Additional investigations

Application forensics

Thumbnail files

Recycle Bin and previous versions

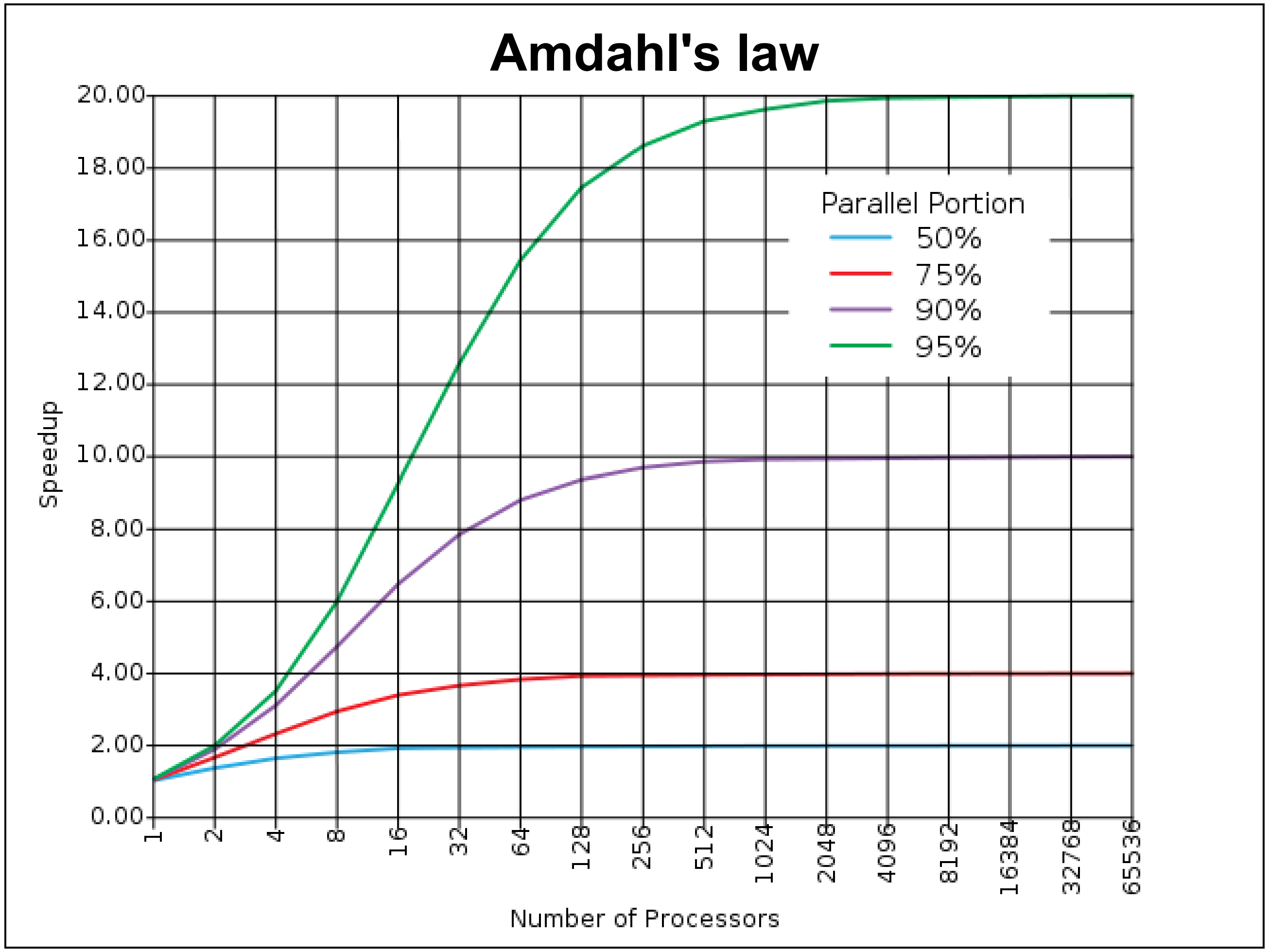
FTK - Link and Spool files

Processer och trådar, CPU

- Olika kategorier av OS - general purpouse, real-tid, embedded
- Operativsystemet hanterar hela tiden ett antal processer och trådar, dvs. applikationer och drivrutiner mm.
- CPU:n växlar mellan de processer/trådar som behöver service utifrån operativsystemets skedulerare (prioritet används oftast), växlingarna sker oerhört snabbt vilket ger sken av samtidighet
- Datorer med single core (en kärna) kan endast köra en process/tråd åt gången, dvs. inga äkta parallella operationer
- Avbrott (interrupt) kan komma från en mjukvaru eller hårdvaru resurs närsomhelst
  - När datorn får ett avbrott stoppar OS:et och CPU:n all annan aktivitet för en oerhört kort stund och ger service till resursen som gjorde förfrågan (man kan därför säga att interrupt har högsta prioriteten)

Multiple processors speedup

- OBS! Gäller ett exekeverande program!
  - http://en.wikipedia.org/wiki/Amdahl%27s_law

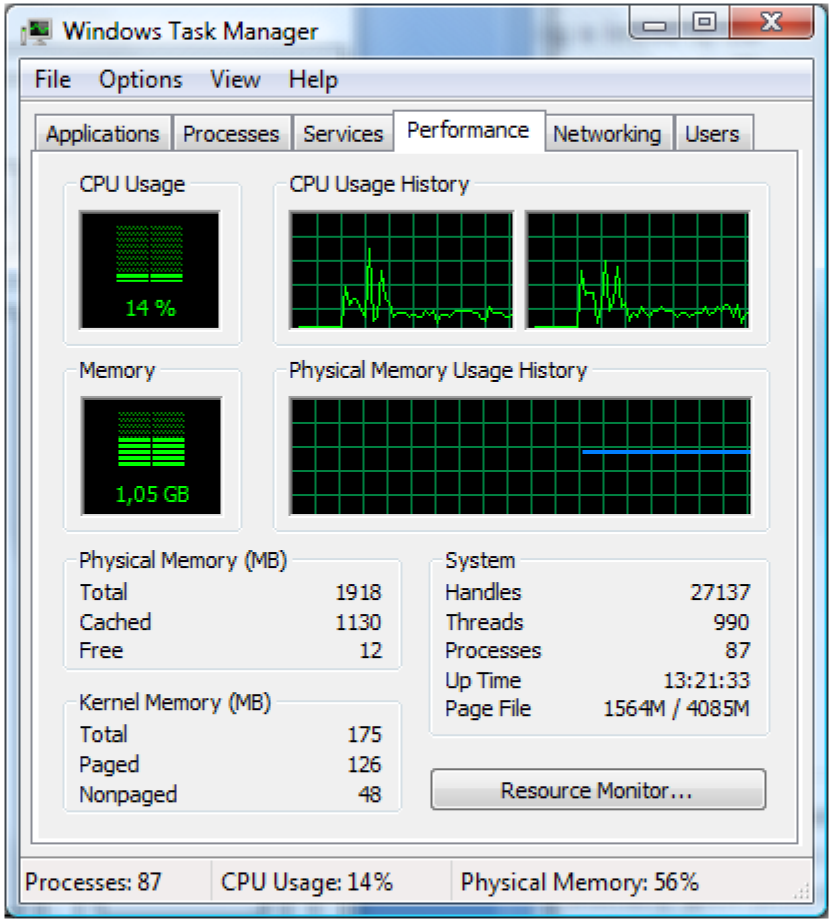


OpenMP

Easy way of implement multi threading/CPU support in already existing single threaded applications!
http://openmp.org

Parallel Extensions finns inbyggt sedan .NET 4.x och Visual Studio 2010

Processer och trådar, CPU



Physical Memory (MB)

Total	1918
Cached	1130
Free	12

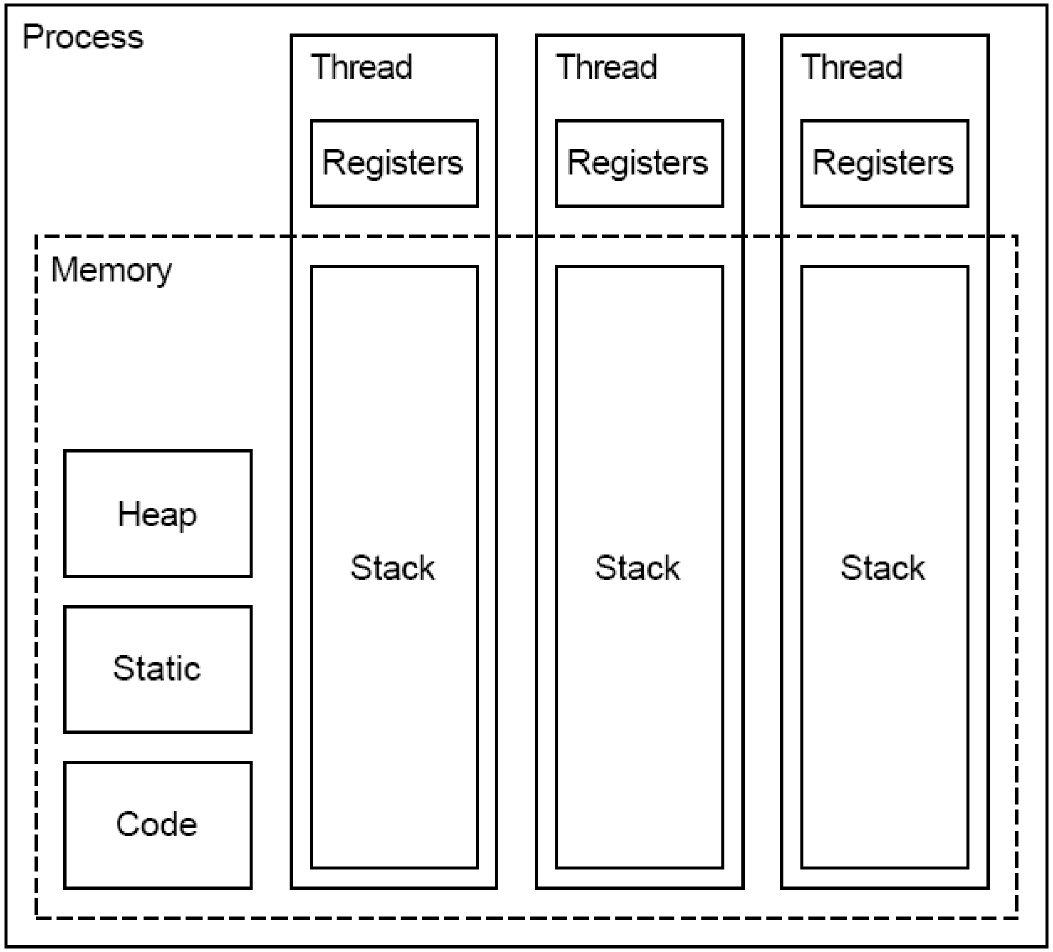
Kernel Memory (MB)

Total	175
Paged	126
Nonpaged	48

System

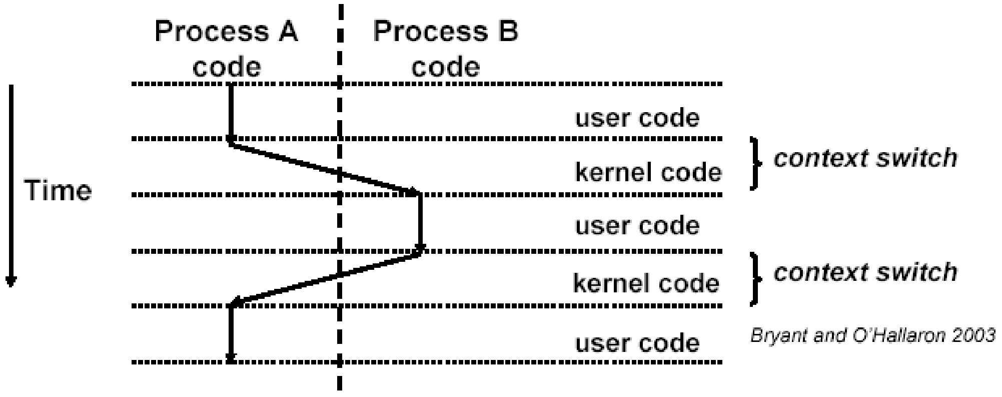
Handles	27137
Threads	990
Processes	87
Up Time	13:21:33
Page File	1564M / 4085M

Processes: 87 CPU Usage: 14% Physical Memory: 56%



Process (Task) Control Block

- Varje process tror den har exklusiv tillgång till allt minne i datorn
- Vid varje växling (context switch) måste viss information om processen sparas undan i OS:ets TCB eller PCB
- TCB/PCB lagrar bland annat:
  - Unika Process id-numret
  - Process status - (exekverande, redo, väntande, blockerad, zombie (terminerad) och prioritet
  - Registerdata och programräknare
  - Öppna filer, IPC information
  - Minnes information
  - Processägarinformation
  - SAT (Security Acces Token)
  - mm.



Threads example C#

```
using System;
using System.Threading;

class ThreadTest {
  static void Main() {
     Thread t = new Thread (WriteY);
    t.Start();                                          // Run WriteY on the new thread
    while (true) Console.Write ("x");                   // Write 'x' forever
  }

  static void WriteY() {
    while (true) Console.Write ("y");                   // Write 'y' forever
  }
}
```

```
xxxxxxxxxxxxxxxyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyyy
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxyyyyyyyyyyyy
 yyyyyyyyyyyyyyyyyyyyyyyyyyyyyxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxyyyyyyyyyyyyyyyyyyyyyyyyyyyyy
 yyyyyyyyyyyyxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 ...
```

Output

- The main thread creates a new thread **t** on which it runs a method that repeatedly prints the character y
- Simultaneously, the main thread repeatedly prints the character x

Minneshantering

- OS tilldelar först minne till sig själv och drivrutiner (tolken mellan elektriska signaler och OS), sedan till applikationer i minnesblock så inga krockar sker
- Ett 32-bitars OS kan adressera ca: 4 GB minne, omkring 3 GB av detta är tillgängligt för applikationer. 64-bitars OS kan adressera?
  - PAE > 36 bit adressering

http://blogs.msdn.com/hiltonl/archive/2007/04/13/the-3gb-not-4gb-ram-problem.aspx

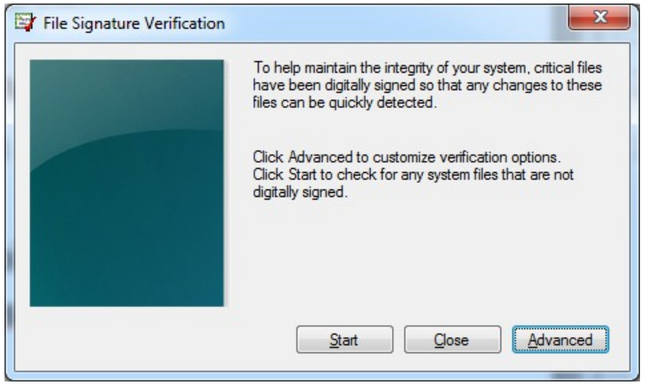
- Olika nivåer på minne
  - Register (CPUns hjärta): GPR och FPU i olika bitlängd < 100
  - L1 Cache i CPU: 32 – 64 kB
  - L2 Cache (i samma kapsel som CPUn): 256 kB
  - L3 delad multicore cache (i samma kapsel som CPUn): 6 - 8 MB
  - RAM: upp till 8 GB är vanligt
  - Virtuellt minne eller swap (fil på hårddisken) : ca (1 - 1.5) * RAM
  - Hårddisken för arkivering och lagring

Applikations forensics I

- Det är viktigt att känna till följande om processer
  - Kontexten processen kör i (user, system eller local/network service)
  - Vad det specifika processnamnet är
  - Fullständiga sökvägen till processen
  - Exekveringstiden och när processen startade
  - Öppna ”handles” (objekt som filer, registernycklar etc.)
  - Mm.
- Om man känner till vilka processer som körs kan det i många fall förklara underligheter i datorn som
  - Attackerarens åtkomst till ej tillåtna areor
  - När en attack mot användaren eller datorn började
  - Metoden för att fånga användares lösenord eller annan info

Applikations forensics II

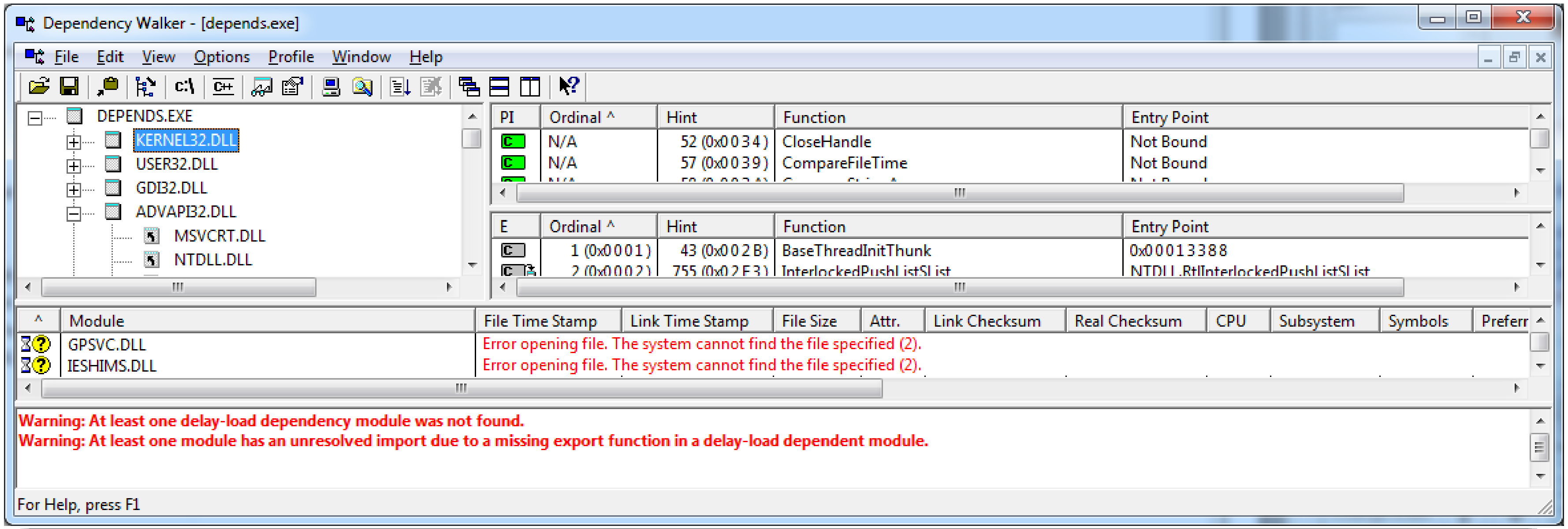
- Ta reda på kontexten (vilken användare är associerad)
  - GNU/Linux
    - ps aux, top, lsof (listar öppna filer), etc.
  - Windows
    - Task Manager, Sysinternals och MS resourcekit verktyg, etc.
- Drivrutiner och DLL:er är svåra att upptäcka
  - DLL injection
- Program som listar inlänkade bibliotek i körbara filer
  - Windows: Dependency Walker, listdlls
  - GNU/Linux: ldd <options> file
- File Signature Verification i Windows
  - Sigverif.exe hittar alla osignerade och förändrade drivrutiner och DLL:er
  - Start > run > sigverif


File Signature Verification

To help maintain the integrity of your system, critical files have been digitally signed so that any changes to these files can be quickly detected.

Click Advanced to customize verification options. Click Start to check for any system files that are not digitally signed.

