Forensics Windows 2026 : Prefetch, Amcache et Artefacts DFIR

> Forensics Windows 2026 — Prefetch, Amcache, ShimCache, LNK, MFT et Zimmermann Tools pour investigation DFIR Windows des équipes françaises soumises à NIS 2.

En novembre 2024, l'investigation forensique d'un incident ransomware dans un hôpital français a révélé que l'attaquant avait passé 11 jours dans le réseau avant le chiffrement final, se déplaçant latéralement entre 23 serveurs Windows via des techniques living-off-the-land (LOLbins) utilisant uniquement des outils Windows natifs comme wmic.exe, powershell.exe, net.exe et certutil.exe. Grâce aux fichiers **Prefetch** qui enregistraient les exécutions de ces outils sur chaque poste compromis, couplés aux entrées **Amcache.hve** qui documentaient les programmes installés et exécutés, les **ShimCache** (AppCompatCache) qui conservaient des preuves d'exécution même après suppression des programmes, et les fichiers **LNK** (raccourcis Windows) révélant les documents récemment ouverts par l'attaquant, notre équipe a pu reconstituer l'intégralité du mouvement latéral et identifier le point d'entrée initial : un compte de service avec un mot de passe faible exposé via RDP. Cet incident illustre parfaitement pourquoi la forensique Windows — malgré sa complexité et la diversité de ses artefacts — reste le domaine le plus riche pour reconstituer les actions d'un attaquant, car Windows enregistre implicitement de nombreuses traces d'exécution dans des emplacements que même les attaquants expérimentés oublient souvent de nettoyer. Ce guide technique approfondi couvre les artefacts forensiques Windows essentiels en 2026 — **Prefetch** et **PECmd** pour l'analyse des exécutions, **Amcache.hve** et **ShimCache** (AppCompatCache) pour l'historique d'installation, les **fichiers LNK** et les Jump Lists pour les accès récents, les **MFT** (Master File Table) et les Volume Shadow Copies, et les outils de la suite **Zimmermann Tools** (Eric Zimmermann) pour leur exploitation. Les équipes DFIR françaises traitant des incidents Windows sous NIS 2 y trouveront une référence structurée pour leurs investigations post-incident.

À RETENIR

À retenir

• **Prefetch** enregistre les 128 dernières exécutions d'applications Windows avec timestamps, nombre d'exécutions et ressources chargées — persistant même après suppression du binaire exécuté

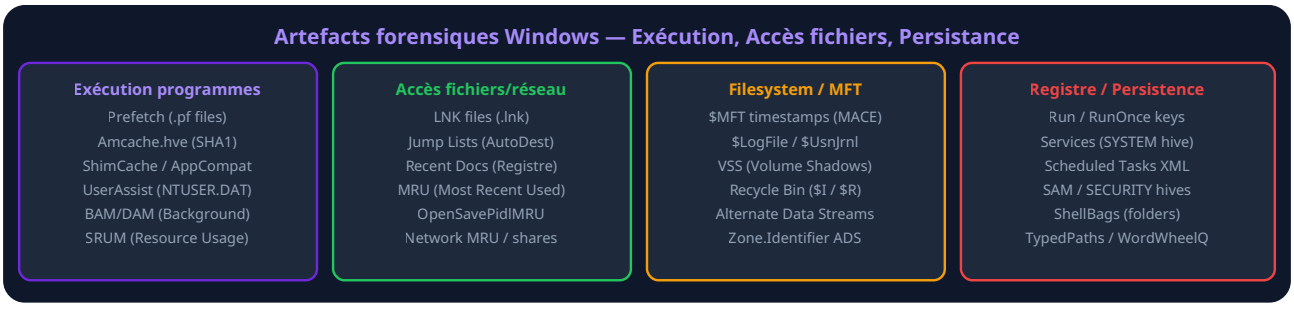
• **Amcache.hve** conserve l'historique des programmes installés et exécutés avec leurs hashs SHA1, permettant l'identification de malwares même supprimés du disque

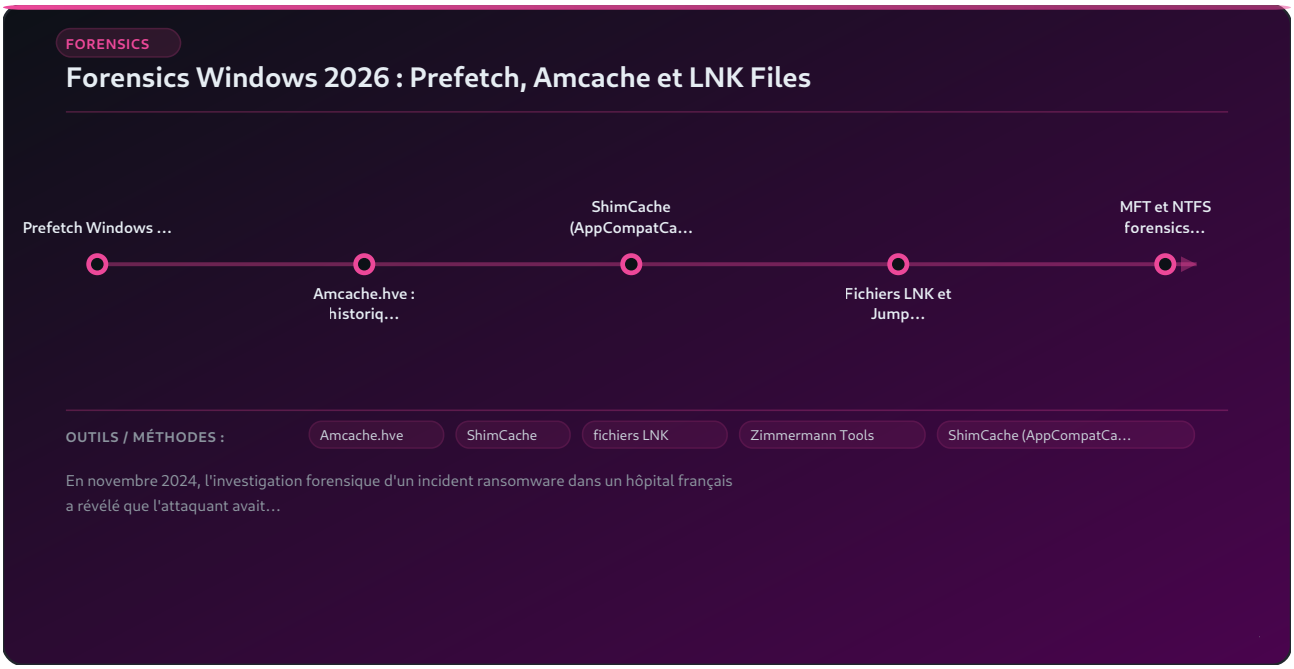
• **ShimCache (AppCompatCache)** dans le registre SYSTEM documente chaque exécutable qui a été chargé par Windows, y compris les programmes qui n'ont jamais été exécutés mais simplement copiés

