Practical Malware Analysis

Kris Kendall and Chad McMillan

MANDIANT
INTELLIGENT INFORMATION SECURITY

Black Hat

Outline

- Why Analyze Malware?
- Creating a Safe Analytical Environment
- Static Analysis Techniques
- Dynamic Analysis Techniques
- Packing
- Finding Malware

What is Malware?

Generally

- Any code that “performs evil”

Today

- Executable content with unknown functionality that is resident on a system of investigative interest
  - Viruses
  - Worms
  - Intrusion Tools
  - Spyware
  - Rootkits

Analyzing Malware

Why Analyze Malware?

- To assess damage
- To discover indicators of compromise
- To determine sophistication level of an intruder
- To identify a vulnerability
- To catch the “bad guy”®
- To answer questions…

Why Analyze Malware?

Business Questions

1. What is the purpose of the malware?
2. How did it get here?
3. Who is targeting us and how good are they?
4. How can I get rid of it?
5. What did they steal?

Why Analyze Malware?

Business Questions

6. How long has it been here?
7. Does it spread on its own?
8. How can I find it on other machines?
9. How do I prevent this from happening in the future?

Why Analyze Malware?

Technical Questions

1. Network Indicators?
2. Host-based Indicators?
3. Persistence Mechanism?
4. Date of Compilation?
5. Date of Installation?

Why Analyze Malware?

Technical Questions

6. What language was it written in?
7. Is it packed?
8. Was it designed to thwart analysis?
9. Does it have any rootkit functionality?

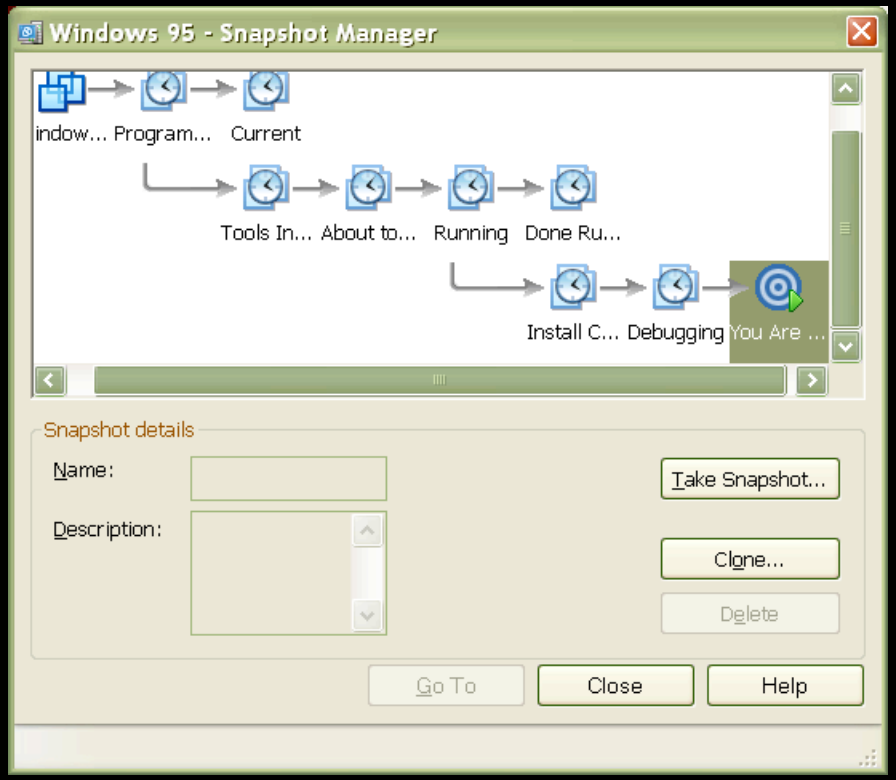
Creating a Safe Analytical Environment

MANDIANT
INTELLIGENT INFORMATION SECURITY

Creating a Safe Environment

- Do Not Run Malware on Your Computer!
- Old And Busted
  - Shove several PCs in a room on an isolated network, create disk images, re-image a target machine to return to pristine state
- The (not so) New Hotness
  - Use virtualization to make things fast and safe
  - VMware (Workstation, Server [free])
  - Parallels (cheap)
  - Microsoft Virtual PC (free)
  - Xen (free)

VMWare Snapshot Manager



Virtualization is not the Only Option

- Use Truman (by Joe Stewart @ Lurhq) to automatically re-image physical machines
  - http://www.lurhq.com/truman/
- Use a hard drive write cache card
  - CoreRestore from CoreProtect
  - Designate a portion of the hard drive as protected, all writes to the protected portion get redirected to another part of the disk
  - Reboot to restore the drive to the previous state

Reduce Risk using Platform Diversity

- If possible, perform static analysis in a different OS than your malware targets
  - Avoid the oh-$@!7 double-click
  - IDA Pro for OS X is coming soon

Creating a Safe Environment

- It is easier to perform analysis if you allow the malware to “call home”…
- However:
  - The attacker might change his behavior
  - By allowing malware to connect to a controlling server, you may be entering a real-time battle with an actual human for control of your analysis (virtual) machine
  - Your IP might become the target for additional attacks (consider using TOR)
  - You may end up attacking other people

Creating a Safe Environment

- Therefore, we usually do not allow malware to touch the real network
  - Use the host-only networking feature of your virtualization platform
  - Establish real services (DNS, Web, etc) on your host OS or other virtual machines
  - Use netcat to create listening ports and interact with text-based client
  - Build custom controlling servers as required (usually in a high-level scripting language)

Virtualization Considerations

- Using a Virtual Machine helps, but…
- Set up the “victim” with no network or host-only networking
- Your virtualization software is not perfect
- Malicious code can detect that it is running in a virtual machine
- A 0-day worm that can exploit a listening service on your host OS will escape the sandbox
  - Even if you are using host-only networking!

Performing Malware Analysis on Windows

Static vs. Dynamic Analysis

- Static Analysis
  - Code is Not Executed
  - Autopsy or Dissection of “Dead” Code
- Dynamic Analysis
  - Observing and Controlling Running (“live”) Code
  - Ant Farm
- The Fastest Path to the Best Answers Will Usually Involve a Combination of Both.

Static Analysis “the dissection”

MANDIANT
INTELLIGENT INFORMATION SECURITY

Static Analysis

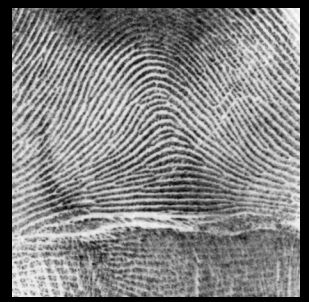
Static Analysis is Safer

- Since we aren't actually running malicious code, we don't have to worry (as much) about creating a safe environment

File Fingerprinting

- As a first step, fingerprint the files you are examining so you will know if they change during analysis
- Use md5deep, md5sum, Hex Workshop, etc

```
krk@ws ~> md5sum hello* > md5sum_hello_files.txt
krk@ws ~> cat md5sum_hello_files.txt
   611957bd6a2ad9642027904a65f3638e  hello
   7ab03b44ac6a20b0fa0cc80b636b0f51  hello.c
   bef5bfe7ddf597c8ea86eecb2cbf52a3  hello_debug
   38e85544dd4349c523430923eafc86ac  hello_static
```



- When you have completed your analysis, or at various points along the way, you should go back and check the md5sums to ensure the values have not changed!

