

Investigation numérique

3. ANALYSE SYSTÈME

0. Introduction

Quelques points de précisions avant de commencer :

- L'analyse des systèmes est une **tâche complexe** et une méthodologie unique **ne peut s'appliquer à tous les systèmes**
- Chaque système d'exploitation possède ses spécificités et artefacts et il faut donc s'adapter à chaque fois pour en tirer le maximum d'informations
- Garder en tête que la plupart des artefacts utilisés en forensic n'ont pas pour but de faciliter une investigation mais servent d'autres buts pour le système ou l'administrateur de celui-ci. Il faut donc **interpréter les données avec vigilance et bien comprendre ce qu'elles indiquent** pour ne pas en déduire des informations erronées.

Sommaire

01

Analyse de journaux

4

02

Systèmes de fichiers

8

03

Artefacts propres à Windows

23

04

Comportements recherchés
lors d'une analyse de système

41

05

Outils d'analyse Windows

50

01

Analyse de journaux

Méthodologie générale

Par journaux, on entend ici les fichiers plats qu'on peut retrouver sur tous les systèmes :

- Equipements réseaux (proxy, pare-feu, routeurs, ...)
- Systèmes d'exploitations (audit.log, system.log, setupapi.dev.log, ...)
- Applications (journaux Apache, PHP, Oracle, ...)

Il apparait donc évident que tous auront des formats différents et contiendront des informations différentes.

Face à une telle diversité, il convient donc de s'adapter et de prendre quelques reflexes pour être certains de les interpréter correctement.

Méthodologie générale

Quelques étapes clefs :

1

Bien comprendre à quelle entité correspondent les champs présents dans les journaux selon la doc / configuration

```
207.241.237.225 - - [17/May/2015:10:05:58 +0000] "GET /blog/tags/examples HTTP/1.0" 200 9208
"http://www.semicomplete.com/blog/tags/" "Mozilla/5.0 (compatible; archive.org_bot
+http://www.archive.org/details/archive.org_bot)"
```



Les formats de logs peuvent parfois être modifiés par les administrateurs

2

Bien comprendre quelle information peut être déduite ou pas de chaque entité

```
RecordId,CreationDate,RecordType,Operation,UserId,AuditData,AssociatedAdminUnits,AssociatedAdminUnitsNames
53324088-7464-4a44-5d15-08dbaaf4fc65,9/1/2023 2:08:58 PM,1,New-InboxRule,redacted@evil-corp.com,{"CreationTime":"","2023-09-
01T14:08:58","Id":"","53324088-7464-4a44-5d15-08dbaaf4fc65","Operation":"","New-
InboxRule","OrganizationId":"","REDACTED","RecordType":1,"ResultStatus":"","True","UserKey":"","1003200124B033D8","UserType":2
,"Version":1,"Workload":"","Exchange","ClientIP":"","161.129.44.200:35425","[...]Parameters":[{"Name":"","AlwaysDeleteOutlookRul
esBlob","Value":"","False"},{"Name":"","Force","Value":"","False"},{"Name":"","MoveToFolder","Value":"","Detected
Items"},{"Name":"","Name","Value":"","."}, {"Name":"","SubjectContainsWords","Value":"","phishing"}, {"Name":"","MarkAsRead"
","Value":"","True"}, {"Name":"","StopProcessingRules","Value":"","True"}], "SessionId":"","939617ff-65f5-49b4-adeb-
b932d8904be5"},"
```

3

Interpréter les données à partir des informations obtenues

“

Le compte redacted@evil-corp.com a été utilisé pour créer une règle de redirection de mail à 2023-09-01T14:08:58 depuis l'adresse IP 161.129.44.200. Cette règle, nommée « . » permet de déplacer tous les mails ayant le mot « phishing » dans leur sujet dans le répertoire « Detected Items » et de les marquer comme lus.

”

A vous de jouer !

Fichier : `~/Forensics/Cours/1_Journaux/0_README.md`

02

Systèmes de fichiers

Systemes de fichiers

Très vaste sujet qui nécessiterait un cours à lui seul.



Dans le cadre de ce cours sur l'investigation numérique, on survolera les systèmes de fichiers les plus utilisés en se limitant à leur **intérêt forensic**.

Une définition rapide :

- Les systèmes de fichiers fournissent aux utilisateurs un moyen de **stocker et organiser les données** dans une **structure hiérarchique**.
- Les principes de fonctionnement des systèmes de fichiers sont indépendant des machines **qui les utilisent**.

