

## Чемоданчик хакера

крис касперски, ака мышцх, по-email

чем вообще ломают программы? правильно, головой, задницей и руками. а только потом программами. но первичны все-таки программы, поскольку именно они формируют сознание, позволяя начинающему сделать свои первые шаги в дремучем лесу машинных кодов. вот только этих программ настолько много, что новичок, попавший на хакерский сайт оказывается в полной растерянности — что качать, а что не надо. мышцх дает предельно сжатый обзор хакерского софта, покрывающий потребности практически любого взломщика.



Рисунок 1 чемоданчик хакера в минимальной комплектации

## отладчики

Лучший отладчик всех времен и народов это, конечно же, **soft-ice**, на котором выросло не одно поколение хакеров. Это интерактивная программа с развитым командным интерфейсом,

представляющим собой компромисс между легкостью освоения и удобством использования. Другими словами, руководство читать обязательно. Никаких интуитивно-понятных менюшек в стиле Turbo-Debugger здесь не будет.

Изначально созданный фирмой Nu-Mega, soft-ice был продан компании Compuware, долгое время распространяющей его в составе уродливо framework'a DriverStudio, 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 так же архитектуру AMD x86-64. То есть лет на пять запаса прочности у soft-ice еще должно хватить, а там... мы что-нибудь придумаем.

```

EAX=09000000  EBX=008E60FC  ECX=00000000  EDX=008E77EC  ESI=00000000
EDI=008E78F0  EBP=0012065C  ESP=0011DD8C  EIP=77E92B8D  o d I s z a P c
CS=001B  DS=0023  SS=0023  ES=0023  FS=0038  GS=0000
byte PROT (0)
0010:008E77EC 6B 65 79 66 69 6C 65 2E-64 61 74 00 00 00 1B 00 keyfile.dat....^
0010:008E77FC 72 01 1B 00 73 01 1B 00-69 E0 1B 00 6F E0 1B 00 r...s...i...o...↑
0010:008E780C 6E 01 1B 00 20 01 1B 00-C4 00 1B 00 C4 ED 1B 00 n... .. ↓
0010:008E781C C4 E8 1B 00 C4 5B 1B 00-C4 F4 1B 00 C4 E8 1B 00 .....[.....]
-----KERNEL32!WritePrivateProfileStringW+16D7-----PROT32-
001B:77E92B88 JMP 77E9DD7F
KERNEL32!CreateFileA
001B:77E92B8D PUSH EBP
001B:77E92B8E MOV EBP,ESP
001B:77E92B90 PUSH DWORD PTR [EBP+08]
001B:77E92B93 CALL 77E94DA1
001B:77E92B98 TEST EAX,EAX
001B:77E92B9A JZ 77EB2885
001B:77E92BA0 PUSH DWORD PTR [EBP+20]
001B:77E92BA3 PUSH DWORD PTR [EBP+1C]
(PASSIVE)-KTEB(81216780)-TID(0270)-kernel32!.text+00011B88
NTICE: Load32 START=71760000 SIZE=29000 KPEB=811871C0 MOD=dsquery
NTICE: Load32 START=76AE0000 SIZE=3E000 KPEB=811871C0 MOD=comdlg32
NTICE: Load32 START=71730000 SIZE=1E000 KPEB=811871C0 MOD=dsuixt
NTICE: Load32 START=77BF0000 SIZE=11000 KPEB=811871C0 MOD=ntdsapi
NTICE: Load32 START=77360000 SIZE=2F000 KPEB=811871C0 MOD=activeds
NTICE: Load32 START=77330000 SIZE=22000 KPEB=811871C0 MOD=adsldpc
NTICE: Load32 START=777D0000 SIZE=1D000 KPEB=811871C0 MOD=winspool
NTICE: Unload32 MOD=KMIXER
:bpx CreateFileA
:x
Break due to BPX KERNEL32!CreateFileA (ET=12.64 seconds)
:wd
:d esp->4
:
Enter a command (H for help) Far

```

Рисунок 2 внешний вид отладчика soft-ice

Найти soft-ice можно на любом хакерском сайте или в Осле. Чтобы не качать всю судию целиком (это же без малого 200 метров), можно воспользоваться пакетом DeMoNiX'a (reversing.kulichki.net) — содержащим в себе один лишь soft-ice, выдернутый из Driver Studio v2.7 build 562, и занимающий всего 2,27 Мбайт. Однако, инсталлятор содержит ошибки, а старая версия не поддерживает новых веяний Microsoft (хотя замечательно идет под W2K, мышцх вообще работает с build'ом 334 и полностью им удовлетворен).

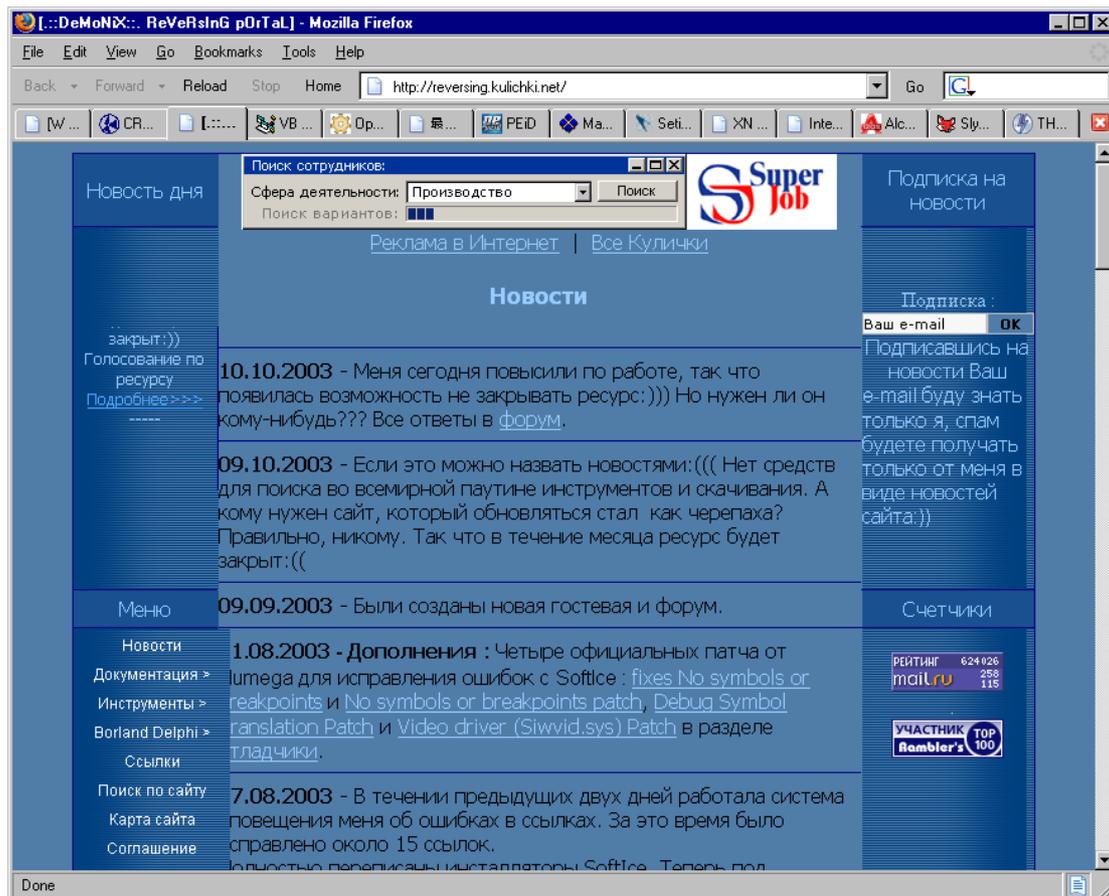


Рисунок 3 сайт DeMoNiX'a

Вместе с soft-ice желательно сразу же установить **IceExt** (sourceforge.net/projects/iceext) — неофициальное расширение, позволяющее скрывать отладчик от взора большинства защит, дампить память, задействовать кириллические кодировки 866/1251, приостанавливать потоки и делать множество других вещей (например, играть в тетрис).

```

EAX=00300E40  EBX=7FFDF000  ECX=004060B8  EDX=00000003  ESI=00000000
EDI=00000000  EBP=0012FFC0  ESP=0012FF84  EIP=00401001  o d I s z a P c
CS=001B  DS=0023  SS=0023  ES=0023  FS=0038  GS=0000
-----test_dump-----byte-----PROT---(0)
0010:00400000  4D 5A 90 00 03 00 00 00-04 00 00 00 FF FF 00 00  MZÉ.....^
0010:00400010  B8 00 00 00 00 00 00 00-40 00 00 00 00 00 00 00  .....@.....↑
0010:00400020  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  .....↓
0010:00400030  00 00 00 00 00 00 00 00-00 00 00 00 D0 00 00 00  .....
-----PROT32-----
001B:00401001  PUSH      00406030
001B:00401006  CALL     00401010
001B:0040100B  POP      ECX
001B:0040100C  RET
001B:0040100D  NOP
001B:0040100E  NOP
001B:0040100F  NOP
001B:00401010  PUSH     EBX
001B:00401011  PUSH     ESI
001B:00401012  MOV      ESI,00406068
(PASSIVE)-KTEB(811EB020)-TID(0190)-test_dump!.text+0001
: !DUMP
Dump memory to disk
!dump FileName Addr Len
Ex:
!dump c:\dump.dat 400000 1000
!dump \??\c:\dump.dat 400000 1000
!dump \??\c:\dump.dat edx+ebx ecx
: !DUMP C:\dumped 400000 7DE8
DUMP: \??\C:\dumped 400000 7de8
:
Enter a command (H for help) test_dum

