

Digital Forensics: Evidence Collection and Preservation Procedures

Ce texte traite de la **collecte et de la conservation des preuves** dans le cadre d'une enquête de **criminalistique numérique**. Il souligne l'importance cruciale de maintenir une **chaîne de possession** rigoureuse et une documentation exhaustive pour garantir l'admissibilité des éléments devant les tribunaux. Les sources détaillent les responsabilités du **premier intervenant**, notamment la sécurisation du périmètre et le choix des ressources internes ou externes. Le document explique également les aspects techniques de la **duplication bit à bit** et les spécificités liées aux divers supports de stockage, tels que les disques **SSD, RAID ou objets connectés**. Enfin, il insiste sur la nécessité d'obtenir les **autorisations légales** appropriées avant toute analyse afin d'éviter la contamination des preuves originales.

Pourquoi la chaîne de possession est-elle cruciale juridiquement ?

La chaîne de possession est cruciale sur le plan juridique pour plusieurs raisons fondamentales :

1. **Prouver l'absence de contamination** : Devant un tribunal, la recevabilité d'une preuve repose sur la capacité à démontrer, **hors de tout doute raisonnable**, que l'élément a été trouvé et préservé sans aucune contamination durant l'enquête. Toute modification non documentée à partir de la première intervention peut être interprétée comme une altération due à l'enquête elle-même, ce qui fragilise grandement la crédibilité des constatations.
2. **Éviter le rejet des preuves par la cour** : Si la défense soulève une objection ou conteste la manière dont un élément a été manipulé et que l'enquêteur est incapable d'y répondre avec exactitude, cela peut entraîner **le rejet de vos preuves** par le tribunal.
3. **Garantir une traçabilité totale et écrite** : Par définition, la chaîne de possession est la **documentation chronologique et la traçabilité écrite** détaillant chaque étape de la vie de la preuve :

sa saisie, sa garde, son contrôle, son transfert, son analyse et sa disposition finale. Elle exige l'utilisation d'un document standardisé comprenant un système d'étiquetage unique, l'identification claire de chaque personne impliquée, ainsi que les dates et heures précises de chaque prélèvement, interaction ou transfert.

4. **Valider la conformité des copies de travail** : Travailler directement sur un support de preuve original est proscrit car cela le contamine et le rend généralement inutilisable juridiquement. Grâce à une chaîne de possession rigoureuse, les enquêteurs peuvent prouver que l'appareil d'origine n'était pas contaminé avant et pendant la réalisation de copies forensiques conformes. Ainsi, même si l'appareil d'origine subit des modifications ultérieures, les conclusions issues des copies d'images bit à bit restent pleinement exploitables devant la justice.

 Je peux vous expliquer comment les experts utilisent concrètement le blocage en écriture et le hachage pour valider scientifiquement que ces copies forensiques restent strictement identiques aux originaux.

Comment fonctionnent le blocage en écriture et le hachage ?

Le blocage en écriture

Le blocage en écriture empêche de manière stricte toute action susceptible de modifier une copie ou un élément de preuve. En criminalistique numérique, toute modification apportée à une copie la rend immédiatement inutilisable sur le plan légal devant un tribunal. Le blocage en écriture fonctionne comme un **mode « lecture seule »** qui supprime toute possibilité d'écriture sur le support.

Il existe deux grands types de solutions :

- **Les bloqueurs physiques (matériels)** : Ce sont des boîtiers d'interface branchés entre le disque d'origine et la station d'analyse. Ils appliquent la protection de deux manières : soit en **autorisant uniquement les commandes qui ne modifient pas les données**, soit en **bloquant spécifiquement la commande d'écriture** tout en laissant passer toutes les autres.
- **Les bloqueurs logiciels** : Ce sont des fonctionnalités de protection directement intégrées au sein des logiciels

d'investigation numérique pour protéger les copies de travail en cours d'examen.

Le cas particulier des disques SSD : Bien que les bloqueurs d'écriture matériels bloquent efficacement les commandes externes de modification (comme la commande TRIM), ils ne peuvent pas empêcher les algorithmes internes de l'appareil (comme l'équilibrage d'usure ou *wear-leveling*) de s'exécuter de manière autonome, ce qui peut altérer la structure physique de l'appareil et modifier sa signature. Pour cette raison, lors de la contestation de copies de SSD, les tribunaux et les experts s'intéressent aux **structures d'adresses logiques** plutôt qu'aux adresses physiques.

Le hachage

Le hachage permet de vérifier de manière scientifique que la copie d'un fichier ou d'un disque est **strictement identique à l'original**. Il s'agit d'un **processus unidirectionnel** (on ne peut pas recréer le fichier à partir du résultat) qui produit un **nombre unique de longueur fixe** se présentant sous la forme d'une suite de chiffres et de lettres. On peut l'assimiler à une **empreinte numérique** du fichier.

Pour être pleinement exploitable en informatique légale, le hachage repose sur **trois règles indispensables** :

1. **La valeur de hachage d'un fichier est imprévisible.**
2. **Deux fichiers différents ne peuvent pas avoir une valeur de hachage identique** (ce phénomène est appelé une « collision de hachage »).
3. **Toute modification du fichier source**, même celle d'un seul caractère ou d'une simple virgule, **génère une valeur de hachage complètement différente.**

Choix de l'algorithme : l'impact sur la crédibilité juridique

La longueur et la complexité de l'empreinte dépendent de l'algorithme utilisé :

- **MD5 et SHA-1** : Ces anciens algorithmes (de 128 et 160 bits respectivement) ont connu des **collisions de hachage**. Bien qu'ils restent utiles en laboratoire pour des tests rapides, leur utilisation est **fortement déconseillée dans un cadre juridique professionnel**. Une défense expérimentée peut exploiter la

moindre possibilité théorique de collision pour soulever un doute devant le tribunal et faire rejeter la preuve.

- **SHA-256 et SHA-512** : Recommandés par le NIST (*National Institute of Standards and Technology*), ces algorithmes de hachage robustes sont les **normes actuelles pour garantir l'admissibilité légale** des preuves numériques.

🔧 Je pourrais également vous expliquer la procédure concrète en 3 étapes recommandée par le NIST pour tester l'efficacité d'un bloqueur d'écriture avant de l'utiliser lors d'une enquête.

Comment tester un bloqueur d'écriture ?

Pour tester l'efficacité d'un bloqueur d'écriture (qu'il soit matériel ou logiciel), le NIST (*National Institute of Standards and Technology*) propose un protocole officiel. Cependant, une **méthode simple et rigoureuse en trois étapes** permet de valider son bon fonctionnement avant de l'utiliser dans une enquête réelle :

1. Préparation du support de test (Mode écriture activé)
 - Connectez un support de stockage de test à votre système d'analyse forensique, **sans activer le blocage en écriture**.
 - **Effacez toutes ses données et formatez-le** pour confirmer qu'il est parfaitement vierge.
 - **Copiez une quantité précise de données** sur le support, puis **supprimez-en une partie**.
 - Créez une image forensique de ce support et **calculez son hachage SHA-256**.
2. Phase de test (Mode blocage en écriture activé)
 - Déconnectez le support, **activez le blocage en écriture**, puis reconnectez-le à votre système.
 - Effectuez une série de tests pour forcer l'écriture : **tentez de copier de nouveaux fichiers, tentez d'en supprimer et tentez de formater le support**. (Le système doit idéalement rejeter ces actions ou les simuler sans modifier le support physique).
 - Créez une nouvelle image du support et **générez le hachage SHA-256 de cette deuxième image**.

3. Contrôle et validation (Mode écriture réactivé)

- **Comparez les deux hachages SHA-256** obtenus aux étapes 1 et 2 : **ils doivent être rigoureusement identiques**, ce qui prouve que vos tentatives d'écriture à l'étape 2 n'ont absolument rien modifié sur le disque.
- Désactivez le blocage en écriture (réactivez l'écriture), reconnectez le support et **supprimez volontairement une partie des données**.
- Créez une troisième image du support et calculez son hachage SHA-256.
- **Assurez-vous que ce troisième hachage est complètement différent** des deux précédents. Cela confirme que le support est bien modifiable et que l'absence de modification à l'étape 2 était uniquement due à l'efficacité du bloqueur d'écriture.

Recommandations pour la crédibilité devant les tribunaux

- **Privilégiez le matériel reconnu** : Si vous craignez que la partie adverse ne conteste un modèle de bloqueur d'écriture spécifique, utilisez des équipements déjà largement éprouvés et acceptés devant les tribunaux (tels que les dispositifs *FastBloc* de Guidance Software ou la gamme *Tableau*), ou tout autre bloqueur validé et recommandé par le NIST.
- **Faites toujours des copies doubles** : Il est recommandé de créer au moins deux copies de chaque élément de preuve. Ainsi, en cas de défaillance imprévue de votre bloqueur d'écriture lors de l'analyse, vous disposerez d'une copie de rechange intacte et vierge pour poursuivre votre travail.

 Je peux vous synthétiser les meilleures pratiques de collecte des preuves recommandées par votre document selon que le système cible est trouvé allumé ou éteint.

Format RAW vs EnCase

Le Format RAW (Brut)

- **Description** : C'est le format de duplication **le plus courant**. Il effectue une copie physique intégrale (bit à bit) qui capture l'ensemble des données du support, y compris ce qui se dissimule

dans l'**espace libre** (slack space, entre les fichiers) ou dans les **clusters de disque corrompus**.

- **Extensions et fichiers** : Les fichiers d'image s'enregistrent avec l'extension **.raw** et se présentent sous la forme de segments nommés **Evidence.001** ou **Evidence.dd1**.
- **Avantages** : Ce format offre l'avantage d'une copie généralement **très rapide** et possède la particularité d'**ignorer les erreurs de lecture** rencontrées lors du processus de copie.