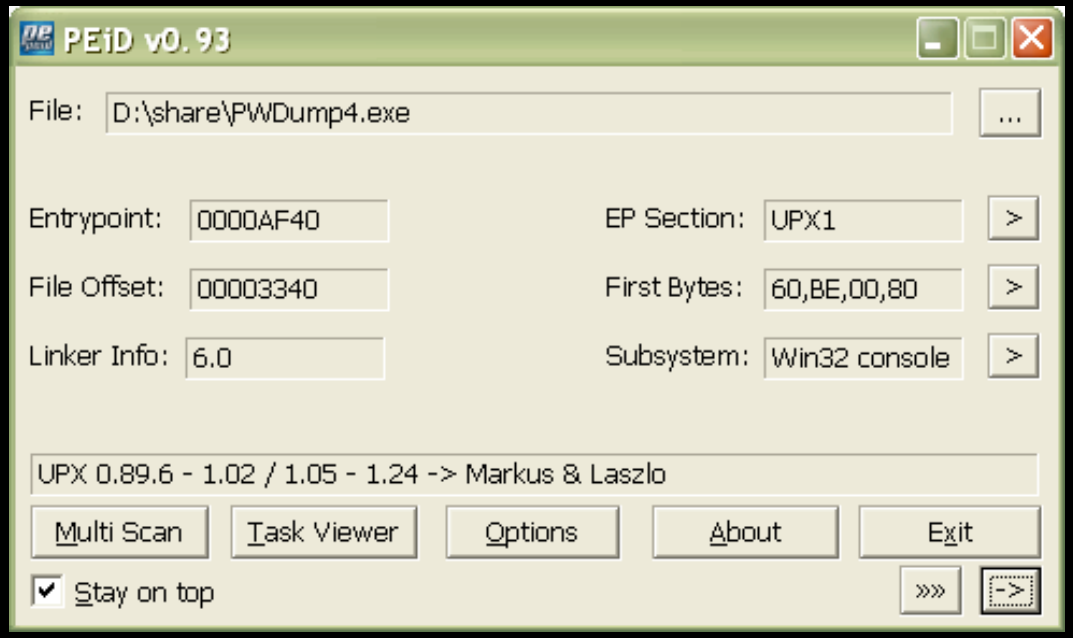
```
krk@ws ~> md5sum -c md5sum_hello_files.txt
```

Virus Scan

- Always scan new malware with an up to date virus scanner.
- Someone else may have already discovered and documented the program you are investigating
- If the code is not sensitive, consider submitting to http://www.virustotal.com

PEiD

- PEiD is a free program that will tell you details about Windows executable files
- Identifies signatures associated with over 600 different “packers” and compilers



“Caprica6”

- Mandiant tool that identifies packed code (amongst other things)
- Covered in detail later in this talk

Strings

- Sometimes things are easy
- First look at the obvious – strings

```
$ strings unknown2.exe
...
 <host> <port>
 -install <host> <port>
 -remove
EC.1
EC.2
cmd.exe
connect thread started!
...
```

- Strings, Bintext, Hex Workshop, IDA Pro
- Be aware of Unicode

Strings

```
C:\analysis> strings
Strings v2.1
Copyright (C) 1999-2003 Mark Russinovich
Systems Internals - www.sysinternals.com
usage: strings [-s] [-n length] [-a] [-u] [-q] <file or directory>
-s      Recurse subdirectories
-n      Minimum string length (default is 3)
-a      Ascii-only search (Unicode and Ascii is default)
-u      Unicode-only search (Unicode and Ascii is default)
-q      Quiet (no banner)
```

Strings

- Be careful about drawing conclusions
- There is nothing stopping the attacker from planting strings meant to deceive the analyst
- However, strings are a good first step and can sometimes even provide attribution

```
rem barok -loveletter(vbe) <i hate go to school>
rem by: spyder / ispyder@mail.com / \
        @GRAMMERSoft Group / Manila,Philippines
On Error Resume Next
dim fso,dirsystem,dirwin,dirtemp,eq,ctr,file,
    vbscopy,dow eq="" ctr=0
Set fso = CreateObject("Scripting.FileSystemObject")
set file = fso.OpenTextFile(WScript.ScriptFullname,1)
```

Conducting Web Research

- Look at unique strings, email addresses, network info
  - But! the intruder/author could be watching for you.
- Search the web
  - Be careful … Google cache != Anonymous
  - You might find other victims, or complete analysis
  - Don't forget newsgroups
- It helps if you know Chinese (or Russian, or Spanish)
  - http://www.google.com/language_tools?hl=en

No Strings Attached

- Point-and-click “packers” make it easy for intruders to obfuscate the contents of binary tools
- More on packers later…

- We can still gather useful information by examining the layout of the executable file

