КОМПЬЮТЕР ГЛАЗАМИ ХАКЕРА

3-е издание

Михаил Флёнов

Модификация операционной системы Windows

Разгон компьютера

Атаки хакеров и защита

Форсирование Интернета

Компьютерные шутки

+ Материалы на www.bhv.ru

bhv®

УДК 681.3.06
ББК 32.973.26-018.2
Ф70

Флёнов М. Е.

Ф70 Компьютер глазами хакера. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 272 с.: ил.

ISBN 978-5-9775-0790-5

Рассмотрены компьютер, операционные системы Windows XP/Vista/7 и Интернет с точки зрения организации безопасной и эффективной работы на ПК. Описаны основные методы атак хакеров и рекомендации, которые позволят сделать компьютер быстрее, надежнее и безопаснее. Представлены примеры накручивания счетчиков на интернет-сайтах и методы взлома простых вариантов защиты программ Shareware. Приведены советы хакеров, которые позволят при путешествии по Интернету не заразиться вирусами и не стать добычей сетевых мошенников, владеющих методами социальной инженерии. Показано, как сделать интерфейс Windows более удобным и привлекательным, компьютер — надежнее и быстрее, а работу в сети — более эффективной. В третьем издании добавлены новые примеры для операционной системы Windows 7. На сайте издательства находятся программы, описанные в книге, а также используемые файлы и дополнительные статьи.

Для пользователей ПК

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Анна Кузьмина*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Наталия Першакова*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Марины Дамбиевой*
Зав. производством	*Николай Тверских*

Подписано в печать 31.01.12.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 21,93.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0790-5

Оглавление

ГЛАВА 1

Интересные настройки Windows

Будем двигаться от очень простого к простому, ведь все на самом деле очень примитивно, если не усложнять себе жизнь. Поэтому после прочтения книги даже на те вопросы, на которые вы не знали ответа, вы сможете воскликнуть: "Как же это было просто!" Я ничего особого изобретать не намерен, а просто соберу интересные (на мой взгляд) темы относительно компьютера в одной книге. И начнем мы с интерфейса ОС Windows и его программ.

Когда я первый раз познакомился с Windows 95, то понял, что полюбил эту ОС по самые иконки. Несмотря на то, что она была нестабильна и выдавала синие экраны, да и переустанавливать ее приходилось раз в пару месяцев, в ней было очень много удобных для простого пользователя и заядлого хакера вещей.

С появлением следующих версий, таких как Windows 98, 2000, моя любовь только укреплялась. С каждой новой версией система усложнялась, и появлялись новые, интересные возможности для выражения своей индивидуальности. Нестабильность и проблемы иногда склоняли меня установить Linux и работать в нем, но с появлением Windows XP я понял, что ни о каком дистрибутиве в "красной шапке" можно больше и не думать. Лучше заплатить подороже, но получить отличную, удобную и стабильную систему. Главное — подойти с правильной стороны и все строго настроить. А тут есть где "разгуляться", и не только для повышения надежности, но и с целью улучшения внешнего вида.

Начиная с Windows Vista, количество интересных изменений, которые можно выполнить в визуальном интерфейсе, сильно сократилось, поэтому и эта глава сильно изменилась по сравнению с предыдущими изданиями. На момент написания этих строк во всем мире семимильными шагами идет переход на Windows 7, а через год нам обещают еще одну новую версию — Windows 8. Архитектурно Windows уже не должна так сильно меняться, как это произошло при переходе с Windows XP на Vista, поэтому бóльшая часть описываемого здесь может быть применима и в ближайших будущих версиях ОС от Microsoft.

Но вернемся к Linux, который я только что упомянул. Если честно, то в Linux я иногда посиживаю, но в последнее время все реже и реже. Процентов 90 своего

компьютерного времени я провожу непосредственно в Windows, и только 10 процентов уходит на Linux в его различных проявлениях.

В предыдущем издании эта глава содержала много информации, касающейся Windows XP и IE версии 6.0, которые сейчас используются все меньше и меньше. Я думаю, что к моменту выхода книги на полки магазинов количество компьютеров на Windows XP сократится еще сильнее, поэтому оставлять устаревшую информацию не имеет смысла. Но чтобы не терять ее вовсе, я вынес все, что касается Windows XP и Internet Explorer до версии 7, в архив, который можно найти на FTP-сервере издательства (см. **ftp://85.249.45.166/9785977507905.zip**, ссылка доступна также со страницы книги на сайте **www.bhv.ru**). Ищите там множество дополнительной информации в папке Doc.

В этом издании я решил значительно сократить этот раздел, потому что параметры постоянно изменяются, и не хотелось бы, чтобы большая часть книги устарела уже завтра. Вместо этого мы рассмотрим наиболее интересные настройки системы.

