

Systèmes de fichiers

Mars 2023

Organisation de journée

- 9h30-11h30 : Cours
- 12h30-15h10 : 2 TDs, un pause, 1 TDs
- 15h10-15h30 : **Evaluation**

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
++++ PhysicalDrive0: UBOX HARDDISK ++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rounix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

Motivation de l'analyse forensique au niveau stockage

- Déceler toutes les activités de niveau système, middleware, applicatif ou utilisateur utilisant le disque
 - Fourni un grand nombre d'informations temporelles (timestamp): MACB
 - Un malware résiste en général au reboot, donc
 1. Il est souvent stocké sur disque
 2. Il est lancé par l'OS, donc il existe un fichier pour cela (y compris parmi ceux du registre)
- ⇒ Détection des rootkits

Pourquoi s'intéresser aux systèmes de fichiers ?

Analyse des fichiers effacés, des métadonnées (journaux et index), « slackspace », ...

Il faut connaître l'organisation des fichiers et répertoires sur le disque pour parfois, reconstituer de l'information partiellement effacée ou volontairement cachée

Niveaux d'abstraction d'un système de fichiers

1. Niveau physique (mémoire flash* ou HDD)
2. Volume logiques (LVM)
3. Niveau logique: partitions (décrit dans la MBR ou GPT)
4. Niveau données: cluster= Groupe de secteurs
5. Système de fichiers (NTFS, FAT, EXT4): métadonnées
6. Fichiers et répertoires

* <https://articles.forensicfocus.com/2016/04/20/ssd-and-emmc-forensics-2016/>

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
+++++ PhysicalDrive0: UBOX HARDDISK +++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rounix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

Master Boot Record

- Depuis PC XT (1983, partition FAT12)
- Chargé par le BIOS
- Contient
 - les informations pour accéder au stockage qui contient le système d'exploitation
 - La table des partitions
- Limitations
 - 4 partitions. Les partitions étendues ont permis de contourner le problème
 - Disque $\leq 2\text{TB}$, avec secteurs de 512 octets
 - Disque $\leq 16\text{TB}$, avec secteurs de 4k ([Advanced Format](#))
- Évolution de la paire BIOS/MBR vers UEFI/GPT, notamment pour standardiser le démarrage, en particulier pour la sécurité ([Secure Boot](#))

Master Boot Record

1. Jump Instruction / boot code
2. 32bits disk signature: 0xf6096c93

Partition entry #0 (offset 0x1be):

1. Status: 0x80 (bootable)
2. CHS address of first sector (obsolete)
3. Type de partition: 7 (NTFS/exFat)
4. CHS address of last sector (obsolete)
5. Logical Block Address (LBA) du premier secteur de la partition: 0x800
6. Nombre de secteurs dans la partition: 0x249ef800

Partition entry #1:

1. Status: 0 (non bootable)
2. CHS address of first sector (obsolete)
3. Type de partition: 7 (NTFS/exFat)
4. CHS address of last sector (obsolete)
5. Logical Block Address (LBA) du premier secteur de la partition: 0x249f0800
6. Nombre de secteurs dans la partition
7. Octets de synchro de fin de secteur: 0x55aa

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
000000000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00
000000000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00
000000000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10
000000000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00
000000000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09
000000000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74
000000000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00
000000000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13
000000000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00
000000000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE
0000000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	3B	E4
0000000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55
0000000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64
0000000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75
0000000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54
0000000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00
000000000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66
000000000110	53	66	55	66	68	00	00	00	66	68	00	7C	00	00	66	68
000000000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD
000000000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4
000000000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD
000000000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8
000000000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69
000000000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72
000000000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69
000000000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E
0000000001A0	67	20	6F	70	65	72	61	74	69	6F	67	20	73	79	73	74
0000000001B0	65	6D	00	00	00	63	7B	9A	F6	09	6C	93	9B	9A	80	20
0000000001C0	21	00	07	FE	FF	FF	00	08	00	00	00	F8	9E	24	00	FE
0000000001D0	FF	FF	07	FE	FF	FF	00	08	9F	24	00	00	09	3D	00	FE
0000000001E0	FF	FF	0F	FE	FF	FF	00	08	A8	61	00	78	38	87	00	00
0000000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA
000000000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Path	Value	Data
\\DosDevices\C:	REG_BINARY	f6 09 6c 93 00 10 10 00 00 00 00
\\DosDevices\D:	REG_BINARY	ab 9f 7a d6 00 7e 00 00 00 00 00
\\DosDevices\E:	REG_BINARY	a8 ef 5f 21 00 00 10 00 00 00 00
\\DosDevices\F:	REG_BINARY	ab 9f 7a d6 00 fe 08 e2 04 00 00
\\DosDevices\G:	REG_BINARY	ab 9f 7a d6 00 9e 0e d4 30 00 00
\\DosDevices\H:	REG_BINARY	ab 9f 7a d6 00 5e 16 e6 aa 00 00
\\DosDevices\I:	REG_BINARY	a8 ef 5f 21 00 00 10 3e 49 00 00
\\DosDevices\J:	REG_BINARY	5f 00 3f 00 3f 00 5f 00 55 00 53
\\DosDevices\K:	REG_BINARY	f6 09 6c 93 00 00 20 50 c3 00 00

MBR: Types de partition

Valeur type	Type
0x01	FAT12
0x0E	FAT16 (LBA)
0x0B	FAT32 (CHS)
0x0C	FAT32 (LBA)
0x0F/0x05	Extended (LBA)
0x07	NTFS & exFAT
0x83	Linux native
0x82	Linux swap
0xEE	EFI

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
++++ PhysicalDrive0: UBOX HARDDISK ++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rovnix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

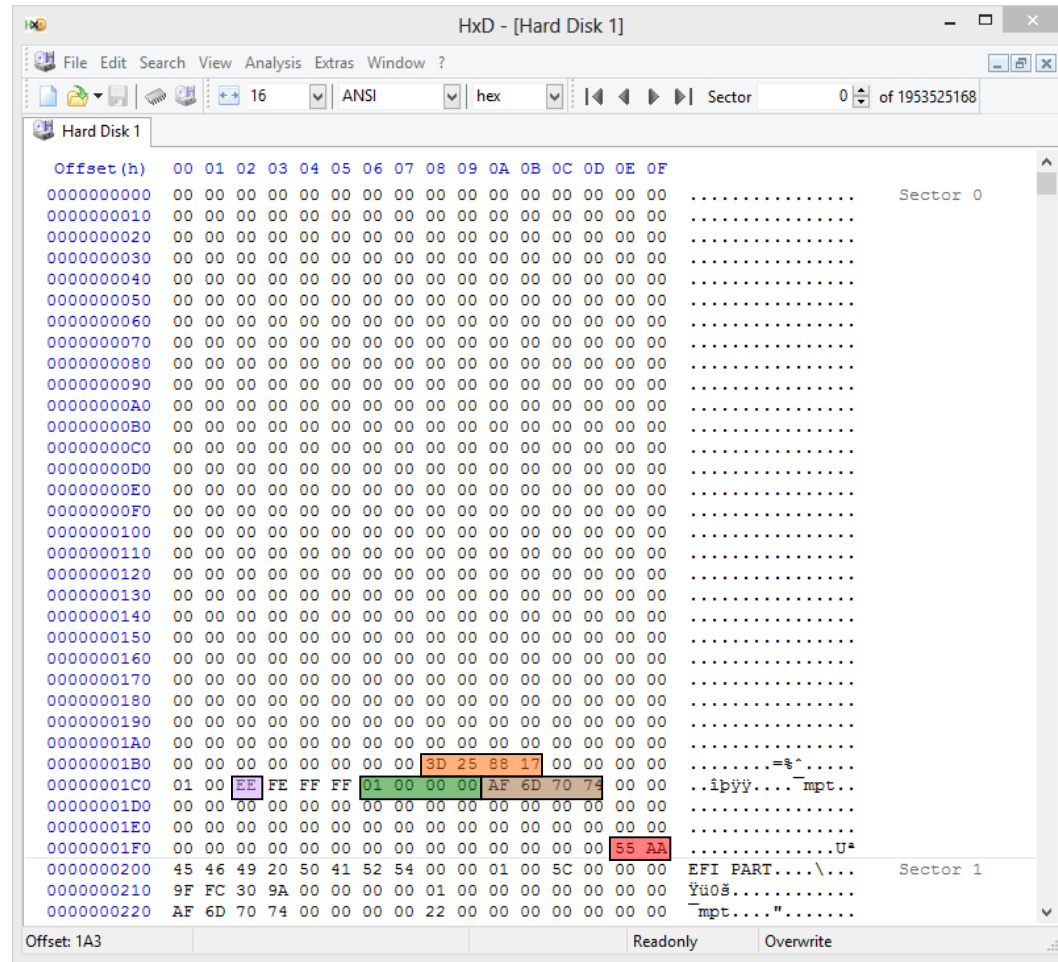
GUID Partition Table (GPT)

- Remplace la MBR à partir de Win8 et les Macs Intel
- 16k pour stocker les descriptions des partitions, avec 128 octets par entrée
- Fait partie de la spécification UEFI, qui remplace le BIOS
 - Utilise une partition FAT pour les applications EFI, lancées avant l'OS
- Cependant certains BIOS modernes permettent l'utilisation de la GPT au lieu d'un MBR

GPT: LBA#0 (protective MBR)

Pour des raisons de compatibilité,
il est défini une unique partition
qui couvre tout le disque physique

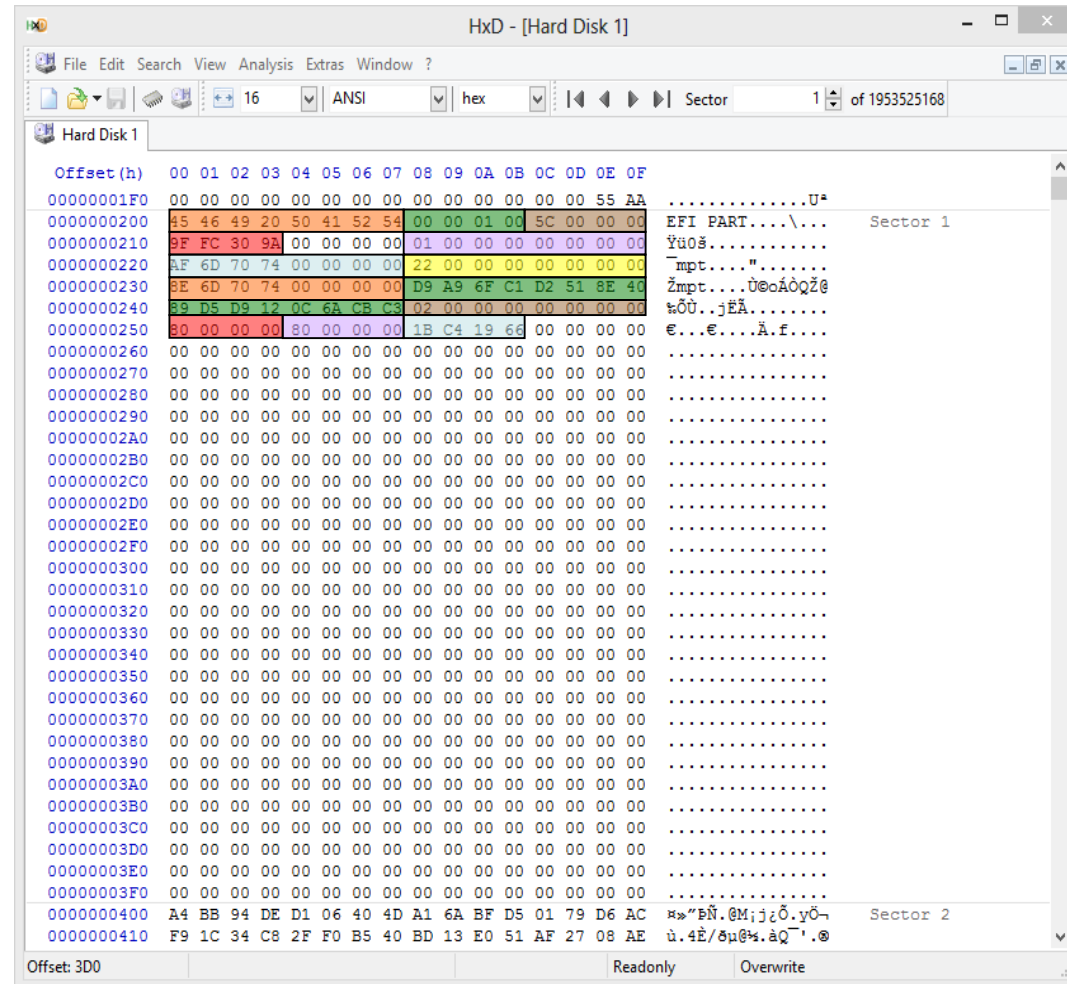
1. Pas de code lancé par le BIOS
2. 32bits disk signature
3. Type de partition: 0xee (EFI)
4. Début de la partition: 1
5. Taille de la partition:
0x74706daf =
1 000 204 885 504 octets
6. Synchro de fin de secteur



LBA#1: entête GPT

GPT= GUID Partition Table

- Signature: « EFI PART »
- Version: 1.0
- Taille de l'entête: 0x5c
- CRC32 de l'entête
- LBA de cet entête: 1
- LBA de l'entête de sauvegarde:
0x74706daf (fin du disque)
- Première LBA pour une partition:
0x22 = 34
- Dernière LBA pour une partition:
juste avant la copie de la GPT
- GUID du disque
- LBA de la table des partitions: 2
- Nombre d'entrée dans la table: 128
- Taille des entrées: 128
- CRC32 de la table



GPT, LBA#2: table des partitions (1)

Partition #0 (Windows Recovery Environment)

1. GUID du type de partition
de94bba4-06d1-4d40-a16a-bfd50179d6ac
2. GUID de la partition
3. Première LBA: 0x800
4. Dernière LBA (include): 0xc87ff (400Mo)
5. Attributs de la partition
6. Nom (36 caractères UTF16LE)

Partition #1 (EFI)

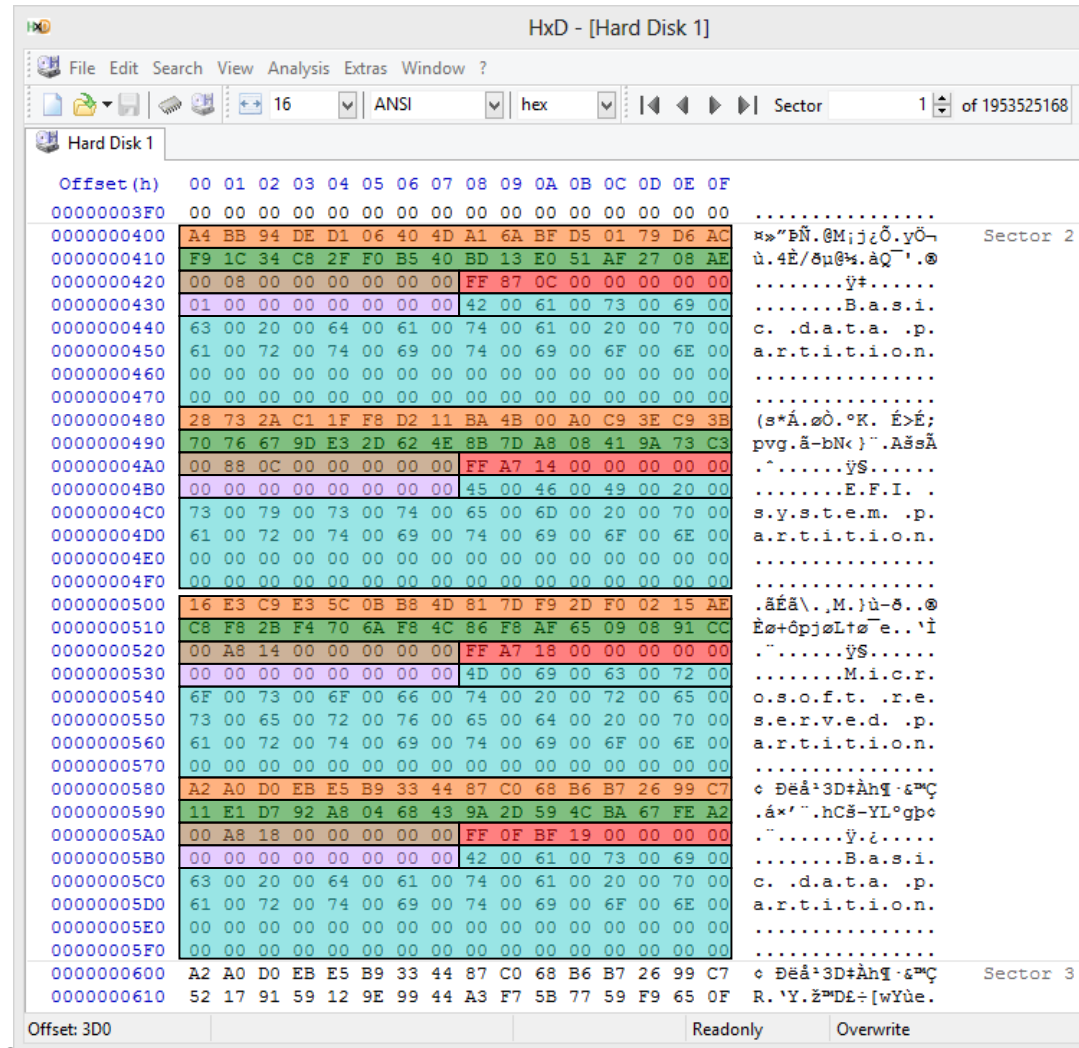
1. GUID du type de partition
C12a7328-f81f-11d2-ba4b-00a0c93ec93b
2. GUID de la partition
3. Première LBA: 0xc8800
4. Dernière LBA (include): 0x14a7ff
5. Attributs de la partition
6. Nom (36 caractères UTF16LE)

Partition #2 (MSR=Microsoft Reserved)

1. GUID du type de partition
2. GUID de la partition
3. Première LBA: 0x14a800
4. Dernière LBA (include): 0x18a7ff
5. Attributs de la partition

Partition #3 (Basic data partition, C: 250Go)

1. GUID du type de partition (NTFS)
2. GUID de la partition
3. Première LBA: 0x18a800
4. Dernière LBA (include)
5. Attributs de la partition



GPT, LBA#2: table des partitions (2)

Partition #4 (data, 600 Go)

1. GUID du type de partition
2. GUID de la partition
3. Première LBA
4. Dernière LBA (inclusive)
5. Attributs de la partition
6. Nom (36 caractères UTF16LE)

Partition #5 (recovery, NTFS)