```

Рисунок 4 снятие дампа с помощью IceExt

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

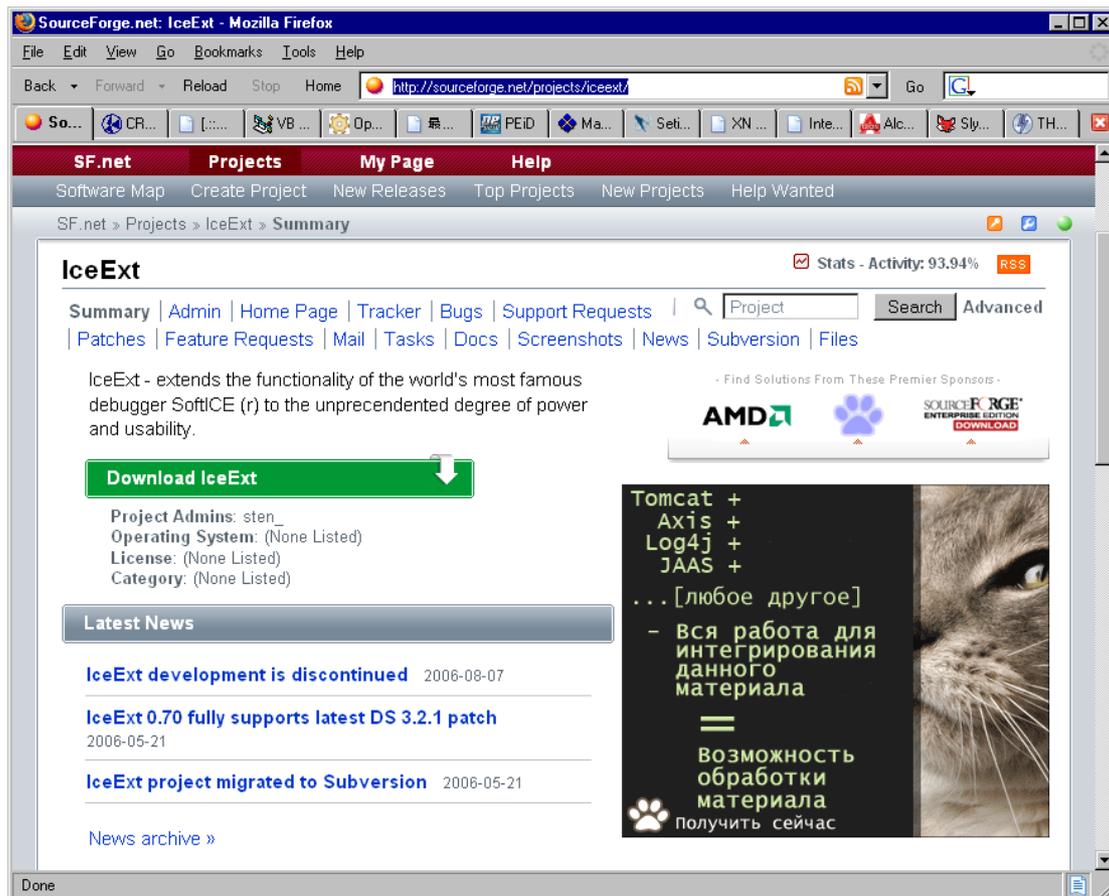


Рисунок 5 отсюда раздают IceExt

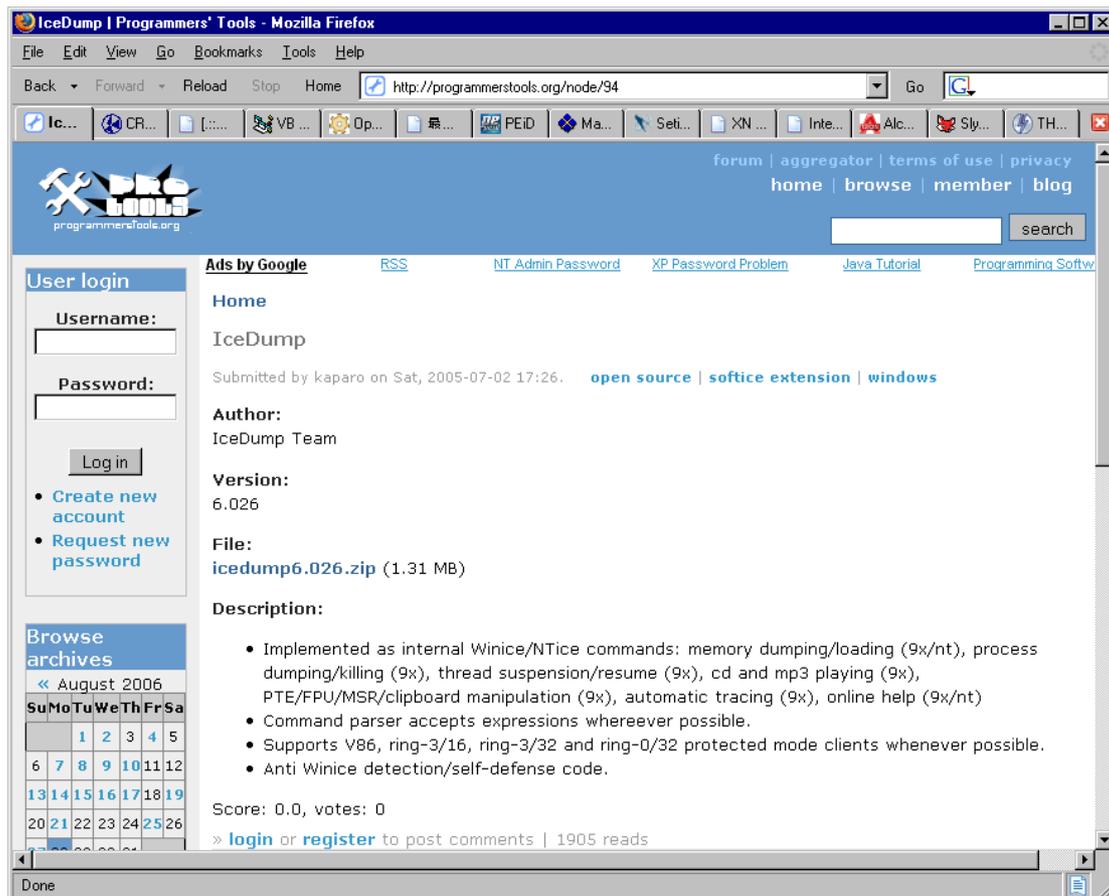
Другое неофициальное расширение для soft-ice **IceDump** (programmerstools.org/system/files?file=icedump6.026.zip) так же умеет делать много полезных вещей, и удачно дополняет IceExt.

```

EAX=00300E40  EBX=7FFDF000  ECX=004060B8  EDX=00000003  ESI=00000000
EDI=00000000  EBP=0012FFC0  ESP=0012FF84  EIP=00401001  o d I s z a P c
CS=001B  DS=0023  SS=0023  ES=0023  FS=0038  GS=0000
-----test_dump!.text+0001-----byte-----PROT-----(0)-----
001B:00401001  68 30 60 40 00 E8 05 00-00 00 59 C3 90 90 90 53  h0`@.ë...Y.ÉÉÉES^
001B:00401011  56 BE 68 60 40 00 57 56-E8 4B 01 00 00 8B F8 8D  U.h`@.WU#K...i.i↑
001B:00401021  44 24 18 50 FF 74 24 18-56 E8 04 02 00 00 56 57  D$.P.t$.U#...UW↓
001B:00401031  8B D8 E8 BE 01 00 00 83-C4 18 8B C3 5F 5E 5B C3  i.ë...â..i.^[.v
-----PROT32-----
001B:00401001  PUSH      00406030
001B:00401006  CALL     00401010
001B:0040100B  POP      ECX
001B:0040100C  RET
001B:0040100D  NOP
001B:0040100E  NOP
001B:0040100F  NOP
001B:00401010  PUSH     EBX
001B:00401011  PUSH     ESI
001B:00401012  MOV      ESI,00406068
(PASSIVE)-KTEB(812121E0)-TID(03B0)-test_dump!.text+0001
:MOD test_dump
hMod Base PEHeader Module Name File Name
00400000 004000D0 test_dump \TEMP\test_dump.exe
:MAP32 test_dump
Owner Obj Name Obj# Address Size Type
test_dump .text 0001 001B:00401000 00003B46 CODE R0
test_dump .rdata 0002 0023:00405000 0000080E IDATA R0
test_dump .data 0003 0023:00406000 00001DE8 IDATA RW
:
Enter a command (H for help) test_dum

```

Рисунок 6 снятие дампа с помощью IceDump



**Рисунок 7 IceDump на programmerstools**

Кстати, сам soft-ice замечательно работает под виртуальной машиной VM Ware, для этого достаточно добавить в vmx-файл пару строк: `raevm = TRUE` и `processor1.use = FALSE`. Так же отмечены проблемы с многоядерными и HT-процессами (хотя и не у всех). Лечится путем отрубания всего этого хозяйства через добавление ключа `/ONECPU` в файл `boot.ini`.

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

Неплохую коллекцию плагинов можно найти на wasm'еи на [www.openrce.org](http://www.openrce.org).

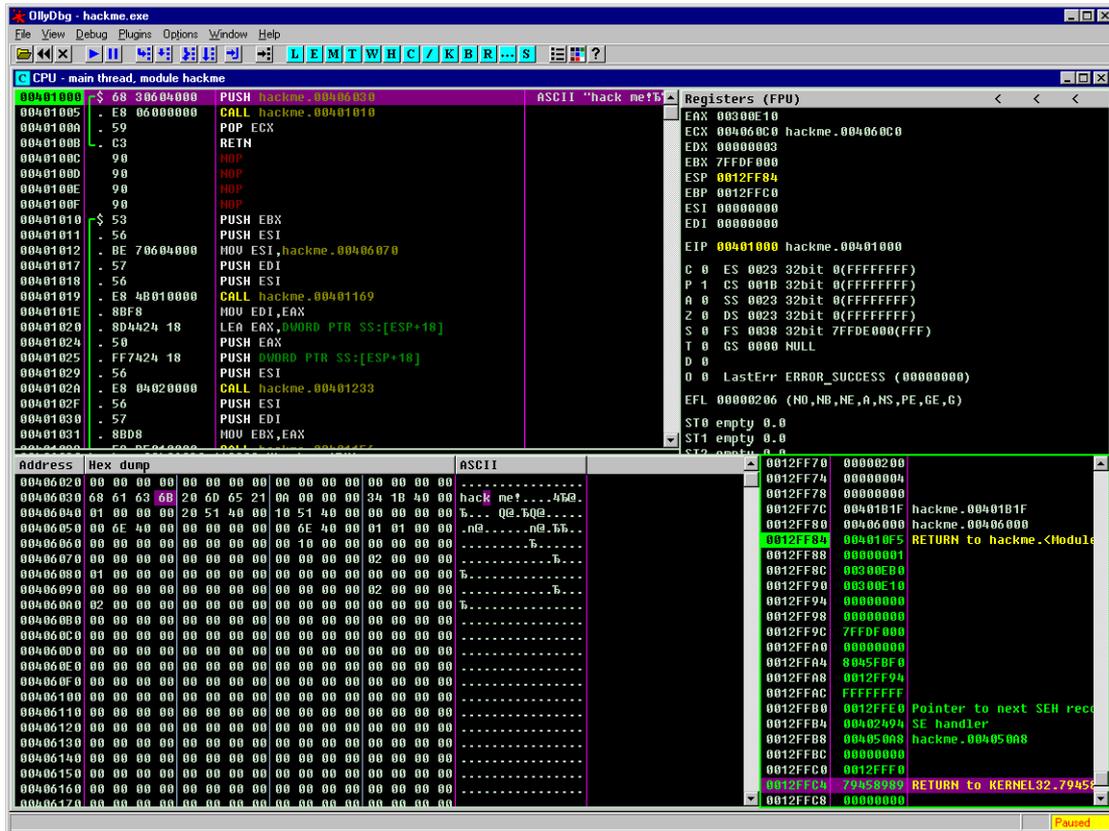


Рисунок 8 внешний вид отладчика OllyDbg

Самый свежий (и пока еще во многом экспериментальный) ядерный отладчик это, бесспорно, **SYSER** ([www.sysersoft.com](http://www.sysersoft.com)), выпущенный нашими китайскими братьями и в настоящее время переживающий стадию активного развития и становления.

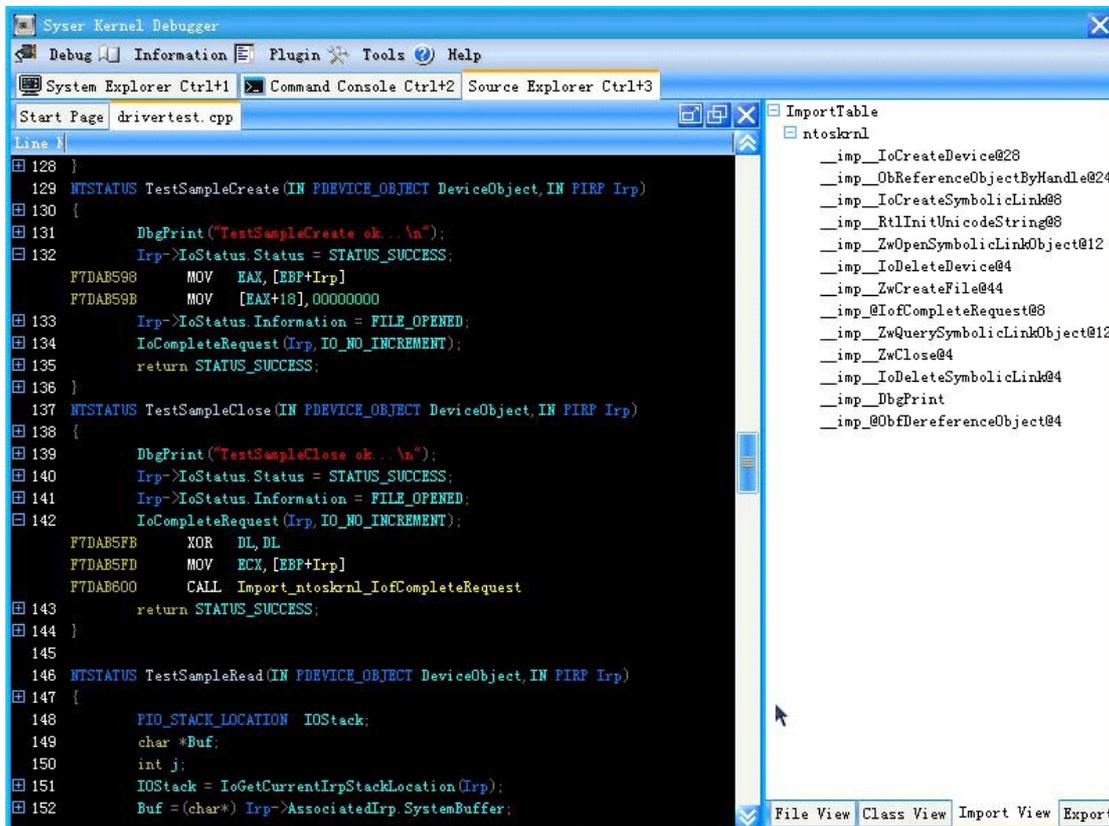


Рисунок 9 отладчик SYSER за отладкой термоядерного драйвера

Довольно много народа пользуется **Microsoft WinDeb**, входящим в состав бесплатного набора Debugging Tools. Он вполне пригоден для взлома только... уж очень неудобен для тех, кто привык к черному экрану soft-ice.

