Крис Касперски,
Ева Рокко

bhv®

ИСКУССТВО ДИЗАССЕМБЛИРОВАНИЯ

- Обзор хакерского инструментария для Windows и Linux
- Техника работы с отладчиками и дизассемблерами
- Практическое дизассемблирование и идентификация ключевых структур языков высокого уровня
- Защитные механизмы и методы их обхода
- Антиотладочные приемы и борьба с ними
- Обфускация кода и ее преодоление

Наиболее полное руководство

+cd

В ПОДЛИННИКЕ®

Крис Касперски
Ева Рокко

ИСКУССТВО ДИЗАССЕМБЛИРОВАНИЯ

Санкт-Петербург
«БХВ-Петербург»
2008

УДК 681.3.06
ББК 32.973.26-018.1
К28

Касперски, К.

К28 Искусство дизассемблирования / К. Касперски, Е. Рокко. — СПб.: БХВ-Петербург, 2008. — 896 с.: ил. + CD-ROM — (В подлиннике)

ISBN 978-5-9775-0082-1

Книга посвящена вопросам и методам дизассемблирования, знание которых позволит эффективно защитить свои программы и создать более оптимизированные программные коды. Объяснены способы идентификации конструкций языков высокого уровня таких, как C/C++ и Pascal, показаны различные подходы к реконструкции алгоритмов. Приводится обзор популярных хакерских инструментов для Windows, UNIX и Linux — отладчиков, дизассемблеров, шестнадцатеричных редакторов, API- и RPC-шпионов, эмуляторов. Рассматривается исследование дампов памяти, защитных механизмов, вредоносного программного кода — вирусов и эксплоитов. Уделено внимание противодействию антиотладочным приемам. К книге прилагается компакт-диск с полноцветными иллюстрациями и кодами рассматриваемых примеров.

Для программистов и продвинутых пользователей

УДК 681.3.06
ББК 32.973.26-018.1

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Наталья Таркова*
Зав. редакцией	*Григорий Добин*
Редактор	*Ольга Кокорева*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.10.07.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 72,24.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-9775-0082-1

Оглавление

Введение

Книга, которую вы сейчас держите в руках, открывает двери в удивительный мир реверсинга (обратной разработки) защитных механизмов. Она адресована всем, кто любит головоломки и готов сделать свои первые шаги на пути к тому, чтобы стать настоящим хакером. Всем, кто проводит свободное (и несвободное) время за копанием в недрах программ и операционной системы. Наконец, всем, кто по роду своей деятельности занимается (постоянно или эпизодически) написанием защит и хочет узнать, как грамотно и гарантированно противостоять вездесущим хакерам.

Книга посвящается обратной разработке — пожалуй, наиболее сложному из аспектов хакерства, ведь дизассемблирование — это искусство. Однако авторы приложили максимум усилий к структурированию материала таким образом, чтобы изложение было выстроено логично, а читатель, овладевая искусством дизассемблирования, двигался "от простого к сложному". В начале книги излагаются базовые основы хакерства — техника работы с отладчиками, дизассемблерами, шестнадцатеричными редакторами, API- и RPC-шпионами, эмуляторами. Приводится широкий обзор популярного хакерского инструментария для Windows, UNIX и Linux, а также демонстрируются практические приемы работы с популярными отладчиками (SoftICE, OllyDbg, WinDbg), от широко известных до нетрадиционных. Подробно описаны приемы идентификации и реконструкции ключевых структур исходного языка — функций (в том числе виртуальных), локальных и глобальных переменных, ветвлений, циклов, объектов и их иерархий, математических операторов и т. д. Наряду с этим демонстрируются различные подходы к анализу алгоритма изучаемых программ и объясняется, как не заблудиться в мегабайтах дизассемблированного кода и избежать разнообразных хитрых ловушек. Значительное внимание уделено таким важным темам, как реконструкция алгоритмов работы защитных механизмов, идентификация ключевых структур языков высокого уровня, таких, как C/C++ и Pascal. Рассматриваются практические методы преодоления антиотладочных приемов, техника снятия дампа с защищенных приложений, преодоление упаковщиков и протекторов. На практических примерах продемонстрированы методы анализа кода вредоносных программ и эксплоитов. Не оставлены без внимания и такие важные темы, как противодействие антиотладочным приемам, исследование упакованного, зашифрованного и умышленно запутанного (Obfuscated) кода, а также другим технологиям, затрудняющим дизассемблирование и отравляющим хакерам жизнь.

ЧАСТЬ I

ОБЗОР

ХАКЕРСКИХ ПРОГРАММ

Глава 1

Инструментарий хакера

С чего начинается хакерство? Некоторые скажут — с изучения C/C++, языка ассемблера, обучения искусству отладки и дизассемблирования… И будут правы. Другие же добавят к этому "джентльменскому списку" изучение архитектуры разнообразных операционных систем, сетевых протоколов, поиск уязвимостей. И тоже будут правы. Но, с другой стороны — а за счет чего хакеры добиваются результатов? Правильно, за счет знаний и упорного труда. Но при этом они не только работают руками и головой, но и пользуются хакерским инструментарием. И правильный подбор программ очень важен, поскольку именно они формируют сознание, позволяя начинающему сделать свои первые шаги в дремучем лесу машинных кодов. Вот только этих программ настолько много, что новичок, впервые попавший на хакерский сайт, оказывается в полной растерянности — что качать, а что не стоит внимания. Основная цель данной главы как раз и состоит в том, чтобы дать предельно сжатый обзор хакерского софта, покрывающего практически любые потребности.

> ***Примечание***
>
> Помимо упорства и стремления к самостоятельному поиску ответов на возникающие вопросы, девизом любого хакера должна стать фраза: "Знай и люби свои инструменты". Иными словами, вы не должны довольствоваться лишь приведенным здесь кратким обзором, который призван лишь обратить ваше внимание на ту или иную программу. Как правило, в комплекте с любой из этих программ поставляется и руководство пользователя, и руководства эти настоятельно рекомендуется внимательно изучать.

Отладчики

Лучший отладчик всех времен и народов — это, конечно же, SoftICE, на котором выросло не одно поколение хакеров. Это — интерактивная программа с развитым командным интерфейсом, представляющим собой компромисс между легкостью освоения и удобством использования (рис. 1.1). Иными словами, без чтения руководства здесь не обойтись, тем более что никаких интуитивно-понятных меню SoftICE не предоставляет[1].

Изначально созданный фирмой NuMega, SoftICE был продан компании Compuware, долгое время распространявшей его в составе громоздкого пакета DriverStudio Framework. К глубокому разочарованию, 3 апреля 2006 по малопонятным причинам компания объявила о прекращении работы над продуктом, похоронив тем самым уникальнейший проект. Последняя версия DriverStudio 3.2 поддерживает всю линейку Windows вплоть до Server 2003, а также архитектуру

[1] Подробное "Руководство пользователя SoftICE" в русском переводе находится здесь: **http://www.podgoretsky.com/ftp/Docs/softice/siug.pdf**. Кроме того, достаточно краткое, но весьма полезное руководство по использованию SoftICE, позволяющее быстро начать работу с данным отладчиком, можно найти по адресу **http://www.reconstructer.org/papers/The%20big%20SoftICE%20howto.pdf**.

