

# Forensic Introduction



**CIRCL**

Computer Incident  
Response Center  
Luxembourg

David Cruciani

david.cruciani@circl.lu

2024-2025

# Me

---

- David Cruciani - david.cruciani@circl.lu
- M2 SSI - Univ Lorraine - 2021
- CIRCL since 2021
- Forensic Analyst
- Developer

# Overview

---

- 1. Introduction - (*Course 1*)
- 2. Understand disk - (*Course 1*)
- 3. Imaging / Cloning and Mounting - (*Course 1*)
- 4. File system analysis - (*Course 2*)
- 5. NTFS - (*Course 2*)
- 6. File System Time Line - (*Course 2*)
- 7. Carving and String Search - (*Course 2*)
- 8. Windows Registry - (*Course 2*)
- 9. Windows Event Logs - (*Course 2*)
- 10. Other Windows Artifacts - (*Course 2*)
- 11. Introduction to Flowintel - (*Course 3*)
- 12. The Exercise - (*Course 3*)

# 1. Introduction

## 1.1 Incident reponse

---

- Someone call for an incident
- Compromised server
- Strange connection from PC of John
- Multiple disks are collected

## 1.1 Incident response - Finding answers

---

- Is there an incident
- System involved at all
- If yes, how and when
- System compromised
- Malware/RAT involved
- Persistence mechanisms
- Root cause of the compromise
- Lateral movement inside LAN
- Access sensitive data
- Data exfiltration
- Illegal content

## 1.1 Incident response - Basic problem: Admin

---

- Get operational asap
  - Re-install
  - Re-image
  - Restore from backup
    - Destroy of evidences
- Analyse the system on his own:
  - Do some investigations
  - Install and run (several) Anti virus
  - Apply updates for OS and Apps
    - Create big noise
    - Overwrite evidences

→ Negative impact on forensics

## 1.2 Preservation of evidences

---

- Legal case:
  - Collect & safe evidences
  - Witness testimony for court
- Use cryptographic hash function
  - Ensure integrity of the evidences
- Make copy of evidences
- Write blocker



## 1.3 Order of Volatility (OOV)

---

- CPU registers → nanoseconds
- CPU cache → nanoseconds
- RAM memory → tens of nanoseconds
- Network state → milliseconds
- Processes running → seconds
- Disk, system settings, data → minutes
- External disks, backup → years
- Optical storage, printouts → tens of years

## 1.4 Forensic Science

---

- Write down everything you see, hear, smell and do
- Scope of the analysis
- Chain of custody
  - A process that tracks the movement of evidence through its collection, safeguarding, and analysis lifecycle by documenting each person who handled the evidence, the date/time it was collected or transferred, and the purpose for any transfers.

→ <https://www.nist.gov/document/sample-chain-custody-formdocx>

## 1.5 Forensic Disciplines

---

- Post-mortem Analysis
- Memory Forensics
- Reverse Engineering
- Code-Deobfuscation
- Network Forensics
- Mobile Forensics
- Cloud Forensics

## 1.6 First Responder: Be prepared

---

- Prepare your toolbox
  - Write Blocker
  - Photo camera
  - Flash light, magnifying glasses
  - Labelling device, labels, tags, stickers
  - Toolkit, screwdriver kits
  - Packing boxes, bags, faraday bag
  - Cable kits, storage devices
  - Anti-static band, network cables
  - Pens, markers, notepads
    - Chain of custody
  - Mouse jiggler
- Talk with people; Take notes
- Identify potential evidences (Computer, devices, paper, ...)

## 1.6 First Responder: First steps

---

- Powered-on versus powered-off
  - Shutdown: Lost of live (memory) data
  - Pull power: Corrupt file system
  - Live analysis: Modify memory and disk
  - Live analysis: Working with compromised binaries?
- USB stick
  - 256 GB USB3
  - File system: exFAT
  - Memory dump: Comae-Toolkit
  - Memory and Live Acquisition: FTK Imager Lite
  - Encrypted Disk Detector - Edd
  - Security Scanner: Nmap command line
  - Sysinternals Suite

## 1.6 First Responder: Live Response

---

- In case of a live analysis:
  - System time
  - Logged-on users
  - Open files
  - Network - connections - status
  - Process information - memory
  - Process / port mapping
  - Clipboard content
  - Services
  - Command history
  - Mapped drives / shares
  - !!! Do not store information on the subject system !!!

