Practical Malware Analysis

Tools and techniques

Incident Response

- Case history
  - A medical clinic with 10 offices found malware on one of their workstations
  - Hired a consultant to clean & re-image that machine
- All done—case closed?

Incident Response

- After malware is found, you need to know
  - Did an attacker implant a rootkit or trojan on your systems?
  - Is the attacker really gone?
  - What did the attacker steal or add?
  - How did the attack get in
    - Root-cause analysis

Breach clean-up cost LinkedIn nearly $1 million, another $2-3 million in upgrades

Summary: *LinkedIn executives reveal on quarterly earnings call just what the June theft of 6.5 million passwords cost the company in forensic work and on-going security updates.*

By John Fontana for Identity Matters | August 3, 2012 -- 17:10 GMT (10:10 PDT)

Follow @johnfontana

Comments 0 | Vote 1 | Like 4 | Tweet 51 | Share | more +

LinkedIn spent nearly $1 million investigating and unraveling the theft of 6.5 million passwords in June and plans to spend up to $3 million more updating security on its social networking site.

- Link Ch 1a

Malware Analysis

- Dissecting malware to understand
  - How it works
  - How to identify it
  - How to defeat or eliminate it
- A critical part of incident response

The Goals of Malware Analysis

- Information required to respond to a network intrusion
  - Exactly what happened
  - Ensure you've located all infected machines and files
  - How to measure and contain the damage
  - Find signatures for intrusion detection systems

Signatures

- Host-based signatures
  - Identify files or registry keys on a victim computer that indicate an infection
  - Focus on what the malware did to the system, not the malware itself
    - Different from antivirus signature
- Network signatures
  - Detect malware by analyzing network traffic
  - More effective when made using malware analysis

Malware Analysis Techniques

Static v. Dynamic Analysis

- Static Analysis
  - Examines malware without running it
  - Tools: VirusTotal, strings, a disassembler like IDA Pro
- Dynamic Analysis
  - Run the malware and monitor its effect
  - Use a virtual machine and take snapshots
  - Tools: RegShot, Process Monitor, Process Hacker, CaptureBAT
  - RAM Analysis: Mandant Redline and Volatility

Basic Analysis

- Basic static analysis
  - View malware without looking at instructions
  - Tools: VirusTotal, strings
  - Quick and easy but fails for advanced malware and can miss important behavior
- Basic dynamic analysis
  - Easy but requires a safe test environment
  - Not effective on all malware

Advanced Analysis

- Advanced static analysis
  - Reverse-engineering with a disassembler
  - Complex, requires understanding of assembly code
- Advanced Dynamic Analysis
  - Run code in a debugger
  - Examines internal state of a running malicious executable

Types of Malware

Types of Malware

- Backdoor
  - Allows attacker to control the system
- Botnet
  - All infected computers receive instructions from the same Command-and-Control (C&C) server
- Downloader
  - Malicious code that exists only to download other malicious code
  - Used when attacker first gains access

Types of Malware

- Information-stealing malware
  - Sniffers, keyloggers, password hash grabbers
- Launcher
  - Malicious program used to launch other malicious programs
  - Often uses nontraditional techniques to ensure stealth or greater access to a system
- Rootkit
  - Malware that conceals the existence of other code
  - Usually paired with a backdoor

Types of Malware

- Scareware
  - Frightens user into buying something
  - Link Ch 1b

Fake FBI warning tricks man into surrendering himself for possession of child porn

29 Jul, 2013 | by Nishtha Kanal | Like 3 | +1 0 | Tweet 3 | Share

Here's a weird one. We've heard of viruses and malware bringing harm to computers but in a rare instance, a "ransomware" has brought a positive outcome. A man in the US turned himself in to the police after a pop-up caused by a ransomware informed him that child porn had been identified on his machine.

Jay Matthew Riley, a 21-year-old from Virginia was browsing the Internet, when a pop-up containing an "FBI warning" informed him that it had detected child pornography on his machine. The message went on to tell Riley to pay up a fine online or face the consequences.

Types of Malware

- Spam-sending malware
  - Attacker rents machine to spammers
- Worms or viruses
  - Malicious code that can copy itself and infect additional computers

Mass v. Targeted Malware

- Mass malware
  - Intended to infect as many machines as possible
  - Most common type
- Targeted malware
  - Tailored to a specific target
  - Very difficult to detect, prevent, and remove
  - Requires advanced analysis
  - Ex: Stuxnet

General Rules for Malware Analysis

General Rules for Malware Analysis

- Don’t Get Caught in Details
  - You don’t need to understand 100% of the code
  - Focus on key features
- Try Several Tools
  - If one tool fails, try another
  - Don’t get stuck on a hard issue, move along
- Malware authors are constantly raising the bar

Basic Static Analysis

Techniques

- Antivirus scanning
- A file’s strings, functions, and headers, etc
  - Using tools such as PEView, PEiD, Ada Pro, etc

Antivirus Scanning

Only a First Step

VirusTotal is an Alphabet product that analyzes suspicious files, URLs, domains and IP addresses to detect malware and other types of threats, and automatically shares them with the security community. To view VirusTotal reports, you'll be submitting file attachment hashes, IP addresses, or domains to VirusTotal.

VirusTotal is convenient, but using it may alert attackers that they've been caught

VirusTotal

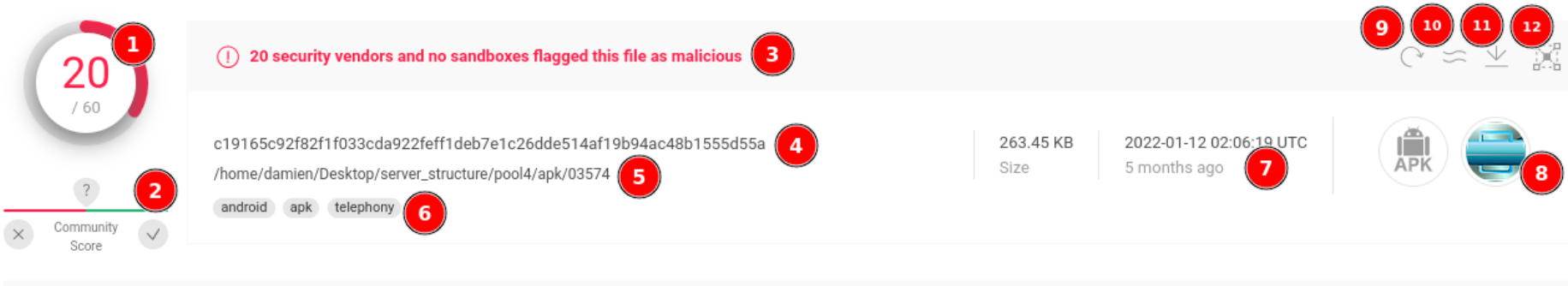
VIRUSTOTAL

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

FILE | URL | SEARCH

Choose file

By submitting data above, you are agreeing to our Terms of Service and Privacy Policy, and to the **sharing of your Sample submission with the security community.** Please do not submit any personal information; VirusTotal is not responsible for the contents of your submission. Learn more.



