

Forensic Analysis of Windows Endpoints and File Systems

Ce texte constitue une introduction technique à l'**analyse forensique des terminaux**, en mettant l'accent sur l'explosion des **objets connectés (IoT)** et des défis qu'ils imposent aux enquêteurs numériques. L'auteur détaille les structures fondamentales des **systèmes de fichiers Windows**, en comparant les limites du format **FAT** à la modernité du format **NTFS**. Les sources présentent divers **outils logiciels**, tels que **Exiftool** ou **TrIDNET**, permettant d'extraire des métadonnées ou d'identifier des fichiers suspects grâce à leurs **nombres magiques**. Une portion importante du contenu est consacrée à la localisation des **artefacts numériques** au sein du **registre Windows** et des dossiers système cachés. Enfin, le document souligne l'importance cruciale de travailler sur des **copies certifiées** afin de préserver l'intégrité des preuves originales.

Comment les objets connectés (IoT) influencent-ils les enquêtes numériques ?

Les objets connectés (IoT) influencent les enquêtes numériques de manière significative, en présentant à la fois des **opportunités inédites de collecte de preuves** et de **nouveaux défis complexes** pour les enquêteurs.

1. Une profusion de preuves comportementales

- **Enregistrement de la vie quotidienne** : L'un des principaux avantages de l'IoT est la quantité accrue de preuves exploitables, car ces équipements enregistrent continuellement des aspects du quotidien de leurs utilisateurs.
- **Reconstitution des faits** :
 - Les **serrures connectées** (Bluetooth ou web) permettent d'identifier qui a déverrouillé une porte ou accédé à une pièce.

- Les **sonnettes vidéo** enregistrent les passages et la présence de personnes à proximité des entrées.
- Les **thermostats intelligents** peuvent signaler l'absence d'un occupant (par exemple en passant en mode économique dès que le téléphone portable associé s'éloigne).
- La croisée de ces données permet aux enquêteurs de **prouver les mouvements, entrées et sorties** de suspects ou de victimes.

2. Une complexité accrue et des failles de sécurité

- **Volume d'équipements à traiter** : La multiplication des objets connectés rend la collecte physique et logique plus lourde, augmentant le nombre de terminaux à analyser lors d'un incident.
- **Vecteurs d'attaque privilégiés** : De nombreux fabricants d'objets connectés n'ont pas pour cœur de métier la sécurité informatique et présentent des lacunes en développement sécurisé. Ces appareils deviennent ainsi des **points d'entrée probables** pour les attaquants cherchant à s'infiltrer dans un réseau d'entreprise.

3. Spécificités et méthodologie d'investigation

L'analyse forensique de l'IoT exige des approches particulières adaptées à leur architecture :

- **Systèmes embarqués Linux** : La plupart des appareils IoT fonctionnent sous une déclinaison de Linux. Lorsqu'un accès shell (VPN, direct ou réseau) est possible, l'analyse s'effectue avec des commandes Linux standard (last, w, ls, ps, lsof, find) pour inspecter les processus, fichiers et connexions.
- **Analyse des fréquences radio (FCC)** : Les identifiants FCC apposés sur les appareils permettent de consulter les fréquences et canaux de communication sans fil enregistrés (ex. via *fccid.io*), afin d'orienter la capture du trafic radio.
- **Les 4 sources de collecte de preuves** :
 - **Le matériel IoT** (via inspection externe/interne, cartes SD ou connexions USB).
 - **L'application mobile dédiée** servant de télécommande ou d'interface de gestion.
 - **Le trafic réseau** généré par l'appareil.

- **Les services cloud** associés à l'équipement ou à l'application mobile.

Quels types de preuves trouve-t-on sur macOS ?

Sur un système **macOS** (OS X), l'analyse forensique permet de collecter une grande variété d'artefacts numériques enregistrés dans des répertoires et fichiers spécifiques :

1. Informations sur le système et l'installation

- **Version du système d'exploitation** : Stockée dans /System/Library/CoreServices/SystemVersion.plist.
- **Date d'installation** : Identifiable via la date de dernière modification de /var/db/.AppleSetupDone.
- **Historique des applications et mises à jour** : Répertoire dans /Library/Receipts/InstallHistory.plist.
- **Dernier utilisateur connecté** : Consigné dans /Library/Preferences/com.apple.loginwindow.plist.
- **Sauvegardes** : Les configurations Time Machine se trouvent dans /Bibliothèque/Préférences/com.apple.TimeMachine.plist, et les sauvegardes d'appareils iOS connectés sont situées dans ~/Bibliothèque/Application Support/MobileSync/Backup/*.

2. Connectivité et partage réseau

- **Historique Bluetooth** : Situé dans /Bibliothèque/Préférences/com.apple.Bluetooth.plist.
- **Réseaux Wi-Fi** : La liste de tous les réseaux sans fil auxquels le système s'est connecté est enregistrée dans /Library/Preferences/SystemConfiguration/com.apple.airport.preferences.plist.
- **Partage de fichiers (SMB)** : Disponible dans /Bibliothèque/Préférences/SystemConfiguration/com.apple.smb.server.plist.

3. Activité de l'utilisateur et fichiers récents

- **Préférences utilisateur** : Stockées sous forme de fichiers .plist dans ~/Bibliothèque/Préférences/*.
- **Éléments récents** : Les applications, documents et serveurs récemment ouverts sont enregistrés dans com.apple.recentitems.plist et dans les fichiers *LSSharedFileList.plist.
- **Messagerie Apple Mail** : Les boîtes aux lettres, la configuration et le contenu des messages se trouvent dans ~/Bibliothèque/Mail/ et com.apple.mail.plist.

4. Navigation Web

- **Safari** : L'historique (History.db), la dernière session (LastSession.plist), les cookies (Cookies.plist), les sites mis en cache (Cache.db) et les sites les plus visités (TopSites.plist) sont stockés dans le dossier ~/Bibliothèque/Safari/ et ~/Library/Caches/.
- **Chrome et Firefox** : L'historique Chrome réside dans ~/Bibliothèque/Application Support/Google/Chrome/Default/History, tandis que Firefox enregistre ses visites et cookies dans des bases SQLite (Places.sqlite, Cookies.sqlite) sous ~/Library/Application Support/Firefox/.

5. Mémoire volatile, fichiers d'échange et journaux

- **Fichiers mémoire et swap** : Le contenu de la RAM sauvegardé lors de la mise en veille est conservé dans /var/vm/sleepimage. Les fichiers d'échange du système se trouvent dans /var/vm/swapfile*.
- **Fichiers journaux (Logs)** : macOS conserve les journaux système Apple (/var/log/asl/*), les journaux d'audit (/var/audit/*) et les journaux d'installation (/var/log/install.log), qui peuvent être examinés via l'application Console.

(Note : La majorité de ces artefacts sont structurés au format .plist (Property List), qui peuvent être sous forme de texte XML brut ou au format binaire).

