Basic Static Techniques

CYBR473 – Malware and Reverse Engineering (2024/T1)

Lecturers: Arman Khouzani (course coordinator), Alvin Valera

Victoria University of Wellington – School of Engineering and Computer Science

Reading Material

- **Part I: Basic Analysis**
  - Ch.1: Basic Static Techniques

"Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software", Michael Sikorski and Andrew Honig, 2012

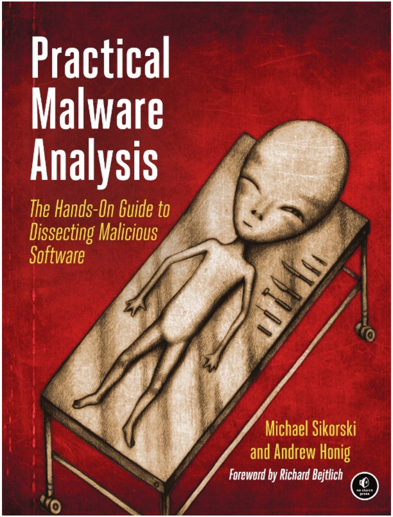

Practical Malware Analysis
The Hands-On Guide to Dissecting Malicious Software
Michael Sikorski and Andrew Honig
Foreword by Richard Bejtlich

Table of contents

1. **Antivirus Scanning**

    Hashing

2. **Finding Strings**

3. ***Packed* and *Obfuscated* Malware**

4. **Portable Executable (PE) File Format**

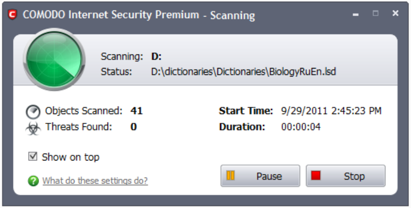
5. **Linked Libraries and Functions**

    Dependency Walker

6. **The PE File *Headers* and *Sections***

Basic Static Techniques

Antivirus scanning



COMODO Internet Security Premium - Scanning

Scanning: D:

Status: D:\dictionaries\Dictionaries\BiologyRuEn.lsd

Objects Scanned: 41

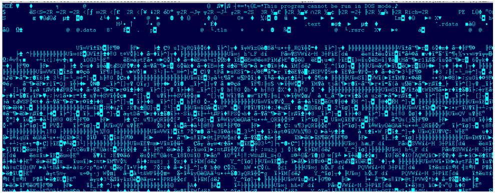
Threats Found: 0

Start Time: 9/29/2011 2:45:23 PM

Duration: 00:00:04

☑ Show on top

What do these settings do?



A file’s strings, functions, and headers

Hashes

Antivirus Scanning

Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

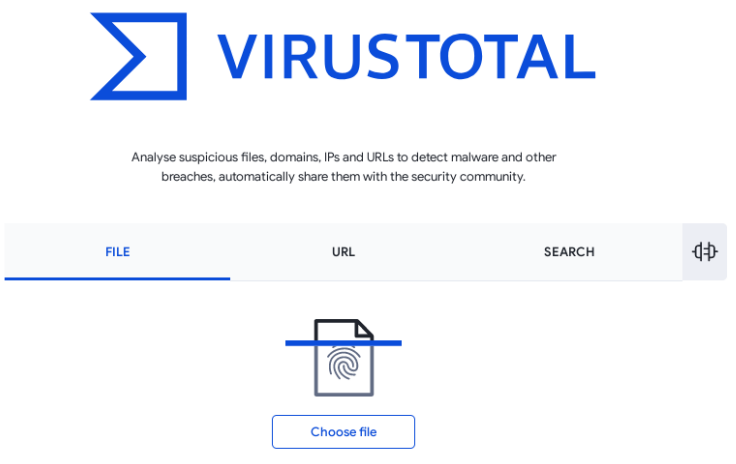
Linked Libraries and Functions

The PE File *Headers* and *Sections*

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.



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

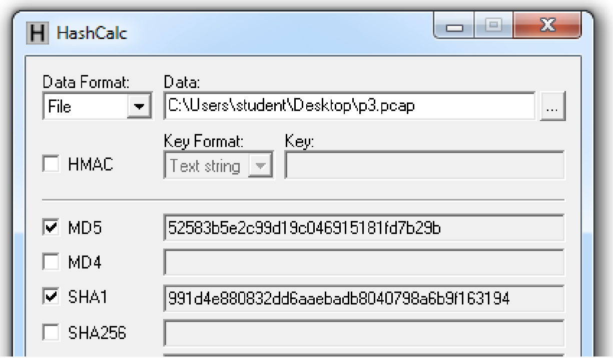
Hashes

MD5 or SHA-1 (or SHA-2).

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

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

Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

Linked Libraries and Functions

The PE File *Headers* and *Sections*

Strings

Any sequence of printable characters is a **string**.

Strings are terminated by a **null** (`0x00`).

ASCII characters are 8 bits long

- also called ANSI

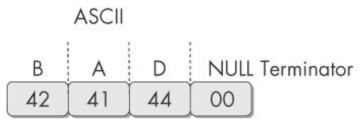


Figure 2-2. ASCII representation of the string BAD

Unicode characters are longer

- Microsoft uses UTF-16 and calls them “wide characters”

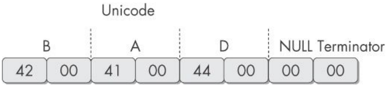
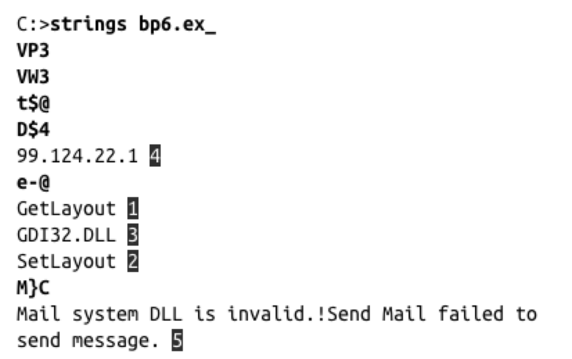


Figure 2-3. Unicode representation of the string BAD

The strings Command

The strings Command: Native in Linux, also available for Windows.