Systemes de fichiers

Une notion importante est la notion de **métadonnée**.

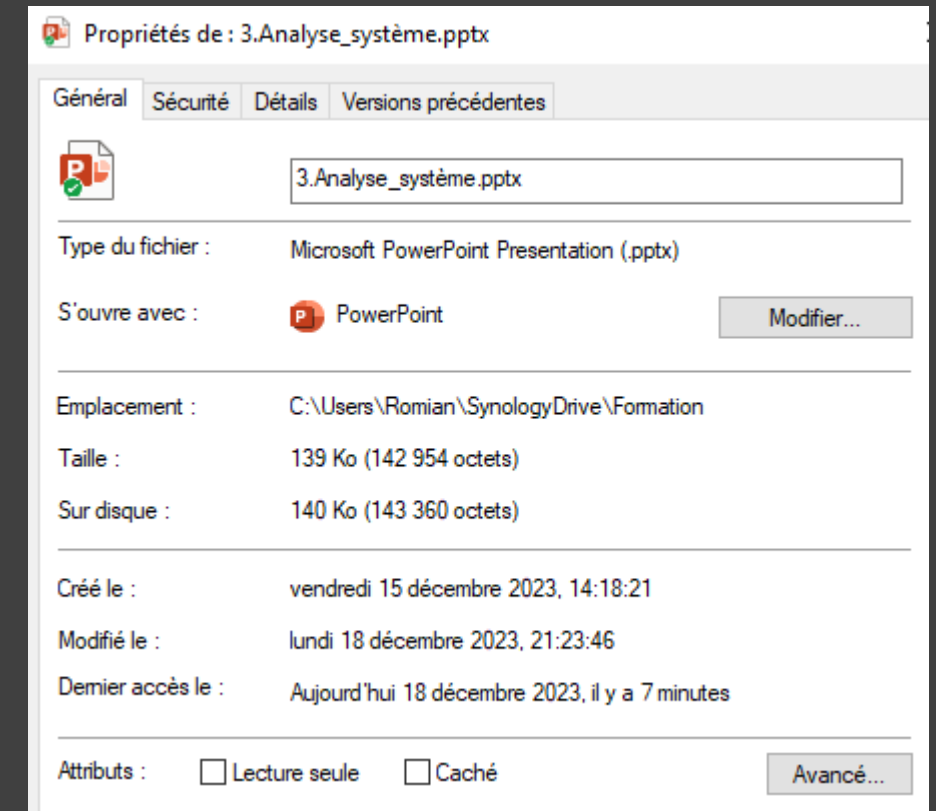
Les métadonnées sont l'ensemble des **informations portant sur les données elles-mêmes**. Par exemple :

- La taille des fichiers
- Les dates de dernier accès / modification / création
- Le chemin d'accès au fichier
- Les droits d'accès et informations de contrôle
- ...

Ce sont ces métadonnées qui vont être particulièrement utiles lors d'une investigation.



On parle ici de métadonnées du point de vue du système de fichier et non dans un sens général. Les métadonnées d'une image par exemple sont bien des données de ce point de vue là.



Systèmes de fichiers : FAT32, ext2 et ext3

On ne rentrera pas dans les spécificités de ces systèmes de fichiers ; on s'intéresse seulement aux **métadonnées qu'ils gardent en mémoire** (seulement les plus importantes pour nous).

FAT32:

Les métadonnées sont stockées dans les *Directory entries* :

- Nom court du fichier (une entrée spéciale est créée si le nom dépasse les 8 caractères)
- Attributs du fichier (RO, Caché, System, Directory, ...)
- Date de création au dixième de seconde
- Date de création à la seconde
- Jour de création
- Jour du dernier accès
- Date de dernière modification à la seconde
- Jour de dernière modification
- Taille du fichier

ext2 et ext3:

Les métadonnées sont stockées dans les *Directory entries* (D) ET dans les *inodes* (I) :

- Nom du fichier et valeur de l'*inode* (D)
- Type de fichier (D)
- Permissions et type de fichier (I)
- Date de dernier accès (I)
- Date de dernière modification (I)
- Date de dernière modification des métadonnées (I)
- Date de suppression (I)



Ext3 est un système de fichier journalisé, ce qui est intéressant d'un point de vue forensic mais ne sera pas abordé plus en détail dans le cours

Systèmes de fichiers : NTFS

Système par défaut sous Windows, c'est aussi l'un des plus intéressant pour nous puisqu'il s'agit d'un **système de fichiers journalisé** et que ses métadonnées sont simples à récupérer.