Établir le tableau comparatif avec Windows

Voici un **tableau comparatif** mettant en vis-à-vis la structure et la localisation des principaux artefacts numériques identifiables sous **macOS** et **Windows** :

Tableau comparatif des artefacts : macOS vs Windows

Catégorie d'artefact	macOS (OS X)	Windows
Systèmes de fichiers principaux	HFS+ et APFS (optimisé pour les disques SSD, avec chiffrement natif, instantanés et horodatage précis).	NTFS (journalisé, stable et sécurisé) ainsi que FAT / FAT32 / exFAT (compatibilité globale mais sécurité et capacités limitées).
Stockage principal de configuration	Fichiers de propriétés .plist (Property List), enregistrés au format XML texte brut ou binaire.	Base de Registre Windows (structure hiérarchique stockée dans %SYSTEMROOT%\system32\config contenant les ruches comme HKLM et HKU).
Informations système & installation	Version système (SystemVersion.plist), date d'installation (.AppleSetupDone), historique (InstallHistory.plist).	Fichiers de configuration sous %SYSTEMROOT%\System32, Registre système et points de restauration (Rp.log, Change.log.x).
Périphériques & Réseaux Wi-Fi	Préférences Wi-Fi (com.apple.airport.preferences.plist), Bluetooth (com.apple.Bluetooth.plist), partages SMB (com.apple.smb.server.plist).	Périphériques USB et SSID des réseaux Wi-Fi enregistrés dans le Registre (USBSTOR, MountedDevices, MountPoints2, USB).
Éléments récents & raccourcis	Éléments récents dans com.apple.recentitems.plist et fichiers *LSSharedFileList.plist.	Raccourcis (.lnk), dossier Récents (...AppData\Roaming\...\Recent), listes MRU et fichiers épinglés/Jump Lists.
Navigation Web	Safari (History.db, LastSession.plist, Cookies.plist, Cache.db), Chrome (History), Firefox (Places.sqlite, Cookies.sqlite).	Historique, favoris, cookies et métadonnées enregistrés sous les dossiers AppData\Roaming\Microsoft\Windows\ (Cookies, Favorites).

Journaux système (Logs)	Journaux système Apple (/var/log/asl/*), d'audit (/var/audit/*) et d'installation (/var/log/install.log), visibles avec l'application Console.	Fichiers au format EVTX dans %SystemRoot%\System32\Winevt\Logs (System.evtx, Security.evtx, Application.evtx).
Fichiers mémoire et d'échange	RAM sauvegardée à la mise en veille (/var/vm/sleepimage) et fichiers d'échange (/var/vm/swapfile*).	Fichiers d'échange mémoire (pagefile / fichiers mémoire cachés) servant d'extension à la mémoire vive.
Dissimulation & Fonctionnalités avancées	Support natif des instantanés (<i>snapshots</i>) et des fichiers éparés sur le système APFS.	Flux de données alternatifs (ADS) sous NTFS (fichier.txt:ads.txt) et espace inutilisé (<i>slack space</i>).

Principales différences d'approche forensique

1. **Registre centralisé vs Fichiers autonomes** : Windows regroupe la quasi-totalité de l'activité utilisateur et de la configuration matérielle au sein des ruches du **Registre**. À l'inverse, macOS privilégie une structure décentralisée basée sur une multitude de fichiers de préférences **.plist** distribués dans l'arborescence utilisateur et système.
2. **Gestion du système de fichiers** : Le système NTFS de Windows permet de masquer du contenu malveillant via des **flux de données alternatifs (ADS)** attachés à des fichiers ordinaires. De son côté, l'environnement macOS s'appuie sur **APFS**, qui introduit des fonctionnalités de sécurité renforcées telles que le chiffrement natif et les instantanés du système.
3. **Formats de journalisation** : Windows exploite un format propriétaire **EVTX** analysable avec des outils dédiés comme *Log Parser* ou *Event Log Viewer*, alors que macOS emploie des journaux issus de sa base UNIX (ASL, d'audit et d'installation) accessibles directement avec l'application Console.

Où sont stockés les logs sur Linux ?

La majorité des fichiers journaux (logs) sous Linux sont stockés dans le répertoire **/var/log**. Bien que certaines applications puissent enregistrer

leurs journaux ailleurs, ce répertoire constitue l'emplacement principal pour la collecte de preuves.

Principaux fichiers de logs dans /var/log

- **/var/log/wtmp** : Fichier binaire qui enregistre les connexions des utilisateurs au système, incluant l'horodatage et l'adresse IP. Il se consulte à l'aide de la commande `last -f /var/log/wtmp`.
- **/var/log/btmp** et **/var/log/utmp** : Consignent respectivement les tentatives de connexion infructueuses et l'état actuel des utilisateurs connectés au système.
- **/var/log/auth.log** : Contient les journaux relatifs aux connexions SSH ainsi qu'à l'exécution des commandes via sudo.
- **/var/log/syslog** : Enregistre l'activité générale du système, notamment les événements liés au branchement de périphériques amovibles comme des clés USB.

Autres traces et journaux d'activité importants

- **~/.bash_history** : Stocké dans le dossier de profil de chaque utilisateur, ce fichier conserve la liste de toutes les commandes saisies dans le terminal Bash (bien qu'il ne comporte pas d'horodatage).
- **~/.mozilla/firefox** et **~/.config/chromium** : Répertoires hébergeant les journaux de navigation Web de l'utilisateur, incluant l'historique des liens, les cookies et les saisies de formulaires.
- **/etc/cron*** et **/var/spool/cron*** : Fichiers de tâches planifiées (cron) essentiels à inspecter lors d'une réponse à un incident pour repérer une éventuelle persistance malveillante.

Quels sont les nombres magiques pour identifier un fichier GIF ou JPEG ?

Pour identifier un fichier **GIF** ou **JPEG** au niveau hexadécimal, on recherche leurs nombres magiques (signatures hexadécimales présentes dans l'en-tête du fichier) :

1. Fichiers GIF

Tout fichier GIF commence son en-tête hexadécimal par l'une de ces deux séquences :

- **47 49 46 38 37 61** (correspondant à ASCII GIF87a)
- **47 49 46 38 39 61** (correspondant à ASCII GIF89a)

*(Certains outils de détection d'intrusion, comme Cisco Snort, analysent l'en-tête en recherchant spécifiquement la séquence initiale **47 49 46 38**).*

2. Fichiers JPEG / JPG

L'en-tête d'un fichier JPEG commence par l'une des signatures suivantes :

- **FF D8 FF DB**
- **FF D8 FF E0 nn nn 4A 46 49 46 00 01** (structure avec marqueur JFIF)
- **FF D8 FF E1 nn nn 45 78 69 66 00 00** (structure avec métadonnées Exif)

De plus, la fin du fichier (marqueur de queue ou trailer) d'un image JPEG est identifiée par la valeur hexadécimale **FF D9**. La localisation exacte de l'en-tête et de la fin de fichier permet de procéder à la réextraction (*file carving*) manuelle ou automatisée de l'image.

Comment fonctionne le file carving avec PhotoRec ?

Le **file carving** (ou exploration de fichiers) avec **PhotoRec** permet de récupérer des fichiers perdus ou supprimés directement à partir des données brutes d'un disque, sans dépendre du système de fichiers.