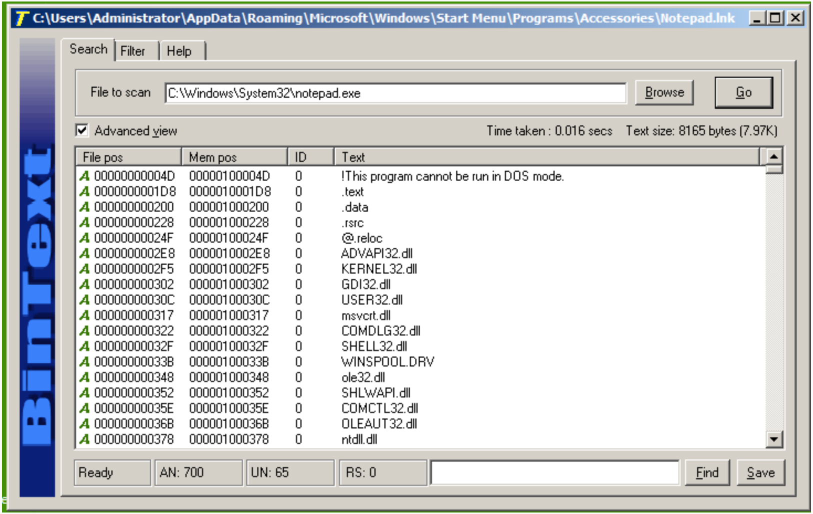
Finds all strings in a file 3 or more characters long. E.g.:



```
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
```

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

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

Packed and *Obfuscated* Malware

Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

Linked Libraries and Functions

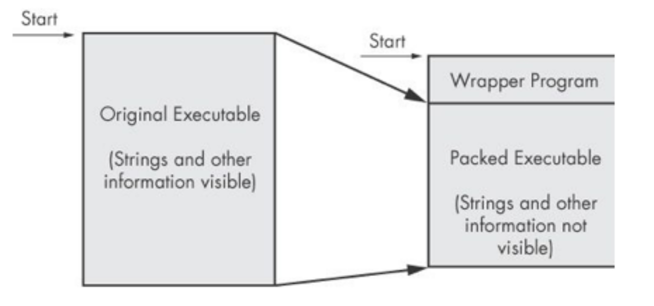
The PE File *Headers* and *Sections*

Packing Files

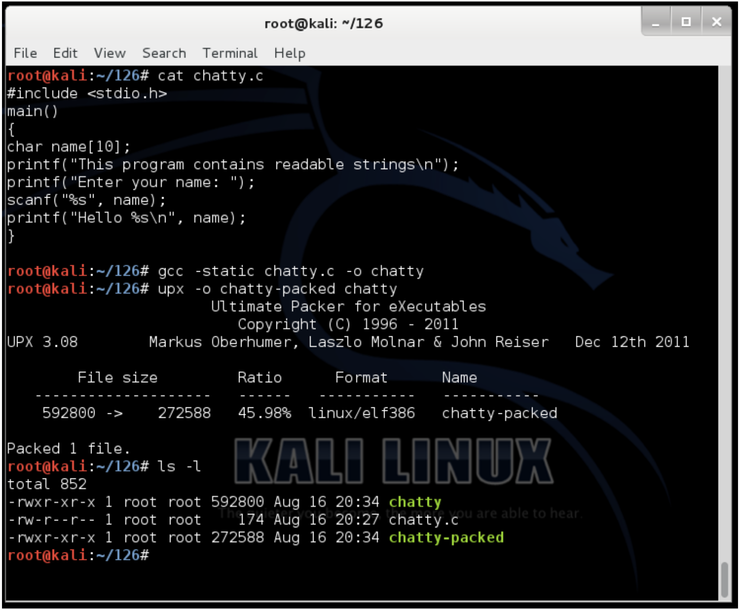
The code is compressed, like a Zip file.

This makes the strings and instructions unreadable.

All you'll see is the **wrapper** – small code that unpacks the file when it is run.


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

UPX – “the Ultimate Packer for eXecutables”


root@kali: ~/126
File Edit View Search Terminal Help
KALI LINUX
The quieter you become the more you are able to hear.

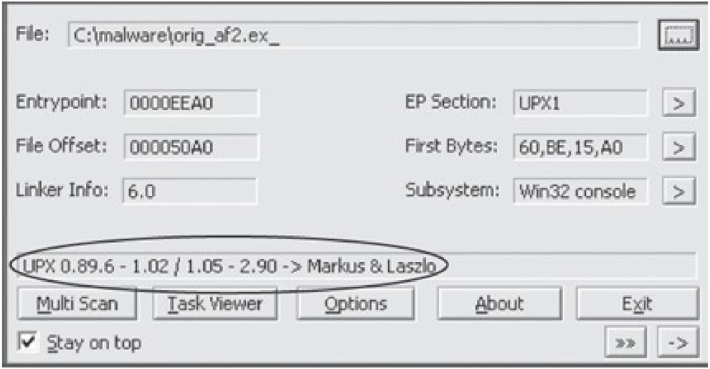
```
root@kali:~/126# cat chatty.c
#include <stdio.h>
main()
{
char name[10];
printf("This program contains readable strings\n");
printf("Enter your name: ");
scanf("%s", name);
printf("Hello %s\n", name);
}

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#
```

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

☑ Stay on top

Figure 2-5. The PEiD program

Note

Many `PEiD` plug-ins will **run** the malware executable without warning!

You need to set up a safe environment for running malware (next week)

Like all programs, especially those used for malware analysis, *PEiD can be subject to vulnerabilities*.

- `PEiD` version 0.92 contains 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 (PE) File Format

Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

Linked Libraries and Functions

The PE File *Headers* and *Sections*

PE Files

PE Files:

- A data structure that contains the information necessary for Windows to load the file.

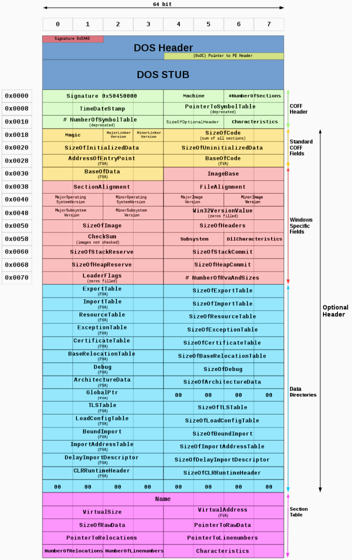
Used by Windows executable files, object code, and DLLs.

Almost every file executed on Windows is in PE format.

