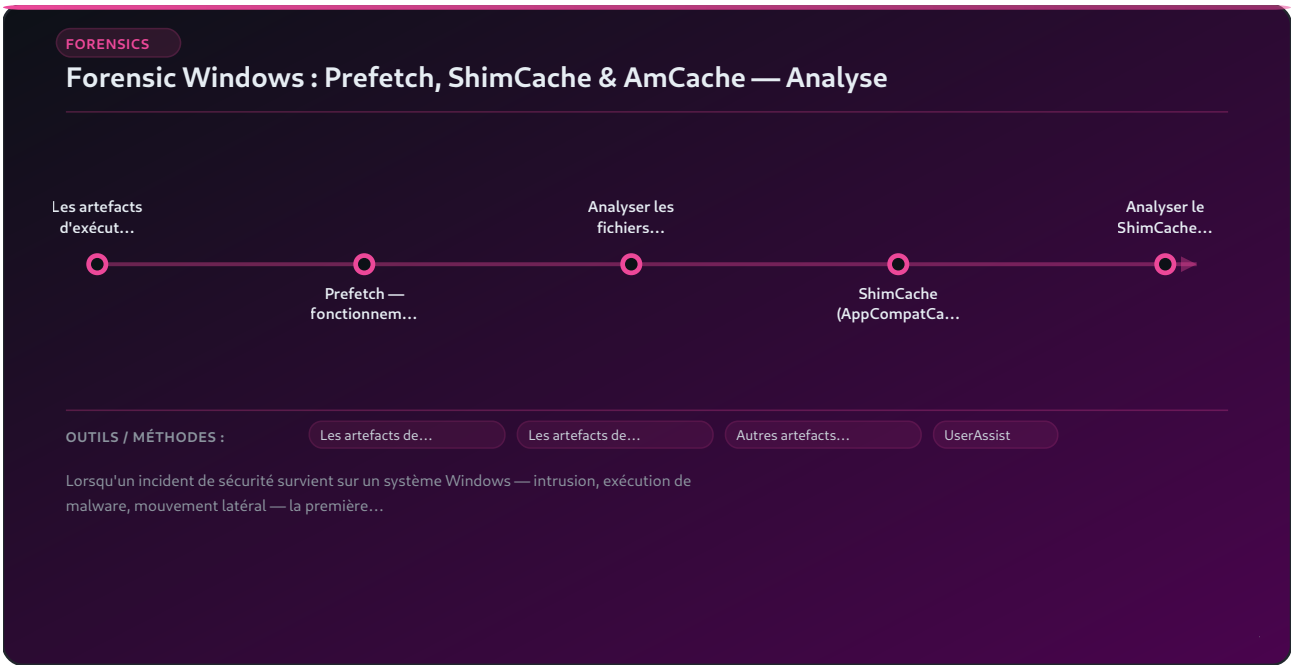
Forensic Windows : analyser les artefacts Prefetch, Shimcache et Amcache

> Guide d'investigation forensique Windows : analyser Prefetch (exécutions programmes), ShimCache et AmCache pour reconstituer une timeline d'attaque. Outils,...

Lorsqu'un incident de sécurité survient sur un système Windows — intrusion, exécution de malware, mouvement latéral — la première question de l'analyste forensique est toujours la même : "Qu'est-ce qui s'est passé, et dans quel ordre ?" Les artefacts Windows répondent à cette question. Le système d'exploitation Microsoft laisse en effet des traces minutieuses de chaque programme exécuté, chaque fichier ouvert, chaque accès réseau. Ces traces persistent après le redémarrage de la machine, après la suppression des fichiers malveillants, parfois même après une tentative de nettoyage par l'attaquant. Parmi tous les artefacts forensiques Windows, trois se distinguent par leur richesse et leur fiabilité pour reconstituer une timeline d'exécution : les fichiers Prefetch, le ShimCache et l'AmCache. Ces trois sources d'information complémentaires permettent de déterminer quels binaires ont été exécutés, à quelle heure, combien de fois, et depuis quel chemin. Ce guide vous présente en détail le fonctionnement de chacun de ces artefacts, les outils d'analyse recommandés — notamment la suite Eric Zimmerman Tools qui fait référence dans la communauté DFIR — et un cas pratique de reconstitution de timeline à partir d'une infection réelle. Vous découvrirez également comment corréler ces données avec d'autres artefacts Windows pour obtenir une image complète de l'attaque.