1.1. Internet Explorer

Большинство программ устанавливается с настройками по умолчанию. И если в основном производители программного обеспечения предоставляют к своим настройками полный визуальный интерфейс, то Microsoft почему-то решила не делать этого. Не все настройки можно изменить визуально в окне параметров, иногда приходится изменять что-то напрямую в реестре. Как любит говорить qa (quality assurance), с которым я сотрудничаю сейчас на работе: "Don’t ask me why it works this way". Реально, иногда очень сложно объяснить, почему менеджеры проектов или разработчики приняли именно такое решение.

Популярный (пока еще) в Windows-мире браузер Internet Explorer, который устанавливается с ОС по умолчанию, грешит такой же проблемой. У него далеко не все параметры можно изменять в окне настроек. Некоторые (иногда очень интересные и полезные параметры) доступны для изменения только напрямую в реестре.

1.1.1. Убить нельзя, помиловать

Среди настроек есть такой параметр, который запрещает пользователю закрывать окна Internet Explorer. Во время путешествия в сети на многих сайтах выскакивает масса всплывающих окон, которые засоряют экран. Если использовать возможность такой настройки, то окна будут только плодиться, а при попытке закрытия появится окно с предупреждением, как на рис. 1.1.

Чтобы сделать Internet Explorer незакрываемым, нужно перейти в реестре в раздел **HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions**. Этот путь может отсутствовать, и нужно будет добавить недостающие разделы. Для этого достаточно щелкнуть правой кнопкой мыши на нужном разделе и в появившемся меню выбрать **Создать | Раздел** (New | Key). Например, если у вас существует только путь **HKEY_CURRENT_USER\Software**

Policies\Microsoft, то щелкните правой кнопкой на строке **Microsoft** и создайте раздел **Internet Explorer**, а затем в нем — **Restrictions**. Когда все разделы будут существовать, создайте параметр **NoBrowserClose** типа `DWORD` и со значением `1`.

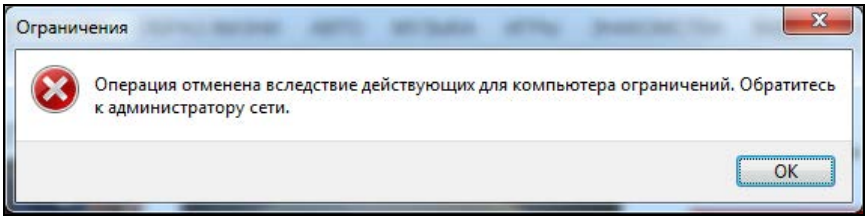

Рис. 1.1. Предупреждение о невозможности закрытия окна IE

Все эти действия можно проделать на компьютере своего друга и посмотреть за его реакцией, когда он попытается закрыть окно. Я однажды подшутил так над своими коллегами по работе. Реакция их была разнообразной. Большинство посчитало, что это было вмешательство вируса.

Чтобы внести все эти изменения на компьютере пользователя, нужно достаточно много времени, а его может и не быть. Чтобы сделать все быстро и незаметно, можно поступить следующим образом:

1. Внести изменения сначала в свой компьютер.
2. Выполнить экспорт файла реестра с опцией **Выбранная ветвь** (в данном случае ветвь **HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\Restrictions**).
3. Записать получившийся файл на флешку.

Теперь идите к компьютеру пользователя, над которым вы намереваетесь подшутить, и говорите, что хотите что-то показать. Вставляете флешку и запускаете файл с расширением reg. Вся необходимая информация будет автоматически добавлена. Не надо больше ничего делать, даже запускать Internet Explorer. Просто скажите, что это не та флешка, и уходите. Ждите, пока пользователь сам не запустит браузер и не встретится с проблемой закрытия программы.

Мне интересно узнать, чем руководствовался тот человек, который придумал ограничение, запрещающее закрывать IE? Я бы с удовольствием поговорил бы с этим человеком, чтобы узнать, для чего это было сделано. А те, над кем подшутили подобным образом, наверно открутили бы нерадивому разработчику голову.

Этот параметр существовал в IE6, и я думал, что его добавили по глупости и уберут из Internet Explorer уже в 7-й версии. Сегодня я проверил 9-ю версию и трюк все еще работает. Немного странно он начинает работать, совсем не сразу. Видимо, браузер кеширует свои параметры и не читает их каждые пять минут или при каждом запуске. Я даже сначала подумал, что параметр не работает, и уже собирался удалить этот раздел из книги. Но прошло некоторое время и я не смог закрыть браузер, а раздел вернулся в книгу.

1.1.2. Количество потоков для скачивания

По умолчанию Internet Explorer очень сильно ограничен в выборе максимального количества подключений из одного процесса к серверу. В зависимости от протокола, версии браузера и подключения это значение может быть от 2 до 6. Конечно же, чем больше установить одновременных подключений между сервером и клиентом, тем лучше и, может быть, даже быстрее. Сервер так же может быть ограничен по скорости на каждое подключение и качать в два и более потока, что способно принести выгоду.