Dependency Walker and Process Explorer


Dependency Walker - [depends.exe]
File Edit View Options Profile Window Help
For Help, press F1

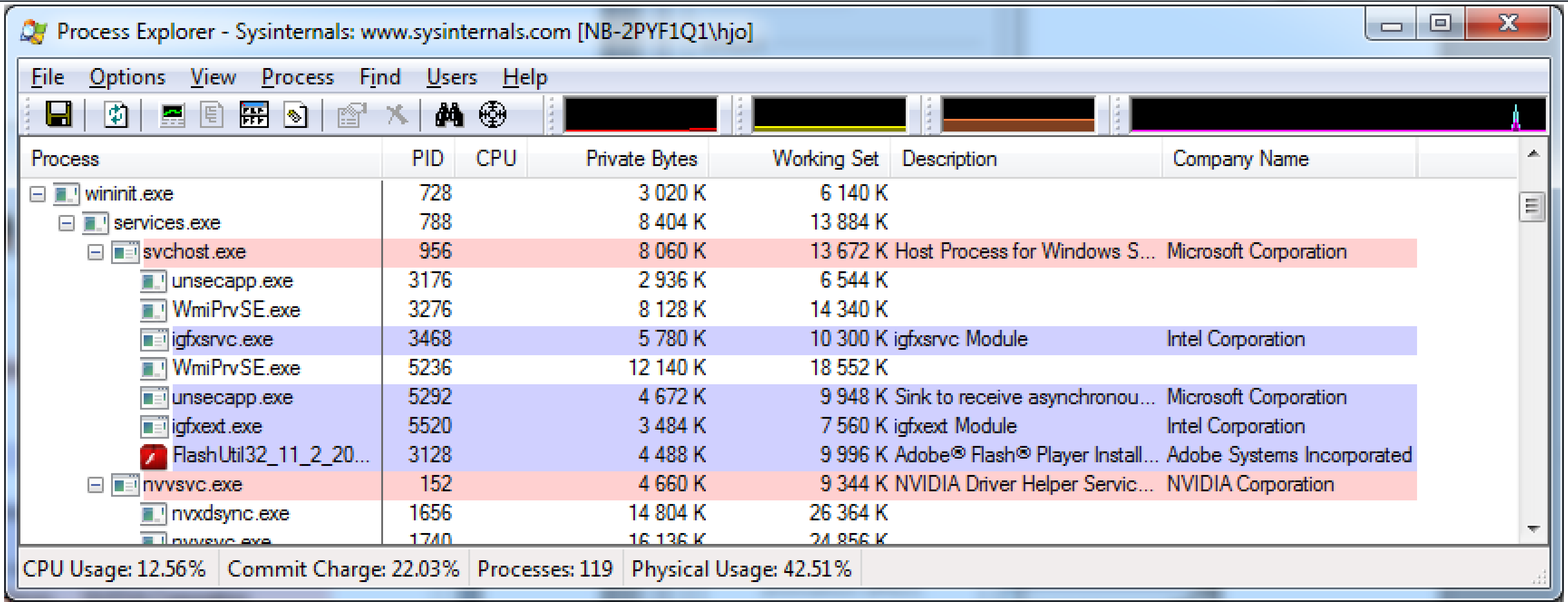
DEPENDS.EXE
- KERNEL32.DLL
- USER32.DLL
- GDI32.DLL
- ADVAPI32.DLL
  - MSVCRT.DLL
  - NTDLL.DLL

PI	Ordinal ^	Hint	Function	Entry Point
🟩C	N/A	52 (0x0034)	CloseHandle	Not Bound
🟩C	N/A	57 (0x0039)	CompareFileTime	Not Bound

E	Ordinal ^	Hint	Function	Entry Point
C	1 (0x0001)	43 (0x002B)	BaseThreadInitThunk	0x00013388
C	2 (0x0002)	755 (0x02F3)	InterlockedPushListSList	NTDLL.RtlInterlockedPushListSList

^	Module	File Time Stamp	Link Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CPU	Subsystem	Symbols	Preferr
	GPSVC.DLL	Error opening file. The system cannot find the file specified (2).									
	IESHIMS.DLL	Error opening file. The system cannot find the file specified (2).									

Warning: At least one delay-load dependency module was not found.
Warning: At least one module has an unresolved import due to a missing export function in a delay-load dependent module.


Process Explorer - Sysinternals: www.sysinternals.com [NB-2PYF1Q1\hjo]
File Options View Process Find Users Help

Process	PID	CPU	Private Bytes	Working Set	Description	Company Name
wininit.exe	728		3 020 K	6 140 K		
services.exe	788		8 404 K	13 884 K		
svchost.exe	956		8 060 K	13 672 K	Host Process for Windows S...	Microsoft Corporation
unsecapp.exe	3176		2 936 K	6 544 K		
WmiPrvSE.exe	3276		8 128 K	14 340 K		
igfxsrvc.exe	3468		5 780 K	10 300 K	igfxsrvc Module	Intel Corporation
WmiPrvSE.exe	5236		12 140 K	18 552 K		
unsecapp.exe	5292		4 672 K	9 948 K	Sink to receive asynchronou...	Microsoft Corporation
igfxext.exe	5520		3 484 K	7 560 K	igfxext Module	Intel Corporation
FlashUtil32_11_2_20...	3128		4 488 K	9 996 K	Adobe® Flash® Player Install...	Adobe Systems Incorporated
nvvsvc.exe	152		4 660 K	9 344 K	NVIDIA Driver Helper Servic...	NVIDIA Corporation
nvxdsync.exe	1656		14 804 K	26 364 K		
nvvsvc.exe	1740		16 136 K	24 856 K		

CPU Usage: 12.56% | Commit Charge: 22.03% | Processes: 119 | Physical Usage: 42.51%

Fientliga DLL:er och bakgrundstjänster

- Fientliga DLL:er kan användas till att förändra säkerheten på ett system
  - Kan gör stort sett nästan vad som helst, endast fantasin sätter gräns!
- Bakgrundstjänster och daemoner som de kallas i Unix
  - Bakgrundstjänster kör oftast som ett task (tråd) i en annan process
  - Tasklist /svc visar alla tjänster (trådar) i varje process
  - I GNU/Linux och Unix brukar man forka processen, dvs. kopiera parent till en ny (child) och sedan döda parent, child fortsätter utan konsol
- De genomgånga verktygen plus ett AV-program och anti root-kit kan lösa de allra enklaste problemen
  - MS Security Essentials, AVG, Avira och Avast är bra gratis anti-virus
  - Bakgrundsprogram i Windows (Task list programs)
    - http://www.answersthatwork.com/Tasklist_pages/tasklist.htm
  - Anti-Rootkit tools
    - RootkitRevealer (Sysinternals), Sophos Anti-Rootkit, Trend Micro RootkitBuster, McAfee Rootkit Detective, GMER, F-Secure BlackLight, **GNU/Linux:** Chkrootkit och Rootkit Hunter

Avancerad filanalys

- Statisk vs. dynamisk analys
- Beskriva vad processens funktion och vad den gör
  - Tillverkare version och innehåll (PE/COFF, ELF, MACH, packad etc.)
  - Dumpa ut strängar: strings, bintext. Systemanrop: strace, ltrace
  - Vilka registernycklar och filer den accessar
  - Kommunikation internt och/eller externt? $\}$ Handles
  - Importerade DLL:er, mm. mm.
  - Sysinternals Process Monitor
    - RegMon, FileMon, TCPMon, ProcessMon
- Reverse Code Engineering (RCE)
  - http://users.du.se/~hjo/cs/common/books/Security%20Warrior/
  - Dumpa ut processen från RAM och titta inuti binären
    - Behövs om binären är packad eller krypterad
  - Disassemblera programmet eller köra i debugger / RCE verktyg
    - IDA Pro
    - OllyDbg

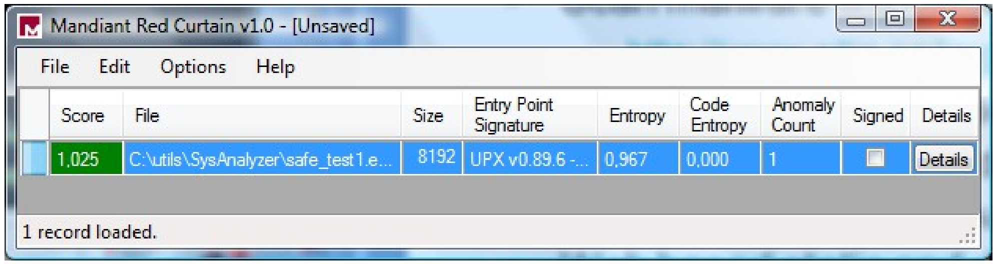
Malware analysis template

http://www.counterhack.net/malware_template.html

	Activity	Observed Results
Static analysis	Load specimen onto victim machine	
	Run antivirus program	
	Research antivirus results and file names	
	Conduct strings analysis	
	Look for scripts	
	Conduct binary analysis	
	Disassemble code	
	Reverse-compile code	
Dynamic analysis	Monitor file changes	
	Monitor file integrity	
	Monitor process activity	
	Monitor local network activity	
	Scan for open ports remotely	
	Scan for vulnerabilities remotely	
	Sniff network activity	
	Check promiscuous mode locally	
	Check promiscuous mode remotely	
	Monitor registry activity	
	Run code with debugger	

Automatic malware analysis

- Scan the malware file with different AntiVirus agents
  - If there is an alert, research AV manufacturers websites
  - If analysis is already done – 90% of your job may be done ☺
    - AV report can be faulty, malcode may be of a new variant etc.
- Web based examples of static and dynamic analysis
  - http://www.virustotal.com - Using all AV-agents?
  - http://www.sunbeltsecurity.com (http://www.cwsandbox.org)
  - http://metascan-online.com/
- Mandiant Red Curtain – very similar to Cerberus in FTK
  - http://www.mandiant.com/mrc
- Search on subject...
- …
- ethical-hacker.net > Blog
  - http://ethicalhackernet.blogspot.com/2008_04_01_archive.html

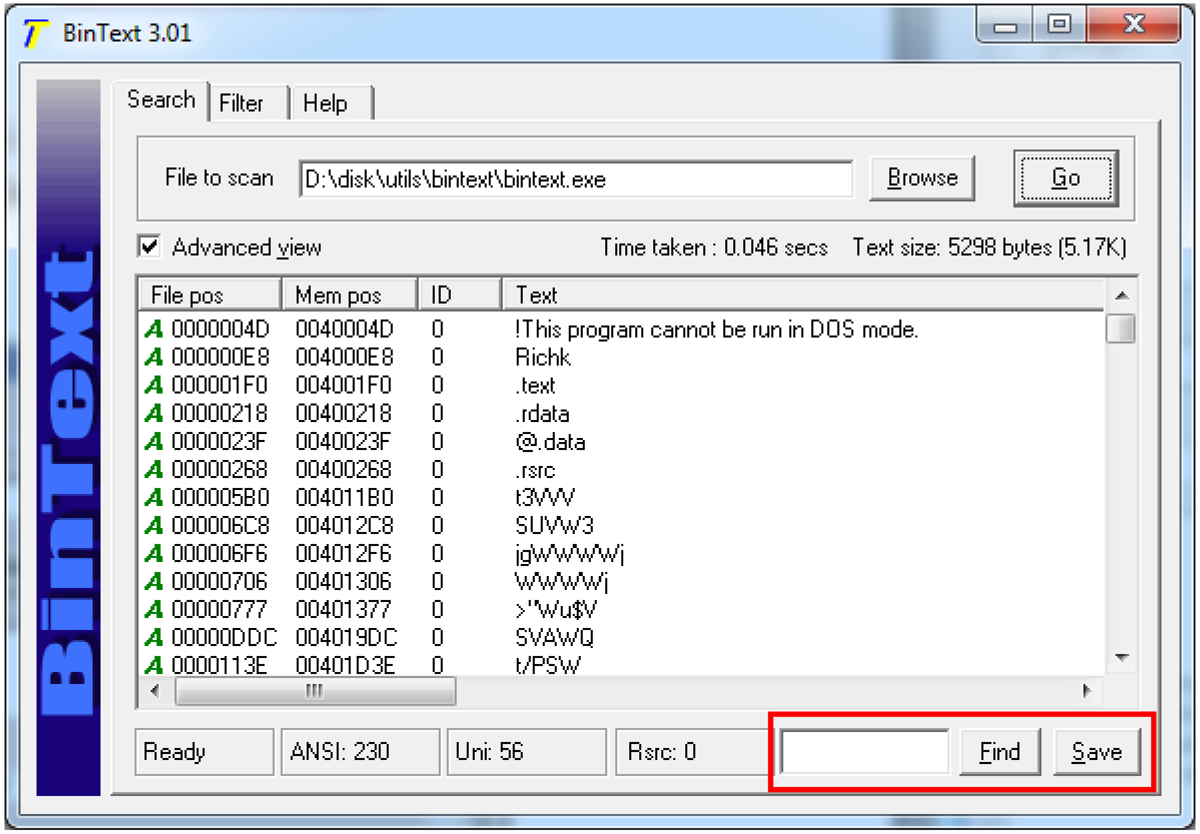


Score	File	Size	Entry Point Signature	Entropy	Code Entropy	Anomaly Count	Signed	Details
1,025	C:\utils\SysAnalyzer\safe_test1.e...	8192	UPX v0.89.6 -...	0,967	0,000	1	☐	Details

1 record loaded.

Strings och Bintext

Undersök vilka textsträngar som finns i binärfilen
Dumpa ut alla potentiella strängar och sök efter en specifik sträng
Exempel



File to scan: D:\disk\utils\bintext\bintext.exe

☑ Advanced view   Time taken : 0.046 secs   Text size: 5298 bytes (5.17K)

```
File pos     Mem pos    ID   Text
A 0000004D   0040004D   0    !This program cannot be run in DOS mode.
A 000000E8   004000E8   0    Richk
A 000001F0   004001F0   0    .text
A 00000218   00400218   0    .rdata
A 0000023F   0040023F   0    @.data
A 00000268   00400268   0    .rsrc
A 000005B0   004011B0   0    t3VVV
A 000006C8   004012C8   0    SUVW3
A 000006F6   004012F6   0    jgWWWWj
A 00000706   00401306   0    WWWWj
A 00000777   00401377   0    >"Wu$V
A 00000DDC   004019DC   0    SVAWQ
A 0000113E   00401D3E   0    t/PSW
```

Ready   ANSI: 230   Uni: 56   Rsrc: 0

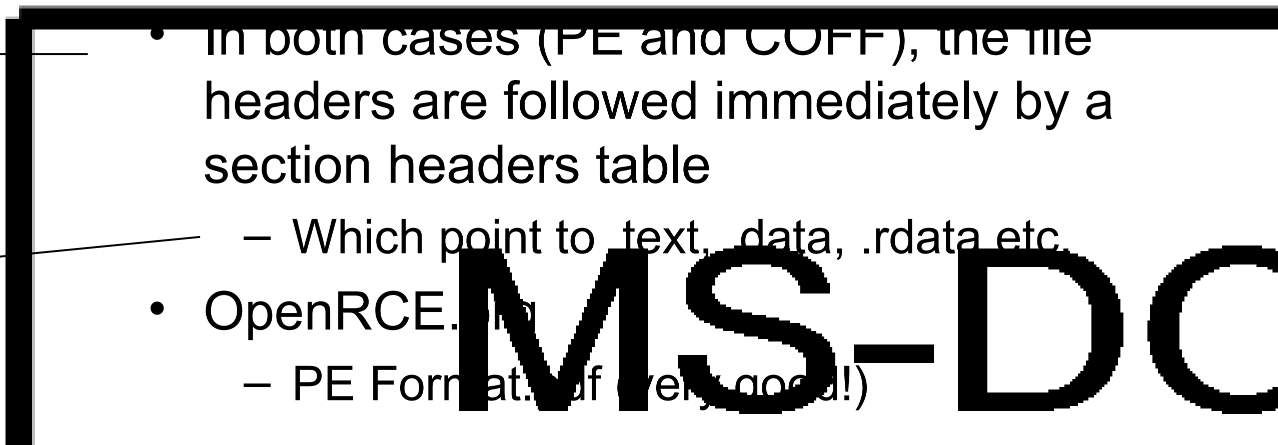
Microsoft PE format

Microsoft Portable Executable and Common Object File Format Specification

http://www.microsoft.com/whdc/system/platform/firmware/PECOFF.mspx

File offset and RVA 0

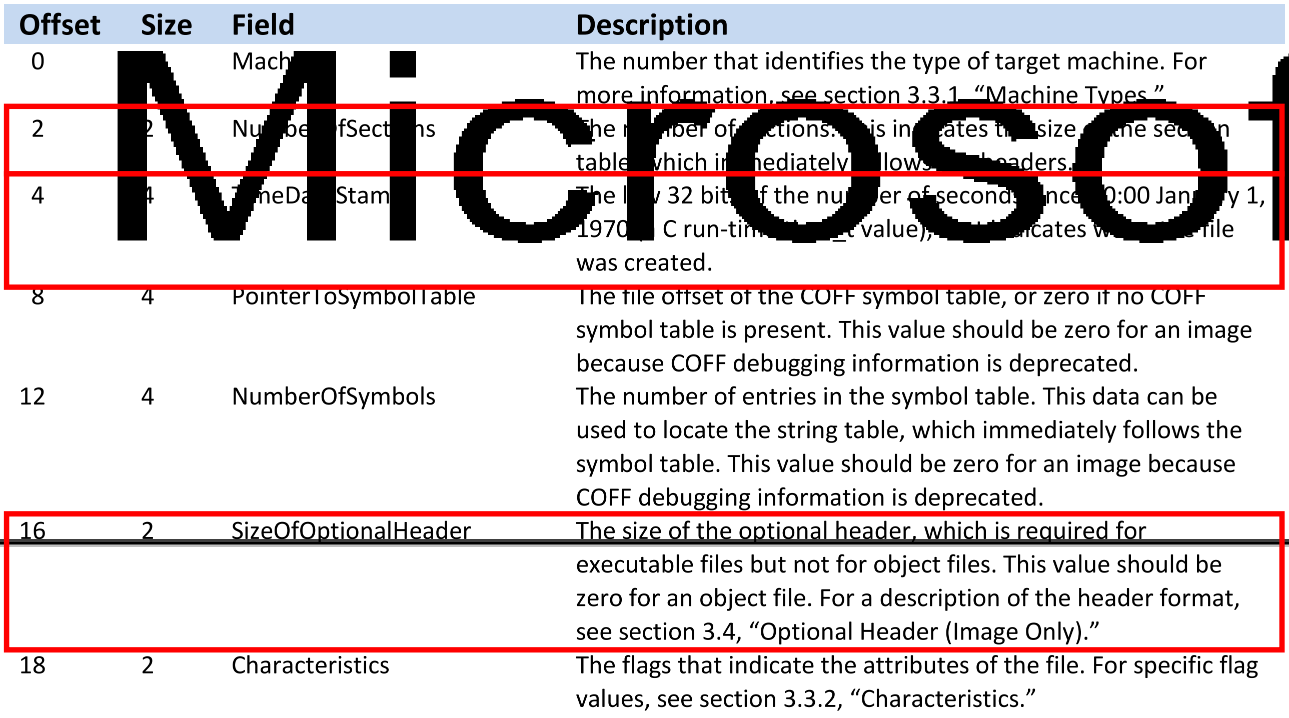
- Portable EXE File Layout
  - Not architecture specific
- The PE file header consists of a
  - MS DOS stub (IMAGE_DOS_HEADER)
  - IMAGE_NT_HEADERS
    - The PE signature (DWORD, PE)
    - The COFF file header (IMAGE_FILE_HEADER)
    - And a **not** so optional header (IMAGE_OPTIONAL_HEADER)
- In both cases (PE and COFF), the file headers are followed immediately by a section headers table
  - Which point to .text, .data, .rdata etc.
- OpenRCE.org
  - PE Format.pdf (very good!)



