прячем файлы и папки

крис касперски ака мыщъх, no-email

заныкать гейму от начальства и порнушку от жены/детей — весьма потребное дело. сам мыщъх велел! в сети существует множество утилит, специально предназначенных для этих целей (как коммерческих, так и бесплатных), но качество маскировки и удобство пользования в большинстве случаев оставляют желать лучшего, однако, при желании "шапку невидимку" можно смастерить и самостоятельно! штатные средства операционной системы таят в себе множество удивительных возможностей, о которых догадываются далеко не все пользователи.

введение

Степень скрытости информации — один из важнейших критериев, гарантирующий, что спрятанный файл или каталог не будет найден посторонним человеком. Тут следует сделать небольшое отступление и обозначить два подхода к криптографии: ***шифрование*** и ***стеганографию***. Шифрование ставит перед собой задачу кодирования информации таким образом, чтобы прочитать ее без знания ключа было невозможно или чрезвычайно трудоемко. Идея же стеганографии заключается в скрытии самого факта наличия информации в данном месте. Естественно, оба метода можно комбинировать друг с другом, в результате чего, даже если присутствие посторонней информации окажется твердо установленным фактом, без знания ключа ее все равно не удастся расшифровать, разве что применить ректотермальный криптоанализ (паяльник в точку пересечения двух прямых — в смысле ног).

Таким образом, утилиты, шифрующие файлы, папки (и даже целые дисковые разделы целиком) для наших целей совершенно непригодны. У незашифрованной папки с "клубничкой", заныканной в неприметном каталоге с огромной степенью вложенности, есть хорошие шансы долгое время оставаться незамеченной, но стоит только ее зашифровать (например, спрятать в rar-архив), как антивирусы тут же начнут ругаться на всех языках, под которые они локализованы, и у "посторонних человеков" возникнет вопрос: а с чего бы это вдруг вам потребовалось шифроваться? Честным людям скрывать нечего!!! Напротив, если человек постоянно пользуются услугами криптографии — значит, он либо параноик (это хреново, но исправимо), либо он что-то заныкал (а вот за это могут уже и побить!). Чтобы не погореть, мы должны скрывать сам факт наличия сокрытых данных! Вот такая рекурсия получается!

Другим критерием оценки криптографических механизмов является их стойкость ко взлому. Начинающие часто спрашивают: стоит ли устанавливать утилиты типа PGP Disk или аналогичные ему шифровальные средства, против которых бессилен даже Пентагон? Ответ зависит от того, какую именно информацию мы собираемся скрывать и насколько часто планируем работать с ней. Работать с зашифрованной информацией все равно, что хранить золото в сундуке, зарытым под яблоней. Понадобилась сотня баксов — откопал, достал, закопал. И так каждый раз. Слишком утомительно, да и небезопасно, поскольку набираемый пароль могут подсмотреть из-за спины или увести key-logger'ом, причем, короткий словарный пароль элементарно подбирается по словарю, а длинный бессмысленный пароль легко забыть, _навсегда_ лишив себя доступа к зашифрованным данным. Если же его записать, то… листок с паролем (или брелок с flash-памятью) может стать добычей взломщика.

Реально защитить свои данные от спецслужб практически невозможно, да мы и не собираемся этим заниматься. Наши задачи гораздо скромнее: дети, жена, коллеги по работе, начальник, администратор и прочие продвинутые пользователи. Что же касается людей в погонах, то тут все зависит от "повезет" или "не повезет". Если экспертизой изъятого компьютера будет заниматься обычный администратор, то обмануть его проще простого. А вот если жесткий диск передадут фирме, специализирующийся на восстановлении данных, для скрупулезного изучения на секторном уровне, то тут дело труба, однако, подобные клинические случаи мы не рассматриваем.

Существует масса относительно простых, но достаточно эффективных механизмов сокрытия информации (для большинства из которых не требуется никаких дополнительных приспособлений, кроме самой операционной системы). Мыщъх пользуется ими далеко не первый год и за это время хорошо изучил их достоинства и недостатки, предлагая отборные

хакерские трюки, проверенные временем и хвостом, а хвост у мышей, как известно, может достигать до 35 см. Вот!

Рисунок 1 скорость перебора паролей пропорциональна температуре паяльника, расположенного в точке пересечения двух прямых

как это работает или обзор маскировочных утилит

Несмотря на то, что мы собираемся рассматривать способы сокрытия файлов и папок, не требующие установки дополнительного программного обеспечения (что в некоторых случаях попросту невозможно), знать, что у них находится под капотом никому не помешает.

Имеющиеся на рынке утилиты можно разделить на два больших и практически непересекающихся класса. Первые устанавливают свой драйвер (службу, резидентную программу), перехватывающую системные вызовы прямо или косвенно относящиеся к поиску, удалению, открытию файлов. После этого им остается всего лишь "подчистить" содержимое каталогов, исключая из них всякое упоминание о сокрытых файлах и подкаталогах, а так же блокируя прямое открытые/удаление файлов по их имени. Это достаточно надежный механизм, однако, ввиду множества присущих ему недостатков, большой популярности он так и не получил.