Но если вам значения по умолчанию не достаточно, то его легко можно увеличить и до 10 всего лишь небольшим изменением в реестре.

Открываем редактор реестра regedit и в **HKEY_LOCAL_MACHINE** переходим в ветку **SOFTWARE\Microsoft\InternetExplorer\MAIN\FeatureControl**. Здесь есть два подраздела, которые могут нас заинтересовать:

- **FEATURE_MAXCONNECTIONSPERSERVER** — максимальное количество подключений к серверу по протоколу HTTP 1.1;
- **FEATURE_MAXCONNECTIONSPER1_0SERVER** — максимальное количество подключений к серверу по протоколу HTTP 1.0.

В обоих разделах есть параметр с именем **explorer.exe**, который задает количество одновременных подключений из одного процесса к серверу. Интересно, что для более старого протокола 1.0 это значение равно 4, а для более нового протокола это значение равняется всего лишь 2. Почему такая несправедливость? Она связана с тем, что для протокола 1.0 значение выбиралось опытным путем по поведению браузеров. А вот для версии 1.1 количество потоков в 2 было обусловлено стандартом, только вот стандарт разрабатывался в 1997-м году, когда большинство пользователей Интернета работало с сетью через медленные модемы.

В настоящее время большинство работает в Интернете через более скоростные подключения. У меня лично в данный момент два варианта выхода в сеть:

- кабельный Интернет. Коаксиальный кабель подключен к Wi-Fi-роутеру, который может работать на стандартах до 802.11n (150 Мбит/с), а провайдер обещает, что его кабель может обеспечить передачу до 10 Мбит/с в мою сторону и до 512 Кбит/с от меня. Так как скорость определяется самым слабым звеном, то больше 10 Мбит/с на скачивание не будет;
- 3G-модем с Wi-Fi-маршрутизатором в одной коробке (я описывал его здесь **http://www.funniestworld.com/Review.aspx?id=781**) и по заявлениям провайдера он должен работать на скорости до 5,76 Мбит/с.

Даже при самых слабых показателях скорость передачи достаточно высокая, чтобы без проблем качать данные даже в 10 потоков. Именно поэтому Internet Explorer 8 сделали чуть более интеллектуально развитым, и он уже определяет максимальное количество потоков в зависимости от соединения. Если это коммутируемое подключение, то количество подключений определяется в зависимости от протокола — 4 или 2 (не забываем, что это значения по умолчанию в реестре и их можно

изменить). Ну а если это высокоскоростное подключение типа интернет-кабеля или Wi-Fi, то браузер будет использовать значение 6.

Чем больше доступно подключений, тем лучше для таких современных технологий как AJAX, но хуже для сервера. Дело в том, что количество подключений на стороне сервера не безгранично. Все имеет свои пределы, просто они иногда очень большие. Но благодаря современным методам кэширования, использования proxy-серверов и современных веб-серверов этим фактором начинают пренебрегать. Особенно клиенты. Мне как пользователю все равно, какие лимиты на сервере, я хочу получать информацию быстрее.

Обратите внимание, что ключ реестра, который мы сейчас рассматриваем, находится в **HKEY_LOCAL_MACHINE**, и это значит, что его изменение отразится на всех пользователях компьютера. Если вы хотите установить для каждого пользователя отдельные настройки, то можно без проблем создать описанные ранее параметры в ветке **SOFTWARE\Microsoft\InternetExplorer\MAIN\FeatureControl**, но только в **HKEY_CURRENT_USER** для текущего пользователя.

Увеличить количество одновременных подключений можно и с помощью групповых политик. Но это отдельная история, да и управление политиками может отсутствовать на компьютере. Просто они ставятся далеко не со всеми редакциями. Поэтому редактирование параметров реестра напрямую намного эффективнее.

На моей практике 6 является вполне достойным значением. Если сервер работает в нормальном режиме, то он будет обрабатывать и более 6 подключений сам. Ну а если сервер слабый, то изменение параметров не поможет.

Сейчас мы отойдем немного от темы и рассмотрим эту проблему с другой стороны — со стороны веб-сервера. Хотя программирование выходит за рамки книги, информация может быть очень полезной.

Итак, мы уже знаем, что Internet Explorer по умолчанию скачивает максимум 6 файлов одновременно. Если на веб-странице 30 и более картинок, то скачивание может быть достаточно долгим, особенно в браузерах IE до версии 8, если качать максимум две картинки. Если тратить на загрузку каждой пары картинок хотя бы 1 секунду, загрузка 30 будет происходить 15 секунд. Это очень много, потому что пользователи не любят ждать столько времени.

