Ch 2: Basic Static Analysis

Techniques

- Antivirus scanning
- Hashes
- A file’s strings, functions, and headers

Antivirus Scanning

Only a First Step

- Malware can easily change its signature and fool the antivirus
- VirusTotal is convenient, but using it may alert attackers that they've been caught
  - Link Ch 2a

VirusTotal is a free service that **analyzes suspicious files and URLs** and facilitates the quick detection of viruses, worms, trojans, and all kinds of malware.

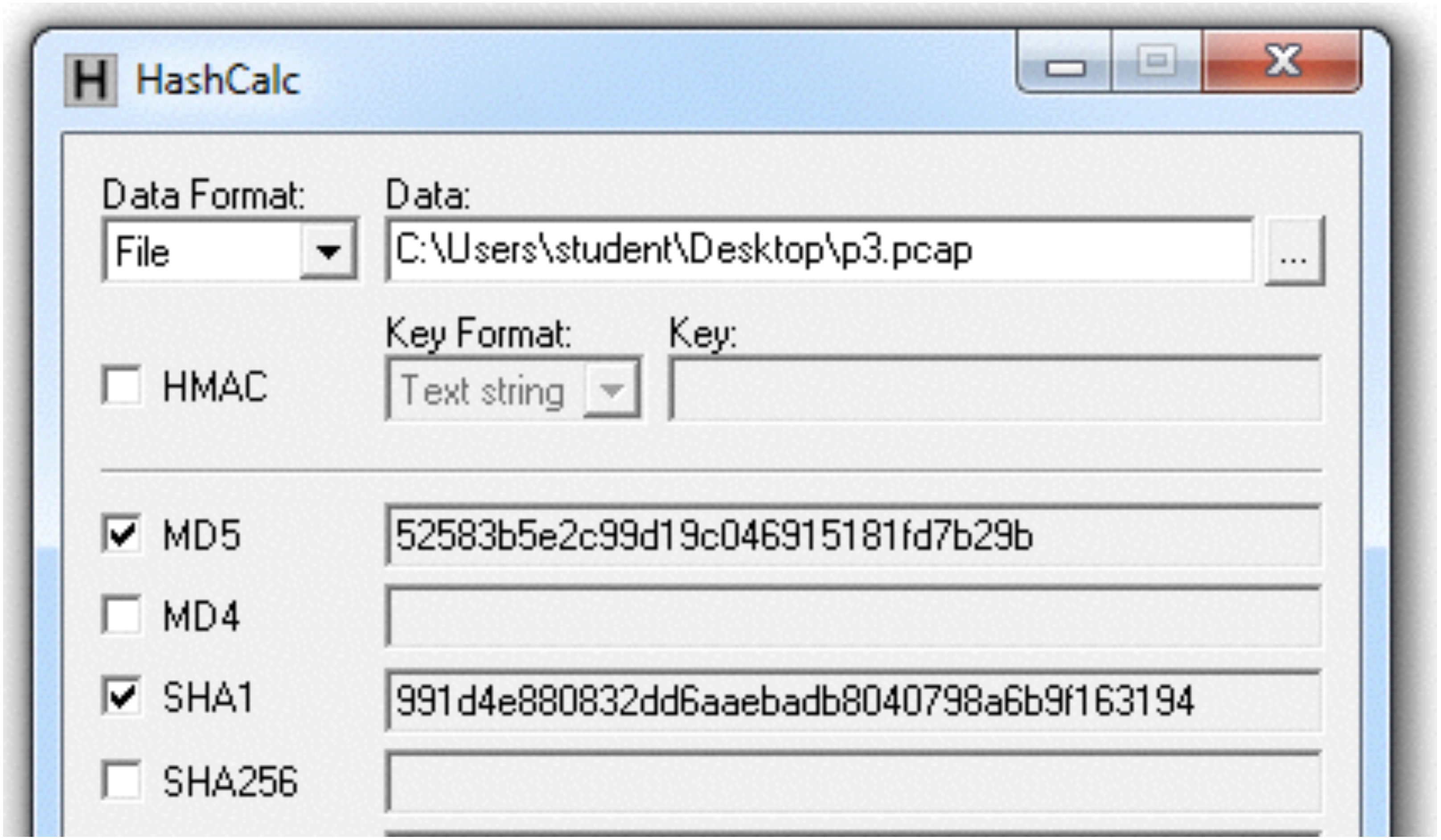
Hashing

A fingerprint for malware

Hashes

- MD5 or SHA-1
- Condenses a file of any size down to a fixed-length fingerprint
- Uniquely identifies a file well in practice
  - There are MD5 collisions but they are not common
  - Collision: two different files with the same hash

HashCalc



HashCalc

Data Format: File

Data: C:\Users\student\Desktop\p3.pcap

[ ] HMAC — Key Format: Text string — Key:

[✓] MD5 52583b5e2c99d19c046915181fd7b29b

[ ] MD4

[✓] SHA1 991d4e880832dd6aaebadb8040798a6b9f163194

[ ] SHA256

Hash Uses

- Label a malware file
- Share the hash with other analysts to identify malware
- Search the hash online to see if someone else has already identified the file

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"