Les artefacts d'exécution Windows — vue d'ensemble

Windows maintient de nombreux registres d'activité pour des raisons de performance, de compatibilité applicative et de journalisation système. Du point de vue forensique, on distingue plusieurs catégories d'artefacts d'exécution.

Les artefacts de performance : Le cache Prefetch est conçu pour accélérer le démarrage des applications en préchargeant leurs données en mémoire. En conservant la liste des fichiers accédés par chaque programme, Windows crée involontairement un journal d'exécution précieux pour les analystes.

Les artefacts de compatibilité applicative : Le ShimCache (AppCompatCache) et l'AmCache sont deux bases de données maintenues par le sous-système de compatibilité applicative de Windows (Application Compatibility Infrastructure). Leur but originel est d'identifier les applications qui nécessitent des correctifs de compatibilité (shims) pour fonctionner sur des versions récentes de Windows. En pratique, ils enregistrent des informations sur presque tous les exécutables qui touchent le système.

Autres artefacts d'exécution importants :

UserAssist : dans le registre `HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`, enregistre les programmes lancés depuis l'interface graphique (ROT13 obfusqué).

RecentDocs / Jump Lists : fichiers récemment ouverts par l'utilisateur, utiles pour identifier les documents accédés par un attaquant.

SRUM (System Resource Usage Monitor) : base de données ESE qui enregistre l'utilisation des ressources par application (CPU, réseau, énergie) sur les 30 derniers jours.

BAM/DAM (Background Activity Monitor) : dans le registre depuis Windows 10, enregistre l'horodatage de la dernière exécution de chaque binaire.

Dans ce guide, nous nous concentrons sur les trois artefacts les plus riches pour la timeline d'exécution : Prefetch, ShimCache et AmCache.

Prefetch — fonctionnement et localisation

Le cache Prefetch de Windows est activé par défaut sur les systèmes client (Windows 10, Windows 11) mais désactivé par défaut sur les serveurs Windows (pour éviter l'overhead sur les SSD serveur). C'est un point critique à vérifier lors d'une investigation : si la machine analysée est un serveur, Prefetch est peut-être inactif.

> ⚡ **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.

Vérification de l'état du Prefetch :

```
reg query "HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParame
# 0 = désactivé, 1 = applications uniquement, 2 = boot uniquement, 3 = les deux (défaut client)
```

Localisation des fichiers Prefetch : `C:\Windows\Prefetch\`

Les fichiers Prefetch ont l'extension `.pf` et suivent une convention de nommage précise : `NOM_EXECUTABLE-HASH.pf`. Par exemple : `MIMIKATZ.EXE-AB12CD34.pf`. Le hash à 8 caractères hexadécimaux est calculé à partir du chemin complet de l'exécutable. Cela signifie que si le même binaire est exécuté depuis deux chemins différents (ex : `C:\Tools\nc.exe` et `C:\Temp\nc.exe`), deux fichiers Prefetch distincts seront créés — ce qui est précieux pour identifier les chemins d'exécution inhabituels.

Informations contenues dans un fichier .pf :

Nom de l'exécutable.

Chemin complet de l'exécutable.

Timestamp de la dernière exécution (jusqu'à 8 exécutions précédentes sur Windows 8+).

Nombre total d'exécutions (run count).

Liste des fichiers et répertoires accédés lors des premières secondes d'exécution (dépendances DLL, fichiers de configuration).

Numéro de version du format Prefetch (17 = XP/2003, 23 = Vista/7, 26 = Windows 8.1, 30 = Windows 10+).