Страница должна появляться максимум через 5 секунд, иначе сайт теряет клиентов. Пять секунд — это максимум, где-то я читал, что пользователи не любят ждать и две секунды, а уходят к конкуренту. Лично я сам тоже не люблю медленные сайты.

Разработчики веб-сайтов используют различные ухищрения для того, чтобы оптимизировать загрузку своих сайтов. Можно создать одну большую картинку, на которой будут находиться все необходимые ресурсы в виде маленьких картинок, а потом с помощью CSS отображать только отдельные части большого холста. Но одна большая картинка неудобна с точки зрения сопровождения. 30 маленьких все же удобнее.

Чтобы браузер смог грузить сразу 20 картинок одновременно, проще использовать следующий трюк: разместить картинки на разных доменах. Каждый хост определя-

ется не адресом (IP), а доменным именем сайта. Например, для веб-страницы **www.flenov.info/blog.php** браузер сможет открыть *X* подключений к домену **www.flenov.info** (где *X* зависит от настроек системы и браузера). Но если половину картинок поместить на **image.flenov.info/images/**, то браузер уже будет думать, что перед ним совершенно другой сервер, и сможет открыть еще *X* соединений к домену **image.flenov.info**, а это значит, что можно качать в два раза больше ресурсов.

Такую оптимизацию очень легко реализовать с помощью DNS. Достаточно только настроить его так, чтобы любые поддомены ***.flenov.info** загружали один и тот же сайт. Что бы вы ни набрали вместо звездочки, на моем сайте все это будет загружать один и тот же мой блог. Я подумываю о том, чтобы использовать поддомены, но на данный момент все работает именно так.

Попробуйте загрузить сайт **www.sonyrewards.com** в FireFox и откройте надстройку FireBug. Эта надстройка позволяет показывать, какие сайты и как загружаются (рис. 1.2). Обратите внимание, что картинки грузятся не с **sonyrewards.com**, а с доменов **image1.sonyrewards.com**, **image2.sonyrewards.com** или **image3.sonyrewards.com**. Ресурсов у сайта много, но за счет того, что практически все браузер может грузить одновременно, сайт грузится достаточно быстро даже у тех, кто использует параметры по умолчанию.

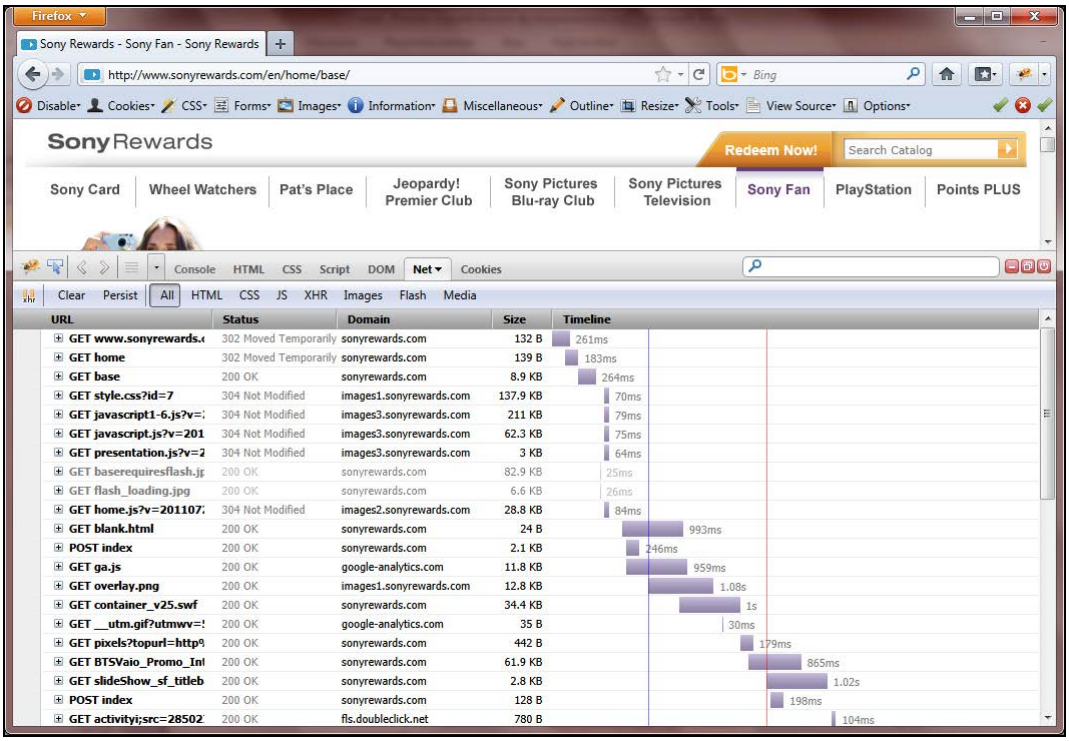

Рис. 1.2. Оптимизация загрузки на стороне сервера

1.2. Windows 7

Не знаю почему, но Microsoft постоянно изменяет методы, которыми она использует различные элементы оформления в своих ОС. Руководству компании наверно нравится менять все каждые 5—7 лет. Но вот в последней версии Windows 7 графическая оболочка вроде бы достигла своего идеала и почти не отличается от Windows Vista.

