Recovering data from a wiped disk

A manual approach

CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:CLEAR*

info@circl.lu

2023-01-31

1. Introduction

- We do not do Black Magic

- Wiping big disk
  - Very time consuming
  - Maybe interupted after some time

- Real case
  - Insider Threat
  - Huge amount of damage
  - Start wiping his disk
    $\rightarrow$ *This disk is wiped*

- Side note
  - This is a simulation of the real case
  - Analysts should be able to read the bytes

2 Getting started

Connect disk to PC

```
dmesg –w

sd 1:0:0:0: Attached scsi generic sg1 type 0
sd 1:0:0:0: [sdb] 15974400 512–byte logical blocks: (8.18 GB/7.62 GiB)
sd 1:0:0:0: [sdb] Write Protect is off
          .....
```

Read the first sectors from disk

```
dd if=/dev/sdb | xxd | less

          .....
00000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
          .....
```

Read sectors from the middle of the disk

```
dd if=/dev/sdb skip=7400000 | xxd | less

          .....
00000140: 24f0 52cd 1fc1 2e45 11c0 4f70 7619 77cd  $.R....E..Opv.w.
00000150: 5243 2a3e 4aad 989f 0a50 cf68 5460 4d4e  RC*>J....P.hT‘MN
00000160: 7663 6f7a ac1a 2f65 2c3a b84b 6c4a 9544  vcoz../e,:.KlJ.D
          .....
```

2 Analysis: Getting started

Read the last sector of the disk

```
dd if=/dev/sdb skip=$((15974400 - 1)) | xxd | less
```

```
00000000: 4546 4920 5041 5254 0000 0100 5c00 0000  EFI PART....\...
00000010: a101 7b1d 0000 0000 ffbf f300 0000 0000  ..{.............
00000020: 0100 0000 0000 0000 2200 0000 0000 0000  ........".......
00000030: debf f300 0000 0000 c6aa d2d4 5971 2f42  ............Yq/B
00000040: b5bc 520a c650 ece1 dfbf f300 0000 0000  ..R..P..........
00000050: 8000 0000 8000 0000 e156 0e4d 0000 0000  .........V.M....
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
          .....
```

- Looks like a cleartext string
- Quick online search: *EFI PART*

  $\rightarrow$ Could be from a GUID Partition Table - GPT

3. GUID Partition Table - GPT

- Legacy: BIOS with DOS Partition Table (with MBR)

- UEFI - Unified Extensible Firmware Interface
  - Replacing classical BIOS over time
  - Introduced 1998 by Intel
  - Introduce GUID Partition Table - GPT

- GPT features
  - Getting rid of classical DOS MBR restrictions
    - 64 Bit for addressing $\rightarrow$ 9.4 Zettabyte
    - Support 128 partitions
    - Keep backwards compatibility with MBR
    - Provied a backup of the GPT header $\rightarrow\rightarrow$ YAY $\leftarrow\leftarrow$

