Armored Malware

Headache for malware analyst

Definition

- A malware designed to be very difficult to reverse engineer and analyse.
- A malware that contains a variety of mechanisms specifically coded to make its detection and decryption very difficult.
- Mostly this technique are deal with reverse engineer when reversing malware.
- Basic static and dynamic analysis doesn’t need this topic.

Why armor?

- Fooling anti-virus software
- Fooling malware analyst
  - Make it confuse, complicated, difficult to analyse

Armored Malware

Armor Features

- Encryption
- Packer
- Cryptor
- Protector
- Compression
- Obfuscation

- Anti Debugging
- Anti Patching
- Anti Tracing
- Anti Unpacking
- Anti Vmware
- Password protected
- Many more…

Packers

- Origins
  - Compression
    - Save space
    - Bandwitdth reduction
- Malware use
  - Bypass AV signatures, avoid detection
  - Prevent reverse engineering

Packers

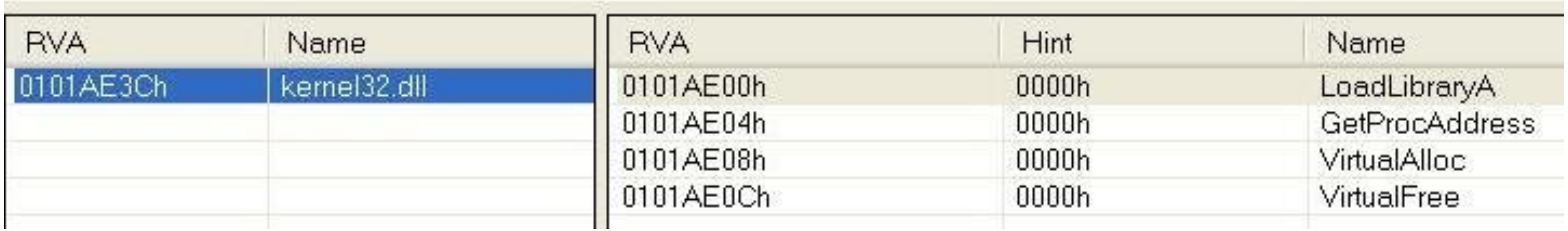
- UPack
- Mew
- UPX
- Packman
- EZIP
- PE-PaCK
- FSG
- Dropper
- Cexe
- PE Diminisher
- PECrypt32
- PESpin
- NSPack
- PEBundle
- PECompact
- Many more…

Side effects of Packing

- No strings
- Few imports result
  - Kernel32.dll
    - LoadLibrary
    - GetProcAddress
    - VirtualAlloc
    - VirtualFree
- High entropy sections
  - Marked as code / executable
  - Large difference in Virtual size of section vs. real size.
- Fewer sections

Side effects – Imports

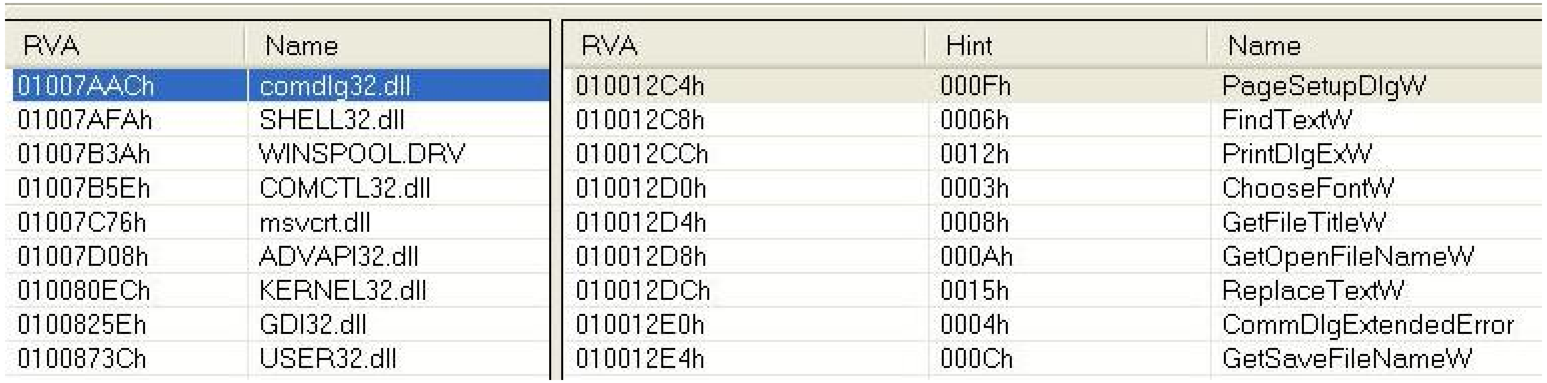
- Packed



RVA	Name
0101AE3Ch	kernel32.dll

RVA	Hint	Name
0101AE00h	0000h	LoadLibraryA
0101AE04h	0000h	GetProcAddress
0101AE08h	0000h	VirtualAlloc
0101AE0Ch	0000h	VirtualFree

- Unpacked

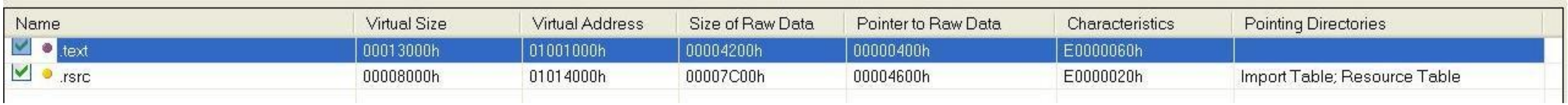


RVA	Name
01007AACh	comdlg32.dll
01007AFAh	SHELL32.dll
01007B3Ah	WINSPOOL.DRV
01007B5Eh	COMCTL32.dll
01007C76h	msvcrt.dll
01007D08h	ADVAPI32.dll
010080ECh	KERNEL32.dll
0100825Eh	GDI32.dll
0100873Ch	USER32.dll

RVA	Hint	Name
010012C4h	000Fh	PageSetupDlgW
010012C8h	0006h	FindTextW
010012CCh	0012h	PrintDlgExW
010012D0h	0003h	ChooseFontW
010012D4h	0008h	GetFileTitleW
010012D8h	000Ah	GetOpenFileNameW
010012DCh	0015h	ReplaceTextW
010012E0h	0004h	CommDlgExtendedError
010012E4h	000Ch	GetSaveFileNameW

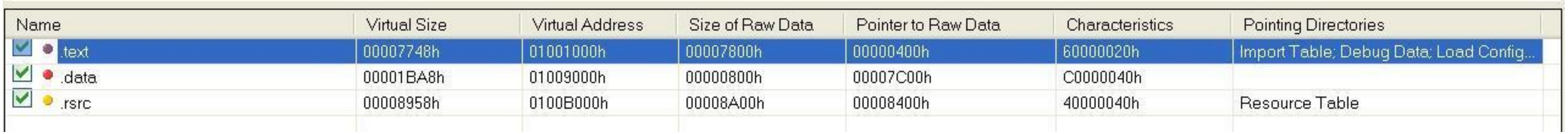
Side effects of Packing – Section Size and Entropy

