Detecting Intrusions

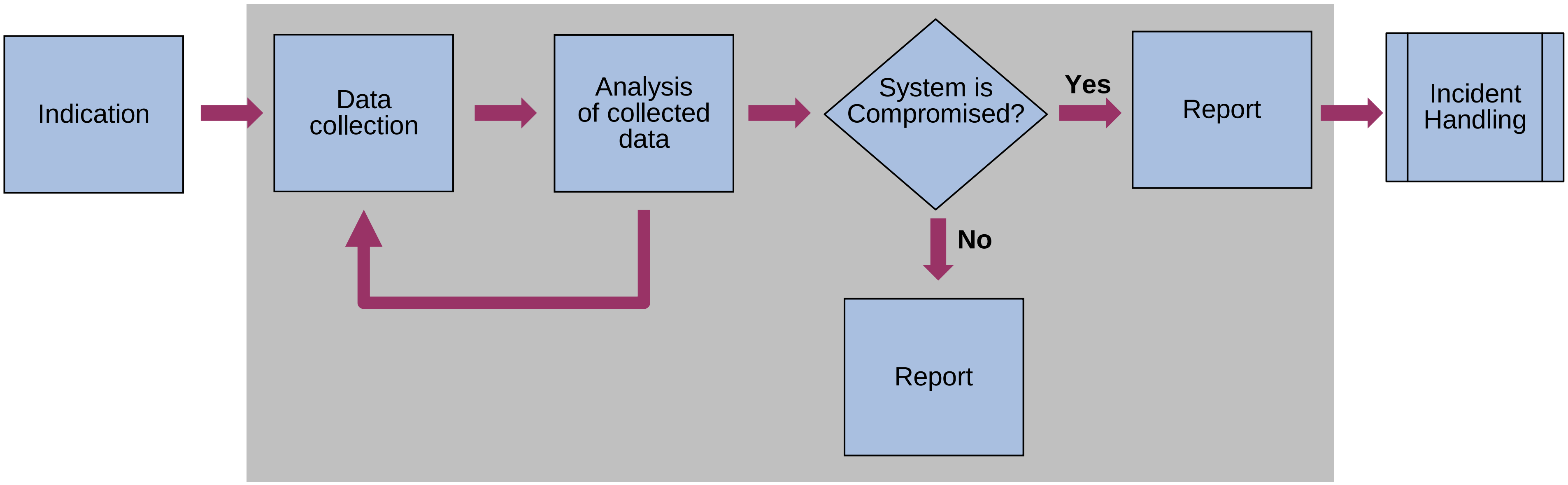
The latest forensics tools and techniques to identify Windows malware infections

Pär Österberg Medina, Sitic

FIRST Conference 2008
Vancouver, June 2008

About the Tutorial

About the Tutorial



The Speaker

Pär Österberg Medina

- CISSP, GCIH
- Experienced with Windows and UNIX, penetration testing.
- Now an incident handler with the Swedish Government CERT, SITIC.

About the Tutorial
Previous presentations

2006

- Sitic – Spring seminar
  http://www.sitic.se/seminarium/sitics-varseminarium/
- SecHeads
- T2'06
  http://www.t2.fi/schedule/2006/#speech8
- Sitic – Seminar about Detecting Intrusions
  http://www.sitic.se/seminarium/seminarium_dec06/

Previous presentations

2007

- Sitic – Seminar about Detecting Intrusions
  http://www.sitic.se/seminarium/seminarium_feb07/
- IP-dagarna
  http://oldweb.iis.se/Internetdagarna/2006/22-forensics/forensics.shtml
- Susec
  http://www.susec.sunet.se/susec/Susecv07/

Previous presentations

FIRST2007

- “Forensic Tools and Techniques to Examine Microsoft Windows”
  - ➔ Andreas Schuster - Deutche Telekom
    http://computer.forensikblog.de/en/

Course outline

- Present methods and techniques an organization can use in order to build a framework which can be used to;
  - Detect a potential computer intrusion or rule it off as a false positive
    - Malware that do not try to hide itself
    - Malware that try to hide itself
  - Detect IT-policy violations

Objective

- The attendees should have a good knowledge of which methods and techniques to use when investigating a suspected computer intrusion
- Memory acquisition and analysis should be a standard part of your incident investigation
- Everybody in this classroom should have come to the conclusion themselves, that an automated method for both collecting and analyzing data is needed when investigating a computer system that is suspected of an intrusion.

Agenda

- Description of the Method
- Data Collection
  - First Responder's Toolkit
  - Order of Volatility
  - Collecting volatile and non volatile data
- Data Analysis
  - Analyzing the data we collected
  - Exercise: Is the system compromised?

What is this course not about

- This is not a course on traditional disk forensics
  - We do not know yet if the system has been compromised which might cause a problem when we have to convince the system owners that a shutdown of the system is necessary
- I will not present a silver bullet solution that will solve all your problems when it comes to live system forensic and incident response
- This course is also not about releasing a the “holy graal” tool

People how have contributed to this course

- Andreas Schuster - Deutche Telekom
  http://computer.forensikblog.de/en/

Big thanks to

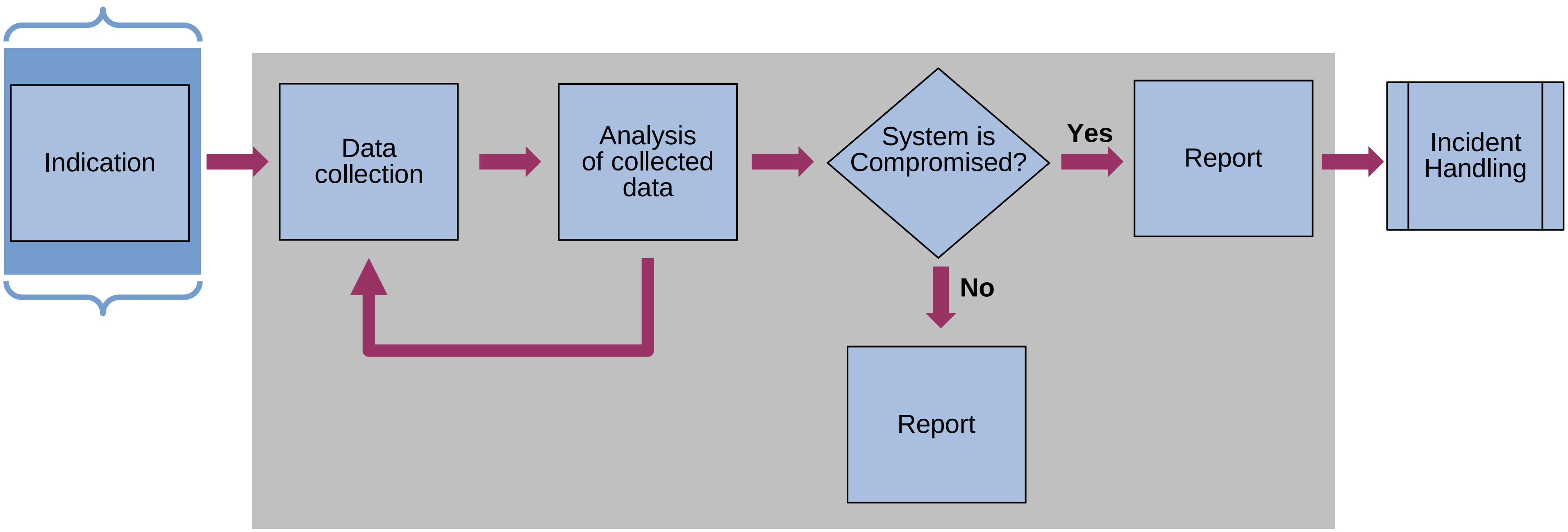
- George M. Garner - GMG Systems, Inc.
  http://www.gmgsystemsinc.com/knttools/

Description of the Method

Why we do the things we do

Description of the Method

Why we do the things we do



Weigh potential damage vs. workload

- Resources
  - ➔ How many hours do we have to spend on investigating a potential intrusion?
    - We do not know if the system has been compromised at this point
- Knowledge
  - ➔ Do we have experienced Incident Handlers on site?
    - Who can perform a forensic investigation of the system?

Automated procedure for collecting and analyzing data (1)

- Script language for automation – Needs to be portable in the data collection part
    - ➔ Windows Batch - preferable before RAM have been collected
    - ➔ Perl, Python or equivalent - after the memory have been collected

Description of the Method
Why we do the things we do

Automated procedure for collecting and analyzing data (2)

- Command Line Interface (CLI)
  - ➔ Touches less on the system that we are investigating
  - ➔ Easier to script

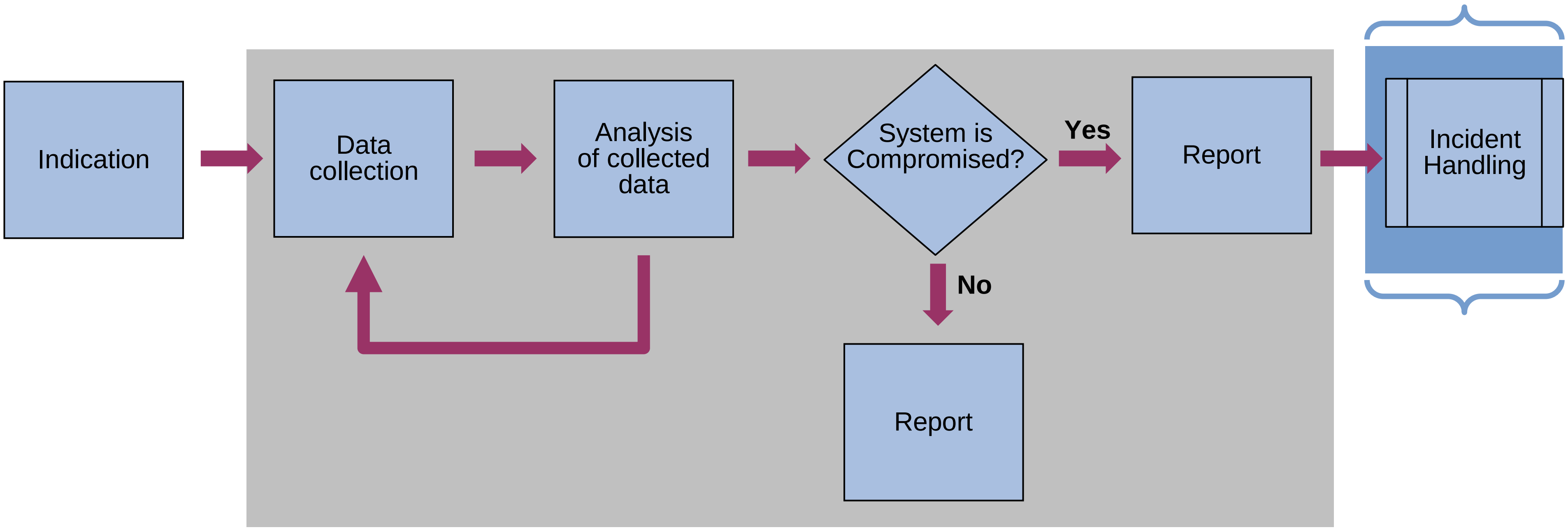
Description of the Method
Why we do the things we do

Automated procedure for collecting and analyzing data (3)

- Publicly available programs
  - ➔ Less resources needed to develop tools
  - ➔ The programs get updated as new versions of Windows get released

Description of the Method

Why we do the things we do



Leave minimal footprint on the system (1)

- Do not write or delete files on the hard drive
- Avoid changing any time attributes of the files
    - ➔ Or at least save them!

Description of the Method

Why we do the things we do

Leave minimal footprint on the system (2)

- Do not make the analysis on the same system that we are investigating
  - ➔ Will change timestamps and write files to the hard drive
  - ➔ The system can be infected and therefore hiding data from us

Description of the Method
Why we do the things we do

Document what is being done to the system

Sitic
Sveriges IT-incidentcentrum

Handläggare:	
System	
Datum:	
Kommentar:	

Tidpunkt	Utförd handling/kommando	Kommentar

Data has precedence over the integrity of the system

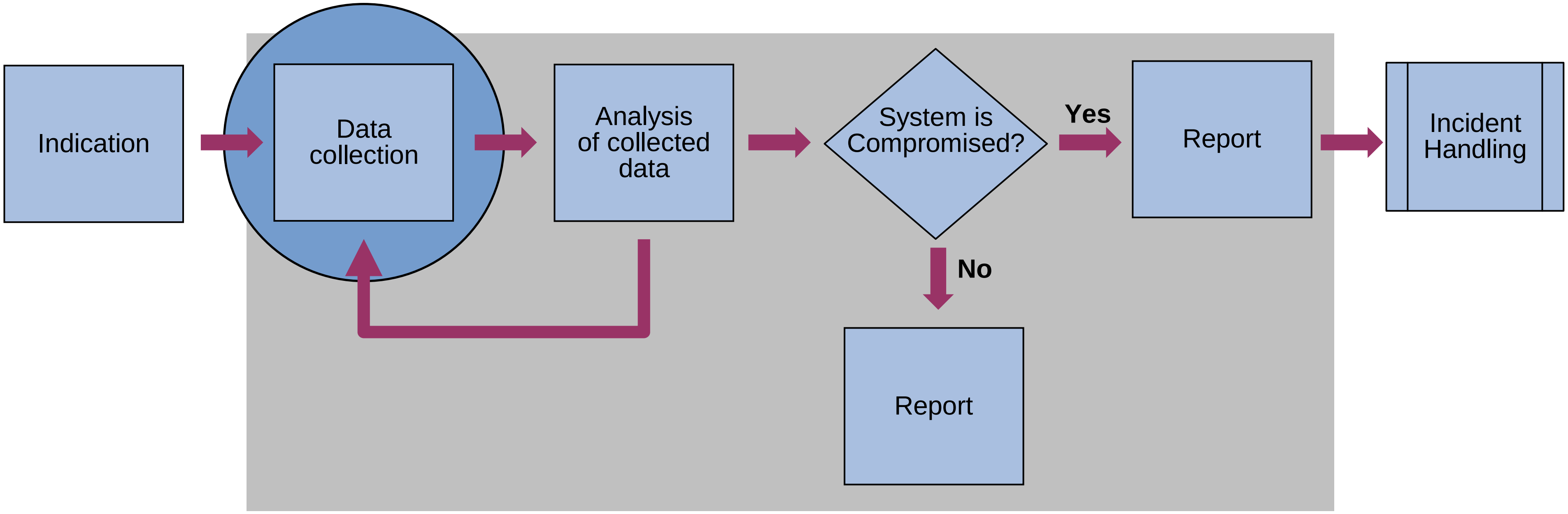
- With no data collected there can be no analysis hence the question if the system has been compromised remains unanswered

Description of the Method
Why we do the things we do

Conclusions

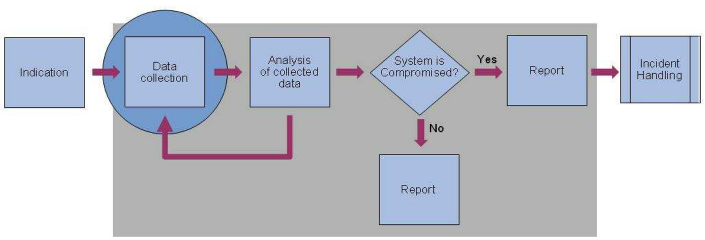
- By using an automated method for collection and analysis, we can;
  - ➔ Reduce the workload for the discovery of an incident
  - ➔ Reduce the knowledge needed by the person that is collecting the data
- Data from an active system is needed if we are to answer the question: Is the system compromised or not?
- Data from an active system can facilitate a full blown computer forensic investigation

Data Collection



Data Collection

First Responder‘s Toolkit



What is the First Responder's Toolkit? (1)

- Write protected media that contains all the program and script needed to acquire the data
  - CDROM
    - Write protected be default
  - USB
    - USB-key write protection switch
    - USB write blocker
    - U3 write protected CDROM emulation

What is the First Responder's Toolkit? (2)

- Trusted binaries with program that we will execute on the system
  - Checked against the right system version, patch level and architecture
  - Add a suffix or prefix (trusted_cmd.exe)
    - Avoid executing the wrong binary by mistake
    - Easier to separate our trusted binaries when we analyze the data
    - Avoid anti-forensic techniques
      Mailbot.AZ (aka Rustock.A) - (BlackLight, Rootkitrevealer, Rkdetector)
      http://www.f-secure.com/v-descs/mailbot_az.shtml

What is the First Responder's Toolkit? (3)

- Trusted binaries with program that we will execute on the system
  - ➔ Change checksums
    - of the whole file (manipulate strings, add extra data)
    - of .text sections (ADMmutate or Hydan)

Avoiding the use of system wide DLLs (1)

- We do not want to use the systems own DLLs since
    - ➔ We do not want to touch the timestamps
    - ➔ We can not trust the systems own DLL-files

trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\trusted_fport.exe	SUCCESS	FileNameInformation
trusted_fport.e:3032	OPEN	C:\WINDOWS\Prefetch\TRUSTED_FPORT.EXE-01820E72.pf	NOT FOUND	Options: Open Access: All
trusted_fport.e:3032	OPEN	C:\seminarium	SUCCESS	Options: Open Directory Access: Traver..
trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\trusted_fport.exe.Local	NOT FOUND	Attributes: Error
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 94208 Length: 16384
trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\PSAPI.DLL	NOT FOUND	Attributes: Error
trusted_fport.e:3032	QUERY INFORMATION	C:\WINDOWS\system32\PSAPI.DLL	SUCCESS	Attributes: A
trusted_fport.e:3032	OPEN	C:\WINDOWS\system32\PSAPI.DLL	SUCCESS	Options: Open Access: Execute
trusted_fport.e:3032	CLOSE	C:\WINDOWS\system32\PSAPI.DLL	SUCCESS	
trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\WS2_32.dll	NOT FOUND	Attributes: Error
trusted_fport.e:3032	QUERY INFORMATION	C:\WINDOWS\system32\WS2_32.dll	SUCCESS	Attributes: A
trusted_fport.e:3032	OPEN	C:\WINDOWS\system32\WS2_32.dll	SUCCESS	Options: Open Access: Execute
trusted_fport.e:3032	CLOSE	C:\WINDOWS\system32\WS2_32.dll	SUCCESS	
trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\WS2HELP.dll	NOT FOUND	Attributes: Error
trusted_fport.e:3032	QUERY INFORMATION	C:\WINDOWS\system32\WS2HELP.dll	SUCCESS	Attributes: A
trusted_fport.e:3032	OPEN	C:\WINDOWS\system32\WS2HELP.dll	SUCCESS	Options: Open Access: Execute
trusted_fport.e:3032	CLOSE	C:\WINDOWS\system32\WS2HELP.dll	SUCCESS	
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 57344 Length: 32768
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 110592 Length: 4096
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 24576 Length: 32768
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 4096 Length: 20480
trusted_fport.e:3032	READ	C:\seminarium\trusted_fport.exe	SUCCESS	Offset: 90112 Length: 4096
trusted_fport.e:3032	QUERY INFORMATION	C:\seminarium\iphlpapi.dll	NOT FOUND	Attributes: Error
trusted_fport.e:3032	QUERY INFORMATION	C:\WINDOWS\system32\iphlpapi.dll	SUCCESS	Attributes: A
trusted_fport.e:3032	OPEN	C:\WINDOWS\system32\iphlpapi.dll	SUCCESS	Options: Open Access: Execute
trusted_fport.e:3032	CLOSE	C:\WINDOWS\system32\iphlpapi.dll	SUCCESS	
trusted_fport.e:3032	CLOSE	C:\seminarium	SUCCESS	

Avoiding the use of system wide DLLs (2)

- Standard Search Order
  http://msdn2.microsoft.com/en-us/library/ms682586(VS.85).aspx

1. The directory specified by lpFileName
2. The current directory (disabled in SafeDllSearchMode)
3. The system directory. Use the GetSystemDirectory function to get the path of this directory
4. The 16-bit system directory. There is no function that obtains the path of this directory, but it is searched
5. The Windows directory. Use the GetWindowsDirectory function to get the path of this directory
6. The directories that are listed in the PATH environment variable. Note that this does not include the per-application path specified by the App Paths registry key

Avoiding the use of system wide DLLs (3)

- Put the DLL files in the same directory

trusted_fport.e:3044	OPEN	C:\SEMINARIUM\FPORT\TRUSTED_FPORT.EXE	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\SEMINARIUM\FPORT\TRUSTED_FPORT.EXE	SUCCESS	Length: 114688
trusted_fport.e:3044	OPEN	C:\SEMINARIUM\FPORT\PSAPI.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 0 Length: 24576
trusted_fport.e:3044	QUERY INFORMATION	C:\SEMINARIUM\FPORT\PSAPI.DLL	SUCCESS	Length: 23040
trusted_fport.e:3044	OPEN	C:\SEMINARIUM\FPORT\WS2_32.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 0 Length: 32768
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 53248 Length: 32768
trusted_fport.e:3044	QUERY INFORMATION	C:\SEMINARIUM\FPORT\WS2_32.DLL	SUCCESS	Length: 82944
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\MSVCRT.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\WINDOWS\SYSTEM32\MSVCRT.DLL	SUCCESS	Length: 343040
trusted_fport.e:3044	OPEN	C:\SEMINARIUM\FPORT\WS2HELP.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 0 Length: 20480
trusted_fport.e:3044	QUERY INFORMATION	C:\SEMINARIUM\FPORT\WS2HELP.DLL	SUCCESS	Length: 19968
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\ADVAPI32.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\WINDOWS\SYSTEM32\ADVAPI32.DLL	SUCCESS	Length: 616960
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\RPCRT4.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\WINDOWS\SYSTEM32\RPCRT4.DLL	SUCCESS	Length: 581120
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\CTYPE.NLS	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\WINDOWS\SYSTEM32\CTYPE.NLS	SUCCESS	Length: 8386
trusted_fport.e:3044	OPEN	C:\SEMINARIUM\FPORT\IPHLPAPI.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 0 Length: 32768
trusted_fport.e:3044	READ	C:	SUCCESS	Offset: 65536 Length: 32768
trusted_fport.e:3044	QUERY INFORMATION	C:\SEMINARIUM\FPORT\IPHLPAPI.DLL	SUCCESS	Length: 94720
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\USER32.DLL	SUCCESS	Options: Open Access: All
trusted_fport.e:3044	QUERY INFORMATION	C:\WINDOWS\SYSTEM32\USER32.DLL	SUCCESS	Length: 577024
trusted_fport.e:3044	OPEN	C:\WINDOWS\SYSTEM32\GDI32.DLL	SUCCESS	Options: Open Access: All

Avoiding the use of system wide DLLs (4)

- Put the DLL files in the same directory
- Dynamic Link Library Redirection
  http://msdn2.microsoft.com/en-us/library/ms682600(VS.85).aspx
  - ➔ A file named just as the binary itself plus a suffix of ‘.local’ causes Windows to check the application directory first whenever it loads a DLL, regardless of the path specified to LoadLibrary or LoadLibraryEx.
  - ➔ As of Windows XP a directory named as the binary plus a suffix of ‘local’ can be used for even more flexibility

Avoiding the use of system wide DLLs (5)

- Put the DLL files in the same directory
- Dynamic Link Library Redirection

trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local	SUCCESS	Attributes: D
System:4	IRP_MJ_QUERY_INFO...	C:\seminarium\fport\trusted_fport.exe	SUCCESS	FileNameInformation
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	Attributes: A
trusted_fport.e:4084	IRP_MJ_CREATE	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	Options: Open Access: Execute
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\psapi.dll	SUCCESS	
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	
System:4	IRP_MJ_QUERY_INFO...	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	FileNameInformation
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	IRP_MJ_CREATE	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	Options: Open Access: Execute
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\ws2_32.dll	SUCCESS	
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	
System:4	IRP_MJ_QUERY_INFO...	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	FileNameInformation
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	IRP_MJ_CREATE	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	Options: Open Access: Execute
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\ws2help.dll	SUCCESS	
trusted_fport.e:4084	IRP_MJ_CLOSE	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	
System:4	IRP_MJ_QUERY_INFO...	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	FileNameInformation
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2_32.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\WS2HELP.dll	SUCCESS	Attributes: A
trusted_fport.e:4084	FASTIO_QUERY_OPEN	C:\seminarium\fport\trusted_fport.exe.Local\PSAPI.DLL	SUCCESS	Attributes: A

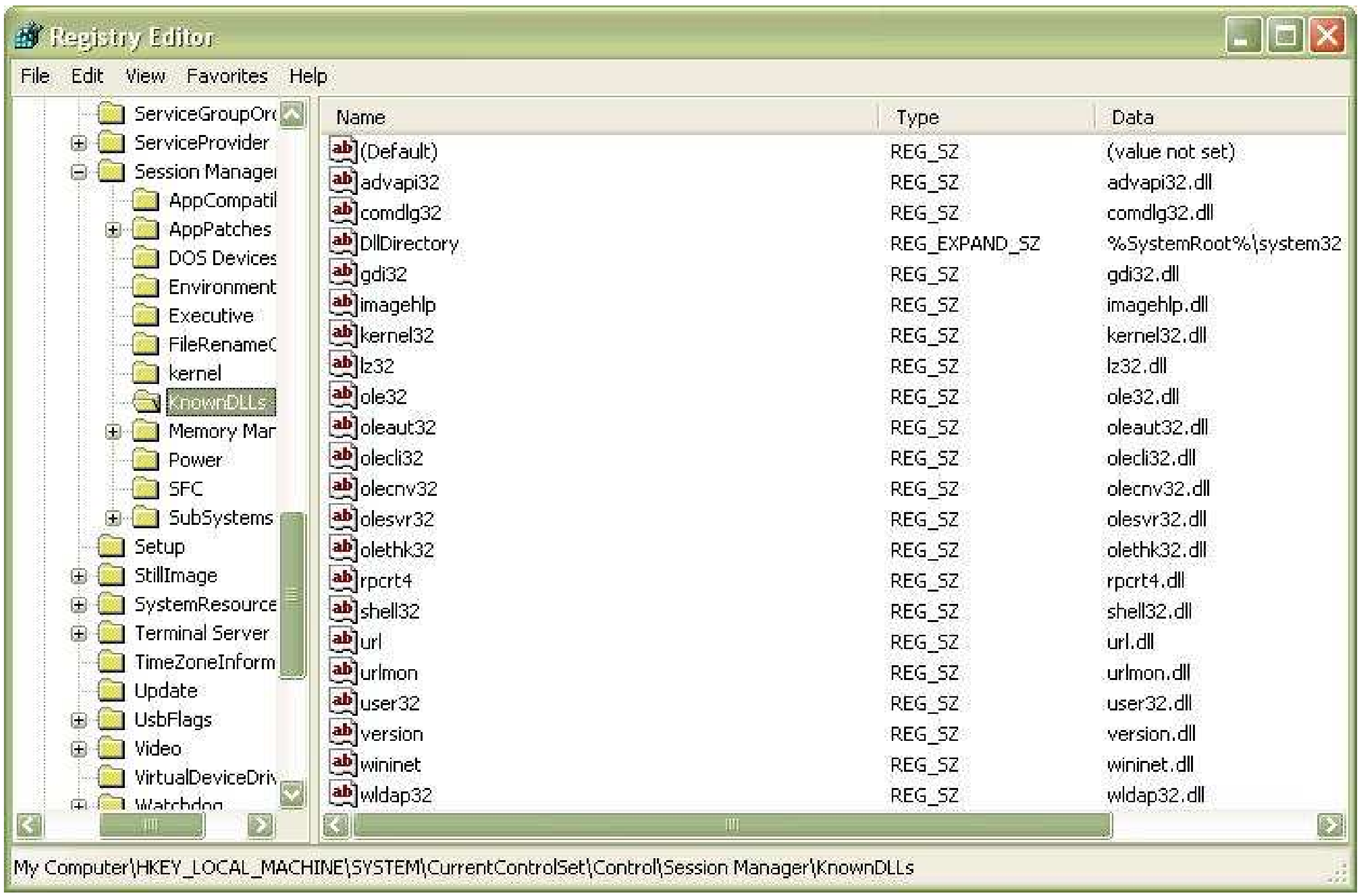
Avoiding the use of system wide DLLs (6)

- Put the DLL files in the same directory
- Dynamic Link Library Redirection
- Edit the PE-header

```
PSAPI.DLL:'EnumProcessModules'
    i trusted_fport.exe     1 r0001a852 r0001a852 PSAPI.DLL:hint0012 PSAPI.DLL:'GetModuleBaseNameA'
    _0001a600 r0001a600 v0041a600 rva_lookup  r0001a784
    _0001a604 r0001a604 v0041a604 timestamp   00000000
    _0001a608 r0001a608 v0041a608 forwarder   00000000
    _0001a60c r0001a60c v0041a60c rva_dllname  r0001a888
    _0001a610 r0001a610 v0041a610 rva_address  r00017134
    i trusted_fport.exe     0 r8000000f r8000000f WS2_32.dll:ord0015
    _0001a614 r0001a614 v0041a614 rva_lookup  r0001a660
    _0001a618 r0001a618 v0041a618 timestamp   00000000
    _0001a61c r0001a61c v0041a61c forwarder   00000000
    _0001a620 r0001a620 v0041a620 rva_dllname  r0001a99e
    _0001a624 r0001a624 v0041a624 rva_address  r00017010
```

Avoiding the use of system wide DLLs (7)

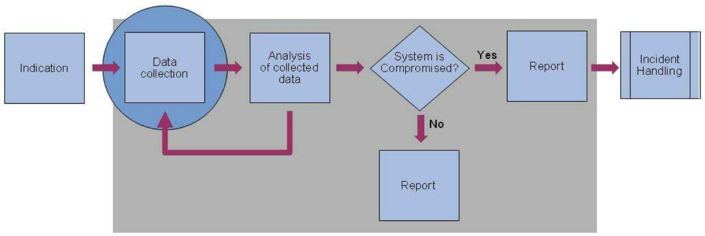
- Does not work on “Known DLLs”



Data Collection

First Responder‘s Toolkit

Exercise 1



Auto starting the data collection (1)

- CDROM
  - ➔ Autorun.inf

    [autorun]

    open=trusted_cmd.exe

Auto starting the data collection (2)

- Non flash-3-tier USB
  - autorun.inf
- USB flash-3-tier
  - U3
  - Flex-IT

Starting a shell that we trust

- Validate a command prompt that is already on the machine
  - Compare with list of known checksums of cmd.exe
- Use a portable system independent shell that is a part of our Toolkit
  - Cygwin
  - SFU (Services for Unix)
  - Portable Command Prompt (Portable Apps)

Run our shell with Administrator privileges

- Almost all of the data collection needs to be done with Administrator privileges
- Do not log off or switch user!

