



IT-Security (ITS) B1

DIKU, E2024



Today's agenda

Forensics defined

Memory forensics

Disk forensics



Forensics defined

Digital forensics is a branch of forensic science encompassing the recovery and investigation of material found on digital devices

Applied in a **corporate**, **civil**, or **criminal** setting (originated in law enforcement)

Applied to a **security** investigation or **personnel** investigation

In security investigations, forensics either means a **root cause or impact analysis** of a cyber-attack, often post-mortem, **or simply techniques** used in the process of uncovering, understanding, and responding to a security incident

In security, **DFIRMA** = digital forensics + incident response + malware analysis



DFIRMA in practice

while true:

- intrusion analysis

- if intrusion suspected:

 - preliminary analysis

 - if intrusion verified:

 - repeat until incident fully contained:

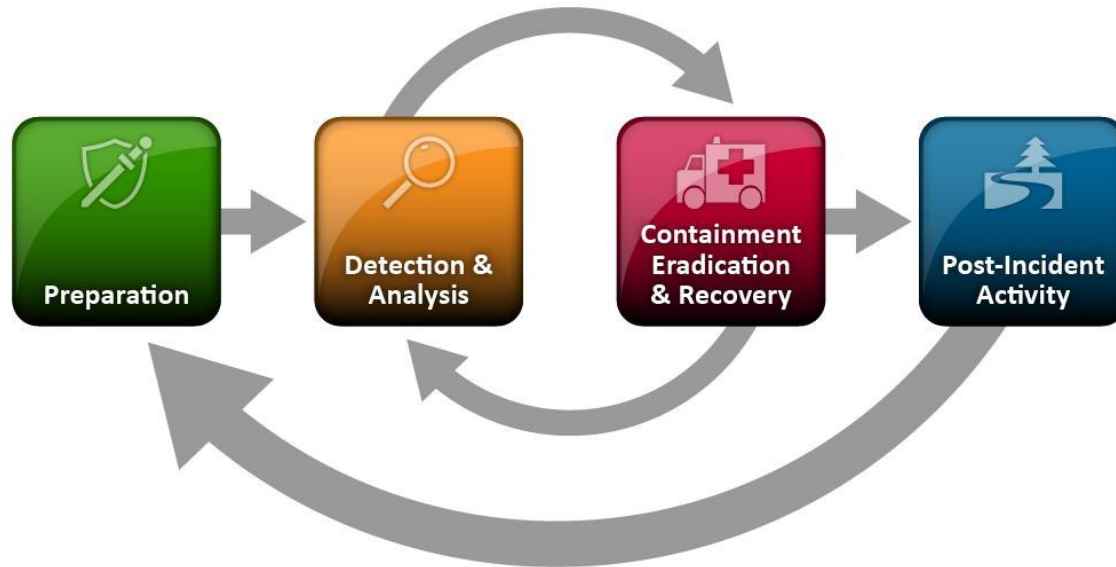
 - forensic analysis

 - malware analysis

 - incident response

- update plans

Recap: Intrusion detection





Many forms of forensics

Digital forensics =

Computer forensics

Memory forensics

Network forensics

Mobile forensics

Etc. forensics



Memory forensics



Memory forensics

From Wikipedia:

“Memory forensics is forensic analysis of a computer's **memory dump**.

Its primary application is investigation of advanced computer attacks which are stealthy enough to avoid leaving data on the computer's hard drive.”



First, get a copy

Live acquisition

Different techniques

Live analysis

Direct analysis of the running kernel

Dead acquisition

Hibernation files, page files

Virtualization - thank you



What to find in memory?

