Recovering data from a wiped disk

A manual approach

CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:CLEAR*

info@circl.lu

2022-10-13

1. Story behind

- Insider Threat
  - Very common
  - Very dangerous
  - All you need: One unhappy employee
- Empolyee
  - Execute critical attacks
  - Wipe disk of his computer
- Wiping big disk
  - Is very time consuming
  - Maybe somebody interupt it after a while
- This presentation
  - Simulates this situation
  - We will recover data manually by hand

2 Analysis: 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 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 center 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

Investigate the last sector of the disk

```
dd if=/dev/sdb skip=15974399 | 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  ................
          .....
```

- We found a string
- `EFI PART`

  $\rightarrow$ Could it be a part GUID Partition Table?

  Quick online search: $\rightarrow$ yes it could

3. GUID Partition Table

- UEFI
  - Unified Extensible Firmware Interface
  - Introduced 1998 by Intel
  - Interface between firmware and Operating System
  - Replacing classical BIOS more and more...
  - Introduce GUID Partition Table - GPT

- GPT
  - GUID: Globally Unique Identifier
  - Getting rid of classical DOS MBR restrictions
    - Use 64 Bit for LBA addressing $\rightarrow$ 9.4 Zettabyte
    - Support 128 partitions
    - Partially backward compatibility with MBR
    - Provied a backup of the GPT header
      $\rightarrow\rightarrow$ WOOT? $\leftarrow\leftarrow$