Из-за своей схожести с rootkit'ми файловые маскировщики обозначенного типа плохо уживаются с антивирусами, попадая под "статью" вредоносной программы неизвестного типа. А кому это понравится?! Администраторы тут же устраивают суровые разборки с раздачей по

обе стороны от технического прогресса (раздачей занимается бригада каратистов быстрого реагирования). К тому же, корректный перехват системных функций реализовать очень сложно и за каждую ошибку приходится расплачиваться нестабильной работой оси и прочими глюками, от которых пользователи совсем не в восторге.

Из коммерческих продуктов, работающих на "топливе" этого вида, наибольшую популярность завоевал Symantec SystemWorks, поддерживающий опцию "Norton Protected Recycle Bin" и создающий скрытый каталог "NPROTECT" в обычной корзине. "Скрытый" не в смысле атрибута "hidden", а реально скрытый от системы и доступный только ему одному. (подробнее об этом и многом другом можно прочитать на блоге Макра Руссиновича: http://blogs.technet.com/markrussinovich/archive/2006/01/15/rootkits-in-commercial-software.aspx).

Другой класс утилит вообще не дотрагивается до системных функций и прячет файлы с каталогами путем поклажи их в специальный контейнер, обычно представляющий собой обычный файл данных, с которым маскировочная программа работает через свой собственный интерфейс, выполненный в стиле "проводника". А чтобы посторонние люди не добрались до спрятанных файлов, они защищаются паролем. Грубо говоря, это тоже самое, что обычный запароленный RAR. Факт сокрытия файлов никак не маскируется и чтобы работать с файлами их нужно извлечь на диск, что не только непрактично, неудобно, но еще и небезопасно (поскольку, следы присутствия извлеченных файлов на диске легко обнаружить любой утилитой типа R-Studio, умеющей восстанавливать удаленные файлы).

Тем не менее, в силу чрезвычайной простоты технической реализации, такой принцип сокрытия чрезвычайно популярен среди разработчиков "маскировочных программ". Знать устройство операционной системы, владеть ассемблером, уметь писать драйвера — не требуется. Достаточно поерзать мышью в DEPLPHI и вот — программа готова!!! Типичным представителем данного класса утилит является условно-бесплатная программа HidesFiles, ознакомиться с которой можно на http://www.hidesfiles.com (см. рис. 2).

Однако, ни ей, ни легионом ее сородичей пользоваться не рекомендуется, поскольку качество шифрования оставляет желать лучшего. В некоторых случаях (для достижения наивысшего быстродействия) никакого шифрования вообще не производится и эталонный пароль хранится в "архиве" открытым текстом (ну или не сам пароль, а его контрольная сумма. несмотря на то, что восстановить пароль по контрольной сумме невозможно, для доступа к данным пароль не нужен, поскольку они лежат в незашифрованном виде, достаточно дизассемблировать программу, чтобы реконструировать формат файла данных и дальше можно работать с ним в обход "пользовательского интерфейса"). Разумеется, "домашние пользователи" на такое не способны, а потому для защиты от них хватит и HidesFlies.