AMD x86-64. То есть лет на пять запаса прочности у SoftICE еще должно хватить, а там хакеры что-нибудь придумают [2].

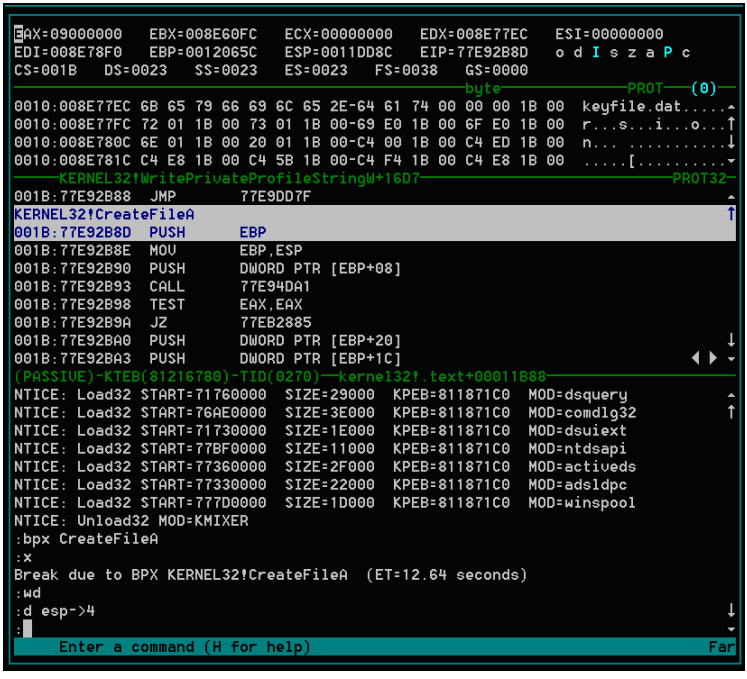

Рис. 1.1. SoftICE — профессионально ориентированный отладчик, на котором выросло не одно поколение хакеров

Найти SoftICE можно практически на любом хакерском сайте (например, здесь: **http://www.woodmann.com/crackz/Tools.htm**). Чтобы не качать целиком весь пакет DriverStudio, можно воспользоваться пакетом DeMoNiX (**http://reversing.kulichki.net**), содержащим только SoftICE, выдернутый из DriverStudio v. 2.7 build 562. Этот пакет занимает всего 2,27 Мбайт. Однако инсталлятор содержит ошибки, а старая версия не поддерживает новых веяний Microsoft (хотя замечательно идет под Windows 2000).

Вместе с SoftICE желательно сразу же установить IceExt (**http://sourceforge.net/projects/iceext**) — неофициальное расширение, позволяющее скрывать присутствие отладчика от большинства защит, сохранять дампы памяти, задействовать кириллические кодировки 866/1251, приостанавливать потоки и выполнять множество других важных задач (рис. 1.2).

Если IceExt откажется запускаться, скорректируйте следующие ключи в данной ветви системного реестра: `HKLM\SYSTEM\CurrentControlSet\Services\NTice: KDHeapSize (DWORD): 0x8000`; `KDStackSize (DWORD): 0x8000`.

Другое неофициальное расширение для SoftICE, IceDump (**http://programmerstools.org/system/files?file=icedump6.026.zip**), также умеет делать много полезных вещей и удачно дополняет IceExt (рис. 1.3).

[2] На данный момент на роль адекватного преемника SoftICE претендует коммерческий отладчик Syser Debugger, о котором речь пойдет чуть далее. Если же вас интересует отладчик Open Source, то возможно, ваше внимание привлечет проект Rasta Ring 0 Debugger (**http://rr0d.droids-corp.org/**).

