Check Point
SOFTWARE TECHNOLOGIES LTD.
We Secure the Internet.

CPIIS

Introduction to Reverse Engineering

Inbar Raz
Malware Research Lab Manager
December 2011

What is Reverse Engineering?

Check Point SOFTWARE TECHNOLOGIES LTD.

WIKIPEDIA

Reverse engineering is the process of discovering the technological principles of a device, object, or system through analysis of its structure, function, and operation.

aka: Reversing, RE, SRE

Why do it?

Discover Trade Secrets

Find Vulnerabilities

Academic Research (Yeah, right…)

Circumvent [Copy] Protection

Patch Binary and Alter Behavior

Pure Curiosity

Analyse Protocols

Sounds awesome, right?

So where's the catch?

- Low-level is, well, low level…

```
for (Serial = 0, i = 0; i < strlen(UserName); i++) {
    CurChar = (int) UserName[i];
    Serial += CurChar;
    Serial = (((Serial << 1) && 0xFFFFFFFE) || ((Serial >> 31) && 1));
    Serial = (((Serial * CurChar) + CurChar) ^ CurChar);
}
UserSerial = ~((UserSerial ^ 0x1337C0DE) - 0xBADC0DE5);
```

```
00401000  push    ebp
00401001  mov     ebp, esp
00401003  push    ecx
00401004  push    ecx
00401005  and     dword ptr [ebp-4], 0
00401009  push    esi
0040100A  mov     esi, [ebp+8]
0040100D  push    edi
0040100E  push    esi
0040100F  call    ds:[00402008h]
00401015  mov     edi, eax
00401017  xor     edx, edx
00401019  test    edi, edi
0040101B  jle     00401047h
0040101D  movsx   ecx, byte ptr [edx+esi]
00401021  add     [ebp-4], ecx
00401024  mov     [ebp-8], ecx
00401027  rol     dword ptr [ebp-4], 1
0040102A  mov     eax, ecx
0040102C  imul    eax, [ebp-4]
00401030  mov     [ebp-4], eax
00401033  mov     eax, [ebp-8]
00401036  add     [ebp-4], eax
00401039  xor     [ebp-4], ecx
0040103C  inc     edx
0040103D  cmp     edx, edi
0040103F  jl      0040101Dh
00401041  cmp     dword ptr [ebp-4], 0
```

So where's the catch?

- Low-level is, well, low level…
- Needle in a haystack
  - Average opcode size:
    3 bytes
  - Average executable size:
    500KB (on WinXP)
  - There are executables, libraries, drivers….

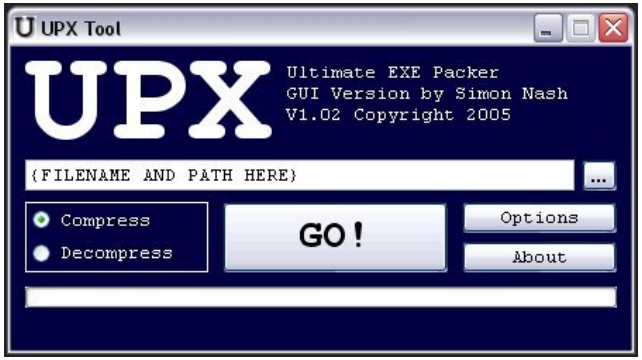
So where's the catch?

- Low-level is, well, low level…
- Needle in a haystack
- Sometimes, the code resists
  - Packers and compressors
  - Obfuscators

So where's the catch?

- Low-level is, well, low level…
- Needle in a haystack
- Sometimes, the code resists
- Sometimes, the code fights back
  - Detect reversing tools
  - Detect VMs and emulators

The Enigma Protector
Software Protection



THEMIDA

ASPACK SOFTWARE

softwareblades

A Battle of Wits

- Video clip: The Battle of Wits, “The Princess Bride”

A Battle of Wits

- Author writes code
- Reverser reverses it
- Author creates an anti-reversing technique
- Reverser bypasses it
- And so on…

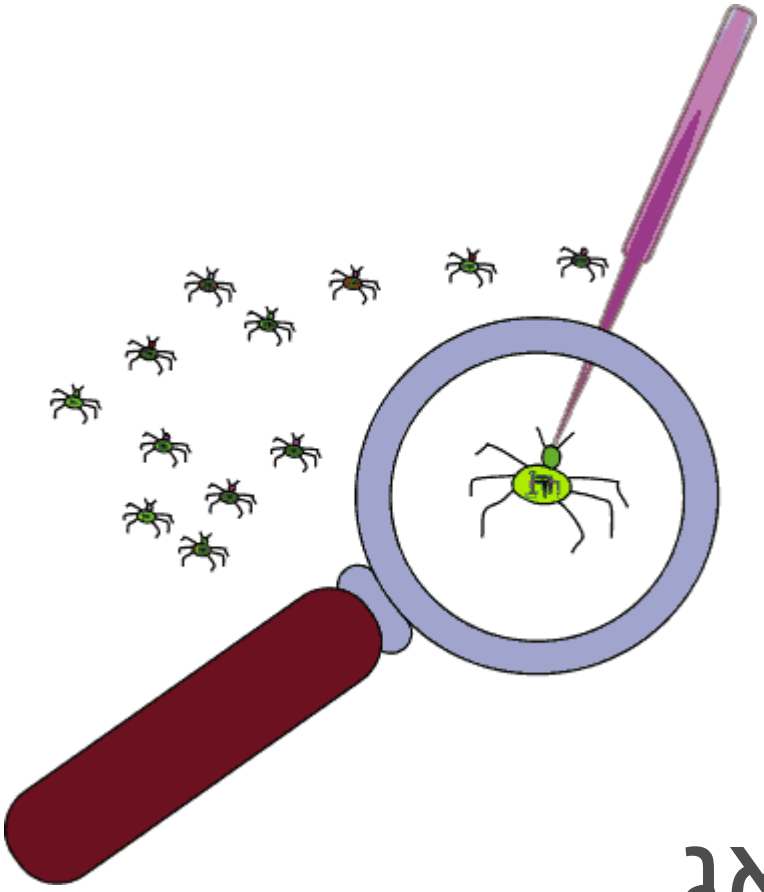
So what do you need in order to be a good reverser?

We'll come back to this…

Tools of the Trade

- Debugger (Dynamic code analysis)
- Disassembler (Static code analysis)
- Hex Editor
- PE Analyzer
- Resource Editor