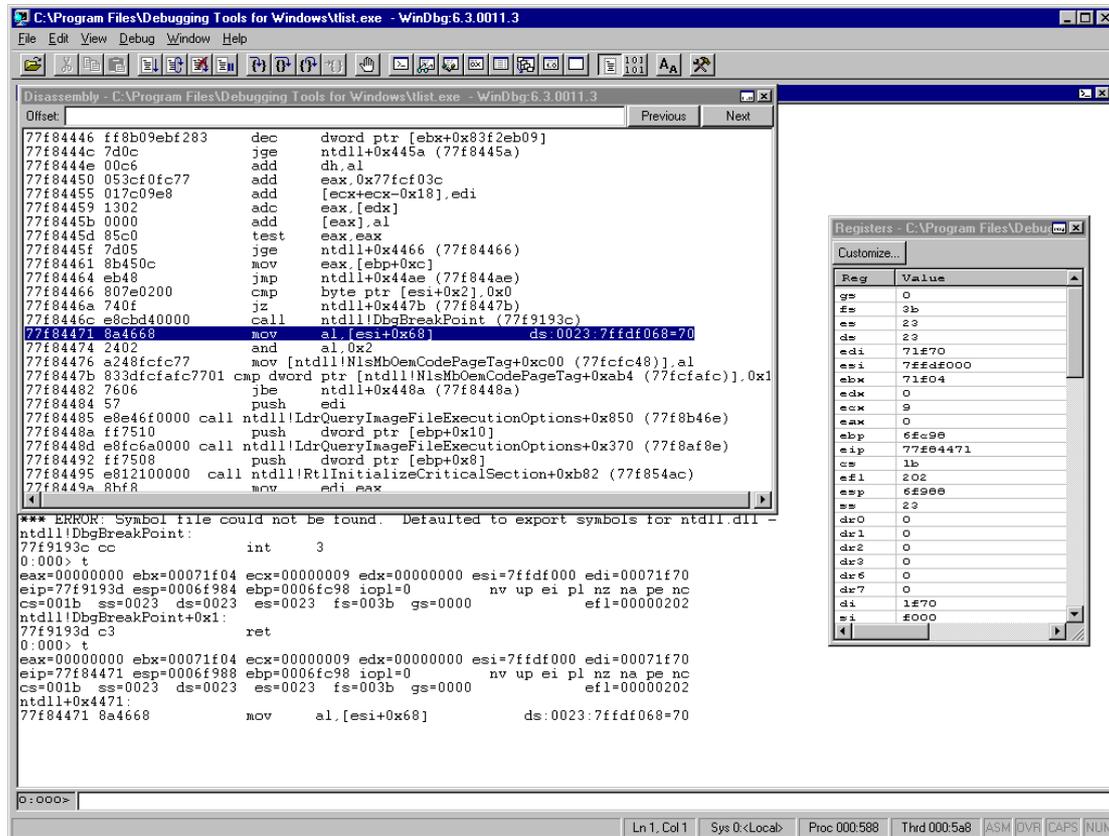


Рисунок 10 отладчик WinDeb

## дизассемблеры

Существует всего лишь один дизассемблер, пригодный для профессиональной работы — **IDA Pro** ([www.idapro.com](http://www.idapro.com)), стоящий немереных денег, но как и всякое другое добро, свободно валяющийся в Осле.



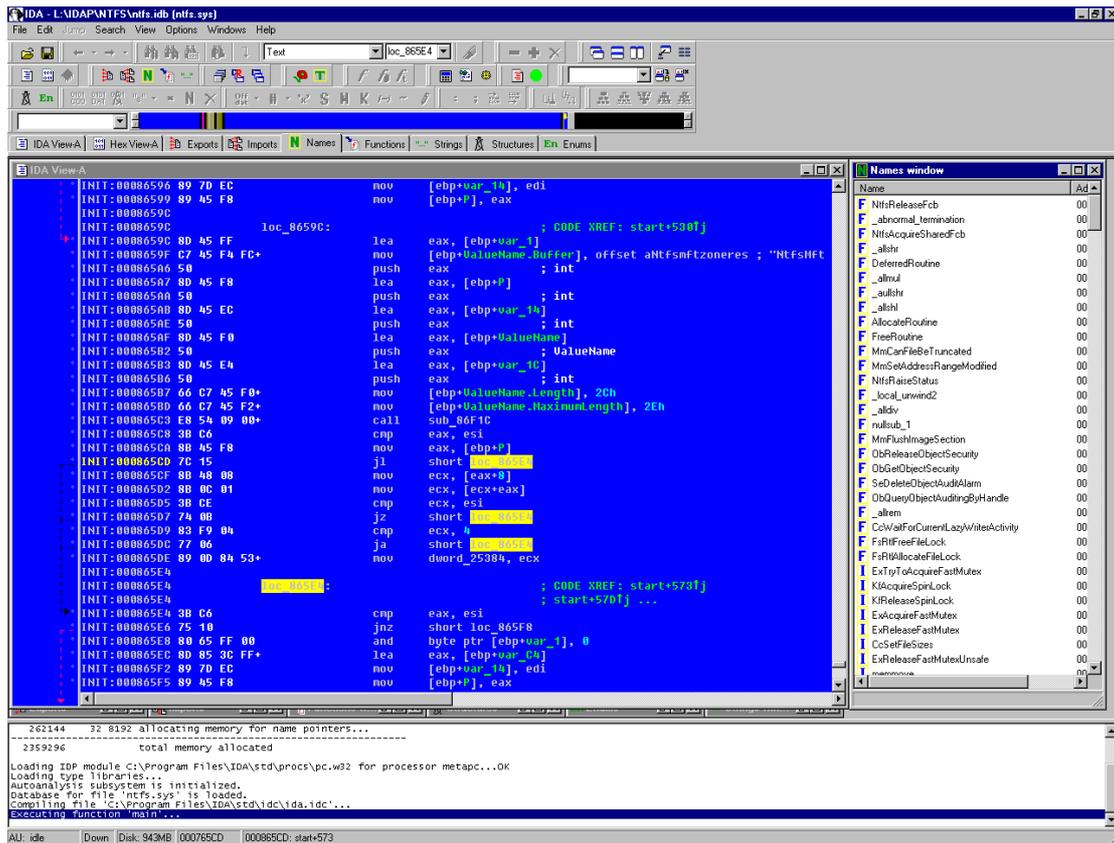


Рисунок 12 графическая версия IDA Pro

Самое главное — IDA Pro – это интерактивный дизассемблер, то есть *инструмент*, позволяющий работать с двоичным файлом, мыслить и творить, а не тупой автомат, заглазывающий хакаемую программу и выплевывающий "готовый" дизассемблированный листинг, в котором все дизассемблированного неправильно.

В последних версиях IDA PRO сделаны определенные подвижки в сторону автоматической распаковки файлов и снятия обфускаторов. Внутреннюю коллекцию плагинов и скриптов можно найти как на официальном сайте, так и на [www.openngse.org](http://www.openngse.org).

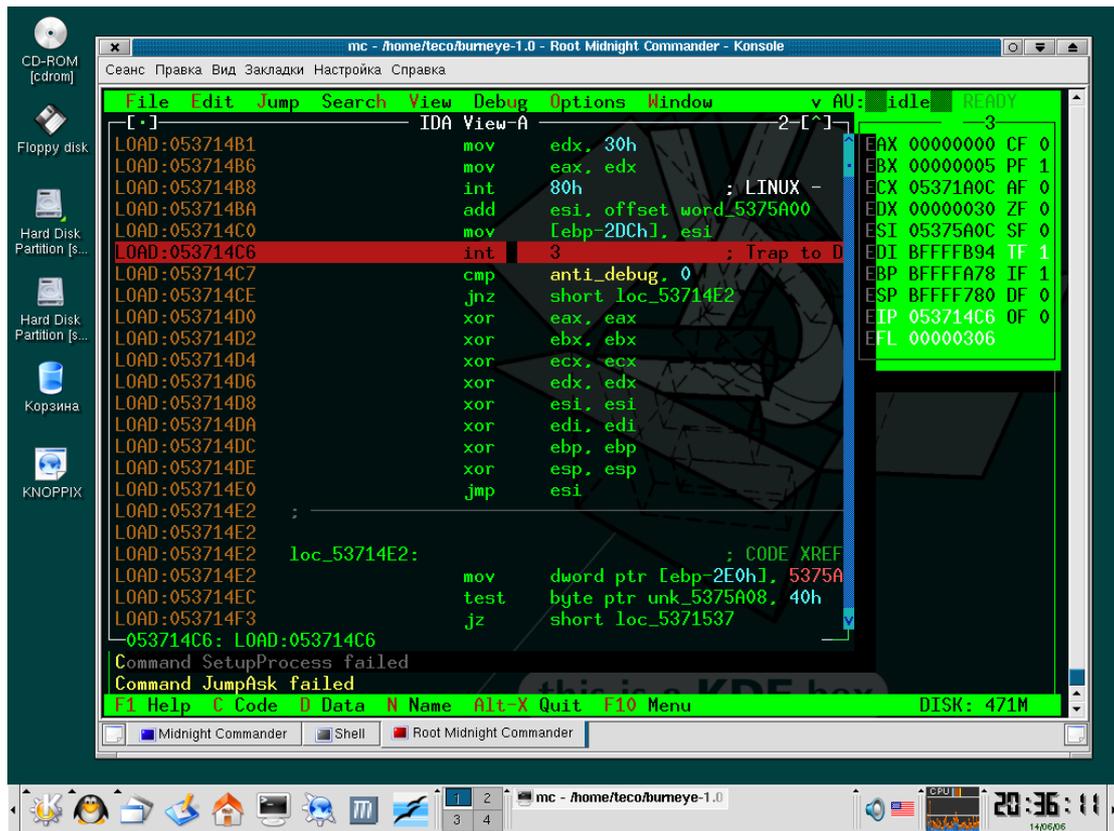


Рисунок 13 IDA Pro под LINUX

Конкурентам до IDA Pro еще расти и расти как до Луны. Тем не менее, народ активно качает бесплатный (ныне заброшенный) дизассемблер и отладчик в одном "флаконе" — WDasm: [www.wasm.ru/baixado.php?mode=tool&id=178](http://www.wasm.ru/baixado.php?mode=tool&id=178) и, судя по всему остается доволен, хотя чтобы взломать с его помощью что-то серьезное, легче задницу напололам разорвать.

```

URSoft W32Dasm Ver 8.93 Program Disassembler/Debugger
Disassembler Project Debug Search Goto Execute Text Functions HexData Refs Help

* Referenced by a CALL at Addresses:
|:0040105C , :004041EE , :004042A2
-----
* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:00460233(U)
|
:00401075 55          push ebp
:00401076 8BEC         mov ebp, esp
:00401078 51          push ecx
:00401079 6A00         push 00000000
:0040107B 6844104000   push 00401044
:00401080 C70120624600 mov dword ptr [ecx], 00466220

* Reference To: KERNEL32.SetConsoleCtrlHandler, Ord:02E3h
|
:00401086 FF1510604600 Call dword ptr [00466010]
:0040108C 85C0         test eax, eax
:0040108E 7515         jne 004010A5
:00401090 8D45FC         lea eax, dword ptr [ebp-04]
:00401093 68D0844600   push 004684D0
:00401098 50          push eax

* Possible StringData Ref from Data Obj ->"SetConsoleCtrlHandler fails"
|
:00401099 C745FCF0304700 mov [ebp-04], 004730F0
:004010A0 E828700500   call 004580CD

* Referenced by a (U)nconditional or (C)onditional Jump at Address:
|:0040108E(C)
|
:004010A5 C9          leave
:004010A6 C3          ret

:004010A7 E854FFFFFF   call 00401000
:004010AC F6D8         neg al
:004010AE 1BC0         sbb eax, eax

```

Рисунок 14 дизассемблер WDASM

Остальные дизассемблеры выглядят еще более убого, поэтому не будем их рассматривать, разве что отметим **Hacker Disassembler Engine** ([patkov-site.narod.ru/lib.html](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 нечто очень на него похожее) из двоичного файла. В полном объеме декомпиляция невозможна в принципе, поскольку компиляция — однонаправленный процесс, причем с потерей данных. Однако, декомпиляторы все-таки существуют и со своей задачей достойно справляются.