3. GUID Partition Table - GPT

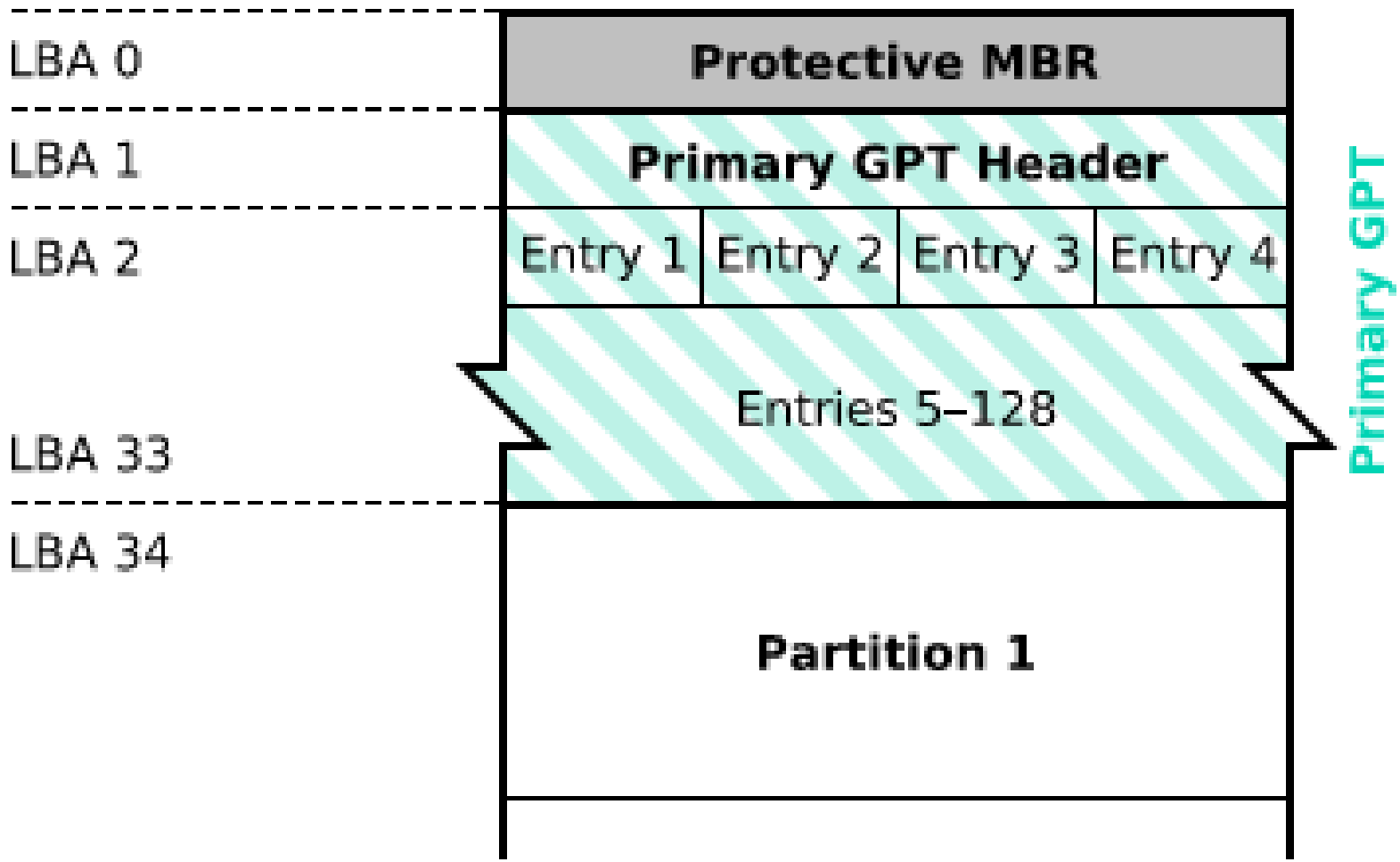


Image (c) wikipedia.org - Image used solely for illustration purposes

3. GUID Partition Table - GPT

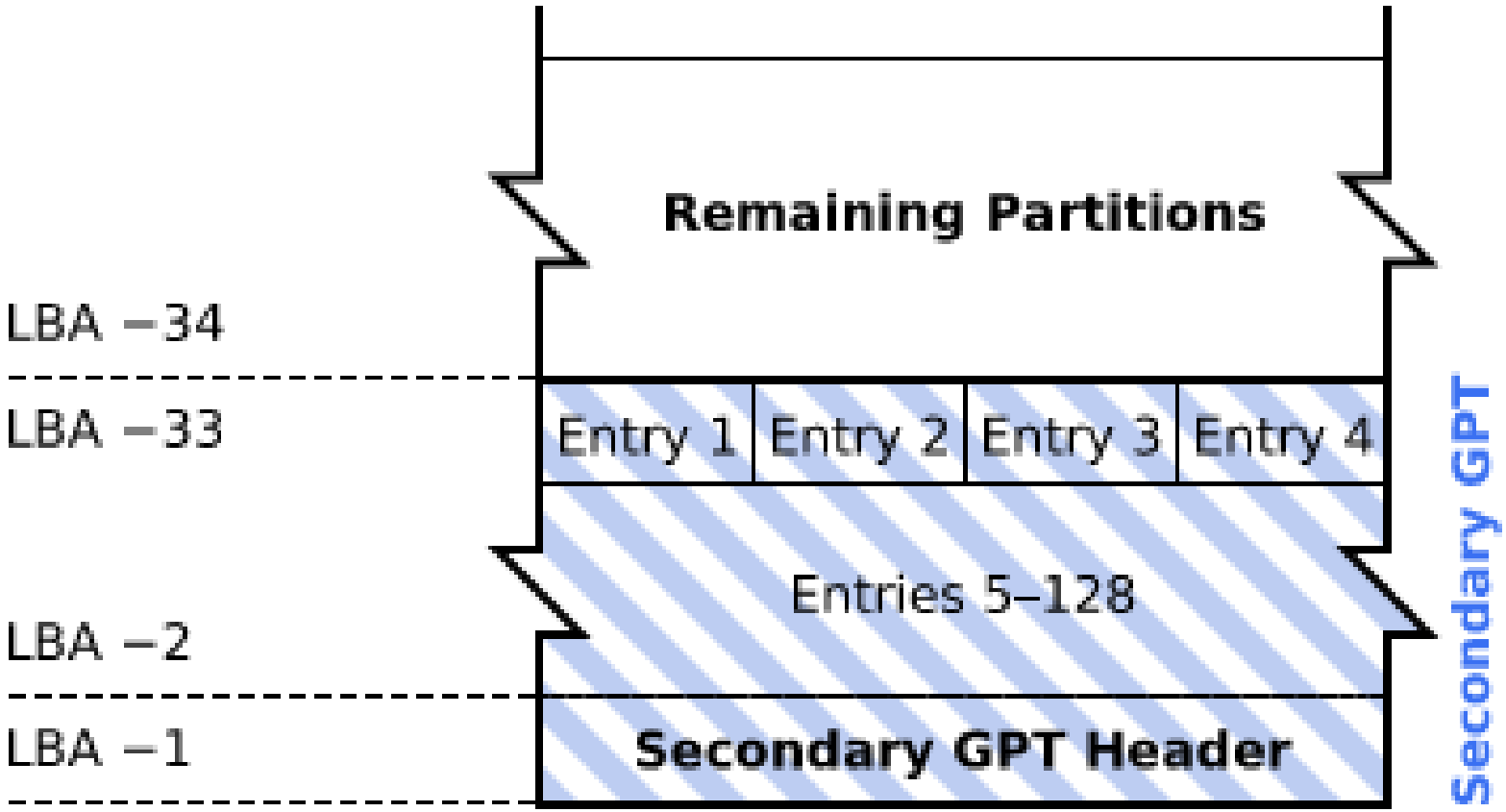


Image (c) wikipedia.org - Image used solely for illustration purposes

3. GUID Partition Table - GPT

Find back partition table Entry 1 & Entry 2

```
dd if=/dev/sdb skip=$((15974400 - 33)) | xxd | less

00000000: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000010: 995a b0ec f740 1549 8ee2 67d1 767d ff0b  .Z...@.I..g.v}..
00000020: 0008 0000 0000 0000 ffa7 7000 0000 0000  ..........p.....
00000030: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
00000040: 3100 0000 0000 0000 0000 0000 0000 0000  1...............
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000090: 8379 5b4f 1b77 5d43 bf13 a646 1c01 3e88  .y[O.w]C...F..>.
000000a0: 00a8 7000 0000 0000 ffb7 f300 0000 0000  ..p.............
000000b0: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
000000c0: 3200 0000 0000 0000 0000 0000 0000 0000  2...............
000000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
          .....
```

1

2

3

4. Analyzing Entry 1

According to the specifications

```
00000000: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000010: 995a b0ec f740 1549 8ee2 67d1 767d ff0b  .Z...@.I..g.v}..
00000020: 0008 0000 0000 0000 ffa7 7000 0000 0000  ..........p.....
00000030: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
00000040: 3100 0000 0000 0000 0000 0000 0000 0000  1...............
```

Offset	Offset	Lenght	Description
0	0	16	Partition type GUID
16	10	16	Unique partition GUID
32	20	8	First LBA (0008 0000 0000 0000) -> 0000 0000 0000 0800 -> 2048
40	28	8	Last LBA (ffa7 7000 0000 0000) -> 0000 0000 0070 a7ff -> 7383039

Read the sectors from partition start

```
dd if=/dev/sdb skip=2048 | xxd | less
```

```
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
.....
```