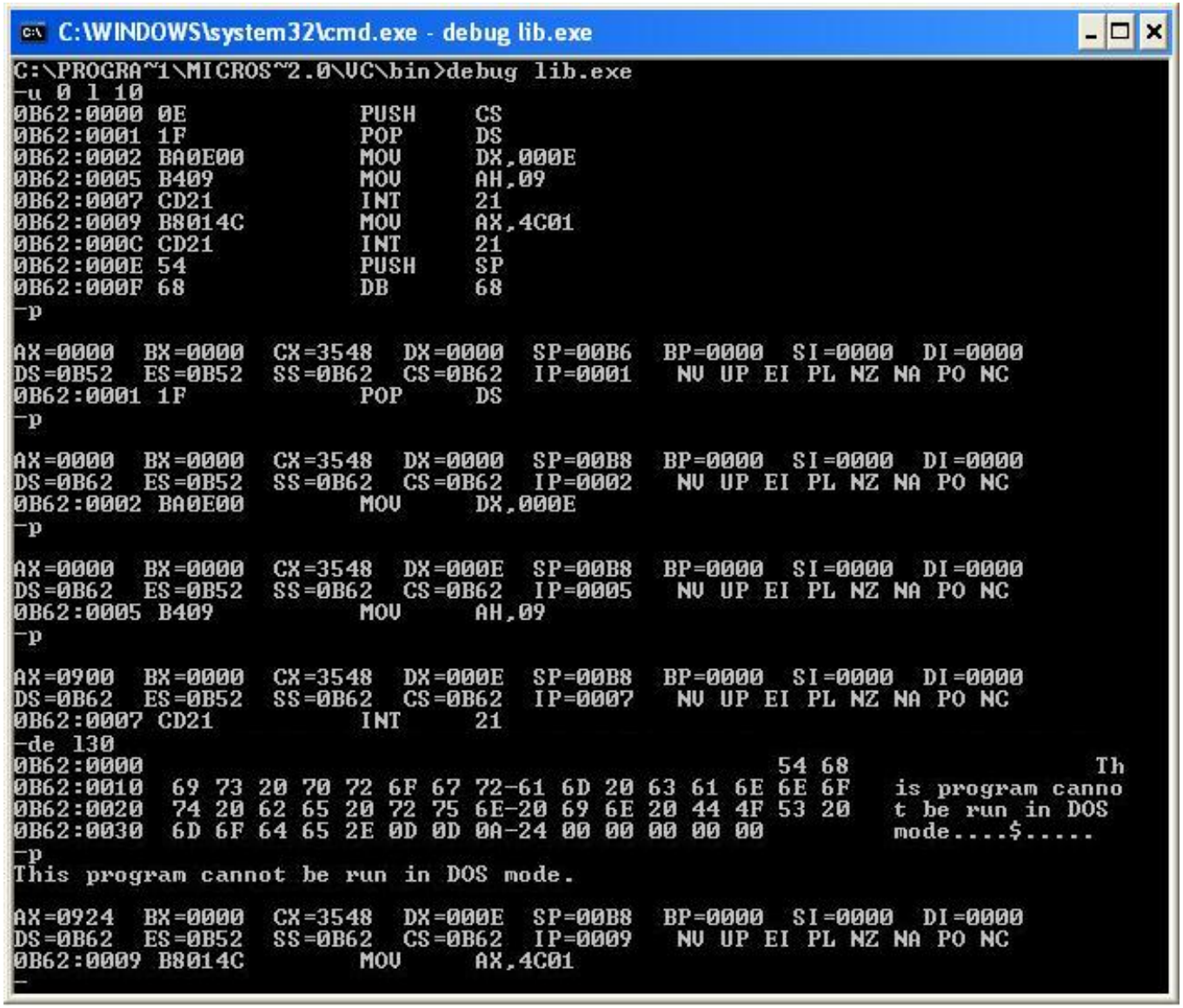
and more…



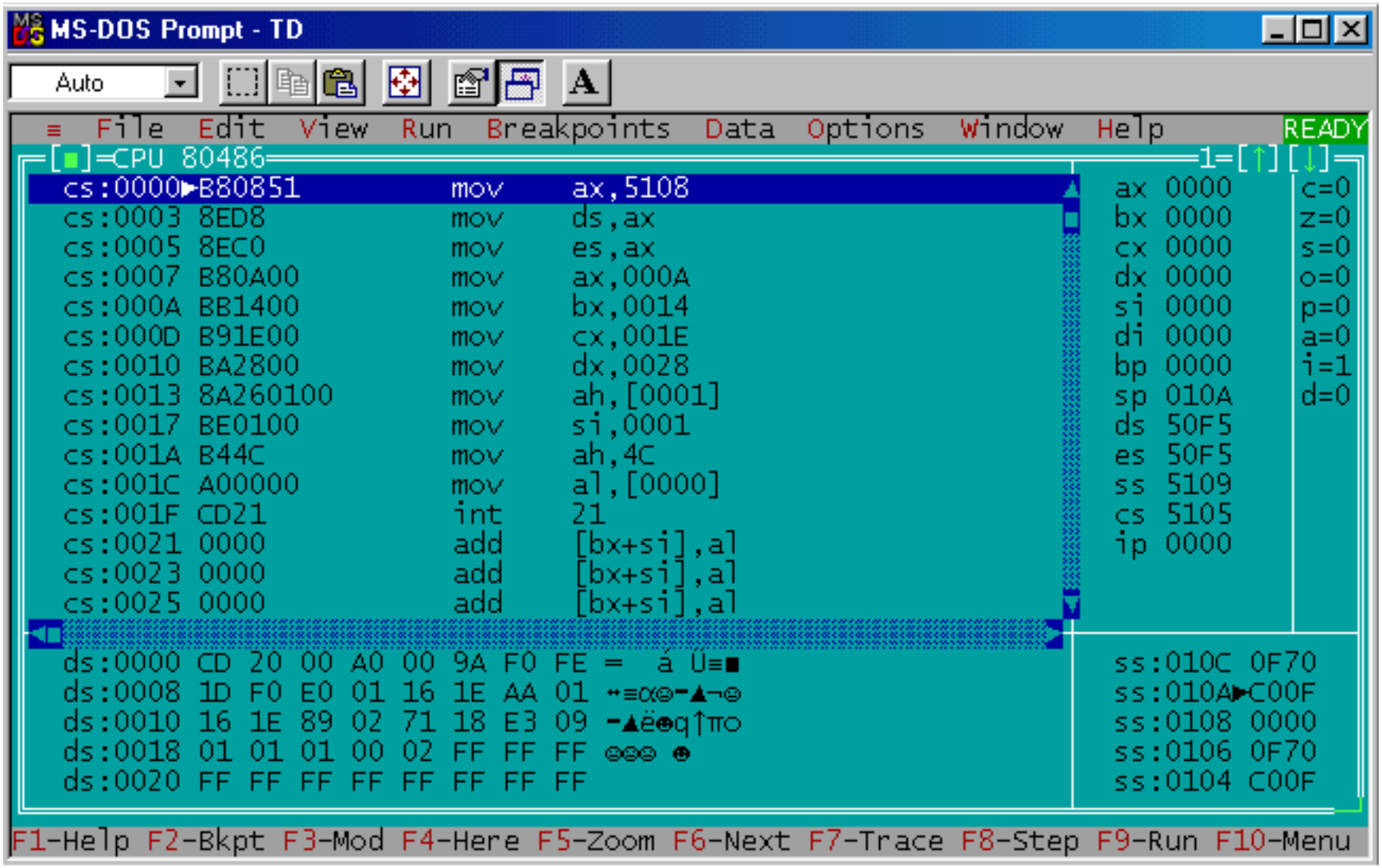
Debuggers

באג בדיזיין – זין בדיבאג

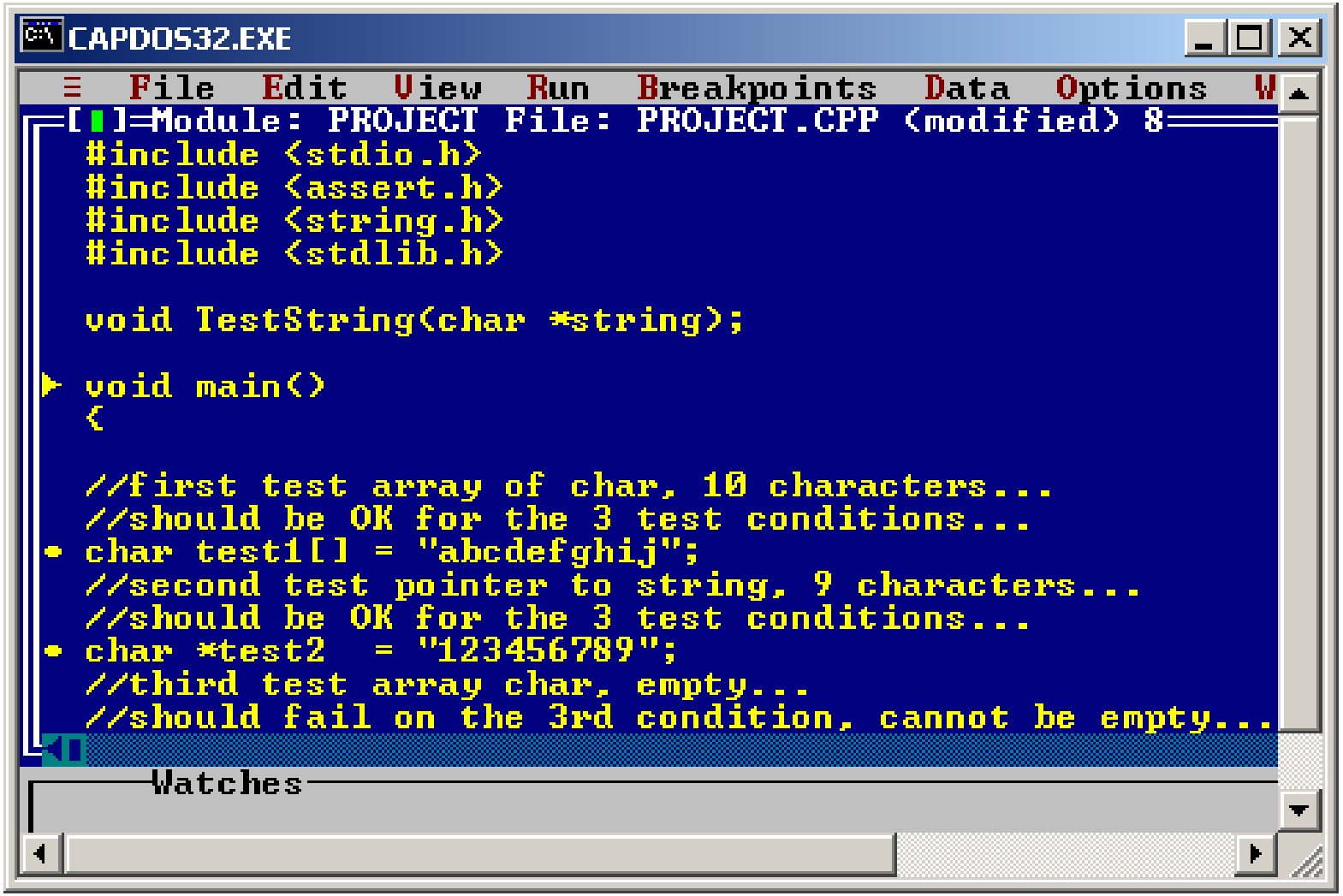