PE Header

PE Header:

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



64 bit

	0	1	2	3	4	5	6	7
	Signature 0x5A4D DOS Header				(0x3C) Pointer to PE Header			
	DOS STUB							
0x0000	Signature 0x50450000				Machine		#NumberOfSections	
0x0008	TimeDateStamp				PointerToSymbolTable (deprecated)			
0x0010	# NumberOfSymbolTable (deprecated)				SizeOfOptionalHeader		Characteristics	
0x0018	Magic		MajorLinker Version	MinorLinker Version	SizeOfCode (sum of all sections)			
0x0020	SizeOfInitializedData				SizeOfUninitializedData			
0x0028	AddressOfEntryPoint (RVA)				BaseOfCode (RVA)			
0x0030	BaseOfData (RVA)				ImageBase			
0x0038	SectionAlignment				FileAlignment			
0x0040	MajorOperating SystemVersion		MinorOperating SystemVersion		MajorImage Version		MinorImage Version	
0x0048	MajorSubsystem Version		MinorSubsystem Version		Win32VersionValue (zeros filled)			
0x0050	SizeOfImage				SizeOfHeaders			
0x0058	CheckSum (images not checked)				Subsystem		DllCharacteristics	
0x0060	SizeOfStackReserve				SizeOfStackCommit			
0x0068	SizeOfHeapReserve				SizeOfHeapCommit			
0x0070	LoaderFlags (zeros filled)				# NumberOfRvaAndSizes			
	ExportTable (RVA)				SizeOfExportTable			
	ImportTable (RVA)				SizeOfImportTable			
	ResourceTable (RVA)				SizeOfResourceTable			
	ExceptionTable (RVA)				SizeOfExceptionTable			
	CertificateTable (RVA)				SizeOfCertificateTable			
	BaseRelocationTable (RVA)				SizeOfBaseRelocationTable			
	Debug (RVA)				SizeOfDebug			
	ArchitectureData (RVA)				SizeOfArchitectureData			
	GlobalPtr (rva)				00	00	00	00
	TLSTable (RVA)				SizeOfTLSTable			
	LoadConfigTable (RVA)				SizeOfLoadConfigTable			
	BoundImport (RVA)				SizeOfBoundImport			
	ImportAddressTable (RVA)				SizeOfImportAddressTable			
	DelayImportDescriptor (RVA)				SizeOfDelayImportDescriptor			
	CLRRuntimeHeader (RVA)				SizeOfCLRRuntimeHeader			
	00	00	00	00	00	00	00	00
	Name							
	VirtualSize				VirtualAddress (RVA)			
	SizeOfRawData				PointerToRawData			
	PointerToRelocations				PointerToLinenumbers			
	NumberOfRelocations		NumberOfLinenumbers		Characteristics			

COFF Header

Standard COFF Fields

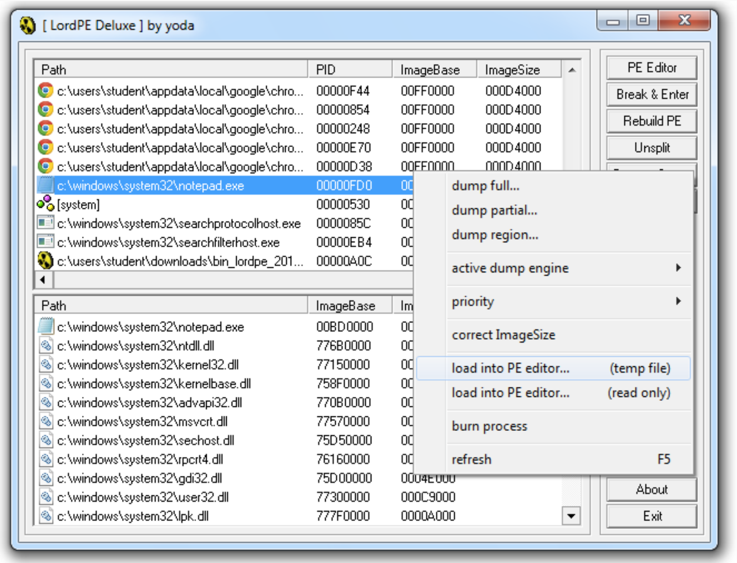
Windows Specific Fields

Data Directories

Section Table

Optional Header

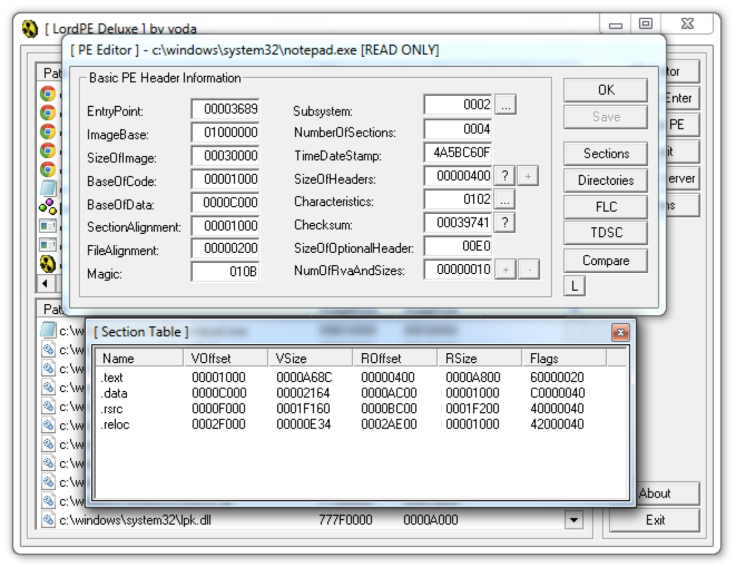
LordPE Demo



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

LordPE Demo: Main Sections



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

Basic PE Header Information

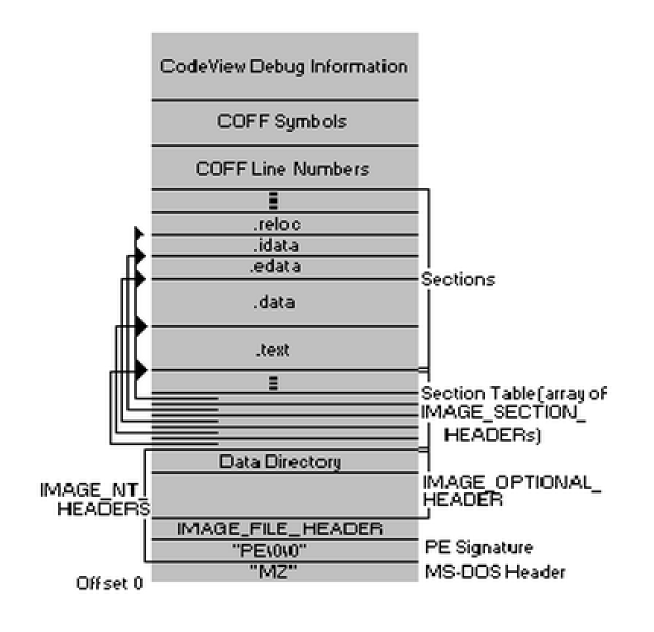
EntryPoint:	00003689	Subsystem:	0002
ImageBase:	01000000	NumberOfSections:	0004
SizeOfImage:	00030000	TimeDateStamp:	4A5BC60F
BaseOfCode:	00001000	SizeOfHeaders:	00000400
BaseOfData:	0000C000	Characteristics:	0102
SectionAlignment:	00001000	Checksum:	00039741
FileAlignment:	00000200	SizeOfOptionalHeader: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