Escalating the current user to Administrator (1)

- runas.exe, WinSudo, Sudo for Windows
  - ➔ Depends on the “Secondary Logon” Service
- Temporally add the current user to the Local Administrator group, execute our shell and the remove the user from the group.
  - ➔ Sudo for Windows by Reinhard Tchorz
    http://www.rt-sw.de/en/freeware/freeware.html

Escalating the current user to Administrator (2)

- Windows Vista – User Account Control (UAC)
  - Consent Prompt - User is administrator
  - Credential Prompt – User is not administrator

Network based communication with the Analysis Server (1)

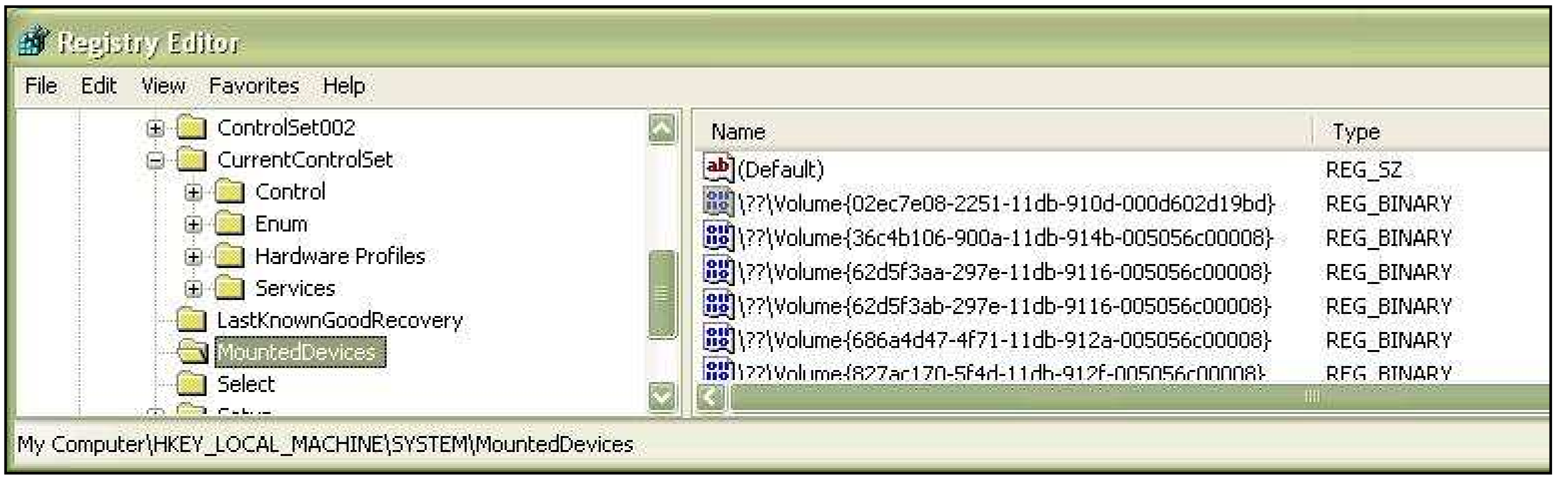
- Netcat
  - ”nc.exe -l -p 4000 > evidence.txt”
  - ”command | nc analys.sitic.se 4000”
- SMB
  - ”command > \\analys.sitic.se\share\evidence.txt”
- TFTP, FTP, HTTP (WebDav, POST or PUT)

Network based communication with the Analysis Server (2)

- Is the communication port blocked?
  - ➔ Personal Firewall rules might be needed to be changed
  - ➔ Corporate Firewall rules might also be needed to be changed

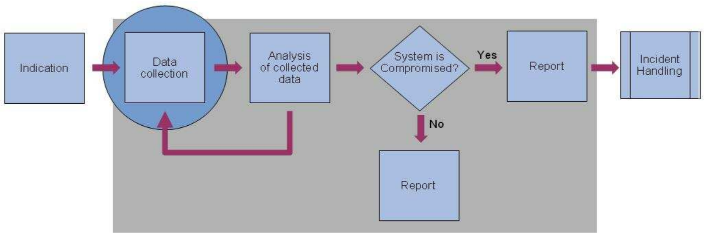
Local communication with the Analysis Server

- External USB or FireWire hard drives
    - ➔ Changes integrity of the system



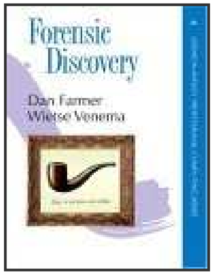
Data Collection

Order of Volatility



Best Practice: Collection of data in the "order of volatility"

- 2002: RFC 3227
  Guidelines for Evidence Collection and Archiving



- 2004: Dan Farmer and Wietse Venema
  Forensic Discovery
- 2006: NIST Special Publication 800-86
  Guide to Integrating Forensic Techniques into Incident Response

Current practice: Pull the plug!

Data Collection
Order of Volatility

What is the proper order of volatility?

RFC 3227

- Registers, cache
- Network status
- Process information
- Main memory
- Temporary file systems
- Disk
- Remote logging and monitoring data that is relevant to the system in question
- Physical configuration, network topology
- Archival media

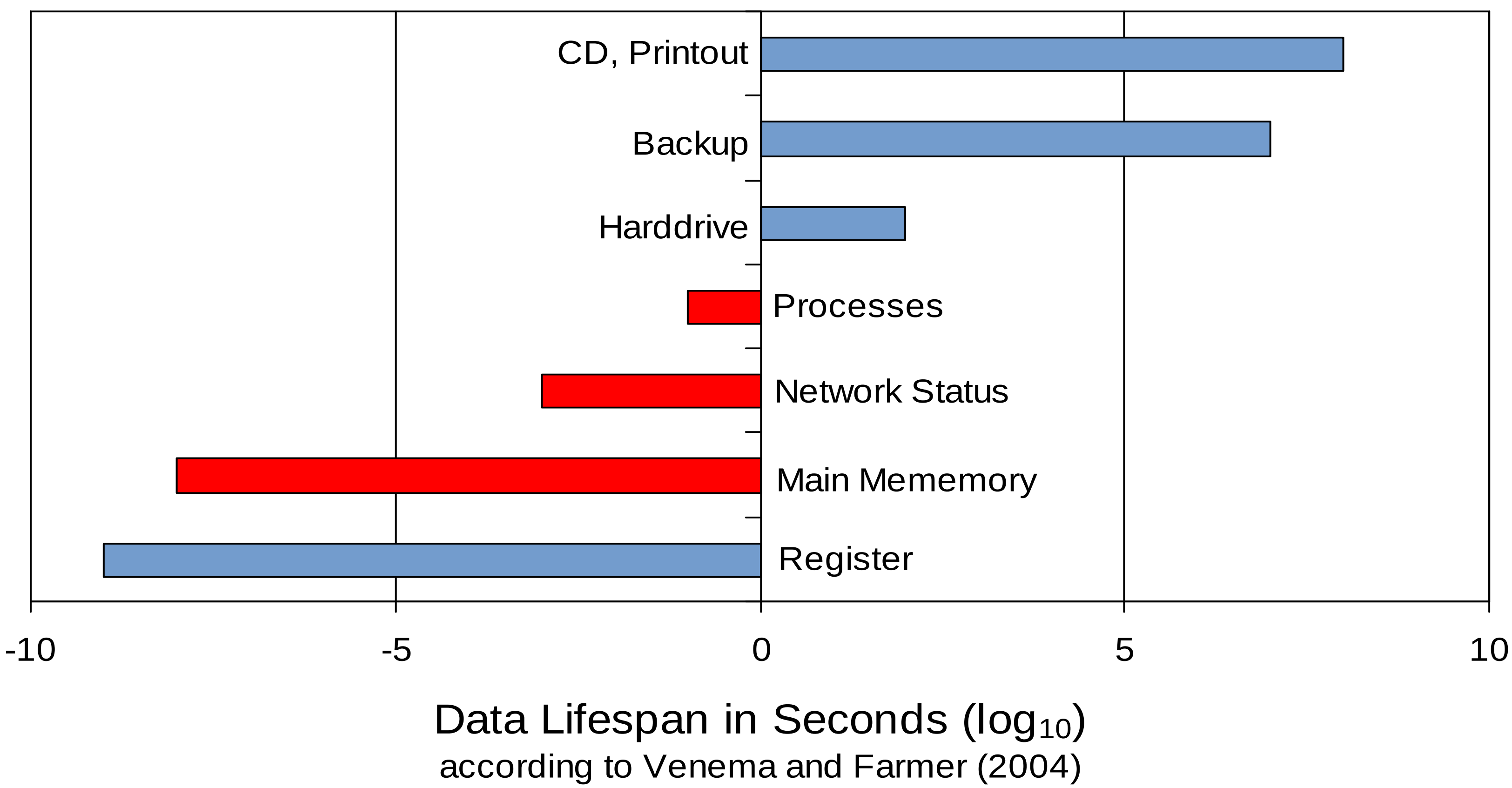
Forensic Discovery

- Registers, peripheral memory, caches, etc.
- Main memory
- Network status
- Process information
- Disk
- Floppies, backup media, etc.
- CD-ROMs, printouts, etc.

NIST SP 800-86

- Network status
- Login sessions
- Main memory
- Process information
- Open files
- Network configuration
- Operating system time

Data Collection
Order of Volatility



Action	% RAM unchanged	
	256 MB RAM	512 MB RAM
Start	100.0	100.0
Idle for 1 hour	90.4	96.7
Idle for 2 hours	79.7	96.1
run dd from Helix CD	76.9	89.8
Idle for 15 hours	74.8	85.6
run WFT from Helix CD	67.2	69.4

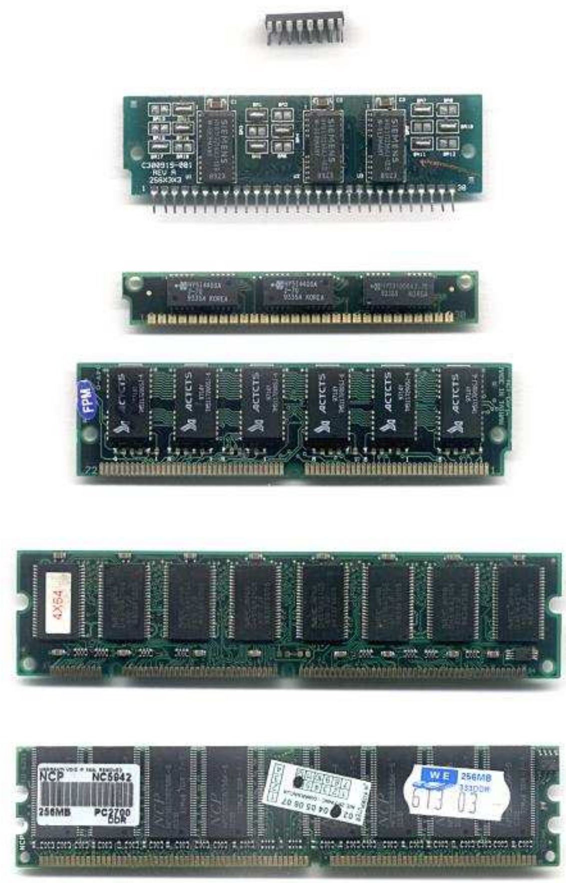
Effects on main memory, according to Walters and Petroni (2006)

Excursus

Concepts of Memory

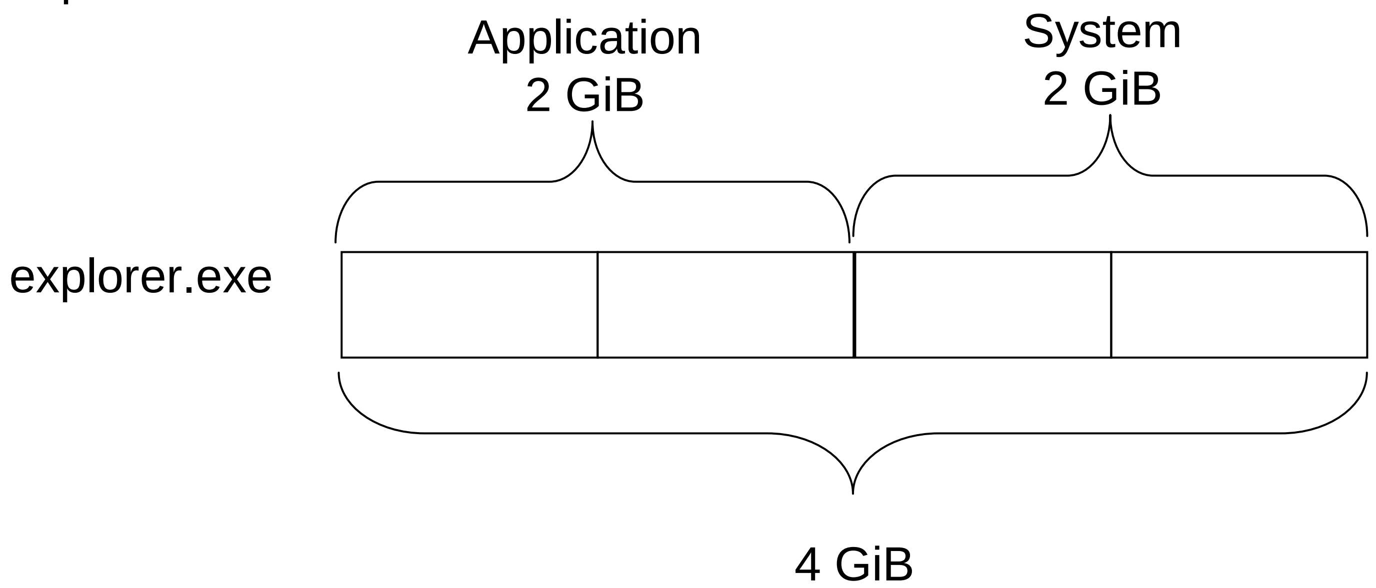
Excursus
Concepts of Memory

- Physical memory is the short-term memory of a computer.
- Rapid decay of information as soon as memory module is disconnected from power and clock sources.
  - ➔ More on the rapid decay later!



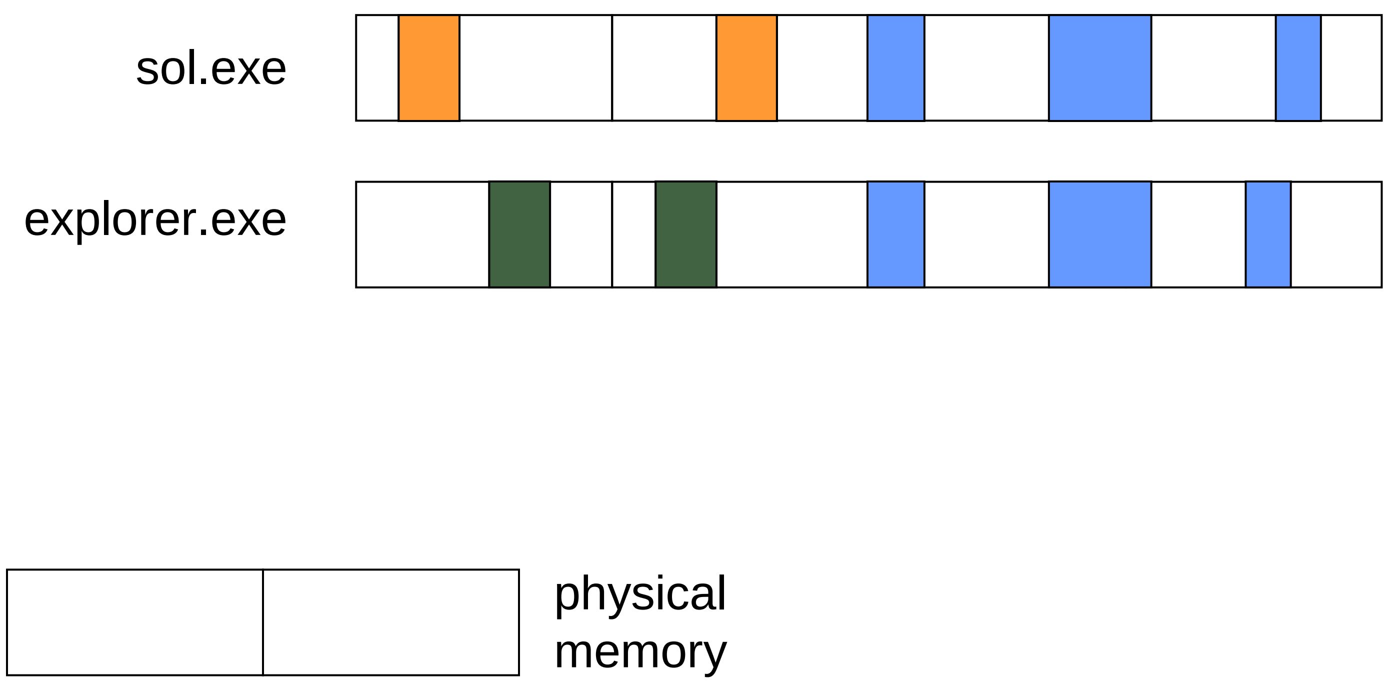
Physical vs. Virtual Memory (1)

- 4 GiB of (virtual) address space per process
- Split into halves



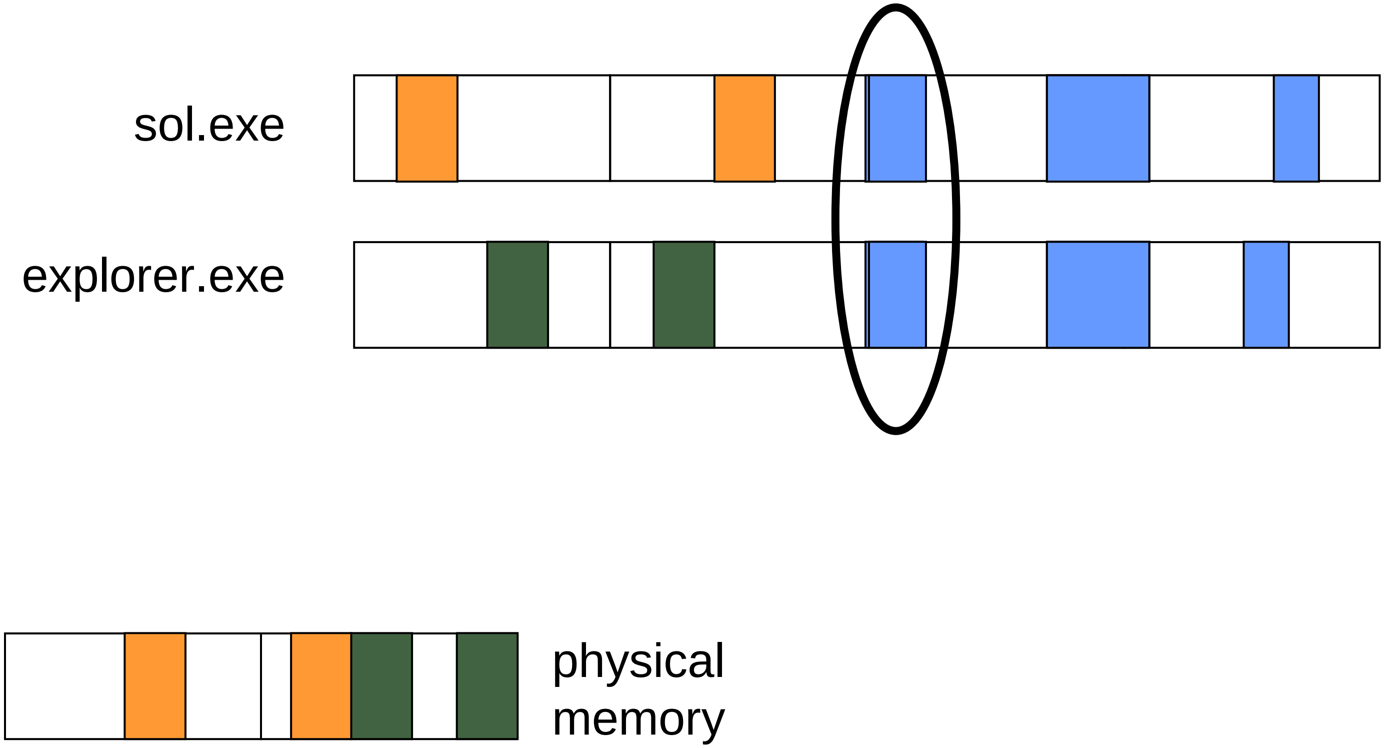
Physical vs. Virtual Memory (2)

- Physical memory is divided into so called “pages”.
- Allocated virtual memory is mapped onto physical memory page by page.

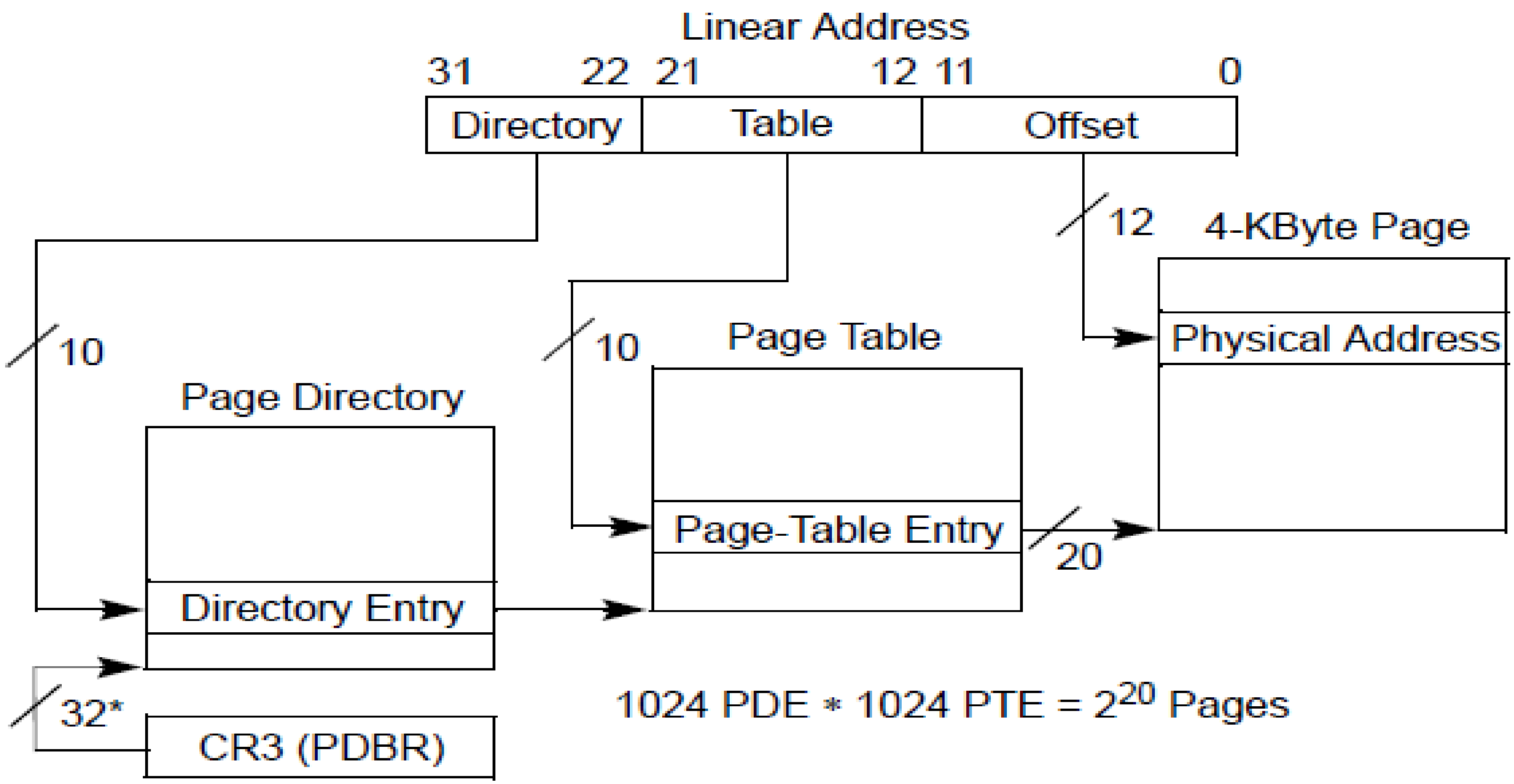


Physical vs. Virtual Memory (3)

- The same page of physical memory can appear at different locations within the same address space or in different address spaces.



x86 and 4k page



*32 bits aligned onto a 4-KByte boundary.

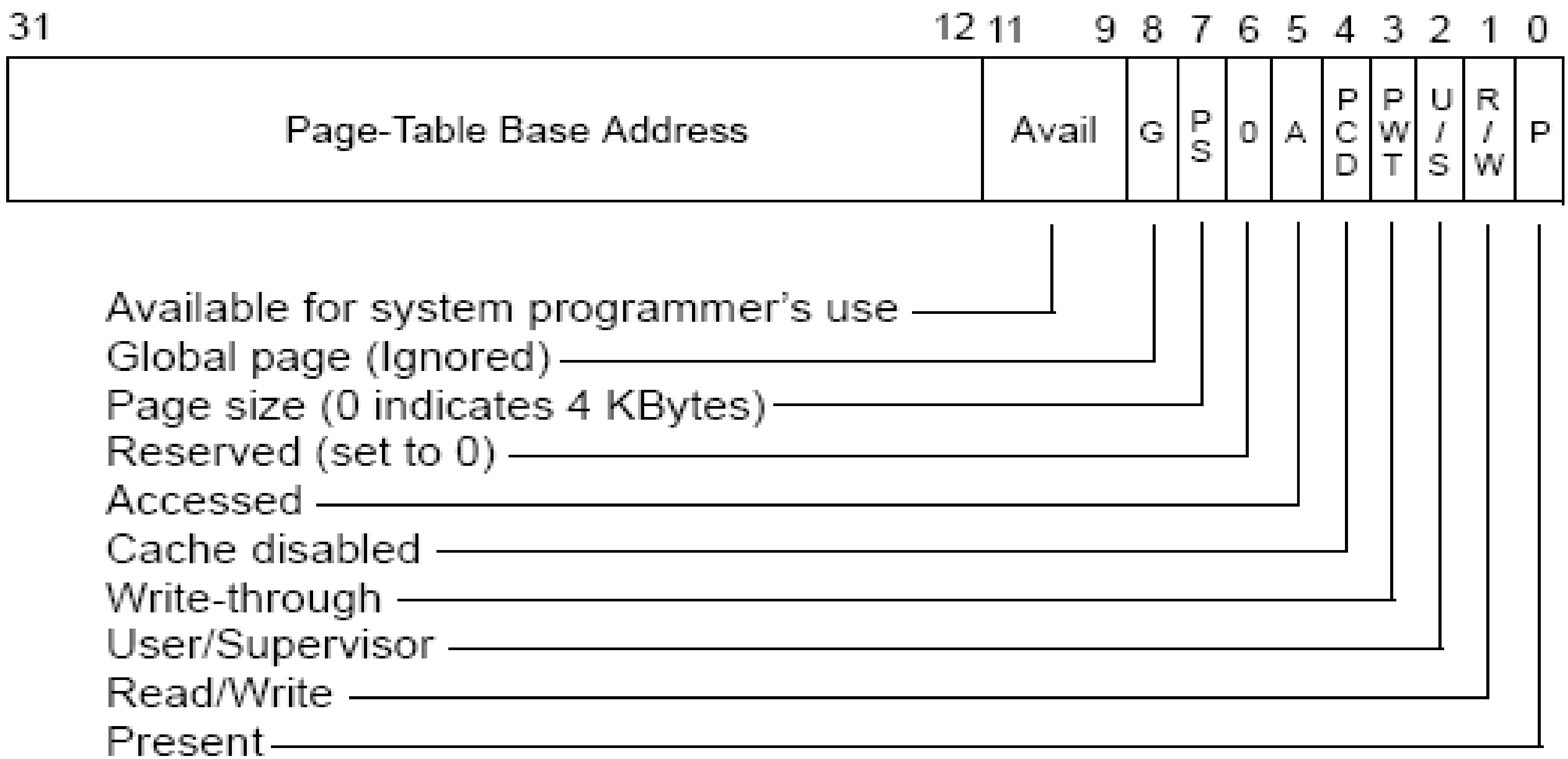
IA-32 Intel Architecture Software Developer's Manual, Vol. 3A

Page Directory and Page Table

- Page Directory:
  - ➔ Provides a bird‘s eye view of a process’ the virtual address space.
  - ➔ States whether a page is 4 kiB or 2/4 MiB.