Limite du Prefetch : Windows maintient un maximum de 128 fichiers Prefetch (Windows 7 et antérieur) ou 1024 fichiers (Windows 8+) dans le répertoire. Sur des systèmes très actifs, les fichiers les moins récents peuvent être écrasés.

Analyser les fichiers Prefetch avec PECmd

PECmd (Prefetch Explorer Command Line) est l'outil de référence pour analyser les fichiers Prefetch. Développé par Eric Zimmerman, il fait partie de la suite d'outils DFIR disponible gratuitement sur son site GitHub.

Téléchargement : Rendez-vous sur ericzimmerman.github.io pour télécharger la dernière version de PECmd.

Analyser un fichier Prefetch unique :

```
PECmd.exe -f "C:\Windows\Prefetch\MIMIKATZ.EXE-AB12CD34.pf"
```

Analyser tous les fichiers Prefetch d'un répertoire :

```
PECmd.exe -d "C:\Windows\Prefetch" --csv "C:\forensics\prefetch_results" --csvf prefetch.csv
```

L'option `--csv` exporte les résultats en CSV, facilement importable dans Excel ou dans un outil de timeline comme Timeline Explorer (également d'Eric Zimmerman).

Exemple de sortie PECmd pour un fichier suspect :

```
 Source file: C:\Windows\Prefetch\POWERSHELL.EXE-B4E28B8F.pf
Created on: 2024-03-15 02:17:43
Modified on: 2024-03-15 02:17:43
Last run: 2024-03-15 02:17:43
Other run times: 2024-03-15 01:45:12, 2024-03-14 23:31:05
Run count: 7
Executable name: POWERSHELL.EXE
Hash: B4E28B8F
Path: \WINDOWS\SYSTEM32\WINDOWSPOWERSHELL\V1.0\POWERSHELL.EXE
Files loaded (93 total):
...
\VOLUME{...}\USERS\COMPROMISED\APPDATA\ROAMING\SVCHOST.PS1
...
```

Dans cet exemple, la présence d'un script `SVCHOST.PS1` dans le profil utilisateur parmi les fichiers chargés par PowerShell est un indicateur fort de compromission. PECmd révèle non seulement l'exécution de PowerShell, mais aussi le nom du script malveillant.

Alternative graphique : WinPrefetchView (NirSoft) — outil GUI simple pour visualiser les fichiers Prefetch, moins complet que PECmd mais utile pour une première analyse rapide.

ShimCache (AppCompatCache) — registre et structure

Le ShimCache, aussi connu sous le nom d'AppCompatCache, est un artefact stocké dans la ruche de registre SYSTEM. Il enregistre les métadonnées des exécutables qui ont interagi avec le système de fichiers Windows, qu'ils aient été exécutés ou simplement "vus" par l'explorateur Windows.

Localisation dans le registre :

```
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache
Valeur: AppCompatCache (type REG_BINARY)
```

Le ShimCache est une structure binaire sérialisée dans cette valeur de registre. Sa structure varie selon la version de Windows :

Windows XP : 96 entrées maximum, format simple.

Windows Vista/7 : 1024 entrées, format enrichi avec taille de fichier et flag d'exécution.

Windows 8/8.1 : 1024 entrées, flag d'exécution supprimé (ShimCache ne distingue plus exécuté vs. vu).

Windows 10/11 : 1024 entrées, format similaire à Windows 8.

Informations contenues dans le ShimCache :

Chemin complet de l'exécutable.

Date de dernière modification du fichier (Modified Time du système de fichiers).

Taille du fichier (sur certaines versions Windows).

Flag d'exécution (sur Windows Vista/7 uniquement) : indique si le binaire a été effectivement exécuté ou seulement référencé.