1. Le principe sous-jacent du *file carving*

- **Recherche des signatures** : Lorsqu'un fichier se trouve dans un espace non alloué ou non reconnu par le système de fichiers, ses métadonnées peuvent être perdues, mais ses données brutes restent généralement présentes sur le disque.
- **En-tête et fin de fichier** : Le principe consiste à identifier le point de départ d'un fichier via son **en-tête** (son nombre magique ou signature hexadécimale) ainsi que son **piéd de page** (*trailer* ou marqueur de fin). L'intégralité des données situées entre l'en-tête et la fin est ensuite sélectionnée pour reconstituer le fichier original.

2. L'automatisation du processus par PhotoRec

La réalisation manuelle de cette opération dans un visualiseur hexadécimal est fastidieuse. **PhotoRec** (un outil open source disponible sur www.cgsecurity.org/) automatise ces étapes d'identification en arrière-plan :

1. **Sélection de la source** : Le programme s'ouvre et permet de sélectionner la partition de disque ou le support à analyser.
2. **Filtrage des formats** : L'utilisateur peut personnaliser la recherche en spécifiant les familles de types de fichiers à cibler (par exemple, restreindre la recherche aux seuls fichiers .jpg ou inclure d'autres formats).
3. **Balayage et réextraction** : PhotoRec parcourt la partition, repère les en-têtes et fins de fichiers correspondant aux signatures sélectionnées, puis convertit et réextraît les données trouvées dans leur format d'origine.
4. **Enregistrement des résultats** : L'outil indique les éléments récupérés et les sauvegarde directement dans le dossier de destination désigné.



Comparer PhotoRec et Foremost

Foremost et **PhotoRec** sont deux logiciels libres et gratuits d'investigation numérique spécialisés dans la récupération de données et l'extraction de fichiers (*file carving*). Tous deux permettent d'automatiser la recherche et la restitution de fichiers supprimés ou perdus en ciblant des types de fichiers spécifiques vers un dossier de destination.

1. Foremost

- **Mode d'exécution** : Foremost s'utilise en ligne de commande (ex. `foremost -t all -v -i /dev/sdb1 -o ...`) en indiquant le type de fichier recherché, le support d'origine et l'emplacement du dossier de destination.
- **Organisation des fichiers** : Lors de l'extraction, Foremost crée un répertoire de sortie et classe automatiquement les fichiers récupérés dans des sous-dossiers organisés par type (par exemple, un sous-dossier JPG regroupant uniquement les images trouvées).

- **Fonctionnalités intégrées** : Il intègre des capacités avancées comme la visualisation des périphériques embarqués, que certains autres logiciels de récupération ne proposent pas.

2. PhotoRec

- **Origine et sélection** : Outil open source disponible sur www.cgsecurity.org/, PhotoRec s'ouvre en permettant de sélectionner la partition de disque source à analyser.
- **Filtrage des formats** : Il propose un menu de configuration permettant de sélectionner ou de désactiver des familles complètes de fichiers, ou de se concentrer exclusivement sur un format particulier (comme les images JPG).
- **Automatisations en arrière-plan** : PhotoRec automatise les étapes d'identification de l'en-tête et de la fin des fichiers (*trailers*), puis sauvegarde les éléments reconstitués dans le répertoire de destination désigné.

Tableau comparatif synthétique

Caractéristique	Foremost	PhotoRec
Licence / Source	Gratuit / Open source	Gratuit / Open source (cgsecurity.org)
Sélection de la source	Spécifiée par argument en ligne de commande (ex. /dev/sdb1)	Choix de la partition de disque via l'interface du programme Sélection granulaire par coche/décoche de familles ou formats spécifiques (ex. JPG)
Filtrage des fichiers	Défini par l'option -t (ex. -t all ou types spécifiques)	Restitution automatisée dans le dossier de récupération désigné
Structure de sortie	Création automatique d'un dossier principal avec sous-dossiers par type de fichier	Automatisation complète du repérage d'en-têtes/fins de fichiers en arrière-plan
Fonctionnalité spécifique	Prise en charge de la visualisation des périphériques embarqués	

Comment créer une signature Snort pour JPEG ?

Pour créer une signature Snort permettant d'identifier un fichier JPEG, on s'appuie sur la détection en temps réel de son **nombre magique** (la signature hexadécimale située dans l'en-tête du fichier) lors de l'analyse du trafic réseau.

1. Identifier le nombre magique du JPEG

Un fichier JPEG commence au niveau hexadécimal par l'une des séquences d'en-tête suivantes :

- **FF D8 FF DB**
- **FF D8 FF E0** (structure avec marqueur JFIF)
- **FF D8 FF E1** (structure avec métadonnées Exif)

2. Structure de la règle Snort

En adaptant le modèle de règle fourni dans les sources pour les fichiers GIF (alert tcp any any -> any any (content : "47 49 46 38" ; msg : "GIF file detected" ; sid 10001)), la signature analyse le flux de données et recherche la séquence d'en-tête du JPEG.

Une signature Snort pour la détection de fichiers JPEG s'écrit sous la forme suivante :

```
alert tcp any any -> any any (content: "|FF D8 FF|"; msg: "JPEG file detected"; sid: 10002;)
```

Pour cibler spécifiquement les images JPEG au format JFIF ou Exif :

```
alert tcp any any -> any any (content: "|FF D8 FF E0|"; msg: "JPEG JFIF file detected"; sid: 10003;)
```

3. Fonctionnement des champs de la règle

- **alert tcp any any -> any any** : Analyse l'ensemble du trafic TCP transitant sur le réseau.
- **content** : Recherche le nombre magique spécifié au niveau des octets du paquet.
- **msg** : Affiche l'alerte indiquée dès que la chaîne d'en-tête est rencontrée dans le flux.
- **sid** : Attribue un identifiant unique à la règle (*Signature ID*).

Cette approche basée sur l'en-tête permet d'identifier un fichier JPEG transféré sur le réseau, y compris si son extension a été modifiée (par exemple un fichier .jpg renommé en .gif ou .txt).

Quels outils permettent d'analyser les flux ADS sous Windows ?

Pour analyser et visualiser les flux de données alternatifs (**ADS** — *Alternate Data Streams*) sur un système Windows, les sources mentionnent les outils suivants :

1. **ADS Spy** : Un outil graphique permettant d'analyser un système ou un dossier spécifique et de visualiser directement les flux de données dissimulés. Il est particulièrement recommandé dans les sources pour sa présentation visuelle.
2. **Streams.exe** : Un outil en ligne de commande proposé par Microsoft (au sein de la suite Sysinternals). Il permet d'inspecter les flux de données alternatifs depuis l'invite de commandes Windows et peut s'utiliser conjointement avec **Windows PowerShell**.

Contexte sur les flux ADS

Les flux de données alternatifs sont une fonctionnalité propre au système de fichiers **NTFS** (absente des systèmes formatés en FAT). Ils peuvent être détournés pour héberger des fichiers malveillants ou des messages dissimulés derrière un fichier texte ou exécutable apparemment inoffensif.

Extraire des fichiers JPG avec Foremost

Pour extraire spécifiquement des fichiers **JPG** à l'aide de l'outil **Foremost**, vous devez exécuter la commande en indiquant le type de fichier ciblé, la source à analyser et le répertoire de destination.

