

Static Analysis: Understanding the PE Header



Tyler Hudak

INCIDENT RESPONDER

@secshoggoth



Overview



Pull key information from the header

Determine functionality

When and where it was created



PE Header



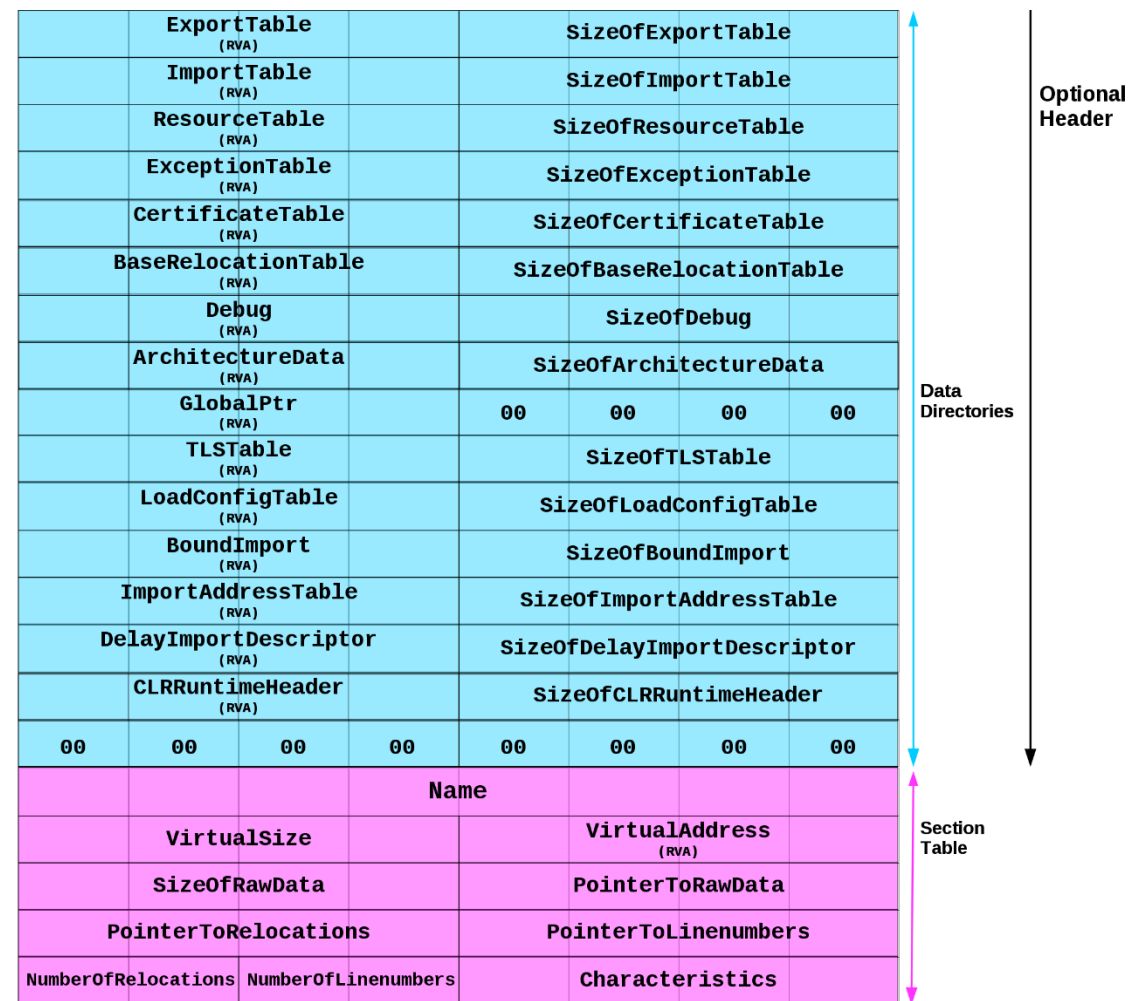
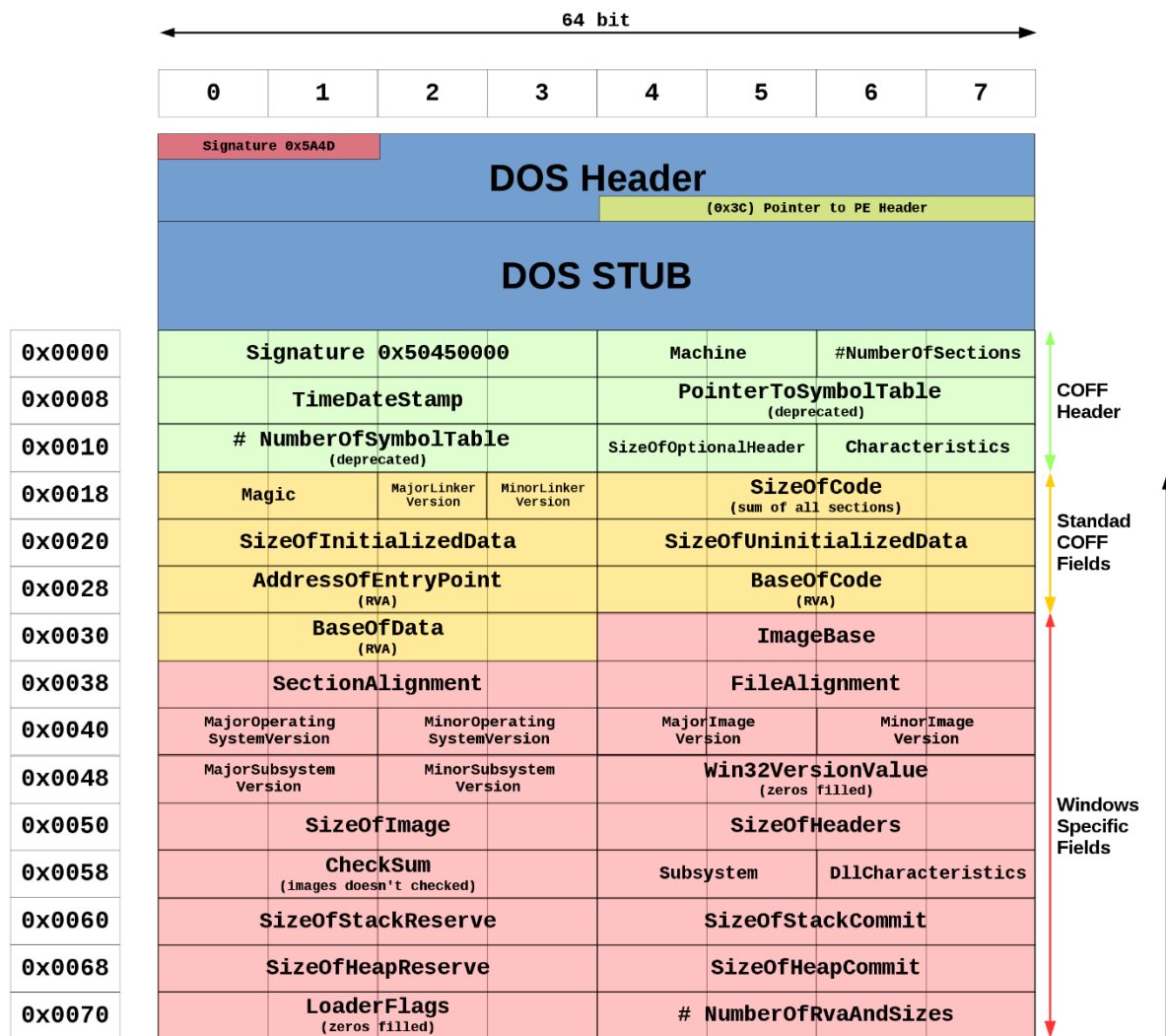
Container