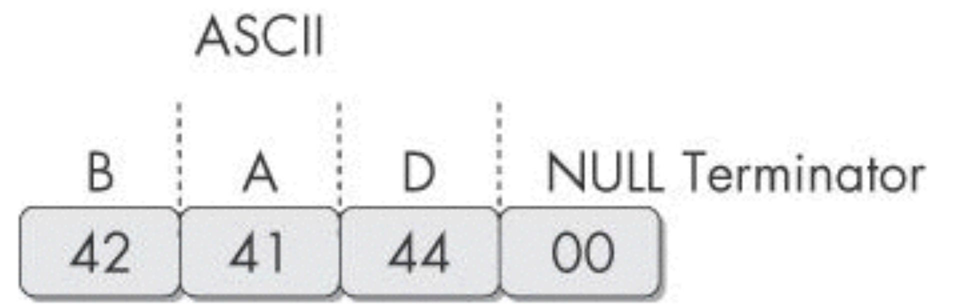


Figure 2-2. ASCII representation of the string BAD

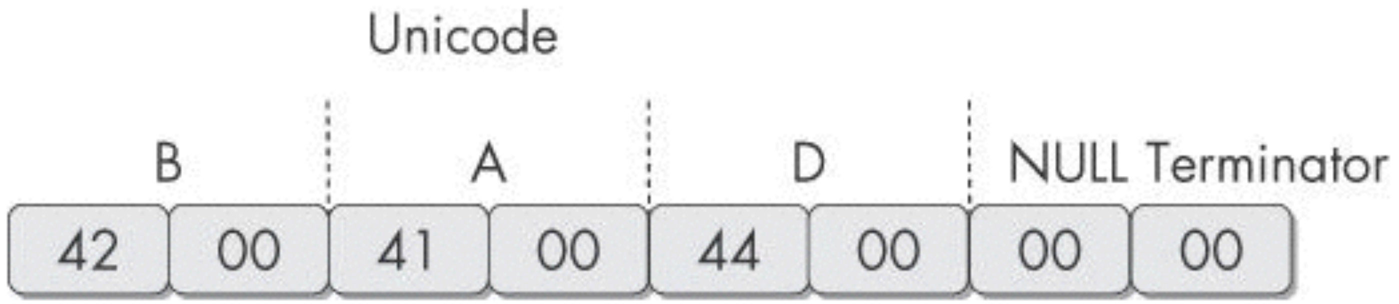


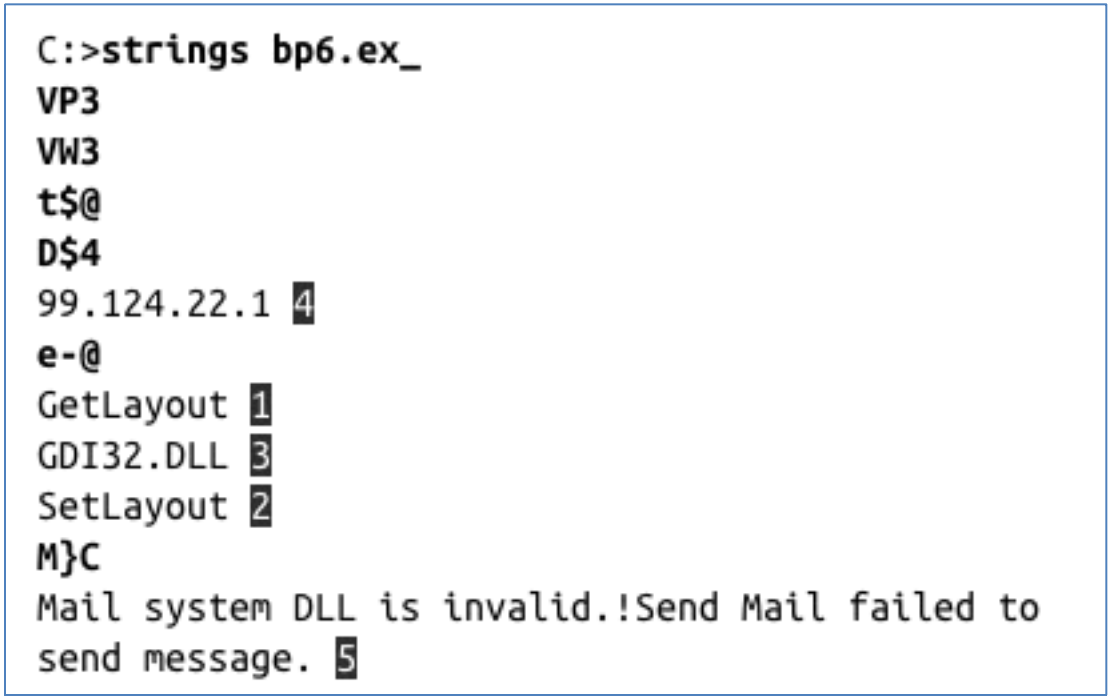
Figure 2-3. Unicode representation of the string BAD