Limitation critique du ShimCache depuis Windows 8 : Le ShimCache est mis à jour en mémoire en temps réel, mais n'est écrit sur disque que lors de l'arrêt propre du système. Si la

machine a été arrêtée brutalement (coupure de courant, crash), les dernières entrées ShimCache peuvent être perdues. De plus, depuis Windows 8, l'absence du flag d'exécution signifie qu'on ne peut pas certifier avec le seul ShimCache qu'un binaire a été exécuté — il peut avoir été simplement parcouru par l'explorateur.

Point d'intérêt forensique : Le ShimCache peut contenir des entrées pour des fichiers qui ont depuis été supprimés. Si un attaquant efface ses outils après utilisation, le ShimCache peut conserver la trace du chemin de ces outils.

Analyser le ShimCache avec AppCompatCacheParser

AppCompatCacheParser est l'outil d'Eric Zimmerman dédié à l'analyse du ShimCache. Il peut analyser le registre d'un système live ou une ruche SYSTEM extraite (pour l'analyse post-mortem).

Analyse du système live :

```
 # Nécessite des droits administrateur
AppCompatCacheParser.exe -t --csv "C:\forensics\" --csvf shimcache.csv
```

Analyse d'une ruche SYSTEM extraite :

```
 AppCompatCacheParser.exe -f "E:\Windows\System32\config\SYSTEM" --csv "C:\forensics\" --csvf shim
```

L'option `-t` trie les entrées par ordre chronologique. Le CSV de sortie contient les colonnes ControlSet, CacheEntryPosition, Path, LastModifiedTimeUTC et Executed (ce dernier uniquement pour Windows Vista/7).

Analyser le résultat : Dans le CSV, recherchez :

Des binaires dans des chemins inhabituels (`C:\Temp\` , `C:\Users\Public\` , `C:\ProgramData\`).

Des noms d'outils offensifs connus (mimikatz, procdump, psexec, netcat, cobalt strike stagers).

Des binaires système (`cmd.exe` , `powershell.exe` , `wscript.exe`) dans des chemins non standard.

Des binaires avec des noms imitant des processus système légitimes (svc-host.exe, svchest.exe).

```
# Exemple de filtrage PowerShell sur le CSV exporté
Import-Csv "C:\forensics\shimcache.csv" | Where-Object {$_.Path -match "Temp|Public|ProgramData"}
```

AmCache — hive et informations SHA1

L'AmCache est une base de données au format hive de registre (même format que les fichiers SYSTEM, SAM, SOFTWARE) stockée dans un fichier dédié. Elle contient des informations beaucoup plus riches que le ShimCache, notamment le hash SHA1 du binaire exécuté — une information de valeur inestimable pour l'investigation.

Localisation : `C:\Windows\AppCompat\Programs\Amcache.hve`

Le fichier Amcache.hve est verrouillé par le système en cours d'utilisation. Pour l'analyser sur un système live, vous devez en faire une copie shadow ou utiliser un outil capable d'accéder aux fichiers verrouillés.

Structure de l'AmCache : La base de données contient plusieurs clés principales :

`Root\File\{VolumeGUID}\{hash}` : entrées pour les fichiers exécutables. Chaque entrée contient le chemin complet, le hash SHA1, la date de compilation PE, la date de première exécution, la date d'installation.

`Root\Programs\{GUID}` : informations sur les applications installées (nom, éditeur, version, date d'installation).

`Root\InventoryApplication` et `Root\InventoryApplicationFile` : inventaire complet des applications (format plus récent, Windows 10+).

Informations clés pour l'investigation :

SHA1 du fichier : permet de comparer avec des bases de menaces (VirusTotal, NSRL). C'est l'information la plus précieuse — même si le fichier a été renommé ou déplacé, son hash SHA1 reste identique.

Date de première exécution (LinkDate) : horodatage de la première fois que le binaire a été exécuté sur ce système.

Date de compilation PE : date de compilation du binaire (peut être falsifiée par l'attaquant, mais reste un indicateur).

Éditeur (Publisher) : informations du certificat de signature ou des métadonnées PE.