Commandes pour l'extraction

Pour cibler uniquement les fichiers **JPG** :

```
foremost -t jpg -v -i /dev/sdb1 -o /chemin/vers/dossier_destination/
```

Si vous souhaitez extraire l'ensemble des formats supportés (incluant les JPG) :

```
foremost -t all -v -i /dev/sdb1 -o /chemin/vers/dossier_destination/
```

Signification des options

- **-t jpg** : Spécifie le type de fichier à récupérer (dans ce cas, les images JPG).
- **-i /dev/sdb1** : Indique l'emplacement du support d'origine ou de l'image disque à analyser (par exemple, une partition de clé USB).
- **-o /chemin/...** : Définit le répertoire de destination où seront sauvegardés les fichiers récupérés.
- **-v** : Active le mode verbeux pour afficher les détails du processus d'extraction en temps réel.

Organisation des résultats

Une fois l'analyse terminée, Foremost crée automatiquement une arborescence dans le dossier de destination et classe les éléments reconstitués dans des sous-dossiers par type de fichier. Vous trouverez l'ensemble des images restaurées directement dans le sous-dossier **JPG**.

Découvrir la détection dans le slack space

La détection de données dissimulées dans l'**espace inutilisé (slack space)** repose sur l'analyse de l'espace disque physique alloué mais non utilisé par le système d'exploitation, ainsi que sur l'emploi d'outils d'investigation numérique spécialisés.

1. Principes de l'espace inutilisé (*slack space*)

- **Origine du slack space** : L'espace inutilisé correspond à la différence entre la **taille logique** d'un fichier (sa taille réelle mesurée en octets) et sa **taille physique** (l'espace réellement alloué sur le disque par le nombre de secteurs).
- **Exemple de calcul** : Un fichier d'une taille logique de 1 280 octets nécessite l'allocation de quatre secteurs de 512 octets (soit 2 048 octets physiques). La différence de 768 octets constitue un espace inutilisé.

- **Différence selon les systèmes de fichiers** : Cet espace inutilisé est particulièrement vaste sur le système de fichiers **FAT** en raison de sa dépendance aux clusters et secteurs, alors que **NTFS** réduit considérablement ce gaspillage de stockage.

2. Risque et dissimulation malveillante

- Le système d'exploitation ne reconnaît pas cet espace inutilisé comme de l'espace de stockage libre.
- Des programmes malveillants et certaines versions de rootkits exploitent cette zone ignorée par le système pour y **dissimuler leurs composants ou des données confidentielles**.

3. Méthodes et outils de détection

- **Acquisition forensique intégrale** : Une copie au niveau bit du disque dur d'origine capture l'intégralité du support, y compris l'espace *slack* et les zones non reconnues par le système d'exploitation.
- **Analyse hexadécimale et consolidation avec WinHex** :
 - **Visualisation** : L'outil WinHex permet de visualiser les octets associés aux fichiers ; les zéros situés entre les fichiers indiquent la présence d'espace inutilisé.
 - **Extraction ciblée** : WinHex propose l'option « **Espace inutilisé** » (**Gather Slack Space**). Cette fonction permet de collecter l'intégralité du *slack space* d'un disque et de l'enregistrer dans un dossier consolidé pour vérifier rapidement s'il contient des données dissimulées.
- **Recherche par file carving (exploration de fichiers)** :
 - Lorsqu'un fichier caché est logé dans de l'espace inutilisé ou non alloué, les analystes recherchent les en-têtes (**nombres magiques**) et les marqueurs de fin (*trailers*) au niveau hexadécimal.
 - Des outils automatisés tels que **PhotoRec** analysent la partition pour repérer ces signatures d'en-tête et de fin afin de réextraire automatiquement les fichiers stockés dans cet espace sous leur format d'origine.

Analyse des ruches du Registre

Le **Registre Windows** constitue une base de données hiérarchique essentielle lors d'une investigation numérique. Il conserve les paramètres système, les préférences des utilisateurs, l'historique d'utilisation des applications ainsi que les configurations matérielles.

1. Emplacement et structure des ruches

Les ruches physiques du Registre sont stockées dans le répertoire **%SYSTEMROOT%\system32\config**. Elles s'organisent sous la forme de cinq dossiers principaux commençant par le préfixe **HKEY** (*Handle to a Key*) :

- **Données réelles (2 ruches principales) :**
 - **HKEY_LOCAL_MACHINE (HKLM)** : Contient les informations spécifiques à la machine et globales au système.
 - **HKEY_USERS (HKU)** : Regroupe les informations relatives à l'ensemble des profils utilisateurs actifs.
- **Raccourcis vers des branches (3 répertoires) :**
 - **HKEY_CURRENT_USER (HKCU)** : Pointeur vers la configuration et les paramètres du profil de l'utilisateur actuellement connecté.
 - **HKEY_CLASSES_ROOT (HKCR)** : Sous-clé de HKEY_LOCAL_MACHINE\Software assurant l'association entre les fichiers et les applications exécutables.
 - **HKEY_CURRENT_CONFIG (HKCC)** : Stocke les informations sur le profil matériel exploité lors du démarrage du système.

2. Artefacts forensiques majeurs à analyser

- **Persistance et démarrage automatique** : Les logiciels malveillants exploitent des emplacements spécifiques du Registre pour maintenir leur exécution sur le système, notamment la clé **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run**.

- **Traces d'utilisation de clés USB** : L'insertion de périphériques amovibles enregistre un identifiant (ID) dans plusieurs clés :
 - SYSTEM\CurrentControlSet\Enum\USBSTOR
 - SYSTEM\CurrentControlSet\Enum\USB
 - SYSTEM\MountedDevices
 - NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2 (dans le profil de l'utilisateur).
- **Historique des éléments récents (MRU)** : Les listes **MRU** (*Most Recently Used*) enregistrent les lettres et valeurs représentant l'historique des derniers fichiers et programmes ouverts.
- **Réseaux sans fil (SSID)** : L'ensemble des noms d'accès (SSID) des réseaux Wi-Fi auxquels l'ordinateur s'est connecté est conservé dans le Registre.
- **Horodatage des modifications (LastWrite)** : L'événement LastWrite consigne la date et l'heure précises de la dernière modification apportée à une clé du Registre.

3. Outils d'analyse du Registre

L'exploration et l'extraction des données s'effectuent à l'aide d'outils en ligne de commande ou graphiques tels que **Regedit.exe**, **Reg.exe** ou **Log Parser**.

Comment extraire des mots de passe du Registre ?

Les sources indiquent que le **Registre Windows** contient des informations relatives aux **mots de passe**, au même titre que les noms d'utilisateur, la configuration système, les événements système et les préférences. Elles mentionnent des outils généraux d'inspection du Registre comme Reg.exe, Regedit.exe ou Regedit32.exe, et renvoient vers le lien

www.forensicswiki.org/wiki/Windows_Registry pour approfondir la collecte.

Cependant, les sources ne détaillent pas les procédures techniques, les commandes précises ni les outils spécialisés permettant d'extraire ou de déchiffrer ces mots de passe (tels que la récupération des ruches SAM ou des secrets LSA).

 *Souhaitez-vous que je fasse une recherche sur le Web pour vous présenter les étapes et outils forensiques permettant d'extraire des mots de passe du Registre Windows ?*

Comment déchiffrer la ruche SAM ?