По сравнению с Windows 8 графическая система снова претерпит изменения, но уже похоже не так сильно. То, что касается оформления Windows XP, и было описано в предыдущих изданиях книги, вы можете найти в электронном архиве на FTP-сервере в каталоге Doc\Windows.

1.2.1. Окно входа в систему

Не знаю почему, но Microsoft зачем-то усложнила смену фонового рисунка загрузки компьютера и окна входа в систему. В Windows 9*x* окно менялось банальным изменением графического файла в корне загрузочного диска, хотя и расширение файла было изменено (если мне не изменяет память, то расширение было dat вместо bmp). В Windows XP спрятали в файл ресурсов, и изменение картинок достаточно сильно усложнилось.

В Windows 7 компания Microsoft (или кто там отвечает за подобные вещи) снова упростила процесс смены картинки, хотя он все равно остался скрытым. Не знаю, почему компания не предоставила нормальной утилиты.

Итак, чтобы изменить фон картинки, которую вы видите при входе в систему, нужно для начала загрузить редактор реестра и перейти в следующий раздел:

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\Background

Тут должен быть параметр **OEMBackground**, который по умолчанию равен `0`. Если какой-то из параметров или разделов не существует, то их можно без проблем создать. Если параметр **OEMBackground** изменить на `1`, то в качестве картинки для окна входа в систему будет использоваться файл:

C:\Windows\System32\oobe\Info\backgrounds\backgroundDefault.jpg

Тут есть одна интересная особенность — этот путь может быть недоступен в некоторых файловых менеджерах. Например, я люблю использовать Total Commander, и если в этом редакторе перейти в папку c:\Windows\System32\oobe\, то в ней будут только два файла и одна папка, но папки info не будет (рис. 1.3).

Ну а если ту же папку открыть в Проводнике Windows, то она заметно преображается и появляется нужная папка info, в которой и следует искать подпапку backgrounds (рис. 1.4).

Имя файла backgroundDefault.jpg используется по умолчанию, если файл для вашего разрешения не найден. Так как вы знаете разрешение своего экрана, то можете поместить туда соответствующую картинку. Но для более универсального решения

можно создать файлы в следующем формате background*XXX*x*XXX*.jpg, где *XXX*x*XXX* — это разрешение экрана. Например, если у вас экран размером в 1280×960 пикселов, то имя файла должно быть background1280x960.jpg.