- Packed



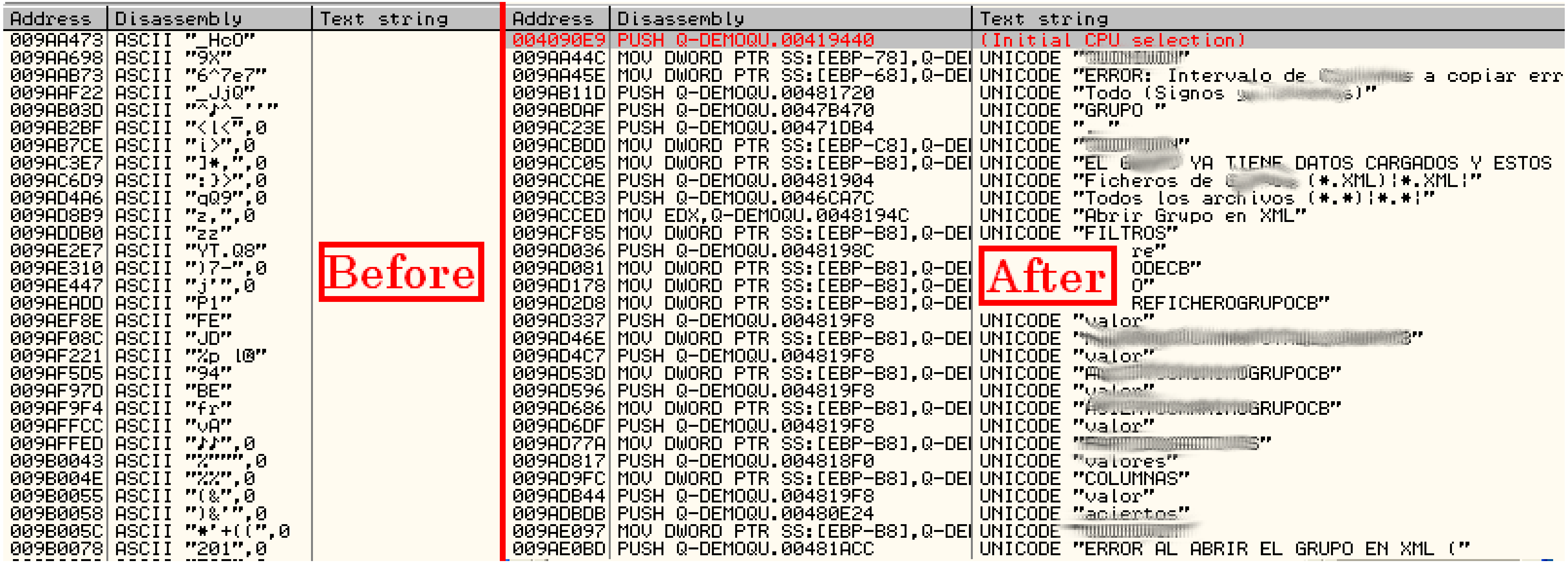
Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
☑ ● .text	00013000h	01001000h	00004200h	00000400h	E0000060h	
☑ ● .rsrc	00008000h	01014000h	00007C00h	00004600h	E0000020h	Import Table; Resource Table

- Unpacked



Name	Virtual Size	Virtual Address	Size of Raw Data	Pointer to Raw Data	Characteristics	Pointing Directories
☑ ● .text	00007748h	01001000h	00007800h	00000400h	60000020h	Import Table; Debug Data; Load Config...
☑ ● .data	00001BA8h	01009000h	00000800h	00007C00h	C0000040h	
☑ ● .rsrc	00008958h	0100B000h	00008A00h	00008400h	40000040h	Resource Table

Strings on Packed binary



```
Address  Disassembly        Text string
009AA473 ASCII "_HcO"
009AA698 ASCII "9X"
009AAB73 ASCII "6^7e7"
009AAF22 ASCII "_JjQ"
009AB03D ASCII "^♪^_'''
009AB2BF ASCII "<l<",0
009AB7CE ASCII "i>",0
009AC3E7 ASCII "]*,",0
009AC6D9 ASCII ":}>",0
009AD4A6 ASCII "qQ9",0
009AD8B9 ASCII "z,",0
009ADDB0 ASCII "zz"
009AE2E7 ASCII "YT.Q8"
009AE310 ASCII ")7-",0
009AE447 ASCII "j'",0
009AEADD ASCII "P1"
009AEF8E ASCII "FE"
009AF08C ASCII "JD"
009AF221 ASCII "%p l@"
009AF5D5 ASCII "94"
009AF97D ASCII "BE"
009AF9F4 ASCII "fr"
009AFFCC ASCII "vA"
009AFFED ASCII "♪♪",0
009B0043 ASCII "%""",0
009B004E ASCII "%%",0
009B0055 ASCII "(&",0
009B0058 ASCII ")&'",0
009B005C ASCII "*'+((",0
009B0078 ASCII "201",0
```

```
Address  Disassembly                          Text string
004090E9 PUSH Q-DEMOQU.00419440               (Initial CPU selection)
009AA44C MOV DWORD PTR SS:[EBP-78],Q-DEI      UNICODE "[redacted]"
009AA45E MOV DWORD PTR SS:[EBP-68],Q-DEI      UNICODE "ERROR: Intervalo de [redacted] a copiar err
009AB11D PUSH Q-DEMOQU.00481720               UNICODE "Todo (Signos [redacted])"
009ABDAF PUSH Q-DEMOQU.0047B470               UNICODE "GRUPO "
009AC23E PUSH Q-DEMOQU.00471DB4               UNICODE ". "
009ACBDD MOV DWORD PTR SS:[EBP-C8],Q-DEI      UNICODE "[redacted]"
009ACC05 MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "EL [redacted] YA TIENE DATOS CARGADOS Y ESTOS
009ACCAE PUSH Q-DEMOQU.00481904               UNICODE "Ficheros de [redacted] (*.XML)|*.XML|"
009ACCB3 PUSH Q-DEMOQU.0046CA7C               UNICODE "Todos los archivos (*.*)|*.*|"
009ACCED MOV EDX,Q-DEMOQU.0048194C            UNICODE "Abrir Grupo en XML"
009ACF85 MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "FILTROS"
009AD036 PUSH Q-DEMOQU.0048198C                       re"
009AD081 MOV DWORD PTR SS:[EBP-B8],Q-DEI              ODECB"
009AD178 MOV DWORD PTR SS:[EBP-B8],Q-DEI              O"
009AD2D8 MOV DWORD PTR SS:[EBP-B8],Q-DEI              REFICHEROGRUPOCB"
009AD337 PUSH Q-DEMOQU.004819F8               UNICODE "valor"
009AD46E MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "[redacted]"
009AD4C7 PUSH Q-DEMOQU.004819F8               UNICODE "valor"
009AD53D MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "[redacted]GRUPOCB"
009AD596 PUSH Q-DEMOQU.004819F8               UNICODE "[redacted]"
009AD686 MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "[redacted]GRUPOCB"
009AD6DF PUSH Q-DEMOQU.004819F8               UNICODE "valor"
009AD77A MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "[redacted]"
009AD817 PUSH Q-DEMOQU.004818F0               UNICODE "valores"
009AD9FC MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE "COLUMNAS"
009ADB44 PUSH Q-DEMOQU.004819F8               UNICODE "valor"
009ADBDB PUSH Q-DEMOQU.00480E24               UNICODE "aciertos"
009AE097 MOV DWORD PTR SS:[EBP-B8],Q-DEI      UNICODE [redacted]
009AE0BD PUSH Q-DEMOQU.00481ACC               UNICODE "ERROR AL ABRIR EL GRUPO EN XML ("
```

