Les disques durs : fonctionnement général et comment en faire une analyse forensique

HackademINT

Table des matières

1 Le disque dur, c'est quoi déjà ?

1.1 Introduction générale

Vous le savez déjà, le disque dur est un composant essentiel de votre ordinateur. Sans lui, impossible de stocker des informations au-delà de l'extinction de votre PC (ce qui, avouons-le, est un peu dommage).
Il s'agit basiquement d'un endroit où sont écrites des données de telle manière qu'elle ne dépendent pas d'une alimentation électrique pour subsister (une telle mémoire est dite *persistante*, à l'inverse de la mémoire RAM par exemple qui est dite *volatile*).
L'appellation "disque dur" a une origine historique qui n'est plus tout à fait vraie aujourd'hui. En effet, nous retrouvons de nos jours deux types de disques :

- Les disques durs (abrégés HDD, Hard Disk Drive) sont de véritables disques en métal, magnétiques, sur lesquels sont stockés de l'information à l'aide de la polarisation : en fonction de si elle est positive ou négative, cela permet de représenter des bits (0 ou 1) et donc de l'information. C'est une tête de lecture qui se déplace sur le disque pour lire / écrire les informations.

FIGURE 1 – Un disque dur HDD ouvert.

- Les disques "électroniques" (nommés SSD, Solid State Drive) sont plus récent et stockent l'information de manière électrique (le fonctionnement détaillé est trop complexe pour être détaillé ici). Leur principal avantage est d'être bien plus rapides que les HDD, car toute leur information est accessible en temps constant, alors que la tête de lecture doit se déplacer à chaque nouvelle lecture pour les HDD.

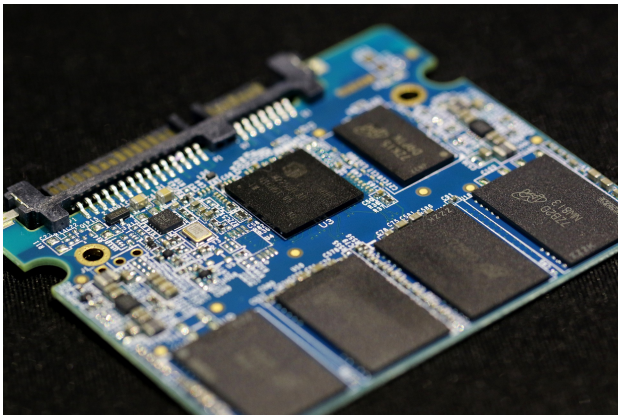

FIGURE 2 – L'intérieur d'un SSD.

Néanmoins, quelque soit le disque utilisé, le fonctionnement du stockage des informations est strictement identique.

1.2 Les connecteurs

C'est bien beau d'avoir un disque dur, mais pour qu'il serve à quelque chose, il doit être relié au reste de l'ordinateur. Pour cela, il y a les *connecteurs*. Il y en a plusieurs :

- USB (Universal Serial Bus) : vous le connaissez tous, c'est le standard des disques durs externes. Il y en a plusieurs versions, les dernières (3 et 3.1) étant les plus rapides.
- l'IDE (Integrated Drive Electronics) est un vieux connecteur que l'on peut retrouver dans de vieilles tours, mais qui n'est plus utilisé. En plus d'être lent, l'IDE avait un problème majeur : son organisation sous forme de "nappes" avait pour conséquence de diminuer la ventilation dans les PC et d'accumuler la poussière, menant à des problèmes de surchauffe si la tour n'était pas nettoyée de temps en temps (ce que quasiment personne ne fait). C'est en partie grâce à l'IDE que toutes les connectiques, aujourd'hui, sont plutôt regroupées par câbles et séparées en cas de nombreux fils.