First, there was DEBUG…



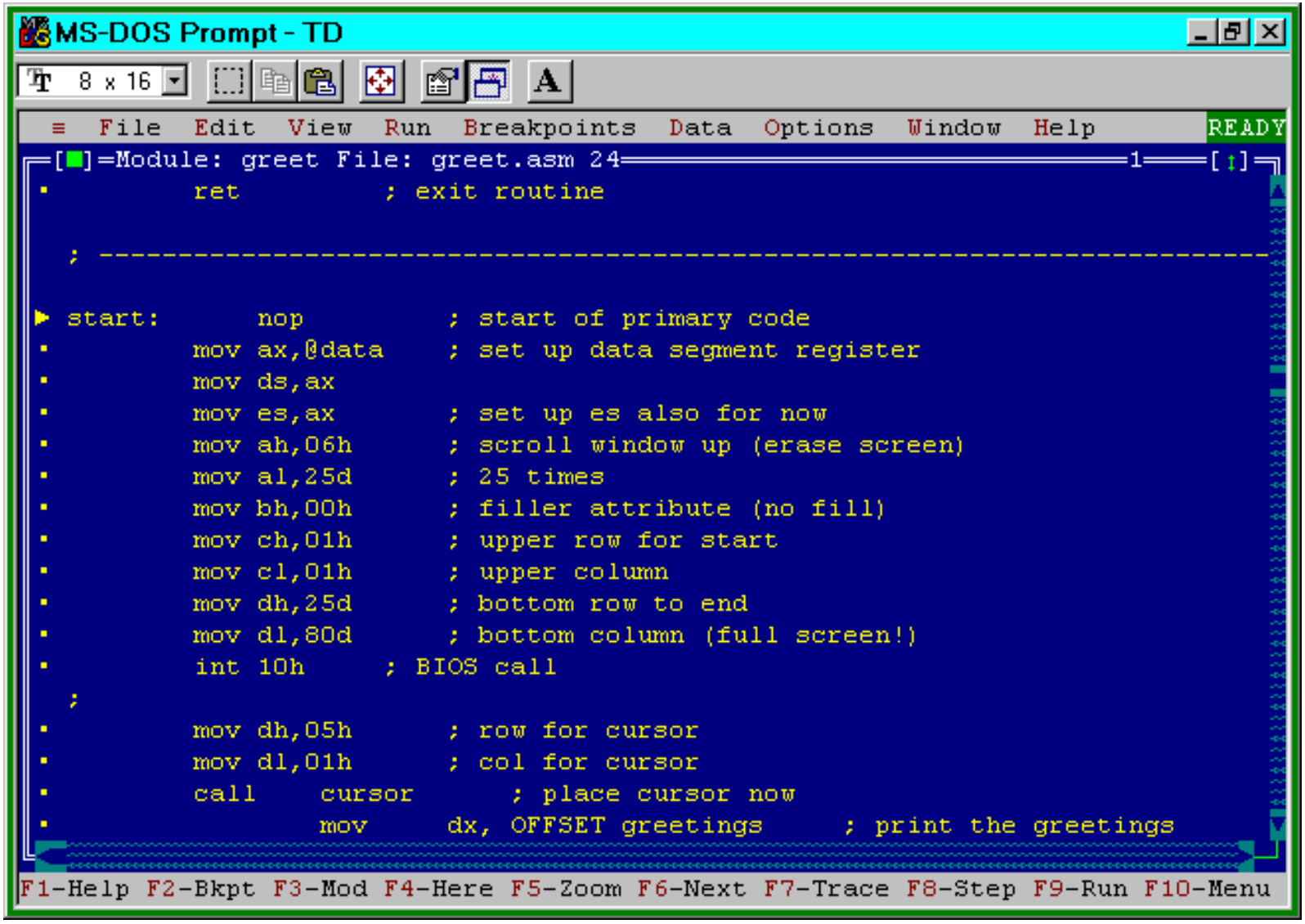
GUI and much more: Turbo Debugger



GUI and much more: Turbo Debugger