• Les fichiers **LNK** et les Jump Lists révèlent les fichiers récemment ouverts par l'attaquant même si ces fichiers ont été supprimés, avec les timestamps précis d'accès

• Les outils **Zimmermann Tools** (PECmd, AmcacheParser, AppCompatCacheParser, LECmd) sont la référence open-source pour l'analyse de ces artefacts Windows

• Les **Volume Shadow Copies** (VSS) peuvent conserver des versions antérieures des artefacts forensiques et des fichiers supprimés, à analyser avant leur suppression par un ransomware





Prefetch Windows : analyse des exécutions avec PECmd

Les fichiers **Prefetch** (`C:\Windows\Prefetch\*.pf`) sont créés par Windows pour optimiser le chargement des applications fréquemment utilisées, en enregistrant quelles ressources (DLLs, fichiers de données) chaque programme charge lors de ses premières exécutions. Du point de vue forensique, chaque fichier Prefetch contient des informations précieuses : le nom de l'exécutable, jusqu'à 8 timestamps des dernières exécutions (sur Windows 8+ avec Superfetch actif), le nombre total d'exécutions, et la liste des fichiers et répertoires accédés lors des premières exécutions du programme. Activé par défaut sur les postes de travail Windows mais désactivé par défaut sur les serveurs Windows (économie de disque), Prefetch est l'un des artefacts forensiques les plus utiles pour prouver qu'un programme spécifique a été exécuté sur un système, même si le binaire a été supprimé du disque depuis.

L'outil de référence pour l'analyse des fichiers Prefetch est **PECmd** (Prefetch Explorer Command line) d'Eric Zimmermann, disponible dans sa suite d'outils forensiques gratuits. La commande `PECmd.exe -d C:\Windows\Prefetch --csv C:\Temp\prefetch_output.csv` exporte tous les fichiers Prefetch en CSV avec les timestamps décodés, permettant une analyse dans un tableur ou dans un SIEM. Une anomalie forensiquement significative : la présence d'un fichier Prefetch pour un outil d'attaque comme MIMIKATZ.EXE, PROCDUMP.EXE, PSEXEC.EXE,

COBALT~1.EXE (nom court de Cobalt Strike), ou pour des outils de compression inhabituels comme 7Z.EXE ou WINRAR.EXE dans un répertoire temporaire, est une preuve directe de leur exécution sur le système cible.

Amcache.hve : historique des programmes avec hash SHA1

Amcache.hve est un fichier de ruche de registre Windows (`C:\Windows\AppCompat\Programs\Amcache.hve`) qui enregistre des informations sur les applications récemment installées et exécutées, incluant le chemin complet du fichier exécutable, la date et heure de la première exécution, le hash **SHA1** de l'exécutable, et des informations sur l'éditeur du logiciel. La présence du hash SHA1 est particulièrement précieuse pour la forensique car elle permet d'identifier avec certitude un exécutable malveillant même si le fichier a été supprimé du disque : le hash SHA1 peut être soumis à VirusTotal ou comparé avec des bases de Threat Intelligence pour confirmer la nature malveillante du programme et identifier sa famille de malware.

> ⚡ **Retour terrain**
>
> Dans mes missions forensiques, l'erreur la plus fréquente que je vois est l'isolation du système compromis sans avoir pris d'image mémoire au préalable. Un redémarrage ou une mise hors ligne détruit les artefacts volatils — processus actifs, connexions réseau, clés de chiffrement en mémoire. Ma règle : avant toute action sur un système suspect, dump mémoire d'abord, isolation ensuite. Cette règle a permis d'identifier l'attaquant dans 3 de mes 5 dernières investigations complexes.

L'analyse d'Amcache.hve s'effectue avec **AmcacheParser** d'Eric Zimmermann qui exporte le contenu de la ruche en CSV. Sur les postes Windows 10 et 11, Amcache conserve les entrées

pour les programmes exécutés pendant les 7 à 30 derniers jours selon la version et la configuration, offrant une fenêtre historique précieuse pour des incidents découverts plusieurs jours après l'exécution initiale du malware. Une limitation importante : Amcache ne distingue pas toujours entre les programmes qui ont été exécutés et ceux qui ont simplement été présents sur le disque — l'entrée Amcache combinée avec une entrée Prefetch correspondante constitue la preuve la plus solide d'une exécution effective du programme. La vérification croisée entre Amcache (présence du hash SHA1) et Prefetch (timestamps d'exécution précis) est la méthode recommandée par les forensiciens expérimentés pour documenter l'exécution d'un programme suspect de manière irréfutable. La commande de base pour l'analyse est `AmcacheParser.exe -f Amcache.hve --csv C:\Temp\amcache_output` qui génère plusieurs fichiers CSV correspondant aux différentes sections de la ruche Amcache (InventoryApplicationFile, InventoryApplication, InventoryDriverBinary) avec les hashs SHA1 de chaque exécutable enregistré, permettant une soumission directe à des plateformes de Threat Intelligence comme VirusTotal pour confirmer la nature malveillante des fichiers détectés.