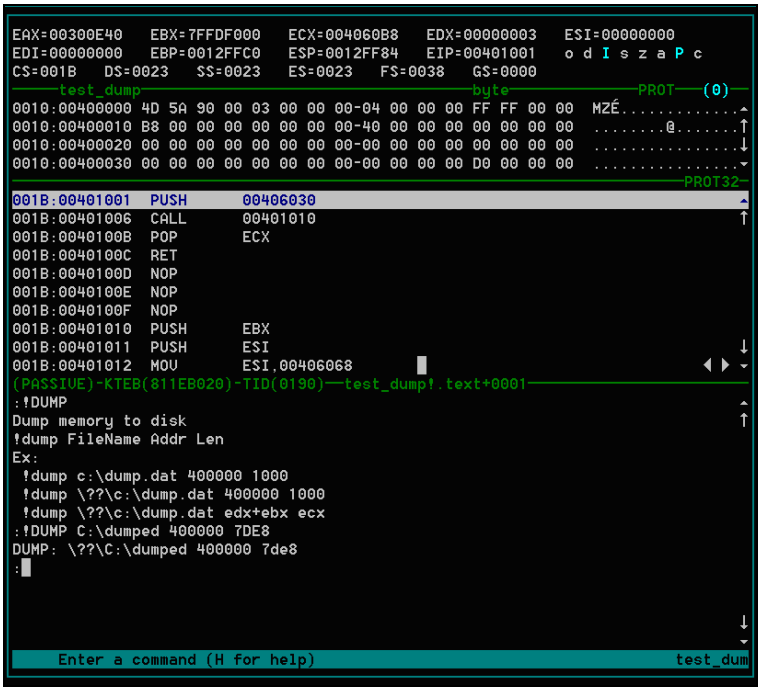

Рис. 1.2. Снятие дампа с помощью IceExt

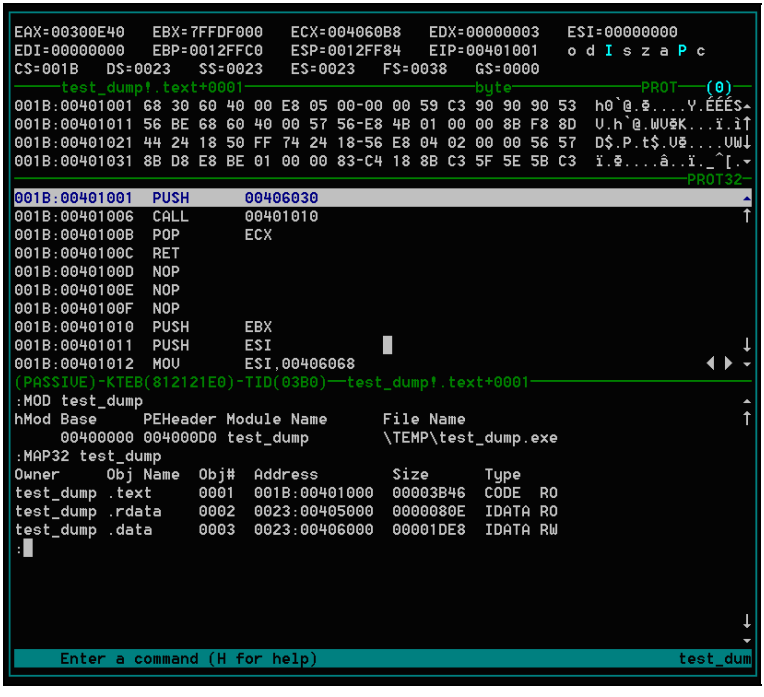

Рис. 1.3. Снятие дампа с помощью IceDump

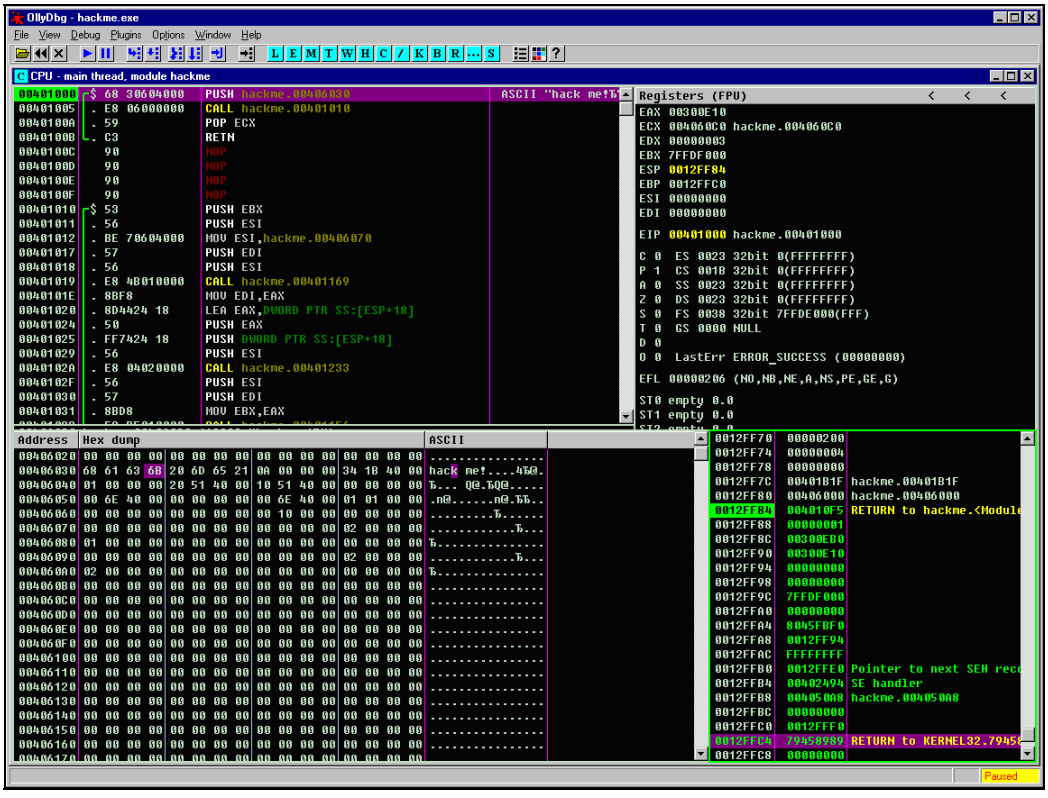

Рис. 1.4. Компактный и шустрый отладчик OllyDbg

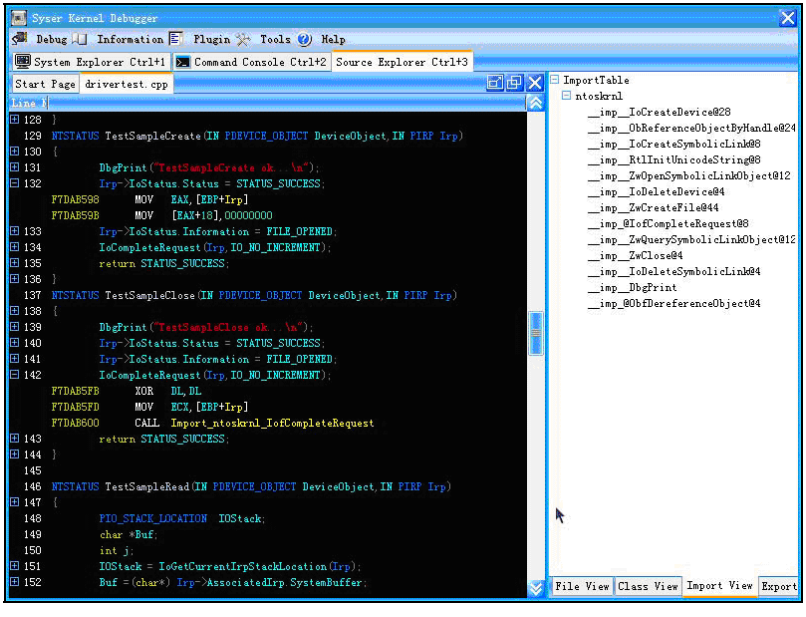

Рис. 1.5. Отладчик Syser Debugger за отладкой термоядерного драйвера

Кстати, сам SoftICE замечательно работает под виртуальной машиной VMware, для этого достаточно добавить в конфигурационный файл с расширением .vmx следующие две строки: `paevm = TRUE` и `processor1.use = FALSE`.

> ***Примечание***
>
> При работе с SoftICE на многоядерных процессорах и процессорах с поддержкой HyperThreading также были отмечены некоторые проблемы (хотя возникают они нерегулярно). Устранить эти проблемы в случае их возникновения можно, добавив ключ `/ONECPU` в файл boot.ini.

Помимо SoftICE существуют и другие отладчики, из которых в первую очередь хотелось бы отметить бесплатный Olly Debugger (**http://www.ollydbg.de**). Это — удобный инструмент прикладного уровня (рис. 1.4), ориентированный на хакерские потребности, поддерживающий механизм плагинов (plug-ins) и собравший вокруг себя целое сообщество, написавшее множество замечательных расширений и дополнений, прячущих OllyDbg от защит, автоматически определяющих оригинальную точку входа в упакованной программе, облегчающих снятие протекторов и т. д.

Неплохие коллекции плагинов можно найти на сайтах **http://www.wasm.ru** и **http://www.openrce.org**.

Самый свежий (и пока еще во многом экспериментальный) ядерный отладчик — это, бесспорно, Syser (**http://www.sysersoft.com**). Данный отладчик выпущен китайскими разработчиками и в настоящее время переживает стадию активного развития и становления (рис. 1.5). Как уже говорилось чуть ранее, этот отладчик претендует на роль преемника SoftICE. Его последняя версия (v. 1.9, датированная 17 мая 2007 года) работает с 32-разрядными версиями Windows 2000/XP/2003/Vista, а также обеспечивает поддержку SMP, HyperThreading и многоядерных процессоров.

Довольно многие программисты пользуются отладчиком уровня ядра Microsoft WinDbg, входящим в состав бесплатного набора Debugging Tools (**http://www.microsoft.com/whdc/devtools/debugging/debugstart.mspx**). Он вполне пригоден для взлома (рис. 1.6). Тем не менее, большинство хакеров, привыкших к черному экрану SoftICE, считают его неудобным.

