

Харлэн Карви

**Криминалистическое исследование
Windows**



Анализ файлов

Содержание этой главы:

- § Файлы журналов
- § Метаданные файлов
- § Альтернативные потоки данных
- § Альтернативные методы анализа

- Ü Краткое изложение
- Ü Быстрое повторение
- Ü Часто задаваемые вопросы

Введение

Операционные системы Windows содержат ряд файлов, имеющих ценность с судебной точки зрения. В действительности многие эксперты не представляют себе, какое богатство данных можно найти в некоторых файлах, которые ОС Windows использует для отслеживания различных действий и функций. Знания о многочисленных местах, где хранятся данные на компьютере, позволяют эксперту подтвердить информацию, найденную в других источниках, и уменьшают величину неопределенности во время анализа. В этой главе будут проанализированы различные файлы, в том числе, файлы журналов, которые можно найти в ОС Windows, общие сведения о файлах, а также другие файлы, представляющие интерес для эксперта. Мы рассмотрим ряд очевидно разных элементов, которые связаны между собой тем, что все они находятся в файлах или файловой системе, независимо от того, встречаются они в удобочитаемом формате ASCII или трудном для понимания двоичном формате.

Файлы журналов

Операционные системы Windows ведут журналы для ряда событий и действий, которые могут представлять интерес для эксперта. Помимо файлов журналов приложений, в которых регистрируются события, касающиеся отдельных приложений, операционные системы Windows также содержат несколько других журналов. В этой главе мы рассмотрим файлы журналов, наиболее актуальные для анализа данных, среди которых, вероятно, самый известный – журнал событий Windows.

Журналы регистрации событий

Журналы регистрации событий – возможно, самые распространенные журналы в ОС Windows, которым приблизительно соответствует демон syslog в Linux. Такие журналы регистрируют ежедневные события, происходящие в Windows, и их можно настроить так (подробнее – в главе 4), чтобы они записывали ряд других дополнительных событий. Эти события поделены на категории, которые реализованы посредством различных журналов событий, таких как журналы событий безопасности, системы и приложений. Эти журналы событий могут предоставить большое количество информации, полезной как для поиска и устранения неисправностей, так и для понимания событий во время судебного анализа.

Совет

В большинстве ОС Windows можно использовать инструмент «auditpol.exe» (из пакета Resource Kit) для просмотра и настройки параметров политики аудита. В ОС Windows XP с пакетом обновления 2 (SP2) и 2003 с пакетом обновления 1 (SP1) инструмент «auditusr.exe» позволяет настраивать политики аудита для каждого пользователя. Например, можно установить аудит входа в систему для всех пользователей, но включить более детальный аудит для отдельного пользователя. В результате применения инструмента «auditusr.exe» изменяется раздел реестра HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Audit\PerUserAuditing\System. Использование этого инструмента может дать эксперту представление о типах событий, которые он должен ожидать увидеть в журнале событий, а также свидетельствовать об уровне технических навыков пользователя или администратора.

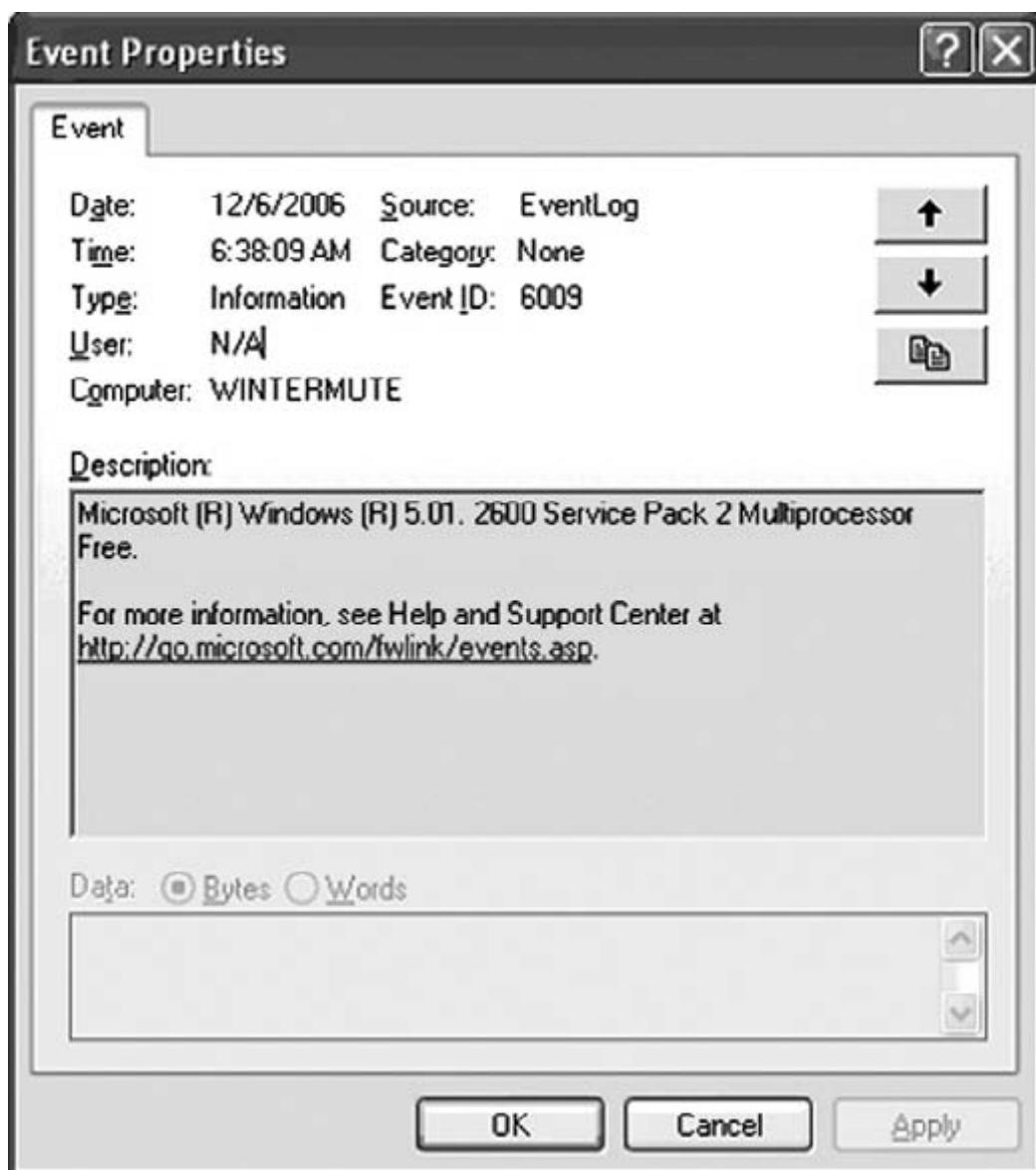
Основные понятия о событиях

В семействе операционных систем Windows NT, от Windows 2000 до XP и 2003, журналы событий имеют двоичную структуру; в файле журнала хранится заголовок и ряд записей о событиях. Как правило, когда в операционной системе происходят определенные события, например, вход в систему или выход из системы, создается запись об этих событиях. Одни события регистрируются по умолчанию, другие – в зависимости от конфигурации аудита, хранящейся в разделе реестра PolAdtEv, как было указано в главе 4. Прочие аспекты конфигурации для журнала регистрации событий (размер файла, период хранения записей и т. д.) содержатся в следующем разделе реестра:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\<Event Log>
```

По умолчанию в операционных системах Windows 2000, XP и 2003 есть журналы событий приложений, безопасности и системы. В компьютерах, сконфигурированных как контроллеры домена, также есть журналы событий службы репликации файлов и службы каталогов, а в компьютерах, сконфигурированных как серверы системы доменных имен (DNS), – журналы событий DNS. Кроме того, в других системах могут быть файлы журналов событий для отдельных приложений.

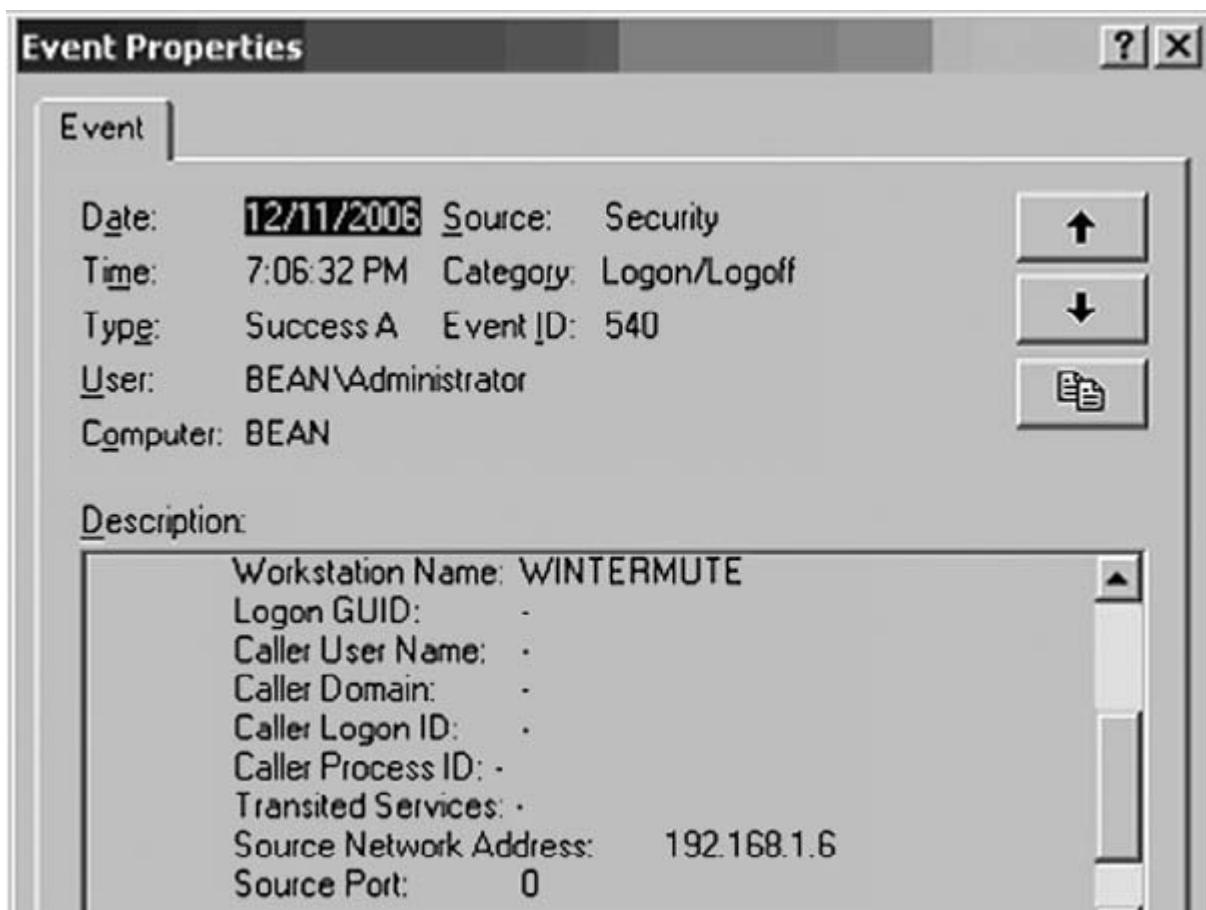
Администраторы обычно взаимодействуют с журналами событий посредством программы «Просмотр событий» (“Event Viewer”), имеющей графический интерфейс пользователя. Когда администратор просматривает запись о событии в ОС Windows XP, то видит окно, похожее на то, что показано на илл. 5.1.



Илл. 5.1. Запись о событии Windows XP в программе «Просмотр событий».

Когда программа «Просмотр событий» (“Event Viewer”) открывает запись о событии, она заполняет данными поле «Описание:» (“Description:"), считывая строковые параметры из записи о событии, а затем определяет местонахождение соответствующего файла сообщений (библиотека динамической компоновки (DLL-файл)) в системе. Файлы сообщений содержат строки сообщений, используемые для поддержки интернационализации в операционных системах Windows, а строковые параметры из записей о событиях вставляются в соответствующих местах в этих строках. Это делает возможным интернационализацию журналов событий, предоставляя строки сообщений о событиях на языке, присущем данной системе (английском, немецком, французском или другом) и просто «заполняя пробелы» необходимой информацией (имя системы, отметки даты и времени и т. д.). Таким образом, видна тесная взаимосвязь между журналом регистрации событий, системным реестром и многими DLL-файлами в ОС Windows. Это также означает, что приложения сторонних разработчиков, выполняющие записи в журнал событий, должны иметь в своем составе собственные файлы сообщений.

До ОС Windows 2003 события входа обычно содержали только NetBIOS-имя компьютера, из которого был выполнен вход. Начиная с Windows 2003, журнал событий безопасности регистрирует как имя, так и IP-адрес компьютера, что показано на илл. 5.2.



Илл. 5.2. Запись о событии ОС Windows 2003 с указанием IP-адреса.

Информация в записи о событии (см. илл. 5.2) может быть чрезвычайно полезной во время расследования, особенно это касается строки «Адрес сети источника» (“Source Network Address”), так как в ней показан IP-адрес удаленного компьютера. Эти данные помогают определить, откуда выполнялись операции входа и попытки входа.

Даже без DLL-файла сообщений не трудно сказать, к чему относятся различные записи о событиях, так как в записи есть и другие идентифицирующие сведения. На илл. 5.2, например, мы видим идентификатор события, источник события и другую информацию, которую можно использовать во время анализа записей о событиях. Здесь также есть отметка даты и времени, которую можно использовать для анализа временной шкалы; на самом деле в записи о событии есть две отметки даты и времени (эта тема будет рассмотрена позднее в этой главе). Microsoft предоставляет много информации относительно некоторых интересующих нас записей о событиях. Например, если включены аудит и регистрация событий входа и выхода (в главе 4 объясняется, как определить это в образе данных), эксперт должен увидеть идентификаторы 528 (успешный вход) и 538 (выход) в журнале событий безопасности. Если он видит несколько записей о событиях с идентификатором 528, ему нужно будет проверить тип входа, потому что существует девять кодов для разных типов входа. В таблице № 5.1 показаны девять кодов для успешных операций входа и дано их описание.

Таблица № 5.1

Типы входа

| Тип входа | Название типа входа | Описание |
|-----------|---------------------|---|
| 2 | <i>Interactive</i> | Этот тип входа означает, что пользователь вошел в консоль |
| 3 | <i>Network</i> | Пользователь или компьютер выполнил вход на данный компьютер из сети, например, с помощью команды <i>net use</i> , получив доступ к сетевому ресурсу, или успешной команды <i>net</i> |

view, направленной на сетевой ресурс. (Этот тип входа был заменен идентификатором события 540.)

| | | |
|----|--------------------------------|---|
| 4 | <i>Batch</i> | Зарезервирован для приложений, выполняющихся в пакетном режиме |
| 5 | <i>Service</i> | Вход службы |
| 6 | <i>Proxy</i> | Не поддерживается |
| 7 | <i>Unlock</i> | Пользователь разблокировал рабочую станцию |
| 8 | <i>NetworkClearText</i> | Пользователь вошел в сеть, а учетные данные пользователя были переданы в незашифрованном виде |
| 9 | <i>NewCredentials</i> | Процесс или поток клонировал свой текущий маркер, но указал новые учетные данные для исходящих соединений |
| 10 | <i>RemoteInteractive</i> | Вход с использованием службы терминалов или подключения к удаленному рабочему столу |
| 11 | <i>CachedInteractive</i> | Пользователь выполнил вход на этот компьютер с учетными данными, хранящимися локально на компьютере (возможно, контролер домена был недоступен для проверки учетных данных) |
| 12 | <i>CachedRemoteInteractive</i> | То же, что <i>RemoteInteractive</i> ; используется для внутренних целей аудита |
| 13 | <i>CachedUnlock</i> | Во время входа выполнена попытка разблокировать рабочую станцию |

В статьях № 299475 (<http://support.microsoft.com/kb/299475>) и № 301677 (<http://support.microsoft.com/kb/301677>) из базы знаний Microsoft описываются события безопасности ОС Windows 2000. Для каждого события безопасности дано краткое описание, а также указаны заполнители (%1, %2 и т. д.), вместо которых вставляются строки из записи о событии.

Совет

Идентификатор события 540 (сетевой вход в систему) впервые появился в Windows 2000. Он означает то же, что идентификатор события 528 типа 3 (успешный вход в систему из сети) и заменяет этот тип входа. Специалисты по расследованию инцидентов, работающие в среде предприятия, могут встретить записи о событиях, в которых упоминается протокол Kerberos, например, событие с идентификатором 672. В статье №301677 (<http://support.microsoft.com/kb/301677>) из базы знаний Microsoft предоставляется информация об этих идентификаторах событий для ОС Windows 2000, а в статье № 274176 (<http://support.microsoft.com/kb/274176>) описано, как связать событие входа в систему для учетной записи с событием создания процесса, например, когда служба запускается с использованием учетной записи пользователя в ОС Windows XP. Записи о событии с идентификатором 672 содержат IP-адрес, который можно использовать во время анализа данных.

В ОС Windows Vista и 2008 события входа стали обозначаться одним идентификатором (4624). В статье № 947226 (<http://support.microsoft.com/kb/947226>) из базы знаний Microsoft предоставлен список идентификаторов событий для Windows Vista и Windows 2008, а также описан способ получения более подробной информации о событиях с помощью инструмента «wevtutil.exe».

Что касается других записей о событиях, существует много сайтов, предоставляющих о них подробные сведения. Полную информацию об отдельных записях

в журнале событий приложений можно получить, обратившись к производителю. По моему мнению, один из лучших сайтов, который помогает получить представление о данных, содержащихся в записях о событиях, – это EventID.net. Часть информации доступна на сайте EventID.net без подписки, но если вы уделяете много времени анализу различных записей о событиях, то плата за подписку будет стоить дополнительной информации и времени, сэкономленного на поиске в Интернете. Во многих случаях, вам нужно будет просто ввести исследуемый идентификатор о событии, чтобы получить сведения о событии, сгенерированные различными источниками, а также ссылки на справочную документацию. Например, если я выполню поиск по событию с идентификатором 6009, то получу четыре разных источника события. Затем я могу щелкнуть по данным для источника, который мне нужен (в данном случае – *EventLog*), и получить комментарии от двух авторов, а также три ссылки на страницы сайта Microsoft, где предоставлена подробная информация об этом идентификаторе события. В данном примере я достаточно быстро узнал, что этот идентификатор события создается при начальной загрузке Windows (значит, время создания записи о событии приблизительно равно времени загрузки системы), и что информация о версии операционной системы записывается в поле «*Описание:*» (“*Description:*”) события.

Совет

Запись о событии с идентификатором 6009 из источника *EventLog* можно использовать, чтобы определить операционную систему главного компьютера и имя компьютера. Запись «*Компьютер:*» (“*Computer:*”) будет содержать имя компьютера, а поле «*Описание:*» (“*Description:*”) в записи о событии будет содержать строку, в которой указана версия операционной системы Windows.

Помимо сайта EventID.net, отличным источником информации о журналах событий Windows является блог Эрика Фицджеральда (Eric Fitzgerald) «Windows Security Logging and Other Esoterica» (<http://blogs.msdn.com/ericfitz/default.aspx>). Блог содержит много полезных сведений о журналах регистрации событий, в том числе о том, как их можно использовать, чтобы отвечать стандартам защиты информации PCI (Payment Card Industry) платежной системы Visa, а также советы и рекомендации по аудиту. В блоге Эрика я нашел множество информации, в том числе описание входа типа 0, а также способы получения подробных данных о событиях безопасности в ОС Windows Vista и Windows 2008. Microsoft также предоставляет расширенный поиск по сайту «Центр сообщений об ошибках и событиях» (www.microsoft.com/technet/support/ee/ee_advanced.aspx), который можно использовать для получения сведений о различных записях журнала событий.

Совет

Помните, как мы обсуждали артефакты съемных USB-накопителей в главе 4? Когда съемный USB-накопитель подключался к компьютеру с ОС Windows 2000, служба съемных носителей создавала запись о событии с идентификатором 134. Когда накопитель отключался, она создавала идентификатор события 135. Эти события больше не отображаются в ОС Windows XP, а в статье № 329463 (<http://support.microsoft.com/kb/329463/en-us>) из базы знаний Microsoft объясняется, почему это происходит. В статье говорится, что после установки исправления «... Netshell больше не ожидает уведомлений о появлении самонастраивающихся (Plug and Play) устройств. Следовательно, вы не будете уведомлены о новых устройствах».

Итак, не следует ожидать увидеть в журнале событий записи о подключении или извлечении съемных запоминающих устройств USB.

Кроме того, вы, возможно, захотите посетить сайт Рэнди Франклина (Randy Franklin) UltimateWindowsSecurity.com; там вы найдете страницы, посвященные

специальному журналу событий безопасности Windows, в том числе справочную таблицу идентификаторов событий и энциклопедию (www.ultimatewindowssecurity.com/encyclopedia.aspx). Если вам нужна информация о журнале событий безопасности, стоит обязательно посетить этот сайт и добавить его в закладки.

Формат файла журнала событий

Иногда во время анализа данных необходимо исследовать файл журнала событий (с расширением .evt) в понятном формате. (Формат журнала событий, обсуждающийся в этом разделе, относится к версиям операционной системы Windows от Windows 2000 до 2003 и не используется в ОС Windows Vista.) Предположим, что после извлечения evt-файла из образа данных вы решили открыть его в программе «Просмотр событий». Или, возможно, вы захотели использовать программу Event Log Explorer (www.eventlogxp.com), а не стандартную программу просмотра событий. Однако, пытаясь сделать это, вы получите сообщение об ошибке, где говорится, что журнал событий поврежден. Кроме того, знание подробностей о структуре файла журнала событий поможет вам найти полезную и ценную информацию во время просмотра свободных кластеров в образе данных.

Журнал регистрации событий Windows (для ОС Windows 2000, XP и 2003) имеет двоичный формат с четкими признаками, которые помогут эксперту распознать и интерпретировать файлы журнала событий или просто записи о событиях из файлов или свободного пространства. Каждый журнал событий состоит из раздела заголовка и ряда записей о событиях, оба этих элемента будут подробно рассмотрены в этой главе. Журнал событий сохраняет информацию в виде кольцевого буфера, поэтому, когда новые записи о событиях добавляются в файл, более старые записи циклически удаляются из файла.

Заголовок журнала событий

Заголовок журнала событий содержится в первых 48 байтах действительного файла журнала. Если evt-файл не был каким-либо образом поврежден, его заголовок будет похож на заголовок примерного файла журнала, показанный на илл. 5.3.

| | |
|------------|--|
| 00000000h: | 30 00 00 00 4C 66 4C 65 01 00 00 00 01 00 00 00 ; 0...LjLe..... |
| 00000010h: | 30 00 00 00 F0 A9 00 00 AD 00 00 00 01 00 00 00 ; 0...@...-..... |
| 00000020h: | 00 00 01 00 09 00 00 00 80 3A 09 00 30 00 00 00 ;€:..0... |

Илл. 5.3. Заголовок файла журнала.

Заголовок файла журнала состоит из 12 отдельных значений двойного слова. В таблице № 5.2 перечислены девять из этих значений и предоставлено описание для каждого из них.

Таблица № 5.2

Структура заголовка журнала событий

| Смещение | Размер | Описание |
|----------|---------|--|
| 0 | 4 байта | Размер записи; для заголовка evt-файла размер равен 0x30 (48) байт. Размер записи о событии – 56 байт. |
| 4 | 4 байта | Магическое число, или сигнатура (<i>LjLe</i>) |
| 16 | 4 байта | Смещение самой старой записи о событии в evt-файле |
| 20 | 4 байта | Смещение следующей записи о событии (которая будет сохранена) в evt-файле |
| 24 | 4 байта | Идентификатор следующей записи о событии |
| 28 | 4 байта | Идентификатор самой старой записи о событии |
| 32 | 4 байта | Максимальный размер evt-файла (из реестра) |
| 40 | 4 байта | Время хранения записей о событиях (из реестра) |
| 44 | 4 байта | Размер записи (повторение двойного слова в смещении 0) |

Важное значение в заголовке – так называемое магическое число, которое отображается как *LfLe*, начиная с четвертого байта (второго двойного слова) в заголовке. Это значение уникально в журнале событий Windows (для ОС Windows 2000, XP и 2003) и связано с записями о событиях. Microsoft называет это значение как *ELF_LOG_SIGNATURE*. (В описании структуры записи о событии на сайте Microsoft сказано, что это «значение двойного слова, всегда установленное в *ELF_LOG_SIGNATURE*».) Обратите внимание, что размер записи (для заголовка – 0x30, или 48 байт) присутствует как в начале, так и в конце записи заголовка. Это позволяет эксперту программным способом (используя код) или вручную (с помощью шестнадцатеричного редактора) определить местонахождение заголовка во время просмотра файла журнала событий, свободного пространства накопителя или файла неизвестного типа. Номера идентификаторов следующей записи о событии (которая будет сохранена) и самой старой записи о событии можно использовать, чтобы определить общее число записей о событиях, которое должен увидеть эксперт.

Примечание

Во время работы с файлами мы используем термин *магическое число*, когда говорим об отдельной последовательности байтов в файле, уникальной для этого файла или типа файла. Эти магические числа используются при проведении анализа сигнатур файлов – метод, который применяется, чтобы определить, имеет ли файл правильное расширение, основываясь на его магическом числе (сигнатуре). В случае с файлами журналов событий магическое число равно 0x654c664c, или 4C 66 4C 65, как показано на илл. 5.3. Несмотря на то, что эта последовательность байтов преобразовывается в строку *LfLe* при изменении порядка следования байтов, она все равно называется магическим числом.

Значения для максимального размера файла журнала событий и времени хранения записей о событиях берутся из реестра системы, в которой хранятся журналы регистрации событий.

Структура записи о событии

В записях о событиях есть несколько значений структуры, похожих на значения заголовка журнала событий, но записи о событиях содержат намного больше информации, что показано на илл. 5.4. Заголовок для записи о событии несколько больше заголовка самого журнала событий (как было описано выше) и имеет размер 56 байт. Несмотря на то, что размер записи, предоставленный в записи о событии (0xF4, или 244 байта), больше, чем 56 байт, первые 56 байт записи составляют заголовок записи о событии.

| | | |
|------------|--|------------------|
| 00000030h: | F4 00 00 00 4C 66 4C 65 01 00 00 00 00 3D E1 20 43 ; | δ...LfLe....=á C |
| 00000040h: | 3D E1 20 43 64 02 00 00 08 00 15 00 06 00 00 00 ; | =á Cd..... |
| 00000050h: | 00 00 00 00 72 00 00 00 1C 00 00 00 56 00 00 00 ; |r.....v... |
| 00000060h: | 00 00 00 00 EE 00 00 00 53 00 65 00 63 00 75 00 ; |i...S.e.c.u. |

Илл. 5.4. Примерная структура записи о событии.

Как видите, магическое число журнала событий присутствует во втором значении двойного слова в записи о событии, как и в случае с заголовком. В таблице № 5.3 представлены подробности о содержимом первых 56 байт записи о событии.

Таблица № 5.3

Структура записи о событии

| Смешение | Размер | Описание |
|-----------------|---------------|--|
| 0 | 4 байта | Длина записи о событии, или размер записи в байтах |
| 4 | 4 байта | Зарезервировано; магическое число |
| 8 | 4 байта | Номер записи |

| | | |
|----|---------|---|
| 12 | 4 байта | Время создания записи; измеряется в формате времени UNIX (число секунд, прошедших с начала 1 января 1970 года в формате всемирного координированного времени (Universal Coordinated Time, далее – UTC)) |
| 16 | 4 байта | Время сохранения записи; измеряется в формате времени UNIX (число секунд, прошедших с начала 1 января 1970 года в формате UTC) |
| 20 | 4 байта | Идентификатор события, который относится к источнику события и однозначно определяет событие; идентификатор события используется вместе с именем источника, чтобы найти соответствующую строку описания в файле сообщений для источника события |
| 24 | 2 байта | Тип события (0x01 = ошибка; 0x10 = неудача; 0x08 = успех; 0x04 = информация; 0x02 = предупреждение) |
| 26 | 2 байта | Число строк |
| 28 | 2 байта | Категория события |
| 30 | 2 байта | Зарезервированные флаги |
| 32 | 4 байта | Номер заключительной записи |
| 36 | 4 байта | Смещение строк; смещение строк с описанием в записи о событии |
| 40 | 4 байта | Размер идентификатора безопасности (SID) пользователя; размер SID пользователя в байтах (если размер равен 0, SID пользователя не предоставлен) |
| 44 | 4 байта | Смещение SID пользователя в этой записи о событии |
| 48 | 4 байта | Длина данных; длина двоичных данных, связанных с этой записью о событии |
| 52 | 4 байта | Смещение данных |

В таблице № 5.3 описаны первые 56 байт записи о событии. Не забывайте, что фактическая длина самой записи указана в первом и последнем двойных словах записи. (Значение размера записи, как и в заголовке файла, присутствует в начале и конце фактической записи.) Имея эту информацию, можно относительно легко анализировать содержимое файлов журнала событий и извлекать записи о событиях.

Знание структуры записи о событии позволяет заново собрать неполные записи о событиях, найденные в свободном пространстве накопителя. Используя магическое число в качестве указателя, эксперт может выполнить поиск по всему свободному пространству. В случае если он найдет магическое число, все, что ему нужно будет сделать, – интерпретировать предыдущее двойное слово, чтобы узнать размер записи о событии, а затем извлечь это число байтов, чтобы получить полную запись о событии. Даже если целая запись о событии недоступна, первые 56 байт предоставляют план действий для восстановления частей записи о событии.

Инструменты и ловушки...

Чтение журналов событий

Однажды я помогал одному эксперту, который был хорошо знаком с ОС Linux и использовал инструмент PyFlag (www.pyflag.net) для судебного анализа данных. Он попросил меня открыть журналы регистрации событий и извлечь имеющиеся записи; он пытался сделать это сам, но, скопировав evt-файлы на компьютер с ОС Windows и попытавшись открыть их с помощью программы «Просмотр событий», он получил сообщение, что файлы повреждены.

Мне уже приходилось исследовать структуру журнала событий и записей о событиях, поэтому я выполнил определенные настройки в своем Perl-скрипте и начал обрабатывать файлы журналов событий, без проблем извлекая все записи о событиях. Однако я обнаружил несоответствие между информацией, полученной из заголовка одного из журналов событий, и тем, что я видел в выходных данных записей о событиях; какой бы способ я не использовал, я всегда получал на одну полную запись больше, чем было указано в информации заголовка. Изучив эту проблему, я установил, что, согласно

интерфейсу прикладного программирования (API), часть журнала событий, предшествующая первой записи, была буферной областью, оставшейся после того, как журнал событий был очищен. Эта область не считывалась интерфейсом API, и, если бы система продолжала работать в обычном режиме, буферная область бы была удалена из кольцевого буфера, когда новые записи о событиях были бы сохранены в файле. И все же этот буфер содержал одну полную запись о событии; так как инструмент, с которым я работал, не использовал интерфейс API для извлечения записей о событиях, а вместо этого считывал файл в двоичном режиме, то во время анализа найденной информации он не распознавал эту буферную область.

Хотя «потерянная» запись о событии не имела большого значения для дела, этот случай показал, насколько полезно (в отношении судебного анализа) понимать формат определенных файлов в ОС Windows, и по возможности разрабатывать инструменты, анализирующие информацию в этих файлах таким способом, который не зависит от интерфейса Windows API. Это не только предоставит эксперту возможность находить «скрытую» информацию, но и позволит ему проводить анализ на платформах, отличных от ОС Windows (в частности в ОС Linux); экспертам не запрещается анализировать образы данных, полученные из ОС Windows, на других платформах.

В каталоге ch5\code\EVT на DVD-диске, идущем в комплекте с этой книгой, содержится несколько Perl-скриптов, которые позволяют вам собрать информацию из файлов журналов событий в ОС Windows 2000, XP и 2003. Скрипт «evtstats.pl» показывает простые статистические данные, полученные из evt-файла:

```
C:\Perl\forensics\evt2xls>evtstats.pl d:\cases\evt\secevent.evt
Max Size of the Event Log file = 65536 bytes
Actual Size of the Event Log file = 65536 bytes
Total number of event records (header info) = 172
Total number of event records (actual count) = 260
Total number of event records (rec_nums) = 260
Total number of event records (sources) = 260
Total number of event records (types) = 260
Total number of event records (IDs) = 260
```

Скрипт анализирует заголовок файла журнала событий и определяет количество записей, которые должны присутствовать в файле, затем обрабатывает содержимое самого журнала событий и, используя различные теги в каждой записи о событии, выполняет фактический подсчет количества записей, находящихся в файле журнала событий.

Совет

Для того чтобы работать на своем компьютере со скриптами из каталога ch5\code\EVT, просто скопируйте файлы каталога на свой компьютер, используемый для анализа данных. Если вы собираетесь использовать Perl-скрипты, не забудьте установить интерпретатор Perl, а также поместить Perl-модуль «ReadEvt.pm» в тот же каталог, где находятся скрипты. Или же можно использовать исполняемые файлы, но при этом необходимо поместить DLL-файл в тот же каталог, где находятся EXE-файлы.

Perl-скрипт «evtrpt.pl» показывает дополнительные статистические данные о файле журнала событий:

```
C:\Perl\forensics\evt2xls>evtrpt.pl d:\cases\evt\secevent.evt
EVT file parsed: d:\cases\evt\secevent.evt (65536 bytes)
Total number of event records counted: 260
-----
Event Source/ID Frequency
Source                                Event ID
```

```
-----
Security           513          4
Security           514         28
Security           515         34
Security           518          4
Security           520          3
Security           528,2        7
Security           528,5        35
Security           529,2        7
Security           538,2        5
Security           538,3        8
Security           540,3       12
Security           551          7
Security           576         42
Security           612          5
Security           615          5
Security           680         14
Security           806          4
Security           848          4
Security           849          4
Security           850         28
Total: 260
-----
Event Type Frequency
Type              Count
-----
AUDIT_SUCCESS     245
AUDIT_FAILURE    15
Total: 260
-----
Date Range (UTC)
Fri Sep 9 01:11:25 2005 to Tue Sep 27 00:38:58 2005
```

Я довольно часто использую скрипт «evtrpt.pl» при проведении анализа журналов событий. Я обычно начинаю с анализа файла куста Security на предмет политики аудита (см. главу 4), чтобы узнать какие типы событий можно встретить в журнале событий, а также определяю, включен ли аудит. Затем, используя «evtrpt.pl», я анализирую файл журнала событий (извлеченный из образа данных), чтобы определить частотность различных идентификаторов и источников событий в файле журнала и установить диапазон дат событий в журнале. Значение времени, которое собирает «evtrpt.pl», – это время создания события (а не время записи информации о событии), что позволяет мне узнать, есть ли в файле журнала записи, находящиеся в пределах временного интервала для данного инцидента. Этот способ позволяет получить много полезной информации о журналах событий, особенно если журнал событий приложения содержит записи о событиях, созданных антивирусной программой.

Еще один Perl-скрипт, «lsevt.pl», использует Perl-модуль «ReadEvt.pm», чтобы обработать файл журнала событий и отобразить записи о событиях, в виде списка, как показано ниже:

| | | |
|----------------|---|--|
| Record Number | : | 251 |
| Source | : | Security |
| Computer Name | : | PETER |
| Event ID | : | 528 |
| Event Type | : | EVENTLOG_AUDIT_SUCCESS |
| Time Generated | : | Mon Sep 26 23:37:51 2005 |
| Time Written | : | Mon Sep 26 23:37:51 2005 |
| SID | : | S-1-5-21-839522115-1801674531-2147200963-1003 |
| Message Str | : | Harlan PETER (0x0,0x141B9C) 2 User32 Negotiate PETER {00000000-0000-0000-0000-000000000000} |
| Record Number | : | 252 |

```

Source : Security
Computer Name : PETER
Event ID : 576
Event Type : EVENTLOG_AUDIT_SUCCESS
Time Generated : Mon Sep 26 23:37:51 2005
Time Written : Mon Sep 26 23:37:51 2005
SID : S-1-5-21-839522115-1801674531-2147200963-1003
Message Str : (0x0,0x141B9C) SeChangeNotifyPrivilege
                SeBackupPrivilege
                SeRestorePrivilege
                SeDebugPrivilege

```

Скрипт «lsevt.pl» также анализирует идентификатор SID пользователя и по возможности преобразовывает его в доступную для чтения информацию, которую можно сопоставить с другими данными (например, из реестра) во время анализа.

Скрипт «lsevt2.pl» предоставляет большую гибкость, чем «lsevt.pl», потому что он позволяет сохранять выходные данные в формате значений, разделенных запятыми (.csv). Таким образом, эксперт может запустить этот скрипт для анализа файла журнала событий Windows, используя следующую команду:

```
C:\Perl>lsevt2.pl -f d:\cases\appevent.evt -c > testevt.csv
```

Затем можно открыть полученный файл «testevt.csv» в программе Excel и выполнить сортировку, поиск и анализ данных. Более того, «lsevt2.pl» – это автономный скрипт, который не требует использования Perl-модуля «ReadEvt.pm». Однако скрипт «lsevt2.pl» не переводит SID пользователя в распознаваемый формат.

«Evt2xls.pl» – это Perl-скрипт, который считывает файл журнала событий, извлекая все записи о событиях, анализирует и записывает их в формат таблицы, совместимый на двоичном уровне с Microsoft Excel. Это позволяет открывать таблицу и сортировать различные поля данных по таким критериям, как источник события (например, чтобы показать все записи о событиях открытия всплывающего окна приложения) или идентификатор события. Для того чтобы использовать «evt2xls.pl», необходимо указать несколько параметров в командной строке, например:

```
C:\perl>evt2xls.pl -e d:\cases\evt\secevent.evt -o d:\cases\secevent.xls
```

В предыдущей команде переключатель *-e* используется, чтобы указать файл журнала событий, который нужно обработать, а переключатель *-o* – чтобы указать имя и местонахождение выходного файла электронной таблицы. Если в командной строке просто ввести **evt2xls.pl** без переключателей, то на экране будет показана информация об использовании синтаксиса скрипта. Например, переключатель *-r* позволяет указать местонахождение файла отчета, похожего на тот, что создается скриптом «evtrpt.pl». Кроме того, с помощью переключателя *-x* можно перечислить через запятую идентификаторы событий, которые нужно исключить из итоговой таблицы. Первоначально эта опция предназначалась для больших журналов событий, содержащих более 65 535 записей, так как в некоторых версиях программы Microsoft Excel есть ограничение на такое количество строк в листе. Однако в программе Excel 2007, очевидно, нет этого ограничения, а другие приложения, например, редактор электронных таблиц из пакета OpenOffice (www.openoffice.org/), должны без проблем открыть этот файл. Более того, скрипт «evt2xls.pl» использует Perl-модуль Spreadsheet::WriteExcel для создания таблицы, а модуль Spreadsheet::Read или Spreadsheet::ParseExcel можно использовать, чтобы извлечь данные из выходного файла таблицы.

Совет

Роб Фабер (Rob Faber) написал отличную статью «Windows log forensics: did you

cover your tracks?» для апрельского издания журнала *INSECURE* 2008 года (номер 16, доступен по адресу www.net-security.org/dl/insecure/INSECURE-Mag-16.pdf). В статье Роб предоставляет исключительную информацию, которую можно использовать в разнообразных экспертизах. Стоит обязательно распечатать если не весь номер журнала *INSECURE*, то хотя бы статью Роба Фабера.

Все эти Perl-скрипты анализируют файлы журналов событий в двоичном режиме, полностью обходя интерфейс Windows API. Таким образом можно не только обрабатывать файлы журналов событий на платформе, отличной от Windows (например, в Mac OS X, Linux и т. д.), но и анализировать журналы событий, даже если программа «Просмотр событий» выдает сообщение, что файл каким-либо образом поврежден. Некоторые скрипты требуют наличия Perl-модуля «ReadEvt.pm», включенного в состав DVD-диска, который идет в комплекте с этой книгой.

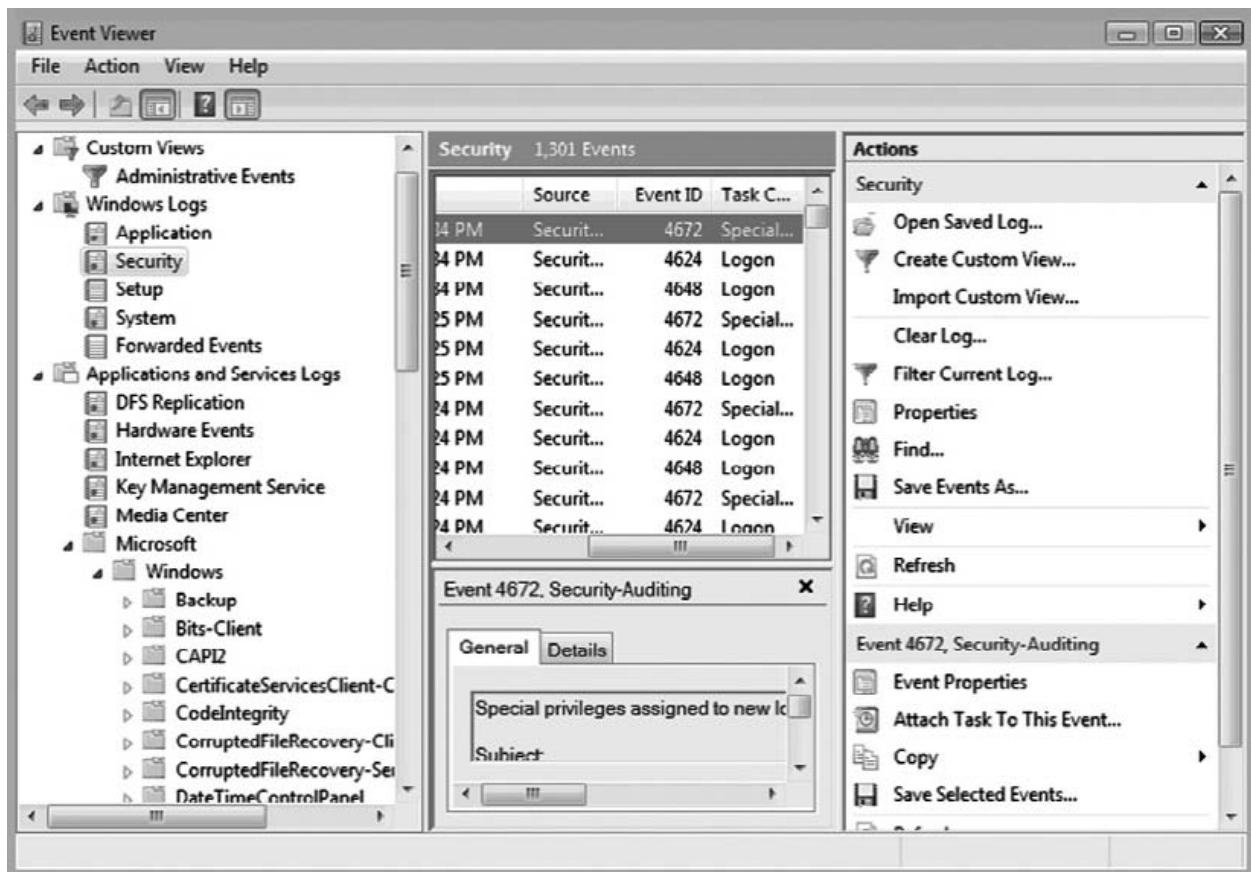
Предупреждение

В феврале 2007 года Andreas Schuster (Andreas Schuster) написал в своем блоге об особом условии, касающемся записей журнала событий, когда запись сохраняется в конце evt-файла, но переносится в начало файла так, что часть записи следует после заголовка. Эта запись будет неправильно интерпретирована инструментами (например, Perl-скриптами, упоминавшимися в этой главе), которые ищут магическое число записи, потому что будет распознана только часть записи. Andreas любезно предоставил пример evt-файла, чтобы программы анализа можно было проверить на это условие и соответствующим образом улучшить (запись в блоге и тестовый файл можно найти по адресу http://computer.forensikblog.de/en/2007/02/a_common_misconception.html).

Журналы событий Vista

С появлением Windows Vista в операционной системе произошло много изменений, в том числе в структуре журнала событий. Например, в английской версии операционной системы это служба теперь называется Windows Event Log, а не Event Logging. Кроме того, ОС Windows Vista использует новый XML-формат для хранения записей о событиях и поддерживает центральный сбор записей о событиях.

К другим изменениям относится то, что, хотя Vista все еще содержит три основные категории журнала событий (приложение, безопасность и система), она теперь имеет широкий выбор категорий, в которых можно регистрировать разные события, см. илл. 5.5.



Илл. 5.5. Программа «Просмотр событий» (“Event Viewer”) в Vista.

Как показано на илл. 5.5, теперь существует больше журналов, в том числе для событий Internet Explorer и событий оборудования. (После установки браузера Internet Explorer версии 7 также добавляется журнал событий Internet Explorer в ОС Windows XP и 2003.) Несмотря на то, что контейнер журнала создается, журнал не включен и, похоже, используется для тестирования совместимости приложений (<http://msdn.microsoft.com/en-us/library/bb250493.aspx>); по этой причине он не имеет большого значения с точки зрения судебного анализа.

Также обратите внимание, что внизу справа на илл. 5.5 есть элемент, который называется «Привязать задачу к событию...» (“Attach Task To This Event...”). Так как инструменты для анализа журналов событий Vista разрабатываются, а специалисты по расследованию инцидентов и судебные эксперты используют эти инструменты, этот элемент будет представлять для них интерес.

Андреас Шустер и Эрик Фицджеральд опубликовали в своих блогах информацию о структурах, используемых для хранения записей о событиях. Кроме того, в блоге Андреаса есть статья, в которой описываются некоторые типы данных, имеющиеся в новых журналах событий (http://computer.forensikblog.de/en/2007/08/evtx_data_types.html), а также статья об анализаторе, преобразовывающем журналы событий Vista в обычный текст (http://computer.forensikblog.de/en/2007/08/evtx_parser.html).

На работающем компьютере с ОС Windows Vista можно использовать команду «wevtutil.exe», чтобы найти информацию о журнале событий Windows, которую не так легко получить через интерфейс программы «Просмотр событий». Например, нижеупомянутая команда покажет список журналов событий, имеющихся в системе:

```
C:\>wevtutil el
```

Затем можно использовать следующую команду, чтобы перечислить сведения о конфигурации для отдельного журнала событий, в том числе имя файла и путь к файлу:

```
C:\>wevtutil gl имя журнала
```

Большая часть информации, показываемая этой командой, также доступна в следующем разделе реестра в ОС Windows Vista:

```
HKEY_LOCAL_MACHINE\System\ControlSet00x\Services\EventLog\log name
```

Эта информация будет полезна как специалистам по расследованию инцидентов, так и судебным экспертам. Необходимо разрабатывать инструменты и методы, позволяющие специалистам по расследованию инцидентов и судебным экспертам извлекать относящуюся к делу информацию из журналов событий в ОС Windows Vista.

Чтобы облегчить анализ данных при исследовании журналов событий из ОС Windows XP и Vista, файлы журнала событий (.evt) можно преобразовать в формат журнала событий Windows (.evtx), используя один из способов, перечисленных в блоге Ask the Performance Team на сайте TechNet корпорации Microsoft (<http://blogs.technet.com/askperf/archive/2007/10/12/windows-vista-and-exported-event-log-files.aspx>).

Журналы IIS

Internet Information Server (IIS) от корпорации Microsoft – это платформа веб-сервера, которая популярна как среди пользователей, так и среди злоумышленников. Ее легко установить – так легко, что администраторы иногда не знают, что на их компьютере работает веб-сервер. Она также, не без основания, считается популярной целью злоумышленников. Часто в веб-сервере появляются уязвимости, связанные с проблемами кода или конфигурации. Если эти проблемы не решать, то злоумышленники могут использовать уязвимости, чтобы получить доступ к программному обеспечению или целевой платформе веб-сервера. Один из лучших способов обнаружить попытки взлома веб-сервера IIS или узнать подробности успешного использования уязвимости – исследовать журналы, созданные веб-сервером.

Журналы веб-сервера IIS чаще всего хранятся в каталоге %WinDir%\System32\LogFiles. Каждый виртуальный сервер имеет свой подкаталог для файлов журналов, имя которого совпадает с именем веб-сервера. В большинстве случаев будет работать только один веб-сервер, поэтому подкаталог для журналов будет называться «W3SVC1». Во время расследования можно найти несколько подкаталогов с именами типа «W3SVCn», где *n* – номер виртуального сервера. Однако администраторы могут изменить стандартное местонахождение журналов и сохранить их в любом месте, даже на общем накопителе в сети. По умолчанию файлы журналов сохраняются в текстовом формате ASCII (также настраивается администраторами), то есть их можно легко открыть и выполнить в них поиск. Во многих случаях файлы журналов могут иметь довольно большой размер, особенно это относится к очень активным веб-сайтам, поэтому будет практически невозможно или неэффективно открыть файл и выполнить в нем поиск вручную. Поиск можно выполнять с использованием Perl-скриптов, утилиты поиска *grep* или, если вы ищите что-то конкретное, с помощью функции поиска, которая есть в любом редакторе.

Кстати, один из самых сложных вопросов, который стоит перед экспертами при проведении поиска, – как найти «иголку» в объемистых журналах веб-сервера? В высокопроизводительных серверах файлы журналов могут иметь довольно большой размер, и найти в них нужные данные будет в высшей степени трудной задачей. Иногда отчет об инциденте помогает сузить период времени, в течение которого произошла атака, и позволяет уменьшить объем обрабатываемых данных. Однако этот способ не всегда работает. Специалисты по расследованию инцидентов довольно часто имеют дело с

компьютерами, которые были взломаны за несколько недель или месяцев до того, как появились отчеты о необычной активности. Так что же делать?

Предупреждение

Во время анализа файлов журналов IIS следует не забывать, что отметки времени для событий будут, по всей вероятности, в формате времени Гринвичу (GMT) (<http://support.microsoft.com/kb/194699>). Когда журналы IIS сохраняются в расширенном формате W3C (по умолчанию), отметки времени регистрируются в формате GMT, а не в формате местного часового пояса для компьютера. В результате новые журналы IIS будут начинаться с момента генерации первой записи после полуночи по Гринвичу (GMT), согласно статье <http://support.microsoft.com/kb/313437>, что также нужно учитывать при проведении анализа.

Кроме того, следует знать о полях, которые могут быть доступны в журналах разных версий IIS. Например, журналы IIS версий 6.0 и 7.0 содержат поле «time-taken» со значением времени, необходимого для обработки запроса (<http://support.microsoft.com/kb/944884>), которое может быть полезным во время анализа журналов

Некоторое время тому назад, в 1997 году, Маркус Ранум (Marcus Ranum) разработал метод, который он назвал «искусственным неведением» (англ. *artificial ignorance*, AI). Основная идея метода заключалась в том, что если удалить из журналов веб-сервера все записи о санкционированных действиях, то оставшиеся записи должны быть связаны с необычными действиями.

Инструменты и ловушки...

Реализация метода «искусственного неведения»

Я применял метод «искусственного неведения» для фильтрации различных элементов, и он оказался очень эффективным. Я написал Perl-скрипт, который устанавливал связь со всеми компьютерами предприятия (я работал в небольшой компании со штатом около 300-400 человек) и собирал содержимое отдельных разделов реестра из компьютеров, которые вошли в домен. Я мог запустить этот скрипт во время обеденного перерыва, а когда возвращался, то получал подробный файл журнала, который можно было легко открыть и проанализировать в Excel. Однако файл был довольно большим, а мне нужно было увидеть только те записи, которые требовали моего внимания. Поэтому я начал исследовать некоторые из полученных записей, и как только подтверждалось, что каждая запись была допустимой, я добавлял ее в файл заведомо правильных записей. Затем я собирал содержимое разделов реестра и регистрировал только те элементы, которые не входили в файл заведомо правильных записей. Через некоторое время я получил не несколько страниц записей, а приблизительно полстраницы элементов, которые нужно было исследовать.

Реализовать этот тип «искусственного неведения» для журналов веб-серверов довольно легко. Например: предположим, что вы исследуете случай предполагаемого взлома веб-сервера, и на этом сервере есть небольшое количество файлов – файл «index.html» и, возможно, дюжина других HTML-файлов, содержащих дополнительную информацию о сайте («about.html», «contact.html», «links.html» и т. д.).

Журналы веб-сервера IIS по умолчанию сохраняются в простом ASCII-формате, поэтому можно легко использовать любимый скриптовый язык, чтобы открыть файл, прочитать каждую запись журнала по отдельности и выполнить её обработку. В журналах IIS, как правило, есть заголовки столбцов в верхней части файлов; эта информация может находиться в другом месте файла, если веб-сервер был перезапущен. Используя заголовки столбцов в качестве ключа, можно проанализировать каждую запись на предмет такой

относящейся к делу информации, как метод запроса (*GET*, *HEAD* или *POST*), запрошенная страница и возвращенный код состояния или ответа (перечень кодов состояния для IIS версий 5.0 и 6.0 доступен по адресу <http://support.microsoft.com/kb/318380>). Если вы обнаружили запрошенную страницу, которой нет в вашем списке заведомо правильных страниц, можно записать имя файла, дату и время запроса, исходный IP-адрес запроса и другую подобную информацию в отдельный файл для анализа.

Предупреждение

Я не предоставляю код для этого метода просто потому, что не все журналы веб-серверов IIS имеют одинаковый формат. Информация, регистрируемая в журналах, настраивается администратором веб-сервера, поэтому я действительно не могу предоставить одно универсальное решение. Кроме того, точные параметры поиска могут отличаться в зависимости от обстоятельств. В одном случае вас могут интересовать все запрошенные страницы, не являющиеся частью веб-сервера, в другом – только запросы, поданные с отдельного IP-адреса или диапазона IP-адресов. Кроме того, вас могут интересовать только запросы, сгенерировавшие отдельные коды ответа.

«Искусственное неведение» – один из способов, который можно использовать при выполнении поиска в журналах веб-серверов; этот способ очень гибок, и его можно реализовывать для различных журналов и файлов. Еще один способ – искать отдельные артефакты, оставшиеся после определенных атак. Этот способ может быть очень полезен в случаях, когда у вас есть дополнительная информация об инфраструктуре, об уровне доступа, полученном злоумышленником, и другие подробности об атаке. Кроме того, если, похоже, что приблизительно во время вторжения появилась информация об определенной уязвимости, или увеличилось число зарегистрированных попыток использования определенного эксплойта, то поиск отдельных артефактов может быть очень эффективным.

Предупреждение

Раньше я удивлялся, как растет популярность некоторых атак, и думал, что это связано с успехом самой атаки. В некоторых случаях доступно большое количество технически подробной информации об атаках. Например, на сайте <http://ferruh.mavituna.com/sql-injection-cheatshee-oku/> доступна документация, посвященная разнообразным атакам с внедрением SQL-кода. Из этой документации вы узнаете, например, что в результате атаки с внедрением SQL-кода в журналах можно найти случаи использования ключевого слова «*sp_password*». Данное ключевое слово обычно используется для смены пароля и указывает системе Microsoft SQL Server не регистрировать команду. Злоумышленники могут использовать его на тот случай, если в SQL Server включено ведение журналов; хотя данные об атаке все равно появятся в журналах веб-сервера, это довольно-таки хитрый трюк.

Например, если веб-сервер IIS использует сервер базы данных Microsoft SQL в качестве фонового сервера, одна из атак, которую следует искать, – внедрение SQL-кода (http://en.wikipedia.org/wiki/SQL_injection). Злоумышленник может использовать запросы, отправленные на веб-сервер для обработки фоновым сервером базы данных, чтобы извлекать информацию, отправлять файлы на сервер или чтобы проникнуть в сетевую инфраструктуру. Верный признак атаки с внедрением SQL-кода – наличие процедуры *xp_cmdshell* в записях файла журнала. *Xp_cmdshell* – расширенная хранимая процедура, являющаяся частью сервера Microsoft SQL, которая позволяет злоумышленнику выполнять команды на сервере базы данных с теми же привилегиями, что есть у самого сервера (обычно это системные привилегии). С середины до конца 2007 года наблюдался ряд таких атак, которые главным образом представляли собой атаки на основе открытых

текстов, так как, как только признаки внедрения SQL-кода обнаруживались в журналах веб-сервера IIS (обычно посредством поиска по ключевому слову *xp_cmdshell*), эксперт мог ясно видеть действия злоумышленника. Во многих случаях злоумышленник проводил разведку в сети, используя инструменты, присущие системе Microsoft SQL, такие как *ipconfig /all*, *nbtstat -c*, *netstat -ano*, разновидности команды *net*, чтобы отобразить другие компьютеры в сети или добавить учетные записи в систему, а также используя «ping.exe» и другие утилиты, чтобы определить возможность сетевого подключения из системы. Затем злоумышленник загружал инструменты на SQL-сервер с помощью утилиты «tftp.exe» или «ftp.exe» (после использования команд *echo*, чтобы создать скриптовый файл FTP). В отдельном случае злоумышленник разбил исполняемый файл на 512-байтовые части, а потом записал каждую часть в таблицу базы данных. После того как все части были загружены в базу данных, злоумышленник указал базе данных (все это выполнялось удаленно через веб-сервер) извлечь эти части, снова собрать их в отдельный файл и запустить его. Удивительно, но это сработало!

Совет

Следует отметить, что во время атаки с внедрением SQL-кода злоумышленник не взламывает веб-сервер и не получает к нему непосредственный доступ. Злоумышленник отправляет специально созданные запросы на веб-сервер, который затем переадресовывает их в базу данных для обработки. При исследовании файлов журналов веб-сервера эксперту также нужно иметь в виду, что код состояния или ответа веб-сервера не указывает, было ли внедрение SQL-кода успешным.

Шли годы, и весной 2008 года в средствах массовой информации появился ряд статей, в которых описывалось использование SQL-кода для нарушения работоспособности веб-серверов путем внедрения вредоносных JavaScript-файлов в страницы веб-сервера. Несмотря на то, что в статьях уделялось особое внимание проблеме, связанной с внедрением SQL-кода, в них ничего не говорилось о, возможно, более опасных атаках, которые приводили к выведению из строя сетевой инфраструктуры потерпевшего. Эксперты стали обнаруживать все более широкомасштабные атаки, а также наблюдалось заметное увеличение сложности применяемых способов атак с внедрением SQL-кода, так как в них больше не использовались ASCII-команды на основе открытых текстов. В результате поиска по ключевым словам не было найдено совпадений для *xp_cmdshell*, даже когда было ясно, что был получен тип доступа, похожий на тот, что можно получить посредством атаки с внедрением SQL-кода. После тщательного изучения было обнаружено, что злоумышленники теперь использовали операторы *DECLARE* и *CAST*, чтобы преобразовать свои команды в шестнадцатеричные строки или в последовательности наборов символов (например, символ «%20» соответствует пробелу). Другие уникальные термины, такие как *nvarchar*, также использовались в операторах при внедрении SQL-кода. Пример того, как выглядит такая запись файла журнала, можно увидеть в статье «The tao of SQL Injection exploits» (<http://dominoyesmaybe.blogspot.com/2008/05/tao-of-sql-injection-exploits.html>). В результате появления новых вариантов атак, необходимо разрабатывать новые методы анализа и обнаружения атак.

По умолчанию веб-сервер IIS сохраняет свои журналы в текстовом формате. Этот формат состоит из ряда полей, последовательность которых отображается в строке *#Fields*: в верхней части файла журнала. Один из способов анализа, который можно использовать для обнаружения атак с внедрением SQL-кода, независимо от кодирования – обработать журналы, извлекая поле *cs-uri-stem*, которое указывает на целевую веб-страницу, например, на «default.asp» или «jobs.asp». Затем для каждого уникального поля *cs-uri-stem* отследите длину полей *cs-uri-query*, которые показывают фактический запрос, введенный для целевой веб-страницы. Команды, используемые для внедрения SQL-кода,

очень часто намного длиннее стандартных запросов, отправляемых на эти веб-страницы во время обычных действий, поэтому вы можете легко отследить записи журнала, представляющие для вас интерес. Поле *cs-uri-stem* также можно использовать, чтобы определить, какие веб-страницы уязвимы для атак с внедрением SQL-кода, опираясь только на содержимое самих журналов.

Можно выполнить поиск ряда других проблем, используя различные ключевые слова или фразы. Например, наличие записи о файле *vti_auth\author.dll* в журналах веб-сервера может свидетельствовать о проблеме (<http://xforce.iss.net/xforce/xfdb/3682>) с правами доступа в серверных расширениях FrontPage, что может привести к искажению внешнего вида веб-страницы. Другие сигнатуры, которые я использовал в прошлом для поиска червя Nimda (www.cert.org/advisories/CA-2001-26.html) (см. раздел «System Footprint» в бюллетене CERT), были связаны с попытками исполнить файлы «cmd.exe» и «tftp.exe» посредством URL-адресов, переданных в веб-браузер.

Полевые заметки...

Журналы веб-сервера

Иногда в мои обязанности входит анализ журналов веб-сервера, и в нескольких случаях я находил явные признаки использования приложений для автоматического сканирования веб-сервера, основываясь на «отпечатках» этих приложений в журналах. Когда я нахожу такие записи, то обычно спрашиваю администратора веб-сервера, подлежит ли организация сканированию на основе требований нормативных документов. Если ответ положительный, даты сканирования и IP-адреса, с которых было выполнено сканирование, можно напрямую связать с санкционированными действиями, но это не всегда так.

Анализ журналов IIS (и других веб-серверов) – обширная тема, которой можно уделить целую главу. Однако, как и в случае с большинством файлов журналов, принципы предварительной обработки данных остаются те же: исключите все записи, соответствующие санкционированным действиям. Или, если вы знаете, что ищете, можно использовать сигнатуры для поиска признаков отдельных действий.

Полевые заметки...

Журналы FTP-сервера

Я помогал расследовать одно дело, в котором неизвестный получил доступ к компьютеру с ОС Windows через программу удаленного управления (такую как WinVNC или pcAnywhere) и использовал установленный FTP-сервер (Microsoft), чтобы передавать файлы на компьютер и из него. Как и веб-сервер IIS, FTP-сервер сохраняет свои журналы в подкаталоге «MSFTPSVCx», который находится в каталоге «LogFiles». Не было никаких признаков, что пользователь попытался сделать что-либо, чтобы скрыть свое присутствие в системе, и мы смогли составить временную шкалу действий, используя журналы FTP-сервера в качестве исходных параметров. Благодаря формату журнала FTP-сервера по умолчанию, мы получили не только отметки даты/времени его посещений и используемое имя пользователя, но и адрес FTP-сервера, с которого осуществлялись подключения. Мы сопоставили эту информацию с отметками даты и времени действий из реестра (например, из разделов UserAssist и т. д.) и из журнала регистрации событий (несколько записей о событиях с идентификатором 10 указывали, что время ожидания соединения с FTP-сервером истекло из-за простоя), чтобы получить более полное представление о действиях пользователя в системе.

Log Parser

Теперь, когда мы рассмотрели журналы событий Windows и журналы веб-сервера IIS, пришло время вспомнить об инструменте от корпорации Microsoft, который чрезвычайно полезен для эксперта (даже несмотря на то, что производитель не уделяет должного внимания поддержке этого инструмента). Инструмент называется Log Parser и позволяет использовать язык SQL для поиска по ряду текстовых, XML- и двоичных файлов, таких как журналы событий или файлы реестра, и выводить данные в текстовом, SQL- или даже syslog-формате. (URL-адрес инструмента довольно длинный и может измениться, поэтому лучший способ найти ссылку на него – выполнить поиск по словам «log parser» в Google).

Чтобы облегчить использование этого инструмента, по адресу www.codeplex.com/visuallogparser доступно средство Visual Log Parser с графическим интерфейсом пользователя.

Log Parser – настолько мощный и при этом недооцененный инструмент, что Габриэле Джузеппини (Gabriele Giuseppini) и Марк Бёрннетт (Mark Burnett) написали книгу *Microsoft Log Parser Toolkit*, которая доступна в издательстве Syngress Publishing (а также в Amazon и книжных магазинах). Кроме того, есть ряд ресурсов, например, сайт Windows Dev Center, предоставляющих примеры использования Log Parser на различных уровнях сложности (www.windowsdevcenter.com/pub/a/windows/2005/07/12/logparser.html). Для одних экспертов использование Log Parser является самым естественным делом, другие же применяют его для особых задач, например для анализа важных журналов событий с нескольких компьютеров.

Журнал веб-браузера

На противоположном конце журналов веб-сервера находится журнал веб-браузера Internet Explorer. Internet Explorer устанавливается вместе с ОС Windows и является браузером по умолчанию для многих пользователей. Очень часто, некоторые веб-сайты (например, для предоставления информации о рабочем времени или командировочных расходах в корпоративной интрасети) могут быть специально разработаны для использования с Internet Explorer и не поддерживают другие браузеры, такие как Firefox и Opera. Когда браузер Internet Explorer применяется для просмотра интернет-страниц, он сохраняет историю своих действий, которую эксперт может использовать, чтобы получить представление об активности пользователя, а также чтобы найти данные для исследования. Файлы журнала Internet Explorer сохраняются в каталоге пользователя в подкаталоге Local Settings\Temporary Internet Files\Content.IE5. По этому адресу эксперт может найти несколько подкаталогов с именами, состоящими из восьми случайных символов. Структура и содержимое этих подкаталогов, а также структура файлов «index.dat» в каждом из них, подробно описана на других ресурсах, поэтому мы не будем здесь повторно рассматривать эту тему. Для анализа журнала браузера на работающих компьютерах эксперты могут использовать инструмент Web Historian (его версия 1.3 была доступна на сайте Mandiant.com на момент написания этой книги). При исследовании образа данных эксперты могут использовать такие инструменты, как Internet History Viewer из программы ProDiscover, чтобы объединить данные журнала браузера в понятный и удобный для просмотра формат. Файл «index.dat» из каждого подкаталога (либо на работающем компьютере, либо извлеченного из образа данных) можно просмотреть с помощью таких инструментов, как Index Dat Spy (www.stevengould.org/index.php?option=com_content&task=view&id=57&Itemid=220) и Index.dat Analyzer (www.systenance.com/indexdat.php).

Совет

При проведении исследования рекомендуется просматривать места, в которых можно найти информацию о том, какие данные должны присутствовать на компьютере.

Например, если аудит в системе настроен на регистрацию успешных входов, и вы можете узнать из реестра, когда различные пользователи последний раз входили в систему, то в журнале событий безопасности следует ожидать увидеть записи о событиях успешного входа. Что касается журнала просмотра интернет-страниц, в Internet Explorer можно настроить количество дней, в течение которых браузер будет сохранять историю посещенных URL-адресов. Этот параметр настройки можно найти в пользовательском кусте (файл «ntuser.dat», или куст HKEY_CURRENT_USER, если пользователь вошел в систему) в разделе \Software\Microsoft\Windows\CurrentVersion\Internet Settings\URL History. Параметр, о котором идет речь, называется *DaysToKeep*, а его значение по умолчанию равно 0x014, или 20 в десятичной системе счисления. Если данные, связанные с этим параметром не соответствуют значению по умолчанию, то можно предположить, что пользователь изменил параметр, вероятно, открыв в Internet Explorer меню «Сервис» (“Tools”), выбрав пункт «Свойства обозревателя» (“Internet Options”), а затем перейдя в раздел «История просмотра» (“History”) во вкладке «Общие» (“General”). Время *LastWrite* этого раздела реестра подскажет вам, когда параметр был изменен.

Многие эксперты знают, как использовать историю просмотра интернет-страниц для документирования действий пользователя. Например, можно найти ссылки на сайты, с которых загружаются вредоносные программы, социальную сеть MySpace.com или другие веб-страницы, которые пользователи не должны посещать. Как и в случае с большинством аспектов судебных артефактов в системе, то, что вы ищите в качестве улик, в действительности зависит от характера дела. Тем не менее, ничего нельзя пропускать; маленькие кусочки информации могут дать вам зацепки или контекст к имеющимся данным или к делу в целом. Однако не все пользователи используют Internet Explorer, существует ряд других браузеров, а именно: Mozilla, Firefox, Opera и Google Chrome. Бесплатные инструменты с веб-сайта NirSoft (www.nirsoft.net/utils/) позволяют просматривать журналы, кэш и файлы «cookie» ряда браузеров, а также (в некоторых случаях) получать пароли, сохраненные самими браузерами. Все эти инструменты могут предоставить очень ценную информацию во время экспертизы.

К другим инструментам для судебного анализа браузеров относятся Firefox Forensics (F3) и Google Chrome Forensics, которые доступны (за плату) на сайте Machor Software (www.machorsoftware.com/home), и Historian, бесплатное средство на немецком языке с сайта Gaijin (www.gaijin.at/dlhistorian.php). Кроме того, есть ряд статей, в которых рассматривается судебная экспертиза браузеров, например, статья Джона Маккэша (John McCash) об анализе браузера Safari в блоге SANS Forensics (<http://sansforensics.wordpress.com/2008/10/22/safari-browser-forensics/>), а также серия статей из двух частей о судебном исследовании браузеров от Кита Джонса (Keith Jones) и Рохита Белани (Rohyt Belani) на веб-сайте SecurityFocus (первая часть – www.securityfocus.com/infocus/1827). Поиск в Google по словам «*browser forensics*» также поможет получить много сведений, однако они не всегда будут такими подробными, как в вышеупомянутых статьях. Тем не менее, в этой области проводится большая работа, поэтому не забывайте отслеживать новую информацию по данной теме.

Другие файлы журналов

ОС Windows сохраняет ряд других, менее известных файлов журналов как во время установки операционной системы, так и в течение повседневных операций. Некоторые из этих журналов предназначены для записи действий и ошибок, происходящих во время установки. Другие файлы журналов создаются или добавляются, только когда происходят определенные события. Эти файлы журналов могут быть очень полезны для эксперта, который понимает не только то, что они существуют, но и то, какие действия являются причиной их создания или увеличения в размере, а также как анализировать и трактовать

имеющуюся в них информацию. В этом разделе мы рассмотрим несколько таких файлов журналов.

Setuplog.txt

Файл «setuplog.txt», расположенный в каталоге «Windows», используется для записи информации во время установки ОС Windows. Возможно, самое главное, что должен знать эксперт об этом файле, это то, что в нем сохраняются временные отметки всех регистрируемых действий, указывая на дату и время установки системы. Эта информация поможет вам при составлении временной шкалы действий в системе.

Ниже показан фрагмент файла «setuplog.txt» из ОС Windows XP с пакетом обновления 2 (SP2):