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

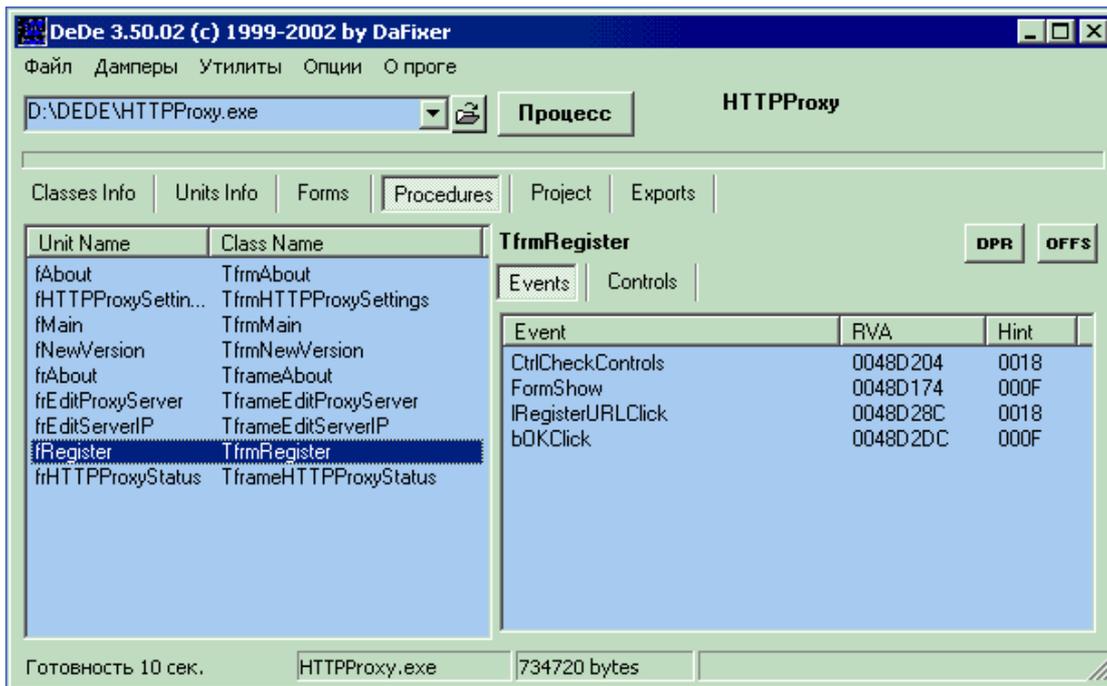


Рисунок 15 декомпилятор DeDe ломает HTTPProxy

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

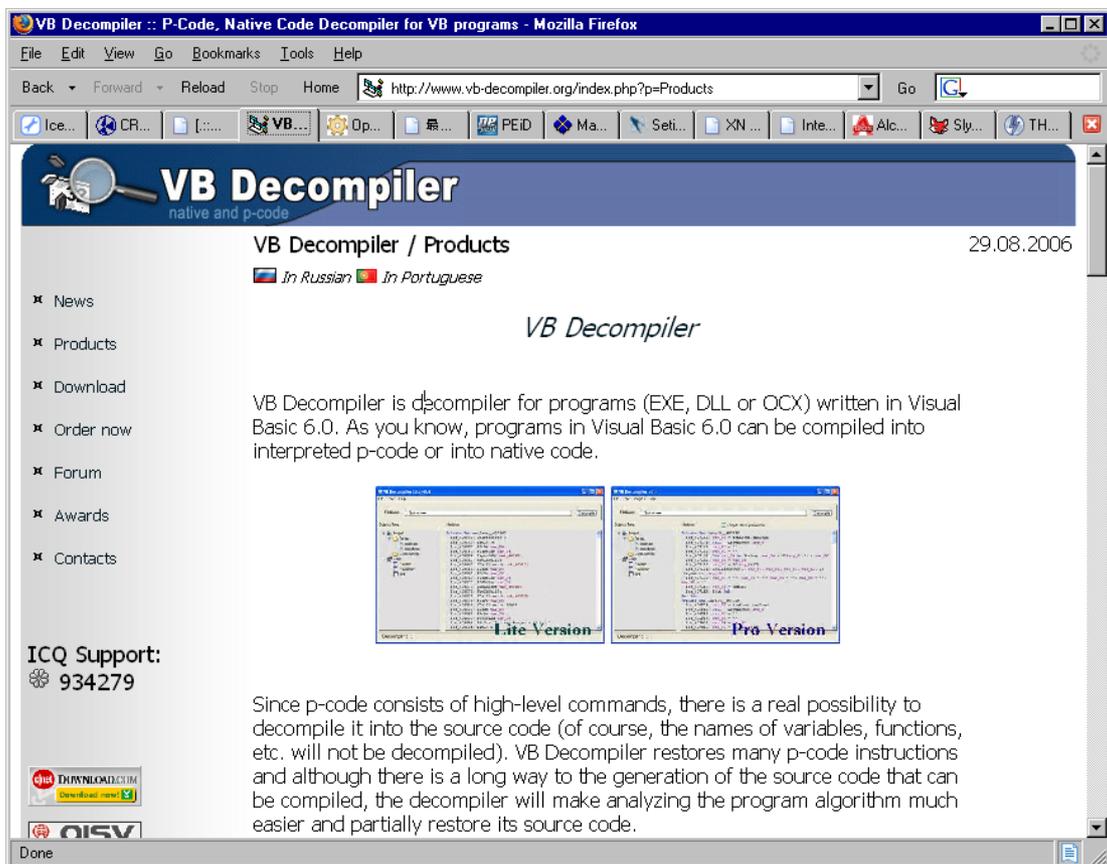


Рисунок 16 официальный сайт одного из лучших декомпиляторов Visual Basic'a

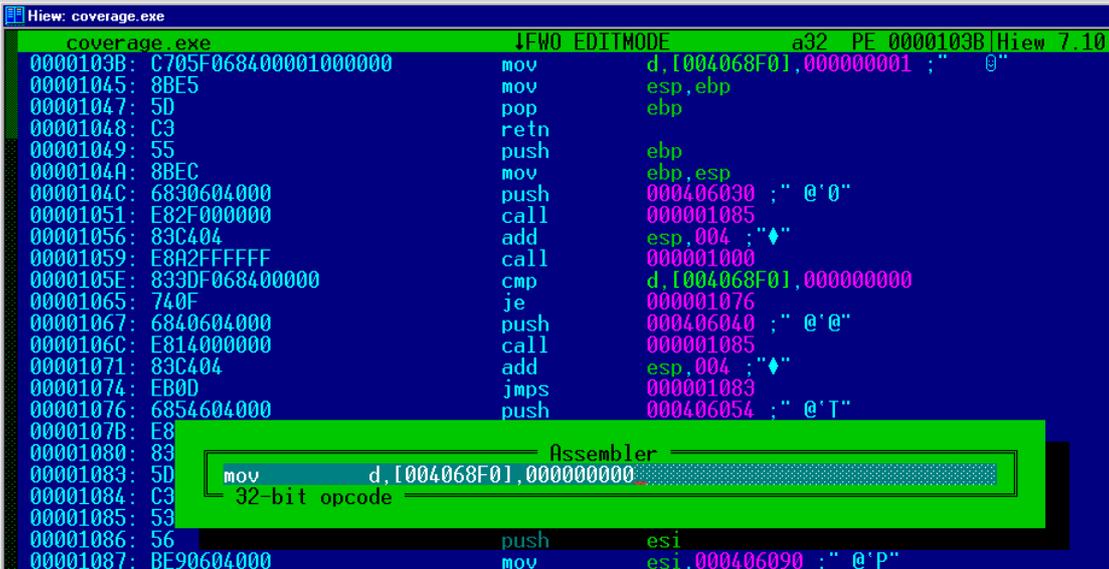
Большой интерес представляют декомпиляторы инсталляторов, поскольку многие проверки (на истечение трального срока, на серийный номер или ключевой файл) производятся как раз на стадии инсталляции. Самый популярный инсталлятор Install Shield и вот куча декомпиляторов к нему: **InstallShield X Unpacker** (programmerstools.org/node/154), **Windows Installshield Decompiler** (programmerstools.org/node/118), **InstallShield Decompiler** (programmerstools.org/node/114) и всякая мелочь типа isDcc (programmerstools.org/node/115).

Что же касается Java и платформы .NET, то с ними замечательно справляется IDA Pro, а если ее под рукой нет, можно воспользоваться специализированными декомпиляторами, которые можно найти на сайтах [www.cracklab.ru](http://www.cracklab.ru) и [www.wasm.ru](http://www.wasm.ru) вместе с декомпиляторами Fox Pro, Clirper'a и прочей экзотики.

## hex-редакторы

Давным-давно hex-редакторы представляли собой простые программы, умеющие всего лишь отображать двоичный файл в шестнадцатеричном виде и править байты по указанным адресам (кстати, вместо них часто использовался редактор диска Norton Disk Editor), но со временем они обросли дизассемблерами, ассемблерами, встроенными калькуляторами, функциями регулярного поиска, научились работать с блоками, понимать различные форматы файлов и даже расшифровывать/зашифровывать фрагменты кода/данных. В общем, эдакий швейцарский ножичек с шестнадцатью лезвиями.

Наибольшую популярность завоевал **HIEW** ([webhost.kemtel.ru/~sen](http://webhost.kemtel.ru/~sen)). Вплоть до версии 6.11 (поддерживающей MZ/PE/NE/LE/ELF-форматы) он распространяется на бесплатной основе, а теперь за него просят денежку, которую мыщх платить не хочет и продолжает пользоваться своей любимой 6.04, в которой гораздо меньше ошибок и багов.



```
coverage.exe          IFWO EDITMODE          a32 PE 0000103B |Hiew 7.10
0000103B: C705F068400010000000  mov     d,[004068F0],00000001 ;" 0"
00001045: 8BE5                mov     esp,ebp
00001047: 5D                  pop     ebp
00001048: C3                  retn
00001049: 55                  push   ebp
0000104A: 8BEC                mov     ebp,esp
0000104C: 6830604000         push   000406030 ;" @'0"
00001051: E82F000000         call   000001085
00001056: 83C404             add     esp,004 ;" 4"
00001059: E8A2FFFF           call   000001000
0000105E: 833DF068400000    cmp     d,[004068F0],00000000
00001065: 740F               je      000001076
00001067: 6840604000         push   000406040 ;" @'@"
0000106C: E814000000         call   000001085
00001071: 83C404             add     esp,004 ;" 4"
00001074: EB0D               jmps   000001083
00001076: 6854604000         push   000406054 ;" @'T"
0000107B: E8
00001080: 83
00001083: 5D
00001084: C3
00001085: 53
00001086: 56
00001087: BE90604000         push   esi
mov     esi,000406090 ;" @'P"
```

Рисунок 17 старый добрый hex-редактор hiew

Другой хороший редактор, по своим возможностям не только не уступающий hiew'у, но даже превосходящий его — это **HTE** ([hte.sourceforge.net](http://hte.sourceforge.net)), распространяющийся в исходных кодах на бесплатной основе и в отличие от hiew'a позволяющий выбирать способ ассемблирования инструкции (если инструкция может быть ассемблирована более чем одним путем), а так же поддерживающий мощную систему перекрестных ссылок, вплотную приближающий его к IDA Pro.

```

ht 0.9.0
File Edit Windows Help
C:\WINNT\notepad.exe
07:09 29.08.2006
<.text> @00005a20 push ebp
entrypoint+0
1006420 !
! *****
! : program entry point
! *****
! entrypoint:
!          push     ebp
1006421 !          mov      ebp, esp
1006423 !          push     0fffffffh
1006425 !          push     offset_1001888
100642a !          push     [ ] choose opcode
100642f !          mov      opcode | disassembly
1006435 !          push     04 01 | add al,1
1006436 !          mov      80 c0 01 | add al,1
100643d !          add
1006440 !          push
1006441 !          push
1006442 !          push
1006443 !          mov
1006446 !          mov
100644d !          push
100644f !          call
1006455 !          add
1006458 !          mov
1006462 !          mov
100646c !          call    dword ptr [MSVCRT.dll:._p_fmode]
1006472 !          mov     ecx, [?data_1008844]
1006478 !          mov     [eax], ecx
100647a !          call   dword ptr [MSVCRT.dll:._p_commode]
1006480 !          mov     edx, [?data_1008840]
1006486 !          mov     [eax], edx
1006488 !          mov     eax, [MSVCRT.dll:._adjust_fdiv]
100648d !          mov     ecx, [eax]
100648f !          mov     [?data_1009940], ecx
1006495 !          call   stub_1006620
100649a !          mov     eax, [data_10085c0]
100649f !          test    eax, eax
1006420/@00005a20

```

Рисунок 18 hex-редактор НТЕ — достойная замена hiew'y

Западные хакеры во всю прутя от коммерческих WinHex'a ([www.winhex.com/winhex/index-m.html](http://www.winhex.com/winhex/index-m.html)) и Hex Workshop'a ([www.bpssoft.com](http://www.bpssoft.com)). Чего они в них нашли — непонятно.