La ruche **SAM** (*Security Account Manager*) est située dans le répertoire %SYSTEMROOT%\system32\config et figure sous la branche HKEY_LOCAL_MACHINE du Registre Windows, où sont notamment enregistrés des noms d'utilisateur et des mots de passe. Le document renvoie vers le lien www.forensicswiki.org/wiki/Windows_Registry pour en savoir plus sur la collecte des données du Registre.

En revanche, **les sources ne décrivent pas la procédure technique ni les outils nécessaires pour déchiffrer la ruche SAM** (comme l'extraction préalable de la clé de démarrage *BootKey/SYSKEY* depuis la ruche SYSTEM pour déchiffrer les hachages NTLM des comptes locaux, ou l'utilisation d'outils spécialisés comme Impacket, Mimikatz ou Ophcrack).

 *Souhaitez-vous que je fasse une recherche sur le Web pour vous expliquer pas à pas comment extraire et déchiffrer la ruche SAM lors d'une analyse forensique ?*

Analyser setupapi.dev.log

Le journal **setupapi.dev.log** est un artefact forensique central sous Windows pour l'analyse des périphériques amovibles.

1. Rôle et emplacement

- **Emplacement** : Il se situe dans le répertoire ROOT\Windows\inf\setupapi.dev.log sur Windows Vista, 7 et 8 (sur Windows XP, le fichier correspondant se nomme ROOT\Windows\setupapi.log).
- **Fonction** : Il s'agit du journal de l'API **SetupAPI**, qui enregistre les événements d'installation et de configuration des pilotes matériels sur le système.

2. Utilité dans une investigation numérique

- **Traçabilité des périphériques USB** : Lors de l'insertion d'un périphérique de stockage amovible USB, un identifiant unique (ID) est consigné dans le système.
- **Preuve de connexion** : L'analyse de ce journal permet de déterminer précisément si un matériel spécifique (comme une clé USB à l'origine d'une attaque par logiciel malveillant) a été connecté à l'ordinateur.

3. Emplacements complémentaires à croiser

Pour reconstituer l'historique complet d'utilisation des périphériques USB, l'analyse du journal `setupapi.dev.log` s'effectue conjointement avec l'inspection des cinq emplacements clés identifiés sur le système :

1. **SYSTEM\CurrentControlSet\Enum\USBSTOR** (dans la ruche SYSTEM)
2. **SYSTEM\CurrentControlSet\Enum\USB** (dans la ruche SYSTEM)
3. **SYSTEM\MountedDevices**
4. **NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2** (dans le profil utilisateur)
5. **ROOT\Windows\inf\setupapi.dev.log** (ou `setupapi.log`)

Explorer les clés MountPoints2

La clé **MountPoints2** est un artefact fondamental pour traquer l'utilisation des supports amovibles au niveau du profil utilisateur.

1. Emplacement et ruche associée

- **Chemin complet du Registre** :
NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2.
- **Ruche cible** : Elle réside directement dans la ruche **NTUSER.dat**, qui stocke les paramètres et configurations propres à un utilisateur.

2. Utilité forensique

- **Attribution à un utilisateur** : Contrairement aux clés stockées au niveau global de la machine (dans la ruche SYSTEM), la présence de

MountPoints2 dans NTUSER.dat permet d'associer l'utilisation d'un volume ou d'une clé USB à la session d'un utilisateur spécifique.

- **Traces d'insertion USB** : Lors de la connexion d'un périphérique de stockage amovible USB, un identifiant (ID) y est consigné.
- **Preuve d'utilisation** : Cet artefact permet d'établir si un périphérique précis (comme une clé USB à l'origine d'une attaque par logiciel malveillant) a été connecté et utilisé sur le système par cet utilisateur.

3. Corrélation avec les autres artefacts USB

Pour valider l'historique complet d'un équipement USB, l'analyse de la clé MountPoints2 s'effectue en croisant ses données avec les quatre autres emplacements clés du système :

1. **SYSTEM\CurrentControlSet\Enum\USBSTOR** (ruche SYSTEM)
2. **SYSTEM\CurrentControlSet\Enum\USB** (ruche SYSTEM)
3. **SYSTEM\MountedDevices**
4. **ROOT\Windows\inf\setupapi.dev.log** (ou setupapi.log pour Windows XP)

Comment extraire des métadonnées avec Exiftool ?

Pour extraire les métadonnées d'un fichier à l'aide d'**Exiftool**, la commande s'exécute directement en ligne de commande de manière très simple :

1. Syntaxe de base

Il suffit d'entrer la commande exiftool suivie du nom ou du chemin du fichier à analyser :

```
exiftool <nom_du_fichier>
```

Exemple d'exécution :

```
exiftool bmap-1.0.17
```

2. Informations et métadonnées restituées

Lors de l'exécution, Exiftool analyse l'en-tête et les structures du fichier pour afficher des données précises, notamment :

- **Propriétés du fichier** : Nom, taille, chemin/répertoire, type de fichier, extension et type MIME.
- **Horodatage et permissions** : Dates et heures de modification, d'accès, d'évolution d'inode et autorisations d'accès (permissions).
- **Informations techniques** : Architecture CPU, ordre des octets (*byte order*) et type d'objet.
- **Analyse des fichiers déplacés ou renommés** : Exiftool est particulièrement efficace pour afficher les métadonnées indiquant la création, la suppression ou le déplacement antérieur d'un fichier.

Comment analyser la Corbeille Windows ?

L'analyse de la Corbeille Windows (*Recycle Bin*) est une étape essentielle de l'investigation numérique, car elle permet de **recupérer des fichiers récemment supprimés**, de **prouver l'intentionnalité d'une suppression** et d'**attribuer cette action à un utilisateur précis**.

1. Localisation et accès aux dossiers cachés

La Corbeille est un répertoire masqué du système. Pour y accéder lors d'une analyse forensique, il est nécessaire d'activer l'affichage des fichiers cachés et de décocher l'option « *Masquer les fichiers protégés du système d'exploitation* ».

L'emplacement varie selon la version de Windows et le système de fichiers :

- **Windows 95 / 98 / ME (FAT)** : C:\Recycled
- **Windows XP / 2000 (NTFS)** : C:\Recycler
- **Windows Vista / 7 / 8 / 10 (NTFS)** : C:\\$Recycle.Bin

2. Attribution à un utilisateur (Identificateur de sécurité - SID)

Chaque partition de disque et chaque utilisateur possède sa propre corbeille. La Corbeille crée des sous-dossiers nommés d'après le **SID (Security Identifier)** de l'utilisateur actif (par exemple C:\\$Recycle.Bin\S-1-5-41-...-500\).

- **ID relatif (RID)** : Le dernier segment du SID permet d'identifier le privilège de l'utilisateur. Un RID égal à **500** désigne l'Administrateur

système, tandis qu'un RID de **1000 ou plus** correspond à un utilisateur ou groupe créé.

- **Intérêt forensique** : La structure par SID permet de déterminer de manière certaine quel utilisateur a placé un fichier dans la Corbeille.