- Page Table:
  - ➔ States whether the page is valid or invalid.
  - ➔ Page Frame Number (physical address / 0x1000)

x86, Page Directory Entry



IA-32 Intel Architecture Software Developer's Manual, Vol. 3A

Important Flags

- Present, bit 0
  - 0 = page is not readily accessible in physical memory, triggers a Page Fault Exception (#PF)
  - 1 = page is accessible
  - also known as Valid flag (Microsoft Windows)
- Page Size, bit 7
  - 0 = page size is 4k, go through Page Table
  - 1 = page size is 4M, direct access to page

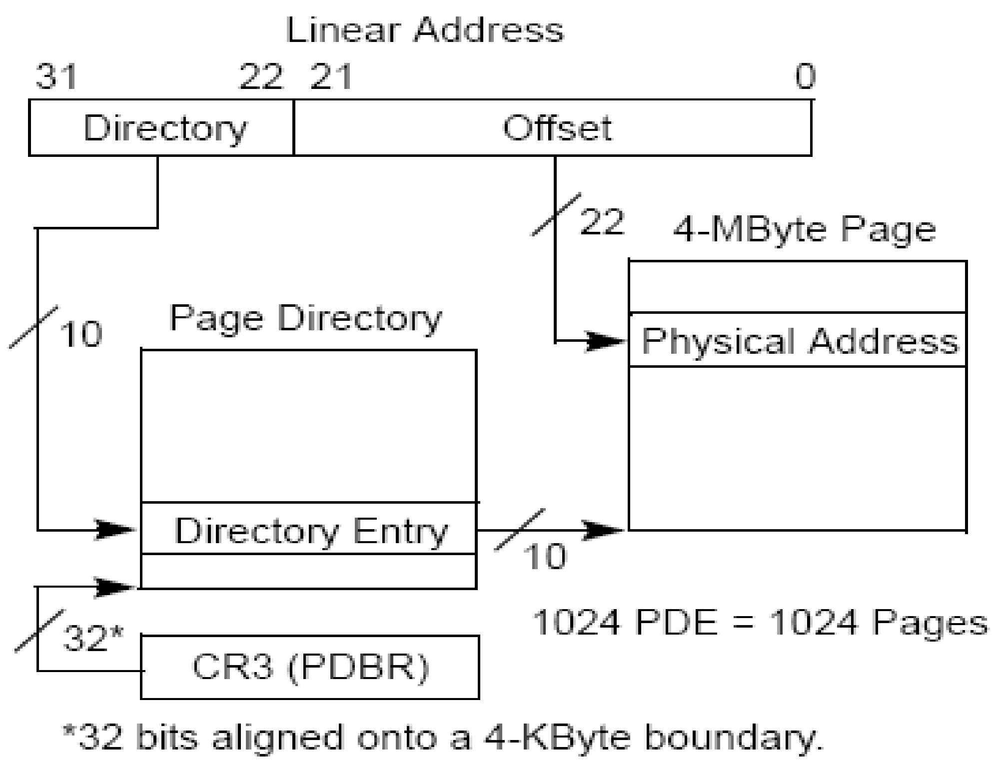
Invalid Pages in Microsoft Windows

- Proper response to a #PF exception is up to the operating system.
- Types of invalid pages:
  - Swap: The page has been moved into a page file.
  - Demand Zero: Return a page filled with NULL bytes.
  - Transition: The page is kept in either one of the modified, written (standby) or free pages lists.
  - Prototype: The page is accessed from different processes. The processes do not reference the desired memory page, but a prototype PTE. The prototype then points to the final page (similar to a symlink).
- Invalid pages may exist in physical memory! (Jesse Kornblum, 2007)

Further Reading

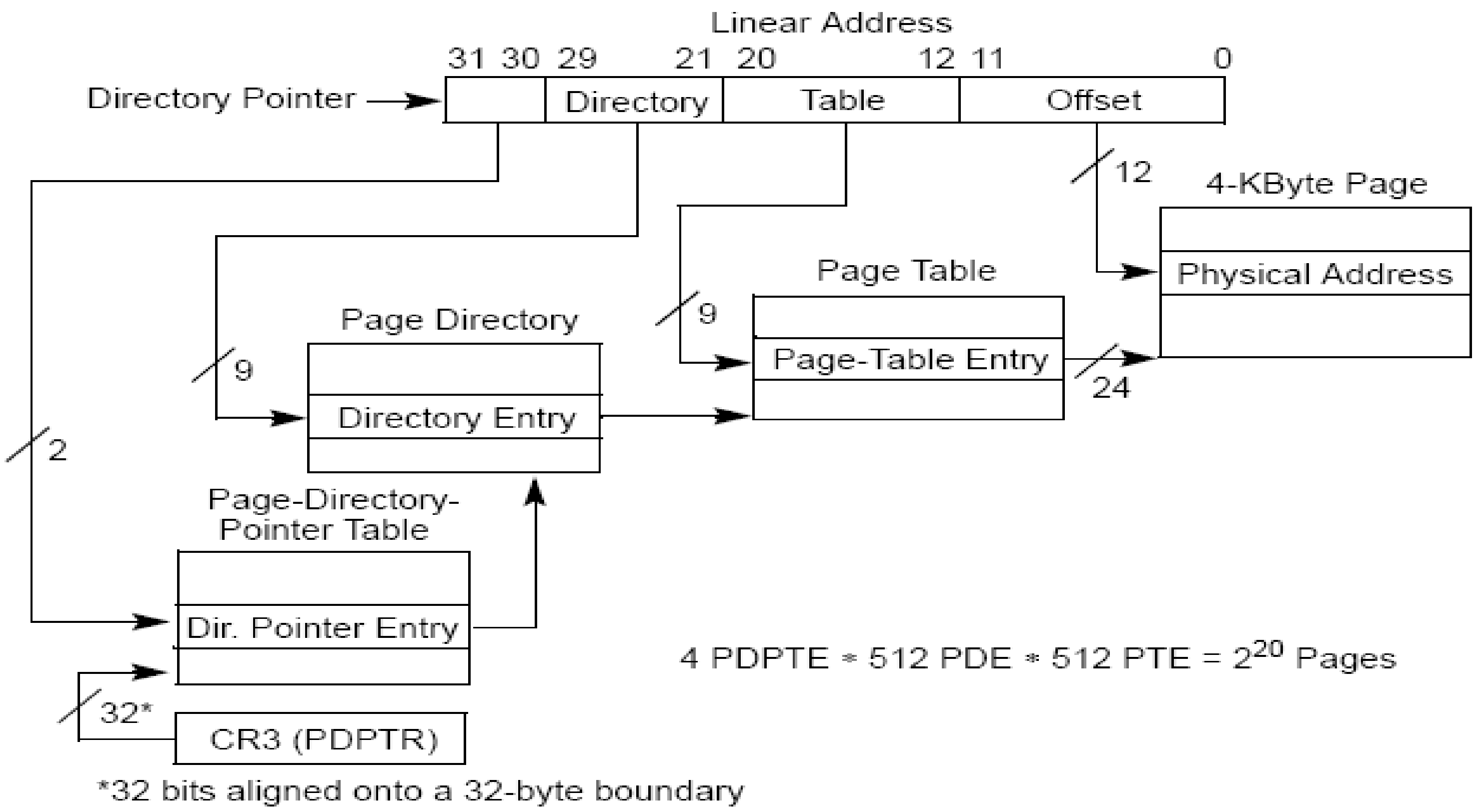
- For details on addressing, page directories and page tables please see:
  - Russinovich and Salomon “Windows Internals“, 4th ed., chapter 7.
  - Intel® 64 and IA-32 Architectures Software Developer's Manual
    http://www.intel.com/products/processor/manuals
  - “Using Every Part of the Buffalo in Windows Memory Analysis”
    by Jesse Kornblum (2007)
    http://jessekornblum.com/research/papers/buffalo.pdf

x86 and 4M page

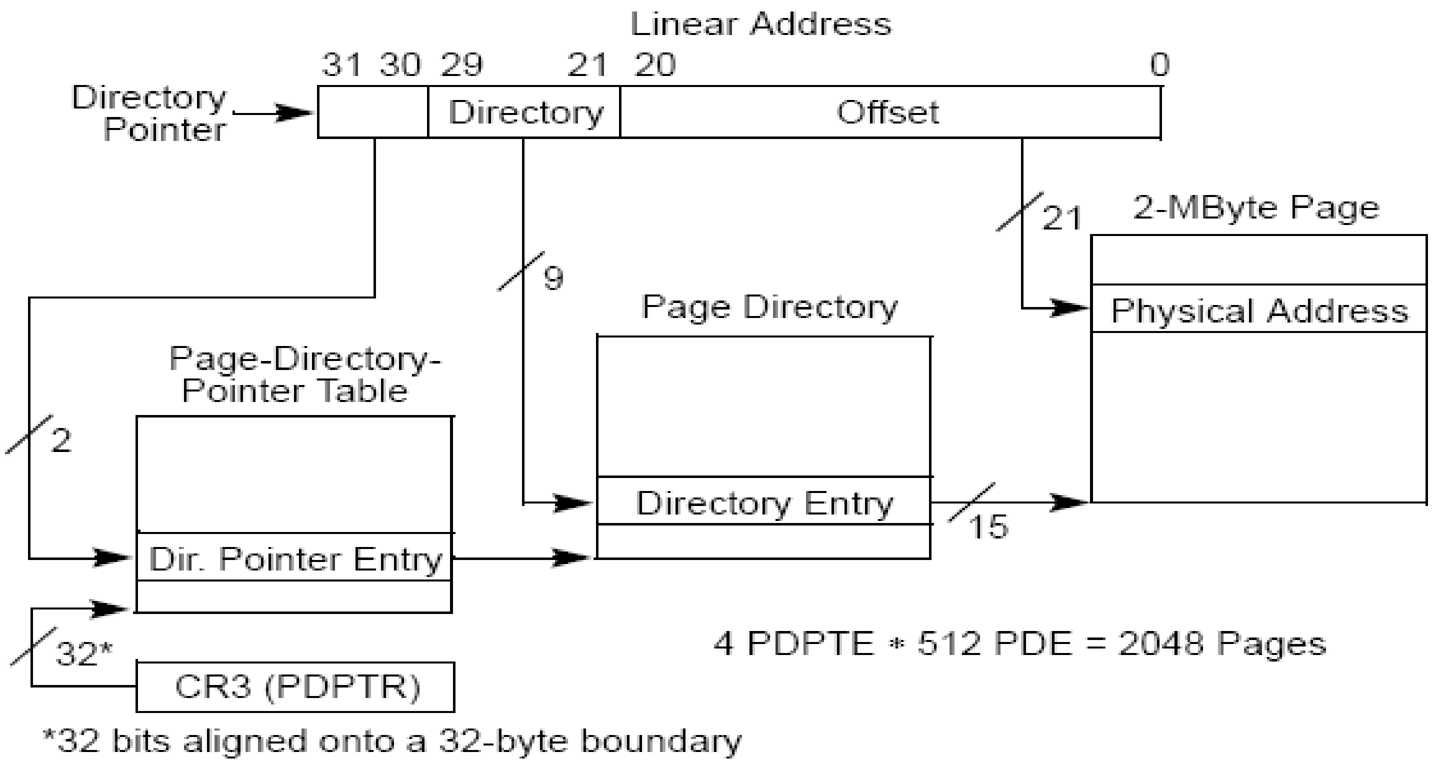


IA-32 Intel Architecture Software Developer's Manual, Vol. 3A

x86, PAE and 4k pages

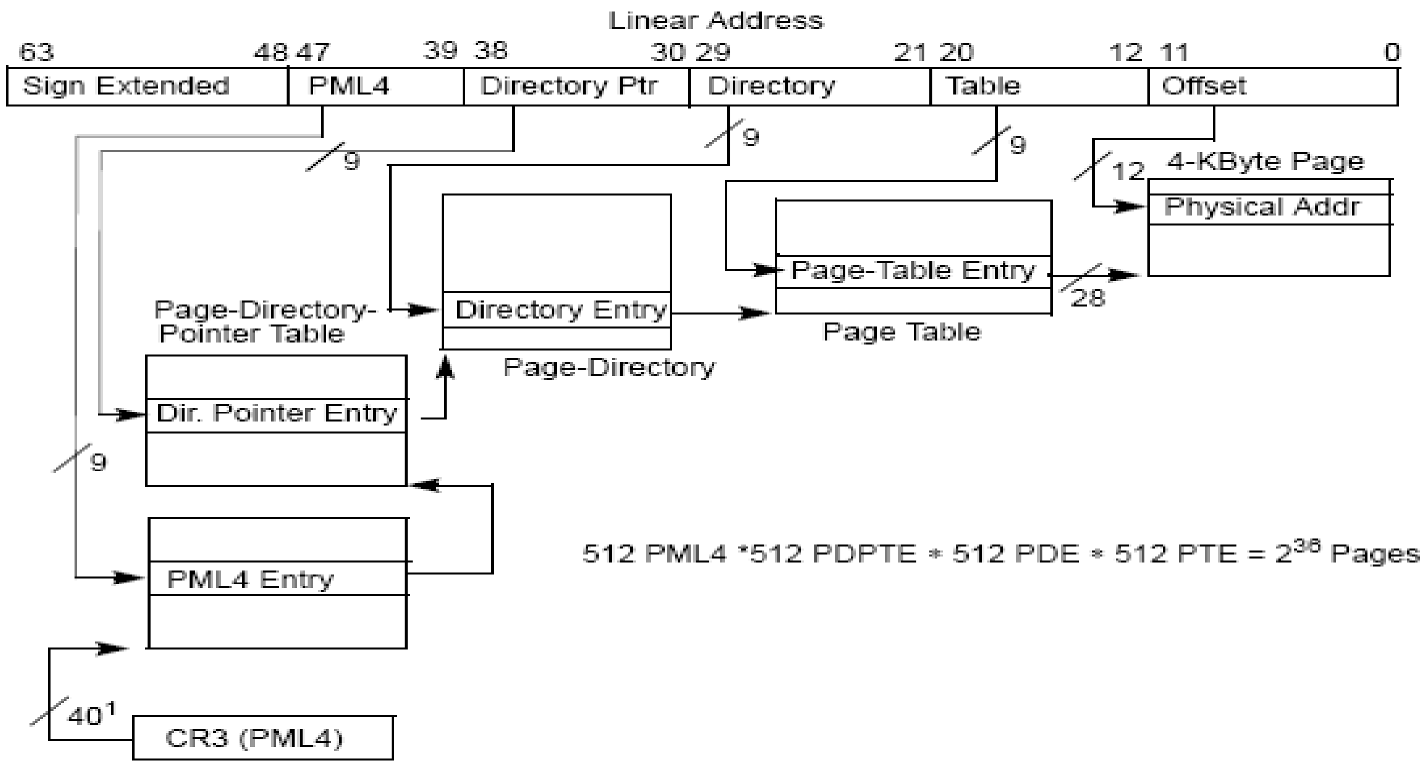


x86, PAE (2M pages)



IA-32 Intel Architecture Software Developer's Manual, Vol. 3A

IA-32e (64bit architecture), 4k pages

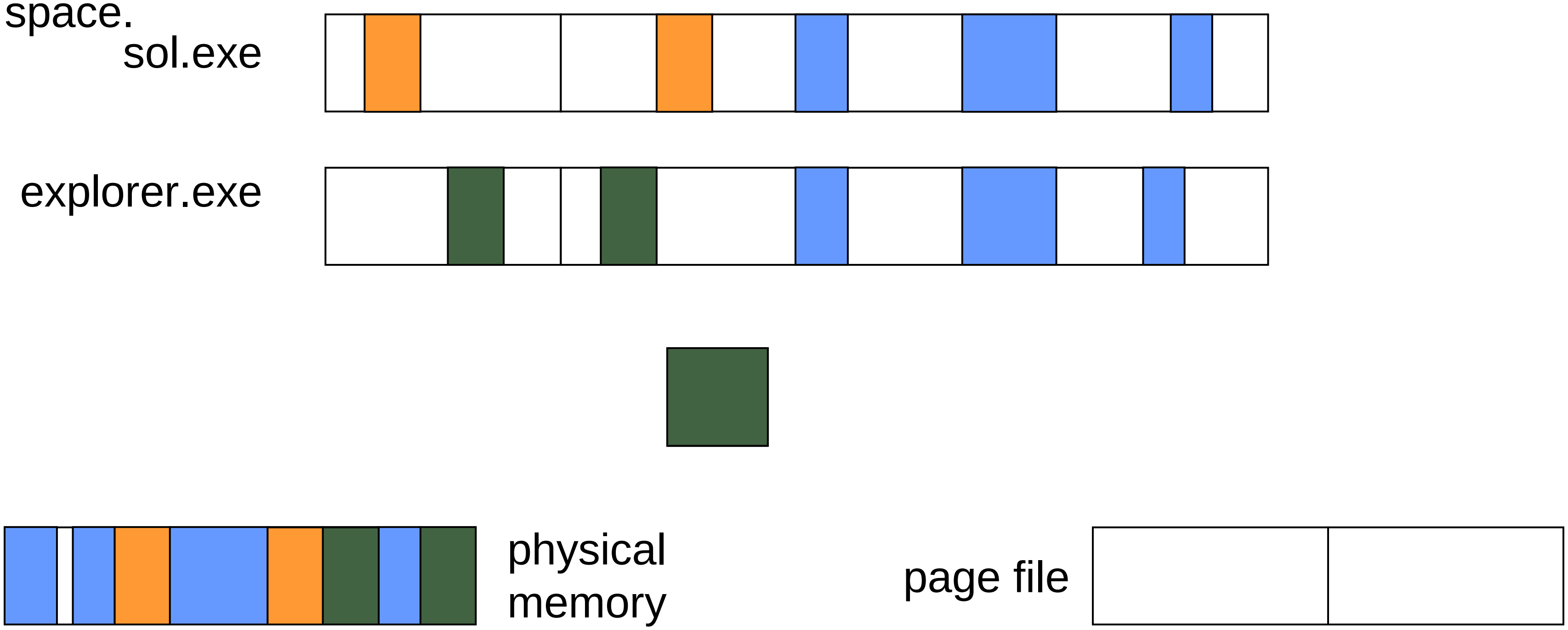


NOTES:
1. 40 bits aligned onto a 4-KByte boundary

IA-32 Intel Architecture Software Developer's Manual, Vol. 3A

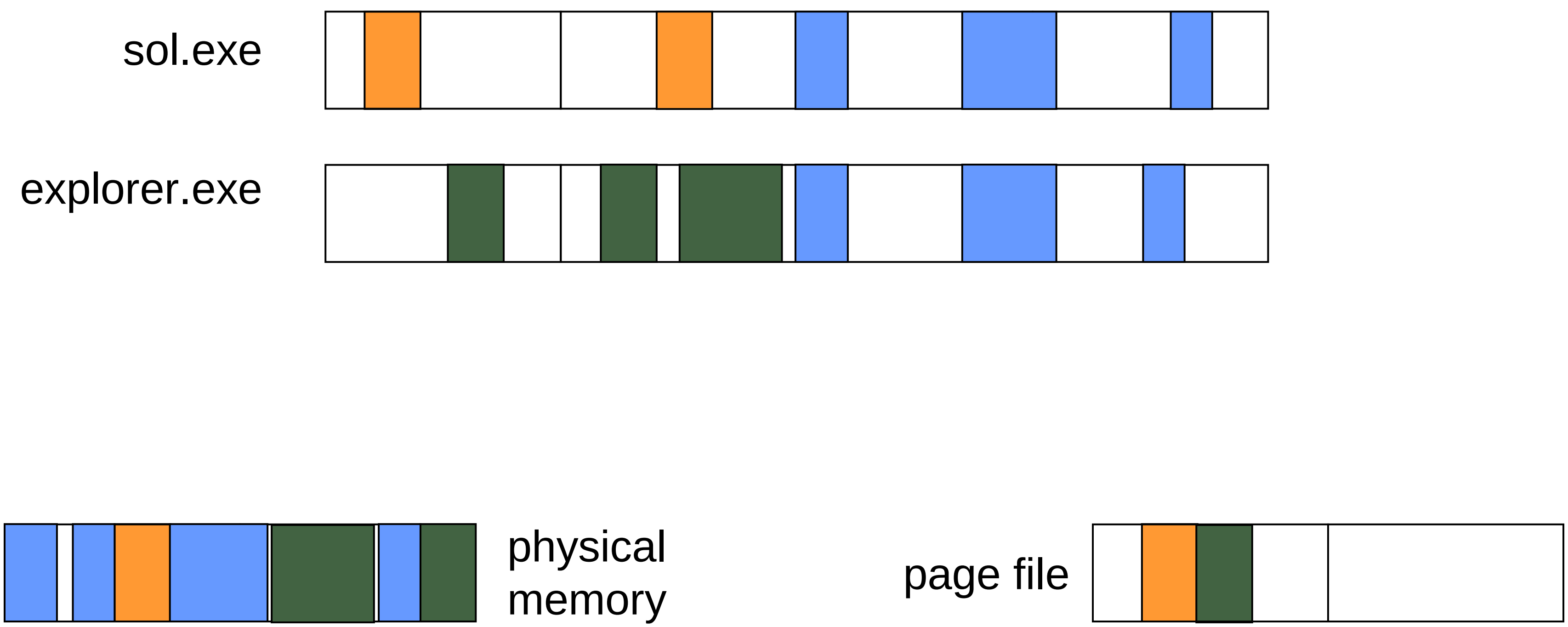
Page file

- Data can be moved from physical memory into a page file to clear some space.



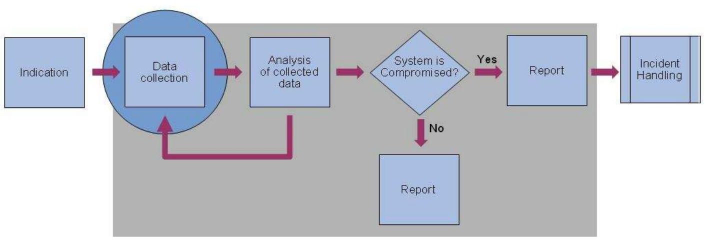
Freed pages

- Memory does not get over written when it is marked as free



Data Collection

Main Memory



Classification of Methods

- Access to main memory
  pure hardware vs. software
- Time of installation
  prior to incident vs. post incident
- Required privileges
  user vs. administrator
- Impact on system
  in vivo vs. post mortem
- Atomicity of image
- Image file format
  raw vs. Microsoft crash dump

Access to Main Memory

Software

- Affects CPU, memory, kernel and drivers.
- Can easily be fooled.
- Costs mainly driven by license.
- Easy to deploy and maintain in a corporate environment.
- Low atomicity of resulting image

Pure Hardware

- Does not utilize the CPU.
- Usually requires extra hardware, FireWire might be an exception.
- Installation requires significant time (more costs).
- Trusted access to memory? Rutkowska attack on DMA
- Higher atomicity of resulting image.

Installation

prior to incident

- Installation required prior to the incident.
- Usually requires a reboot.
- Does not tamper with evidence.
- Permanently adds (privileged) code to system, increases exposure to attacks.

post incident

- Installation possible after the incident occurred.
- Could interfere with evidence.
- “Installed” only as long as needed.

Required Privileges

Unprivileged

- User-level access.
- No (secondary) logon required.
- Minimized impact on evidence.

Privileged

- Administrator / SYSTEM privileges.
- Requires either installation prior to incident or (secondary) logon.
- High impact on evidence in case of a (secondary) logon.

Impact on system

Low

- in-vivo: system continues to work.
- Degraded performance during imaging, reverts to normal afterwards.
- Generally should be safe even on servers.
- Low atomicity of resulting image.

High

- post-mortem: system forced to crash.
- System out of service for time required to obtain the dump and reboot. Extra time may be required to restore functionality afterwards.
- Acceptable only for clients. Generally best choice under lab conditions.
- High atomicity of resulting image.

Atomicity of Image

Low

- “blurred” image.
- Inconsistent state; may confuse tools and examiners (e.g. dangling pointers).
- Significant problem for analysis of user data.
- Low impact on analysis of kernel data.

High

- “crisp” image.
- Consistent state.
- Usually difficult to achieve..

Dump file format

Raw

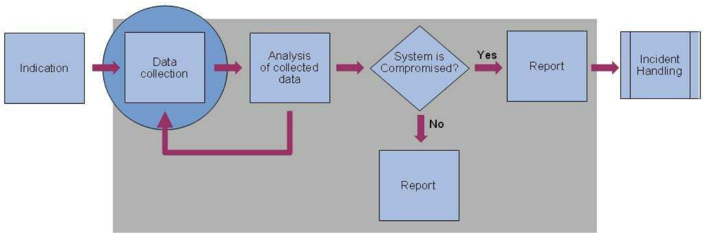
- 1:1 copy of physical memory.
- offset == physical address
- Several proof-of-concept tools only operate on this format.

Crashdump

- Extension .DMP
- CPU state information
- One or many blocks of physical memory.
- Holes, e.g. Bios, DMA, AGP video.
- Extra data from devices that use nt!KeRegisterBugCheckReasonCallback.
- Microsoft Tools require this format.

Data Collection

Main Memory- Tools and Techniques



Dedicated Hardware - Tribble

- by Brian Carrier and Joe Grand (2004)
  http://www.digital-evidence.org/papers/tribble-preprint.pdf
- PCI add-in card
- HLT to CPU
- DMA busmaster
- Output via RS-232
- NOT available.

Dedicated Hardware - Copilot

- by Komoku
- Paper presented at 14th USENIX Security Symposium, 2004.
  http://www.usenix.org/events/sec04/tech/full_papers/petroni/petroni.pdf
- PCI add-in card with single-board microcomputer
- DMA
- Evaluates kernel data structures while the (host) system is running.
- NOT available to the public.

FireWire (1)

- Dornseif and Becher (2004)
  - 0wned by an iPod
    http://md.hudora.de/presentations/firewire/PacSec2004.pdf
  - Hacking with Fire
    http://md.hudora.de/presentations/firewire/2004-firewire-21c3.pdf
- Boileau (2006)
  http://www.security-assessment.com/files/presentations/ab_firewire_rux2k6-final.pdf

FireWire (2)

- OHCI controller can read and write the first 4 GiB of main memory
- Quinn “The Eskimo“ (2003) FireStarter modifies video memory of connected Mac
- Dornseif and Becher (2004)
  0wned by an iPod
  http://md.hudora.de/presentations/firewire/PacSec2004.pdf
- Boileau (2006)
  “Hit by a Bus:Physical Access Attacks with Firewire”
  http://www.security-assessment.com/files/presentations/ab_firewire_rux2k6-final.pdf

FireWire - Drawbacks

- Frequently found on laptops, but it’s rare on desktops.
- Unexpected hang (Vidstrom 2006)
  http://www.ntsecurity.nu/onmymind/2006/2006-09-02.html
- Memory access can be controlled by malicious software (Rutkowska 2007)
- If the examiner can access the suspect, can the suspect access the examiner also?

FireWire - Characteristics

- **Access to main memory**
  hardware
- **Time of installation**
  post incident
- **Required privileges**
  physical access
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  raw

dd

- Most popular method in literature.
- Windows makes physical memory accessible through the \\.\PhysicalMemory and \\.\DebugMemory devices. Copy from device to file.

dd - Implementations

- ~~Port by George. M. Garner Jr.~~
  ~~http://users.erols.com/gmgarner/forensics/~~
- X-Ways Capture (does a lot of other things, too)
  http://www.x-ways.com/capture/

dd - Drawbacks

- Cache coherency on Windows 2000 (Vidstrom 2006)
  http://www.ntsecurity.nu/onmymind/2006/2006-06-01.html

- Devices are not accessible from userland on Windows 64bit, Windows Server 2003 SP 1 and Vista for security reasons.

  ➔ load your own driver or use symlinks

dd - Characteristics

- **Access to main memory**
  software
- **Time of installation**
  post incident
- **Required privileges**
  administrator
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  raw

KnTDD

- by GMG Systems, Inc. (George M. Garner Jr)
  http://www.gmgsystemsinc.com/knttools/
- Accesses physical memory through a driver.
- Also obtains for later analysis
  - kernel and network driver binaries
  - system status as seen from userland
- Enterprise edition allows for digitally signed work packages and encrypted evidence.

- **KnTDD - Characteristics**
- **Access to main memory**
  software
- **Time of installation**
  post incident
- **Required privileges**
  administrator
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  raw and dmp at the same time

ManTech's Memory DD

- By ManTech International Corporation
  http://www.mantech.com/msma/MDD.asp
- Accesses physical memory through a driver.
- Free version available on SourceForge

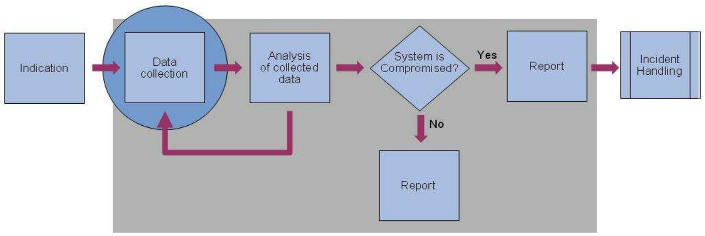
ManTech's Memory DD - Characteristics

- **Access to main memory**
  software
- **Time of installation**
  post incident
- **Required privileges**
  administrator
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  raw

Data Collection

Main memory

Exercise 2



Agent based tools

- The one who hooks first, stays.
- The one who hooks deeper, stays.
- Products:
  - WetStone LifeWire Investigator
    http://www.wetstonetech.com/
  - Technology Pathways ProDiscover IR
    http://www.techpathways.com/ProDiscoverIR.htm
  - Guidance Software EnCase Enterprise
    http://www.encase.com/products/ee_index.aspx
  - Agile RiskManagement Nigilant32 (free)
    http://www.agilerm.net/publications_4.html

Agent based tools - Characteristics

- **Access to main memory**
  software
- **Time of installation**
  pre incident
- **Required privileges**
  administrator (installation)
  unprivileged (activation)
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  raw

LiveKD

- Microsoft's Debugger (kd, WinDbg) can't dump memory on a kernel local connection.
- LiveKD presents live physical memory like a static dump file.
- Requires MS Debugger, LiveKd and a minimum set of debug symbols (PDB) for kernel and HAL.
- Exact software versions must be known prior to memory acquisition!
- From the debugger prompt run
  .dump /f filename

LiveKD - Characteristics

- **Access to main memory**
  software
- **Time of installation**
  pre incident
- **Required privileges**
  administrator
- **Impact on system**
  low
- **Atomicity of image**
  low
- **Image file format**
  dmp

Forced Crash

- Configure system to create a dump on crash.
- Provide means to force a crash.
- Make system crash when needed.
- What happens?
  - ➔ Upon boot: creates dedicated copy of miniport storage driver, named dump_xyz.
  - ➔ Upon crash: writes physical memory into page file on system volume.
  - ➔ Upon reboot: SMSS checks page file for dump signature and locks file.
  - ➔ Winlogon again checks for signature and extracts dump out off page file

Forced Crash - Preparation

- Go to Control Panel > System Properties > Advanced > Startup and Recovery > Settings
  - ➔ The Page File must be of the same size or greater as the memory installed
- For “Write debugging information” chose either the complete or kernel memory dump.
- Can be done conveniently through a registry patch file (.reg)

Forced Crash - Activation

- Kill csrss.exe (Client Server Subsystem).
- Write your own driver that calls nt!KeBugCheck or nt!KeBugCheckEx.
- NotMyFault from Sysinternals
  http://download.sysinternals.com/Files/Notmyfault.zip
- SystemDump from Citrix
  http://support.citrix.com/article/CTX111072
- Bang from OSR
  http://www.osronline.com/article.cfm?article=153
- Activate crash sequence in PS/2 keyboard driver (USB supported in Windows 2003 SP 1).