1) and 3) The total number of VirusTotal partners who consider this file harmful (in this case, 20) out of the total number of partners who reviewed the file (in this case, 60).

2) The reputation of the given URL as determined by VirusTotal's Community (registered users). Users sometimes vote on files and URLs submitted to VirusTotal, these users in turn have a reputation themselves, the *community score* condenses the votes performed on a given item weighted by the reputation of the users that casted these votes.

4) SHA-256 (a cryptographic hash function) is a unique way to identify a file and used in the security industry to unambiguously refer to a particular threat.

5) File name of last submission, and access to search by file names.

6) Tags.

7) The date and time (UTC) of the review.

8) Icon for the file type.

9) Button to reanalyze the file.

10) Search for similar files.

11) Download sample.

12) Explore the file in VirusTotal Graph.

Resource

- See the link below for the online documentation of VirusTotal reports:

https://support.virustotal.com/hc/en-us/articles/115002719069-Reports

Finding Strings

Strings

- Any sequence of printable characters is a **string**
- Strings are terminated by a **null** (0x00)
- ASCII characters are 8 bits long
  - Now called ANSI
- Unicode characters are 16 bits long
  - Microsoft calls them "wide characters"

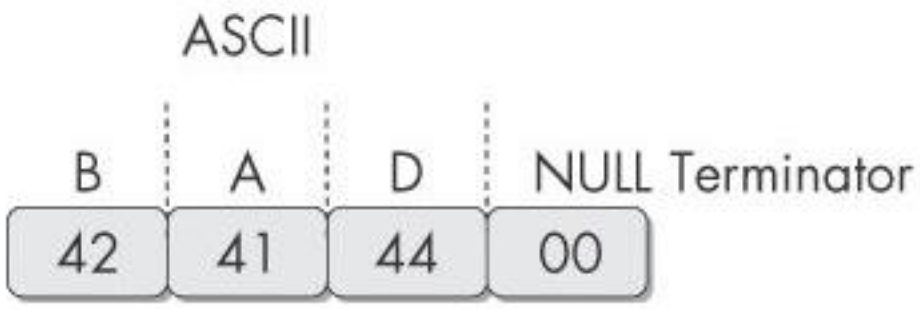


Figure 2-2. ASCII representation of the string BAD

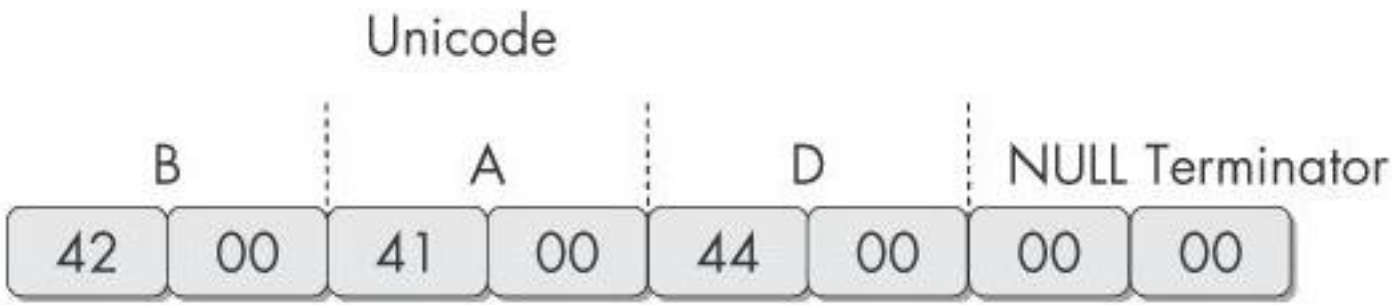


Figure 2-3. Unicode representation of the string BAD

The strings Command

- Native in Linux, also available for Windows
- Finds all strings in a file 3 or more characters long

The strings Command

- Bold items can be ignored
- **GetLayout** and **SetLayout** are Windows functions
- GDI32.DLL is a Dynamic Link Library

```
C:>strings bp6.ex_
VP3
VW3
t$@
D$4
99.124.22.1 4
e-@
GetLayout 1
GDI32.DLL 3
SetLayout 2
M}C
Mail system DLL is invalid.!Send Mail failed to
send message. 5
```

Download Strings v2.54

Strings v2.54

Article • 06/22/2021 • 2 minutes to read • 6 contributors

Feedback

By Mark Russinovich

Published: June 22, 2021

Download Strings (534 KB)

Introduction

Working on NT and Win2K means that executables and object files will many times have embedded UNICODE strings that you cannot easily see with a standard ASCII strings or grep programs. So we decided to roll our own. Strings just scans the file you pass it for UNICODE (or ASCII) strings of a default length of 3 or more UNICODE (or ASCII) characters. Note that it works under Windows 95 as well.

Using Strings

Usage:

https://learn.microsoft.com/en-us/sysinternals/downloads/strings

What is DLL file?

- A DLL is a library that contains code and data that can be used by more than one program at the same time.
- For example, in Windows operating systems, the Comdlg32 DLL performs common dialog box related functions. Each program can use the functionality that is contained in this DLL to implement an **Open** dialog box. It helps promote code reuse and efficient memory usage.

https://learn.microsoft.com/

PEview

- Install the PEview tool from: http://wjradburn.com/software/ (first link on the page: PEview version 0.9.9 (.zip 31KB)).

- **PEview** provides a quick and easy way to view the structure and content of 32-bit Portable Executable (PE) files.
- This PE file viewer displays header, section, directory, import table, export table, and resource information within EXE, DLL, OBJ, LIB, DBG, and other file types.

Portable Executable (PE) File Format

EXE Files

PE Files

- Used by Windows executable files, object code, and DLLs
- A data structure that contains the information necessary for Windows to load the file
- Almost every file executed on Windows is in PE format

PE Header

- Information about the code
- Type of application
- Required library functions
- Space requirements

PEview - C:\Documents and Settings\Administrator\Desktop\0170.dll

File View Go Help

- IMAGE_SECTION_HEADEI
- SECTION .text
- SECTION .rdata
  - IMPORT Address Table
  - DELAY IMPORT DLL Na
  - DELAY IMPORT Descrip
  - DELAY IMPORT Name T
  - DELAY IMPORT Hints/N
  - IMPORT Directory Table
  - IMPORT Name Table
  - IMPORT Hints/Names &
  - IMAGE_EXPORT_DIREC
  - EXPORT Address Table
  - EXPORT Name Pointer T
  - EXPORT Ordinal Table
  - EXPORT Names
- SECTION .data
- SECTION .rsrc
- SECTION .reloc

pFile	Data	Description	Value
0000C168	0000378E	Function RVA	0001 EuropeAppliesC
0000C16C	0000378F	Function RVA	0002 FromH
0000C170	00003C6B	Function RVA	0003 LimitationLimite
0000C174	00004EA9	Function RVA	0004 SandyfordON
0000C178	00004BE9	Function RVA	0005 ServingIf
0000C17C	00004FF8	Function RVA	0006 YouBlockMiddle
0000C180	00004EAA	Function RVA	0007 YourYou