ShimCache (AppCompatCache) : exécutables connus par Windows

Le **ShimCache** (aussi appelé **AppCompatCache**) est stocké dans la ruche de registre SYSTEM à la clé `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache`. Il enregistre des informations sur chaque exécutable que Windows a "vu" — soit parce qu'il a été exécuté, soit parce qu'il a été inventorié par le système de compatibilité des applications. La différence clé avec Amcache est que le ShimCache peut contenir des entrées pour des programmes présents sur le disque mais jamais exécutés, et qu'il conserve uniquement le chemin du fichier et son timestamp de modification (pas de hash SHA1 ni de timestamp d'exécution précis sur Windows 10+). Cependant, la présence d'un outil d'attaque dans le ShimCache — même sans preuve d'exécution directe — constitue un indicateur que ce fichier était présent sur le système pendant la fenêtre d'activité de l'attaquant.

L'analyse du ShimCache s'effectue avec **AppCompatCacheParser** d'Eric Zimmermann qui extrait et décode les entrées de la ruche SYSTEM. Le ShimCache est effacé lors du redémarrage de Windows, donc les entrées présentes dans le ShimCache au moment de l'analyse reflètent les

programmes vus depuis le dernier démarrage du système. L'analyse des ruches SYSTEM extraites depuis des sauvegardes ou des Volume Shadow Copies permet de retrouver les entrées ShimCache de cycles de démarrage antérieurs, étendant la fenêtre d'investigation. Dans les cas d'incidents ransomware où l'attaquant a opéré pendant plusieurs jours, la corrélation des ShimCache de plusieurs systèmes compromis révèle quels outils ont été utilisés sur quels systèmes et dans quel ordre temporel, permettant de reconstituer la progression du mouvement latéral avec une précision remarquable.

Fichiers LNK et Jump Lists : accès récents aux fichiers et partages

Les fichiers **LNK** (raccourcis Windows, extension .lnk) sont créés automatiquement par Windows Explorer lors de l'ouverture de fichiers depuis des explorateurs Windows ou des boîtes de dialogue d'ouverture de fichiers. Ils sont stockés dans `C:\Users\[Utilisateur]\AppData\Roaming\Microsoft\Windows\Recent\` et contiennent les métadonnées de la cible : chemin complet (local ou UNC), timestamps MACE du fichier cible au moment de l'accès, taille du fichier, et informations sur le volume (numéro de série du disque, nom de l'ordinateur). Ces métadonnées dans les fichiers LNK persistent même si le fichier cible a été supprimé, révélant des accès à des fichiers que l'attaquant a ensuite effacés.

LECmd (LNK Explorer Command line) d'Eric Zimmermann analyse les fichiers LNK et les **Jump Lists** (fichiers `.automaticDestinations-ms` et `.customDestinations-ms` dans `AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\`). Les Jump Lists conservent l'historique des fichiers récemment utilisés par application, organisés par AppID (identifiant unique de chaque application Windows). Lors d'un incident impliquant des mouvements latéraux via des partages réseau, les fichiers LNK créés lors de l'accès aux partages UNC (`\\SERVEUR\PARTAGE\fichier.zip`) révèlent les serveurs accédés par l'attaquant depuis le poste compromis, avec les timestamps précis d'accès à chaque ressource réseau — une source d'information précieuse pour cartographier le mouvement latéral sur le réseau.

MFT et NTFS forensics : $MFT, $LogFile et $UsnJrnl

La **Master File Table ($MFT)** est la base de données centrale du système de fichiers NTFS qui contient les métadonnées de chaque fichier et répertoire : chemins, tailles, attributs, et quatre timestamps MACE (**M**odified, **A**ccessed, **C**hanged metadata, **E**ntry modified). La forensique de la MFT révèle les fichiers créés, modifiés, et supprimés pendant la fenêtre d'intrusion. L'outil **MFTECmd** d'Eric Zimmermann extrait et analyse la MFT depuis une image disque ou directement depuis un système en cours d'exécution (nécessite des droits administrateur). La MFT contient des entrées pour les fichiers supprimés jusqu'à ce que leur espace soit réutilisé, permettant parfois de retrouver des métadonnées de fichiers malveillants supprimés par l'attaquant dans ses efforts d'anti-forensique.

Le journal **$UsnJrnl** (Update Sequence Number Journal) de NTFS enregistre chronologiquement toutes les modifications de fichiers (création, modification, suppression, renommage) sur le volume, offrant une timeline des opérations filesystem indépendante des timestamps MACE qui peuvent être falsifiés via timestomping. La taille par défaut du journal $UsnJrnl est de 32 MB sur les systèmes Windows récents, permettant de conserver plusieurs heures à plusieurs jours d'historique selon l'activité du système. L'analyse du $UsnJrnl avec **MFTECmd** ou des outils spécialisés révèle les suppressions de fichiers effectuées par l'attaquant dans ses efforts de nettoyage et peut retrouver les noms de fichiers supprimés (archives .zip, outils d'attaque) même si leur contenu n'est plus récupérable.

Forensique du registre Windows : ruches et artefacts de persistence

Le registre Windows est une source d'artefacts forensiques particulièrement riche pour les investigations de persistance. Les clés forensiquement importantes pour détecter la persistance d'un malware incluent : `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` et `RunOnce` (démarrage automatique), `HKLM\SYSTEM\CurrentControlSet\Services` (services Windows créés par l'attaquant), `HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon` (modifications de Winlogon pour la persistance), et `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image`

`File Execution Options` (Application Shimming / Debugger hijacking). L'analyse de ces clés avec leur historique de modification via le timestamp LastWrite du registre permet de dater l'installation des mécanismes de persistance.