```
08/07/2006 16:14:22.921,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,6434,
BEGIN_SECTION,Installing Windows NT
08/07/2006 16:14:24.921,d:\xpsprtm\base\ntsetup\syssetup\wizard.c,1568,,,
SETUP: Calculating registry size
08/07/2006 16:14:24.921,d:\xpsprtm\base\ntsetup\syssetup\wizard.c,1599,,,
SETUP: Calculated time for Win9x migration = 120 seconds
08/07/2006 16:14:24.937,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,6465,
BEGIN_SECTION,Initialization
08/07/2006 16:14:24.984,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,6585,
BEGIN_SECTION,Common Initialization
08/07/2006 16:14:25.000,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,1674,
BEGIN_SECTION,Initializing action log
08/07/2006 16:14:25.046,d:\xpsprtm\base\ntsetup\syssetup\log.c,133,,GUI mode
Setup has started.
08/07/2006 16:14:25.078,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,1679,
END_SECTION,Initializing action log
08/07/2006 16:14:25.093,d:\xpsprtm\base\ntsetup\syssetup\syssetup.c,1764,
BEGIN_SECTION,Creating setup background window
```

Предупреждение

Во время написания этой книги я исследовал разные версии ОС Windows в отношении файла «setuplog.txt». В ОС Windows 2000 в этом файле были отметки времени, но отсутствовала информация о датах. В ОС Windows XP и 2003 содержимое файлов было похоже в том, что каждая запись содержала отметку времени с датой. Мне не удалось найти файл «setuplog.txt» в ОС Windows Vista.

Как показано в фрагменте файла «setuplog.txt» из Windows XP, отметка даты и времени содержится в каждой записи.

Предупреждение

Если во время анализа образа данных отметки времени, показанные в файле «setuplog.txt», лишены всякого смысла (временная шкала не соответствует другой полученной информации), то система, возможно, была установлена из образа, созданного программой клонирования, или восстановлена из резервной копии данных. Не забывайте, что файл «setuplog.txt» регистрирует действия, происходящие во время установки, поэтому, чтобы файл содержал полезную информацию, операционная система должна быть инсталлирована на компьютер, а не восстановлена из резервной копии.

Setupact.log

Файл «setupact.log», расположенный в каталоге «Windows», содержит список действий, происходивших во время графической стадии процесса установки. В ОС Windows 2000, XP и 2003 в этом файле не будет отметок времени, связанных с различными зарегистрированными действиями, но даты создания и последнего изменения файла предоставят эксперту сведения о времени установки операционной системы. В ОС

Windows Vista этот файл содержит отметки даты и времени для многих зарегистрированных действий.

Setupapi.log

Файл «setupapi.log», расположенный в каталоге «Windows», содержит записи об установке устройств, пакетов обновлений и пакетов исправлений в ОС Windows. Журналы в Windows XP и более поздних версиях ОС Windows ведутся подробнее, чем в предыдущих версиях, и хотя Microsoft использует этот файл главным образом для поиска и устранения неполадок, информация в нем может быть чрезвычайно полезной для эксперта.

Microsoft предоставляет документ «Troubleshooting Device Installation with the SetupAPI Log File», содержащий большое количество ценной информации о файле «setupapi.log». Например, файл «setupapi.log» содержит раздел заголовка об установке Windows, в котором указана версия операционной системы и другая информация. Если файл «setupapi.log» по какой-либо причине был удален, операционная система создает новый файл и вставляет в него заголовок об установке Windows.

Сведения об установке устройств также записываются в этот файл вместе с отметками времени, которые эксперт может использовать, чтобы отслеживать такие действия в системе. В главе 4 вы видели, что, когда съемное запоминающее USB-устройство (флеш-накопитель, iPod и т. д.) подключается к компьютеру с ОС Windows, изменения записываются в системный реестр. Когда отдельный вид устройства подключается к компьютеру впервые, выполняется поиск и загрузка драйвера, поддерживающего это устройство. В случаях, когда несколько экземпляров одинакового типа устройства подключаются к компьютеру с ОС Windows, поиск драйвера выполняется только для первого подключаемого устройства. При последующем подключении устройств такого же типа к компьютеру, информация в реестре обновляется. Посмотрите на этот фрагмент из файла «setupapi.log»:

```
[2006/10/18 14:11:53 1040.8 Driver Install]
#-019 Searching for hardware ID(s): usbstor\disksony_sony_
dsc_5.00,usbstor\disksony_sony_dsc_,usbstor\disksony_,usb
stor\sony_sony_dsc_5,sony_sony_dsc_5,usbstor\gendisk,gend
isk
#-018 Searching for compatible ID(s): usbstor\disk,usbstor\raw
#-198 Command line processed: C:\WINDOWS\system32\services.exe
#I022 Found "GenDisk" in C:\WINDOWS\inf\disk.inf; Device: "Disk drive";
Driver:
"Disk drive"; Provider: "Microsoft"; Mfg: "(Standard disk drives)"; Section
name:
"disk_install".
#I023 Actual install section: [disk_install.NT]. Rank: 0x00000006. Effective
driver date: 07/01/2001.
#-166 Device install function: DIF_SELECTBESTCOMPATDRV.
#I063 Selected driver installs from section [disk_install] in "c:\windows\
inf\disk.inf".
#I320 Class GUID of device remains: {4D36E967-E325-11CE-BFC1-08002BE10318}.
#I060 Set selected driver.
#I058 Selected best compatible driver.
#-166 Device install function: DIF_INSTALLDEVICEFILES.
#I124 Doing copy-only install of "USBSTOR\DISK&VEN_SONY&PROD_SONY_
DSC&REV_5.00\6&1655167&0".
```

В этом фрагменте журнала мы видим, что съемное запоминающее USB-устройство, изготовленное корпорацией Sony, было впервые подключено к компьютеру 18 октября 2006 года. Исходя из того, что мы узнали в главе 4, можно определить по последней записи журнала, что у устройства нет серийного номера. Однако отметка даты и времени

в разделе «Driver Install» показывает, когда устройство было впервые подключено к компьютеру, что можно использовать вместе с временем *LastWrite* соответствующего раздела реестра, чтобы определить временную шкалу использования устройства в компьютере.

Netsetup.log

Файл «netsetup.log» создается во время установки системы; в Windows XP он находится в папке Windows\Debug. В этом файле записывается информация о членстве компьютера в рабочей группе и домене, а также сохраняются отметки времени всех регистрируемых записей. Отметки времени в файле «netsetup.log» записываются в тот же период времени, что и отметки в файле «setuplog.txt». Дополнительные записи добавляются в файл, если рабочая группа или домен компьютера изменяется. Например, я установил операционную систему Windows XP на личном ноутбуке 7 августа 2006 года, о чем свидетельствуют отметки времени в журналах «netsetup.log» и «setuplog.txt». Девятнадцатого ноября 2006 года я изменил рабочую группу (сменил группу WorkGroup на Home) компьютера, включив общий доступ к файлам. Эта информация была записана в файл «netsetup.log» вместе с соответствующими отметками времени. Записи также добавляются в этот файл журнала, если компьютер добавляется в домен или удаляется из него.

Журнал планировщика заданий

Доступ к службе планировщика заданий в системах Windows можно получить посредством файла «at.exe» или «Мастера планирования заданий» (“Scheduled Task Wizard”) в панели управления. Эта служба позволяет пользователю с правами администратора планировать задание, которое будет запущено в определенный момент времени в будущем или будет выполняться в указанное время ежедневно, еженедельно или ежемесячно. Она очень эффективна для администрирования компьютера или целой сети. Эта же служба используется злоумышленниками, которые хотят, чтобы их вредоносная программа постоянно выполнялась во взломанной системе; в действительности ряд вредоносных программ (например, Conficker/Downadup) используют именно этот способ как средство обеспечения сохраняемости в зараженной системе. К счастью эта служба ведет журнал заданий, которые были запущены, в файле «schedlgu.txt». Этот файл журнал фактически имеет имя по умолчанию, связанное с параметром *LogFile*, который находится в следующем разделе реестра:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SchedulingAgent

В ОС Windows XP файл «schedlgu.txt» находится в каталоге «Windows» по умолчанию (C:\Windows), а в ОС Windows 2003 и Vista – в каталоге «Tasks» (C:\Windows\Tasks).

Совет

Microsoft предоставляет множество информации в статьях из базы знаний. В частности, в статье № 169443 (<http://support.microsoft.com/kb/169443>) объясняется, как ограничить размер файла журнала назначенных заданий.

Обратите внимание, что для создания назначенного задания требуются права администратора; очень часто вредоносная программа, заражающая компьютер и использующая этот (или другой) механизм сохраняемости, достигает своей цели потому, что обычные пользователи имеют права администратора.

Если планировщик заданий используется не администратором, то эксперт должен увидеть записи, в которых указывается дата и время начала и завершения работы службы планировщика заданий. Так как служба планировщика заданий обычно настроена на

запуск при загрузке компьютера, эта информация может помочь эксперту определить время начала и завершения работы компьютера.

Если задание было назначено и выполнено, в файле «schedlgu.txt» будут находиться записи, похожие на те, что показаны ниже (фрагмент из файла журнала планировщика в Windows XP):

```
"At1.job" (regedit.exe)
    Started 9/26/2006 4:35:00 PM
"At1.job" (regedit.exe)
    Finished 9/26/2006 4:35:04 PM
    Result: The task completed with an exit code of (0).
"Pinball.job" (PINBALL.EXE)
    Started 9/26/2006 4:36:00 PM
"Pinball.job" (PINBALL.EXE)
    Finished 9/26/2006 4:36:07 PM
    Result: The task completed with an exit code of (0).
```

Первое задание было назначено с помощью «at.exe», а второе (pinball.job) – с помощью мастера планирования заданий. Эти файлы заданий (с расширением .job) хранятся в каталоге Windows\Tasks.

Записки из подполья...

Как скрывать назначенные задания

Существует эффективный способ скрытия назначенных заданий. Создайте назначенное задание с помощью команды «at.exe» или мастера планирования заданий. Перейдите в панель управления, откройте приложение «**Назначенные задания**» (“Scheduled Tasks”) и удостоверьтесь, что в нем указано только что созданное задание. Теперь закройте приложение, откройте командную строку, перейдите к каталогу **Windows\Tasks** и используйте «**attrib.exe**», чтобы установить атрибут «скрытый» для job-файла. После этого снова откройте приложение «**Назначенные задания**», и вы больше не увидите там созданного задания. Конечно, должны применяться обычные ограничения к командной строке (необходимо использовать правильный переключатель с командой *dir*) и проводнику Windows (по умолчанию в нем не отражаются скрытые файлы). Тем не менее, задание будет выполняться, если вы его запланируете.

По правде говоря, я попался на этот трюк во время написания первой книги. Я написал текст и выполнил вышеупомянутую процедуру с карточной игрой «Косынка», но забыл удалить файл задания. Затем я отправился домой, где работал все выходные. Когда я вернулся в свой офис, на рабочем столе была открыта игра «Косынка», и сначала я подумал, что кто-то был в моем офисе! Затем я понял, что произошло, и удалил job-файл

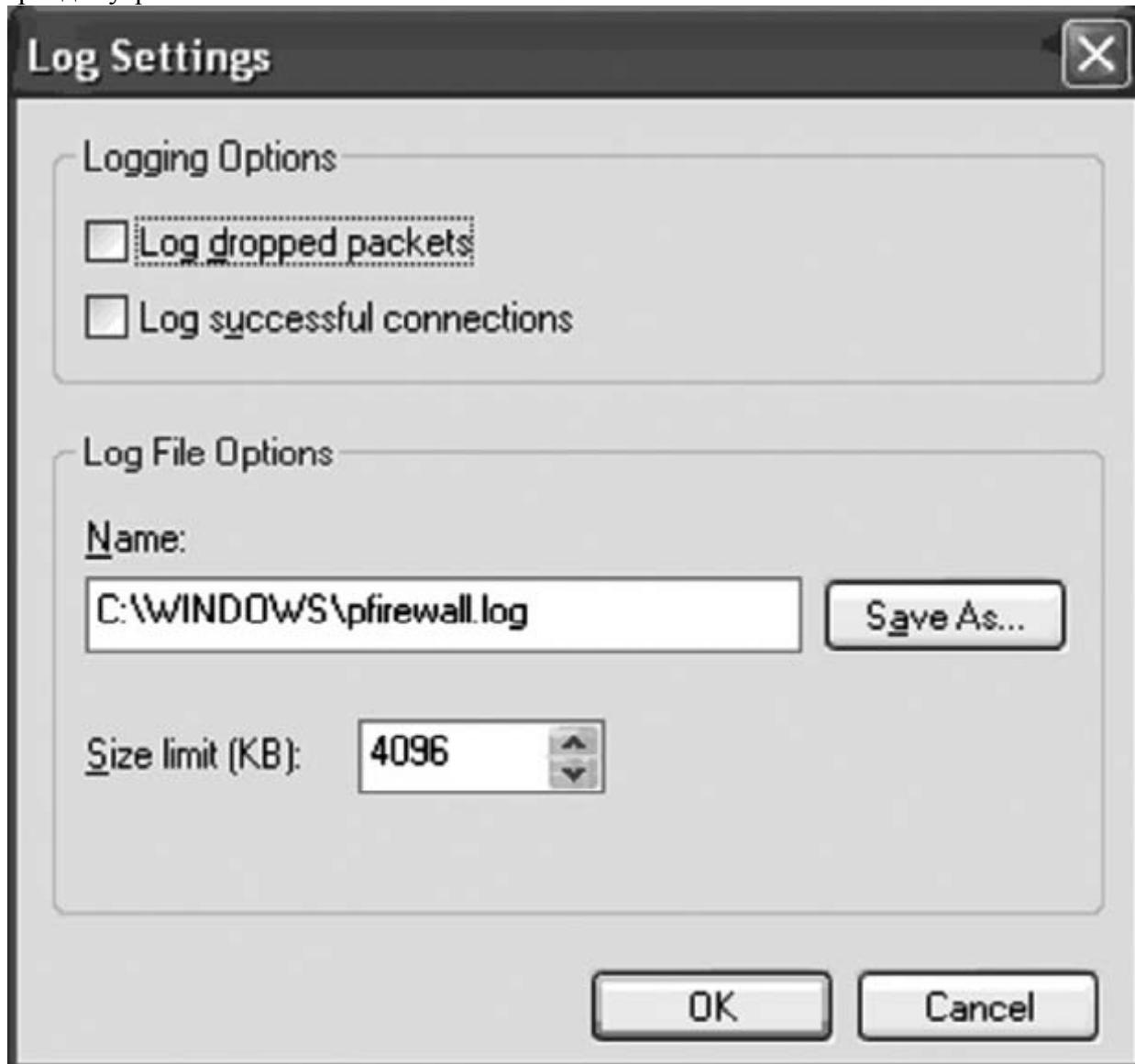
К сожалению, в файле журнала не регистрируется полный путь к исполняемому файлу, который запускается заданием, но предоставляется указание на то, когда программа была запущена с помощью службы планировщика заданий.

Журналы брандмауэра Windows XP

Большинство из нас знают о компонентах брандмауэра, встроенных в ОС Windows XP, возможно, из средств массовой информации и из-за проблем, решенных в Windows XP с пакетом обновления 2 (SP2). Большая часть пользователей никогда даже не видели брандмауэр Windows XP или не взаимодействовали с ним, хотя он включен по умолчанию. Брандмауэр можно отключить (некоторые вредоносные программы пытаются сделать это), и такое действие может быть частью конфигурационной схемы в корпорации, чтобы облегчить управление этими компьютерами. Кроме того, брандмауэр

можно вручную настроить так, чтобы отдельным приложениям был разрешен доступ к сети.

В брандмауэре Windows XP есть файл журнала, в котором регистрируются различные события, но по умолчанию ведение журнала не включено. На илл. 5.6 показаны параметры по умолчанию в диалоговом окне «Параметры журнала» («Log Settings») для брандмауэра.



Илл. 5.6. Параметры журнала брандмауэра в ОС Windows XP.

Как видите, опции ведения журнала достаточно ограниченные. Ведение журнала не включено по умолчанию, поэтому вы, вероятно, не найдете журнал брандмауэра «pfirewall.log» в большинстве системах. Отсутствие файла журнала не означает, что брандмауэр отключен. Тем не менее, если вы вдруг найдете в системе экземпляр файла журнала, сможете легко и просто понять его формат. Фрагмент из примерного журнала брандмауэра показан ниже:

```
#Version: 1.0
#Software: Microsoft Internet Connection Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size
tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info
2003-10-10 10:21:11 DROP ICMP 131.107.0.2 131.107.0.1 - - 60 - - - 8 0 -
2003-10-10 10:21:16 DROP ICMP 131.107.0.2 131.107.0.1 - - 60 - - - 8 0 -
2003-10-10 10:21:21 DROP ICMP 131.107.0.2 131.107.0.1 - - 60 - - - 8 0 -
```

```

2003-10-10 10:21:26 DROP ICMP 131.107.0.2 131.107.0.1 - - 60 - - - 8 0 -
2003-10-10 10:21:34 DROP TCP 131.107.0.2 131.107.0.1 1045 21 48 S
1226886480 0 16384 - - -
2003-10-10 10:21:37 DROP TCP 131.107.0.2 131.107.0.1 1045 21 48 S
1226886480 0 16384 - - -
2003-10-10 10:21:43 DROP TCP 131.107.0.2 131.107.0.1 1045 21 48 S
1226886480 0 16384 - - -

```

Тер *Fields* в заголовке журнала брандмауэра указывает, к чему относятся различные части записей журнала и как интерпретировать информацию в файле журнала. Из записей в фрагменте журнала «*pfirewall.log*» мы видим, что несколько ICMP-пакетов (возможно, из приложения «*ping.exe*») было отброшено; то же относится к нескольким попыткам подключиться к компьютеру, используя порт 21 (порт для FTP-серверов по умолчанию).

Предупреждение

Часто бывает трудно интерпретировать записи о действиях в файле «*pfirewall.log*», не имея более детального представления о компьютере и его окружении. Например, когда я просматривал другие журналы сетевой активности, такие как журналы корпоративного брандмауэра или системы обнаружения вторжений, администратор спросил меня, что означает эта активность. В случае с отдельным компьютером попытки получить доступ к таким известным портам, как порт 80 (веб-сервер) или порт 21 (FTP-сервер), не всегда означают, что что-то выполняется в этой системе, а скорее то, что кто-то, возможно, пытается определить, чем используется этот порт. Это может означать разведывательные действия, например, сканирование портов. Если в журналах показано, что похожие действия направлены на несколько компьютеров в одно и то же время, это свидетельствует о широкомасштабном сканировании портов. Если запись журнала показывает действие, направленное на отдельный порт, это необязательно означает, что порт в системе был открыт (служба прослушивала этот порт). Это явление, которое обычно неправильно понимают, особенно когда дело касается широкомасштабного сканирования, направленного на порты, используемые троянскими приложениями удаленного администрирования.

Для облегчения просмотра существует ряд бесплатных программ, которые проанализируют этот файл и помогут интерпретировать его данные, вплоть до цветового выделения некоторых записей. Можно выполнить поиск в Google по словам «XP», «firewall» и «viewer», чтобы найти программу, отвечающую вашим потребностям.

Совет

На DVD-диске, который идет в комплекте с этой книгой, есть каталог «Chapter 5», содержащий подкаталог «samples». В этом подкаталоге находится файл с именем «*nmap_xp_scan.txt*», который содержит командную строку, используемую, чтобы запустить программу Nmap для сканирования ОС Windows XP SP2 (с включенным брандмауэром), а также результаты сканирования, отправленные на стандартное устройство вывода. Другой файл с именем «*pfirewall_nmap_scan.txt*» содержит информацию о зарегистрированных пакетах, отправленных на целевой компьютер. Для удобства просмотра сканирование Nmap было запущено с IP-адреса 192.168.1.28, а IP-адрес целевой системы – 192.168.1.6.

Mrt.log

Помимо программ для обеспечения безопасности, таких как брандмауэры, Microsoft также использует приложения для решения проблем с вредоносным ПО; одно из таких приложений называется «Средство удаления вредоносных программ» («Malicious Software Removal Tool») (<http://support.microsoft.com/kb/890830>). Подобно инструменту

Stinger от McAfee (<http://vil.nai.com/vil/stinger/>), это средство предназначено не для обнаружения всех вредоносных программ и защиты от них, а для поиска и удаления отдельных вредоносных программ, перечисленных в статье № 890830 из базы знаний Microsoft. Следует отметить, что приблизительно каждый месяц в базу данных средства добавляется несколько новых угроз, которые можно устраниć; в июне 2008 этот инструмент версии 1.42 мог удалить в общей сложности восемь вредоносных программ.

Файл журнала для средства удаления вредоносных программ называется «mrt.log» и находится в каталоге %WinDir%\Debug. Он содержит информацию о версии инструмента, дате установки и результатах сканирования, как показано ниже:

```
Microsoft Windows Malicious Software Removal Tool v2.5, December 2008
Started On Fri Dec 12 06:55:23 2008-
Results Summary:
-----
No infection found.
Return code: 0
Microsoft Windows Malicious Software Removal Tool Finished On
Fri Dec 12 06:56:52 2008
```

Эта информация может помочь экспертам получить представление об угрозах, которым, возможно, была подвержена система, и о вредоносных программах, которые, возможно, были удалены.

В том же каталоге можно найти файл «mrting.log», который содержит похожую информацию только без результатов сканирования.

Журналы программы «Доктор Ватсон»

Программа «Доктор Ватсон» (<http://support.microsoft.com/kb/308538/>) довольно давно входит в состав операционной системы Windows, но сегодня о ней мало говорят. При возникновении ошибок в работе приложений «Доктор Ватсон» собирает информацию о системе и программной ошибке в текстовый файл, который затем может быть отправлен специалистам службы поддержки для устранения неисправностей. Эта информация может также быть полезна при исследовании проблем в работе системы.

Текстовый файл журнала, создаваемый программой «Доктор Ватсон», называется «drwtsn32.log» и содержится в следующем каталоге:

```
C:\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson
```

Информация о конфигурации программы «Доктор Ватсон» находится в следующем разделе реестра:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DrWatson
```

Этот раздел реестра содержит ряд параметров, отображаемых в графическом интерфейсе программы «Доктор Ватсон», к которому можно получить доступ, открыв меню «Пуск» (“Start”), выбрав пункт «Выполнить» (“Run”) и введя команду drwtsn32. По умолчанию файл журнала сохраняет информацию о 10 программных ошибках. Эти параметры показывают эксперту, что он должен ожидать увидеть, если произошла ошибка в системе.