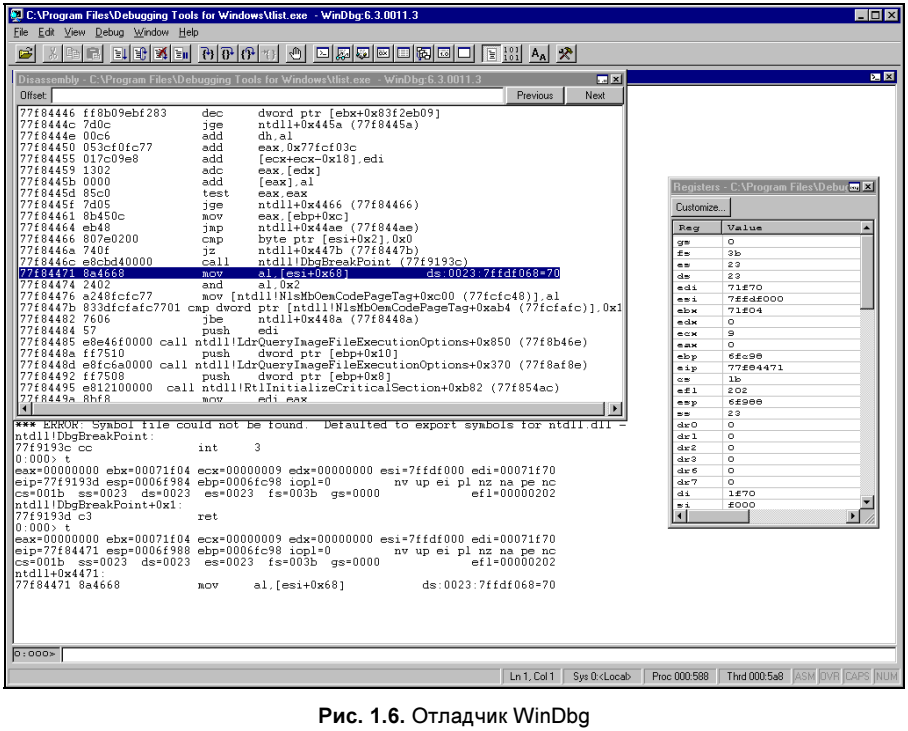

Рис. 1.6. Отладчик WinDbg

Дизассемблеры

Существует всего лишь один дизассемблер, пригодный для по-настоящему профессиональной работы — IDA Pro (**http://www.idapro.com**). Этот дизассемблер имеет консольную (рис. 1.7) и графическую (рис. 1.8) версии. IDA Pro воспринимает огромное количество форматов файлов и множество типов процессоров, легко справляясь с байт-кодом виртуальных машин Java и .NET, поддерживает макросы, плагины и скрипты, содержит интегрированный отладчик, работает под MS-DOS, Windows, Linux (рис. 1.9) и обладает уникальной способностью распознавать имена стандартных библиотечных функций по их сигнатурам.

Существует несколько версий IDA Pro — бесплатная (freeware), стандартная (standard) и расширенная (advanced). Бесплатную версию IDA Pro можно скачать по адресу **http://www.dirfile.com/ida_pro_freeware_version.htm**. Стоит, правда, отметить, что бесплатная версия, по сравнению со стандартной и расширенной, обладает ограниченными возможностями. Из всех процессорных архитектур поддерживается только x86, а функция поддержки подключаемых модулей попросту отсутствует. Что касается стандартной и расширенной версий, то они, как и любое хорошее программное обеспечение, стоят дорого (хотя, если хорошо поискать в файлообменных сетях, их можно стянуть оттуда бесплатно).

Основное достоинство IDA Pro состоит в том, что это — интерактивный дизассемблер, то есть интеллектуальный инструмент, позволяющий работать с двоичным файлом, мыслить и творить, а не тупой автомат, заглатывающий исследуемую программу и выплевывающий "готовый" дизассемблированный листинг, в котором все дизассемблировано неправильно.