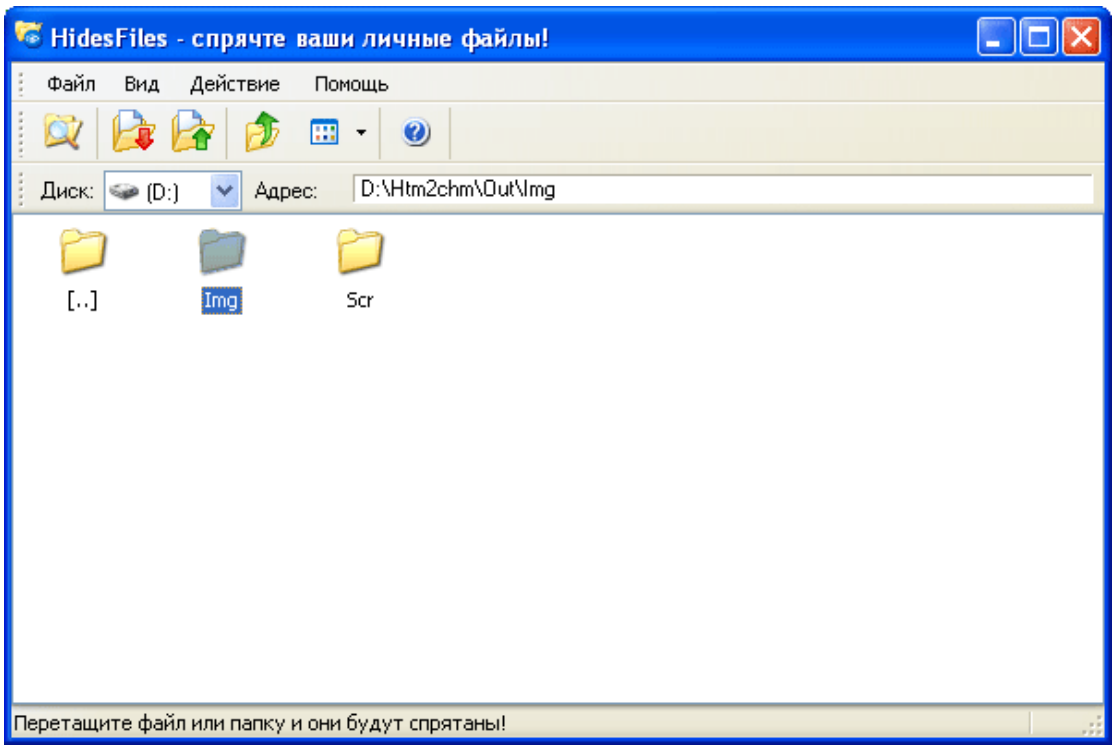

Рисунок 2 внешний вид программы HidesFiles, прячущей файлы и папки от посторонних глаз

>>> врезка игры с pkzip'ом

Классический способ маскировки игрушек и порнушки сводится к упаковке файлов в какой-нибудь архив (например, ZIP или RAR) с последующим изменением расширения на .DAT или что-то вроде того. Считается, что тупой администратор поведется как пионер и ни о чем не догадается. Три раза "ха"! Администратор, быть может, и не догадается (он ведь и не подряжался проверять все файлы в системе), но антивирусы со включенной опцией "проверять архивы" проигнорируют расширение и опознают тип архива по его содержимому, в результате чего в протокол попадет полный список проверенных файлов и архив с нетипичным расширением тут же привлечет к себе внимание, после чего будет безжалостно удален.

Чтобы избежать расправы рекомендуется вооружиться hiew'ом (или любым другим hex-редактором) и изменить первые два байта заголовка на что-нибудь такое… эдакое… например, просто поменять их местами, чтобы не забыть (см. рис. 3).

Теперь (после смены расширения) ни антивирус, ни даже сам архиватор ни за что не сможет догадаться об истинном формате файла и спокойно пропустит его даже не жуя. Единственная зацепка, могущая вызывать подозрения у администратора — это неприлично огромный размер файла. Стратегия поиска сокрытых данных в стиле "найди десятку самых длинных файлов и присмотрись к ним повнимательнее" палит незадачливых хакеров только так! Поэтому, всегда разбивайте архив на несколько файлов разного размера, подбирая ее так, чтобы они не слишком выделялись среди остальных.

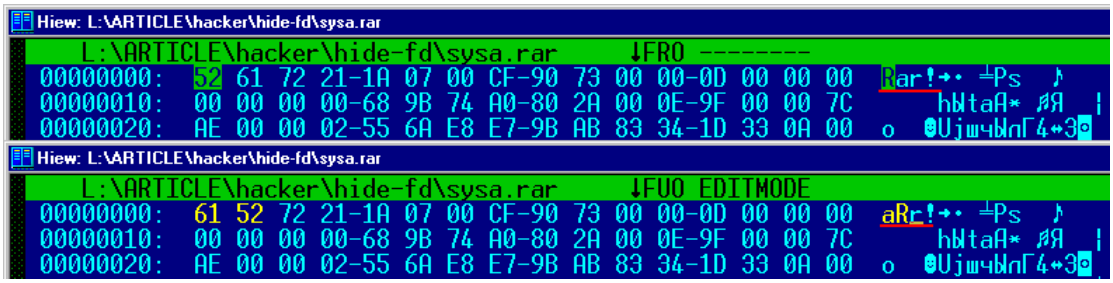

Рисунок 3 перестановка местами первых двух байт заголовка архива препятствует его распаковки (вверху показан оригинальный файл, внизу — измененный)

монтирование и демонтирование дисковых томов

Операционные системы семейства NT (к числу которых относится сама NT, W2K, XP и Vista) поддерживают механизм монтирования (mount) дисковых томов (аналогичный тому, что имеется в UNIX), однако, в отличии от UNIX'a, в NT диски монтируются автоматически при их подключении (для несъемных носителей это означает, что они монтируются всегда) и пользователю нет никакой нужды задумываться об этом.

Между тем, если размонтировать дисковый том (грубо говоря, "отобрать" у него букву) он исчезнет из "Моего компьютера", FAR'a и чтобы получить доступ к его содержимому необходимо выполнить операцию монтирования (о которой осведомлены далеко не все администраторы, не говоря уже за рядовых пользователей). Причем, это совершенно законная и абсолютно безопасная операция!!!

Хорошая идея — при разбивке диска выделить один раздел под секретные файлы, монтируя его только на время работы с ними. Ни антивирусы, ни дисковые доктора, ни прочие анти-хакерские средства не заподозрят и следов "измены". В принципе, изменить разбивку диска можно и на лету, достаточно воспользоваться PQMagic или аналогичной программой. (***внимание***: *_все_ программы, разбивающие диск на лету не застрахованы от ошибок и могут угробить один или несколько разделов, без малейших шансов на восстановление, поэтому обязательно зарезервируйтесь перед разбиением*).