The strings Command

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

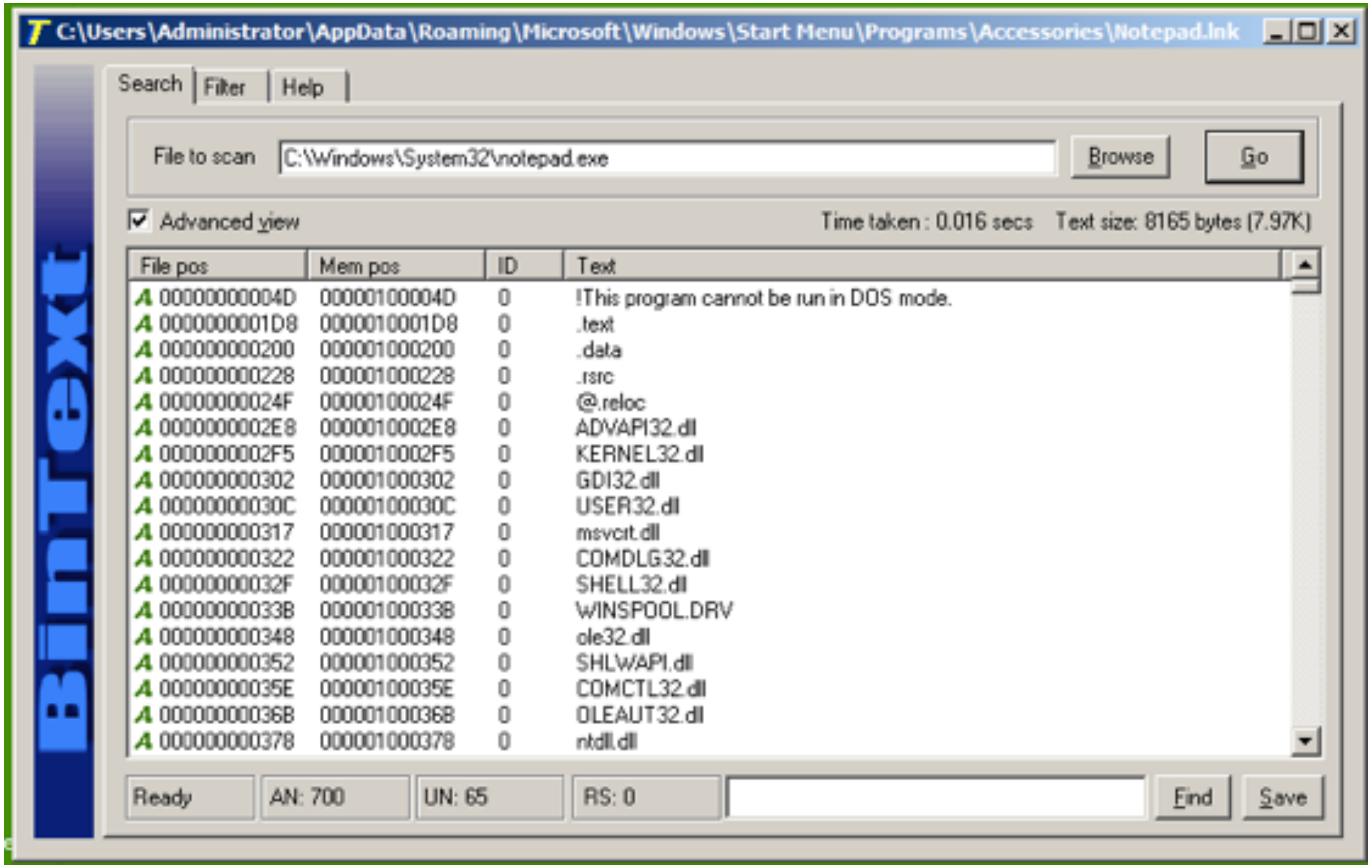
The strings Command

- 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
```

BinText



C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Accessories\Notepad.lnk

File to scan: C:\Windows\System32\notepad.exe

Advanced view: ✓

Time taken : 0.016 secs   Text size: 8165 bytes (7.97K)

File pos	Mem pos	ID	Text
A 00000000004D	00000100004D	0	!This program cannot be run in DOS mode.
A 0000000001D8	0000010001D8	0	.text
A 000000000200	000001000200	0	.data
A 000000000228	000001000228	0	.rsrc
A 00000000024F	00000100024F	0	@.reloc
A 0000000002E8	0000010002E8	0	ADVAPI32.dll
A 0000000002F5	0000010002F5	0	KERNEL32.dll