Microsoft PE/COFF format

- Common Object File Format
  - PE structure is derived from COFF
- A COFF object file header consists of a
  - PE/COFF file header (IMAGE_FILE_HEADER)
  - And the optional header (IMAGE_OPTIONAL_HEADER)

PE/COFF
IMAGE_FILE_HEADER

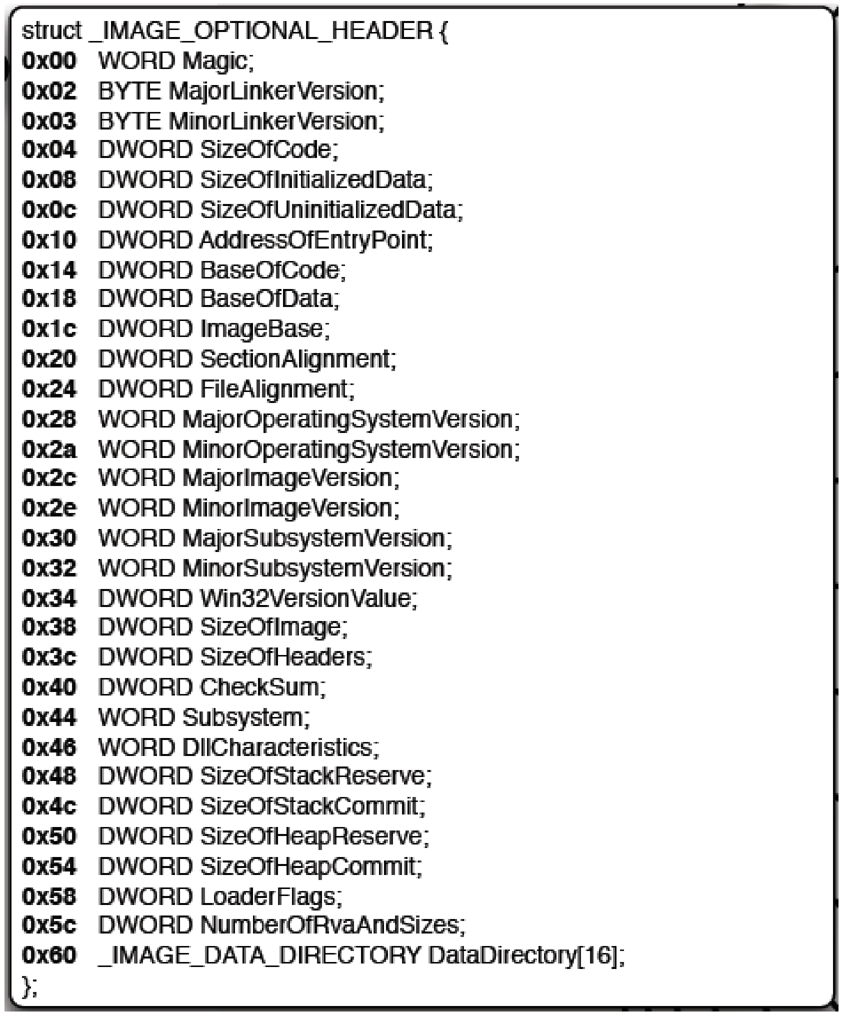


Offset	Size	Field	Description
0	2	Machine	The number that identifies the type of target machine. For more information, see section 3.3.1, "Machine Types."
2	2	NumberOfSections	The number of sections. This indicates the size of the section table, which immediately follows the headers.
4	4	TimeDateStamp	The low 32 bits of the number of seconds since 00:00 January 1, 1970 (a C run-time time_t value), that indicates when the file was created.
8	4	PointerToSymbolTable	The file offset of the COFF symbol table, or zero if no COFF symbol table is present. This value should be zero for an image because COFF debugging information is deprecated.
12	4	NumberOfSymbols	The number of entries in the symbol table. This data can be used to locate the string table, which immediately follows the symbol table. This value should be zero for an image because COFF debugging information is deprecated.
16	2	SizeOfOptionalHeader	The size of the optional header, which is required for executable files but not for object files. This value should be zero for an object file. For a description of the header format, see section 3.4, "Optional Header (Image Only)."
18	2	Characteristics	The flags that indicate the attributes of the file. For specific flag values, see section 3.3.2, "Characteristics."

Microsof

Microsoft PE/COFF format

- Optional header (IMAGE_OPTIONAL_HEADER)
  - Magic - 32/64 bit application
  - Address Of Entry Point
  - Base of Code and Data
  - Image Base
  - Subsystem, Dll Characteristics
  - Etc...
- IMAGE_DATA_DIRECTORY
  - Size and RVA to
    - [0] Export table
    - [1] Import Descriptor Table
    - [12] Import Address Table
    - Etc. 16 entries in total

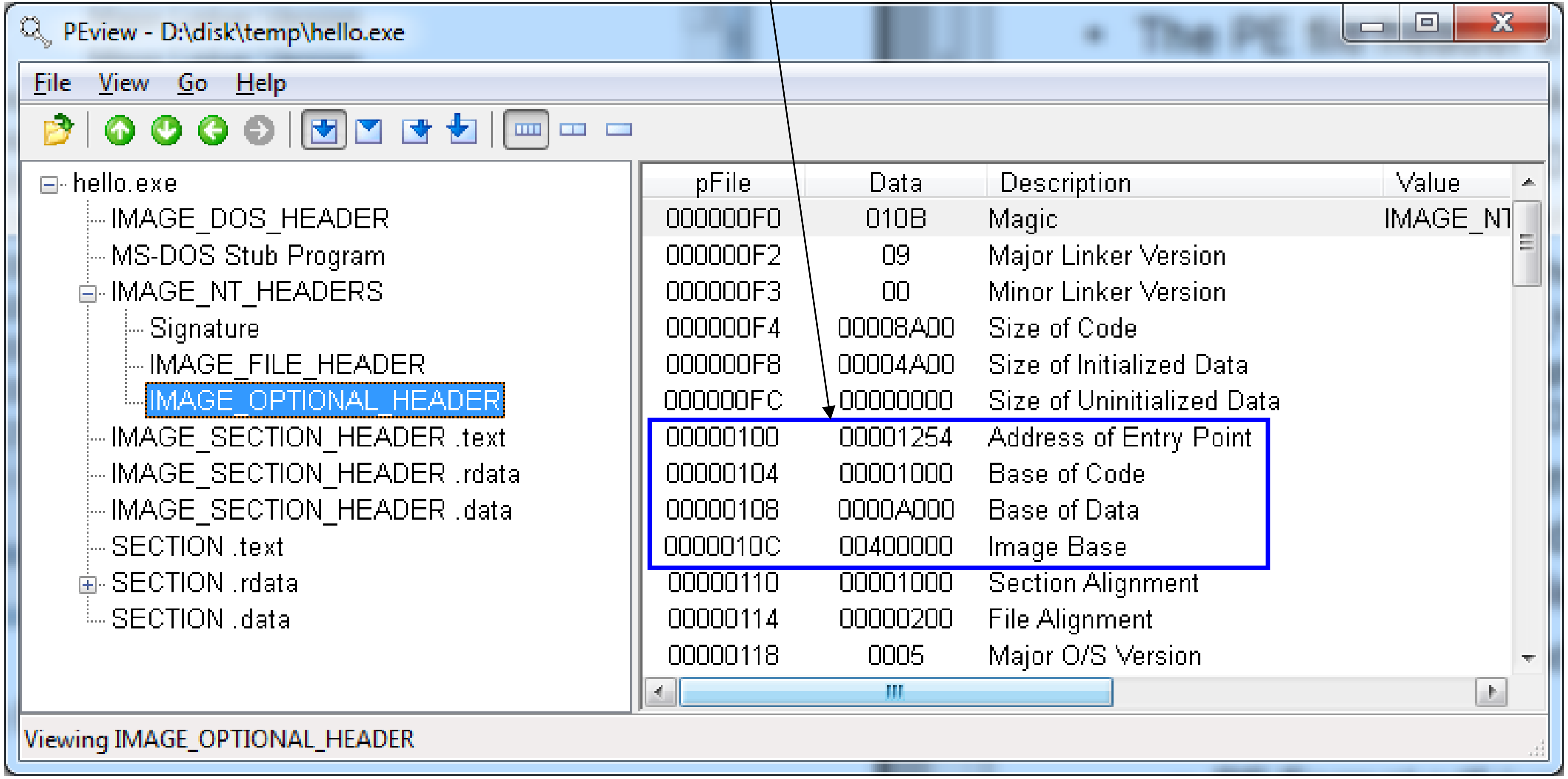


```
struct _IMAGE_OPTIONAL_HEADER {
0x00   WORD Magic;
0x02   BYTE MajorLinkerVersion;
0x03   BYTE MinorLinkerVersion;
0x04   DWORD SizeOfCode;
0x08   DWORD SizeOfInitializedData;
0x0c   DWORD SizeOfUninitializedData;
0x10   DWORD AddressOfEntryPoint;
0x14   DWORD BaseOfCode;
0x18   DWORD BaseOfData;
0x1c   DWORD ImageBase;
0x20   DWORD SectionAlignment;
0x24   DWORD FileAlignment;
0x28   WORD MajorOperatingSystemVersion;
0x2a   WORD MinorOperatingSystemVersion;
0x2c   WORD MajorImageVersion;
0x2e   WORD MinorImageVersion;
0x30   WORD MajorSubsystemVersion;
0x32   WORD MinorSubsystemVersion;
0x34   DWORD Win32VersionValue;
0x38   DWORD SizeOfImage;
0x3c   DWORD SizeOfHeaders;
0x40   DWORD CheckSum;
0x44   WORD Subsystem;
0x46   WORD DllCharacteristics;
0x48   DWORD SizeOfStackReserve;
0x4c   DWORD SizeOfStackCommit;
0x50   DWORD SizeOfHeapReserve;
0x54   DWORD SizeOfHeapCommit;
0x58   DWORD LoaderFlags;
0x5c   DWORD NumberOfRvaAndSizes;
0x60   _IMAGE_DATA_DIRECTORY DataDirectory[16];
};
```

- An In-Depth Look into the Win32 Portable Executable File Format
  - http://msdn.microsoft.com/en-us/magazine/cc301805.aspx

PEview - cons.exe

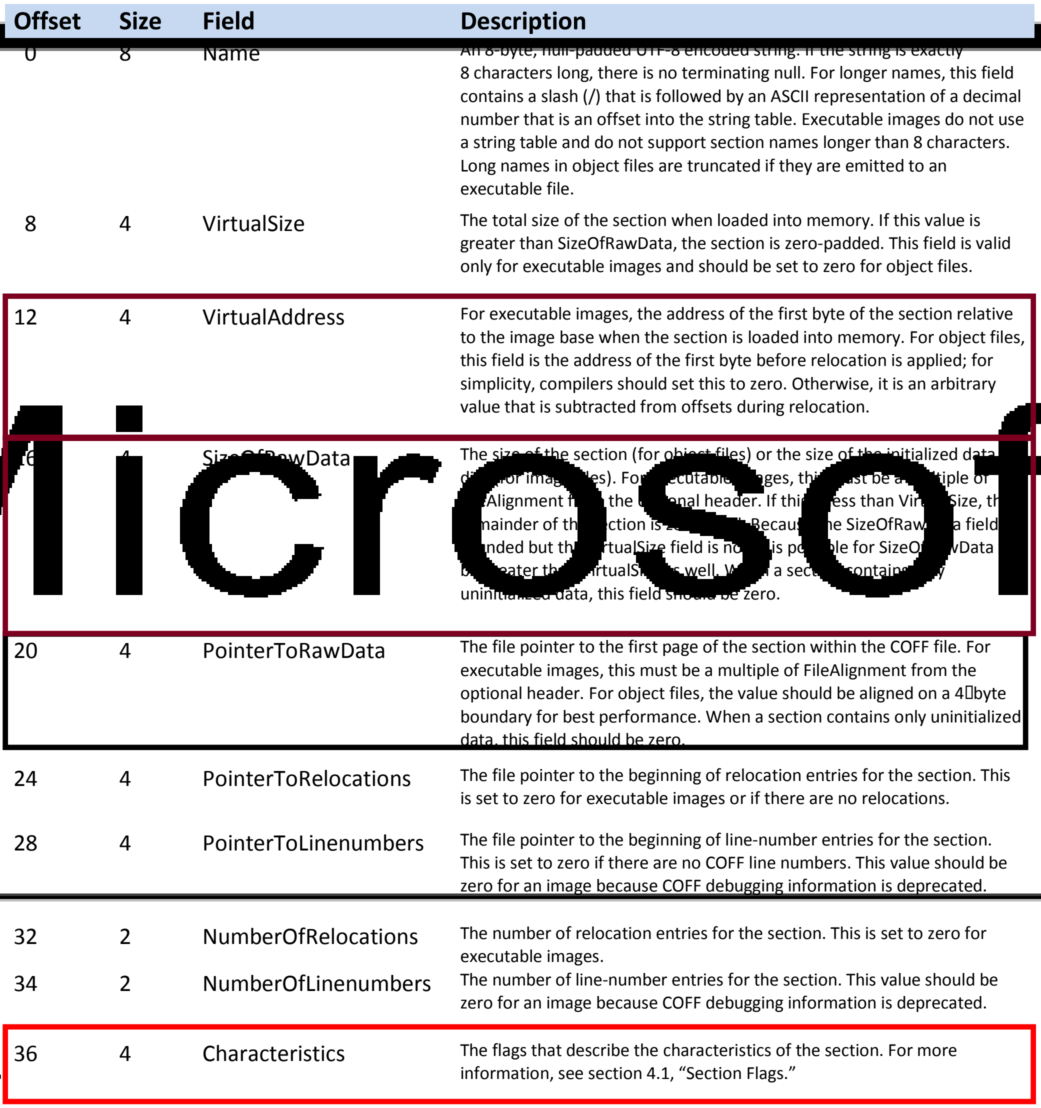
Offsets from Image Base



pFile	Data	Description	Value
000000F0	010B	Magic	IMAGE_NT
000000F2	09	Major Linker Version	
000000F3	00	Minor Linker Version	
000000F4	00008A00	Size of Code	
000000F8	00004A00	Size of Initialized Data	
000000FC	00000000	Size of Uninitialized Data	
00000100	00001254	Address of Entry Point	
00000104	00001000	Base of Code	
00000108	0000A000	Base of Data	
0000010C	00400000	Image Base	
00000110	00001000	Section Alignment	
00000114	00000200	File Alignment	
00000118	0005	Major O/S Version	

Microsoft PE/COFF format

- Section header

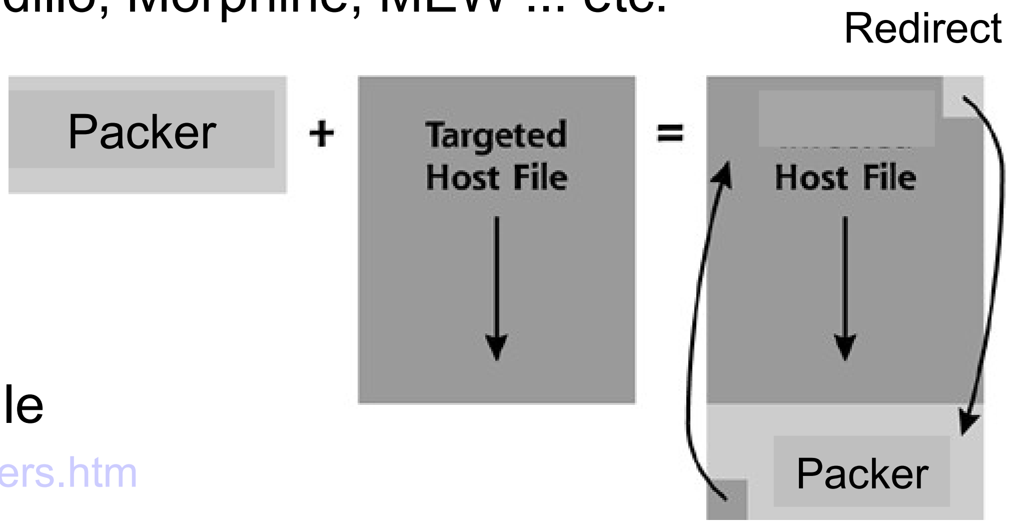


Offset	Size	Field	Description
0	8	Name	An 8-byte, null-padded UTF-8 encoded string. If the string is exactly 8 characters long, there is no terminating null. For longer names, this field contains a slash (/) that is followed by an ASCII representation of a decimal number that is an offset into the string table. Executable images do not use a string table and do not support section names longer than 8 characters. Long names in object files are truncated if they are emitted to an executable file.
8	4	VirtualSize	The total size of the section when loaded into memory. If this value is greater than SizeOfRawData, the section is zero-padded. This field is valid only for executable images and should be set to zero for object files.
12	4	VirtualAddress	For executable images, the address of the first byte of the section relative to the image base when the section is loaded into memory. For object files, this field is the address of the first byte before relocation is applied; for simplicity, compilers should set this to zero. Otherwise, it is an arbitrary value that is subtracted from offsets during relocation.
16	4	SizeOfRawData	The size of the section (for object files) or the size of the initialized data [illegible] image files). For executable images, this must be a multiple of [illegible]Alignment from the optional header. If this is less than VirtualSize, the remainder of the section is [illegible]. Because the SizeOfRawData field is rounded but the VirtualSize field is not, it is possible for SizeOfRawData to be greater than VirtualSize as well. When a section contains only uninitialized data, this field should be zero.
20	4	PointerToRawData	The file pointer to the first page of the section within the COFF file. For executable images, this must be a multiple of FileAlignment from the optional header. For object files, the value should be aligned on a 4byte boundary for best performance. When a section contains only uninitialized data, this field should be zero.
24	4	PointerToRelocations	The file pointer to the beginning of relocation entries for the section. This is set to zero for executable images or if there are no relocations.
28	4	PointerToLinenumbers	The file pointer to the beginning of line-number entries for the section. This is set to zero if there are no COFF line numbers. This value should be zero for an image because COFF debugging information is deprecated.
32	2	NumberOfRelocations	The number of relocation entries for the section. This is set to zero for executable images.
34	2	NumberOfLinenumbers	The number of line-number entries for the section. This value should be zero for an image because COFF debugging information is deprecated.
36	4	Characteristics	The flags that describe the characteristics of the section. For more information, see section 4.1, "Section Flags."