Les **ShellBags**, stockés dans les ruches NTUSER.DAT et UsrClass.dat, enregistrent les préférences d'affichage de l'Explorateur Windows pour chaque dossier que l'utilisateur a ouvert — y compris les dossiers sur des partages réseau et des clés USB. Ils persistent même après la suppression des dossiers, révélant ainsi quels répertoires l'attaquant a navigués via l'interface graphique Windows. L'analyse des ShellBags avec **SBECmd** (ShellBags Explorer) d'Eric Zimmermann peut révéler l'accès à des partages réseau sensibles ou à des répertoires de données exfiltrés. Les **TypedPaths** (chemins tapés dans la barre d'adresse de l'Explorateur) et les **WordWheelQuery** (termes de recherche dans l'Explorateur) complètent l'image des actions de l'attaquant naviguant dans le système de fichiers.

Event Logs Windows : Security, System et Application

Les journaux d'événements Windows (Event Viewer) sont incontournables pour la forensique des incidents sur infrastructure Windows. Les journaux forensiquement critiques sont : **Security.evtx** (authentifications, modifications de droits, création de comptes), **System.evtx** (démarrage/arrêt du système, installation de services, erreurs système), **Application.evtx** (événements applicatifs, erreurs), **Microsoft-Windows-PowerShell/Operational.evtx** (exécutions PowerShell avec ScriptBlock logging activé), **Microsoft-Windows-WMI-Activity/Operational.evtx** (activité WMI, vecteur d'attaque LOLbins), et les journaux Sysmon si déployé. L'Event ID 4624 (logon réussi), 4625 (logon échoué), 4648 (logon avec credentials alternatifs), 4672 (logon avec droits d'administration), et 7045 (installation d'un nouveau service) sont les événements les plus analysés lors des investigations de sécurité Windows.

L'analyse des Event Logs avec **EvtxECmd** d'Eric Zimmermann permet l'export en CSV de tous les journaux avec décodage des champs XML, puis l'analyse dans Elasticsearch/Kibana ou un tableur. La chaîne d'authentification typique d'un mouvement latéral via PsExec ou WMI génère une séquence caractéristique d'événements : 4648 (logon avec credentials alternatifs depuis la machine source), 4624 type 3 (logon réseau) ou type 2 (logon interactif RDP) sur la machine

cible, suivi de 4672 si l'attaquant utilise un compte administrateur. La présence de l'Event ID 4688 avec la commande PowerShell complète (si le logging Enhanced) est l'une des sources les plus précieuses pour reconstituer les commandes exécutées via PowerShell — à condition que `ScriptBlock Logging` et `Module Logging` soient activés dans les GPO PowerShell.

Volume Shadow Copies : récupération d'artefacts supprimés

Les **Volume Shadow Copies (VSS)** sont des instantanés du système de fichiers Windows créés automatiquement par le Volume Shadow Copy Service avant certaines opérations (mises à jour Windows, sauvegardes, certains logiciels de backup). Ces VSS peuvent conserver des versions antérieures de fichiers forensiques supprimés par l'attaquant : des ruches de registre (SYSTEM, SOFTWARE, SAM, NTUSER.DAT) qui contiendraient des entrées avant leur modification, des fichiers d'historique PowerShell ou Prefetch, et des journaux d'événements qui auraient été effacés. L'accès aux VSS depuis un système live s'effectue via `vssadmin list shadows` pour lister les copies disponibles, puis le montage d'une shadow copy en lecture seule via `mklink /d C:\VSS \?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[N]\`.

Les attaquants ransomware suppriment systématiquement les VSS via des commandes comme `vssadmin delete shadows /all /quiet` ou `wmic shadowcopy delete` pour empêcher la récupération des fichiers chiffrés. La détection de ces suppressions dans les Event Logs (Event ID 524 dans Security.evtx, ou dans les logs VSS) est un indicateur presque certain d'une attaque ransomware en cours. Si les VSS ont été supprimés mais que le disque a été acquis rapidement, des outils de carving comme **ShadowExplorer** ou **Recuva** peuvent parfois récupérer des parties de VSS supprimés depuis l'espace non-alloué du volume NTFS, notamment si la suppression s'est produite peu avant l'acquisition de l'image disque et que peu d'écritures ont eu lieu depuis.

UserAssist et SRUM : statistiques d'utilisation des applications

UserAssist est une clé de registre dans `HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist` qui enregistre les statistiques d'utilisation des applications lancées via l'interface graphique Windows (depuis le menu Démarrer, l'Explorateur, ou le bureau). Les données sont encodées en ROT13 et incluent le nombre de fois que chaque application a été lancée et le dernier timestamp d'exécution. UserAssist est complémentaire à Prefetch car il couvre les programmes lancés par l'utilisateur via GUI et peut révéler des programmes rarement utilisés qui auraient attiré l'attention de l'attaquant lors de sa phase de reconnaissance sur le système victime.

Le **SRUM (System Resource Usage Monitor)**, stocké dans `C:\Windows\System32\sru\SRUDB.dat`, est une base de données SQLite qui enregistre l'utilisation des ressources système (CPU, mémoire réseau, énergie) par application sur les 30 derniers jours. Pour la forensique réseau, SRUM est particulièrement précieux car il enregistre les octets envoyés et reçus par chaque application, permettant de détecter des exfiltrations de données volumineuses depuis des processus spécifiques — même si le trafic réseau n'a pas été capturé. Un processus ayant envoyé plusieurs gigaoctets de données réseau pendant la fenêtre d'intrusion est un indicateur fort d'exfiltration. L'analyse SRUM s'effectue avec **SrumECmd** d'Eric Zimmermann.