A 000000000302	000001000302	0	GDI32.dll
A 00000000030C	00000100030C	0	USER32.dll
A 000000000317	000001000317	0	msvcrt.dll
A 000000000322	000001000322	0	COMDLG32.dll
A 00000000032F	00000100032F	0	SHELL32.dll
A 00000000033B	00000100033B	0	WINSPOOL.DRV
A 000000000348	000001000348	0	ole32.dll
A 000000000352	000001000352	0	SHLWAPI.dll
A 00000000035E	00000100035E	0	COMCTL32.dll
A 00000000036B	00000100036B	0	OLEAUT32.dll
A 000000000378	000001000378	0	ntdll.dll

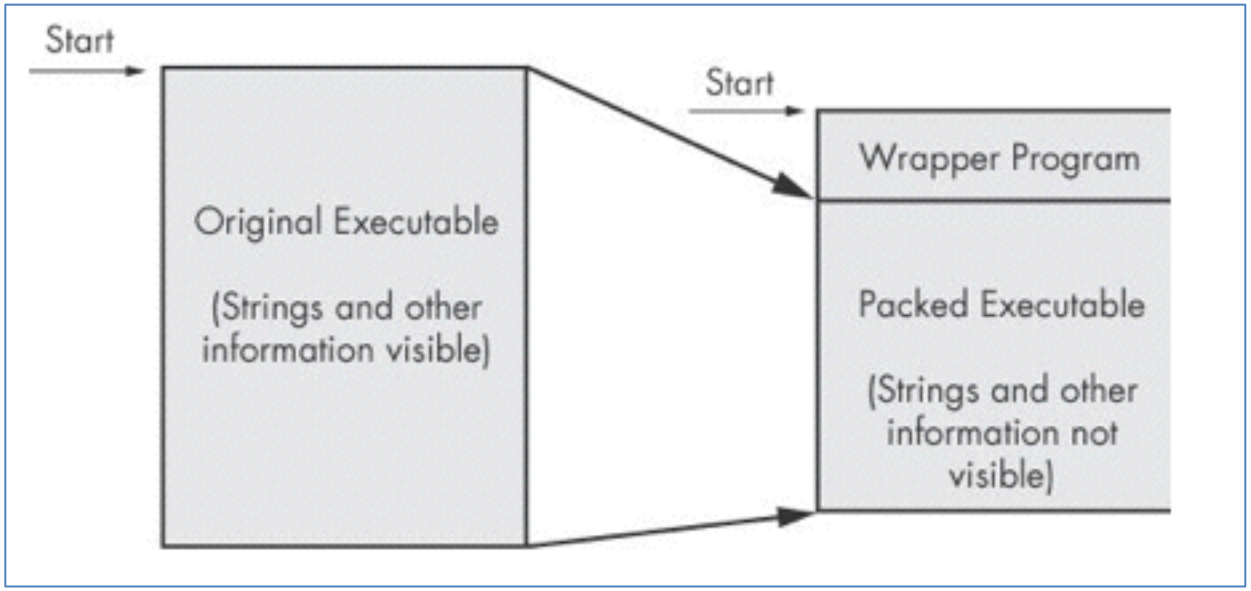
Ready | AN: 700 | UN: 65 | RS: 0

- Link Ch 2i

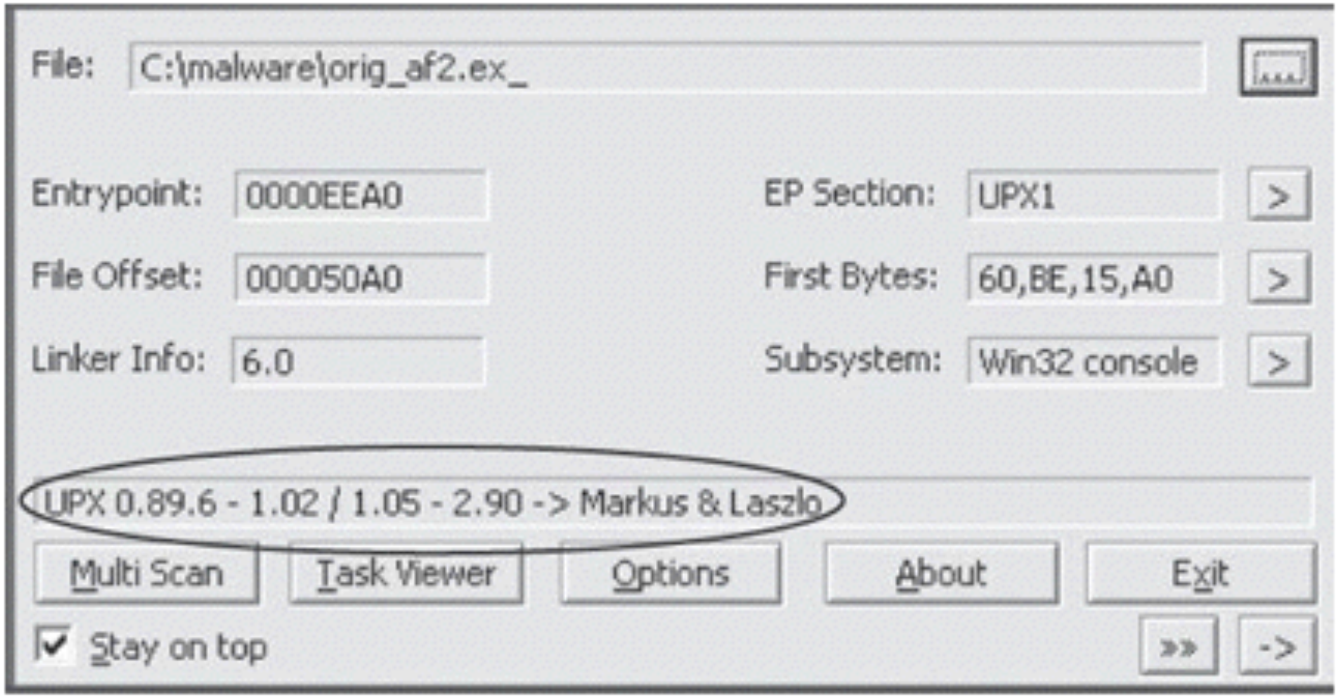
Packed and Obfuscated Malware

Packing Files

- The code is compressed, like 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