3. GUID Partition Table

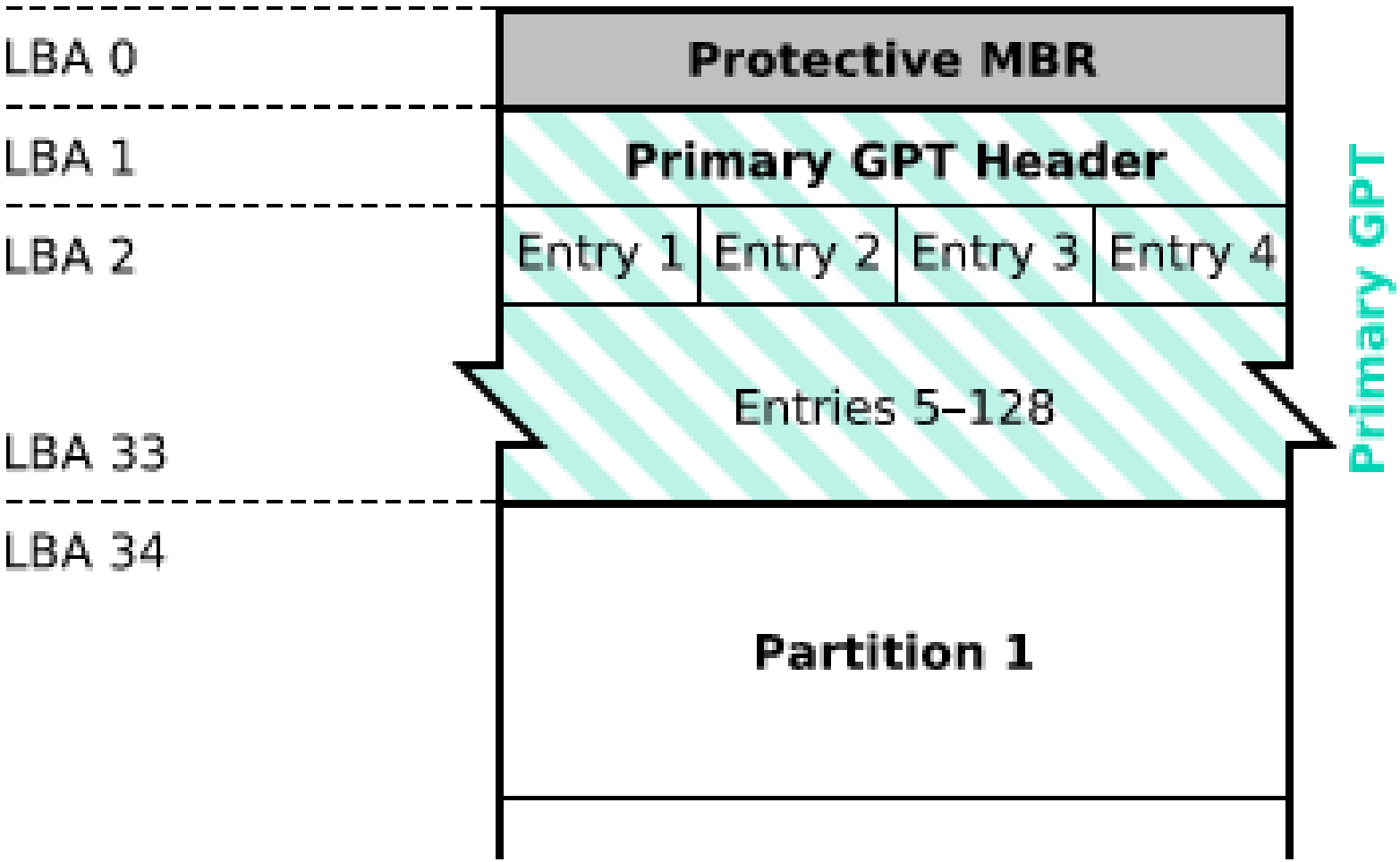

Image (c) wikipedia.org - Image used solely for illustration purposes

3. GUID Partition Table

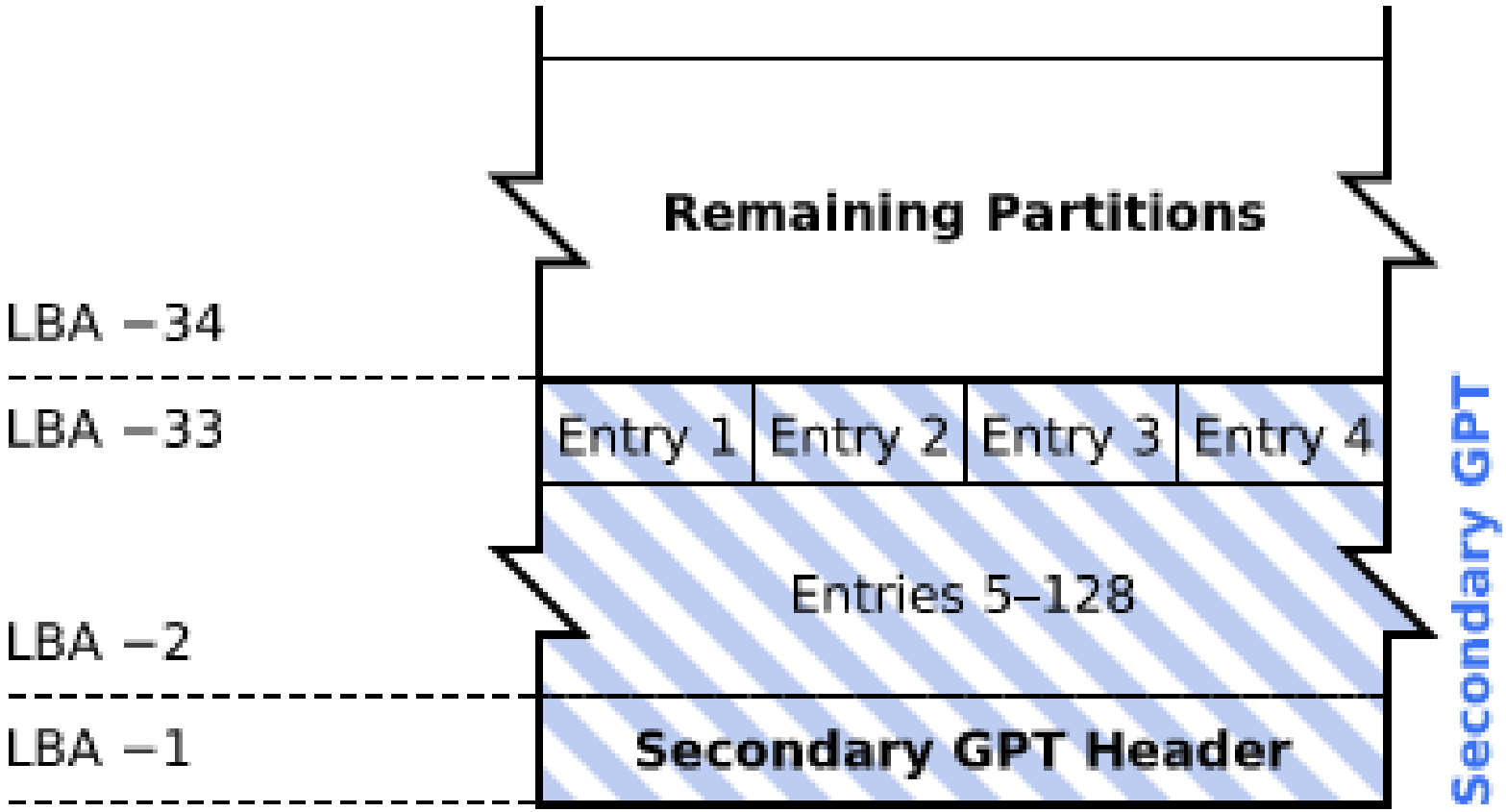

Image (c) wikipedia.org - Image used solely for illustration purposes

3. GUID Partition Table

Can we also find back partition table entries?

```
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  ................
          .....
```