## 1.6 First Responder: Live Response

---

- Isolate system from (WiFi) network
- Perform memory dump
- Shutdown and do disk image (If possible)
- Logical image of live system (Possible issues)

## 1.7 Post-mortem Analysis

---

- Hardware layer & acquisition
  - Best copy (in the safe)
  - Working copy (on a NAS)
  - Working copy attached with Write Blocker
  - Disk volumes and partitions
  - Simple tools: dmesg, dd, mount
- Sector layer
  - Carving: foremost, scalpel, testdisk/photorec
  - String search
- File system layer
  - FAT, NTFS
  - File system timeline
  - Restore deleted files



## 1.7 Post-mortem Analysis

---

- OS layer
  - Registry
  - Event logs
  - Volume shadow copies
  - Prefetch files
- Application layer
  - AV logs
  - Browser history: IE, firefox, chrome
  - Email
  - Office files & PDFs
- Searching for malware
  - TEMP folders
  - Startup folders
  - Windows tasks

## 1.8 Forensic Distributions

---

- Commercial
  - [EnCase Forensic](#)
  - [F-Response](#)
  - [Forensic Toolkit](#)
  - [Helix Enterprise](#)
  - [X-Ways Forensics](#)
  - [Magnet Axion](#)
- Open source tools
  - [Kali Linux](#)
  - [SANS SIFT](#)
- Consider using your favorite Linux and add tools
- Sometimes a Windows based VM could be helpful

## 2. Understand disk

## 2.1 Some history

---

- Magnetic storage
  - Tapes
  - Floppy disks
  - Hard disks
- Optical storage
  - Compact disks - CD
  - Digital versatile disk - DVD
  - Blu-ray disk
- Non-volatile memory
  - USB flash drive
  - Solid state drive
  - Flash memory cards

## 2.2 ATA Disks

---

- ATA-3: Hard disk password
- ATA-4: HPA - Host Protected Area
  - Not accessible by OS / user
  - Persistent data - Survive format and re-installation
  - Vendor area - Created by manufacturer
  - Diagnostics and recovery tools
- ATA-6: DCO - Device Configuration Overlay
  - Supports manufacturers with a layer of abstraction
    - Use standard parts
      - To build different products
      - Example: Disks reports unique amount of sectors
- ATA-7: Serial ATA

## 2.3 Hidden Sectors

---

- Create hidden message

```
$ echo -n 'MySecret 123456' | dd of=/dev/sdb seek=3500000000
```

```
$ dd if=/bin/dd of=/dev/sdb seek=3500000001
148+1 records in
148+1 records out
76000 bytes (76 kB, 74 KiB) copied, 0,022659 s, 3,4 MB/s
```

- Create HPA

```
$ hdparm --yes-i-know-what-i-am-doing -N p3000000000 /dev/sdb
setting max visible sectors to 3000000000 (permanent)
max sectors    = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled
```

Power cycle your device after every ACCESSIBLE MAX ADDRESS

## 2.3 Hidden Sectors

---

- Create partition and format

```
$ dmesg
sd 1:0:0:0: [sdb] 3000000000 512-byte logical blocks: (1.54 TB/1.40 TiB)

$ fdisk /dev/sdb
primary
2048
2999999999

$ mkfs.ntfs -L CIRCL.DFIR -f /dev/sdb1
Creating NTFS volume structures.
mkntfs completed successfully. Have a nice day.
```

## 2.3 Hidden Sectors

---

- Investigate disk layout:

```
$ fdisk -l /dev/sdb
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	2999999999	2999997952	1,4T	7	HPFS/NTFS/exFAT

- Investigate last accessible sector:

```
$ dd if=/dev/sdb skip=2999999999 status=none | xxd
```

```
00000000: eb52 904e 5446 5320 2020 2000 0208 0000  .R.NTFS      .....
.....
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ....U.
```



## 2.3 Hidden Sectors

---

- Try to access hidden message

```
$ dd if=/dev/sdb skip=3500000000 count=1 | xxd
dd: /dev/sdb: cannot skip: Invalid argument
0+0 records in
```

- Resize HPA

```
$ hdparm -N /dev/sdb
max sectors    = 3000000000/3907029168, ACCESSIBLE MAX ADDRESS enabled
```

```
$ hdparm --yes-i-know-what-i-am-doing -N p3900000000 /dev/sdb
max sectors    = 3900000000/3907029168, ACCESSIBLE MAX ADDRESS enabled
```

Power cycle your device after every ACCESSIBLE MAX ADDRESS

## 2.3 Hidden Sectors

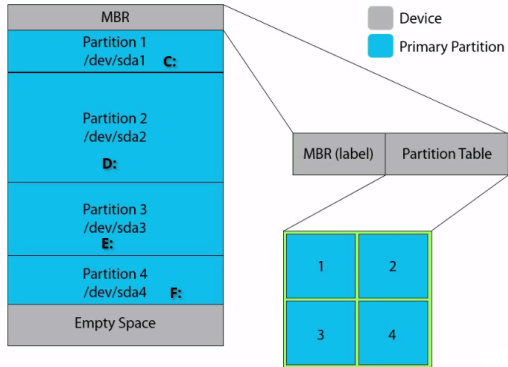
---

- Recover hidden message