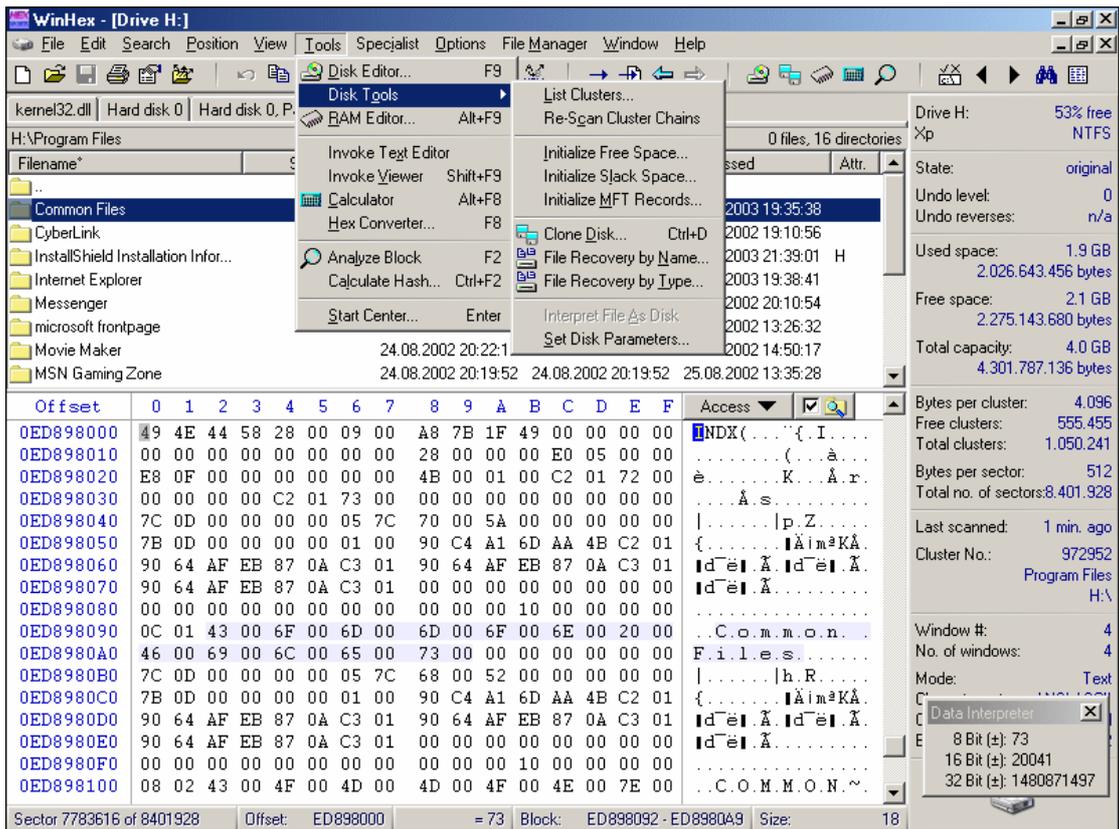


Рисунок 19 коммерческий hex-редактор WinHex популярный на западе

Ни ассемблера, ни дизассемблера в нем нет и навряд ли появятся в дальнейшем, зато есть калькулятор контрольных и хэш сумм (типа CRC16, CRC32, MD5, SHA-1), что в некоторых случаях оказывается очень удобным.

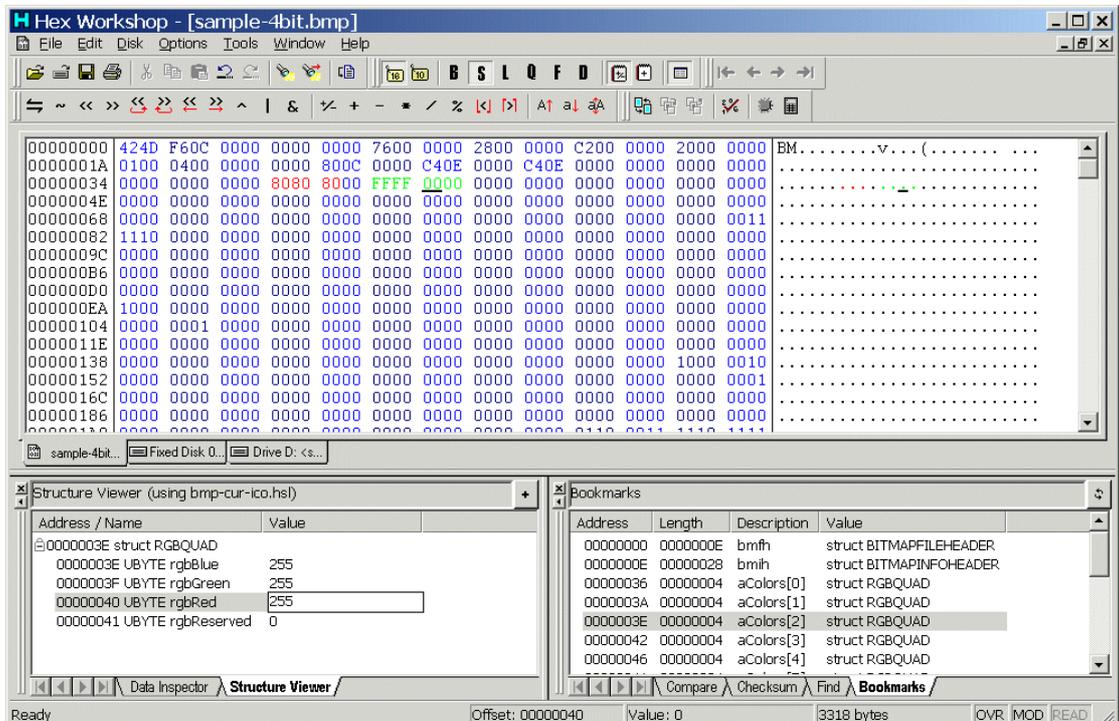


Рисунок 20 коммерческий hex-редактор Hex Workshop популярный на западе

## распаковщики

Все больше и больше программ распространяются в упакованном виде (или защищаются протекторами, что еще хуже), в результате чего их непосредственное дизассемблирование становится невозможным, а, поскольку, многие упаковщики/проекторы содержат анти-отладочные приемы, то страдает и отладка.

Попытки создать универсальный распаковщик многократно предпринимались еще со времен MS-DOS и всякий раз проваливались, поскольку разработчики защит придумывали новую гадость. Тем не менее в состав большинства хакерских инструментов (IDA Pro, OllyDbg) входят генетические распаковщики, справляющие с несложными защитами. Сложные же приходится распаковывать руками (тому, как это сделать посвящено множество статей, которые легко найти в сети). Когда же один и тот же упаковщик встречается хакеру десятый раз кряду, он матерится и пишет автоматический/полуавтоматический распаковщик, чтобы облегчить себе работу. Коллекции таких распаковщиков собраны на [www.exetools.com/unpackers.htm](http://www.exetools.com/unpackers.htm), [programmerstools.org/taxonomy/term/16](http://programmerstools.org/taxonomy/term/16), [www.woodmann.com/crackz/Packers.htm](http://www.woodmann.com/crackz/Packers.htm) и других сайтах. Проблема 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, распаковывая программу руками (**и мы уже писали как**).

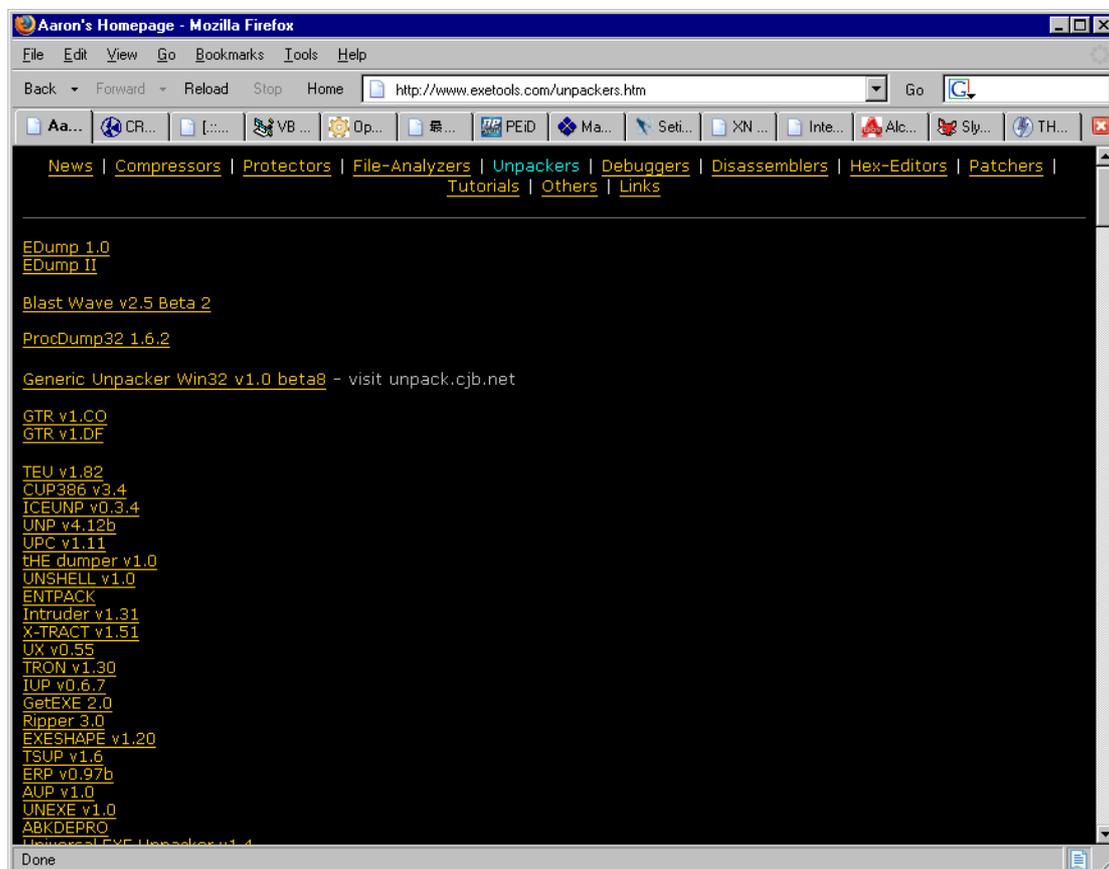


Рисунок 21 большая коллекция распаковщиков на exetools

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

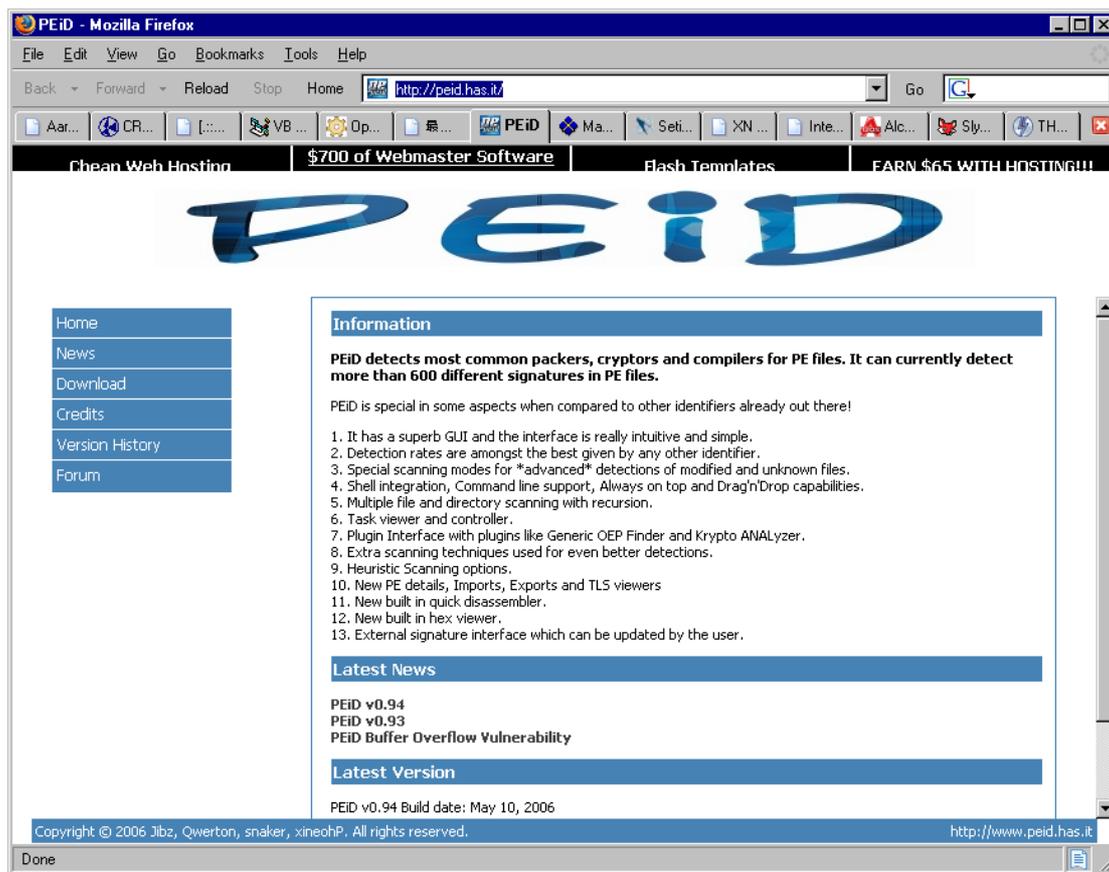


Рисунок 22 официальный сайт утилиты PEiD

## дамперы

Снятие дампа с работающей программы — универсальный способ распаковки, убивающий практически все упаковщики и большую часть протекторов, правда, над полученным дампом еще предстоит как следует поработать и мышь рекомендует использовать дампы лишь для дизассемблирования. Сдампленная программа может работать неустойчиво, периодически падая в самый ответственный момент. Но это ладно, это все лирика. Забьем на лирику и обратимся к практике.