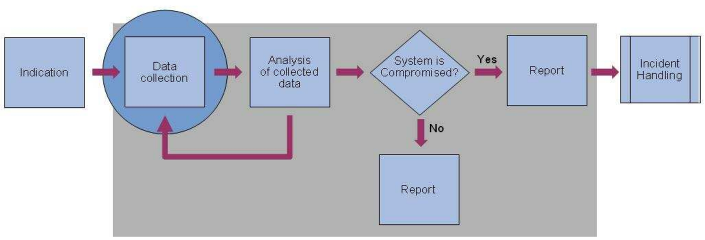
Forced Crash - Characteristics

- **Access to main memory**
  software
- **Time of installation**
  pre incident
- **Required privileges**
  administrator (installation)
  unprivileged (activation)
- **Impact on system**
  high
- **Atomicity of image**
  high
- **Image file format**
  dmp

Data Collection

Main memory

Exercise 3



Anti-forensic attacks (1)

- Ddefy
  - by D. Bilby (2006) http://www.blackhat.com/presentations/bh-jp-06/BH-JP-06-Bilby-up.pdf
  - Hooks entry for `nt!NtMapViewofSection` in System Service Descriptor Table (SSDT).
  - Monitors access to \\.\PhysicalMemory.

Anti-forensic attacks (2)

- Shadow Walker
  - by Sparks and Butler (2005) http://www.blackhat.com/presentations/bh-jp-05/bh-jp-05-sparks-butler.pdf
  - Controls the contents of memory viewed by another application or driver.
  - Modifies page fault handler, marks page as not present, then flushes the Translation Lookaside Buffer (TLB).

Anti-forensic attacks (3)

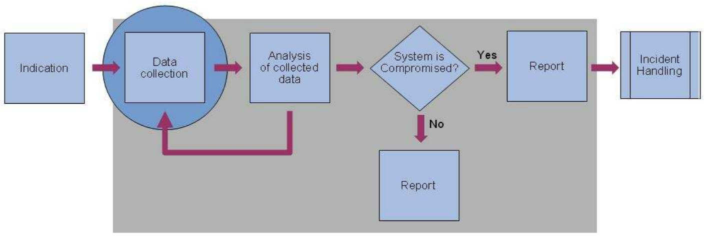
- Redirecting physical memory access
  - by J. Rutkowska (2007) http://invisiblethings.org/papers/cheating-hardware-memory-acquisition-updated.ppt
  - Manipulates configuration of Northbridge.
  - At the same physical address CPU and DMA see different
- Clever software could overcome attack. http://fshypervisor.wordpress.com/2007/05/23//part-a-auscert/

BodySnatcher

- by Bradley Schatz, Evimetry
  http://www.evimetry.com.au/
- Injects a minimal and trusted operating system kernel into the target system
- Not publicly available.

Data Collection

Main Memory- Other sources



Pagefile.sys

- Contains memory pages of kernel (paged pool) and userland processes.
- Age of data highly depends on the system's memory load.
- Really helpful, more about that in the analysis session.

Hibernate.sys

- Does NOT contain all physical memory available to Windows.
- Undocumented file format/data compression algorithm.
- Matthieu Suiche and Nicolas Ruff, 2007
  "Enter Sandman (why you never should go to sleep)"
  http://www.msuiche.net/pres/PacSec07-slides-0.4.pdf
  library and Python bindings enables one to read and write hibernate.sys

Hibernate.sys - Format

File format

Field	Content
Header	PO_MEMORY_IMAGE structure
Page list	Not sure – might be a list of “free pages” for loader use
Processor State	CONTEXT + SPECIAL_REGISTERS structures
Memory Range Array #1	*Header:* list entries count + next list offset + checksum *List:* Up to 255 entries *List entry:* start page + end page + checksum
Xpress Blocks Array #1	*Magic:* “\x81\x81xpress” (Windows > 2000) *Header:* size + checksum + other *Content:* compressed data
Memory Range Array #2	(…)

Conclusion FIRST2007

- You can’t trust the kernel of a compromised system.
- You can’t trust the hardware of a compromised system.
- But you have to rely on both, hardware and software …
- … until someone comes up with a better architecture!

Cold Booting

- Based on research from Princeton University
  - J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten http://citp.princeton.edu/memory
- Showed that memory could retain their contents for seconds to minutes after power is lost.
- Cut the power and boot up the system with a very low memory-impact OS that dumps the memory.
- Freeze the memory modules and transport them to a secure location. Data will survive up to 10 minutes without power.

Cold Booting - Implementations

- msramdump by Robert Wesley McGrew
  http://www.mcgrewsecurity.com/projects/msramdmp/

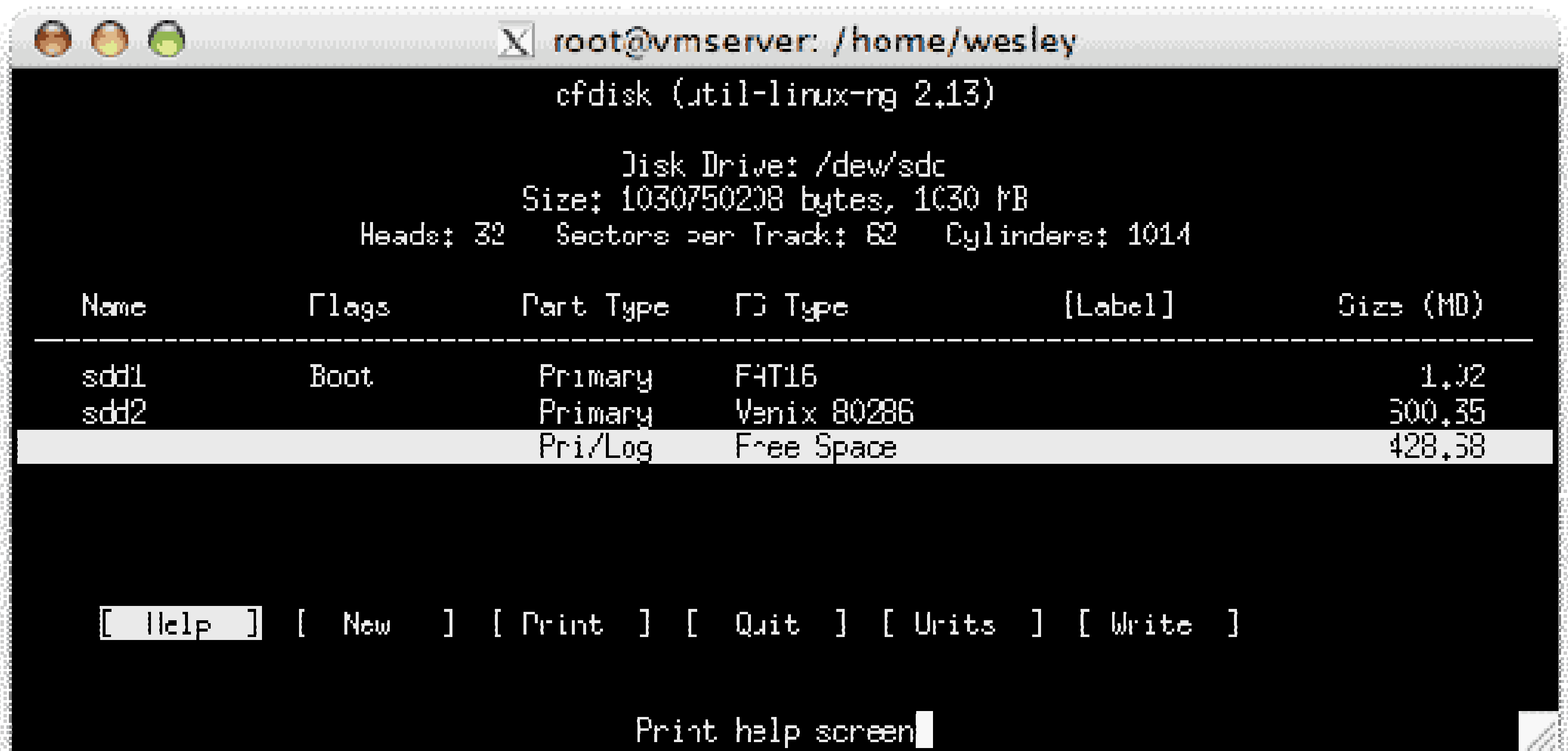
- Knopix
  http://www.knopix.org

Cold Booting - Characteristics

- **Access to main memory**
  software and/or hardware
- **Time of installation**
  post incident
- **Required privileges**
  none
- **Impact on system**
  high
- **Atomicity of image**
  high
- **Image file format**
  raw

Cold Booting – msrramdump

- Bootable external media – USB Hard drive



Cold Booting – msrramdump

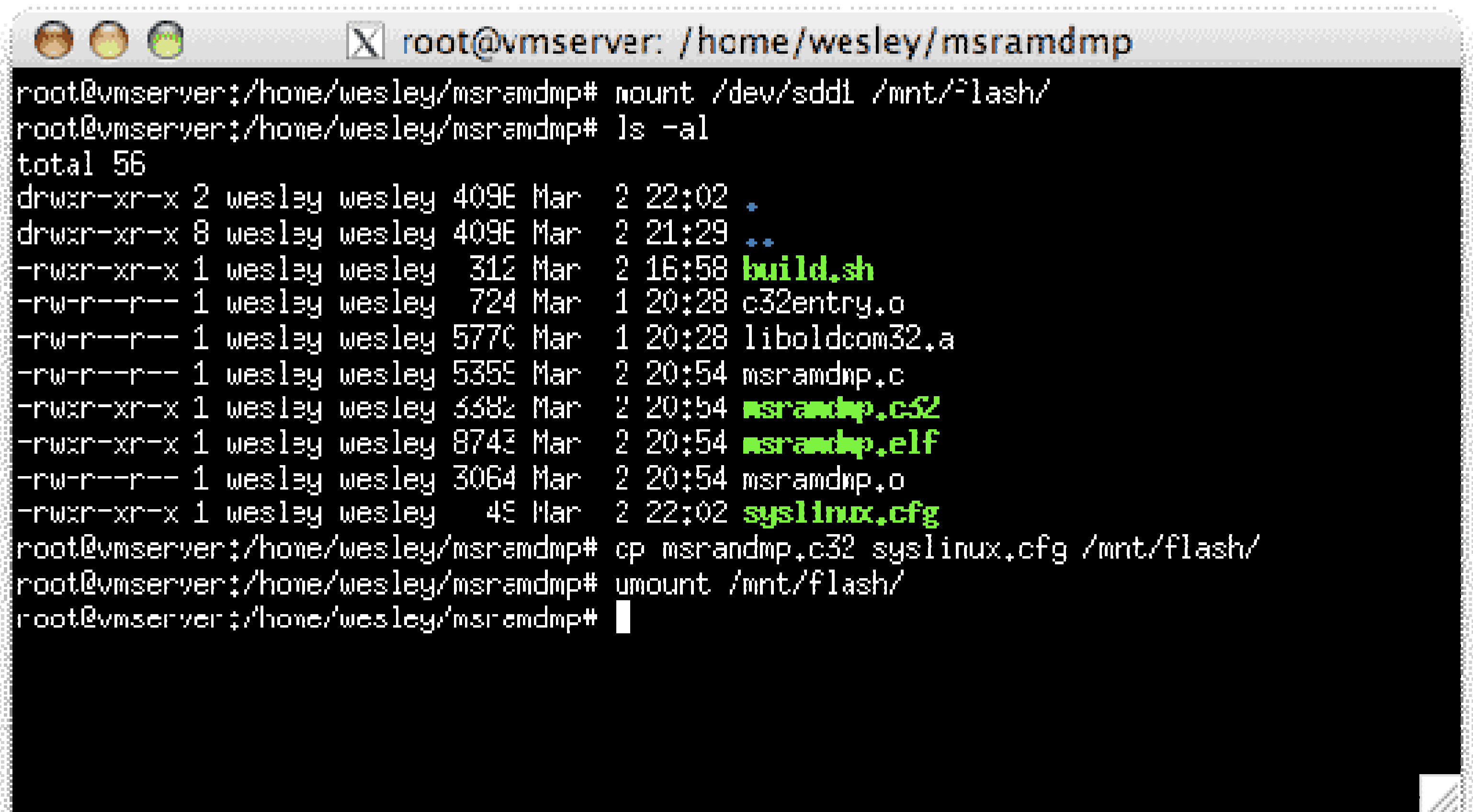
- Copy the MBR to the disk and install Syslinux on the disk

root@vmserver: /home/wesley/syslinux-3.61/mbr

```
root@vmserver:/home/wesley/syslinux-3.61/mbr# ls -al mbr.bin
-rwxr-xr-x 1 wesley wesley 404 Feb 29 14:19 mbr.bin
root@vmserver:/home/wesley/syslinux-3.61/mbr# dd if=mbr.bin of=/dev/sdd
0+1 records in
0+1 records out
404 bytes (404 B) copied, 0.216795 seconds, 1.9 kB/s
root@vmserver:/home/wesley/syslinux-3.61/mbr#
```

Cold Booting – msrramdump

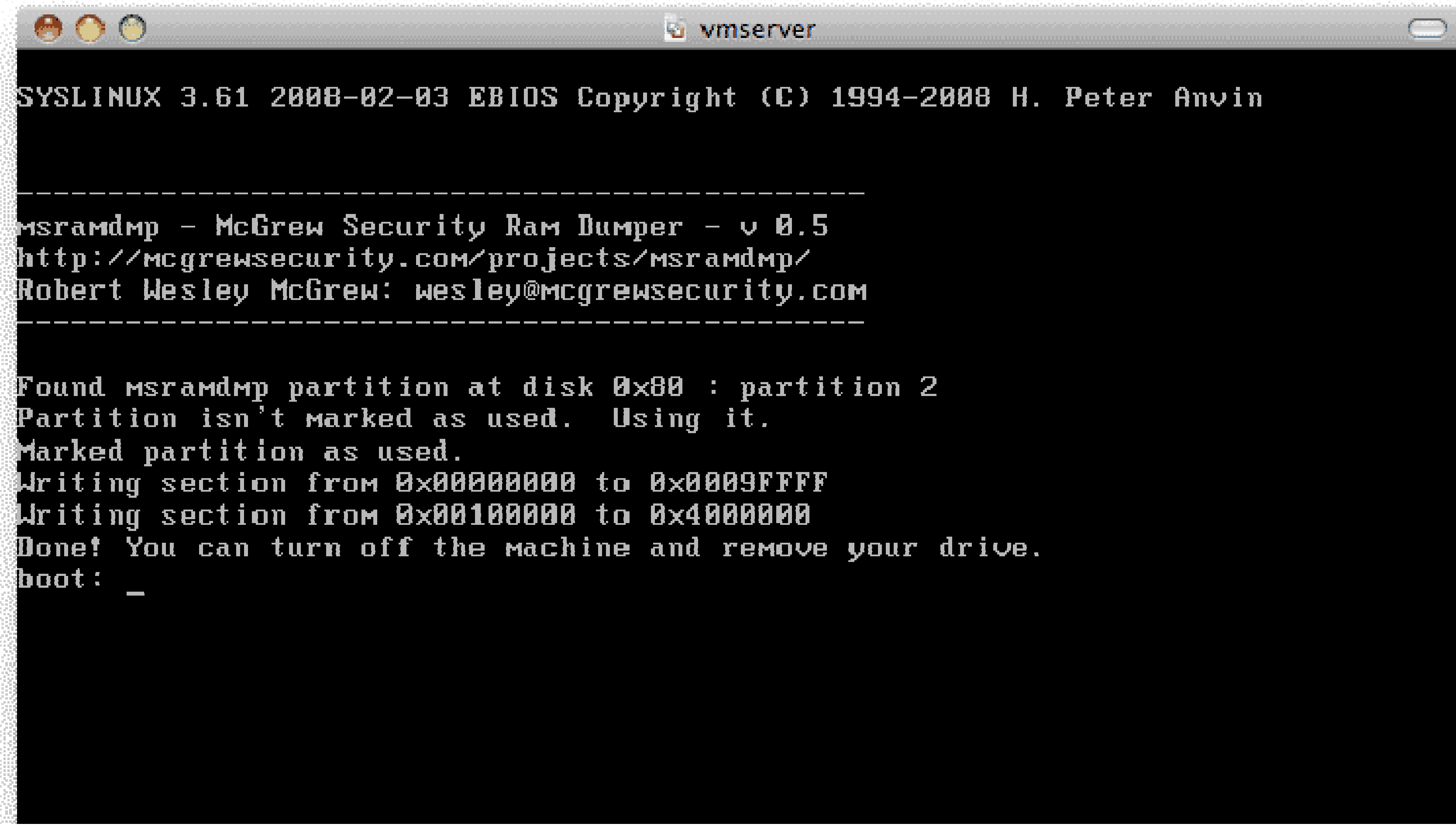
- Mount the FAT partition and copy all the necessary programs to the disk



Data Collection
Main Memory

Cold Booting – msrramdump

- Cut the power, start-up on the USB-disk and start to dump the memory

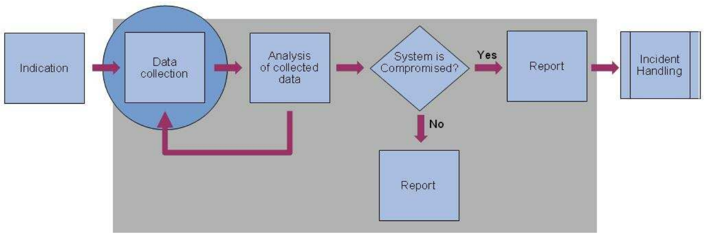


Conclusion FIRST2008

- You can’t trust the kernel of a compromised system
  - ➔ Coldboot the system and then acquire the memory
- You can’t trust the hardware of a compromised system
  - ➔ Transport the memory to a trusted hardware and dump the memory from that system

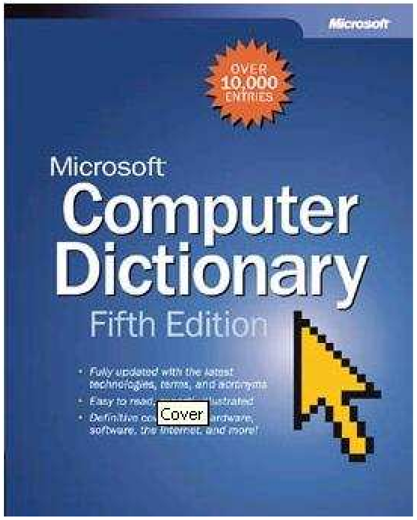
Data Collection

Paged Memory



Pagefile.sys

- **paging file** n. A hidden file on the hard disk that operating systems (such as Windows, Mac OS X, and UNIX) use to hold parts of programs and data files that do not fit in memory. The paging file and physical memory, or RAM, make up virtual memory. Data is moved from the paging file to memory as needed and moved from memory to the paging file to make room for new data in memory. Also called: swap file.

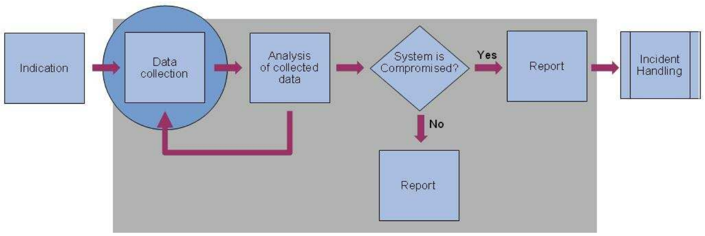


Pagefile.sys

- Located in the root directory if configured for that partition
- Can not be copied using standard methods

Excursus

Bypassing Windows File Protection



Raw Device

Windows Driver Kit: Glossary

- **raw device**
  - A device running in *raw mode*.
- **raw mode**
  - The mode of operation in which a device's driver stack does not include a function driver. A device running in raw mode is being controlled primarily by the bus driver. Upper-level, lower-level, and/or bus filter drivers might be included in the driver stack. If a bus driver can control a device in raw mode, it sets **RawDeviceOK** in the DEVICE_CAPABILITIES structure.

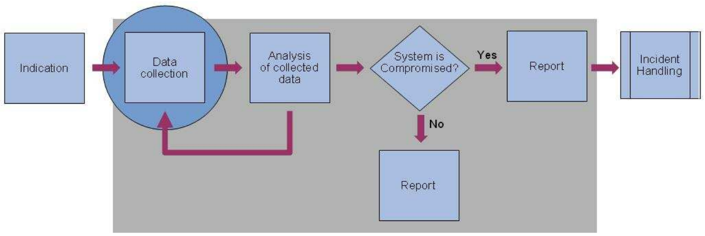
Method nr 1

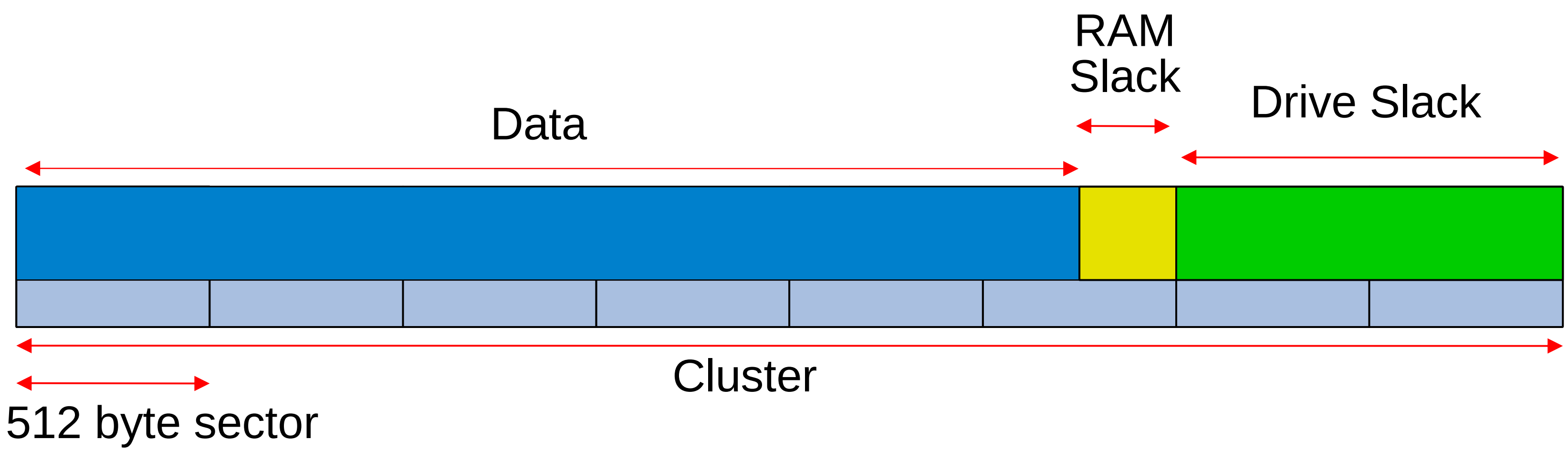
- List all the allocated clusters and write them to STDOUT using raw disk access
- Tools to use:
  - nfi.exe
    from “Windows NT 4.0 and Windows 2000 OEM Support Tools”
  - dd.exe
    from FAU (Forensic Acquisition Utilities) by George M. Garner Jr.
    GMG Systems, Inc
    http://www.gmgsystemsinc.com/fau/

Excursus

Bypassing Windows File Protection

Demo of Method nr 1




RAM
Slack
Drive Slack
Data
Cluster
512 byte sector

Method nr 2

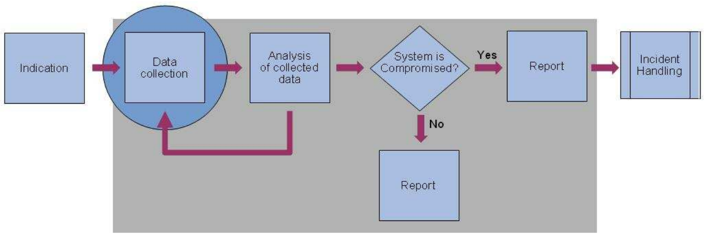
- List the $Mft entry and use that as input to icat.
- Tools to use:
  - ifind and icat
    Both are available from Brian Carrier's Sleuthkit
    http://www.sleuthkit.org/sleuthkit/

    Version 2.03 or earlier compiled with cygwin will work

Excursus

Bypassing Windows File Protection

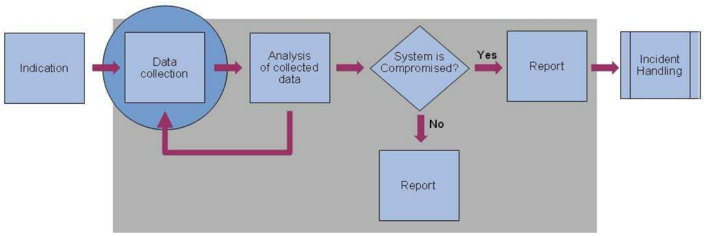
Demo of Method nr 2



Excursus

Bypassing Windows File Protection

Exercise 2



Problems

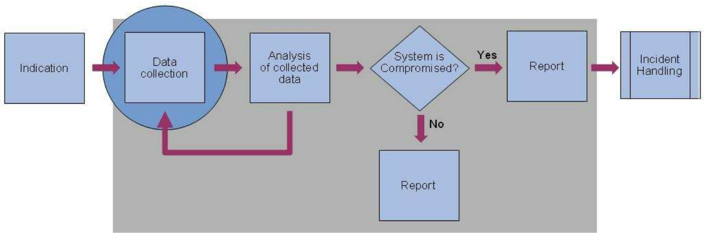
- Disk Encryption
- Compression
- Sparse Files

Anti Forensic techniques

- Hooking RawDevice
- Hooking Low Level functions

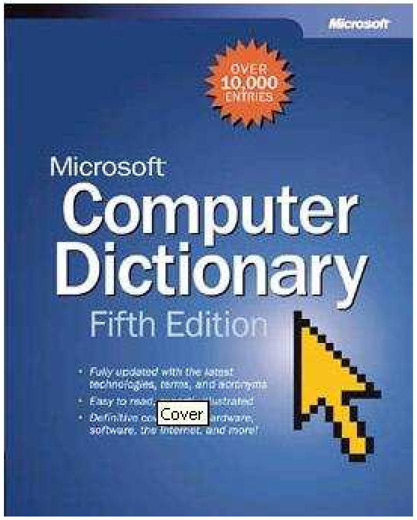
Data Collection

File system meta data



NTFS

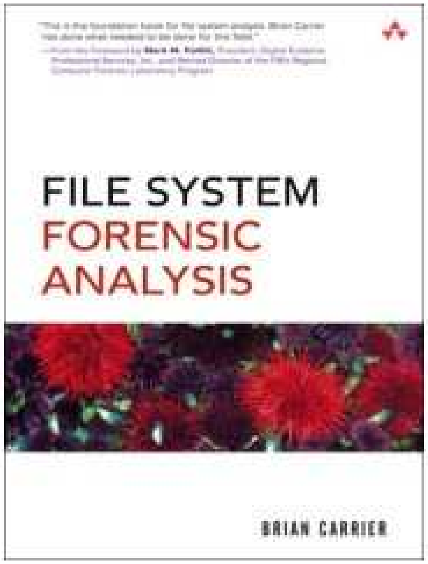
- **NTFS** n. Acronym for NT file system. An advanced file system designed for use specifically with the Windows NT operating system. It supports long filenames, full security access control, file system recovery, extremely large storage media, and various features for the Windows NT POSIX subsystem. It also supports object-oriented applications by treating all files as objects with user-defined and system-defined attributes.



NTFS

- Everything is a File

  One of the most important concepts in understanding the design of NTFS is that important data are allocated to files. This includes the basic file system administrative data that are typically hidden by other file systems. In fact, the files that contain the administrative data can be located anywhere in the volume, like a normal file can. Therefore, an NTFS file system does not have a specific layout like other file systems do. The entire file system is considered a data area, and any sector can be allocated to a file. The only consistent layout is that the first sectors of the volume contain the boot sector and boot code.