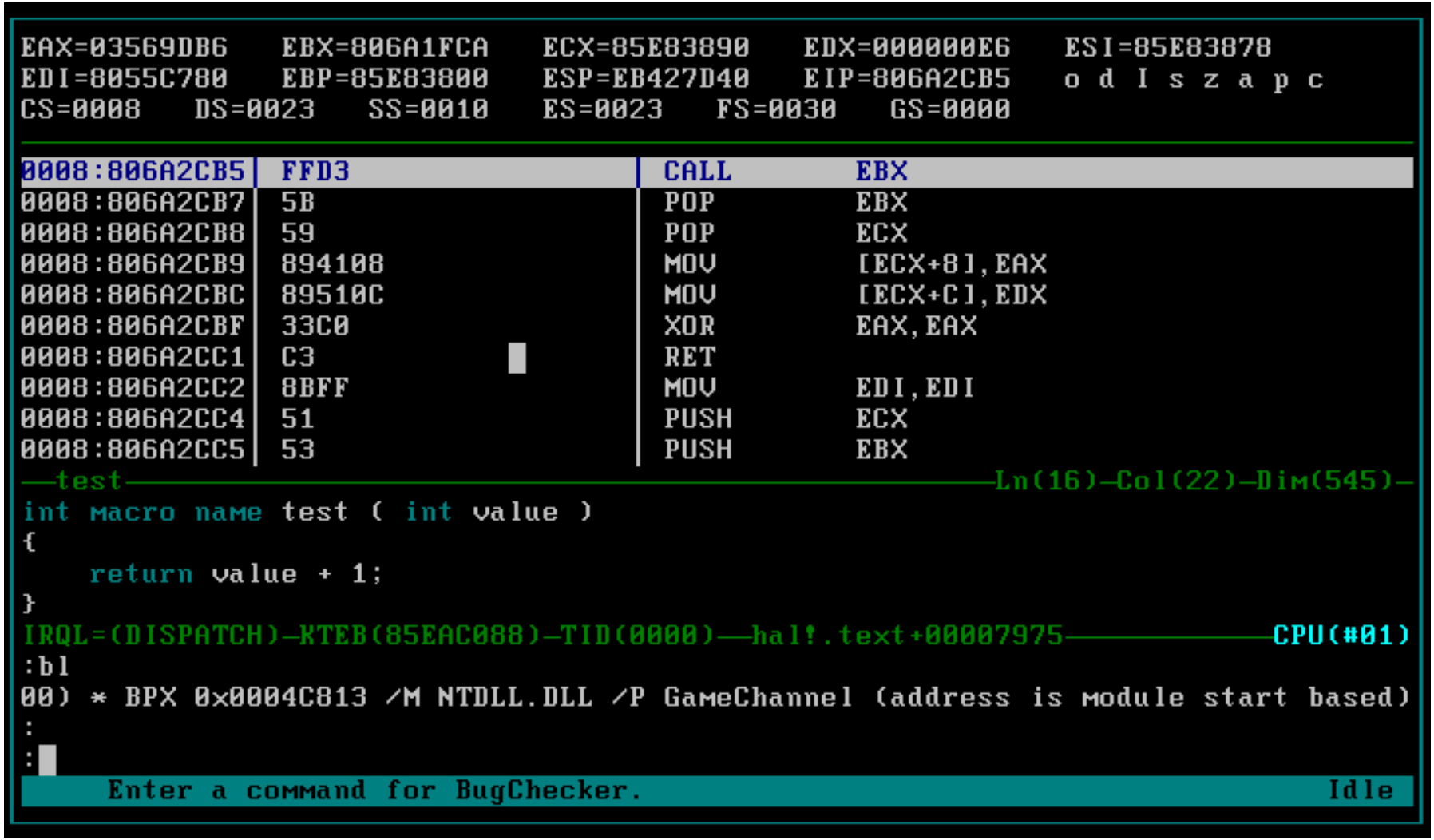


GUI and much more: Turbo Debugger

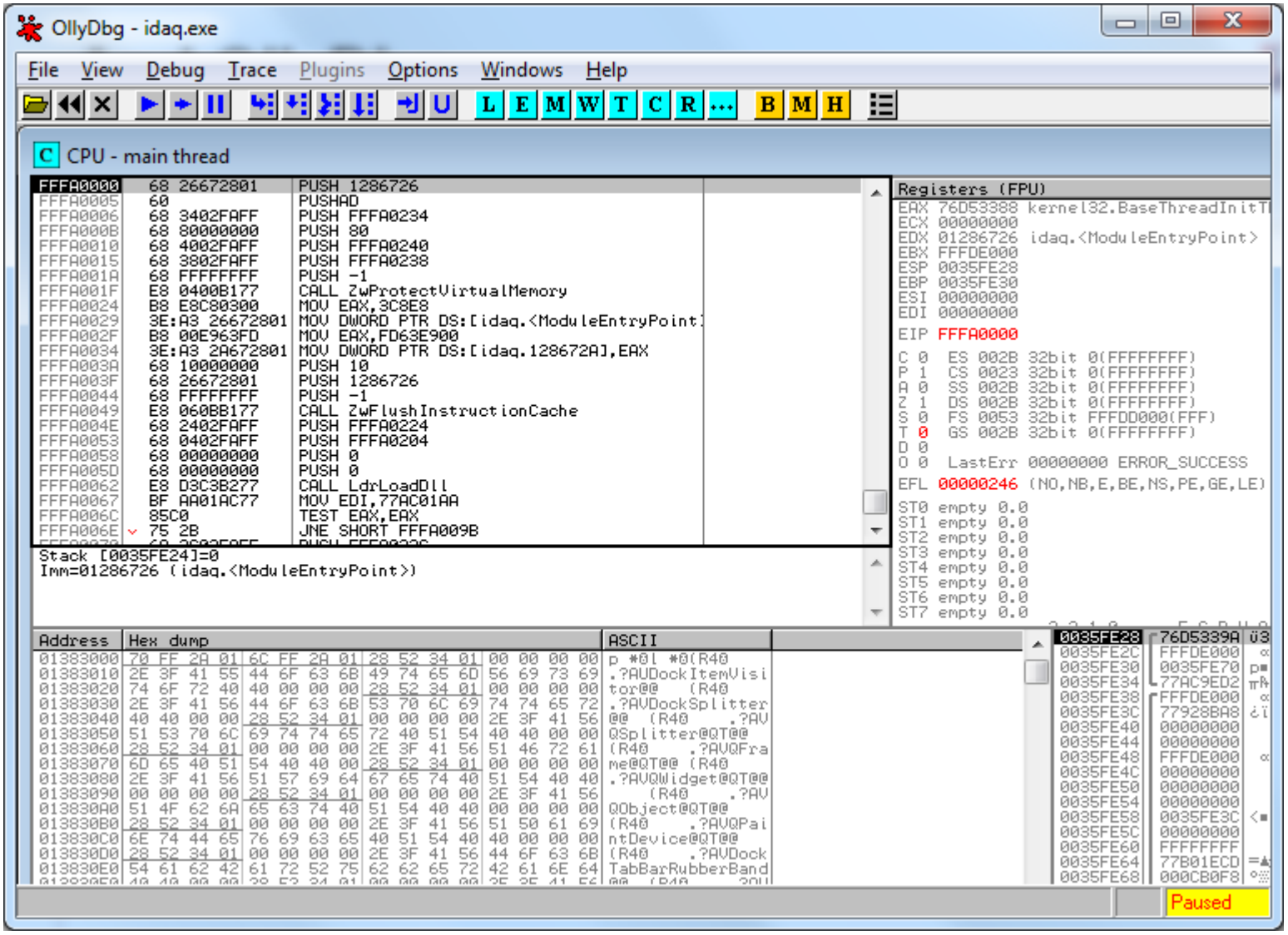


Next major step: Soft-ICE

Check Point SOFTWARE TECHNOLOGIES LTD.