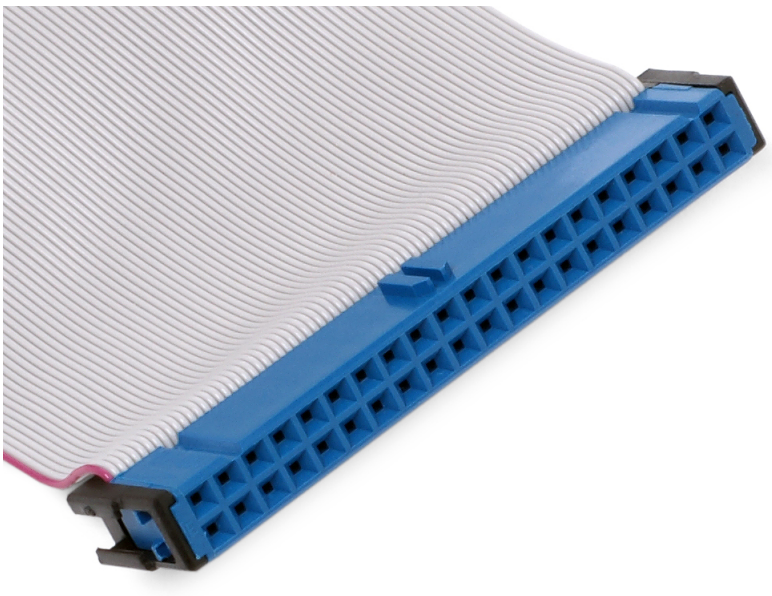

FIGURE 3 – UNe nappe IDE et son connecteur.

- Le SATA (Serial Advanced Technology Attachment) est le successeur de l'IDE. Il est présent partout, et a connu plusieurs évolutions. Il a connu plusieurs évolutions, notamment en termes de vitesse, au fil des années.

FIGURE 4 – Un connecteur SATA.

2 Les données sur un disque dur

2.1 Les partitions

Vous vous en doutiez peut-être (ou pas), mais vos données ne sont pas écrites en vrac sur votre disque dur. Plus étonnant peut-être, elles ne sont pas écrits avec la même organisation que vos documents.
En fait, il existe plusieurs systèmes de fichiers qui déterminent comment sont rangés vos fichiers sur le disque dur. De plus, votre disque dur peut être découpé en plusieurs *partitions* : il s'agit

de morceaux indépendants de votre disque, qui peuvent utiliser de systèmes de fichiers différents. Cela est utile pour par exemple un PC avec dual boot, mais aussi pour plein d'autres choses. Voici un exemple de différentes partitions que vous pouvez avoir sur un disque :

- Des partitions de données : classique, c'est là où sont stockées vos données. Elle est associée à un système de fichiers, qui sont détaillés dans la partie suivante.
- La swap : sous Linux, la swap est une partition qui sert, si nécessaire, à étendre la mémoire vive de votre PC : c'est utile si vous utilisez des logiciels particulièrement gourmands, ou si vous mettez votre PC en mode hivernage.
- La partition de boot : c'est la partition qui est lue par votre BIOS et qui lui permet de savoir que faire pour lancer le système d'exploitation. Dans le cas de Linux, cette partition contient GRUB (ou équivalent).
- Les partitions de crash Windows : présentes uniquement sur les systèmes Windows, ces partitions permettent de récupérer des données en cas de crash (écran bleu) et ont d'autres fonctionnalités. Leur fonctionnement exact n'est pas très connu, il est donc recommandé de ne pas y toucher.