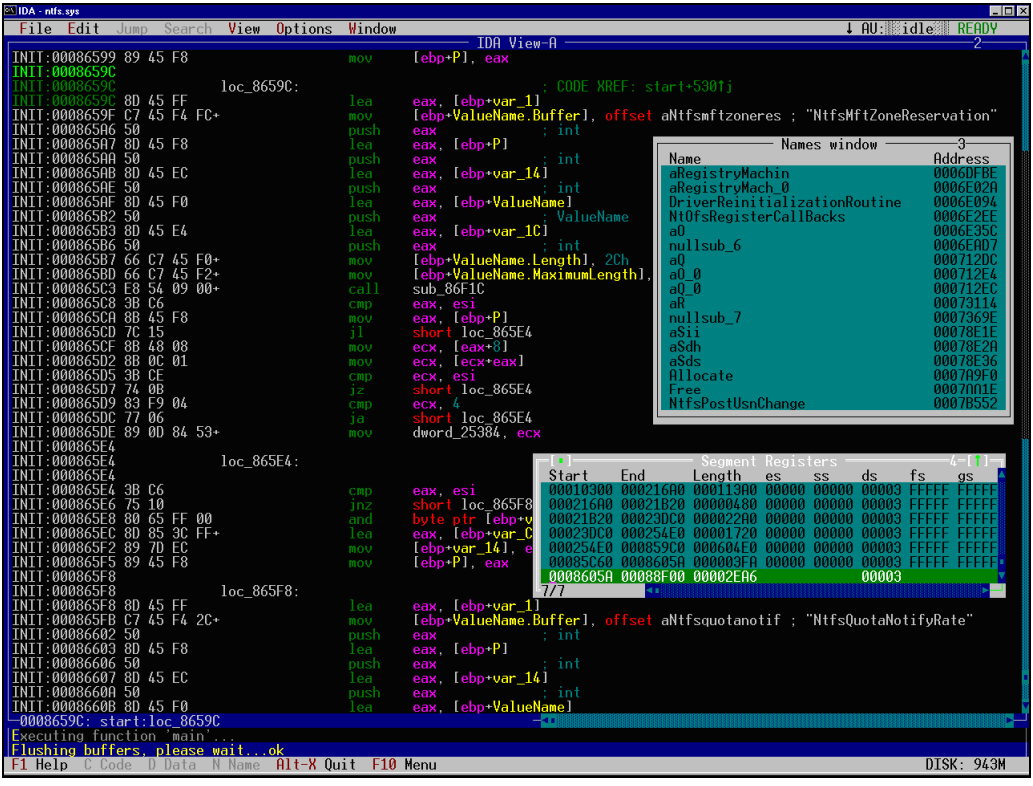

Рис. 1.7. Консольная версия IDA Pro

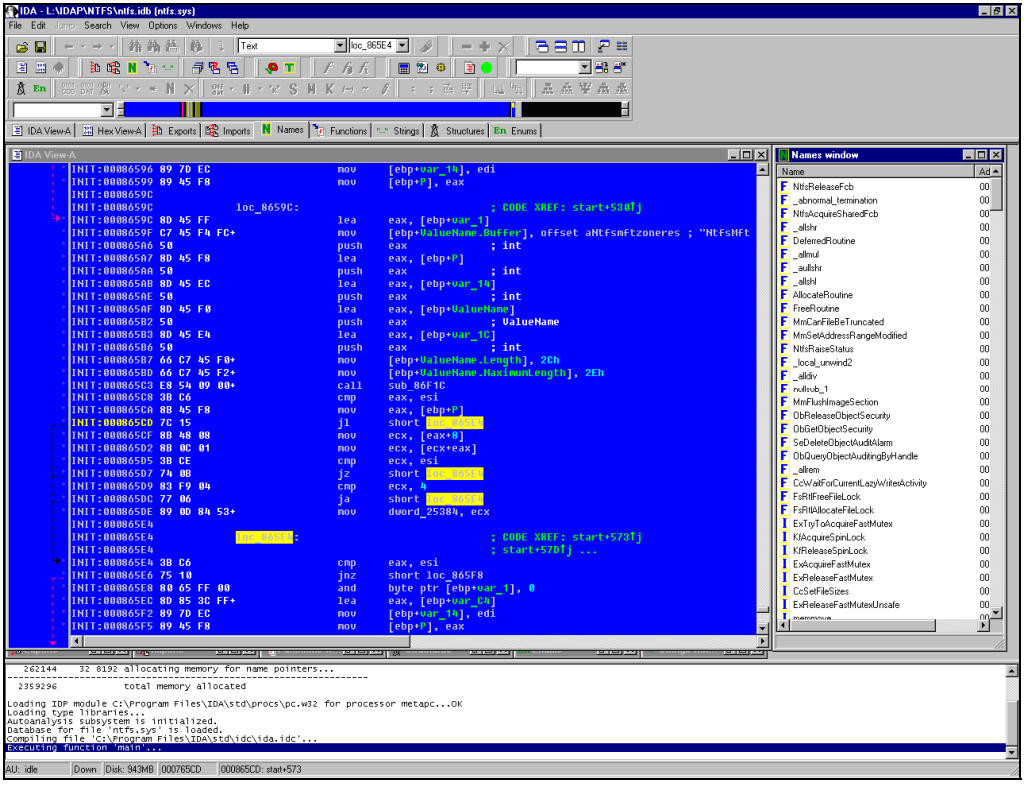

Рис. 1.8. Графическая версия IDA Pro

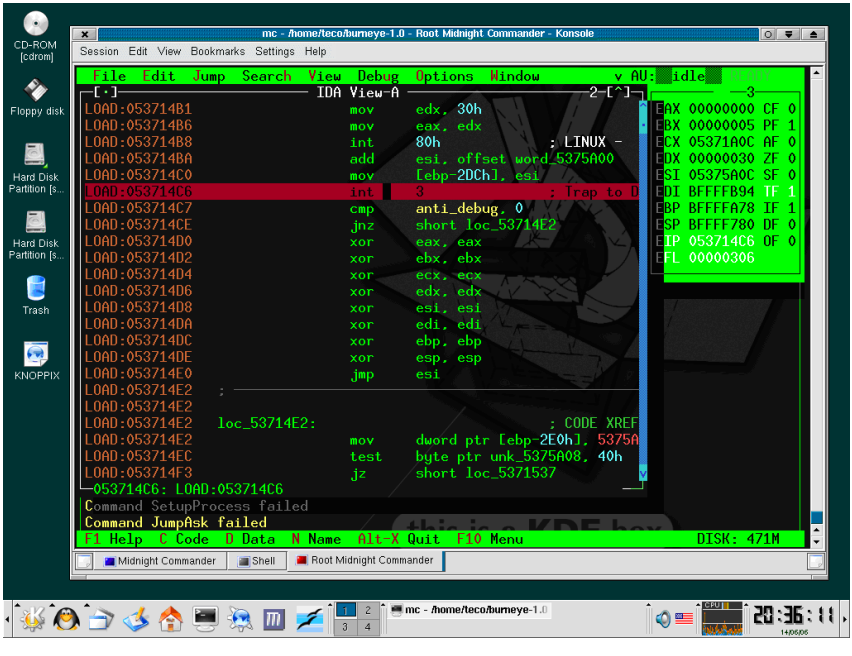

Рис. 1.9. IDA Pro под Linux

В последних версиях IDA Pro сделаны определенные подвижки в сторону автоматической распаковки файлов и снятия обфускаторов[3]. Внушительную коллекцию плагинов и скриптов можно найти как на официальном сайте, так и на сайте **http://www.openrce.org**.

Конкурирующие продукты не выдерживают никакого сравнения с IDA Pro. Тем не менее, народ активно качает бесплатный (ныне заброшенный) дизассемблер с функциями отладчика — W32Dasm (**http://www.wasm.ru/baixado.php?mode=tool&id=178**) и, судя по всему, остается доволен (рис. 1.10).

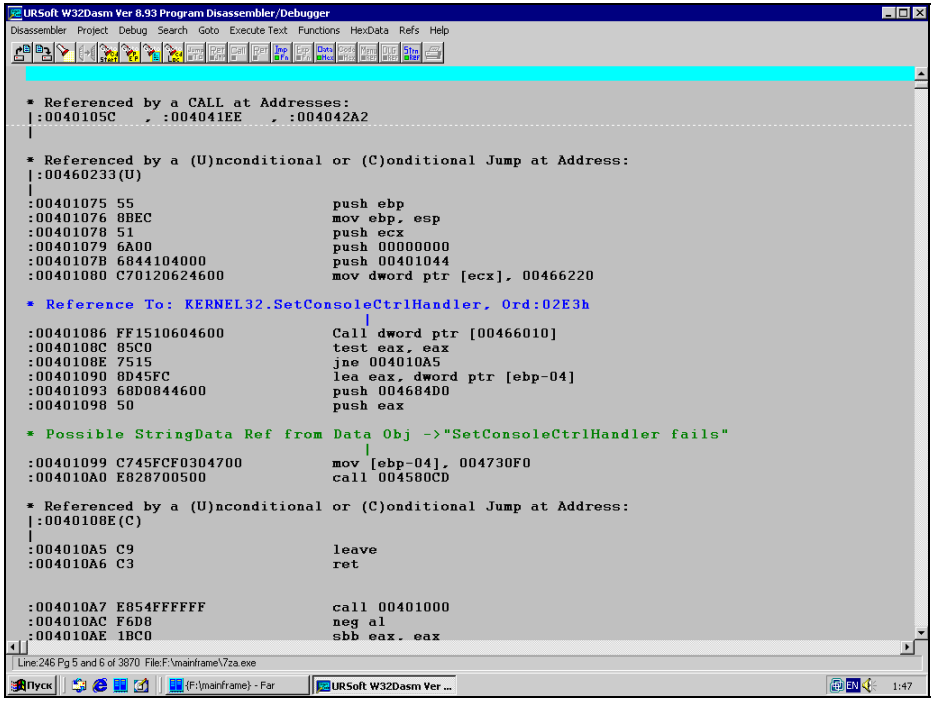

Рис. 1.10. Дизассемблер WDASM

Остальные дизассемблеры обладают еще более ограниченными возможностями, и поэтому здесь не рассматриваются. Единственный из такого рода продуктов, заслуживающий упоминания — это Hacker Disassembler Engine (**http://patkov-site.narod.ru/lib.html**), представляющий собой дизассемблер длин, распространяющийся в исходных текстах и предназначенный для встраивания в различные хакерские программы, занимающиеся перехватом функций, автоматической распаковкой, генерацией полиморфного кода и т. д.