Монтировать диски можно как из командной строки, так и через графический интерфейс, что намного нагляднее. Вот с него-то мы и начнем! Итак, "пуск" → "настройка" → "панель управления" → "администрирование" → "управление компьютером". Во вкладке

"Структура" находим "Управление дисками" и смотрим какие диски есть на нашем компьютере (см. рис. 4).

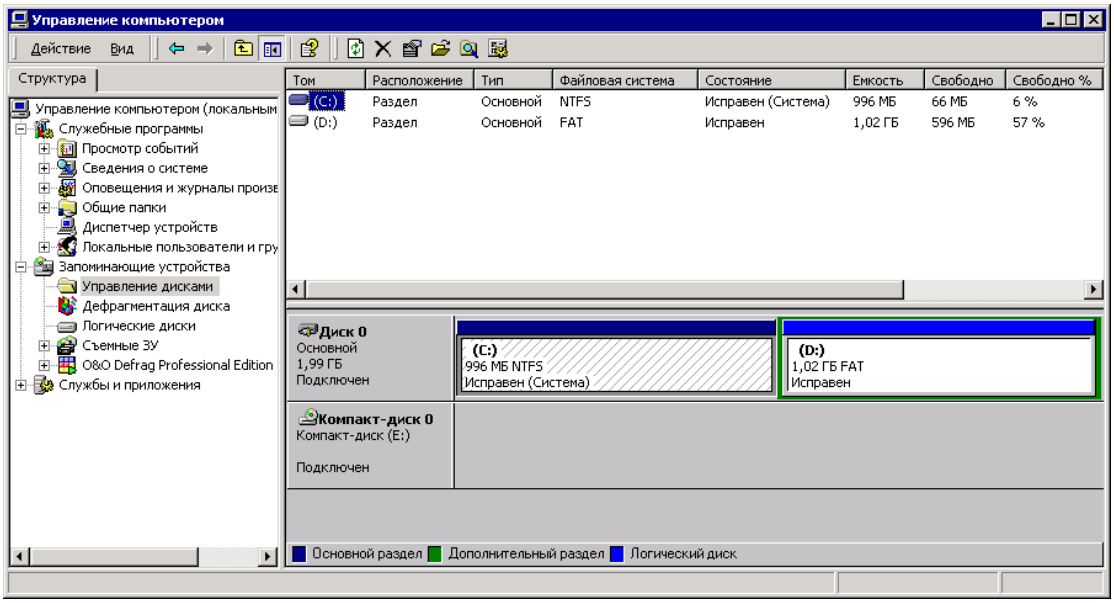

Рисунок 4 управление дисками через консоль администрирования

Допустим, мы хотим демонтировать диск "D:". Щелкаем по нему левой клавишей, в открывшемся контекстом меню выбираем пункт "Изменение буквы диска и пути диска". Появляется еще одно диалоговое окно, с выделенной буквой диска и кнопками: "добавить", "изменить", "удалить" и "зарыть". Нажимаем "удалить", после чего закрываем окно, выходим из системы "управления компьютером" и видим, что диск "D:" не отображается ни в "проводнике", ни в FAR'e, ни в командной строке.

Выполняем перезагрузку (если есть такое желание), убеждаясь, что при загрузке системы удаленный диск более не монтируется в автоматическом режиме. Как же его вернуть обратно?! Очень просто! Заходим в "управление дисками" (как было показано выше), щелкаем мышью по безымянному прямоугольнику, выбираем "Изменение буквы диска и пути диска", нажимаем кнопку "добавить" и выбираем любую букву из передоложенных. Совершенно необязательно выбирает именно диск "D:" это вполне может быть и "X:" (а почему бы и нет?), и даже путь к существующему NTFS-каталогу (но о каталогах мы поговорим позднее).

После нажатия на "ОК" смонтированный диск тут же появится в "моем компьютере", FAR'e и остальных менеджерах. Перезагружать компьютер для этого не требуется (правда некоторые программы, перечисляющие диски при запуске, могут не заметить появления нового диска, поэтому их следует закрыть и открыть вновь — впрочем, таких программ очень немного и с каждым днем становится все меньше и меньше).