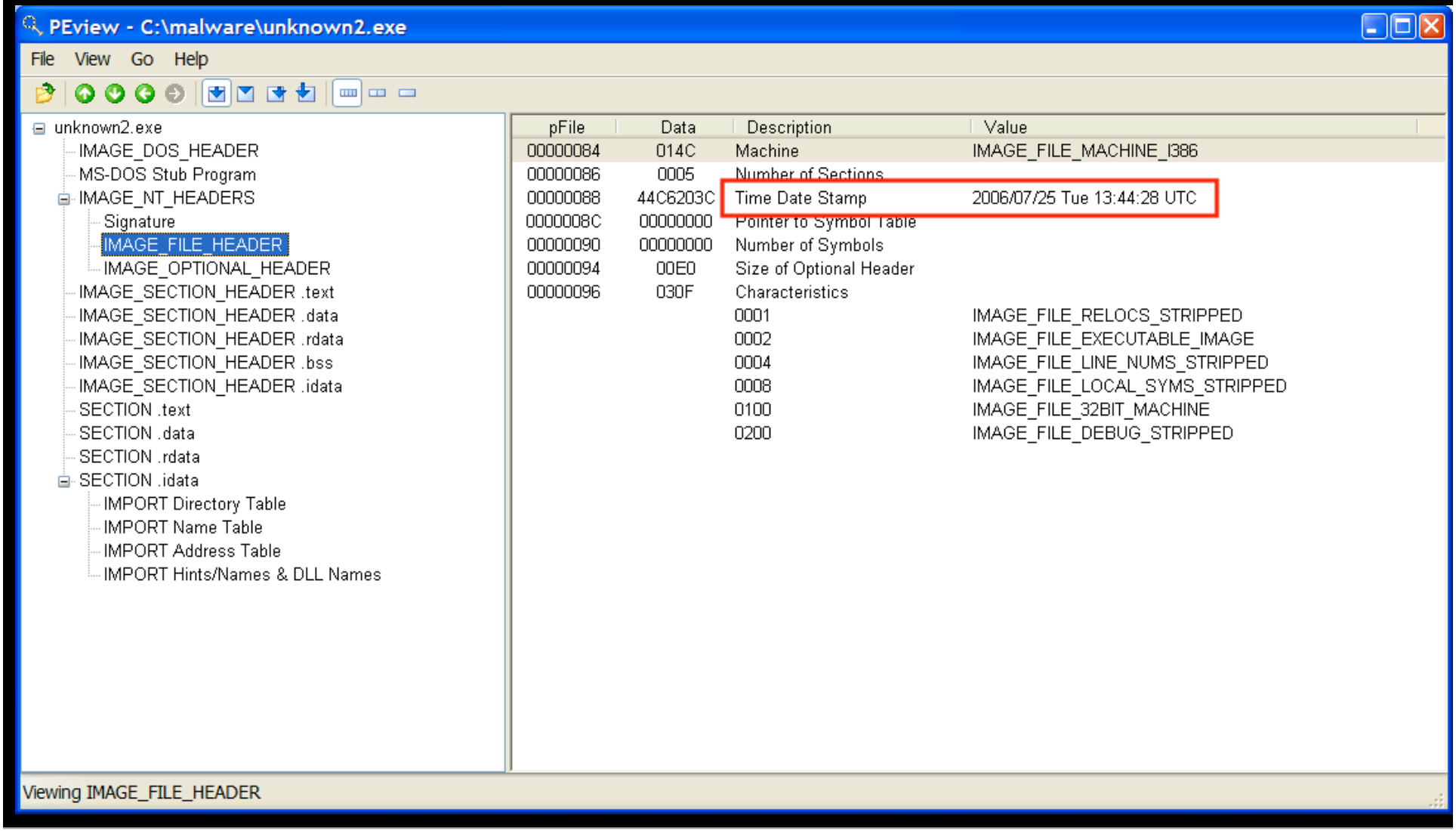
Inside the PE Format

- Executable File Formats
  - Windows: PE (Portable Executable)
    - www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx
  - Linux: ELF (Executable and Linking Format)
    - www.skyfree.org/linux/references/ELF_Format.pdf
- Useful Information
  - Imports
  - Exports
  - Metadata
  - Resources

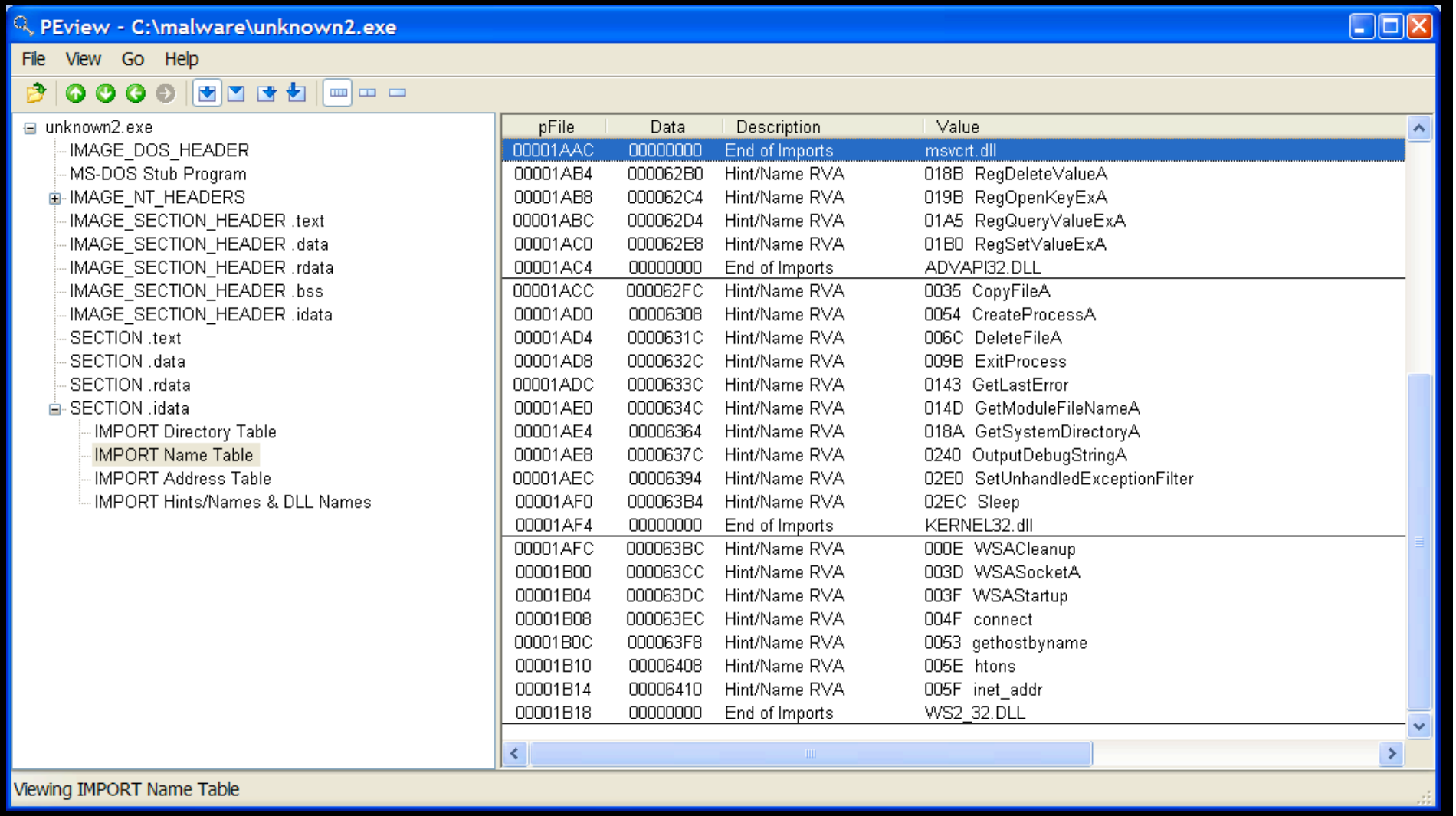
Inside the PE

- Tools
  - PEview – Wayne Radburn
    http://www.magma.ca/~wjr/
  - Depends – Steve Miller
    http://www.dependencywalker.com
  - PEBrowse Professional – Russ Osterlund
    http://www.smidgeonsoft.com
  - Objdump – Cygwin
    http://www.cygwin.com
  - IDA Pro – DataRescue
    http://www.datarescue.be
  - Resource Hacker – Angus Johnson
    http://www.angusj.com/resourcehacker/