Linked Libraries and Functions

Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

Linked Libraries and Functions

The PE File *Headers* and *Sections*

Imports

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

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

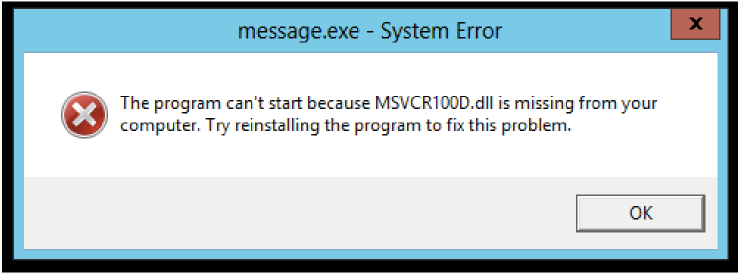
Runtime Linking:

- 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

Dynamic Linking:

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



message.exe - System Error

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

They can reveal what the program does

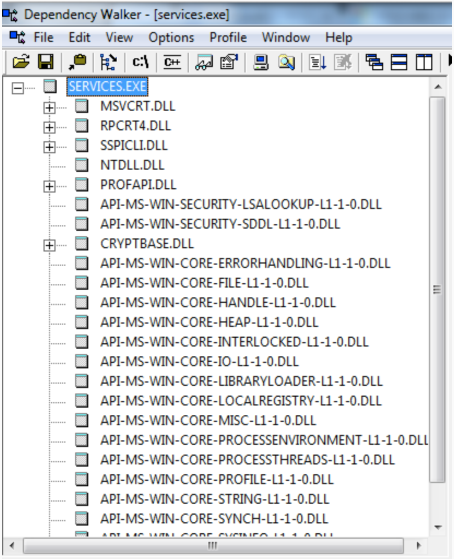
- e.g.: URLDownloadToFile indicates that the program downloads something

Dependency Walker

`Dependency Walker`: **Shows Dynamically Linked Functions**

- Normal programs have many DLLs.
- Malware often has very few DLLs.

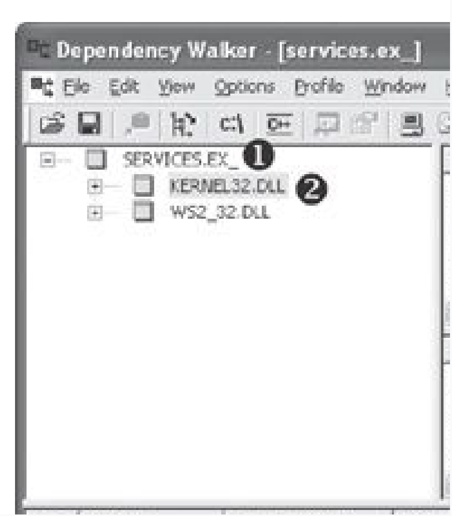
Dependency Walker example: `Services.exe`



Dependency Walker - [services.exe]
File Edit View Options Profile Window Help

```
SERVICES.EXE
    MSVCRT.DLL
    RPCRT4.DLL
    SSPICLI.DLL
    NTDLL.DLL
    PROFAPI.DLL
    API-MS-WIN-SECURITY-LSALOOKUP-L1-1-0.DLL
    API-MS-WIN-SECURITY-SDDL-L1-1-0.DLL
    CRYPTBASE.DLL
    API-MS-WIN-CORE-ERRORHANDLING-L1-1-0.DLL
    API-MS-WIN-CORE-FILE-L1-1-0.DLL
    API-MS-WIN-CORE-HANDLE-L1-1-0.DLL
    API-MS-WIN-CORE-HEAP-L1-1-0.DLL
    API-MS-WIN-CORE-INTERLOCKED-L1-1-0.DLL
    API-MS-WIN-CORE-IO-L1-1-0.DLL
    API-MS-WIN-CORE-LIBRARYLOADER-L1-1-0.DLL
    API-MS-WIN-CORE-LOCALREGISTRY-L1-1-0.DLL
    API-MS-WIN-CORE-MISC-L1-1-0.DLL
    API-MS-WIN-CORE-PROCESSENVIRONMENT-L1-1-0.DLL
    API-MS-WIN-CORE-PROCESSTHREADS-L1-1-0.DLL
    API-MS-WIN-CORE-PROFILE-L1-1-0.DLL
    API-MS-WIN-CORE-STRING-L1-1-0.DLL
    API-MS-WIN-CORE-SYNCH-L1-1-0.DLL
    API-MS-WIN-CORE-SYSINFO-L1-1-0.DLL
```

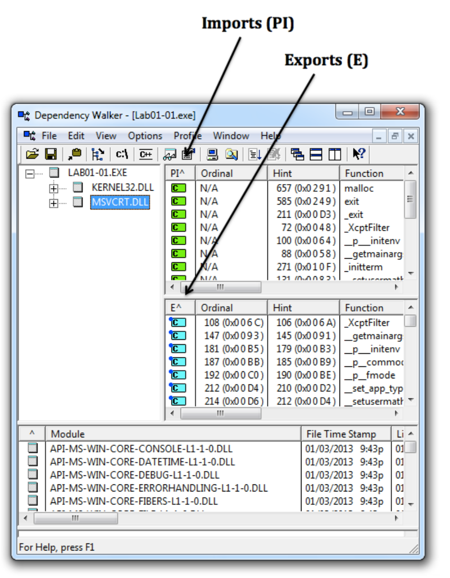
Dependency Walker example: `Services.ex_` (malware)



Dependency Walker - [services.ex_]