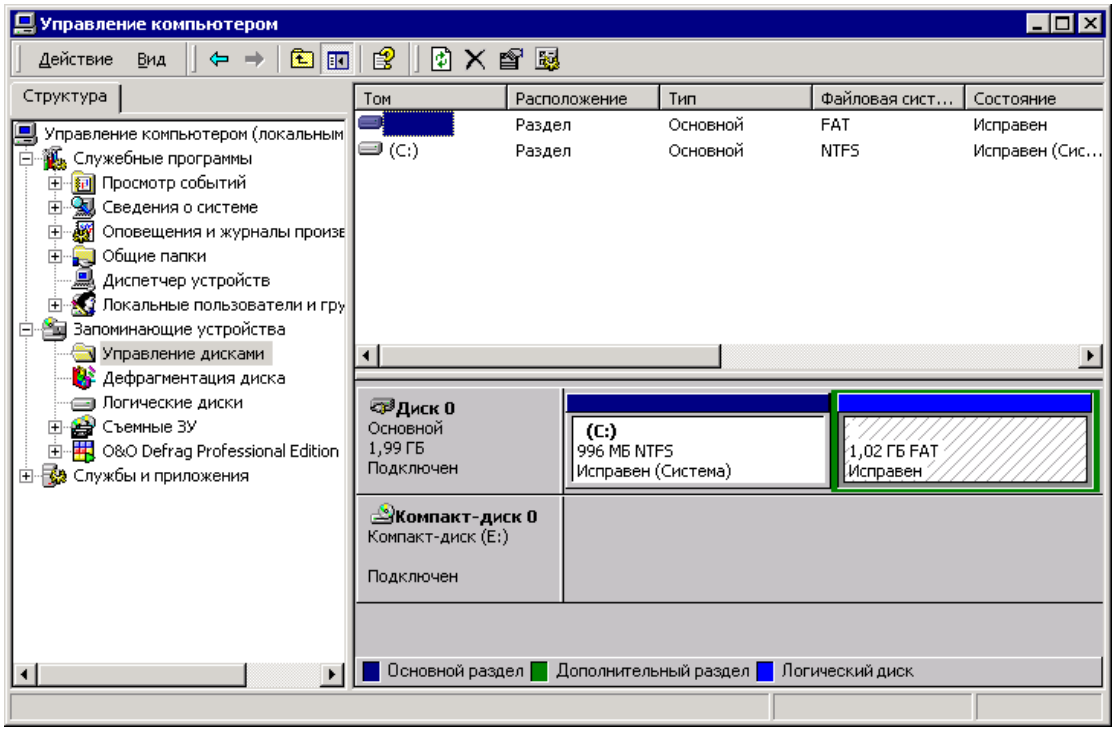

Рисунок 5 теперь с диском D: уже не ассоциирована никакая буква и доступ к нему из "Моего компьютера" или FAR'а невозможен!

А теперь перейдем к командной строке, которая удобна тем, что монтировать/размонтировать диски можно не только мышью, но и командным файлом, запускаемым одним щелчком или даже вызываемым "горячей" комбинацией клавиш).

Начиная с W2K (а, может быть, и еще раньше), в комплект штатной поставки системы входит утилита **moutvol.exe**, которая, как и следует из нее названия, предназначена для (раз)монтирования дисков.

При запуске без параметров она выдаст список логических дисков вместе с назначенными им буквами или точками подключения (см. рис. 6).

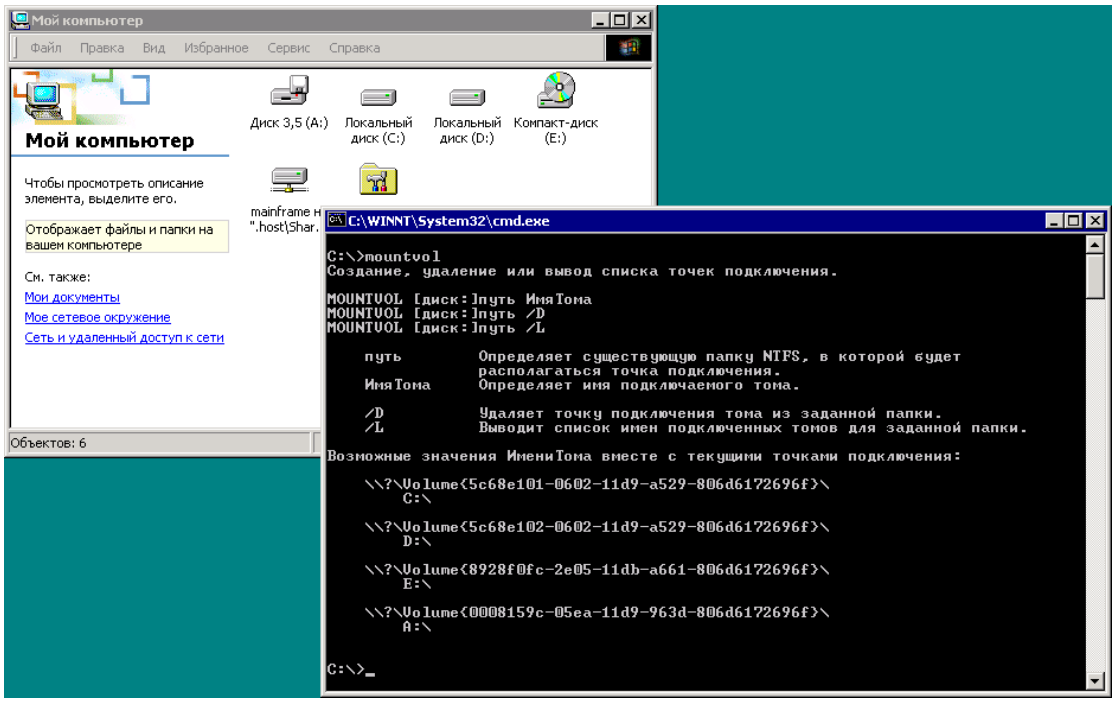

Рисунок 6 просмотр логических дисков и точек подключения с помощью штатной утилиты mountvol.exe