- N sections headers point out where code, data, resources etc. are stored
- Characteristics – sections flags RWX etc.
- Name can be set by programmer
- RVA = Relative Virtual Adress
- Virtual (or target) Address = RVA + Load (or Base) address

Executable (PE/COFF) obfuscation

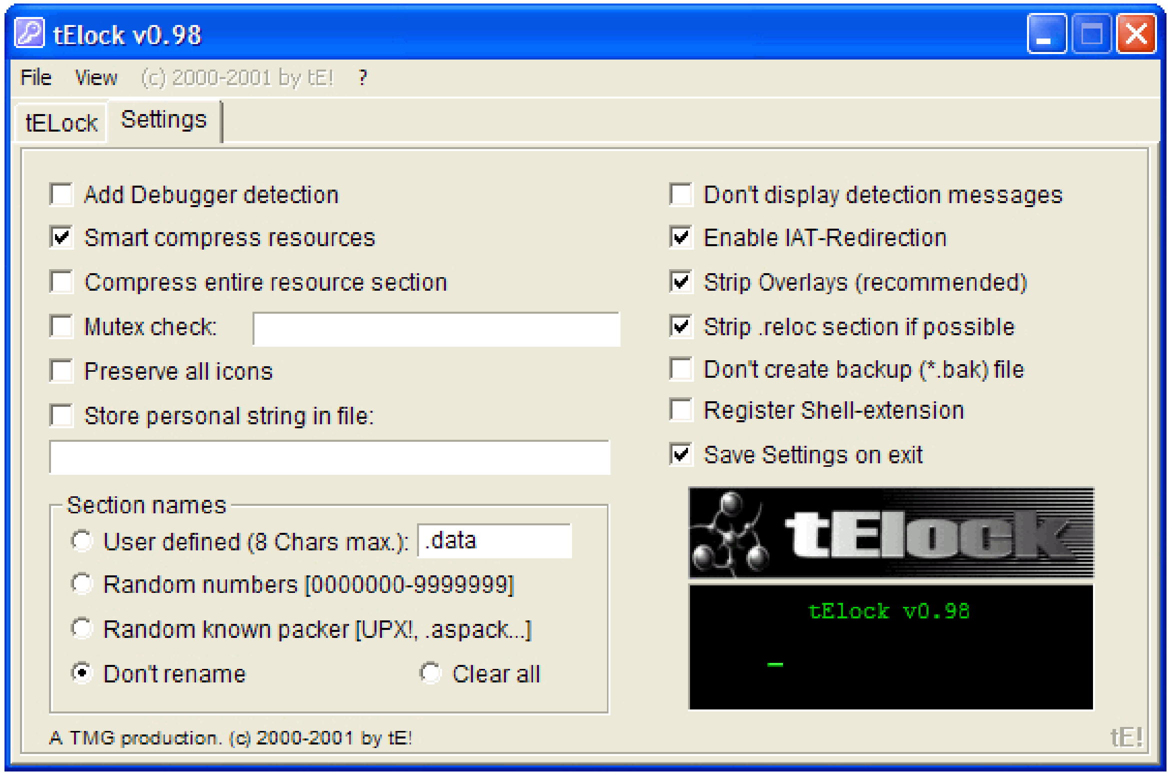
- Binders and droppers
  - Bind two applications into one, mainly used for trojans
- Packers or compressors
  - Compress the binarys sections to make it smaller and harder to detect and analyse
  - Works much like a virus appending an application and when unpacked in memory the entry point is reset to original
  - ASPack, UPX, FSG, Armadillo, Morphine, MEW ... etc.
  - Scan for section names indicating a packer
  - Special tools is neded to unpack the binary, then dump and rebuild it Image dump != MS .dmp file



http://www.woodmann.com/crackz/Packers.htm

Executable (PE/COFF) obfuscation

- Cryptors
  - As packers but with encrypted sections usually with anti-dissasembly and anti-debugging techniques, also
  - Rebuilding the import adress tables at runtime
- Example: tElock



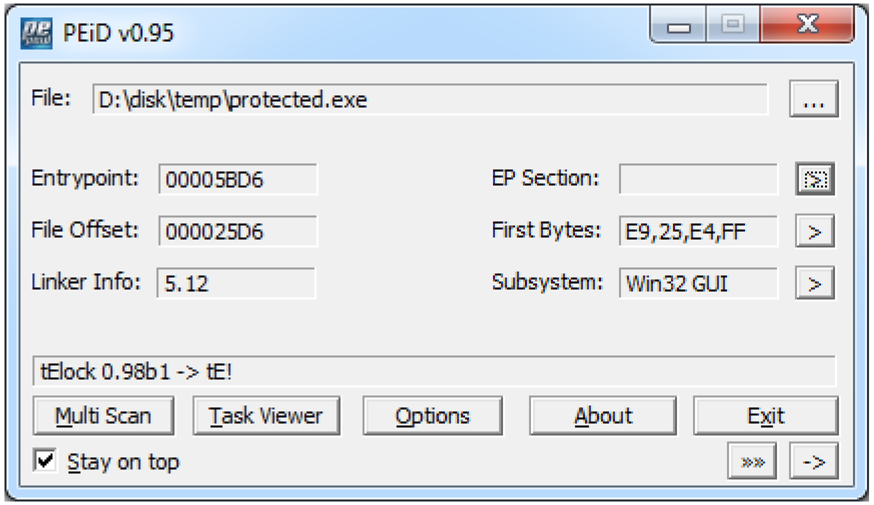
- [ ] Add Debugger detection
- [x] Smart compress resources
- [ ] Compress entire resource section
- [ ] Mutex check:
- [ ] Preserve all icons
- [ ] Store personal string in file:
- [ ] Don't display detection messages
- [x] Enable IAT-Redirection
- [x] Strip Overlays (recommended)
- [x] Strip .reloc section if possible
- [ ] Don't create backup (*.bak) file
- [ ] Register Shell-extension
- [x] Save Settings on exit

Section names
- ○ User defined (8 Chars max.): .data
- ○ Random numbers [0000000-9999999]
- ○ Random known packer [UPX!, .aspack...]
- ● Don't rename
- ○ Clear all

A TMG production. (c) 2000-2001 by tE!

PE/COFF tools...

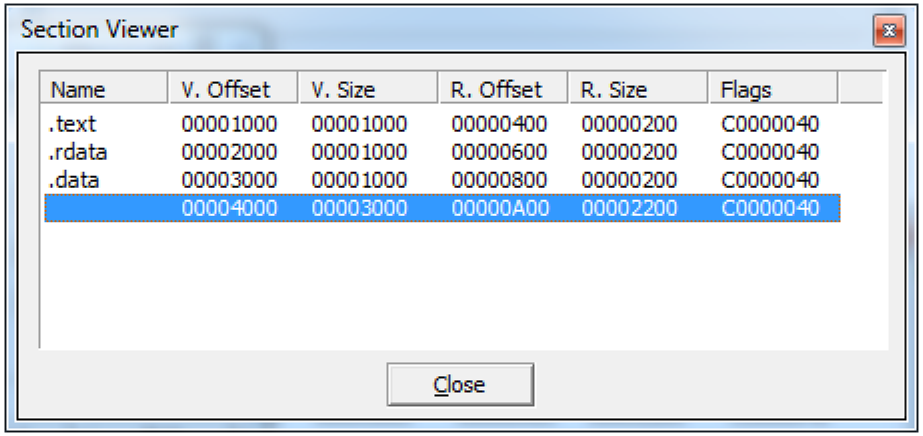
- PEiD
- PE.explorer
- PETools
- ProcDump32
- LordPE
- PEdump
- PEview
- Periscope
- FileAlyzer
- 7zip can dump PE/COFF sections to files (.data, .text etc.)
- Perl (ch6 WFA)
  - Pedmp.pl
  - Fvi.pl (resources)


PEiD v0.95

File: D:\disk\temp\protected.exe

Entrypoint: 00005BD6 EP Section:
File Offset: 000025D6 First Bytes: E9,25,E4,FF
Linker Info: 5.12 Subsystem: Win32 GUI

tElock 0.98b1 -> tE!


Section Viewer

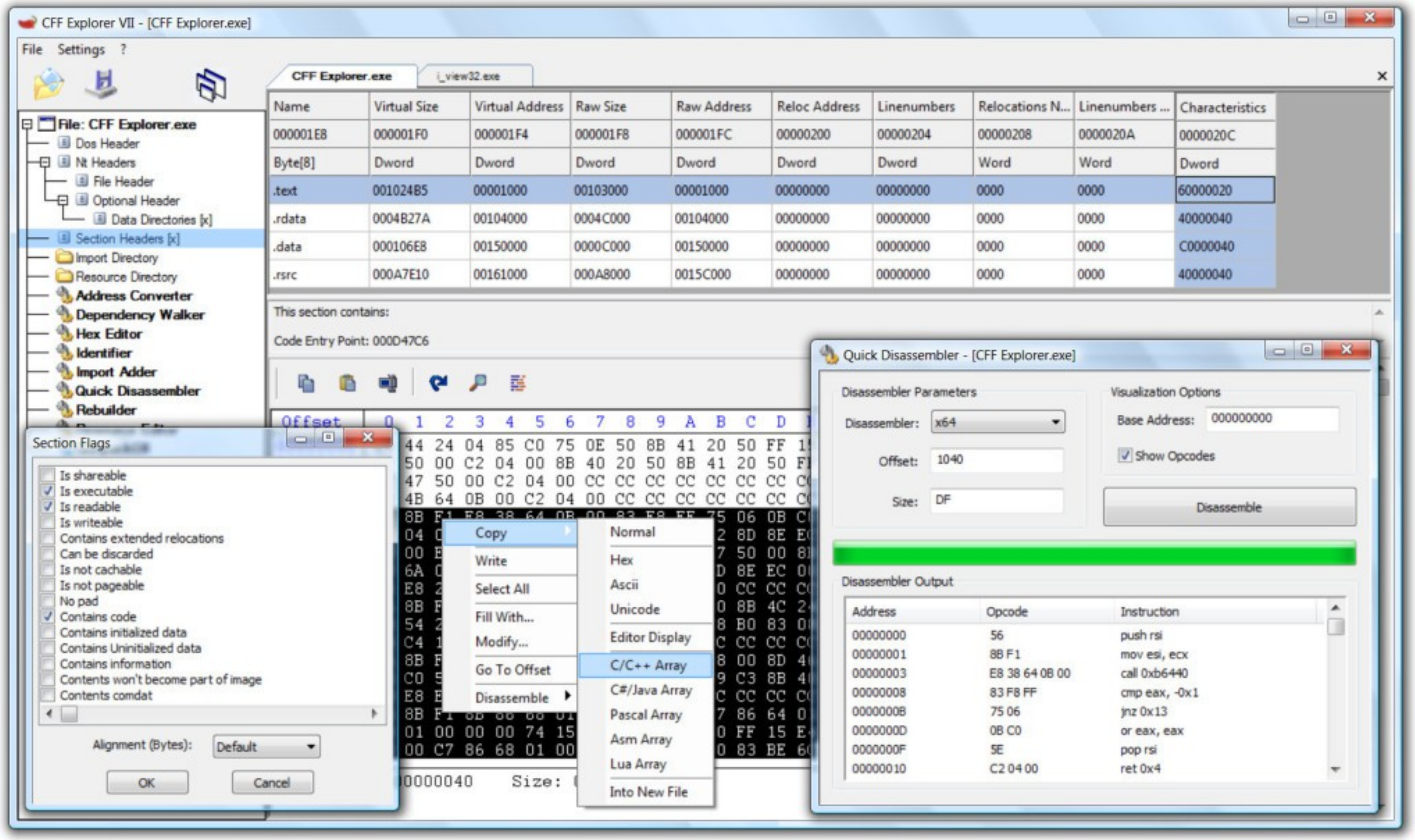
Name	V. Offset	V. Size	R. Offset	R. Size	Flags
.text	00001000	00001000	00000400	00000200	C0000040
.rdata	00002000	00001000	00000600	00000200	C0000040
.data	00003000	00001000	00000800	00000200	C0000040
	00004000	00003000	00000A00	00002200	C0000040

Address of entry point (EP) should be located in .text or .code

CFF Explorer

A freeware suite of tools. The PE editor has full support for PE32/64. Special fields description and modification (.NET supported), utilities, rebuilder, hex editor, import adder, signature scanner, signature manager, extension support, scripting, disassembler, dependency walker etc. The suite is available for x86, x64 and Itanium.

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



Name	Virtual Size	Virtual Address	Raw Size	Raw Address	Reloc Address	Linenumbers	Relocations N...	Linenumbers ...	Characteristics
000001E8	000001F0	000001F4	000001F8	000001FC	00000200	00000204	00000208	0000020A	0000020C
Byte[8]	Dword	Dword	Dword	Dword	Dword	Dword	Word	Word	Dword
.text	001024B5	00001000	00103000	00001000	00000000	00000000	0000	0000	60000020
.rdata	0004B27A	00104000	0004C000	00104000	00000000	00000000	0000	0000	40000040
.data	000106E8	00150000	0000C000	00150000	00000000	00000000	0000	0000	C0000040
.rsrc	000A7E10	00161000	000A8000	0015C000	00000000	00000000	0000	0000	40000040

This section contains:

Code Entry Point: 000D47C6

Section Flags

- [ ] Is shareable
- [✓] Is executable
- [✓] Is readable
- [ ] Is writeable
- [ ] Contains extended relocations
- [ ] Can be discarded
- [ ] Is not cachable
- [ ] Is not pageable
- [ ] No pad
- [✓] Contains code
- [ ] Contains initialized data
- [ ] Contains Uninitialized data
- [ ] Contains information
- [ ] Contents won't become part of image
- [ ] Contents comdat

Alignment (Bytes): Default

Quick Disassembler - [CFF Explorer.exe]

Disassembler Parameters

Disassembler: x64

Offset: 1040

Size: DF

Visualization Options

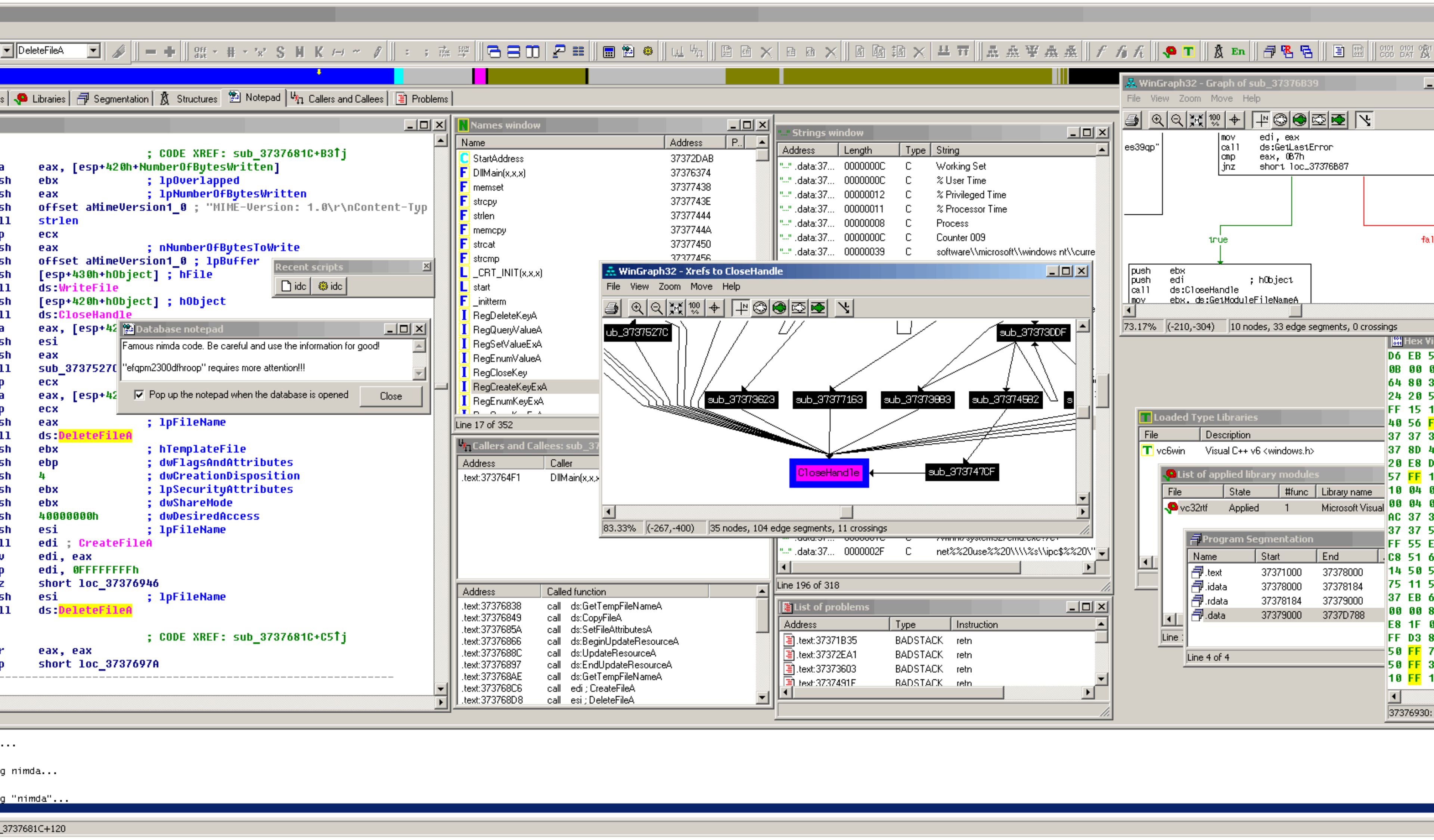
Base Address: 000000000

[✓] Show Opcodes

Disassembler Output

Address	Opcode	Instruction
00000000	56	push rsi
00000001	8B F1	mov esi, ecx
00000003	E8 38 64 0B 00	call 0xb6440
00000008	83 F8 FF	cmp eax, -0x1
0000000B	75 06	jnz 0x13
0000000D	0B C0	or eax, eax
0000000F	5E	pop rsi
00000010	C2 04 00	ret 0x4

IDA Pro disassembler (and debugger)



```
                    ; CODE XREF: sub_3737681C+B3↑j
a    eax, [esp+420h+NumberOfBytesWritten]
sh   ebx                ; lpOverlapped
sh   eax                ; lpNumberOfBytesWritten
sh   offset aMimeVersion1_0 ; "MIME-Version: 1.0\r\nContent-Typ
ll   strlen
p    ecx
sh   eax                ; nNumberOfBytesToWrite
sh   offset aMimeVersion1_0 ; lpBuffer
sh   [esp+430h+hObject] ; hFile
ll   ds:WriteFile
sh   [esp+420h+hObject] ; hObject
ll   ds:CloseHandle
a    eax, [esp+42
sh   esi
sh   eax
ll   sub_3737527C
p    ecx
a    eax, [esp+42
p    ecx
sh   eax                ; lpFileName
ll   ds:DeleteFileA
sh   ebx                ; hTemplateFile
sh   ebp                ; dwFlagsAndAttributes
sh   4                  ; dwCreationDisposition
sh   ebx                ; lpSecurityAttributes
sh   ebx                ; dwShareMode
sh   40000000h          ; dwDesiredAccess
sh   esi                ; lpFileName
ll   edi ; CreateFileA
v    edi, eax
p    edi, 0FFFFFFFFh
z    short loc_37376946
sh   esi                ; lpFileName
ll   ds:DeleteFileA

                    ; CODE XREF: sub_3737681C+C5↑j
r    eax, eax
p    short loc_3737697A
```

Database notepad

Famous nimda code. Be careful and use the information for good!

"efqpm2300dfhroop" requires more attention!!!

[✓] Pop up the notepad when the database is opened — Close

Names window

Name	Address	P..
C StartAddress	37372DAB	
F DllMain(x,x,x)	37376374	
F memset	37377438	
F strcpy	3737743E	
F strlen	37377444	
F memcpy	3737744A	
F strcat	37377450	
F strcmp	37377456	
L _CRT_INIT(x,x,x)		
L start		
F _initterm		
I RegDeleteKeyA		
I RegQueryValueA		
I RegSetValueExA		
I RegEnumValueA		
I RegCloseKey		
I RegCreateKeyExA		
I RegEnumKeyExA		

Line 17 of 352

Strings window

Address	Length	Type	String
.data:37...	0000000C	C	Working Set
.data:37...	0000000C	C	% User Time
.data:37...	00000012	C	% Privileged Time
.data:37...	00000011	C	% Processor Time
.data:37...	00000008	C	Process
.data:37...	0000000C	C	Counter 009
.data:37...	00000039	C	software\\microsoft\\windows nt\\curre
.data:37...	0000002F	C	net%%20use%%20\\\\%s\\ipc$%%20\"

Line 196 of 318

WinGraph32 - Xrefs to CloseHandle

File View Zoom Move Help

sub_3737527C, sub_37373623, sub_37377163, sub_37373883, sub_37374582, sub_373730DF, sub_373747CF, CloseHandle

83.33% (-267,-400) 35 nodes, 104 edge segments, 11 crossings

Callers and Callees: sub_37

Address	Caller
.text:373764F1	DllMain(x,x,x

Address	Called function
.text:37376838	call ds:GetTempFileNameA
.text:37376849	call ds:CopyFileA
.text:3737685A	call ds:SetFileAttributesA
.text:37376866	call ds:BeginUpdateResourceA
.text:3737688C	call ds:UpdateResourceA
.text:37376897	call ds:EndUpdateResourceA
.text:373768AE	call ds:GetTempFileNameA
.text:373768C6	call edi ; CreateFileA
.text:373768D8	call esi ; DeleteFileA

List of problems

Address	Type	Instruction
.text:37371B35	BADSTACK	retn
.text:37372EA1	BADSTACK	retn
.text:37373603	BADSTACK	retn
.text:3737491F	BADSTACK	retn

```
es39qp"
mov   edi, eax
call  ds:GetLastError
cmp   eax, 0B7h
jnz   short loc_37376B87
true                          fal
push  ebx
push  edi                     ; hObject
call  ds:CloseHandle
mov   ebx, ds:GetModuleFileNameA
```

73.17% (-210,-304) 10 nodes, 33 edge segments, 0 crossings

Loaded Type Libraries

File	Description
vc6win	Visual C++ v6 <windows.h>

List of applied library modules

File	State	#func	Library name
vc32rtf	Applied	1	Microsoft Visual

Program Segmentation

Name	Start	End
.text	37371000	37378000
.idata	37378000	37378184
.rdata	37378184	37379000
.data	37379000	3737D788