2.2 Le système de fichier

On a parlé des systèmes de fichiers dans la partie précédente, mais revenons-y en détails : un système de fichiers permet, sur le disque dur, de gérer de nombreuses choses : gestion de la lecture / écriture, de l'espace disque, des noms de fichiers, des métadonnées, de l'arborescence, des permissions... Il en existe plusieurs, dont les principaux sont présentés juste après.
Au vu des multiples tâches accomplies par un système de fichiers et de leurs fonctionnement variés il est assez difficile de les comparer, mais un critère en particulier ressort : celui de la *fragmentation*. En effet, à force de (ré)écrire en permanence des données sur un disque, il arrive que les données se retrouvent éparpillées partout sur le disque (ce problème est surtout lié aux HDD). Ainsi, la tête de lecture devant plus se déplacer pour atteindre les données, le temps de lecture et d'écriture était rallongé, et le seul moyen d'y remédier était de défragmenter le disque, c'est-à-dire tout rassembler et réécrire au même endroit, une opération longue et coûteuse.

- FAT (File Allocation Table) est le plus vieux système de fichiers présenté ici. Développé en 1977 pour les disquettes, il a été ensuite adapté pour les disques durs. Il en existe 3 versions majeures : FAT12, FAT16 et FAT32. Ce système n'est quasiment plus utilisé aujourd'hui (par exemple, le maximum de données pouvant être stocké sur un FAT16 est de 4Go...), et est connu pour énormément se fragmenter.
- l'exFAT (Extensible File Allocation Table), appelé improprement FAT64 (ce n'est pas une évolution de FAT !) est un système de fichiers développé par Microsoft et resté propriétaire jusqu'en 2019. Il est aujourd'hui le standard sur beaucoup de cartes SD (appareils photo, consoles de jeu...), mais pas pour les téléphones portables qui utilisent d'autres formats pour diverses raisons.
- le format ext (Extended File System) a été créé spécifiquement pour le noyau Linux. Il en existe 4 versions : ext n'a quasiment jamais été utilisé, ext2 a été très répandu dans les années 90, ext3 a pris le relais pour les années 2000 avec de nombreuses améliorations en termes de performances (notamment pour la fragmentation), et ext4 est présent depuis les années 2010 sur tous les Linux récents. Ext3 et ext4 sont connus pour très peu se fragmenter.
- NTFS (New Technology File System) est un système propriétaire, développé par Microsoft pour Windows, à partir de Windows NT. Il en existe plusieurs versions, mais c'est ce système qui est toujours utilisé sur Windows aujourd'hui. Il a connu moins d'évolutions que ext et est moins performant en termes de fragmentation.

2.3 Les inodes

Les inodes sont des structures de données particulières qui stockent des données sur les fichiers autres que leurs données brutes. Ils ne sont pas présents sur tous les systèmes de fichiers, mais au moins sur ext. Ils permettent de stocker, en fonction des versions, des métadonnées, le nom du fichier, son créateur, ses permissions, sa taille, la date de sa création...
Chaque inode a un numéro d'identifiant unique qui correspond à un seul fichier dans l'ensemble de la partition. On peut voir les numéros d'inode de ses fichiers sous Linux avec ls -i.
Les inodes ont une importance vitale en forensics, car c'est eux qui détiennent des informations vitales sur le fichier, et notamment **s'il a été supprimé**. En effet, lors de la suppression d'un fichier, celui-ci ne disparaît pas du disque dur : on indique simplement, dans l'inode, que ce fichier n'est plus utilisé, ce qui permet au système de fichiers de réécrire dessus s'il a besoin de place, et

qui fait qu'on ne le "voit plus" dans le navigateur de fichiers standard.
C'est par exemple le fonctionnement de la "Corbeille" sous Windows et certaines distributions Linux : ce dossier spécifique liste en fait les fichiers toujours présents sur le disque mais dont les inodes ont été marqués comme supprimés, ce qui permet de les restaurer facilement. A noter que "vider la Corbeille" ne supprime pas plus vos fichiers qu'une suppression normale...
Ainsi, tant qu'un fichier "supprimé" ne se fait pas réécrire dessus, ses données restent disponibles sur le disque, et si votre explorateur de fichier standard ne vous l'affichera pas, en regardant le contenu brut de votre disque, vous pourrez récupérer votre fichier...