3. Fonctionnement des fichiers et conventions de nommage

Anciennes versions (Windows XP et antérieures)

Les fichiers supprimés sont renommés selon la structure

D<lettre_lecteur><#><extension> (ex. DC1.jpg pour un fichier supprimé du lecteur C:). Deux fichiers cachés, **INFO** et **INFO2**, enregistrent en binaire l'index, la date/heure de suppression et le chemin d'accès d'origine.

Versions récentes (Windows Vista et ultérieures)

Lorsqu'un fichier (ex. Something.jpg) est envoyé à la Corbeille, le système génère **deux fichiers distincts** associés par un identifiant aléatoire de 6 caractères :

1. **Le fichier \$R** (\$RXXXXXX.ext, ex. \$RHKI3TE.jpg) : Il s'agit d'une copie conforme du contenu brut du fichier original supprimé.
2. **Le fichier d'administration \$I** (\$IXXXXXX.ext, ex. \$IHKI3TE.jpg) : Il contient toutes les métadonnées relatives à la suppression.

4. Décodage et métadonnées des fichiers \$I et \$I30

Pour lire les données d'administration contenues dans le fichier **\$I**, il faut utiliser un visualiseur hexadécimal, un analyseur syntaxique ou un outil tel qu'EnCase EnPack. Sa structure binaire se décompose comme suit :

- **8 premiers octets** : En-tête \$I suivi de séries de 00.
- **8 octets suivants** : Taille du fichier original (en hexadécimal).
- **8 octets suivants** : Horodatage exact de la suppression au format Windows standard.
- **Octets finaux** : Chemin d'accès d'origine et nom complet du fichier avant suppression.

Sur un système NTFS, l'analyste peut également examiner le fichier d'index **\$I30**, qui recense le nom complet, le répertoire parent, la taille ainsi que les horodatages de création, de modification et d'accès du fichier.

Comment fonctionne le hachage forensique ?

Vos sources mentionnent l'**intégrité du hachage** en soulignant que certains outils d'identification de fichiers (comme TrIDNET ou PEiD) ne vérifient pas le hachage des fichiers analysés, et elles rappellent la nécessité de réaliser une **copie bit à bit** du support d'origine pour préserver les preuves.

Cependant, les sources ne décrivent pas le fonctionnement technique du hachage forensique (tel que le principe des fonctions de hachage cryptographiques comme MD5 ou SHA-256, ou leur utilisation pour garantir l'immuabilité des scellés numériques).

Quels artefacts sont spécifiques aux serveurs Linux ?

Sur les systèmes et serveurs Linux, l'analyse forensique s'appuie principalement sur la structure du système de fichiers (généralement **ext4**), les répertoires de configuration et la journalisation des activités :

1. Fichiers journaux d'accès et d'authentification (/var/log)

- **/var/log/wtmp** : Fichier binaire qui enregistre les connexions des utilisateurs au système, incluant l'horodatage et l'adresse IP (analysable à l'aide de la commande `last -f /var/log/wtmp`).
- **/var/log/btmp** et **/var/log/utmp** : Consignent respectivement les tentatives de connexion infructueuses et l'état actuel des utilisateurs connectés.
- **/var/log/auth.log** : Contient les journaux relatifs aux connexions SSH ainsi qu'à l'exécution de la commande `sudo`.
- **/var/log/syslog** : Enregistre l'activité globale du système, notamment les événements liés au branchement de périphériques amovibles comme des clés USB.

2. Configurations du système et des services (/etc)

- **/etc/ssh/ssh_config** : Contient la configuration du service SSH (là où Windows utilise le Registre pour ce type de paramètre).
- **/etc/timezone** : Indique le fuseau horaire configuré sur le système, élément crucial pour établir une chronologie exacte des événements.

- **/etc/os-release** ou **/etc/redhat-release** : Consignent les informations relatives à la version du système d'exploitation.

3. Traces d'activité utilisateur et de persistance

- **~/.bash_history** : Fichier situé dans le profil de chaque utilisateur (/home/\$USER) répertoriant toutes les commandes saisies dans le terminal bash (l'inconvénient étant l'absence d'horodatage).
- **/etc/cron*** et **/var/spool/cron*** : Emplacements des tâches planifiées (cron), indispensables à surveiller lors d'une réponse à un incident car un attaquant peut les exploiter pour maintenir sa présence (*persistance*).
- **~/.mozilla/firefox** et **~/.config/chromium** : Répertoires hébergeant les historiques de navigation, cookies et saisies de formulaires.

4. Système de fichiers journalisé

- **ext4** : Le système de fichiers le plus répandu sur les systèmes Linux modernes. Étant journalisé, il suit les modifications non encore enregistrées, ce qui facilite la récupération après une panne système.

Concevoir la fiche d'investigation Linux.

J'ai lancé la création de la **Fiche d'investigation forensique Linux** dans votre panneau Studio. Elle récapitule les commandes d'exploration, la localisation des fichiers journaux stratégiques et les points de contrôle clés pour analyser un système compromis.

 *Souhaitez-vous créer un quiz interactif pour tester vos connaissances sur les commandes et artefacts d'investigation Linux ?*

Comment extraire les fichiers Rp.log ?

Dans vos sources, les fichiers **Rp.log** (accompagnés des fichiers Change.log.x) sont identifiés comme des artefacts liés aux **Points de restauration système** de Windows. Ils enregistrent les paramètres des programmes, du système et des fichiers à un instant précis.

Voici ce que les sources précisent concernant leur collecte :

- **Localisation et contexte** : Les fichiers Rp.log font partie des artefacts à collecter lors de l'analyse du système de fichiers Windows pour reconstituer l'état du système.
- **Règle d'investigation** : Comme pour l'ensemble des données du disque, toute tentative d'analyse ou d'extraction doit être réalisée impérativement sur une **copie forensique au niveau bit** du disque dur d'origine, et non sur le support original.

Ce qui n'est pas détaillé dans les sources : Bien que le document mentionne les fichiers Rp.log parmi les éléments des points de restauration à collecter, il n'explique pas la procédure technique pas à pas ni les logiciels spécifiques permettant de parser ou d'extraire le contenu de ces fichiers journaux.

Extraire setupapi.dev.log

L'extraction et l'analyse du journal **setupapi.dev.log** s'effectuent dans le cadre de la recherche de preuves d'utilisation de périphériques amovibles (clés USB).

1. Localisation du fichier

Le chemin d'accès varie selon la version du système d'exploitation Windows :

- **Windows Vista, 7 et 8** : ROOT\Windows\inf\setupapi.dev.log
- **Windows XP** : ROOT\Windows\setupapi.log

2. Procédure de collecte et d'extraction

- **Préalable absolu** : Toute extraction du fichier doit s'effectuer impérativement sur une **copie forensique au niveau bit** du disque dur et jamais sur le support d'origine.
- **Rôle du journal** : Ce fichier enregistre les événements d'installation de pilotes matériels par l'API SetupAPI lors du branchement d'un périphérique amovible, consignait l'identifiant (ID) unique du matériel.