Абракадабра в стиле "\\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\" представляет собой идентификатор тома, используемой системой при "общении" с ним на низком уровне. Формально, NT позволяет нам (в порыве мазохизма) указывать идентификатор диска вместо его буквы (не зависимо от того, есть у него эта буква или она удалена), однако, это слишком громоздко, непотребно и неудобно (см. листинг 1).

```
$dir /w \\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\
 Том в устройстве \\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\ имеет метку BACKUP
 Серийный номер тома: 48FF-7B94

 Содержимое папки \\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\

chasingamy.srt               [dex]                        [disk-cd]
[distr]                      [emule]                      Forth_Java.txt
Forth_Java.zip               [game]                       IFRAME.exp.doc
[masm32]                     [OLD-OLD]                    sexgo.exe
sex_do_it.bat                SubRip.srt                   [systemV]
[temp]                       новый год и новые планы.eml
               8 файлов        322 679 байт
               9 папок   3 331 104 768 байт свободно
```

Листинг 1 "прямое" обращение к диску по его идентификатору

Для размонтирования диска (удаления буквы) достаточно вызвать mountvol.exe с ключом /D (от delete – удаление) и буквой удаляемого диска (см. листинг. 2), после чего диск тут же исчезнет из "моего компьютера".

```
$mountvol.exe D: /D
```

Листинг 2 размонтирование логического диска D:

Хорошо! Диск "D:" успешно удален. Остается только разобраться как вернуть его обратно. Нет ничего проще! Вызываем mountvol.exe идентификатором логического диска и буквой, которую мы собираемся ему присвоить. Ой, а если мы не помним этот длинный и ужасный идентификатор — что делать нам тогда?! Ну не помним, так не помним! Невелика беда. Запускаем утилиту mountvol.exe без ключей и она выводит список всех идентификаторов (см. рис. 7), причем рядом с идентификаторами, с которыми не ассоциирована ни одна буква, будет торчать надпись "нет точек подключения". Выделяем мышью все от одинарной до двойной косой черты и нажатием левой кнопки мыши вставляем в командую строку следом за командой "mountvol.exe D:" (см. листинг 3).

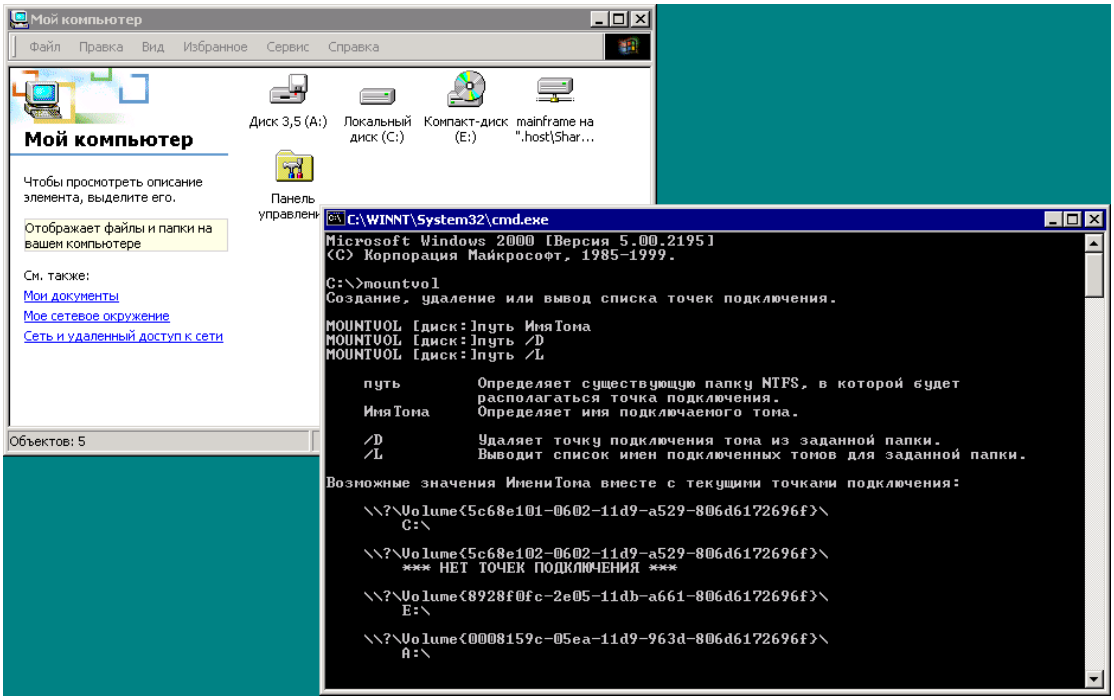

Рисунок 7 просмотр идентификаторов размонтированных логических дисков

```
$mountvol.exe D: \\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\
```

Листинг 3 монтирование размонтированного логического диска

Как мы видим, диск "D:" исправно появляется в "моем компьютере" и в прочих местах. Хочешь — работай с ним как с обыкновенным диском, не хочешь — форматируй! Впрочем, нет, форматировать лучше не надо. Пришло время познакомится с одной удивительной возможностью NT, поддерживаемой всеми UNIX'ми, но отсутствующей в линейке 9x. А именно — возможности монтирования логического диска на папку другого диска.