4. Analyzing Entry 2

According to the specifications

```
00000080: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000090: 8379 5b4f 1b77 5d43 bf13 a646 1c01 3e88  .y[O.w]C...F..>.
000000a0: 00a8 7000 0000 0000 ffb7 f300 0000 0000  ..p.............
000000b0: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
000000c0: 3200 0000 0000 0000 0000 0000 0000 0000  2...............
```

Offset	Offset	Lenght	Description
0	0	16	Partition type GUID
16	10	16	Unique partition GUID
32	20	8	First LBA (00a8 7000 0000 0000) -> 0000 0000 0070 a800 -> 7383040
40	28	8	Last LBA (ffb7 f300 0000 0000) -> 0000 0000 00f3 b7ff -> 15972351

Read the sectors from partition start

```
dd if=/dev/sdb skip=7383040 | xxd | less
```

```
00000000: 4c55 4b53 babe 0002 0000 0000 0000 4000  LUKS..........@.
00000010: 0000 0000 0000 0003 0000 0000 0000 0000  ................
*
```

5.1 GPT Repair: Play-Script

- Create a generic Protective MBR
  - Copy empty sector LBA 0 to disk
  - Create a single partition table entry protectiing the disk
  - Write the sector back from disk to LBA 0

- Recover Primary GPT Header
  - Read sector LBA -1 to disk
  - Modify some values: Secondary GPT $\rightarrow$ Primay GPT
  - Write this sector back to LBA 1

- Recover partition table entries
  - Read sector LBA -33 to disk
  - Write this sector back to LBA 2

5.2 GPT Repair: Protective MBR

Offset	Lenght	Description				
1be	64	Partition Table:	4 possible partitions 16 byte per partition			
1c2	1	Partition Type:	0b FAT32 07 NTFS/exFAT ee GPT			
				Big Endian		Little Endian
1c6	4	Starting LBA	1	—> 0000 0001	—>	0100 0000
1ca	4	Size in sectors	15974399	—> 00f3 bfff	—>	ffbf f300
1fe	2	Signature	55aa			

```
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
      .....
      .....
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001c0: 0000 ee00 0000 0100 0000 ffbf f300 0000  ................
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

```
dd if=/dev/sdb of=mbr skip=0 count=1
dd if=mbr of=/dev/sdb seek=0 count=1
```

5.3 GPT Repair: Primary GPT Header

```
dd if=/dev/sdb of=gpth skip=$((15974400 - 1)) count=1
```

Offset	Lenght	Description
0	8	EFI PART
8	4	Revision (0000 0100)
c	4	Header size (5c00 0000) -> 92 bytes
10	4	CRC32: (a101 7b1d) —> bc4f 886a
18	8	LBA of this header (ffbf f300 0000 0000) —> 0100 0000 0000 0000
20	8	LBA of other header (0100 0000 0000 0000) —> ffbf f300 0000 0000
48	8	LBA Partition Table (dfbf f300 0000 0000) —> 0200 0000 0000 0000

```
00000000: 4546 4920 5041 5254 0000 0100 5c00 0000  EFI PART....\...
00000010: a101 7b1d 0000 0000 ffbf f300 0000 0000  ..{.............
00000020: 0100 0000 0000 0000 2200 0000 0000 0000  ........".......
00000030: debf f300 0000 0000 c6aa d2d4 5971 2f42  ............Yq/B
00000040: b5bc 520a c650 ece1 dfbf f300 0000 0000  ..R..P..........
00000050: 8000 0000 8000 0000 e156 0e4d 0000 0000  .........V.M....
     .....
```

```
dd if=gpth of=/dev/sdb seek=1 conv=notrunc
```

5.3 GPT Repair: Primary GPT Header

```
dd if=/dev/sdb of=gpth skip=$((15974400 - 1)) count=1
```

Offset	Lenght	Description			
0	8	EFI PART			
8	4	Revision (0000 0100)			
c	4	Header size (5c00 0000) -> 92 bytes			
10	4	CRC32:	(a101 7b1d)	—>	bc4f 886a
18	8	LBA of this header	(ffbf f300 0000 0000)	—>	0100 0000 0000 0000
20	8	LBA of other header	(0100 0000 0000 0000)	—>	ffbf f300 0000 0000
48	8	LBA Partition Table	(dfbf f300 0000 0000)	—>	0200 0000 0000 0000