Line 4 of 4

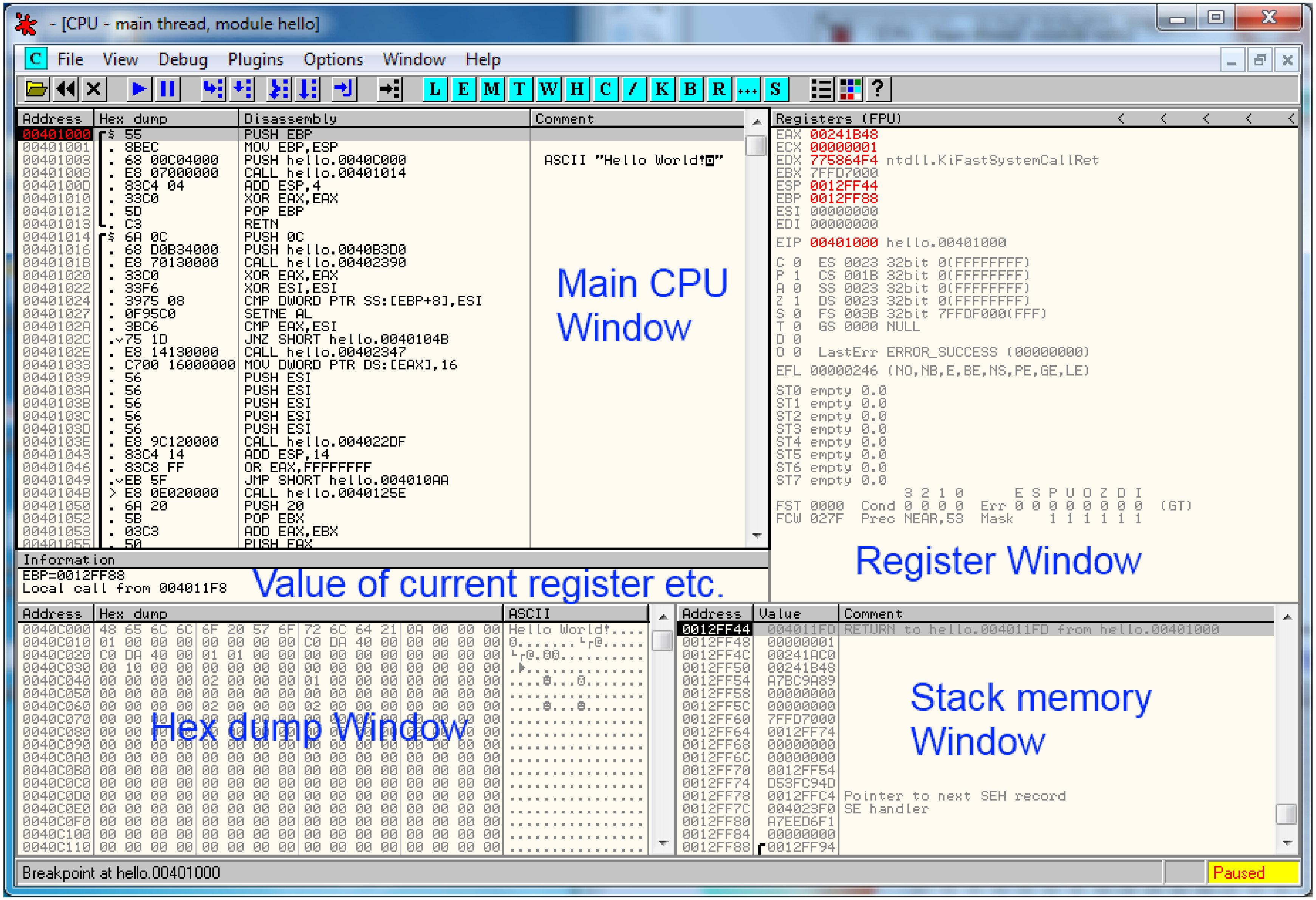
```
D6 EB 5
0B 00 0
64 80 3
24 20 5
FF 15 1
40 56 F
37 37 3
37 8D 4
20 E8 D
57 FF 1
10 04 0
00 04 0
AC 37 3
37 37 5
FF 55 E
C8 51 6
14 50 5
75 11 5
37 EB 6
00 00 8
E8 1F 0
FF D3 8
50 FF 7
50 FF 3
10 FF 1
37376930:
```

```
...
g nimda...
g "nimda"...
```

_3737681C+120

OllyDbg

OllyDbg
Win32 Symbolic Debugger


- [CPU - main thread, module hello]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S ?
Main CPU Window
Register Window
Value of current register etc.
Hex dump Window
Stack memory Window
Paused

```
Address   Hex dump          Disassembly                         Comment
00401000 ┌$ 55              PUSH EBP
00401001 │. 8BEC            MOV EBP,ESP
00401003 │. 68 00C04000     PUSH hello.0040C000                 ASCII "Hello World!◘"
00401008 │. E8 07000000     CALL hello.00401014
0040100D │. 83C4 04         ADD ESP,4
00401010 │. 33C0            XOR EAX,EAX
00401012 │. 5D              POP EBP
00401013 └. C3              RETN
00401014 ┌$ 6A 0C           PUSH 0C
00401016 │. 68 D0B34000     PUSH hello.0040B3D0
0040101B │. E8 70130000     CALL hello.00402390
00401020 │. 33C0            XOR EAX,EAX
00401022 │. 33F6            XOR ESI,ESI
00401024 │. 3975 08         CMP DWORD PTR SS:[EBP+8],ESI
00401027 │. 0F95C0          SETNE AL
0040102A │. 3BC6            CMP EAX,ESI
0040102C │.v75 1D           JNZ SHORT hello.0040104B
0040102E │. E8 14130000     CALL hello.00402347
00401033 │. C700 16000000   MOV DWORD PTR DS:[EAX],16
00401039 │. 56              PUSH ESI
0040103A │. 56              PUSH ESI
0040103B │. 56              PUSH ESI
0040103C │. 56              PUSH ESI
0040103D │. 56              PUSH ESI
0040103E │. E8 9C120000     CALL hello.004022DF
00401043 │. 83C4 14         ADD ESP,14
00401046 │. 83C8 FF         OR EAX,FFFFFFFF
00401049 │.vEB 5F           JMP SHORT hello.004010AA
0040104B │> E8 0E020000     CALL hello.0040125E
00401050 │. 6A 20           PUSH 20
00401052 │. 5B              POP EBX
00401053 │. 03C3            ADD EAX,EBX
00401055 │. 50              PUSH EAX
```

```
Registers (FPU)
EAX 00241B48
ECX 00000001
EDX 775864F4 ntdll.KiFastSystemCallRet
EBX 7FFD7000
ESP 0012FF44
EBP 0012FF88
ESI 00000000
EDI 00000000

EIP 00401000 hello.00401000

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFDF000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS (00000000)

EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0
ST5 empty 0.0
ST6 empty 0.0
ST7 empty 0.0
               3 2 1 0      E S P U O Z D I
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0 0 0 0  (GT)
FCW 027F  Prec NEAR,53  Mask    1 1 1 1 1 1
```

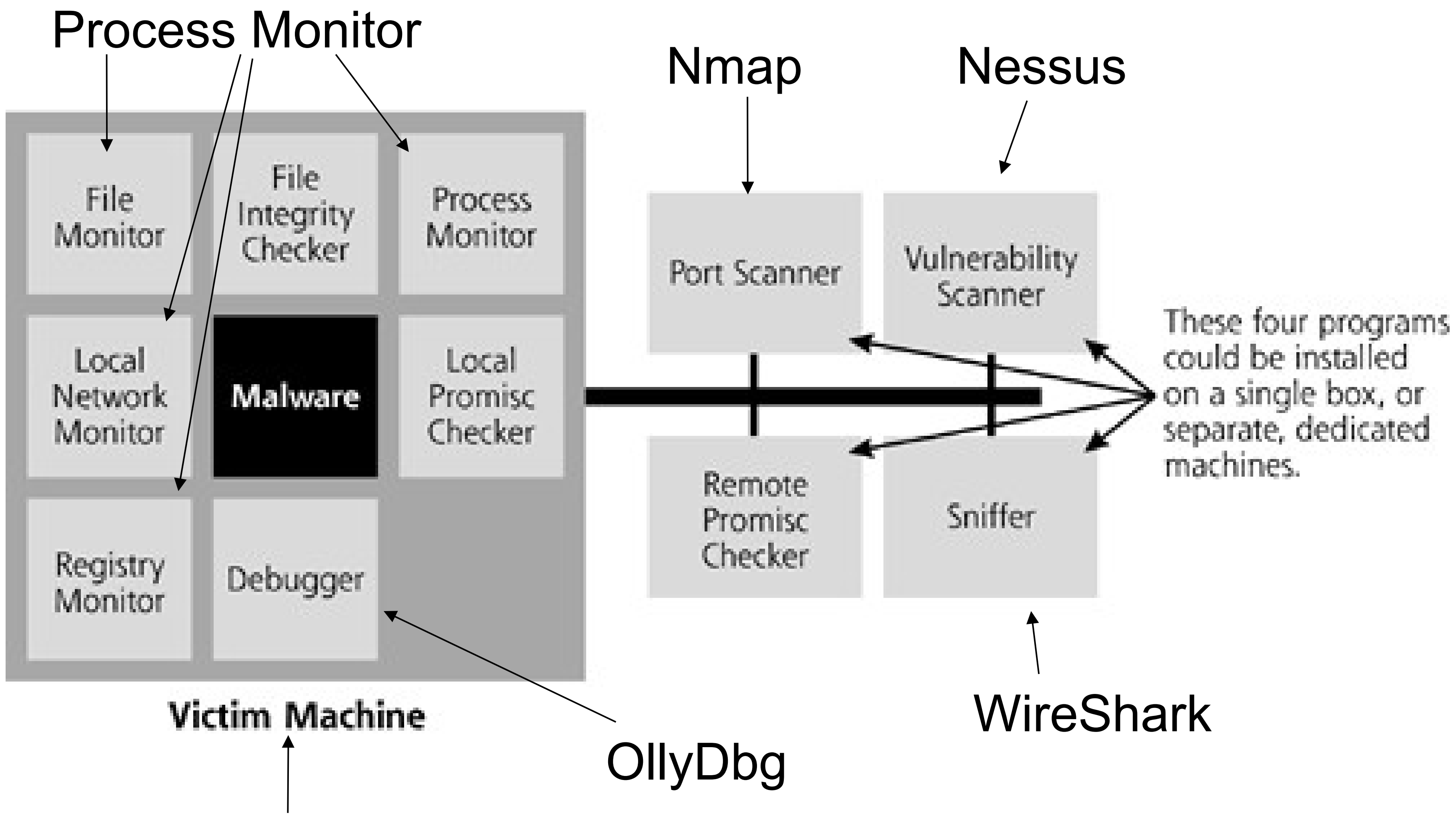
```
Information
EBP=0012FF88
Local call from 004011F8
```

```
Address   Hex dump                                         ASCII
0040C000  48 65 6C 6C 6F 20 57 6F 72 6C 64 21 0A 00 00 00  Hello World!....
0040C010  01 00 00 00 00 00 00 00 C0 DA 40 00 00 00 00 00  0.......└┌@.....
0040C020  C0 DA 40 00 01 01 00 00 00 00 00 00 00 00 00 00  └┌@.00..........
0040C030  00 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .►..............
0040C040  00 00 00 00 02 00 00 00 01 00 00 00 00 00 00 00  ....☻...☺.......
0040C050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C060  00 00 00 00 02 00 00 00 02 00 00 00 00 00 00 00  ....☻...☻.......
0040C070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C0F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040C110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
Address   Value     Comment
0012FF44  004011FD  RETURN to hello.004011FD from hello.00401000
0012FF48  00000001
0012FF4C  00241AC0
0012FF50  00241B48
0012FF54  A7BC9A89
0012FF58  00000000
0012FF5C  00000000
0012FF60  7FFD7000
0012FF64  0012FF74
0012FF68  00000000
0012FF6C  00000000
0012FF70  0012FF54
0012FF74  D53FC94D
0012FF78  0012FFC4  Pointer to next SEH record
0012FF7C  004023F0  SE handler
0012FF80  A7EED6F1
0012FF84  00000000
0012FF88 ┌0012FF94
```

```
Breakpoint at hello.00401000
```

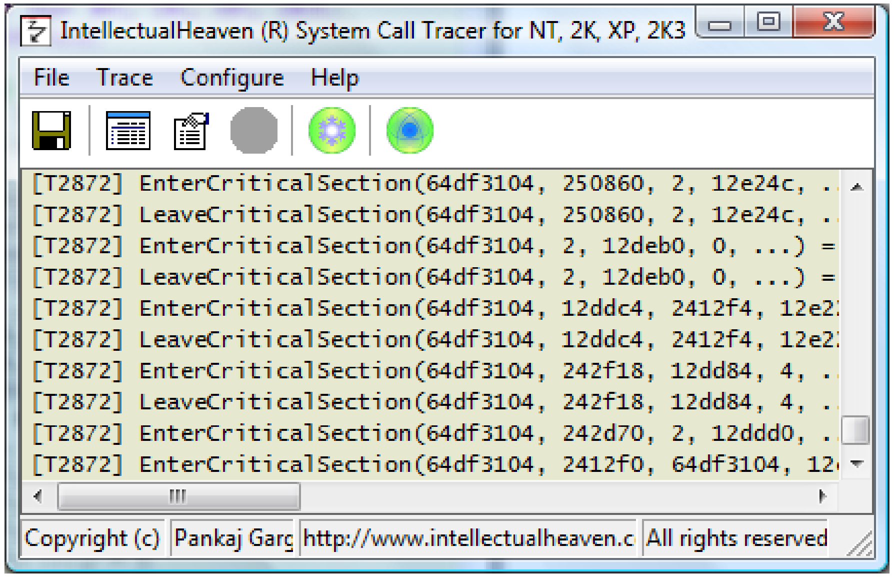
Dynamic analysis



If possible a virtual machine!
VMware, VirtualBox etc.

Additional dynamic analysis methods

- Enable auditing for process tracking in event log (failure and success events)
  - auditpol.exe /enable /process:all
- Non real-time registry or file snapshot tools as RegShot
  - Not to be used for longer time since you don't see
    - Keys or files that have been searched for
    - Timeline when keys or files were accessed
- System call trace (ported)
  - Strace NT 0.8 beta


IntellectualHeaven (R) System Call Tracer for NT, 2K, XP, 2K3
File
Trace
Configure
Help

```
[T2872] EnterCriticalSection(64df3104, 250860, 2, 12e24c, .
[T2872] LeaveCriticalSection(64df3104, 250860, 2, 12e24c, .
[T2872] EnterCriticalSection(64df3104, 2, 12deb0, 0, ...) =
[T2872] LeaveCriticalSection(64df3104, 2, 12deb0, 0, ...) =
[T2872] EnterCriticalSection(64df3104, 12ddc4, 2412f4, 12e2
[T2872] LeaveCriticalSection(64df3104, 12ddc4, 2412f4, 12e2
[T2872] EnterCriticalSection(64df3104, 242f18, 12dd84, 4, .
[T2872] LeaveCriticalSection(64df3104, 242f18, 12dd84, 4, .
[T2872] EnterCriticalSection(64df3104, 242d70, 2, 12ddd0, .
[T2872] EnterCriticalSection(64df3104, 2412f0, 64df3104, 12
```

Copyright (c) | Pankaj Garg | http://www.intellectualheaven.c | All rights reserved

Malware Analyst's Cookbook and DVD: Tools and Techniques for Fighting Malicious Code

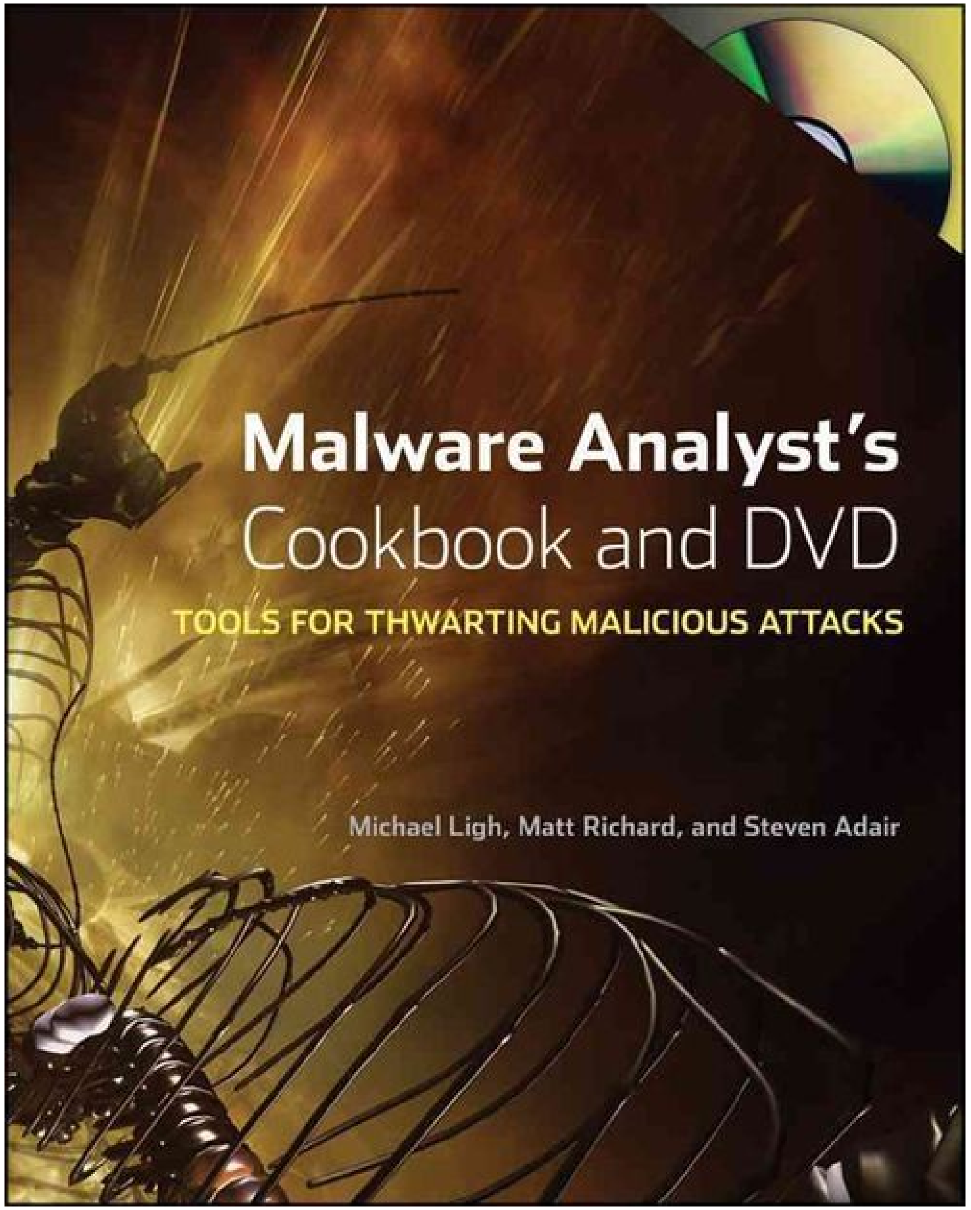
- Boken har ett stort antal verktyg på en DVD (vilken kan laddas ner) som är mycket intressanta!
- DVDn ligger på: [server]\malware\malwarecookbook.com
- **Password “infected”**
- DVDn på internet

http://www.malwarecookbook.com/

Full pott på Amazon, läs recensionerna för att veta mer.

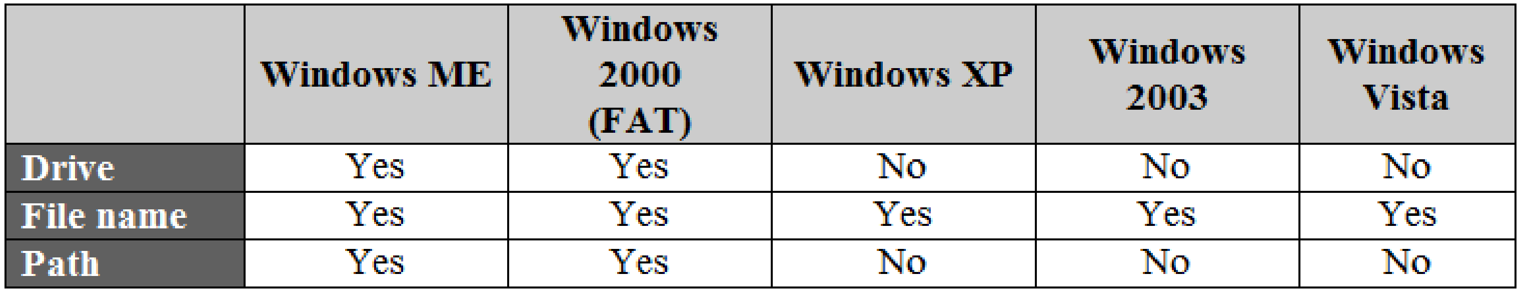
Helt enkelt bästa boken i ämnet!