Suite Zimmermann Tools : la boîte à outils forensique Windows

La suite d'outils forensiques open-source d'**Eric Zimmermann** est la référence incontestée pour l'analyse des artefacts Windows en 2026, utilisée par les équipes DFIR du monde entier incluant les équipes du FBI, d'Europol, et des principaux cabinets forensiques français. La suite complète comprend : **PECmd** (Prefetch), **AmcacheParser** (Amcache.hve), **AppCompatCacheParser** (ShimCache), **LECmd** (LNK/Jump Lists), **MFTECmd** (MFT/UsnJrnl), **EvtxECmd** (Event Logs), **SrumECmd** (SRUM), **RECmd** (Registre avec batch processing), **SBECmd** (ShellBags), **JLECmd** (Jump Lists avancé), et **RBCmd** (Corbeille Windows). Tous ces outils sont

disponibles gratuitement sur ericzimmerman.github.io et acceptent des exports CSV/JSON standardisés.

Le script **KAPE (Kroll Artifact Parser and Extractor)**, développé par Eric Zimmermann pour Kroll Consulting, est le framework d'automatisation forensique qui orchestre la collecte et l'analyse des artefacts Windows avec les Zimmermann Tools. KAPE utilise des fichiers de cibles (`.tkape`) pour définir quels artefacts collecter et des fichiers de modules (`.mkape`) pour définir quels outils d'analyse appliquer. La commande KAPE `kape.exe --tsource C: --target !SANS_Triage --tdest C:\Temp\Collection --mdest C:\Temp\Analysis --module !EZParser` collecte et analyse automatiquement tous les artefacts forensiques Windows essentiels en moins de 15 minutes sur un poste standard. KAPE est devenu l'outil de triage forensique Windows de référence pour les premières heures d'intervention, permettant aux équipes non spécialisées de collecter correctement les preuves avant l'arrivée d'un analyste forensique senior.

Forensique PowerShell : ScriptBlock Logging et transcriptions

PowerShell est le vecteur d'attaque dominant dans les incidents Windows modernes, utilisé pour tout : téléchargement de malwares depuis Internet (IEX), mouvement latéral (Invoke-Command, Enter-PSSession), manipulation du registre et des services, extraction de credentials (Invoke-Mimikatz), et persistance via des profils PowerShell modifiés. La journalisation forensique de PowerShell nécessite l'activation de trois fonctionnalités dans les GPO : **ScriptBlock Logging** (journalise le contenu complet de chaque bloc de script PowerShell exécuté), **Module Logging** (journalise les appels de modules et cmdlets), et **Transcription** (enregistre les sessions PowerShell complètes dans des fichiers texte). Ces journalisations génèrent des Event IDs 4103, 4104 et 800 dans `Microsoft-Windows-PowerShell/Operational.evtx` .

Sans ScriptBlock Logging activé, la forensique des commandes PowerShell est extrêmement limitée. Un attaquant utilisant PowerShell avec le flag `-EncodedCommand` (commandes encodées en Base64) ne laisse aucune trace textuelle lisible dans les logs si le ScriptBlock Logging n'est pas actif — la commande encodée est journalisée mais son contenu décodé reste inaccessible. L'activation du ScriptBlock Logging peut être rétroactive dans le sens où elle journalise immédiatement dès son activation, mais les sessions PowerShell passées ne sont pas

rétroactivement journalisées. C'est pourquoi l'activation du ScriptBlock Logging doit être prioritaire dans le durcissement des postes Windows pour la forensique et la détection, bien avant qu'un incident ne survienne. Un scénario fréquent en investigation forensique : l'analyste découvre que des commandes PowerShell malveillantes ont été exécutées (Amcache et Prefetch montrent l'exécution de powershell.exe), mais sans ScriptBlock Logging activé, il est impossible de déterminer quelles commandes ont été exécutées et donc quel payload a été téléchargé ou quelle persistence a été installée. C'est une impasse forensique qui force l'analyste à se rabattre sur des techniques indirectes comme l'analyse des connexions réseau DNS ou des artefacts de persistance dans le registre pour reconstituer partiellement les actions de l'attaquant. Notre service Microsoft 365 Security inclut la vérification et l'activation de ces paramétrages PowerShell sur l'ensemble du parc lors des audits de configuration Windows.

Forensique réseau Windows : artefacts de connexion et DNS

La forensique réseau sur un système Windows compromis s'appuie sur plusieurs sources d'artefacts. Le cache DNS (`ipconfig /displaydns` ou la commande PowerShell `Get-DnsClientCache`) conserve les résolutions DNS récentes du système, révélant les domaines contactés par les processus même si les connexions TCP correspondantes ont été fermées. Les entrées du cache ARP (`arp -a`) révèlent les systèmes avec lesquels le poste a communiqué récemment sur le réseau local. Le cache des partages réseau mappés (`net use`) révèle les partages réseau récemment accédés, et les entrées de la table de routage (`route print`) peuvent révéler des routes ajoutées par un malware pour rediriger le trafic vers une infrastructure C2.

Les logs du pare-feu Windows (`C:\Windows\System32\LogFiles\Firewall\pfirewall.log` si activé) et les journaux ETW (Event Tracing for Windows) du provider réseau Microsoft-Windows-Security-Auditing contiennent les connexions réseau avec PID du processus initiateur. L'Event ID 5156 (The Windows Filtering Platform has permitted a connection) dans Security.evtx journalise chaque connexion réseau autorisée avec l'application source et les adresses IP/ports source et destination — une mine d'informations pour reconstituer les communications réseau d'un malware. Pour les investigations nécessitant une analyse réseau approfondie, la technique MITRE ATT&CK T1049 (System Network Connections Discovery)

documente les commandes utilisées par les attaquants pour découvrir le réseau, et les mêmes commandes laissent des traces dans les Event Logs et Prefetch pour les investigateurs.