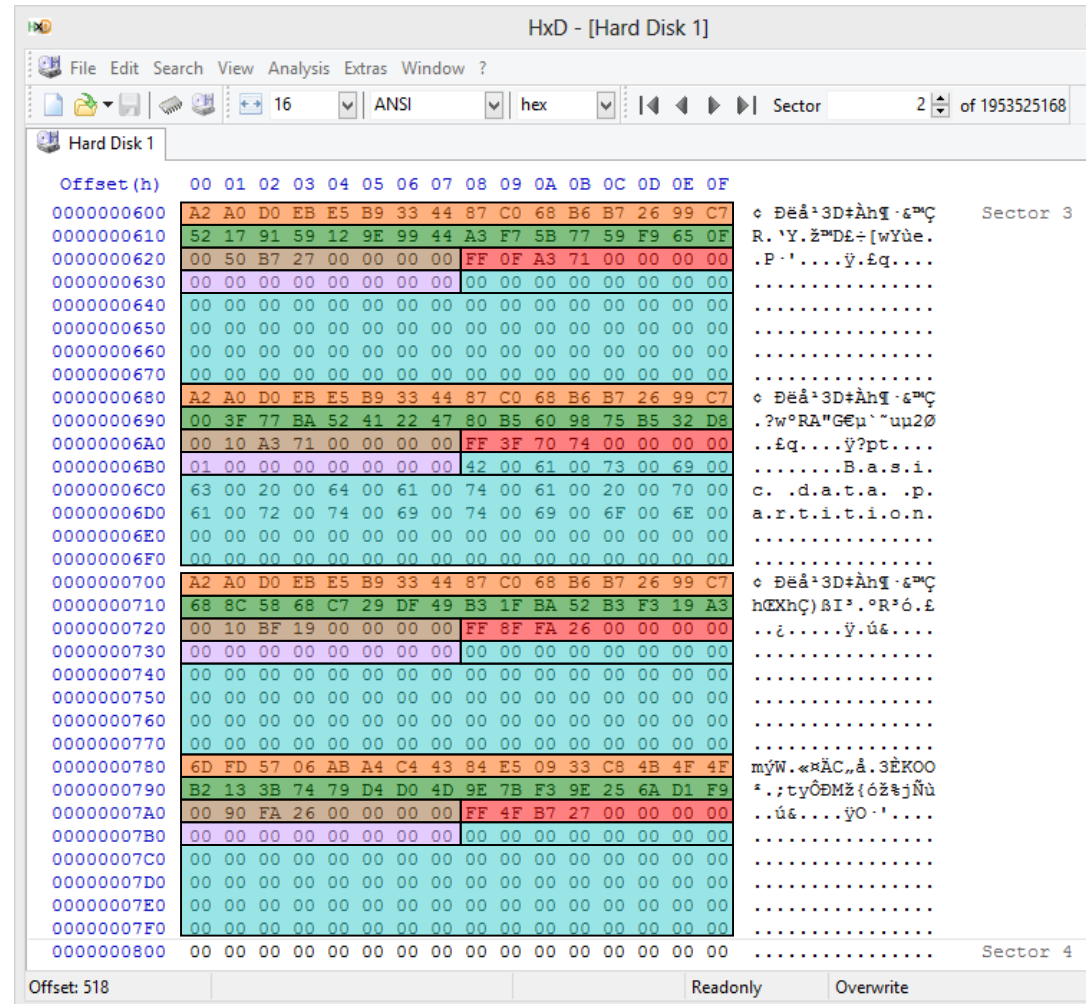
1. GUID du type de partition
2. GUID de la partition
3. Première LBA:
4. Dernière LBA (inclusive)

Partition #6 (Linux, EXT4, 105 Go)

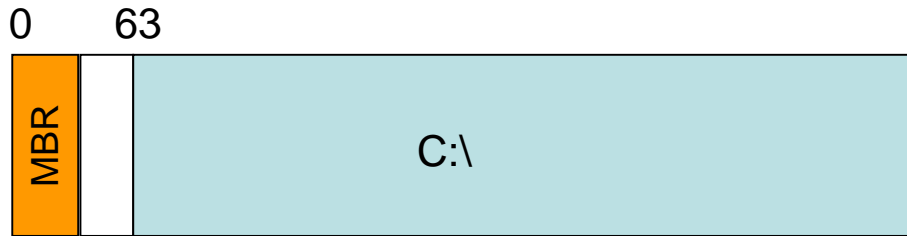
1. GUID du type de partition
now: 0fc63daf-8483-4772-8e79-3d69d84777de4
2. GUID de la partition
3. Première LBA:
4. Dernière LBA (inclusive)

Partition #7 (Linux swap)

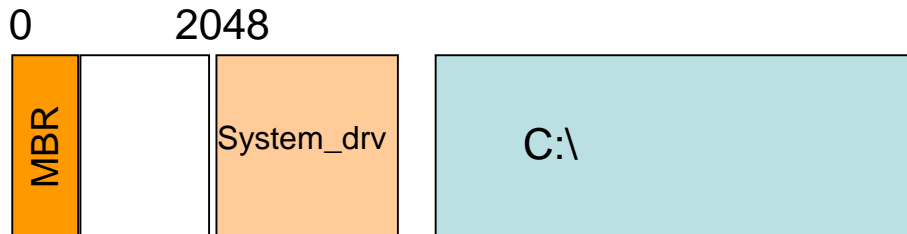
1. GUID du type de partition
0657fd6d-43c4-84e5-0933c84b4f4f
2. GUID de la partition
3. Première LBA:
4. Dernière LBA (inclusive)



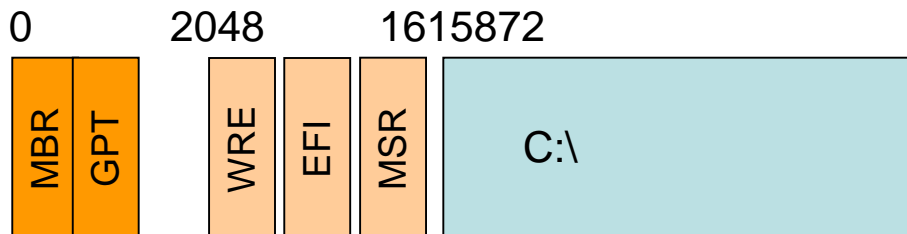
Partitionnements typiques



XP / 2000 : première partition habituellement au secteur 63.



Win7 / Vista : première partition habituellement au secteur 2048 (1Mo)



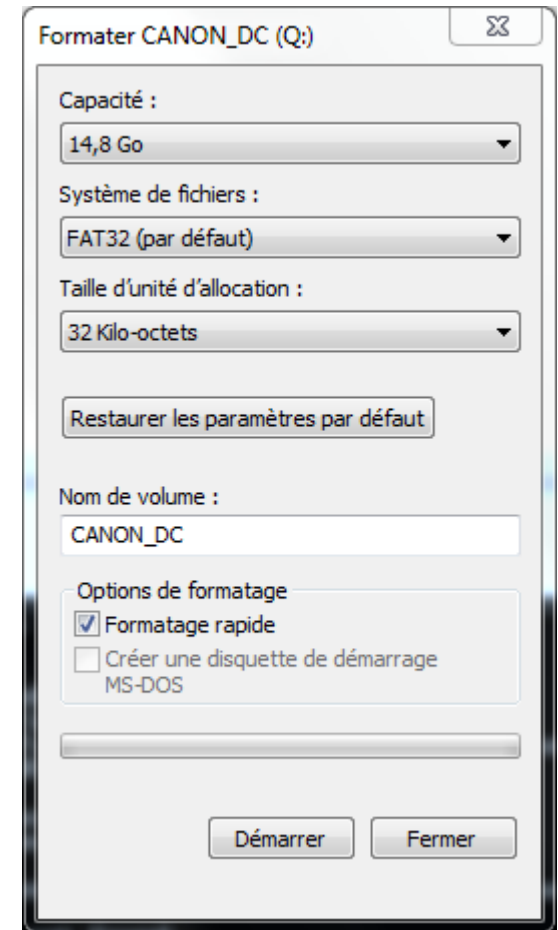
Win8/10 + UEFI

Agenda

- Physique/logique/partition
- Partitions
 - MBR
 - GPT
- **Système de fichiers**
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit

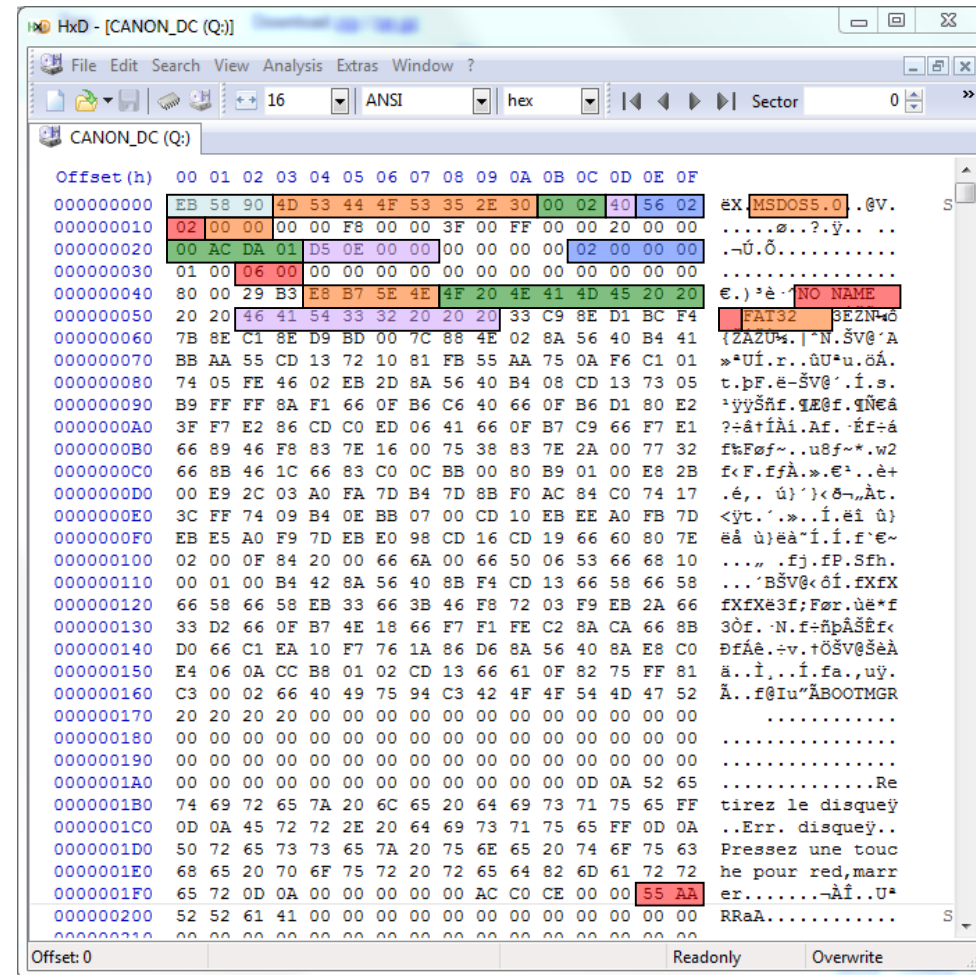
FAT32

- Depuis Windows 95 OSR
- « File Allocation Table », évolution de FAT12 et FAT16. Créé en 1977 par Bill Gates
- Taille des clusters (groupe de secteurs): de 512 octets à 32Ko
- « FAT32 », pour 2^{32} clusters, mais 2^{28} seulement en pratique
- Taille maximale
 - Windows: disque de 32Go max
 - Théorique (limitation MBR): partition de 2To
- Taille maximale d'un fichier: 4Go (attention lors de la création d'un dump mémoire sur clé USB si RAM > 4Go)



FAT32: Secteur de boot

1. Jump Instruction
2. OEM ID: « MSDOS5.0 »
3. Taille secteur: 0x200 = 512
4. Nb secteurs / cluster: 0x40 = 64.
64*512=32ko
5. Nb secteurs réservés: 0x256. Offset de la FAT=0x256*512 = 0x4ac00
6. Nb table FAT: 2
7. Entrées à la racine: 0
8. Nb total de secteurs: 0x1daac00
9. Nb secteurs dans chaque FAT: 0xed5
10. Numéro premier cluster de la racine: 2
11. Numéro du secteur de la copie du secteur de boot: 6. Offset=0xc00
12. Volume ID: 5EB7-E8B3
13. Volume name: « NO NAME »
14. Système ID: « FAT32 »
15. Octets de synchro de fin de secteur: 0x55aa



Référence:

<http://msdn.microsoft.com/en-us/windows/hardware/gg463080.aspx>

(c) 2023 Laurent Clévy

20

FAT32: structure

MBR

...

Secteur de boot

...

FAT1 (après secteurs réservés)

FAT2

Zone des clusters (premier cluster= cluster#2)

FAT32: 1ere FAT

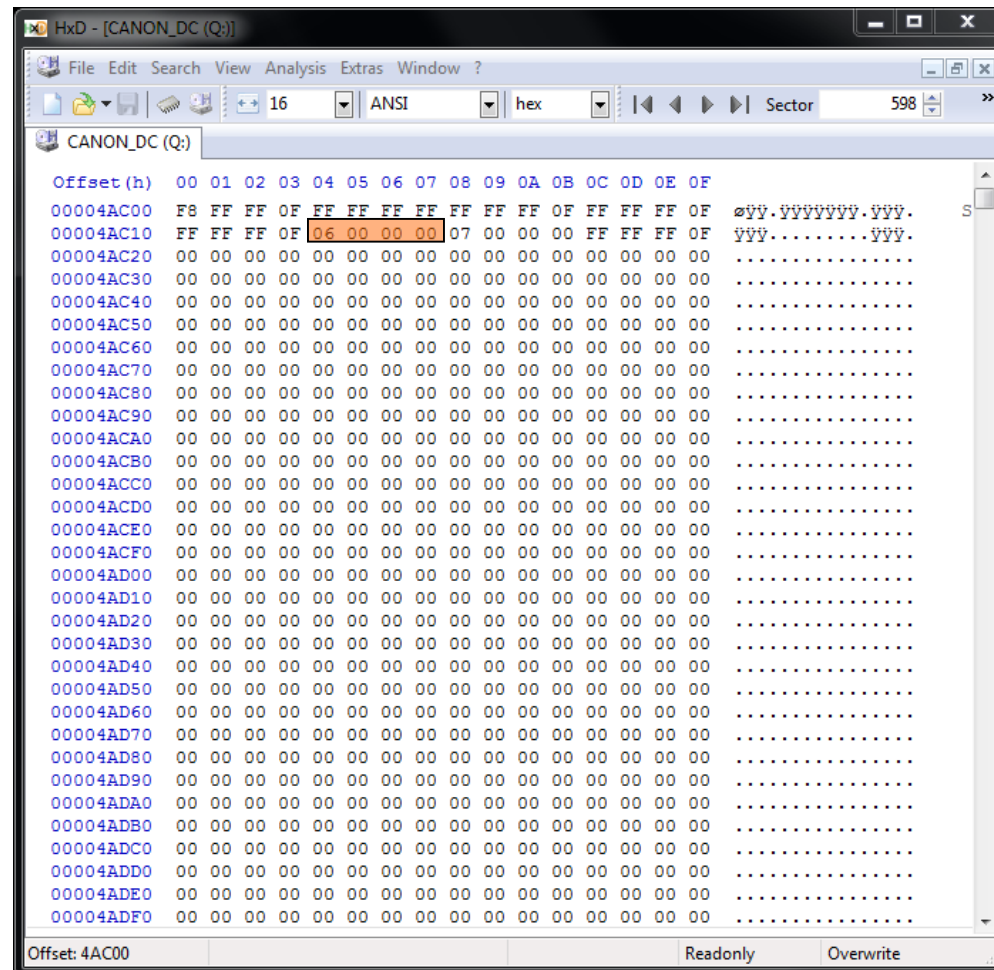
Offset de la FAT = $0x256 * 512 = 0x4ac00$

FAT³², donc les entrées de la FAT font 32bits

Indique si le cluster est libre (00000000) ou alloué (0x0ffffff8, 0xffffffff ou 0x0fffffff)

Chaîne les clusters d'un même fichier

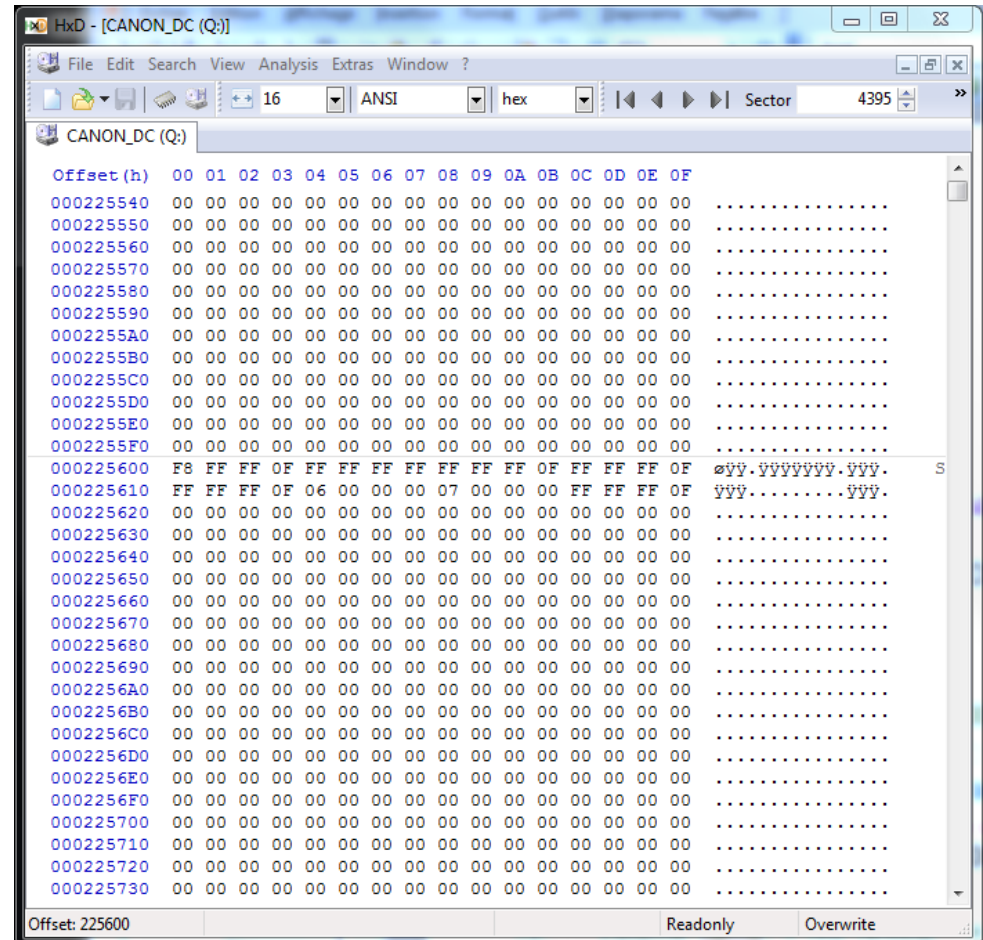
La première entrée est #0,
donc en orange, l'entrée pour le cluster 5



FAT32: 2eme FAT

Fin de la 1^{ère} FAT et début de la 2^{ème}:
 $0x4ac00 + 0xed5 \times 512 = 0x225600$