PE analys exempel med python i slutet av presentationen!


Malware Analyst's
Cookbook and DVD
TOOLS FOR THWARTING MALICIOUS ATTACKS
Michael Ligh, Matt Richard, and Steven Adair

Thumbs.db filer i Windows före Vista

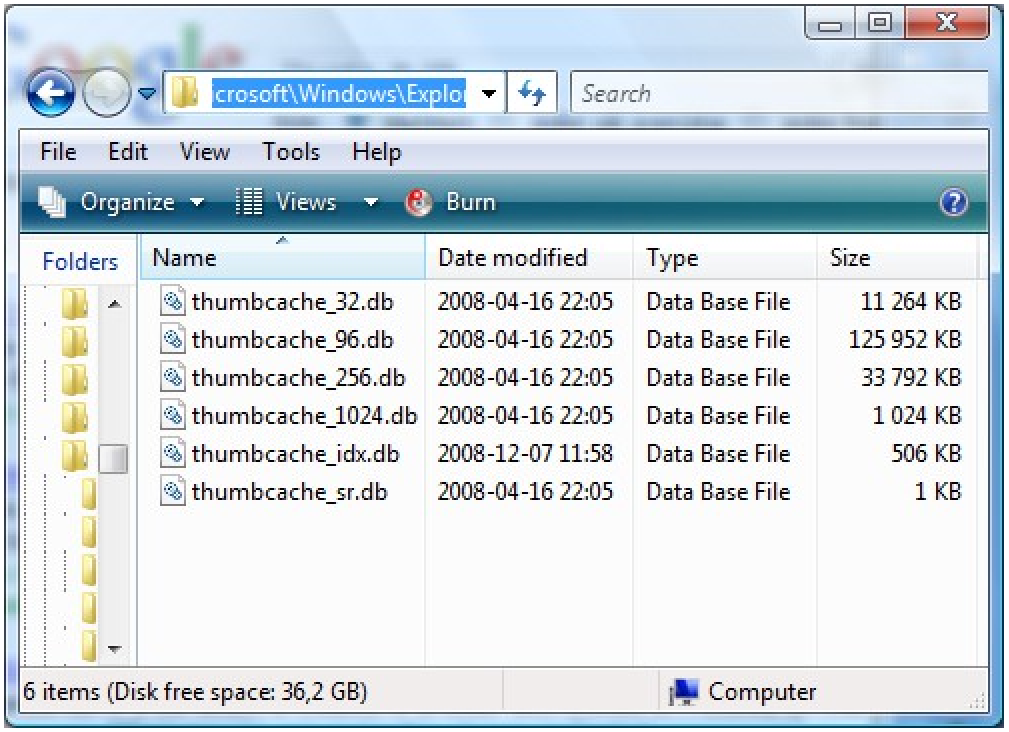
- Spar en liten jpeg miniatyr bild i en thumbs.db fil för varje mapp där JPEG, BMP, GIF bilder existerar
- Se tabell nedan för vad som sparas utöver miniatyren



	Windows ME	Windows 2000 (FAT)	Windows XP	Windows 2003	Windows Vista
Drive	Yes	Yes	No	No	No
File name	Yes	Yes	Yes	Yes	Yes
Path	Yes	Yes	No	No	No

- Skapas som en OS system fil – därför vanligen dold
- När en bild visas skapas ett entry i DB - när stora bilden raderas, raderas ej innehållet i DB!
- Krypterade bildfilers miniatyr bild kan även lagras!
- Läs mer i AccesDatas wp.Thumbs_DB_Files.en_us.pdf
  - Verkligt case och övning i att hasha thumbnails

Thumbnail cache in Vista/7



Name	Date modified	Type	Size
thumbcache_32.db	2008-04-16 22:05	Data Base File	11 264 KB
thumbcache_96.db	2008-04-16 22:05	Data Base File	125 952 KB
thumbcache_256.db	2008-04-16 22:05	Data Base File	33 792 KB
thumbcache_1024.db	2008-04-16 22:05	Data Base File	1 024 KB
thumbcache_idx.db	2008-12-07 11:58	Data Base File	506 KB
thumbcache_sr.db	2008-04-16 22:05	Data Base File	1 KB

6 items (Disk free space: 36,2 GB)

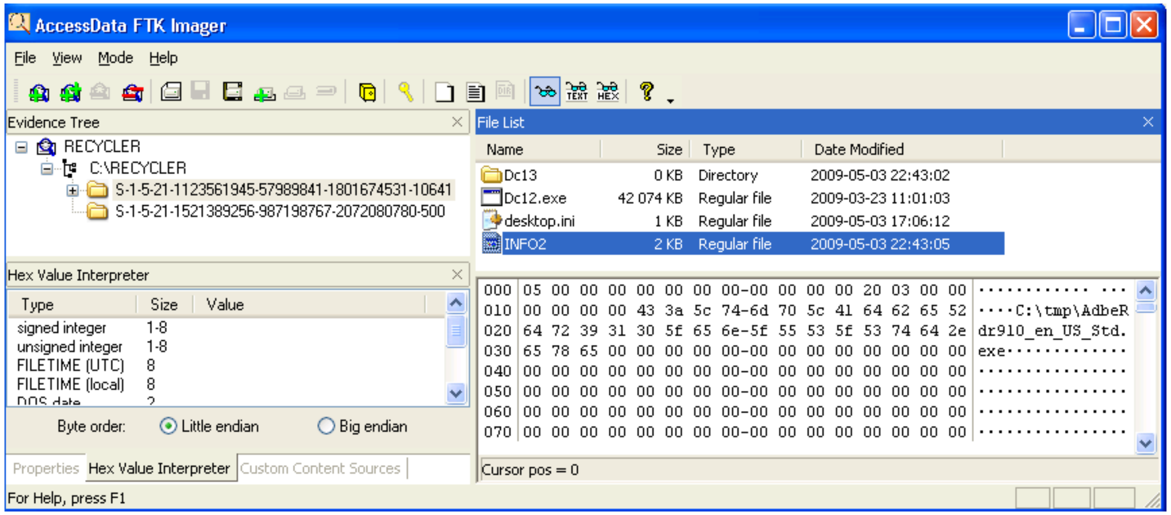
There are several thumbs DB viewers available

- The thumbnail cache that is used in Windows XP/2003, named THUMBS.DB has been replaced with a centralized thumbs database named (either) “thumbcache_32 db”, “thumbcache_96.db”, “thumbcache_256.db” or “thumbcache_1024.db”.
- These centralized caches now hold all thumbnails on the system, depending on their size
- These caches are located in the directory of:

“C:\Users\<user name>\AppData\Local\Microsoft\Windows\Explorer”

Recycle Bin/RECYCLER

- Recycle Bin
  - Om en fil/mapp raderas (flyttas) till Recycle Bin (papperskorgen) så utförs följande
    - Filen/mappen döps om till DcX.ext/DcX och läggs i respektive användares Recycle Bin (SID mapp)
    - Behåller posten i MFT som använd
- INFO2 filen
  - Innehåller metadata om filer/mappar som raderats av användaren
  - På bestämda offsets kan information om varje raderad fil/mapp data (record) utläsas
- ODESSA
  - White paper – Recycler_Bin_ Record_ Reconstruction.pdf
  - Rifiuti – A Recycle Bin Analysis Tool



Evidence Tree

```
RECYCLER
  C:\RECYCLER
    S-1-5-21-1123561945-57989841-1801674531-10641
    S-1-5-21-1521389256-987198767-2072080780-500
```

File List

Name	Size	Type	Date Modified
Dc13	0 KB	Directory	2009-05-03 22:43:02
Dc12.exe	42 074 KB	Regular file	2009-03-23 11:01:03
desktop.ini	1 KB	Regular file	2009-05-03 17:06:12
INFO2	2 KB	Regular file	2009-05-03 22:43:05

Hex Value Interpreter

Type	Size	Value
signed integer	1-8	
unsigned integer	1-8	
FILETIME (UTC)	8	
FILETIME (local)	8	
DOS date	2	

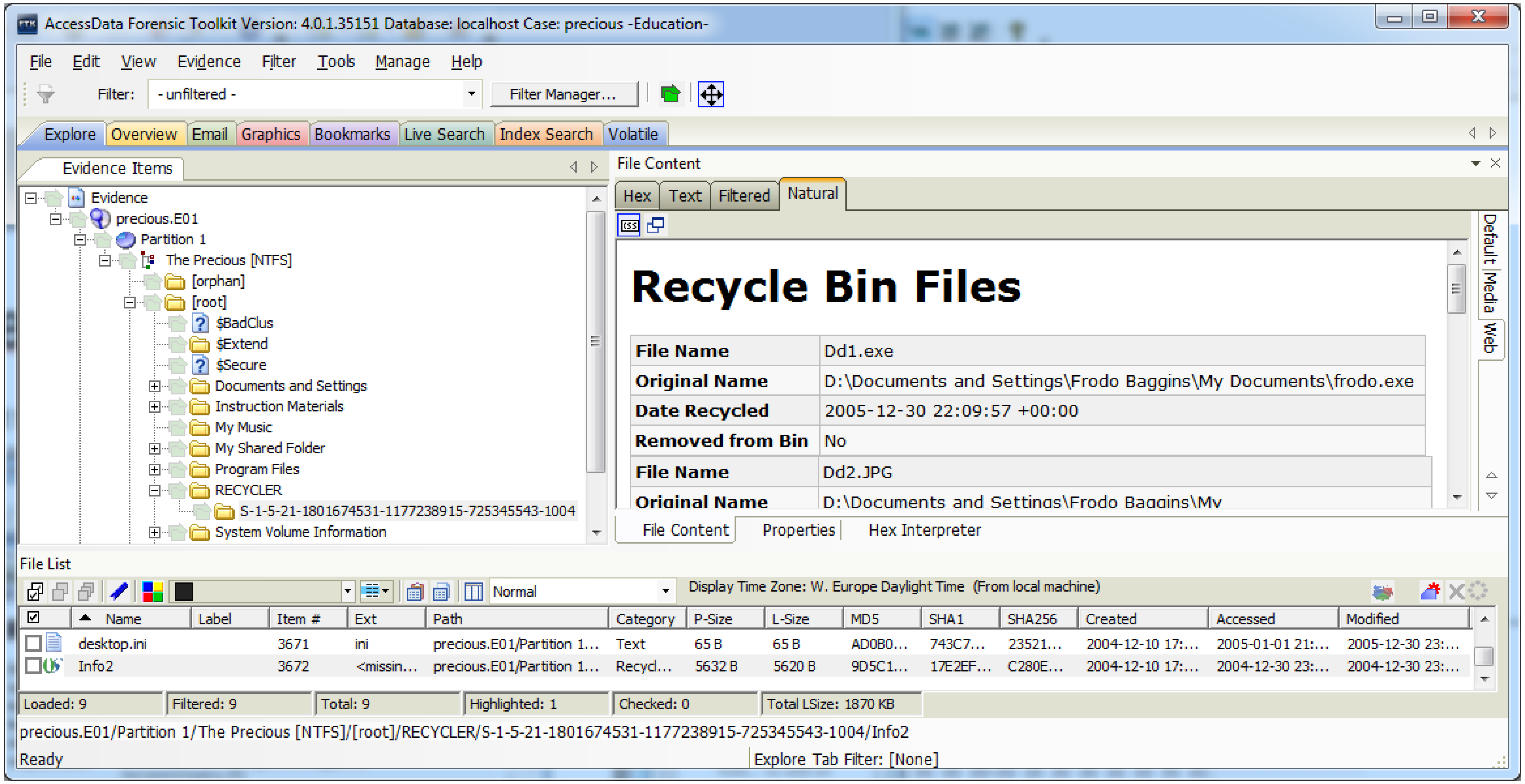
Byte order: [✓ Little endian] Big endian

```
000 05 00 00 00 00 00 00 00-00 00 00 00 20 03 00 00 ············ ···
010 00 00 00 00 43 3a 5c 74-6d 70 5c 41 64 62 65 52 ····C:\tmp\AdbeR
020 64 72 39 31 30 5f 65 6e-5f 55 53 5f 53 74 64 2e dr910_en_US_Std.
030 65 78 65 00 00 00 00 00-00 00 00 00 00 00 00 00 exe·············
040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ················
```

Cursor pos = 0

The INFO2 file

- FTK 4 automatically parses the INFO2 file
- Deleted file path, file index number, file drive location, file date and time, file physical size etc.


AccessData Forensic Toolkit Version: 4.0.1.35151 Database: localhost Case: precious -Education-

Filter: - unfiltered -

Evidence Items

- Evidence
  - precious.E01
    - Partition 1
      - The Precious [NTFS]
        - [orphan]
        - [root]
          - $BadClus
          - $Extend
          - $Secure
          - Documents and Settings
          - Instruction Materials
          - My Music
          - My Shared Folder
          - Program Files
          - RECYCLER
            - S-1-5-21-1801674531-1177238915-725345543-1004
          - System Volume Information

File Content

Recycle Bin Files

File Name	Dd1.exe
Original Name	D:\Documents and Settings\Frodo Baggins\My Documents\frodo.exe
Date Recycled	2005-12-30 22:09:57 +00:00
Removed from Bin	No
File Name	Dd2.JPG
Original Name	D:\Documents and Settings\Frodo Baggins\My

File List

Display Time Zone: W. Europe Daylight Time (From local machine)

☑	Name	Label	Item #	Ext	Path	Category	P-Size	L-Size	MD5	SHA1	SHA256	Created	Accessed	Modified
☐	desktop.ini		3671	ini	precious.E01/Partition 1...	Text	65 B	65 B	AD0B0...	743C7...	23521...	2004-12-10 17:...	2005-01-01 21:...	2005-12-30 23:...
☐	Info2		3672	<missin...	precious.E01/Partition 1...	Recycl...	5632 B	5620 B	9D5C1...	17E2EF...	C280E...	2004-12-10 17:...	2004-12-30 23:...	2004-12-30 23:...

Loaded: 9 | Filtered: 9 | Total: 9 | Highlighted: 1 | Checked: 0 | Total LSize: 1870 KB

precious.E01/Partition 1/The Precious [NTFS]/[root]/RECYCLER/S-1-5-21-1801674531-1177238915-725345543-1004/Info2

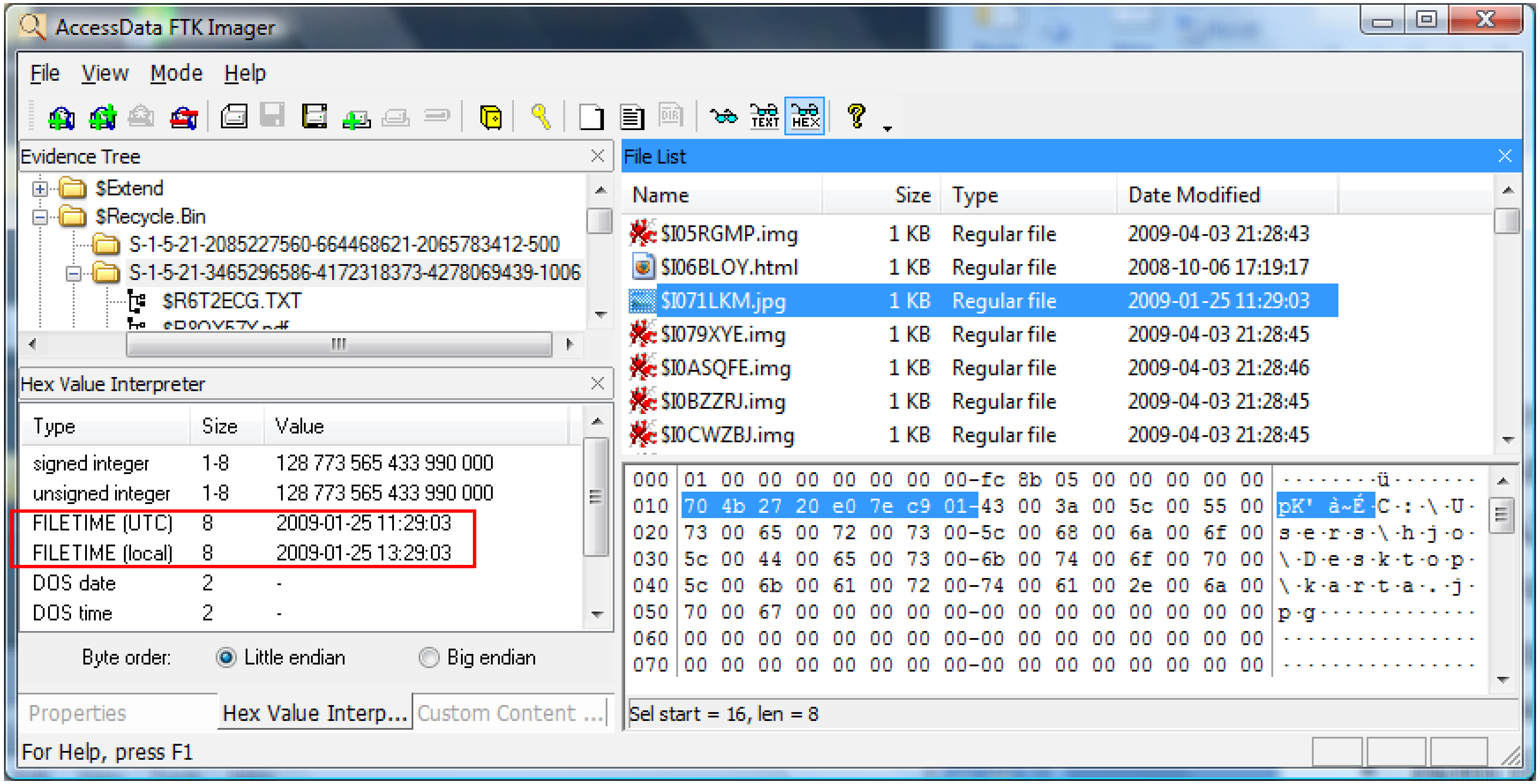
Ready | Explore Tab Filter: [None]

Recycle Bin in Vista/7

- The contents of the recycle bin has changed in Windows Vista and the name of the folder itself has changed to ”$Recycle.bin”
- The INFO2 file that is present in Windows 2000/XP/2003 has been removed
- In Windows Vista/7, two files are created when a file is deleted into the recycle bin
- Both file have the **same** random looking name, but the names are proceeded with a “$R” or “$I”
  - The file or folder with the “$R” at the beginning of the name is actually the data of the deleted file or folder
  - The file with the “$I” at the beginning of the name contains the path of where the file originally resided, as well as the date and time it was deleted

Recycle Bin in Vista/7

- In addition, it is important to note that the users Recycle Bin is created the first time the user logs into their account, not the first time a file/folder is deleted as in Windows 2000/XP/2003



Evidence Tree:

```
$Extend
$Recycle.Bin
  S-1-5-21-2085227560-664468621-2065783412-500
  S-1-5-21-3465296586-4172318373-4278069439-1006
    $R6T2ECG.TXT
```

Name	Size	Type	Date Modified
$I05RGMP.img	1 KB	Regular file	2009-04-03 21:28:43
$I06BLOY.html	1 KB	Regular file	2008-10-06 17:19:17
$I071LKM.jpg	1 KB	Regular file	2009-01-25 11:29:03
$I079XYE.img	1 KB	Regular file	2009-04-03 21:28:45
$I0ASQFE.img	1 KB	Regular file	2009-04-03 21:28:46
$I0BZZRJ.img	1 KB	Regular file	2009-04-03 21:28:45
$I0CWZBJ.img	1 KB	Regular file	2009-04-03 21:28:45