Start
Original Executable
(Strings and other information visible)
Start
Wrapper Program
Packed Executable
(Strings and other information not visible)

Detecting Packers with PEiD


File: C:\malware\orig_af2.ex_

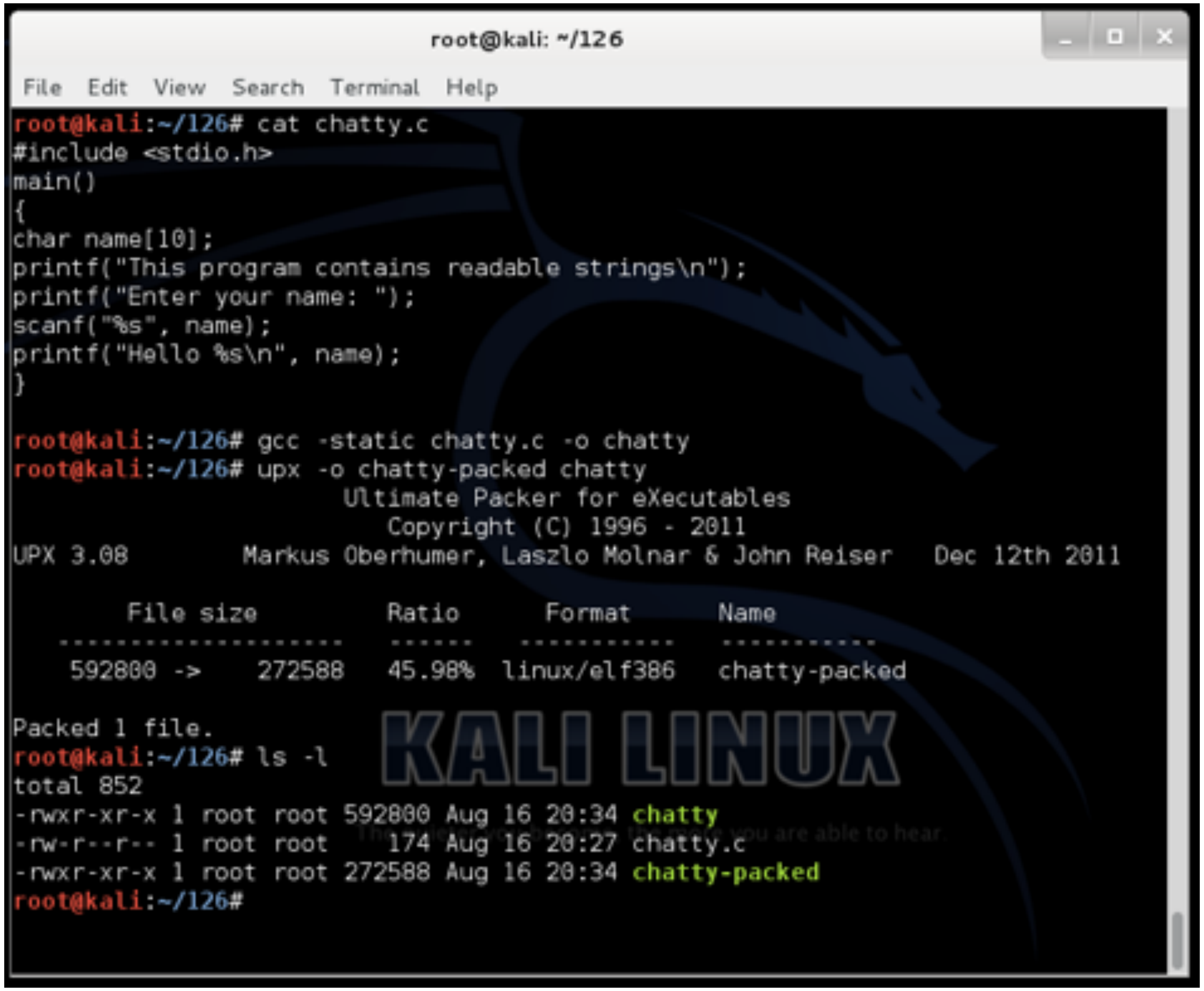
Entrypoint: 0000EEA0
File Offset: 000050A0
Linker Info: 6.0

EP Section: UPX1
First Bytes: 60,BE,15,A0
Subsystem: Win32 console

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

Figure 2-5. The PEiD program

Demo: UPX



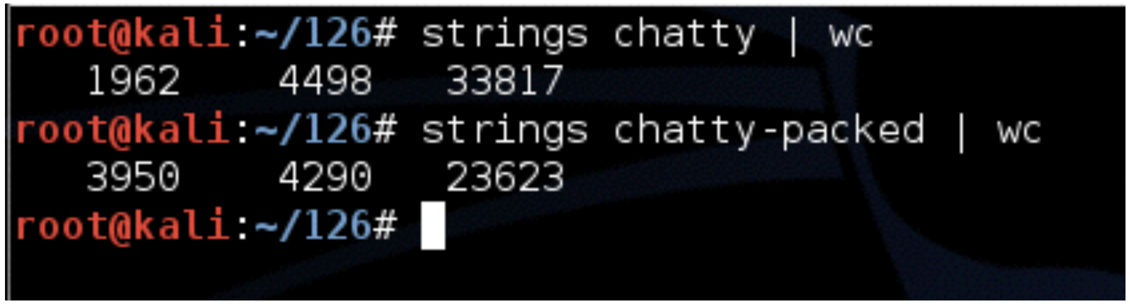
```
root@kali:~/126# cat chatty.c
#include <stdio.h>
main()
{
char name[10];
printf("This program contains readable strings\n");
printf("Enter your name: ");
scanf("%s", name);
printf("Hello %s\n", name);
}