Running processes

Listening sockets

Open connections

Encryption keys

Credentials

Memory only malware

Closed connections

Terminated processes

Open file handles

Deobfuscated code

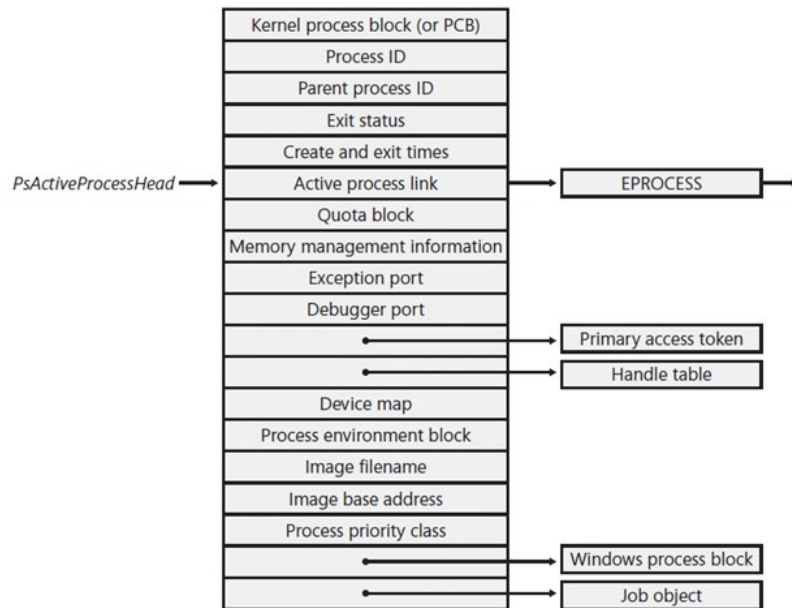


Memory forensic analysis process

- 1: Find rogue processes
- 2: Analyse DLLs
- 3: Review network artefacts
- 4: Look for evidence of code injections
- 5: Dump suspicious processes → further analysis

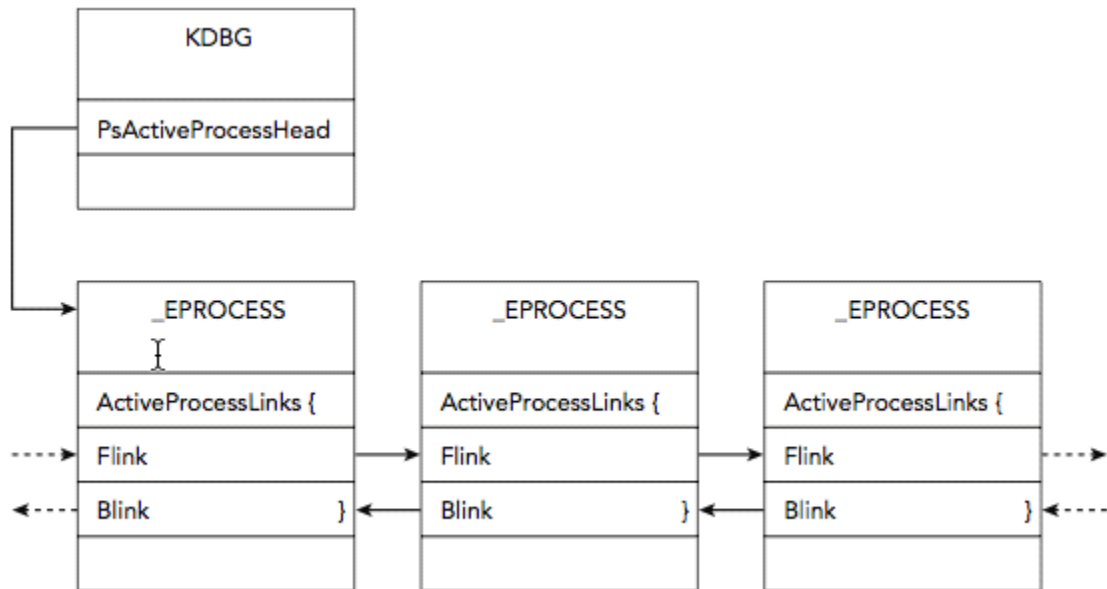
How to find processes (on Windows)

Process scanning:



How to find processes (on Windows)

Process enumeration:





Step 1 revisited: Find rogue processes

Those that:

- Hide

- Have odd parents

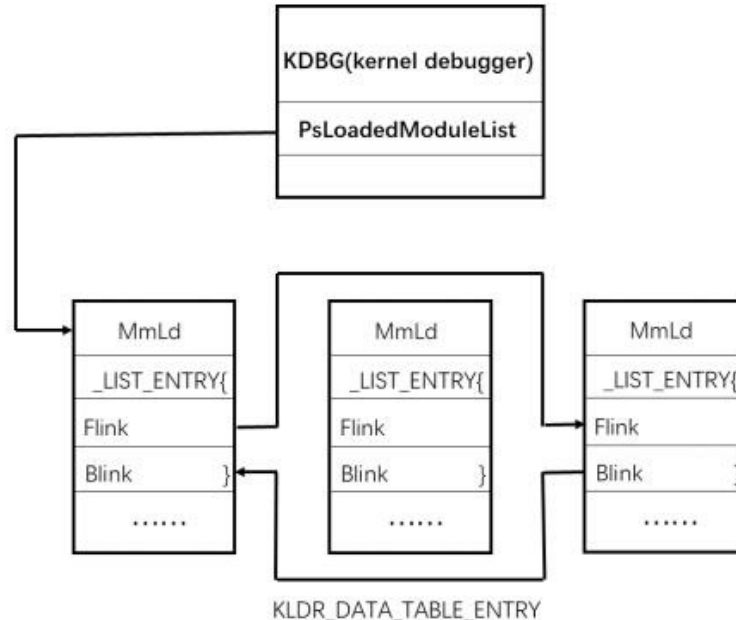
- Do network comm but shouldn't

- Have unusually many handles open

- Contain maliciously injected code

- ...

Direct kernel objection manipulation (DKOM)