При возникновении ошибки информация, сохраняемая программой «Доктор Ватсон», добавляется в конец файла «drwtsn32.log». Сначала «Доктор Ватсон» записывает в файл раздел, который начинается строкой «Иключение в приложении: » (“Application exception occurred:”). Этот раздел содержит информацию о программе, которая вызвала ошибку, а также дату и время возникновения ошибки:

App: C:\Perl\bin\perl.exe (pid=4040)
When: 8/21/2006 @ 10:17:35.859

Обратите внимание, что имя программы, вызвавшей ошибку, включает в себя полный путь к исполняемому файлу, а ниже указана отметка даты и времени. Как вы поняли из предыдущих глав, эта информация может быть полезной для эксперта, особенно в случаях, когда рассматриваемая программа является вредоносным средством или объектом, помещенным в систему в результате вторжения или несанкционированного использования. Затем «Доктор Ватсон» записывает сведения о системе, список запущенных процессов, список модулей (DLL-файлов), загружаемых программой, и копии стека, которые можно использовать для поиска и устранения неисправностей программы. Эксперт может использовать эту информацию, чтобы узнать, какой пользователь входил в систему в определенное время, какие программы выполнялись (что также показывает, какие приложения были установлены), и какие DLL-файлы были загружены программой, вызвавшей ошибку (что может также показать объекты модуля поддержки браузера (ВНО), установленные посредством Internet Explorer, любые DLL-файлы, внедренные в процесс, чтобы нарушить его работоспособность, и т. д.).

Примечание

Журнал программы «Доктор Ватсон» может быть чрезвычайно полезен при демонстрации или подтверждении данных из временной шкалы действий в системе. Однажды пользователь, получивший доступ к компьютеру, загрузил на него несколько инструментов, но при попытке установить некоторые из них возникла ошибка. Мы нашли журналы, свидетельствующие о его доступе в систему, журналы, показывающие, что он загрузил эти инструменты, а также IP-адрес, с которого было произведено подключение, записи журнала событий, показывающие всплывающее сообщение об ошибке приложения, и журнал программы «Доктор Ватсон», показывающий, в работе какого приложения произошел сбой. Помимо этой информации, у нас был контекст пользователя для приложения, когда произошел его сбой, а также список других приложений, выполняющихся во время сбоя. Все эти сведения помогли подтвердить наше представление о том, какие приложения уже были на компьютере до того, как пользователь получил к нему доступ, какие приложения были добавлены в систему, и когда он их использовал.

«Доктор Ватсон» также создает файл аварийного дампа памяти («user.dmp»), который находится в том же каталоге, что и текстовый файл журнала. Этот дамп содержит частные страницы памяти, используемые процессом во время возникновения ошибки, но не содержит кодовые страницы из исполняемых файлов (EXE, DLL и т. д.). Файл «user.dmp» можно открыть с помощью инструмента WinDbg, который входит в состав средств отладки (Debugging Tools) от Microsoft. Однако этот файл перезаписывается после каждой ошибки, поэтому вы увидите только файл «user.dmp», содержащий данные о последней ошибке. Тем не менее, имеющийся файл «user.dmp» может содержать чрезвычайно полезную информацию, такую как пароли, незашифрованные данные или другие признаки действий пользователя.

Cbs.log

В состав ОС Windows Vista и 2008 входит приложение «Диспетчер пакетов» («Package Manager»), которое используется для установки и удаления различных пакетов в этих операционных системах. Диспетчер пакетов сохраняет свои журналы в файле %WinDir%\Logs\Cbs\cbs.log. Microsoft предоставляет отличную статью, в которой объясняется, как анализировать записи в этом файле журнала (<http://support.microsoft.com/kb/928228>), и эксперт может найти в нем полезные сведения,

которые помогут ему решить проблему. Например, средство проверки ресурсов Windows (sfc.exe) делает записи в этот файл, проверяя во время сканирования, что защищенные системные файлы не изменились. В статье № 928228 из базы знаний Microsoft предоставлен пример сканирования, во время которого не было обнаружено ошибок, а также пример сканирования, во время которого найден и исправлен поврежденный файл. Этот журнал может быть хорошим источником информации для эксперта, показывая или исключая проблемы с поврежденными файлами. Согласно статье № 954402 из базы знаний Microsoft (<http://support.microsoft.com/kb/954402>), в файле «cbs.log» в ОС Windows 2008 можно найти записи о том, что некоторые файлы не были исправлены, хотя сообщается, что проверка завершена успешно.

Файлы аварийного дампа памяти

Мы рассматривали файлы аварийного дампа памяти в главе 3. Я подумал, что для полноты картины не помешает также упомянуть их в этой главе.

В главе 3 мы говорили о способах настройки и создания файлов аварийного дампа памяти, но в большинстве случаев я обнаруживал, что сами системы совсем не изменялись. Если во время экспертизы или расследования инцидента вы найдете файл аварийного дампа памяти, рекомендуется просмотреть его содержимое. Можно использовать такие инструменты, как «dumpchk.exe» (для Windows 2000/2003, см. <http://support.microsoft.com/kb/156280>; для XP см. <http://support.microsoft.com/kb/315271>), чтобы проверить дамп памяти и убедиться, что он правильный. Затем можно загрузить файл в средство отладки (например, в WinDbg) и использовать такие команды как *!process 0 0*, чтобы просмотреть список процессов, выполнявшихся во время сбоя, или *!m kv*, чтобы просмотреть список загруженных драйверов, работающих в режиме ядра. Более того, можно использовать такие инструменты, как «strings.exe», «bintext.exe» и выражения grep, чтобы найти конкретную информацию.

Корзина

Большинство судебных экспертов знают о том что, после того, как файл удаляется, он на самом деле не исчезает. Истинность этого факта стала еще более очевидна с появлением корзины на рабочем столе Windows. Корзина существует как метафора для выбрасывания файлов, как будто мы их сминаем и бросаем в корзину для ненужных бумаг. Но корзина также позволяет нам находить и восстанавливать файлы, которые мы «случайно» выбросили. Мы можем открыть корзину, выбрать ранее удаленные файлы и восстановить их в предыдущее местонахождение.

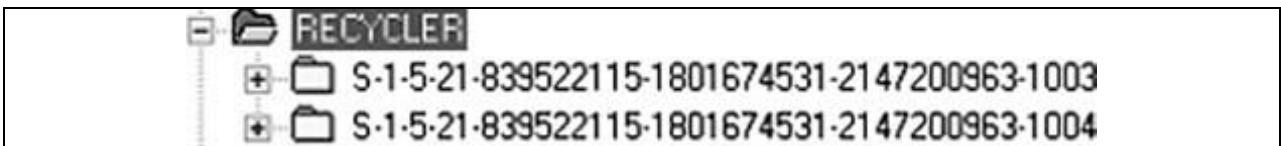
Итак, когда какой-нибудь объект удаляется через оболочку – то есть когда пользователь выбирает файл на рабочем столе или в проводнике Windows и «удаляет» его – он на самом деле не исчезает. Файл просто перемещается в корзину, которая по умолчанию отображается в файловой структуре как каталог «Recycler» в корне каждого логического диска. Во многих случаях этот каталог может предоставить большое количество информации, относящейся к расследованию.

Для того чтобы лучше понять, как информацию этого каталога можно использовать в качестве улик, давайте посмотрим, что происходит, когда пользователь удаляет файл через оболочку. Как только пользователь, вошедший в систему, начинает удалять файлы через оболочку (в отличие от использования команды *del* или *erase* в командной строке), в каталоге «Recycler» создается подкаталог для этого пользователя; этому каталогу присваивается имя, состоящее из идентификатора SID пользователя. Например, в командной строке путь к подкаталогу будет выглядеть приблизительно так:

```
C:\RECYCLER\S-1-5-21-1454471165-630328440-725345543-1003>
```

Когда вы открываете корзину через значок на рабочем столе, то автоматически открывается подкаталог пользователя. Поэтому, если бы вы получили доступ к ноутбуку, на которым выполнен вход с использованием учетной записи пользователя, и открыли корзину, чтобы просмотреть ее содержимое, вы бы увидели файлы, которые пользователь «удалил». Если бы вам нужно было переключиться между учетными записями и повторить процесс, вы бы автоматически увидели файлы, удаленные в учетной записи активного пользователя.

При просмотре каталога «Recycler» в образе данных, вы увидите подкаталог для каждого активного пользователя в системе, который удалял файлы через оболочку, как показано на илл. 5.7:



Илл. 5.7. Пример каталога корзины, отображаемого в ProDiscover.

В каждом подкаталоге вы, возможно, увидите несколько файлов, количество которых зависит от активности пользователя и от того, как часто пользователь очищал корзину. Файлы, отправляемые в корзину, сохраняются в соответствии со специальным соглашением об именах (<http://support.microsoft.com/kb/136517>), разобравшись в котором, можно относительно легко определить отдельные типы файлов или установить, какие файлы представляют интерес для дела. Когда файл перемещается в корзину, он переименовывается с использованием следующего соглашения:

D<символ исходного накопителя><#>.〈исходное расширение〉

Имя файла начинается с буквы *D*, за которой следует символ накопителя, с которого был удален файл, а затем порядковый номер файла, начиная с нуля (т. е. пятому удаленному файлу будет присвоен номер 4). Исходное расширение файла сохраняется. Кроме того, добавляется запись в файл INFO2 (в этом каталоге), который представляет собой файл журнала всех удаленных файлов, находящихся в данный момент в корзине. Порядковый номер удаленного файла служит ссылкой на исходные имя и путь к файлу, которые хранятся в файле INFO2.

К счастью, Кит Джонс (Keith Jones), ранее работавший в компаниях Foundstone и Mandiant, задокументировал формат файла INFO2, чтобы эта информация была более полезной для эксперта. Файл INFO2 содержит записи, соответствующие каждому удаленному файлу в корзине; каждая запись содержит номер записи, обозначение накопителя, отметку с указанием времени перемещения файла в корзину, размер файла, исходное имя файла и исходный полный путь к файлу в форматах ASCII и Unicode.

Файл INFO2 начинается с 16-байтового заголовка, в котором последнее значение двойного слова является размером каждой записи. Это значение равно 0x320 (с порядком байтов от младшего к старшему), что при преобразовании дает 800 байт. Первая запись начинается сразу после заголовка, и ее общая длина равна 800 байт.

На первое двойное слово (четыре байта) можно не обращать внимания. Исходное имя файла и путь к нему (в формате ASCII) хранятся в строке с завершающим нулем, которая начинается после первого двойного слова и занимает первые 260 байт записи. Открыв файл INFO2, вы увидите, что большая часть пространства, занятого именем файла в формате ASCII, заполнена нулями. Эти нули можно отбросить, чтобы получить только имя файла. Ниже перечислены остальные элементы записи:

- § Номер записи содержится в двойном слове, находящемся по смещению 264 относительно начала записи.
- § Обозначение накопителя содержится в двойном слове, находящемся по смещению 268 относительно начала записи. Обозначение накопителя служит для того, чтобы определить, с какого накопителя был удален файл; 2 = C:\, 3 = D:\ и т. д.

- § Отметка с указанием времени перемещения файла в корзину – это 64-разрядное значение структуры *FILETIME*, находящееся по смещению 272 относительно начала записи.
- § Размер удаленного файла (кратный размеру кластера) содержится в двойном слове, находящемся по смещению 280 относительно начала записи.

Исходное имя файла в формате Unicode занимает остальную часть записи, начиная со смещения 284 относительно начала записи и до конца (516 байт). Просто отбросив пустые байты, мы получим путь к файлу и имя файла на английском языке в формате ASCII. (Размер символа в формате Unicode равен двум байтам, поэтому, удалив пустые байты из второй половины формата Unicode, мы получим просто формат ASCII на английском языке.)

Perl-скрипт «*grecbin.pl*», расположенный на DVD-носителе, который идет в комплекте с этой книгой, восстановит различные элементы из каждой записи и покажет номер записи, отметку времени, указывающую, когда файл был перемещен в корзину (в формате UTC; параметры часового пояса для системы не учитываются), и исходные имя файла и путь к файлу. Скрипт использует путь к файлу *INFO2* в качестве единственного аргумента, а его выходными данными можно легко управлять, что позволяет получить любой формат, необходимый для эксперта.

Кит Джонс также предоставляет инструмент, который называется *Rifiuti* (итал. мусор), для анализа файла *INFO2*. *Rifiuti* – бесплатный инструмент (доступный на сайте [Foundstone.com](http://www.foondstone.com)), позволяющий проанализировать файл *INFO2* и сохранить выходные данные в формате, который можно легко просмотреть с помощью любой программы для работы с электронными таблицами.

Записки из подполья...

Тщательно изучайте содержимое корзины

Экспертам следует также обратить пристальное внимание на файлы, находящиеся в каталоге «*Recycler*», но не хранящиеся ни в одном из подкаталогов с именем в виде SID пользователя, а также на файлы, которые не соответствуют соглашению об именах для файлов, перемещенных в корзину. Это может свидетельствовать о злонамеренных действиях пользователя или вредоносной программы, имеющих целью скрыть файл. Эксперты должны знать, что такие программы, как Norton AntiVirus могут использовать корзину; приложение Norton Recycle Bin Protector помещает файл «*nprotect.log*» в каталог. Компания Datalifter, выпускающая инструменты для судебного анализа, предоставляет программу *NProtect Viewer* (www.datalifter.com/tutorial/bt/NProtect_Using_NProtect.htm), которая анализирует файл «*nprotect.log*». *NProtect Viewer* входит в состав пакета Datalifter .Net Bonus Tools.

Одно из первых действий, которое я выполняю во время исследования образа данных, – проверяю время последнего изменения файла *INFO2*. Таким образом я узнаю, когда последняя запись была добавлена в файл *INFO2*, что приблизительно равно времени перемещения соответствующего файла в корзину. Если подкаталог пользователя в каталоге «*Recycler*» содержит только файлы «*desktop.ini*» и *INFO2*, и файл *INFO2* имеет небольшой размер, то время последнего изменения соответствует времени, когда пользователь последний раз очистил корзину (т. е. щелкнул по значку «**Корзина**» (“Recycle Bin”) правой кнопкой мыши и выбрал пункт «**Очистить корзину**» (“Empty Recycle Bin”) в контекстном меню).

Корзина в ОС Windows Vista

Еще один аспект операционной системы Windows, изменившийся с появлением ОС Vista, – основная структура для способа реализации корзины. Хотя это изменение не заметно для пользователя, оно предоставляет очень полезный источник для судебного эксперта, о чём пишет Митчел Макор (Mitchell Machor) в своей статье «The Forensic Analysis of the Microsoft Windows Vista Recycle Bin» (www.forensicfocus.com/downloads/forensic-analysis-vista-recycle-bin.pdf). Как и в предыдущих версиях Windows, файлы, удаляемые пользователем, по-прежнему связаны с идентификатором SID пользователя, но теперь они находятся в каталоге C:\\$Recycle.Bin. Отличие при обработке удаленных файлов в ОС Vista состоит в том, что удаленому файлу присваивается имя, начинающееся с символов «\$R», за которыми следуют шесть случайных символов, а затем – исходное расширение файла. Кроме того, создается второй файл с таким же именем (но с символами «\$I» вместо «\$R»), который содержит сведения, похожие на те, что находятся в файле INFO2. Однако в этом «индексном» файле в корзине ОС Vista хранится только исходное имя файла, исходный размер файла, а также дата и время удаления файла.

Точки восстановления системы в Windows XP

Мы обсуждали файлы реестра, хранящиеся в точках восстановления системы Windows XP, в главе 4. В этой главе мы рассмотрим другие файлы журналов, хранящиеся в этих точках восстановления.

Файлы «rp.log»

«Rp.log» – это файл журнала, находящийся в каталоге точки восстановления (RPxx). Этот журнал точки восстановления содержит значение, указывающее на тип точки восстановления, описательное имя для события, во время которого была создана точка восстановления (т. е. установка приложения или драйвера устройства, удаление приложения и т. п.), и 64-разрядное значение структуры *FILETIME*, указывающее время создания точки восстановления. Тип точки восстановления – это 4-байтовое значение (двойное слово), начинающееся в четвертом байте файла. Описание точки восстановления – строка в формате Unicode с завершающим нулем, которая начинается по смещению 16 (0x10) относительно начала файла, а дата и время создания – это 8-байтовое значение (четверное слово), расположенное по смещению 528 (0x210) относительно начала файла.

Для того чтобы собрать информацию о точках восстановления, можно на работающем компьютере запустить Perl-скрипт «sr.pl» (расположенный на носителе, который идет в комплекте с этой книгой; это тот же Perl-скрипт «sr.pl», что рассматривался в главе 4). Скрипт реализовывает класс WMI *SystemRestore*, чтобы получить доступ к значениям *RestorePointType*, *Description* и *CreationTime* для каждой точки восстановления и показывает их пользователю.

Perl-скрипт «sysrestore.pl» (расположенный на носителе, который идет в комплекте с этой книгой) – это скрипт типа ProScript, который можно использовать в программе ProDiscover, чтобы получить информацию из файлов «rp.log», находящихся в каталогах точек восстановления в образе данных из Windows XP (который открывается в ProDiscover). Скрипт открывает файл «rp.log» в каждом каталоге и извлекает описание и дату создания точки восстановления.

Описание точки восстановления может быть полезным для эксперта, особенно если он ищет информацию об установке или удалении приложения. Точки восстановления системы создаются при установке приложений и неподписанных драйверов, при установке обновлений и при выполнении операций восстановления. Точки восстановления можно также создать вручную.

При создании точки восстановления описание события, вызвавшего создание точки, записывается в файл «*gr.log*». Часто вы увидите описание *System Checkpoint*, которое относится к точке восстановления, создаваемой ОС Windows XP каждые 24 часа (параметр по умолчанию). Описание *Software Distribution Service* относится к установке обновлений Windows. Мне также встречались в системах такие описания, как *Installed QuickTime*, *Removed ProDiscover 4.8a* и *Installed Windows Media Player 11*. Описание может подсказать эксперту дату установки или удаления определенного приложения.

Дата создания точки восстановления может также помочь эксперту в других случаях. Она не только добавляет информацию к временной шкале действий в системе, но и помогает эксперту определить, было ли изменено системное время. Если последовательные точки восстановления (последовательные исходя из номера точки, например RP80, RP81, RP82 и т. д.) имеют непоследовательные даты создания, это может свидетельствовать о том, что кто-то изменял системное время.

Файлы «*change.log.x*»

После того как точка восстановления создана, продолжается наблюдение за важными файлами системы и приложений, чтобы систему можно было восстановить в определенное состояние. Изменения в файлах регистрируются, и при необходимости сохраняется весь файл, чтобы систему можно было восстановить. Эти изменения записываются в файлах «*change.log*», которые находятся в каталогах точек восстановления. По мере того как в наблюдаемых файлах обнаруживаются изменения, исходное имя файла записывается в файл «*change.log*» вместе с порядковым номером и другой необходимой информацией, такой как тип произошедшего изменения (удаление файла, изменение атрибутов файла или изменения содержимого). Если наблюдаемый файл необходимо сохранить (например, в случае операции удаления), он копируется в каталог точки восстановления и переименовывается в формат *Axxxxxx.ext*, где *x* представляет последовательный номер, а *.ext* – исходное расширение файла.

При перезапуске системы к имени файла «*change.log*» добавляется порядковый номер (имя файла «*change.log*» изменяется на «*change.log.1*»), и создается новый файл «*change.log*». Однако вы не найдете файл «*change.log*» в каталогах точек восстановления; вместо этого вы увидите несколько файлов с именем типа «*change.log.x*», где *x* – это номер файла «*change.log*».

Каждый файл «*change.log.x*» состоит из нескольких записей журнала изменений. Мне удалось найти веб-сайт, содержащий подробную информацию о двоичном формате этих записей (включая сведения о «магическом» числе 0xABCD12, используемом для распознавания записей журнала изменений в свободном пространстве накопителя). Используя информацию с этого сайта, я смог создать Perl-скрипт, анализирующий и интерпретирующий содержимое файлов «*change.log.x*». Perl-скрипт «*lscl.pl*» (аббревиатура от англ. *LiSt Change Log – вывести данные журнала изменений*) находится на DVD-диске, который идет в комплекте с этой книгой.

Совет

Файл «*fifo.log*» – еще один журнал, сохраняемый функцией восстановления системы и расположенный в корне каталога точек восстановления. Когда дисковое пространство, резервируемое для точек восстановления, заполняется на 90%, функция восстановления начнет удалять самые старые точки, пока пространство, занимаемое точками восстановления, не сократится до 75% от максимального размера (параметр по умолчанию или значение, указанное пользователем). В файле «*fifo.log*» сохраняется список точек восстановления, удаленных с наблюдаемого накопителя, а также дата и время их удаления. Точки восстановления хранятся 90 дней; по истечении этого срока они удаляются.

Служба теневого копирования томов в ОС Windows Vista

В Windows Vista используется функция, похожая на системные точки восстановления в Windows XP; эта функция называется «Теневое копирование томов» (“Volume Shadow Copy”) (<http://technet.microsoft.com/en-us/library/cc785914.aspx>). Как и точки восстановления Windows XP, теневые копии тома сохраняются в каталоге «System Volume Information»; эта функция включена в Vista по умолчанию. Обычно теневые копии тома создаются во время начальной загрузки системы, но они также могут создаваться в других случаях. Как и в случае с точками восстановления XP, теневые копии тома в ОС Windows Vista могут содержать множество информации, представляющей интерес для судебного эксперта. Однако, в отличие от точек восстановления XP, доступ к теневым копиям томов, очевидно, можно получить только на работающем компьютере с ОС Vista, используя утилиту «vssadmin.exe». Дополнительную информацию о загрузке образа данных ОС Windows см. в разделе «Альтернативные методы анализа» этой главы.

Совет

Кристофер Харгривз (Christopher Hargreaves) и Говард Чиверс (Howard Chivers) написали статью «Potential Impacts of Windows Vista on Digital Investigations» (www.forensicfocus.com/downloads/potential-impact-windows-vista.pdf), расширенная версия которой доступна в журнале *Journal of Digital Investigation*. В статье они описывают способ получения доступа к данным, хранящимся в теневых копиях тома, похожий на способ, описанный в этом разделе. Помимо способа, рассматриваемого в этом разделе, можно использовать программу ShadowExplorer (www.shadowexplorer.com), чтобы получить доступ к данным, а также к файлам и папкам в теневых копиях тома на работающем компьютере с ОС Windows Vista (и Windows 2003, если эта функция включена).

С помощью утилиты «vssadmin.exe» на работающем компьютере с ОС Windows Vista можно использовать следующую команду, чтобы вывести список имеющихся теневых копий тома:

```
C:\>vssadmin list shadows /for=c:\
```

Получив список теневых копий тома, можно создать символьную ссылку на любую теневую копию с помощью утилиты «mklink.exe», как показано ниже (где *n* – это номер распознанной теневой копии тома):

```
C:\>mklink /d C:\Voln \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopyn
```

На этом этапе файлы теневых копий тома будут доступны через C:\Voln (например, C:\Vol3 будет символьной ссылкой на HarddiskVolumeShadowCopy3). Помимо этого процесса, можно использовать утилиту «dd.exe», чтобы создать образ отдельной теневой копии тома, о чем упоминает Роб Ли (Rob Lee) в блоге SANS Forensic (<http://sansforensics.wordpress.com/2008/10/10/shadow-forensics/>). Роб пишет, что для создания образа теневой копии тома можно использовать следующую команду, запущенную с USB-накопителя:

```
F:\>dd.exe if=\\.\\HarddiskVolumeShadowCopy4 of=f:\\snapshot4.img --localwrt
```

Утилиту «dd.exe» можно найти на сайте <http://gmgsystemsinc.com/fau/>.

Специалисты по расследованию инцидентов и эксперты должны знать о том, что доступ к теневым копиям тома, очевидно, можно получить только на работающем компьютере с ОС Windows Vista, чтобы предпринять необходимые меры для сохранения

данных. К таким мерам относится создание образа данных теневых копий тома с работающих компьютеров или использование имени пользователя и пароля, необходимых для получения доступа к образу данных, который был загружен на компьютере, чтобы обратиться к теневым копиям тома.

Файлы упреждающей выборки

Начиная с ОС Windows XP, в операционных системах Microsoft используется так называемая упреждающая выборка данных для улучшения производительности системы. В ОС Windows XP, 2003 и Vista по умолчанию выполняется упреждающая выборка при загрузке системы, а в XP и Vista также осуществляется упреждающая выборка при запуске приложений.

Чтобы обеспечить упреждающую выборку для загрузки системы, диспетчер кэша наблюдает за ошибками страниц физической памяти (требующих чтение данных с накопителя) и ошибками программной страницы (требующих добавления данных, находящихся в памяти, к рабочему набору процесса); наблюдение проводится в течение первых двух минут процесса загрузки, первой минуты после запуска всех служб Windows или первых 30 секунд после запуска пользовательской оболочки (в зависимости от того, какое из этих трех событий произойдет первым). Данные об ошибках обрабатываются вместе с ссылками на файлы и каталоги, к которым осуществляется доступ, что в итоге позволяет получить доступ ко всем этим данным из отдельного файла, а не из разных файлов и каталогов, разбросанных по всему накопителю. Это в свою очередь уменьшает количество времени, необходимого для загрузки системы.

Во время упреждающей выборки для приложения диспетчер кэша выполняет наблюдение в течение первых 10 секунд после запуска процесса. После того как эти данные обработаны, они записываются в файл с расширением .pf в каталоге Windows\Prefetch. Имя файла создается с использованием имени приложения, за которым следует тире и шестнадцатеричное представление хэша для пути к приложению. Таким образом, для одной и той же программы, запущенной из разных мест, будут созданы разные pf-файлы. Например, в ОС Windows XP будет создано два разных pf-файла, когда программа «Блокнот» («Notepad») будет запущена из каталога C:\Windows и из каталога C:\Windows\system32. (Почему-то в ОС Windows XP есть копия программы «Блокнот» в каждом из этих каталогов.)