Une partition NTFS contient plusieurs fichiers spéciaux permettant de gérer efficacement le système de fichiers.

L'un de ces fichiers est particulièrement intéressant : le fichier spécial nommé **\$MFT** (pour *Master File Table*).

Cette table est une **suite d'entrée de taille fixe** (classiquement 1024 octets) contenant les métadonnées (et parfois les données) de **tous les fichiers** présents sur le système de fichier.

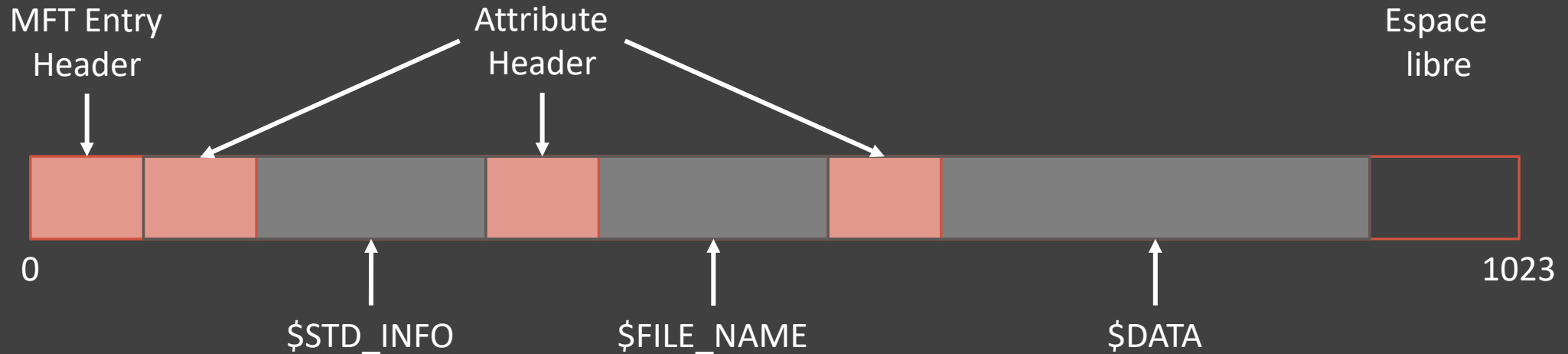
Il faut la voir comme un gros tableau avec une entrée par fichier. C'est un **index des fichiers du disque** qui permet au système de savoir à tout moment où trouver un fichier, connaître sa taille, etc.



L'**inode** de la \$MFT est toujours **0**. Sa position exacte sur le disque (le secteur sur lequel elle se trouve) est quant à elle définie par un autre fichier spécial: **\$Boot**.

Systèmes de fichiers : NTFS

Une entrée de la \$MFT est constituée de plusieurs attributs (jusqu'à 65 535 !), dont les trois plus fréquents sont les suivants :



- \$STD_INFO : Toujours présent, il contient les métadonnées les plus utiles au fonctionnement du système, notamment les timestamps affichés à l'utilisateur et les *flags* définissant le type de fichier
- \$FILE_NAME : Toujours présent, il contient d'autres données importantes dont le nom et la taille du fichier
- \$DATA : Peut être vide et n'a pas de structure définie. Dans certains cas, peut contenir les données du fichier



Il y a d'autres types d'attributs, et ils peuvent être stockés sur plusieurs entrées de la \$MFT si nécessaire.

Systèmes de fichiers : NTFS

Revenons dans un premier temps sur \$STD_INFO et \$FILE_NAME et sur les métadonnées utiles qu'ils contiennent :

Information	\$STD_INFO	\$FILE_NAME
Date de création (B)	Oui	Oui
Date de dernière modification (M)	Oui	Oui
Date de dernière modification de la \$MFT (C)	Oui	Oui
Date de dernier accès (A)	Oui	Oui
Flags (RO, Hidden, System, Archive, Device, ...)	Oui	Oui
Référence au répertoire parent	Non	Oui
Nom du fichier	Non	Oui
Taille du fichier	Non	Oui



Il est fréquent de désigner les *timestamps* d'un fichier par les lettres MACB. Attention à ne pas confondre *Modify* et *Change* ! Le premier porte sur les données, le deuxième sur les métadonnées.



On pourrait être tenté, avec ce tableau, de considérer l'attribut \$FILE_NAME comme une source unique d'informations et se passer de \$STD_INFO. On verra juste après qu'il n'en est rien !