Identify packed program

- The program has few imports, and particularly if the only imports are LoadLibrary and GetProcAddress .
- When the program is opened in IDA Pro, only a small amount of code is recognized by the automatic analysis.
- When the program is opened in OllyDbg, there is a warning that the program may be packed.
- The program shows section names that indicate a particular packer (such as UPX0).
- The program has abnormal section sizes, such as a .text section with a Size of Raw Data of 0 and Virtual Size of nonzero.

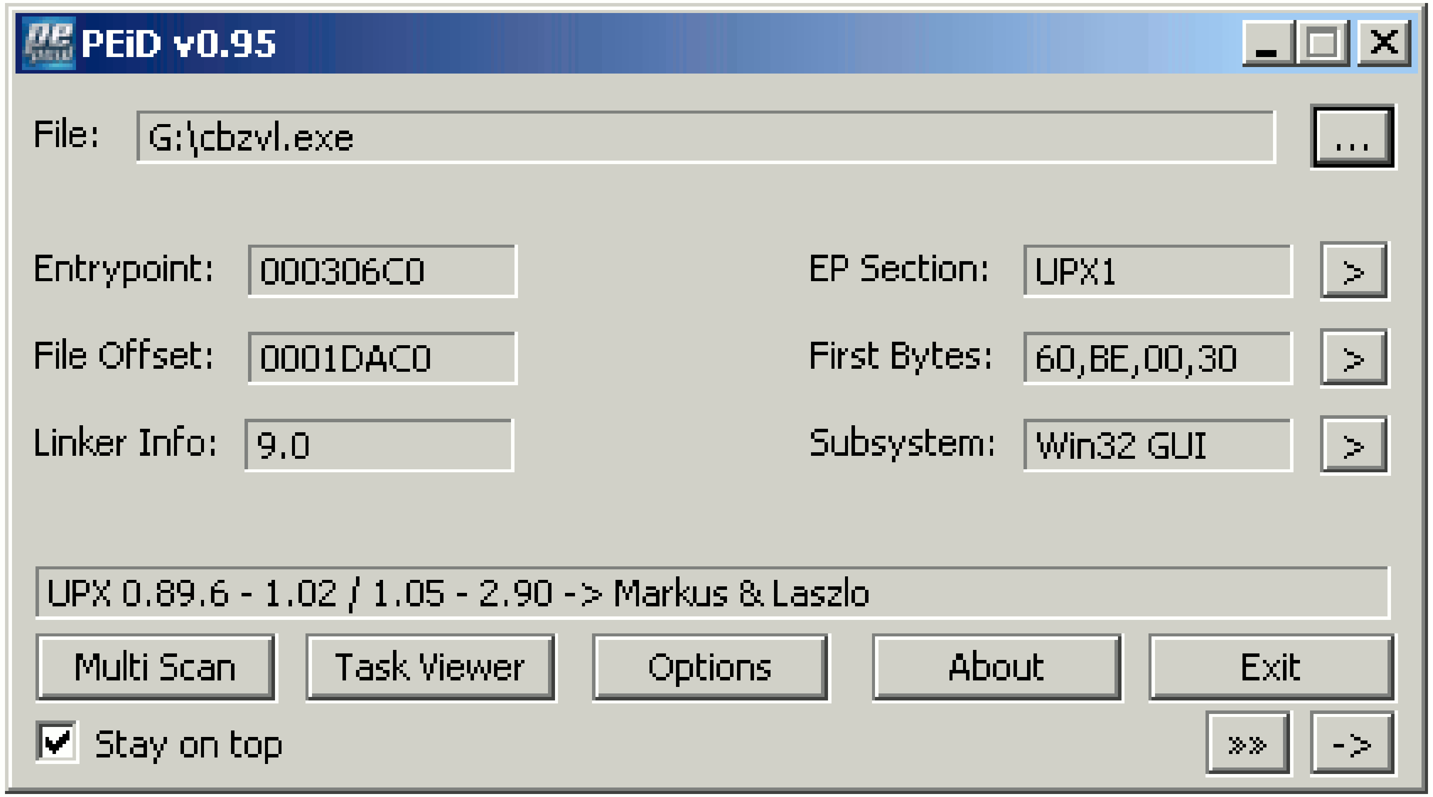
Packer detection tool

- Tools like
  - PEiD
  - ExeInfo PE
  - DiE

PEiD

- PEiD detects most common packers, cryptors and compilers for PE files.
- It can currently detect more than 470 different signatures in PE files.