Analyser l'AmCache avec AmcacheParser

AmcacheParser, toujours d'Eric Zimmerman, est l'outil de référence pour extraire et analyser le contenu de la base AmCache.

Copie du fichier verrouillé sur système live (avec RawCopy ou VSS) :

```
 # Via Volume Shadow Copy (si disponible)
vssadmin list shadows
# Copier depuis le shadow volume
copy "\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\Windows\AppCompat\Programs\Amcache.hve" C:\f
```

Analyse avec AmcacheParser :

```
 AmcacheParser.exe -f "C:\forensics\Amcache.hve" --csv "C:\forensics\" --csvf amcache.csv -i
```

L'option `-i` inclut les entrées du registre d'inventaire. La sortie génère plusieurs CSV selon le type d'entrée : UnassociatedFileEntries (fichiers non associés à une application), AssociatedFileEntries, ProgramEntries.

Vérifier les hashes SHA1 sur VirusTotal :

```
 # Exemple de script PowerShell pour vérifier les hashes
$apikey = "VOTRE_API_VT"
Import-Csv "C:\forensics\amcache_UnassociatedFileEntries.csv" | ForEach-Object {
    $sha1 = $_.SHA1
    if ($sha1 -ne "") {
        $result = Invoke-RestMethod "https://www.virustotal.com/vtapi/v2/file/report?apikey=$apike
        if ($result.positives -gt 0) {
            Write-Host "SUSPECT: $($_.FullPath) - $($result.positives)/$($result.total) détections
        }
    }
}
```

Cette approche permet d'identifier rapidement les binaires connus malveillants parmi les centaines d'entrées AmCache, sans analyser manuellement chaque fichier.

Reconstituer une timeline d'attaque — cas pratique

Imaginons un scénario réaliste : une alerte EDR signale une activité suspecte sur le poste d'un utilisateur commercial. L'équipe SOC suspecte une compromission. Voici comment les artefacts Prefetch, ShimCache et AmCache permettent de reconstituer la timeline.

Étape 1 — Collecte des artefacts :

```
 # Sur la machine compromise (avec droits admin)
PECmd.exe -d "C:\Windows\Prefetch" --csv "\\SRV-FORENSICS\cases\CASE001\prefetch" --csvf prefetch.
AppCompatCacheParser.exe -t --csv "\\SRV-FORENSICS\cases\CASE001\" --csvf shimcache.csv
# AmCache : via VSS ou après acquisition image disque
```

Étape 2 — Identification des binaires suspects dans le Prefetch : On découvre dans les fichiers Prefetch :

`POWERSHELL.EXE-B4E28B8F.pf` — exécuté à 02:17 (heure inhabituelle), chargeant un script depuis `C:\Users\user\AppData\Roaming\update.ps1`.

`MSHTA.EXE-C1D2E3F4.pf` — exécuté à 02:16, juste avant PowerShell.

`CERTUTIL.EXE-A1B2C3D4.pf` — exécuté à 02:18, outil utilisé pour télécharger des fichiers.

Étape 3 — Confirmation dans le ShimCache : Le ShimCache confirme la présence de `C:\Users\user\AppData\Roaming\svchost32.exe` — un binaire nommé pour imiter `svchost.exe` mais situé dans un chemin utilisateur.

Étape 4 — Validation via AmCache : L'AmCache contient une entrée pour `svchost32.exe` avec un hash SHA1. La vérification sur VirusTotal révèle que ce hash correspond à un implant Cobalt Strike connu, détecté par 45/71 antivirus.

Étape 5 — Reconstitution de la chaîne d'infection :

02:15 — L'utilisateur a cliqué sur un lien malveillant (email de phishing, identifié via les logs Exchange).

02:16 — `mshta.exe` exécute un HTA malveillant (dropper initial).

02:17 — PowerShell télécharge et exécute `update.ps1` (CertUtil comme alternative).

02:18 — `svchost32.exe` (Cobalt Strike) est déposé et exécuté.

02:20 — Début du mouvement latéral vers d'autres postes (corroboré par les logs réseau).