Timestomping et anti-forensique MACE : détection et contre-mesures

Le **timestomping** est une technique anti-forensique consistant à modifier les timestamps MACE (Modified, Accessed, Changed, Entry Modified) des fichiers malveillants pour les faire apparaître comme des fichiers système légitimes anciens, évitant ainsi qu'ils ressortent lors d'une recherche de fichiers récemment créés. Sur Windows, les timestamps sont stockés à deux endroits dans la MFT NTFS : les timestamps **$STANDARD_INFORMATION** (accessibles via `dir`, `ls`, `Get-Item`) et les timestamps **$FILE_NAME** (plus difficiles à modifier car nécessitant un accès raw au système de fichiers ou des outils spécialisés). Un écart entre les timestamps $STANDARD_INFORMATION et $FILE_NAME d'un fichier est un indicateur fort de timestomping.

La détection du timestomping s'effectue via l'analyse de la MFT avec **MFTECmd** : si le timestamp $STANDARD_INFORMATION d'un fichier est antérieur à son timestamp $FILE_NAME créé (ce qui est physiquement impossible sans manipulation), ou si le champ `ReferenceCount` dans la MFT indique des incohérences, le timestomping est confirmé. Les outils forensiques avancés comme **Plaso** extraient et comparent automatiquement ces deux ensembles de timestamps pour chaque fichier, permettant d'identifier les fichiers timestompés dans une timeline. Pour une investigation forensique destinée à un cadre judiciaire ou à une notification ANSSI dans le contexte NIS 2, la documentation du timestomping est essentielle car elle démontre l'intentionnalité de l'attaquant dans ses efforts de dissimulation, un élément important pour qualifier juridiquement l'acte malveillant. En pratique, les attaquants utilisent des outils comme *Metasploit timestomp*, des scripts PowerShell personnalisés, ou des techniques de manipulation directe de la MFT via des API Windows de bas niveau pour modifier les timestamps $STANDARD_INFORMATION. La corrélation entre les timestamps de la MFT, les entrées Prefetch correspondantes, et les Event Logs de création de fichier (Sysmon Event ID 11) permet de démontrer le timestomping même lorsque l'attaquant a modifié les deux ensembles de

timestamps avec des valeurs cohérentes entre elles mais incohérentes avec les autres sources forensiques.

KAPE et triage forensique Windows : collecte automatisée en 15 minutes

Le triage forensique Windows en urgence suit une séquence optimisée pour maximiser la collection d'artefacts volatils et non-volatils dans un minimum de temps. La première action est l'isolation réseau du système (débranchement câble réseau ou désactivation Wi-Fi) sans extinction, pour préserver la mémoire RAM et les processus actifs tout en coupant les communications C2. La deuxième action est la capture de la liste des processus, connexions réseau, et variables d'environnement via des commandes PowerShell exportées en fichier texte sur le support USB de triage. La troisième action est l'acquisition de la mémoire RAM avec WinPmem ou DumpIt, suivie du lancement de KAPE sur un support USB préparé à l'avance avec la cible `!SANS_Triage` qui collecte automatiquement Prefetch, Amcache, ShimCache, LNK files, Event Logs, ruches de registre, et tous les artefacts critiques en moins de 15 minutes sur un poste standard, permettant à l'analyste forensique senior d'analyser les preuves à distance sans nécessiter un déplacement immédiat sur site.

Le kit KAPE pré-configuré doit être préparé avant tout incident et stocké sur un support USB dédié aux interventions forensiques, avec les outils Zimmermann intégrés dans les modules KAPE pour une analyse immédiate sur site. La formation des équipes IT de première réponse à l'utilisation de KAPE permet de démarrer la collecte d'artefacts dans les premières heures d'un incident, même avant l'arrivée d'un spécialiste forensique. Notre service de RSSI externalisé inclut la préparation des kits de triage forensique Windows et la formation des équipes à leur utilisation pour les organisations françaises soumises à NIS 2 qui doivent être en mesure de documenter et de notifier les incidents dans les délais réglementaires imposés par la directive.

Artefact	Emplacement Windows	Outil Zimmermann	Information clé
Prefetch	C:\Windows\Prefetch*.pf	PECmd	Timestamps exécution (x8)
Amcache.hve	C:\Windows\AppCompat\Programs\	AmcacheParser	SHA1 des exécutables
ShimCache	SYSTEM hive (registre)	AppCompatCacheParser	Exécutables "vus" par Windows
LNK Files	%APPDATA%\Recent\	LECmd	Fichiers et partages accédés
$MFT / $UsnJrnl	C:\ (système de fichiers)	MFTECmd	Fichiers créés/supprimés
Event Logs	C:\Windows\System32\winevt\	EvtxECmd	Auth, services, PowerShell
SRUM	C:\Windows\System32\sru\	SrumECmd	Trafic réseau par processus
UserAssist	NTUSER.DAT (registre)	RECmd	Applications lancées GUI

Sysmon : journalisation avancée des événements Windows

Sysmon (System Monitor) est un service Windows gratuit de Microsoft Sysinternals qui s'intègre au démarrage de Windows et reste en mémoire à travers les redémarrages pour journaliser les activités système dans le journal des événements Windows. Contrairement aux Event Logs natifs qui journalisent peu de détails sur les processus, les connexions réseau, ou les modifications de fichiers, Sysmon fournit des informations enrichies : **Event ID 1** (création de processus avec ligne de commande complète, hash du processus, et ProcessGUID), **Event ID 3** (connexion réseau avec processus source, adresse destination, et port), **Event ID 7** (chargement de module DLL, utile pour détecter le DLL hijacking), **Event ID 11** (création de fichier avec chemin complet), et **Event ID 13** (modification de valeur de registre). La journalisation Sysmon est persistante à travers les redémarrages et disponible depuis n'importe quelle timeline d'investigation incluant les Event Logs, contrairement aux informations de processus volatiles disponibles uniquement en mémoire RAM.