root@kali:~/126# gcc -static chatty.c -o chatty
root@kali:~/126# upx -o chatty-packed chatty
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2011
UPX 3.08       Markus Oberhumer, Laszlo Molnar & John Reiser   Dec 12th 2011

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    592800 ->    272588   45.98%  linux/elf386   chatty-packed

Packed 1 file.
root@kali:~/126# ls -l
total 852
-rwxr-xr-x 1 root root 592800 Aug 16 20:34 chatty
-rw-r--r-- 1 root root    174 Aug 16 20:27 chatty.c
-rwxr-xr-x 1 root root 272588 Aug 16 20:34 chatty-packed
root@kali:~/126#
```

Packing Obfuscates Strings



```
root@kali:~/126# strings chatty | wc
   1962    4498   33817
root@kali:~/126# strings chatty-packed | wc
   3950    4290   23623
root@kali:~/126# 
```

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.

Portable Executable File Format

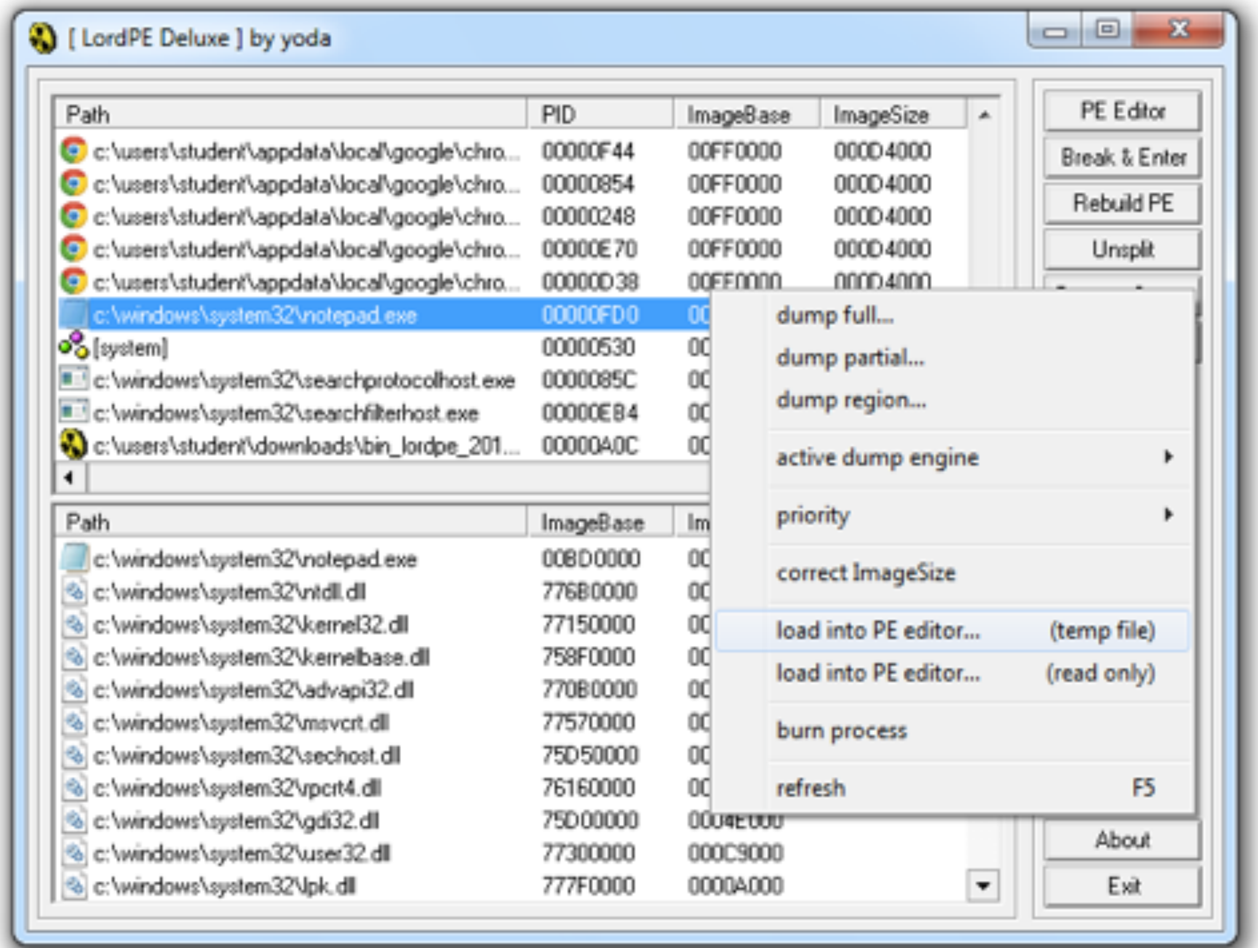
PE Files

- 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

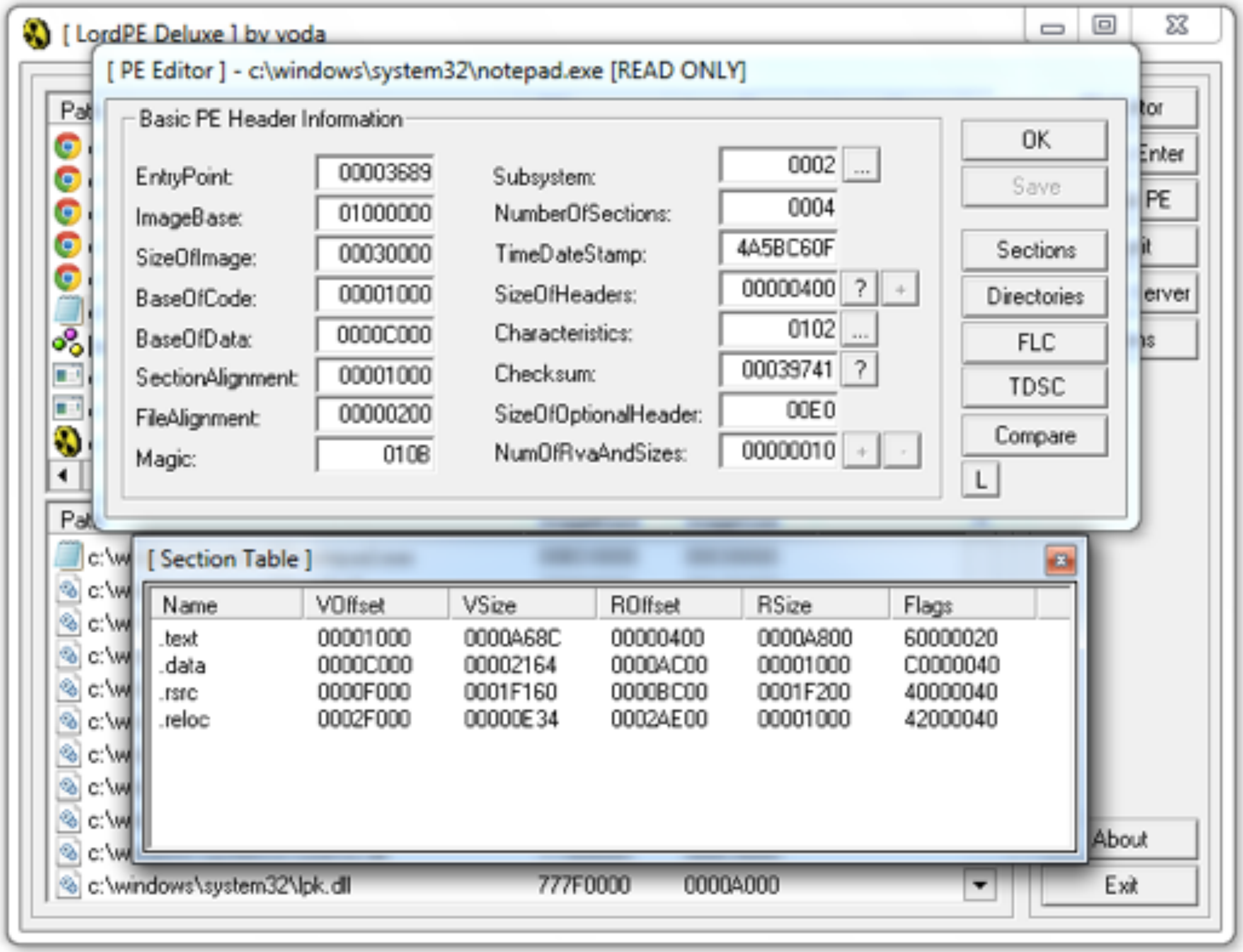
LordPE Demo



Path	PID	ImageBase	ImageSize
c:\users\student\appdata\local\google\chro...	00000F44	00FF0000	000D4000
c:\users\student\appdata\local\google\chro...	00000854	00FF0000	000D4000
c:\users\student\appdata\local\google\chro...	00000248	00FF0000	000D4000
c:\users\student\appdata\local\google\chro...	00000E70	00FF0000	000D4000
c:\users\student\appdata\local\google\chro...	00000D38	00FF0000	000D4000