SERVICES.EX_ ❶
KERNEL32.DLL ❷
WS2_32.DLL

Imports & Exports in Dependency Walker



Imports (PI)

Exports (E)

LAB01-01.EXE
- KERNEL32.DLL
- MSVCRT.DLL

PI^	Ordinal	Hint	Function
C	N/A	657 (0x0291)	malloc
C	N/A	585 (0x0249)	exit
C	N/A	211 (0x00D3)	_exit
C	N/A	72 (0x0048)	_XcptFilter
C	N/A	100 (0x0064)	__p__initenv
C	N/A	88 (0x0058)	__getmainarg
C	N/A	271 (0x010F)	_initterm

E^	Ordinal	Hint	Function
C	108 (0x006C)	106 (0x006A)	_XcptFilter
C	147 (0x0093)	145 (0x0091)	__getmainarg
C	181 (0x00B5)	179 (0x00B3)	__p__initenv
C	187 (0x00BB)	185 (0x00B9)	__p__commo
C	192 (0x00C0)	190 (0x00BE)	__p__fmode
C	212 (0x00D4)	210 (0x00D2)	__set_app_typ
C	214 (0x00D6)	212 (0x00D4)	__setusermatl

^	Module	File Time Stamp	Li
	API-MS-WIN-CORE-CONSOLE-L1-1-0.DLL	01/03/2013 9:43p	01
	API-MS-WIN-CORE-DATETIME-L1-1-0.DLL	01/03/2013 9:43p	01
	API-MS-WIN-CORE-DEBUG-L1-1-0.DLL	01/03/2013 9:43p	01
	API-MS-WIN-CORE-ERRORHANDLING-L1-1-0.DLL	01/03/2013 9:43p	01
	API-MS-WIN-CORE-FIBERS-L1-1-0.DLL	01/03/2013 9:43p	01

For Help, press F1

Common DLLs

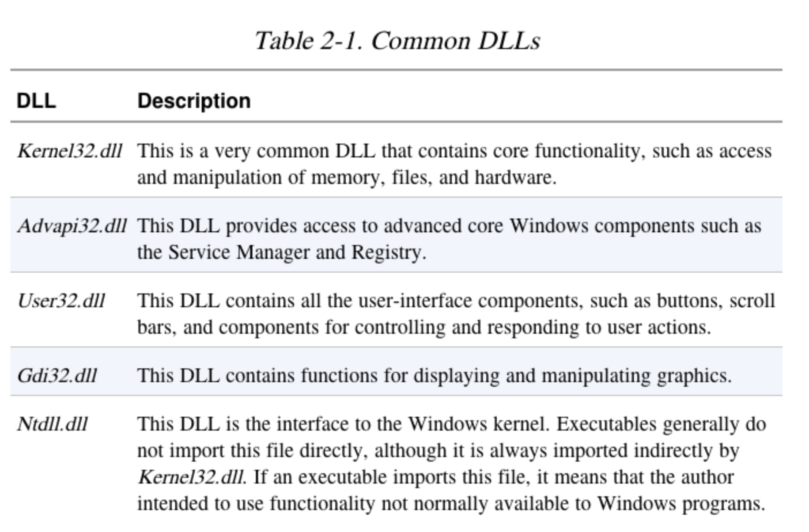
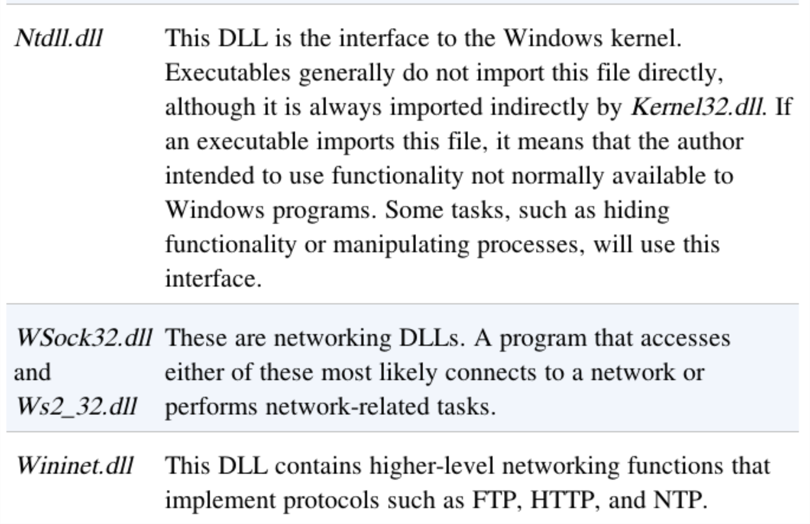


Table 2-1. Common DLLs

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.

Common DLLs (cont.)



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.