La configuration de Sysmon s'effectue via un fichier de configuration XML qui définit quels événements journaliser et avec quels filtres, pour éviter la surcharge de logs. Les configurations de référence les plus utilisées sont celles de *SwiftOnSecurity* et de *Olaf Hartong* (Sysmon Modular), disponibles sur GitHub, qui offrent une couverture complète des techniques ATT&CK tout en filtrant les bruits de fond (processus Windows légitimes très fréquents). Pour les équipes DFIR françaises, Sysmon est devenu un prérequis pour la détection avancée des incidents Windows : sans Sysmon, les Event Logs natifs ne contiennent souvent pas assez de détails pour reconstruire une chaîne d'attaque complète, particulièrement pour les techniques LOLbins qui utilisent des processus Windows légitimes à des fins malveillantes. Notre service de mise en conformité NIS 2 recommande systématiquement le déploiement de Sysmon via GPO sur l'ensemble du parc Windows comme mesure prioritaire de préparation à la détection et à la réponse aux incidents, bien avant qu'un incident réel ne survienne dans l'organisation.

Corbeille Windows et Alternate Data Streams : artefacts cachés

La **Corbeille Windows** est une source forensique souvent négligée mais potentiellement très révélatrice. Lorsqu'un utilisateur ou un attaquant supprime un fichier via l'Explorateur Windows, Windows ne supprime pas immédiatement le fichier mais le déplace dans le répertoire caché `$Recycle.Bin` (Windows Vista+) où il est renommé avec un nom aléatoire. Le fichier `$I` contient le chemin d'origine du fichier supprimé, la taille du fichier, et la date et heure de suppression — des informations qui persistent même si le fichier `$R` correspondant a été définitivement supprimé depuis. L'outil **RBCmd** d'Eric Zimmermann analyse les fichiers `$I` de la Corbeille et exporte les métadonnées des fichiers supprimés en CSV, permettant de retrouver quels fichiers un attaquant a supprimé et à quel moment exact. Dans plusieurs incidents ransomware analysés par notre équipe, la Corbeille Windows a révélé que l'attaquant avait supprimé des archives ZIP contenant les données exfiltrées après leur transfert, fournissant une preuve directe de l'exfiltration même sans capturer le trafic réseau.

Les **Alternate Data Streams (ADS)** NTFS sont une fonctionnalité peu connue du système de fichiers Windows qui permet d'associer plusieurs flux de données à un même fichier. Le flux principal nommé `::$DATA` contient le contenu visible du fichier, tandis que des flux additionnels

peuvent être cachés en ajoutant `:nom_stream` après le nom du fichier. Les attaquants utilisent les ADS pour dissimuler des charges utiles ou des configurations malveillantes dans des fichiers légitimes, car les outils standards comme `dir` et l'Explorateur Windows n'affichent pas les ADS. L'ADS le plus forensiquement important est `Zone.Identifier` que Windows ajoute automatiquement à chaque fichier téléchargé depuis Internet, indiquant l'URL d'origine du téléchargement — une preuve que certains fichiers malveillants ont été téléchargés depuis Internet plutôt que copiés depuis un support local. La commande PowerShell `Get-Item * -Stream *` ou l'outil Streams de Sysinternals permettent de lister tous les ADS sur un volume, révélant les cachettes potentielles utilisées par les attaquants pour dissimuler des outils d'attaque ou des données sensibles exfiltrées.

Forensique mémoire Windows : Volatility 3 et acquisition RAM

La forensique mémoire Windows — l'analyse de la RAM d'un système compromis — est une discipline complémentaire à la forensique du système de fichiers car la mémoire contient des artefacts que les attaquants ne peuvent pas facilement effacer : les processus en cours d'exécution, les connexions réseau actives, les clés de chiffrement en mémoire, le code malveillant injecté dans des processus légitimes (process injection, reflective DLL loading), et les artefacts de malwares opérant uniquement en mémoire (fileless malware). L'acquisition de la mémoire Windows nécessite un outil d'acquisition live comme **WinPmem** (version open-source), **DumpIt**, ou **FTK Imager** qui créent une image brute de la RAM sans redémarrage du système. L'acquisition doit être effectuée le plus tôt possible dans l'intervention forensique, car la mémoire est volatile et son contenu est définitivement perdu lors d'un redémarrage, emportant avec elle toutes les preuves des processus malveillants actifs et des communications C2 en cours.

L'analyse de l'image mémoire Windows s'effectue principalement avec **Volatility 3** qui supporte les profils Windows 10, Windows 11, et Windows Server 2022 avec des plugins spécialisés : `windows.pslist` (liste des processus), `windows.cmdline` (lignes de commande des processus), `windows.netscan` (connexions réseau et sockets), `windows.malfind` (détection des injections de code en mémoire par recherche de permissions RWX dans des zones non-mappées à des fichiers PE), et `windows.dlllist` (DLLs chargées par chaque processus pour détecter le DLL hijacking).

Pour les incidents ransomware, Volatility peut révéler les processus de chiffrement actifs et parfois extraire les clés de chiffrement de la mémoire avant qu'elles ne soient détruites par le redémarrage du système imposé par le ransomware. Notre service de sécurité Windows avancée inclut des procédures documentées d'acquisition mémoire pour les systèmes Windows dans les scénarios d'incident les plus courants, avec des outils USB pré-configurés pour un déploiement immédiat lors d'un incident.

Plaso et super-timeline Windows : corrélation multi-artefacts

Plaso (log2timeline) supporte nativement les artefacts Windows via ses parsers spécialisés : parsers Prefetch, Amcache, ShimCache, LNK, Event Logs (EVTX), registre Windows (REG), et système de fichiers NTFS. La création d'une super-timeline Windows avec Plaso depuis une image disque produit une chronologie unifiée de tous les événements depuis toutes ces sources, permettant une analyse contextuelle qui révèle les relations causales entre les événements : par exemple, l'exécution d'un programme via Prefetch corrélée avec la création simultanée d'une entrée de service dans l'Event Log 7045 et d'une clé de Run dans le registre prouve l'installation d'un mécanisme de persistance à un moment précis.