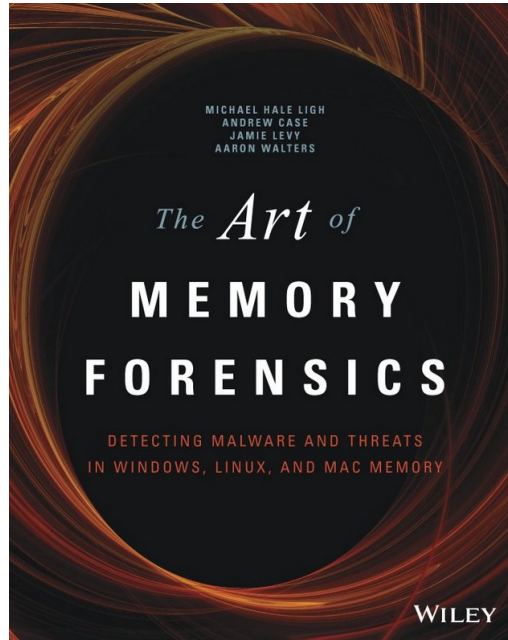
Example: Stuxnet



Volatility and Stuxnet

```
Terminal
File Edit View Search Terminal Help
[zeus stux]$ python volatility/vol.py -f stux.mem --profile=WinXPSP3x86 pslist
Volatility Foundation Volatility Framework 2.5
Offset(V) Name PID PPID Thds Hnds Sess Wow64 Start Exit
-----
0x823c8030 System 4 0 59 403 ----- 0
0x820df020 smss.exe 376 4 3 19 0 0 2010-10-29 17:08:53 UTC+0000
0x821a2da0 csrss.exe 600 376 11 395 0 0 2010-10-29 17:08:54 UTC+0000
0x81da5650 winlogon.exe 624 376 19 570 0 0 2010-10-29 17:08:54 UTC+0000
0x82073020 services.exe 668 624 21 431 0 0 2010-10-29 17:08:54 UTC+0000
0x81e70020 lsass.exe 680 624 19 342 0 0 2010-10-29 17:08:54 UTC+0000
0x823315d8 vmacthlp.exe 844 668 1 25 0 0 2010-10-29 17:08:55 UTC+0000
0x81db8da0 svchost.exe 856 668 17 193 0 0 2010-10-29 17:08:55 UTC+0000
0x81e61da0 svchost.exe 940 668 13 312 0 0 2010-10-29 17:08:55 UTC+0000
0x822843e8 svchost.exe 1032 668 61 1169 0 0 2010-10-29 17:08:55 UTC+0000
0x81e19b20 svchost.exe 1080 668 5 80 0 0 2010-10-29 17:08:55 UTC+0000
0x81ff7020 svchost.exe 1200 668 14 197 0 0 2010-10-29 17:08:55 UTC+0000
0x81fee8b0 spoolsv.exe 1412 668 10 118 0 0 2010-10-29 17:08:56 UTC+0000
0x81e0eda0 jqs.exe 1580 668 5 148 0 0 2010-10-29 17:09:05 UTC+0000
0x81fe52d0 vmtoolsd.exe 1664 668 5 284 0 0 2010-10-29 17:09:05 UTC+0000
0x821a0568 VMUpgradeHelper 1816 668 3 96 0 0 2010-10-29 17:09:08 UTC+0000
0x8205ada0 alg.exe 188 668 6 107 0 0 2010-10-29 17:09:09 UTC+0000
0x820ec7e8 explorer.exe 1196 1728 16 582 0 0 2010-10-29 17:11:49 UTC+0000
0x820ecc10 wscntfy.exe 2040 1032 1 28 0 0 2010-10-29 17:11:49 UTC+0000
0x81e86978 TSVNCache.exe 324 1196 7 54 0 0 2010-10-29 17:11:49 UTC+0000
0x81fc5da0 VMwareTray.exe 1912 1196 1 50 0 0 2010-10-29 17:11:50 UTC+0000
0x81e6b660 VMwareUser.exe 1356 1196 9 251 0 0 2010-10-29 17:11:50 UTC+0000
0x8210d478 jusched.exe 1712 1196 1 26 0 0 2010-10-29 17:11:50 UTC+0000
0x82279998 imapi.exe 756 668 4 116 0 0 2010-10-29 17:11:54 UTC+0000
0x822b9a10 wuaucflt.exe 976 1032 3 133 0 0 2010-10-29 17:12:03 UTC+0000
0x81c543a0 Procmon.exe 660 1196 13 189 0 0 2011-06-03 04:25:56 UTC+0000
0x81fa5390 wmiprvse.exe 1872 856 5 134 0 0 2011-06-03 04:25:58 UTC+0000
0x81c498c8 lsass.exe 868 668 2 23 0 0 2011-06-03 04:26:55 UTC+0000
0x81c47c00 lsass.exe 1928 668 4 65 0 0 2011-06-03 04:26:55 UTC+0000
0x81c0cda0 cmd.exe 968 1664 0 ----- 0 2011-06-03 04:31:35 UTC+0000
0x81f14938 ipconfig.exe 304 968 0 ----- 0 2011-06-03 04:31:35 UTC+0000
[zeus stux]$ python volatility/vol.py -f stux.mem --profile=WinXPSP3x86 pslist | grep lsass
Volatility Foundation Volatility Framework 2.5
0x81e70020 lsass.exe 680 624 19 342 0 0 2010-10-29 17:08:54 UTC+0000
0x81c498c8 lsass.exe 868 668 2 23 0 0 2011-06-03 04:26:55 UTC+0000
0x81c47c00 lsass.exe 1928 668 4 65 0 0 2011-06-03 04:26:55 UTC+0000
[zeus stux]$
```