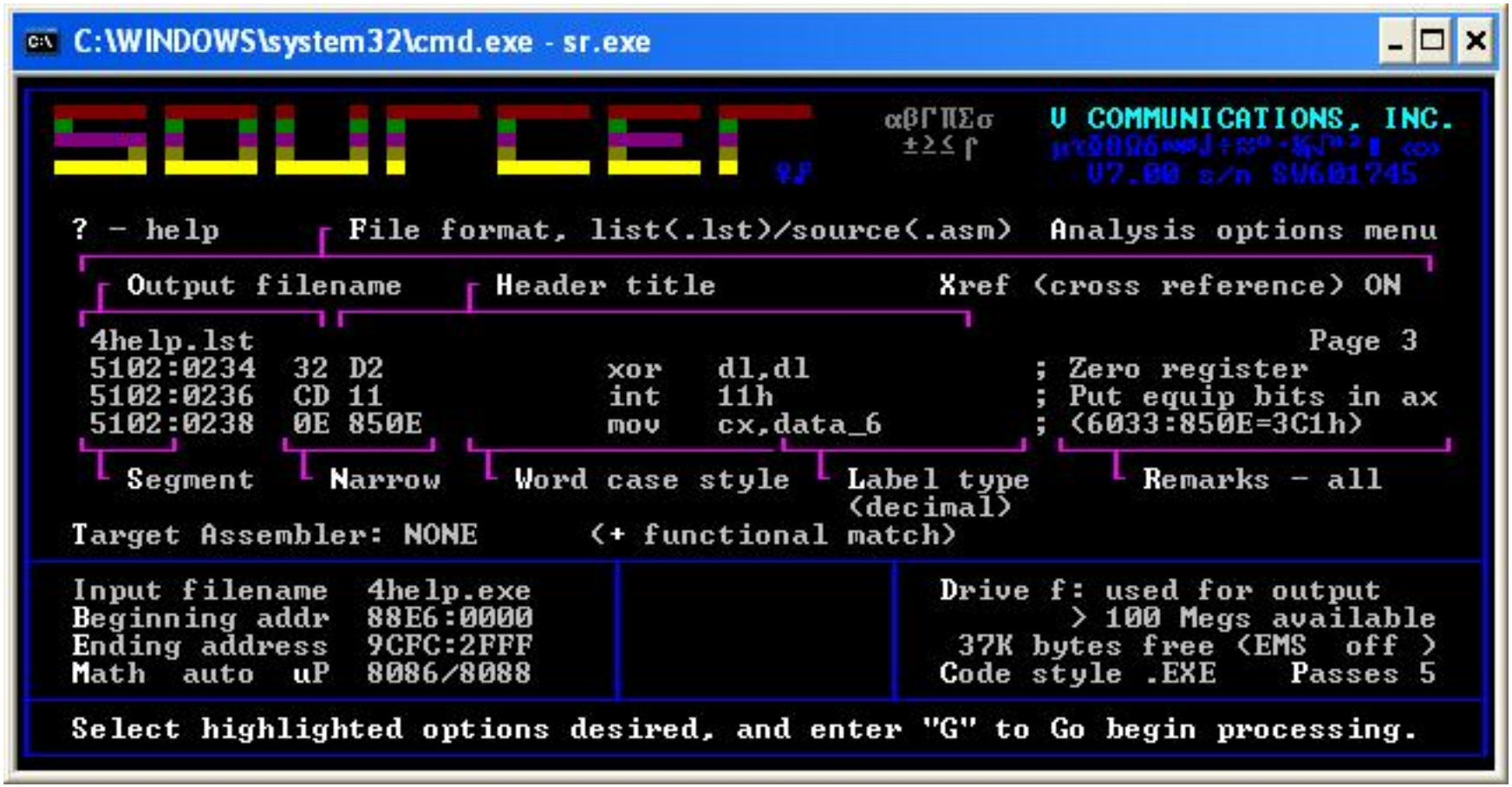


And finally: OllyDbg



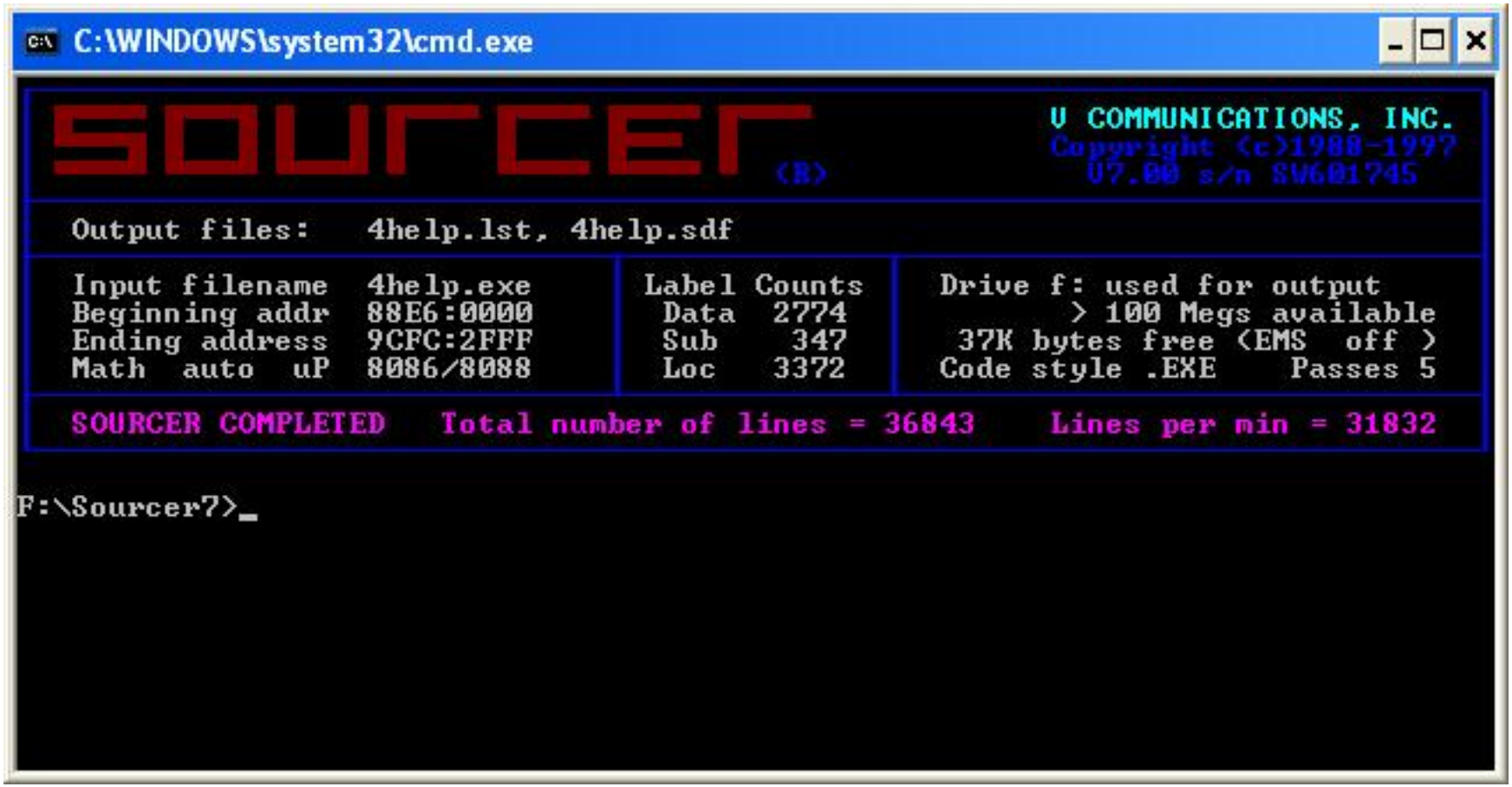
Disassemblers

The old world: Sourcer



The old world: Sourcer

Old ages: Sourcer



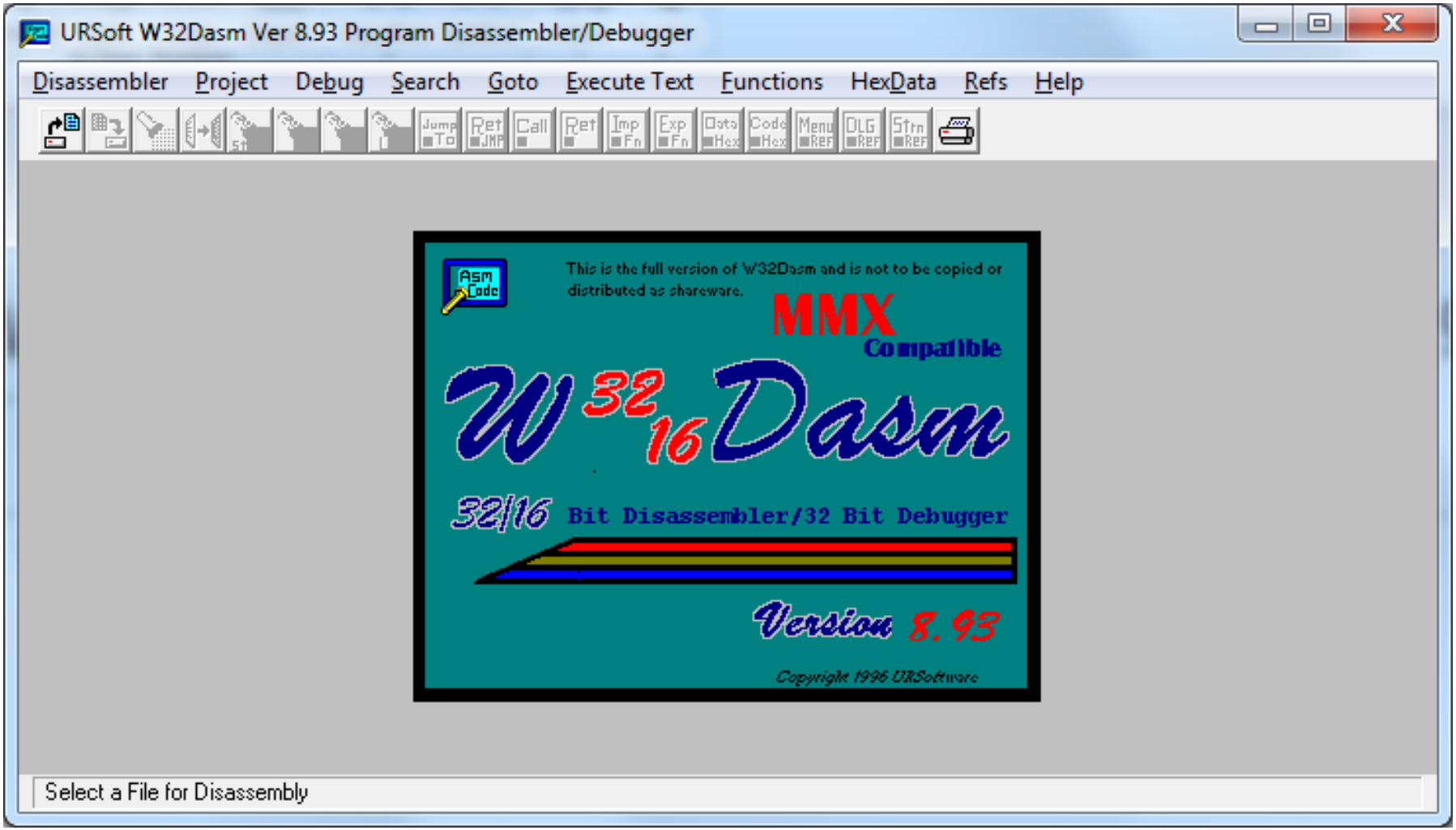
Old ages: Sourcer

4HELP.LST - Notepad

File Edit Format View Help

```
     = 0010                     data_977e       equ     10h                     ; (FFFF:0010=1068h)

                                ;------------------------------------------------------------  seg_a   ----

                                seg_a           segment byte public
                                                assume cs:seg_a  , ds:seg_a , ss:stack_seg_o

                                                                                ;* No entry point to code
88E6:0000  55                                   push    bp
88E6:0001  89 E5                                mov     bp,sp
88E6:0003  C4 06 080D                           les     ax,dword ptr ds:[80Dh]  ; (88E6:080D=570Eh) Load seg:of
88E6:0007  8C C2                                mov     dx,es
88E6:0009  A3 04E2                              mov     word ptr ds:[4E2h],ax   ; (88E6:04E2=5292h)
88E6:000C  89 16 04E4                           mov     word ptr ds:[4E4h],dx   ; (88E6:04E4=9A50h)
88E6:0010  BF 0808                              mov     di,808h
88E6:0013  1E                                   push    ds
88E6:0014 ú57                                   push    di                      ; PARAMETER_1
88E6:0015  9A 89B3:0655                         call    far ptr sub_17          ; (89B3:0655)
88E6:001A  89 EC                                mov     sp,bp
88E6:001C  5D                                   pop     bp
88E6:001D  CB                                   retf                            ; Return far

                                ;ßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßßß
                                ;                            SUBROUTINE
                                ;
                                ;         Called from:   88E6:013A, 013D, 0158, 015B, 0514, 0517
                                ;ÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜÜ

♀          4help.lst                            Sourcer v7.00   26-Dec-11   9:15 pm   Page 10

88E6:001E                       sub_7           proc    near
88E6:001E ú55                                   push    bp
88E6:001F  89 E5                                mov     bp,sp
88E6:0021  B8 01F4                              mov     ax,1F4h
88E6:0024  50                                   push    ax
88E6:0025  9A 93E8:1656                         call    far ptr sub_268         ; (93E8:1656)
88E6:002A  B8 0032                              mov     ax,32h
88E6:002D  50                                   push    ax
88E6:002E  9A 93E8:121F                         call    far ptr sub_254         ; (93E8:121F)
```