```
$ dd if=/dev/sdb skip=3500000000 count=1 status=none  
00000000: 4d79 5365 6372 6574 2031 3233 3435 3600  MySecret 123456.
```

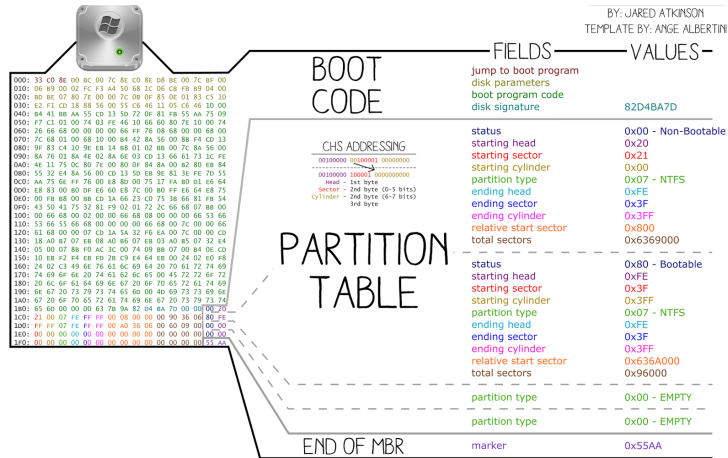
## 2.4 Disk structure

### MBR Partition Scheme



<https://www.golinuxhub.com/2014/11/understanding-partition-scheme-mbr-vs/>

## 2.4.1 MBR



## 2.4.2 CHS

---

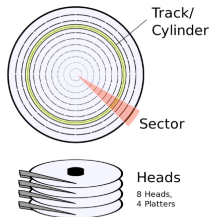
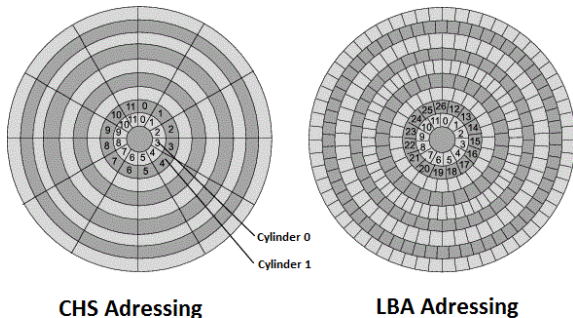


Image (c) wikipedia.org - Image used solely for illustration purposes

- C -> Cylinder, [0, 1023]
- H -> Head, [0, 254]
- S -> Sector, [1, 63]
- $1024 * 255 * 63 * 512 = 8,422,686,720$  bytes  $\rightarrow$  8G

## 2.4.3 CHS vs LBA

---

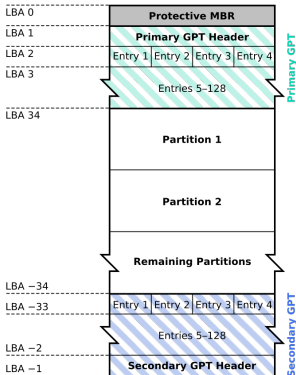


<https://benkixmiller.com/2023/10/08/chs-vs-lba-addressing/>

- $\$(2^{32} * 512 / 1024^3) == 2 \text{ TByte}$
- 48 bits  $\rightarrow$  144,000,000 GB  $\rightarrow$  144 000 TB

## 2.4.4 GPT

### GUID Partition Table Scheme



## 2.4.4 GPT

### GPT HEADER

```
200 45 46 49 20 50 41 52 54 00 00 01 00 5c 00 00 00
210 f3 73 9f 97 01 00 00 00 00 00 00 00 00 00 00 00
220 ff ff 3f 01 00 00 00 00 22 00 00 00 00 00 00 00
230 0e ff 3f 01 00 00 00 00 10 f1 13 f9 35 08 f1 4c
240 96 c7 38 08 50 b4 a4 20 02 00 00 00 00 00 00 00
250 80 00 00 00 80 00 00 00 3b 04 a4 f8
```

signature  
revision  
header size  
header CRC32  
my LBA  
alternate LBA  
first usable LBA  
last usable LBA  
disk guid  
partition entry LBA  
# of partition entries  
size of partition entry  
partition entry array CRC32

EFI PART  
1.0  
92  
979f73f3  
1  
20971519  
34  
20971486  
f913e110-0835-4cf1-96c7-380b5db4a42d  
2 (sector containing of partition table)  
128  
128  
f8a4043b

### PARTITION ARRAY