Systèmes de fichiers : NTFS

ATTENTION : Les *timestamps* présents dans \$STD_INFO et \$FILE_NAME ne sont pas mis à jour de la même manière par Windows ! Les deux attributs sont donc primordiaux pour bien retracer l'histoire du fichier sur le système.

Windows® Time Rules								
\$ STANDARD_INFORMATION								
File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
Modified – Time of File Creation	Modified – No Change	Modified – Time of Data Modification	Modified – No Change	Modified – Inherited from Original	Modified – No Change	Modified – Inherited from Original	Modified – Inherited from Original	Modified – No Change
Access – Time of File Creation	Access – Time of Access (No Change only on NTFS Win7+)	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of File Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – Time of Data Modification	Metadata – Time of File Rename	Metadata – Time of File Copy	Metadata – Time of Local File Move	Metadata – Inherited from Original	Metadata – Inherited from Original	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of File Move via CLI	Creation – Inherited from Original	Creation – No Change
\$ FILENAME								
File Creation	File Access	File Modification	File Rename	File Copy	Local File Move	Volume File Move (move via CLI)	Volume File Move (cut/paste via Explorer)	File Deletion
Modified – Time of File Creation	Modified – No Change	Modified – No Change	Modified – No Change	Modified – Time of File Copy	Modified – No Change	Modified – Time of Move via CLI	Modified – Time of Cut/Paste	Modified – No Change
Access – Time of File Creation	Access – No Change	Access – No Change	Access – No Change	Access – Time of File Copy	Access – No Change	Access – Time of Move via CLI	Access – Time of Cut/Paste	Access – No Change
Metadata – Time of File Creation	Metadata – No Change	Metadata – No Change	Metadata – No Change	Metadata – Time of File Copy	Metadata – No Change	Metadata – Time of Move via CLI	Metadata – Time of Cut/Paste	Metadata – No Change
Creation – Time of File Creation	Creation – No Change	Creation – No Change	Creation – No Change	Creation – Time of File Copy	Creation – No Change	Creation – Time of Move via CLI	Creation – Time of Cut/Paste	Creation – No Change

Crédit : SANS Institute
(<https://www.sans.org/posters/windows-forensic-analysis/>)

Systèmes de fichiers : NTFS

Et évidemment, ça change à chaque version (et sous-version). Ici un comparatif Windows 10 21H2 / Windows 11 :

	\$STANDARD_INFORMATION															
	File Creation	File Access	File Modification	File Rename	File Copy (copy-paste) to same folder		File Copy (copy-paste) to new folder		Local File Move (cut-paste)	Volume File Move (CU)	Volume File Move (cut-paste)		File Recycled	File Deletion (shift delete)	Create ADS	Modify ADS
					Original file	New copy	Original file	New copy			Original file record	New file				
Last modified time	Time of File Creation	No Change	Time of Modification	No Change	No Change	Inherited from original	No Change	Inherited from original	No Change	Inherited from original	No Change	Inherited from original	No Change	No Change	Time of ADS creation	Time of ADS modification
Last access time	Time of File Creation	Win 10: Time of Access (No change on NTFS volumes if system volume > 128GB) Win 11: Time of Access	Time of Modification (approx. on Win 11)	Win 10: No change Win 11: Approx Time of Modification	Win 10: Time of File Copy Win 11: Approx time of File Copy	Time of File Copy (approx. on Win 11)	Win 10: Time of File Copy Win 11: Approx time of File Copy	Time of File Copy	Win 10: No change Win 11: Time of file move	Time of file move	Win 10: No change Win 11: Time of file move	Time of file move	No Change	No Change	Time of ADS creation	Time of ADS modification
Metadata time	Time of File Creation	No Change	Time of Modification	Time of Modification	No Change	Inherited from original	No Change	Win 10: Time of File Copy Win 11: Inherited from original	Time of file move	Win 10: Inherited from original Win 11: Time of file move	Win 10: No change Win 11: Time of file move	Win 10: Inherited from original Win 11: Time of file move	Time of recycle	No Change	Time of ADS creation	Time of ADS modification
Creation time	Time of File Creation	No Change	No Change	No Change	No Change	Time of File Copy	No Change	Time of File Copy	No Change	Time of file move	No Change	Inherited from original	No Change	No Change	No Change	No Change