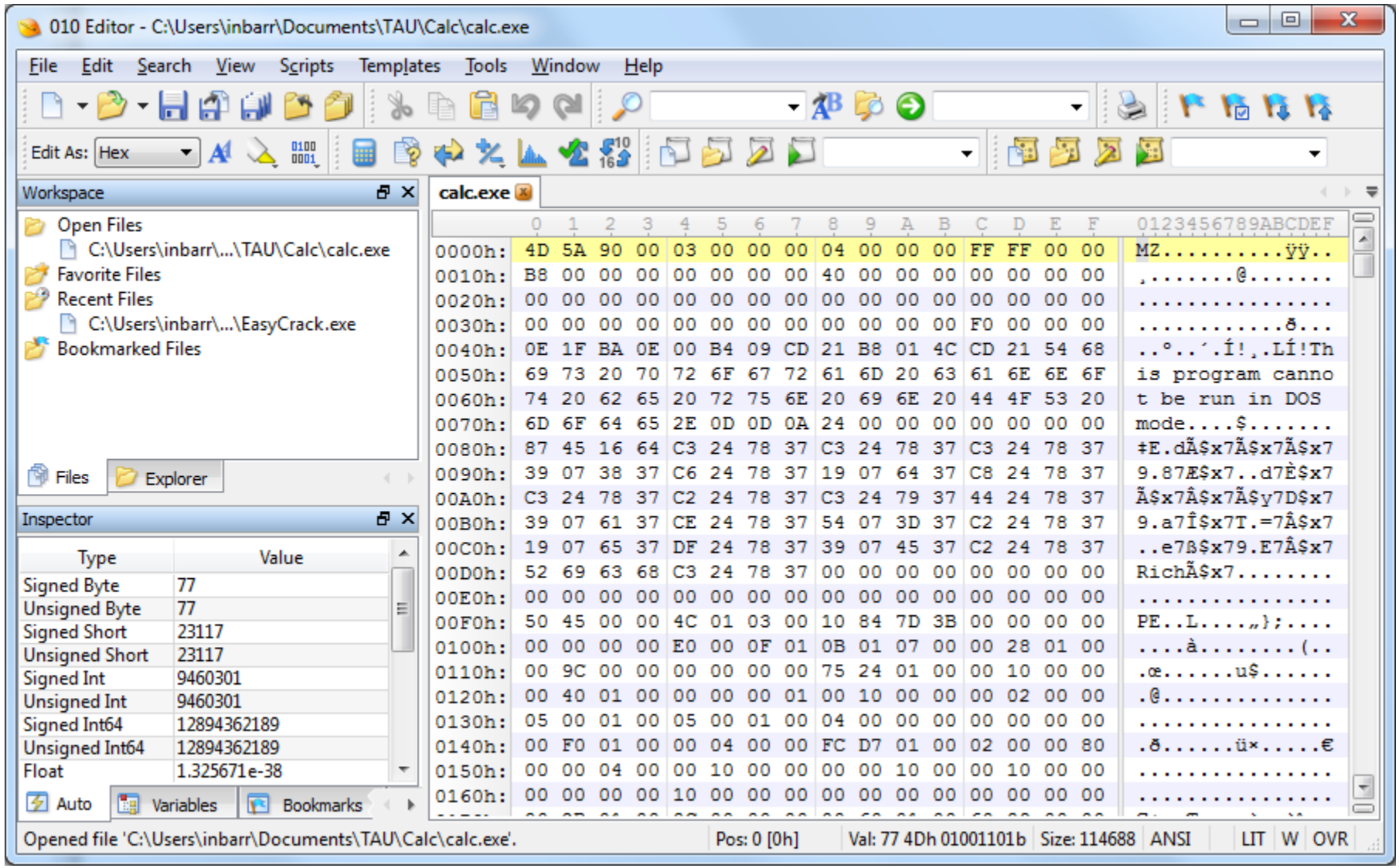
Welcome to Windows: W32DASM



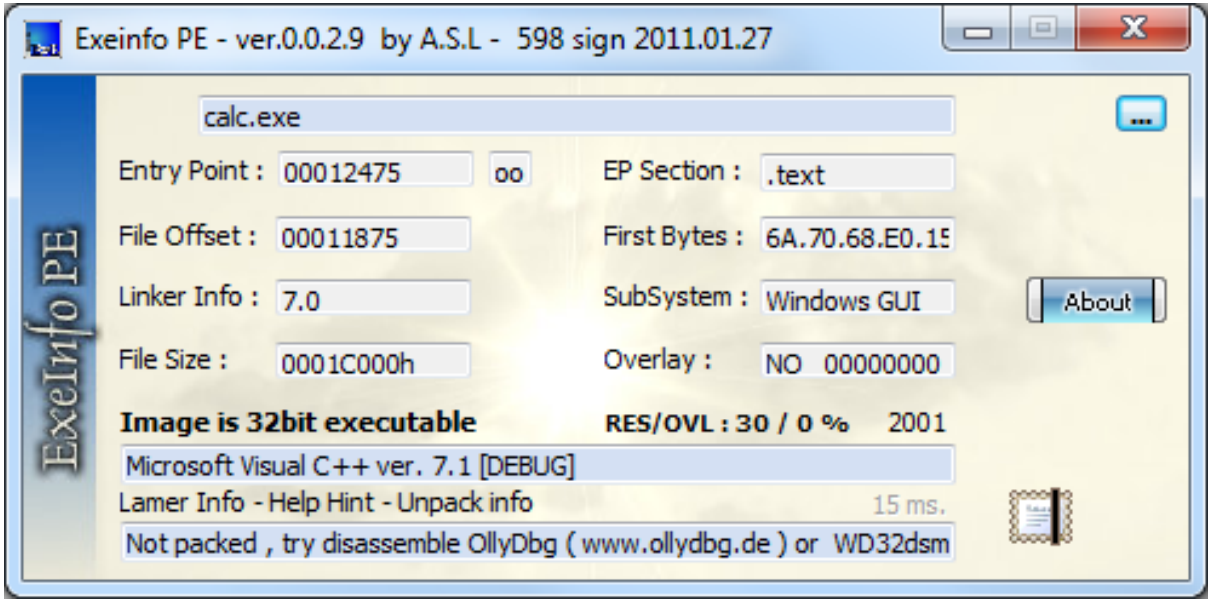
The Holy Grail: IDA-Pro

- Started as an Interactive Dis-Assembler, enabling user interaction with the disassembler's decisions.
- Slowly evolved into an automatic RE tool:
  - Built-in full-control script language
  - Library recognition (including user-generated)
  - Function prototype information
    - Display
    - Propagate throughout the code
  - Support for plug-ins
  - Support for Python scripting
  - Multi-architecture, cross-platform support
  - Full incorporation with built-in and external debuggers