```
400 16 e3 c9 e3 5c 0b 88 40 81 79 f9 20 f0 02 15 ae
410 47 8a 1a ff f8 08 ab 43 b4 10 53 69 7f 08 23 23
420 22 00 00 00 00 00 00 21 00 01 00 00 00 00 00
430 00 00 00 00 00 00 00 40 00 69 00 63 00 72 00
440 6f 00 73 00 6f 00 66 00 74 00 20 00 72 00 65 00
450 73 00 65 00 72 00 76 00 65 00 64 00 20 00 70 00
460 61 00 72 00 74 00 69 00 74 00 69 00 6f 00 6e 00
470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

partition type guid  
unique partition guid  
starting LBA  
ending LBA  
attributes  
partition name

e3c9e316-0b5c-4db8-817d-f92df00215ae  
ffa18a47-08f8-43ab-b410-53697f0b2323  
34  
65569  
0  
Microsoft reserved partition

```
480 a2 a0 d0 eb e5 b9 33 44 87 c0 68 b6 87 26 99 c7
490 42 ae 76 60 c1 b6 8e 4f 80 42 20 cd 36 60 26 b4
4a0 00 08 01 00 00 00 00 ff 07 00 00 00 00 00 00
4b0 00 00 00 00 00 00 00 42 00 61 00 73 00 69 00
4c0 63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00
4d0 61 00 72 00 74 00 69 00 74 00 69 00 6f 00 6e 00
4e0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
4f0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

partition type guid  
unique partition guid  
starting LBA  
ending LBA  
attributes  
partition name

ebd0a0a2-b9e5-4433-87c0-68b6b72699c7  
6d76ae42-b6c1-4f8e-8d42-20cd366026b4  
67584  
2164735  
0  
Basic data partition

```
500 a2 a0 d0 eb e5 b9 33 44 87 c0 68 b6 87 26 99 c7
510 3a 5c 79 d6 4d 8a b4 4f 91 a0 48 88 12 cc e0 27
520 00 08 00 00 00 00 00 ff 07 41 00 00 00 00 00
530 00 00 00 00 00 00 00 42 00 61 00 73 00 69 00
540 63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00
550 61 00 72 00 74 00 69 00 74 00 69 00 6f 00 6e 00
560 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
570 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

partition type guid  
unique partition guid  
starting LBA  
ending LBA  
attributes  
partition name

ebd0a0a2-b9e5-4433-87c0-68b6b72699c7  
d6795c3a-8a4d-4fb4-91a0-488812cce027  
2164736  
4261887  
0  
Basic data partition



### 3. Imaging / Cloning and Mounting

## 3.1 Disk Imaging / Cloning

---

- Clone disk-2-disk
  - Different sizes
  - This will wipe target disk!
- Clone disk-2-image
  - Clear boundaries
  - One big file
  - Break file into chunks
- Image file format
  - RAW
  - AFF (Advanced Forensic Format)
  - EWF (Expert Witness Format)
  - Please no 3rd party formats
- Write-Blockers
  - Hardware

## 3.2 dd - disk imaging rudimentary

---

- Copy files from usb stick using dd:

```
$ dd if=img_1.txt of=out_1.txt bs=512
```

```
<input file>    <output file>  <block size>  
3+0 records in  
3+0 records out  
1536 bytes (1.5 kB) copied, 0.000126 s, 12.2 MB/s
```

- Skip and count options:

```
dd if=img_3.txt bs=512 skip=0 count=1 status=none | less  
dd if=img_3.txt bs=512 skip=1 count=1 status=none | less  
dd if=img_3.txt bs=512 skip=2 count=1 status=none | less
```

## 3.2 dd - disk imaging rudimentary

---

- Play with bs, skip and count option

```
dd if=img_3.txt bs=1 skip=$((512*3)) count=16 status=none
```

```
dd if=img_3.txt bs=16 skip=$((32*3)) count=1 status=none
```

- Exercise: dd | xxd | less

```
dd if=img_3.txt bs=512 skip=3 count=1 status=none | xxd | less
```

```
00000000: 4f76 6572 6865 6164 2031 3233 3435 3637  Overhead 1234567
00000010: 3839 3020 204d 6573 7361 6765 2d31 2020  890  Message-1
00000020: 3039 3837 3635 3433 3231 2020 2020 2020  0987654321
00000030: 2020 2020 2020 20
```