Декомпиляторы

Декомпиляцией называется процесс получения исходного текста программы (или чего-то очень на него похожего) из двоичного файла. В полном объеме декомпиляция принципиально невозможна, поскольку компиляция — однонаправленный процесс, причем с потерей данных. Однако декомпиляторы все-таки существуют и со своей задачей достойно справляются.

[3] Более подробную информацию о распаковке файлов, снятии протекторов и обфускаторов можно найти в *части V* данной книги.

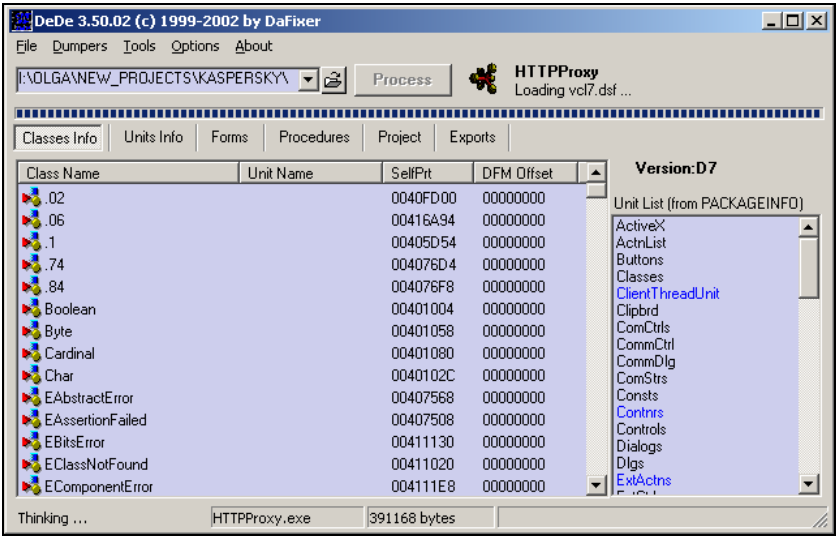

Рис. 1.11. Декомпилятор DeDe ломает HTTPProxy

Для программ, написанных на Delphi и Borland Builder с использованием RTTI, возможно восстановить исходную структуру классов вплоть до имен функций-членов, а также реконструировать формы и "вычислить" адреса обработчиков каждого из элементов. Допустим, у нас имеется диалоговое окно **Registration** с кнопкой **OK**, и мы хотим знать, какая процедура считывает серийный номер и что она с ним делает. Нет ничего проще! Берем бесплатный декомпилятор DeDe (**http://programmerstools.org/node/120**), декомпилируем программу и вперед (рис. 1.11)!

Для Visual Basic существуют свои декомпиляторы, лучшим из которых считается VB Decompiler от GPcH (**http://www.vb-decompiler.org/index.php?p=Products**). Другие бейсик-декомпиляторы, в том числе VB RezQ (**http://www.vbrezq.com/**), VBDE (**http://programmerstools.org/node/129**) и Spices.Decompiler (**http://programmerstools.org/node/635**) также полезно положить в свой хакерский чемоданчик.

Особый интерес представляют декомпиляторы программ-инсталляторов, поскольку многие проверки (на истечение срока работы демо-версии, на серийный номер или ключевой файл) производятся как раз на стадии инсталляции. Самый популярный инсталлятор — это InstallShield. Для него имеется множество удобных декомпиляторов. Вот только некоторые из них:

- InstallShield X Unpacker (**http://programmerstools.org/node/154**)
- Windows InstallShield Decompiler (**http://programmerstools.org/node/118**)
- InstallShield Decompiler (**http://programmerstools.org/node/114**)
- isDcc (**http://programmerstools.org/node/115**)

Что же касается Java и платформы .NET, то с ними замечательно справляется IDA Pro. Если же у вас нет под рукой IDA Pro, можно воспользоваться специализированными декомпиляторами, которые можно найти на сайтах **http://www.cracklab.ru** и **http://www.wasm.ru** вместе с декомпиляторами Fox Pro, Clipper и прочей экзотики.

Шестнадцатеричные редакторы

Давным-давно шестнадцатеричные редакторы (hex-редакторы) представали собой простые программы, умеющие лишь отображать двоичные файлы в шестнадцатеричном виде и править бай-

ты по указанным адресам[4]. Однако со временем они обросли встроенными дизассемблерами, ассемблерами, калькуляторами, функциями поиска регулярных выражений, научились работать с блоками, распознавать различные форматы файлов и даже расшифровывать/зашифровывать фрагменты кода или данных. В общем, превратились в эдакие швейцарские ножички с шестнадцатью лезвиями.

Наибольшую популярность завоевал HIEW (**http://webhost.kemtel.ru/~sen**). Вплоть до версии 6.11 (поддерживающей форматы MZ/PE/NE/LE/ELF) он распространялся на бесплатной основе (рис. 1.12). Стоит, правда, отметить, что старые и бесплатные версии обладают достаточными возможностями, и к тому же содержат гораздо меньше ошибок и багов.