La super-timeline Windows est l'outil analytique le plus puissant pour les investigations forensiques complexes car elle élimine la nécessité de basculer manuellement entre de nombreux outils et sources, réduisant le risque de rater des corrélations critiques entre des événements dans des sources différentes. L'analyste peut filtrer la timeline sur la fenêtre temporelle d'intérêt, chercher par nom de fichier ou nom de processus, et exporter les événements sélectionnés dans un rapport d'incident structuré. Dans les investigations NIS 2 qui doivent documenter l'impact de l'incident pour la notification à l'ANSSI, la super-timeline Plaso constitue une base documentaire objective et multi-sources difficile à contester. La commande de base pour créer une super-timeline Windows est `log2timeline.py --parsers win7,winxp,webhist,firefox,chrome timeline.plaso /chemin/image.raw` suivi de `psort.py -o l2tcsv -w timeline.csv timeline.plaso` pour exporter en CSV analysable dans un tableur ou dans Timesketch, le frontend web open-source développé par Google pour la visualisation collaborative des timelines forensiques. Notre service d'audit de sécurité Windows inclut une évaluation de la configuration

forensique de vos systèmes Windows et les recommandations pour activer les artefacts manquants (ScriptBlock Logging, Sysmon, auditpol) avant qu'un incident ne survienne dans votre organisation.

Questions fréquentes

Le Prefetch est-il disponible sur les serveurs Windows ?

Non, Prefetch est désactivé par défaut sur les versions Windows Server pour des raisons de performance disque. Il peut être activé manuellement en modifiant la valeur de registre `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher` à 3. Sur les postes de travail Windows 10 et 11, Prefetch est actif par défaut et constitute l'un des artefacts forensiques les plus fiables disponibles.

Combien de temps les entrées Amcache.hve sont-elles conservées ?

Amcache.hve conserve les entrées pour les programmes exécutés pendant environ 30 jours sur les systèmes Windows 10 et 11 par défaut. Les entrées les plus anciennes sont purgées lors de mises à jour Windows ou lors de la maintenance automatique de Windows. Contrairement à Prefetch qui conserve les 128 derniers programmes exécutés, Amcache a une limite temporelle plus qu'une limite par nombre d'entrées. Les Volume Shadow Copies peuvent préserver des versions antérieures d'Amcache.hve pour étendre la fenêtre d'investigation.

Le timestomping peut-il complètement tromper un analyste forensique expérimenté ?

Non si l'analyste compare les timestamps $STANDARD_INFORMATION et $FILE_NAME dans la MFT NTFS. Ces deux ensembles de timestamps sont modifiés indépendamment, et les outils courants de timestomping modifient uniquement $STANDARD_INFORMATION via les APIs Windows standard. Un écart entre les deux ensembles de timestamps d'un fichier est un

indicateur infaillible de timestomping, même si les deux timestamps sont dans le passé. De plus, les entrées dans Prefetch, Amcache, et les Event Logs fournissent des timestamps supplémentaires impossibles à tous modifier simultanément sans laisser des traces de la tentative de modification.

Comment activer ScriptBlock Logging PowerShell via GPO ?

Via la GPO : Computer Configuration → Administrative Templates → Windows Components → Windows PowerShell → Turn on PowerShell Script Block Logging → Enabled. La configuration recommandée inclut également "Log script block invocation start/stop events" et l'activation du Module Logging pour les modules PowerShell standards utilisés par les attaquants (ActiveDirectory, NetSecurity, etc.). Ces paramètres génèrent les Event IDs 4103 et 4104 dans le journal Microsoft-Windows-PowerShell/Operational.evtx de chaque poste.

Les Zimmermann Tools sont-ils utilisables dans un cadre judiciaire français ?

Oui, les Zimmermann Tools sont des outils forensiques reconnus et utilisés par des experts judiciaires en France et dans toute l'Europe. Leur utilisation dans le cadre d'une investigation judiciaire requiert la documentation de la version utilisée, des commandes exécutées, et des outputs générés, ainsi que la vérification de l'intégrité des sources analysées (hashes SHA256 avant et après analyse). Le respect du protocole de chaîne de custody et l'absence de modification des sources sont les conditions primordiales pour la recevabilité des conclusions en justice.

KAPE est-il gratuit pour les entreprises françaises ?

KAPE est disponible gratuitement pour les usages non commerciaux (équipes forensiques internes, institutions académiques). Pour une utilisation commerciale dans le cadre de prestations forensiques facturées à des clients, une licence commerciale est requise auprès de Kroll Consulting. Les outils Zimmermann (PECmd, AmcacheParser, etc.) que KAPE orchestre sont entièrement gratuits et open-source sans restriction d'usage commercial.

Accompagnement forensique Windows : expertise et formation DFIR

La forensique Windows nécessite une expertise technique spécialisée combinant la connaissance approfondie des internals Windows (NTFS, registre, artefacts forensiques spécifiques à chaque version de Windows), la maîtrise opérationnelle de la suite Zimmermann Tools et de Plaso, et l'expérience des procédures légales de préservation des preuves pour les dossiers judiciaires ou les notifications ANSSI NIS 2. Notre service de RSSI externalisé propose une capacité d'intervention forensique Windows avec SLAs définis et des équipes formées aux dernières techniques d'analyse. Pour une préparation proactive, nous proposons également des formations DFIR Windows pour les équipes IT, incluant l'utilisation de KAPE et des Zimmermann Tools, les procédures de triage de premier niveau, et la rédaction de rapports d'incident conformes NIS 2.

Contactez-nous via le diagnostic NIS 2 pour évaluer votre maturité forensique Windows actuelle et recevoir un plan d'amélioration adapté, incluant la configuration des artefacts manquants (ScriptBlock Logging, Sysmon, auditpol), la préparation des kits KAPE de triage, et la définition des procédures de réponse à incident Windows adaptées à votre organisation. Notre service de pentest Active Directory peut également réaliser des simulations d'incident Windows pour tester l'efficacité de votre collecte forensique en conditions réelles.