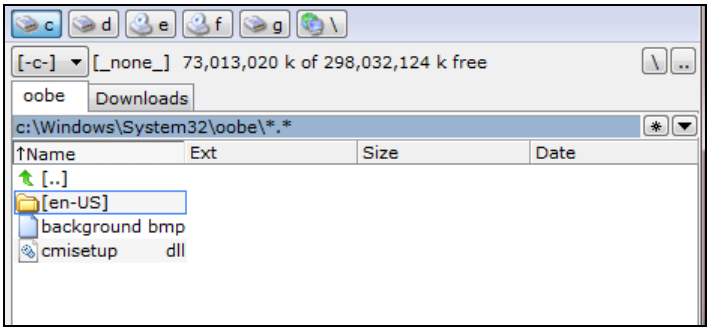

Рис. 1.3. Папка oobe в Total Commander

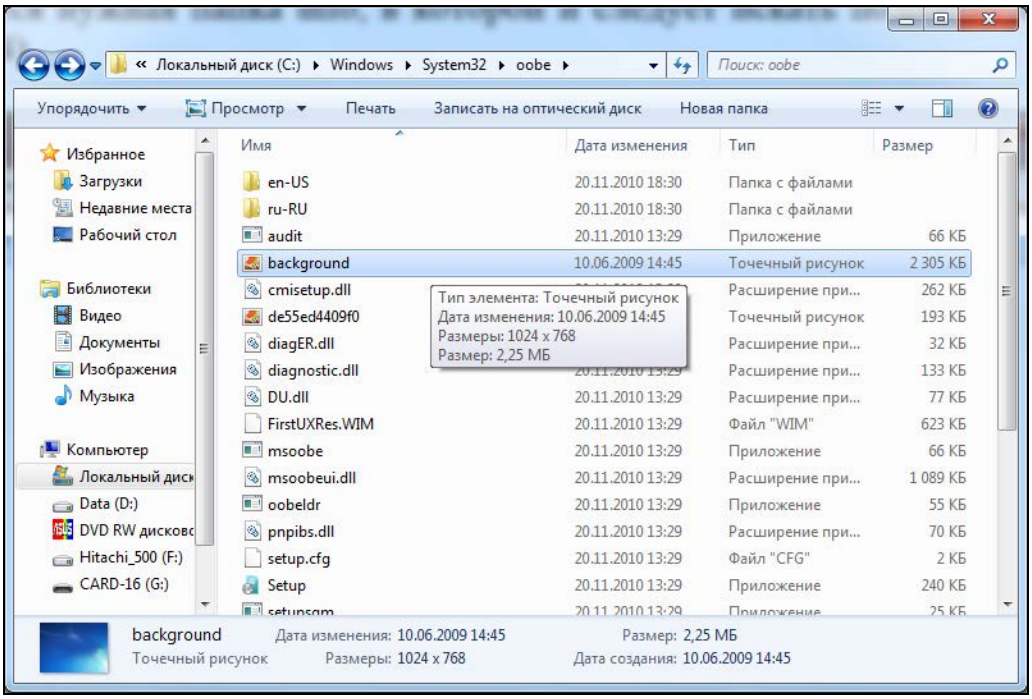

Рис. 1.4. Папка oobe в Проводнике Windows

1.2.2. Рабочий стол

Лично я люблю чистый рабочий стол, на котором нет ничего лишнего. На данный момент у меня на нем расположены только один ярлык Корзины (Recycle Bin) и один ярлык документа, с которым я работаю.

Если посмотреть на рабочий стол компьютера моей жены, то там файлу приземлиться негде. Весь рабочий стол заполнен ярлыками.

Есть несколько способов убрать все с рабочего стола. Самый жестокий — это удалить ярлыки. Хотя нет, все ярлыки удалить не получится. Есть один, который не удаляется — Корзина. Если щелкнуть по ней правой кнопкой мыши, то там вы не найдете пункта удаления. Чтобы удалить Корзину, переходим в реестре к разделу:

HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\Desktop\NameSpace

Здесь ищем подраздел, у которого значение по умолчанию равно `Recycle Bin` (рис. 1.5). У меня таким оказался **{645FF040-5081-101B-9F08-00AA002F954E}**. Я тут не могу утверждать, но опыты показывают, что на всех системах Windows Vista и Windows 7 Корзина имеет этот код. Но просто на всякий случай убедитесь, что именно она перед вами.

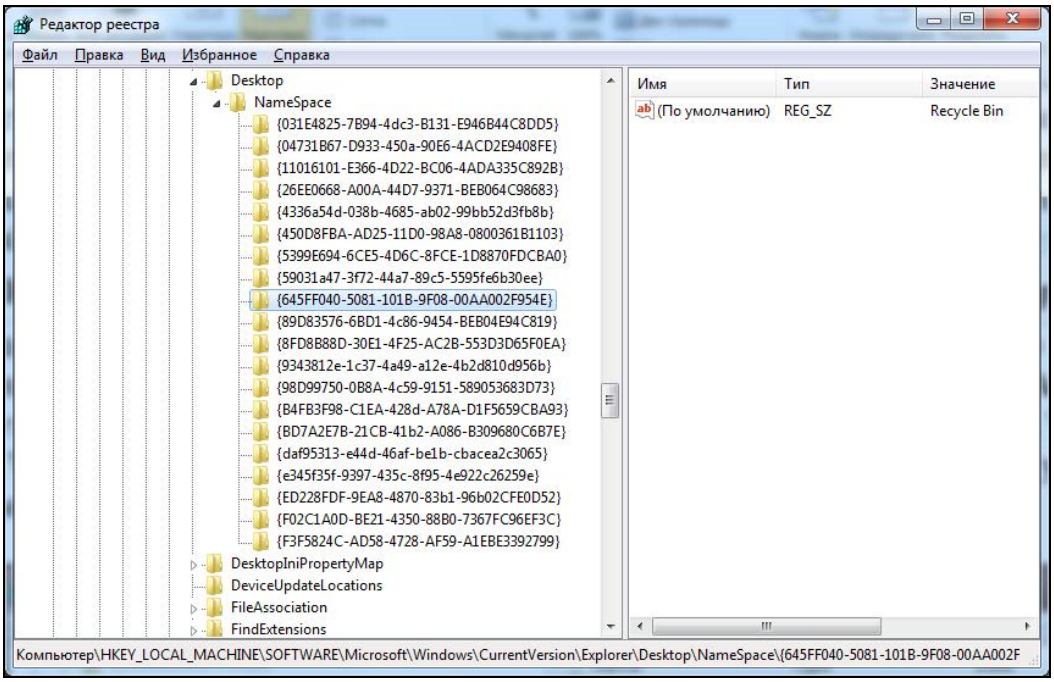

Рис. 1.5. Раздел реестра, отвечающий за Корзину на рабочем столе

Так как мы удалили Корзину, ее нужно как-то чистить. Ее все еще можно очистить, например, вручную. Для этого в своем менеджере файлов сделайте так, чтобы отображались скрытые файлы. На каждом диске вы сможете увидеть скрытую папку с именем $Recycle.Bin (рис. 1.6). Внутри этой папки будет еще несколько папок, но для доступа к ним нужны права администратора, и там ничего интересного нет. Вас больше должна интересовать папка, у которой даже ярлык — корзина. У меня это S-1-5-21-4060577442-2030883239-2705281912-1000. Если зайти в эту папку, то вы увидите удаленные вами файлы, и очистка Корзины в принципе заключается в удалении этих файлов с диска.