Corrélation avec d'autres artefacts Windows

Les artefacts Prefetch, ShimCache et AmCache sont plus puissants encore lorsqu'ils sont corrélés avec d'autres sources de données forensiques.

Event Logs Windows (Event ID 4688 — Process Creation) : Si l'audit des processus est activé (via GPO : Computer Configuration → Windows Settings → Security Settings → Advanced Audit Policy → Process Creation), l'Event ID 4688 enregistre chaque création de processus avec la ligne de commande complète. Cette information, combinée avec les timestamps Prefetch, permet de reconstituer la commande exacte exécutée.

```
Get-WinEvent -LogName Security -FilterXPath "*[System[EventID=4688] and EventData[Data[@Name='New
```

SRUM (System Resource Usage Monitor) : La base de données SRUM (`C:\Windows\System32\sru\SRUDB.dat`) conserve 30 jours d'historique d'utilisation des ressources par

application. Elle peut confirmer qu'une application a généré du trafic réseau (connexion C2) ou consommé beaucoup de CPU (chiffrement).

Fichiers LNK (raccourcis) : Les fichiers LNK créés automatiquement dans `%APPDATA%\Microsoft\Windows\Recent\` enregistrent les fichiers récemment ouverts avec les timestamps d'accès, de création et de modification de la cible.

$MFT (Master File Table NTFS) : L'analyse de la MFT avec MFTECmd (toujours Eric Zimmerman) permet de retrouver des fichiers supprimés et de vérifier les timestamps NTFS (créé, modifié, accédé, modifié dans la MFT — le "MACE").

Pour approfondir votre pratique forensique, consultez notre service d'investigation numérique et nos articles sur la threat intelligence. La gestion des incidents de sécurité est également couverte dans notre guide sur la réponse aux incidents. Pour les équipes SOC, notre article sur la détection des menaces avec un SIEM complète ce guide.

Les ressources incontournables pour la forensique Windows : la suite complète Eric Zimmerman Tools et les formations DFIR du SANS Institute.

Acquisition forensique légale et chaîne de custody

Dans le cadre d'une investigation susceptible d'aboutir à des poursuites judiciaires ou à un litige, la rigueur dans la collecte des preuves est primordiale. Une preuve mal collectée peut être inutilisable devant un tribunal.

Write blocker matériel : Pour l'acquisition d'un disque dur, utilisez toujours un write blocker matériel (Tableau, CRU WiebeTech) pour empêcher toute écriture sur le disque source. L'acquisition logicielle (sans write blocker) risque de modifier les timestamps d'accès et d'invalider les preuves.

Calcul de hash de l'image disque :

```
# Avec FTK Imager ou dcfldd
dcfldd if=/dev/sdb hash=sha256 hashwindow=1G hashlog=hash.log of=case001.dd
```

Le hash SHA256 de l'image disque originale et de toutes les copies doit être documenté et correspond à l'empreinte de l'intégrité de la preuve.

Documentation de la chaîne de custody : Chaque étape de manipulation de la preuve doit être documentée : qui a accédé à la preuve, quand, pourquoi, et quelles actions ont été effectuées. Cette documentation (Chain of Custody form) accompagne la preuve tout au long de la procédure.

Acquisition de la mémoire vive (RAM) : Pour les malwares qui n'existent qu'en mémoire (fileless malware), l'acquisition de la RAM est indispensable. Utilisez WinPmem ou DumpIt avant d'éteindre la machine. L'extinction de la machine sans acquisition RAM peut effacer des preuves irremplaçables.

Questions fréquentes sur la forensique Windows

Quelle est la différence entre ShimCache et AmCache pour la forensique ?