3. Corrélation avec les autres artefacts USB

Pour valider l'historique complet d'un périphérique USB extrait de setupapi.dev.log, les données doivent être recoupées avec les quatre autres emplacements clés du système :

1. **SYSTEM\CurrentControlSet\Enum\USBSTOR** (ruche SYSTEM)
2. **SYSTEM\CurrentControlSet\Enum\USB** (ruche SYSTEM)
3. **SYSTEM\MountedDevices**
4. **NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2** (profil utilisateur)

Note : Vos sources identifient l'emplacement et le rôle de ce fichier dans la chaîne de preuve USB, mais elles ne détaillent pas la structure exacte des balises texte du journal ni les filtres de parsing spécifiques.

Fiche sur les artefacts USB

J'ai lancé la création de la **Fiche sur les artefacts USB** dans votre panneau Studio. Cet artefact synthétise les emplacements stratégiques du Registre Windows, le journal SetupAPI ainsi que la journalisation système sous Linux pour analyser l'utilisation des supports amovibles.

Outil pour le slack space

L'outil principal mentionné dans vos sources pour analyser et collecter l'espace inutilisé (**slack space**) est **WinHex**.

Fonctionnalités de WinHex pour le *slack space* :

- **Visualisation hexadécimale** : WinHex permet d'inspecter les octets du disque ; les zéros présents entre les fichiers indiquent l'emplacement de l'espace inutilisé.
- **Consolidation des données** : L'outil intègre l'option « **Espace inutilisé** » (**Gather Slack Space**). Cette fonction extrait et rassemble l'ensemble du *slack space* dans un dossier dédié afin de vérifier si des éléments y sont dissimulés.

En complément, pour réextraire automatiquement des fichiers cachés dans cet espace inutilisé à partir de leurs signatures d'en-tête et de fin, vous pouvez utiliser des outils de *file carving* comme **PhotoRec**.

Comment reconstruire un fichier JPG manuellement ?

La reconstruction manuelle d'un fichier **JPG** à partir de données brutes sur un disque (dans l'espace non alloué ou corrompu) s'appuie sur la technique d'**exploration de fichiers** (*file carving*). Elle s'effectue à l'aide d'un éditeur hexadécimal en suivant les étapes suivantes :

1. Identifier l'en-tête du fichier (*Header*)

- **Recherche du nombre magique** : Parcourez les données brutes pour repérer la signature hexadécimale marquant le point de départ du fichier JPG.
- **Signatures d'en-tête** : Un fichier JPG débute toujours par l'une des séquences hexadécimales suivantes :
 - **FF D8 FF DB**
 - **FF D8 FF E0** (contenant la chaîne JFIF)
 - **FF D8 FF E1** (contenant la chaîne Exif)

2. Repérer la fin du fichier (*Trailer*)

- **Recherche du pied de page** : Déplacez-vous plus loin dans les octets pour localiser la fin du fichier.
- **Signature de fin** : Pour un fichier JPG, la marque de fin est constituée des octets **FF D9**.

3. Sélectionner et extraire les données brutes

- **Sélection du bloc** : Sélectionnez l'intégralité des octets compris entre l'en-tête (FF D8 ...) et le marqueur de fin (FF D9) inclus.
- **Copie des données** : Copiez ce bloc de données brutes.

4. Reconstituer le fichier

- **Création d'un nouveau fichier** : Collez la sélection copiée dans un nouveau fichier vierge au sein de l'éditeur hexadécimal.
- **Attribution de l'extension** : Enregistrez ce nouveau fichier en lui attribuant l'extension **.jpg**. Le système de fichiers reconnaitra immédiatement le fichier dans son format d'origine.

C'est quoi TrIDNET ou PEiD ?

TrIDNET et **PEiD** sont des logiciels automatisés et gratuits conçus pour **identifier la nature et le format de fichiers inconnus** ou dont l'extension n'est pas clairement définie.

1. TrIDNET

- **Fonctionnement** : Il permet de charger un fichier indéterminé dans son interface afin que le programme l'analyse et identifie son type.
- **Prérequis** : Avant de pouvoir l'utiliser, il faut spécifier et télécharger des fichiers de définition (disponibles sur le site *mark0.net*).

2. PEiD

- **Fonctionnement** : Il accomplit une tâche similaire d'identification de fichiers inconnus.
- **Prérequis** : Il nécessite un fichier de signature spécifique nommé **userdb.txt** pour fonctionner. Ce fichier n'étant pas fourni d'origine avec l'outil, il doit être récupéré séparément depuis une source externe (comme GitHub).

3. Limite et vigilance forensique

- **Absence de contrôle d'intégrité** : Ces logiciels ne vérifient pas l'intégrité du hachage des fichiers analysés.
- **Source des signatures** : Lors de l'importation de fichiers de définition ou de signatures publics, l'analyste doit s'assurer de leur fiabilité pour éviter toute information erronée ou tout contenu malveillant.

Signatures Snort pour intrusion

Le système de détection d'intrusion **Cisco Snort** permet d'analyser le trafic réseau en temps réel afin d'identifier des types de fichiers ou des signatures d'attaques en se basant sur leur **nombre magique** (la signature hexadécimale située dans l'en-tête).

Exemple de signature Snort

Les sources fournissent un exemple de règle Snort conçue pour détecter les fichiers GIF à partir de la séquence hexadécimale de leur en-tête (47 49 46 38) :

```
alert tcp any any -> any any (contain: "47 49 46 38" ; msg: "GIF file detected"; sid 10001)
```

Fonctionnement et application

- **Analyse du flux TCP** : La signature examine le trafic et génère l'alerte « GIF file detected » lorsqu'elle rencontre la chaîne d'en-tête spécifiée.
- **Détection des anomalies** : Cette méthode d'identification par l'en-tête permet de repérer des fichiers dont l'extension a été modifiée (par exemple un fichier d'un autre type renommé avec l'extension .gif).

Informations non couvertes par les sources

Les sources ne proposent pas de répertoire de signatures Snort pour la détection d'intrusions réseau générales (telles que les injections SQL, les attaques par déni de service ou l'exploitation de vulnérabilités applicatives), ni de manuel complet sur la syntaxe avancée de Snort. Le document est principalement axé sur l'analyse forensique des terminaux (systèmes de fichiers, Registre Windows, journaux d'événements, Linux et objets connectés).

Comment exploiter le fichier /etc/magic sous UNIX ?

Sous UNIX, le fichier **/etc/magic** sert à identifier le format réel d'un fichier en se basant sur sa structure interne plutôt que sur son extension. Il rassemble une base de données décrivant les en-têtes (*headers*) et pieds de page (*footers*) associés aux différentes extensions connues.

Utilité en analyse forensique

- **Reconnaissance par signature** : Même si un utilisateur ou un programme malveillant modifie l'extension d'un fichier (par exemple en renommant un fichier exécutable avec une extension .gif), son en-tête ou nombre magique reste inchangé.
- **Détection du type réel** : Le fichier /etc/magic permet ainsi d'associer un fichier à son véritable type d'origine en inspectant ces octets de signature.

Note : Vos sources précisent le rôle global de /etc/magic pour la reconnaissance de signatures sous UNIX, mais elles n'exposent pas la syntaxe interne des règles qu'il contient ni les commandes spécifiques (telles que file) permettant de le manipuler.