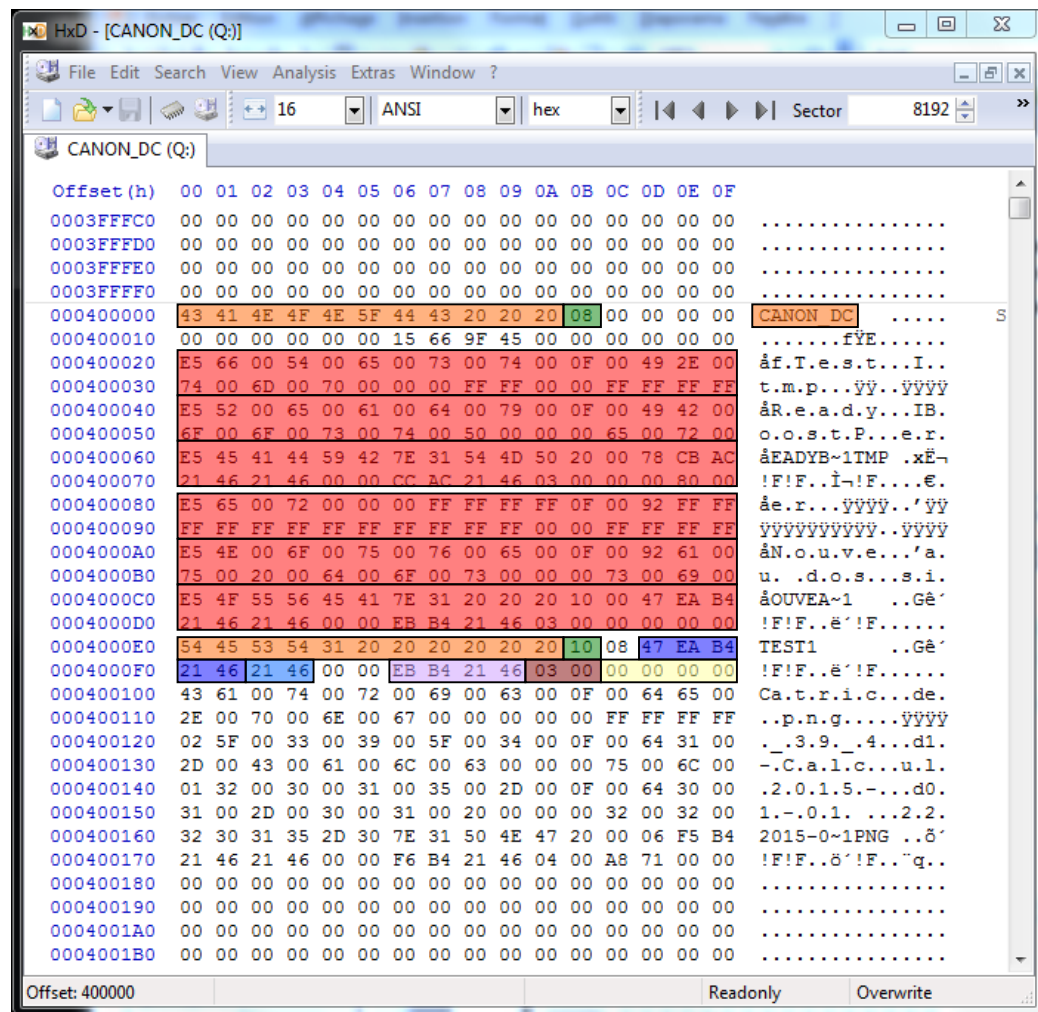
Fin de la 2eme FAT:
 $0x225600 + 0xed5 \times 512 = 0x400000$



FAT32: répertoire racine (1)

Cluster2 = 0x400000 (après la 2^e FAT)

1. Nombre du volume: « canon_dc »
car attribut = 8 à l'offset 0xB
2. 2 entrées effacées avec noms longs,
car le premier octet à la valeur spéciale
0xe5
car attribut = 0xF à l'offset 0xB
3. Nom du répertoire: « test1 »,
attribut: 0x10 (sous répertoire)
4. Heure et date de création (à 10ms prêt)
5. Date dernier accès
6. Heure et de date de dernière
modification
7. Numéro du premier cluster du fichier: 3
8. Taille du fichier: 0



FAT32: répertoire racine (2)

1. Nombre du fichier long
car attribut = 0xF à l'offset 0xB
2. Heure et date de création (à 10ms prêt)
3. Date dernier accès
4. Heure et de date de dernière modification
5. Numéro du premier cluster du fichier: 4.
offset=0x410000
6. Taille du fichier: 0x71a8 = 29096
(1 cluster)

000400B0	75 00 20 00 64 00 6F 00 73 00 00 00 73 00 69 00	u. .d.o.s...s.i.
000400C0	E5 4F 55 56 45 41 7E 31 20 20 20 10 00 47 EA B4	ÀOUVEA~1 ..Gè'
000400D0	21 46 21 46 00 00 EB B4 21 46 03 00 00 00 00 00	!F!F..ë'!F.....
000400E0	54 45 53 54 31 20 20 20 20 20 20 10 08 47 EA B4	TEST1 ..Gè'
000400F0	21 46 21 46 00 00 EB B4 21 46 03 00 00 00 00 00	!F!F..ë'!F.....
00040100	43 61 00 74 00 72 00 69 00 63 00 0F 00 64 65 00	Ca.t.r.i.c...de.
00040110	2E 00 70 00 6E 00 67 00 00 00 00 00 00 FF FF FF FF	..p.n.g....ÿÿÿÿ
00040120	02 5F 00 33 00 39 00 5F 00 34 00 0F 00 64 31 00	..3.9...4...d1.
00040130	2D 00 43 00 61 00 6C 00 63 00 00 00 75 00 6C 00	..C.a.l.c...u.l.
00040140	01 32 00 30 00 31 00 35 00 2D 00 0F 00 64 30 00	..2.0.1.5.-...d0.
00040150	31 00 2D 00 30 00 31 00 20 00 00 00 32 00 32 00	1.-.0.1. ...2.2.
00040160	32 30 31 35 2D 30 7E 31 50 4E 47 20 00 06 F5 B4	2015-0~1PNG ..8'
00040170	21 46 21 46 00 00 F6 B4 21 46 04 00 A8 71 00 00	!F!F..ë'!F..q..
00040180	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00040190	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000401A0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000401B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Offset: 400000 Readonly Overwrite

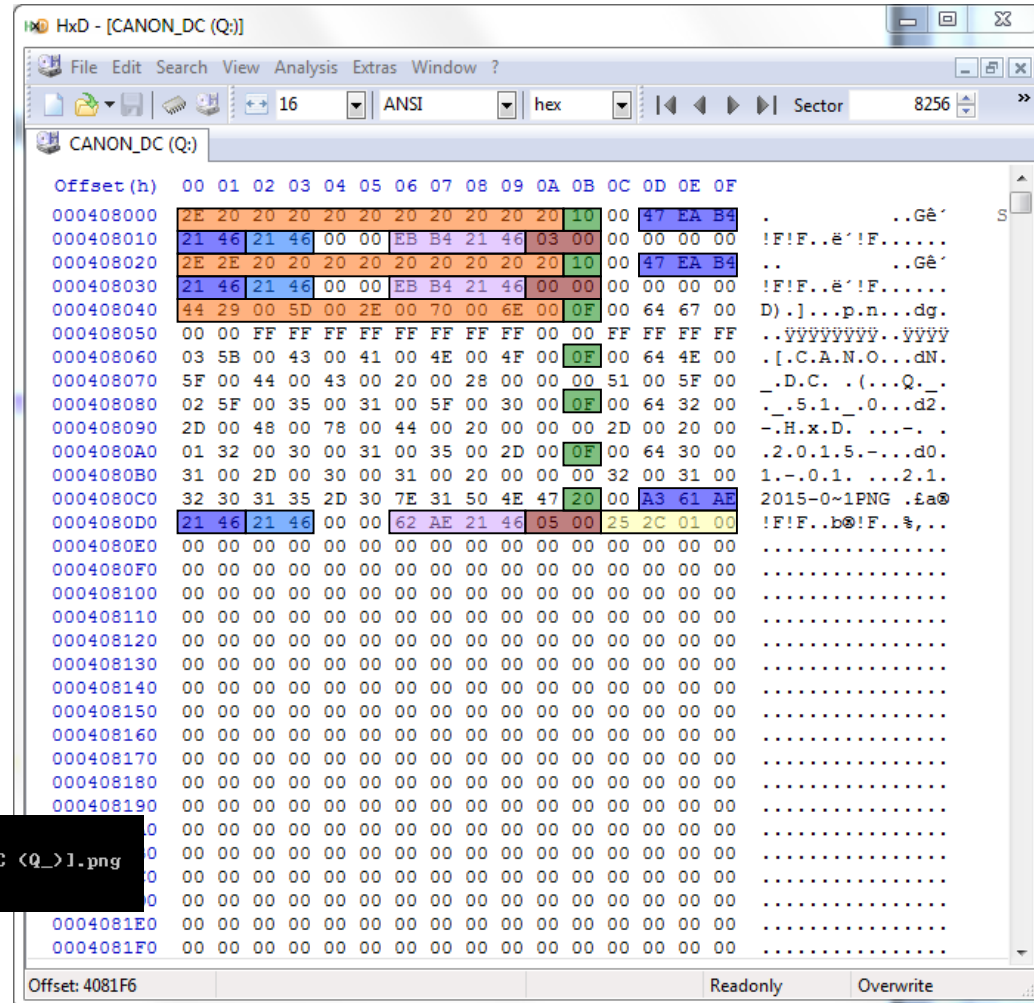
```

Répertoire de Q:\
01/01/2015 22:39 <REP> test1
01/01/2015 22:39 29 096 2015-01-01 22_39_41-Calculatrice.png
1 fichier(s) 29 096 octets
1 Rép(s) 15 922 954 240 octets libres
  
```

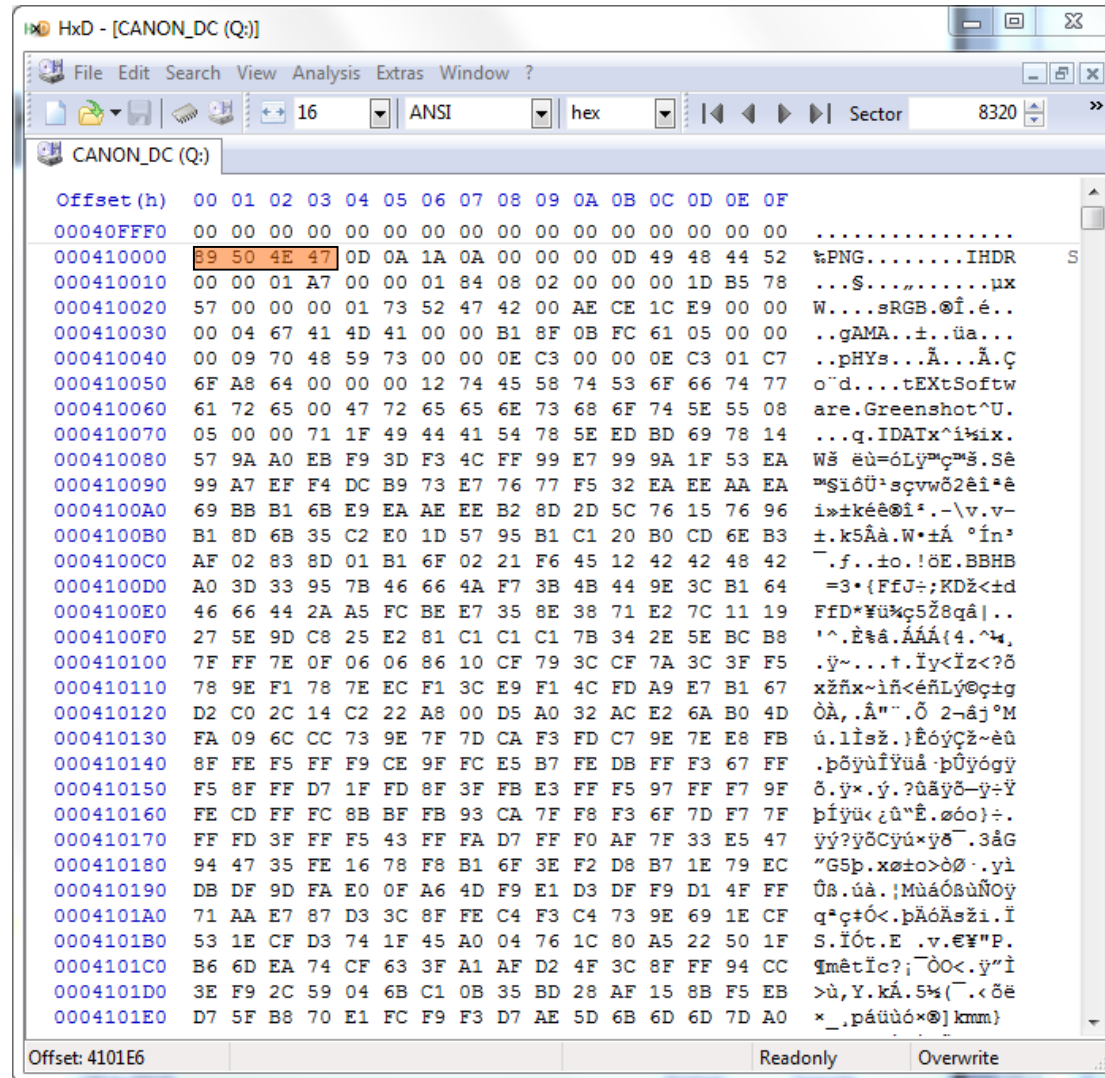
FAT32: répertoire « test1 »

1. Répertoire: « . », au cluster 3
attribut = 0x10
2. Répertoire: « .. », au cluster 0
attribut = 0x10
3. Fichier au nom long
attribut = 0xF
4. Heure et date de création
5. Date dernier accès
6. Heure et de date de dernière modification
7. Numéro du premier cluster du fichier: 5.
8. Taille du fichier: 0x12c25 = 76837

```
01/01/2015 22:39 <REP> .
01/01/2015 22:39 <REP>
01/01/2015 21:51 76 837 2015-01-01 21_51_02-HxD - [CANON_DC <Q_>].png
1 fichier(s) 76 837 octets
2 Rép(s) 15 922 954 240 octets libres
```