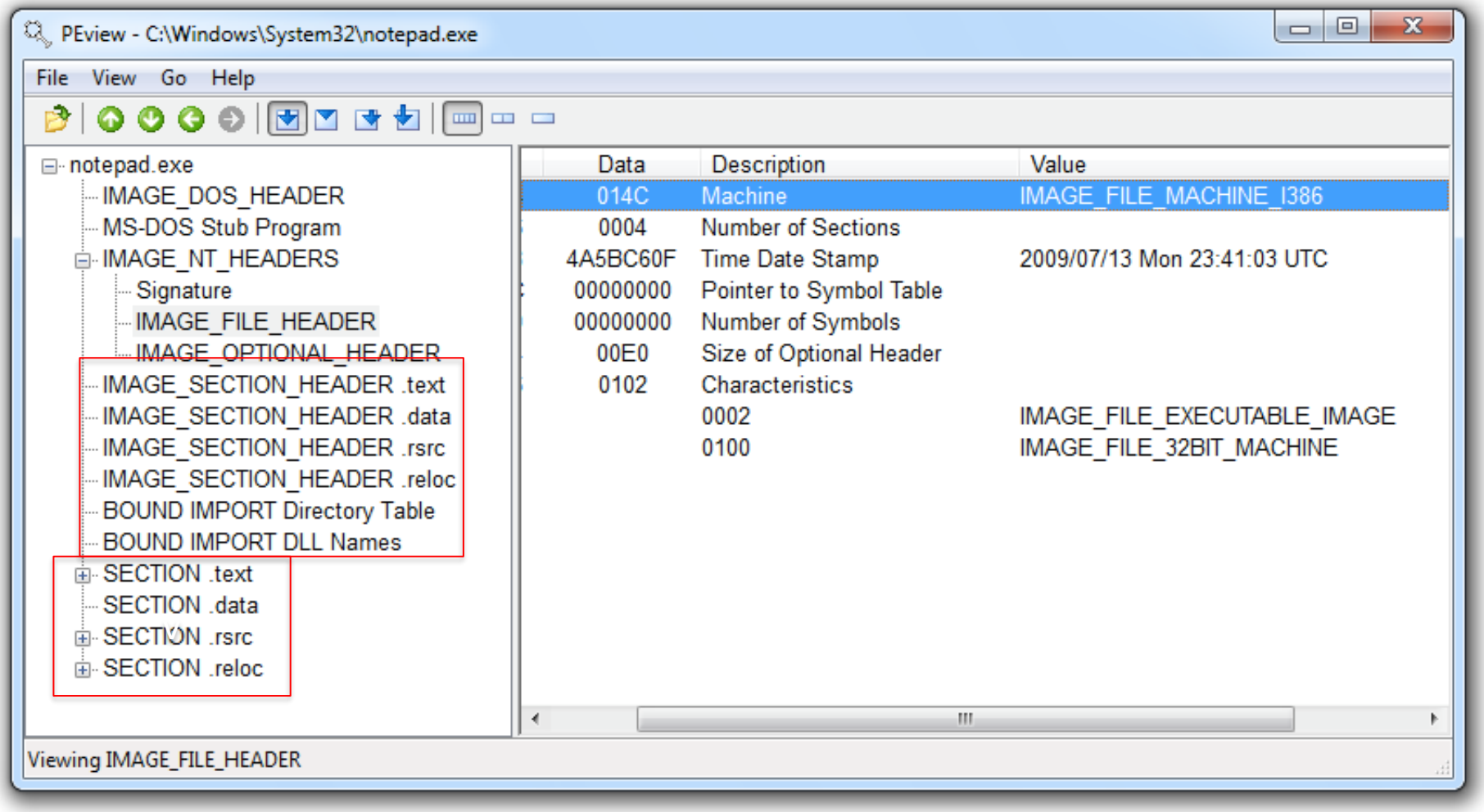
Viewing EXPORT Address Table

The PE File Headers and Sections

Important PE Sections

- **.text** -- instructions for the CPU to execute
- **.rdata** -- imports & exports
- **.data** – global data
- **.rsrc** – strings, icons, images, menus

PEView


PEview - C:\Windows\System32\notepad.exe
File View Go Help
notepad.exe
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
Signature
IMAGE_FILE_HEADER
IMAGE_OPTIONAL_HEADER
IMAGE_SECTION_HEADER .text
IMAGE_SECTION_HEADER .data
IMAGE_SECTION_HEADER .rsrc
IMAGE_SECTION_HEADER .reloc
BOUND IMPORT Directory Table
BOUND IMPORT DLL Names
SECTION .text
SECTION .data
SECTION .rsrc
SECTION .reloc
Data Description Value
014C Machine IMAGE_FILE_MACHINE_I386
0004 Number of Sections
4A5BC60F Time Date Stamp 2009/07/13 Mon 23:41:03 UTC
00000000 Pointer to Symbol Table
00000000 Number of Symbols
00E0 Size of Optional Header
0102 Characteristics
0002 IMAGE_FILE_EXECUTABLE_IMAGE
0100 IMAGE_FILE_32BIT_MACHINE
Viewing IMAGE_FILE_HEADER

Time Date Stamp

- Shows when this executable was compiled
- Older programs are more likely to be known to antivirus software
- But sometimes the date is wrong
  - All Delphi programs show June 19, 1992
  - Date can also be faked

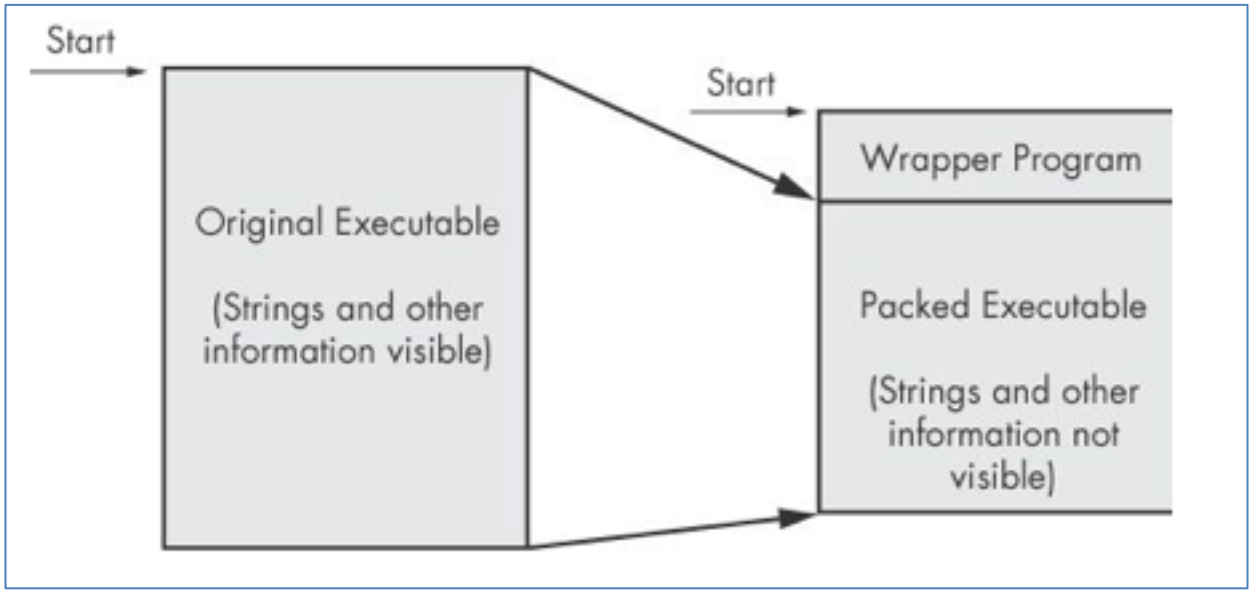
IMAGE_SECTION_HEADER

- Virtual Size – RAM
- Size of Raw Data – DISK
- For **.text** section, normally equal, or nearly equal
- Packed executables show Virtual Size much larger than Size of Raw Data for **.text** section