Le Format EnCase Evidence

- **Description** : Contrairement au format brut, le format EnCase est un format **propriétaire** développé spécifiquement pour l'application d'investigation numérique *EnCase*. Il crée lui aussi une copie numérique complète de l'original.
- **Extensions et fichiers** : Il utilise l'extension **.e**. Les fichiers générés se nomment **Evidence.E01**, **Evidence.E02**, etc. Il existe deux versions de ce format, la version 2 ayant été introduite avec EnCase 7.
- **Métadonnées** : Il permet d'inclure un stockage, bien que **limité**, pour les métadonnées de l'acquisition.
- **Intérêt stratégique au tribunal** : Choisir le format EnCase peut être une tactique délibérée pour **tester les ressources de l'avocat adverse**. EnCase étant largement accepté par les tribunaux, l'avocat de la partie adverse se retrouve obligé d'acquérir ce même logiciel pour analyser la preuve. S'il dispose d'un budget limité ou s'il manque d'expérience en informatique légale, le coût élevé de cette suite logicielle peut le déstabiliser ou le pousser à montrer des signes de faiblesse.

*(Note : Vos sources décrivent également une troisième option courante, le format ouvert **AFF** (Advanced Forensics Format), qui permet d'ajuster la compression, de stocker des métadonnées arbitraires et de vérifier les erreurs).*

 Souhaitez-vous que je vous détaille la différence fondamentale que fait votre document entre une sauvegarde classique et une copie conforme bit à bit (comme RAW ou EnCase) ?

Sauvegarde vs copie bit à bit

La différence entre une sauvegarde classique et une copie conforme bit à bit est fondamentale en criminalistique numérique :

1. La sauvegarde classique

- **Cible sélective** : Elle se concentre uniquement sur les **données visibles et considérées comme pertinentes** par le système d'exploitation.
- **Zones ignorées** : Elle ignore de larges portions du support physique, notamment l'**espace libre** (slack space / données intercalées) ainsi que les **clusters de disque considérés comme corrompus**.
- **Risque d'enquête** : Cette méthode est insuffisante pour une investigation numérique. En effet, des preuves effacées ou des menaces informatiques redoutables (comme des **rootkits**) sont connues pour se dissimuler spécifiquement dans ces espaces non alloués ou corrompus qu'une sauvegarde classique ignore complètement.

2. La copie bit à bit (ou copie forensique)

- **Capture binaire intégrale** : Elle réalise une copie physique miroir, bit par bit, de la totalité du support de stockage. Elle ne se contente pas de copier les fichiers, elle duplique la structure brute.
- **Préservation absolue des données** : Elle englobe tout ce qui se trouve sur le disque, y compris les données actives, l'**espace libre** entre les fichiers et les **clusters corrompus**.
- **Valeur juridique devant les tribunaux** : C'est la méthode obligatoire pour toute enquête forensique. Les tribunaux exigent généralement qu'une image intégrale du disque soit capturée pour valider les conclusions et écarter toute contestation de la défense sur l'intégrité de la preuve.

 Je peux vous présenter les outils spécifiques (comme *dd*, *dcfldd* ou *ddrescue*) détaillés dans vos sources pour réaliser et optimiser ces copies bit à bit selon l'état du disque cible.

Quels outils recommande le NIST ?

Le NIST (*National Institute of Standards and Technology*) recommande l'utilisation de standards et de protocoles spécifiques pour garantir la rigueur scientifique et la recevabilité juridique des éléments de preuve numérique :

1. **Algorithmes de hachage robustes :**

- a. Le NIST recommande l'usage d'algorithmes de hachage de la famille SHA, plus particulièrement **SHA-224, SHA-256 ou SHA-512**, pour la validation professionnelle des images forensiques.
- b. À l'inverse, l'utilisation d'algorithmes obsolètes comme **MD5 et SHA-1** est fortement déconseillée dans un contexte juridique, car ils ont présenté des faiblesses et des collisions de hachage.

2. **Bloqueurs d'écriture validés :**

- a. Le document indique que **tout bloqueur d'écriture recommandé ou validé par le NIST convient** pour préserver l'intégrité des données d'origine lors de l'acquisition.
- b. Pour vérifier la conformité de vos outils de copie, le NIST publie des listes de tests d'outils d'investigation numérique sur son site dédié (www.cfft.nist.gov), qui constituent une ressource de référence.

3. **Protocole de test officiel :**

- a. Afin de s'assurer du bon fonctionnement d'un bloqueur d'écriture avant de le présenter devant un tribunal, le NIST propose un protocole de test officiel téléchargeable à l'adresse : www.cfft.nist.gov/HWB-ATP-19.pdf.

 Je peux vous détailler le fonctionnement des outils de capture de RAM (comme Dumplt et LiME) ou d'analyse de mémoire vive (comme Volatility) présentés dans votre document.

C'est quoi le format AFF ?

Le format **AFF (Advanced Forensics Format)** est une option courante pour enregistrer les données d'une copie numérique ainsi que leurs métadonnées.

Voici ses caractéristiques fondamentales selon votre document :

- **Norme ouverte** : Contrairement à des solutions propriétaires (comme le format EnCase), l'AFF est un format ouvert, ce qui garantit qu'un très grand nombre d'outils d'analyse et de duplication acceptent et prennent en charge ce type d'image.
- **Avantages techniques clés** : Certains professionnels le considèrent comme la méthode de copie privilégiée pour plusieurs raisons :
 - La possibilité de stocker des **métadonnées arbitraires**.
 - La flexibilité d'**ajuster la compression** pour économiser de l'espace de stockage sans altérer les données d'origine.
 - La capacité intégrée de **vérifier les erreurs** de copie.
- **Nomenclature des fichiers** : Lors de l'enregistrement de l'image forensique, les segments de fichiers générés prennent l'extension. AFF et se nomment chronologiquement **Evidence.AFF1**, **Evidence.AFF2**, etc.

 Souhaitez-vous que je dresse un tableau comparatif synthétique des trois grands formats d'acquisition du document (RAW, EnCase et AFF) pour vous aider à choisir le bon format selon le contexte de l'enquête ?

Comparer RAW, EnCase et AFF

Voici une comparaison détaillée des trois principaux formats d'acquisition de preuves numériques présentés dans vos sources : **RAW (Brut)**, **EnCase (.E)** et **AFF (Advanced Forensics Format)**.

Tableau comparatif des formats d'acquisition

Caractéristique	Format RAW	Format EnCase	Format AFF
Type de norme	Standard générique (brut)	Propriétaire (lié à l'application <i>EnCase</i>)	Norme ouverte (<i>open standard</i>)
Fichiers générés	Evidence.001, Evidence.dd1, etc.	Evidence.E01, Evidence.E02, etc.	Evidence.AFF1, Evidence.AFF2, etc.
Extension	.raw	.e	.AFF
Métadonnées	Aucune	Stockage limité de métadonnées	Stockage de métadonnées arbitraires

Gestion des erreurs	Ignore les erreurs lors de la copie	Gérée par la suite propriétaire	Capacité intégrée de vérification des erreurs
Compression	Aucune (taille brute du support)	Selon les options d'EnCase	Possibilité d' ajuster la compression pour économiser de l'espace
Atout principal	Copie généralement très rapide .	Tactique de déstabilisation juridique de l'adversaire.	Flexibilité maximale et compatibilité multi-outils.

Analyse détaillée de chaque format

1. Le Format RAW (Brut)

Le format RAW est **le plus couramment utilisé** pour la duplication forensique. Il réalise une copie physique intégrale (bit à bit) de l'original, englobant l'intégralité du support de stockage y compris les structures non visibles comme l'**espace libre** (*slack space*) et les **clusters de disque corrompus**. En plus d'être très rapide, son grand intérêt réside dans sa capacité à ignorer les erreurs physiques du disque pour mener la copie à terme.

2. Le Format EnCase

Ce format effectue également une copie numérique binaire complète, mais il est **propriétaire** et développé pour la suite judiciaire *EnCase*. Bien que ses capacités de stockage de métadonnées soient limitées, il présente un **intérêt tactique majeur devant un tribunal**. EnCase étant une référence acceptée par les cours de justice, forcer l'usage de ce format oblige la partie adverse à acheter cette suite logicielle très onéreuse pour analyser la preuve. Si l'avocat adverse dispose d'un budget limité ou manque d'expérience en informatique légale, cette barrière financière et technique peut l'amener à montrer des signes de faiblesse ou à contester maladroitement la procédure.

3. Le Format AFF (Advanced Forensics Format)

L'AFF est une **norme ouverte** acceptée par une immense variété d'outils d'investigation. Certains experts le considèrent comme la méthode de copie idéale car il élimine les contraintes des formats propriétaires tout en offrant des fonctionnalités absentes du format RAW. Il permet notamment d'incorporer des métadonnées personnalisées (arbitraires), de compresser l'image pour économiser de

l'espace disque (sans perte de données) et de vérifier scientifiquement la présence d'erreurs d'écriture directement dans le fichier image.

 Je pourrais vous présenter la différence entre l'outil classique *dd* et *dcfldd* (créé par le Département de la Défense des États-Unis), qui permet notamment de calculer le hachage de ces images à la volée pendant le transfert.

Expliquez dd et dcfldd

L'outil dd (Disk Duplicator)

dd est le programme de duplication de disque **le plus populaire et le plus simple** disponible sur les systèmes Linux.

- **Fonctionnement** : Il effectue une copie physique brute, bit à bit, d'une source vers une destination.
- **Syntaxe de base** : La commande s'exécute sous la forme suivante : `dd if=<source> of=<destination> bs=<taille en octets>`.
Par exemple, pour copier une clé USB /dev/sdb vers un dossier média, on saisit : `dd if=/dev/sdb of=/media/diskcopy.dd`.
- **Limitation visuelle** : Lors de copies volumineuses, le système Linux peut sembler totalement bloqué sans donner de retour visuel. Sur certains systèmes Linux, il est toutefois possible de suivre la progression du transfert en envoyant un signal SIGINFO à l'aide du raccourci clavier **Ctrl+T**.

L'outil dcfldd

Face aux limitations de l'outil classique dd lors d'enquêtes professionnelles, le **Département de la Défense des États-Unis** a créé la commande dcfldd pour offrir des options de contrôle indispensables aux experts en criminalistique.