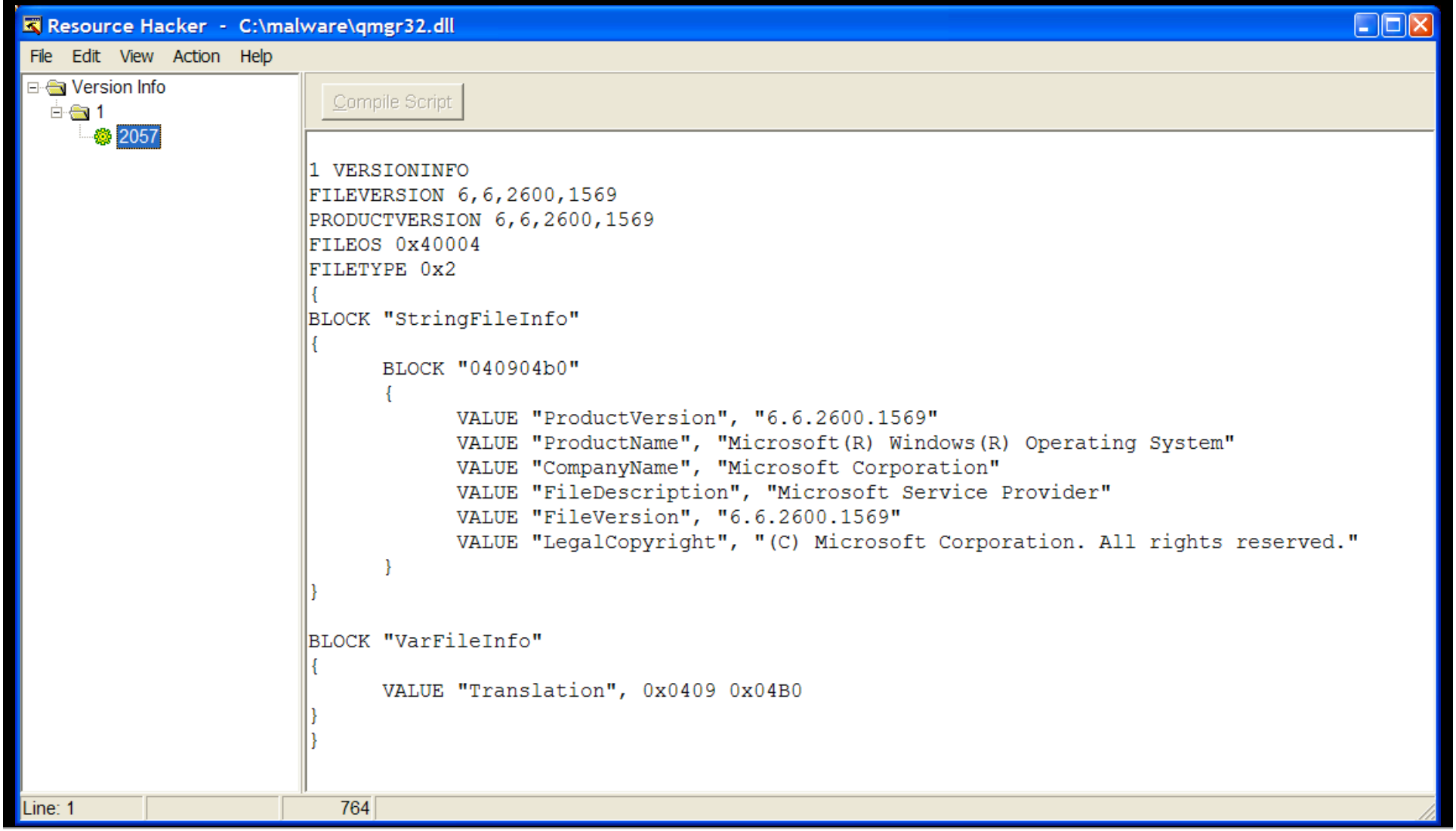
PEview


PEview - C:\malware\unknown2.exe
File View Go Help
unknown2.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 .rdata
IMAGE_SECTION_HEADER .bss
IMAGE_SECTION_HEADER .idata
SECTION .text
SECTION .data
SECTION .rdata
SECTION .idata
IMPORT Directory Table
IMPORT Name Table
IMPORT Address Table
IMPORT Hints/Names & DLL Names
pFile Data Description Value
00000084 014C Machine IMAGE_FILE_MACHINE_I386
00000086 0005 Number of Sections
00000088 44C6203C Time Date Stamp 2006/07/25 Tue 13:44:28 UTC
0000008C 00000000 Pointer to Symbol Table
00000090 00000000 Number of Symbols
00000094 00E0 Size of Optional Header
00000096 030F Characteristics
0001 IMAGE_FILE_RELOCS_STRIPPED
0002 IMAGE_FILE_EXECUTABLE_IMAGE
0004 IMAGE_FILE_LINE_NUMS_STRIPPED
0008 IMAGE_FILE_LOCAL_SYMS_STRIPPED
0100 IMAGE_FILE_32BIT_MACHINE
0200 IMAGE_FILE_DEBUG_STRIPPED
Viewing IMAGE_FILE_HEADER