Зачем это может понадобиться?! Все мы привыкли к тому, что имеются диски "C:", "D:", "E:", однако, некоторые считают такой расклад неудобным и с удовольствием предпочли бы работать с одним логическим диском. Или вот… возьмем такой жизненный случай. У нас есть диски C: и D:, причем на C: свободного места нет совсем, а на D: его завались. Было бы здорово перенести часть программ на D:, но… это невозможно сделать без их переустановки, поскольку практически все они привязываются к букве диска на который установлены. Однако, могущество операционной системы NT позволяет преодолеть это ограничение весьма простым и элегантным путем.

Передоложим, на диске C: находится каталог C:\KPNC, содержащий множество установленных приложений. Что мы делаем? Переносим их в корневой каталог диска "D:" (так чтобы каталог C:\KPNC) оказался пустым, после чего удаляем букву D: и монтируем логический диск на каталог C:\KPNC, подставляя его вместо буквы диска (см. листинг 4):

```
$REM размонтируем D:
$mountvol.exe D: /D

$REM монтируем логический диск на каталог C:\KPNC
$mountvol.exe C:\KPNC \\?\Volume{fd9b89a6-5675-11d9-9ed9-bd1445c469e4}\
```

Листинг 4 монтирование логического диска на каталог

Диск "D:" послушно удалился из "Моего компьютера" и его как будто бы нет, однако, открыв каталог C:\KPNC мы увидим его содержимое в целости и сохранности. Со всеми программами которые мы туда перенести и они (программы в смысле) продолжат работать как ни в чем ни бывало, поскольку с их точки зрения совершенно ничего не изменилось.

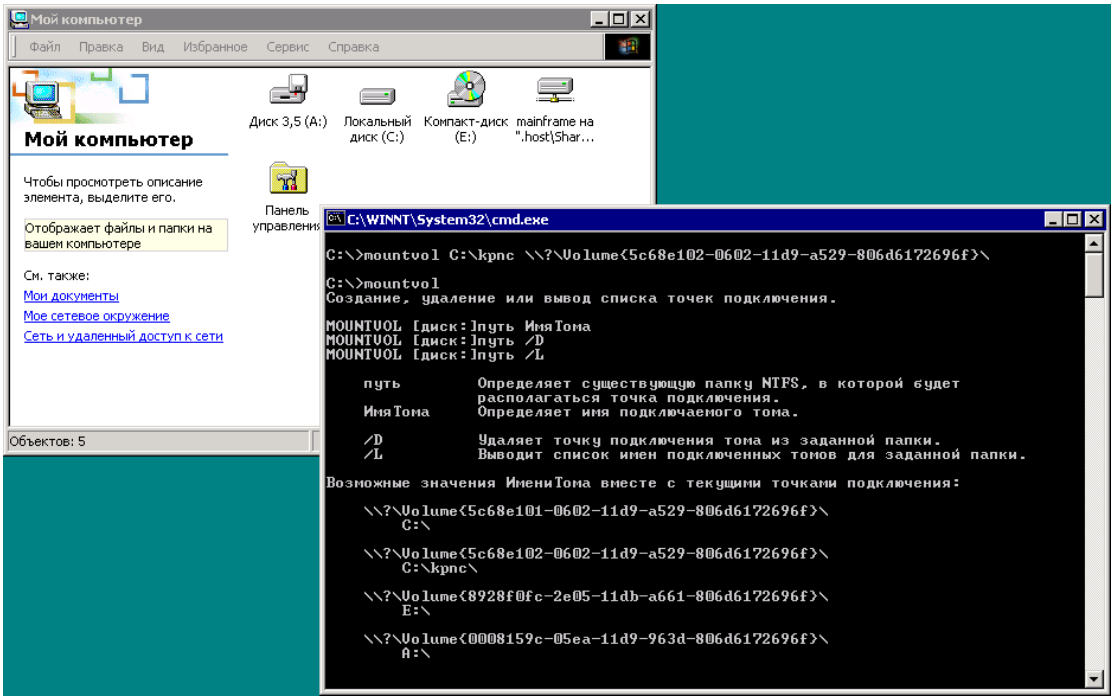

Рисунок 8 логический диск (в девичестве диск "D:") смонтирован на каталог C:\KPNC

Два маленьких замечания на последок. Первое: для монтирования/размонтирования дисков необходимо обладать правами администратора. Второе: монтирование диска на непустой каталог невозможно, так что даже не пытайтесь это делать.