Main interface of PEiD



File: G:\cbzvl.exe

Entrypoint: 000306C0

File Offset: 0001DAC0

Linker Info: 9.0

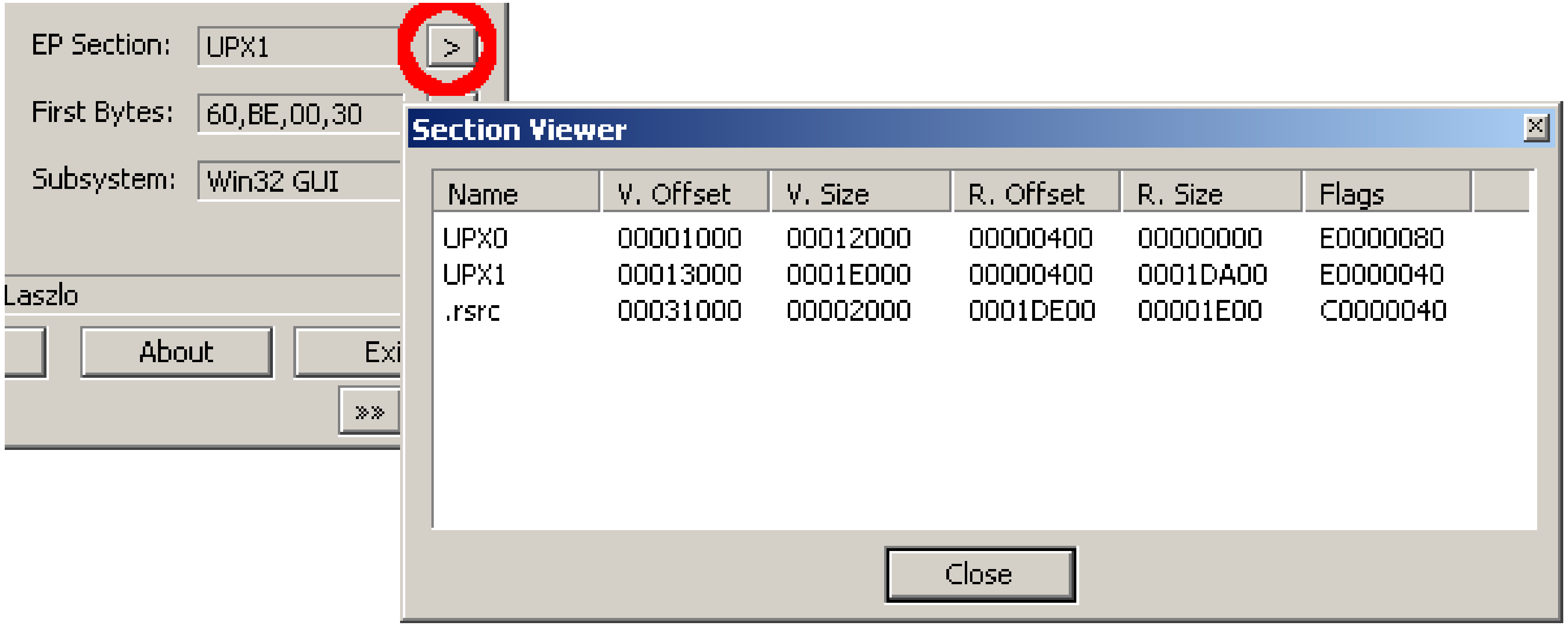
EP Section: UPX1

First Bytes: 60,BE,00,30

Subsystem: Win32 GUI

UPX 0.89.6 - 1.02 / 1.05 - 2.90 -> Markus & Laszlo

PEiD section viewer



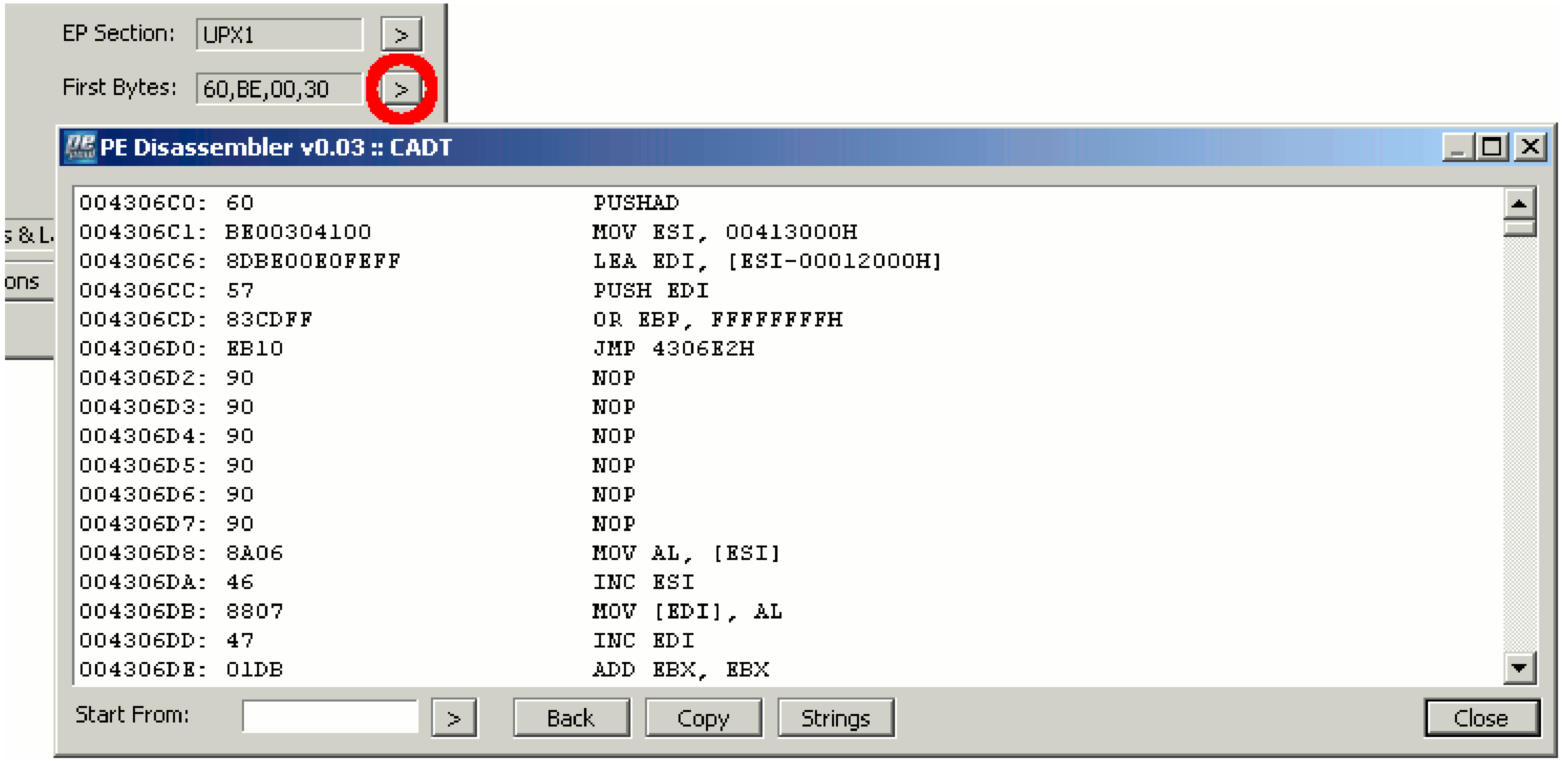
EP Section: UPX1

First Bytes: 60,BE,00,30

Subsystem: Win32 GUI

Name	V. Offset	V. Size	R. Offset	R. Size	Flags
UPX0	00001000	00012000	00000400	00000000	E0000080
UPX1	00013000	0001E000	00000400	0001DA00	E0000040
.rsrc	00031000	00002000	0001DE00	00001E00	C0000040