PEview

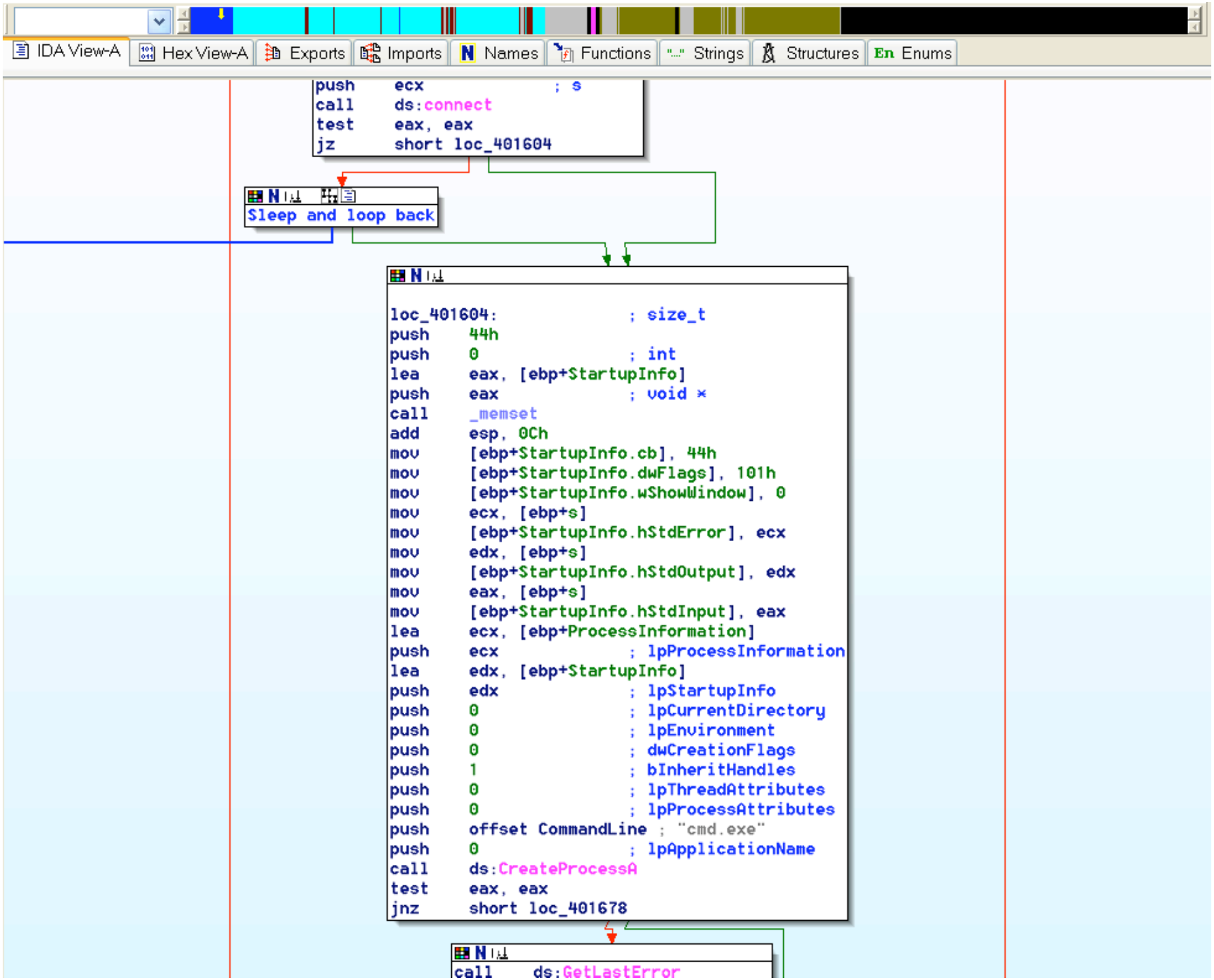

PEview - C:\malware\unknown2.exe
File View Go Help
unknown2.exe
IMAGE_DOS_HEADER
MS-DOS Stub Program
IMAGE_NT_HEADERS
IMAGE_SECTION_HEADER .text
IMAGE_SECTION_HEADER .data
IMAGE_SECTION_HEADER .rdata
IMAGE_SECTION_HEADER .bss
IMAGE_SECTION_HEADER .idata
SECTION .text
SECTION .data
SECTION .rdata
SECTION .idata
IMPORT Directory Table
IMPORT Name Table
IMPORT Address Table
IMPORT Hints/Names & DLL Names
pFile Data Description Value
00001AAC 00000000 End of Imports msvcrt.dll
00001AB4 000062B0 Hint/Name RVA 018B RegDeleteValueA
00001AB8 000062C4 Hint/Name RVA 019B RegOpenKeyExA
00001ABC 000062D4 Hint/Name RVA 01A5 RegQueryValueExA
00001AC0 000062E8 Hint/Name RVA 01B0 RegSetValueExA
00001AC4 00000000 End of Imports ADVAPI32.DLL
00001ACC 000062FC Hint/Name RVA 0035 CopyFileA
00001AD0 00006308 Hint/Name RVA 0054 CreateProcessA
00001AD4 0000631C Hint/Name RVA 006C DeleteFileA
00001AD8 0000632C Hint/Name RVA 009B ExitProcess
00001ADC 0000633C Hint/Name RVA 0143 GetLastError
00001AE0 0000634C Hint/Name RVA 014D GetModuleFileNameA
00001AE4 00006364 Hint/Name RVA 018A GetSystemDirectoryA
00001AE8 0000637C Hint/Name RVA 0240 OutputDebugStringA
00001AEC 00006394 Hint/Name RVA 02E0 SetUnhandledExceptionFilter
00001AF0 000063B4 Hint/Name RVA 02EC Sleep
00001AF4 00000000 End of Imports KERNEL32.dll
00001AFC 000063BC Hint/Name RVA 000E WSACleanup
00001B00 000063CC Hint/Name RVA 003D WSASocketA
00001B04 000063DC Hint/Name RVA 003F WSAStartup
00001B08 000063EC Hint/Name RVA 004F connect
00001B0C 000063F8 Hint/Name RVA 0053 gethostbyname
00001B10 00006408 Hint/Name RVA 005E htons
00001B14 00006410 Hint/Name RVA 005F inet_addr
00001B18 00000000 End of Imports WS2_32.DLL
Viewing IMPORT Name Table

Resource Hacker



Disassembly

- Automated disassemblers can take machine code and “reverse” it to a slightly higher-level
- Many tools can disassemble x86 code
  - Objdump, Python w/ libdisassemble, IDA Pro
- But, IDA Pro is what everyone uses
- Manual examination of disassembly is somewhat painstaking, slow, and can be hard
  - Keep your goals in mind and don’t get bogged down


IDA View-A
Hex View-A
Exports
Imports
Names
Functions
Strings
Structures
Enums
push ecx ; s
call ds:connect
test eax, eax
jz short loc_401604
Sleep and loop back
loc_401604: ; size_t
push 44h
push 0 ; int
lea eax, [ebp+StartupInfo]
push eax ; void *
call _memset
add esp, 0Ch
mov [ebp+StartupInfo.cb], 44h
mov [ebp+StartupInfo.dwFlags], 101h
mov [ebp+StartupInfo.wShowWindow], 0
mov ecx, [ebp+s]
mov [ebp+StartupInfo.hStdError], ecx
mov edx, [ebp+s]
mov [ebp+StartupInfo.hStdOutput], edx
mov eax, [ebp+s]
mov [ebp+StartupInfo.hStdInput], eax
lea ecx, [ebp+ProcessInformation]
push ecx ; lpProcessInformation
lea edx, [ebp+StartupInfo]
push edx ; lpStartupInfo
push 0 ; lpCurrentDirectory
push 0 ; lpEnvironment
push 0 ; dwCreationFlags
push 1 ; bInheritHandles
push 0 ; lpThreadAttributes
push 0 ; lpProcessAttributes
push offset CommandLine ; "cmd.exe"
push 0 ; lpApplicationName
call ds:CreateProcessA
test eax, eax
jnz short loc_401678
call ds:GetLastError

Performing Dynamic Analysis

MANDIANT
INTELLIGENT INFORMATION SECURITY

Dynamic Analysis

- Static Analysis will reveal some immediate information
- Exhaustive static analysis could theoretically answer any question, but it is slow and hard
- Usually you care more about “what” malware is doing than “how” it is being accomplished
- Dynamic analysis is conducted by observing and manipulating malware as it runs

Safe Environment

- Our nice, safe analytical environment wasn't that important during static analysis
- As soon as you run an unknown piece of code on your system, nothing that's writable can be trusted
- In general we will need to run the program many times. Snapshots make life easier