NTFS

- Everything is a File
  - $Mft
  - $MftMirr
  - $LogFile
  - $Volume
  - $AttrDef
  - $BitMap
  - $Boot
  - $BadClus
  - $Secure
  - $Upcase
  - $Extend

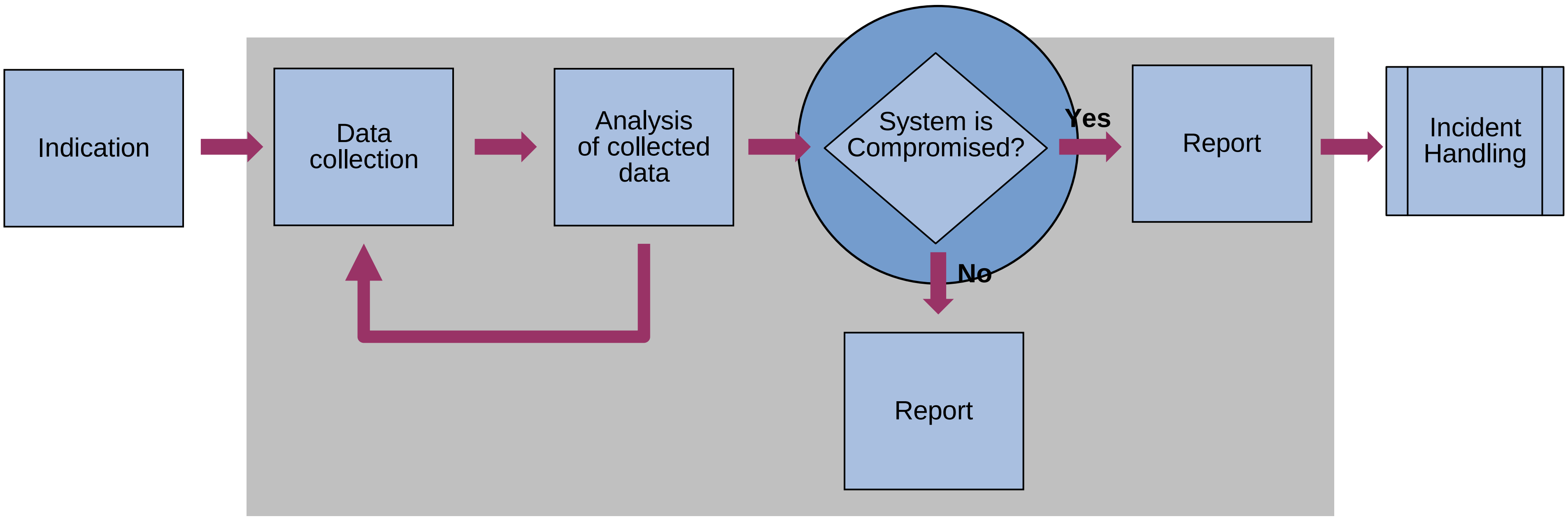
What is not acquired when collecting only Meta Data

- The actual content of the file
- Slack Space
  - Drive Slack
  - Volume Slack
  - File System Slack
- DCO (Device Configuration Overlay)
- HPA (Host Protected Area)

What is gained by only collecting Meta Data

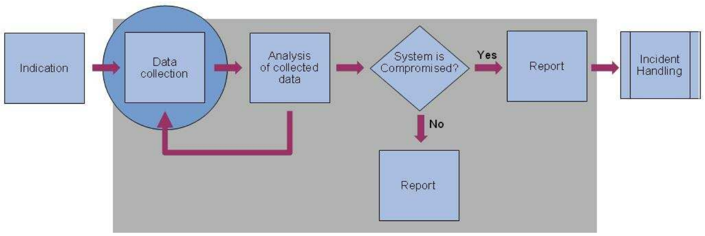
- Speed
  - Acquisition of data takes less time
  - Analyzing the data is also less time consuming
- Remember that we do not know if the system has been compromised at this point

Data Collection
File system meta data



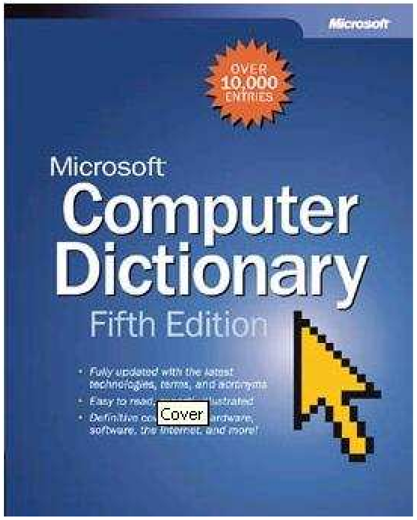
Data Collection

Windows Registry



Windows Registry

- **registry** n. A central hierarchical database in Windows 9x, Windows CE, Windows NT, and Windows 2000 used to store information necessary to configure the system for one or more users, applications, and hardware devices. The Registry contains information that Windows continually references during operation, such as profiles for each user, the applications installed on the computer and the types of documents each can create, property sheet settings for folders and application icons, what hardware exists on the system, and which ports are being used. The Registry replaces most of the text-based .ini files used in Windows 3. x and MS-DOS configuration files, such as AUTOEXEC.BAT and CONFIG.SYS. Although the Registry is common to the several Windows platforms, there are some differences among them. Also called: system registry.



Windows Registry - Logical Structure

- HKEY_CLASSES_ROOT
- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_CONFIG

Windows Registry - Real Structure

- HKEY_CURRENT_USER
- HKEY_LOCAL_MACHINE

Windows Registry - System Wide Hives on Disk

- HKEY_LOCAL_MACHINE\SYSTEM
- HKEY_LOCAL_MACHINE\SAM
- HKEY_LOCAL_MACHINE\SECURITY
- HKEY_LOCAL_MACHINE\SOFTWARE
- HKEY_USERS\.DEFAULT
    - ➔ All located under %windir%\System32\Config\

Windows Registry - System Wide Hives in Memory

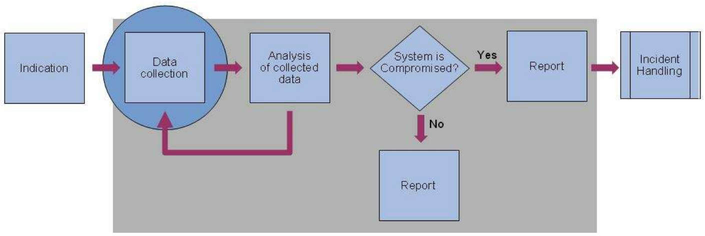
- HKEY_LOCAL_MACHINE\HARDWARE
- HKEY_LOCAL_MACHINE\SYSTEM\Clone (Windows 2000)

Per User Hives on Disk

- HKEY_USERS\<SID>
  - Located under %USERPROFILE%\Ntuser.dat
- HKEY_USERS\<SID>_Classes
  - Located under %USERPROFILE%\Local Settings\Application Data\Microsoft\Windows\Usrclass.dat

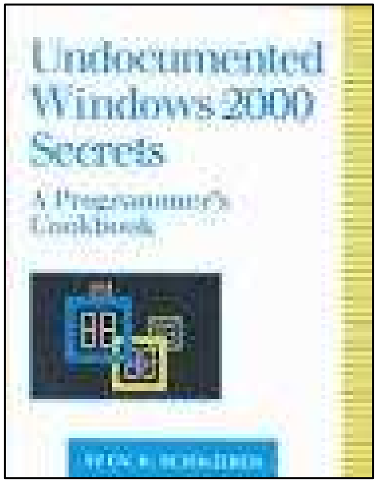
Data Collection

Windows Internal Objects



Windows Internal Objects (1)

- "There is hardly anything more fascinating in the internals of Windows 2000 than the world of its objects. If the memory space of an operating system is viewed as the surface of a planet, the objects are the creatures living on it. Several types of objects exist – small and large ones, simple and complex ones – and they interact in various ways."



Windows Internal Objects (2)

- **Process**
Environment of a loaded binary.

- **Thread**
Execution of a loaded binary.

- **Section**

- **File**
Instance of an opened file or device.

- **Access token**
Access privileges of a process or thread.

- **Key**
Pointer into the Windows registry.

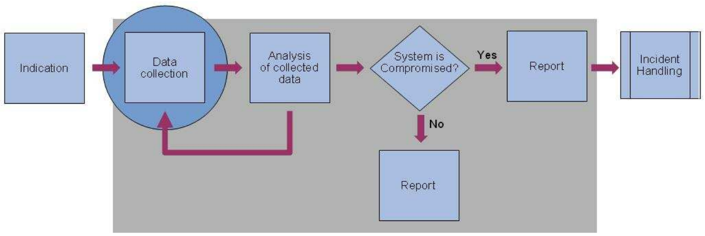
- **Driver**
Extends the kernel.

- **Device**

- **Symbolic link**
makes objects accessible under a new identifier

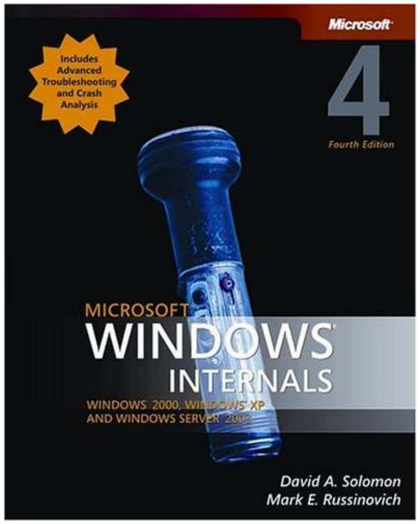
Data Collection

Process Information



Processes (1)

- **process** The virtual address space and control information necessary for the execution of a set of thread objects.

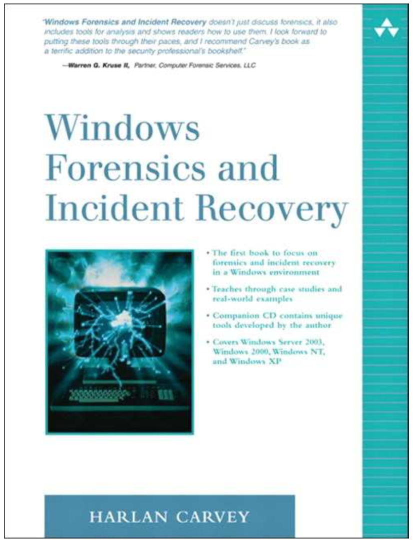


Data Collection

Collecting Process Information

Processes (2)

- What its executable image is
- What command line was used to initiate it
- How long the process has been running
- The security context that it runs in
- Which modules or libraries (DLLs) it accesses
- What memory the process uses



Processes - Tools to use (1)

- tlist.exe (Debugging Tools for Windows)
  http://www.microsoft.com/whdc/devtools/debugging/default.mspx
- pslist.exe (Sysinternals)
  http://www.microsoft.com/technet/sysinternals/utilities/pslist.mspx
  - ➔ Memory & Thread information
- tasklist.exe (WinXP & Win2003 Native)
  - ➔ Security Context

Processes - Tools to use (2)

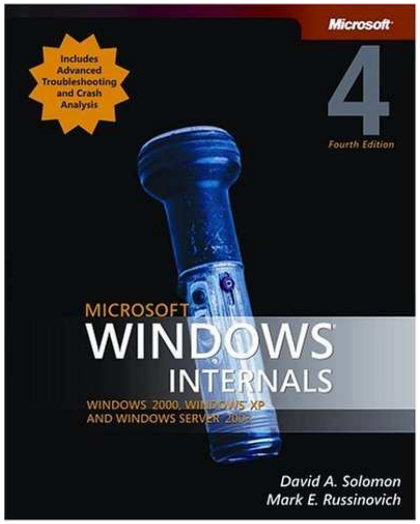
- cmdline.exe (DiamondCS)
  http://www.diamondcs.com.au/index.php?page=console-cmdline
  - Full path to the executable
  - Full Command line for the process
- pulist.exe (Win2000 Resource Kit)
  - Security Context
- Add all Running processes to the list of files to collect

Processes - Tools to dump Process Memory

- Tools to use:
  - Userdump - Microsoft OEM Support Tools
    http://support.microsoft.com/kb/253066
  - X-Ways Capture
    http://www.x-ways.net/capture/
  - Process Dumper by Tobias Klein
    http://www.trapkit.de/research/forensic/pd/index.html

Services

- **Server Processes** User processes that are Windows services, such as the Event Log and Schedule services. Many add-on server applications, such as Microsoft SQL Server and Microsoft Exchange Server, also include components that run as Windows services.

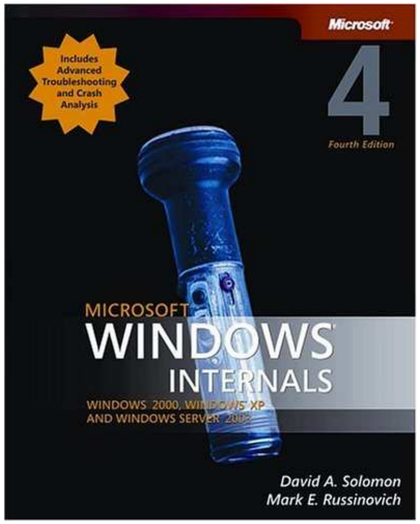


Services

- Tools to use:
  - GSD (Get Service Dacl) (Arne Vidström)
    http://ntsecurity.nu/toolbox/gsd/
  - tasklist.exe
    Native in Windows XP and above

DLL files

- **Dynamic-link library (DLL)** A set of callable subroutines linked as a binary image that can be dynamically loaded by applications that use them.

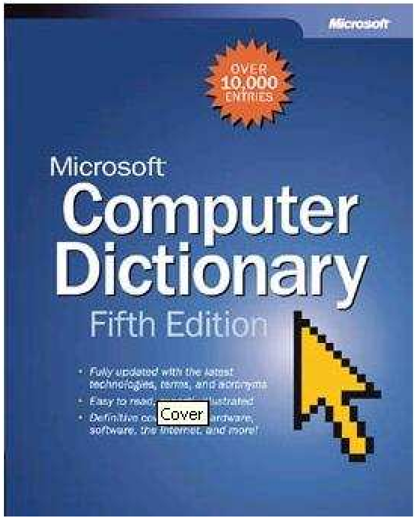


DLL files

- Tools to use:
  - tlist.exe (Debugging Tools for Windows)
    http://www.microsoft.com/whdc/devtools/debugging/
  - listmodules.exe by Arne Vidström
    http://ntsecurity.nu/toolbox/listmodules/
  - listdll.exe (Sysinternals)
    http://www.microsoft.com/technet/sysinternals/utilities/listdlls.mspx
    - Full path to DLL
    - Changes 'LastAccessed'
- Add all DLL files to the list of files to collect

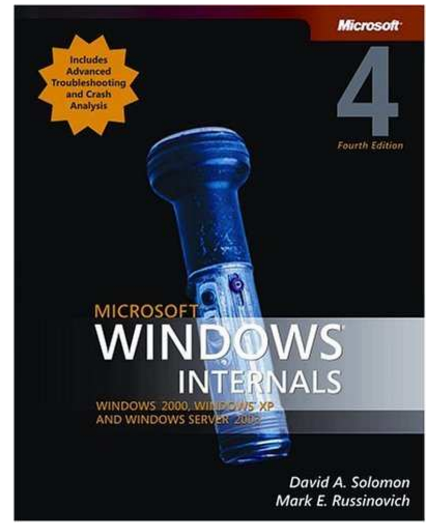
Handle (1)

- **handle** n. Any token that a program can use to identify and access an object such as a device, a file, a window, or a dialog box.



Handle (2)

- Only the executive and drivers are allowed to directly access kernel objects. Processes in user-mode have to acquire a handle prior to any operation on an object.



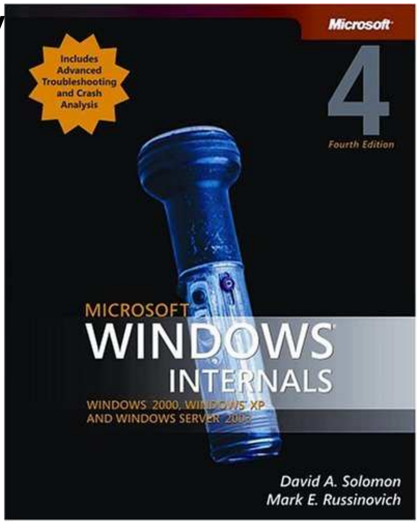
Handle - Tools to use

- handle.exe (Sysinternals)
  http://www.microsoft.com/technet/sysinternals/utilities/Handle.mspx

- Add all files with open handles to the list of files to collect

Device Drivers

- **“Device Drivers** Loadable kernel-mode modules (typically ending in .sys) that interface between the I/O system and the relevant hardware. Device drivers on Windows don't manipulate hardware devices directly, but rather they call parts of the hardware application layer (HAL) to interface with the hardware.”

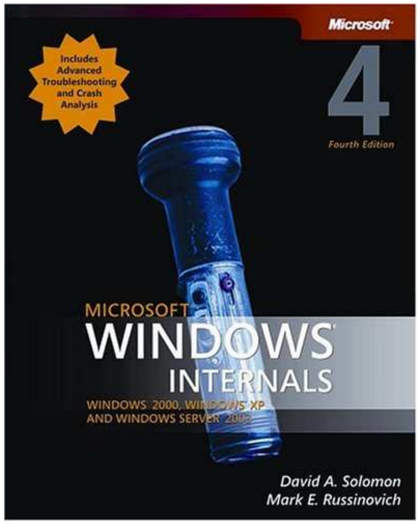


Device Drivers

- Tools to use:
  - listdrivers.exe (Arne Vidström)
    http://ntsecurity.nu/toolbox/listdrivers/
  - Device Console from DDK - Windows Driver Development Kit
    http://www.microsoft.com/whdc/devtools/ddk/default.mspx
  - ListObj (Arne Vidström) - prints the entire Windows object space
    http://vidstrom.net/otools/listobj/
- Add all Device Drivers to the list of files to collect

Device Objects

- **“Device Objects** A data structure that represents a physical, logical, or virtual device on the system and describes its characteristics, such as the alignment it requires for buffers and the location of its device queue to hold incoming I/O request packets.”

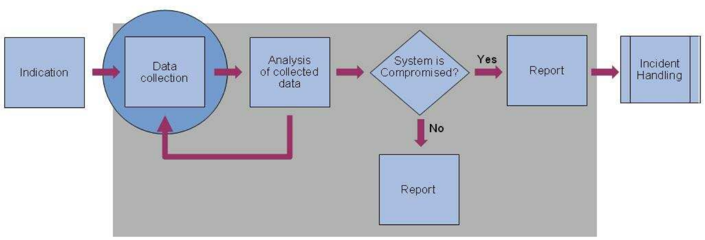


Device Objects

- Tools to use:
  - ➔ lListObj (Arne Vidström) - prints the entire Windows object space
    http://vidstrom.net/otools/listobj/
- Add all Device Objects to the list of files to collect

Data Collection

Network Information

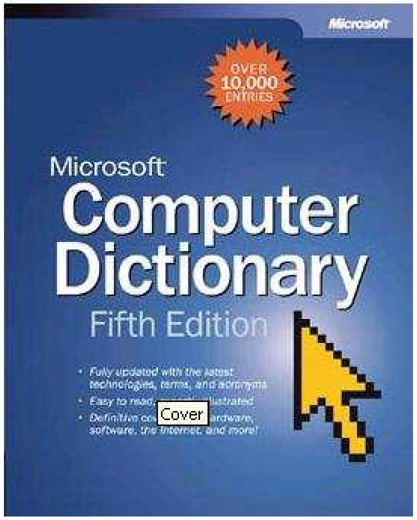


Network Interface Cards

- Tools to use:
  - ➔ ipconfig.exe (Native System Command)

ARP

- **ARP** n. Acronym for Address Resolution Protocol. A TCP/IP protocol for determining the hardware address (or physical address) of a node on a local area network connected to the Internet, when only the IP address (or logical address) is known. An ARP request is sent to the network, and the node that has the IP address responds with its hardware address. Although ARP technically refers only to finding the hardware address, and RARP (for Reverse ARP) refers to the reverse procedure, ARP is commonly used for both senses.

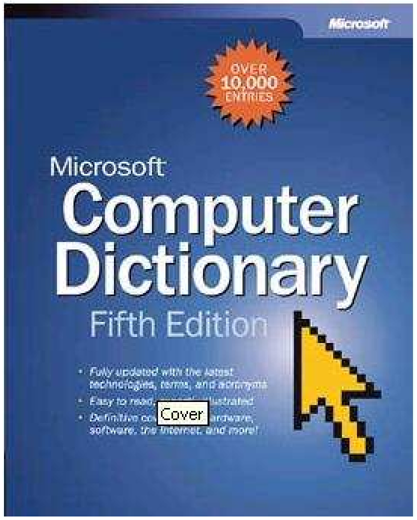


ARP

- Tools to use:
  - ➔ arp.exe (Native System Command)

Active Network Connections

- **socket** n. 1. An identifier for a particular service on a particular node on a network. The socket consists of a node address and a port number, which identifies the service. For example, port 80 on an Internet node indicates a Web server.



Active Network Connections

- Tools to use:
  - netstat.exe (Native system command)
  - fport.exe (Foundstone)
    http://www.foundstone.com/resources/proddesc/fport.htm
  - openports.exe (DiamondCS)
    http://www.diamondcs.com.au/openports/

NetBIOS over TCP/IP

- Tools to use:
  - nbtstat.exe (Native system command)

Files opened remotely

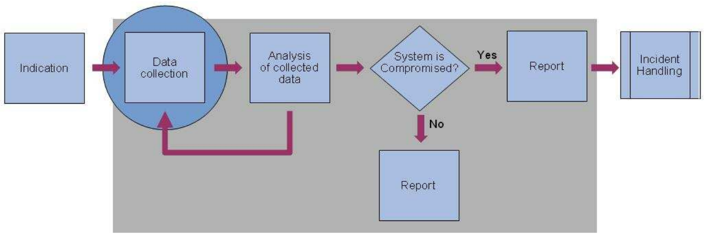
- Tools to use:
  - psfile.exe (Sysinternals)
    http://www.microsoft.com/technet/sysinternals/Networking/PsFile.mspx
  - net.exe (Native System Command)

Logged on remote users

- Tools to use:
  - psloggedon.exe (Sysinternals)
    http://www.microsoft.com/technet/sysinternals/Networking/PsFile.mspx
  - net.exe sessions (Native System Command)
  - netusers.exe (Sommarsoft)
  - loggonsessions.exe (Sysinternals)

Data Collection

Non Volatile data



System information

- Tools to use:
  - psinfo.exe (Sysinternals) http://www.microsoft.com/technet/sysinternals/SystemInformation/PsInfo.mspx
    - Installed Applications & Hotfixes
  - systeminfo.exe (Windows Native)
  - psservice.exe (Sysinternals)
  - cpuid.exe (Arne Vidström)

NTFS

- Directory Listing from Usermode
- Tools to use
  - dir (built-in command)
  - find.exe (windows port of unix command)

Windows registry

- Listing from Usermode
- Tools to use
  - ➔ reg.exe (Resource Kit)
  - ➔ accesschk.exe (sysinternals)
  - ➔ subinacl.exe
  - ➔ regdump.exe (Win2K Resource Kit)
  - ➔ Regtool (Cygwin)

Log files - OS specific

- System, Application and Security
  - ➔ Located under %windir%\System32\Config\

Log files - Per application

- Exchange, IIS, Apache,
  - ➔ Location highly dependent of application

Interesting files (1)

- Everything running
  - Processes
  - Drivers
  - DLLs

Interesting files (2)

- Everything being started
  - Autorunsc (Sysinternals)
    http://www.sysinternals.com

Interesting files (3)

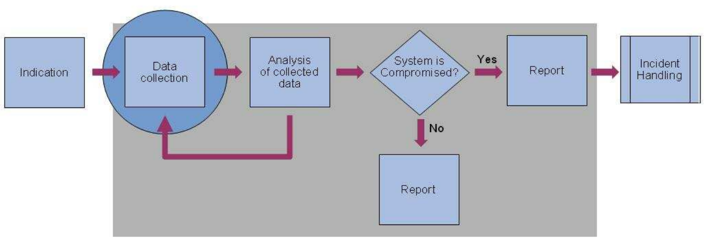
- Per user (%USERPROFILE%)
  - NTUSER.DAT
  - Application Data
  - Cookies
  - Recent

Checksums of files

- Algorithms
  - MD5
  - SHA-1
- Tools change 'LastAccessed'

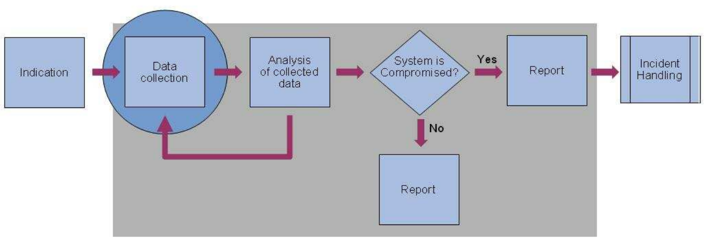
Data Collection

Putting it all together

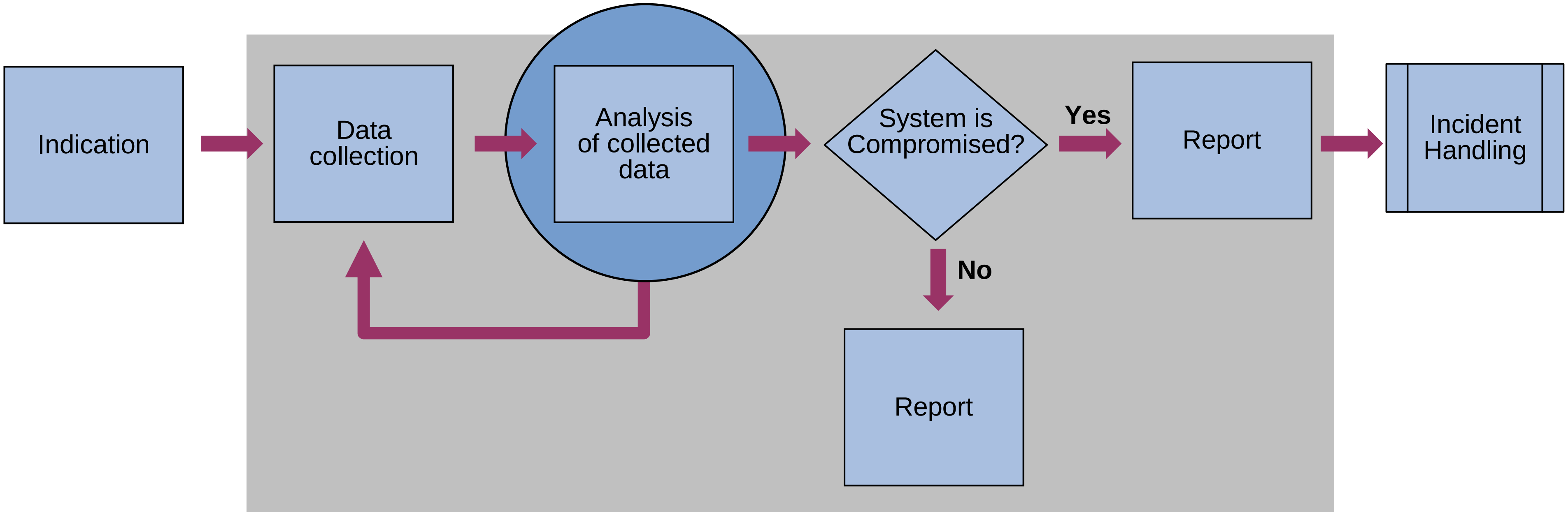


Data Collection

Question and Answers

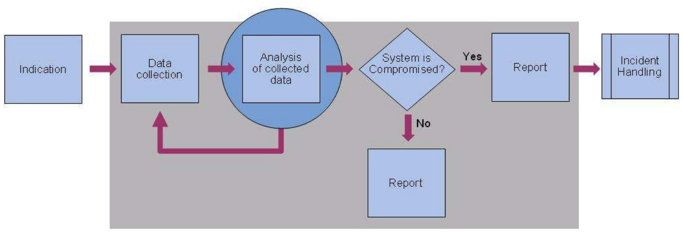


Incident Flowchart



Data Analysis

Analysis Method



Windows vs. Linux as the choice of the analysis platform (1)

- Script based tools (perl, python)
  - Works in general just as fine on Windows as on Linux
  - Win32 perl modules works natively on Windows
- Analyzing Crash Dumps
  - Windows Debugger works only on the Windows platform

Windows vs. Linux as the choice of the analysis platform (2)

- Conclusion
  - ➔ Use the Windows platform when analyzing a suspected Windows intrusions!