PE disassembler



EP Section: UPX1

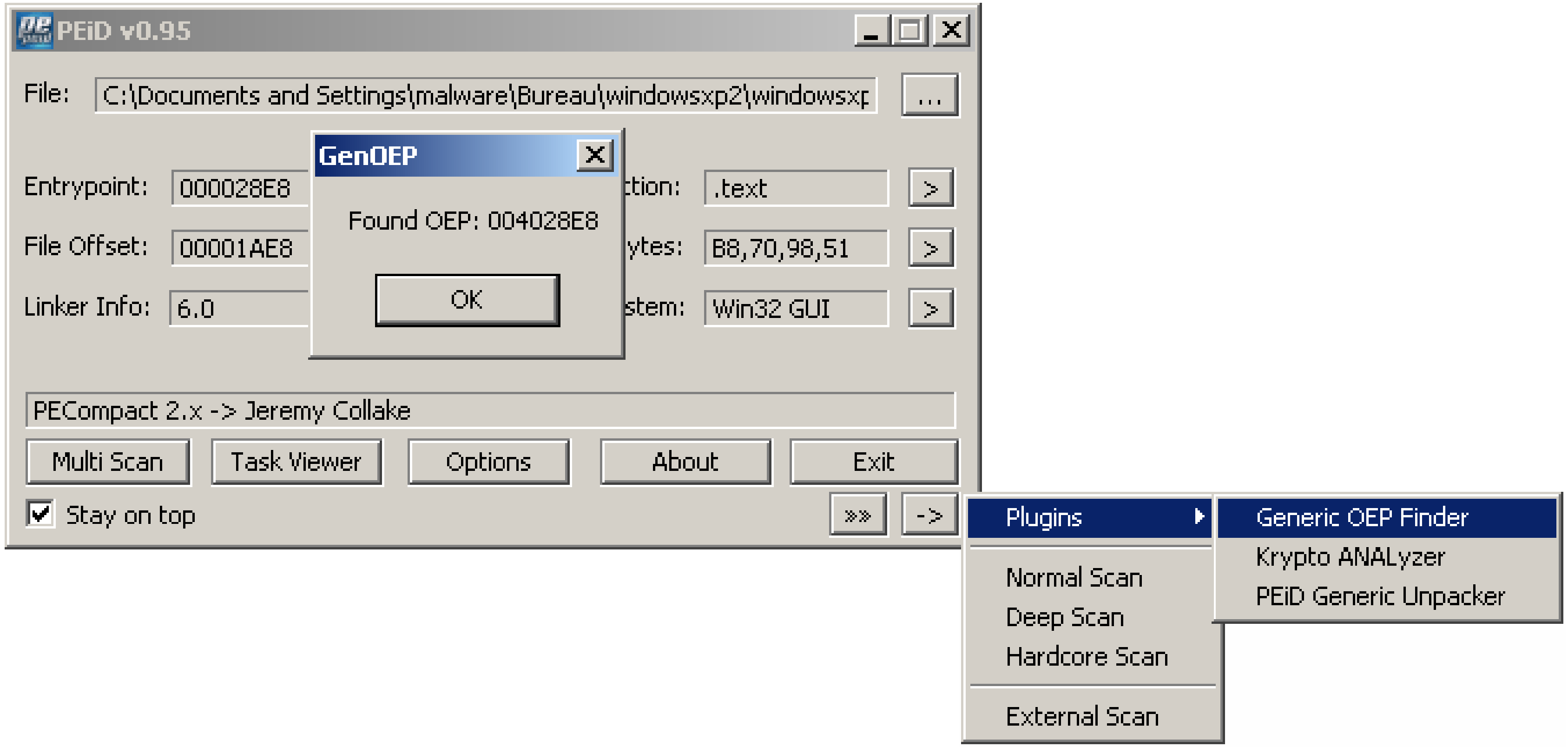
First Bytes: 60,BE,00,30

PE Disassembler v0.03 :: CADT

```
004306C0:  60                                PUSHAD
004306C1:  BE00304100                        MOV ESI, 00413000H
004306C6:  8DBE00E0FEFF                      LEA EDI, [ESI-00012000H]
004306CC:  57                                PUSH EDI
004306CD:  83CDFF                            OR EBP, FFFFFFFFH
004306D0:  EB10                              JMP 4306E2H
004306D2:  90                                NOP
004306D3:  90                                NOP
004306D4:  90                                NOP
004306D5:  90                                NOP
004306D6:  90                                NOP
004306D7:  90                                NOP
004306D8:  8A06                              MOV AL, [ESI]
004306DA:  46                                INC ESI
004306DB:  8807                              MOV [EDI], AL
004306DD:  47                                INC EDI
004306DE:  01DB                              ADD EBX, EBX
```

Start From:

Generic OEP Finder



PEiD v0.95

File: C:\Documents and Settings\malware\Bureau\windowsxp2\windowsxp

Entrypoint: 000028E8

File Offset: 00001AE8

Linker Info: 6.0

tion: .text

ytes: B8,70,98,51

stem: Win32 GUI

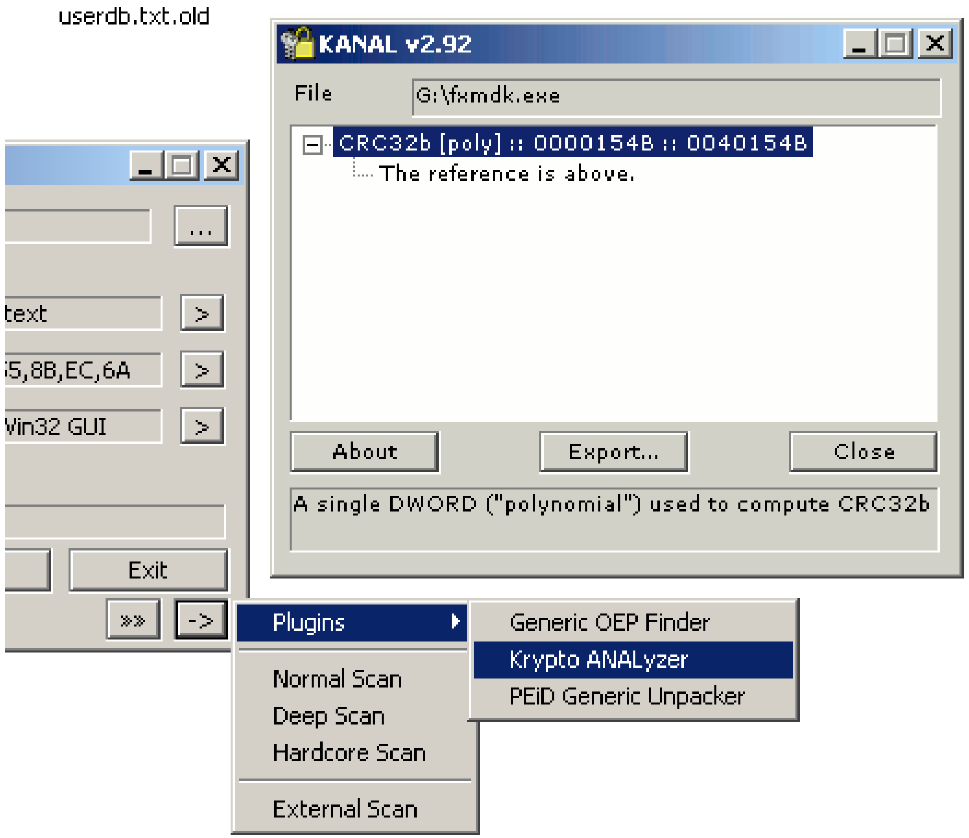
PECompact 2.x -> Jeremy Collake

☑ Stay on top

GenOEP

Found OEP: 004028E8

Krypto Analyzer



File: G:\fxmdk.exe

```
CRC32b [poly] :: 0000154B :: 0040154B
    The reference is above.
```