Packed and Obfuscated Malware

Packing Files

- The code is compressed, like a Zip file
- This makes the strings and instructions unreadable
- All you'll see is the **wrapper** – small code that unpacks the file when it is run



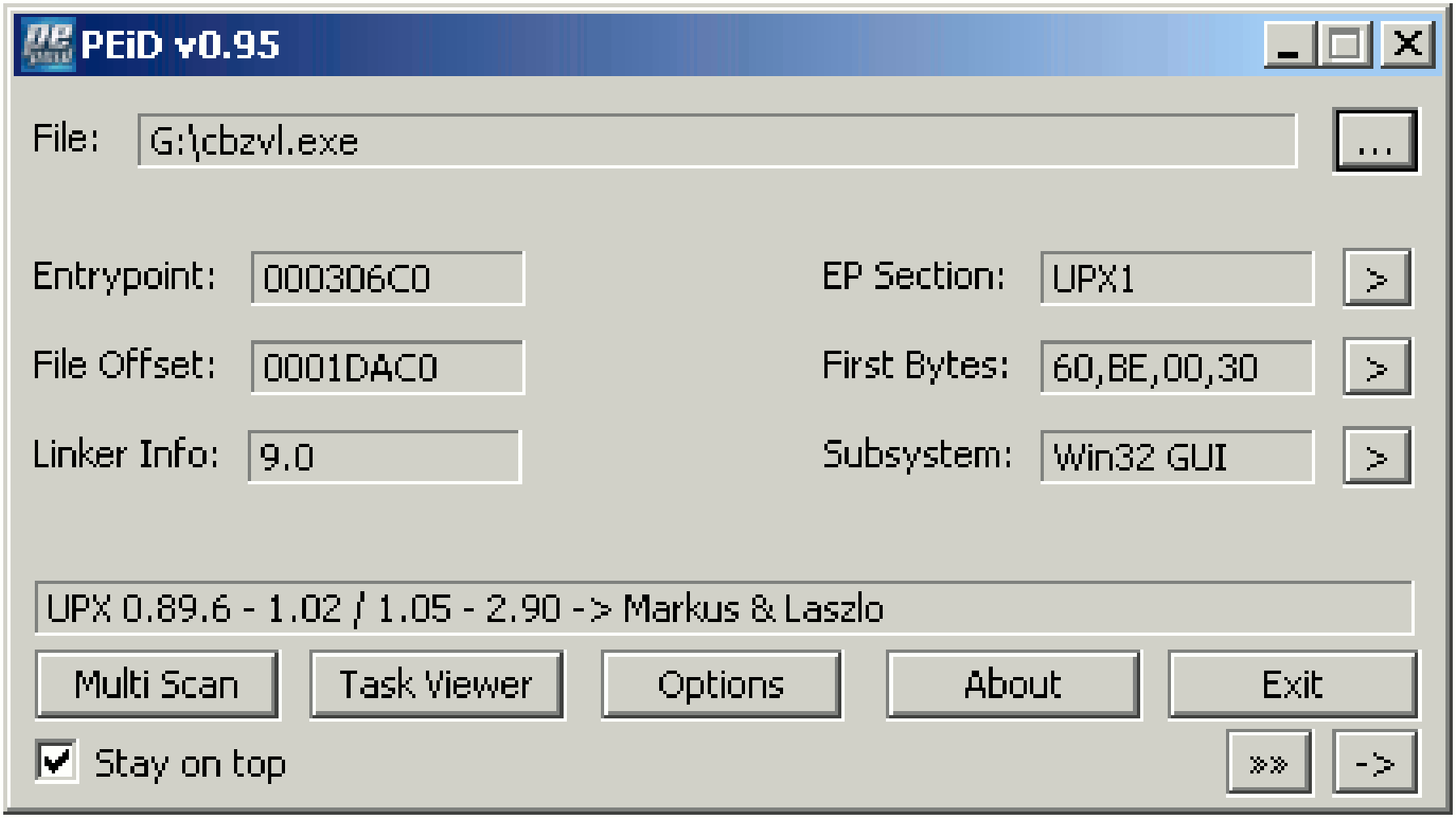
PEiD

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

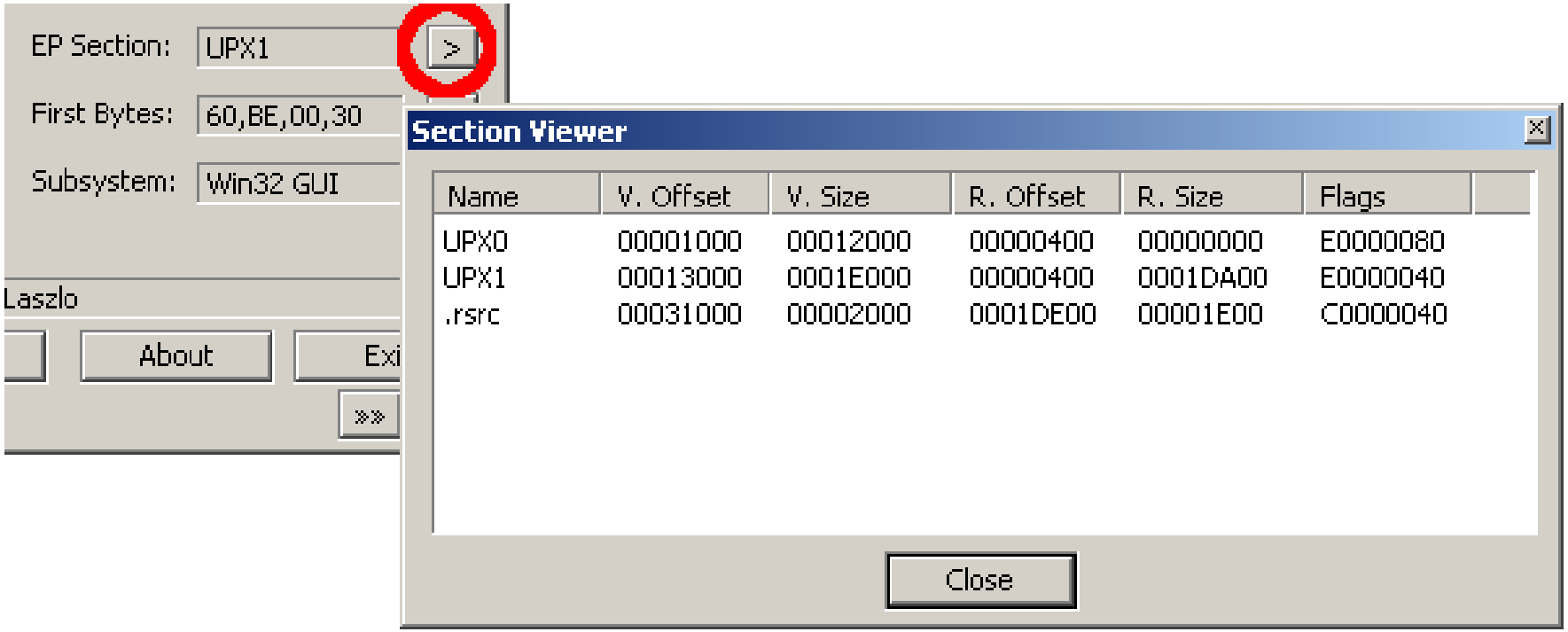
- To download, you can use the link below:
  - https://softfamous.com/postdownload-file/peid/12446/4719/