Bien que dcfldd fonctionne de manière similaire à dd dans sa syntaxe de base, il intègre des **fonctionnalités avancées exclusives** :

1. **Le hachage à la volée** : C'est l'atout majeur de l'outil. Il calcule l'empreinte numérique des données d'entrée pendant leur transfert (prenant en charge simultanément md5, sha1, sha256, sha384 et sha512). Cela évite d'avoir à exécuter une validation de hachage

manuelle et séparée une fois la copie terminée, ce qui représente un **gain de temps considérable**.

2. **L'affichage en temps réel de l'état d'avancement** : Il affiche en continu la quantité de données transférées et une estimation du temps restant.
3. **La vérification de l'image et de l'effacement** : Il compare bit à bit le disque cible avec le fichier d'entrée (ou un modèle d'effacement) pour s'assurer que l'opération s'est déroulée de manière parfaite.
4. **L'effacement flexible des disques** : Il permet d'effacer rapidement et de manière sécurisée un support de stockage selon un modèle déterminé.
5. **Les sorties multiples** : Il permet d'écrire de manière simultanée vers plusieurs fichiers ou disques durs.
6. **Le fractionnement avancé de la sortie** : Il permet de découper l'image générée en plusieurs fichiers avec une configuration bien plus poussée que la commande Linux standard split.
7. **La redirection native des journaux** : Il envoie nativement toutes ses données de journalisation (logs) directement dans des fichiers ou des commandes pour préserver la traçabilité de l'acquisition.
8. **Syntaxe de base** : `dcfldd if=<source> hash=<type de hachage> of=<destination> bs=<taille en octets> <vérification des erreurs>`.
9. **Exemple de hachage multiple et d'écriture de fichiers journaux** : `dcfldd if=/dev/sdb hash=md5, sha512 md5log=md5.txt sha512log=sha512.txt of=finalfile.dd`.

🔧 Je peux également vous expliquer comment l'outil Netcat permet d'associer ces commandes de duplication pour transférer une image disque directement à travers un réseau vers un serveur distant.

Transférer des images via Netcat

L'utilisation de **Netcat** en combinaison avec la commande `dd` permet d'envoyer l'image d'un disque dur directement à travers le réseau vers un serveur distant. Cette méthode est particulièrement utile pour éviter de perdre du temps à copier l'image localement avant de devoir transférer le fichier lourd dans un second temps.

Voici comment configurer le transfert selon le rôle de chaque machine :

1. Configuration depuis le système source (Émetteur)

Pour lancer la copie d'un disque local et rediriger son flux de données directement à travers le réseau vers la machine de destination, vous devez utiliser un tube (|) pour lier dd et netcat.

- **Syntaxe générale** : `dd if=<source> bs=<taille> | netcat <adresse IP distante> <port>`
- **Exemple concret** : `dd if=/dev/sdb bs=16065b | netcat 192.168.10.25 1234`

Dans cet exemple, dd lit la clé USB cible branchée sur /dev/sdb avec une taille de bloc (bs) spécifiée, puis transmet le flux à Netcat, qui l'envoie à la machine distante ayant l'IP 192.168.10.25 sur le port 1234.

2. Configuration depuis le système distant (Récepteur)

Pour réceptionner l'image disque sur la machine distante qui va enregistrer les données, la commande est inversée : la source devient Netcat (qui écoute le réseau) et la destination est le disque ou le fichier de sortie géré par dd.

- **Syntaxe générale** : `netcat -l -p [port utilisé] | dd of=<source à copier> bs=<taille en octets>`
- **Exemple concret** : `netcat -l -p 1234 | dd of=/dev/sdb bs=16065b`

Détail des options utilisées :

- **-l** : Indique à Netcat d'écouter (*listen*) une connexion entrante plutôt que d'en initier une vers un hôte distant.
- **-p 1234** : Spécifie le port que Netcat doit écouter (assurez-vous que ce port est libre et n'est pas utilisé par un autre processus sur la machine).
- **dd of=/dev/sdb** : Reçoit le flux de données de Netcat et écrit la copie bit à bit vers la destination désignée (ici, le disque /dev/sdb).

Pour approfondir l'utilisation de cet outil réseau, vous pouvez consulter sa documentation sur <https://linux.die.net/man/1/nc>.

 Je peux vous expliquer comment compresser ou fractionner ces images volumineuses sans perte (avec des outils comme gzip ou split) afin d'optimiser l'espace de stockage avant de procéder à leur transfert.

Expliquez les outils DumpIt et LiME

DumpIt et **LiME** sont deux outils gratuits et extrêmement populaires conçus pour effectuer l'acquisition (ou le vidage/*dump*) de la **mémoire volatile (RAM)** d'un système informatique.

Dans une enquête de criminalistique numérique, la capture de la RAM est une étape indispensable, car elle permet de figer et de conserver des preuves qui s'effacent dès que l'appareil est mis hors tension. Ces données volatiles contiennent des indices précieux : les processus en cours d'exécution, l'identité des utilisateurs connectés, des mots de passe en clair manipulés par les applications, l'historique des commandes réseau, ou encore des logiciels malveillants « sans fichier » (qui résident exclusivement en mémoire).

Voici comment fonctionnent ces deux outils selon vos sources :

1. DumpIt (pour Windows)

DumpIt est une application d'acquisition de mémoire vive réputée pour sa grande simplicité d'utilisation sur les systèmes Windows.

- **Préparation** : L'outil est généralement téléchargé sous forme d'archive compressée (zip). Une fois décompressé, l'enquêteur obtient un simple fichier exécutable qu'il doit exécuter directement sur la machine cible. Il est d'ailleurs recommandé de bien s'assurer de l'authenticité de l'exécutable avant son lancement pour éviter toute contamination.
- **Fonctionnement** : À l'exécution, une interface en ligne de commande s'ouvre et pose une question simple : l'utilisateur souhaite-t-il sauvegarder la mémoire actuelle dans le dossier depuis lequel l'exécutable a été lancé ?
- **Résultat** : Si l'enquêteur confirme en saisissant « oui », DumpIt réalise instantanément une copie binaire intégrale et conforme de la RAM et l'enregistre localement sous la forme d'un fichier au format brut (**.raw**).

2. LiME (pour Linux)

LiME (pour *Linux Memory Extractor*) remplit un rôle identique à celui de DumpIt, mais s'adresse spécifiquement aux environnements Linux. Son

processus est plus technique et s'intègre directement au fonctionnement interne du système d'exploitation :

- **Téléchargement et compilation** : LiME est récupéré depuis un dépôt GitHub. L'enquêteur doit obligatoirement compiler l'outil directement pour le noyau du système cible. Pour ce faire, il extrait l'archive, se positionne dans le dossier /LiME-master/src et saisit la commande **make**. Cette compilation génère un module noyau Linux (un fichier d'extension .ko, par exemple lime-4.3.0-kali1-amd64.ko).
- **Exécution** : Pour démarrer la capture de la RAM, il faut insérer ce module directement dans le noyau à l'aide de la commande `sudo insmod` tout en spécifiant le chemin de sortie et le format d'enregistrement souhaité. La commande recommandée est la suivante : `sudo insmod lime-4.3.0-kali1-amd64.ko "path=/root/RAMDump.lime format=raw"` Cette instruction génère l'image mémoire (ici RAMDump.lime) directement au format brut (**.raw**) dans le dossier racine.
- **Résolution des erreurs fréquentes** :
 - *Problèmes de compatibilité* : Sur certaines versions de Linux comme Ubuntu (notamment XUbuntu 16.04), des erreurs peuvent survenir avec la version standard de LiME ou lors de l'installation par le gestionnaire de paquets via `apt-get install lime-forensics-dkms`.
 - *Erreur de module déjà chargé* : Si l'on tente d'exécuter la commande `insmod` une seconde fois, Linux renvoie une erreur indiquant que le fichier ou le module existe déjà. Pour débloquer la situation, il suffit de décharger le module actif en saisissant la commande **rmmod lime**. L'enquêteur peut vérifier la présence de ces modules en mémoire via la commande **lsmod** avant de les supprimer pour que LiME fonctionne à nouveau de façon optimale.

Une fois ces captures de RAM complétées et enregistrées, les fichiers bruts générés par DumpIt ou LiME peuvent être importés dans un outil d'analyse spécialisé comme **Volatility** afin d'en extraire méthodiquement les preuves.

🕵️ Je peux vous détailler le fonctionnement de Volatility pour analyser l'image RAM que vous venez de générer, et vous montrer comment extraire la liste des processus ou l'historique des lignes de commande.

Comment fonctionne Volatility ?

Volatility est un outil de criminalistique numérique extrêmement populaire, fourni avec Kali Linux (et également disponible sous Windows sous forme d'exécutable .exe), spécialement conçu pour **explorer et analyser les images de mémoire volatile (dumps de RAM)**.

Son fonctionnement repose sur un processus méthodique en ligne de commande, exécuté directement depuis un terminal :

1. L'identification du type de RAM (Le profil)

Puisqu'un enquêteur travaille souvent avec plusieurs fichiers de RAM provenant de différents systèmes, il est parfois impossible de connaître d'emblée l'origine exacte du fichier. Volatility a absolument besoin d'un profil pour interpréter correctement la mémoire.

- **Estimation du profil** : L'enquêteur utilise d'abord la commande `volatility -f <fichier de dump> imageinfo`. Volatility analyse le fichier et fournit une liste de "Profils suggérés" (il s'agit d'une estimation).
- **Validation de précision (kdbgscan)** : Pour identifier avec certitude le profil et l'adresse `kdbg` corrects, on utilise l'option `kdbgscan` : `volatility -f <fichier> --profile=<profil estimé> kdbgscan`. Cette commande effectue des vérifications de cohérence pour éliminer les faux positifs et fournit des détails précieux comme les décalages (*offsets*). Cette étape évite d'obtenir des résultats incorrects lors de l'exécution des commandes d'analyse suivantes.

2. L'extraction des preuves (Plugins et Commandes)

Une fois le bon profil identifié et configuré grâce à l'option `-profile=<type de profil>`, Volatility permet d'exécuter des commandes ciblées pour extraire méthodiquement des informations clés de la RAM :

- **Analyse des processus et détection de logiciels malveillants** :
 - `pslist` : affiche la liste de tous les processus en cours d'exécution au moment de la capture.