4. Analyze partition table entry 1

```
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  ................
```

Offset	Offset	Lenght	Description
0	0	16	Partition type GUID
16	10	16	Unique partition GUID
32	20	8	First LBA (00 08 00 00 00 00 00 00) -> 0x0800 -> 2048
40	28	8	Last LBA (ff a7 70 00 00 00 00 00) -> 0x70a7ff -> 7383039

```
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. Analyze partition table entry 2

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

```
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  ................
```

Offset	Offset	Lenght	Description
0	0	16	Partition type GUID
16	10	16	Unique partition GUID
32	20	8	First LBA (00 a8 70 00 00 00 00 00) -> 0x70a800 -> 7383040
40	28	8	Last LBA (ff b7 f3 00 00 00 00 00) -> 0xf3b7ff -> 15972351

```
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. Data Recovery: Play-Script

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

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

- Recover Protective MBR
  - Read empty sector LBA 0 to disk
  - Create a simpe protective partition table entry
  - Write this sector back to LBA 0

6. Data Recovery: Partition table entries

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

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


dd if=/dev/sdb of=lba2 skip=$((15974400 - 33)) count=1


dd if=lba2 of=/dev/sdb seek=2 conv=notrunc


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

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...............
.....
```

7. Data Recovery: Primary GPT Header

```
dd if=/dev/sdb of=gpt2 skip=$((15974400 - 1)) count=1
```

```
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....
.....
```

Offset	Offset	Lenght	Description
0	0	8	EFI PART
8	8	4	Revision (00 00 01 00)
12	c	4	Header size (5c 00 00 00) -> 92 bytes
16	10	4	CRC32: (a1 01 7b 1d) -> bc 4f 88 6a
24	18	8	LBA of this header (ff bf f3 00 00 00 00 00) -> 00 00 00 00 00 f3 bf ff -> 15974399
32	20	8	LBA of other header (01 00 00 00 00 00 00 00) -> 00 00 00 00 00 00 00 01 -> 1
72	48	8	Start LBA: Partition Table (df bf f3 00 00 00 00 00) -> 00 f3 bf df -> 15974367

```
dd if=gpt1 of=/dev/sdb seek=1 conv=notrunc
```

7. Data Recovery: Primary GPT Header

```
dd if=/dev/sdb of=gpt2 skip=$((15974400 - 1)) count=1
```

```
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....
.....
```

Offset	Offset	Lenght	Description
0	0	8	EFI PART
8	8	4	Revision (00 00 01 00)
12	c	4	Header size (5c 00 00 00) -> 92 bytes
16	10	4	CRC32: (a1 01 7b 1d) -> bc 4f 88 6a
24	18	8	LBA of this header (ff bf f3 00 00 00 00 00) -> 00 00 00 00 00 f3 bf ff -> 15974399
32	20	8	LBA of other header (01 00 00 00 00 00 00 00) -> 00 00 00 00 00 00 00 01 -> 1
72	48	8	Start LBA: Partition Table (df bf f3 00 00 00 00 00) -> 00 f3 bf df -> 15974367

```
dd if=gpt1 of=/dev/sdb seek=1 conv=notrunc
```

8. Data Recovery: 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
1c6	4	Starting LBA	00 00 00 01 01 00 00 00
1ca	4	Size in sectors	15974399 00 f3 bf ff ff bf f3 00
1fe	2	Signature	55 aa

```
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.
```

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

9. Data Recovery: Review the results

```
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...............
      *
      *
```

9. Data Recovery: Review 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
```

9. Data Recovery: Review the results

Partition NTFS2 is fully repaired

10. End Note:

Forensic tools will spot the secondary GPT without problem

There are tools for automatically repair, for sure

We did this repair manually by intention

An analyst should

- → Be able to do it manually
- → Do it manually 2 or 3 times to learn how it works
- → Know what to do, if your tools don't work

Recovering data from a wiped disk

A manual approach

CIRCL
Computer Incident
Response Center
Luxembourg

CIRCL *TLP:CLEAR*

info@circl.lu

2022-10-13