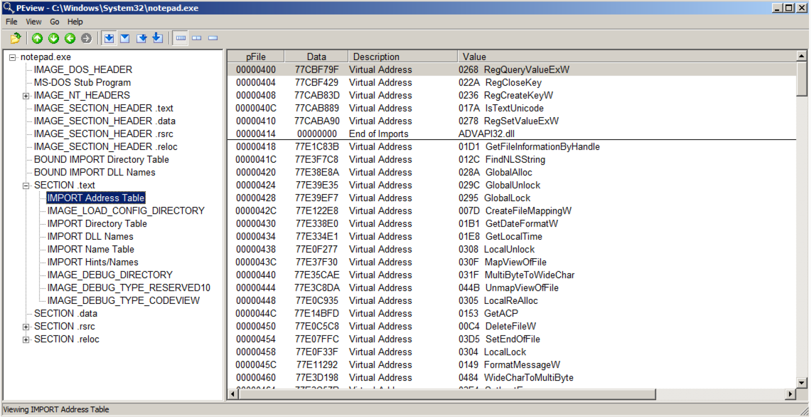
Exports

DLLs **export** functions

EXEs **import** functions

Both exports and imports are listed in the PE header

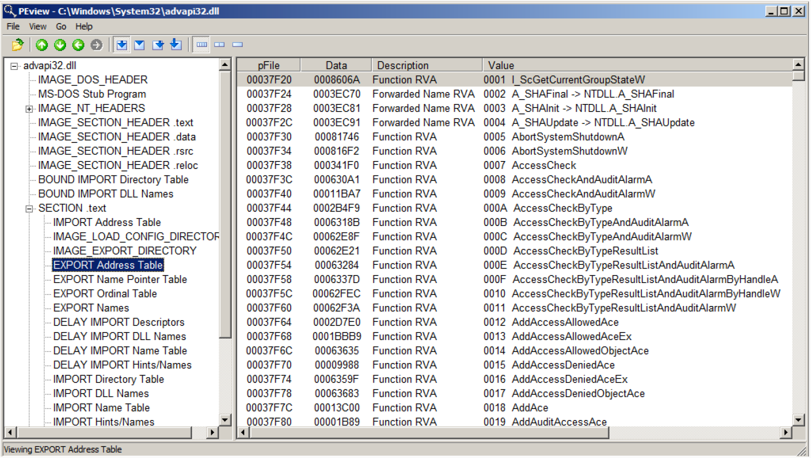
Notepad.exe



pFile	Data	Description	Value
00000400	77CBF79F	Virtual Address	0268 RegQueryValueExW
00000404	77CBF429	Virtual Address	022A RegCloseKey
00000408	77CAB83D	Virtual Address	0236 RegCreateKeyW
0000040C	77CAB889	Virtual Address	017A IsTextUnicode
00000410	77CABA90	Virtual Address	0278 RegSetValueExW
00000414	00000000	End of Imports	ADVAPI32.dll
00000418	77E1C83B	Virtual Address	01D1 GetFileInformationByHandle
0000041C	77E3F7C8	Virtual Address	012C FindNLSString
00000420	77E38E8A	Virtual Address	028A GlobalAlloc
00000424	77E39E35	Virtual Address	029C GlobalUnlock
00000428	77E39EF7	Virtual Address	0295 GlobalLock
0000042C	77E122E8	Virtual Address	007D CreateFileMappingW
00000430	77E338E0	Virtual Address	01B1 GetDateFormatW
00000434	77E334E1	Virtual Address	01E8 GetLocalTime
00000438	77E0F277	Virtual Address	0308 LocalUnlock
0000043C	77E37F30	Virtual Address	030F MapViewOfFile
00000440	77E35CAE	Virtual Address	031F MultiByteToWideChar
00000444	77E3C8DA	Virtual Address	044B UnmapViewOfFile
00000448	77E0C935	Virtual Address	0305 LocalReAlloc
0000044C	77E14BFD	Virtual Address	0153 GetACP
00000450	77E0C5C8	Virtual Address	00C4 DeleteFileW
00000454	77E07FFC	Virtual Address	03D5 SetEndOfFile
00000458	77E0F33F	Virtual Address	0304 LocalLock
0000045C	77E11292	Virtual Address	0149 FormatMessageW
00000460	77E3D198	Virtual Address	0484 WideCharToMultiByte

Viewing IMPORT Address Table

Advapi32.dll

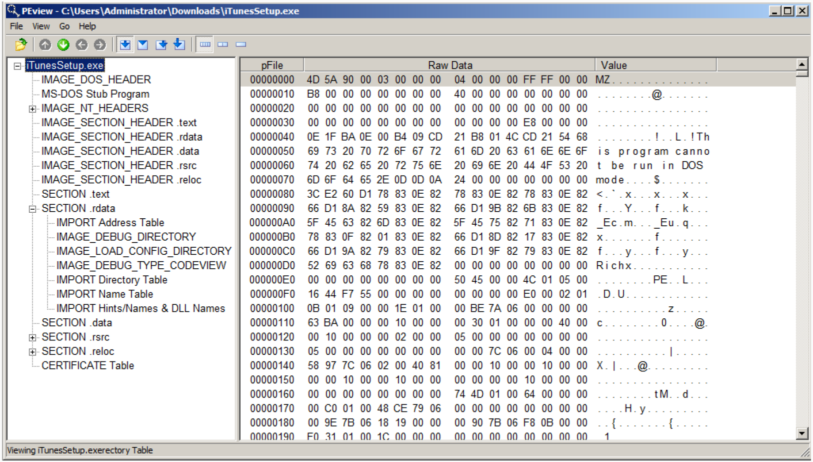


- advapi32.dll
  - IMAGE_DOS_HEADER
  - MS-DOS Stub Program
  - IMAGE_NT_HEADERS
  - IMAGE_SECTION_HEADER .text
  - IMAGE_SECTION_HEADER .data
  - IMAGE_SECTION_HEADER .rsrc
  - IMAGE_SECTION_HEADER .reloc
  - BOUND IMPORT Directory Table
  - BOUND IMPORT DLL Names
  - SECTION .text
    - IMPORT Address Table
    - IMAGE_LOAD_CONFIG_DIRECTOF
    - IMAGE_EXPORT_DIRECTORY
    - EXPORT Address Table
    - EXPORT Name Pointer Table
    - EXPORT Ordinal Table
    - EXPORT Names
    - DELAY IMPORT Descriptors
    - DELAY IMPORT DLL Names
    - DELAY IMPORT Name Table
    - DELAY IMPORT Hints/Names
    - IMPORT Directory Table
    - IMPORT DLL Names
    - IMPORT Name Table
    - IMPORT Hints/Names

pFile	Data	Description	Value
00037F20	0008606A	Function RVA	0001 I_ScGetCurrentGroupStateW
00037F24	0003EC70	Forwarded Name RVA	0002 A_SHAFinal -> NTDLL.A_SHAFinal
00037F28	0003EC81	Forwarded Name RVA	0003 A_SHAInit -> NTDLL.A_SHAInit
00037F2C	0003EC91	Forwarded Name RVA	0004 A_SHAUpdate -> NTDLL.A_SHAUpdate
00037F30	00081746	Function RVA	0005 AbortSystemShutdownA
00037F34	000816F2	Function RVA	0006 AbortSystemShutdownW
00037F38	000341F0	Function RVA	0007 AccessCheck
00037F3C	000630A1	Function RVA	0008 AccessCheckAndAuditAlarmA
00037F40	00011BA7	Function RVA	0009 AccessCheckAndAuditAlarmW
00037F44	0002B4F9	Function RVA	000A AccessCheckByType
00037F48	0006318B	Function RVA	000B AccessCheckByTypeAndAuditAlarmA
00037F4C	00062E8F	Function RVA	000C AccessCheckByTypeAndAuditAlarmW
00037F50	00062E21	Function RVA	000D AccessCheckByTypeResultList
00037F54	00063284	Function RVA	000E AccessCheckByTypeResultListAndAuditAlarmA
00037F58	0006337D	Function RVA	000F AccessCheckByTypeResultListAndAuditAlarmByHandleA
00037F5C	00062FEC	Function RVA	0010 AccessCheckByTypeResultListAndAuditAlarmByHandleW
00037F60	00062F3A	Function RVA	0011 AccessCheckByTypeResultListAndAuditAlarmW
00037F64	0002D7E0	Function RVA	0012 AddAccessAllowedAce
00037F68	0001BBB9	Function RVA	0013 AddAccessAllowedAceEx
00037F6C	00063635	Function RVA	0014 AddAccessAllowedObjectAce
00037F70	00009988	Function RVA	0015 AddAccessDeniedAce
00037F74	0006359F	Function RVA	0016 AddAccessDeniedAceEx
00037F78	00063683	Function RVA	0017 AddAccessDeniedObjectAce
00037F7C	00013C00	Function RVA	0018 AddAce
00037F80	00001B89	Function RVA	0019 AddAuditAccessAce