- pscan : permet, en combinaison avec pslist, d'identifier les processus dissimulés ou cachés par des logiciels malveillants.
- dlllist : dresse la liste des DLL chargées en mémoire pour chaque processus.
- **Activités réseau et lignes de commande :**
 - Connections et connscan : listent l'ensemble des connexions réseau ouvertes au moment de l'acquisition.
 - cmdscan : affiche toutes les commandes saisies en ligne de commande (ce qui est extrêmement précieux pour inspecter l'activité sur les terminaux de centres de données).
- **Sécurité et historique d'activité :**
 - hashdump : permet d'afficher les hachages des mots de passe système.
 - iehistory : récupère l'historique de navigation web (même si le nom de la commande fait référence à "Internet Explorer").

3. Gestion de la compatibilité des profils

- **Vérification des profils supportés** : Si l'on soupçonne que le fichier image n'est pas pris en charge par l'installation actuelle de l'outil, on peut saisir la commande volatility --info pour afficher la liste de tous les profils et formats supportés localement.
- **Création de profils personnalisés** : Si Volatility ne dispose d'aucun profil adapté (ce qui arrive fréquemment pour des systèmes Linux), il est possible de créer un profil personnalisé en compilant les structures appropriées conformément à la documentation de l'outil.

✱ Je peux vous préparer un aide-mémoire complet sous forme de document de référence pour votre kit d'intervention, récapitulant les commandes indispensables de capture (DumpIt/LiME) et d'analyse (Volatility).

Comment fonctionne le hachage à la volée avec dcfldd ?

Le hachage à la volée avec la commande de duplication **dcfldd** fonctionne en calculant l'empreinte numérique (le hachage) des données d'entrée **pendant leur transfert** vers le fichier de destination.

Voici les détails clés de son fonctionnement selon vos sources :

- **Un gain de temps important** : Le calcul s'effectuant simultanément durant la copie, cela évite d'avoir à exécuter une validation séparée des fichiers une fois la duplication terminée.
- **Calculs simultanés multiples** : Vous pouvez exécuter plusieurs vérifications de hachage en même temps. Les types de hachage utilisables avec dcfldd sont **md5**, **sha1**, **sha256**, **sha384** et **sha512**.
- **Journalisation automatique** : Les résultats des hachages calculés à la volée sont enregistrés directement dans des fichiers journaux (logs) spécifiques pour préserver l'intégrité et la traçabilité forensique.

Structure de la commande

Le format général de la commande dcfldd incluant le hachage est le suivant :

```
dcfldd if=[fichier source] hash=[type(s) de hachage]  
hashwindow=[fréquence de calcul du hachage] [hashtype]log=[nom du  
fichier journal de hachage] bs=[nombre d'octets à lire] of=[fichier de  
sortie].dd
```

Par exemple, pour copier une clé USB située sur /dev/sdb tout en calculant simultanément ses hachages MD5 et SHA-512 pour les stocker dans des fichiers de log, la commande s'écrit : dcfldd if=/dev/sdb hash=md5, sha512 md5log=md5.txt sha512log=sha512.txt of=finalfile.dd

 Je peux vous détailler d'autres fonctionnalités uniques de dcfldd, comme le fractionnement automatique de la sortie en plusieurs segments ou ses capacités de vérification d'erreurs.

Expliquez le fractionnement des images volumineuses.

Le fractionnement des images volumineuses est une méthode essentielle en criminalistique numérique pour gérer l'espace de stockage important consommé par les copies de disques durs.

Pour que cette opération soit valide devant un tribunal, elle doit impérativement être réalisée à l'aide d'un **fractionnement sans perte**, garantissant qu'aucune modification ou contamination n'altère la preuve d'origine.

Voici comment fonctionne ce processus sous Linux, de la division à la reconstruction, selon vos sources :

1. Le fractionnement avec la commande split

Sous Linux, la commande la plus courante et la plus simple pour diviser un fichier volumineux est **split**. Sa syntaxe générale s'écrit de la façon suivante : `split [option(s)] [fichier d'entrée] [préfixe de sortie]`

Elle propose deux méthodes principales de découpage :

- **Le découpage par taille** : Par exemple, la commande `split -b 22 example.txt new` divise le fichier d'entrée en segments de 22 bits (la taille étant définie par l'option `-b`). Les segments générés seront automatiquement nommés selon le préfixe choisi, suivi d'un compteur alphabétique (`newaa`, `newab`, `newac`, etc.).
- **Le découpage par lignes** : Par exemple, la commande `split -l 200 example.txt new` crée des segments de fichiers contenant chacun un maximum de 200 lignes de texte, nommés de la même manière (`newaa`, `newab`, etc.).

2. La reconstruction (La fusion)

Pour pouvoir analyser ou visualiser les données de l'image forensique, vous devez obligatoirement reconstituer le fichier d'origine :

1. **Le regroupement** : La première étape indispensable consiste à **déplacer tous les segments découpés dans un seul et même dossier** afin de pouvoir exécuter une commande unique de combinaison.
2. **La fusion avec cat** : On utilise ensuite la commande de concaténation standard selon la structure suivante : `cat [nom du fichier]* > [nom du fichier de sortie]` *Exemple concret* : Pour fusionner l'ensemble des segments portant le préfixe `new` (comme `newaa`, `newab`, etc.), on saisit : `cat new* > combinednew` Le

système va combiner tous ces segments pour recréer un fichier unique et complet nommé combinednew.

3. La validation absolue par hachage

En informatique légale, la compression ou le fractionnement comportent toujours un risque d'erreur lié au transfert, à l'application ou au système de copie. Il est donc obligatoire de valider scientifiquement l'intégrité de la copie :

- L'enquêteur doit **calculer la valeur de hachage du fichier avant l'opération**, puis **calculer à nouveau le hachage du fichier combiné après sa reconstruction**.
- Si les empreintes numériques (les valeurs de hachage) sont parfaitement identiques, la preuve est considérée comme une **copie conforme**. Si les hachages diffèrent, cela signifie que le fichier a subi une modification ou une corruption.
- **Recommandation de sécurité** : Il est fortement conseillé de tester ces fonctions de fractionnement et de compression sur une copie de test au préalable, plutôt que de manipuler directement l'image originale de la preuve.



Je peux vous expliquer en détail le fonctionnement des outils de compression sans perte présentés dans vos sources (tels que gzip, bzip et xz), ainsi que leur utilisation combinée avec la commande d'archivage tar.

Comment fonctionne l'outil Autopsy ?

Autopsy est une application open source populaire intégrée à Kali Linux, spécialement conçue pour structurer, documenter et gérer le processus de traçabilité d'une enquête de criminalistique numérique. Voici comment fonctionne cet outil étape par étape :

1. Lancement de l'outil

Pour démarrer Autopsy, vous devez vous rendre dans le menu de Kali Linux, aller dans **Applications > Analyse forensique**, puis sélectionner **Autopsy**. Ce clic ouvre automatiquement une fenêtre de terminal qui vous indique que l'application est en cours d'exécution en arrière-plan.

2. Accès à l'interface graphique (GUI)

Vous devez impérativement **garder la fenêtre de terminal ouverte**. Ouvrez ensuite votre navigateur web (comme Iceweasel) et rendez-vous à l'adresse locale indiquée par le terminal (généralement <http://localhost:9999/autopsy>). C'est ce navigateur qui vous servira d'interface visuelle pour manipuler l'outil.

3. Configuration et documentation de l'enquête

Une fois sur l'interface graphique d'Autopsy, vous avez la possibilité d'ouvrir un dossier existant ou de cliquer sur « **Nouvelle enquête** ». Lors de la création d'un nouveau cas, l'application vous guide à travers un formulaire rigoureux pour documenter l'enquête, en vous demandant de renseigner :

- Les **noms des personnes impliquées**.
- La **méthode de vérification** employée par les enquêteurs pendant leur travail.
- Les **images des artefacts** identifiés sur la scène.

4. Fonctionnalités d'analyse et de préservation

Au-delà de la simple documentation administrative, Autopsy intègre des fonctionnalités techniques indispensables pour préserver la valeur juridique de vos preuves :

- Elle gère et intègre les **copies d'images au niveau bit** (copies forensiques physiques conformes).
- Elle propose des méthodes scientifiques pour **valider l'intégrité des éléments stockés à l'aide de hachages** (MD5, SHA, etc.), garantissant ainsi l'absence de toute contamination.

(Note : Pour approfondir l'utilisation de cet outil open source, vous pouvez consulter leur documentation officielle sur

www.sleuthkit.org/autopsy.)

🚨 Je peux vous détailler le protocole rigoureux de première intervention sur une scène de crime pour garantir que l'environnement physique et numérique soit parfaitement sécurisé avant même de lancer vos outils d'analyse.

Détaillez le fonctionnement de Volatility

Volatility est un outil d'analyse forensique très populaire, intégré à Kali Linux (et également disponible sous Windows sous la forme d'un fichier

exécutable .exe fonctionnant de la même manière). Il est conçu pour explorer et analyser les captures de mémoire volatile (dumps de RAM). Voici les étapes clés et le fonctionnement détaillé de cet outil :

1. Lancement et identification du type de RAM (Le profil)

Puisqu'il arrive fréquemment qu'une équipe d'analystes travaille sur plusieurs fichiers de RAM issus de systèmes différents sans en connaître l'origine exacte, la première étape indispensable consiste à identifier le type de RAM.

- **Commande d'estimation** : Après avoir accédé au dossier contenant la capture, vous devez lancer le programme en terminal à l'aide de la commande : `volatility -f <fichier de dump mémoire> imageinfo`.
- **Résultat** : Volatility analyse le fichier et fournit une liste d'estimations appelée « **Profils suggérés** ».

2. Validation de précision avec kdbgscan

Les informations d'imageinfo n'étant que des estimations, il est recommandé de valider le profil avec précision pour éviter les erreurs lors de l'exécution des plug-ins (comme obtenir des résultats erronés avec pslist parce que Volatility utilise le premier kdbg trouvé au lieu du bon).