Type	Size	Value
signed integer	1-8	128 773 565 433 990 000
unsigned integer	1-8	128 773 565 433 990 000
FILETIME (UTC)	8	2009-01-25 11:29:03
FILETIME (local)	8	2009-01-25 13:29:03
DOS date	2	-
DOS time	2	-

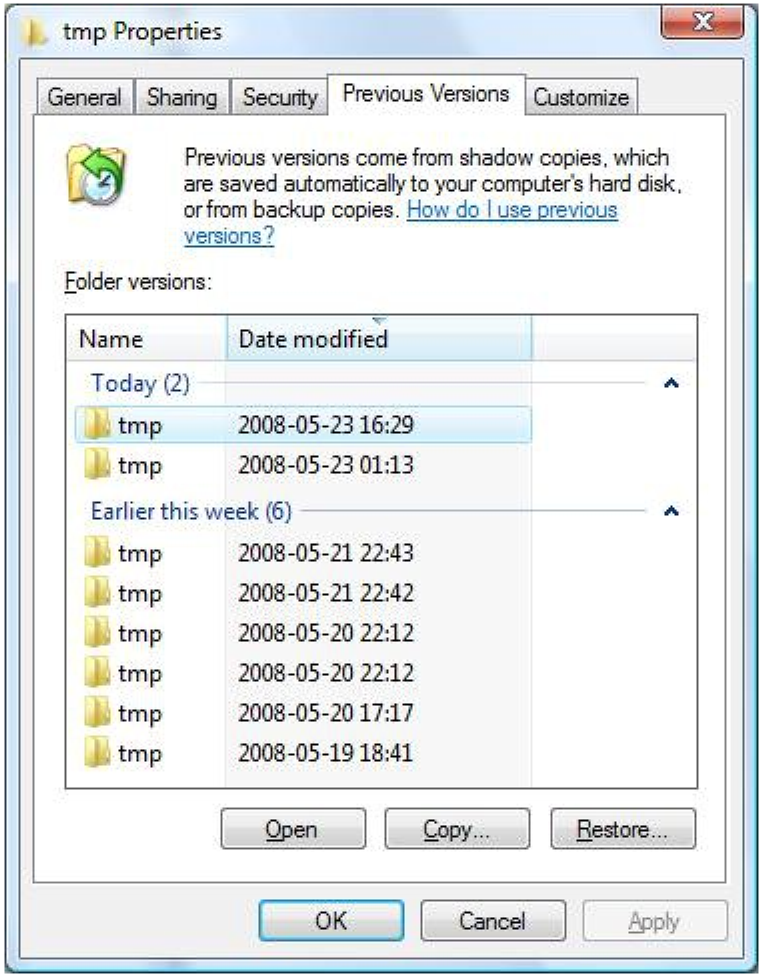
Byte order: [✓ Little endian] Big endian

```
000 01 00 00 00 00 00 00 00-fc 8b 05 00 00 00 00 00  ········ü·······
010 70 4b 27 20 e0 7e c9 01-43 00 3a 00 5c 00 55 00  pK' à~É·C·:·\·U·
020 73 00 65 00 72 00 73 00-5c 00 68 00 6a 00 6f 00  s·e·r·s·\·h·j·o·
030 5c 00 44 00 65 00 73 00-6b 00 74 00 6f 00 70 00  \·D·e·s·k·t·o·p·
040 5c 00 6b 00 61 00 72 00-74 00 61 00 2e 00 6a 00  \·k·a·r·t·a·.·j·
050 70 00 67 00 00 00 00 00-00 00 00 00 00 00 00 00  p·g·············
060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
```

Sel start = 16, len = 8

Volume Shadow Service / Previous Version

- Recycle bin on steroids!
- Shadow copy
  - Business and Ultimate
  - Automatically creates restore points in what changed
  - Only save incremental info
- Saves
  - Deleted and to big data
  - Overwritten data
  - Corrupted data
  - Shift-deleted data



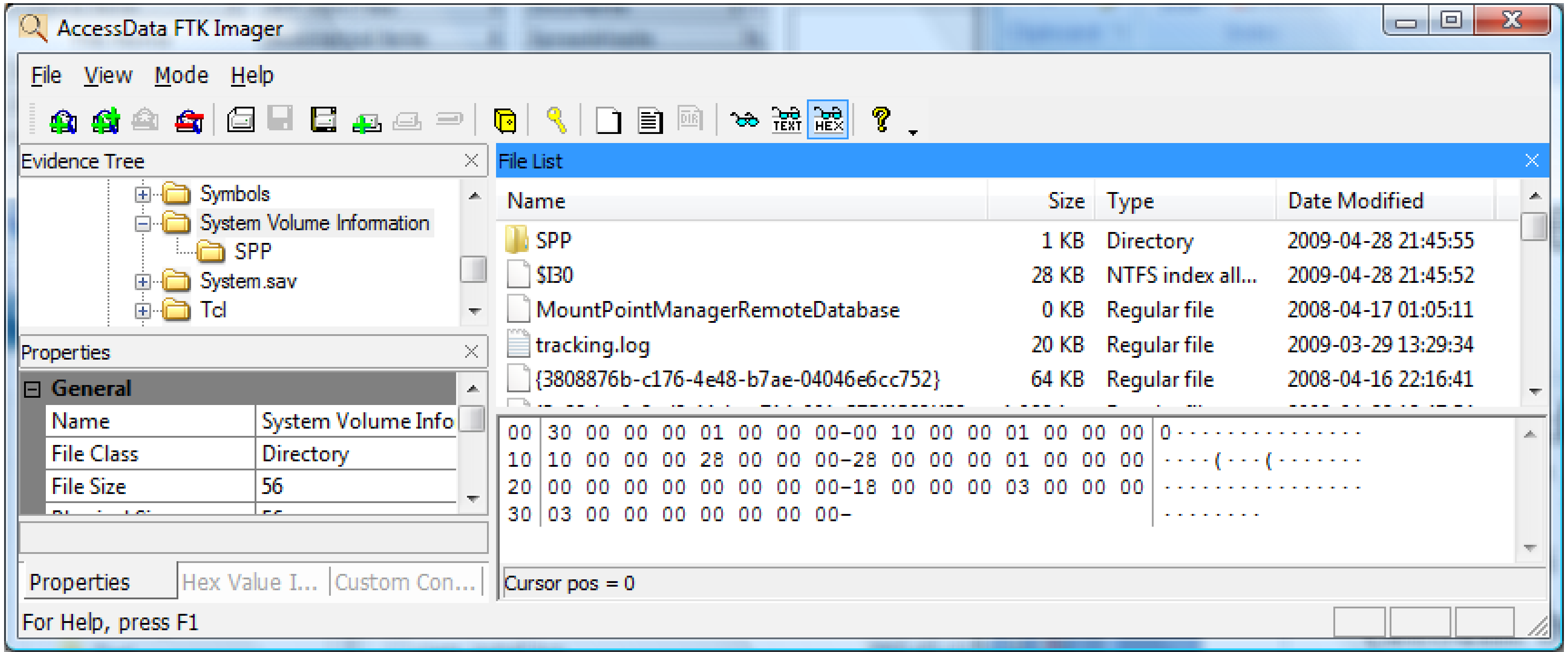
Previous versions come from shadow copies, which are saved automatically to your computer's hard disk, or from backup copies. How do I use previous versions?

Folder versions:

Name	Date modified
Today (2)	
tmp	2008-05-23 16:29
tmp	2008-05-23 01:13
Earlier this week (6)	
tmp	2008-05-21 22:43
tmp	2008-05-21 22:42
tmp	2008-05-20 22:12
tmp	2008-05-20 22:12
tmp	2008-05-20 17:17
tmp	2008-05-19 18:41

Volume Shadow Service / Previous Version

- The block level changes that are saved by the "previous version" feature are stored in the System Volume Information folder as part of a restore point.
- This data is not encrypted (absent bitlocker) and can be easily searched. In the root of the "System Volume Information" folder, several files can be seen with GUIDs as the filename.



General	
Name	System Volume Info
File Class	Directory
File Size	56

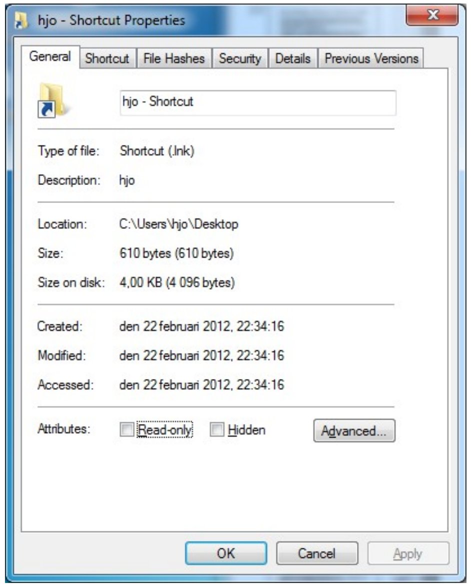
Name	Size	Type	Date Modified
SPP	1 KB	Directory	2009-04-28 21:45:55
$I30	28 KB	NTFS index all...	2009-04-28 21:45:52
MountPointManagerRemoteDatabase	0 KB	Regular file	2008-04-17 01:05:11
tracking.log	20 KB	Regular file	2009-03-29 13:29:34
{3808876b-c176-4e48-b7ae-04046e6cc752}	64 KB	Regular file	2008-04-16 22:16:41

```
00  30 00 00 00 01 00 00 00-00 10 00 00 01 00 00 00   0...............
10  10 00 00 00 28 00 00 00-28 00 00 00 01 00 00 00   ....(...(.......
20  00 00 00 00 00 00 00 00-18 00 00 00 03 00 00 00   ................
30  03 00 00 00 00 00 00 00-                           ........
```

Cursor pos = 0

What are Link Files?

- Link files are shortcuts to for example recently opened files or other resources and contains
  - Target path information
  - Target date/time information
  - User-defined link comments
  - Target machine addressing



hjo - Shortcut

Type of file: Shortcut (.lnk)
Description: hjo

Location: C:\Users\hjo\Desktop
Size: 610 bytes (610 bytes)
Size on disk: 4,00 KB (4 096 bytes)

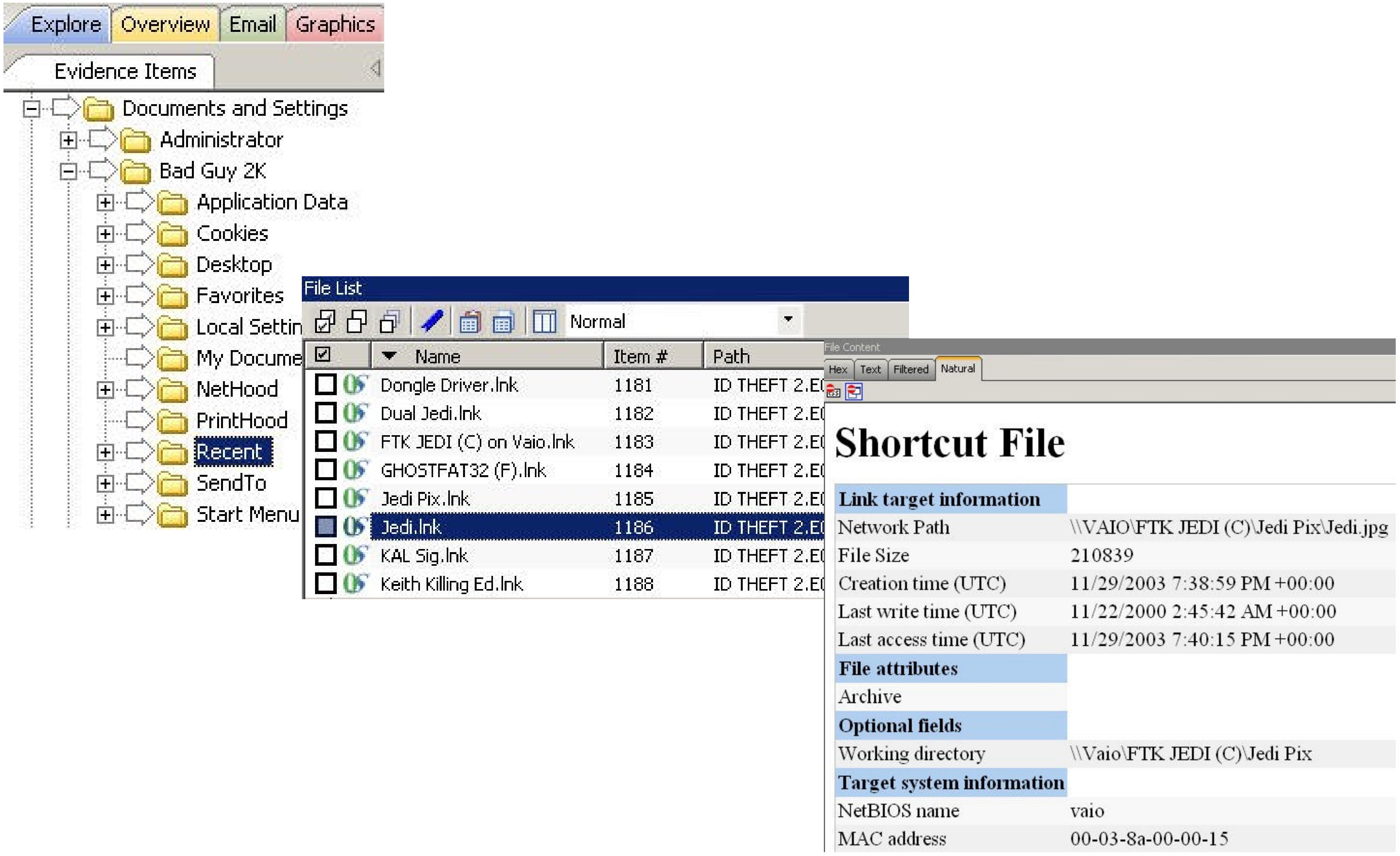
Created: den 22 februari 2012, 22:34:16
Modified: den 22 februari 2012, 22:34:16
Accessed: den 22 februari 2012, 22:34:16

Attributes: ☐ Read-only ☐ Hidden

Either a .lnk or .url extension

- Standard shortcuts
  - Link to local or network programs, files – usually executables or documents, folders, printers or computers
- Uniform Resource Locator (URL) shortcuts
  - Link to entities that can be referenced by a valid URL – usually a Web page

Recent Files – Location

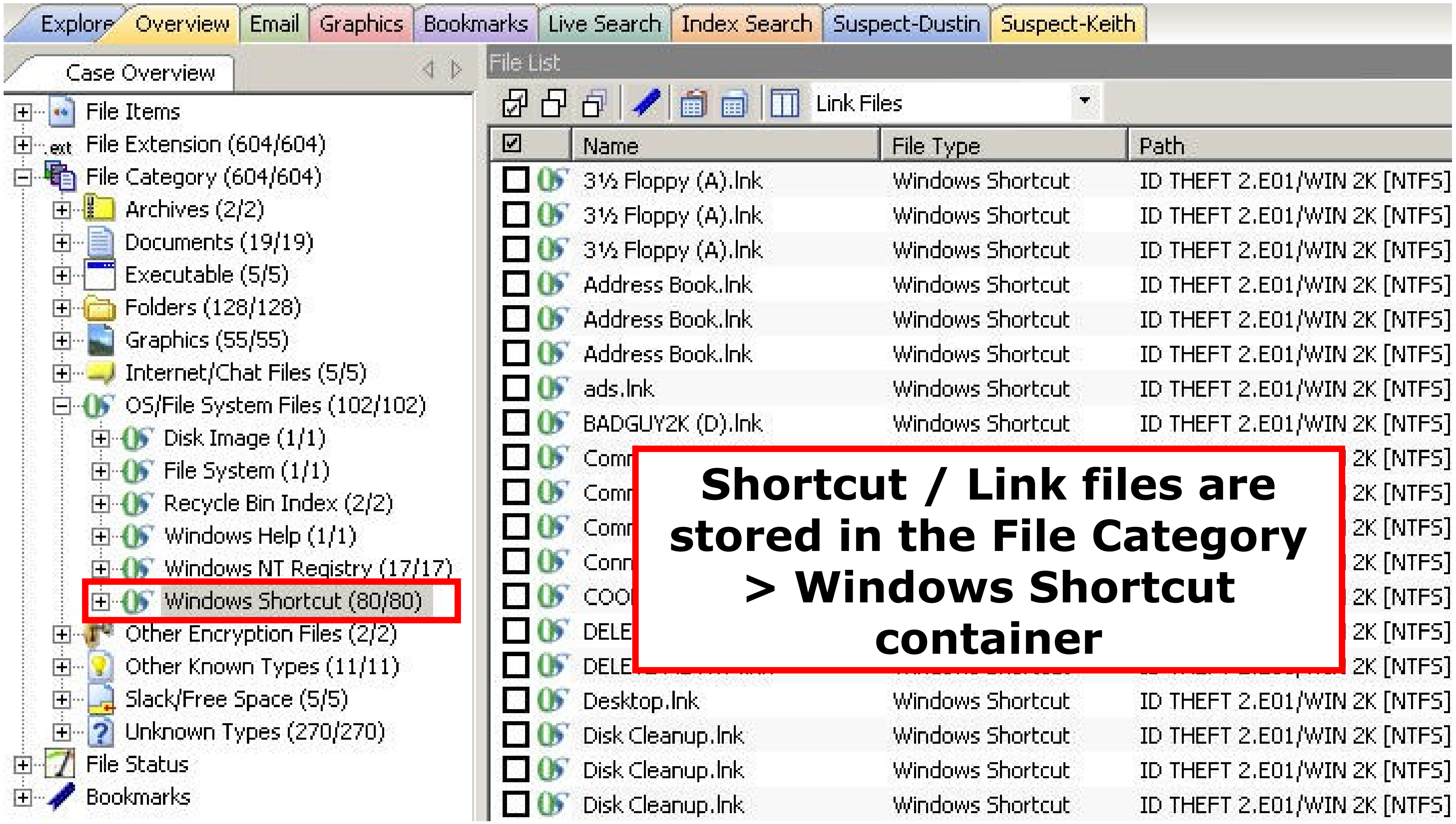


Name	Item #	Path
Dongle Driver.lnk	1181	ID THEFT 2.E
Dual Jedi.lnk	1182	ID THEFT 2.E
FTK JEDI (C) on Vaio.lnk	1183	ID THEFT 2.E
GHOSTFAT32 (F).lnk	1184	ID THEFT 2.E
Jedi Pix.lnk	1185	ID THEFT 2.E
Jedi.lnk	1186	ID THEFT 2.E
KAL Sig.lnk	1187	ID THEFT 2.E
Keith Killing Ed.lnk	1188	ID THEFT 2.E

Shortcut File

Link target information	
Network Path	\\VAIO\FTK JEDI (C)\Jedi Pix\Jedi.jpg
File Size	210839
Creation time (UTC)	11/29/2003 7:38:59 PM +00:00
Last write time (UTC)	11/22/2000 2:45:42 AM +00:00
Last access time (UTC)	11/29/2003 7:40:15 PM +00:00
File attributes	
Archive	
Optional fields	
Working directory	\\Vaio\FTK JEDI (C)\Jedi Pix
Target system information	
NetBIOS name	vaio
MAC address	00-03-8a-00-00-15

FTK Classification



Explore | Overview | Email | Graphics | Bookmarks | Live Search | Index Search | Suspect-Dustin | Suspect-Keith