Analysis Methodology - What are we looking for

- Malware that do not try to hide itself
  - No rootkit technology being used
- Malware that try to hide itself
  - The malware is using rootkit technology to hide its presence
- Traces of system activity in order to build a timeline of the incident

Analysis Methodology - Malware that do not try to hide itself (1)

- Log files
  - Signs of intrusions
- NTFS Meta data
  - Known suspicious file names
  - Files that the Local Administrator do not have access to
  - Files added at the suspected time of the intrusion

Analysis Methodology - Malware that do not try to hide itself (2)

- Windows Registry
  - Known registry keys used by Malware
  - Registry keys added at the suspected time of the intrusion

Analysis Methodology - Malware that do not try to hide itself (3)

- Files Collected during the data acquisition
  - Known checksums
  - Static Analysis
- Network Information
  - Listening ports
  - Established connections

Analysis Methodology - Malware that try to hide itself (1)

- NTFS Meta data
  - Files hidden from user mode
- Windows Registry
  - Keys hidden from user mode

Analysis Methodology - Malware that try to hide itself (2)

- Memory Dump
  - Objects hidden from user mode
  - Inspection of system tables
  - Integrity checking of binaries loaded in memory

Analysis Methodology - Traces of system activity (1)

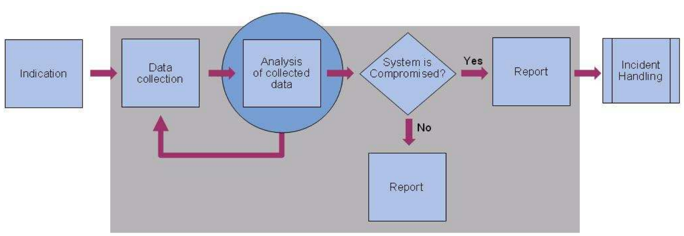
- NTFS Meta data
  - Added files
  - Changed files
  - Deleted files
- Collected files
  - INFO2 Records (Recycle Bin)
  - Cookies

Analysis Methodology - Traces of system activity (2)

- Memory dump
  - Processes and Threads
  - Network Activity

Data Analysis

Log Files



Windows Event Logs (1)

- Data of interest
  - Signs of intrusions
  - Time stamps to add to our time line analysis
- Online resources
  - Loganalysis.org
  - Event ID mapping
    - EventID.Net
    - Microsoft Events and Errors Message Center

Windows Event Logs (2)

- Format of the log file

Windows Event Logs (1)

- Tools to use
  - GrokEVT by Timothy Morgan (Sentinel Chicken Networks)
    http://projects.sentinelchicken.org/grokevt/
  - FCCU evtreader.pl (d-fence.be)
    http://www.d-fence.be/loadcd?target=fccu.evtreader.1.1.tar.gz

Windows Vista – Event logging

- XML Schema

Other types of text based Log Files (1)

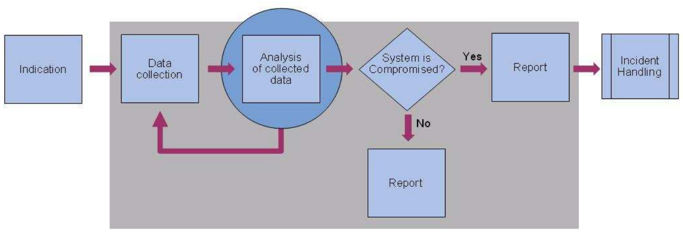
- Data of interest
  - Signs of intrusions
  - Time stamps to add to our time line analysis
- Firewall log
  - MS Firewall

Other types of text based Log Files (2)

- Tools to use
    - grep, sed, perl
    - PyFLAG

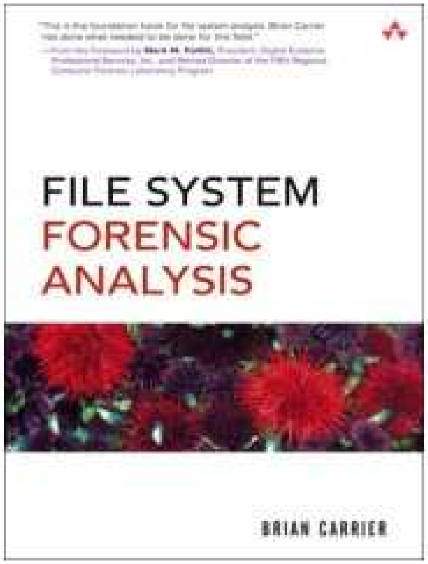
Data Analysis

NTFS Meta Data



$MFT

- **MFT Concepts:** The Master File Table (MFT) is the heart of NTFS because it contains the information about all files and directories. Every file and directory has at least one entry in the table, and the entries by themselves are very simple. They are 1 KB in size, but only the first 42 bytes have a defined purpose. The remaining bytes store attributes, which are small data structures that have a very specific purpose. For example, one attribute is used to store the file's name, and another is used to store the file's content.



$MFT- Data of interest

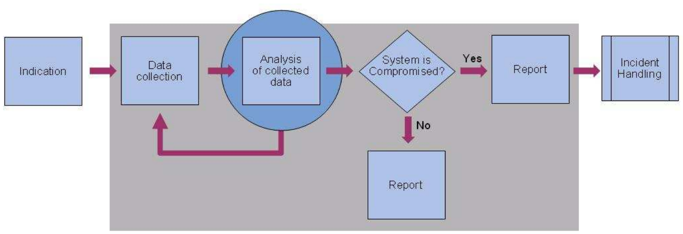
- Time stamps to add to our time line analysis
  - Modified
  - Accesed
  - Created
  - Entry updated
- Known suspicious filenames
- Deviation between directory listening and $MFT (cross-view diff)

$MFT- Tools to use

- No publicly available tools to do our analysis with!

Excursus

Analysis of a $MFT entry



Excursus

Analysis of $MFT entry

```
0000: 4649 4c45 3000 0300 37ec 2517 0000 0000  FILE0...7.%.....
0010: 0300 0100 3800 0100 7001 0000 0004 0000  ....8...p.......
0020: 0000 0000 0000 0000 0400 0000 7c33 0000  ............|3..
0030: 0600 0000 0000 0000 1000 0000 6000 0000  ............`...
0040: 0000 0000 0000 0000 4800 0000 1800 0000  ........H.......
0050: 70ec 468d 43cf c601 a053 909c 43cf c601  p.F.C....S..C...
0060: a053 909c 43cf c601 a053 909c 43cf c601  .S..C....S..C...
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 0000 0000 8301 0000 0000 0000 0000 0000  ................
0090: 0000 0000 0000 0000 3000 0000 7000 0000  ........0...p...
00a0: 0000 0000 0000 0200 5200 0000 1800 0100  ........R.......
00b0: 7b33 0000 0000 0300 70ec 468d 43cf c601  {3......p.F.C...
00c0: 70ec 468d 43cf c601 70ec 468d 43cf c601  p.F.C...p.F.C...
00d0: 70ec 468d 43cf c601 0000 0000 0000 0000  p.F.C...........
00e0: 0000 0000 0000 0000 2000 0000 0000 0000  ........ .......
00f0: 0803 7400 6500 7300 7400 2e00 7400 7800  ..t.e.s.t...t.x.
0100: 7400 0000 0000 0000 8000 0000 2800 0000  t...........(...
0110: 0000 1800 0000 0100 0e00 0000 1800 0000  ................
0120: 7468 6973 2069 7320 6120 7465 7374 0000  this is a test..
0130: 8000 0000 3800 0000 0007 1800 0000 0300  ....8...........
0140: 0d00 0000 2800 0000 6100 6400 7300 2e00  ....(...a.d.s...
0150: 7400 7800 7400 0000 7468 6973 2069 7320  t.x.t...this is
0160: 6120 6164 7300 0000 ffff ffff 8279 4711  a ads........yG.
```

Offset to first attribute

Attribute Type Identifier

0x10 $STANDARD_INFORMATION

Length of Attribute

Excursus

Analysis of $MFT entry

```
0000: 4649 4c45 3000 0300 37ec 2517 0000 0000  FILE0...7.%.....
0010: 0300 0100 3800 0100 7001 0000 0004 0000  ....8...p.......
0020: 0000 0000 0000 0000 0400 0000 7c33 0000  ............|3..
0030: 0600 0000 0000 0000 1000 0000 6000 0000  ............`...
0040: 0000 0000 0000 0000 4800 0000 1800 0000  ........H.......
0050: 70ec 468d 43cf c601 a053 909c 43cf c601  p.F.C....S..C...
0060: a053 909c 43cf c601 a053 909c 43cf c601  .S..C....S..C...
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 0000 0000 8301 0000 0000 0000 0000 0000  ................
0090: 0000 0000 0000 0000 3000 0000 7000 0000  ........0...p...
00a0: 0000 0000 0000 0200 5200 0000 1800 0100  ........R.......
00b0: 7b33 0000 0000 0300 70ec 468d 43cf c601  {3......p.F.C...
00c0: 70ec 468d 43cf c601 70ec 468d 43cf c601  p.F.C...p.F.C...
00d0: 70ec 468d 43cf c601 0000 0000 0000 0000  p.F.C...........
00e0: 0000 0000 0000 0000 2000 0000 0000 0000  ........ .......
00f0: 0803 7400 6500 7300 7400 2e00 7400 7800  ..t.e.s.t...t.x.
0100: 7400 0000 0000 0000 8000 0000 2800 0000  t...........(...
0110: 0000 1800 0000 0100 0e00 0000 1800 0000  ................
0120: 7468 6973 2069 7320 6120 7465 7374 0000  this is a test..
0130: 8000 0000 3800 0000 0007 1800 0000 0300  ....8...........
0140: 0d00 0000 2800 0000 6100 6400 7300 2e00  ....(...a.d.s...
0150: 7400 7800 7400 0000 7468 6973 2069 7320  t.x.t...this is
0160: 6120 6164 7300 0000 ffff ffff 8279 4711  a ads........yG.
```

$STANDARD_INFORMATION (0x10)

Offset	Size	OS	Description
~	~		Standard Attribute Header
0x00	8		C Time - File Creation
0x08	8		A Time - File Altered
0x10	8		M Time - MFT Changed
0x18	8		R Time - File Read
0x20	4		DOS File Permissions
0x24	4		Maximum Number of Versions
0x28	4		Version Number
0x2C	4		Class Id
0x30	4	2K	Owner Id
0x34	4	2K	Security Id
0x38	8	2K	Quota Charged
0x40	8	2K	Update Sequence Number (USN)

Analysis of $MFT entry

```
0000: 4649 4c45 3000 0300 37ec 2517 0000 0000  FILE0...7.%.....
0010: 0300 0100 3800 0100 7001 0000 0004 0000  ....8...p.......
0020: 0000 0000 0000 0000 0400 0000 7c33 0000  ............|3..
0030: 0600 0000 0000 0000 1000 0000 6000 0000  ............`...
0040: 0000 0000 0000 0000 4800 0000 1800 0000  ........H.......
0050: 70ec 468d 43cf c601 a053 909c 43cf c601  p.F.C....S..C...
0060: a053 909c 43cf c601 a053 909c 43cf c601  .S..C....S..C...
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 0000 0000 8301 0000 0000 0000 0000 0000  ................
0090: 0000 0000 0000 0000 3000 0000 7000 0000  ........0...p...
00a0: 0000 0000 0000 0200 5200 0000 1800 0100  ........R.......
00b0: 7b33 0000 0000 0300 70ec 468d 43cf c601  {3......p.F.C...
00c0: 70ec 468d 43cf c601 70ec 468d 43cf c601  p.F.C...p.F.C...
00d0: 70ec 468d 43cf c601 0000 0000 0000 0000  p.F.C...........
00e0: 0000 0000 0000 0000 2000 0000 0000 0000  ........ .......
00f0: 0803 7400 6500 7300 7400 2e00 7400 7800  ..t.e.s.t...t.x.
0100: 7400 0000 0000 0000 8000 0000 2800 0000  t...........(...
0110: 0000 1800 0000 0100 0e00 0000 1800 0000  ................
0120: 7468 6973 2069 7320 6120 7465 7374 0000  this is a test..
0130: 8000 0000 3800 0000 0007 1800 0000 0300  ....8...........
0140: 0d00 0000 2800 0000 6100 6400 7300 2e00  ....(...a.d.s...
0150: 7400 7800 7400 0000 7468 6973 2069 7320  t.x.t...this is
0160: 6120 6164 7300 0000 ffff ffff 8279 4711  a ads........yG.
```

$FILE_NAME(0x30)

Offset	Size	Description
~	~	Standard Attribute Header
0x00	8	File reference to the parent directory.
0x08	8	C Time - File Creation
0x10	8	A Time - File Altered
0x18	8	M Time - MFT Changed
0x20	8	R Time - File Read
0x28	8	Allocated size of the file
0x30	8	Real size of the file
0x38	4	Flags, e.g. Directory, compressed, hidden
0x3c	4	Used by EAs and Reparse
0x40	1	Filename length in characters (L)
0x41	1	Filename namespace 0x42 2L Filename in Unicode (not null terminated)

Excursus
Analysis of $MFT entry

```
0000: 4649 4c45 3000 0300 37ec 2517 0000 0000  FILE0...7.%.....
0010: 0300 0100 3800 0100 7001 0000 0004 0000  ....8...p.......
0020: 0000 0000 0000 0000 0400 0000 7c33 0000  ............|3..
0030: 0600 0000 0000 0000 1000 0000 6000 0000  ............`...
0040: 0000 0000 0000 0000 4800 0000 1800 0000  ........H.......
0050: 70ec 468d 43cf c601 a053 909c 43cf c601  p.F.C....S..C...
0060: a053 909c 43cf c601 a053 909c 43cf c601  .S..C....S..C...
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 0000 0000 8301 0000 0000 0000 0000 0000  ................
0090: 0000 0000 0000 0000 3000 0000 7000 0000  ........0...p...
00a0: 0000 0000 0000 0200 5200 0000 1800 0100  ........R.......
00b0: 7b33 0000 0000 0300 70ec 468d 43cf c601  {3......p.F.C...
00c0: 70ec 468d 43cf c601 70ec 468d 43cf c601  p.F.C...p.F.C...
00d0: 70ec 468d 43cf c601 0000 0000 0000 0000  p.F.C...........
00e0: 0000 0000 0000 0000 2000 0000 0000 0000  ........ .......
00f0: 0803 7400 6500 7300 7400 2e00 7400 7800  ..t.e.s.t...t.x.
0100: 7400 0000 0000 0000 8000 0000 2800 0000  t...........(...
0110: 0000 1800 0000 0100 0e00 0000 1800 0000  ................
0120: 7468 6973 2069 7320 6120 7465 7374 0000  this is a test..
0130: 8000 0000 3800 0000 0007 1800 0000 0300  ....8...........
0140: 0d00 0000 2800 0000 6100 6400 7300 2e00  ....(...a.d.s...
0150: 7400 7800 7400 0000 7468 6973 2069 7320  t.x.t...this is
0160: 6120 6164 7300 0000 ffff ffff 8279 4711  a ads........yG.
```

$DATA(0x80)

Offset	Size	Description
~	~	Standard Attribute Header
0x00		Any data

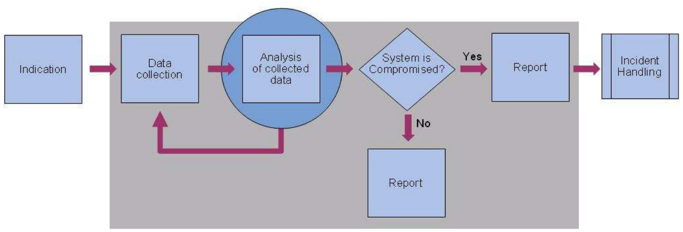
```
0000: 4649 4c45 3000 0300 37ec 2517 0000 0000  FILE0...7.%.....
0010: 0300 0100 3800 0100 7001 0000 0004 0000  ....8...p.......
0020: 0000 0000 0000 0000 0400 0000 7c33 0000  ............|3..
0030: 0600 0000 0000 0000 1000 0000 6000 0000  ............`...
0040: 0000 0000 0000 0000 4800 0000 1800 0000  ........H.......
0050: 70ec 468d 43cf c601 a053 909c 43cf c601  p.F.C....S..C...
0060: a053 909c 43cf c601 a053 909c 43cf c601  .S..C....S..C...
0070: 2000 0000 0000 0000 0000 0000 0000 0000   ...............
0080: 0000 0000 8301 0000 0000 0000 0000 0000  ................
0090: 0000 0000 0000 0000 3000 0000 7000 0000  ........0...p...
00a0: 0000 0000 0000 0200 5200 0000 1800 0100  ........R.......
00b0: 7b33 0000 0000 0300 70ec 468d 43cf c601  {3......p.F.C...
00c0: 70ec 468d 43cf c601 70ec 468d 43cf c601  p.F.C...p.F.C...
00d0: 70ec 468d 43cf c601 0000 0000 0000 0000  p.F.C...........
00e0: 0000 0000 0000 0000 2000 0000 0000 0000  ........ .......
00f0: 0803 7400 6500 7300 7400 2e00 7400 7800  ..t.e.s.t...t.x.
0100: 7400 0000 0000 0000 8000 0000 2800 0000  t...........(...
0110: 0000 1800 0000 0100 0e00 0000 1800 0000  ................
0120: 7468 6973 2069 7320 6120 7465 7374 0000  this is a test..
0130: 8000 0000 3800 0000 0007 1800 0000 0300  ....8...........
0140: 0d00 0000 2800 0000 6100 6400 7300 2e00  ....(...a.d.s...
0150: 7400 7800 7400 0000 7468 6973 2069 7320  t.x.t...this is
0160: 6120 6164 7300 0000 ffff ffff 8279 4711  a ads........yG.
```

$DATA(0x80)

Offset	Size	Description
~	~	Standard Attribute Header
0x00		Any data

Data Analysis

NTFS Meta Data (continued)



$MFT - Cross-view diff

```
C:\WINDOWS\system32\cmd.exe
### files missing from user mode
4       /$AttrDef
8       /$BadClus
6       /$Bitmap
7       /$Boot
11      /$Extend
25      /$Extend/$ObjId
24      /$Extend/$Quota
26      /$Extend/$Reparse
2       /$LogFile
0       /$MFT
1       /$MFTMirr
9       /$Secure
10      /$UpCase
3       /$Volume
22429   /Documents and Settings/All Users/Documents/My Pictures/Sample Pictures/Thumbs.db:encryptable
14548   /Documents and Settings/user/Desktop/Att_fanga_en_DDoS_kiddie.pdf:Zone.Identifier
33234   /Documents and Settings/user/Desktop/SP28809.exe:Zone.Identifier
23253   /Documents and Settings/user/Desktop/SP28849.exe:Zone.Identifier
24595   /Documents and Settings/user/Desktop/verafigueiredo.mov:Zone.Identifier
45327   /Documents and Settings/user/Local Settings/History/History.IE5/MSHist012005112820051205
47806   /Documents and Settings/user/Local Settings/History/History.IE5/MSHist012005120720051208
46122   /Documents and Settings/user/Local Settings/History/History.IE5/MSHist012005121020051211
```

$MFT - Known suspicious file names

- Viruslist.com
- Counter Spy (Sunbelt-Software)
  http://research.sunbelt-software.com/WhatYouShouldKnow.aspx

Threat Details

Copyright © 2007 Sunbelt-Software. Reproduction in whole or in part without permission is prohibited.

Adrenaline Worm

Type	Worm
Level	High
Level Description	High risks are typically installed without user interaction through security exploits, and can severely compromise system security. Such risks may open illicit network connections, use polymorphic tactics to self-mutate, disable security software, modify system files, and install additional malware. These risks may also collect and transmit personally identifiable information (PII) without your consent and severely degrade the performance and stability of your computer.
Advice Type	Remove
Alias	IRC-Worm.Adrenaline, W32/Scrambler
File Traces	cygnus.exe dupripper.07.exe irc-worm.adrenaline.exe littlejo.07.exe

$BadClus

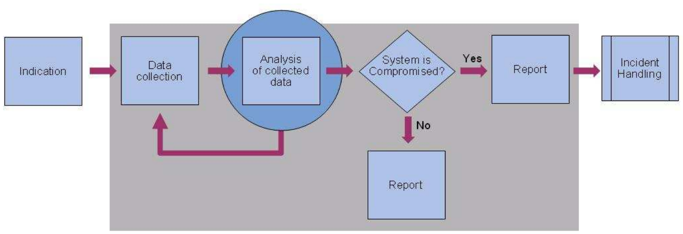
- Used for not letting the OS use clusters marked as bad. Modern hard disks usually handle bad sectors themself.
- Data of interest
  - ➔ $Bad attribute - Check for excessive use

Anti-forensic attacks

- Metasploit Anti-forensics (Vincent Liu and Patrick Stach)
  http://www.metasploit.com/
  - Slacker
  - Timestomp

Data Analysis

Windows Registry



Windows Registry - Data of interest

- Data of interest
  - Time Line
  - Known obfuscation techniques
  - Deviation between user mode listening and raw file - cross-view diff
  - Interesting registry keys

Windows Registry - Tools to use

- reglookup (Sentinel Chicken Networks)
  http://projects.sentinelchicken.org/reglookup/
- Offline Registry Parser by Harlan Carvey
  http://downloads.sourceforge.net/windowsir/regp_1_1.zip
- Parse::Win32Registry

Windows Registry - Obfuscation techniques (1)

- Keys with built-in ”Null” characters

Windows Registry - Obfuscation techniques (2)

- Values that are of 256-259 characters in length

Windows Registry Editor Utility String Concealment Weakness

Secunia Advisory:	SA16560
Release Date:	2005-08-24
Last Update:	2006-02-06
Critical:	Not critical
Impact:	Spoofing
Where:	Local system
Solution Status:	Unpatched
OS:	Microsoft Windows 2000 Advanced Server Microsoft Windows 2000 Datacenter Server Microsoft Windows 2000 Professional Microsoft Windows 2000 Server Microsoft Windows XP Home Edition Microsoft Windows XP Professional

Description:
Igor Franchuk has discovered a weakness in Microsoft Windows, which can be exploited to hide certain information.

The weakness is caused due to an error in the Registry Editor Utility (regedit.exe) when handling long string names. This can be exploited to hide strings in a registry key by creating a string with a long name, which causes this string and any subsequently created strings in the key to be hidden.

Successful exploitation e.g. makes it possible for malware to hide strings in the "Run" registry key. However, these hidden strings created after the string with the overly long name will still be executed when the user logs in.

Contact Secunia for a customised vulnerability solution

Secunia
Stay Secure

Secunia Poll

What is your primary protection against being hacked?

- Antivirus technology
- Firewall technology
- My operating system
- Use uncommon software
- Patching
- Careful behaviour
- Other

See Results Vote!

Most Popular Advisories

1. Adobe Reader / Acrobat Multiple Vulnerabilities

Windows Registry - Cross-view diff

- Deviation between usermode and the raw registry file

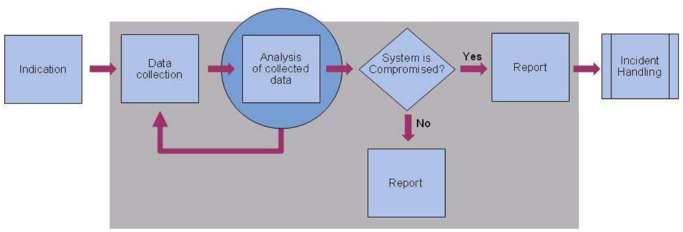
C:\WINDOWS\system32\cmd.exe

```
### keys hidden from user mode
/SYSTEM/ControlSet001/Services/MRxDAV/EncryptedDirectories/(null) SZ,,
/SYSTEM/ControlSet003/Services/MRxDAV/EncryptedDirectories/(null) SZ,,
### keys with no read permissions
/SAM/SAM
/SECURITY
/SYSTEM/ControlSet001/Control/Class/{4D36E965-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:29
/SYSTEM/ControlSet001/Control/Class/{4D36E967-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:29
/SYSTEM/ControlSet001/Control/Class/{4D36E968-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30
/SYSTEM/ControlSet001/Control/Class/{4D36E969-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30
/SYSTEM/ControlSet001/Control/Class/{4D36E96A-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:32
/SYSTEM/ControlSet001/Control/Class/{4D36E97B-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:35
/SYSTEM/ControlSet001/Control/Class/{4D36E980-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30
/SYSTEM/ControlSet003/Control/Class/{4D36E965-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:29
/SYSTEM/ControlSet003/Control/Class/{4D36E967-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:29
/SYSTEM/ControlSet003/Control/Class/{4D36E968-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30
/SYSTEM/ControlSet003/Control/Class/{4D36E969-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30
/SYSTEM/ControlSet003/Control/Class/{4D36E96A-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:32
/SYSTEM/ControlSet003/Control/Class/{4D36E97B-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:35
/SYSTEM/ControlSet003/Control/Class/{4D36E980-E325-11CE-BFC1-08002BE10318}/Properties KEY,,2004-08-20 19:22:30

D:\response\server>
```

Data Analysis

Collected Files



Files Collected during the data acquisition

- Everything running and accessed
  - Running processes
  - Loaded DLLs and drivers
  - Handles that resolves to a file
- Everything being started
  - Registry keys
  - Startup files

Cryptographic hashes - Algorithms

- MD5
  http://en.wikipedia.org/wiki/MD5
  - Hash collisions
  - MD5 Collision Generation by Patrick Stach and Vincent Liu
    http://www.stachliu.com/research_collisions.html
- SHA-1
  http://en.wikipedia.org/wiki/SHA1
  - Hash collisions

Cryptographic hashes - Conclusion

- Use both algorithms! At least when identifying know good files

Cryptographic hashes - Tools to use

- hfind.exe (The Sleuthkit)
  http://www.sleuthkit.org/
  - ➔ Creates index files for the hash database and use that index file to look up a hash value. Described in “The Sleuth Kit Informer” nr 6 and 7.
- md5deep and sha1deep by Jesse Kornblum
  http://md5deep.sourceforge.net/
- md5.exe (cygwin)
- sha1.exe (cygwin)

Cryptographic hashes – Resources (1)

- Online databases
  - NIST - National Software Reference Library
  - Hashkeeper (only available for Law Enforcement and CERT organizations)
- Some of the web sites that list Malware hashes
  - CastleCops
    http://hashes.castlecops.com/
  - Avira
  - HijackThis
  - Spyware Browser AntiSpyware

Cryptographic hashes – Resources (2)

- Generate your own databases of known good files
  - newfind.pl – NIST
    http://www.nsrl.nist.gov/perl/
  - md5deep and sha1deep by Jesse Kornblum
    http://md5deep.sourceforge.net/

Cryptographic hashes – Anti-forensic Attacks

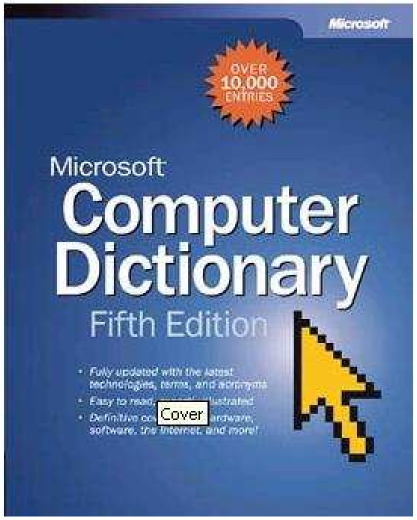
- Attacks against MD5 hashes
  - MD5 and MD4 Collision Generators (Vincent Liu and Patrick Stach) http://www.stachliu.com/research_collisions.html
- Attacks against SHA-1 hashes

Methods for determining File Type

- Check files for data in particular fixed formats
  - ➔ file.exe
    - uses ‘magic’ database
- Look at the file extension

Portable Executable

- **portable executable file** n. The file format used for -executable programs as well as for files that are linked together to form executable programs.
  - .cpl
  - .dll
  - .drv
  - exe
  - .scr
  - .sys
  - .ocx



Portable Executable - Header information

- Tools to use
  - periscope.exe by Arne Vidström
    http://ntsecurity.nu/toolbox/periscope/
  - PE-Header
  - Win32::File::VersionInfo

Portable Executable - Packed files

- Tools to use
    - PEid
      http://peid.has.it/
    - Sigbuster by Toni Koivunen – F-Secure
        - Available to Law enforcement and CERT-organizations

Online Anti-Virus resources

- Virustotal (Hispasec Sistemas)
- Jotti's malware scan
- File Scanner (Kaspersky Lab)

Virustotal

Complete scanning result of "ipv6monl.dll", received in VirusTotal at 11.06.2006, 11:46:09 (CET).

STATUS: FINISHED

Antivirus	Version	Update	Result
AntiVir	7.2.0.37	11.06.2006	TR/Spy.BZub.EC.2
Authentium	4.93.8	11.05.2006	W32/Goldun.gen1
Avast	4.7.892.0	11.03.2006	no virus found
AVG	386	11.04.2006	PSW.Generic2.OPF
BitDefender	7.2	11.06.2006	Trojan.Proxy.Cimuz.AO
CAT-QuickHeal	8.00	11.04.2006	no virus found
ClamAV	devel-20060426	11.06.2006	Trojan.Bzub-38
DrWeb	4.33	11.06.2006	Trojan.PWS.Tanspy
eTrust-InoculateIT	23.73.47	11.06.2006	no virus found
eTrust-Vet	30.3.3178	11.06.2006	Win32/Brospy.CT
Ewido	4.0	11.05.2006	Logger.BZub.ey
Fortinet	2.82.0.0	11.06.2006	suspicious
F-Prot	3.16f	11.04.2006	W32/Goldun.gen1
F-Prot4	4.2.1.29	11.04.2006	W32/Goldun.gen1
Ikarus	0.2.65.0	11.05.2006	no virus found
Kaspersky	4.0.2.24	11.06.2006	Trojan-Spy.Win32.BZub.ey
McAfee	4888	11.03.2006	Generic PWS.q
Microsoft	1.1609	11.06.2006	PWS:Win32/Cimuz.gen
NOD32v2	1.1854	11.06.2006	probably a variant of Win32/Spy.BZub
Norman	5.80.02	11.06.2006	W32/Goldun.gen1
Panda	9.0.0.4	11.06.2006	Suspicious file
Sophos	4.10.0	10.26.2006	Troj/Cimuz-Gen
TheHacker	6.0.1.112	11.03.2006	Trojan/Spy.BZub.ey
UNA	1.83	11.03.2006	Trojan.Spy.Win32.BZub.59D1
VBA32	3.11.1	11.06.2006	suspected of Malware.Agent.4
VirusBuster	4.3.15:9	11.05.2006	TrojanSpy.Agent.BD.Gen

Aditional Information
File size: 67288 bytes
MD5: 290bac6046976d2d5b76e90dcace4cba
SHA1: 09279679f2d9032d02bf020da25c7f14e9dd5bcc
packers: UPX
packers: UPX
packers: UPX
packers: UPX

Online sandbox tools

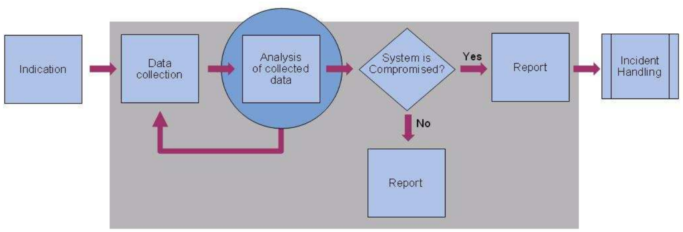
- Norman's Sandbox
- CWSandbox (Carsten Willems)
    - ➔ Sunbelt Sandbox

Analysis methodology for the collected files

- Use a white list and throw all the files that have a matching hash away
- Determine the file type and use appropriate tools
- Use online resources like Virus Total and CWSandbox
- Do a dynamic analysis of the file

System is compromised?