System Monitoring

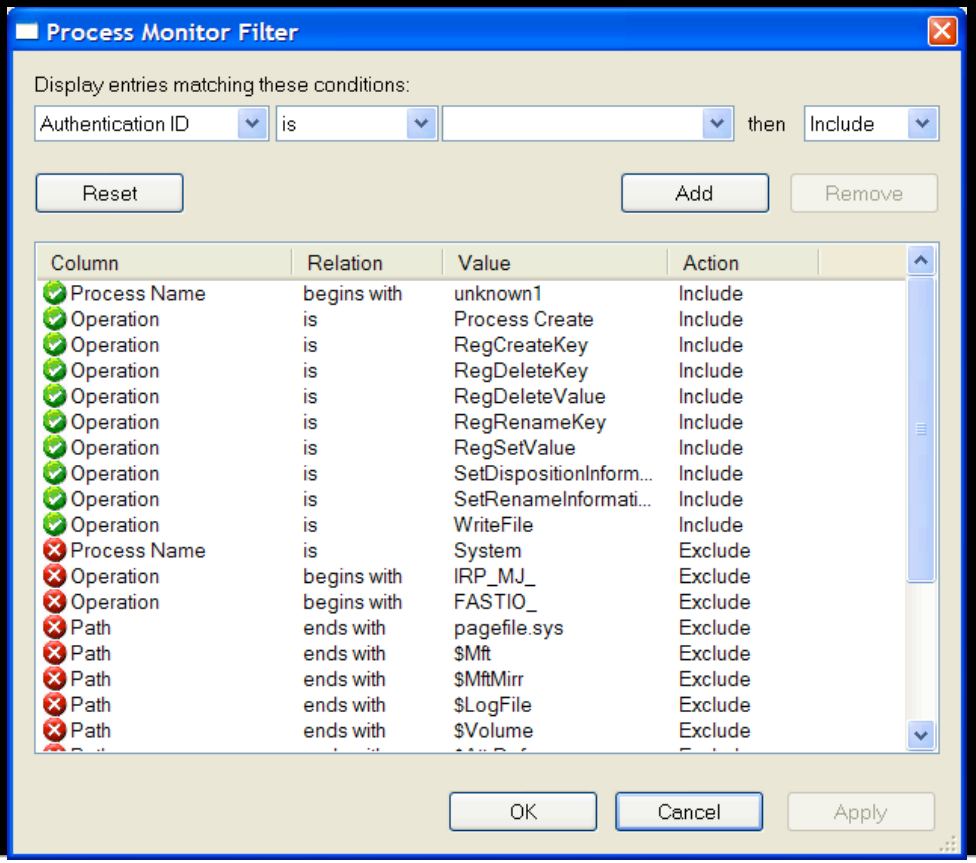
- What we are after
  - Registry Activity
  - File Activity
  - Process Activity
  - Network Traffic
- The tools
  - SysInternals Process Monitor
  - Wireshark
  - + a whole bunch of other stuff

Process Monitor

- Process Monitor is a SysInternals tool that records information about File System, Registry, and Process/Thread activity
- If you liked Filemon and Regmon—you'll really like Process Monitor
- Changes from Filemon/Regmon:
  - Procmon will record everything, user can change display filter at will
  - Procmon tracks process activity

Process Monitor

- The key to effective use of Process Monitor for malware analysis is filter configuration



Wireshark

- Wireshark is a protocol analyzer that captures and decodes network traffic
- Wireshark is not aware of what process generates traffic
- As with Process Monitor, the key is using filters to focus on what is relevant

Dynamic Analysis Example

- Use Process Monitor and Wireshark to quickly reveal the behavior of a malicious program

Other Tools

- Port Explorer
  - http://www.diamondcs.com.au/portexplorer/
  - Monitors network traffic at the connection level
- Malcode Analysts Pack
  - http://labs.idefense.com/labs-software.php?show=8
  - fakeDNS
- Paros, Fiddler
  - Web proxies that can capture and modify traffic
- Norman Sandbox

Beyond System Monitoring

- In some cases, simple Static Analysis and System Monitoring will answer your questions
- Beyond this point, you need a debugger
  - Windbg (Microsoft)
  - Ollydbg (Oleh Yuschuk)
  - Ida Pro (Datarescue)
- Or a scriptable debugger
  - Paimei - http://pedram.redhive.com/PaiMei/docs/
  - Vtrace - http://www.kenshoto.com/vtrace/

Armored Malware

MANDIANT
INTELLIGENT INFORMATION SECURITY

Armor Features

- Encryption
- Compression
- Obfuscation
- Anti-Patching
  - CRC Checking
- Anti-Tracing
  - SoftICE, ICEDump Detection Code.
  - Crashes OS if they are Found in Memory
- Anti-Unpacking
- Anti-Vmware
- Polymorphic/Self-Mutating
- Restrictive Dates
- Password Protected
- Configuration Files

Viruses can Circumvent AV?

We're offering anti-detection service for any type of windows modules. There are many ways how to make your module undetected hence you can see below quite complicated price table with examples. To order this service write a mail with full description of what you need to holy_father@phreaker.net. Feel free to write a mail if you're not sure how much would your order cost or if you have special demands (e.g. bypassing any detector that is not in list).

feature	Morphine	Hacker defender	Hacker defender driver	Other (no driver or libraries)	Libraries	Drivers
basic fee	€ 30.00	€ 20.00[0]		€ 15.00	€ 15.00	€ 15.00
morphined[1]	x	+ € 02.50	x	+ € 02.50	+ € 02.50	x
morphined - unique[2]	x	+ € 25.00	x	+ € 20.00	+ € 20.00	x
per AV[3]	+ € 10.00	+ € 05.00	+ € 05.00	+ € 08.00	+ € 09.00	+ € 10.00
all AV[3]	x	+ € 25.00	+ € 30.00	+ € 30.00	+ € 35.00	+ € 40.00
unique version[4]	+ € 20.00	+ € 25.00	+ € 20.00	x	x	x
source code	+ € 20.00	+ € 30.00	+ € 15.00	– € 10.00[5]	– € 10.00[5]	– € 10.00[5]
no driver	x	+ € 10.00[6]	x	x	x	x
special	x	special[7]	x	x	x	x

Packers

- Origins
  - Compression
    - Bandwidth reduction
    - Save space
- Current use
  - Bypass anti-virus signatures
  - Prevent reverse engineering

Packers

- **UPack** by **Dwing**. **08.IV.2005**.
- **Mew** by **Northfox**. **22.IX.2004**.
- **UPX** by **Laszlo & Markus**. **03.VII.2004**.
- **Packman** by **bubba**. **27.II.2005**.
- **EZIP** by **Jonathan Clark**. **21.VII.2001**.
- **PE-PaCK** by **ANAKiN**. **12.I.1999**.
- **FSG** by **bart**. **24.V.2004**.
- **Dropper** by **Gem**. **13.III.2005**.
- **CExe** by **Scott**. **20.III.2003**.
- **PE Diminisher** by **tERAPHY**. **11.IX.1999**.
- **PECRYPT32** by **random**, **killa** and **acpizer**. **12.I.1999**.
- **PESpin** by **cyberbob**. **09.III.2005**.
- **NSPack** by **North star Tech**. **05.VI.2005**.
- **eXPressor** by **CGSoftLabs**. **28.III.2005**.
- **Thinstall** by **Jonathan Clark**. **29.III.2005**
- **PEBundle** by **Jeremy Collake**. **12.III.2004**.
- **PECompact** by **DevelTek**. **06.IV.2005**.
- **AS-Pack (shareware)** by **Solodovnikov Alexey**. **07.I.2002**.
- **NeoLite (shareware)** by **NeoWorx Inc**. **04.IV.1999**.
- **WWPack 32** by **Piotr Warezak**. **07.VII.2000**.
- **ARM Protector** by **SMoKE**. **22.IX.2004**.