3 Disque dur et Linux

3.1 Localisation et convention de nommage

Sous Linux, comme vous le savez sûrement déjà, "tout est fichier" : cela signifie que c'est aussi le cas pour votre disque dur et ses partitions. La plupart des éléments de votre ordinateur sont en fait regroupés dans le dossier /dev (pour devices) : cpu, souris, clavier, entrées USB... et votre disque.
Pour simplifier la manipulation de différents disques sous Linux, une convention de nommage a été mise en place, ce qui permet de savoir quel disque on manipule dès le premier coup d'oeil :

- Les deux premières lettres (fd, hd, sd) correspondent à la connectique (ce qui signifie que sd et hd n'ont rien à voir avec SSD et HDD ! !). fd est pour "floppy disk" ou disquette et a disparu aujourd'hui, hd correspond à une connection IDE et sd à une connection SATA (et éventuellement d'autres).
- La lettre suivante correspond au numéro du disque. Ainsi hda correspond à votre premier disque IDE, hdb à votre deuxième, etc.
- Le numéro qui suit correspond aux partitions. sda3 représente donc la troisième partition de votre premier disque SATA. Simple, non ?

> Il est possible de retrouver d'autres nommages, notamment pour les connecteurs. Ainsi, ceux qui possède un PC avec un SSD récent ont peut-être un disque du genre nvme0n1 : cela correspond à une connectique récente (NVME), qui apparaît de plus en plus sur les SSD car elle est plus rapide et permet de mieux exploiter les possibilités d'un gros SSD (notamment en termes de parallélisme).

3.2 Montage d'une partition

Nous avons vu que le disque dur est découpé en partitions, chacune avec un rôle spécifique. Lorsque vous accédez à un disque dur (ou même à une clé USB), vous accédez en fait à une partition et non au disque directement (vous n'avez peut-être jamais fait la différence, mais sur vos clés USB, il y a en fait une unique partition qui est rendue accessible automatiquement, mais c'est bien une partition et pas le contenu du disque directement).
L'opération qui consiste à choisir une partition sur un disque et à l'associer à un endroit dans votre arborescence de fichiers afin de pouvoir y accéder s'appelle *le montage*. C'est une opération particulière, qui nécessite par défaut les droits root.

> Attention : même si la plupart des systèmes d'exploitation le font automatiquement, cela reste une assez mauvaise idée en termes de cybersécurité de monter une clé USB externe automatiquement : s'il s'agit d'une clé malicieuse, elle a ainsi accès directement au PC de la victime, et ce parfois alors que le PC est en veille avec l'écran verrouillé ! Monter les clés USB à la main et taper son mot de passe à chaque fois peut paraître long et fastidieux, mais cela vous assure que seules les clés USB que vous avez explicitement autorisées ont accès à votre PC.

Par défaut, sous Linux, au moins deux partitions sont montées au démarrage de votre PC. Il s'agit de votre partitions de données, montée par défaut à la racine (/), et de la partition de boot, accessible dans /boot. Le fichier de configuration des partitions des disques internes et comment elles doivent être montées est contenu dans /etc/fstab. Il est lu à chaque démarrage par Linux pour savoir ce qu'il doit faire pour mettre en place votre système de fichiers. Ne le modifiez pas ou vous risquez fort de casser votre Linux ! !

Pour monter / démonter une clé USB, il existe des commandes dédiées que nous verrons juste après. Il est important de monter une partition dans un dossier vide : sinon, les dossiers contenus dans le dossier seront cachés tant que vous ne démonterez pas la partition. Par convention, les partitions sont montées dans le dossier /mnt lorsqu'elles sont montées à la main, et dans /media lorsqu'elles sont montées automatiquement par votre OS, mais vous pouvez monter une partition presque n'importe où dans votre système de fichiers.

3.3 Les outils

Il existe de nombreux outils, sous Linux, qui permettent de manipuler des disques durs. Je vais présenter les plus classiques. Pour Windows, l'utilitaire principal est le gestionnaire de disques, présent par défaut.