- **La commande kdbgscan** effectue des vérifications de cohérence pour réduire les faux positifs et identifier avec certitude le profil et l'adresse kdbg corrects.
- **Syntaxe** : `volatility -f <fichier de vidage mémoire> --profile=<profil estimé> kdbgscan`.
- **Données obtenues** : Cette analyse révèle des détails cruciaux tels que le type d'image précis, les décalages (*offsets*), etc.. Une fois le profil exact déterminé, vous devez l'appliquer à toutes vos commandes ultérieures avec l'option `-profile=<type de profil>`.

3. Extraction des preuves à l'aide de plug-ins et de commandes

Volatility propose de nombreuses commandes dédiées pour extraire des détails de la capture de RAM :

- **pslist** : Affiche la liste de tous les processus en cours d'exécution au moment de l'acquisition.
- **pscan** : Utilisée notamment en combinaison avec pslist pour identifier les processus cachés (très utile lors des enquêtes sur les logiciels malveillants).

- **hashdump** : Extrait et affiche les hachages de mots de passe système.
- **iehistory** : Affiche l'historique de navigation (bien que le nom de la commande fasse spécifiquement référence à Internet Explorer).
- **dlllist** : Permet de lister les DLL (Dynamic Link Libraries) chargées pour chaque processus.
- **cmdscan** : Affiche toutes les commandes exécutées en ligne de commande (ce qui s'avère extrêmement précieux pour l'analyse des terminaux de centres de données).
- **Connections et connscan** : Fournissent la liste complète des connexions réseau qui étaient ouvertes au moment de l'acquisition.

4. Gestion de la compatibilité et profils personnalisés

- **Vérification des profils supportés** : Si vous n'êtes pas certain que votre installation de Volatility prenne en charge un fichier image spécifique, vous pouvez lister les profils locaux avec la commande : `volatility --info`.
- **Profils personnalisés (Linux)** : Par exemple, si votre installation ne dispose d'aucun profil pour Linux, vous pouvez apprendre à créer un profil personnalisé en vous référant à la documentation officielle sur le dépôt GitHub de l'outil.

 Nous pourrions explorer d'autres outils de duplication forensique ou de montage d'images présentés dans votre manuel, comme Guymager ou WinHex.

Comment protéger un disque dur original lors de l'acquisition ?

Pour protéger un disque dur original lors de l'acquisition de données et garantir que la preuve reste admissible devant un tribunal, vous devez suivre un protocole strict fondé sur les principes suivants :

1. La règle d'or : Ne jamais travailler directement sur l'original

Il ne faut **jamais travailler directement sur le support de preuve original**, car cela risque de le contaminer et de le rendre inutilisable dans la plupart des situations juridiques. L'objectif est de réaliser des copies conformes au niveau binaire (images bit à bit) et de **conserver**

l'original en lieu sûr (dans un coffre-fort de qualité, après l'avoir scellé dans un sac antistatique sécurisé).

2. L'utilisation obligatoire d'un bloqueur d'écriture (*Write Blocker*)

Pour éviter toute modification accidentelle ou automatique du disque d'origine par le système d'analyse, il est indispensable d'utiliser un bloqueur d'écriture qui supprime toute possibilité d'écriture en appliquant un mode « lecture seule » :

- **Les bloqueurs matériels (physiques)** : Ce sont des boîtiers d'interface branchés entre le disque d'origine et votre station de travail forensique. Ils agissent soit en autorisant uniquement les commandes système qui ne modifient pas les données, soit en bloquant spécifiquement toutes les requêtes de commande d'écriture.
- **Recommandation juridique** : Devant un tribunal, il est fortement conseillé d'utiliser des matériels éprouvés et acceptés par la justice, tels que les dispositifs **FastBloc** (de Guidance Software), la gamme **Tableau**, ou tout bloqueur validé et recommandé par le **NIST**.
- **Les bloqueurs logiciels** : Ce sont généralement des options intégrées directement aux logiciels d'investigation numérique pour protéger vos copies de travail en cours d'examen.

3. Le cas critique des disques SSD et de la mémoire Flash

Il faut être extrêmement vigilant avec les disques SSD : les bloqueurs d'écriture physiques peuvent bloquer les commandes externes (comme TRIM), mais **ils ne peuvent pas empêcher les algorithmes internes d'équilibrage d'usure** (*wear-leveling*) propres au SSD de s'exécuter de façon autonome. Ces opérations internes peuvent modifier physiquement les données et donc changer la signature de hachage du disque, donnant l'impression que vous avez contaminé la preuve. Pour cette raison, lors d'une acquisition sur SSD, il est recommandé de **privilégier la validation des structures d'adresses logiques** plutôt que physiques afin d'éliminer les modifications internes dues au fonctionnement du disque.

4. Le protocole de préservation en 3 étapes

Une fois le disque connecté via un bloqueur d'écriture matériel validé, la procédure standard pour figer la preuve comprend :

1. **Créer au moins deux copies** au niveau binaire (bit à bit) du support d'origine (ce qui permet d'avoir une copie de rechange intacte en cas de défaillance matérielle).
2. **Valider immédiatement chaque copie par hachage** (idéalement avec un algorithme fort comme SHA-256 ou SHA-512) pour prouver scientifiquement l'intégrité des données copiées.
3. **Activer le blocage d'écriture sur vos propres copies forensiques** de travail afin d'éviter toute altération durant votre analyse.

🔧 Je peux vous détailler le protocole complet en 3 étapes du NIST pour tester l'efficacité de vos bloqueurs d'écriture avant de les engager sur une véritable scène de crime.

Comment fonctionne Guymager ?

Guymager est une application d'interface graphique disponible sous Kali Linux dans la section « **Analyse forensique** », conçue pour réaliser des copies conformes de disques durs de manière simple et intuitive.

Son fonctionnement suit un protocole visuel précis :

1. **Détection des disques** : Au lancement de Guymager, l'interface affiche automatiquement tous les disques durs connectés et disponibles. Si un support n'apparaît pas immédiatement, il peut être nécessaire de le monter manuellement au préalable pour qu'il soit détecté.
2. **Lancement de l'acquisition** : Pour démarrer le processus, vous devez effectuer un **clic droit sur le disque ciblé** dans la liste et sélectionner l'option « **Acquérir une image** ».
3. **Configuration de l'image** : Une fenêtre de paramétrage s'ouvre alors pour vous permettre de configurer l'acquisition :
 - a. **Format de fichier** : Vous pouvez choisir entre une image brute Linux dd (extension .dd ou .raw) ou le format Expert Witness de Guymager (extension .Exx).
 - b. **Documentation de l'enquête** : Pour respecter la chaîne de possession, vous devez remplir des champs d'identification indispensables tels que le numéro de dossier (*Case number*), le numéro de la preuve (*Evidence number*), le nom

de l'enquêteur (*Examiner*), une description et des notes de contexte.

- c. **Destination** : Vous indiquez le dossier d'enregistrement, le nom du fichier image et le nom du fichier d'information joint.
 - d. **Hachage et vérification** : L'interface vous permet de cocher les algorithmes de hachage à calculer pour valider l'intégrité de la copie (MD5, SHA-1 ou SHA-256) et d'activer l'option de relecture de la source après l'acquisition pour vérifier que la copie est rigoureusement identique.
4. **Exécution** : Une fois tous les paramètres saisis, il vous suffit de cliquer sur le bouton « **Démarrer** » (Start) pour lancer la copie physique et générer le fichier image à l'emplacement de stockage spécifié.

 Je peux vous présenter les outils gratuits équivalents présentés dans vos sources pour l'environnement Windows (comme FTK Imager, Arsenal Image Mounter ou OSFMount) si vous n'utilisez pas Kali Linux.

Comparer dd et dcfldd

La commande classique **dd** et la commande améliorée **dcfldd** sont deux outils de duplication Linux couramment utilisés pour réaliser des copies physiques bit à bit. Cependant, elles répondent à des besoins de niveaux différents lors d'une enquête numérique :

1. La commande dd (Disk Duplicator)

Il s'agit du programme de copie de disque **le plus populaire et le plus simple** disponible sous Linux.

- **Fonctionnement** : Il effectue une copie brute et linéaire d'une source vers une destination.
- **Syntaxe** : `dd if=<source> of=<destination> bs=<taille en octets>.`
- **Limitation visuelle** : Lors de copies volumineuses, le système Linux peut sembler complètement bloqué et ne fournit aucun retour direct sur l'avancement. Sur certains systèmes, il est toutefois possible d'afficher la progression en envoyant un signal SIGINFO à l'aide de la combinaison de touches **Ctrl+T**.

2. La commande dcfldd

Créée spécifiquement par le **département de la Défense des États-Unis**, cette commande a été développée pour combler les manques de dd en fournissant aux enquêteurs des fonctionnalités forensiques avancées.

Voici les options exclusives intégrées dans dcfldd qui ne sont pas disponibles dans dd :

- **Hachage à la volée** : Il calcule l'empreinte numérique des données pendant leur transfert, ce qui garantit leur intégrité et évite d'avoir à lancer une vérification de hachage manuelle et séparée une fois la copie terminée. Vous pouvez calculer simultanément plusieurs types de hachages parmi les options suivantes : md5, sha1, sha256, sha384 et sha512.
- **Affichage de l'état d'avancement** : Il indique en temps réel la quantité de données déjà transférées et estime le temps restant.
- **Vérification de l'image et de l'effacement** : Il vérifie la correspondance exacte, bit à bit, entre le disque cible et le fichier (ou le modèle) d'entrée spécifié.
- **Effacement de disque flexible** : Il permet de remettre à zéro ou d'effacer rapidement un support selon un modèle connu.
- **Sorties multiples** : Il permet d'écrire simultanément la copie vers plusieurs fichiers ou disques de destination.
- **Fractionnement de la sortie** : Il découpe automatiquement l'image générée en plusieurs morceaux avec des paramètres de configuration beaucoup plus poussés que la commande Linux standard split.
- **Journalisation et redirection des logs** : Il envoie nativement toutes ses données d'activité et ses journaux de hachage vers des commandes ou des fichiers textes d'audit.
- **Syntaxe** : dcfldd if=<source> hash=<type de hachage> of=<destination> bs=<taille en octets> <vérification des erreurs>.