>>> врезка строго дозирование разрушение индексов (для профессионалов)

Файловая система NTFS примечательна тем, что в ней нет каталогов (в том понимании, которое вкладываем в этот термин MS-DOS). в NTFS есть только индексы, и как в любой базе данных, индексы являются лишь _вспомогательными_ структурами данных, используемые для быстрого поиска (вывод содержимого заданной директории есть ничто иное как операция поиска принадлежащих ему файлов и подкаталогов).

Причем в NTFS можно индексировать не только директории, но и другие атрибуты, например, индексировать файлы по размеру (правда, это не реализовано в текущих версиях драйвера).

Основная информация о файлах и порядке их размещения на диске хранится в служебном файле, именуемом $MFT, который содержит _все_ необходимое для нормального функционирования FS, в том числе и каталоги, которые являются обычными файлами, ну... или практически обычными...

Индексы _дублируют_ их содержимое. При открытии файла поиск идет только по $MFT, поскольку как известно поиск в линейном массиве намного быстрее, чем обход кучи деревьев, разбросанных по диску ;) тем более, не нужно забывать, что у файла может быть несколько имен, но индексы допускают лишь включение одного. т. е. файл, имеющий несколько атрибутов имен (типа MS-DOS-имя, POSIX-имя, NTFS-имя) крути не крути, а нужно искать в $MFT. Индексы используются командой DIR, но API-функции операционной системы (такие как CreateFile, CreateProcess) "пляшут" от $MFS, поэтому можно скрыть папку Windows, не нарушив ее работоспособность!!!!

Это достигается за счет удаления каталога из индексов любым подходящим дисковым редактором (например, NtExplorer от Runtime Software) – тогда она не будет отображаться _нигде_, однако, "прямой" запуск (путь + имя файла) продолжит нормально работать. Другим словами, если мы удаляем папку Windows из индексов, команда "DIR C:", как и следовало ожидать, не покажет ее, но вот "DIR C:\Windows" отработает нормально. Впрочем, при первом же запуске chkdsk'a он ее "вылечит", восстановив недостающую запись в индексах. Ну тут, правда, его можно обломать, только тогда придется перестраивать кучу структур данных на диске (подобнее о которых можно прочитать в моей книжке "восстановление данных —

практическое руководство"), а это утомительно и небезопасно. Зато такое сокрытие не требует присутствия резидентов в памяти, абсолютно безглючно и ничем (кроме chkdsk'a) не обнаруживаться...

Еще на NTFS-разделах можно создавать "виртуальные папки" (термин взят мыщъхем из мира web). Допустим мы имеем папку "C:\X\Y\". Так вот, папка Y существует, а X - нет. Точнее, она существует, но у нее удален uplink на корень C:\ и потому добраться до Y можно только зная полный путь. Chkdsk на это внимание уже не обращает ;)

Перейти в папку C:\X так же нельзя, "DIR C:\X" скажет, что нету здесь никакой папки X и отродясь не бывало. Таким образом, Y надежно скрыта от глаз пользователя и антивирусов. В ней же можно держать все, что угодно... правда, антивирусы сканируют запущенные процессы и потому положить в нее резидента не получится, как не получится положить троянский плугин к IE, поскольку нам придется прописывать полный путь, а его-то аверии захавают... Но зато спрятать игрушку, видофильмы или прочую порнушку можно без труда и напряжения мозговых извилин.

Во всяком случае на W2k мыщъх проделывал этот трюк неоднократно, особенно на ZIP-дискетах — не путать с PKZIP архиватором! (была необходимость пронести на них исполняемые файлы, перед этим отдав их на проверку администратору. ну он посмотрел, — а там чисто одни нормальные файлы, никакие не исполняемые ;) ну и... кстати, с ZIP'ом намного более безопасно экспериментировать, чем с hdd, и он быстрее чем виртуальный диск под VMWare...

заключение

Помимо рассмотренных способов сокрытия файлов и папок существует масса других привлекательных трюков, однако, рассмотреть их в рамках скромной журнальной статьи нет никакой возможности, тем более, что всякий описанный трюк утрачивает свою "магическую" силу, и найти спрятанный файл сможет любой желающий. Напротив, трюки, выдуманные вами самостоятельно, как правило, оказываются намного более надежными. Их не берут ни антивирусы, ни суровые администраторы, ни жены, ни начальники. А вот дети — находят! Невероятно, но факт!!! Так что играть в прятками с детьми взрослым сложнее всего. Дети мыслят совсем не так как мы. У них гибкий мозг, еще не засранный штаммами, природная любознательность и потрясающая наблюдательность. Так что лучше всего смотреть порнушку вместе с детьми. Все равно ведь раскопают! Главное — чтобы жена не узнала! (Хотя это, впрочем, смотря какая жена — в этом мире все относительное. Даже невидимые истребители модели Stealth появляются на длинноволновых радарах. Вот тебе и невидимка!).

Рисунок 9 файл-невидимка