```
00000000: 4546 4920 5041 5254 0000 0100 5c00 0000  EFI PART....\...
00000010: bc4f 886a 0000 0000 0100 0000 0000 0000  .O.j............
00000020: ffbf f300 0000 0000 2200 0000 0000 0000  ........".......
00000030: debf f300 0000 0000 c6aa d2d4 5971 2f42  ............Yq/B
00000040: b5bc 520a c650 ece1 0200 0000 0000 0000  ..R..P..........
00000050: 8000 0000 8000 0000 e156 0e4d 0000 0000  .........V.M....
     .....
```

```
dd if=gpth of=/dev/sdb seek=1 conv=notrunc
```

5.4 GPT Repair: Partition Table Entry 1 & Entry 2

Before Operation

```
dd if=/dev/sdb skip=2 count=1 status=none | xxd | less
```

```
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
*
```

```
dd if=/dev/sdb of=/dev/sdb skip=$((15974400 - 33)) count=1 seek=2 conv=notrunc
```

After Operation

```
00000000: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000010: 995a b0ec f740 1549 8ee2 67d1 767d ff0b  .Z...@.I..g.v}..
00000020: 0008 0000 0000 0000 ffa7 7000 0000 0000  ..........p.....
00000030: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
00000040: 3100 0000 0000 0000 0000 0000 0000 0000  1...............
00000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000080: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000090: 8379 5b4f 1b77 5d43 bf13 a646 1c01 3e88  .y[O.w]C...F..>.
000000a0: 00a8 7000 0000 0000 ffb7 f300 0000 0000  ..p.............
000000b0: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
000000c0: 3200 0000 0000 0000 0000 0000 0000 0000  2...............
*
```

5.5 GPT Repair: Summary

```
dd if=/dev/sdb count=3 status=none | xxd | less
```

```
00000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
      .....
      .....
000001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001c0: 0000 ee00 0000 0100 0000 ffbf f300 0000  ................
000001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
000001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
00000200: 4546 4920 5041 5254 0000 0100 5c00 0000  EFI PART....\...
00000210: bc4f 886a 0000 0000 0100 0000 0000 0000  ..{.............
00000220: ffbf f300 0000 0000 2200 0000 0000 0000  ........".......
00000230: debf f300 0000 0000 c6aa d2d4 5971 2f42  ............Yq/B
00000240: b5bc 520a c650 ece1 0200 0000 0000 0000  ..R..P..........
00000250: 8000 0000 8000 0000 e156 0e4d 0000 0000  .........V.M....
      .....
      .....
00000470: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000480: a2a0 d0eb e5b9 3344 87c0 68b6 b726 99c7  ......3D..h..&..
00000490: 8379 5b4f 1b77 5d43 bf13 a646 1c01 3e88  .y[O.w]C...F..>.
000004a0: 00a8 7000 0000 0000 ffb7 f300 0000 0000  ..p.............
000004b0: 0000 0000 0000 0000 6400 6900 7300 6b00  ........d.i.s.k.
000004c0: 3200 0000 0000 0000 0000 0000 0000 0000  2...............
      .....
      .....
```

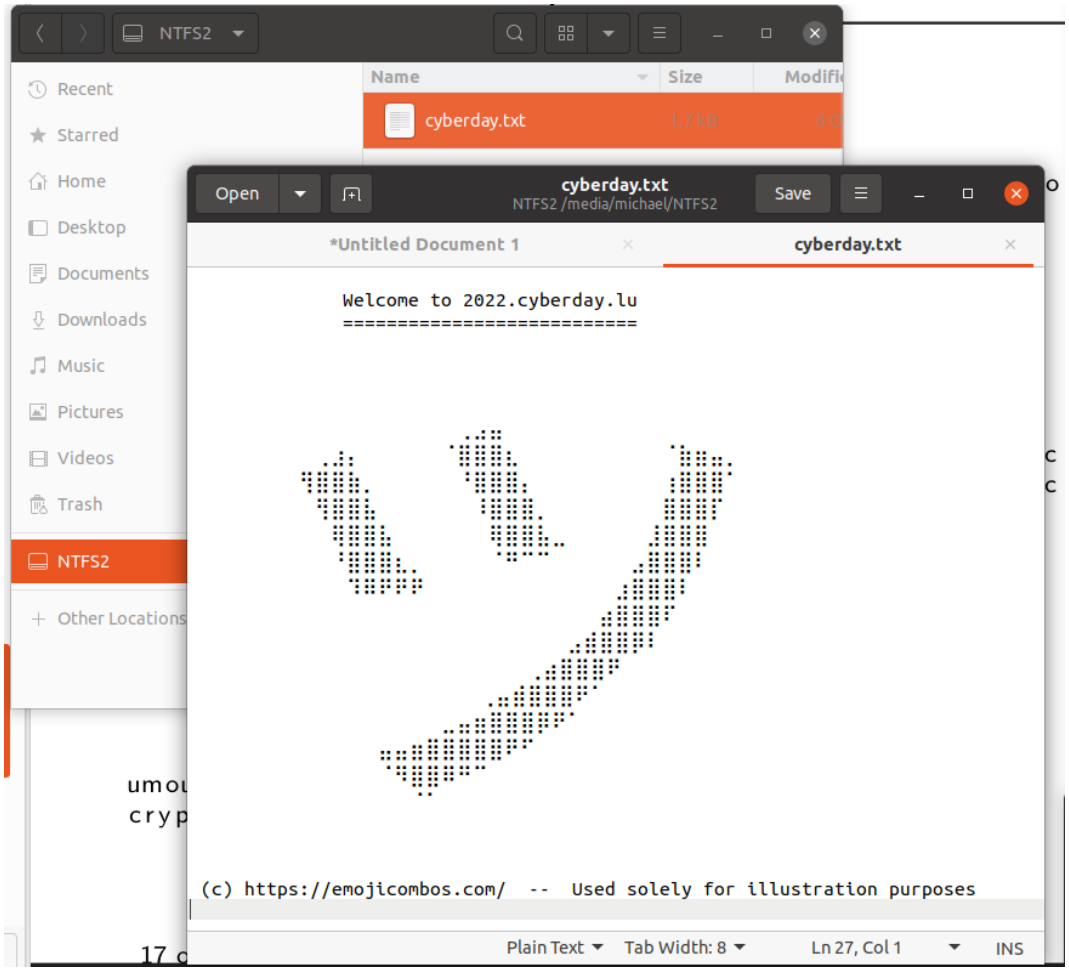
6. Test the results