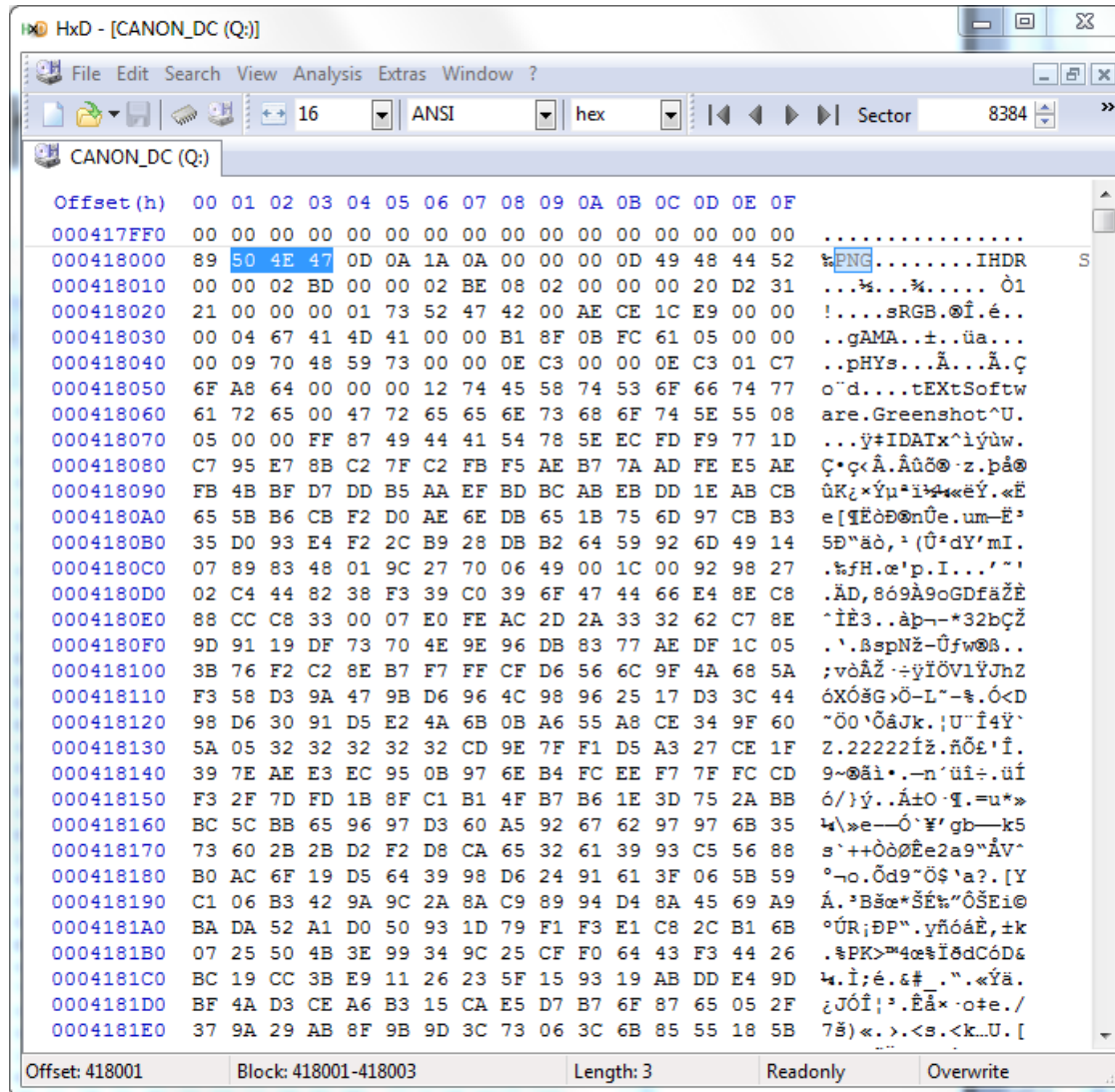
FAT32: fichier au cluster 4



The screenshot shows the HxD hex editor interface. The title bar reads 'HxD - [CANON_DC (Q:)]'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Analysis', 'Extras', and 'Window'. The toolbar shows various file operations and a status bar at the bottom indicates 'Offset: 4101E6', 'ReadOnly', and 'Overwrite'. The main display area shows a hex dump of a file named 'CANON_DC (Q:)'. The address range is from 00040FFF0 to 0004101E0. The hex data is displayed in columns, and the ASCII representation is shown on the right. The file appears to be a PNG image, as indicated by the 'PNG.....IHDR' signature at offset 000410000. The hex value 89 50 4E 47 is highlighted in orange at offset 000410000.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00040FFF0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000410000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
000410010	00	00	01	A7	00	00	01	84	08	02	00	00	00	1D	B5	78	...S.....ux
000410020	57	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00	W....sRGB.0i.e.
000410030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	..gAMA..t..ua...
000410040	00	09	70	48	59	73	00	00	0E	C3	00	00	0E	C3	01	C7	..pHYs...A...A.C
000410050	6F	A8	64	00	00	00	12	74	45	58	74	53	6F	66	74	77	o'd....tEXtSoftw
000410060	61	72	65	00	47	72	65	65	6E	73	68	6F	74	5E	55	08	are.Greenshot^U
000410070	05	00	00	71	1F	49	44	41	54	78	5E	ED	BD	69	78	14	...q.IDATx^ix.
000410080	57	9A	A0	EB	F9	3D	F3	4C	FF	99	E7	99	9A	1F	53	EA	W\$ eù=óLy™ç™š.Sê
000410090	99	A7	EF	F4	DC	B9	73	E7	76	77	F5	32	EA	EE	AA	EA	™\$iôÛ²sqvwô2êi²ê
0004100A0	69	BB	B1	6B	E9	EA	AE	EE	B2	8D	2D	5C	76	15	76	96	i»±kêê0i².-\v.v-
0004100B0	B1	8D	6B	35	C2	E0	1D	57	95	B1	C1	20	B0	CD	6E	B3	±.k5Âa.W.±Á °In²
0004100C0	AF	02	83	8D	01	B1	6F	02	21	F6	45	12	42	42	48	42	².f...to.¡0E.BBHB
0004100D0	A0	3D	33	95	7B	46	66	4A	F7	3B	4B	44	9E	3C	B1	64	=3•{FfJ+;KDž<±d
0004100E0	46	66	44	2A	A5	FC	BE	E7	35	8E	38	71	E2	7C	11	19	FfD*¥ü%ç5ž8qâ ..
0004100F0	27	5E	9D	C8	25	E2	81	C1	C1	C1	7B	34	2E	5E	BC	B8	¹.Êšâ.ÂÂÂ(4.~4.
000410100	7F	FF	7E	0F	06	06	86	10	CF	79	3C	CF	7A	3C	3F	F5	.ÿ~...t.ÿy<ÿz<?ô
000410110	78	9E	F1	78	7E	EC	F1	3C	E9	F1	4C	FD	A9	E7	B1	67	xžñx~iñ<éñLy0ç±g
000410120	D2	C0	2C	14	C2	22	A8	00	D5	A0	32	AC	E2	6A	B0	4D	ÒÀ,.Â"™.Ö 2-âj°M
000410130	FA	09	6C	CC	73	9E	7F	7D	CA	F3	FD	C7	9E	7E	E8	FB	ú.lisž.}ÊôýÇž~èû
000410140	8F	FE	F5	FF	F9	CE	9F	FC	E5	B7	FE	DB	FF	F3	67	FF	.pôÿùîÿuâ²pÿÿôÿÿ
000410150	F5	8F	FF	D7	1F	FD	8F	3F	FB	E3	FF	F5	97	FF	F7	9F	ô.ÿ×.ÿ.¿ûâÿô-ÿ-ÿ
000410160	FE	CD	FF	FC	8B	BF	FB	93	CA	7F	F8	F3	6F	7D	F7	7F	píÿüç¿û"Ê.ø00)÷.
000410170	FF	FD	3F	FF	F5	43	FF	FA	D7	FF	F0	AF	7F	33	E5	47	ÿÿ?ÿôCÿú×ÿô³.3âG
000410180	94	47	35	FE	16	78	F8	B1	6F	3E	F2	D8	B7	1E	79	EC	"G5p.xø±0>ò0°.ÿi
000410190	DB	DF	9D	FA	E0	0F	A6	4D	F9	E1	D3	DF	F9	D1	4F	FF	Ûš.úa.¡MúaôšûN0ÿ
0004101A0	71	AA	E7	87	D3	3C	8F	FE	C4	F3	C4	73	9E	69	1E	CF	q²ç±Ó<.pÂôÂšzi.ÿ
0004101B0	53	1E	CF	D3	74	1F	45	AE	04	76	1C	80	A5	22	50	1F	S.Í0t.E .v.€¥"P.
0004101C0	B6	6D	EA	74	CF	63	3F	A1	AF	D2	4F	3C	8F	FF	94	CC	¶mêtÿç?;¬00<.ÿ"i
0004101D0	3E	F9	2C	59	04	6B	C1	0B	35	BD	28	AF	15	8B	F5	EB	>ù,Y.kÂ.5%{².ôê
0004101E0	D7	5F	B8	70	E1	FC	F9	F3	D7	AE	5D	6B	6D	6D	7D	A0	×_.pâüü0×0]kmm}

FAT32: fichier au cluster 5



The screenshot shows the HxD hex editor interface. The title bar reads 'HxD - [CANON_DC (Q:)]'. The menu bar includes 'File', 'Edit', 'Search', 'View', 'Analysis', 'Extras', and 'Window'. The toolbar shows various file operations and a '16' byte selection. The status bar at the top indicates 'ANSI' encoding, 'hex' view, and 'Sector 8384'. The main window displays a hex dump of a file named 'CANON_DC (Q:)'. The first column shows 'Offset (h)' from 000417FF0 to 0004181E0. The second column shows the hex data. The third column shows the ASCII representation. The data starts with a PNG header: 'PNG.....IHDR'. The status bar at the bottom shows 'Offset: 418001', 'Block: 418001-418003', 'Length: 3', 'ReadOnly', and 'Overwrite'.

```
Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000417FF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
000418000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
000418010 00 00 02 BD 00 00 02 BE 08 02 00 00 00 20 D2 31 ...%...%.... Ô1
000418020 21 00 00 00 01 73 52 47 42 00 AE CE 1C E9 00 00 !....sRGB.©Î.é..
000418030 00 04 67 41 4D 41 00 00 B1 8F 0B FC 61 05 00 00 ..gAMA..±..üa...
000418040 00 09 70 48 59 73 00 00 0E C3 00 00 0E C3 01 C7 ..pHYs...Ä...Ä.Ç
000418050 6F A8 64 00 00 00 12 74 45 58 74 53 6F 66 74 77 o`d....tEXtSoftw
000418060 61 72 65 00 47 72 65 65 6E 73 68 6F 74 5E 55 08 are.Greenshot^U.
000418070 05 00 00 FF 87 49 44 41 54 78 5E EC FD F9 77 1D ...ÿ+IDATx^iÿûw.
000418080 C7 95 E7 8B C2 7F C2 FB F5 AE B7 7A AD FE E5 AE Ç•ç•Ä.Äûð•z.pää
000418090 FB 4B BF D7 DD B5 AA EF BD BC AB EB DD 1E AB CB ûKç×ÿµ*ÿ•«eÿ.«Ë
0004180A0 65 5B B6 CB F2 D0 AE 6E DB 65 1B 75 6D 97 CB B3 e[ÿËòðonÛe.um-Ë³
0004180B0 35 D0 93 E4 F2 2C B9 28 DB B2 64 59 92 6D 49 14 5B"àò,³(Û³dÿmI.
0004180C0 07 89 83 48 01 9C 27 70 06 49 00 1C 00 92 98 27 .%fH.œ'p.I...'~'
0004180D0 02 C4 44 82 38 F3 39 C0 39 6F 47 44 66 E4 8E C8 .ÄD,8ó9Ä9oGDfäZË
0004180E0 88 CC C8 33 00 07 E0 FE AC 2D 2A 33 32 62 C7 8E ^ËË3..äp-~*32bÇZ
0004180F0 9D 91 19 DF 73 70 4E 9E 96 DB 83 77 AE DF 1C 05 .\$.aspNž-Ûfw@B..
000418100 3B 76 F2 C2 8E B7 F7 FF CF D6 56 6C 9F 4A 68 5A ;vðÄZ-~ÿÏÖVlÿJhZ
000418110 F3 58 D3 9A 47 9B D6 96 4C 98 96 25 17 D3 3C 44 óXóšG>Ö-L"-$.ó<D
000418120 98 D6 30 91 D5 E2 4A 6B 0B A6 55 A8 CE 34 9F 60 ~Öö'ÖâJk.¡Û"í4ÿ'
000418130 5A 05 32 32 32 32 CD 9E 7F F1 D5 A3 27 CE 1F Z.22222îž.ñö£'î.
000418140 39 7E AE E3 EC 95 0B 97 6E B4 FC EE F7 7F FC CD 9~@äi•.-n'úi÷.uí
000418150 F3 2F 7D FD 1B 8F C1 B1 4F B7 B6 1E 3D 75 2A BB ó/}ý..Ä±Ö`ÿ.=u*»
000418160 BC 5C BB 65 96 97 D3 60 A5 92 67 62 97 97 6B 35 ¼\»e—ó`ÿ'gb—k5
000418170 73 60 2B 2B D2 F2 D8 CA 65 32 61 39 93 C5 56 88 s`++òððËe2a9"ÄV^
000418180 B0 AC 6F 19 D5 64 39 98 D6 24 91 61 3F 06 5B 59 °-o.Öd9"Ö$'a?.[Y
000418190 C1 06 B3 42 9A 9C 2A 8A C9 89 94 D4 8A 45 69 A9 Á.'Bšœ*ŠË%~"ÖŠEi@
0004181A0 BA DA 52 A1 D0 50 93 1D 79 F1 F3 E1 C8 2C B1 6B °ÛR;ðP".yñóáË,±k
0004181B0 07 25 50 4B 3E 99 34 9C 25 CF F0 64 43 F3 44 26 .%PK>»4œ$ËðdCód&
0004181C0 BC 19 CC 3B E9 11 26 23 5F 15 93 19 AB DD E4 9D ¼.î:é.£#_".«ÿä.
0004181D0 BF 4A D3 CE A6 B3 15 CA E5 D7 B7 6F 87 65 05 2F çJÓî!;³.ËÄ×·o+e./
0004181E0 37 9A 29 AB 8F 9B 9D 3C 73 06 3C 6B 85 55 18 5B 7š)«.>.<s.<k..U.[
```

Offset: 418001 Block: 418001-418003 Length: 3 ReadOnly Overwrite

FAT32: chaînage dans la FAT

Fichier au cluster 5: 76837 octets

Fichier au cluster 4: 29096 octets

Taille d'un cluster: 32k

Entrée #5 de la FAT:

6 = prochain cluster du fichier est le 6

Entrée #6 de la FAT:

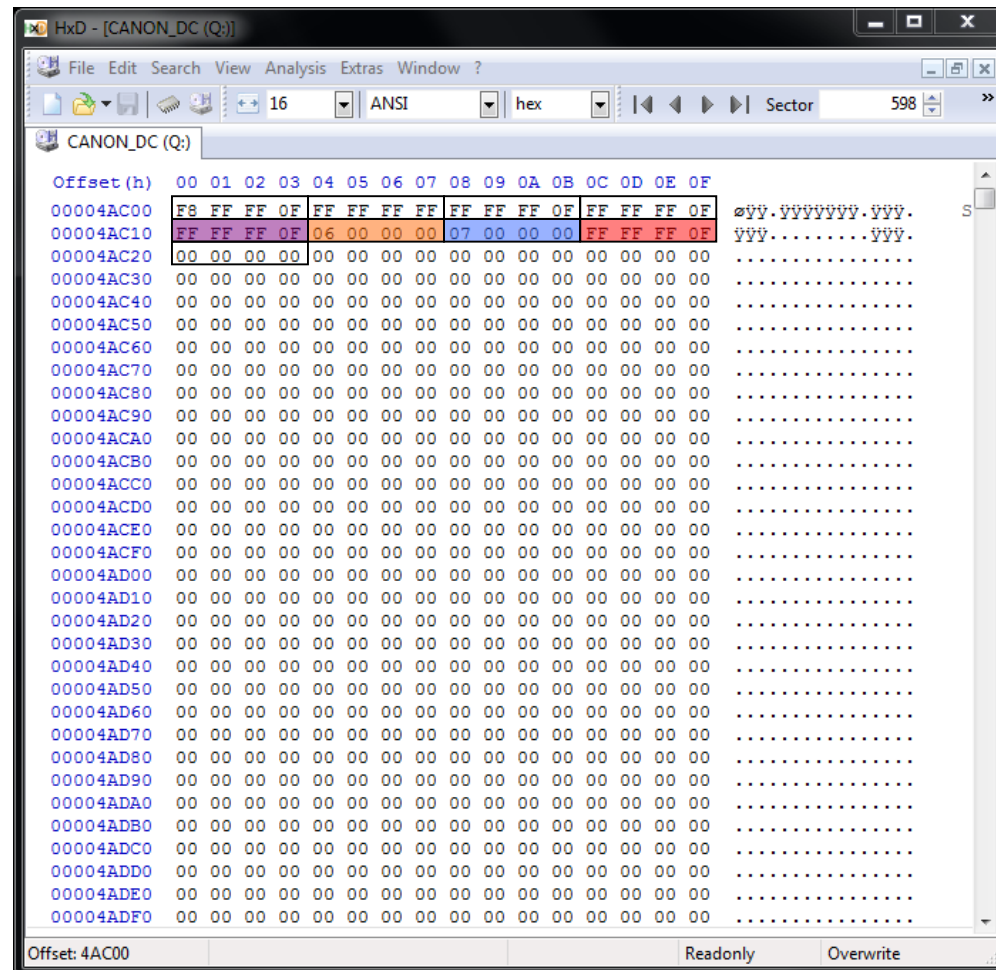
7 = prochain cluster du fichier est le 7

Entrée #7 de la FAT:

0x0fffffff = fin de la chaîne de 3 clusters

Expliquer la valeur pour le cluster 4

Et les 4 premiers clusters ?



FAT32: Résumé

	Offset (octets)	taille	secteur	cluster	Origine de l'info
MBR	0	512	0	N/A	Toujours secteur 0
Secteur de boot	$s \cdot 512$	512	s	N/A	MBR
FAT1	$0x256 \cdot 512 = 0x4ac00$	0xed5 secteurs	0x256	N/A	Secteur de boot
FAT2	$0x4ac00 + 0xed5 \cdot 512 = 0x225600$	0xed5 secteurs	$0x256 + 0xed5$	N/A	Secteur de boot
Zone des clusters	$0x225600 + 0xed5 \cdot 512 = 0x400000$		$0x256 + 0xed5 \cdot 2$	#2	Secteur de boot
Cluster#n (4)	$0x400000 + 512 \cdot 64 \cdot (4-2) = 0x410000$	64 secteurs	$0x256 + 0xed5 \cdot 2 + 64 \cdot (n-2)$	#4	Secteur de boot

Agenda

- Physique/logique/partition
- Partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit

ExFAT

Amélioration de FAT32,
avec quelques fonctions de NTFS:

- Secteur jusqu'à 4096 octets
- Cluster jusqu'à 32Mo
- Dates en UTC
- Noms de fichiers de 255 caractères max, en unicode
- VBR de 12 secteurs
- Taille maximale d'un fichier: $2^{64}-1$
- Pour les cartes mémoire SDXC (2009)
- Infos temporelles:
<http://www.ntfs.com/exfat-time-stamp.htm>

ExFAT: structure

MBR

...

VBR primaire de 12 secteurs

Backup VBR de 12 secteurs

...

FAT principale

Backup FAT

Clusters

exFAT: MBR

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	01
000001C0	14	00	07	03	60	DA	33	00	00	00	4D	ED	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55 AA

1. Bootable = 0x80 (oui)
2. Type de partition=7
3. Premier secteur = 0x33
4. Taille en secteur = 0xED4D
5. Synchro fin de secteur

Supporté par TSK depuis la 4.2.0

exFAT: VBR#1

Taille de la VBR=12 secteurs

1. **Jmp instruction**
2. **Signature: « EXFAT »**
3. **VBR1 address= sector number: 0x33
(début disque) = 0x33*512=0x6600**
4. **Total volume size in sectors:
0xED4D = 32mo**
5. **Address of FAT#1: secteur 0x80 (+VBR)**
6. **Size of FAT: 0x40 sectors**
7. **Data/cluster region address: 0x100 (+VBR)**
8. **Number of Cluster number in the Cluster heap**
9. **Cluster address of root dir: 5**
10. **Volume S/N**
11. **exFat version: 1.0**
12. **Flags**
13. **Bytes per sectors: 9 ($2^9 = 512$ bytes)**
14. **Sectors per clusters: 3 ($2^3 = 8$)**
15. **End of bootcode**
16. **Synchro fin de secteur**

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00006600	EB	76	90	45	58	46	41	54	20	20	20	00	00	00	00	00	èv.EXFAT
00006610	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006620	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006630	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006640	33	00	00	00	00	00	00	00	00	4D	ED	00	00	00	00	00	3.....Mi.....
00006650	80	00	00	00	40	00	00	00	00	00	01	00	00	89	1D	00	€...@.....%...
00006660	05	00	00	00	F0	08	37	F2	00	01	00	00	09	03	01	808.7ò.....€
00006670	00	00	00	00	00	00	00	00	00	33	C9	8E	D1	BC	F0	7B3ÉŽÑ-8{Ž
00006680	D9	A0	FB	7D	B4	7D	8B	F0	AC	98	40	74	0C	48	74	0E	Û û}'}<8-~@t.Ht.
00006690	B4	0E	BB	07	00	CD	10	EB	EF	A0	FD	7D	EB	E6	CD	16	'...».î.ëi ý)ëaî.
000066A0	CD	19	00	00	00	00	00	00	00	00	00	00	00	00	00	00	î.....
000066B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000066C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000066D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000066E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000066F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006700	0D	0A	52	65	74	69	72	65	7A	20	6C	65	20	64	69	73	..Retirez le dis
00006710	71	75	65	FF	0D	0A	45	72	72	2E	20	64	69	73	71	75	queÿ..Err. disqu
00006720	65	FF	0D	0A	50	72	65	73	73	65	7A	20	75	6E	65	20	eÿ..Pressez une
00006730	74	6F	75	63	68	65	20	70	6F	75	72	20	72	65	64	82	touche pour red,
00006740	6D	61	72	72	65	72	0D	0A	00	00	00	00	00	00	00	00	marrer.....
00006750	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006760	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006770	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006780	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00006790	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000067A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000067B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	FFÿÿ
000067C0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000067D0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000067E0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
000067F0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	00	14	22	55	ÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00006800	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

[/reverse-engineering-microsoft-exfat-file-system-33274](https://reverse-engineering-microsoft-exfat-file-system-33274)

exFAT: exercice

Utiliser disk1.001.txt

- Trouver l'offset de la VBR2
- Trouver l'offset de la FAT1
- Trouver l'offset du 1^{er} cluster

En bonus, vous pouvez aller voir l'outil suivant (non utile pour l'exercice): <https://github.com/lclevy/exfatDump>

exFAT: précision

Dans l'exercice, le répertoire racine est indiqué au cluster 5 dans la VBR,
mais situé à l'index 3 dans la table des clusters, **car dans ExFAT, il n'y a pas de cluster #0 et #1**

Agenda

- Physique/logique/partition
- Partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit

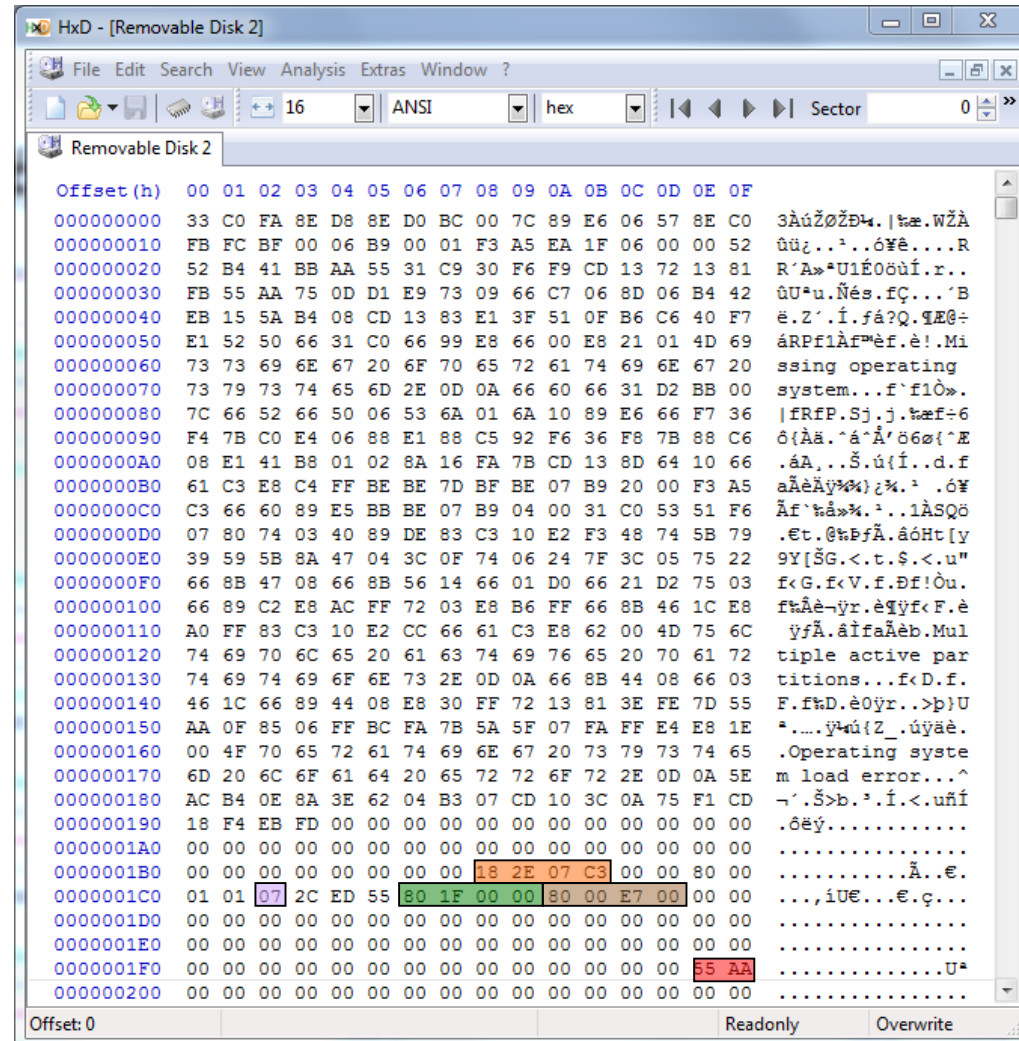
NTFS

- New Technology File System
- Depuis 1993 (Windows NT= v1.0; Win XP = v3.1)
- Support des fichiers longs (255 octets) et localisation
- Système de fichier fiable avec journalisation
- Supporte la compression, le chiffrement, le contrôle d'accès, les quotas
- Support des fichiers creux (sparse), les liens hard et soft
- Meilleure utilisation de la taille des clusters: moins de place non utilisée (« slack space »)
- Nouvelles limites
 - Fichiers: 16TB
 - Partition: 256TB (avec GPT)
 - Nombre de fichiers: $2^{32} = 4$ milliards
- Meilleure informations temporelles:
<https://articles.forensicfocus.com/2013/04/06/interpretation-of-ntfs-timestamps/>

NTFS: exemple de MBR

Clé USB de 8 Go

1. 32bits disk signature:
0x182e07c3
2. Type de partition: 7 (NTFS)
3. Début de la 1ère partition:
0x1f80 = secteur 8064
4. Taille de la partition:
0xe70080= 15138944 secteurs,
7 751 139 328 octets
5. Synchro de fin de secteur



NTFS: exemple de MBR

The Sleuth Kit (<http://www.sleuthkit.org/sleuthkit/>):

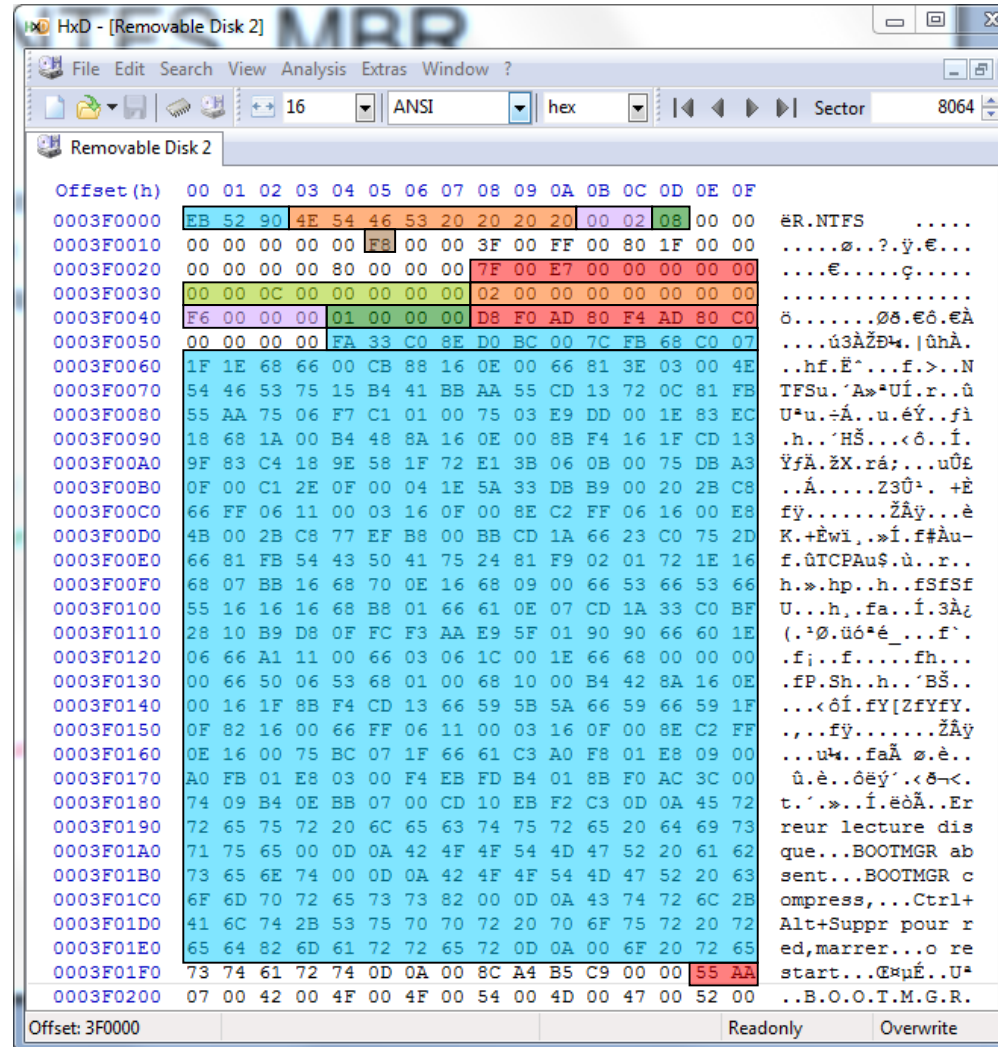
mmls liste les informations sur les partitions d'un disque
« mm » for media management

```
I:\>mmls kingston_usbkey_yellow.E01
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
00:	Meta	00000000000	00000000000	00000000001	Primary Table <#0>
01:	-----	00000000000	00000008063	00000008064	Unallocated
02:	00:00	00000008064	0015147007	0015138944	NTFS <0x07>

NTFS: Volume Boot Record

1. **Jmp instruction / code**
2. **Signature**
3. **Taille d'un secteur:**
0x200 = 512 octets
4. **Nombre de secteurs par cluster: 8**
Taille d'un cluster: 8*512 = 4k
5. **Media descriptor: 0xf8**
6. **Taille du volume en secteurs:**
0xe7007f
7. **Premier cluster de la \$MFT:**
0xc0000 = cluster 786432 (+VBR).
\$MFT au secteur 786432*8 (+8064)
= secteur 6299520
8. **Premier cluster de \$MFTMirr: 2**
9. **Taille des entrées \$FILE: 0xf6=-10**
interprétation: 2^(-1*-10)=1024 octets
10. **Taille du buffer \$INDX: 1 cluster**
11. **S/N du volume**



NTFS: exemple de MBR

Avec The Sleuth Kit:

Fsstat: liste les informations le système de fichiers

```
I:\>fsstat -o 8064 kingston_usbkey_yellow.E01
FILE SYSTEM INFORMATION
-----
File System Type: NTFS
Volume Serial Number: C080ADF480ADF0D8
OEM Name: NTFS
Volume Name: KINGSTON
Version: Windows XP

METADATA INFORMATION
-----
First Cluster of MFT: 786432
First Cluster of MFT Mirror: 2
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 256
Root Directory: 5