https://www.aldeid.com/wiki/PEiD

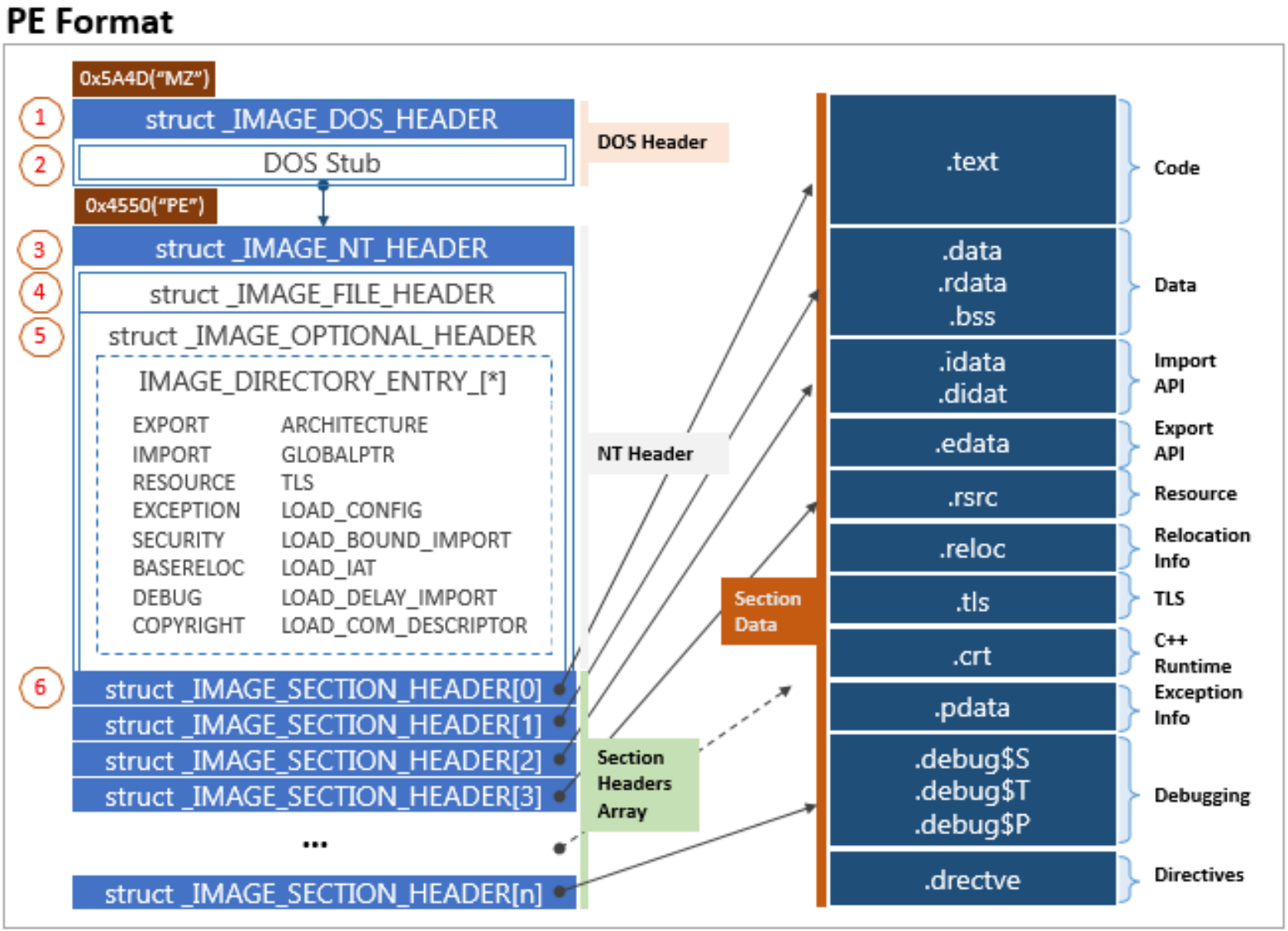
PEiD Main Interface



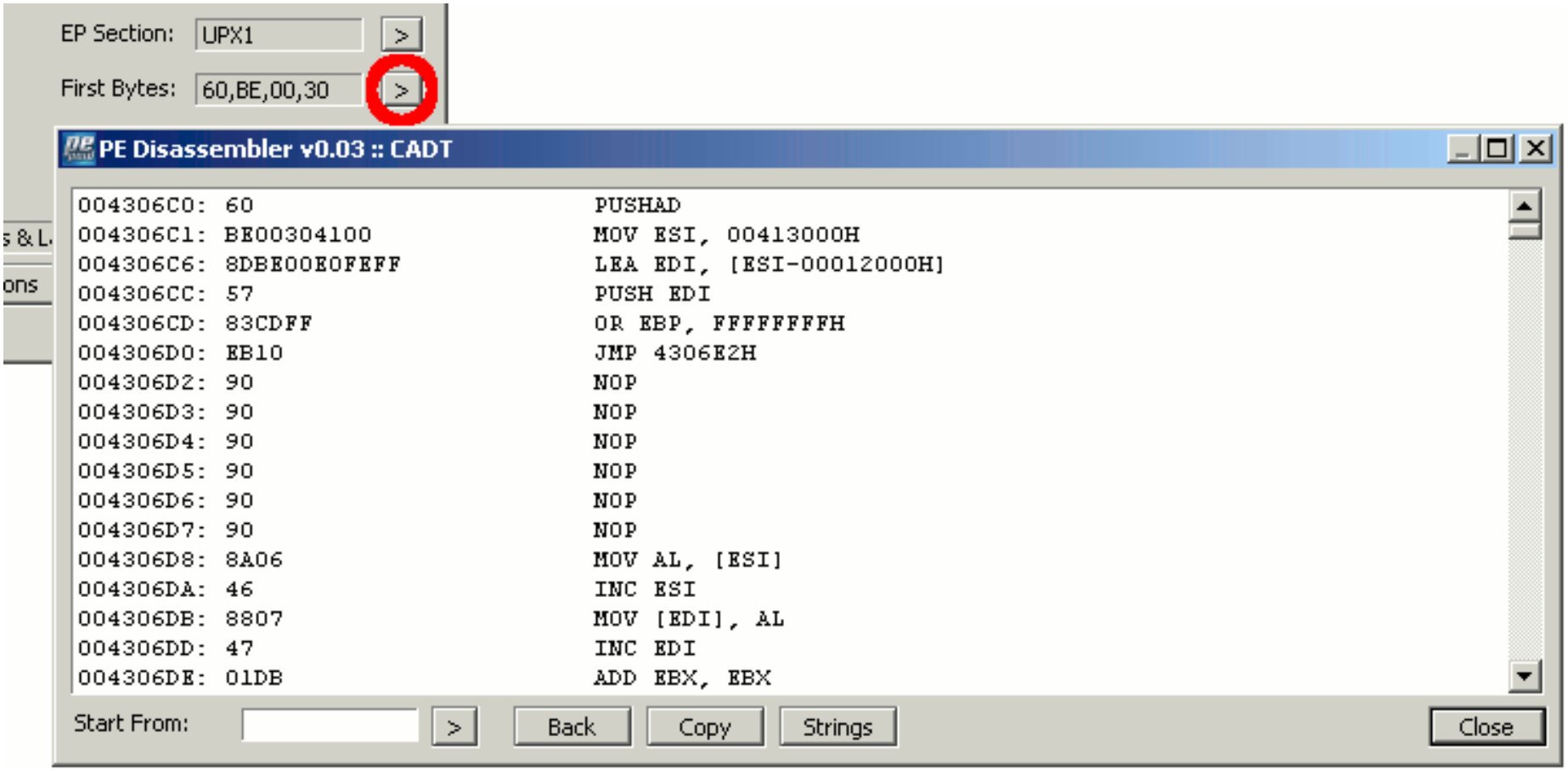
Section Viewer in PEiD



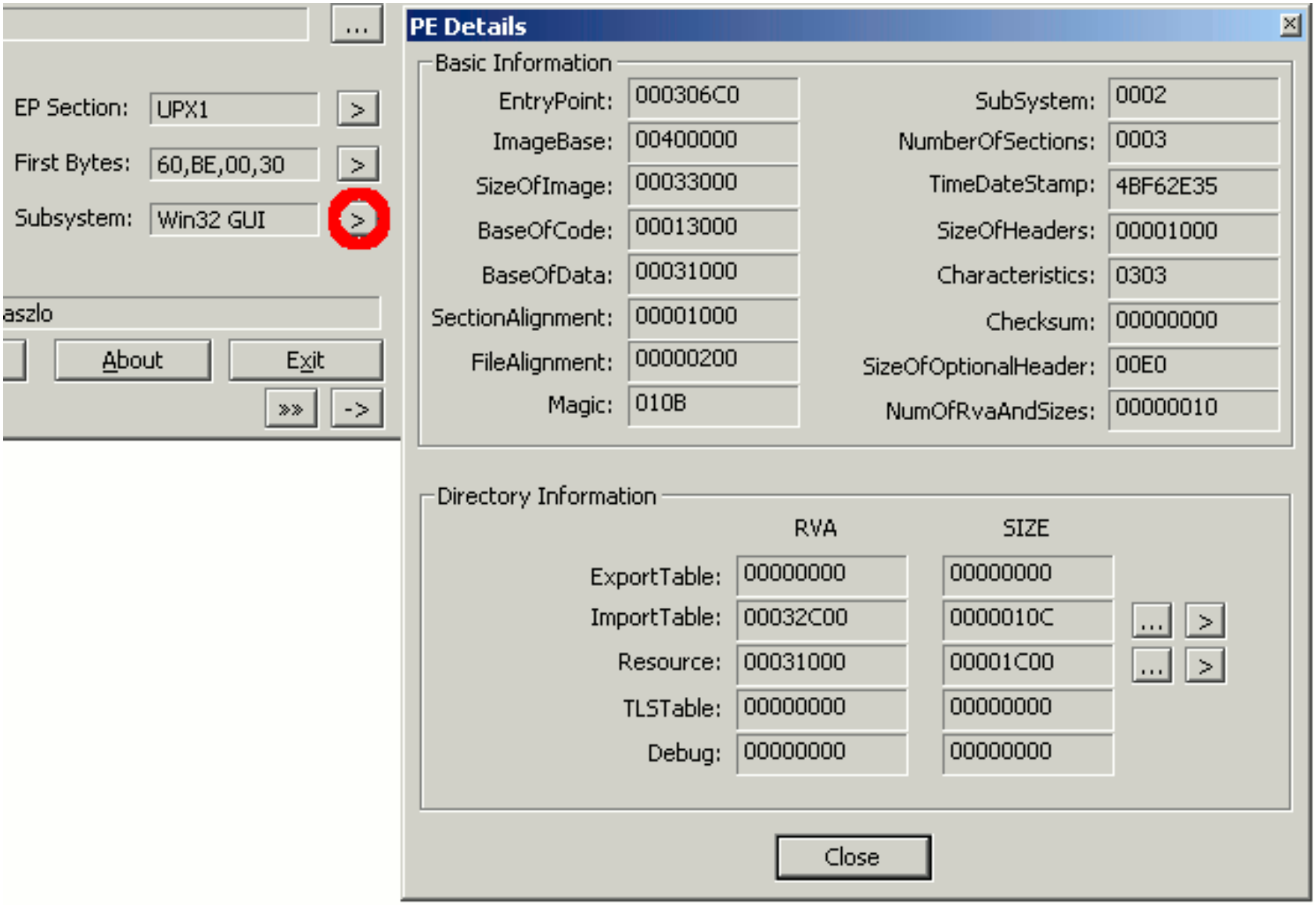
PE File Format-Sections in PEiD



PE disassembler



PE Details


...
EP Section: UPX1 >
First Bytes: 60,BE,00,30 >
Subsystem: Win32 GUI >
aszlo
About
Exit
»»
->
PE Details
Basic Information
EntryPoint: 000306C0
ImageBase: 00400000
SizeOfImage: 00033000
BaseOfCode: 00013000
BaseOfData: 00031000
SectionAlignment: 00001000
FileAlignment: 00000200
Magic: 010B
SubSystem: 0002
NumberOfSections: 0003
TimeDateStamp: 4BF62E35
SizeOfHeaders: 00001000
Characteristics: 0303
Checksum: 00000000
SizeOfOptionalHeader: 00E0
NumOfRvaAndSizes: 00000010
Directory Information
RVA
SIZE
ExportTable: 00000000 00000000
ImportTable: 00032C00 0000010C
Resource: 00031000 00001C00
TLSTable: 00000000 00000000
Debug: 00000000 00000000
Close