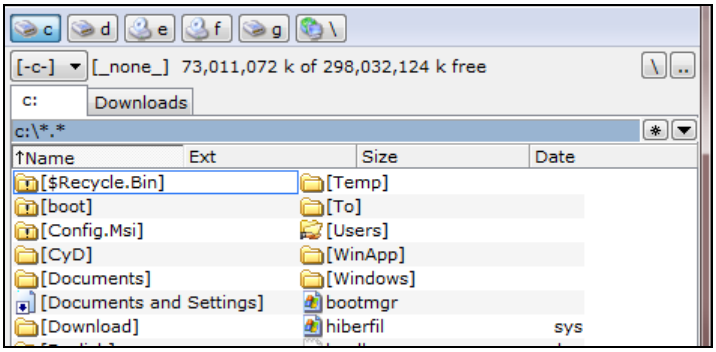

Рис. 1.6. Корзина — это просто скрытая папка в корне диска

Получается, что для доступа к Корзине достаточно в файловом менеджере перейти в папку c:\$Recycle.Bin\S-1-5-21-4060577442-2030883239-2705281912-1000\. Но это не всегда удобно, и у каждого диска своя папка для Корзины. Это сделано для того, чтобы проще было удалять. Дело в том, что если перемещать файл из реальной папки в папку Корзины внутри одного диска, достаточно пользоваться операцией переименования. Вы как бы переименовываете путь к файлу, перемещая его в новое место, и это происходит мгновенно. Если же источник и приемник на разных дисках, то тут уже придется копировать файл с одного места в другое и потом удалять из источника.

Но Корзине можно найти вполне удобное и полезное место для жизни — окно **Компьютер** (Computer). Тут располагаются все диски и сюда же можно добавить Корзину. Для этого переходим в раздел реестра:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\MyComputer\NameSpace

И добавляем уже знакомый нам магический ключик **{645FF040-5081-101B-9F08-00AA002F954E}**. Теперь ваше окно **Компьютер** будет содержать и Корзину — рис. 1.7.

Но если вы хотите убрать значки просто ради шутки (а шутки мы будем рассматривать отдельно), то не обязательно прибегать к жестокому методу удаления этих самых значков. Я не настолько жестокий, поэтому предлагаю воспользоваться более простым способом. В реестре переходим в следующий раздел:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer

И здесь создаем `DWORD`-параметр **NoDesktop** (он, скорее всего, не будет существовать). Если этому параметру установить значение `1`, то все ярлыки исчезнут с рабочего стола. Чтобы вернуть все значки на их место, просто изменяем на `0`.