Самым первым (и самым неумелым) был **ProcDump**, затем появился **Lord PE**, учитывающий горький опыт своего предшественника и способный сохранять дампы даже в тех случаях, когда PE-заголовки умышленно искажены защитной, а доступ к некоторым страницам памяти отсутствует (атрибут `PAGE_NOACCESS`). Венцом эволюции стал **PE-TOOLS**, базовый комплект поставки которого можно найти практически на любом хакерском сервере, например, на WASM'e ([www.wasm.ru/baixado.php?mode=tool&id=124](http://www.wasm.ru/baixado.php?mode=tool&id=124)) или на CrackLab'e ([www.cracklab.ru/download.php?action=get&n=MTU1](http://www.cracklab.ru/download.php?action=get&n=MTU1)), а свежие обновления лежат на "родном" сайте проекта [neoh.iatp.by](http://neoh.iatp.by), кстати говоря, уже несколько раз поменявшим свой адрес (по непонятным причинам, базовый пакет на нем отсутствует).

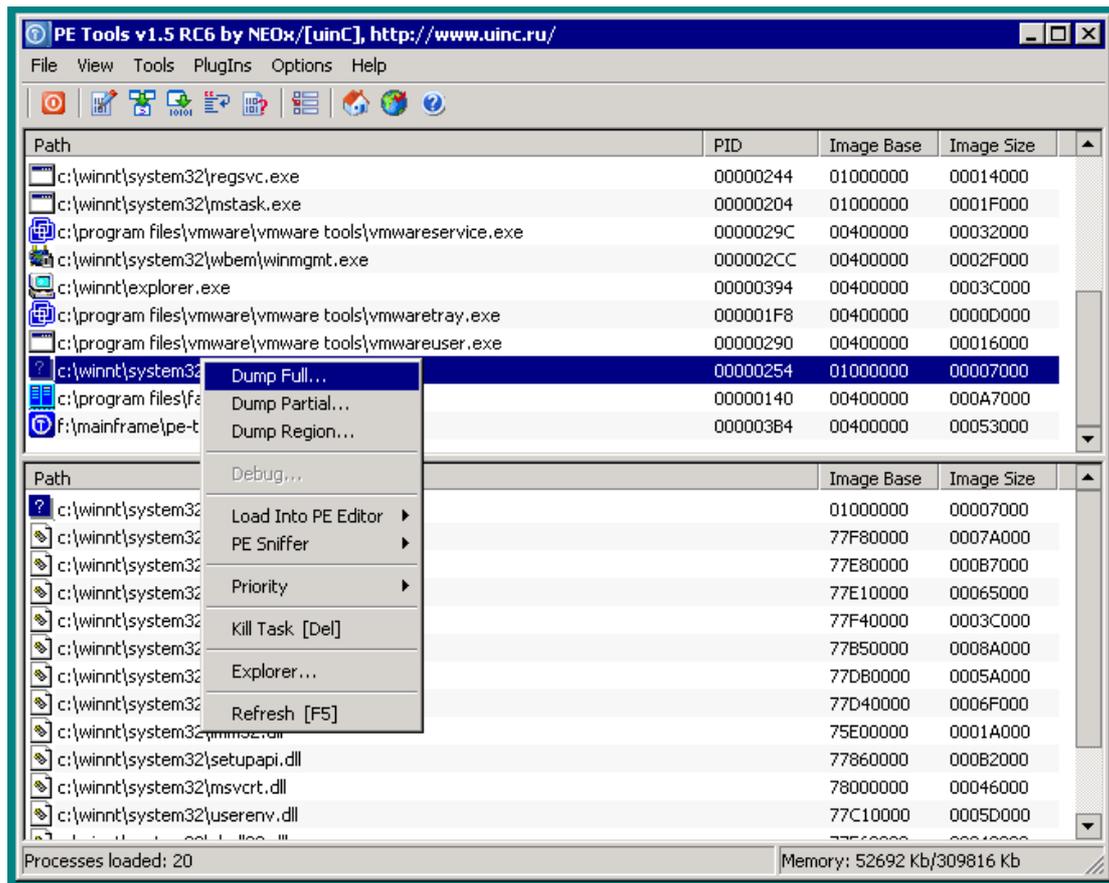


Рисунок 23 PE-TOOLS – один из лучших дамперов PE-файлов

После снятия дампа необходимо как минимум восстановить таблицу импорта, а иногда еще и таблицу перемещаемых элементов вместе с секцией ресурсов. Импорт лучше всего восстанавливать знаменитым **Import REConstructor**'ом, который вместе с **ReloX**'ом (восстанавливающим таблицу перемещаемых элементов) и минимально работающим генетическим распаковщиком можно найти там же, где и PE-TOOLS: [wave.prohosting.com/mackt/main.htm](http://wave.prohosting.com/mackt/main.htm). А вот здесь лежит коллекция программ для восстановления таблицы ресурсов [www.wasm.ru/baixado.php?mode=tool&id=156](http://www.wasm.ru/baixado.php?mode=tool&id=156), если же ни одна из них не справится со своей задачей, то, быть может, поможет бесплатный **Resource Binder**: [www.setisoft.com/ru/redirect.php?dlid=89](http://www.setisoft.com/ru/redirect.php?dlid=89)

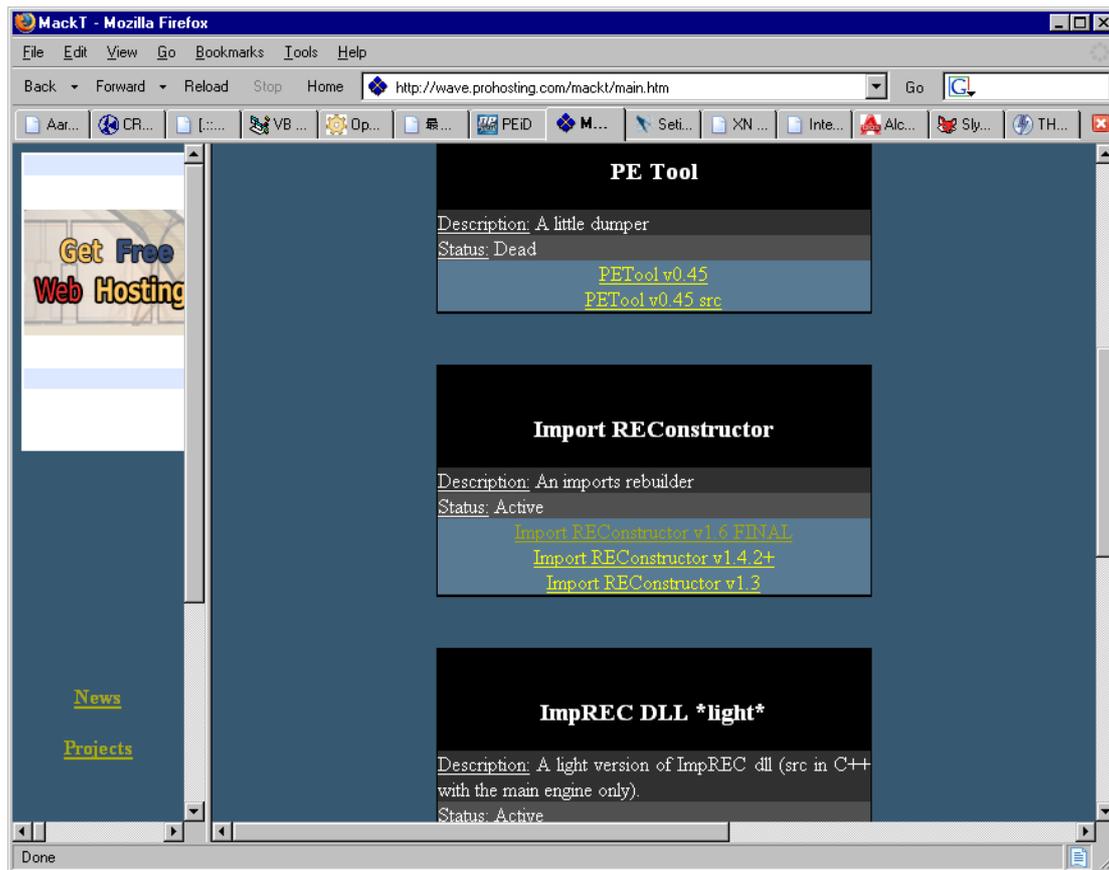


Рисунок 24 на этом сайте можно найти Import REConstructor и другие программы

## ***редакторы ресурсов***

Редактировать ресурсы приходится во многих случаях. Например, чтобы сменить текст диалогового окна, разблокировать элемент управления, перебить логотип и т. д. и т. п. Формально, редактор ресурсов входит в каждый Windows-компилятор, в том числе и в Microsoft Visual Studio, вот только после редактирования ресурсов файл зачастую становится неработоспособным! Это потому что штатный редактор ресурсов к таким задачам непригоден!

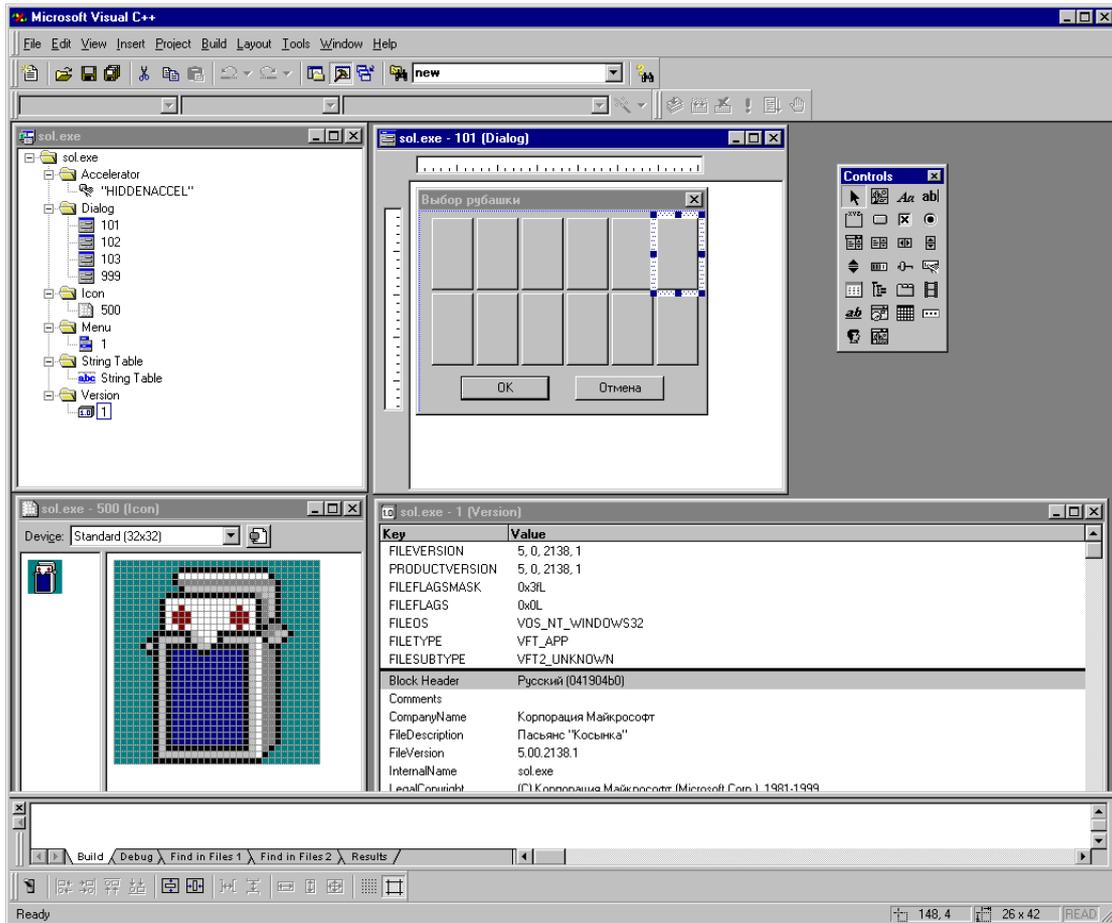


Рисунок 25 редактирование ресурсов в Microsoft Visual Studio

Лучим хакерским редактором был и остается коммерческий **Restorator Resource Editor** ([www.bome.com/Restorator](http://www.bome.com/Restorator)), который может практически все, что нужно и даже чуточку больше. Из бесплатных утилит в первую очередь хотелось бы отметить **XN Resource Editor** ([www.wilsonc.demon.co.uk/d10resourceeditor.htm](http://www.wilsonc.demon.co.uk/d10resourceeditor.htm)), написанный на совсем не хакерском языке DELPHI и распространяющегося в исходных текстах, что позволяет наращивать функционал программы, затачивая ее под свои собственные нужды (если вы знаете DELPHI, конечно).