Side effects of Packing

- No strings (legitimate)
- “Few” imports
  - 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 of Packing - Imports

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

Packed

RVA	Name
0101AE3Ch	kernel32.dll

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

Side Effects of Packing – Section Size and Entropy

Unpacked : Entropy (st dev): 0.7653

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

Packed : Entropy (st dev): 1.0666

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

Strings on Packed Binary

```
C:\analysis>strings sak.exe
Strings v2.1
Copyright (C) 1999-2003 Mark Russinovich
Systems Internals - www.sysinternals.com
!Windows Program
$PE
@.data
.idata
$s!
;Ot
(!B
KERNEL32.dll
LoadLibraryA
GetProcAddress
DM.D
&DS
d'D
~E-
```

So, Packing == Bad?

- No … there are legitimately packed apps
  - Google Desktop Search – “Troubleshoot Network.exe” : PECompact V 2
  - Adobe Acrobat 7.0 – AdobeUpdateManager.exe : PECompact V 2
- So, how do you tell the difference?

Knowing the difference

- Collect as much static info about a PE as feasible
- Correlate the extracted information
- Score / rate what you know

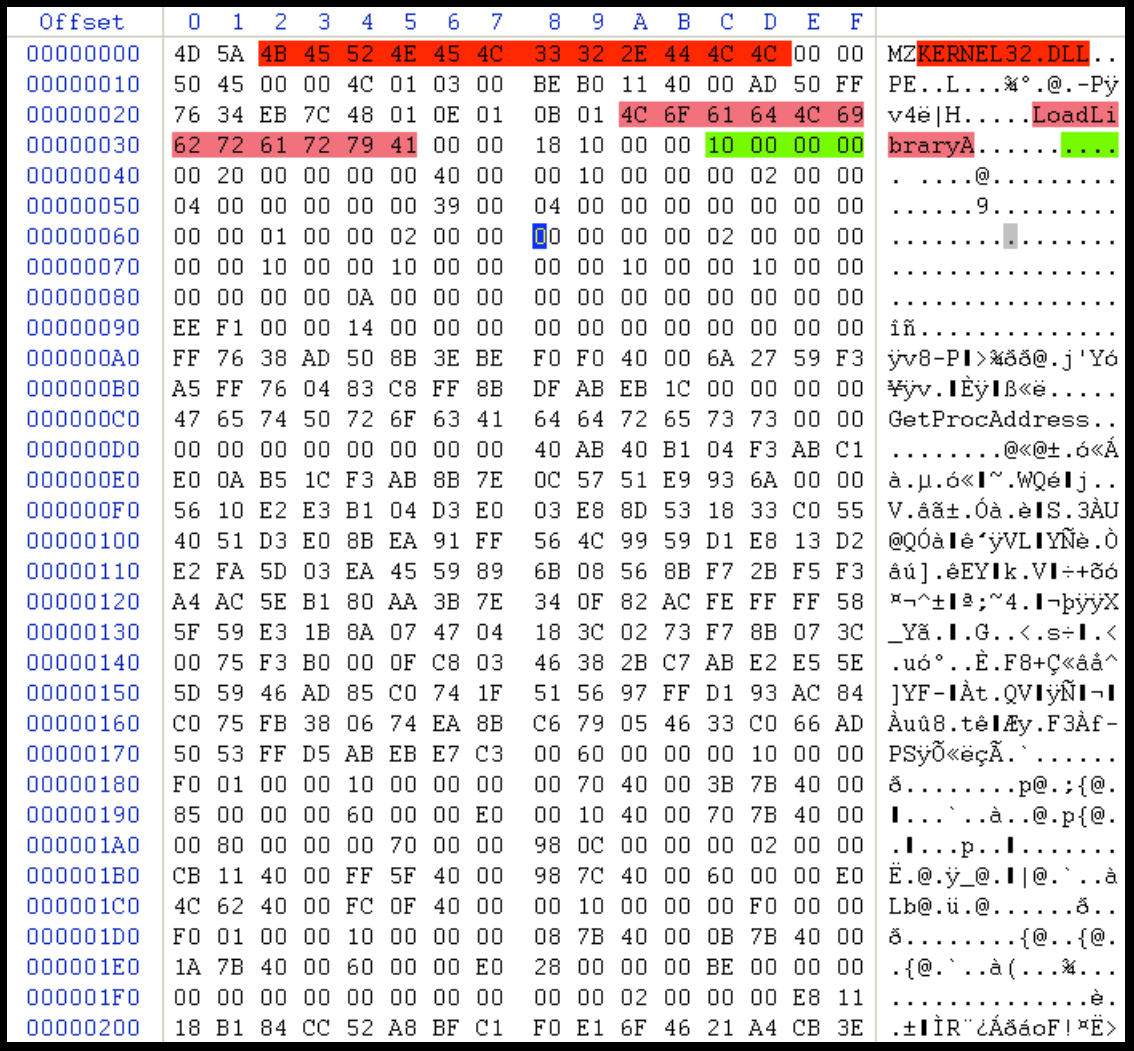
“Caprica Six”

- PE Header Anomolies
  - Incorrect image size
  - Unaligned sections
  - Non-ASCII section names
  - Overlapping headers
- Entry point signature detection
  - Quick scan using xor from the entry point
- Full section roaming signature detection (using modified Boyer-Moore search)
  - Signatures are developed subjectively (by reverse engineering) to lower FP’s and obtain signatures for “code” that would be difficult to modify (and are necessary for unpacking)

“Caprica Six” (cont’)

- Comparison of EP and roaming signature
  - Used to determine trying to hide packer use
    - Adding a new section with “fake code”
    - Instruction swaps at the entry
- Full section entropy calculation (sliding window check with standard deviation – patent application submitted)
  - Comparison against section characteristics (code, executable … not resource)
- PE Import extraction
- Digital Signature checking (code signing with X509 Cert)
- Additional info (section names, section sizes, etc).
- Scoring