Detecting Packers with PEiD

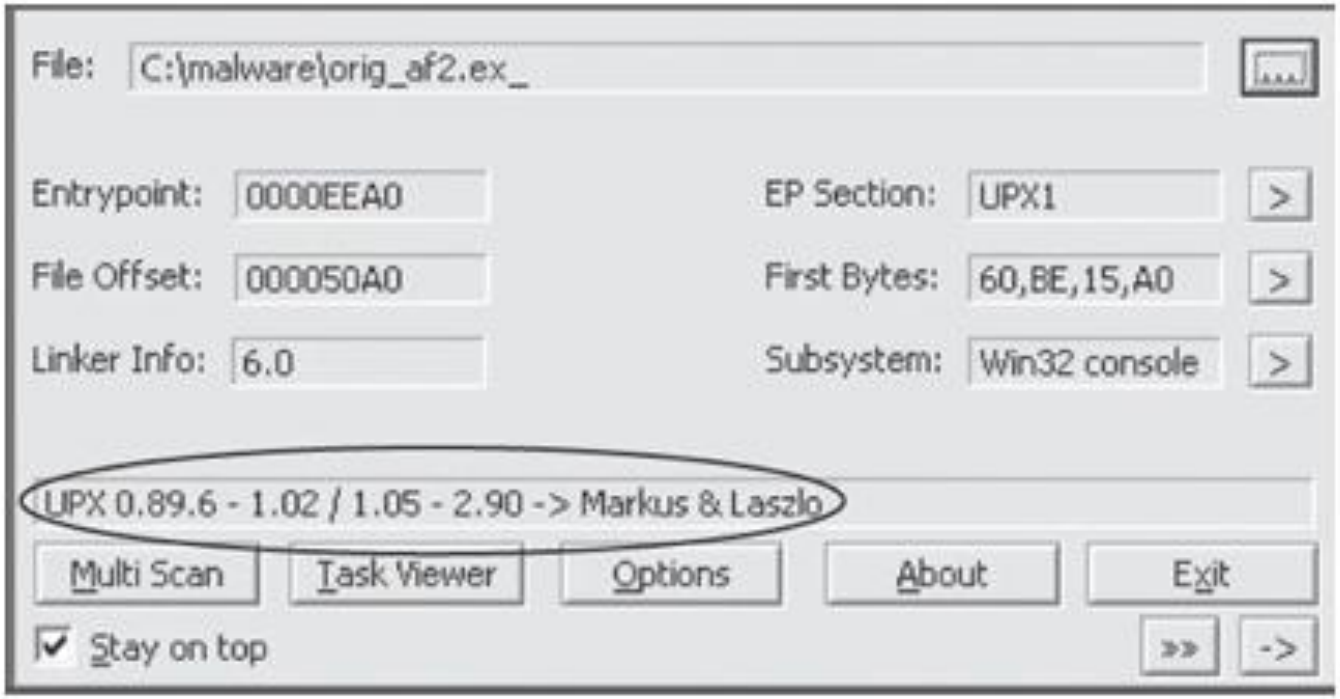
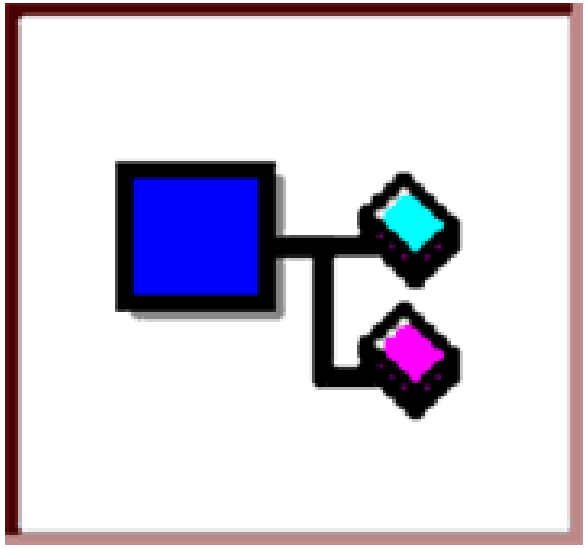


Figure 2-5. The PEiD program

NOTE

Many PEiD plug-ins will run the malware executable without warning! (See Chapter 3 to learn how to set up a safe environment for running malware.) Also, like all programs, especially those used for malware analysis, PEiD can be subject to vulnerabilities. For example, PEiD version 0.92 contained a buffer overflow that allowed an attacker to execute arbitrary code. This would have allowed a clever malware writer to write a program to exploit the malware analyst's machine. Be sure to use the latest version of PEiD.