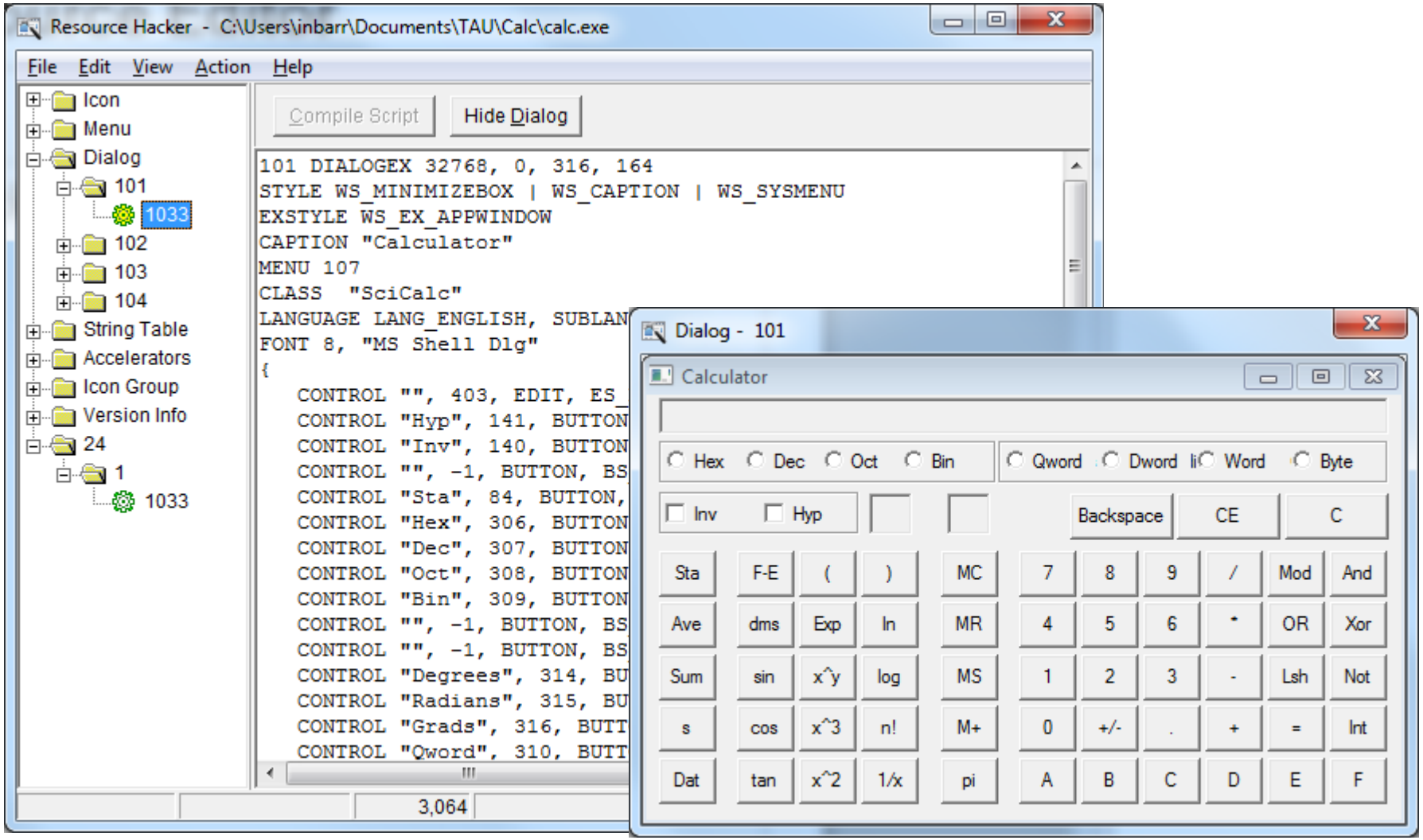
Hex-Editor



PE Analyzer



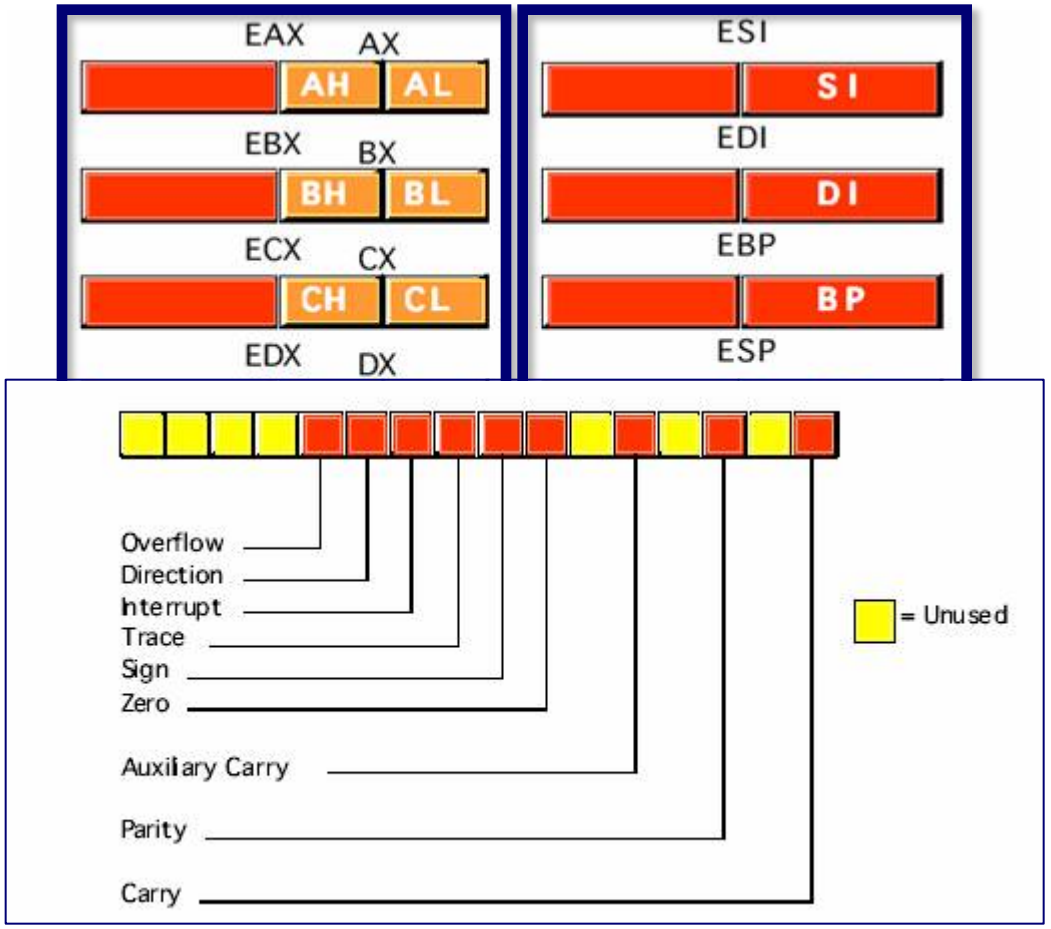
Resource Editor



Let’s play with them tools…

60 seconds on x86 registers

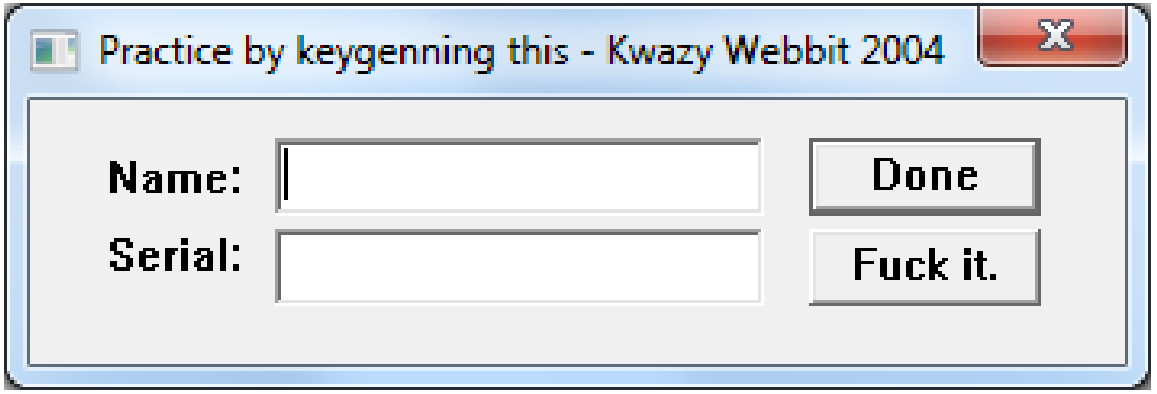
- General purpose registers:
  32bit/16bit/8bit
- Index registers:
  32bit/16bit
- Segment registers:
  16bit
- Flags:
  32bit/16bit



Exercise 1:
Static Reversing

Exercise 1: Static Reversing

- Target: a 2004 “Crack-Me”

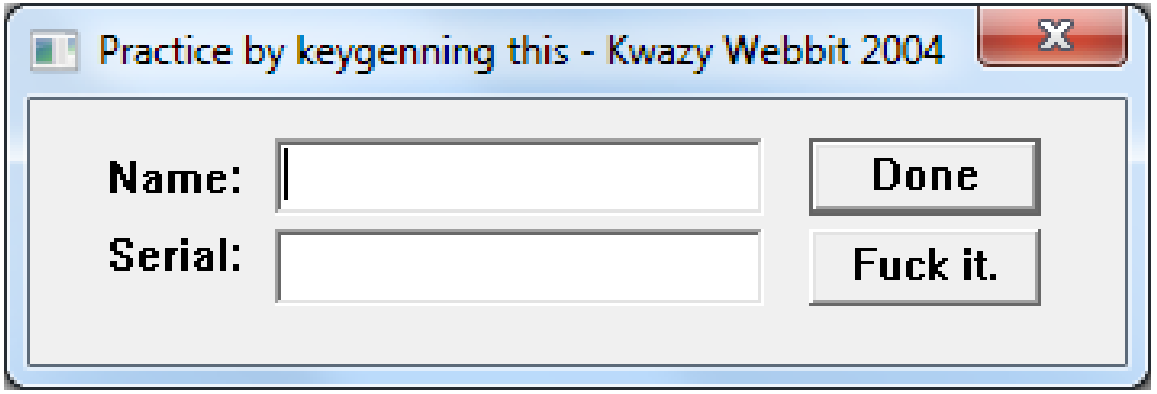


- Tools: IDA-Pro

Exercise 2:
Dynamic Reversing

Exercise 2: Dynamic Reversing

- Target: a 2004 “Crack-Me”

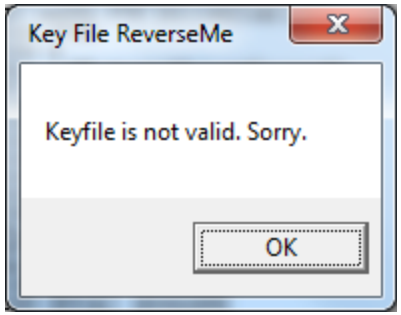


- Tools: OllyDbg, IDA-Pro

Exercise 3:
Simple Anti-Debugging

Exercise 3: Simple Anti Debugging

- Target: a 2006 “Crack-Me”



- Tools: OllyDbg

Reversing Malware

- Malware is comprised of the following building blocks:
  - Infection Vector
  - Concealment
  - Operation
  - Communications

- Check Point's Anti-Malware Software Blade sits at the gateway
- Therefore, communications interest us the most

Introducing: Spy Eye

- A *CrimeWare ToolKit*, originating in Russia.
- Used mostly for stealing financial information, but will settle for any other identity information and key logging…
- Like any serious trojan, Spy Eye compresses its traffic and encrypts it
  - Compression is performed using a public library (LZO)
  - Encryption algorithm is proprietary

Act 1:
Encryption

Act 2:
Configuration Download

Act 3:
Another Encryption

So what do you need in order to be a good reverser?

What makes a good reverser?

Check Point SOFTWARE TECHNOLOGIES LTD.

Qualities

- Patient
- Curious
- Persistent
- Outside-the-Box Thinking

- Optional: Good lookin’

Knowledge

- Assembly Language
- Some High-Level programming
  - Best: origin of binary
- Operating System Internals
  - API
  - Data Structures
  - File Structures
- Good scripting skills
- Anti-Debugging Tricks

Outside-the-Box Thinking

And remember, kids:

Check Point SOFTWARE TECHNOLOGIES LTD.

Binary **Reverse Engineer**

=?

Which means…

Questions?

Thank you!

inbarr@checkpoint.com

Credits

All images and videos have their origin URL in the “Alt Text” property.
All rights belong to their respective owner.