Example 1



Analyzing auto started processes

Looking for good matching hashes:

290bac6046976d2d5b76e90dcace4cba Hash Not Found C\WINDOWS\system32\ipv6monl.dll

ipv6monl.dll - PE-Header

File Version : 5.1.2600.2180
Product Version : 5.1.2600.2180
OS : Unknown/Win32
Type : DLL
CompanyName : Microsoft Corporation
FileDescription : Software Installation Extenstion
FileVersion : 5.1.2600.2190 (xpsp_sp2_rtm.041803-2198)
InternalName : Software Installation Snapin Extenstion
Copyright :
Trademarks :
OrigFileName : ipv6.dll
ProductName : Microsoft« Windows« Operating System
ProductVersion : 5.1.2600.2190
PrivateBuild :
SpecialBuild :

ipv6monl.dll - Google

Nätet Bilder Grupper Kategori

ipv6monl.dll spyware OR spy OR malware OF Sök Avancerad sökning Inställningar

Sök: webben sidor på svenska sidor från Sverige

Nätet Resultat **1** - **10** av ungefär **298** vid sökning efter **ipv6monl.dll spyware OR spy OR malware**

Tips: Sök endast efter **svenska** resultat. Du kan ställa in sökspråk i Inställningar

Troj/Cimuz-AX - **Spyware Trojan** - Sophos threat analysis
Analysis of the Troj/Cimuz-AX **Spyware Trojan**, with information on its **...** BZub; **Trojan-Spy**.Win32.BZub.dt. Protection. Download **virus** identity (IDE) file **...**
www.sophos.com/security/analyses/trojcimuzax.html - 16k - Cachad - Liknande sidor

Troj/Cimuz-AW - **Spyware Trojan** - Sophos threat analysis
Analysis of the Troj/Cimuz-AW **Spyware Trojan**, with information on its behaviour and recovery **...** The file **ipv6monl.dll** is detected as Troj/Cimuz-Gen. **...**
www.sophos.com/security/analyses/trojcimuzaw.html - 16k - Cachad - Liknande sidor
[Fler resultat från www.sophos.com]

IDG.se Eforum - **Virus** problem igen!
O23 - Service: AVG Anti-**Spyware** Guard - Anti-**Malware** Development a.s. - C:\Program **...** 2006-10-09 23:20 64216 --a------ C:\WINDOWS\system32**ipv6monl.dll** **...**
eforum.idg.se/viewmsg.asp?entriesid=875617 - 164k - Cachad - Liknande sidor

AusCERT - AL-2006.0097 -- [Win] -- Flickr site spoofed by **trojan** email
IMPACT: The **trojan malware** intercepts web browser form data to capture online banking **...** The **malware** is installed as C:\windows\system32**ipv6monl.dll** and **...**
www.auscert.org.au/render.html?it=6907 - 19k - Cachad - Liknande sidor

ipv6monl.dll - Dangerous - Greatis Software
UnHackMe - **ROOTKIT** KILLER! It is a time to check your computer. **...** Need help ? Get rid of a **Virus** / **Trojan** / Adware / **Spyware** ? RegRun - User's Choice **...**
www.greatis.com/appdata/d/i/ipv6monl.dll.htm - 22k - Kompletterande resultat - Cachad - Liknande sidor

ipv6monl.dll - VirusTotal

Complete scanning result of "ipv6monl.dll", received in VirusTotal at 11.06.2006, 11:46:09 (CET).

STATUS: FINISHED

Antivirus	Version	Update	Result
AntiVir	7.2.0.37	11.06.2006	TR/Spy.BZub.EC.2
Authentium	4.93.8	11.05.2006	W32/Goldun.gen1
Avast	4.7.892.0	11.03.2006	no virus found
AVG	386	11.04.2006	PSW.Generic2.OPF
BitDefender	7.2	11.06.2006	Trojan.Proxy.Cimuz.AO
CAT-QuickHeal	8.00	11.04.2006	no virus found
ClamAV	devel-20060426	11.06.2006	Trojan.Bzub-38
DrWeb	4.33	11.06.2006	Trojan.PWS.Tanspy
eTrust-InoculateIT	23.73.47	11.06.2006	no virus found
eTrust-Vet	30.3.3178	11.06.2006	Win32/Brospy.CT
Ewido	4.0	11.05.2006	Logger.BZub.ey
Fortinet	2.82.0.0	11.06.2006	suspicious
F-Prot	3.16f	11.04.2006	W32/Goldun.gen1
F-Prot4	4.2.1.29	11.04.2006	W32/Goldun.gen1
Ikarus	0.2.65.0	11.05.2006	no virus found
Kaspersky	4.0.2.24	11.06.2006	Trojan-Spy.Win32.BZub.ey
McAfee	4888	11.03.2006	Generic PWS.q
Microsoft	1.1609	11.06.2006	PWS:Win32/Cimuz.gen
NOD32v2	1.1854	11.06.2006	probably a variant of Win32/Spy.BZub
Norman	5.80.02	11.06.2006	W32/Goldun.gen1
Panda	9.0.0.4	11.06.2006	Suspicious file
Sophos	4.10.0	10.26.2006	Troj/Cimuz-Gen
TheHacker	6.0.1.112	11.03.2006	Trojan/Spy.BZub.ey
UNA	1.83	11.03.2006	Trojan.Spy.Win32.BZub.59D1
VBA32	3.11.1	11.06.2006	suspected of Malware.Agent.4
VirusBuster	4.3.15:9	11.05.2006	TrojanSpy.Agent.BD.Gen

Aditional Information
File size: 67288 bytes
MD5: 290bac6046976d2d5b76e90dcace4cba
SHA1: 09279679f2d9032d02bf020da25c7f14e9dd5bcc
packers: UPX
packers: UPX
packers: UPX
packers: UPX

Recorded network traffic

Stream Content

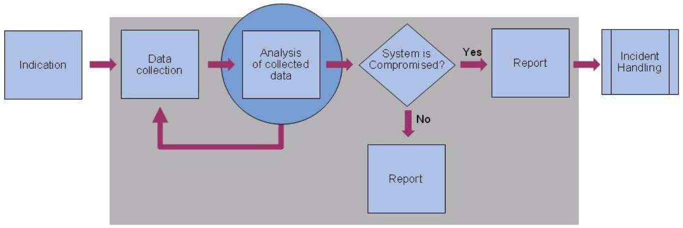
```
GET /flickr.html HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: [redacted]
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Wed, 25 Oct 2006 15:05:51 GMT
Server: Apache/2.0.54 (Debian GNU/Linux) mod_jk/1.2.6 DAV/2 mod_python/3.1.3 Python/2.3.5 PHP/4.3.10-16 mod_ssl/2.0.54 OpenSSL/0.9.7e
mod_perl/1.999.21 Perl/v5.8.4
Last-Modified: Fri, 20 Oct 2006 00:02:33 GMT
ETag: "d01636ca-31a7-d225f840"
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 3856
Connection: close
Content-Type: text/html
```


```
............is.H..S....k2...0..&.mH<.k
.T>....^..F......}.[...g..*.V.....t>...........%..?..8#F.....Y...\.h.......y..G]..._....:.h..oF....D<rY...nt..............LD..g4...C.R
[.h."JfQ......tb../b^T.-}f.[.:1"..Yx.1..M......v`.X<:g'.#[>...3.>.>
.?[...oyD].[.../..B.k...x..o.?R$z.-.>...K.....4....T..H=....?..p.s.....?...s..U`.a..p....%.)..u.".....r.].h......\.1E
+.S.N.......*......"v.=..bs...$...L.R,.2.O".Q.-$.1...2{.08.Q......
.E
....yN(wN...  M...+..C.....>..<.....iT[..........Iy.......4..p........8".C.NO..\./..M.F]>...P......8
..sj.pEpD.'.......)N.H/...!_...".+..\....o.bc.z.......w.o.....w..\....Y.j...J.b~..b.....hvD..$...Z
$...x!..YtD."....&...2jS......L...R..=.^.6.h...D......#.h.n.......).<&o`.  .W/.#..l.U.1...."1?".....Q......5T._g.......g-..o.`.!..5iJG.<....."w.-.
r..UGUXI...
U..v>~.r.7...........l.)?o=F....l..<./..5..dK..&......m.c4.D.Gu..e8<.....G.w....p.
.[.;.k.#..U.......B...+..
..d>./g.. 10.1'..^M;..{.d...$F......n..I.?..{.......=.....R.z{.....P +..S....=q
(....m...g........7.p ?..*..i1C.D.U..b..Ss.......UA.Ea......U..' `eYl
..D.c....J.......n}%.....D6.,Jd"?...gdq...&d.z.  ....-.....o%.b.......
...(.m.9..q.`.0...[P..s.<>L+.s...7.'.U..+...?DL)..$+jr[vK.;$.....$...ECnIa
}..5<:.s]v.YA...c..'D.i..<d..-eiOwS..5-.R..a...
......./...R".d..n f..h=.9.....T'.M.../I...B.q.m7.....U@..a(..7..67.mnW.F+6.U.a=...Mz,....*$..k.X.J`....p.....W*.s.".P.Q..`":vY.1.*.z...J...f
(\..m.*.o.J..[.....*&.mx......Y...q.*K.......-...d.......v$..1..3..(d.>.aH.l....Il'?jM...z....2r.K:f.n@..UN.c.:
+z.*.L*.....o.^.......=..>"`:...8...6..H..R}.y.K;..q...s.q.....r.3.u+.6Pu..5..+.J
25I....(...>k.  .eJ....,d.YM.q.I.....A}P..j..=KM7v._..xl..%w.X....a..O.mi<....@Sa...o.......R.#.W....m..#..  ...6..M.)er.}..vW.....
.P3qhO.pk..
.)K.^.!....].Q..  ....wD..@.(7.l..ER..onp.0.Q`A}.^r7S&...~..Q..y&........B..........*.....={...C9...^v.S.....V6..<...T.n.V(x..s.....V.w..n4c.
+jg....*?"i..@.d.3..\.A.+............!e8......<...U.t...T..z...:.~.....s..>f.o.?...axb...A..\.'f.V...B.m.
\O......H..y.;..@....GoMg.j......H.N.b.U.4...P.FW...C....VYx.......:E..9%6..\...M,.6!?G$.H.
..Go.....;...691....e....[..=.....\x..v7..1..@.\|."[............J..u#..s{C....W.....
M.r..,K...4.wbxj.e.9.f.  .,R..`.}...R#n........LX..>S.e..N..A.....C.........0~.Wq..F6.
.y.
```

Data Analysis

Memory Dump



Memory dump – Data of interest

- Processes (running and terminated)
- Drivers (create threads running in the context of the system process)
- Threads (running and terminated)
- Network activity (listening, active and closed sockets)
- Timestamps of all sort

Memory dump – Different types of dumps

- Physical Memory Dump
- Microsoft Crash dump
- Pagefile
- Hiberfile

Physical Memory dump

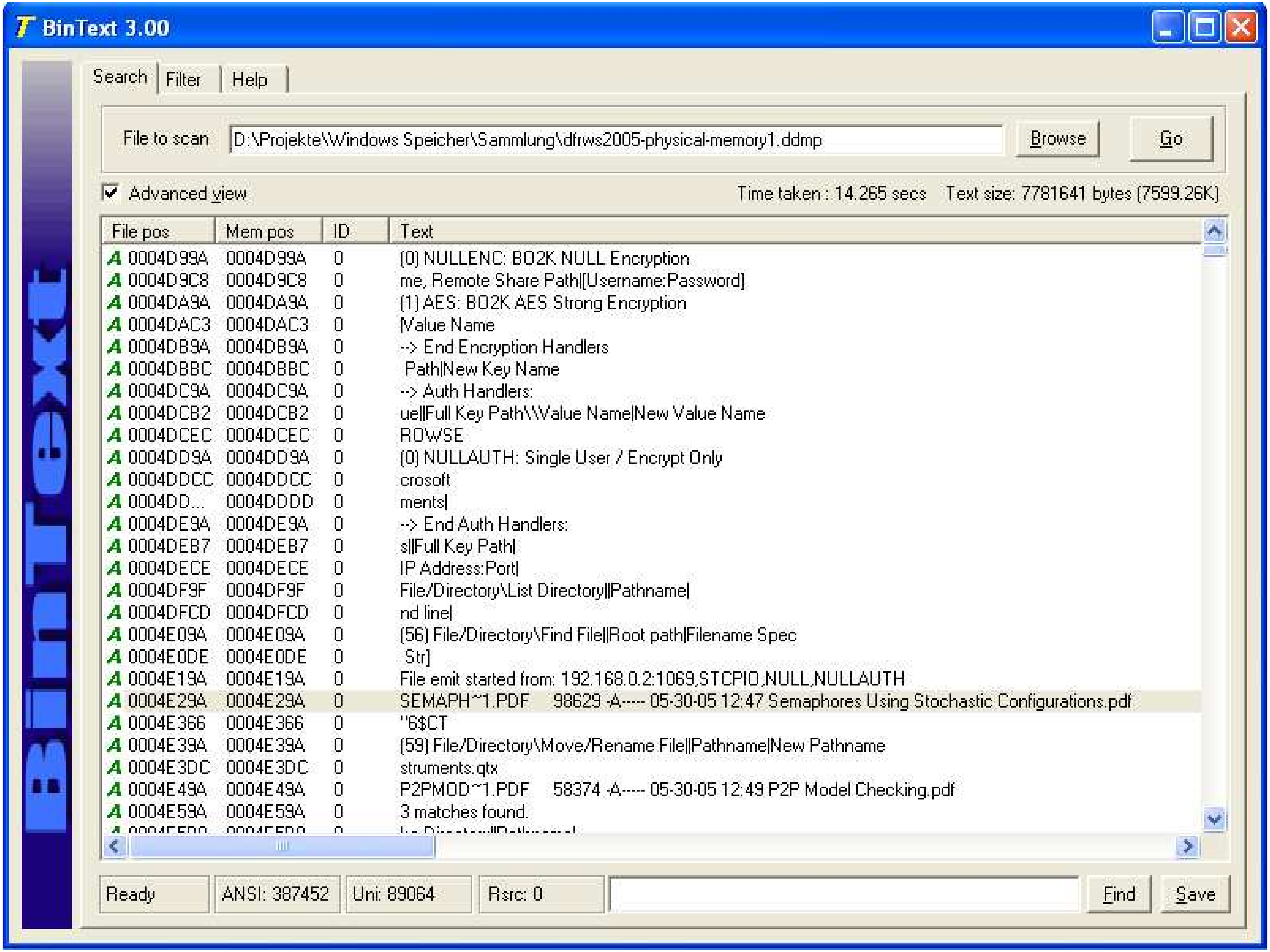
- 1:1 mapping of the physical address space
- Does not have conceptual information about processes

Methods to enumerate information

1. Look for a printable string
2. Reconstruct internal data structures
3. Search for static signatures of kernel data structures

Method 1: Search for sequences of printable characters.

- Some implementations:
  - UNIX strings(1) generally only catches ASCII text
    GNU: mind the option "–e" to catch Unicode strings
  - Sysinternals strings
    defaults to Unicode and ASCII, minimum length 3 characters
    http://www.microsoft.com/technet/sysinternals/utilities/strings.mspx
  - Foundstone BinText
    by Robin Keir / Foundstone


BinText 3.00
Search
Filter
Help
File to scan
D:\Projekte\Windows Speicher\Sammlung\dfrws2005-physical-memory1.ddmp
Browse
Go
Advanced view
Time taken : 14.265 secs
Text size: 7781641 bytes (7599.26K)
File pos
Mem pos
ID
Text
A 0004D99A 0004D99A 0 [0] NULLENC: BO2K NULL Encryption
A 0004D9C8 0004D9C8 0 me, Remote Share Path|[Username:Password]
A 0004DA9A 0004DA9A 0 [1] AES: BO2K AES Strong Encryption
A 0004DAC3 0004DAC3 0 Value Name
A 0004DB9A 0004DB9A 0 --> End Encryption Handlers
A 0004DBBC 0004DBBC 0 Path|New Key Name
A 0004DC9A 0004DC9A 0 --> Auth Handlers:
A 0004DCB2 0004DCB2 0 ue||Full Key Path\\Value Name|New Value Name
A 0004DCEC 0004DCEC 0 ROWSE
A 0004DD9A 0004DD9A 0 [0] NULLAUTH: Single User / Encrypt Only
A 0004DDCC 0004DDCC 0 crosoft
A 0004DD... 0004DDDD 0 ments|
A 0004DE9A 0004DE9A 0 --> End Auth Handlers:
A 0004DEB7 0004DEB7 0 s||Full Key Path|
A 0004DECE 0004DECE 0 IP Address:Port|
A 0004DF9F 0004DF9F 0 File/Directory\List Directory||Pathname|
A 0004DFCD 0004DFCD 0 nd line|
A 0004E09A 0004E09A 0 [56] File/Directory\Find File||Root path|Filename Spec
A 0004E0DE 0004E0DE 0 Str]
A 0004E19A 0004E19A 0 File emit started from: 192.168.0.2:1069,STCPIO,NULL,NULLAUTH
A 0004E29A 0004E29A 0 SEMAPH~1.PDF 98629 -A---- 05-30-05 12:47 Semaphores Using Stochastic Configurations.pdf
A 0004E366 0004E366 0 "6$CT
A 0004E39A 0004E39A 0 [59] File/Directory\Move/Rename File||Pathname|New Pathname
A 0004E3DC 0004E3DC 0 struments.qtx
A 0004E49A 0004E49A 0 P2PMOD~1.PDF 58374 -A---- 05-30-05 12:49 P2P Model Checking.pdf
A 0004E59A 0004E59A 0 3 matches found.
Ready
ANSI: 387452
Uni: 89064
Rsrc: 0
Find
Save

Method 1: Search for sequences of printable characters.

- Drawbacks:
  - No context, difficult to interpret.
  - A lot of interesting information is not in a printable format:
    - Timestamps (FILETIME, uint32)
    - IP addresses

And how can we find that?

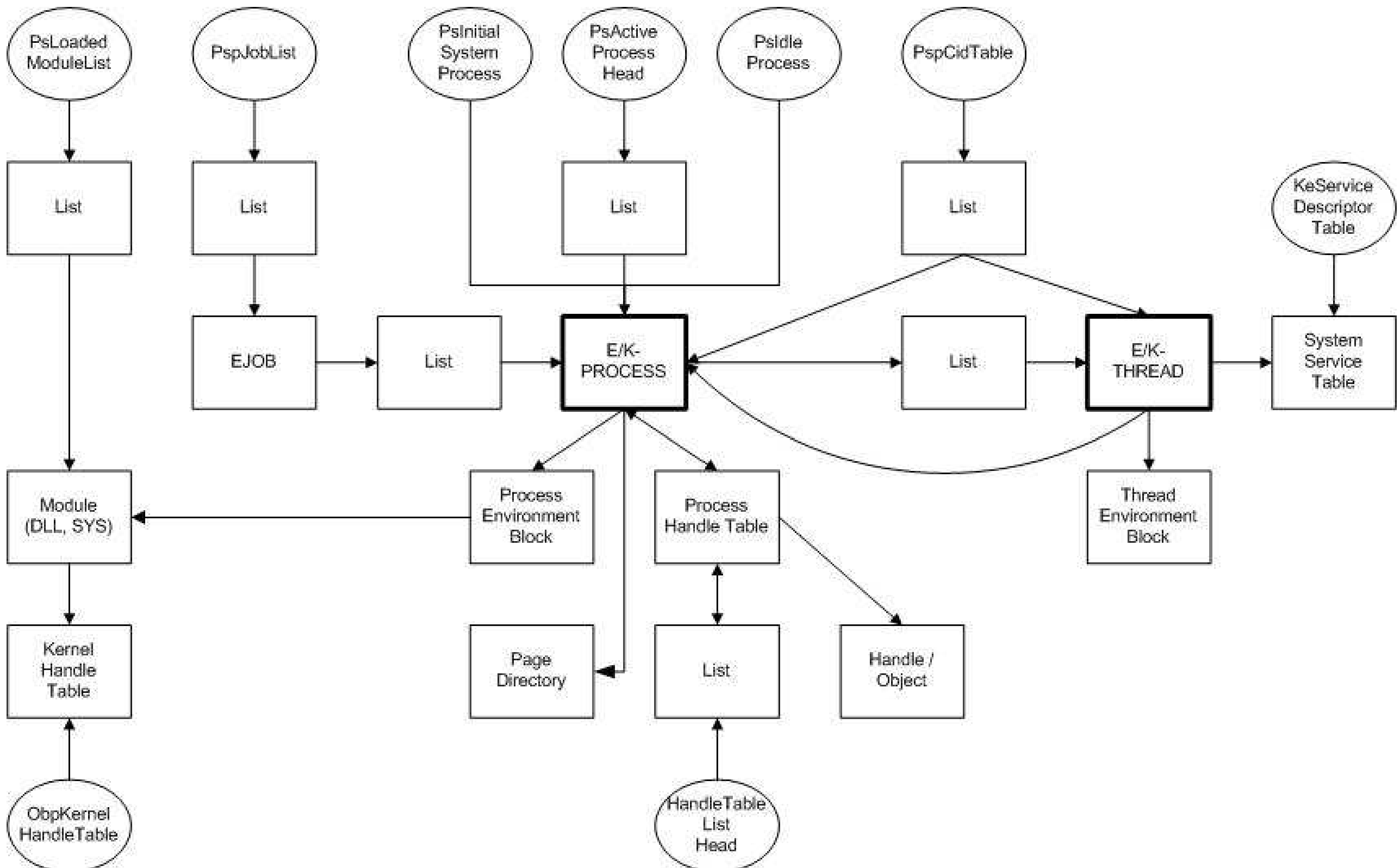
1. Look for printable text.
2. Reconstruct internal data structures.
3. Search for static signatures of kernel data structures.
4. Look for deviations between the results from different levels and from usermode (cross-view detection).
5. Look for “bad” structures.

Method 2: Reconstruct internal data structures.

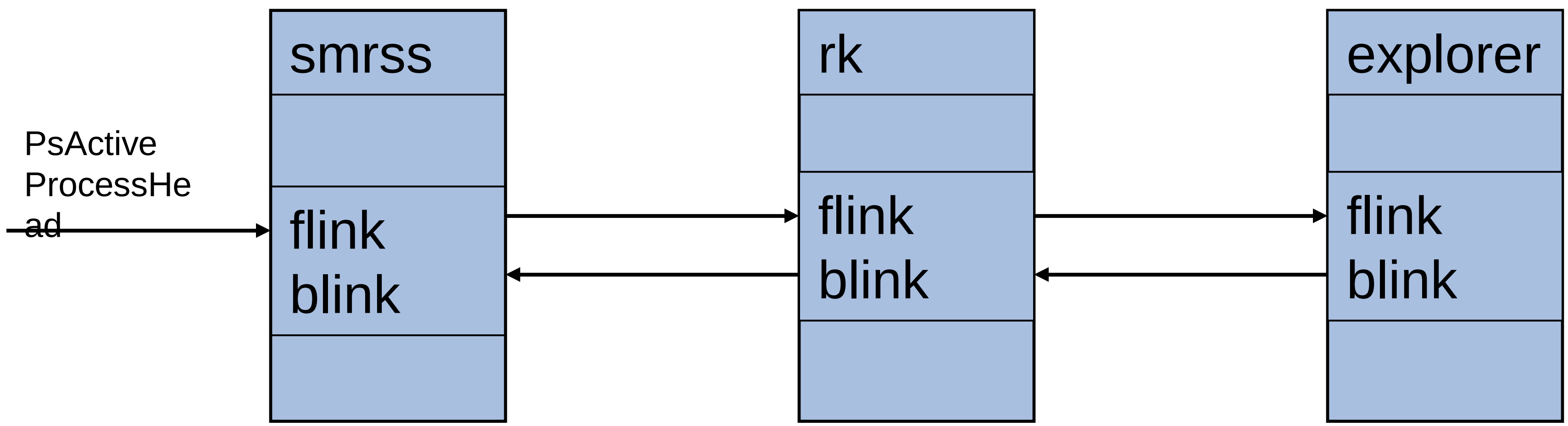
- Most data is kept in Lists and Trees.
- From a known starting point reconstruct and follow the list/tree and enumerate the objects found (aka “list-walking”).
- The most important structure is: _LIST_ENTRY, a double-linked list element.

```
kd> dt _LIST_ENTRY
   +0x000 Flink            : Ptr32 _LIST_ENTRY
   +0x004 Blink            : Ptr32 _LIST_ENTRY
```

Data Analysis
Physical Memory Dump



Enumerating the list of processes



Method 3: Search for signatures of kernel data structures.

- Simple, brute-force searching.
- Largely independent from the dump file format.
- Fast, low memory requirements.
- Problems:
  - ➔ Assuring a sufficient selectivity.
  - ➔ Signature should be based on essential data, otherwise it can be easily defeated.

Method 3: Search for static signatures of kernel data structures.

- Memory management – POOL_HEADER
- Object management – OBJECT_HEADER
- Object – EPROCESS in this example

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
E1:FB30h:	2C	CB	1C	FF	00	00	00	00	00	00	00	00	00	00	00	00	,...............
E1:FB40h:	04	80	01	16	50	72	6F	E3	02	00	00	00	01	00	00	00	....Pro.........
E1:FB50h:	60	51	E2	FC	00	00	00	20	20	B6	46	80	78	0C	00	E1	`Q..... .F.x...
E1:FB60h:	03	00	1B	00	01	00	00	00	68	CB	1C	FF	68	CB	1C	FF	........h...h...
E1:FB70h:	70	CB	1C	FF	70	CB	1C	FF	00	80	C9	06	00	90	05	07	p...p...........
E1:FB80h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....

E1:FD30h:	04	80	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
E1:FD40h:	E8	07	E0	FC	00	00	00	00	48	CD	1C	FF	48	CD	1C	FF	........H...H...
E1:FD50h:	00	00	00	00	00	00	00	00	00	00	00	00	64	66	72	77	............dfrw
E1:FD60h:	73	32	30	30	35	2E	65	78	65	00	00	00	00	00	00	00	s2005.exe.......
E1:FD70h:	00	02	00	04	00	00	00	00	00	00	00	00	00	00	00	00	....
E1:FD80h:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
E1:FD90h:	00	00	00	00	00	00	00	00	0A	00	00	00	00	00	00	00	....
E1:FDA0h:	03	00	00	00	00	00	00	00	26	00	00	00	00	00	00	00	........&.......

Method 3, Memory Management Layer.

- Memory is managed through the CPU's Memory Management Unit (MMU).
- Allocation granularity is a whole page (usually 4 kiB).
- Concept of "pools": several pages are preallocated to form a pool of memory.
- Small allocations from pool, granularity 8 Bytes (Windows 2000: 32 Bytes).
- Mostly2 Pools:
  - ➔ non-paged pool (frequently used information like Processes, Threads)
  - ➔ paged-pool (allocations also can be found in page file)

Set of Allocators:

- nt!ExAllocatePool - deprecated
- nt!ExAllocatePoolWithTag – most common
- nt!ExAllocatePoolWithQuotaTag – charges current process
- nt!ExAllocatePoolWithTagPriority – specifies importance of request

…

Matching set of Deallocators:

- nt!ExFreePool
- nt!ExFreePoolWithTag

…

Some subsystems provide their own set of (de)allocators.

_POOL_HEADER structure

```
>dt nt!_POOL_HEADER
  +0x000 PreviousSize           : Pos 0, 9 Bits
  +0x000 PoolIndex              : Pos 9, 7 Bits
  +0x002 BlockSize              : Pos 0, 9 Bits
  +0x002 PoolType               : Pos 9, 7 Bits
  +0x004 PoolTag                : Uint4B
  +0x004 AllocatorBackTraceIndex : Uint2B
  +0x006 PoolTagHash            : Uint2B
```

BlockSize:

- size of this allocation
- pointer to next allocation

PreviousSize:

- size of the previous allocation
- pointer to previous allocation
- 0 for the first allocation in a page

Both:

- measured in units of 8 bytes (Windows 2000: 32 bytes).
- includes the _POOL_HEADER (8 bytes), so must be 1 at least.

Pool type:

- Declared in Windows Development Kit, file wdm.h.
- values used in memory increased by 1.

Distinction:

- 0 = block is free (deallocated)
- odd = non-paged pool
- even = paged pool

.

PoolTag:

- According to documentation of ExAllocatePoolWithTag in MSDN:
  - ➔ up to 4 character literals
  - ➔ ASCII values between 0 and 127
  - ➔ stored in little-endian (reverse) byte-order
    ‘1234’ stored as ‘4321’
  - ➔ every allocation code path should use a unique pool tag
  - ➔ “protection” bit for kernel objects
- There is no registry for pool tags.
- Every application is free to use any pool tag!