Viewing EXPORT Address Table

iTunesSetup.exe



```
iTunesSetup.exe
    IMAGE_DOS_HEADER
    MS-DOS Stub Program
    IMAGE_NT_HEADERS
    IMAGE_SECTION_HEADER .text
    IMAGE_SECTION_HEADER .rdata
    IMAGE_SECTION_HEADER .data
    IMAGE_SECTION_HEADER .rsrc
    IMAGE_SECTION_HEADER .reloc
    SECTION .text
    SECTION .rdata
        IMPORT Address Table
        IMAGE_DEBUG_DIRECTORY
        IMAGE_LOAD_CONFIG_DIRECTORY
        IMAGE_DEBUG_TYPE_CODEVIEW
        IMPORT Directory Table
        IMPORT Name Table
        IMPORT Hints/Names & DLL Names
    SECTION .data
    SECTION .rsrc
    SECTION .reloc
    CERTIFICATE Table
```

```
pFile     Raw Data                                          Value
00000000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ..............
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 E8 00 00 00  ................
00000040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68  ........!..L.!Th
00000050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode....$.......
00000080  3C E2 60 D1 78 83 0E 82  78 83 0E 82 78 83 0E 82  <.`.x...x...x...
00000090  66 D1 8A 82 59 83 0E 82  66 D1 9B 82 6B 83 0E 82  f...Y...f...k...
000000A0  5F 45 63 82 6D 83 0E 82  5F 45 75 82 71 83 0E 82  _Ec.m..._Eu.q...
000000B0  78 83 0F 82 01 83 0E 82  66 D1 8D 82 17 83 0E 82  x.......f.......
000000C0  66 D1 9A 82 79 83 0E 82  66 D1 9F 82 79 83 0E 82  f...y...f...y...
000000D0  52 69 63 68 78 83 0E 82  00 00 00 00 00 00 00 00  Richx...........
000000E0  00 00 00 00 00 00 00 00  50 45 00 00 4C 01 05 00  ........PE..L...
000000F0  16 44 F7 55 00 00 00 00  00 00 00 00 E0 00 02 01  .D.U............
00000100  0B 01 09 00 00 1E 01 00  00 BE 7A 06 00 00 00 00  ..........z.....
00000110  63 BA 00 00 00 10 00 00  00 30 01 00 00 00 40 00  c........0....@.
00000120  00 10 00 00 00 02 00 00  05 00 00 00 00 00 00 00  ................
00000130  05 00 00 00 00 00 00 00  00 00 7C 06 00 04 00 00  ..........|.....
00000140  58 97 7C 06 02 00 40 81  00 00 10 00 00 10 00 00  X.|...@.........
00000150  00 00 10 00 00 10 00 00  00 00 00 00 10 00 00 00  ................
00000160  00 00 00 00 00 00 00 00  74 4D 01 00 64 00 00 00  ........tM..d...
00000170  00 C0 01 00 48 CE 79 06  00 00 00 00 00 00 00 00  ....H.y.........
00000180  00 9E 7B 06 18 19 00 00  00 90 7B 06 F8 0B 00 00  ..{.......{.....
00000190  F0 31 01 00 1C 00 00 00  00 00 00 00 00 00 00 00  .1
```

Viewing iTunesSetup.exerectory Table

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.

E.g.: A Packed Program

Very few functions.

All you see is the unpacker.