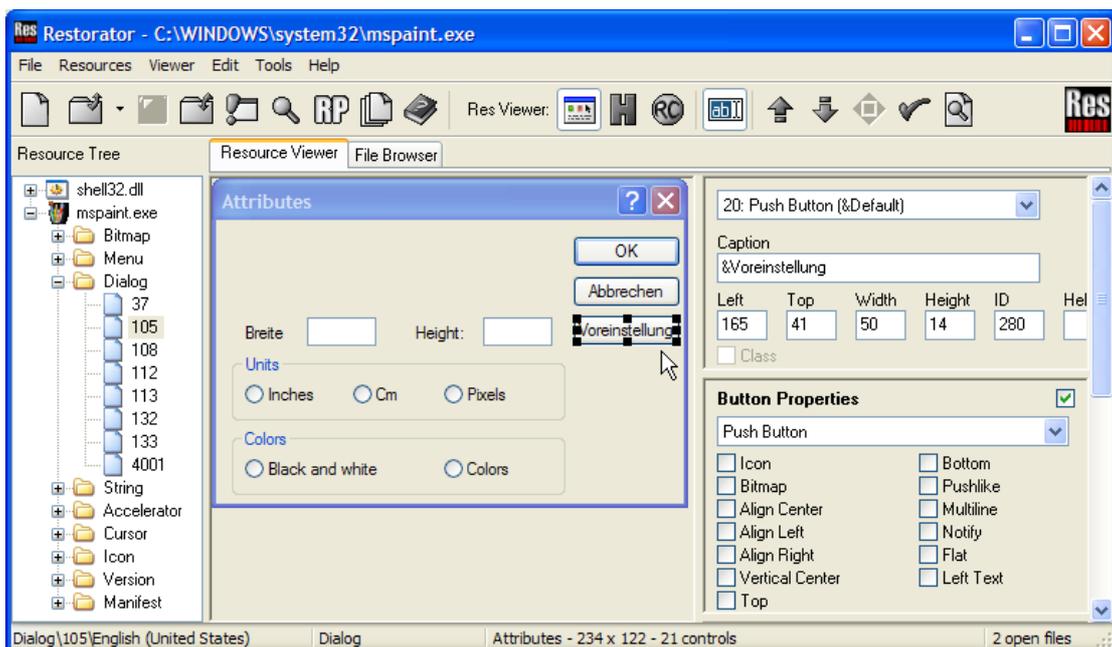


Рисунок 26 редактирование ресурсов в Restorator Resource Editor

## ШПИОНЫ

В основном используется два типа шпионов — шпионы Windows-сообщений и API-шпионы. Первые следят за посылкой сообщений окнам и элементам управления, вторые — за вызовом API-функций, включая функции, экспортируемые динамическими библиотеками, поставляемыми вместе с программой. Шпионаж — лучшее (и наиболее дешевое — в смысле усилий и времени) средство, позволяющее узнать чем "дышит" защищенная программа.

Вполне достойный шпион сообщений входит в штатную поставку Microsoft Visual Studio и называется **Spyxx.exe**. Аналогичный по возможностям шпион, но только с открытыми исходными текстами лежит на [www.catch22.net/software/winspy.asp](http://www.catch22.net/software/winspy.asp) и совершенно бесплатен.

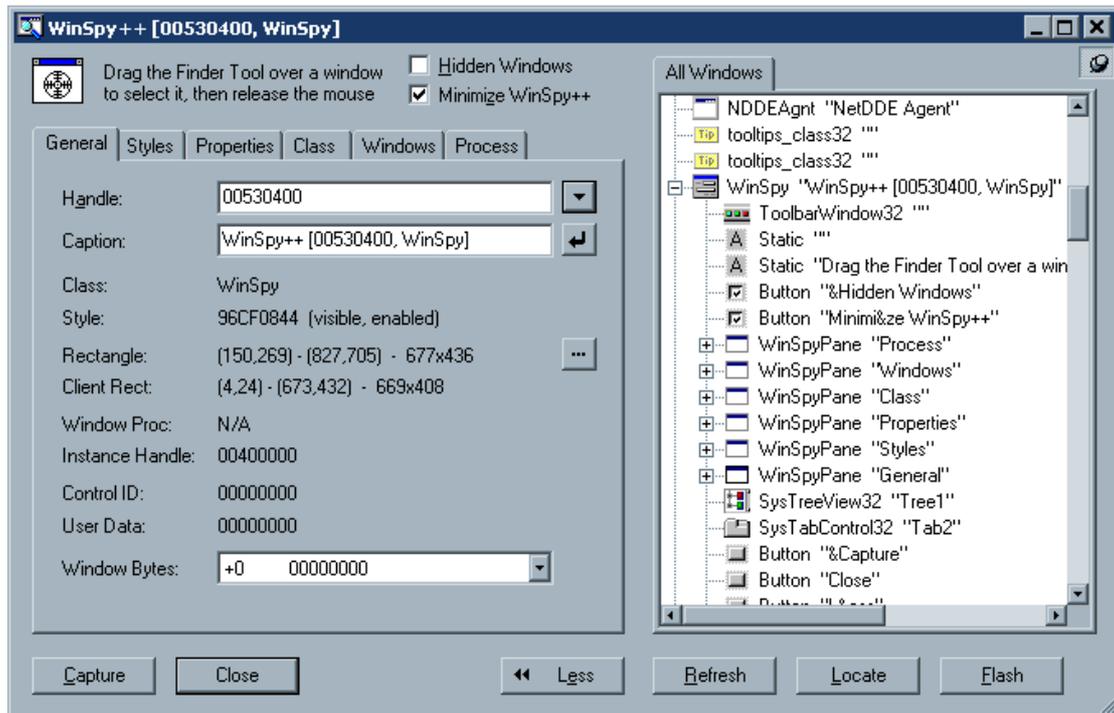


Рисунок 27 шпионаж за Windows-сообщениями при помощи бесплатной утилиты WinSpy

Из API-шпионов, лучшим на мой мышцѣный взгляд является **Kerberos** от Рустема Фасихова ([www.wasm.ru/baixado.php?mode=tool&id=313](http://www.wasm.ru/baixado.php?mode=tool&id=313)), который взялся за клавиатуру тогда, когда остальные шпионы перестали его устивать. Тем не менее, о вкусах не спорят и многие пользуются **APISpy32** (таким же бесплатным как и Kerberos) который можно раздобыть на [www.internals.com](http://www.internals.com). Впрочем, любой нормальный отладчик (как-то soft-ice, OllyDbg) можно настроить так, чтобы он выполнял функции API-шпиона, причем действуя по очень избирательному шаблону, избавляющему нас от просмотра многокилометровых листингов, генерируемых Kerberos'ом и APISpy32.

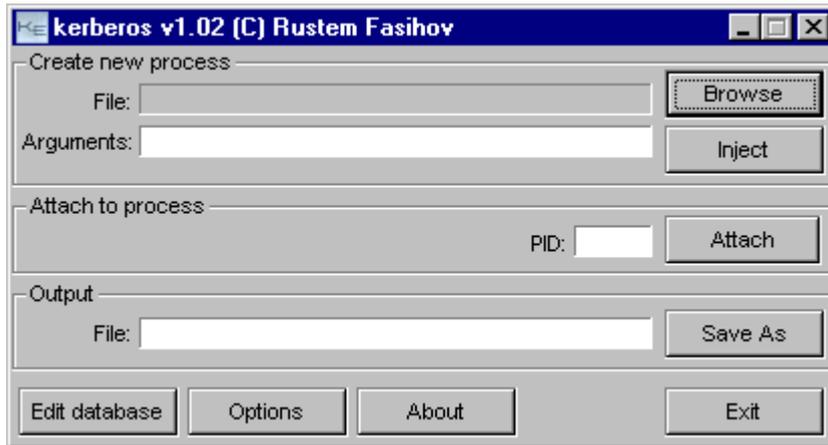


Рисунок 28 внешний вид API-шпиона Kerberos от Рустама Фасихова

## мониторы

Чтобы узнать к каким файлам или ветвям реестра обращается подопытная программа достаточно воспользоваться файловым монитором и монитором реестра соответственно.

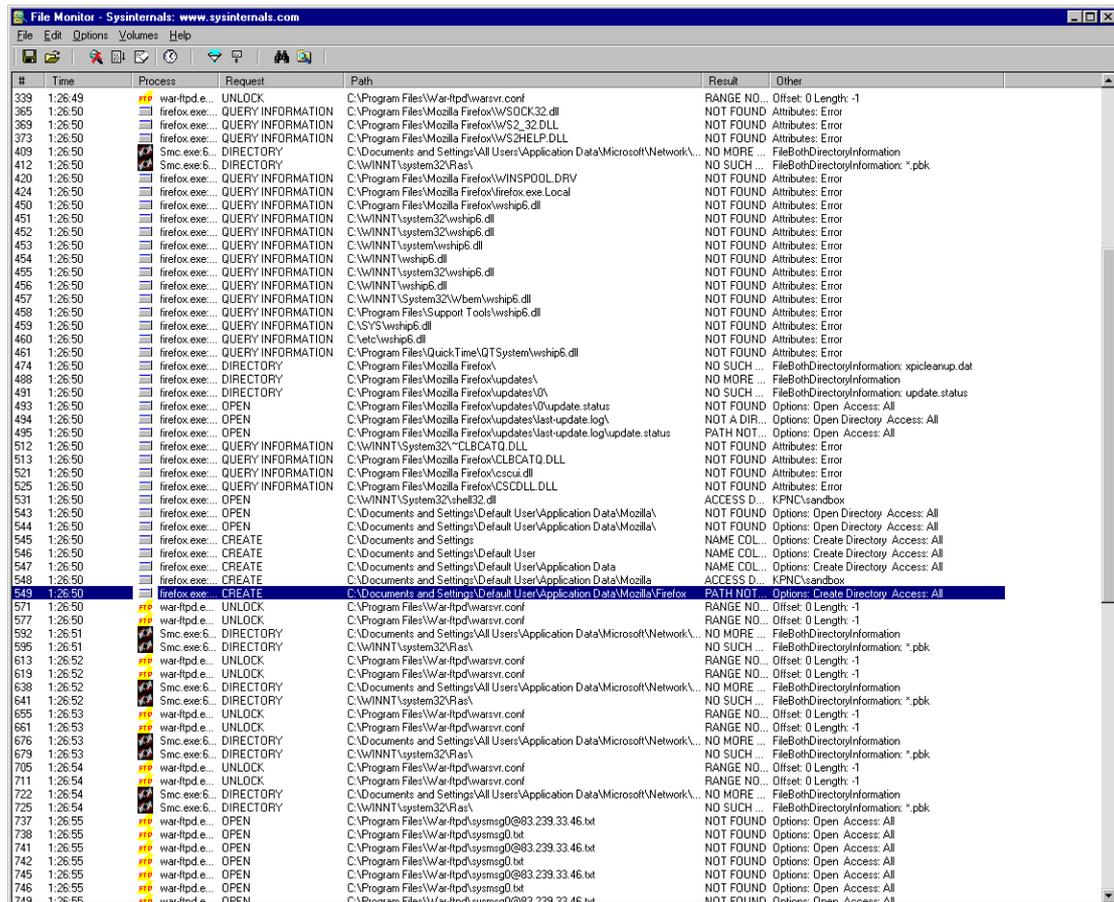


Рисунок 29 файловый монитор Марка Русиновича за работой

Оба они были написаны легендарным исследователем недр Windows Марком Русиновичем и долгое время распространялись совершенно бесплатно через некоммерческий сайт [www.sysinternals.com](http://www.sysinternals.com), однако, в июне 2006 года Русинович продан Microsoft и хотя его utility обещают остаться бесплатными и впредь, скорее всего они будут бесплатными *только для легальных пользователей Windows*, так что спешите качать пока дают на шару.