Case Overview

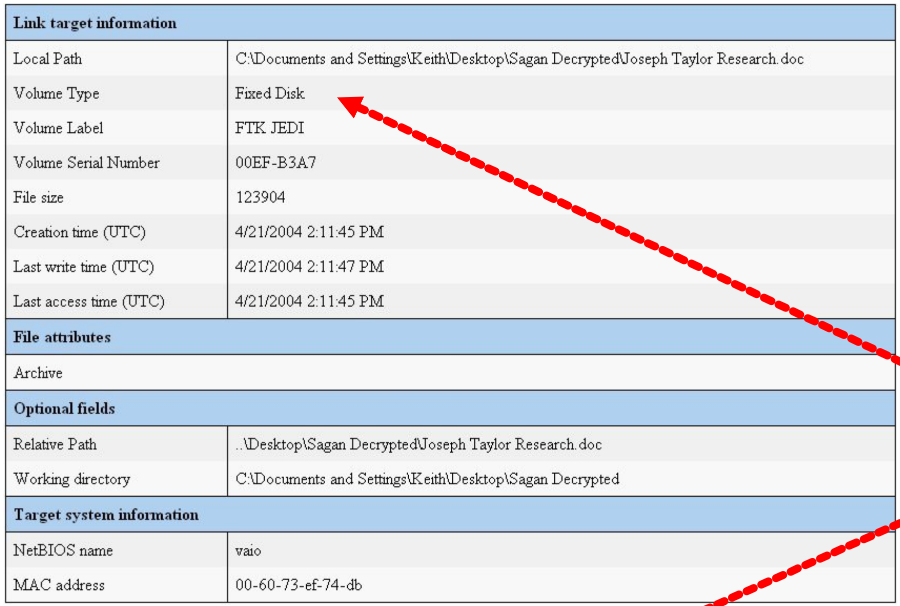
- File Items
- .ext File Extension (604/604)
- File Category (604/604)
  - Archives (2/2)
  - Documents (19/19)
  - Executable (5/5)
  - Folders (128/128)
  - Graphics (55/55)
  - Internet/Chat Files (5/5)
  - OS/File System Files (102/102)
    - Disk Image (1/1)
    - File System (1/1)
    - Recycle Bin Index (2/2)
    - Windows Help (1/1)
    - Windows NT Registry (17/17)
    - Windows Shortcut (80/80)
  - Other Encryption Files (2/2)
  - Other Known Types (11/11)
  - Slack/Free Space (5/5)
  - Unknown Types (270/270)
- File Status
- Bookmarks

File List — Link Files

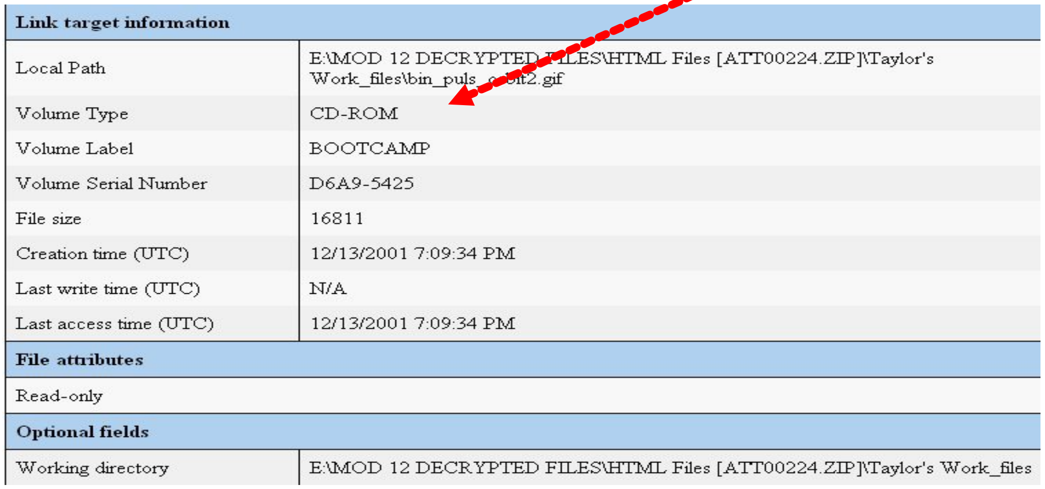
Name	File Type	Path
3½ Floppy (A).lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
3½ Floppy (A).lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
3½ Floppy (A).lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Address Book.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Address Book.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Address Book.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
ads.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
BADGUY2K (D).lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Com		2K [NTFS]
Com		2K [NTFS]
Com		2K [NTFS]
Con		2K [NTFS]
COO		2K [NTFS]
DELE		2K [NTFS]
DELE		2K [NTFS]
Desktop.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Disk Cleanup.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Disk Cleanup.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]
Disk Cleanup.lnk	Windows Shortcut	ID THEFT 2.E01/WIN 2K [NTFS]

Shortcut / Link files are stored in the File Category > Windows Shortcut container

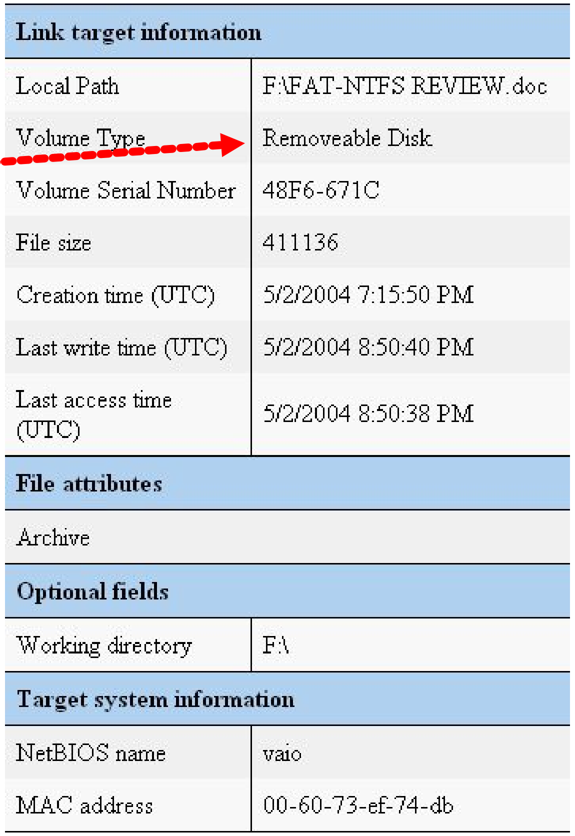
Link Information



Link target information	
Local Path	C:\Documents and Settings\Keith\Desktop\Sagan Decrypted\Joseph Taylor Research.doc
Volume Type	Fixed Disk
Volume Label	FTK JEDI
Volume Serial Number	00EF-B3A7
File size	123904
Creation time (UTC)	4/21/2004 2:11:45 PM
Last write time (UTC)	4/21/2004 2:11:47 PM
Last access time (UTC)	4/21/2004 2:11:45 PM
File attributes	
Archive	
Optional fields	
Relative Path	..\Desktop\Sagan Decrypted\Joseph Taylor Research.doc
Working directory	C:\Documents and Settings\Keith\Desktop\Sagan Decrypted
Target system information	
NetBIOS name	vaio
MAC address	00-60-73-ef-74-db



Link target information	
Local Path	E:\MOD 12 DECRYPTED FILES\HTML Files [ATT00224.ZIP]\Taylor's Work_files\bin_puls_orbit2.gif
Volume Type	CD-ROM
Volume Label	BOOTCAMP
Volume Serial Number	D6A9-5425
File size	16811
Creation time (UTC)	12/13/2001 7:09:34 PM
Last write time (UTC)	N/A
Last access time (UTC)	12/13/2001 7:09:34 PM
File attributes	
Read-only	
Optional fields	
Working directory	E:\MOD 12 DECRYPTED FILES\HTML Files [ATT00224.ZIP]\Taylor's Work_files

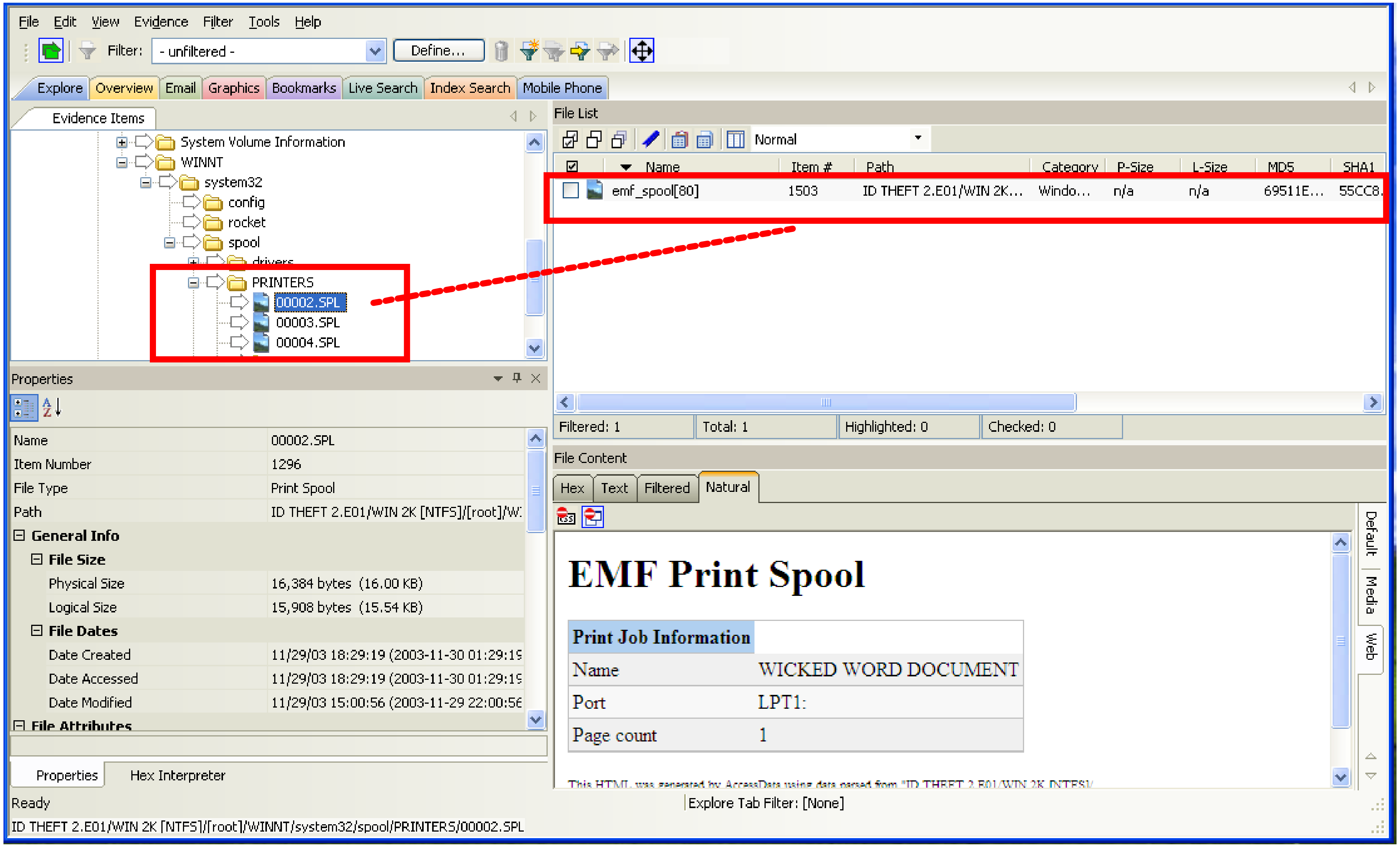


Link target information	
Local Path	F:\FAT-NTFS REVIEW.doc
Volume Type	Removeable Disk
Volume Serial Number	48F6-671C
File size	411136
Creation time (UTC)	5/2/2004 7:15:50 PM
Last write time (UTC)	5/2/2004 8:50:40 PM
Last access time (UTC)	5/2/2004 8:50:38 PM
File attributes	
Archive	
Optional fields	
Working directory	F:\
Target system information	
NetBIOS name	vaio
MAC address	00-60-73-ef-74-db

What are Spool Files?

- Local print provider actions
  - Writes print job contents to a SPL file
  - Creates a separate graphics file (EMF) for each page
  - Creates an admin shadow file (SHD) that contains
    - User name/Machine name
    - Document name and data type
- Default Windows print spool location
  - %SystemRoot%\System32\Spool\PRINTERS
- Spooled document pages are stored as .EMF or graphic files – These are in the Graphics container in FTK
- SPL and SHD files are deleted after the job prints – FTK can data carve the EMF graphics
- Print jobs might also be spooled through the Windows swap / page file

Spool and EMF Files


File Edit View Evidence Filter Tools Help
Define...
Explore Overview Email Graphics Bookmarks Live Search Index Search Mobile Phone
Evidence Items
File List
Properties
Hex Text Filtered Natural
Default Media Web
Properties Hex Interpreter

Filter: - unfiltered -

System Volume Information
WINNT
system32
config
rocket
spool
drivers
PRINTERS
00002.SPL
00003.SPL
00004.SPL

Normal

	Name	Item #	Path	Category	P-Size	L-Size	MD5	SHA1
☐	emf_spool[80]	1503	ID THEFT 2.E01/WIN 2K...	Windo...	n/a	n/a	69511E...	55CC8

Filtered: 1 Total: 1 Highlighted: 0 Checked: 0

Name	00002.SPL
Item Number	1296
File Type	Print Spool
Path	ID THEFT 2.E01/WIN 2K [NTFS]/[root]/W
General Info	
File Size	
Physical Size	16,384 bytes (16.00 KB)
Logical Size	15,908 bytes (15.54 KB)
File Dates	
Date Created	11/29/03 18:29:19 (2003-11-30 01:29:19
Date Accessed	11/29/03 18:29:19 (2003-11-30 01:29:19
Date Modified	11/29/03 15:00:56 (2003-11-29 22:00:56
File Attributes	

EMF Print Spool

Print Job Information	
Name	WICKED WORD DOCUMENT
Port	LPT1:
Page count	1

This HTML was generated by AccessData using data parsed from "ID THEFT 2.E01/WIN 2K [NTFS]/

Ready

Explore Tab Filter: [None]

ID THEFT 2.E01/WIN 2K [NTFS]/[root]/WINNT/system32/spool/PRINTERS/00002.SPL

Detecting malicious PE files 1

- Executables on Windows must conform to the PE/COFF (Portable Executable/Common Object File Format) specification. This includes, but is not limited to, console and GUI applications (.exe), Dynamic Link Libraries (.dll), kernel drivers (.sys), and ActiveX controls (.ocx)
- For a good introduction, see Matt Pietrek's two-part series: Peering Inside the PE and An In-Depth Look into the Win32 Portable Executable File Format
  - http://msdn.microsoft.com/en-us/magazine/ms809762.aspx
  - http://msdn.microsoft.com/en-us/magazine/cc301805.aspx
- The malware book authors show you several ways to detect suspicious files based on values in the PE header. Thus, independent of any antivirus scanners, you can use heuristics to quickly determine which files exhibit suspicious attributes. The code for this recipe uses Ero Carrera's pefile, which is a Python module for parsing PE headers.
  - http://code.google.com/p/pefile/
- You can find the book script, named pescanner.py on the book's DVD in **Malwarecookbook\3\8**. It currently detects the following criterias:

Detecting malicious PE files 2

- Files with TLS entries
  - TLS entries are functions that execute before the program’s main thread, thus before the initial breakpoint set by debuggers. Malware typically uses TLS entries to run code before your debugger gets control. The pescanner.py script prints the addresses of all TLS callback functions.
- Files with resource directories
  - Resource directories can contain arbitrary data types such as icons, cursors, and configurations. If you’re scanning an entire system32 directory, then you will likely find many false positives because resource directories are legitimate. However, if you’re scanning a folder full of malware, the presence of a resource directory likely indicates that the file drops another executable at run-time. The pescanner.py script extracts all resources from the PE file and runs them through the file type identification process described earlier

Detecting malicious PE files 3

- Suspicious IAT entries
  - Imported functions can indicate how a program behaves at run-time. You can create a list of API functions that are suspicious and then produce an alert whenever you find a malware sample that imports a function from your list. The pescanner.py script has a default list of about 15 APIs, but it’s up to you to add additional ones.
- Suspicious entry point sections
  - An entry point section is the name of the PE section that contains the AddressOfEntryPoint. The AddressOfEntryPoint value for legitimate, or non-packed, files typically resides in a section named .code or .text for user mode programs, and PAGE or INIT for kernel drivers. Therefore, you can detect potentially packed files if the entry point resides in a section that is not in your list of known-good sections.

Detecting malicious PE files 4

- Sections with zero-length raw sizes
  - The raw size is the amount of bytes that a section requires in the file on disk (as opposed to bytes required when the section is mapped into memory). The most common reason a raw size would be zero on disk but greater than zero in memory is because packers copy decrypted instructions or data into the section at run-time.
- Sections with extremely low or high entropy
  - Entropy is a value between 0 and 8 that describes the randomness of data. Encrypted or compressed data typically has high entropy, whereas a long string of the same character has low entropy. By calculating entropy, you can get a good idea of which sections in a PE file contain packed or abnormal code.

Detecting malicious PE files 5

- Invalid timestamps
  - The TimeDateStamp field is a 32-bit value (the number of seconds since December 31st, 1969, 4 P.M.) that indicates when the linker or compiler produced the PE file. Malware authors (and packers) obscure this value to hide the true build date. If pescanner.py detects an invalid date, it produces an alert.
- File version information
  - A PE file's version information may contain the name of the person or company who created the file, a description of the file, a version and/or build number, the original file name, and other comments. This type of information is not available in all PE files, but many times malware authors will accidentally leave it in or intentionally forget the values. In both cases, the information yields interesting forensic evidence.

Using pescanner.py

- Using SIFT workstation or apt-get based distro
  - Set VMware appliance to NAT if using SIFT
  - sudo su -
  - apt-get install python-magic
  - apt-get install upx-ucl
  - apt-get install ssdeep
  - apt-get install python-pefile (break pescanner on SIFT?, skip it)
  - Install Yara as usual (./configure, make, make install)
- Windows - Activestate Python etc. Use x86 even if your machine is x64 to avoid non-existent modules
  - pypm install python-magic
    - Or from https://github.com/ahupp/python-magic
  - pefile-1.2.10-102 (see earlier slide for download)
    - python setup.py install
  - yara-python-1.5.win32-py2.x.exe

pescanner.py output 1

```
sansforensics@SIFT-Workstation:~$ python pescanner.py protected.exe
################################################################################
Record 0
################################################################################
Meta-data
================================================================================
File:    protected.exe
Size:    11264 bytes
Type:    PE32 executable for MS Windows (GUI) Intel 80386 32-bit
MD5:     2779ead9dc8c7b62b1738e09240ed103
SHA1:    8b9f29d2bb2f99949a9f4a943b93a87d6d8b06a5
ssdeep:  192:nAdgDAraVWWcN9fNZhz9TYsoIs5/KFle0bGmKUXNOWFLr [REMOVED]
Date:    0x483EBDA6 [Thu May 29 14:28:54 2008 UTC]
EP:      0x405bd6  3/4 [SUSPICIOUS]
CRC:     Claimed: 0xb3c9, Actual: 0xb3c9
Sections
================================================================================
Name       VirtAddr     VirtSize     RawSize      Entropy
--------------------------------------------------------------------------------
.text      0x1000       0x1000       0x200        7.602876    [SUSPICIOUS]
.rdata     0x2000       0x1000       0x200        7.622602    [SUSPICIOUS]
.data      0x3000       0x1000       0x200        7.615053    [SUSPICIOUS]
           0x4000       0x3000       0x2200       7.610693    [SUSPICIOUS]
```

pescanner.py output 2

```
sansforensics@SIFT-Workstation:~$ python pescanner.py upx_sample.exe
################################################################################
Record 0
################################################################################
Meta-data
============================================================
File:    upx_sample.exe
Size:    72704 bytes
Type:    MS-DOS executable PE  for MS Windows (GUI) Intel 80386 32-bit
MD5:     22a9c61c71fa5cef552a94e479dfe41e
SHA1:    14ac258df52d0131c5984b00dc14960ee94e6aad
ssdeep:  1536:JxXOg1j5jBWSNzrpGhDZuiq3AC+wcnG4Pqvtuz+ [REMOVED]
Date:    0x49277573 [Sat Nov 22 02:58:59 2008 UTC]
EP:      0x4292e0 (UPX1) [SUSPICIOUS]
Packers: UPX 2.90 [LZMA] -> Markus Oberhumer, Laszlo Molnar & John Reiser
Sections
============================================================
Name       VirtAddr     VirtSize     RawSize      Entropy
------------------------------------------------------------
UPX0       0x1000       0x17000      0x0          0.000000    [SUSPICIOUS]
UPX1       0x18000      0x12000      0x11600      7.912755    [SUSPICIOUS]
UPX2       0x2a000      0x1000       0x200        2.71365
```