- Exercise: Find the secret password behind sector 3

## 3.2 dd - disk imaging rudimentary

---

- Exercise: Continue an interrupted imaging process

## 3.2 dd - disk imaging rudimentary

---

- Exercise: Continue an interrupted imaging process

```
$ dd if=img_2.txt of=broken.raw bs=512 skip=0 count=2 status=none
```

```
||      img_2.txt      ..... 1591 Aug 13 14:40 img_2.txt*  
||      broken.raw     ..... 1024 Aug 13 15:05 broken.raw
```

```
dd if=img_2.txt of=broken.raw bs=512 skip=2 seek=2 status=none
```

```
md5sum  img_2.txt f319b1cc9d424a923a8c83c3e67185f1
```

```
md5sum  broken.raw f319b1cc9d424a923a8c83c3e67185f1
```

## 3.2 dd - disk imaging rudimentary - Tools

---

- dd
- ddrescue, gddrescue, dd\_rescue
- dc3dd - Department of Defense Cyber Crime Center
- dcfldd - Defense Computer Forensic Labs
- rdd-copy, netcat, socat, ssh
- Guymager

## 3.3 Connecting devices

---

- List all disk

`sudo lsblk -o NAME,FSTYPE,SIZE,MOUNTPOINT,LABEL`

- Where there are mount:
  - `/dev/sd*` # SCSI, SATA
    - `/dev/sda1` # Partition 1 on disk 1
    - `/dev/sda2` # Partition 2 on disk 1
    - ...
  - `/dev/hd*` # IDE. EIDE
  - `/dev/md*` # RAID
  - `/dev/nvme*n*` # NVME devices



## 3.3 Connecting devices

---

- `dmesg`

```
[106834.127269] sd 6:0:0:0: Attached scsi generic sg1 type 0
[106834.127503] sd 6:0:0:0: [sdb] 15826944 512-byte logical blocks: (8.10 GB/7.54 GiB)
[106834.130380] sd 6:0:0:0: [sdb] Write Protect is off
```

- `fdisk -l /dev/sda`

```
Disk /dev/sda: 7.62 GiB, 8178892800 bytes, 15974400 sectors
Disk model: Flash Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x223f3288
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		2048	10485759	10483712	?	7	HPFS/NTFS/exFAT

## 3.3 Connecting devices

---

- `fdisk -l /dev/sda`

Disk /dev/sda: 7.62 GiB, 8178892800 bytes, 15974400 sectors

Disk model: Flash Disk

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0x223f3288

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		2048	10485759	10483712	5G	7	HPFS/NTFS/exFAT

## 3.4 Imaging devices

---

- Image your usb stick
- What are the steps ?

## 3.4 Imaging devices

---

```
dd if=/dev/sda of=circl_dfir.dd bs=512
dd if=/dev/sda of=/home/david/Desktop/circl_dfir_ntfs.dd skip=2048
dd if=/dev/sda1 of=/home/david/Desktop/circl_dfir_ntfs_2.dd
```

## 3.5 Mounting

---

- Basics

`mkdir /mnt/ntfs` *# Create mount point*

`mount /dev/sda1 /mnt/ntfs` *# Mounting*

`mount -o ro,remount /dev/sda1 /mnt/ntfs` *# Re-mounting*

`umount /mnt/ntfs` *# Un-mounting*

`umount /dev/sda1` *# Also un-mounting*

## 3.5 Mounting

---

- More Advanced:

*# Mounting readonly, no journaling, no executable*

```
mount -o ro,noload,noexec /dev/sda1 /mnt/ntfs
```

```
mount -o ro,noload,noexec,remount /dev/sda1 /mnt/ntfs
```

*# Mounting with offset. mounting from image files*

```
mount -o ro,noload,noexec,offset=$((512*2048)) circl-dfir.dd /mnt/ntfs
```

*# Mounting NTFS file systems*

```
mount -o ro,noload,noexec,offset=$((512*2048)),  
      show_sys_files,streams_interface=windows circl-dfir.dd /mnt/ntfs
```

## 3.5 Mounting

---

```
dd if=/dev/sda of=circl_dfir.dd bs=512
dd if=/dev/sda of=/home/david/Desktop/circl_dfir_ntfs.dd skip=2048
dd if=/dev/sda1 of=/home/david/Desktop/circl_dfir_ntfs_2.dd

mount -o offset=$((512*2048)) circl_dfir.dd mnt_pt
mount circl_dfir.dd mnt_pt
mount circl_dfir.dd mnt_pt
```

## Contact and Reference

---

- david.cruciani@circl.lu
- <https://github.com/DavidCruciani>
- info@circl.lu