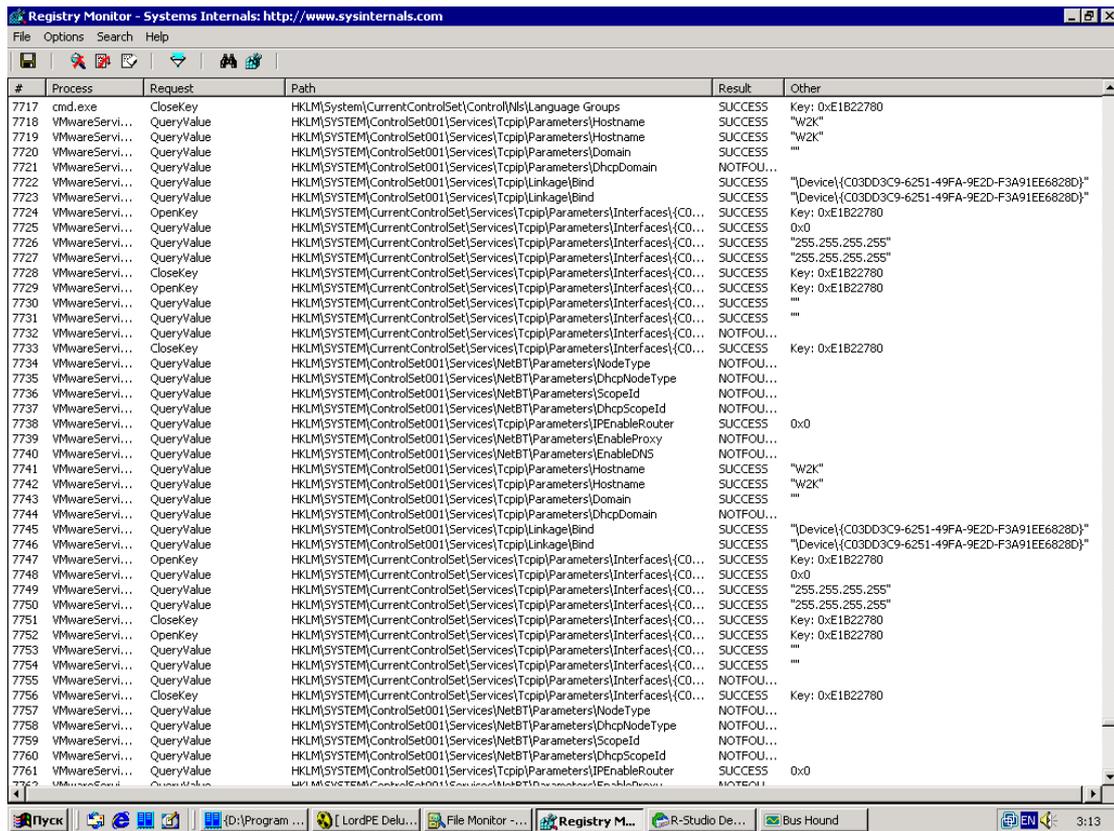


Рисунок 30 монитор реестра Марка Руссиновича

## модификаторы

Существует два диаметрально противоположных подхода к взлому программ. Самое трудное (но самое идеологически правильное и наименее всего наказуемое) создание своих собственных генераторов серийных номеров, ключевых файлов и т. д. Проанализировав, как работает оригинальный генератор, хакер пишет точно такой же и раздает его всем кому надо (и кому не надо — тоже). Однако, это слишком утомительно, тем более что большинство защит нейтрализуются правкой нескольких байт. Вот только распространять взломанный файл нельзя. За это могут и по лапкам дать. К тому же, как правило, exe/dll слишком тяжелы для распространения, поэтому, возникает естественная идея — распространять не сам взломанный файл, а список байт с адресами, которые надо исправить. Понятное дело, никакой юзер с hiew'ом внутрь программы не полезет, поэтому на помощь приходит автоматизация.

Получить список различий между оригинальным и взломанным файлом поможет утилита **fc.exe**, входящая в штатный комплект поставки Windows, а вот, чтобы внести исправления в exe/dll, понадобится утилита-модификатор, которую можно написать буквально за несколько минут, а если писать лень, то вот коллекция уже готовых: [www.wasm.ru/baixado.php?mode=tool&id=35](http://www.wasm.ru/baixado.php?mode=tool&id=35).

```
C:\WINNT\system32\CMD.EXE
Microsoft Windows 2000 [Версия 5.00.2195]
(C) Корпорация Майкрософт, 1985-2000.

L:\ARTICLE\hacker\hack-toolz>fc /b demo.exe demo_hacked.exe
Сравнение файлов demo.exe и DEMO_HACKED.EXE
000010A4: 75 74
000010FE: E8 90
000010FF: 22 90
00001100: 0A 90
00001101: 00 90
00001102: 00 90

L:\ARTICLE\hacker\hack-toolz>_
```

Рисунок 31 поиск различий между оригинальной и хакнутой версией файла с помощью штатной утилиты FC.EXE

Хуже есть программа упакована/защищена протектором. Тогда ее приходится править уже налету, непосредственно в оперативной памяти, для чего пригодится **Process Patcher** ([www.wasm.ru/baixado.php?mode=tool&id=38](http://www.wasm.ru/baixado.php?mode=tool&id=38)), **R!SC's Process Patcher** ([www.wasm.ru/baixado.php?mode=tool&id=39](http://www.wasm.ru/baixado.php?mode=tool&id=39)) или **\*ABEL\* Self Learning Loader Generator** ([www.wasm.ru/baixado.php?mode=tool&id=144](http://www.wasm.ru/baixado.php?mode=tool&id=144)). Последняя программа отличается тем, что ищет исправляемые байты не по фиксированным смещениям, а по регулярным шаблонам, что позволяет ей в большинстве случаев переживать выход новой, слегка измененной версии ломаемой программы (если, конечно, изменения затронули не защитный механизм, а что-нибудь другое).

### **копировщики защищенных дисков**

Копировать защищенные диски — это совсем не по-хакерски, а гораздо более по пиратски, тем не менее, заниматься этой деятельностью всем приходится, так что лишними они не будут.

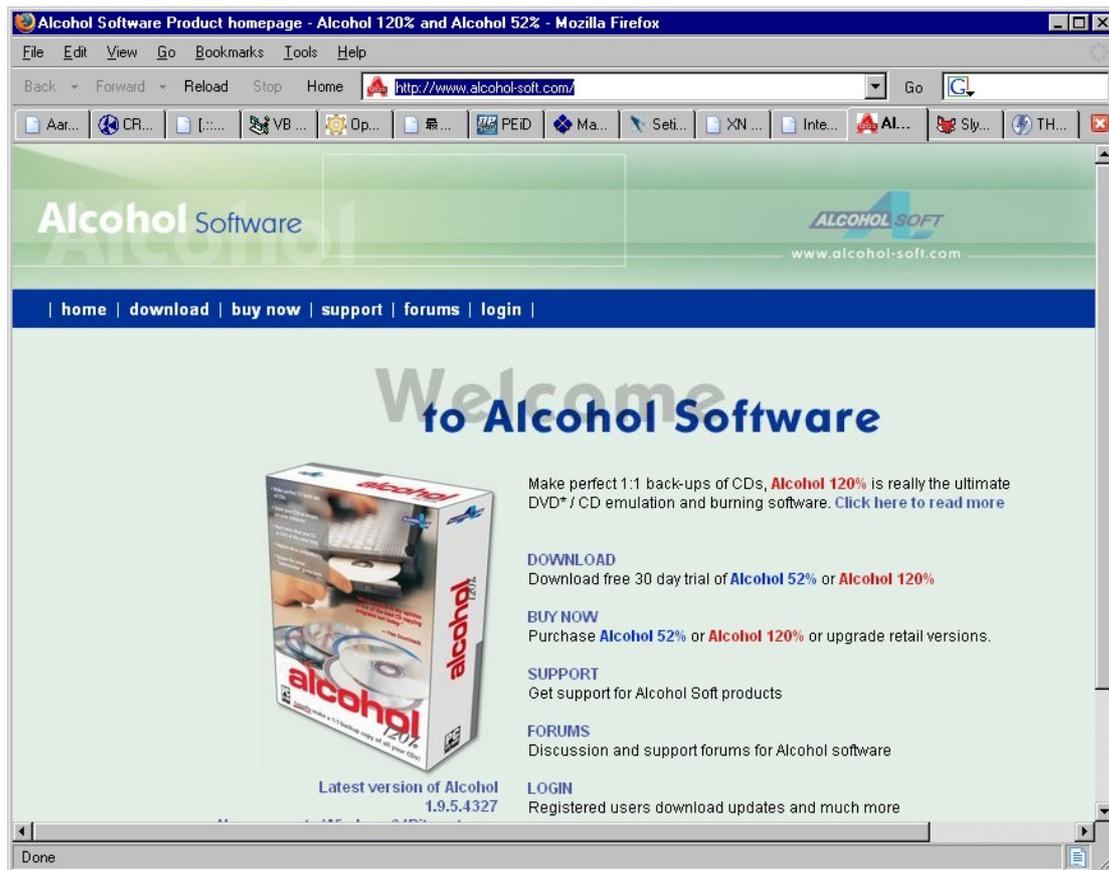


Рисунок 32 официальный сайт копировщика защищенных дисков Alcohol 120%

Пара лучших коммерческих копировщиков это, бесспорно, Alcohol 120% (www.alcohol-soft.com) и CloneCD (www.slysoft.com/en/clonedcd.html). Бесплатная утилита Daemon Tools (www.daemon-tools.cc) позволяет монтировать образы, снятые двумя этими копировщиками как виртуальные диски, образ которых лежит на HDD. Очень удобно!

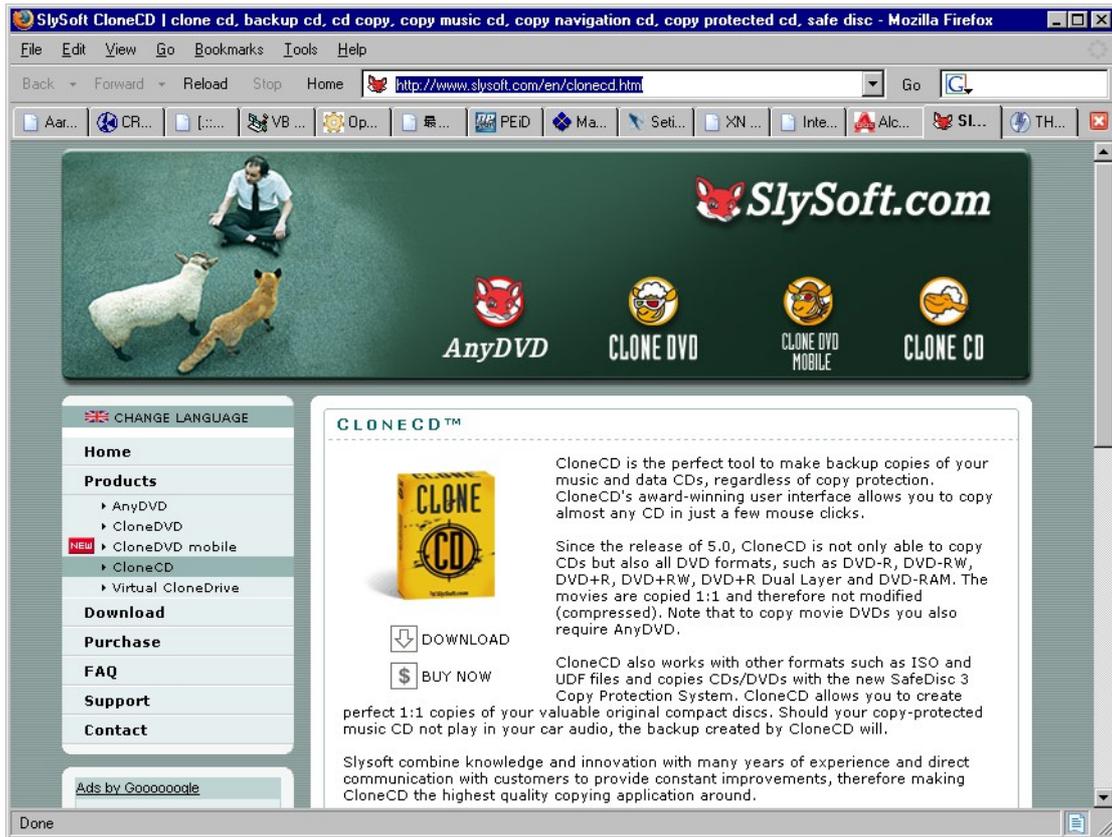


Рисунок 33 официальный сайт копировщика защищенных дисков CloneCD