Contains all of the information needed for the OS to execute the program



Director

Directs the OS where the pieces of the executable go in memory



File Header

c:\malware\antivirus.exe

indicators (2/7)

virustotal (n/a)

dos-stub (160 bytes)

file-header (20 bytes)

optional-header (224 bytes)

directories (3/15)

sections (4)

libraries (4)

imports (20/48)

exports (n/a)

exceptions (n/a)

tls-callbacks (n/a)

resources (3/3)

strings (34/2318)

debug (n/a)

manifest (n/a)










version (n/a)

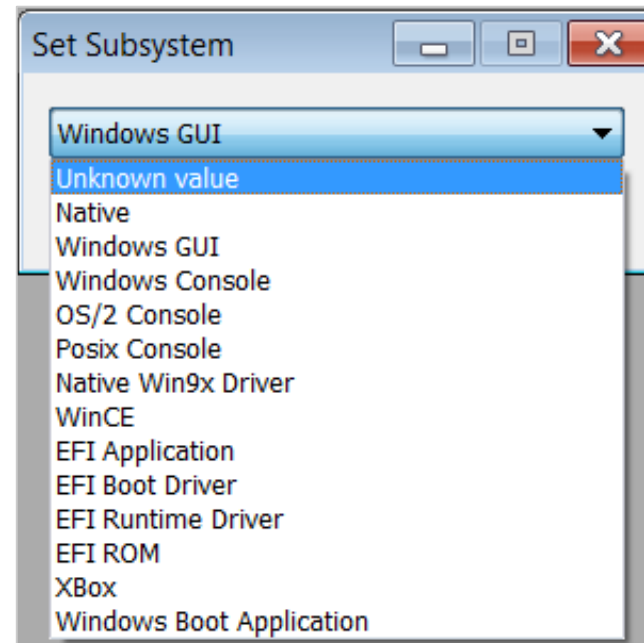
certificate (n/a)

property	value
signature	0x00004550
machine	Intel
sections	4
stamp	0x46641262 (Mon Jun 04 09:23:46 2007)
PointerToSymbolTable	0x00000000
symbols	0x0000 (0)
SizeOfOptionalHeader	0x00E0 (224 bytes)
processor-32bit	true
Relocation stripped	true
Large Address aware	false
uniprocessor-only	false
system-image	false
dynamic-link library	false
executable	true
debug information stripped	false
if on a removable media, copy and run from...	false



Optional Header

 File: antivirus.exe <ul style="list-style-type: none"> Dos Header Nt Headers<ul style="list-style-type: none"> File Header Optional Header Data Directories [x] Section Headers [x] Import Directory Resource Directory	Win32VersionValue	0000012C	Dword	00000000	
	SizeOfImage	00000130	Dword	0001A000	
	SizeOfHeaders	00000134	Dword	00000400	
	CheckSum	00000138	Dword	00000000	
	Subsystem	0000013C	Word	0002	Windows GUI



PE Sections

Logical segments of data, each with a name and characteristics.



Section Names

Name	Purpose
<code>.code</code>	Executable code
<code>.text</code>	Executable code
<code>.data</code>	Program Data
<code>.rdata</code>	Program Data
<code>.idata</code>	Import Table
<code>.edata</code>	Export Table
<code>.rsrc</code>	Resources



Section Characteristics

Read

Write

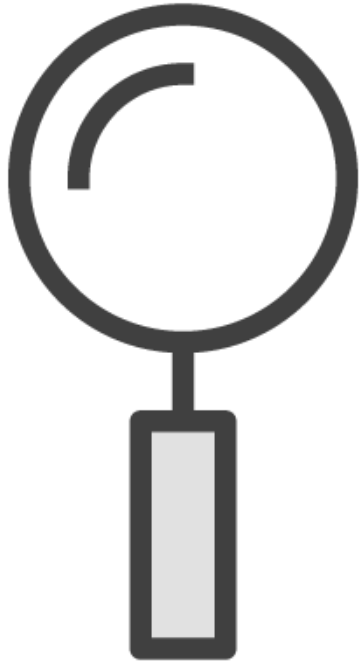
Execute

**Initialized
Data**

**Uninitialized
Data**



Packers and Sections



Random, Missing or Known Section Names

- UPX0, UPX1, UPX2, UPX!