Further reading



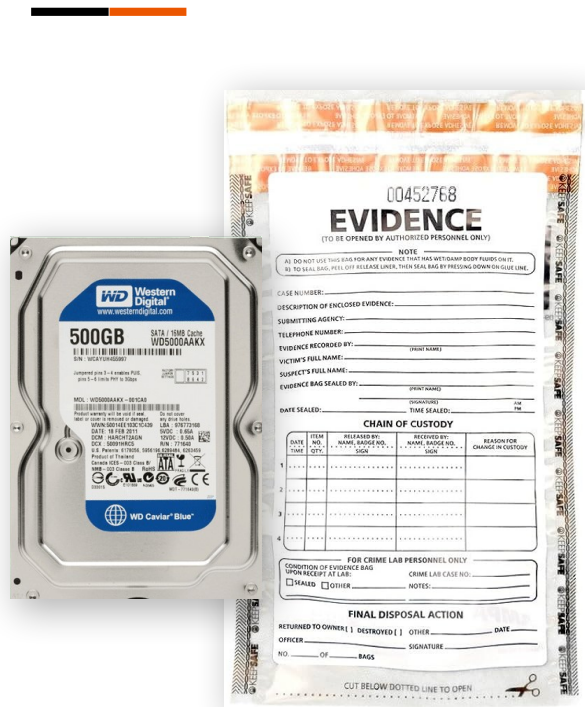


Disk (or, file system) forensics



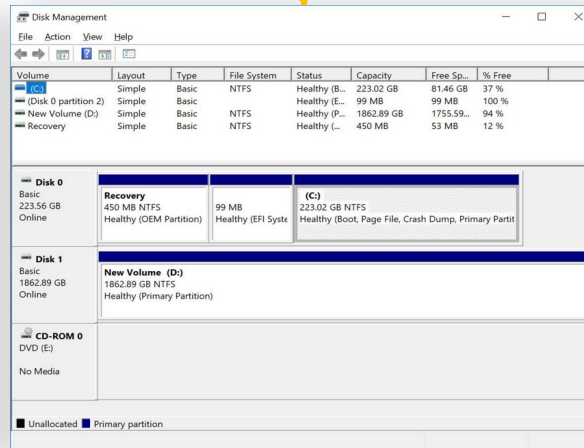
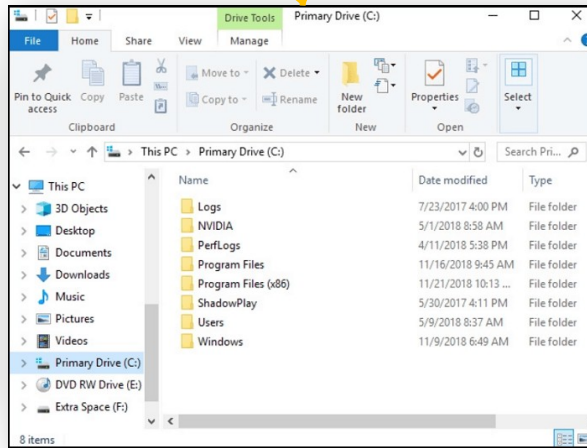
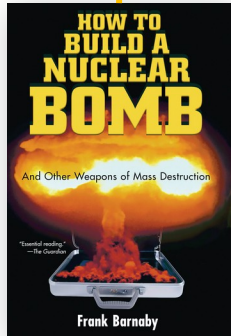
"Vi fik ham.
Bombeplanen
lå på hans
bærbar."