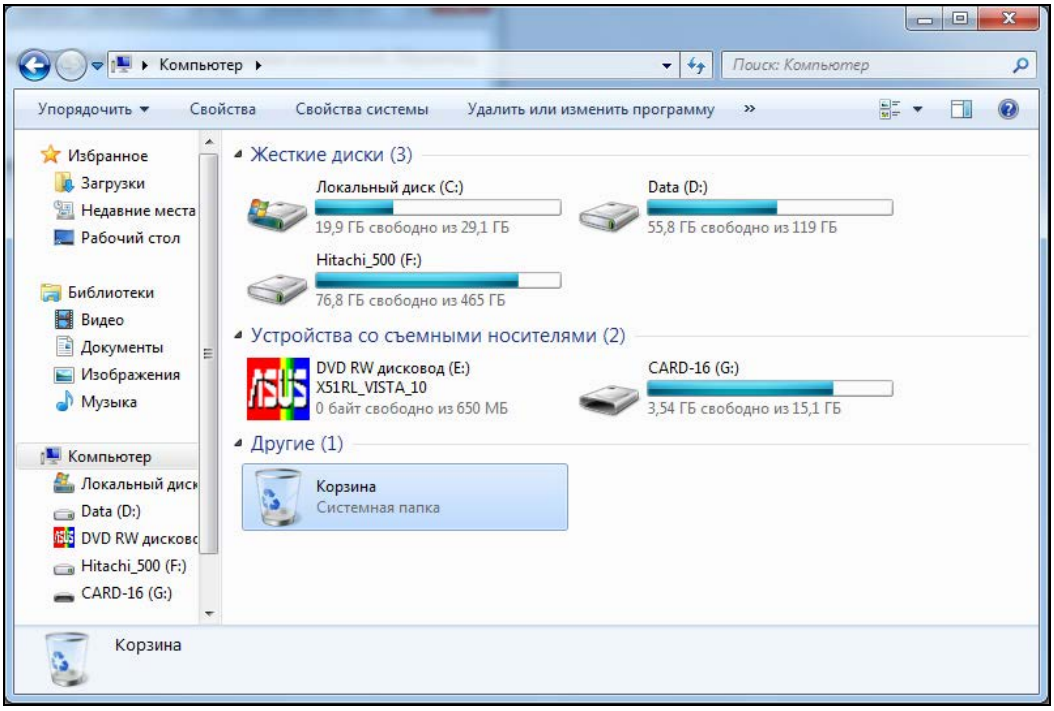

Рис. 1.7. Корзина в вашем компьютере

ГЛАВА 2

Внутренний мир Windows

Если в предыдущей главе мы обсуждали Windows весьма поверхностно, то здесь мы рассмотрим проблемы настройки глубже и детальнее. Вы узнаете, из чего состоят программы, и это позволит вам изменять практически любой софт по собственному усмотрению.

В этой главе нам предстоит познакомиться с великолепной программой Restorator, с помощью которой вы сможете редактировать ресурсы исполняемых файлов и динамических библиотек. В качестве практических примеров мы отредактируем загрузчики Windows XP и программы входа в систему.

Изменение ресурсов, которое мы будем рассматривать в данной главе, применимо в равной степени к любой версии ОС. Но я затрону только загрузчики Windows XP/Vista/7, которые имеют новый формат и содержат намного больше интересных для хакера настроек. Однотипные же ресурсы в Windows 9*x* реализованы проще, и о них уже много сказано в Интернете, так что нет смысла повторяться. (Да и есть ли у кого еще эта ОС?)

2.1. Ресурсы Windows

Прежде чем приступать к серьезным изменениям системы, мы должны немного познакомиться с теорией. Основой этого раздела будет работа с ресурсами программ, и именно о них мы сейчас поговорим с научной точки зрения.

Что такое ресурсы и для чего они нужны? Чтобы понять это, достаточно увидеть, что может быть в ресурсах, а это картинки, значки, строки и внешний вид диалоговых окон. Программа использует ресурсы в своей работе, а мы можем получить к ним доступ и изменить, а значит, повлиять на внешний вид и даже на поведение программы.

Классические исполняемые файлы Windows имеют расширение exe. В общем виде они состоят из следующих частей:

- заголовок;
- исполняемый код;
- ресурсы.

Существуют еще и .NET-сборки, но это уже отдельная история.

Заголовок содержит служебную информацию, которую ОС использует при запуске файла. Например, здесь записана точка, начиная с которой должен выполняться исполняемый код. Это очень важная информация для любой программы. Помимо этого, можно узнать, где размещаются ресурсы программы (чаще всего — после исполняемого кода, но возможны и исключения).

Исполняемый код мы изменять не будем, это достаточно сложно и нужны знания Ассемблера и сложных программ отладки приложений. Ну а с ресурсами познакомимся достаточно подробно, потому что здесь для настоящего хакера кроется много интересного.

Все ресурсы разбиты по разделам:

- Bitmap — картинки, высвечиваются в окнах программы;
- Menu — меню, обеспечивают удобный доступ к функциям приложения, структурируя их в однородные группы;
- Dialog — всевозможные окна диалогов;
- Stringtable — таблица с сообщениями, которые используются в строках состояния или в окнах диалогов;
- Accelerator — сочетания клавиш для быстрого вызова каких-либо команд;
- Cursor — различные курсоры;
- Icon — рисунки определенного размера, чаще используются для отображения в виде значка формы в свернутом состоянии;
- Versioninfo — информация о версии. В дальнейшем мы этот раздел использовать не будем, поэтому забудьте о его существовании и то, что я о нем упоминал :).

Все ресурсы хранятся в открытом виде и доступны для редактирования. Ресурсы могут быть не только в исполняемых файлах, но и в динамических библиотеках (dll), программах-заставках (scr), отдельных файлах ресурсов (res) и в некоторых других типах файлов.

Руками какой-либо из ресурсов изменить невозможно, но программ для их редактирования великое множество. Практически в каждом языке программирования есть утилита или встроенный модуль, который позволяет изменять ресурсы:

- Borland Resource Workshop — поставляется с некоторыми средствами разработки фирмы Borland;
- Microsoft Visual Studio — среда разработки от Microsoft, которая может открывать исполняемые файлы для редактирования ресурсов.

Тут надо заметить, что модули, написанные на разных языках программирования, могут иметь разные типы ресурсов. Например, компилятор Visual C++ создает про-