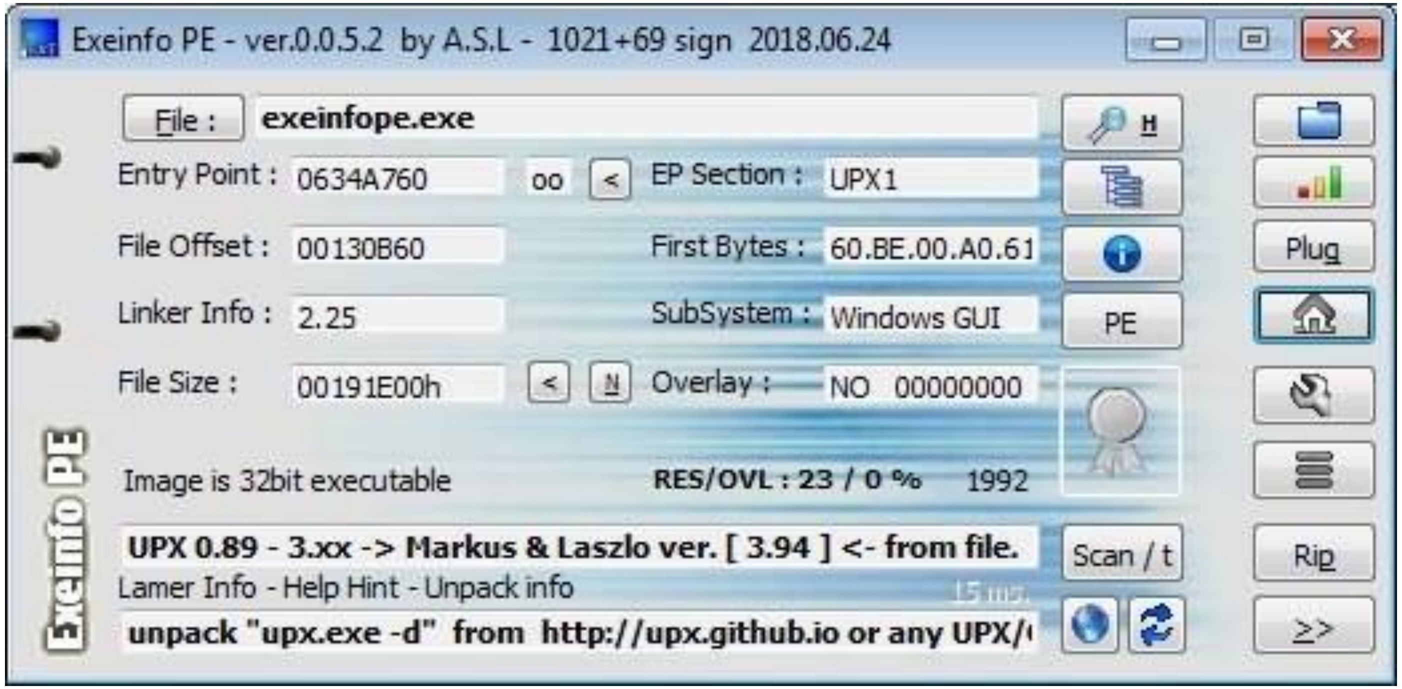
A single DWORD ("polynomial") used to compute CRC32b

Detect with Exeinfo PE

Exeinfo PE

- A program that lets you verify .exe files and check out all their properties.
- Another piece of info provided is the exact size and the point of entry.
- PE checker for packers, exe protectors, packer detector with solve hint for unpack