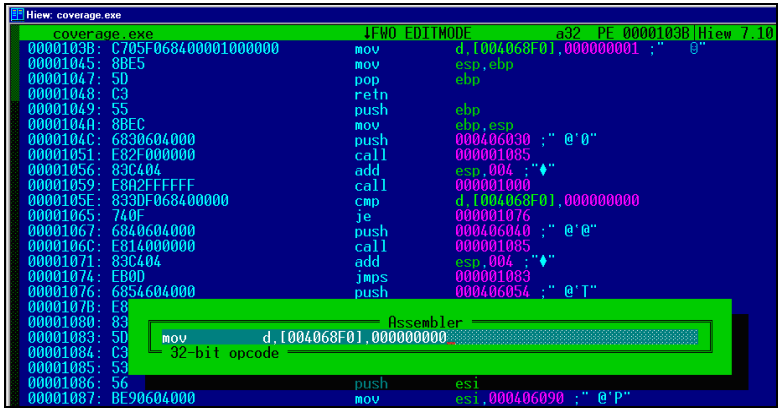

Рис. 1.12. Старый добрый hex-редактор HIEW

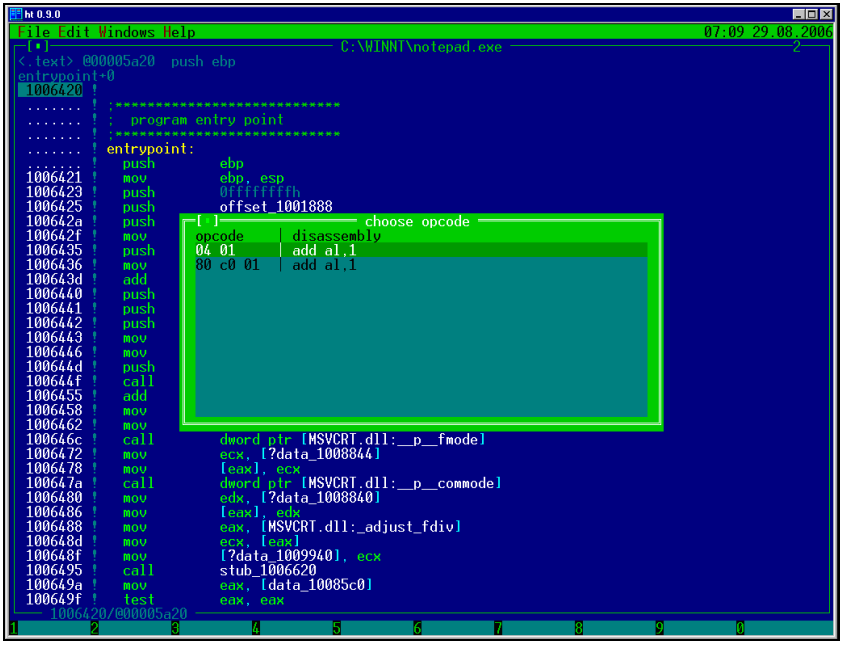

Рис. 1.13. Hex-редактор HTE — достойная замена HIEW

[4] Кстати, вместо них часто использовался редактор диска Norton Disk Editor.

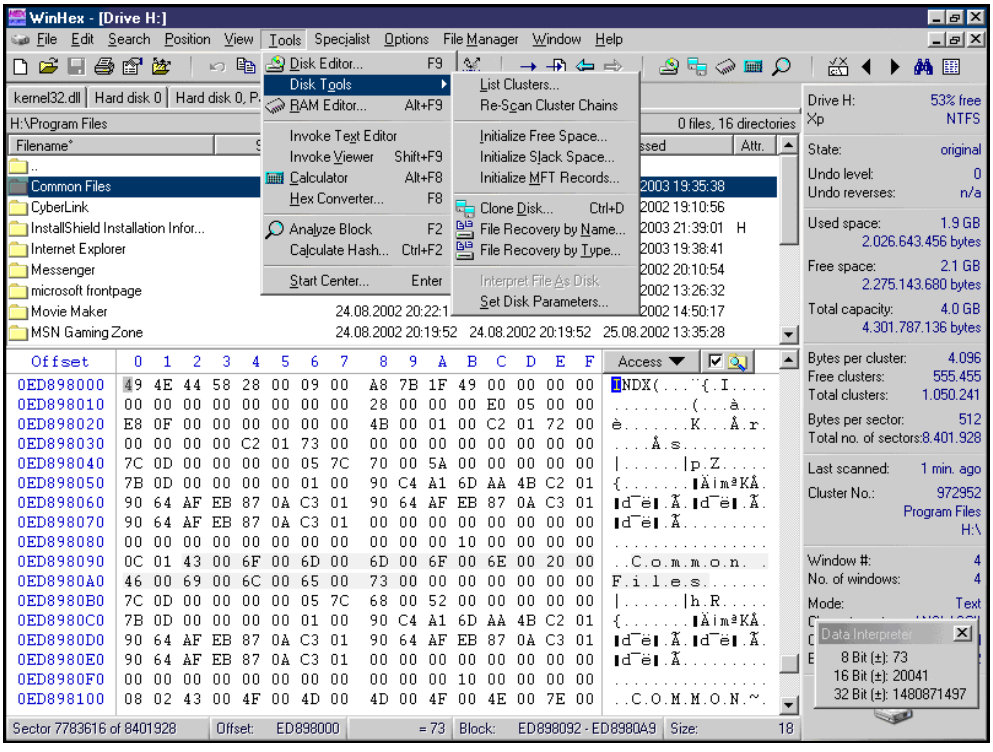

Рис. 1.14. Коммерческий hex-редактор WinHex

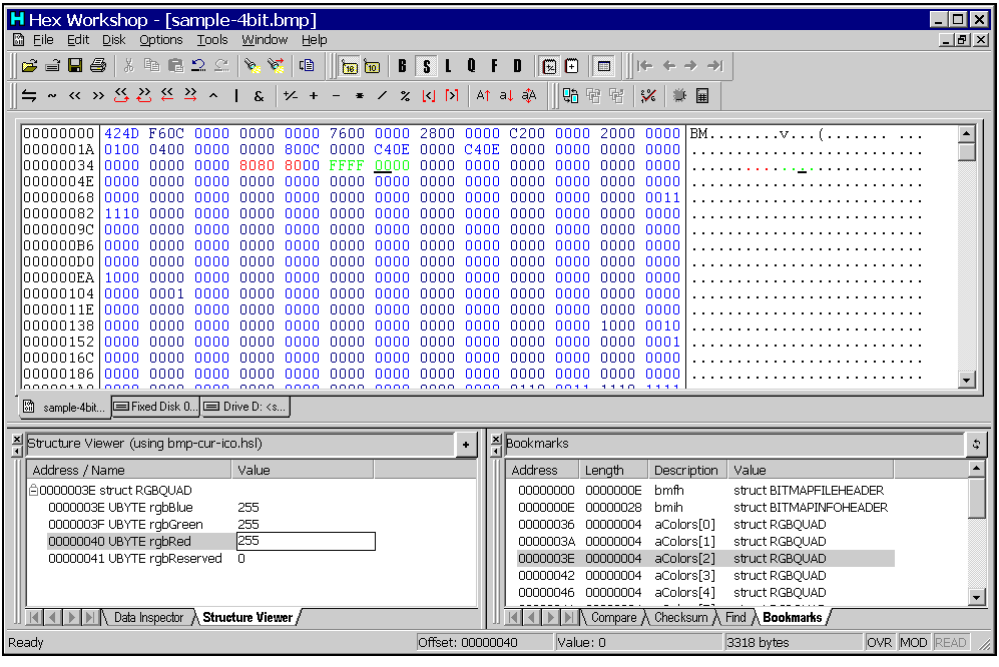

Рис. 1.15. Коммерческий hex-редактор Hex Workshop

Другой хороший редактор, по своим возможностям не только не уступающий HIEW, но даже превосходящий его, — это HTE (**http://hte.sourceforge.net**), распространяющийся в исходных кодах на бесплатной основе. В отличие от HIEW, HTE позволяет выбирать способ ассемблирования инструкции (если инструкция может быть ассемблирована несколькими путями), а также поддерживает мощную систему перекрестных ссылок, вплотную приближающую его к бесплатной версии IDA Pro (рис. 1.13).