- mount : pour monter une partition, on utilise la commande mount (aucune excuse pour ne pas la connaître celle-là). La syntaxe est simple :
  mount <partition> <dossier cible>
  Par exemple, mount /dev/sda1 /mnt/monDD permet de monter votre première partition de votre premier disque dans /mnt/monDD. Gardez à l'esprit que cette commande nécessite les droits root.
- umount : permet de démonter une partition en l'indiquant :
  umount <partition>
  Cela revient à "éjecter" la clé sous Windows. Cette commande échouera si votre partition est occupée (opérations de lecture ou d'écriture). Il est important de bien démonter les partitions non utilisées, afin de ne pas perdre de données.
- lsblk : il s'agit du ls des disques et partitions. Son avanatge (énorme dans certains cas) est qu'il ne nécessite pas les droits root.
- fdisk : il fait partie d'un groupe plus étendu d'outils (fdisk, cfdisk, sfdisk...) qui permettent de faire beaucoup de choses sur les disques / partitions, de les déplacer, redimensionner... Notamment fdisk -l fait environ la même chose que lsblk. Ces commandes nécessitent d'être root.
- parted : la commande parted est la commande historique pour modifier ses partitions. Elle permet de faire l'essentiel en termes de management de partitions et est assez claire d'utilisation. A noter que pour modifier une partition (comme avec n'importe quel outil), il faut qu'elle ne soit pas montée.
- gparted : parted mais en interface graphique. Autrement plus agréable, plus visuel et permet en général de faire moins de bêtises.
- dd : c'est le "cp des disques durs". Il ne copie pas des fichiers ou des dossiers, mais des partitions complètes, et surtout copie *l'intégralité du contenu bit à bit*, ce qui signifie aussi les fichiers corrompus, effacés... Bref tout le contenu du disque. C'est la commande par excellence pour faire un dump de disque dur pour l'analyser plus tard. Sa syntaxe de base est simple :
  dd if=<entrée> of=<sortie> [bs=<taille des blocs>]
  Le paramètre bs correspond à la taille des blocs que vous voulez copier. Par défaut, dd copie votre disque octet par octet, ce qui peut le ralentir. Il n'existe pas de valeur parfaite car cela dépend des disques et des PC, mais bs=512, bs=1024, bs=2048 ou bs=4096 sont des tailles fréquemment utilisées. Exemple : dd if=/dev/sdb2 of=/tmp/dump dd=1024
- shred : c'est lui, le véritable rm auquel vous pensiez lorsque vous étiez jeunes et innocents. shred efface votre fichier, mais réécrit aussi plusieurs fois dessus pour s'assurer de bien le faire disparaître. La syntaxe de base est :
  shred <fichier>
  Il possède aussi des options permettant de choisir le nombre de fois que vous voulez réécrire, si vous voulez camoufler le shredding, comment il doit être supprimé dans le système de fichiers... Bref, enfin une commande qui vous permet de faire disparaître des fichiers pour de bon.

4 Le forensic sur des disques durs

4.1 Pourquoi analyser des disques durs ?

Il existe de multiples raisons de vouloir analyser un disque dur. Récupérer des données corrompues ou effacées, obtenir de nouvelles informations sur un fichier, mais aussi des choses plus avancées comme retrouver un rootkit qui utilisait un module kernel pour se cacher du système

d'exploitation...
Cette analyse peut être complexe pour diverses raisons. L'un des problèmes les plus importants est la taille : si vous analysez un disque complet, vous risquez d'avoir vite des difficultés de place : avec un dump d'un disque de 2To, difficile de le déplacer ou de travailler directement sur sa machine...