```
fdisk -l /dev/sdb

Disk /dev/sdb: 7,63 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: gpt
Disk identifier: D4D2AAC6-7159-422F-B5BC-520AC650ECE1

Device       Start      End Sectors  Size Type
/dev/sdb1     2048  7383039 7380992  3,5G Microsoft basic data
/dev/sdb2  7383040 15972351 8589312  4,1G Microsoft basic data
```

```
cryptsetup open /dev/sdb2 NTFS_2
Enter passphrase for /dev/sdb2:
mount -o ro /dev/mapper/NTFS_2 /media/michael/ntfs/
```

```
umount /media/michael/ntfs/
cryptsetup close NTFS_2
```

6. Test the results



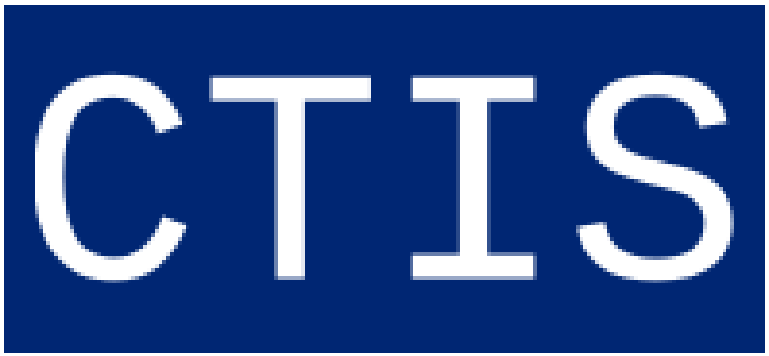
Partition NTFS2 is fully repaired

Safe the Date

CTI Summit & HACK.LU

Week 42 October 2023

16.10.2023 – 20.10.2023



Recovering data from a wiped disk

A manual approach

CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:CLEAR*

info@circl.lu

2023-01-31