Функция упреждающей выборки управляет с помощью следующего раздела реестра:

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet00x\Control\Session Manager\Memory Management\PrefetchParameters
```

В этом разделе есть параметр с именем *EnablePrefetcher*. Данные, связанные с этим параметром, покажут вам, какой тип упреждающей выборки использует система:

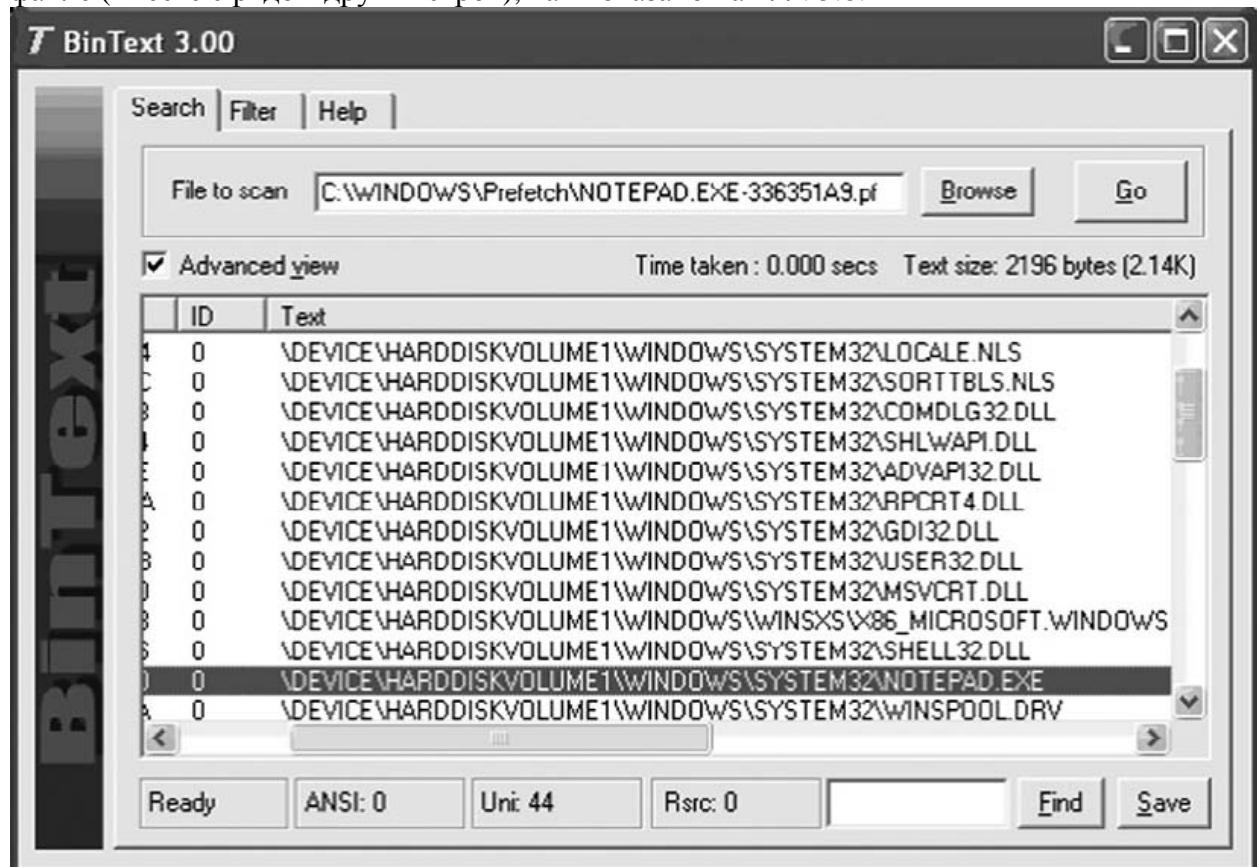
- § 0: упреждающая выборка отключена;
- § 1: упреждающая выборка для приложений включена;
- § 2: упреждающая выборка для загрузки системы включена;
- § 3: упреждающая выборка для запуска приложений и загрузки системы включена.

В ОС Windows XP и Vista значение по умолчанию для параметра *EnablePrefetcher* – 3; это значение равно 2 в ОС Windows 2003. Интересно, что упреждающая выборка для приложений в ОС Windows XP имеет ограничение на хранение 128 pf-файлов.

Некоторая информация в pf-файле в каталоге «Prefetch» может оказаться чрезвычайно полезной для эксперта. По смещению 144 относительно начала файла находится 4-байтовое значение, соответствующее числу запусков приложения. По смещению 120 относительно начала файла находится 64-разрядное значение структуры

FILETIME, соответствующее времени последнего запуска приложения. Это значение хранится в формате UTC, который аналогичен времени по Гринвичу. Perl-скрипт «prefetch.pl» на DVD-диске, который идет в комплекте с этой книгой, – это скрипт типа ProScript, анализирующий pf-файлы в каталоге «Prefetch» и извлекающий из них число запусков и время последнего запуска приложений. Perl-скрипт «pref.pl» (скомпилированная исполняемая версия скрипта также доступна на сопроводительном DVD-диске) анализирует каталог «Prefetch» на работающем компьютере и извлекает отметки времени MAC (дополнительную информацию об этих отметках см. в следующем разделе) и время последнего запуска из pf-файлов, отправляя выходные данные на консоль в формате значений, разделенных запятыми (подходящем для открытия в программе Excel).

Путь к приложению, которое было запущено, сохраняется в Unicode-строке в pf-файле (вместе с рядом других строк), как показано на илл. 5.8.



Илл. 5.8. Пример пути к приложению в pf-файле.

Можно сопоставить различные данные из pf-файла и информацию из реестра (см. главу 4) или журнала событий, чтобы определить, кто входил в систему, какие приложения запускались, и кто запускал эти приложения. Одно из преимуществ такого сопоставления состоит в том, что, если пользователь устанавливает и запускает приложение, а затем удаляет его, сведения об этом приложении могут остаться в каталоге «Prefetch». Когда я разговаривал с сотрудниками правоохранительных органов о проблемах, связанных с применением стеганографических приложений в интернет-преступлениях, все они говорили, что обычно ищут скрытую информацию, только если что-то указывает на то, что подобное приложение использовалось. Существование pf-файла с именем определенного приложения может быть таким признаком.

Функция SuperFetch в OC Windows Vista

В состав Windows Vista включена функция упреждающей выборки, которая называется SuperFetch и создает файлы, похожие на данные упреждающей выборки для

приложений в Windows XP. Однако смещения для различных метаданных в этих файлах немного отличаются от смещений, используемых в файлах упреждающей выборки для приложений в Windows XP. Perl-скрипт «vista_pref.pl», а также соответствующий скомпилированный исполняемый скрипт могут извлекать дату последнего запуска приложения из файла упреждающей выборки в ОС Windows Vista (оба скрипта доступны на носителе, который идет в комплекте с этой книгой).

Файлы ярлыков

Файлы ярлыков могут оказаться полезными во время экспертизы. Подумайте о том, как ярлыки (файлы с расширением .lnk) создаются и как к ним осуществляется доступ в обычном повседневном использовании. Пользователь получает доступ к документу на накопителе, запоминающем устройстве или сетевом ресурсе, а ярлык создается в папке «Недавние документы» (“Recent”) («Недавние документы» – это скрытая папка в каталоге профиля пользователя). Ярлыки могут предоставить информацию о файлах (или сетевых ресурсах), к которым пользователь получал доступ, а также об устройствах, которые пользователь, возможно, подключал к компьютеру в определенный момент времени. Некоторые коммерческие инструменты для судебного анализа данных, такие как Forensic Toolkit (FTK) от компании AccessData и EnCase от компании Guidance Software, предоставляют возможность проанализировать содержимое lnk-файлов, чтобы показать встроенную в них информацию. Также можно использовать бесплатную программу Windows File Analyzer (WFA) от компании MiTeC, чтобы проанализировать информацию в lnk-файле. Недавно Джесси Хагер (Jesse Hager) опубликовал документ «The Windows Shortcut File Format», в котором задокументированы смещения и размеры различных компонентов файла ярлыка. Натан Вайлбахер (Nathan Weilbacher) написал статью (www.forensicfocus.com/link-file-evidentiary-value) для сайта ForensicFocus.com, в которой он ссылается на документ Дж. Хагера и детально описывает доказательственную ценность файлов ярлыков Windows.

Perl-скрипт «lslnk.pl» (находящийся на DVD-носителе, который идет в комплекте с этой книгой) учитывает большую часть сведений из документа Дж. Хагера и позволяет эксперту просмотреть внутреннее содержимое файлов ярлыков Windows, отображая такую информацию, как отметки времени MAC целевого файла, параметры различных флагов и атрибутов, а также сведения о локальном томе, пример которых показан ниже:

```
Shortcut file is on a local volume.
Volume Name = C-DISK
Volume Type = Fixed
Volume SN = 0x303d30de
```

Если целевой файл находится на сетевом ресурсе, скрипт «lslnk.pl» извлекает путь к этому ресурсу в формате, показанном ниже:

```
File is on a network share.
Network Share name = \\192.168.1.22\c$ Z:
```

Скрипт «lslnk.pl» открывает файл ярлыка в двоичном режиме и анализирует содержимое, не используя интерфейс Windows API. Этот скрипт можно использовать в любой системе, поддерживающей язык Perl. Джейк Каннингэм (Jake Cunningham) написал похожий Perl-скрипт, который называется «lnk-parse.pl» и доступен на веб-сайте JAFAT (<http://jafat.sourceforge.net/files.html>).

Метаданные файлов

Термин *метаданные* означает «данные о данных». Наиболее известные метаданные о файлах в ОС Windows – это отметки времени MAC; в данном случае аббревиатура MAC означает время изменения (*modified*), доступа (*accessed*) и создания (*created*). Отметки времени MAC относятся к времени последнего изменения файла (когда данные были добавлены в файл или удалены из него), времени последнего доступа (когда файл был открыт последний раз) и времени первоначального создания. Способ того, как операционная система управляет этими отметками времени, зависит от используемой файловой системы. Например, в файловой системе FAT отметки времени сохраняются на основе местного времени компьютерной системы, а в файловой системе NTFS отметки времени MAC сохраняются в формате UTC, который аналогичен времени по Гринвичу. Когда приложения, например, проводник Windows, отображают отметки времени MAC, нужно принимать во внимание параметры часового пояса и перехода на летнее время. Более того, в файловой системе FAT время создания записывается с точностью до 10 миллисекунд, время изменения – до 2 секунд, а время последнего доступа – до 1 дня (в действительности записывается дата последнего доступа, что нельзя назвать подробными сведениями). В файловой системе NTFS время последнего доступа записывается с точностью до 1 часа.

Предупреждение

В ОС Windows параметр реестра *NtfsDisableLastAccessUpdate* (находящийся в разделе `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\FileSystem`) позволяет отключить обновление времени последнего доступа в операционной системе (для этого параметр *DWORD* нужно установить в значение «1»). Хотя такая настройка рекомендуется для высокопроизводительных серверов (чтобы оптимизировать быстродействие и увеличить общее время отклика), она может усложнить работу судебного эксперта, особенно когда определение времени доступа к файлам является важной частью дела. Это значение можно установить с помощью команды *fsutil* в Windows XP и 2003, и оно установлено (т. е. обновление времени последнего доступа отключено) по умолчанию в Windows Vista. Это означает, что судебному эксперту нужно разработать дополнительные способы и методы анализа и опираться на другие источники данных для исследования.

Еще один момент, который может быть интересен экспертам, – способ отображения отметок времени MAC (<http://support.microsoft.com/?kbid=299648>) для файлов и каталогов в зависимости от различных действий перемещения и копирования.

В файловой системе FAT16:

- § **Копируйте «myfile.txt» из каталога C:\ в C:\subdir** В файле «myfile.txt» сохраняется та же дата изменения, а дата создания обновляется до текущих даты и времени.
- § **Переместите «myfile.txt» из каталога C:\ в C:\subdir** В файле «myfile.txt» сохраняются те же даты изменения и создания.
- § **Копируйте «myfile.txt» из раздела FAT16 в раздел NTFS** В файле «myfile.txt» сохраняется та же дата изменения, а дата создания обновляется до текущих даты и времени.
- § **Переместите «myfile.txt» из раздела FAT16 в раздел NTFS** В файле «myfile.txt» сохраняются те же даты изменения и создания.

В файловой системе NTFS:

- § **Копируйте «myfile.txt» из каталога C:\ в C:\subdir** В файле «myfile.txt» сохраняется та же дата изменения, а дата создания обновляется до текущих даты и времени.

§ Переместите «myfile.txt» из каталога C:\ в C:\subdir В файле «myfile.txt» сохраняются те же даты изменения и создания.

Короче говоря, независимо от используемой файловой системы, если файл копируется, дата создания обновляется до текущих даты и времени; если файл перемещается, дата создания не изменяется. Дата изменения обновляется, когда в файл вносятся изменения.

Записки из подполья...

Изменение отметок времени MAC

Каким бы полезными не были отметки времени MAC для расследования, нужно иметь в виду, что есть люди, которые могут активно пытаться скрыть данные в системе, изменяя отметки времени MAC в файлах. Я демонстрировал на конференциях применение инструментов, позволяющих пользователю изменять отметки времени MAC, используя Perl-скрипты для получения доступа к необходимым и (полностью документированным) функциям Windows API, чтобы сначала создать файл, затем изменить дату создания на шесть лет вперед, а дату изменения – на два года назад. Такие действия могут запутать расследование, и как после этого можно доверять *каким-либо* отметкам времени MAC?

Но и это еще не все. На веб-сайте Metasploit Project есть раздел Anti-Forensics Project (www.metasploit.org/research/projects/antiforensics/), содержащий инструмент «timestomp.exe», который позволяет злоумышленнику изменять не только отметки времени MAC файла, но и отметку даты/времени изменения записи, указывающую на время изменения атрибутов файла. Надеюсь, что, дойдя до этого места в книге, вы поняли, что антикриминалистические средства предназначены для того, чтобы сбить с толку эксперта, а не отдельное приложение для судебного анализа данных.

В оставшейся части этого раздела рассматриваются метаданные, встроенные в различные форматы файлов.

Документы Word

Метаданные, содержащиеся в документах Word, давно являются предметом обсуждения. Документы Word – это составные документы, основанные на технологии связывания и внедрения объектов (OLE), которая определяет «структуру файла внутри файла». Помимо сведений о форматировании, документы Word могут содержать довольно много дополнительной информации, которая не видна пользователю, в зависимости от пользовательского представления документа. Например, документ может хранить не только данные о последних исправлениях, но и список последних 10 авторов, редактировавших документ. Таким образом, существует вероятность раскрытия информации о пользователях или организациях. Это наглядно показал Ричард М. Смит (Richard M. Smith) в середине 2003 года на примере документа, опубликованного премьер-министром Великобритании Тони Блэром (www.computerbytesman.com/privacy/blair.htm). В феврале 2003 года правительство Блэра разместило в Интернете документ об иракских организациях разведки и безопасности. Преподаватель Кембриджского университета определил, что часть содержимого этого документа была первоначально написана американским исследователем в Ираке. Это заставило некоторых людей внимательнее изучить этот документ. Обсуждая проблему раскрытия информации, преподаватель продемонстрировал данные, которые он смог извлечь из этого документа Word, содержащие список последних 10 авторов, редактировавших документ. Эта информация поставила в неловкое положение аппарат премьер-министра Тони Блэра.

В статье на своем веб-сайте преподаватель упомянул о программе, которую он создал, чтобы извлечь эту информацию из документа Word, однако эта программа не предоставлена для публичного использования. Я написал Perl-скрипт «wmd.pl», который включен в состав сопроводительного DVD-диска, анализирующий двоичный заголовок

документа Word и извлекающий из него некоторую информацию. Для того чтобы извлечь информацию, скрипт использует Perl-модули (скрипт не использует интерфейс API Microsoft Word, поэтому его можно запускать в любых системах, поддерживающих Perl и имеющих необходимые установленные модули, как указано в прагмах *use* для скрипта). Выходные данные скрипта, примененного к документу Блэра, выглядят так:

```
-----
Statistics
-----
File      = g:\book2\ch5\blair.doc
Size      = 65024 bytes
Magic     = 0xa5ec (Word 8.0)
Version   = 193
LangID    = English (US)
Document was created on Windows.
Magic Created : MS Word 97
Magic Revised : MS Word 97
-----
Last Author(s) Info
-----
1 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
2 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
3 : cic22 : C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
4 : JPratt : C:\TEMP\Iraq - security.doc
5 : JPratt : A:\Iraq - security.doc
6 : ablackshaw : C:\ABlackshaw\Iraq - security.doc
7 : ablackshaw : C:\ABlackshaw\A;Iraq - security.doc
8 : ablackshaw : A:\Iraq - security.doc
9 : MKhan : C:\TEMP\Iraq - security.doc
10 : MKhan : C:\WINNT\Profiles\mkhan\Desktop\Iraq.doc
-----
Summary Information
-----
Title      : Iraq- ITS INFRASTRUCTURE OF CONCEALMENT, DECEPTION AND
INTIMIDATION
Subject    :
Authress   : default
LastAuth   : MKhan
RevNum    : 4
AppName   : Microsoft Word 8.0
Created    : 03.02.2003, 09:31:00
Last Saved : 03.02.2003, 11:18:00
Last Printed : 30.01.2003, 21:33:00
-----
Document Summary Information
-----
Organization : default
```

Как видите некоторые данные, «скрытые» в документах Word, могут быть довольно компрометирующими и неприятными. Помимо последних 10 авторов, скрипт показывает, на какой платформе (Windows или Mac) был создан документ, а также какая версия программы Word использовалась для создания и последующего редактирования документа. Скрипт также извлекает сводные сведения из документа (которые будут рассмотрены позднее в разделе «Альтернативные потоки данных в NTFS» в этой главе).

В состав DVD-диска, который идет в комплекте с этой книгой, также входит небольшая утилита «oledmp.pl». Она использует те же Perl-модули, что и «wmd.pl», но выполняет несколько другую функцию. «Oledmp.pl» показывает OLE-потоки и корзины,

встроенные в документ Word, а также те же сводные сведения, что извлекает скрипт «wmd.pl», как показано в следующем примере выходных данных:

```
C:\Perl>oledmp.pl blair.doc
ListStreams
Stream : *CompObj
Stream : WordDocument
Stream : *DocumentSummaryInformation
Stream : ObjectPool
Stream : 1Table
Stream : *SummaryInformation
Trash Bin           Size
BigBlocks          0
SystemSpace        940
SmallBlocks        0
FileEndSpace       1450
Summary Information
subject
lastauth           MKhan
lastprinted        30.01.2003, 21:33:00
appname            Microsoft Word 8.0
created             03.02.2003, 09:31:00
lastsaved           03.02.2003, 11:18:00
revnum              4
title               Iraq- ITS INFRASTRUCTURE OF CONCEALMENT, DECEPTION AND
                    INTIMIDATION
authress            default
1Table
1 cic22           C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
2 cic22           C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
3 cic22           C:\DOCUME~1\phamill\LOCALS~1\Temp\AutoRecovery save of Iraq -
security.asd
4 JPratt           C:\TEMP\Iraq - security.doc
5 JPratt           A:\Iraq - security.doc
6 ablackshaw       C:\ABlackshaw\Iraq - security.doc
7 ablackshaw       C:\ABlackshaw\A\Iraq - security.doc
8 ablackshaw       A:\Iraq - security.doc
9 MKhan            C:\TEMP\Iraq - security.doc
10 MKhan           C:\WINNT\Profiles\mkhan\Desktop\Iraq.doc
```

В данных *ListStreams* показаны имена различных OLE-потоков, составляющих документ Word. Microsoft называет технологию OLE «файловой системой внутри файла», а эти имена потоков называет «файлами» в документе.

Предупреждение

Иногда просто поражаешься тому, сколько информации можно обнаружить в метаданных документа Word. Попробуйте провести небольшой эксперимент: найдите на работающем файловом сервере (конечно, имея права доступа) какие-нибудь документы Word, например, те, что обычно рассылают клиентам, и посмотрите, что скрытые метаданные «рассказывают» о документе. Я попытался сделать что-то похожее, только использовал Google вместо корпоративного файлового сервера. Из-за большого числа полученных ответов я ограничил поиск доменами .mil и .gov, но все равно нашел больше документов, чем мне было нужно.

Интересно, что когда я писал свою первую книгу, один из технических рецензентов не хотел, чтобы я знал, как его зовут, и специально попросил, чтобы издательство не сообщало мне никакой информации о нем. Более того, когда этот рецензент заполнил бланки рецензии в документах Word, он сохранил содержимое как обычный текстовый

документ в кодировке ASCII, удалив таким образом все метаданные. Полагаю, он действительно не хотел, чтобы я знал, кто он!

Эти метаданные могут не только представлять угрозу раскрытия информации о пользователе или организации, но также могут быть полезны эксперту, который ищет отдельные сведения о документах. Это очень важно в делах, связанных с представлением доказательств в электронной форме, особенно если поиск ключевых слов или фраз ограничен видимым текстом документов.

Для полноты картины нужно упомянуть еще несколько моментов, прежде чем переходить к рассмотрению следующей темы. Во-первых, Microsoft предоставляет пользователям информацию о метаданных в документе Word и способах уменьшить количество доступных метаданных. Во-вторых, документы Word – не единственные файлы пакета Microsoft Office, имеющие проблемы с метаданными. Для решения этих вопросов Microsoft предлагает ознакомиться со следующими статьями из базы знаний:

- § № 223790: WD97: «How to Minimize Metadata in Microsoft Word Documents»;
- § № 223396: OFF: «How to Minimize Metadata in Microsoft Office Documents»;
- § 223789: XL: «How to Minimize Metadata in Microsoft Excel Workbooks»;
- § № 223793: PPT97: «How to Minimize Metadata in Microsoft PowerPoint Presentations»;
- § № 290945: «How to Minimize Metadata in Word 2002»;
- § № 825576: «Удаление метаданных из документов в Word 2003» («How to Minimize Metadata in Word 2003»).

Помимо этих статей из базы знаний, Microsoft также предоставляет средство удаления скрытых данных (<http://support.microsoft.com/kb/834427>) в виде надстройки к Office 2003 и XP. Авторы могут использовать это средство, чтобы удалить множество метаданных из документов. Это отличный инструмент, гарантирующий, что количество доступных метаданных сведено к минимуму, даже если вы будете сохранять созданный документ в другом формате, например, в PDF.

Записки из подполья...

Программа Merge Streams

Программа Merge Streams (www.ntkernel.com/w&p.php?id=23), доступная на сайте NT Kernel Resources, реализовывает интересный аспект OLE-документов пакета Microsoft Office. Вкратце, она позволяет объединить таблицу Excel и документ Word в один файл. Программа имеет простой графический интерфейс, через который можно выбрать документ Word и таблицу Excel и объединить их. Предположим, у вас в каталоге есть каждый из этих документов. Запустив эту программу и объединив два документа, вы получите документ Word, размер которого больше, чем размер исходного документа Word, а также больше, чем размер исходной таблицы Excel. Однако если бы вы удалили таблицу Excel, изменили расширение документа Word на .xls, а затем дважды щелкнули по файлу, на рабочем столе открылась бы таблица Excel без каких-либо данных исходного документа Word или его содержимого. Изменив расширение файла обратно на .doc, вы сможете открыть документ Word без каких-либо видимых данных таблицы Excel.

Рассматривая эту тему на конференциях, я обычно показываю, как работает данная программа. Чаще всего я демонстрирую это с точки зрения корпоративного пользователя, который пытается тайно вынести из организации таблицу финансовых прогнозов или контактную информацию, относящуюся к важному тендеру. Все, что нужно сделать пользователю – объединить таблицу Excel с документом Word (содержащим какую-нибудь безвредную информацию, например, текст письма), а затем копировать документ Word на флеш-накопитель. Если кто-нибудь остановит пользователя на выходе и проверит содержимое флеш-накопителя, то увидит только документ Word.

Разговаривая с сотрудниками правоохранительных органов, я использую несколько

другой подход. Предположим, что у сотрудника компании есть незаконные изображения, которыми он хотел бы поделиться со своими друзьями. Он копирует изображения в документ Word, затем находит таблицу Excel на файловом сервере, к которому все сотрудники могут (и имеют законную потребность) получить доступ, и объединяет файлы. Затем он изменяет расширение документа Word на расширение таблицы и сообщает своим друзьям о том, что сделал. Таким образом он может распространять изображения, не оставляя следов.

Обнаружить использование такой программы, как Merge Streams, вовсе не является сверхсложной задачей. Используя скрипты, которые имеют функциональные возможности, подобные скрипту «oledmp.pl», упоминавшемуся ранее в этой главе, можно вывести список OLE-потоков, составляющих документ Word. Если вы увидите имена потоков (Workbook, Worksheet и т. д.), свидетельствующие о наличии таблицы Excel, то этот документ Word определенно стоит исследовать.

Совет

Perl-скрипт «oledmp.pl» чрезвычайно эффективен во время исследований, связанных с таблицами Excel и презентациями PowerPoint. Однажды я производил экспертизу компьютера, с которого, как подозревал клиент, совершалось мошенничество; при этом использовались номера счетов, к которым сотрудник имел доступ как часть повседневного круга обязанностей. Используя список ключевых слов, созданный с помощью клиента, я нашел на компьютере таблицу Excel, извлек ее из образа и предоставил ее клиенту для изучения. В свой отчет я смог включить информацию о том, откуда возник файл (согласно местонахождению файла, это было вложение Outlook), когда пользователь получил доступ к файлу (основываясь на данных, найденных в реестре), а также о том, что пользователь редактировал, а затем распечатал таблицу. Сведения о последних двух фактах были получены из метаданных таблицы с помощью скрипта «oledmp.pl».

Кори Алтейд (Cory Althiede) недавно обратил мое внимание на еще одно средство извлечения потенциально полезной информации из файлов Microsoft Word и других OLE-документов. Создавая рукопись этой книги, я выделял текст в файле, копировал его в буфер обмена, вставлял текст в документ, над которым я работал, а затем правильно форматировал этот текст. Однако если текст перетаскивается в документ Microsoft Word мышью, он становится вложением. Если вам нужно извлечь эти вложения OLE-документа, Кори советует использовать отличный инструмент, который называется b2xtranslator (<http://b2xtranslator.sourceforge.net/>). Согласно справочному разделу веб-сайта, цель этого инструмента – позволить пользователю осуществить переход с формата двоичных документов на новый формат XML/zip, используемый в последних версиях пакета Microsoft Office (например, перейти с формата .doc на формат .docx). Страница документации, на которую можно перейти с главной страницы сайта, содержит информацию об основных этапах работы инструмента и о том, как получить доступ к различным OLE-объектам, вложенным в документ Word или таблицу Excel. Если вам нужно сделать что-то большее, чем просто узнать последнего автора или дату распечатывания OLE-документа, стоит подумать об использовании этого инструмента.

PDF-документы

Файлы формата PDF также могут содержать такие метаданные, как имя автора, дату создания файла и название приложения, использовавшегося для создания PDF-файла. Часто метаданные показывают, что PDF-файл был создан на компьютере Mac или путем преобразования документа Word в формат PDF. Как и в случае с документами Word, эти метаданные могут представлять угрозу раскрытия информации. Однако, в зависимости от ситуации, эта информация может быть также полезна для эксперта, например, чтобы

помочь представить доказательство в электронной форме или чтобы показать, что определенное приложение было установлено на компьютере пользователя.

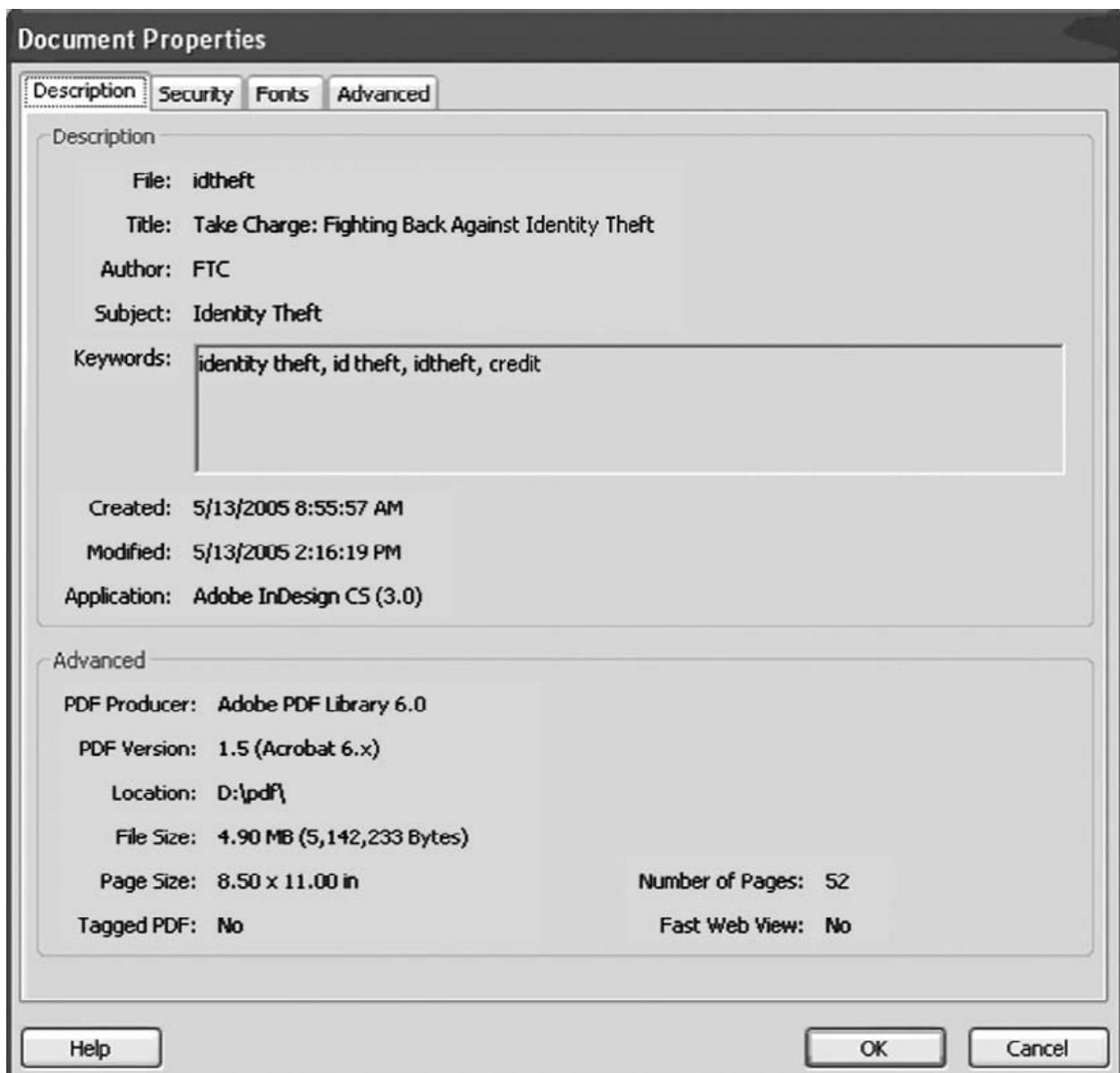
В состав сопроводительного DVD-диска включены два Perl-скрипта («pdfmeta.pl» и «pdfdump.pl»), которые я использую для извлечения метаданных из PDF-файлов. Единственное различие между ними состоит в том, что они используют разные Perl-модули для взаимодействия с PDF-файлами. Если честно, я использовал эти скрипты с переменным успехом; в некоторых случаях оба скрипта успешно извлекают метаданные из PDF-файла, тогда как в других случаях тому или иному скрипту не удавалось сделать это по какой-либо причине. В качестве теста я использовал Google для поиска нескольких примерных PDF-файлов и нашел два: один с сайта FTC, другой с сайта IRS. PDF-файл с сайта FTC назывался «idtheft.pdf», и скрипт «pdfmeta.pl» возвратил следующую информацию:

```
C:\Perl>pdfmeta.pl d:\pdf\idtheft.pdf
Author          FTC
CreationDate   D:20050513135557Z
Creator         Adobe InDesign CS (3.0)
Keywords        identity theft, id theft, idtheft, credit
ModDate        D:20050513151619-04'00'
Producer       Adobe PDF Library 6.0
Subject         Identity Theft
Title          Take Charge: Fighting Back Against Identity Theft
```

PDF-файл, загруженный с сайта IRS, был копией налоговой формы W-4 за 2006 год и назывался «fw4.pdf». Скрипт «pdfmeta.pl» возвратил следующую информацию:

```
C:\Perl>pdfmeta.pl d:\pdf\fw4.pdf
Author          SE:W:CAR:MP
CreationDate   D:20051208083254-05'00'
Creator         OneForm Designer Plus
Keywords        Fillable
ModDate        D:20060721144654-04'00'
Producer       APJavaScript 2.2.1 Windows SPDF_1112 Oct 3 2005
Subject         Employee's Withholding Allowance Certificate
Title          2006 Form W-4
```

Оба эти примера довольно безобидны, но они позволяют легко увидеть, как метаданные в PDF-файлах можно использовать при представлении доказательств в электронной форме, и понять, что метаданные следует учитывать при проведении поиска по ключевым словам. Если у вас не получается извлечь метаданные с помощью обоих Perl-скриптов, предоставленных с этой книгой, можно использовать старый проверенный способ – открыть файл в программе Adobe Reader (доступной на сайте Adobe.com) и в меню «Файл» (“File”) выбрать пункт «Свойства документа» (“Document Properties”). Вкладка «Описание» (“Description”) в диалоговом окне «Свойства документа» (“Document Properties”) содержит все имеющиеся метаданные. На илл. 5.9 показаны свойства документа для файла «idtheft.pdf».



Илл. 5.9. Свойства документа «idtheft.pdf».

Осенью 2008 года Дидье Стивенс (Didier Stevens) разработал инструмент «pdf-parser.py» на языке Python. Инструмент доступен по адресу <http://blog.didierstevens.com/programs/pdf-tools/#pdf-parser>; на сайте также есть ссылка на иллюстрации, на которых демонстрируется работа этого средства. Согласно Дидье, этот Python-скрипт «обрабатывает PDF документ, чтобы определить основные элементы, используемые в анализируемом файле. Он не преобразовывает PDF-документ».

Совет

Интерпретатор языка Python можно бесплатно загрузить с сайта ActiveState.com; это тот же сайт, на котором доступен интерпретатор языка Perl.

Скрипт «pdf-parser.py» извлекает различные метаданные и содержимое из PDF-документа, в том числе объекты и JavaScript-код, встроенные в документ. Например, Дидье опубликовал в своем блоге сообщение (<http://blog.didierstevens.com/2008/11/10/shoulder-surfing-a-malicious-pdf-author/>), в котором описывается анализ информации из вредоносного PDF-документа, содержащего код, использующий уязвимость JavaScript-функции util.printf (<http://cve.mitre.org/cgi-bin/cvename.cgi?name=2008-2992>).

Didier также предоставляет свой Python-скрипт, ExtractScripts (<http://blog.didierstevens.com/programs/extractscripts/>), который извлекает в отдельные файлы потенциально вредоносные скрипты, встроенные в HTML-файлы.

Файлы изображений

Файлы документов Word – не единственные файлы, содержащие внутренние метаданные. В феврале 2006 года в журнале *Washington Post Magazine* была опубликована статья о «пастухе» ботов (злоумышленник, который заражает компьютеры программами-роботами (ботами), а затем управляет этими сетями или даже сдает их в аренду); интернет-версия этого рассказа включала в себя JPEG-изображение. Хотя автор этой истории старался сохранить личность владельца бот-сети в тайне, JPEG-изображение содержало заметки фотографа, в которых указывалось место (город и штат), где был сделан этот снимок.

Метаданные, имеющиеся в JPEG-изображении, в основном зависят от приложения, в котором изображение было создано или изменено. Например, цифровые камеры внедряют в изображение информацию в формате EXIF, которая может включать в себя модель и производителя камеры (к сожалению, похоже, что серийный номер не используется или не сохраняется) и даже хранить миниатюру или аудиоинформацию (EXIF использует формат каталога файла изображения TIFF). Такие приложения, как Adobe Photoshop, имеют собственные метаданные, которые они добавляют в JPEG-файлы.

Такие инструменты, как Exifer (www.friedemann-schmidt.com/software/exifer/), IrfanView (www.irfanview.com) и Perl-модуль Image::MetaData::JPEG позволяют вам просматривать, извлекать и в некоторых случаях изменять метаданные, внедренные в файлы JPEG-изображений. Программа ProDiscover также может показывать данные EXIF, найденные в JPEG-изображении. Крис Браун (Chris Brown) из компании Technology Pathways предоставляет документ (<http://toorcon.techpathways.com/cs/forums/storage/8/11/EXIF.pdf>), в котором описываются данные EXIF и отчасти формат JPEG-файла.

Анализ сигнатур файлов

Во время экспертизы вам могут встретиться файлы с необычным расширением или файлы, имеющие знакомые расширения, но находящиеся в необычных местах. В таких случаях можно использовать *анализ сигнатур файлов*, чтобы определить тип этих файлов, а также получить некоторое представление о технических способностях злоумышленника. Один из способов определить правильный тип файлов (независимо от их расширений) – использовать анализ файловых сигнатур.

Анализ файловых сигнатур представляет собой сбор информации из первых 20 байт файла и поиск отдельной сигнатуры или магического числа, указывающего на тип и назначение файла. Разные типы файлов имеют разные сигнатурные, и эти сигнатурные не зависят от расширения файла. Надо сказать, что злоумышленники часто меняют расширение файла, чтобы при просмотре этого файла в проводнике Windows он отображался со значком, который эффективно скрывает содержимое и назначение файла. Однажды я проводил анализ чат-бота, который я назвал «russiantopz» (www.securityfocus.com/infocus/1618). Этот чат-бот размещал ряд файлов в зараженной системе и добавлял этим файлам расширения .drv и .dll, чтобы они выглядели как обычные файлы, на которые администраторы редко обращают внимание. В конце концов, в большинстве случаев, когда администратор открывает файл с одним из таких расширений в шестнадцатеричном редакторе, все что он видит, – это набор двоичных данных. Во время анализа я, как ни странно, открыл эти файлы и смог увидеть, что они содержат текстовую информацию, а именно сведения о конфигурации и данные о действиях, которые предпринимал бот, когда получал команду.

Инструменты для судебного анализа данных, такие как ProDiscover, позволяют эксперту легко выполнять анализ сигнатур и просматривать результаты. При проведении анализа эти инструменты получают расширение файла и сравнивают сигнатуру, связанную с этим расширением, с информацией, содержащейся в первых 20 байт файла. Например, сигнатура переносимых исполняемых файлов Windows (формат PE) начинается с букв *MZ*, которые находятся в первых 2 байтах PE-файла (*MZ* – упоминание о Марке Збиковском (Mark Zbikowski) [http://en.wikipedia.org/wiki/Mark_Zbikowski], разработчике из Microsoft). Исполняемые файлы могут иметь расширения .exe, .dll, .sys, .osx, .drv и другие, как видно из файла «headersig.txt», используемом программой ProDiscover в качестве «базы данных» расширений и сигнатур файлов. Короче говоря, если файл имеет расширение исполняемого файла, у него должна быть правильная сигнатура исполняемого файла. Файлы, не имеющие правильных сигнатур, соответствующих их расширениям, должны быть подвергнуты дальнейшему анализу.

Такие файлы изображений, как JPEG и GIF, также имеют свои собственные сигнатуры. Сигнатура для JPEG файла – JFIF, а сигнатура для GIF файла – GIF87a или GIF89a. На илл. 5.10 показана сигнатура для PDF-документа, или символы %PDF-, за которыми следует версия формата PDF для этого файла.

00000000h: 25 50 44 46 2D 31 2E 35 0D 25 E2 E3 CF D3 0D 0A ; %PDF-1.5.%äÍÓ..

Илл. 5.10. Сигнатура PDF-файла.

Perl-скрипт «sigs.pl», расположенный на DVD-носителе, который идет в комплекте с этой книгой, позволит вам выполнить анализ сигнатур файлов на работающих компьютерах. Он проанализирует файл, каталог файлов или все файлы в структуре каталогов, чтобы определить, соответствуют ли расширения сигнатурам файлов. Скрипт использует тот файл «headersig.txt» от компании Technology Pathways в качестве «базы данных» сигнатур файлов по умолчанию, однако можно также использовать другие списки такого же формата. Когда скрипт анализирует файлы, он не только определяет, соответствует ли расширение сигнатуре файла, но и предупреждает эксперта, если расширение файла не найдено в «базе данных». В таком случае скрипт предлагает расширение и сигнатуру, чтобы эксперт мог обновить свою базу данных, если он считает это необходимым. По умолчанию скрипт отправляет свои результаты на консоль в формате значений, разделенных запятыми (.csv), чтобы их можно было перенаправить в файл и открыть в программе Excel для удобного анализа.

Альтернативные потоки данных в NTFS

Альтернативные потоки данных – свойство файловой системы NTFS, о котором мало что знают и которое плохо понимают системные администраторы. В конце концов, зачем им это? На первый взгляд, альтернативные потоки данных незаметно используются некоторыми приложениями Microsoft, поэтому они не могут причинить вред, ведь так?

Давайте посмотрим на это с другой стороны. Предположим, что существует способ создавать законные файлы в ОС Windows, файлы, которые могут содержать не только данные, но и скрипты или исполняемый код, и эти файлы можно создавать или запускать, но в операционной системе нет собственных средств, позволяющих обнаружить присутствие этих файлов. Это правда. В ОС Windows есть все инструменты, необходимые для создания и изменения альтернативных потоков данных, а также для управления ими, но нет собственных средств, чтобы просмотреть наличие этих потоков. Хотя это не совсем верно, так как, начиная с Windows Vista, для команды *dir* появился переключатель, позволяющий вам увидеть альтернативные потоки данных. Вскоре мы рассмотрим эту возможность.

Итак, что же такое альтернативные потоки данных, откуда они берутся и как используются? Альтернативные потоки данных – функция файловой системы NTFS, введенная с появлением Windows NT 3.1. Альтернативные потоки данных были добавлены в файловую систему для поддержки иерархической файловой системы

(Hierarchical File System, HFS), используемой Macintosh. HFS использует ветви ресурсов, чтобы файловая система могла сохранять такие метаданные о файле, как значки, меню или диалоговые окна. Эта функция была включена в состав файловой системы NTFS, но она никогда не была темой для широкого обсуждения. На самом деле долгое время альтернативные потоки данных практически не обсуждались, и по этой теме было очень мало информации, даже от корпорации Microsoft. Хотя приложения Microsoft и функциональные возможности в оболочке позволяют создавать отдельные альтернативные потоки данных, факт остается фактом – альтернативные потоки данных редко используются в повседневной работе. Злоумышленники поняли это и применяют альтернативные потоки данных для скрытия инструментов, даже как часть руткитов. Это эффективный подход, потому что некоторые антивирусные программы либо не сканируют альтернативные потоки данных, либо не делают этого по умолчанию. Следовательно, вредоносная программа, попавшая в систему в альтернативном потоке данных, возможно, не будет обнаружена или удалена/помещена в карантин антивирусным приложением.

Записки из подполья...

Использование альтернативных потоков данных

В конце 1990-х годов, работая консультантом, я участвовал в нескольких тестах на проникновение и оценках уязвимости систем. Когда во время теста на проникновение мы получали доступ к компьютеру с ОС Windows, то оставляли на нем альтернативный поток данных, если у нас было на это разрешение. Это никак не влияло на работу системы, потому что мы оставляли только небольшое текстовое сообщение размером несколько байт. Однако таким способом мы сообщали системному администратору, что проникли в систему и предоставляли ему доказательство этого. Я встречал специалистов по тестированию на проникновение, которые копировали все свои инструменты на взломанную систему в альтернативных потоках данных.

Создание альтернативных потоков данных

Создать альтернативный поток данных довольно просто; более того, некоторые приложения Microsoft делают это автоматически. Любой пользователь может сделать это, если у него есть возможность создавать файлы. Например, самый простой способ создать альтернативный поток данных – ввести следующую команду:

```
D:\ads>notepad myfile.txt:ads.txt
```

Вначале вы увидите диалоговое окно, в котором спрашивается, хотите ли вы создать новый файл. Нажмите «Да» (“Yes”), введите какой-нибудь текст в окно, сохраните файл, а затем закройте окно программы «Блокнот» (“Notepad”). Если на этом этапе вы введете команду **dir**, то увидите что файл с именем «myfile.txt» имеет размер 0 байт, хотя вы только что ввели текст в «Блокнот».

Еще один способ создать альтернативный поток данных – использовать команду **echo**:

```
D:\ads>echo "This is another ADS test file" > myfile.txt:ads2.txt
```

Итак, вы создали два альтернативных потока данных; теперь при каждом вводе команды **dir** или при просмотре содержимого каталога в проводнике Windows, вы увидите отдельный файл в каталоге, и этот файл будет иметь размер 0 байт.

Следующий способ создать альтернативный поток данных – использовать команду **type**, чтобы скопировать другой файл в альтернативный поток данных:

```
D:\ads>type c:\windows\system32\sol.exe > myfile.txt:ads3.exe
```

Таким образом вы копируете содержимое файла с именем «sol.exe» (который является карточной игрой «Косынка» в ОС Windows 2000, XP и 2003) в альтернативный поток данных. Можно использовать те же команды в ОС Vista, чтобы создавать альтернативные потоки данных, хотя для некоторых приложений (например, для игры «Косынка») пути к исполняемым файлам могут быть другими.

Альтернативные потоки данных можно также добавлять к спискам каталогам, используя следующий синтаксис:

```
D:\ads>echo "This is an ADS attached to a directory" > :ads.txt
```

Обратите внимание, что вы не указывали отдельное имя файла. Это приводит к тому, что альтернативный поток данных добавляется к списку каталогов, в данном случае D:\ads.

Альтернативные потоки данных можно создавать и другими способами; часто вы даже не будете знать о том, что такие потоки создаются. Если вы щелкните правой кнопкой мыши по файлу, и выберите пункт «Свойства» (“Properties”), то увидите, что одна из вкладок называется «Сводка» (“Summary”) (похоже, что эта вкладка не доступна в ОС Vista). Вы можете вводить любую информацию в различные текстовые поля, и когда вы сохраните эту информацию, нажав «OK», она сохранится в альтернативном потоке данных (но если вы работаете с документом Office, введенная информация сохранится в структурированном хранилище или в самом OLE-документе).

Более того, диспетчер вложений (Attachment Manager, <http://support.microsoft.com/kb/883260>), который входит в состав ОС Windows XP с пакетом обновления 2 (SP2), добавляет альтернативный поток данных к файлам, загружаемым из Интернета или извлекаемым из электронной почты в виде вложений (через браузер Internet Explorer или программу Outlook соответственно). Когда вы загружаете файл с помощью Internet Explorer, он сохраняется в выбранном месте, а альтернативный поток данных с именем Zone.Identifier добавляется к файлу (конечно, при условии, что файловая система – NTFS; иначе, согласно статье № 883260 из базы знаний Microsoft, диспетчер вложений автоматически прервет свою работу). Альтернативный поток данных добавляется к файлу, чтобы при попытке открыть или выполнить файл появлялось диалоговое окно с предупреждением, что этот файл не безопасен.

Отображение альтернативных потоков данных

Теперь, когда вы знаете, как создавать альтернативные потоки данных, нужно научиться их обнаруживать. Как я уже говорил, в ОС Windows нет собственных средств, позволяющих отображать произвольные альтернативные потоки данных. Вы не увидите эти потоки через проводник Windows, и команда *dir* также будет бесполезна. Хотя последнее утверждение не совсем верное; в ОС Windows Vista можно отобразить альтернативные потоки данных с помощью команды *dir*, используя переключатель /r, как показано на илл. 5.11.

```
C:\ads>dir /r
Volume in drive C has no label.
Volume Serial Number is 98A5-80D5

Directory of C:\ads

11/20/2006  07:17 PM    <DIR>          .
11/20/2006  07:17 PM    <DIR>          ..
11/20/2006  07:33 PM          0 myfile.txt
                                23 myfile.txt:ads.txt:$DATA
                                34 myfile.txt:ads2.txt:$DATA
                                982,528 myfile.txt:ads3.exe:$DATA
                               1 File(s)      0 bytes
                               2 Dir(s)  14,823,571,456 bytes free
```

Илл. 5.11. Пример отображения альтернативных потоков данных в Windows Vista.

На илл. 5.11 показаны результаты запуска команды *dir /r* в Windows Vista после того, как несколько альтернативных потоков данных были созданы одним из способов, рассмотренных в разделе «Создание альтернативных потоков данных» (в этом разделе мы создавали альтернативные потоки данных в Windows XP).

В других операционных системах Windows (2000, XP и 2003) нужно использовать сторонние утилиты, чтобы отобразить альтернативные потоки данных. Моя любимая – «lads.exe» (www.heysoft.de/Frames/f_sw_la_en.htm), созданная Фрэнком Хейне (Frank Heyne). «Lads.exe» – это инструмент командной строки, который можно запускать для анализа любого каталога.

```
D:\tools>lads d:\ads
LADS - Freeware version 4.00
(C) Copyright 1998-2004 Frank Heyne Software (http://www.heysoft.de)
This program lists files with alternate data streams (ADS)
Use LADS on your own risk!
Scanning directory d:\ads\
      size      ADS in file
-----
      0      d:\ads\myfile.txt:ads.txt
     34      d:\ads\myfile.txt:ads2.txt
1032192      d:\ads\myfile.txt:ads3.exe
1032226      bytes in 3 ADS listed
```

«Lads.exe» – лишь один из доступных инструментов, позволяющий отобразить альтернативные потоки данных в Windows. Также существуют другие инструменты командой строки, инструменты с графическим интерфейсом пользователя и даже инструменты, устанавливаемые в виде подключаемых модулей оболочки, чтобы можно было отобразить альтернативные потоки данных посредством пользовательского интерфейса в проводнике Windows.

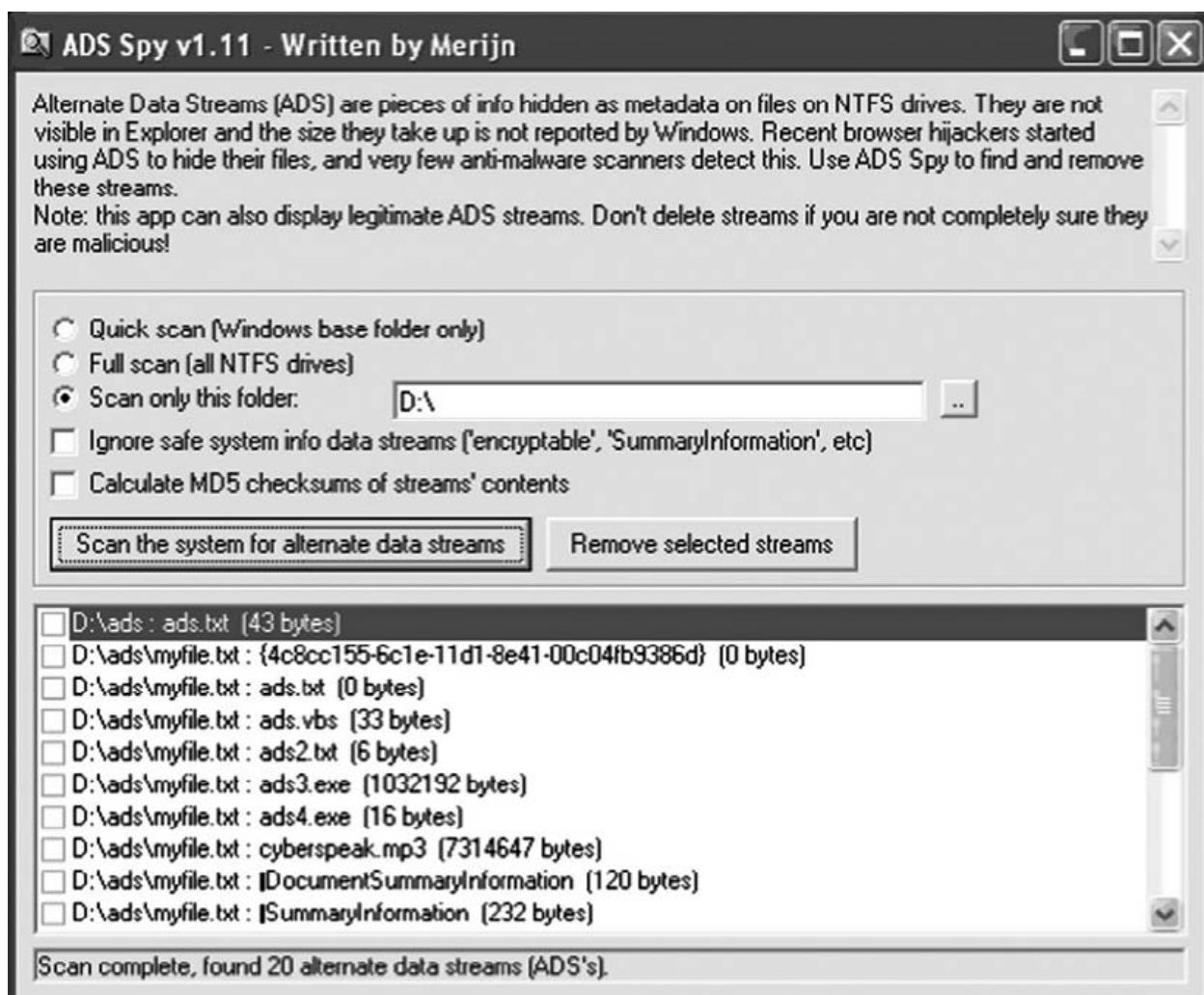
Альтернативные потоки данных, присоединяемые к файлу посредством добавления сводной информации в файл (что упоминалось в предыдущем разделе), отличаются от альтернативных потоков данных, которые мы создавали ранее. Например, если мы добавим сводную информацию к файлу «myfile.txt», а затем запустим «lads.exe», то увидим:

```
size      ADS in file
-----
120      d:\ads\myfile.txt: *DocumentSummaryInformation
232      d:\ads\myfile.txt: *SummaryInformation
0      d:\ads\myfile.txt:ads.txt
34      d:\ads\myfile.txt:ads2.txt
1032192      d:\ads\myfile.txt:ads3.exe
0      d:\ads\myfile.txt:{4c8cc155-6c1e-11d1-8e41-00c04fb9386d}
```

В выходных данных «lads.exe» видно, что к файлу «myfile.txt» было присоединено три дополнительных альтернативных потока данных: один, который отображается в виде глобального уникального идентификатора GUID (и имеет размер 0 байт), и два других, начинающихся с символа *. В последних двух потоках сохраняется информация, введенная во вкладке «Сводка» (“Summary”) диалогового окна «Свойства» (“Properties”).

Иногда можно встретить альтернативный поток данных, который называется AFP_AfpInfo или AFP_Resource. В таком случае следует проверить, есть ли на этом компьютере установленные и включенные файловые службы для Macintosh. Если да, то поток без имени, возможно, был скопирован с компьютера Macintosh через протокол AppleTalk. В таком случае ветвь данных для файла сохраняется в файл, например, с именем «myfile.txt». Ветвь ресурсов затем сохраняется в поток myfile.txt:AFP_Resource, а информация об атрибутах – в myfile.txt:AFP_AfpInfo.

Как упоминалось выше, существуют другие инструменты для отображения альтернативных потоков данных. «Streams.exe» (от Sysinternals, доступный на сайте Microsoft), «lns.exe» (от Арне Видстрома (Arne Vidstrom), доступный на сайте NTSecurity.nu) и «sfind.exe» (входящий в состав Forensic Toolkit и доступный на сайте Foundstone.com) – это инструменты командной строки, похожие на утилиту «lads.exe». ADS Detector – это подключаемый модуль оболочки (т. е. проводника Windows), доступный на сайте CodeProject.com и позволяющий просматривать незашифрованные альтернативные потоки данных файла в реальном масштабе времени. Наконец, CrucialADS (с сайта CrucialSecurity.com) и ADS Spy (с сайта SpyWareInfo.com) – это инструменты с графическим интерфейсом пользователя для отображения альтернативных потоков данных. ADS Spy, показанный на илл. 5.12, также позволяет пользователю удалять выбранные альтернативные потоки данных.



Илл. 5.12. Графический интерфейс программы ADS Spy.

После того как вы найдете альтернативный поток данных, можно просмотреть содержимое файла, открыв его в программе «Блокнот» или применив утилиту *cat*, часть пакета UnxUtils с сайта SourceForge.net. Можно использовать *cat*, чтобы просмотреть содержимое альтернативного потока данных в консоли (т. е. в стандартном устройстве вывода) или перенаправить выходные данные команды в отдельный файл.

Предупреждение

В 2000 году два хакера под псевдонимами Benny и Ratter, которые тогда были членами группы создателей вирусов, известной как 29A (шестнадцатеричное представление числа 666), выпустили вирус W2K.Stream, использующий альтернативные потоки данных. Вирус заражал файл, заменял его, а затем копировал исходный файл в альтернативный поток данных. Например, если вирус заражал файл «notepad.exe», он заменял исполняемый файл и копировал исходную программу «Блокнот» в поток Notepad.exe:STR. Вирус работал только в файловых системах NTFS. Если файловая система была отформатирована как FAT, то происходило только заражение файла, а альтернативный поток данных не создавался.

В июне 2006 года в блоге компании, выпускающей антивирус F-Secure, было опубликовано сообщение, в котором описывался драйвер руткита режима ядра Mailbot.AZ (также известен как Rustock.A), который делает обнаружение особенно трудным, скрывая себя в альтернативном потоке данных (подробную информацию о руткитах см. в главе 7). Более того, по имеющимся данным, этот альтернативный поток данных нельзя отобразить с помощью инструментов, обнаруживающих альтернативные потоки данных, так как он скрывается рутkitом. Очень хитро!

Использование альтернативных потоков данных

Вероятно, вам интересно, для чего используются альтернативные потоки данных, помимо скрытия информации. Оказывается, их можно использовать в нескольких случаях. Например, можно вложить исполняемый файл в альтернативный поток данных и запустить его оттуда. Используйте команду *type*, как мы делали раньше, чтобы поместить исполняемый файл в альтернативный поток данных:

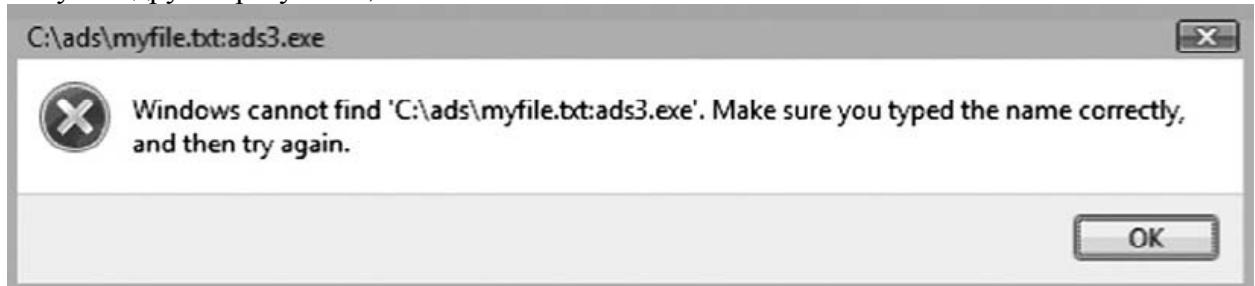
```
D:\ads>type c:\windows\system32\sol.exe > myfile.txt:ads4.exe
```

В данном случае мы поместили игру «Косынка» в альтернативный поток данных. Это хороший пример, потому что при его выполнении открывается графический интерфейс игры, позволяющий убедиться, что все работает должным образом. Чтобы выполнить программу введите следующую команду:

```
D:\ads>start .\myfile.txt:ads4.exe
```

Как видите, открывается графический интерфейс программы «Косынка». Помимо исполняемых файлов, в альтернативных потоках данных можно также легко скрывать скрипты (Windows Scripting Host [WSH], Perl и т. д.) и запускать их оттуда. Инструменты WSH (cscript.exe, wscript.exe), как и Perl, без проблем запускают скрипты, скрытые в альтернативных потоках данных; даже веб-сервер IIS обеспечит выполнение HTML-файлов и скриптов, скрытых в альтернативных потоках данных (что является отличным способом продемонстрировать успешную атаку сервера («захват флага»)).

При попытке выполнить альтернативный поток данных в ОС Windows Vista мы получим другой результат, как показано на илл. 5.13.



Илл. 5.13. Диалоговое окно, отображаемое при попытке выполнить альтернативный поток данных в ОС Windows Vista.

При попытках запустить альтернативный поток данных («myfile.txt:ads3.exe» содержит версию программы «Косынка» для Vista), в том числе с помощью различных версий команды *start*, а также с использованием строки «Выполнить» (“Run”) из меню «Пуск» (“Start”), были получены те же результаты. Однако запуск WSH-скриптов из альтернативного потока данных был выполнен в ОС Windows Vista без ошибок.

Еще один интересный способ использования альтернативных потоков данных – скрытие мультимедийных файлов. В альтернативных потоках данных можно скрыть фильмы и подкасты, а затем запустить проигрыватель Windows Media Player из командной строки, чтобы открыть мультимедийный файл:

```
wmpplayer d:\ads\myfile.txt:ciberspeak.mp3
```

Используя этот способ, я прослушал подкаст CyberSpeak. Интересно, что хотя подкаст был запущен из командной строки, его имя появилось в следующем разделе реестра:

```
HKEY_CURRENT_USER\Software\Microsoft\MediaPlayer\Player\RecentFileList
```

Эта запись содержалась в списке в данных, связанных с параметром *File0*, что свидетельствует о том, что при каждом добавлении нового файла в этот список, имя файла добавляется в верхнюю часть списка, а более старые имена файлов сдвигают вниз; чем меньше номер файла, тем новее файл. Как вы поняли из главы 4, получив время *LastWrite* из раздела реестра, можно узнать, когда к файлу был получен доступ посредством проигрывателя Windows Media Player.

Предупреждение

Во время просмотра пробного дела в программе ProDiscover, я заметил, что в корзине было несколько альтернативных потоков данных. Чтобы выделить альтернативные потоки данных, программа ProDiscover отображает их красным шрифтом. Я удалил несколько файлов, с которыми работал, и один из этих файлов был загружен из Интернета. Я заметил, что для этого файла показан альтернативный поток данных Zone.Identifier (файл был загружен с помощью браузера Internet Explorer), но счетчик записей для общего числа файлов в INFO2 не отображает наличие этого альтернативного потока данных.

Удаление альтернативных потоков данных

Теперь, когда вы поняли, как создаются и используются альтернативные потоки данных, нужно узнать, как их можно удалить. Для этого существуют несколько способов, и вы можете выбрать любой из них в зависимости от ваших потребностей и предпочтений.

Один из способов удалить альтернативный поток данных – просто удалить файл, к которому присоединен этот поток данных. Однако очевидно, что если исходный файл был очень важен для вас (документ, таблица, файл изображения), то вы потеряете эти данные.

Для сохранения исходных данных можно использовать команду *tupe*, чтобы копировать содержимое исходного потока (в нашем случае «myfile.txt») в файл с другим именем, а затем удалить исходный файл. Еще один вариант – копировать файл на носитель с файловой системой, отличной от NTFS. Как вы помните, альтернативные потоки данных характерны для файловой системы NTFS, поэтому при копировании файла на дискету (помните такие?), флеш-накопитель или другой раздел, отформатированный в FAT, FAT32 или другой файловой системе, альтернативный поток данных эффективно удаляется (например, можно передать файл по FTP на компьютер с операционной системой Linux, в котором накопитель отформатирован в ext2, а затем обратно).

Но что делать, если альтернативный поток данных, который вы обнаружили, присоединен к списку каталогов, например C:\ или C:\windows\system32? Вы не можете просто удалить каталог, а его копирование в другую файловую систему и обратно будет довольно трудно осуществить. Так что же делать? Используя команду *echo*, можно преобразовать альтернативный поток данных в безвредный текстовый файл независимо от его содержимого. Используя предыдущий пример, где мы копировали программу «Косынка» в альтернативный поток данных, мы можем запустить утилиту «lads.exe» и получить информацию об этом альтернативном потоке данных:

```
56832 d:\ads\myfile.txt:ads4.exe
```

Итак, у нас есть альтернативный поток данных размером 56832 байта, и мы уже знаем, что это исполняемый файл. Поэтому введите следующую команду:

```
D:\ads>echo "deleted ADS" > myfile.txt:ads4.exe
```

Снова запустив утилиту «lads.exe», мы увидим, что размер файла изменился:

```
16 d:\ads\myfile.txt:ads4.exe
```

Итак, мы эффективно «позабочились» об альтернативном потоке данных; хотя мы его не удалили, мы преобразовали его в безвредные данные. Можно также написать сообщение в альтернативном потоке данных, указывая тип найденного потока, время его удаления и ваше имя.

Наконец, еще один способ – использовать программу ADS Spy с графическим интерфейсом пользователя, о которой мы говорили выше.

Альтернативные потоки данных: заключение

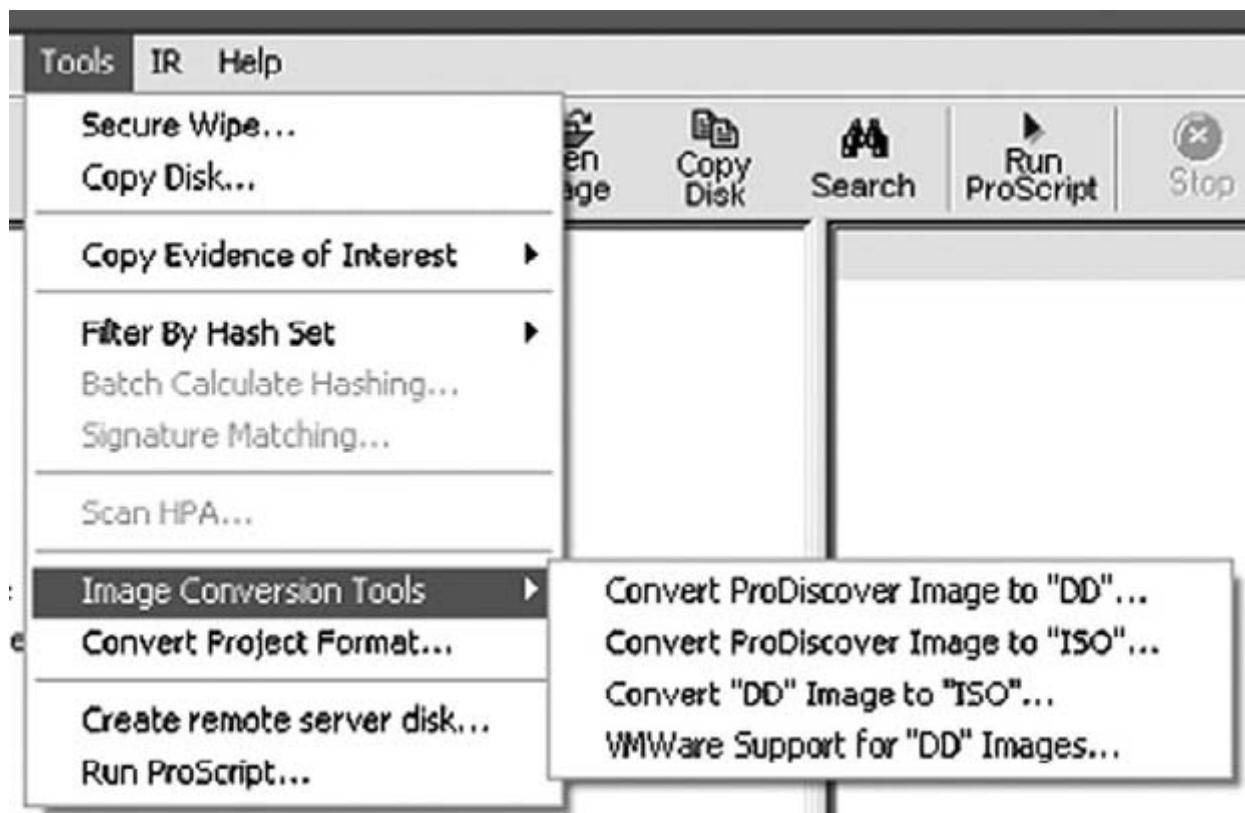
Мы рассмотрели много информации об альтернативных потоках данных, обсудив способы их создания, использования и удаления. Необходимо учитывать эту информацию при расследовании инцидентов или проведении компьютерно-технических экспертиз. Присутствие альтернативных потоков данных – настолько необычное явление, что некоторые программы для судебного анализа данных выделяют их красным шрифтом. Однако не все альтернативные потоки данных являются вредоносными по своей природе; вы видели, что некоторые приложения используют их в процессе своей обычной работы.

Эксперт должен не забывать просматривать содержимое альтернативного потока данных. Если альтернативный поток данных назван с помощью одной из схем именований, используемых известными, надежными приложениями, это вовсе не значит, что в этом потоке не может находиться вредоносное содержимое. Иначе говоря, не стоит считать альтернативный поток данных безопасным только потому, что он называется AFP_AfpInfo. Злоумышленники любят размещать вредоносные программы в обычных местах, присваивая им такие имена, на которые администратор или судебный эксперт, вероятнее всего, не обратит внимания.

Альтернативные методы анализа

Иногда во время компьютерно-технической экспертизы вам, возможно, потребуется выполнить задачи, которые становятся намного более трудоемкими, если вы работаете только с образом накопителя. Предположим, вы захотите просканировать систему на наличие вредоносных программ, таких как трояны, программы удаленного администрирования или шпионское ПО. Во время работы с образом системы у вас есть отдельный файл (или чаще всего несколько файлов, которые вместе равны размеру исходного НЖМД), и вам нужен способ, чтобы просканировать все объекты в этом образе. Поэтому, вместо того чтобы извлекать все объекты из образа, можно использовать инструменты, преобразовывающие образ в формат, подходящий для сканирования.

Один из таких инструментов доступен в программе ProDiscover. Начиная с версии 4.85 программы ProDiscover, этот инструмент позволяет преобразовать образ из собственного формата ProDiscover или формата dd в формат ISO. ProDiscover также имеет возможность создавать файлы, необходимые для загрузки образа в программе VMware. На илл. 5.14 показаны эти новые опции.

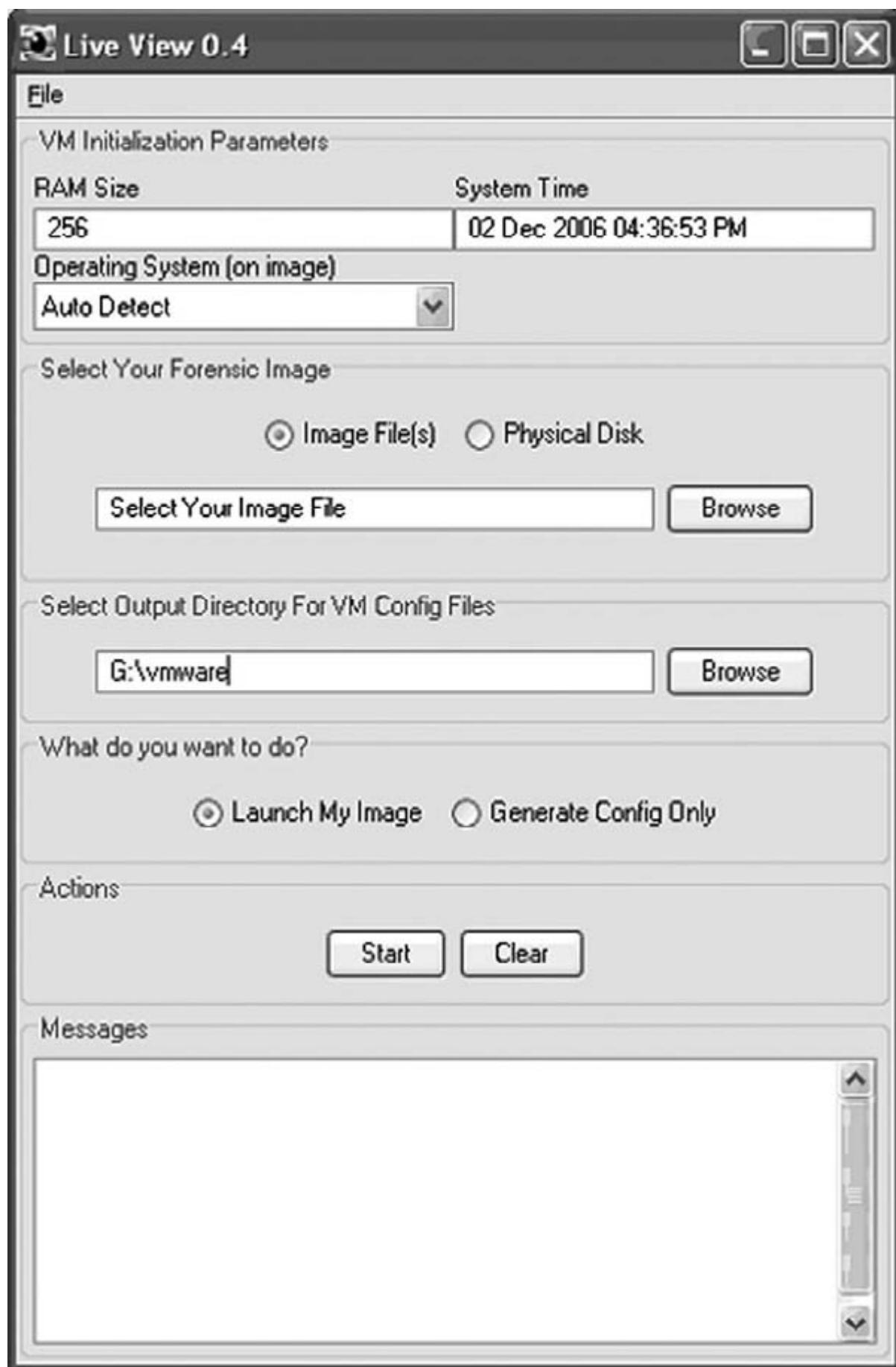


Илл. 5.14. Новые инструменты в меню ProDiscover.

Как показано на илл. 5.14, программу ProDiscover можно использовать для преобразования образов из собственного формата ProDiscover (.eve) в формат *dd* или из формата ProDiscover или *dd* в формат ISO 9660 Joliet. ProDiscover можно также использовать для создания файлов, необходимых для загрузки образа в программе VMware, что похоже на функцию инструмента VMware P2V Assistant (P2V означает «из физического в виртуальный»). Используя такие инструменты, можно загрузить систему, чтобы выполнить дополнительные задачи анализа, например, поиск вирусов и шпионских программ, или чтобы просто увидеть, как «выглядела» система во время своей работы. Иногда это может быть очень полезным при расследовании дела, потому что во время анализа образа данных достаточно трудно определить тип работающей системы (учитывая взаимодействия между различными параметрами конфигураций, установленными программами и т. д.).

Технический онлайн-семинар на сайте Technology Pathways Resource Center (www.techpathways.com/DesktopDefault.aspx?tabindex=8&tabid=14) познакомит вас со способами использования инструментов ProDiscover для монтирования образа в VMware. Для участия в онлайн-семинаре требуется клиентское программное обеспечение с сайта WebEx.com.

Еще один бесплатный и очень удобный в использовании инструмент для загрузки образа в программе VMware – это Live View (<http://liveview.sourceforge.net>), который доступен на сайте CERT. Live View использует простой графический интерфейс пользователя (см. илл. 5.15), чтобы познакомить вас с параметрами конфигурации, требуемыми для настройки образа, который будет загружен в программе VMware, и автоматически создает необходимые для этого файлы.



Илл. 5.15. Графический интерфейс Live View.

Запуск Live View – простой и интуитивный процесс. Live View поддерживает большинство версий ОС Windows и имеет ограниченную поддержку ОС Linux. Я

несколько раз успешно использовал эту программу, чтобы загрузить образ и войти в загруженную систему.

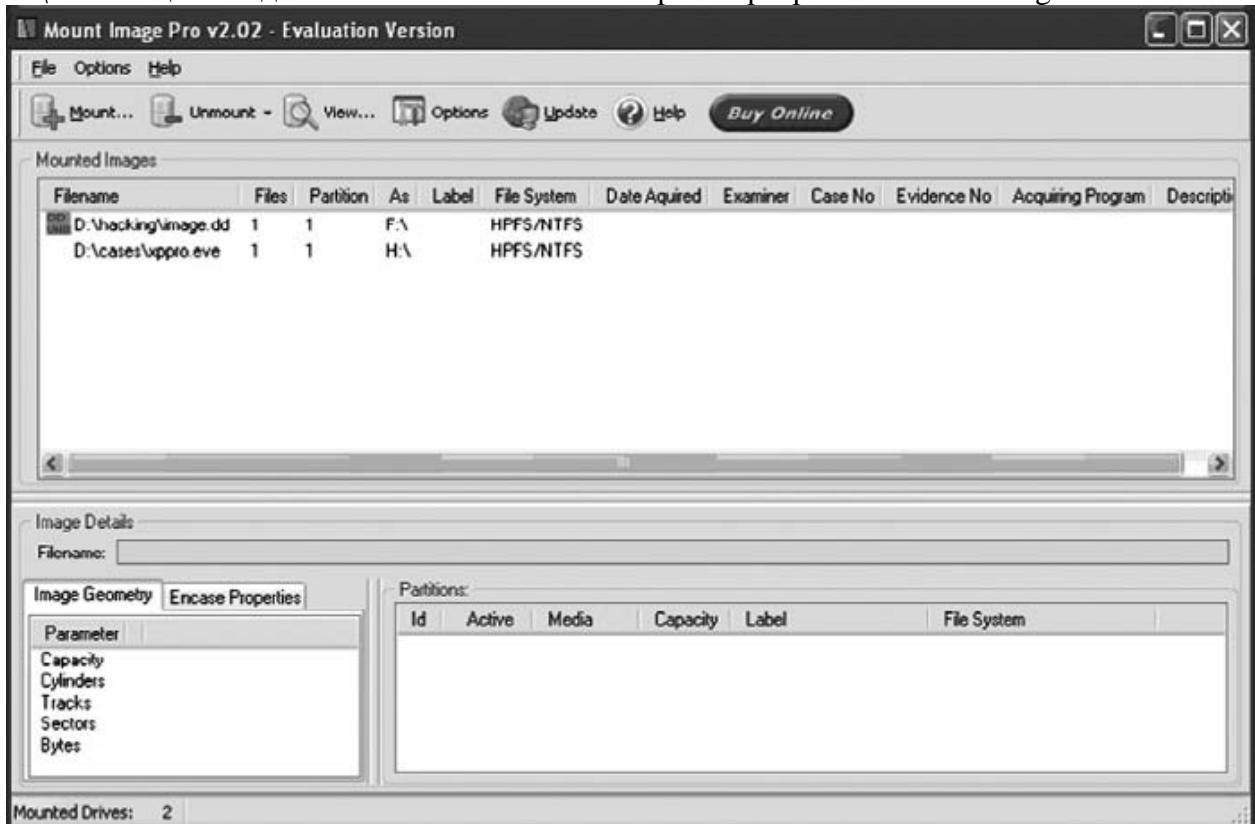
Примечание

После того как у вас будет файл-образ, который нужно загрузить, вы можете выполнять с ним различные действия. Если вы используете продукт VMware Workstation и настроили его на сетевое подключение через мост, можно включить сетевой интерфейс для только что загруженного образа данных и сканировать образ так же, как вы выполняете сканирование портов и/или поиск уязвимостей. Возможно, вы захотите войти в работающую систему, поэтому, если учетная запись администратора или пользователя защищена паролем, вам нужно будет связаться с ИТ-отделом организации, где работает пользователь, чтобы получить этот пароль, или подобрать пароль, используя специальные средства или пароли, найденные во время судебной экспертизы компьютера.

Загрузка образа с помощью Live View – операция, которую эксперт может сделать частью процесса анализа, чтобы «увидеть систему в том виде, в котором ее видел пользователь». Альтернатива использованию таких инструментов, как Live View, – монтирование образа как накопителя в режиме только для чтения.

Монтирование образа

Еще один отличный (хотя и не бесплатный) инструмент – это Mount Image Pro (далее MIP, доступный на сайте www.mountimage.com). MIP – замечательное средство, позволяющее монтировать образ на работающем компьютере как накопитель в режиме только для чтения. На илл. 5.16 показаны два образа, монтированные как накопители F:\ и H:\ с помощью 30-дневной ознакомительной версии программы Mount Image Pro.



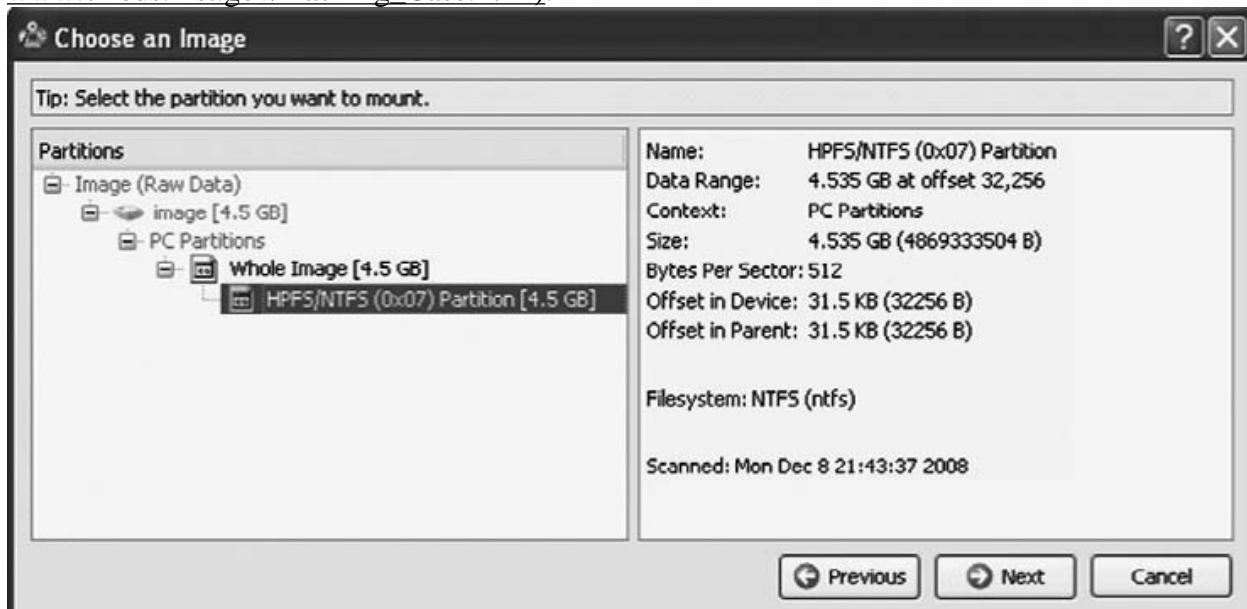
Илл. 5.16. Образы, монтированные как накопители с помощью Mount Image Pro версии 2.02.

MIP не загружает образ и не позволяет получить доступ к образу как к работающей системе; вы не сможете извлечь выполняющиеся процессы этой системы из физической памяти. Вместо этого программа монтирует образ как отдельный накопитель, чтобы

доступ к файлам в образе можно было получить так же, как к любому другому накопителю, и делает это в режиме только для чтения, чтобы в образ нельзя было внести никаких изменений. Для проверки возможностей программы я использовал утилиту «md5deep.exe», чтобы вычислить криптографический хэш образа клонированной системы, содержащейся в отдельном файле. Затем я использовал MIP, чтобы монтировать образ как накопитель, получил доступ к некоторым файлам и копировал несколько файлов с монтированного накопителя на другой раздел своего компьютера. После того как я завершил другие действия, в том числе запуск нескольких Perl-скриптов для анализа файлов в монтированном накопителе, я размонтировал накопитель и полностью завершил работу программы MIP. Затем я снова применил «md5deep.exe» в файлу-образу, и возвращенное хэш-значение совпало с первым вычисленным хэш-значением, что подтверждает тот факт, что образ монтируется в режиме только для чтения. Создание и проверка криптографических хэш-значений с помощью известных и принятых алгоритмов должны быть частью стандартных рабочих процедур, если вы используете такие инструменты, как Mount Image Pro. (Существует несколько бесплатных инструментов, реализовывающих такие алгоритмы хэширования, как MD5, SHA-1 и SHA-256.)

Еще один мощный инструмент, который можно использовать для монтирования образа данных как накопителя в режиме только для чтения, – это Smart Mount от компании ASR Data (www.asrdata.com/SmartMount/). Согласно Энди Розену (Andy Rosen), автору этого инструмента, Smart Mount предлагает значительно больше функциональных возможностей (например, она работает на платформах Windows и Linux, монтирует защищенные паролем файлы формата .E01 без пароля и т. д.), чем MIP, и хотя это коммерческий продукт, который необходимо приобретать, также доступна его ознакомительная версия.

SmartMount монтирует разнообразные файлы-образы, в том числе файлы формата .vmdk программы VMware и файлы формата EWF программы EnCase, а также файлы-образы необработанных данных. На илл. 5.17 показана часть процесса использования Smart Mount для монтирования файла-образа как накопителя в режиме только для чтения (обратите внимание, что образ данных доступен на сайте www.cfreds.nist.gov/Hacking_Case.html).



Илл. 5.17. Монтирование файла-образа с помощью Smart Mount.

Можно также использовать бесплатные инструменты для монтирования образов данных как накопителей, хотя и с различными уровнями удобства и функциональности. Два таких инструмента – это VDK (<http://chitchat.at.infoseek.co.jp/vmware/vdk.html>) и ImDisk (www.ltr-data.se/opencode.html). VDK устанавливается как исполняемый файл на основе интерфейса командной строки (vdk.exe) и как драйвер (vdk.sys). Введите в

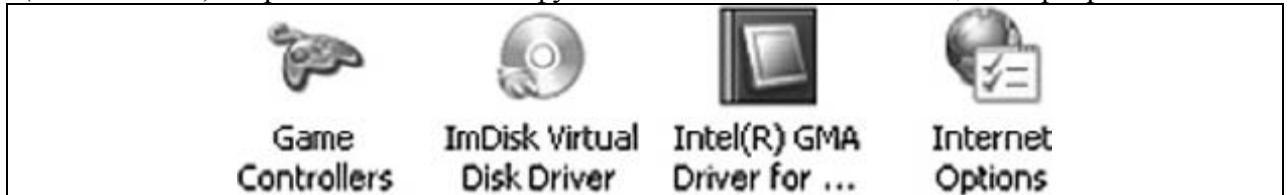
командной строке команду **vdk help**, чтобы ознакомиться с различными опциями программы VDK, или загрузите и установите программу VDKWin (на данный момент доступна версия 1.1) с графическим интерфейсом пользователя (<http://petruska.stardock.net/Software/VMware.html>), которая позволяет монтировать файлы образы как виртуальные накопители в режиме только для чтения. VDK позволяет монтировать образы накопителей виртуальной машины VMware (файлы .vmdk) и образы необработанных данных как накопители в режиме только для чтения. VDK можно использовать для выполнения других функций, в том числе для получения информации о файле-образе; например, для того чтобы получить сведения о разделах в образе, используйте команду *view*:

```
D:\vdk\vdk view D:\hacking\image.dd
```

Выходные данные этой команды выглядят следующим образом:

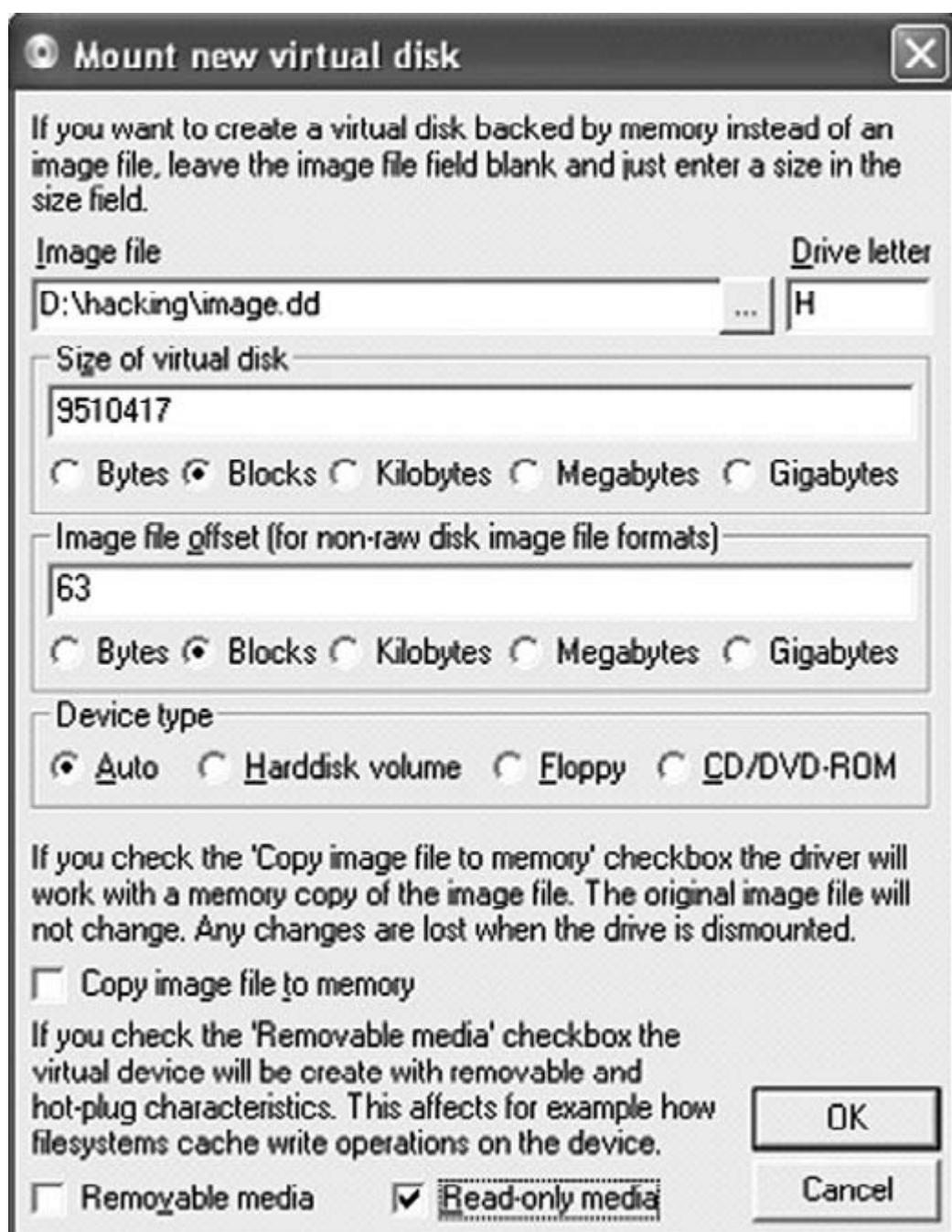
| | | | |
|-----------------|--------------|---------------------------|---------------|
| Image Name | : | image | |
| Disk Capacity | : | 9514260 sectors (4645 MB) | |
| Number Of Files | : | 1 | |
| Type | Size | Path | |
| ----- | ----- | ----- | |
| FLAT | 9514260 | d:\hacking\image.dd | |
| Partitions | : | | |
| # | Start Sector | Length in sectors | Type |
| -- | ----- | ----- | ----- |
| 0 | 0 | 9514260 (4645 MB) | <disk> |
| 1 | 63 | 9510417 (4643 MB) | 07h:HPFS/NTFS |

ImDisk устанавливается как консольное приложение и элемент панели управления (см. илл. 5.18) и предлагает такие же функциональные возможности, как программа VDK.



Илл. 5.18. Значок ImDisk в панели управления.

На илл. 5.19 показано диалоговое окно ImDisk, которое появляется после выбора файла-образа. На следующем этапе выбранный файл-образ будет монтирован в режиме только для чтения как виртуальный накопитель с символом H:\.



Илл. 5.19. Диалоговое окно программы ImDisk.

Монтирование образов как накопителей в режиме только для чтения имеет множество преимуществ, особенно в области обработки и анализа данных. К данным из образов можно в автоматическом режиме применять несколько инструментов, например, приложения для поиска вирусов и шпионских программ, утилиты для анализа сигнатур файлов и инструменты для отображения альтернативных потоков данных NTFS. Вместо того чтобы выбирать интересующие вас файлы в образе, а затем копировать их из образа для более подробного анализа, можно автоматизировать многие из этих операций с помощью Perl-скриптов.

Обнаружение вредоносных программ

Одна из самых трудных задач анализа, с которой сталкиваются многие эксперты, – это поиск вредоносных программ в системе или образе данных. Однажды я исследовал образ системы, в которой, по словам пользователя, происходили подозрительные события. В конце концов я нашел вредоносную программу, которая вызывала эти события, но

использование инструмента для монтирования образа как накопителя в режиме только для чтения не только позволило мне найти отдельную вредоносную программу намного быстрее, выполнив ее поиск с помощью антивирусного приложения, но также позволило мне автоматизировать поиск по нескольким образом, чтобы найти ту же вредоносную программу или использовать несколько инструментов для поиска вредоносных или шпионских программ. Это также помогло мне в одном отдельном случае, когда начальное заражение системы произошло за два года до создания образа данных. Возможность сканировать файлы в образе данных так же, как файлы на работающем компьютере, но не изменения их никоим образом, может быть чрезвычайно полезной для эксперта.

Совет

При попытке определить, содержит ли образ данных вредоносные программы, можно использовать файлы журналов установленного антивирусного приложения, если таковое присутствует. Например, файл «mrt.log», упоминавшийся ранее в этой главе, может дать вам некоторое представление о том, от чего была защищена система. Другие антивирусные приложения сохраняют файлы журналов в других местах на накопителе в зависимости от версии приложения. Программа McAfee VirusScan Enterprise версии 8.0i сохраняет свои файлы журналов в каталоге C:\Documents and Settings\All Users\Application Data\Network Associates\VirusScan, тогда как другая версия этого приложения сохраняет свой файл «onaccessscanlog.txt» в каталоге C:\Documents and Settings\All Users\Application Data\McAfee\DesktopProtection. Также не забудьте проверить журнал событий приложений на наличие записей, созданных антивирусными программами.

После монтирования образа данных (конечно, в режиме только для чтения) можно просканировать его с помощью антивирусного приложения на ваш выбор. Существует различные коммерческие и бесплатные антивирусные решения, и обычно рекомендуется выполнять сканирование, используя несколько антивирусных приложений. Клаус Валка (Claus Valca) перечислил некоторые переносимые антивирусные приложения в своем блоге Grand Stream Dreams (<http://grandstreamdreams.blogspot.com/2008/11/portable-anti-virusmalware-security.html>). При использовании любых инструментов для поиска вирусов и шпионских программ, удостоверьтесь, что настроили эти инструменты так, чтобы они не удаляли, не помещали в карантин или по-другому не изменяли любые обнаруженные файлы. Многие из таких инструментов выпускаются с опцией только предупреждать пользователя и не предпринимать других действий, поэтому убедитесь, что выбрали эти опции.

Так как монтированный образ предоставляет доступ к своим файлам в режиме только для чтения, вы можете обратиться к этим файлам так же, как к обычным файлам на компьютере, не изменения их содержимого. Поэтому скриптовые языки, такие как Perl, возвращают дескрипторы файла при открытии файлов каталогов, что делает анализ каталогов точек восстановления и упреждающей выборки простым процессом. Perl-скриптом, используемым на работающих компьютерах для выполнения этих функций, нужно только указать соответствующие места. Программа RegRipper, подробно рассмотренная в главе 4, работает так же; эксперт может монтировать созданный файл-образ как накопитель в режиме только для чтения, а затем указать программе RegRipper соответствующие файлы кустов реестра. В примерах, использованных в разделе «Монтирование образа», эксперт указал программе RegRipper каталог H:\Windows\system32\config, где находятся файлы кустов реестра в монтированном образе. Использование инструментов, таких как RegRipper, таким образом позволит проверить места автозапуска в системном реестре на предмет возможных признаков вредоносных программ (для проверки мест автозапуска в файловой системе можно использовать другие инструменты).

Совет

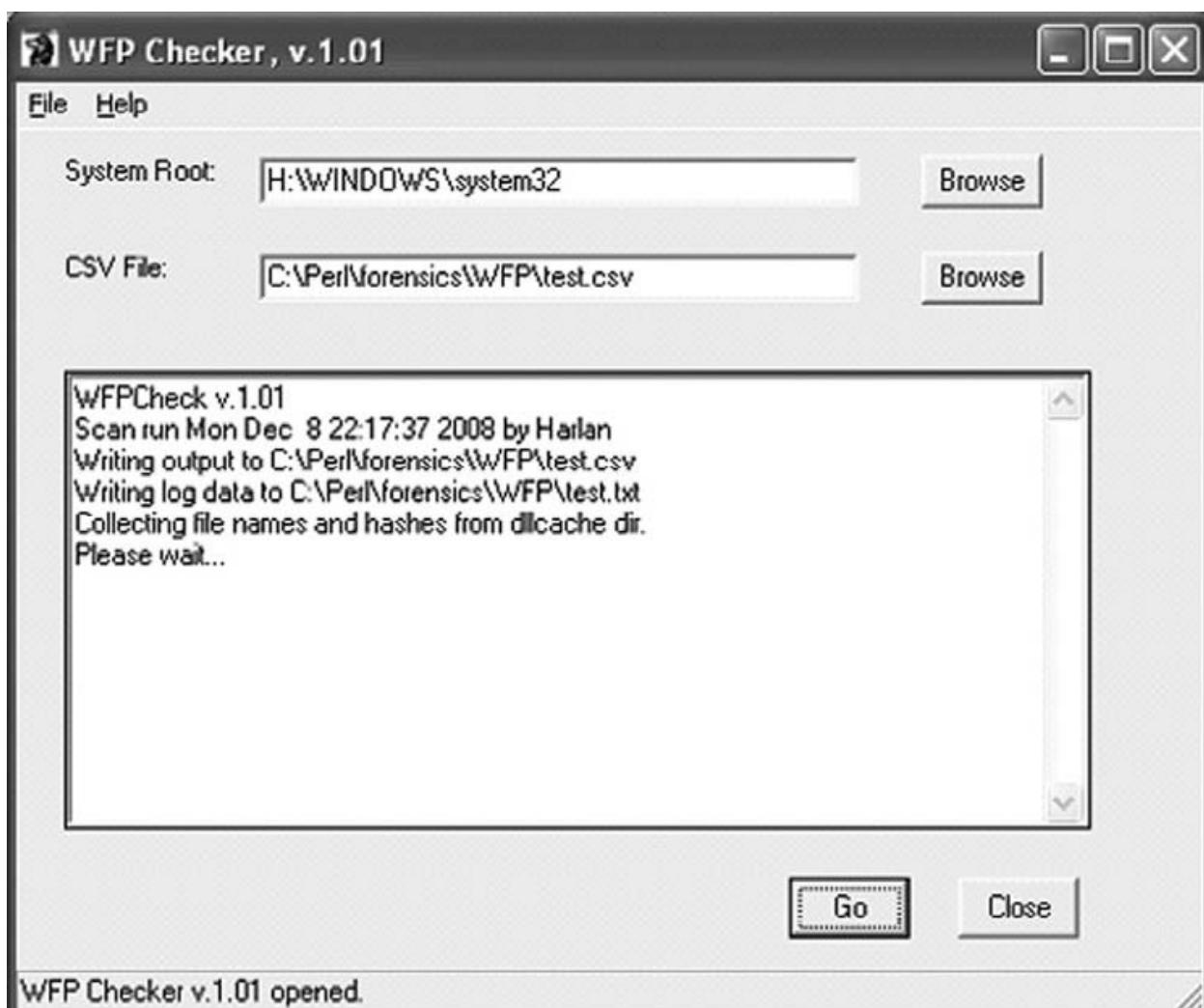
Места автозапуска – это места в операционных системах Windows, позволяющие приложениям запускаться практически без взаимодействия с пользователем. Как вы поняли из главы 4, в реестре Windows, а также в файловой системе есть много таких мест. В главе 1 вы узнали, что можно использовать приложение Autoruns от компании Microsoft (доступное на сайте Sysinternals) или его аналог с интерфейсом командной строки, «autorunsc.exe», для быстрой проверки мест автозапуска на работающем компьютере.

Еще один пример связан с моим инструментом, который называется WFPCheck (WFPCheck не доступен на носителе, идущем в комплекте с этой книгой). На саммите SANS Forensic Summit в октябре 2008 года консультанты компании Mandiant упоминали о вредоносной программе, которая могла заражать файлы, защищенные функцией «Защита файлов Windows» (“Windows File Protection”, WFP), и эти файлы оставались зараженными. Я читал сообщение о подобных вредоносных программах ранее и провел исследование по этому вопросу. Я обнаружил, что существует недокументированный вызов функции API с именем *SfcFileException* (www.bitsum.com/aboutwfp.asp), которая, предположительно, приостанавливает WFP на одну минуту. Функция WFP следит в фоновом режиме за изменениями в файлах и активируется, когда получает уведомление об изменении в одном из определенных защищенных файлов, а не спрашивает защищенные файлы регулярно, чтобы определить, были ли они каким-либо образом изменены. Приостановления функции WFP на одну минуту более чем достаточно для того, чтобы заразить файл, а после того, как WFP возобновляет свою работу, она никак не может определить, что защищенный файл был изменен.

Совет

Функция API *SfcFileException* вызывается инструментом *WfpDeprotect* (www.bitsum.com/wfpdeprotect.php).

В результате я решил создать инструмент WFPCheck («wfpchk.pl»). WFPCheck сначала считывает список файлов из каталога system32\ dllcache и создает хэш-значения MD5 для каждого файла. Затем WFPCheck выполняет поиск всех файлов, сначала в каталоге «system32», а потом во всем разделе (пропуская каталоги «system32» и system32\ dllcache). Когда обнаруживается файл с таким же именем, как и у одного из защищенных файлов, WFPCheck создает MD5-хэш для этого файла и сравнивает его с уже имеющимся хэш-значением. WFPCheck сохраняет свои результаты в файл в формате значений, разделенных запятыми (.csv), и записывает текстовый журнал своих действий в тот же каталог, где находится файл .csv. На илл. 5.20 показана утилита WFPCheck, анализирующая монтированный файл-образ.



WFP Checker v.1.01 opened.

Илл. 5.20. WFPCheck анализирует монтированный файл-образ.

CSV-файл, созданный WFPCheck, состоит из трех столбцов: имя файла из каталога system32\ dllcache, полный путь к месту, где был найден файл (вне каталога «dllcache»), и результаты сравнения. Если WFPCheck обнаруживает, что созданные хэш-значения совпадают, то в столбце результатов будет указано «Совпадение» (“Match”) (см. илл. 5.21). Если хэш-значения не совпадают, то в столбце результатов появится слово «Несовпадение» (“Mismatch”), за которым следуют два набора разделенных двоеточием значений: размеры соответствующих файлов, а затем соответствующие им хэш-значения. Сами эти два набора чисел разделены знаком тире.

| File | Full Path | Result |
|--------------|----------------------------------|--------|
| 12520437.cpx | H:\WINDOWS\system32\12520437.cpx | Match |
| 12520850.cpx | H:\WINDOWS\system32\12520850.cpx | Match |
| 6to4svc.dll | H:\WINDOWS\system32\6to4svc.dll | Match |
| aaaamon.dll | H:\WINDOWS\system32\aaaamon.dll | Match |
| access.cpl | H:\WINDOWS\system32\access.cpl | Match |
| acctres.dll | H:\WINDOWS\system32\acctres.dll | Match |
| accwiz.exe | H:\WINDOWS\system32\accwiz.exe | Match |

Илл. 5.21. Фрагмент выходных данных csv-файла программы WFPCheck.

Предупреждение

В системах, в которых было выполнено обновление, но не были удалены старые файлы, WFPCheck может показать несколько результатов «Несовпадение» (“Mismatch”). Это происходит потому, что при обновлении системы файлы заменяются, и, если более старые версии файлов от предыдущих обновлений остаются в файловой системе, WFPCheck, вероятно, обнаружит, что хэш защищенного файла совпадает с его копией из каталога «`dllcache`», но определит, что хэш-значения более старых файлов с тем же именем не совпадают. При использовании WFPCheck необходимо внимательно изучать полученные данные, так как существует вероятность ошибочных результатов. Чтобы уменьшить количество ошибочных результатов, я написал еще одну версию утилиты WFPCheck с именем WFPCheckf, которая при необходимости извлекает информацию о версии файла из раздела ресурсов файла (объяснение различных разделов исполняемого файла см. в главе 6).

Еще один способ найти потенциальные вредоносные программы в образе данных, монтированном как накопитель в режиме только для чтения, – использовать утилиту «`sigcheck.exe`» от Microsoft (<http://technet.microsoft.com/en-us/sysinternals/bb897441.aspx>). Она проверяет, имеет ли файл цифровую подпись, а также создает дамп информации о версии файла, если таковая имеется. Например, следующая команда проверит содержимое каталога `Windows\system32` в монтированном файле-образе (монтированном как накопитель H:) на предмет всех неподписанных исполняемых файлов:

```
D:\tools>sigcheck -u -e H:\Windows\system32
```

Хотя этот метод не является наиболее точным, его можно использовать вместе с другими применяемыми методами, чтобы провести всесторонний анализ файлов в образе данных при поиске вредоносных программ.

Анализ временной шкалы

Анализ временной шкалы – это легкий для представления и понимания способ установления или связывания последовательности событий, который могут использовать, например, специалисты по расследованию инцидентов. Большая часть работы, выполняемая экспертами, в некоторой степени связана с временем; например, первое, что я делаю, приняв вызов об инциденте, – отмечаю время, в которое мне позвонили. Во время начальной оценки происшествия я пытаюсь определить, когда пользователь, позвонивший мне, впервые получил информацию об инциденте, например, когда он заметил что-либо необычное или принял уведомление из внешнего источника. Эти сведения очень часто помогают мне уменьшить количество информации, которую я буду искать во время анализа, и сузить область поиска (изображения, дампы памяти, журналы и т. д.) в клонированных данных. Все это имеет отношение к данным, имеющим некоторое связанное с ними значение времени или содержащим определенную отметку времени.

Как только специалист начинает расследование инцидента, на него обрушивается огромное количество данных, имеющих отметки времени. Файлы в образе данных содержат отметки о времени их создания, изменения и последнего доступа к ним. Файлы журналов в системе (журналы регистрации событий Windows, журналы веб-сервера IIS, журналы службы FTP, файлы журналов антивирусного приложения и т. д.) содержат записи, в которых указывается время, связанное с определенными событиями. Как вы поняли в главе 4, отметки времени в виде времени *LastWrite* относятся не только к разделам реестра; например, в случае со списком последних использованных файлов, последнее событие связано с отметкой времени *LastWrite* этого раздела. Кроме того, эксперту доступны различные параметры реестра, содержащие в своих данных значения времени, а также имеется довольно много других данных с отметками времени, что

позволяет ему не только получить представление о времени инцидента, но и, возможно, определить дополнительные артефакты инцидента.

Чаще всего эксперты начинают разрабатывать временную шкалу, собирая данные с отметками времени и добавляя их в электронную таблицу. Одно из преимуществ такого вида создания временной шкалы состоит в том, что добавлять новые события по мере их обнаружения – относительно просто, и при этом записи можно сортировать, чтобы разместить их в точной последовательности на временной шкале. Этот процесс также полезен для предварительной обработки данных (так как эксперт добавляет только те значения, которые представляют для него интерес), но он может быть достаточно трудоемким. Кроме того, этот процесс нельзя детально масштабировать, так как, получая данные из современных операционных систем и других источников во время экспертизы, эксперт очень скоро будет завален огромным количеством событий, которые могут не иметь отношения к текущему делу.

Чтобы автоматизировать сбор данных, имеющих отметки времени, можно использовать инструмент fls из пакета The Sleuth Kit (www.sleuthkit.org), созданного Брайаном Кэрриером (Brian Carrier). Инструмент fls перечисляет все каталоги и файлы, а также недавно удаленные файлы в образе данных и выводит результаты в так называемый промежуточный файл (англ. *body file*, см. http://wiki.sleuthkit.org/index.php?title=Body_file). Промежуточный файл можно затем обработать с помощью Perl-скрипта «mactime.pl», чтобы преобразовать файл во временную шкалу в более понятном, текстовом формате. Страница [Timelines](http://wiki.sleuthkit.org/index.php?title=Timelines) на веб-сайте Sleuth Kit (<http://wiki.sleuthkit.org/index.php?title=Timelines>) предоставляет дополнительную информацию об этом процессе.

Показать данные о файлах с отметками времени из образа данных, используя утилиту «fls.exe», относительно просто. С помощью следующей команды можно отобразить выходные данные в консоли (т. е. в стандартном устройстве вывода) в формате скрипта «mactime» или промежуточного файла:

```
D:\tools\tsk>fls -m C: -i raw -f ntfs -l -r d:\cases\xp\xp.001
```

Обратите внимание, что для того чтобы показать только удаленные записи в образе данных, нужно добавить переключатель *-d*. Подробную информацию об использовании fls можно получить на справочной странице для этого инструмента, которая находится по адресу www.sleuthkit.org/sleuthkit/man/fls.html. Выходные данные утилиты fls разделены вертикальной чертой, и их можно перенаправить в файл для хранения или последующего анализа. Ниже показан фрагмент выходных данных fls:

```
0|C:/Program Files/Internet Explorer/Connection Wizard/inetwiz.exe|5091-128-3|r/
rrwxrwxrwx|0|0|20480|1201700419|1057579200|1201700199|1201700199
0|C:/Program Files/Internet Explorer/Connection Wizard/isignup.exe|5092-128-3|r/
rrwxrwxrwx|0|0|16384|1201700420|1057579200|1201700199|1201700199
0|C:/Program Files/Internet Explorer/Connection Wizard/msicw.isp|5107-128-1|r/
rrwxrwxrwx|0|0|158|1201700200|1057579200|1201700200|1201700200
0|C:/Program Files/Internet Explorer/Connection Wizard/msn.isp|5108-128-1|r/
rrwxrwxrwx|0|0|197|1201700200|1057579200|1201700200|1201700200
0|C:/Program Files/Internet Explorer/Connection Wizard/phone.icw|4949-128-3|r/
rrwxrwxrwx|0|0|2921|1201700184|1057579200|1201700184|1201700184
```

Эти данные отображаются в промежуточном файле, и их можно затем проанализировать с помощью Perl-скрипта «mactime», чтобы создать временную шкалу. Отметки времени, связанные с файлом приведены в формат 32-разрядных значений времени UNIX, даже когда сами отметки времени *сохраняются* файловой системой (речь идет об образе накопителя, в котором используется файловая система NTFS) как 64-разрядные значения структуры *FILETIME*. Кроме того, можно проанализировать содержимое промежуточного файла с помощью утилиты Ex-Tip

(<http://sourceforge.net/projects/ex-tip/>), созданной Майклом Клоппертом (Michael Cloppert), и просмотреть выходные данные в несколько другом формате. Мы подробно рассмотрим Ex-Tip в главе 8.

Краткое изложение

Большинство из нас знает, или слышало, что не существует двух одинаковых расследований. Каждое расследование, которое мы проводим, отличается от предыдущего так же, как снежинки. Однако некоторые основные понятия могут быть одинаковыми для всех расследований, а знания о том, где искать подтверждающую информацию, могут иметь важнейшее значение. Слишком часто наша работа зависит от внешних факторов и крайних сроков, поэтому очень важно знать, где искать информацию, подтверждающую те или иные действия, помимо тех данных, что предоставляют программы для судебного анализа данных с графическим интерфейсом пользователя. Из-за недостатка времени и ресурсов многие расследования сводятся к поиску ключевых слов или отдельных файлов, хотя нам могло бы быть доступно большое количество информации, если бы мы знали, где искать и какие вопросы задавать. Помимо отдельных файлов (незаконные изображения, вредоносные программы), мы можем исследовать несколько недокументированных (или малоизвестных) форматов файлов, чтобы получить лучшее представление о том, что и где произошло в системе.

Помимо других аспектов судебного анализа, необходимо знать, где искать данные и где данные должны находиться исходя из того, как операционная система и приложения отвечают на действия пользователя. Знания о том, где должны находиться файлы журналов и какой формат они имеют, помогают получить ценные сведения во время расследования; факт отсутствия этих артефактов является не менее важной информацией.

Отсутствие подробной документации о различных файловых форматах (а также о расширениях определенных файлов) всегда было проблемой для судебной экспертизы. Решение этой проблемы состоит в тщательном, документированном исследовании этих файловых форматов и предоставлении этой информации другим экспертам. Это относится не только к файлам и форматам файлов из операционных систем Windows, исследуемых в данный момент (Windows 2000, XP и 2003), но также и к более новым версиям, как например, Vista.

Быстрое повторение

Файлы журналов

- § Большая часть традиционной компьютерно-технической экспертизы вращается вокруг наличия файлов или фрагментов файлов. Операционные системы Windows сохраняют ряд файлов, которые можно включить в традиционную процедуру просмотра, чтобы обеспечить большую степень детализации анализа.
- § Многие файлы журналов, сохраняемые ОС Windows, содержат отметки времени, которые можно использовать во время анализа временной шкалы действий в системе.

Метаданные файлов

- § Термин *метаданные* означает «данные о данных». Они представляют собой дополнительные данные о файле, которые находятся отдельно от фактического содержимого файла (т. е. отдельно от той области, где эксперт выполняет текстовый поиск).
- § Многие приложения сохраняют метаданные о файле или документе в самом файле.

Альтернативные методы анализа

- § Помимо традиционных средств судебной компьютерной экспертизы, существуют дополнительные методы анализа данных.
- § Загрузка файла-образа в виртуальной среде может предоставить эксперту дополнительные средства как для анализа системы, так и для представления собранных данных другим лицам (например, присяжным).
- § Получение доступа к образу как к накопителю в режиме только для чтения предоставляет эксперту способ для быстрого поиска вирусов, троянских и других вредоносных программ.

Часто задаваемые вопросы

Вопрос: Я выполнял в образе поиск интернет-страниц, просмотренных пользователем, и обнаружил, что профиль «Default User» содержит записи в журнале браузера. Что это означает?

Ответ: Хотя мы не обсуждали историю просмотра интернет-страниц в этой главе (эта тема подробно рассмотрена в других источниках), мне часто задают этот вопрос, и на самом деле я сам сталкивался с этой проблемой при проведении анализа данных. Роберт Хенсинг (Robert Hensing), сотрудник Microsoft, рассказывает об этой проблеме в своем блоге (http://blogs.technet.com/robert_hensing/default.aspx). Вкратце, в профиле «Default User» по умолчанию нет временных интернет-файлов или журнала просмотра веб-страниц. Если в этом профиле обнаружен журнал браузера, это означает, что кто-то с правами доступа на уровне системы использовал функции WinInet API. Я наблюдал это в случаях, когда злоумышленнику удавалось получить доступ на уровне системы и запустить утилиту «wget.exe», чтобы загрузить инструменты на взломанный компьютер. Так как файл «wget.exe» использует WinInet API, очевидно, что журнал браузера находился в каталоге «Temporary Internet Files» учетной записи «Default User». Для демонстрации этой ситуации Роберт предоставляет отличный пример, запуская браузер Internet Explorer как запланированное задание так, чтобы он запускался с учетными данными системы. Затем журнал браузера заполняется данными для учетной записи «Default User». После этого можно провести анализ журнала браузера и файла «index.dat», используя инструмент Internet History Viewer в программе ProDiscover или Web Historian от компании Mandiant.com.

Вопрос: Я создал образ НЖМД, и на первый взгляд мне кажется, что в системе находится мало данных. Как я понял, пользователь, «владевший» этим компьютером, работал в организации несколько лет и недавно уволился при подозрительных обстоятельствах. Судя по дате, сохраненной в реестре, система была установлена приблизительно месяц назад. Какие подходы можно использовать с точки зрения анализа?

Ответ: Это вопрос часто возникает, когда эксперт во время анализа системы считает, что операционная система была переустановлена непосредственно перед тем, как был создан образ. Дело действительно может быть в этом, но, прежде чем остановиться на данной версии, эксперт может исследовать несколько мест, чтобы получить больше данных по этому вопросу. Дата и время установки системы записывается во время установки в параметр *InstallDate* в следующем разделе реестра:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion

Данные, связанные с параметром *InstallDate*, – это двойное слово, представляющее количество секунд, прошедших с начала 1 января 1970 года. В статье № 235162 из базы знаний Microsoft (<http://support.microsoft.com/?id=235162>) говорится, что это значение может быть записано неправильно. Другие места, которые эксперт может исследовать,

чтобы подтвердить это значение, – это отметки даты и времени в записях файла «setuplog.txt», отметки времени *LastWrite* в разделах реестра Service и т. д. Также не забудьте проверить отметки времени *LastWrite* разделов реестра для учетной записи пользователя в файле «SAM». Дополнительные сведения об извлечении информации из реестра (из разделов UserAssist и т. п.) см. в главе 4. К другим областям, подлежащим исследованию, относятся каталог «Prefetch» в ОС Windows XP и отметки времени создания, доступа и изменения в каталогах профиля пользователя.

Вопрос: Я слышал о таком понятии, как *антикриминалистика* – когда кто-то прилагает особые усилия и использует специальные средства, чтобы скрыть улики от судебного эксперта. Что можно предпринять в таких случаях?

Ответ: Теории и выводы судебных экспертов никогда не должны основываться на каком-то отдельном элементе данных. Наоборот, выводы по мере возможности должны быть подтверждены различными фактами. Во многих случаях попытки скрыть данные создают свои собственные артефакты. Эксперт должен понимать, как ведет себя операционная система и какие обстоятельства и события становятся причиной создания определенных артефактов, чтобы обнаружить признаки различных действий. (Обратите внимание, что, если значение изменено, это все равно считается артефактом). Также имейте в виду, что факт отсутствия артефакта в том месте, где он должен находиться, сам по себе является артефактом.

Содержание

| | |
|--|----|
| Введение | 2 |
| Файлы журналов | 2 |
| Журналы регистрации событий | 2 |
| Основные понятия о событиях | 3 |
| Формат файла журнала событий | 8 |
| Заголовок журнала событий | 8 |
| Структура записи о событии | 9 |
| Журналы событий Vista | 14 |
| Журналы IIS | 16 |
| Log Parser | 21 |
| Журнал веб-браузера | 21 |
| Другие файлы журналов | 22 |
| Setuplog.txt | 23 |
| Setupact.log | 23 |
| Setupapi.log | 24 |
| Netsetup.log | 25 |
| Журнал планировщика заданий | 25 |
| Журналы брандмауэра Windows XP | 26 |
| Mrt.log | 28 |
| Журналы программы «Доктор Ватсон» | 29 |
| Cbs.log | 30 |
| Файлы аварийного дампа памяти | 31 |
| Корзина | 31 |
| Корзина в ОС Windows Vista | 34 |
| Точки восстановления системы в Windows XP | 34 |
| Файлы «gr.log» | 34 |
| Файлы «change.log.x» | 35 |
| Служба теневого копирования томов в ОС Windows Vista | 36 |
| Файлы упреждающей выборки | 37 |
| Функция SuperFetch в ОС Windows Vista | 38 |
| Файлы ярлыков | 39 |
| Метаданные файлов | 40 |
| Документы Word | 41 |
| PDF-документы | 45 |
| Файлы изображений | 48 |
| Анализ сигнатур файлов | 48 |
| Альтернативные потоки данных в NTFS | 49 |
| Создание альтернативных потоков данных | 50 |
| Отображение альтернативных потоков данных | 51 |
| Использование альтернативных потоков данных | 55 |
| Удаление альтернативных потоков данных | 56 |
| Альтернативные потоки данных: заключение | 57 |
| Альтернативные методы анализа | 57 |
| Монтирование образа | 60 |
| Обнаружение вредоносных программ | 63 |
| Анализ временной шкалы | 67 |
| Краткое изложение | 69 |
| Быстрое повторение | 69 |
| Часто задаваемые вопросы | 70 |



<http://computer-forensics-lab.org>

Перевод:
Бочков Д.С.
Капинус О.В.
Михайлов И.Ю.