Detect packed binary and unpack info



Exeinfo PE - ver.0.0.5.2 by A.S.L - 1021+69 sign 2018.06.24

File : exeinfope.exe

Entry Point : 0634A760

EP Section : UPX1

File Offset : 00130B60

First Bytes : 60.BE.00.A0.61

Linker Info : 2.25

SubSystem : Windows GUI

File Size : 00191E00h

Overlay : NO 00000000

Image is 32bit executable

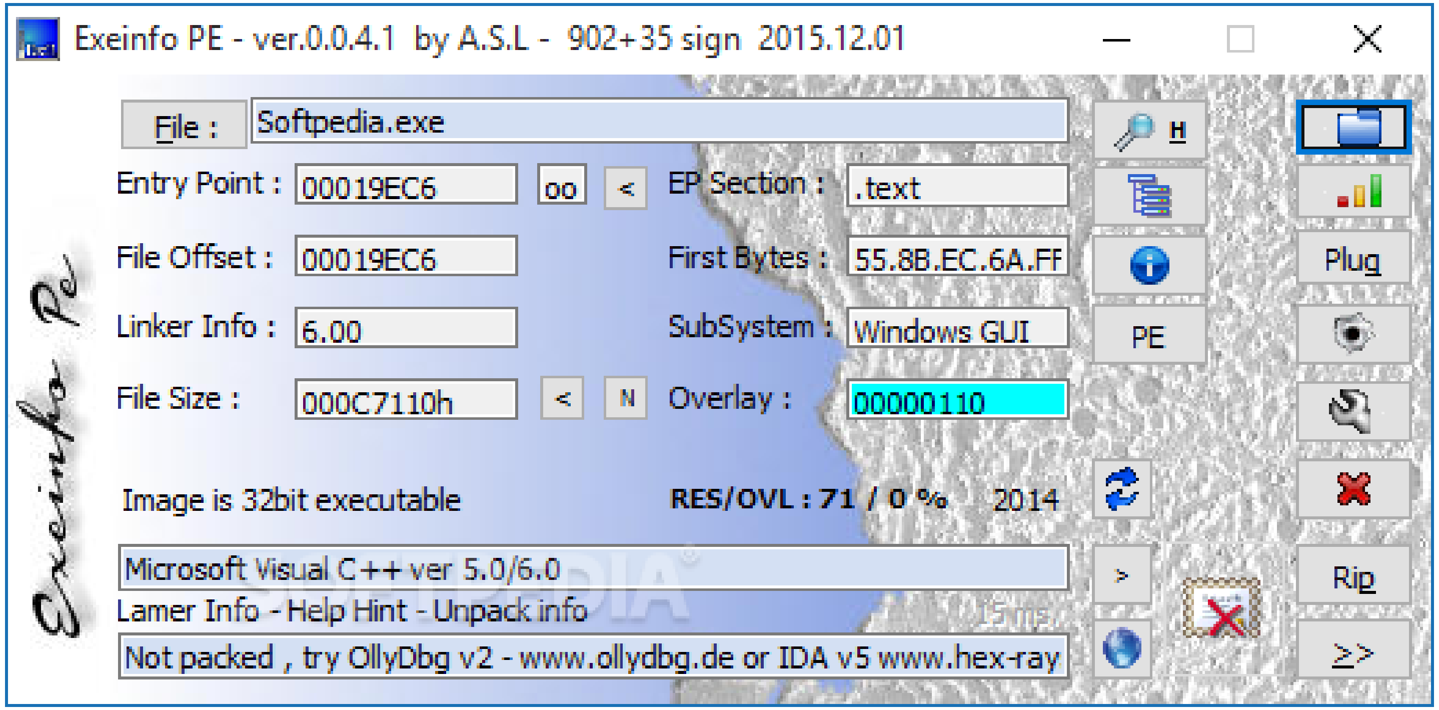
RES/OVL : 23 / 0 % 1992

UPX 0.89 - 3.xx -> Markus & Laszlo ver. [3.94] <- from file.

Lamer Info - Help Hint - Unpack info

unpack "upx.exe -d" from http://upx.github.io or any UPX/(

Not packed binary in Exeinfo PE



Exeinfo PE - ver.0.0.4.1 by A.S.L - 902+35 sign 2015.12.01

File : Softpedia.exe

Entry Point : 00019EC6

EP Section : .text

File Offset : 00019EC6

First Bytes : 55.8B.EC.6A.FF

Linker Info : 6.00

SubSystem : Windows GUI

File Size : 000C7110h

Overlay : 00000110

Image is 32bit executable

RES/OVL : 71 / 0 % 2014

Microsoft Visual C++ ver 5.0/6.0

Lamer Info - Help Hint - Unpack info

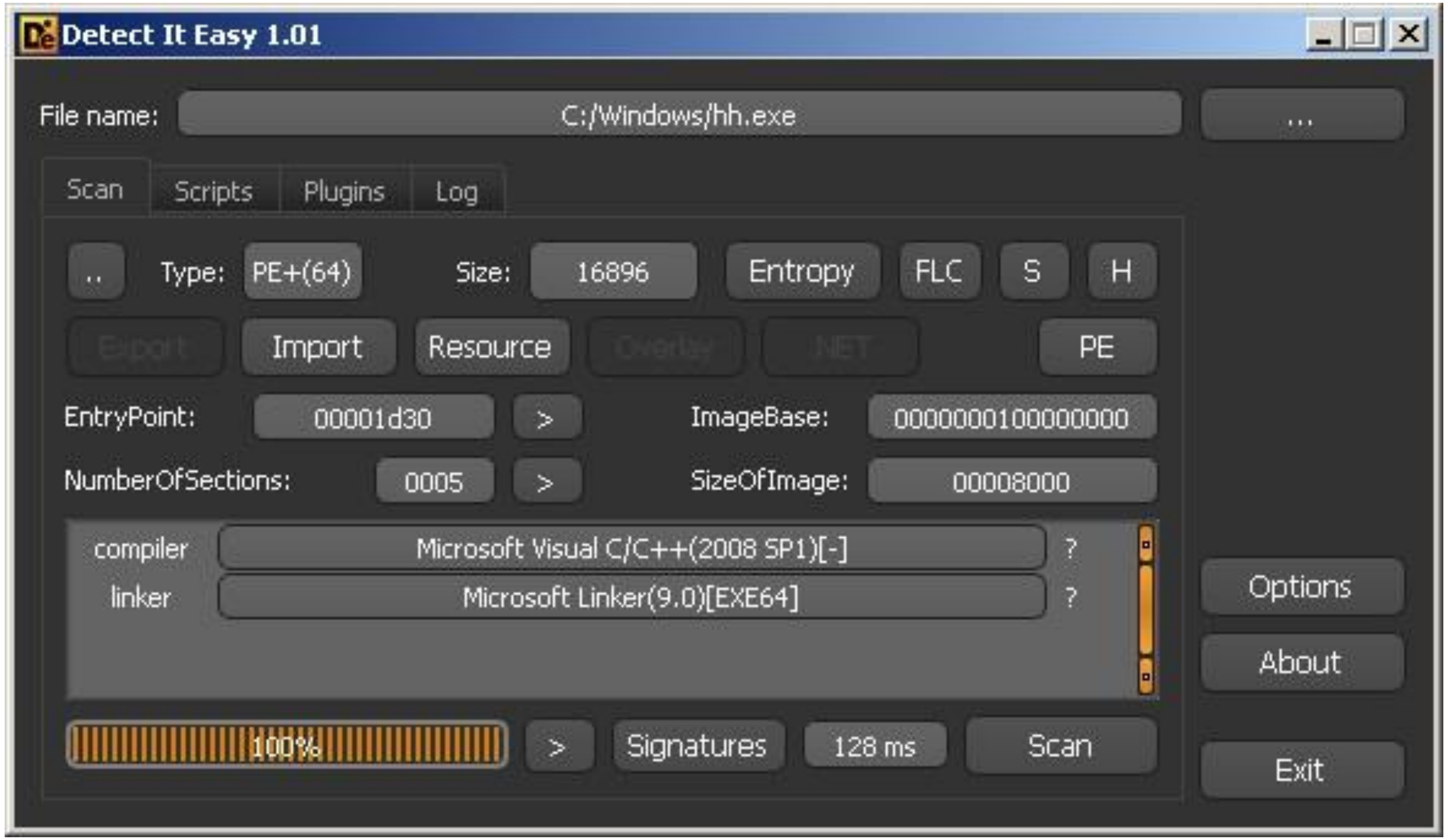
Not packed , try OllyDbg v2 - www.ollydbg.de or IDA v5 www.hex-ray

Detect using Detect it Easy (DiE)

DiE

- A program to determining types of files.
- An application that has been built as a packer identifier in order to help define a file type.
- Easily identify over 200 file types from their contents.