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 1892366
Total Sector Range: 0 - 15138942
```

Structure générale de la Master File Table

- Table
 - Entrée #0 (MFT)
 - Entête « FILE »
 - Attributs
 - Attributs
 - ...
 - Entrée #1
 - Entête « FILE »
 - Attributs
 - Attributs
 - ...
 - ...

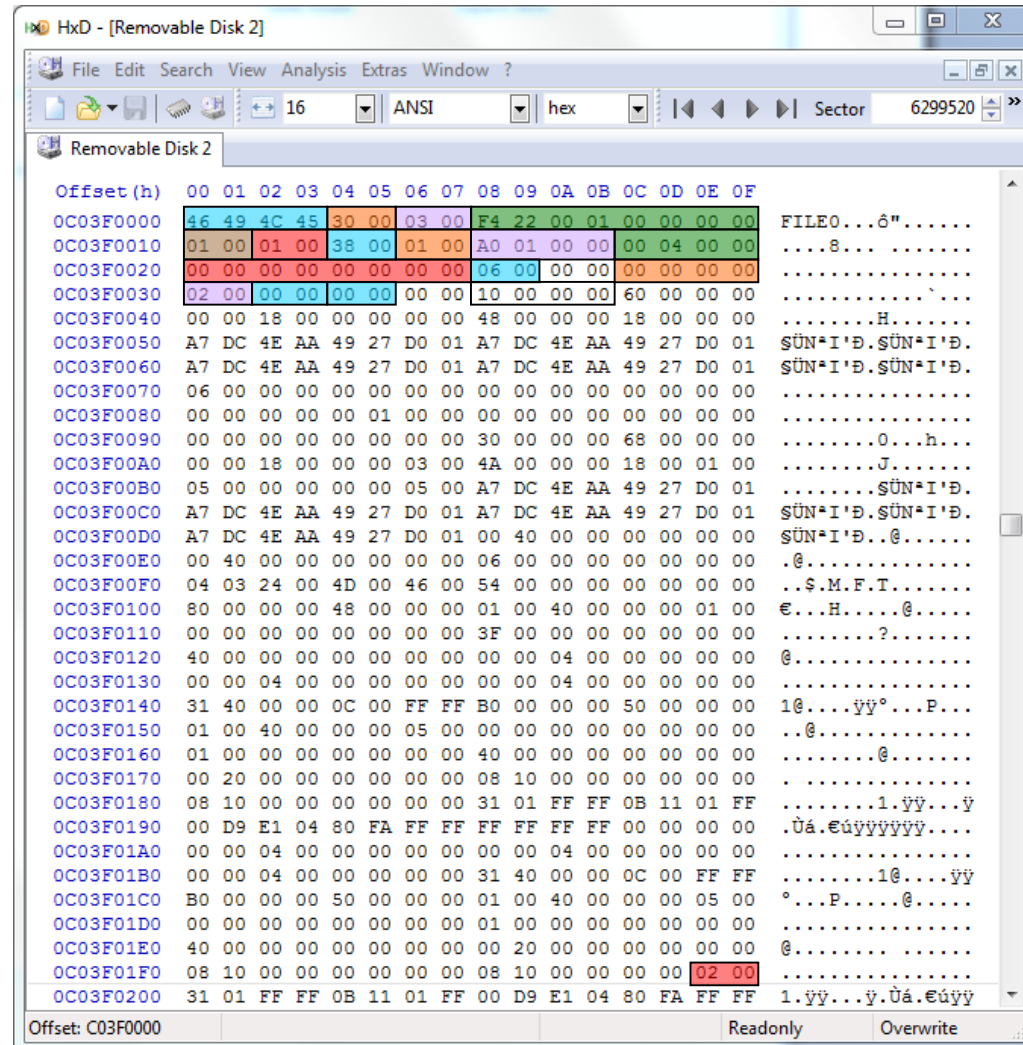
<https://www.ntfs.com/ntfs-mft.htm>

MFT: Entête FILE

La première entrée (#0) de la MFT est la MFT elle-même.

Format: entête (0x38 octets), puis les attributs

1. Signature
2. Offset vers fixup data: 0x30
3. Fixup data size: 3 valeurs de 2 octets.
Valeur USN=2, et 2 valeurs patchées aux offsets 0x1fe et 0x3fe
4. \$logfile sequence number
5. Sequence number: 1 (nombre de recyclage)
6. Hardlink (nombre d'attributs \$FILE): 1
7. Offset 1er attribut: 0x38
8. Flags: 0x0001 (file in use)
9. Taille réelle de l'entrée: 0x1a0
10. Taille allouée pour l'entrée: 0x400 (1024)
11. File reference to base record: 0
12. Nombre d'attributs: 6
13. Inode # of this record: 0
14. Fixup (USN): 0x0002
15. Valeurs originales écrasées par USN: 0 et 0
16. Fixup USN: 0x0002.
Valeur originale=0 (offset 0x32)



MFT: attribut \$Standard_Information

Attribut \$STANDARD_INFORMATION:

Entête générique d'un attribut:

1. Signature de l'attribut: 0x10
2. Taille de l'attribut: 0x60
3. Resident: 0 = oui
4. Name length: 0
5. Offset to name: 0x18
6. Flags: 0x0000
7. Attribute Id: 0000

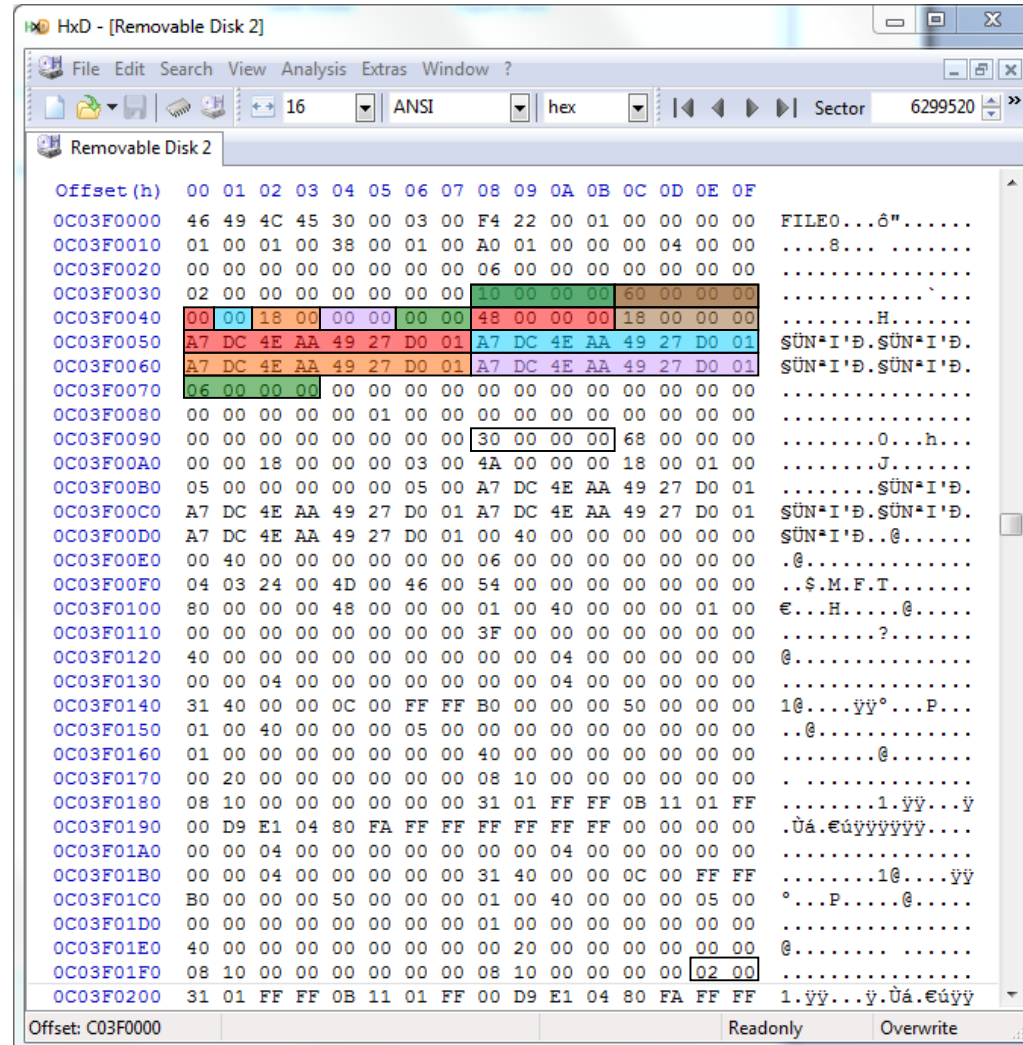
Pour un attribut resident:

8. Length of attribute: 0x48
9. Offset to attribute: 0x0018

Pour cet attribut:

10. Creation time (B, Birth)
11. Content modified time (M)
12. Metadata modified time (C)
13. Last accessed time (A)
14. Flags: 6 = 2 (hidden) | 4 (system)

Offsets: de 0x38 à 0x98 (0x38+0x60)



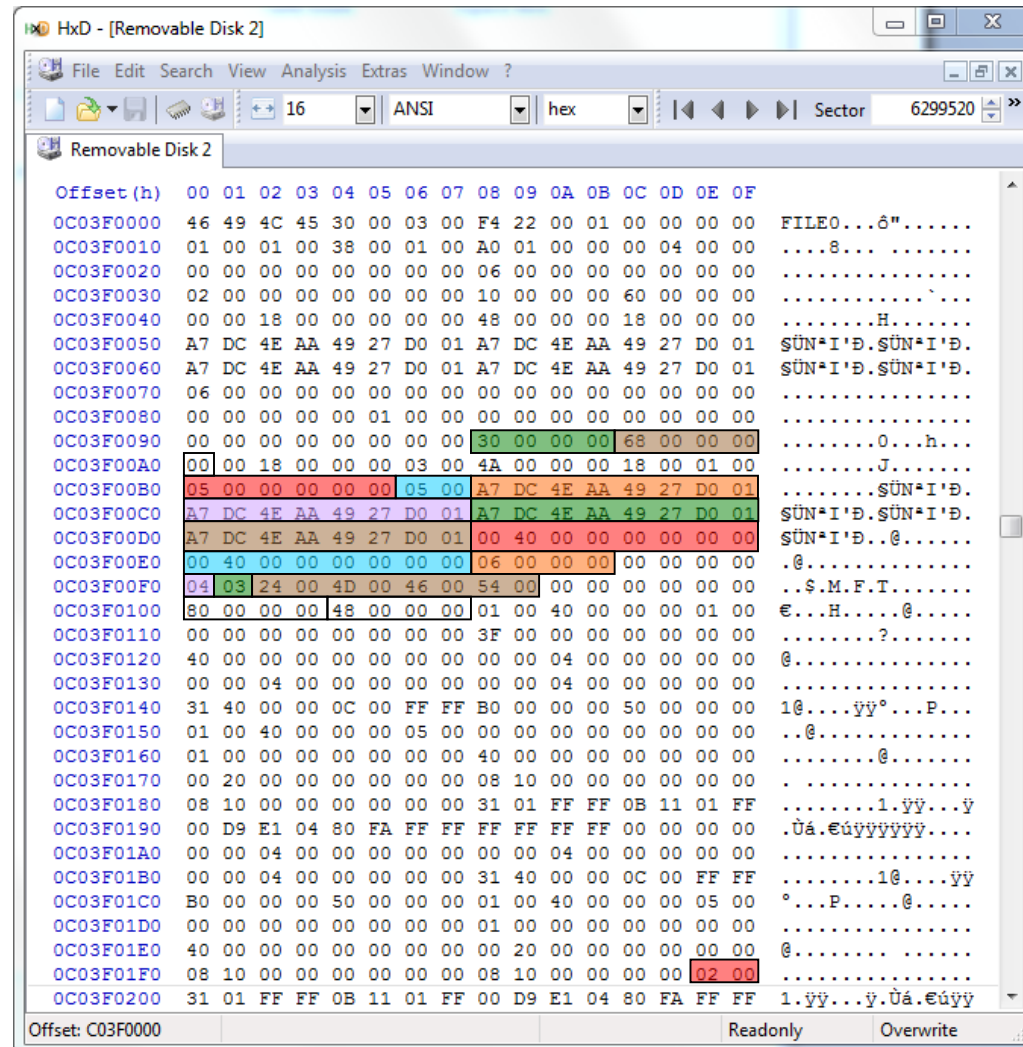
NTFS: attributs

Type	Nom	Description
0x10	\$Standard_information	Horodatage, flags
0x20	\$Attribute_List	Lorsqu'il y a trop d'attributs pour une seule entrée (1k) de la MFT
0x30	\$File_Name	Répertoire parent, horodatage, taille, flags et nom
0x40	\$Object_Id	Nom du volume, version de NTFS, dirty flag
0x50	\$Security_Descriptor	Info de sécurité et ACLs
0x60	\$Volume_Name	
0x70	\$Volume_Information	Version NTFS et drapeau
0x80	\$Data	Contenu du fichiers
0x90	\$Index_Root	Entête de l'index
0xa0	\$Index_Allocation	Contenu de l'index
0xb0	\$Bitmap	Allocation de l'index
0xc0	\$Reparse_Point	Extensions NTFS. Utilisé pour les soft et hard links, les points de montages.
0x100	\$Logged_Util_Stream	Contenu pour le journal ou les clés de chiffrement

MFT: attributs \$File_Name

Attribut \$File_Name:

1. Signature de l'attribut: 0x30
2. Taille de l'attribut: 0x68
3. Resident: 0=oui
4. Numéro de l'entrée de la MFT pour le répertoire parent (=répertoire racine): 5
5. Numéro de séquence pour cette entrée de la MFT
6. Timestamps: B, M, C, A
7. Physical size: 0x4000
8. Logical size: 0x4000
9. Flags: 6
10. Name length: 4
11. Namespace type: 3 (Win32/DOS)
12. Nom: « \$MFT »



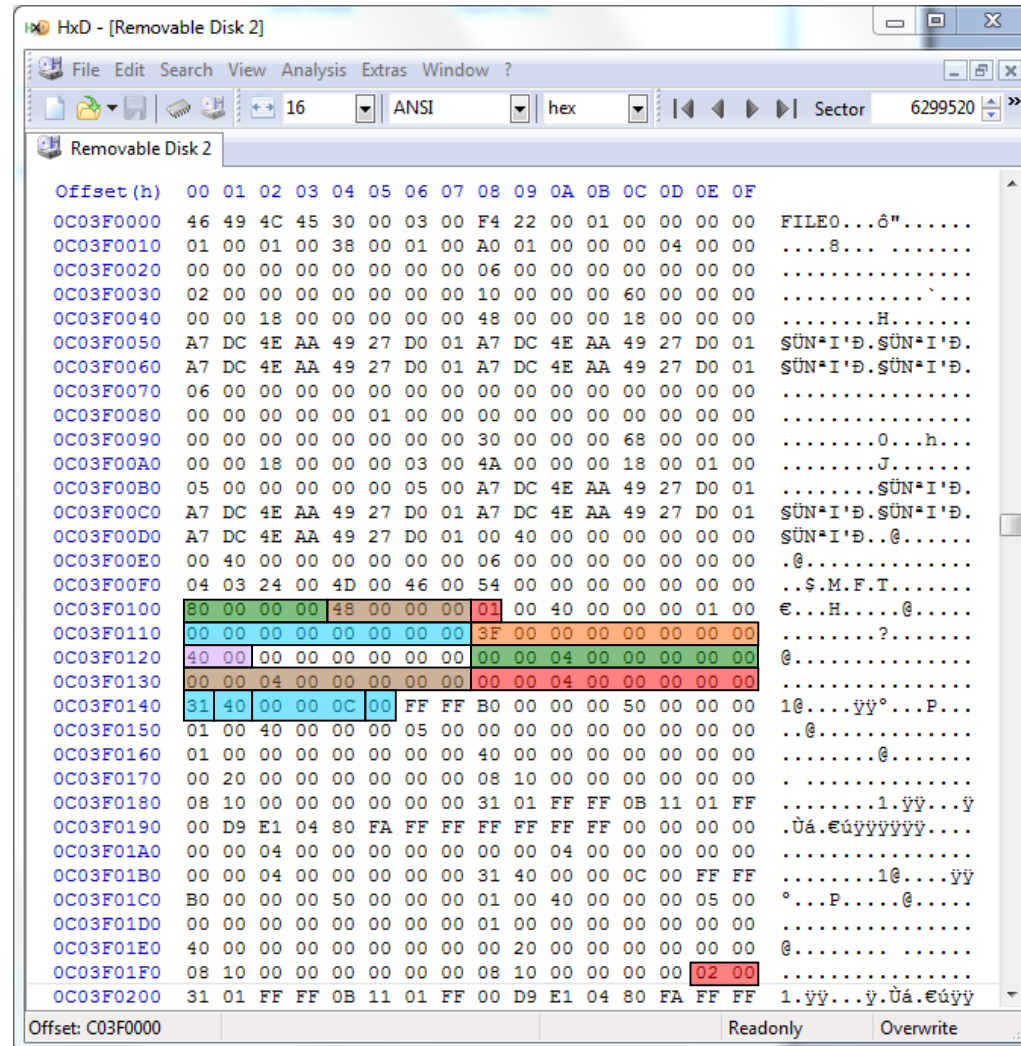
MFT: \$Data

Attribut \$Data (offset 0x100):

1. Signature de l'attribut: 0x80
2. Taille de l'attribut: 0x48
3. Resident: 1=non. 0 pour oui: dans le cas de petits fichiers (600 octets environ), le contenu est stocké dans l'attribut \$data directement.
4. Starting Virtual Cluster Number: 0
5. Ending Virtual Cluster Number: 0x3f
6. Offset to Data Run (chaînage des cluster de données): 0x40. $0x100 + 0x40 = 0x140$
7. Allocated size
8. True size
9. Initialized size
True < Initialized < Allocated
10. Data runs
0x31: read 1 byte for the length (0x40), then read 3 bytes for first cluster number

0x40: length = 0x40 clusters

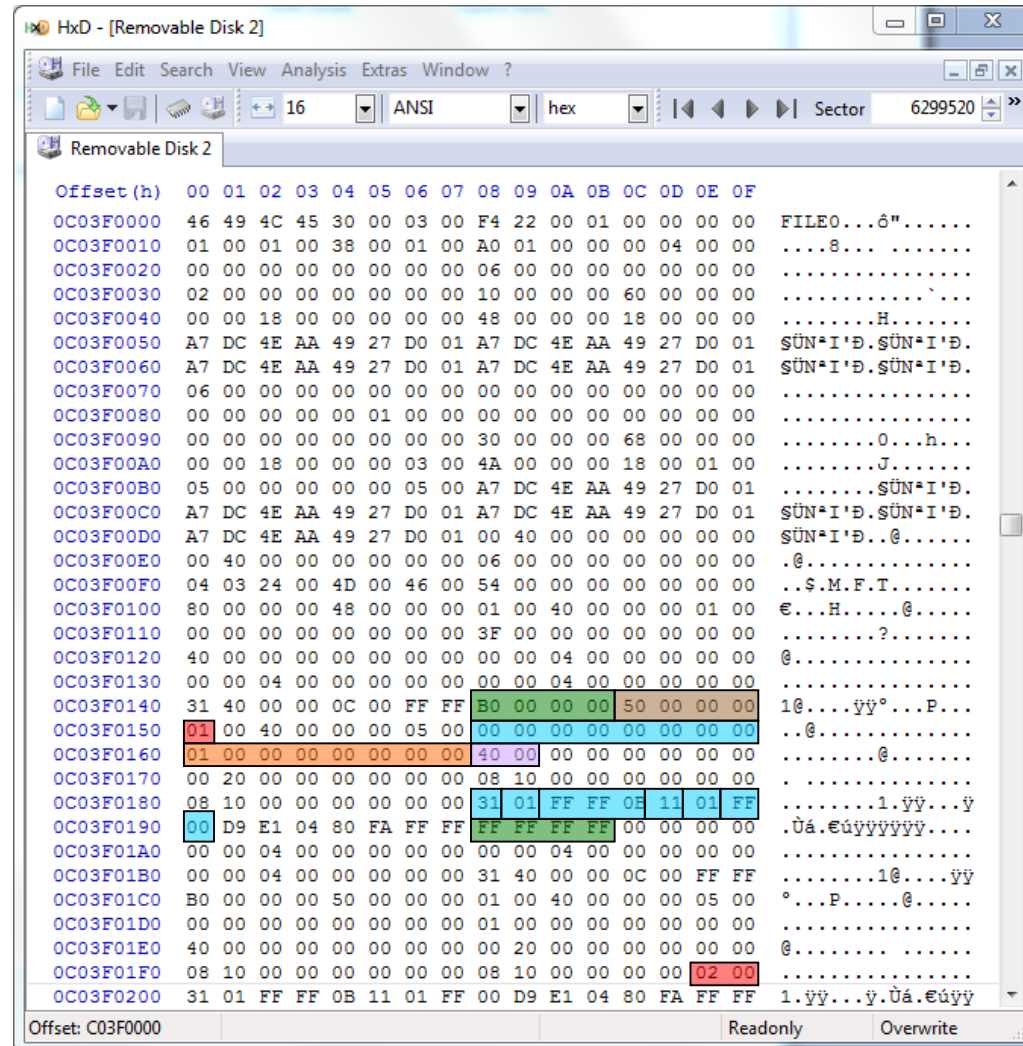
0x0c0000: first data cluster of \$MFT
0x00: end of data runs



MFT: autres attributs

Attribut \$Bitmap (offset 0x148):

- Signature de l'attribut: 0xb0
- Taille de l'attribut: 0x50
- Resident: 1=non.
- Starting Virtual Cluster Number: 0
- Ending Virtual Cluster Number: 0
- Offset to Data Run (chaînage des cluster de données): 0x40. $0x148 + 0x40 = 0x188$
- Signature de l'attribut: 0xffffffff = fin des attributs de cette entrée de la MFT



NTFS: attributs de \$MFT

TSK: *istat* pour afficher les metadata sur l'entrée MFT #0

```
I:\>istat -o 8064 kingston_usbkey_yellow.E01 0
MFT Entry Header Values:
Entry: 0          Sequence: 1
$LogFile Sequence Number: 16786164
Allocated File
Links: 1

$STANDARD_INFORMATION Attribute Values:
Flags: Hidden, System
Owner ID: 0
Security ID: 256 (< >)
Created:          2015-01-03 12:37:33 <Paris, Madrid>
File Modified:    2015-01-03 12:37:33 <Paris, Madrid>
MFT Modified:     2015-01-03 12:37:33 <Paris, Madrid>
Accessed:         2015-01-03 12:37:33 <Paris, Madrid>

$FILE_NAME Attribute Values:
Flags: Hidden, System
Name: $MFT
Parent MFT Entry: 5          Sequence: 5
Allocated Size: 16384        Actual Size: 16384
Created:          2015-01-03 12:37:33 <Paris, Madrid>
File Modified:    2015-01-03 12:37:33 <Paris, Madrid>
MFT Modified:     2015-01-03 12:37:33 <Paris, Madrid>
Accessed:         2015-01-03 12:37:33 <Paris, Madrid>

Attributes:
Type: $STANDARD_INFORMATION <16-0> Name: N/A Resident size: 72
Type: $FILE_NAME <48-3> Name: N/A Resident size: 74
Type: $DATA <128-1> Name: N/A Non-Resident size: 262144 init_size: 262144
786432 786433 786434 786435 786436 786437 786438 786439
786440 786441 786442 786443 786444 786445 786446 786447
786448 786449 786450 786451 786452 786453 786454 786455
786456 786457 786458 786459 786460 786461 786462 786463
786464 786465 786466 786467 786468 786469 786470 786471
786472 786473 786474 786475 786476 786477 786478 786479
786480 786481 786482 786483 786484 786485 786486 786487
786488 786489 786490 786491 786492 786493 786494 786495
Type: $BITMAP <176-5> Name: N/A Non-Resident size: 4104 init_size: 4104
786431 786430
```

NTFS: entrées de la MFT

TSK: *fls* pour lister les fichiers

```
I:\>fls -o 8064 kingston_usbkey_yellow.E01
r/r 4-128-4:      $AttrDef
r/r 8-128-2:      $BadClus
r/r 8-128-1:      $BadClus:$Bad
r/r 6-128-4:      $Bitmap
r/r 7-128-1:      $Boot
d/d 11-144-4:     $Extend
r/r 2-128-1:      $LogFile
r/r 0-128-1:      $MFT
r/r 1-128-1:      $MFTMirr
r/r 9-128-8:      $Secure:$SDS
r/r 9-144-11:     $Secure:$SDH
r/r 9-144-5:      $Secure:$SII
r/r 10-128-1:     $UpCase
r/r 3-128-3:      $Volume
r/r 35-128-1:     2015-01-02 22_04_00-HxD - [Hard Disk 1].png
d/d 36-144-1:     test3
-r/r * 38-128-4:  ReadyBoostPerfTest.tmp
d/d 256:          $OrphanFiles
```

r/r 0-128-1: \$MFT

0=entrée 0

NTFS: entrées des fichiers systèmes

#entrée	Filename	Description
0	\$MFT	Master File Table
1	\$MFTMirr	Copie des 4 premières entrée de la MFT
2	\$LOGFILE	Journal transactionnel des métadonnées
3	\$VOLUME	Nom du volume, version de NTFS, dirty flag
4	\$ATTRDEF	NTFS Attribute definitions
5	.	Répertoire racine
6	\$BITMAP	Etat d'allocation des clusters
7	\$BOOT	
8	\$BADCLUSTER	Clusters défectueux
9	\$SECURE	Propriétaire des fichiers et répertoires, notamment
10	\$UPCASE	Table des caractères Unicodes pour les tris
11	\$EXTEND	Un répertoire contenant les extension: \$ObjId, \$Quota, \$Reparse, \$UsnJrnl (journal des fichiers)

NTFS: exercice

- Utiliser l'archive ntfs.zip
- Etudier le fichier 0.txt et trouver l'offset de la VBR
- L'offset est le mot de passe (de la forme 0xn timers) de 1.zip
- Etudier le contenu de 1.txt et trouver l'offset de la MFT
- Déchiffrer 2.zip avec cet offset comme mot de passe (forme 0xn timers)
- Quelle est l'offset de l'entrée de la MFT du répertoire racine ? La réponse est le mdp pour accéder au dump et vérifier la réponse avec TSK

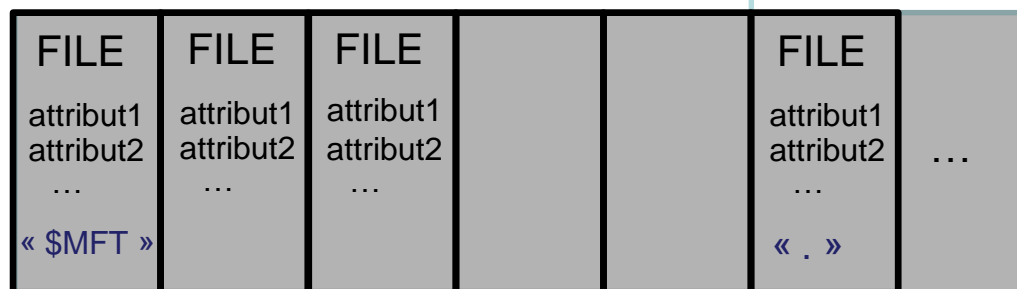
Exercice NTFS: un dessin ?

0 512



0 1

0



0 1 2 3 4 5 ...

Offset (octets)

Adresse en secteurs

Adresse en clusters

Adresse dans la MFT

Exercice NTFS: TSK mmls

```
>mmls disk2.001
```

```
DOS Partition Table
```

```
Offset Sector: 0
```

```
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
000:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
001:	-----	0000000000	0000000038	0000000039	Unallocated
002:	000:000	0000000039	0000121855	0000121817	NTFS / exFAT (0x07)

Exercice NTFS: TSK fsstat

```
>fsstat -o 39 disk2.001
```

```
FILE SYSTEM INFORMATION
```

```
-----  
File System Type: NTFS
```

```
Volume Serial Number: 7820B17620B13BC6
```

```
OEM Name: NTFS
```

```
Volume Name: EOS_DIGITAL
```

```
Version: Windows XP
```

```
METADATA INFORMATION
```

```
-----  
First Cluster of MFT: 5075
```

```
First Cluster of MFT Mirror: 2
```

```
Size of MFT Entries: 1024 bytes
```

```
Size of Index Records: 4096 bytes
```

```
Range: 0 - 256
```

```
Root Directory: 5
```

```
CONTENT INFORMATION
```

```
-----  
Sector Size: 512
```

```
Cluster Size: 4096
```

```
Total Cluster Range: 0 - 15226
```

```
Total Sector Range: 0 - 121815
```

```
...
```

Exercice NTFS: TSK fsstat

...