PE Anomalies – Overlapping Header



Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	4D	5A	4B	45	52	4E	45	4C	33	32	2E	44	4C	4C	00	00	MZKERNEL32.DLL..
00000010	50	45	00	00	4C	01	03	00	BE	B0	11	40	00	AD	50	FF	PE..L...¾°.@.-Pÿ
00000020	76	34	EB	7C	48	01	0E	01	0B	01	4C	6F	61	64	4C	69	v4ë|H.....LoadLi
00000030	62	72	61	72	79	41	00	00	18	10	00	00	10	00	00	00	braryA.........
00000040	00	20	00	00	00	00	40	00	00	10	00	00	00	02	00	00	. ....@.........
00000050	04	00	00	00	00	00	39	00	04	00	00	00	00	00	00	00	......9.........
00000060	00	00	01	00	00	02	00	00	00	00	00	00	02	00	00	00	....
00000070	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00	....
00000080	00	00	00	00	0A	00	00	00	00	00	00	00	00	00	00	00	....
00000090	EE	F1	00	00	14	00	00	00	00	00	00	00	00	00	00	00	îñ..............
000000A0	FF	76	38	AD	50	8B	3E	BE	F0	F0	40	00	6A	27	59	F3	ÿv8-P‹>¾ðð@.j'Yó
000000B0	A5	FF	76	04	83	C8	FF	8B	DF	AB	EB	1C	00	00	00	00	¥ÿv.ƒÈÿ‹ß«ë.....
000000C0	47	65	74	50	72	6F	63	41	64	64	72	65	73	73	00	00	GetProcAddress..
000000D0	00	00	00	00	00	00	00	00	40	AB	40	B1	04	F3	AB	C1	........@«@±.ó«Á
000000E0	E0	0A	B5	1C	F3	AB	8B	7E	0C	57	51	E9	93	6A	00	00	à.µ.ó«‹~.WQé“j..
000000F0	56	10	E2	E3	B1	04	D3	E0	03	E8	8D	53	18	33	C0	55	V.âã±.Óà.è‹S.3ÀU
00000100	40	51	D3	E0	8B	EA	91	FF	56	4C	99	59	D1	E8	13	D2	@QÓà‹ê‘ÿVL™YÑè.Ò
00000110	E2	FA	5D	03	EA	45	59	89	6B	08	56	8B	F7	2B	F5	F3	âú].êEY‰k.V‹÷+õó
00000120	A4	AC	5E	B1	80	AA	3B	7E	34	0F	82	AC	FE	FF	FF	58	¤¬^±€ª;~4.‚¬þÿÿX
00000130	5F	59	E3	1B	8A	07	47	04	18	3C	02	73	F7	8B	07	3C	_Yã.Š.G..<.s÷‹.<
00000140	00	75	F3	B0	00	0F	C8	03	46	38	2B	C7	AB	E2	E5	5E	.uó°..È.F8+Ç«âå^
00000150	5D	59	46	AD	85	C0	74	1F	51	56	97	FF	D1	93	AC	84	]YF-…Àt.QV—ÿÑ“¬„
00000160	C0	75	FB	38	06	74	EA	8B	C6	79	05	46	33	C0	66	AD	Àuû8.têŠÆy.F3Àf-
00000170	50	53	FF	D5	AB	EB	E7	C3	00	60	00	00	00	10	00	00	PSÿÕ«ëçÃ.`......
00000180	F0	01	00	00	10	00	00	00	00	70	40	00	3B	7B	40	00	ð........p@.;{@.
00000190	85	00	00	00	60	00	00	E0	00	10	40	00	70	7B	40	00	…...`..à..@.p{@.
000001A0	00	80	00	00	00	70	00	00	98	0C	00	00	00	02	00	00	.€...p..˜.......
000001B0	CB	11	40	00	FF	5F	40	00	98	7C	40	00	60	00	00	E0	Ë.@.ÿ_@.˜|@.`..à
000001C0	4C	62	40	00	FC	0F	40	00	00	10	00	00	00	F0	00	00	Lb@.ü.@......ð..
000001D0	F0	01	00	00	10	00	00	00	08	7B	40	00	0B	7B	40	00	ð........{@..{@.
000001E0	1A	7B	40	00	60	00	00	E0	28	00	00	00	BE	00	00	00	.{@.`..à(...¾...
000001F0	00	00	00	00	00	00	00	00	00	00	02	00	00	00	E8	11	..............è.
00000200	18	B1	84	CC	52	A8	BF	C1	F0	E1	6F	46	21	A4	CB	3E	.±„ÌR¨¿Áðáof!¤Ë>

- DOS Header is actually overlapped by NT Headers
- Loader allows for this (fields in header are ignored)
- Module and functions are scattered!
- What compiler does THIS???

Roaming Signature – Multiple detections

- Signature detected in 2 sections
  - .data
  - .rsrc
- Why? Embedded executable ALSO packed!!

```
- <Section>
    <Name>.rsrc</Name>
    <Type>Resource</Type>
    <LengthInBytes>81920</LengthInBytes>
    <DetectedCharacteristics>Read</DetectedCharacteristics>
  - <Entropy>
      <AverageValue>1.1230640</AverageValue>
    </Entropy>
  - <DetectedSignatureKeys>
      <string>Aspack v 1.02X - 1.08X</string>
    </DetectedSignatureKeys>
  </Section>
- <Section>
    <Name>.data</Name>
    <Type>None</Type>
    <LengthInBytes>4096</LengthInBytes>
    <DetectedCharacteristics>Read Write</DetectedCharacteristics>
  - <Entropy>
      <AverageValue>0.3580073</AverageValue>
    </Entropy>
  - <DetectedSignatureKeys>
      <string>Aspack v 1.02X - 1.08X</string>
    </DetectedSignatureKeys>
  </Section>
</Sections>
```

Demonstration

Unpacking

- Ollydbg → OllyScript → OllyDump
- Ollydbg → bp in Library → OllyDump
- UnFSG, upx, etc
- PEiD
- ProcDump
- OEPFinder
- etc…

Other Unpackers

- Ollydbg with the Ollydump plugin and a variety of OllyScripts *
- IDAPro with the “Universal Unpacker Plugin”.
- Generic Unpacker Win32 by Christoph Gabler. 31.VII.2001.
  Win32 Intro by Vitaly Evseenko. 21.IX.1999.
- UN-PACK by Snow Panther. 21.IV.2003.
- UNPE-SHiELD by G-RoM. 1.VI.1999 de-CodeCrypt by xOANINO.
  10.V.2000.
- Ni2Untelock by Ni2. 31.XII.2000.
- DeYoda by C-ripper. 18.II.2001.
- UnPEProt by Lorian. 23.I.1999.
- DePE-PACK by Unknown One. 03.V.2002.
- Un-FSG by SMoKE. 12.I.2003.
- un-ASPack by dtg. 26.VIII.1999.
- StealthKiller by Snow Panther. 04.IX.2002.

Questions?

kris.kendall@mandiant.com
chad.mcmillan@mandiant.com