DiE interface



Detect It Easy 1.01

File name: C:/Windows/hh.exe

Type: PE+(64)

Size: 16896

EntryPoint: 00001d30

ImageBase: 0000000100000000

NumberOfSections: 0005

SizeOfImage: 00008000

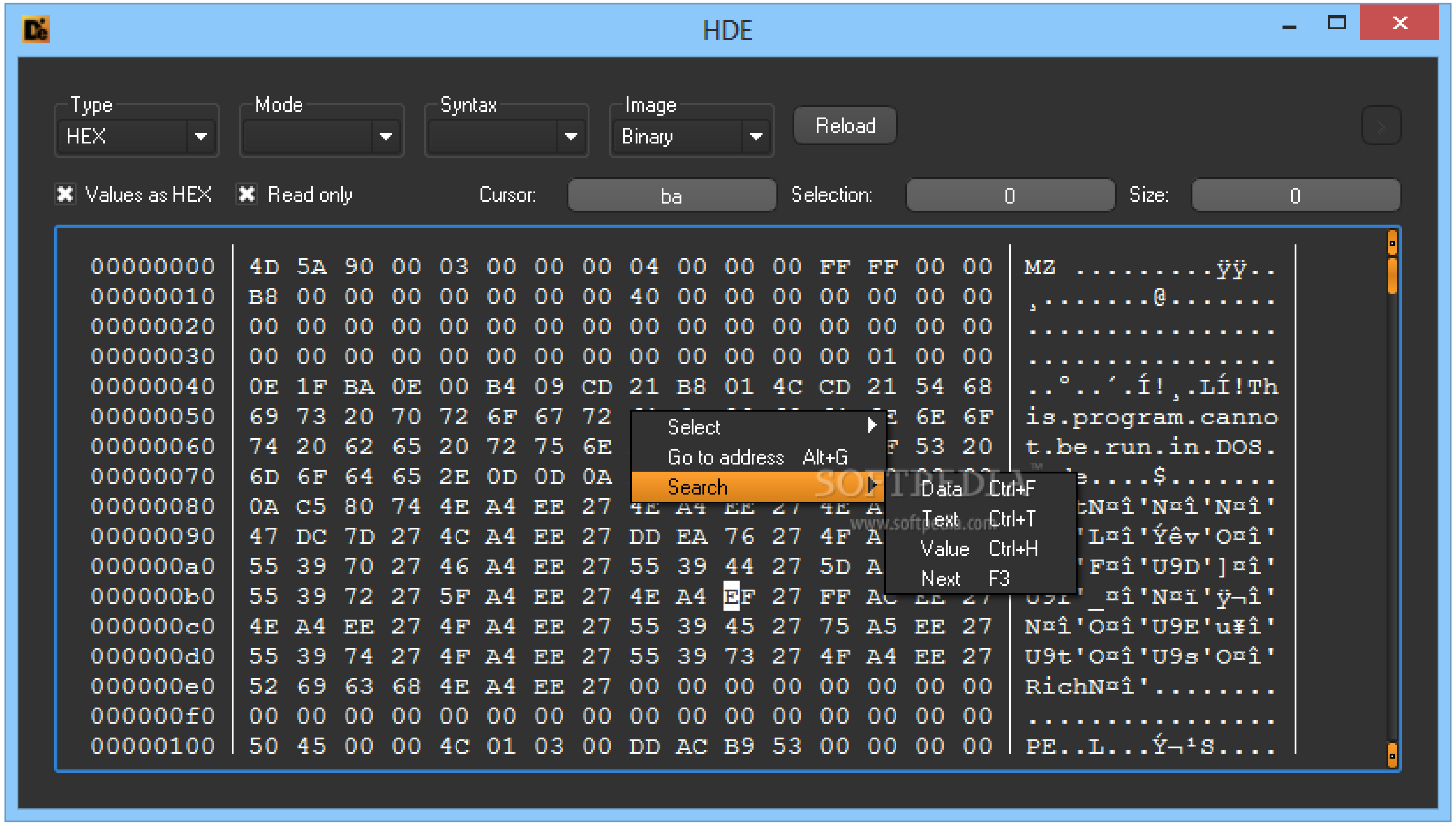
compiler Microsoft Visual C/C++(2008 SP1)[-]

linker Microsoft Linker(9.0)[EXE64]

100%

128 ms

Hex viewer in DiE



Type: HEX Mode: Syntax: Image: Binary

☒ Values as HEX ☒ Read only Cursor: ba Selection: 0 Size: 0

```
00000000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
00000010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
00000040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00000050  69 73 20 70 72 6F 67 72                E 6E 6F  is.program.canno
00000060  74 20 62 65 20 72 75 6E                F 53 20  t.be.run.in.DOS.
00000070  6D 6F 64 65 2E 0D 0D 0A                           e....$.......
00000080  0A C5 80 74 4E A4 EE 27 4E A4 EE 27 4E A          tN¤î'N¤î'N¤î'
00000090  47 DC 7D 27 4C A4 EE 27 DD EA 76 27 4F A          'L¤î'Ýêv'O¤î'
000000a0  55 39 70 27 46 A4 EE 27 55 39 44 27 5D A          'F¤î'U9D']¤î'
000000b0  55 39 72 27 5F A4 EE 27 4E A4 EF 27 FF AC EE 27     '_¤î'N¤ï'ÿ¬î'
000000c0  4E A4 EE 27 4F A4 EE 27 55 39 45 27 75 A5 EE 27  N¤î'O¤î'U9E'u¥î'
000000d0  55 39 74 27 4F A4 EE 27 55 39 73 27 4F A4 EE 27  U9t'O¤î'U9s'O¤î'
000000e0  52 69 63 68 4E A4 EE 27 00 00 00 00 00 00 00 00  RichN¤î'........
000000f0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000100  50 45 00 00 4C 01 03 00 DD AC B9 53 00 00 00 00  PE..L...Ý¬¹S....
```

Unpacking

- Automated
  - Easy
  - NSPack, UPack, and UPX
- Manual
  - Hard
  - Themida, ASPack, many more..

UPX

- Let’s try pack and unpack a binary with a common packer named UPX.

Malwares author ‘s advances techniques

Anti-disassembly

- Uses specially crafted code or data in a program to cause disassembly analysis tools to produce an incorrect program listing.
- Malware authors use anti-disassembly techniques to delay or prevent analysis of malicious code.
- Expert malware reverse engineer are required.
- Preventing certain automated analysis techniques.

Anti Debugging

- Popular anti-cracking and anti-reverse engineering protection techniques
- Malware authors know that malware analysts use debuggers to figure out how malware operates.
- The main goal of various anti-reverse engineering techniques is simply to complicate the process as much as possible.
- **IsDebuggerPresent**
- **PEB (Process Environment Block)**
- **NtGlobalFlag**
- **Many more…**

IsDebuggerPresent

- Simplest anti-debugging method is calling the IsDebuggerPresent function.
- This function detects if the calling process is being debugged by a user-mode debugger.

IsDebuggerPresent

```
int main()
{
  if (IsDebuggerPresent())
  {
    std::cout << "Stop debugging program!" << std::endl;
    exit(-1);
  }
  return 0;
}
```

Anti VM

- Malware authors sometimes use anti-virtual machine (anti-VM) techniques to thwart attempts at analysis.
- The malware attempts to detect whether it is being run inside a virtual machine.
- If a virtual machine is detected, it can act differently or simply not run.

Anti VM (cont.)

- Anti-VM techniques are most commonly found in malware that is widely deployed, such as bots, scareware, and spyware mostly
- Because honeypots often use virtual machines and because this malware typically targets the average user’s machine, which is unlikely to be running a virtual machine

Conclusion

- This chapter covered a large number of strategies for dealing with packed software.
- Now we know how hard to be a malware analyst.

To master this

- Student need to have knowledge in Portable Executable concept
- Reverse engineering
- Do some research