 Je peux vous présenter l'outil de secours **ddrescue** détaillé dans vos sources, qui s'avère indispensable lorsque le disque dur cible est endommagé ou présente des clusters corrompus qui font échouer dd.

Quels risques présentent les SSD ?

Les disques SSD (et les clés USB) présentent des défis majeurs et des risques techniques importants en criminalistique numérique par rapport aux disques durs mécaniques traditionnels :

1. **La répartition non linéaire des données** : Contrairement aux disques durs mécaniques qui utilisent un mappage linéaire des secteurs, les SSD répartissent les données de manière aléatoire sur le support. Il est donc particulièrement difficile pour un enquêteur de prédire l'emplacement exact des données.
2. **Les algorithmes d'équilibrage d'usure (wear-leveling)** : Pour optimiser leur fonctionnement et améliorer leurs performances, les fabricants intègrent des logiciels qui répartissent les processus d'écriture sur tous les blocs de mémoire flash disponibles. Ces algorithmes internes compliquent considérablement la recherche de données et peuvent parfois même détruire les informations recherchées.
3. **La commande TRIM (Purge définitive)** : Sur un disque traditionnel, les secteurs supprimés sont simplement marqués comme espaces libres mais conservent les données jusqu'à leur remplacement. Sur un SSD, la commande TRIM effectue une purge automatique qui efface physiquement les données pour libérer immédiatement les secteurs de stockage. Une fois la commande TRIM exécutée, la récupération des données devient très improbable.
4. **La compression propriétaire** : Certains SSD utilisent des schémas de compression propres au fabricant qui peuvent altérer ou corrompre la lisibilité directe des données lors de l'extraction forensique.
5. **L'inefficacité partielle des bloqueurs d'écriture (Altération du hachage)** : Bien qu'un bloqueur d'écriture physique empêche efficacement les commandes externes (comme TRIM), **il ne peut pas bloquer les algorithmes internes d'équilibrage d'usure** du SSD. Étant donné que ces processus internes s'exécutent de manière autonome, ils peuvent modifier l'état physique du disque et générer une nouvelle valeur de hachage. Devant un tribunal,

cela peut donner la fausse impression que l'enquêteur a manipulé ou contaminé les preuves.

6. **Le dilemme de la mise hors tension** : En règle générale, un enquêteur forensique ne doit jamais éteindre ni débrancher un système en marche pour éviter toute modification. Cependant, si des données sont activement menacées de suppression sur un SSD, le maintien sous tension risque de déclencher TRIM et de détruire définitivement les preuves. L'enquêteur doit alors évaluer ce risque et peut être contraint d'enfreindre la règle standard en coupant l'alimentation de l'appareil.

Comment s'en prémunir juridiquement ?

Pour contrer le risque de modification physique autonome du SSD (et la variation de sa signature numérique), les sources recommandent d'abandonner l'analyse des adresses physiques lors de la validation des copies. Il est crucial de **privilégier les structures d'adresses logiques**. Cette approche cible uniquement le contenu des données copiées et élimine de l'équation les modifications physiques internes dues au fonctionnement automatique du SSD.

🔧 Je peux vous détailler le protocole de première intervention recommandé pour un système trouvé allumé afin d'éviter qu'un processus automatique ou une mauvaise manipulation ne détruise les preuves en mémoire.

Expliquer le kdbgscan

L'option **kdbgscan** de Volatility est une commande d'analyse approfondie essentielle pour identifier avec certitude le profil de système d'exploitation et l'adresse KDBG (*Kernel Debugger Block*) corrects au sein d'une capture de mémoire vive (RAM).

Voici comment elle fonctionne et pourquoi elle est indispensable :

1. Le problème de l'estimation initiale (imageinfo)

Lorsque vous commencez l'analyse d'un dump de RAM avec Volatility, vous utilisez généralement la commande imageinfo. Cependant, les profils suggérés par imageinfo ne sont que des **estimations**. Se fier uniquement à cette première analyse peut conduire à des erreurs lors de l'exécution de plug-ins d'investigation ultérieurs.

2. Le rôle correctif et préventif de kdbgscan

La commande kdbgscan vient valider ou corriger l'estimation d'imageinfo. Elle effectue des **vérifications de cohérence rigoureuses** afin de réduire considérablement les faux positifs.

Cette étape est particulièrement utile pour éviter des dysfonctionnements majeurs :

- **Éviter les résultats erronés** : Si vous lancez une commande comme pslist (pour lister les processus actifs) avec un mauvais profil, vous risquez d'obtenir des données totalement fausses ou incohérentes.
- **Résoudre les conflits de structures** : Une erreur fréquente survient lorsque Volatility utilise par défaut le tout premier bloc KDBG qu'il détecte dans l'échantillon de mémoire, alors qu'il en existe un autre, correct et complet, situé ailleurs. kdbgscan permet de localiser précisément le bon bloc.

3. Comment l'exécuter ?

La bonne pratique consiste à exécuter kdbgscan immédiatement après avoir obtenu les estimations d' imageinfo. La syntaxe recommandée est la suivante :

```
volatility -f <fichier_de_dump> --profile=<profil_estimé> kdbgscan
```

4. Les informations révélées

Une fois l'analyse terminée, kdbgscan affiche des détails techniques cruciaux pour configurer vos prochaines analyses, notamment :

- Le **type d'image** de manière définitive.
- Les **décalages** (*offsets*) exacts de la structure du noyau.

Une fois cette adresse KDBG validée, vous pouvez exécuter toutes vos autres commandes d'extraction (processus, réseau, clés de registre) en toute confiance en appliquant le profil exact validé.

 Si vous le souhaitez, nous pourrions explorer les commandes spécifiques de Volatility permettant d'analyser les connexions réseau actives ou d'extraire les hachages de mots de passe de cette image RAM.

Détecter les logiciels malveillants avec pscan

Dans le cadre de l'analyse d'une capture de mémoire vive avec **Volatility**, la commande **pscan** est un outil essentiel pour détecter les menaces dissimulées.

Voici comment elle fonctionne et s'intègre dans votre analyse selon vos sources :

- **Détection des processus cachés** : Lorsque vous enquêtez sur un logiciel malveillant, la commande classique **pslist** vous permet d'obtenir la liste des processus en cours d'exécution au moment de l'acquisition. Cependant, certains malwares tentent activement de se masquer pour échapper à cette liste. C'est là qu'intervient **pscan** : en combinant l'utilisation de **pslist** et de **pscan**, vous pouvez identifier avec précision ces processus cachés qui tentent de se soustraire à l'observation.
- **Indispensable pour les malwares résidant uniquement en mémoire** : Certains logiciels malveillants sophistiqués sont conçus pour ne jamais s'écrire sur le disque dur et s'exécuter exclusivement au sein de la mémoire vive (RAM). L'analyse de la RAM à l'aide de commandes comme **pscan** est donc le seul moyen d'identifier leur présence et de comprendre comment ils fonctionnent.

*Remarque : Bien que vos sources soulignent l'importance de la commande **pscan** pour révéler ces anomalies dissimulées, elles ne détaillent pas ses arguments spécifiques de ligne de commande ni ses mécanismes internes de balayage.*

 Je peux vous présenter les autres commandes de Volatility complémentaires pour pister l'activité d'un malware, comme l'analyse des connexions réseau ouvertes ou la recherche des DLL chargées en mémoire.

Première intervention : étapes ?

Lors d'une **première intervention** sur une scène de crime numérique, la première équipe d'intervention doit suivre un protocole extrêmement rigoureux. La moindre erreur ou omission à cette étape peut entraîner la contamination des preuves et conduire à leur rejet devant un tribunal.

Voici les étapes clés et obligatoires d'une première intervention :

1. Sécurisation physique et élimination des risques

- **Priorité absolue à la sécurité** : Le premier intervenant doit éliminer tout facteur de danger ou de risque physique dans la zone (par exemple, s'assurer qu'un suspect ne puisse pas faire irruption pendant la fouille).
- **Délimitation du périmètre** : Il faut immédiatement mettre en place un périmètre de sécurité (à l'aide de ruban de police ou d'un dispositif d'isolement) pour empêcher l'accès aux personnes non autorisées et éviter que le criminel potentiel n'altère la scène. Si nécessaire, le soutien des forces de l'ordre peut être sollicité pour maintenir ce périmètre.

2. Prise de décision stratégique

Le premier intervenant doit évaluer la situation pour choisir la meilleure orientation pour l'enquête :

- **Contacter les urgences** (112 ou 911) si des vies humaines ou des infrastructures vitales (comme des équipements hospitaliers) sont menacées.
- **Solliciter les autorités de police** (en mesurant le risque de perte de contrôle sur la confidentialité ou la saisie de matériels critiques pour l'entreprise).
- **Faire appel à des consultants externes spécialisés** (pour externaliser la responsabilité, bénéficier d'outils récents ou parer à un manque de ressources internes).
- **Mobiliser l'équipe interne de réponse aux incidents** ou **combiner ces options** sous la direction d'un chef de projet dédié.

3. Documentation systématique (Anti-contamination)

- **Figurer l'environnement** : Tout doit être documenté avant de commencer l'analyse ou de déplacer le matériel. Toute action non documentée sera perçue comme une contamination par la défense.
- **Prise de photos** : Prenez des photos de l'écran, du câblage, des appareils connectés et de l'emplacement des artefacts sous différents angles. Il est fortement recommandé d'utiliser un **appareil photo numérique dédié** plutôt qu'un téléphone portable, cette méthode étant mieux acceptée par les tribunaux.

- **Journal de bord** : Tenez un journal de bord écrit (stylo et papier) ou utilisez un enregistreur numérique en commentant chaque étape à voix haute.