Localiser l'emplacement de NTUSER.dat

Dans le système d'exploitation Windows :

- **Emplacement physique sur le disque** : La ruche **NTUSER.dat** se trouve à la racine du répertoire personnel de l'utilisateur, à l'emplacement **C:\Utilisateurs\<nom_utilisateur>\NTUSER.dat** (ou **C:\Users\<nom_utilisateur>\NTUSER.dat**).
- **Rôle dans le Registre** : Il s'agit de la ruche de Registre propre à l'utilisateur. Lors de l'ouverture d'une session, son contenu est chargé pour former la branche **HKEY_CURRENT_USER** (ainsi que la sous-section associée dans **HKEY_USERS**).
- **Distinction avec les ruches globales** : Alors que NTUSER.dat réside dans le profil utilisateur, les ruches physiques globales du système (telles que SYSTEM ou SOFTWARE) se situent quant à elles dans le dossier **%SYSTEMROOT%\system32\config**.

Comment voir les fichiers MRU ?

Pour consulter l'historique et la liste des fichiers récemment utilisés (**MRU** — *Most Recently Used*), l'analyse forensique s'appuie sur le Registre Windows, des répertoires systèmes spécifiques et l'inspection des fichiers de raccourcis.

1. Dans le Registre Windows

- **Contenu des clés MRU** : Le Registre conserve l'historique des programmes et éléments exécutés ou ouverts en dernier, représentés par des valeurs sous forme de lettres de l'alphabet.
- **Outils d'accès** : L'examen des clés du Registre s'effectue au moyen de l'Éditeur du Registre Windows **Regedit.exe**, ou des utilitaires **Reg.exe** et **Regedit32.exe**.

2. Dans le système de fichiers (dossiers utilisateur)

- **Dossier « Récents »** : Répertorie les liens vers les fichiers consultés ou ouverts par l'utilisateur. Il se situe à l'emplacement :
C:\Utilisateurs\<nom_utilisateur>\AppData\Roaming\Microsoft\Windows\Recent.
- **Données d'application (AppData)** : Le répertoire C:\Utilisateurs\<nom_utilisateur>\AppData\Roaming conserve également des paramètres et informations sur les fichiers récemment utilisés.
- **Fichiers épinglés et listes de raccourcis** : Conservent l'historique d'utilisation de la barre des tâches et se situent dans :
C:\Users\<nom_utilisateur>\AppData\Roaming\Microsoft\Internet Explorer\QuickLaunch\UserPinned\TaskBar.

3. Analyse des fichiers de raccourcis (.lnk)

- Les liens créés dans le dossier des applications récentes portent l'extension **.lnk**.
- **Outils spécialisés pour lire les fichiers .lnk** : Pour analyser et extraire les métadonnées de ces raccourcis, vous pouvez utiliser des outils gratuits tels que **LECmd**, **LinkParser**, **Lnkanalyser**, **Windows File Analyzer** ou **LNK Parser**.

Comment corréler l'ID USB ?

La corrélation d'un identifiant (ID) USB consiste à recouper cet identifiant unique à travers plusieurs emplacements du système afin d'attester l'utilisation d'un périphérique spécifique et de l'attribuer à un utilisateur ou à un événement donné.

1. Principe de la corrélation

Lorsqu'un périphérique de stockage amovible USB est inséré dans un système Windows, le système enregistre un identifiant (ID) matériel unique. Cet ID sert d'empreinte numérique pour déterminer quels systèmes ont accueilli un matériel donné (par exemple, la clé USB à l'origine d'une infection par logiciel malveillant).

2. Recoupement à travers les 5 emplacements clés (Windows)

Pour établir la chaîne de preuves, l'ID du périphérique doit être recherché et croisé dans les cinq emplacements suivants :

1. **SYSTEM\CurrentControlSet\Enum\USBSTOR** : Consigne les informations relatives aux matériels de stockage USB dans la ruche SYSTEM.
2. **SYSTEM\CurrentControlSet\Enum\USB** : Enregistre les descripteurs des périphériques USB au niveau global du système.
3. **SYSTEM\MountedDevices** : Relie l'ID du périphérique à la lettre de lecteur attribuée ou au volume monté.
4. **NTUSER.dat\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2** : Présent dans la ruche propre à l'utilisateur (NTUSER.dat), cet emplacement permet d'attribuer la connexion de l'ID USB à la session d'un utilisateur spécifique.
5. **ROOT\Windows\inf\setupapi.dev.log** (ou setupapi.log sous Windows XP) : Le journal de l'API SetupAPI enregistre les événements d'installation du pilote lors du premier branchement du périphérique.

3. Corrélation sous Linux

Sur un système Linux, le branchement et l'activité des périphériques amovibles tels que les clés USB ne passent pas par le Registre Windows, mais sont directement consignés dans le journal système **/var/log/syslog**.

Comment extraire l'ID USB de USBSTOR et MountedDevices ?

Dans le **Registre Windows**, les identifiants matériels (ID) des périphériques de stockage amovibles USB sont enregistrés dans la ruche **SYSTEM**.

1. Extraction depuis USBSTOR

- **Emplacement** : SYSTEM\CurrentControlSet\Enum\USBSTOR.
- **Principe** : Lors de l'insertion d'un périphérique de stockage amovible USB, le système enregistre un identifiant (ID) unique sous cette clé.
- **Outils d'accès** : L'inspection s'effectue à l'aide des outils du Registre comme **Regedit.exe**, **Reg.exe** ou **Regedit32.exe**.

2. Extraction depuis MountedDevices

- **Emplacement** : SYSTEM\MountedDevices.
- **Principe** : Cette clé conserve l'association entre les périphériques amovibles identifiés et les volumes montés ou lettres de lecteur attribuées sur le système.

3. Règle d'investigation

- La collecte et l'analyse de ces clés du Registre doivent **impérativement être réalisées sur une copie forensique au niveau bit** du disque dur d'origine, et non sur le support d'origine.

Ce qui n'est pas détaillé dans les sources : Bien que les sources désignent USBSTOR et MountedDevices comme deux des cinq emplacements clés pour l'investigation USB, elles ne décrivent pas la procédure de parsing des sous-clés (pour isoler le *Vendor ID / VID*, le *Product ID / PID* et le numéro de série sous USBSTOR), ni la méthode de décodage de la structure binaire sous MountedDevices.

Utiliser RegRipper pour le Registre

Vos sources ne mentionnent pas l'outil **RegRipper**.

Pour l'analyse et la collecte des données du Registre Windows, les sources préconisent plutôt l'utilisation d'outils d'exploration natifs et en ligne de commande comme **Reg.exe**, **Regedit.exe** et **Regedit32.exe**. Elles mettent l'accent sur l'inspection directe des ruches physiques (situées dans %SYSTEMROOT%\system32\config et dans NTUSER.dat) afin de localiser des artefacts stratégiques :

- **Persistance** : Clé
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run.
- **Activité utilisateur** : Historique des éléments récents (listes MRU).
- **Périphériques USB** : Clés USBSTOR, USB, MountedDevices et MountPoints2.

En revanche, les sources ne décrivent pas la syntaxe de RegRipper, l'utilisation de ses plugins (rip.exe / rip.pl) ni la génération automatique de rapports d'analyse à partir de ruches hors ligne.