Le ShimCache (AppCompatCache) est stocké dans le registre SYSTEM et indique si un fichier a été "vu" par Windows, avec le chemin et la date de dernière modification du fichier. Il ne contient pas de hash SHA1. L'AmCache (Amcache.hve) est un fichier de base de données distinct, beaucoup plus riche : il contient le hash SHA1 de chaque binaire exécuté, la date de première exécution, les informations de signature numérique et les métadonnées PE. L'AmCache est plus fiable pour confirmer l'exécution, mais n'existe que depuis Windows 8. Pour les systèmes Windows 7, le ShimCache et le Prefetch restent les principales sources.

Comment analyser les artefacts Prefetch si le service est désactivé sur un serveur Windows ?

Sur les serveurs Windows, le Prefetch est désactivé par défaut. Pour les investigations sur serveur, concentrez-vous sur le ShimCache, l'AmCache, les Event Logs (4688, 4624, 4625), et si possible le SRUM. Vous pouvez aussi activer le Prefetch manuellement après l'incident pour les futures investigations : modifiez la clé `HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\EnablePrefetcher` à 3. Notez qu'activer le Prefetch sur un SSD rapide n'a plus l'impact de performance qu'il avait sur les disques magnétiques.

Un attaquant peut-il effacer les entrées Prefetch, ShimCache ou AmCache pour couvrir ses traces ?

Oui, c'est techniquement possible. Un attaquant disposant de droits administrateur peut supprimer les fichiers .pf du répertoire Prefetch, vider le ShimCache en modifiant la valeur de registre, ou supprimer/remplacer le fichier Amcache.hve. Cependant, ces actions de nettoyage laissent elles-mêmes des traces (événements de suppression dans la MFT, Event Logs) et sont détectables par un analyste attentif. De plus, beaucoup d'attaquants ne pensent pas à effacer ces artefacts. L'AmCache en particulier est souvent oublié car il est moins connu que le Prefetch.

Comment Timeline Explorer facilite-t-il la corrélation des artefacts forensiques ?

Timeline Explorer est un outil graphique d'Eric Zimmerman qui permet d'importer simultanément les CSV générés par PECmd, AppCompatCacheParser, AmcacheParser, et de nombreux autres outils de la suite. Il affiche toutes les entrées dans une vue chronologique unifiée, avec filtrage, coloration des lignes suspectes et recherche fulltext. C'est l'outil idéal pour reconstituer une timeline complète en corrélant toutes les sources forensiques. Il remplace avantageusement l'analyse manuelle dans Excel pour les cas complexes.

Quelle est la durée de rétention des données dans le ShimCache et l'AmCache ?

Le ShimCache maintient jusqu'à 1024 entrées (Windows 10+) en mémoire et les persiste lors de l'arrêt propre du système. Les entrées les plus anciennes sont écrasées par les nouvelles quand la limite est atteinte. L'AmCache n'a pas de limite documentée explicite, mais en pratique sur un

système actif depuis plusieurs années, il peut contenir des milliers d'entrées couvrant toute la vie du système. C'est d'ailleurs une de ses forces pour les investigations : il peut contenir des traces d'exécutables supprimés depuis longtemps, offrant une mémoire historique bien plus longue que le Prefetch.

À RETENIR

À retenir

Prefetch, ShimCache et AmCache sont trois artefacts complémentaires : Prefetch confirme l'exécution avec timestamps, ShimCache couvre tous les binaires "vus", AmCache fournit le hash SHA1 permettant la corrélation avec des bases de menaces.

PECmd, AppCompatCacheParser et AmcacheParser (suite Eric Zimmerman) sont les outils de référence, gratuits et régulièrement mis à jour.

Le ShimCache n'est écrit sur disque qu'à l'arrêt propre du système — les machines arrêtées brutalement peuvent avoir des données ShimCache incomplètes.

La corrélation avec Event ID 4688, SRUM et la MFT permet de reconstituer une timeline d'attaque complète et chronologiquement précise.

En contexte légal, toujours utiliser un write blocker matériel et documenter rigoureusement la chaîne de custody pour garantir la recevabilité des preuves.

Sources et références

MITRE ATT&CK — Techniques de collecte et d'investigation

ANSSI — Guide d'investigation numérique