4. Identification et initialisation de la chaîne de possession

- **Inventaire des supports** : Repérez tous les appareils dotés d'une capacité de stockage (ordinateurs, téléphones, imprimantes, serveurs, clés USB, objets connectés, etc.).
- **Chaîne de possession** : Remplissez un document de traçabilité standardisé (comme un formulaire généré par le logiciel Autopsy ou un modèle officiel) en attribuant un étiquetage unique, en identifiant toutes les personnes impliquées, ainsi que les dates et heures précises de chaque manipulation.

5. Vérification des autorisations légales

Avant d'exploiter ou de saisir un artefact, vous devez impérativement obtenir l'approbation légale nécessaire :

- **Accord du propriétaire** ou de l'organisation (pour les appareils professionnels dont l'usage est encadré par une charte informatique signée par l'employé).
- **Mandat judiciaire** si la loi l'exige.
- **Accords de confidentialité (NDA)** si vous manipulez des secrets industriels ou des configurations de sécurité réseau sensibles.

6. Conditionnement, transport et stockage sécurisé

- **Gestion des connexions réseau** : Si un appareil est déconnecté du réseau, documentez méticuleusement l'emplacement de chaque câble. Pour les appareils sans fil, utilisez une **cage de Faraday** afin d'empêcher toute modification ou suppression de données à distance pendant le transport.
- **Attention aux pièges de sécurité** : Évaluez les risques liés aux programmes de protection des données au repos, qui peuvent déclencher un chiffrement automatique du disque en cas de déconnexion réseau ou de perte de courant.
- **Préservation physique** : Scellez les preuves dans des **sacs antistatiques** étiquetés avec les numéros de preuve correspondants. Pour un ordinateur en marche qu'il est interdit d'éteindre, utilisez une alimentation de secours pendant le transport.

- **Stockage final** : À l'arrivée au laboratoire, déposez l'original dans un **coffre-fort de qualité** et documentez chaque accès de manière chronologique avant de procéder à la duplication (copie bit à bit).

🔦 Souhaitez-vous que je vous détaille les mesures techniques spécifiques à prendre immédiatement selon que vous découvrez un ordinateur allumé (recherche de données volatiles) ou éteint sur la scène de crime ?

Comment utiliser ddrescue ?

ddrescue est un outil de duplication de disque particulièrement recommandé lorsqu'un support d'origine présente de nombreuses erreurs physiques ou des clusters corrompus. Contrairement à la commande classique `dd` qui peut expirer ou s'interrompre en cas d'anomalie, `ddrescue` est capable d'ignorer les erreurs de lecture et de poursuivre la copie pour tenter de récupérer un maximum de données. Syntaxe générale de la commande

Pour exécuter cet outil sous Linux, la structure de base à respecter est la suivante : `ddrescue [option(s)] <fichier d'entrée> <fichier de sortie> [fichier journal]`

Exemple concret de duplication forensique

Dans le cadre d'une acquisition sur un disque endommagé, la commande recommandée se présente ainsi : `ddrescue -r3 dev/sdb diskcopy.dd ddrescue.log`

Voici l'explication de chaque paramètre utilisé dans cet exemple :

- **-r3** : Cette option indique à l'outil de tenter de relire et de corriger les secteurs défectueux **trois fois** avant de les abandonner.
- **dev/sdb** : Représente le support d'entrée à copier (ici, la clé ou le disque USB d'origine). Notez qu'il est également possible de désigner des portions ou partitions plus petites du lecteur au lieu de copier l'intégralité du support d'un seul coup.
- **diskcopy.dd** : Le fichier d'image de sortie dans lequel les données copiées seront écrites.

- **ddrescue.log** : Il s'agit du **fichier journal**, un élément absolument crucial. Il sert à suivre précisément la progression de la récupération pour chaque secteur du disque. Si le processus de copie vient à être interrompu, ce fichier permet de **reprendre la duplication exactement là où elle s'est arrêtée**, ou de tenter de corriger spécifiquement les secteurs défectueux précédemment ignorés. Sans ce fichier journal, vous seriez contraint de recommencer toute l'opération depuis le début.

Bonnes pratiques et mises en garde importantes

- **Stratégie par étapes** : Lors de la récupération d'un disque dur très endommagé, il est conseillé de récupérer d'abord les parties saines et spécifiques du disque, puis de tenter de récupérer le reste dans un second temps, en vous appuyant sur le fichier journal pour suivre l'avancement.
- **Risques de corruption** : Les développeurs de ddrescue déconseillent fortement de tenter la récupération de partitions qui sont montées en mode lecture/écriture, ou de disques présentant des erreurs d'Entrée/Sortie (E/S). Bien que l'outil puisse fonctionner dans ces situations, cela comporte des risques réels d'échec ou de **perte de données supplémentaires** sur le support d'origine.

(Pour explorer l'intégralité des options avancées de cet outil, vous pouvez consulter son manuel officiel à l'adresse :

www.gnu.org/software/ddrescue/manual/ddrescue_manual.html).

 Je peux vous expliquer comment compresser sans perte ou archiver ces fichiers d'images volumineux (avec des outils comme gzip, xz ou la commande tar) pour optimiser l'utilisation de votre espace de stockage forensique.

Comment récupérer les données supprimées par TRIM ?

La récupération de données après l'exécution de la commande TRIM sur un SSD ou une mémoire flash est **très improbable**. Contrairement aux disques durs mécaniques traditionnels qui marquent simplement les secteurs supprimés comme libres tout en conservant les données jusqu'à leur remplacement, TRIM effectue une purge qui supprime

physiquement les données pour libérer immédiatement les secteurs de stockage.

Face à cette contrainte technique, le protocole recommande plusieurs mesures de préservation :

- **La prévention plutôt que la récupération** : Il est grandement préférable d'empêcher la suppression des données d'un disque dur plutôt que de tenter de les récupérer après une telle opération.
- **L'extinction d'urgence du système** : Bien que la règle habituelle pour un enquêteur forensique soit de ne pas éteindre ni débrancher un système en marche pour éviter la contamination, vous pouvez enfreindre cette règle avec les disques SSD si vous estimez que les données sont activement menacées de suppression, afin d'interrompre le processus avant que TRIM ne soit exécuté.
- **L'utilisation de bloqueurs d'écriture** : Lors de l'analyse, l'utilisation de bloqueurs d'écriture peut empêcher l'exécution de la commande TRIM sur le SSD.
- **L'analyse basée sur les adresses logiques** : Si le hachage physique du disque a déjà été modifié par le fonctionnement interne du SSD (comme TRIM ou l'équilibrage d'usure), il est crucial de concentrer votre copie et votre défense juridique sur les **structures d'adresses logiques** plutôt que physiques. Cette approche cible uniquement les données copiées et élimine les modifications autonomes du support.

 Je pourrais vous rédiger une fiche de procédure récapitulant les étapes spécifiques à suivre pour la manipulation et la saisie des disques SSD lors d'une première intervention.

Outils pour objets connectés (IoT)

Dans votre document, les objets connectés (IoT) sont présentés comme des cas particuliers où le stockage des données est souvent inaccessible de manière classique (impossible de s'y connecter directement avec un ordinateur ou d'accéder au support de stockage), car **la mémoire est directement intégrée à l'appareil**.

En ce qui concerne les **outils et procédures d'acquisition**, le document indique qu'il existe des milliers d'appareils différents avec chacun sa propre méthode, et qu'aucun outil logiciel standard n'est décrit dans ce chapitre. À la place, l'auteur recommande l'approche suivante :

1. **Le démontage physique (Ingénierie matériel)** : Si cela est autorisé, la technique consiste à **ouvrir physiquement l'appareil**, quitte à l'endommager, en le démontant afin d'accéder directement à la mémoire et aux composants internes, une méthode courante chez les ingénieurs en matériel informatique.
2. **La recherche de documentation externe (Google et YouTube)** : L'auteur conseille d'utiliser **Google** et de rechercher des tutoriels ou des vidéos (par exemple sur **YouTube**) pour vérifier si un autre spécialiste a déjà documenté la façon d'accéder à ce modèle spécifique d'appareil.
3. **L'autorisation et l'évaluation des risques** : Il est indispensable d'**obtenir l'autorisation du propriétaire du matériel** et de bien évaluer les risques de destruction de l'appareil ou des données avant de tenter une ouverture physique.

Le manuel précise également que l'étude détaillée des objets connectés (IoT) sera abordée dans le chapitre suivant de l'ouvrage.

 Comme ce chapitre n'entre pas dans le détail des outils techniques pour l'IoT, je peux effectuer une recherche sur le web pour identifier les protocoles matériels d'extraction (comme JTAG ou UART) ou les logiciels d'analyse forensique d'objets connectés les plus réputés actuellement.

Détails sur WinHex ?

Selon vos sources, **WinHex** est un outil fonctionnant sous Windows présenté comme une solution utile pour la criminalistique numérique. Il propose trois grandes fonctionnalités pour l'enquêteur :

- **Analyse binaire et détection de données cachées** : WinHex permet d'analyser visuellement des copies de disques durs. Cette interface permet notamment de repérer l'**espace libre** (*slack space*), qui apparaît sous la forme de bits « 00 » entre les données. Bien que cet espace soit normalement ignoré par le système d'exploitation, il peut être utilisé pour dissimuler des données ou héberger des programmes malveillants comme des rootkits.
- **Validation de hachage** : WinHex intègre des fonctions de validation de hachage permettant de calculer et de comparer les empreintes numériques des fichiers afin de garantir scientifiquement l'intégrité des copies forensiques.
- **Blocage d'écriture logiciel** : Le programme dispose d'une option permettant d'activer un bloqueur d'écriture logiciel. Cette fonction applique un mode « lecture seule » pour protéger vos copies de travail de toute modification ou contamination accidentelle pendant que vous les examinez.

 Souhaitez-vous que nous examinions le fonctionnement d'autres outils d'acquisition et de montage d'images spécifiques à l'environnement Windows également cités dans votre document, comme FTK Imager ou Arsenal Image Mounter ?

Expliquer le format Expert Witness (.Exx).

Le format **Expert Witness** (connu sous l'extension de fichier **.Exx**, telle que .E01, .E02, etc.) est le format d'image forensique associé à l'application **EnCase**.