Odd Permissions

- Write and Execute

Import Address Table

Contains a list of the DLLs and APIs loaded by the program at runtime.



Import Address Table

Import
Address
Table

Kernel32.dll
VirtualAlloc
ReadFile
WriteFile

OS
Memory

Kernel32.dll	
0x12345	VirtualAlloc
0x67890	ReadFile
0xABCDE	WriteFile



Import Address Table

Import
Address
Table

Kernel32.dll	
0x12345	
ReadFile	
WriteFile	

OS
Memory

Kernel32.dll	
0x12345	VirtualAlloc
0x67890	ReadFile
0xABCD E	WriteFile



Import Address Table

Import
Address
Table

Kernel32.dll	
0x12345	
0x67890	
WriteFile	

OS
Memory

Kernel32.dll	
0x12345	VirtualAlloc
0x67890	ReadFile
0xABCDE	WriteFile



Import Address Table

Import
Address
Table

Kernel32.dll
0x12345
0x67890
0xABCDE

OS
Memory

Kernel32.dll	
0x12345	VirtualAlloc
0x67890	ReadFile
0xABCDE	WriteFile



Ordinals

Import
Address
Table

Kernel32.dll
Ordinal: 0001
Ordinal: 0002
Ordinal: 0003

OS
Memory

Kernel32.dll		
0001	0x12345	VirtualAlloc
0002	0x67890	ReadFile
0003	0xABCDE	WriteFile



Ordinals

Import
Address
Table

Kernel32.dll
0x12345
0x67890
0xABCDE

OS
Memory

Kernel32.dll		
0001	0x12345	VirtualAlloc
0002	0x67890	ReadFile
0003	0xABCDE	WriteFile



Common APIs

Memory Operations

VirtualAlloc
VirtualProtect
VirtualFree

File Operations

CreateFile
ReadFile
WriteFile
DeleteFile

Registry Operations

RegCreateKey
RegDeleteKey
RegGetValue
RegSetValueEx
RegSetKeyValue



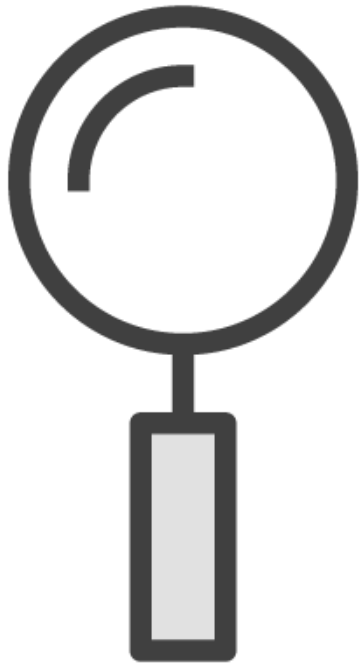
Common APIs (2)

Network Operations	
connect	InternetConnect
accept	InternetReadFile
send	InternetWriteFile
recv	gethostbyaddr
listen	gethostbyname

Misc
LoadLibrary
GetProcAddress
IsDebuggerPresent
WriteProcessMemory
CreateRemoteThread



Packers and IAT



Few Imports

- LoadLibrary
- GetProcAddress
- VirtualAlloc
- Exception: .NET mscorie.dll, _CorExeMain

Fake Imports

Resources

Raw data attached to the program.