Западные хакеры очень любят коммерческие шестнадцатеричные редакторы вроде WinHex (**http://www.winhex.com/winhex/index-m.html**) и Hex Workshop (**http://www.bpsoft.com**). Что привлекательного они в них нашли — непонятно. Ни WinHex (рис. 1.14), ни Hex Workshop (рис. 1.15) не содержат никаких ассемблеров или дизассемблеров, причем маловероятно, что эти функции появятся в дальнейшем. Единственное положительное качество этих редакторов, которое можно отметить, — это наличие калькулятора контрольных и хэш-сумм (например, CRC16, CRC32, MD5, SHA-1), что в некоторых случаях оказывается очень удобным.

Распаковщики

Все больше и больше программ распространяются в упакованном виде (или защищаются протекторами, что еще хуже). Как результат, непосредственное дизассемблирование таких программ становится невозможным. Наконец, поскольку многие упаковщики/протекторы содержат антиотладочные приемы, то страдает и отладка.

Попытки создать универсальный распаковщик многократно предпринимались еще со времен MS-DOS. Всякий раз эти попытки оканчивались полным провалом, поскольку разработчики защит придумывали новую гадость. Тем не менее, в состав большинства хакерских инструментов (IDA Pro, OllyDbg) входят универсальные распаковщики, справляющиеся с несложными защитами. Что касается сложных защит, то, столкнувшись с одной из них, хакер вынужден распаковывать защищенный файл вручную. В *части V* данной книги этот вопрос будет рассмотрен более подробно. Пока же отметим, что когда же один и тот же упаковщик встречается хакеру десятый раз кряду, он садится за написание автоматического или полуавтоматического распаковщика, чтобы облегчить себе работу. Коллекции таких утилит собраны на сайтах **http://www.exetools.com/unpackers.htm**, **http://programmerstools.org/taxonomy/term/16**, **http://www.woodmann.com/crackz/Packers.htm**. Основная проблема состоит в том, что каждый такой распаковщик рассчитан на строго определенную версию упаковщика/протектора и с другими работать просто не может! Чем чаще обновляется упаковщик/протектор, тем сложнее найти подходящий распаковщик, поэтому лучше полагаться только на самого себя.

Кстати, прежде чем искать распаковщик, неплохо бы для начала выяснить: чем же вообще защищена ломаемая программа? В этом поможет бесплатная утилита PEiD (**http://peid.has.it**), содержащая огромную базу сигнатур. Хотя эта программа довольно часто ошибается или дает расплывчатый результат, но, тем не менее, это все-таки лучше, чем совсем ничего.

Дамперы

Снятие дампа с работающей программы — универсальный способ распаковки, позволяющий справиться практически с любым упаковщиком и большинством протекторов. Правда, над полученным дампом еще предстоит как следует поработать, поэтому такие дампы рекомендуется использовать лишь для дизассемблирования. Сломанная таким путем программа может работать неустойчиво, периодически падая в самый ответственный момент.

Какие же дамперы имеются в вашем распоряжении? Первой из утилит этого класса (и самой неуклюжей) была программа ProcDump (**http://www.fortunecity.com/millenium/firemansam/962/html/procdump.html**). Затем появился дампер Lord PE (**http://www.softpedia.com/get/Programming/File-Editors/LordPE.shtml**), учитывающий горький опыт своего предшественника и способный

сохранять дамп даже в тех случаях, когда заголовок файла PE умышленно искажен защитой, а доступ к некоторым страницам памяти отсутствует (атрибут `PAGE_NOACCESS`). Венцом эволюции стал дампер PE Tools (рис. 1.16), базовый комплект поставки которого можно найти практически на любом хакерском сервере, например, на WASM (**http://www.wasm.ru/baixado.php?mode=tool&id=124**) или на CrackLab (**http://www.cracklab.ru/download.php?action=get&n=MTU1**), а свежие обновления лежат на "родном" сайте проекта **http://petools.org.ru/petools.shtml**.

Примечание

"Родной" сайт проекта часто меняет свой адрес.

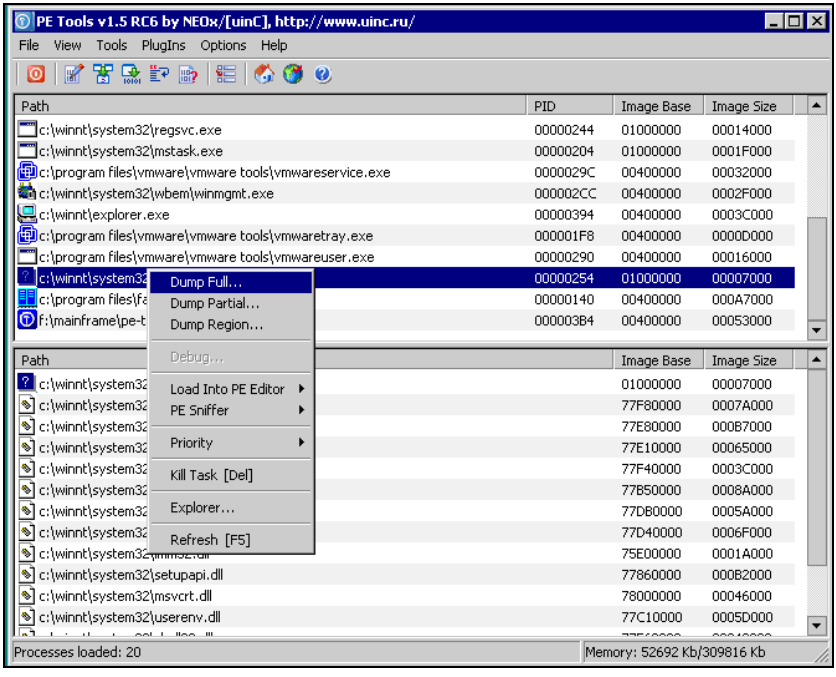

Рис. 1.16. PE Tools — один из лучших дамперов PE-файлов

После снятия дампа необходимо, как минимум, восстановить таблицу импорта, а иногда еще и таблицу перемещаемых элементов вместе с секцией ресурсов. Таблицу импорта лучше всего восстанавливать знаменитой утилитой Import REConstructor, которую вместе с утилитой ReloX (восстанавливающей таблицу перемещаемых элементов) и минимально работающим универсальным распаковщиком можно найти по адресу: **http://wave.prohosting.com/mackt/main.htm**. А вот здесь лежит коллекция программ для восстановления таблицы ресурсов: **http://www.wasm.ru/baixado.php?mode=tool&id=156**. Если же ни одна из этих утилит не справится со своей задачей, то, быть может, поможет бесплатная программа Resource Binder: **http://www.setisoft.com/ru/redirect.php?dlid=89**.

Редакторы ресурсов

Редактировать ресурсы приходится во многих случаях. Например, чтобы сменить текст диалогового окна, разблокировать элемент управления, заменить логотип и т. д. Формально, редактор ресурсов входит в каждый Windows-компилятор, в том числе и в Microsoft Visual Studio. Вот только