Voici ses caractéristiques et son fonctionnement détaillés :

- **Nature du format** : C'est un format **propriétaire** conçu spécifiquement pour la suite d'analyse judiciaire *EnCase*. Malgré cette nature propriétaire, il est devenu un standard de l'industrie pris en charge par de nombreux autres outils d'acquisition, comme *Guymager* sous Kali Linux (qui propose l'option *Expert Witness Format*, *sub-format Guymager* avec l'extension .Exx).

- **Copie physique complète** : À l'image du format brut (RAW), l'acquisition au format Expert Witness réalise une copie numérique binaire complète et conforme (bit à bit) de l'intégralité du support de stockage d'origine.
- **Structure des fichiers** : Lors de la copie, les fichiers d'images volumineuses sont automatiquement segmentés et enregistrés sous la forme de fichiers successifs nommés Evidence.E01, Evidence.E02, etc. (ou .Exx selon les outils).
- **Intégration de métadonnées** : Contrairement au format RAW (qui ne contient que les données brutes du disque), le format Expert Witness offre un **stockage intégré (bien que limité) de métadonnées**. Lors de la configuration de l'acquisition (comme sur l'interface de Guymager), l'enquêteur peut y inscrire des informations de traçabilité essentielles : le numéro de dossier (*Case number*), le numéro de la preuve (*Evidence number*), le nom de l'enquêteur (*Examiner*), une description et des notes de contexte.
- **Évolution** : Le format existe en deux versions principales, la version 2 ayant été introduite avec la sortie de l'application *EnCase 7*.
- **Intérêt stratégique devant les tribunaux** : Ce format étant universellement accepté par les instances juridiques, l'utiliser peut s'avérer être une tactique pour évaluer la solidité de la partie adverse. En effet, l'avocat adverse se retrouve contraint d'acquérir la suite logicielle EnCase, qui s'avère particulièrement onéreuse. S'il manque d'expérience en informatique légale ou si ses ressources financières sont limitées, cette contrainte peut le déstabiliser ou révéler ses faiblesses durant la procédure.

⚖️ Je peux vous détailler les différences de flexibilité et de compression entre ce format propriétaire et le format ouvert **AFF (Advanced Forensics Format)**.

Comment reconstruire un RAID ?

La reconstitution manuelle d'un système RAID (*Redundant Array of Independent Disks*) dans un cadre d'analyse forensique est une opération technique délicate qui s'articule autour des étapes suivantes :

1. **Identifier la taille des blocs** : Il s'agit de la première mesure indispensable pour comprendre la structure de répartition des données.
2. **Déterminer la période physique de la grappe** : Cette analyse s'effectue en traçant et en suivant les blocs de parité répartis sur le disque.
3. **Construire la carte de répartition** : L'enquêteur doit inspecter les blocs physiques successifs dans l'image et analyser les différents blocs physiques occupant la même position.

Au cours de ce processus, le réarrangement des disques produit généralement un motif (ou schéma) récurrent. Ce schéma sert de guide pour reconstituer précisément la carte de répartition, rendant possible la récupération des données du système.

Défis techniques et outils associés

- **Difficultés courantes** : L'enquêteur peut être confronté à des entêtes ou des disques endommagés (générant des erreurs de lecture/écriture), à des contrôleurs RAID propriétaires nécessitant des produits spécifiques pour être lus, ou encore à une absence totale d'informations concernant la configuration du contrôleur d'origine ou du BIOS (lorsque l'on ne dispose que du disque dur nu).
- **Logiciels d'assistance** : Pour vous aider dans cette tâche, plusieurs outils spécialisés sont disponibles, notamment **EnCase**, **OSForensics** et **Mount Image Pro**.
- **Recommandation importante** : Étant donné la complexité de ces manipulations, ce processus requiert très probablement l'intervention de professionnels spécialisés en récupération RAID pour être réalisé correctement.

 Je peux vous présenter un comparatif détaillé des différents niveaux de RAID (de RAID 0 à RAID 15) mentionnés dans vos sources afin d'identifier leurs spécificités en matière de parité, de performance et de tolérance aux pannes.

Expliquer JTAG et UART

Vos sources ne mentionnent pas et n'expliquent pas les termes **JTAG** et **UART**.

Pour ce qui est de l'accès aux composants physiques, vos sources indiquent simplement que pour certains appareils mobiles ou objets connectés (IoT), la mémoire est directement intégrée et inaccessible par les voies logicielles classiques. Dans ces situations, le protocole décrit consiste à :

- **Ouvrir et démonter physiquement l'appareil** (quitte à l'endommager) pour obtenir un accès physique direct à la mémoire et aux puces internes.
- **Rechercher des guides externes** sur Google ou YouTube pour vérifier si une méthode d'accès matériel a déjà été documentée pour ce modèle spécifique d'appareil.

Le document précise que l'analyse de ces équipements IoT est abordée plus en détail dans le chapitre suivant.

Souhaitez-vous que j'effectue des **recherches sur le Web** pour vous expliquer en détail ce que sont les interfaces de test et de débogage **JTAG** et **UART**, et comment elles sont utilisées en pratique pour l'extraction de données physiques (*hardware forensics*) ?

Expliquez le fractionnement des images volumineuses.

Le fractionnement des images volumineuses de disques durs est une méthode indispensable pour gérer l'espace de stockage important consommé par les sauvegardes forensiques. Cependant, pour que cette opération soit valide devant un tribunal, elle doit impérativement être réalisée de manière **sans perte** afin d'exclure tout risque d'altération ou de contamination des preuves.

Voici comment fonctionne ce processus sous Linux, de la division à la reconstruction, selon vos sources :

1. Le fractionnement avec la commande split

Sous Linux, la commande la plus courante et la plus simple pour diviser un fichier volumineux est **split**. Sa syntaxe générale s'écrit de la façon suivante : `split [option(s)] [fichier d'entrée] [préfixe de sortie]`.

Elle propose deux méthodes principales de découpage :

- **Le découpage par taille** (option -b) : Par exemple, la commande `split -b 22 exemple.txt new` divise le fichier d'entrée en segments de 22 bits. Les segments générés seront automatiquement nommés selon le préfixe choisi, suivi d'un compteur alphabétique (newaa, newab, newac, etc.).
- **Le découpage par lignes** (option -l) : Par exemple, la commande `split -l 200 exemple.txt new` crée des segments de fichiers contenant chacun un maximum de 200 lignes de texte, nommés de la même manière (newaa, newab, etc.).

2. La reconstruction (La fusion)

Pour pouvoir analyser ou visualiser les données de l'image forensique, vous devez obligatoirement reconstituer le fichier d'origine :

1. **Le regroupement** : La première étape indispensable consiste à **déplacer tous les segments découpés dans un seul et même dossier** afin de pouvoir exécuter une commande unique de combinaison.
2. **La fusion avec cat** : On utilise ensuite la commande de concaténation standard selon la structure suivante : `cat [nom du fichier]* > [nom du fichier de sortie]`. *Exemple concret* : Pour fusionner l'ensemble des segments portant le préfixe new (comme newaa, newab, etc.), on saisit : `cat new* > combinednew`. Le système va combiner tous ces segments pour recréer un fichier unique et complet nommé combinednew.

3. La validation absolue par hachage

En informatique légale, la compression ou le fractionnement comportent toujours un risque d'erreur ou de modification lié au transfert, à l'application ou au système de copie. Il est donc obligatoire de valider scientifiquement l'intégrité de la copie :

- L'enquêteur doit **calculer la valeur de hachage du fichier avant l'opération**, puis **calculer à nouveau le hachage du fichier combiné après sa reconstruction**.

- Si les empreintes numériques (les valeurs de hachage) sont parfaitement identiques, la preuve est considérée comme une **copie conforme**. Si les hachages diffèrent, cela signifie que le fichier a subi une modification ou une corruption.
- **Recommandation de sécurité** : Il est fortement conseillé de tester ces fonctions de fractionnement et de compression sur une copie de test au préalable, plutôt que de manipuler directement l'image originale de la preuve.

 Si vous le souhaitez, nous pouvons examiner comment combiner ces techniques de fractionnement avec les outils de compression sans perte (gzip, bzip, xz) et d'archivage (tar) présentés dans vos sources.

Qu'est-ce que l'espace libre (slack space) ?

L'**espace libre** (communément appelé **slack space** ou désigné par le terme **données intercalées** dans vos sources) correspond à l'**espace physique qui se trouve entre les fichiers** sur un support de stockage.

Caractéristiques de l'espace libre

- **Invisible pour le système** : Cet espace vide est généralement **ignoré par les ordinateurs** et les systèmes d'exploitation. Lorsque l'on analyse un disque avec un outil comme *WinHex*, l'espace libre se manifeste visuellement sous la forme de bits « 00 » situés entre les blocs de données actives.
- **Exclu des sauvegardes traditionnelles** : Les sauvegardes de données classiques se concentrent uniquement sur les données considérées comme pertinentes par le système, laissant de côté l'espace libre ainsi que les clusters identifiés comme corrompus.

Importance en criminalistique numérique

L'analyse de cet espace est une étape critique lors d'une investigation forensique pour deux raisons majeures :

1. **La dissimulation de données** : Des personnes malveillantes peuvent utiliser le *slack space* pour y injecter et y dissimuler des données confidentielles ou des preuves.
2. **La présence de codes malveillants** : Les **rootkits** sont particulièrement connus pour se loger et se cacher dans cet

espace afin de contourner les outils de détection du système d'exploitation.

Pour cette raison, un enquêteur ne doit jamais se limiter à une simple copie des fichiers visibles du système. Il est indispensable de réaliser une **copie physique conforme, bit à bit (comme au format RAW)**, qui permet de capturer l'intégralité du disque dur et d'aspirer tout ce qui se cache dans l'espace libre pour l'analyser ultérieurement.



Je peux vous expliquer comment d'autres formats d'image forensique (comme le format AFF ou le format Expert Witness d'EnCase) gèrent le stockage des métadonnées et la compression de ces espaces vides par rapport au format brut RAW.