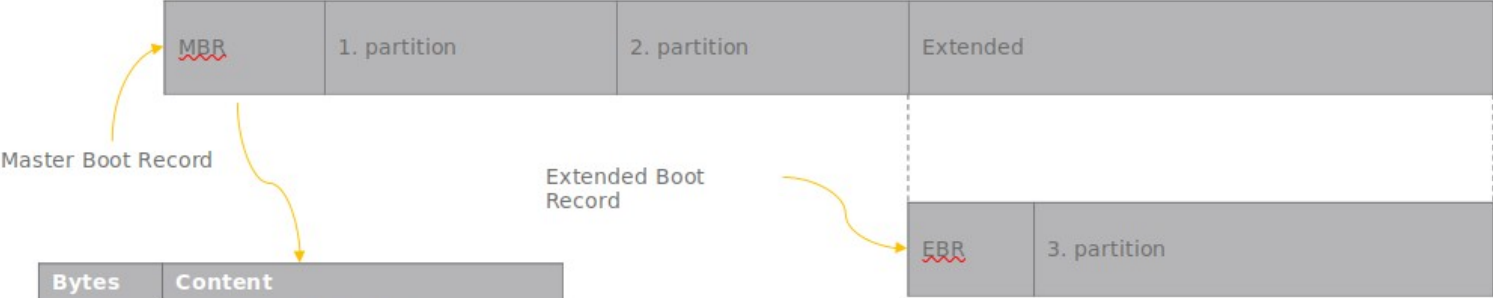




Copy
+ calculate hashsum



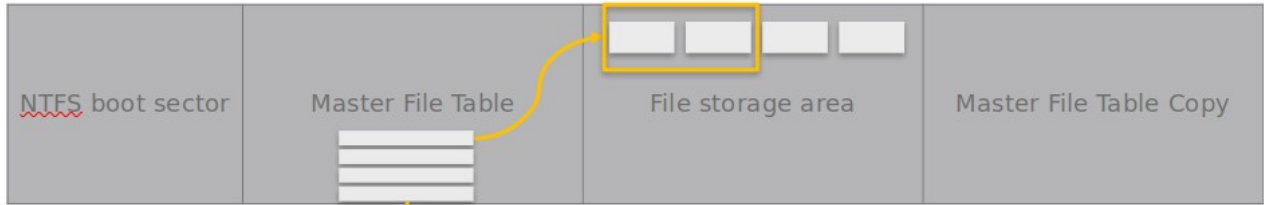




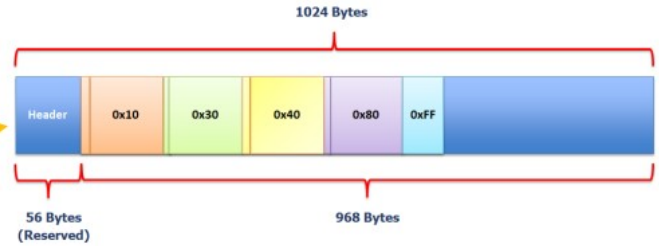
Bytes	Content
0-445	Upstart code, disk signature
446-461	Partition entry 1
462-477	Partition entry 2
478-493	Partition entry 3
494-509	Partition entry 4
510-511	MBR/EBR signature (0xAA55)

Byte s	Content
0	0x00 not boot, 0x80 boot
1-3	Cylinder-head-sector (CHS) of start sector
4	Partition type
5-7	Cylinder-head-sector (CHS) of end sector
8-11	Logical block addressing (LBA) of start sector
12-15	Number of sectors in partition

Type	FAT12	FAT16	FAT32	Linux native	Linux swap	Extended	NFTS
Hex value	0x01	0x0E	0x0C	0x83	0x82	0x05	0x07



0	\$MFT
1	\$MFTMirr
2	\$LogFile
3	\$Volume
4	\$Attr Def
5	.
6	\$Bitmap
7	\$Boot
8	\$BadClus
9	\$Secure
10	\$UpCase
11	\$Extend
12-23	Reserved
24	\$Extend\$Quota
25	\$Extend\$ObjId
26	\$Extend\$Reparse
27-	Beginning of regular file entries.





Deleted != destroyed

When a file is deleted, **data still exists** on disk until overwritten

If overwritten, **remnants may still exist** in

- extra copies of the file

- page/swap/hibernation file, or

- elsewhere on the disk due to (de)fragmentation

However, if disk wiped, only just once, recovery infeasible

Think libraries



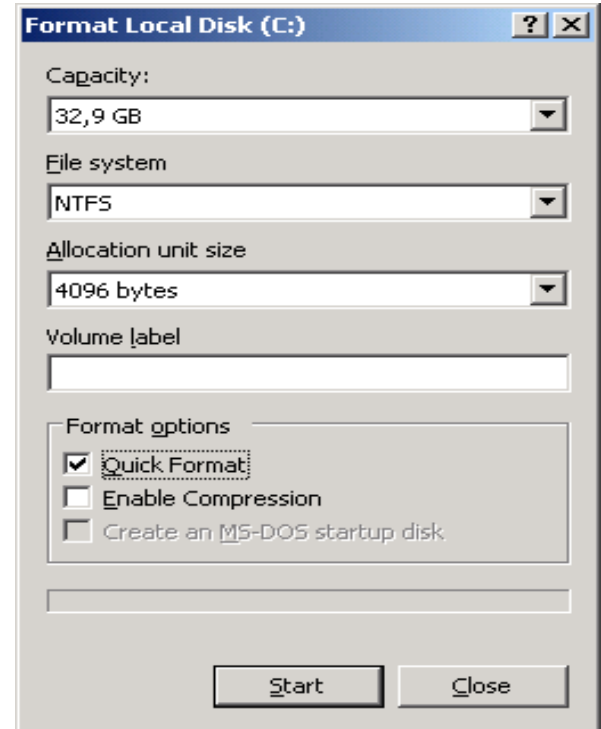
Format is not wiping

Formats create and replace file system structures

Files are not overwritten

Regular formats take longer as the disk is scanned for bad sectors

Use wiping software for wiping



Further reading