c:\windows\system32\notepad.exe	00000FD0	00	
[system]	00000530	00	
c:\windows\system32\searchprotocolhost.exe	0000085C	00	
c:\windows\system32\searchfilterhost.exe	00000EB4	00	
c:\users\student\downloads\bin_lordpe_201...	00000A0C	00	

Path	ImageBase	Im
c:\windows\system32\notepad.exe	00BD0000	00
c:\windows\system32\ntdll.dll	776B0000	00
c:\windows\system32\kernel32.dll	77150000	00
c:\windows\system32\kernelbase.dll	758F0000	00
c:\windows\system32\advapi32.dll	770B0000	00
c:\windows\system32\msvcrt.dll	77570000	00
c:\windows\system32\sechost.dll	75D50000	00
c:\windows\system32\rpcrt4.dll	76160000	00
c:\windows\system32\gdi32.dll	75D00000	0004E000
c:\windows\system32\user32.dll	77300000	000C9000
c:\windows\system32\lpk.dll	777F0000	0000A000

Main Sections



[PE Editor] - c:\windows\system32\notepad.exe [READ ONLY]

Basic PE Header Information

EntryPoint:	00003689	Subsystem:	0002
ImageBase:	01000000	NumberOfSections:	0004
SizeOfImage:	00030000	TimeDateStamp:	4A5BC60F
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	0000C000	Characteristics:	0102
SectionAlignment:	00001000	Checksum:	00039741
FileAlignment:	00000200	SizeOfOptionalHeader:	00E0
Magic:	010B	NumOfRvaAndSizes:	00000010

[Section Table]

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00001000	0000A68C	00000400	0000A800	60000020
.data	0000C000	00002164	0000AC00	00001000	C0000040
.rsrc	0000F000	0001F160	0000BC00	0001F200	40000040