4.2 Les outils : The Sleuth Kit et Autopsy

L'outil de prédilection pour analyser un dump de disque dur est la suite d'outils en ligne de commande The Sleuth Kit (TSK), couplé si souhaité avec l'interface graphique associée Autopsy. Ces outils sont open source si jamais vous souhaitez vous intéresser de plus près à leur fonctionnement.
TSK fonctionne très bien et peut être installé assez facilement. Sa liste de commandes est claire et concise et son manuel bien écrit. Les commandes les plus utilisées sont fls (comme pour ls mais sur une image disque), et fcat (comme cat, mais aussi sur une image disque). Pour fonctionner, ces commandes ne prennent pas des noms de fichiers ou des chemins, mais des numéros d'inode (qui sont affichés par défaut par fls).
Le seul problème de cette approche est que ca devient vite laborieux d'explorer un disque dur complet à coup de commandes. Autopsy répond à ce problème en proposant une interface graphique pratique, rapide et simple d'utilisation. Son interface ressemble à un explorateur de fichiers classique, avec des onglets supplémentaires permettant d'exploiter toutes les informations que peut apporter TSK.

5 Pour aller plus loin

Vous avez maintenant les bases de l'analyse de disques. Toutefois vous allez vous rendre compte qu'il existe bien plus de choses pouvant se retrouver sur un disque dur que ce qui est présenté ici. J'en fait une petite présentation rapide, sans rentrer dans les détails, pour que vous sachiez à quoi vous attendre et que vous preniez les bons réflexes.

5.1 Les formats

Le dump effectué par dd est en fait un format dit "brut". Il existe différents formats de dump de disque, certains compressés. Autopsy en lit une bonne partie, mais il vous sera parfois nécessaire de convertir. La commande file sur votre dump devrait vous dire de quel format il s'agit : ensuite, Google est votre ami. Les formats de dump les plus connus sont l'EWF et l'AFF.

5.2 RAID

Le RAID (Redundant Array of Independent Disks) est une technologie de virtualisation qui permet de transformer plusieurs disques physiques en un seul stockage virtuel, pour des objectifs de redondance, de performance, ou autre. Il en existe plusieurs "versions" (RAID1, RAID2, RAID4, RAID5...) qui ont des objectifs et des technologies différents.
L'un des avantages du RAID (certaines versions) est que si un disque meurt, le système continue de fonctionner et dès que le disque est remplacé, les données sont redupliquées et le RAID se répare tout seul.
Pour analyser un disque physique contenant du RAID, il faut d'abord se renseigner sur la version, son fonctionnement, puis trouver les outils adéquats pour manipuler l'image.

5.3 LUKS et autres moyens de chiffrement

C'est bien facile de retrouver des données sur un disque en clair, mais c'est une autre paires de manches si celui-ci est chiffré. Il existe de nombreux moyens de chiffrement, certains vieux et potentiellement vulnérables, d'autres considérés incassables.
Pour identifier si un disque est chiffré, il suffit de regarder ce qu'il y a dessus : comme il n'y aura pas de système de fichiers apparents, tous les outils standards râleront et la plupart d'entre eux vous signaleront un message du type "high entropy detected". Cela signifie en gros que c'est le bazar sur votre disque, ce qui signifie soit que le disque a été shreddé, soit qu'il est chiffré.
Pour identifier le système de chiffrement, vous pouvez tenter la commande file, ou de monter l'image. Si mount reconnaît le format, il vous le dira dans l'erreur. Sinon, éditeur hexadécimal pour trouver des indices (chaînes de caractères, hash...). Un chiffrement classique est LUKS (Linux

Unified Key Setup), qui est le chiffrement par défaut de Linux.
Sur un disque chiffré, il y a en général un mot de passe. Peu importe le chiffrement, si le mot de passe est faible, vous avez peut-être votre chance. Attention, par sécurité les hash utilisés sur les derniers systèmes de chiffrement sont très longs. Trouvez un outil de bruteforce adapté ou extrayez le hash, et googlez-le ou lancez hashcat / john avec une liste adaptée au type de hash.