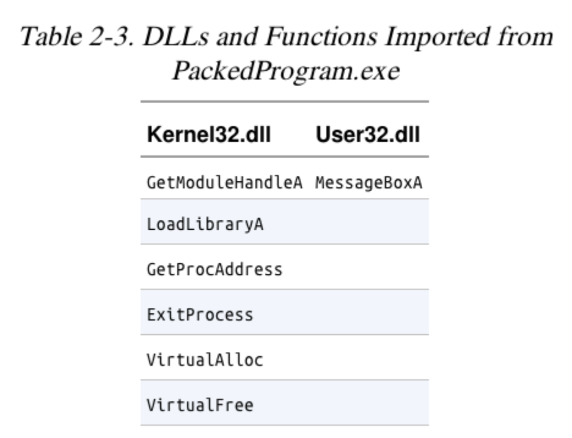


Table 2-3. DLLs and Functions Imported from PackedProgram.exe

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

The PE File *Headers* and *Sections*

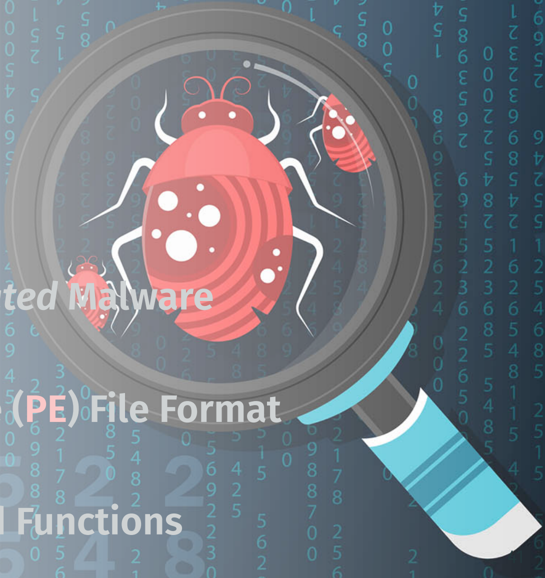
Antivirus Scanning

Finding Strings

Packed and *Obfuscated* Malware

Portable Executable (PE) File Format

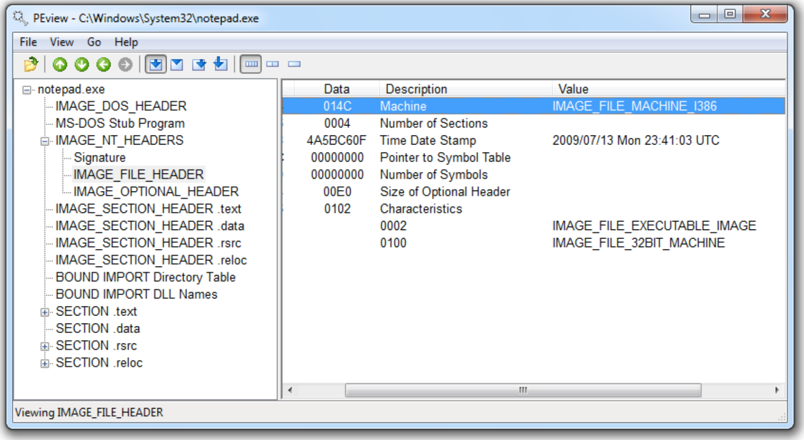
Linked Libraries and Functions



Important PE Sections

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

PEView



- 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

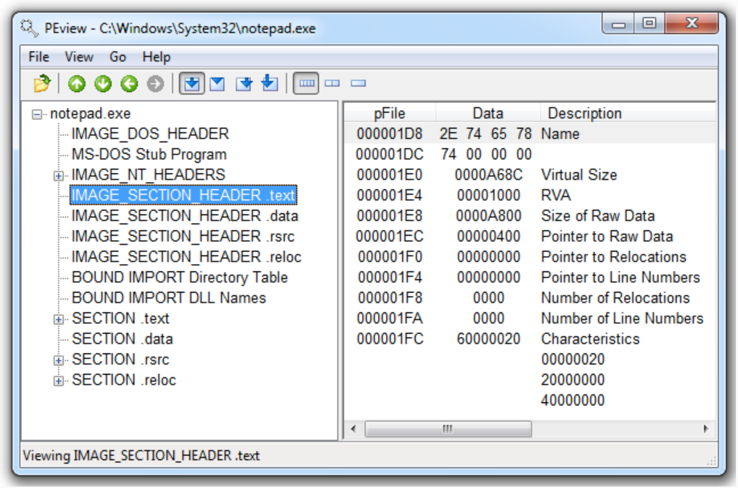
Virtual Size – RAM

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

Not Packed



PEview - C:\Windows\System32\notepad.exe
File View Go Help
notepad.exe
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
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

pFile	Data	Description
000001D8	2E 74 65 78	Name
000001DC	74 00 00 00	
000001E0	0000A68C	Virtual Size
000001E4	00001000	RVA
000001E8	0000A800	Size of Raw Data
000001EC	00000400	Pointer to Raw Data
000001F0	00000000	Pointer to Relocations
000001F4	00000000	Pointer to Line Numbers
000001F8	0000	Number of Relocations
000001FA	0000	Number of Line Numbers
000001FC	60000020	Characteristics
		00000020
		20000000
		40000000

Viewing IMAGE_SECTION_HEADER .text

Section Information for `PackedProgram.exe`

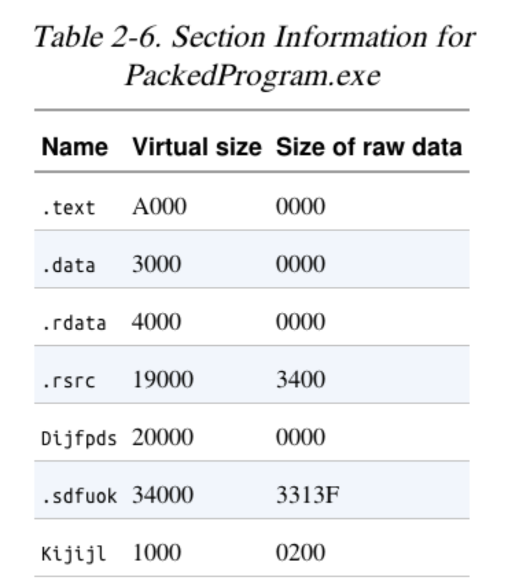


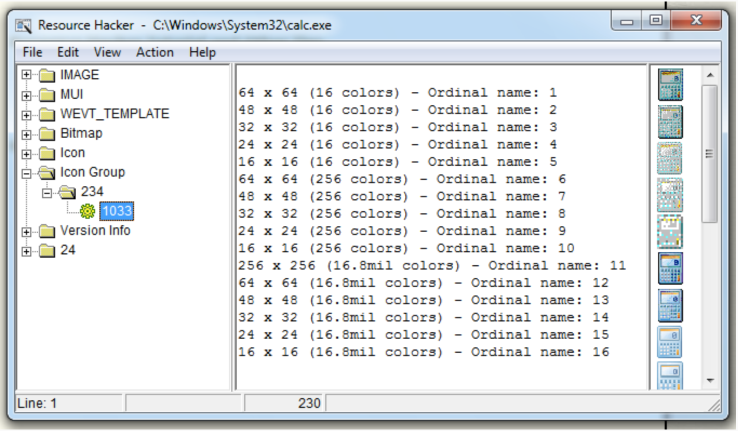
Table 2-6. Section Information for PackedProgram.exe

Name	Virtual size	Size of raw data
`.text`	A000	0000
`.data`	3000	0000
`.rdata`	4000	0000
`.rsrc`	19000	3400
`Dijfpds`	20000	0000
`.sdfuok`	34000	3313F
`Kijijl`	1000	0200

Resource Hacker

Lets you browse the `.rsrc` section.

- Strings, icons, and menus



```
64 x 64 (16 colors) - Ordinal name: 1
48 x 48 (16 colors) - Ordinal name: 2
32 x 32 (16 colors) - Ordinal name: 3
24 x 24 (16 colors) - Ordinal name: 4
16 x 16 (16 colors) - Ordinal name: 5
64 x 64 (256 colors) - Ordinal name: 6
48 x 48 (256 colors) - Ordinal name: 7
32 x 32 (256 colors) - Ordinal name: 8
24 x 24 (256 colors) - Ordinal name: 9
16 x 16 (256 colors) - Ordinal name: 10
256 x 256 (16.8mil colors) - Ordinal name: 11
64 x 64 (16.8mil colors) - Ordinal name: 12
48 x 48 (16.8mil colors) - Ordinal name: 13
32 x 32 (16.8mil colors) - Ordinal name: 14
24 x 24 (16.8mil colors) - Ordinal name: 15
16 x 16 (16.8mil colors) - Ordinal name: 16
```

Next: Malware Analysis in Virtual Machines