Method 3, Object Management Layer.

```
struct _OBJECT_HEADER, 12 elements, 0x20 bytes
 +0x000 PointerCount       : Int4B
 +0x004 HandleCount        : Int4B
 +0x004 SEntry             : Ptr32
 +0x008 Type               : Ptr32 to struct _OBJECT_TYPE
 +0x00c NameInfoOffset     : UChar
 +0x00d HandleInfoOffset   : UChar
 +0x00e QuotaInfoOffset    : UChar
 +0x00f Flags              : UChar
 +0x010 ObjectCreateInfo   : Ptr32
 +0x010 QuotaBlockCharged  : Ptr32
 +0x014 SecurityDescriptor : Ptr32
 +0x018 Body
```

```
struct _OBJECT_TYPE, 12 elements, 0x190 bytes

   +0x000 Mutex              : struct _ERESOURCE
   +0x038 TypeList           : struct _LIST_ENTRY
   +0x040 Name               : struct _UNICODE_STRING
   +0x048 DefaultObject      : Ptr32 to Void
   +0x04c Index              : Uint4B
   +0x050 TotalNumberOfObjects : Uint4B
   +0x054 TotalNumberOfHandles : Uint4B
   +0x058 HighWaterNumberOfObjects : Uint4B
   +0x05c HighWaterNumberOfHandles : Uint4B
   +0x060 TypeInfo           : struct _OBJECT_TYPE_INITIALIZER
   +0x0ac Key                : Uint4B
   +0x0b0 ObjectLocks        : [4] struct _ERESOURCE
```

PoolTags to look for - nt!ObpAllocateObject

```
004D7BD4 CheckForTag:
004D7BD4      cmp edi, esi ; null object?
004D7BD6      mov eax, 'TjbO' ; default pool tag
004D7BDB      jz short AllocateMemory
004D7BDD      mov eax, [edi+_OBJECT_TYPE.Key]
004D7BE3 AllocateMemory:
004D7BE3      or eax, 80000000h ; set protection bit
004D7BE8      push eax ; Tag
004D7BE9      mov eax, [ebp+arg_10]
004D7BEC      add ecx, eax
004D7BEE      push ecx ; NumberOfBytes
004D7BEF      push edx ; PoolType
004D7BF0      call _ExAllocatePoolWithTag@12
```

TypePointers

- Type pointer depends on:
  - OS version
  - amount of main memory
  - other factors?
- Values to scan for:
  - PsJobType
  - PsProcessType
  - PsThreadType
  - magic numer 0xbad0b0b0, indicates a defunct object (not necessarily a process or thread)
- The object layer is not suitable to generate static signatures.

Method 3, Object Specifics – Processes and Threads.

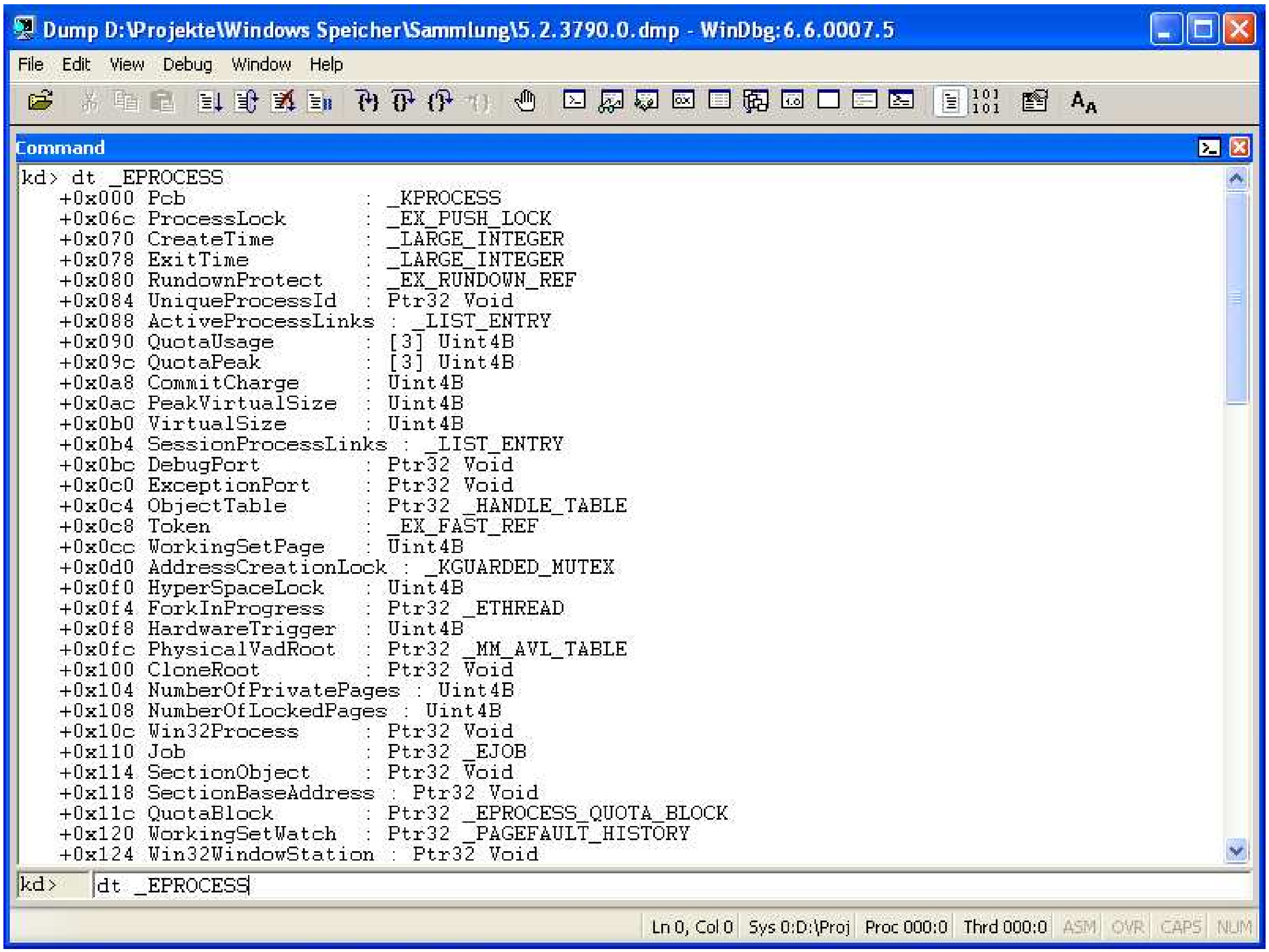
```
struct _EPROCESS, 94 elements, 0x290 bytes
 +0x000 Pcb                    : struct _KPROCESS
  +0x000 Header                 : struct _DISPATCHER_HEADER
   +0x000 Type                   : 0x3
   +0x001 Absolute               : 0
   +0x002 Size                   : 0x1b
   +0x003 Inserted               : 0
   +0x004 SignalState            : 0
   +0x008 WaitListHead           : struct _LIST_ENTRY
…
 +0x070 LockEvent              : struct _KEVENT
  +0x000 Header                 : struct _DISPATCHER_HEADER
…
 +0x130 WorkingSetLock         : struct _FAST_MUTEX
  +0x000 Header                 : struct _DISPATCHER_HEADER
```

Method 3, Object Specifics – Drivers.

```
struct _DRIVER_OBJECT, 15 elements, 0xa8 bytes
   +0x000 Type             : Int2B
   +0x002 Size             : Int2B
   +0x004 DeviceObject     : Ptr32 to struct _DEVICE_OBJECT
   +0x008 Flags            : Uint4B
   +0x00c DriverStart      : Ptr32 to Void
   +0x010 DriverSize       : Uint4B
   +0x014 DriverSection    : Ptr32 to Void
   +0x018 DriverExtension  : Ptr32 to struct _DRIVER_EXTENSION
   +0x01c DriverName       : struct _UNICODE_STRING
   +0x024 HardwareDatabase : Ptr32 to struct _UNICODE_STRING
   +0x028 FastIoDispatch   : Ptr32 to struct _FAST_IO_DISPATCH
   +0x02c DriverInit       : Ptr32 to     long
   +0x030 DriverStartIo    : Ptr32 to     void
   +0x034 DriverUnload     : Ptr32 to     void
   +0x038 MajorFunction    : [28] Ptr32 to     long
```

Excursus

Microsoft’s Debugging Tools



Dump D:\Projekte\Windows Speicher\Sammlung\5.2.3790.0.dmp - WinDbg:6.6.0007.5
File Edit View Debug Window Help
Command
kd> dt _EPROCESS
+0x000 Pcb : _KPROCESS
+0x06c ProcessLock : _EX_PUSH_LOCK
+0x070 CreateTime : _LARGE_INTEGER
+0x078 ExitTime : _LARGE_INTEGER
+0x080 RundownProtect : _EX_RUNDOWN_REF
+0x084 UniqueProcessId : Ptr32 Void
+0x088 ActiveProcessLinks : _LIST_ENTRY
+0x090 QuotaUsage : [3] Uint4B
+0x09c QuotaPeak : [3] Uint4B
+0x0a8 CommitCharge : Uint4B
+0x0ac PeakVirtualSize : Uint4B
+0x0b0 VirtualSize : Uint4B
+0x0b4 SessionProcessLinks : _LIST_ENTRY
+0x0bc DebugPort : Ptr32 Void
+0x0c0 ExceptionPort : Ptr32 Void
+0x0c4 ObjectTable : Ptr32 _HANDLE_TABLE
+0x0c8 Token : _EX_FAST_REF
+0x0cc WorkingSetPage : Uint4B
+0x0d0 AddressCreationLock : _KGUARDED_MUTEX
+0x0f0 HyperSpaceLock : Uint4B
+0x0f4 ForkInProgress : Ptr32 _ETHREAD
+0x0f8 HardwareTrigger : Uint4B
+0x0fc PhysicalVadRoot : Ptr32 _MM_AVL_TABLE
+0x100 CloneRoot : Ptr32 Void
+0x104 NumberOfPrivatePages : Uint4B
+0x108 NumberOfLockedPages : Uint4B
+0x10c Win32Process : Ptr32 Void
+0x110 Job : Ptr32 _EJOB
+0x114 SectionObject : Ptr32 Void
+0x118 SectionBaseAddress : Ptr32 Void
+0x11c QuotaBlock : Ptr32 _EPROCESS_QUOTA_BLOCK
+0x120 WorkingSetWatch : Ptr32 _PAGEFAULT_HISTORY
+0x124 Win32WindowStation : Ptr32 Void
kd> dt _EPROCESS
Ln 0, Col 0 Sys 0:D:\Proj Proc 000:0 Thrd 000:0 ASM OVR CAPS NUM

Display Commands

- db – display BYTEs and ASCII values
- dw – display WORDs
- dd – display DWORDs
- da – display ASCII characters
- du – display UNICODE characters
- there are some more
- d – display the next block of data in the same format

Display Commands

- d* commands default to virtual addresses
  - mind the proper process context!
  - set context with `.process`
- for physical addresses use:
  - `d* /p`
  - `!db`, `!dw`, `!dd`, `!du` (there's no !da)

Display Commands

- dt – display type definition
- Syntax: `dt` options module`!`structure field address
- Options:
  - `-v` – verbosely report size and element count of a structure
  - `-b` – recurse
  - `-p` – apply to physical address
  - `-r` – recursively display substructure
  - `-r`*n* – recursively display substructure, limited to *n* (1-9) levels

Resolve Symbols

- A symbol is a named address.
- To resolve a symbol: ? symbol
- To dereference a symbol as a pointer: `poi(`symbol`)`

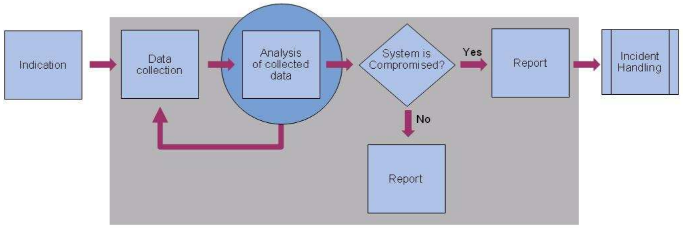
```
kd> dd PsActiveProcessHead L1
805604d8  817cca50

kd> ? PsActiveProcessHead
Evaluate expression: -2141846312 = 805604d8

kd> ? poi(PsActiveProcessHead)
Evaluate expression: -2122528176 = 817cca50
```

Data Analysis

Memory Dump (continued)



Tools to use – Crash Dumps (DMP)

- Microsoft Debugger
- Microsoft Kernel Memory Space Analyzer
- Both are powerful tools, but not intended for forensic purposes.

Tools to use – Raw Dumps (dd)

- kern.pl by Harlan Carvey
  - searches for kernel image at several fixed physical addresses (M. Burdach 2005)
  - when found, evaluates VERSION resource
- os.pl by Harlan Carvey
  - Fingerprinting based on physical addresses, PID of system/idle process etc.
- Both are available from http://downloads.sourceforge.net/windowsir/ostest.zip

Tools to use – Raw Dumps (dd)

- PoolFinder
  http://computer.forensikblog.de/files/poolfinder/poolfinder-current.zip
- Searches for structures on the memory allocation layer.
- Also works on crash dumps, though results are harder to interpret.

Tools to use – Raw Dumps (dd)

- PTFinder
  http://computer.forensikblog.de/files/ptfinder/ptfinder-current.zip
- Searches for processes and threads on the object layer.
- Also works on crash dumps, though parts of the results are harder to interpret.
- Display of process/thread tree requires GraphViz, ZGRviewer is recommended.
- Front end by Richard F.McQuown
  http://www.forensiczone.com/ptfinderfe/PTFinderFE.htm

Tools to use – Raw Dumps (dd)

- Volatility by Aaron Walters and Nick L. Petroni
  https://www.volatilesystems.com/default/volatility
- Lists DLLs, open files, sockets, TCP connections.
- Volatility employs both list-walking amnd scanning routines

Tools to use – Raw Dumps (dd)

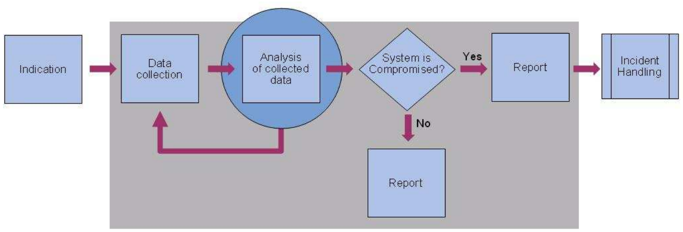
- KnTLIst by GMG Systems, Inc.
  http://www.gmgsystemsinc.com/knttools/
- Runs in batch-mode.
- Gives you an enormous amount of information (more than 2 MB of text, depending on the case).
- Commercial, limited distribution.

Methodology

- Determine dump file type.
- Determine OS version.
- Chose suitable tools.
- Identify processes, threads, drivers and other objects depending on the case.
- Look for unusual data structures and hidden objects.
    - ➔ Cross-view detection
    - ➔ “Exploit the rootkit paradox” (J. Kornblum).
- Build timeline of events.

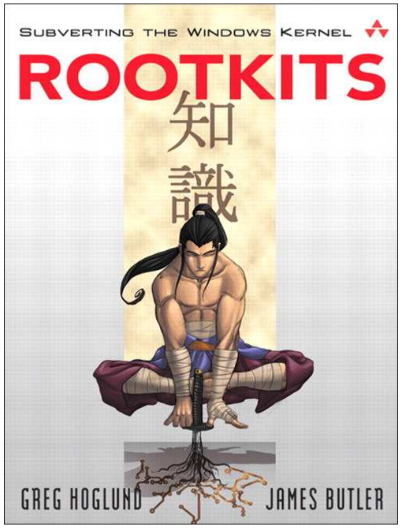
Excursus

Rootkit



Rootkit

- The term rootkit has been around for more than 10 years. A rootkit is a "kit" consisting of small and useful programs that allow an attacker to maintain access to "root," the most powerful user on a computer. In other words, a rootkit is a set of programs and code that allows a permanent or consistent, undetectable presence on a computer.



Different types of rootkit

- User Mode (Ring3)
- Kernel Mode (Ring0)
- Virtualized
- Hardware/Firmware

Rootkit classification

- Type 0
- Type 1
- Type 2
- Type 3

Hardware/Firmware rootkits

- ACPI
  - John Heasman - Implementing and Detecting Implementing and Detecting an ACPI BIOS Rootkit https://www.blackhat.com/presentations/bh-federal-06/BH-Fed-06-Heasman.pdf
- PCI
  - John Heasman - Implementing and Detecting a PCI Rootkit http://www.ngssoftware.com/research/papers/Implementing_And_Detecting_A_PCI_Rootkit.pdf
- Not covered in this course

Virtualization rootkits

- Subvirt
  - Samuel T. King, Peter M. Chen, Yi-Min Wang, Chad Verbowski, Helen J. Wang and Jacob R. Lorch
    www.eecs.umich.edu/~pmchen/papers/king06.pdf
- Blue Pill
  - Joanna Rutkowska
    http://theinvisiblethings.blogspot.com/2006/06/introducing-blue-pill.html
- Not covered in this course

Kernel Mode rootkits (Ring0)

- Executes with the same privileges as the operating system
- Usually works by hooking OS System tables

User Mode rootkits (Ring3)

- Executes with the same privileges as the existing application

Persistant rootkits vs. Memory-based rootkits (1)

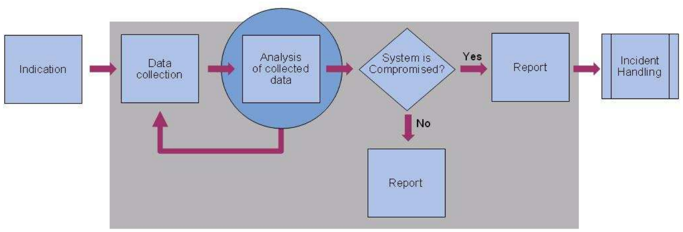
- Persistent Rootkits wants to survive a reboot, hence the rootkit must be initiated from some ware
  - ➔ Registry keys (run keys, file extensions)
  - ➔ Startup files (win.ini, system.ini, config.nt, autoexec.nt)
  - ➔ Patching binaries on disk (Boot Loader, Kernel, Drivers)
  - ➔ using non-existing SafeDllSearchMode
  - ➔ Add-on to an existing application (BHO, Firefox/Thunderbird extensions)
  - ➔ Master Boot Record (MBR)

Persistent rootkits vs. Memory-based rootkits (2)

- Memory-based Rootkits (stealth by design) exist only in memory and does care about surviving a reboot
  - ➔ Most traces of this types of rootkits disappears when the system is rebooted.

Data Analysis

Different rootkit techniques and how we detect it

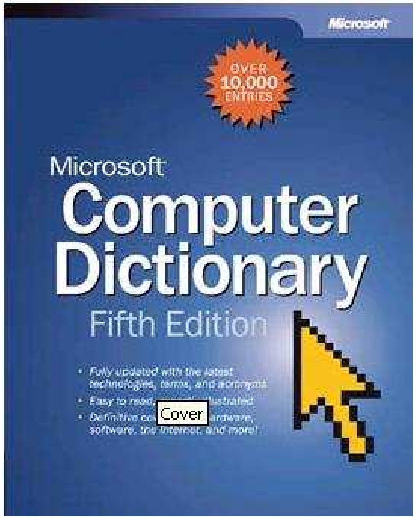


Patching the binary on disk

- Usually old-school user mode rootkits
- Ways to detect the infection
  - Checksums
  - Static analysis of binaries
  - Online resources

Hooking

- **hook** n. A location in a routine or program in which the programmer can connect or insert other routines for the purpose of debugging or enhancing functionality.



Function hooking – Classification

- Hooking of a single program (API hooking)
- Hooking of system tables or exported functions
- Hooking unexported functions

Patching the binary in memory (Hot Patching)

- Ways to detect the infection
  - !chkimg - detects corruption in the images of executable files by comparing them to the image on disk
  - !chksym - detects corruption in the images of executable files by comparing them to the copy on a symbol store or other file repository
  - Inspect system tables and functions

Hooking descriptor tables

- GDT (Global Descriptor Table)
- LDT (Local Descriptor Table)
- IDT (interrupt Descriptor Table)

Hooking descriptor tables

- IDT (Interrupt Descriptor Table) - Each CPU has its own interrupt table
  - ➔ kd> !idt -a (Windows XP and later versions)

Function hooking - Hooking a single program (API hooking)

- Hooking IAT (Import Address Table)
- Hooking Window Messages
- False positives (DLL forwarding)

Function Hooking - IDT

- IRP (I/O Request Packets) Tables
- IDT (Interrupt Descriptor Table) - Each CPU has its own interrupt table
  - ➔ kd> !idt -a (Windows XP and later versions)

Function Hooking - SSDT (1)

- SSDT (System Service Dispatch Table)
  - ➔ nt!KeServiceDescriptorTableShadow
  - ➔ nt!KeServiceDescriptorTable
  - ➔ win32k!W32pServiceTable

Function Hooking - SSDT (2)

- SSDT (System Service Dispatch Table)
  - kd> dps poi (nt!KeServiceDescriptorTableShadow) l dwo (nt!KeServiceDescriptorTableShadow + 0n8)

Function Hooking - System wide hook (2)

- Affects every process in the system
  - IAT
  - EAT
  - SDT
  - SST
  - KiServiceTable
- Ways to detect the infection

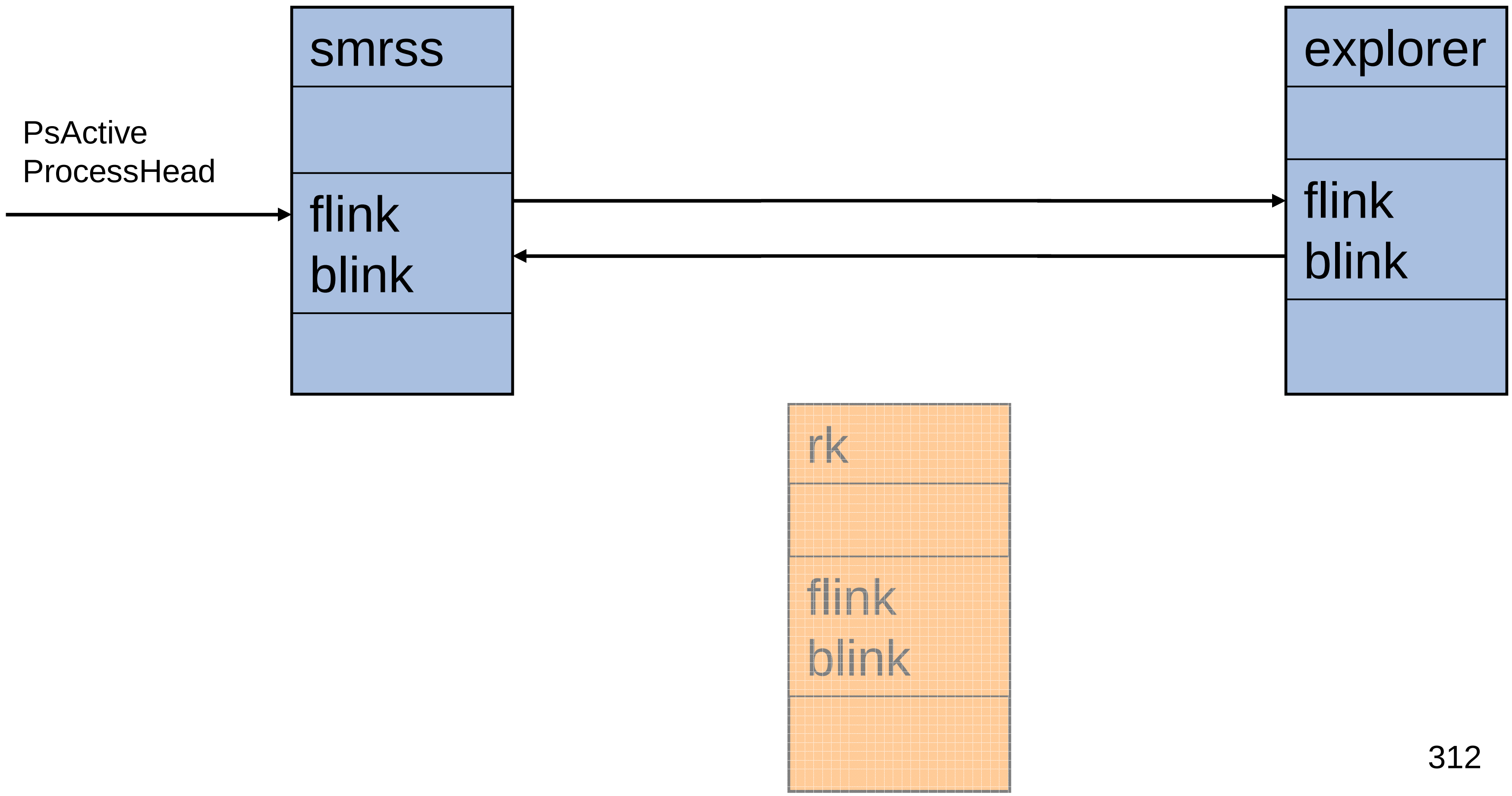
Function Hooking - Inline function hooking (Hot Patching)

- Replaces code inside the original function
- Ways to detect the infection
  - !chkimg
  - enumerate all exported functions
    - kd> x *!*
    - kd> u address – Compare with a list of known instructions

Function Hooking – Hooking unexported functions

- Replaces code in the original function
- Ways to detect the infection
  - kd> u
  - Compare with a list of known instructions

DKOM - Direct Kernel Object Manipulation (1)



DKOM - Direct Kernel Object Manipulation (2)

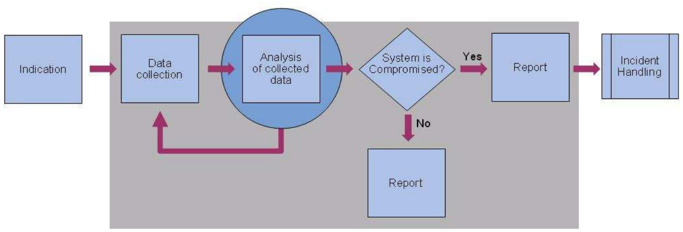
- Works by unlinking doubly linked lists
- Ways to detect the infection
  - Cross view detection
    - List all loaded objects (processes, threads and drivers) by following the memory pool allocations
    - List all threads that are waiting for processor cycles
    - Compare with list enumerated from doubly linked lists

Different rootkit techniques and how we detect it

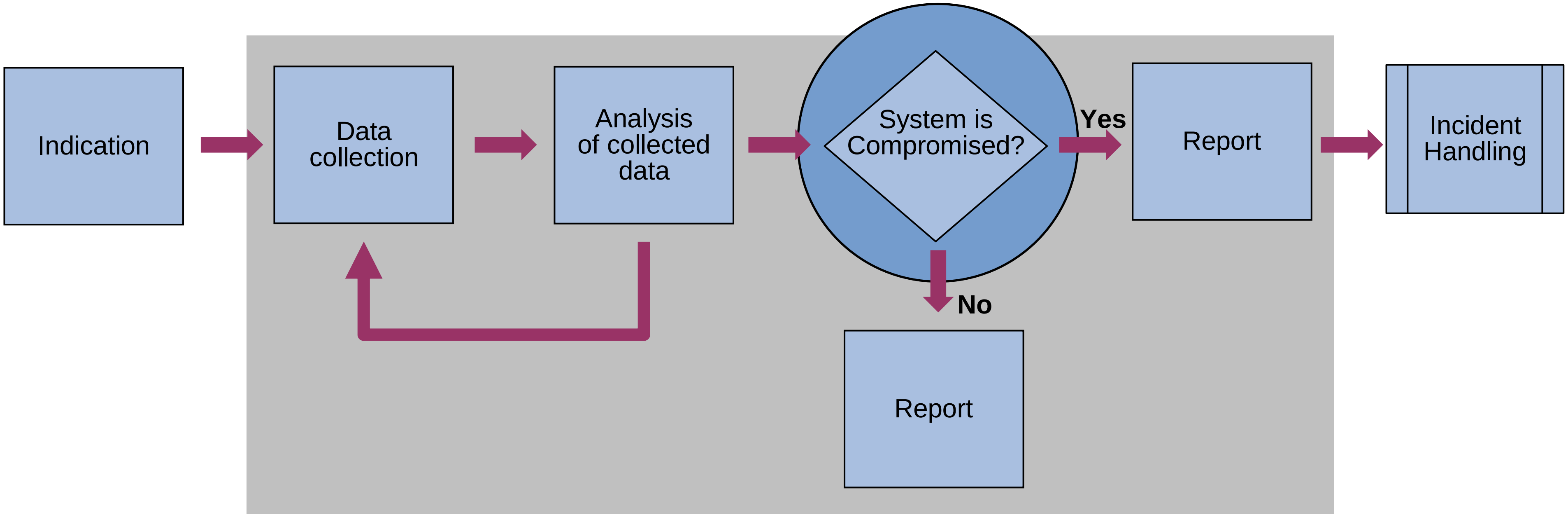
Injecting threads in running processes

- Leaching the process
- Ways to detect the infection

Questions & Answers

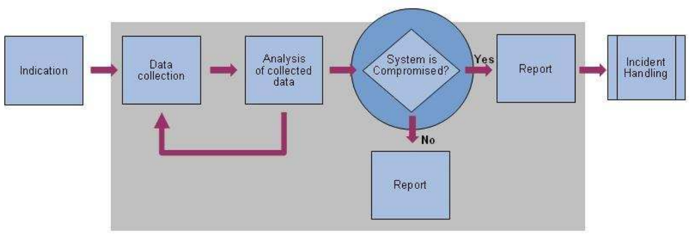


Incident Flowchart



Exercise

Is the system compromised?



Exercise
Is the system compromised?

Exercise 1

- Leaching the process
- Ways to detect the infection

Questions & Answers

Thank you for your attention!

Pär Österberg Medina

Sveriges IT-Incident Centrum
par.osterberg@sitic.se