.reloc	0002F000	00000E34	0002AE00	00001000	42000040

c:\windows\system32\lpk.dll 777F0000 0000A000

There are a lot more sections

- But the main ones are enough for now
- Link Ch 2c

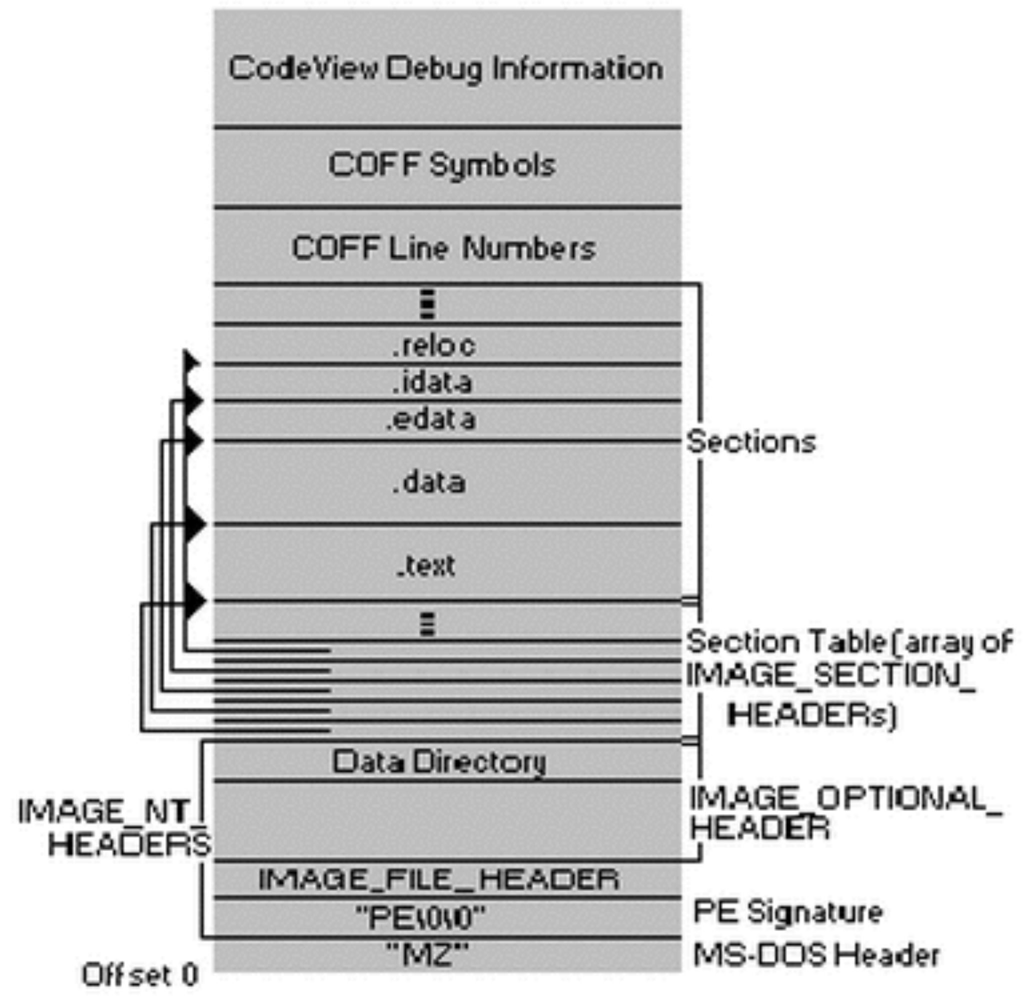


Figure 1. The PE file format

Linked Libraries and Functions

Imports

- Functions used by a program that are stored in a different program, such as library
- Connected to the main EXE by **Linking**
- Can be linked three ways
  - **Statically**
  - At **Runtime**
  - **Dynamically**

Static Linking

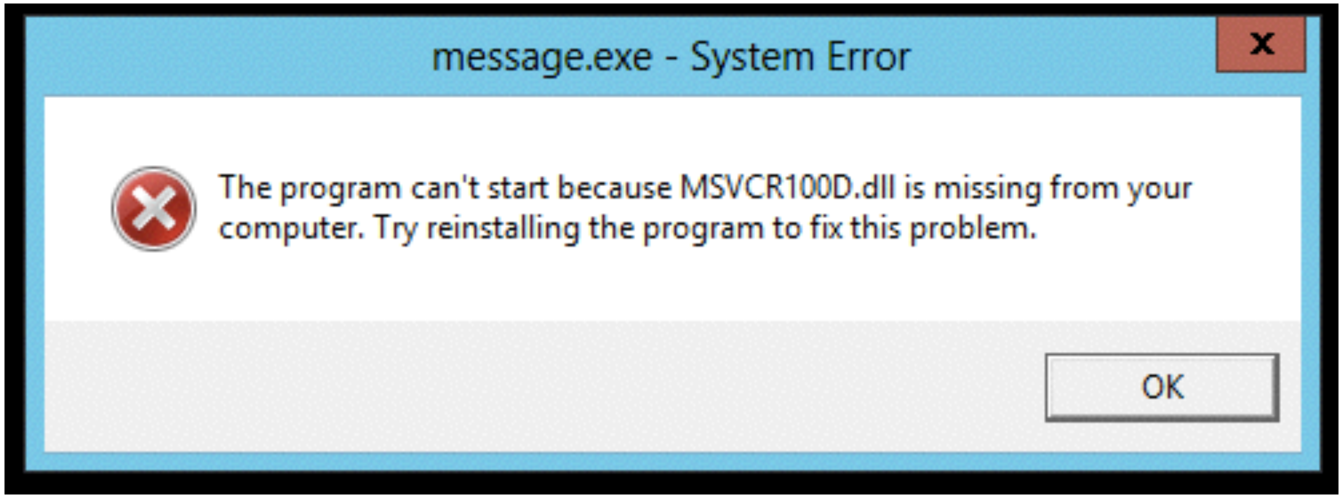
- Rarely used for Windows executables
- Common in Unix and Linux
- All code from the library is copied into the executable
- Makes executable large in size

Runtime Linking

- Unpopular in friendly programs
- Common in malware, especially packed or obfuscated malware
- Connect to libraries only when needed, not when the program starts
- Most commonly done with the **LoadLibrary** and **GetProcAddress** functions

Dynamic Linking

- Most common method
- Host OS searches for necessary libraries when the program is loaded



The program can't start because MSVCR100D.dll is missing from your computer. Try reinstalling the program to fix this problem.

Clues in Libraries

- The PE header lists every library and function that will be loaded
- Their names can reveal what the program does
- **URLDownloadToFile** indicates that the program downloads something