Resource Uses

Icons

Images

Menus

Hidden
Executables
and Configs

Fake
Documents



Resource Languages

type	name	signature	standard	size (1262...	md5	entro...	language (1)
MUI	1	MUI	x	240	FC5665075982ECFC83A649D72...	2.621	English United States
Bitmap	51209	Bitmap	x	22042	8C41EA4A8F0ABCC24D58E998...	6.610	English United States
Icon	1	Windows	x	1640	5E0424A037ED1CF4B86D9CAE...	3.146	English United States
Icon	2	Windows	x	744	E90A939E1107E27E1D95C25E2...	3.463	English United States
Icon	3	Windows	x	488	44B38E737F03387A86DB70708...	3.415	English United States
Icon	4	Windows	x	296	4C7576E8F541BB3E4915569E5...	3.191	English United States
Icon	5	Windows	x	3752	7684234AAE030B0E361B77C54...	5.339	English United States
Icon	6	Windows	x	2216	30678F5B06BC441A5BD8ED28...	5.887	English United States
Icon	7	Windows	x	1736	C50E91E6D59210580879F7BC5...	5.778	English United States
Icon	8	Windows	x	1384	011BDE7B9C82D9453B722295...	3.503	English United States
Icon	9	Windows	x	72024	489350E7DBC2BD241EEEF92...	7.927	English United States

type	name	signature	standard	size (8705...	md5	entro...	language (1)
DLL	101	unknown	-	44280	C107AA913FC4AAB2BA8258D2...	7.996	Russian
TEH	114	unknown	-	18	1069C786672AE57FE7E1ECBD0...	3.572	Russian
XML	103	unknown	-	42757	36675F2F9AB28AD3711FFF0C5...	5.467	Russian



Resource Timestamp

[-] Resource Directory
[-] Resource Directory Entry 1, Name: DLL
 [+] Resource Directory
[-] Resource Directory Entry 2, Name: TEH
 [+] Resource Directory
[-] Resource Directory Entry 3, Name: XML
 [+] Resource Directory

Member	Offset	Size	Value
Characteristics	00001858	Dword	00000000
TimeDateStamp	0000185C	Dword	00000000



CFF Explorer

Parses PE header

Shows raw details

<http://www.ntcore.com/exsuite.php>



PE Studio

More high level, decoded information

Strings analysis

Blacklisted APIs

<https://www.winitor.com/>



Demo



PE Header Analysis

<http://bit.ly/malfund-1>



budget-report.exe

Compiled in 1998?!?

Does not appear to be packed

- Normal looking sections
- IAT normal size

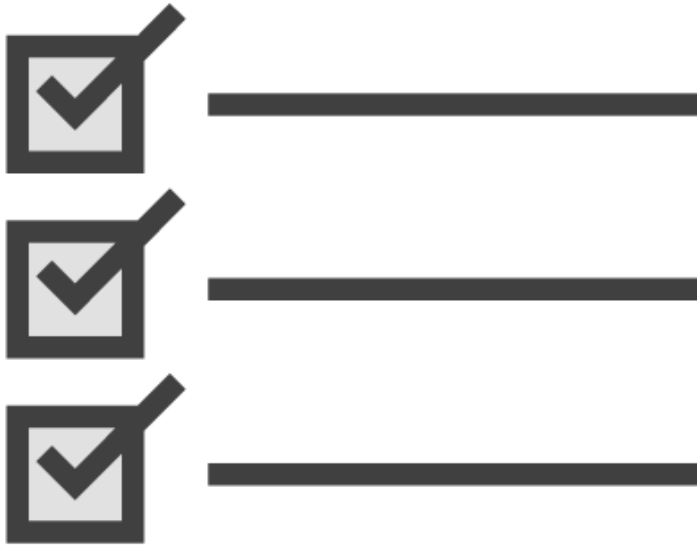
Functionality

- Modifies the registry
- Writes to the file system
- Communicates over the network
- Executes programs

From a “US English” computer



Packers



Sections

- Random or well known names
- Odd permissions

IAT

- Few imports

Summary



Contains information the OS needs to execute the program

When and where the malware was created

Malware functionality