\$AttrDef Attribute Values:

\$STANDARD_INFORMATION (16) Size: 48-72 Flags: Resident
\$ATTRIBUTE_LIST (32) Size: No Limit Flags: Non-resident
\$FILE_NAME (48) Size: 68-578 Flags: Resident, Index
\$OBJECT_ID (64) Size: 0-256 Flags: Resident
\$SECURITY_DESCRIPTOR (80) Size: No Limit Flags: Non-resident
\$VOLUME_NAME (96) Size: 2-256 Flags: Resident
\$VOLUME_INFORMATION (112) Size: 12-12 Flags: Resident
\$DATA (128) Size: No Limit Flags:
\$INDEX_ROOT (144) Size: No Limit Flags: Resident
\$INDEX_ALLOCATION (160) Size: No Limit Flags: Non-resident
\$BITMAP (176) Size: No Limit Flags: Non-resident
\$REPARSE_POINT (192) Size: 0-16384 Flags: Non-resident
\$EA_INFORMATION (208) Size: 8-8 Flags: Resident
\$EA (224) Size: 0-65536 Flags:
\$LOGGED_UTILITY_STREAM (256) Size: 0-65536 Flags: Non-resident

Exercice NTFS: TSK istat

```
>istat -o 39 disk2.001 0
```

MFT Entry Header Values:

Entry: 0 Sequence: 1

\$LogFile Sequence Number: 1057524

Allocated File

Links: 1

\$STANDARD_INFORMATION Attribute Values:

Flags: Hidden, System

Owner ID: 0

Security ID: 256 ()

Created: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d (*birth*)

File Modified: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d

MFT Modified: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d (*change*)

Accessed: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d

\$FILE_NAME Attribute Values:

Flags: Hidden, System

Name: \$MFT

Parent MFT Entry: 5 Sequence: 5

Allocated Size: 16384 Actual Size: 16384

Created: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure dFile (*birth*)

Modified: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d

MFT Modified: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d (*change*)

Accessed: 2015-09-14 20:03:37.232212000 (Paris, Madrid (heure d

Exercice NTFS: TSK istat

...

Attributes:

Type: \$STANDARD_INFORMATION (16-0)	Name: N/A	Resident	size: 72
Type: \$FILE_NAME (48-3)	Name: N/A	Resident	size: 74
Type: \$DATA (128-1)	Name: N/A	Non-Resident	size: 262144 init_size: 262144
5075 5076 5077 5078 5079 5080 5081 5082			
5083 5084 5085 5086 5087 5088 5089 5090			
5091 5092 5093 5094 5095 5096 5097 5098			
5099 5100 5101 5102 5103 5104 5105 5106			
5107 5108 5109 5110 5111 5112 5113 5114			
5115 5116 5117 5118 5119 5120 5121 5122			
5123 5124 5125 5126 5127 5128 5129 5130			
5131 5132 5133 5134 5135 5136 5137 5138			
Type: \$BITMAP (176-5)	Name: N/A	Non-Resident	size: 4104 init_size: 4104
5074 4560			

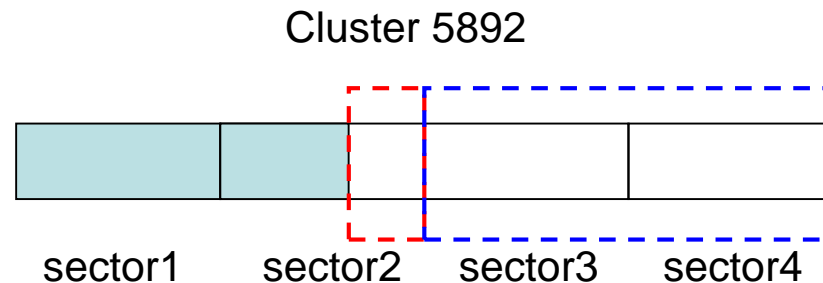
Vocabulaire: Slack space

Slack space:

espace non utilisé entre la fin du fichier et la fin du dernier cluster

Plus exactement, il se décompose en:

1. **RAM slack space:** espace entre fin de fichier et fin de secteur
2. **Drive slack space:** espace entre le début du prochain secteur et fin de cluster



Sous Windows, normalement le RAM slack space est rempli de 0. l'inverse est suspect!

NTFS: modification des timestamp selon les actions sur les fichiers

Windows® Time Rules									
\$ STANDARD_INFORMATION									
	File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
M	Modified – Time of File Creation	Modified – No Change	Modified – Time of Data Modification	Modified – No Change	Modified – Inherited from Original	Modified – No Change	Modified – Inherited from Original	Modified – Inherited from Original	Modified – No Change
A	Access – Time of File Creation	Access – Time of Access (no change only on NTFS with+)	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of File Move via CLI	Access – Time of Cut/Paste	Access – No Change
C	Metadata – Time of File Creation	Metadata – No Change	Metadata – Time of Data Modification	Metadata – Time of File Rename	Metadata – Time of File Copy	Metadata – Time of Local File Move	Metadata – Inherited from Original	Metadata – Inherited from Original	Metadata – No Change
B	Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of File Move via CLI	Creation – Inherited from Original	Creation – No Change
\$ FILENAME									
	File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
M	Modified – Time of File Creation	Modified – No Change	Modified – No Change	Modified – No Change	Modified – Time of File Copy	Modified – No Change	Modified – Time of Move via CLI	Modified – Time of Cut/Paste	Modified – No Change
A	Access – Time of File Creation	Access – No Change	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of Move via CLI	Access – Time of Cut/Paste	Access – No Change
C	Metadata – Time of File Creation	Metadata – No Change	Metadata – No Change	Metadata – No Change	Metadata – Time of File Copy	Metadata – No Change	Metadata – Time of Move via CLI	Metadata – Time of Cut/Paste	Metadata – No Change
B	Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of Move via CLI	Creation – Time of Cut/Paste	Creation – No Change

Autres sources d'information (1)

\$LogFile (journal transactionnel, buffer cyclique)

<https://github.com/jschicht/LogFileParser>

\$i30 (INDX)

Index des répertoires : autres données MACB

Permet parfois de montrer l'existence antérieure d'une entrée dans l'index, aujourd'hui effacée

Voir <https://github.com/harelsegev/INDXRipper> (for carving)

Voir <https://github.com/williballenthin/INDXParse> (targeted)

Voir <https://www.youtube.com/watch?v=x-M-wyq3BXA> (13Cubed)

Analyse de la \$MFT

<https://github.com/EricZimmerman/MFTECmd>

Sortie en csv ou json

Analyse aussi \$J, \$Logfile, \$Boot, \$SDS

Et ... Timeline Explorer est meilleur d'Excel ou
Notepad++ pour lire les CSV

<https://www.youtube.com/watch?v=Hy8Zlc86tCo>

Autres sources d'information (2)

\$UsnJrnl (journal des modifications)

<http://journeyintoir.blogspot.com/2013/01/re-introducing-usnjrnl.html>

jp (journal parser): https://tzworks.net/prototype_page.php?proto_id=5

Information pour chaque entrée:

- heure/date du change
- raison du modification (file_deleted, data_append, ...)
- nom du répertoire/fichier
- attributs du répertoire/fichier
- numéro d'entrée dans la MFT du répertoire/fichier
- “record number” du repertoire parent
- Security ID
- Update Sequence Number
- source de la modification

Forensic d'une partition NTFS

Extraire les données de la MFT pour analyse

Extraire les données de \$UsnJrnl (journal des changements)

Détecter les incohérences d'horodatage entre \$Standard_info et \$File_Name. Temps \$SI < \$FN!!

Les valeurs des 100^e de nanosecondes sont à 0: bizarre

Sur une partition peu remplie, le numéro de cluster s'incrémente avec le temps. Pourquoi le cluster# est très différent de ceux des autres fichiers créés dans la même minute ?

Comparer la date de compilation avec la date de création NTFS

Recherche des fichiers cachés (rootkit) ou effacés

Pour modifier le date d'un fichier: timestomp

Anomalies de dates

Comparer la date de compilation avec les dates NTFS

```
ExifTool Version Number      : 9.03
File Name                    : PeStudio.exe
Directory                   : ../PeStudio845
File Size                   : 551 KB
File Modification Date/Time  : 2014:12:10 17:05:54+01:00
File Access Date/Time       : 2015:01:05 22:55:49+01:00
File Permissions            : rw-rw-rw-
File Type                   : Win32 EXE
MIME Type                   : application/octet-stream
Machine Type                : Intel 386 or later, and compatibles
Time Stamp                  : 2014:12:10 16:45:11+01:00
PE Type                     : PE32
Linker Version              : 9.0
Code Size                   : 409600
Initialized Data Size       : 153600
Uninitialized Data Size     : 0
Entry Point                 : 0x4c9fd
OS Version                  : 5.0
Image Version               : 0.0
Subsystem Version           : 5.0
Subsystem                   : Windows GUI
File Version Number         : 8.45.0.0
Product Version Number      : 8.45.0.0
File Flags Mask             : 0x003f
File Flags                  : Private build, Special build
File OS                     : Win32
Object File Type            : Executable application
File Subtype                : 0
Language Code               : English (U.S.)
Character Set               : Unicode
Comments                    : Malware Early Triage
Company Name                : www.winator.com
File Description             : Malware Early Triage - www.winator.com
File Version                : 8.45.0.0
Internal Name               : pestudio.exe
Legal Copyright              : Copyright (C) 2009-2014 Marc Ochseneier
Legal Trademarks             : www.winator.com
Original Filename           : pestudio.exe
Product Name                : PESTudio
Product Version             : 8.45.0.0
```

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - **EXT4**
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
++++ PhysicalDrive0: UBOX HARDDISK ++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rounix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

EXT4

Principes

- Unité d'allocation: *Block* pour les données, *Inode* pour les métadonnées
- *Block*: groupe de secteur (1k à 64k), 4k en pratique comme les pages mémoires
- *Inode*: 256 octets
- L'allocation des *Blocks* sont gérés en « *Block group* », afin de minimiser la fragmentation (l'allocation est physiquement co-localisée).
Pour chaque *Bloc group*, un « *Group descriptor* » décrit:
 - *Super Block*
 - *Block* bitmap (si un bloc de données libre ou utilisé)
 - *Inode* bitmap (si un bloc de métadonnées libre ou utilisé)
 - Table des *inodes*: stocke les métadonnées et pointe vers les blocks de données
 - Data blocks

Fiabilité:

- Système de journal (fiabilité): jbd2. Depuis ext3
- Ext4: ajout de checksum pour toutes les structures importantes ext4/jbd2

EXT4

Autres particularités:

Optimisation de l'espace:

- Ext3: Tables d'inode, puis tables de tables, puis tables de tables de tables. Ext4: sous forme d'arbre (« extent tree »): beaucoup moins de métadonnées
- Fonction « Bigalloc » pour utiliser une unité d'allocation >4k (block), on utilise le cluster à la place du block

Répertoires stockés sous forme d'arbre binaire « haché ». Optimise la recherche, l'ajout et le retrait des entrées. Depuis ext3

Inodes de 0 à 11 réservés:

- Répertoire racine en inode#2, boot loader en inode#5, journal en inode#8,
...

EXT4: MBR

2 partitions:

Partition#0:

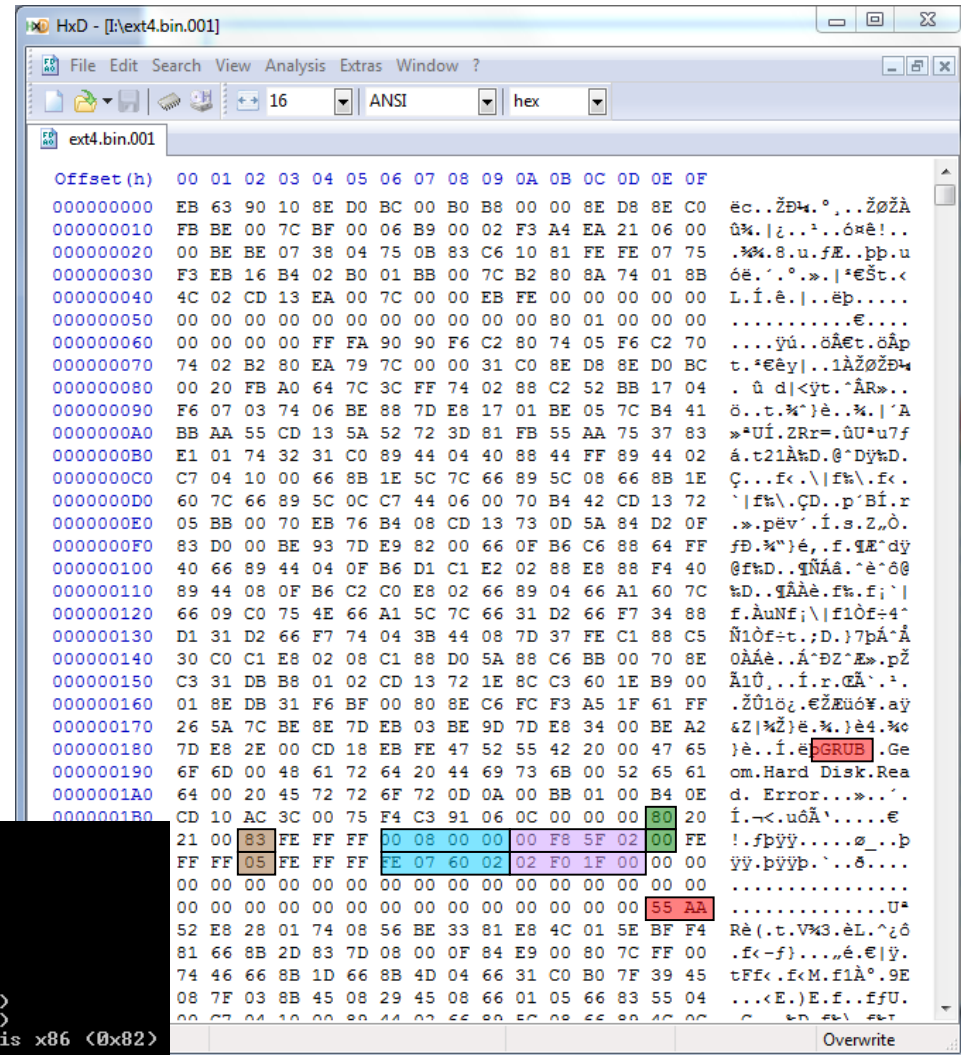
1. 0x80: Bootable
2. Partition type: 0x83
3. Début de la partition: 0x800 (2048)
4. Taille de la partition: 0x25ff800 (39843840)

Partition#1:

1. 0x00: non bootable
2. Partition type: 0x05 (extended)
3. Début de la partition: 0x26007fe (39847934)
4. Taille de la partition: 0x1ff002 (2093058)
5. Synchro fin de secteur

DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

	Slot	Start	End	Length	Description
00:	Meta	0000000000	0000000000	0000000001	Primary Table (#0)
01:	-----	0000000000	0000002047	0000002048	Unallocated
02:	00:00	0000002048	0039845887	0039843840	Linux (0x83)
03:	-----	0039845888	0039847935	0000002048	Unallocated
04:	Meta	0039847934	0041940991	0002093058	DOS Extended (0x05)
05:	Meta	0039847934	0039847934	0000000001	Extended Table (#1)
06:	01:00	0039847936	0041940991	0002093056	Linux Swap / Solaris x86 (0x82)
07:	-----	0041940992	0041943039	0000002048	Unallocated

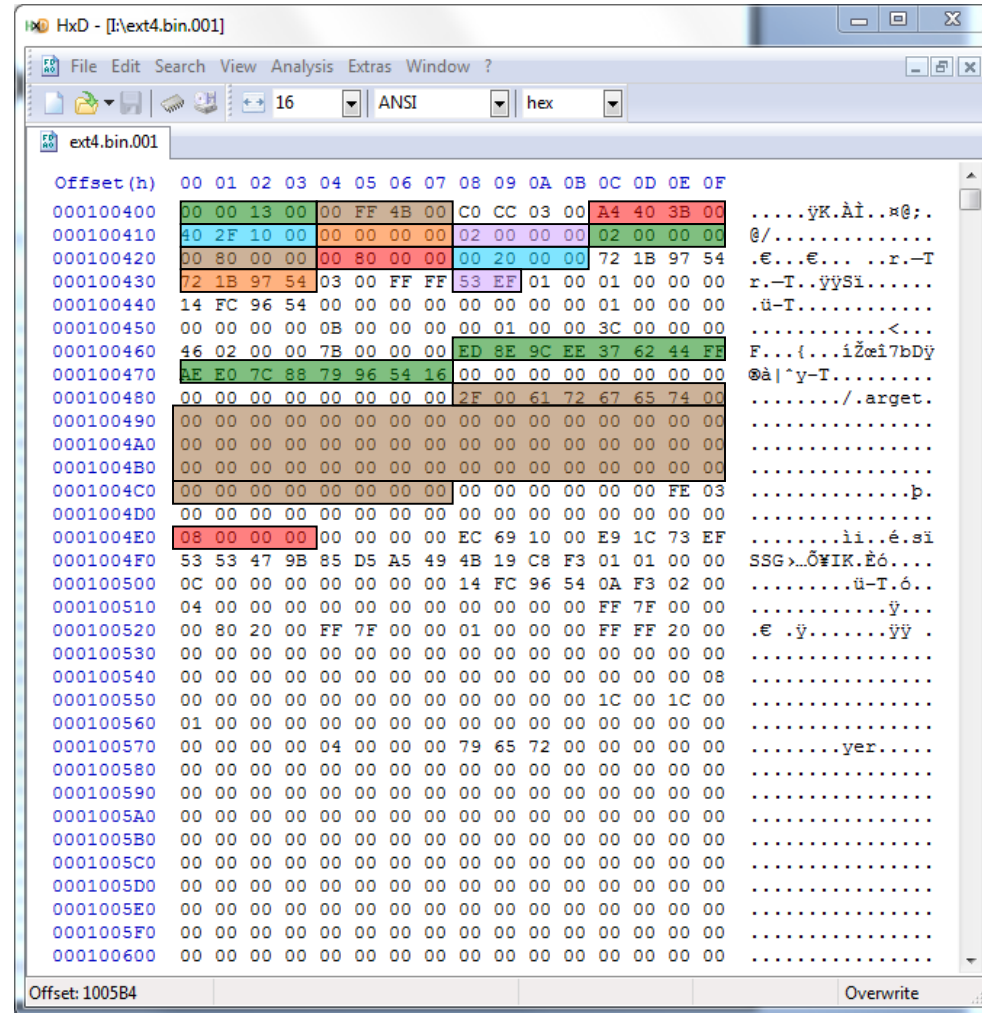


EXT4: the Super Block (1)

Super block: 1024 octets

Quelques champs du Super Block:

1. Total inode count: 0x130000 (1245184)
2. Total block count: 0x4bff00 (4980480)
3. Free block count: 0x3b40a4 (3883172)
4. Free inode count: 0x102f40 (1060672)
5. First data block: 0
6. Block size: 2. means $2^{(10+2)} = 4096$ bytes
- 7.
8. Block per groups: 0x8000
- 9.
10. Inodes per group: 0x2000
11. Write time sinch epoch
12. Signature: 0xef53
13. 128-bit UUID for volume
14. Last Mount point
15. Inode number for journal



https://ext4.wiki.kernel.org/index.php/Ext4_Disk_Layout

(c) 2023 Laurent Clévy

EXT4: the Super Block (2)

```
FILE SYSTEM INFORMATION
-----
File System Type: Ext4
Volume Name:
Volume ID: 16549679887ce0aeff446237ee9c8eed

Last Written at: 2014-12-21 20:11:46 (Paris, Madrid)
Last Checked at: 2014-12-21 17:57:56 (Paris, Madrid)

Last Mounted at: 2014-12-21 20:11:46 (Paris, Madrid)
Unmounted properly
Last mounted on: /

Source OS: Linux
Dynamic Structure
Compat Features: Journal, Ext Attributes, Resize Inode, Dir Index
InCompat Features: Filetype, Needs Recovery, Extents, Flexible Block Groups,
Read Only Compat Features: Sparse Super, Large File, Huge File, Extra Inode Size

Journal ID: 00
Journal Inode: 8

METADATA INFORMATION
-----
Inode Range: 1 - 1245185
Root Directory: 2
Free Inodes: 1060672
Inode Size: 256
Orphan Inodes: 1075692, 1075695, 1050112, 1075399, 661003, 661002, 1075552, 1075551, 661000, 657456, 657455, 923594, 923
075057, 1075032, 1065168, 1075034,

CONTENT INFORMATION
-----
Block Groups Per Flex Group: 16
Block Range: 0 - 4980479
Block Size: 4096
Free Blocks: 3883172

BLOCK GROUP INFORMATION
-----
Number of Block Groups: 152
Inodes per group: 8192
Blocks per group: 32768

Group: 0:
  Block Group Flags: [INODE_ZEROED]
  Inode Range: 1 - 8192
  Block Range: 0 - 32767
  Layout:
    Super Block: 0 - 0
    Group Descriptor Table: 1 - 2
    Group Descriptor Growth Blocks: 3 - 1024
    Data bitmap: 1025 - 1025
    Inode bitmap: 1041 - 1041
    Inode Table: 1057 - 1560
    Data Blocks: 9249 - 32767
  Free Inodes: 171 (2%)
  Free Blocks: 22299 (68%)
  Total Directories: 1105
  Stored Checksum: 0x6D87

Group: 1:
  Block Group Flags: [INODE_ZEROED]
  Inode Range: 8193 - 16384
  Block Range: 32768 - 65535
  Layout:
    Super Block: 32768 - 32768
    Group Descriptor Table: 32769 - 32770
    Group Descriptor Growth Blocks: 32771 - 33792
    Data bitmap: 1026 - 1026
    Inode bitmap: 1042 - 1042
    Inode Table: 1569 - 2080
    Data Blocks: 33793 - 65535
  Free Inodes: 4632 (56%)
  Free Blocks: 150 (0%)
  Total Directories: 45
  Stored Checksum: 0xF9ED
```

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
++++ PhysicalDrive0: UBOX HARDDISK ++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rovnix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

Carving

- Extraction des données (depuis une image disque ou mémoire)
- Basée sur des signatures
- Outils importants:
 - Foremost: <http://foremost.sourceforge.net/>
 - **Bulk_extractor**. Retrouve beaucoup de données!
https://github.com/simsong/bulk_extractor
<http://volatility-labs.blogspot.fr/2015/01/incorporating-disk-forensics-with.html?m=1>
 - Photorec: <http://www.cgsecurity.org/wiki/PhotoRec>
 - Et grep...

Agenda

- Introduction
- Définir les partitions
 - MBR
 - GPT
- Système de fichiers
 - FAT32
 - ExFAT
 - NTFS
 - EXT4
- Carving
- The Sleuth Kit



Tigzy @TigzyRK · 24m

Soon a big improvment in #RogueKiller #MBR section, with #VBR scanner.

```
++++ PhysicalDrive0: UBOX HARDDISK ++++
---- User ----
[MBR] c708b764ca9daa4f8f33e4e8b3b517da
[BSP] f4eb87199eee8a432bb482bb55118447 : Windows XP
Partition table:
0 - [ACTIVE] NTFS (0x7) [VISIBLE] Offset (sectors): 63 ! Size: 4086 MB
!UBR Bootstrap: 644c40d310a73426ba4e8ff7940ac5bb : Windows XP
!UBR Bootloader: a052dae9d7664ba78ae9c733fc55d79e : Rounix
User = LL1 ... OK
User = LL2 ... OK
```



Expand

The Sleuth Kit

- Bibliothèque et ensemble d'outils créés par Brian Carrier pour analyser les systèmes de fichiers
- Supporte NTFS, ExFAT, EXT4, HFS, ...
- Fonctionne sous Windows et Linux
- Ensemble d'outils pouvant être combinés entre eux (comme « sort | cut | less »)
- Outils organisés par niveaux du système de fichier

TSK: principaux outils

Niveau image / secteurs	
mmls	Affiche les partitions d'un disque
img_cat	Affiche les secteurs
Niveau système de fichiers	
fsstat	Affiche les détails d'une partition
Niveau bloc de données (préfixe blk = cluster)	
blkstat	Statistiques à propos d'un bloc
blkls	Liste le contenu des bloc effacés
blkcalc	Calcule la correspondance entre l'espace non alloué et l'espace complet
blkcat	Récupère le contenu d'un bloc
Niveau métadonnées (préfixe <i>i</i> comme inode)	
istat	Statistiques à propos d'un inode (entrée dans la table des fichiers)
icat	Contenu des blocks associé à un inode
ifind	Block -> inode match
Niveau fichier	
fls	Liste les fichiers et répertoire
ffind	Fichier -> inode

TSK: mmls (image)

```
$ mmls disk3.001
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors
```

	Slot	Start	End	Length	Description
000:	Meta	0000000000	0000000000	0000000001	Primary Table <#0>
001:	-----	0000000000	0000000038	0000000039	Unallocated
002:	000:000	0000000039	0000121855	0000121817	DOS FAT16 <0x06>

mmls analyse la MBR/GPT et fourni des informations sur les partitions

TSK: fsstat (partition)

```
$ fsstat -o 2048 disk.E01
FILE SYSTEM INFORMATION
-----
File System Type: NTFS
Volume Serial Number: 2E5646AD5646761D
OEM Name: NTFS
Version: Windows XP

METADATA INFORMATION
-----
First Cluster of MFT: 786432
First Cluster of MFT Mirror: 2
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 59648
Root Directory: 5

CONTENT INFORMATION
-----
Sector Size: 512
Cluster Size: 4096
Total Cluster Range: 0 - 15728126
Total Sector Range: 0 - 125825022

$AttrDef Attribute Values:
$STANDARD_INFORMATION <16>   Size: 48-72   Flags: Resident
$ATTRIBUTE_LIST <32>         Size: No Limit  Flags: Non-resident
$FILE_NAME <48>              Size: 68-578   Flags: Resident,Index
$OBJECT_ID <64>              Size: 0-256    Flags: Resident
$SECURITY_DESCRIPTOR <80>    Size: No Limit  Flags: Non-resident
$VOLUME_NAME <96>            Size: 2-256    Flags: Resident
$VOLUME_INFORMATION <112>    Size: 12-12    Flags: Resident
$DATA <128>                  Size: No Limit  Flags:
$INDEX_ROOT <144>            Size: No Limit  Flags: Resident
$INDEX_ALLOCATION <160>       Size: No Limit  Flags: Non-resident
$BITMAP <176>                Size: No Limit  Flags: Non-resident
$REPARSE_POINT <192>         Size: 0-16384   Flags: Non-resident
$EA_INFORMATION <208>        Size: 8-8       Flags: Resident
$EA <224>                    Size: 0-65536   Flags:
$LOGGED_UTILITY_STREAM <256> Size: 0-65536   Flags: Non-resident
```


TSK: fls

```
$ fls -o 2048 disk.E01
r/r 4-128-4:      $AttrDef
r/r 8-128-2:      $BadClus
r/r 8-128-1:      $BadClus:$Bad
r/r 6-128-4:      $Bitmap
r/r 7-128-1:      $Boot
d/d 11-144-4:     $Extend
r/r 2-128-1:      $LogFile
r/r 0-128-1:      $MFT
r/r 1-128-1:      $MFTMirr
d/d 57-144-1:     $Recycle.Bin
r/r 9-128-8:      $Secure:$SDS
r/r 9-144-16:     $Secure:$SI1
r/r 9-144-17:     $Secure:$SDH
r/r 10-128-1:     $UpCase
r/r 3-128-3:      $Volume
d/d 58094-144-5:  Boot
r/r 58147-128-1:  bootmgr
r/r 58158-128-3:  BOOTSECT.BAK
d/d 13672-144-1:  Documents and Settings
r/r 58183-128-1:  pagefile.sys
d/d 58-144-1:     PerfLogs
d/d 60-144-6:     Program Files
d/d 235-144-6:    Program Files <x86>
d/d 353-144-6:    ProgramData
d/d 22730-144-1:  Recovery
d/d 15963-144-6:  System Volume Information
d/d 443-144-5:    Users
d/d 602-144-5:    Windows
d/d 59648:        $OrphanFiles
```

- Liste les entrées de la racine, en parcourant la MFT

- « r/r 0-128-1: \$MFT »

r/r: regular file, d/d: directory

0-128-1 = address-type-id = entrée#0 – type128 – id1, voir [MetadataAddress](#)

TSK: fls

```
usage: fls.exe [-adDFlpruvV] [-f fstype] [-i imgtype] [-b dev_sector_size] [-m dir/] [-o imgoffset] [-z ZONE] [-s seconds] image [images] [inode]
```

If [inode] is not given, the root directory is used

-a: Display "." and ".." entries

-d: Display deleted entries only

-D: Display only directories

-F: Display only files

-l: Display long version (like ls -l)

-i imgtype: Format of image file (use '-i list' for supported types)

-b dev_sector_size: The size (in bytes) of the device sectors

-f fstype: File system type (use '-f list' for supported types)

-m: Display output in mactime input format with
dir/ as the actual mount point of the image

-o imgoffset: Offset into image file (in sectors)

-p: Display full path for each file

-r: Recurse on directory entries

-u: Display undeleted entries only

-v: verbose output to stderr

-V: Print version

-z: Time zone of original machine (i.e. EST5EDT or GMT) (only useful with -l)

-s seconds: Time skew of original machine (in seconds) (only useful with -l & -m)

TSK: fls

fls permet également de produire un résultat sous la format “body”, utilisable par l'utilitaire **mactime**, afin de faire un “timeline” des événements du système de fichiers

```
$ fls -r -o 39 -m "j" disk.001 >fls_bodyfile
```

```
$ mactime.exe -b fls_bodyfile
```

Xxx Xxx 00 0000 00:00:00	2507122	..c.	r/rrwxrwxrwx	0	0	12	j/IMG_20150118_173555.jpg (deleted)
	0	.acb	r/rrwxrwxrwx	0	0	3	j/EOS_DIGITAL (Volume Label Entry)
	3099843	..c.	r/rrwxrwxrwx	0	0	6	j/IMG_20150118_173559.jpg (deleted)
	2645739	..c.	r/rrwxrwxrwx	0	0	9	j/IMG_20150118_173550.jpg
Sun Jan 18 2015 00:00:00	2507122	.a..	r/rrwxrwxrwx	0	0	12	j/IMG_20150118_173555.jpg (deleted)
	3099843	.a..	r/rrwxrwxrwx	0	0	6	j/IMG_20150118_173559.jpg (deleted)
	2645739	.a..	r/rrwxrwxrwx	0	0	9	j/IMG_20150118_173550.jpg
Sun Jan 18 2015 17:20:10	0	m...	r/rrwxrwxrwx	0	0	3	j/EOS_DIGITAL (Volume Label Entry)
Sun Jan 18 2015 17:35:52	2645739	m...	r/rrwxrwxrwx	0	0	9	j/IMG_20150118_173550.jpg
Sun Jan 18 2015 17:35:56	2507122	m...	r/rrwxrwxrwx	0	0	12	j/IMG_20150118_173555.jpg (deleted)
Sun Jan 18 2015 17:36:00	3099843	m...	r/rrwxrwxrwx	0	0	6	j/IMG_20150118_173559.jpg (deleted)
Sun Jan 18 2015 17:38:00	3099843	...b	r/rrwxrwxrwx	0	0	6	j/IMG_20150118_173559.jpg (deleted)
Sun Jan 18 2015 17:38:06	2645739	...b	r/rrwxrwxrwx	0	0	9	j/IMG_20150118_173550.jpg
Sun Jan 18 2015 17:38:11	2507122	...b	r/rrwxrwxrwx	0	0	12	j/IMG_20150118_173555.jpg (deleted)

Exercice: Carving avec TSK

- On possède une image disque « disk3.001 » et l'on cherche les images Jpeg effacées
- Une image Jpeg commence par les 3 octets suivants: 0xff, 0xd8, 0xff
- Nous allons donc faire du **carving** (recherche par ce motif « ffd8ff ») pour trouver le premier cluster des images effacées. On suppose donc que les clusters de ces images sont contigus
- On va d'abord travailler sur l'espace non alloué qui contient peut être, par chance, ces images effacées

Docker, Overlay2

- Comment analyser le contenu d'un Docker et son système de fichiers ?
- <https://fr.slideshare.net/JoelLathrop2/docker-forensics>

Autres systèmes de fichiers et de stockages

ReFS: Resilient File System (Win 2012)

<https://github.com/libyal/libfsrefs>

<https://github.com/Dafti/pyrefs>

APFS (MacOS 10.12):

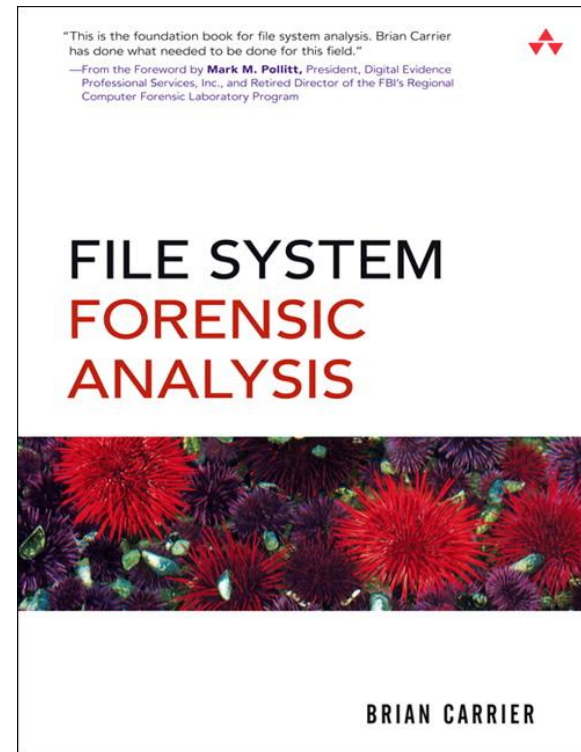
https://www.researchgate.net/publication/319573636_Decoding_the_APFS_file_system

Logical Volume Manager:

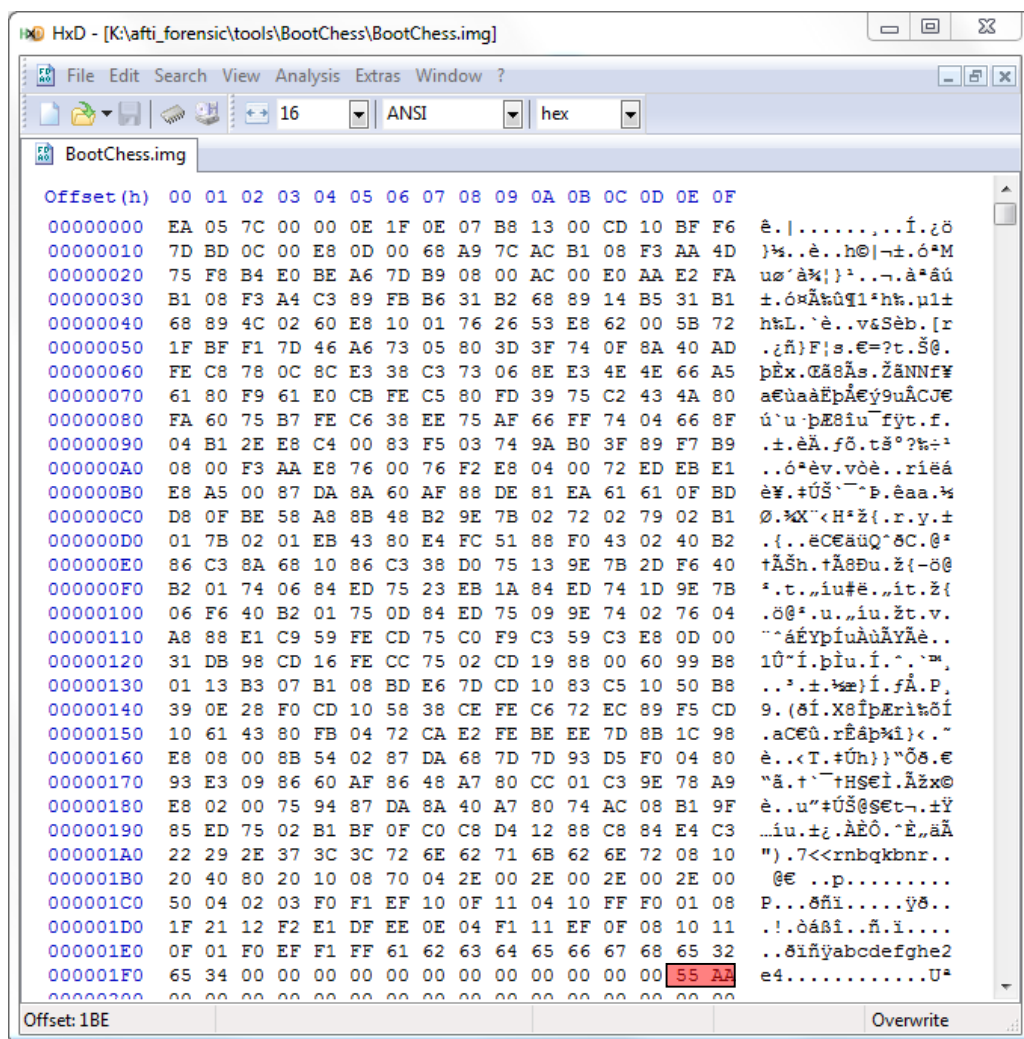
http://mo.morsi.org/blog/2016/03/29/LVM_Internals/

Références additionnelles

- [File system Forensic analysis](#),
Brian Carrier.



Bonus: que contient ce secteur?



- BootChess

<http://www.pouet.net/prod.php?which=64962>