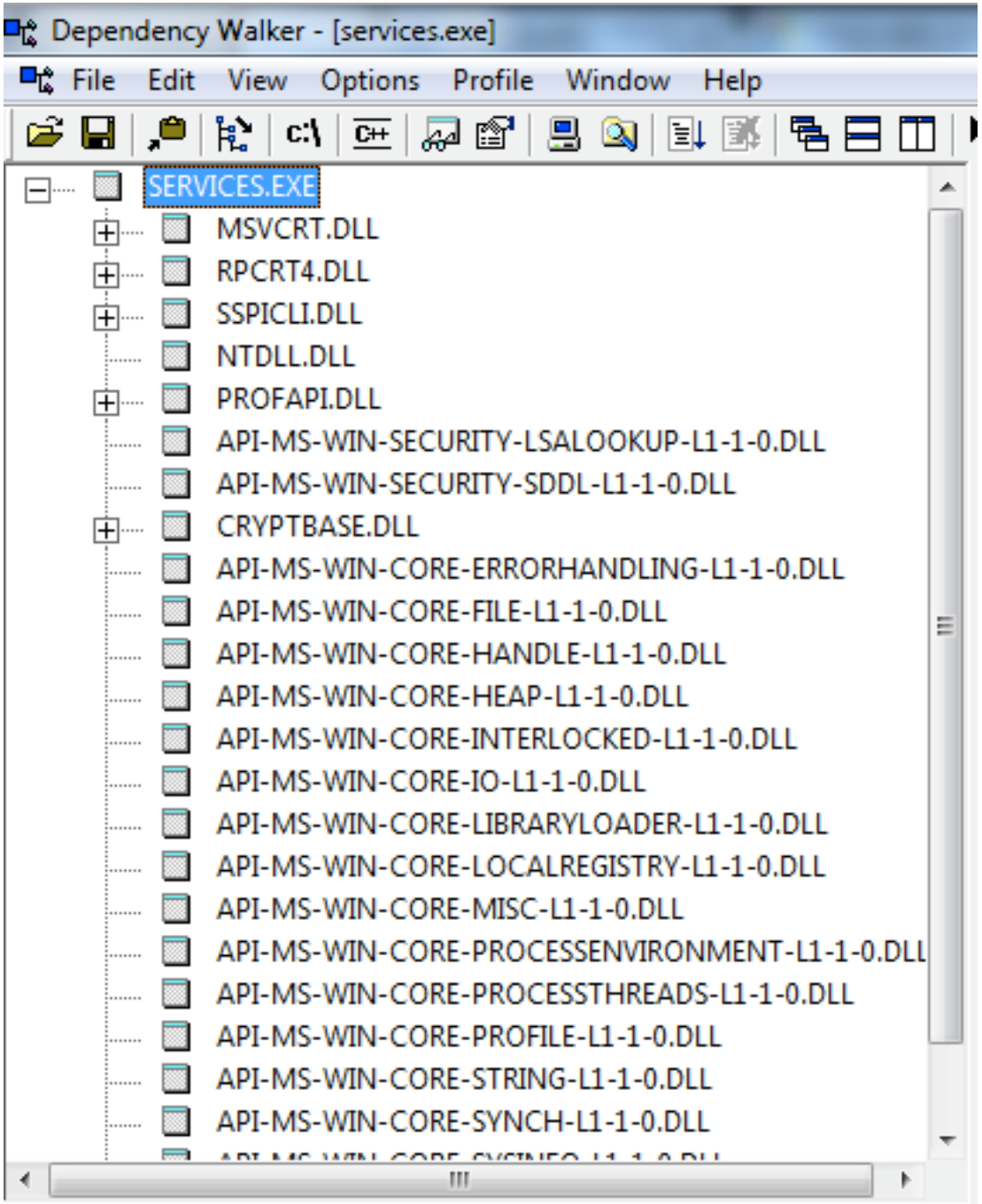


Dependency Walker

Shows Dynamically Linked Functions

- Normal programs have a lot of DLLs
- Malware often has very few DLLs

Services.exe



Imports & Exports in Dependency Walker

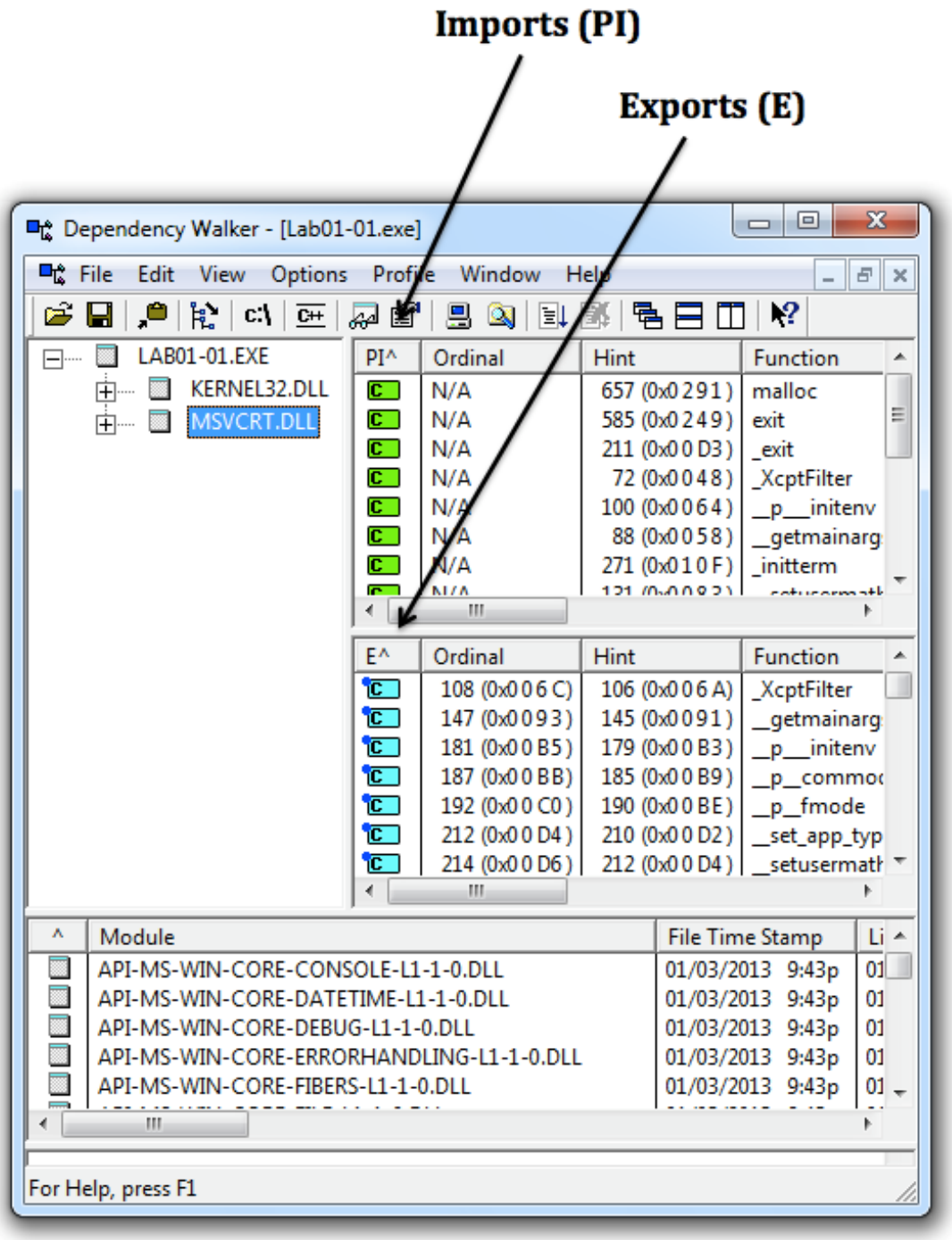


Table 2-1. Common DLLs

DLL	Description
Kernel32.dll	This is a very common DLL that contains core functionality, such as access and manipulation of memory, files, and hardware.
Advapi32.dll	This DLL provides access to advanced core Windows components such as the Service Manager and Registry.
User32.dll	This DLL contains all the user-interface components, such as buttons, scroll bars, and components for controlling and responding to user actions.
Gdi32.dll	This DLL contains functions for displaying and manipulating graphics.

Ntdll.dll	This DLL is the interface to the Windows kernel. Executables generally do not import this file directly, although it is always imported indirectly by *Kernel32.dll*. If an executable imports this file, it means that the author intended to use functionality not normally available to Windows programs. Some tasks, such as hiding functionality or manipulating processes, will use this interface.
WSock32.dll and *Ws2_32.dll*	These are networking DLLs. A program that accesses either of these most likely connects to a network or performs network-related tasks.
Wininet.dll	This DLL contains higher-level networking functions that implement protocols such as FTP, HTTP, and NTP.

Exports

- DLLs **export** functions
- EXEs **import** functions
- Both exports and imports are listed in the PE header
- The book says exports are rare in EXEs, but I see a ton of exports in innocent EXEs

Example: Keylogger

- Imports User32.dll and uses the function **SetWindowsHookEx** which is a popular way keyloggers receive keyboard inputs
- It exports **LowLevelKeyboardProc** and **LowLevelMouseProc** to send the data elsewhere
- It uses **RegisterHotKey** to define a special keystroke like Ctrl+Shift+P to harvest the collected data

Ex: A Packed Program

- Very few functions
- All you see is the unpacker

Table 2-3. DLLs and Functions Imported from PackedProgram.exe

Kernel32.dll	User32.dll
GetModuleHandleA	MessageBoxA
LoadLibraryA	
GetProcAddress	
ExitProcess	
VirtualAlloc	
VirtualFree	