Systemes de fichiers : NTFS

	\$STANDARD_INFORMATION															
	File Creation	File Access	File Modification	File Rename	File Copy (copy-paste) to same folder		File Copy (copy-paste) to new folder		Local File Move (cut-paste)	Volume File Move (CLI)	Volume File Move (cut-paste)		File Recycled	File Deletion (shift delete)	Create ADS	Modify ADS
					Original file	New copy	Original file	New copy			Original file record	New file				
Last modified time	Time of File Creation	No Change	Time of Modification	No Change	No Change	Inherited from original	No Change	Inherited from original	No Change	Inherited from original	No Change	Inherited from original	No Change	No Change	Time of ADS creation	Time of ADS modification
Last access time	Time of File Creation	Win 10: Time of Access (No change on NTFS volumes if system volume > 128GB) Win 11: Time of Access	Time of Modification (approx. on Win 11)	Win 10: No change Win 11: Approx Time of Modification	Win 10: Time of File Copy Win 11: Approx time of File Copy	Time of File Copy (approx. on Win 11)	Win 10: Time of File Copy Win 11: Approx time of File Copy	Time of File Copy	Win 10: No change Win 11: Time of file move	Time of file move	Win 10: No change Win 11: Time of file move	Time of file move	No Change	No Change	Time of ADS creation	Time of ADS modification
Metadata time	Time of File Creation	No Change	Time of Modification	Time of Modification	No Change	Inherited from original	No Change	Win 10: Time of File Copy Win 11: Inherited from original	Time of file move	Win 10: Inherited from original Win 11: Time of file move	Win 10: No change Win 11: Time of file move	Win 10: Inherited from original Win 11: Time of file move	Time of recycle	No Change	Time of ADS creation	Time of ADS modification
Creation time	Time of File Creation	No Change	No Change	No Change	No Change	Time of File Copy	No Change	Time of File Copy	No Change	Time of file move	No Change	Inherited from original	No Change	No Change	No Change	No Change

A vous de jouer !

Fichier : ~/Forensics/Cours/2_FileSystem/1_README.md

Systèmes de fichiers : NTFS

\$DATA :

On l'a compris, aucune structure définie n'existe pour cet attribut. Néanmoins, plusieurs éléments intéressants peuvent être notés :

- Il s'agit souvent du dernier attribut d'un fichier et il peut donc prendre tout l'espace restant sur les 1024 octets de l'entrée \$MFT. Ceci a pour conséquence concrète que si le fichier fait moins de 700 octets (environ), le système va s'épargner la peine de le stocker ailleurs et **tout le contenu sera présent dans l'attribut \$DATA**. Ceci implique qu'il est parfois possible de **recupérer le contenu d'un fichier directement depuis la \$MFT**, même si celui-ci a été supprimé (pour peu que l'entrée de la \$MFT n'est pas été réécrite).
- Un fichier peut avoir **plusieurs attributs \$DATA**. Les attributs \$DATA supplémentaires sont alors appelés des *Alternate Data Stream (ADS)* et sont alors nécessairement nommés.

Systemes de fichiers : NTFS

Example :

46	49	4C	45	30	00	03	00	8A	DC	6C	5D	1B	00	00	00	14	00	01	00	38	00	01	00	70	01	00	00	00	04	00	00	FILE0...	ŠÜ1]8...	p.....	
00	00	00	00	00	00	00	00	05	00	00	00	38	5F	12	00	07	00	00	00	00	00	00	00	10	00	00	00	60	00	00	008		
00	00	00	00	00	00	00	00	48	00	00	00	18	00	00	00	9A	F6	E0	97	35	34	DA	01	C7	68	3E	FD	35	34	DA	01H.....	šöà-54Ů.	Çh>ý54Ů.		
31	41	08	15	36	34	DA	01	56	A9	A2	15	36	34	DA	01	20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	1A..64Ů.V@ç.64Ů.		
00	00	00	00	4A	0C	00	00	00	00	00	00	00	00	00	00	E8	2C	2C	DE	04	00	00	00	30	00	00	00	70	00	00	00J.....	è,,ß....	0...p...		
00	00	00	00	00	00	04	00	56	00	00	00	18	00	01	00	32	5F	12	00	00	00	0E	00	9A	F6	E0	97	35	34	DA	01V.....	2_.....	šöà-54Ů.		
9A	F6	E0	97	35	34	DA	01	9A	F6	E0	97	35	34	DA	01	9A	F6	E0	97	35	34	DA	01	00	00	00	00	00	00	00	00	00	00	šöà-54Ů.šöà-54Ů.šöà-54Ů.....		
00	00	00	00	00	00	00	00	20	00	00	00	00	00	00	00	0A	03	73	00	63	00	72	00	69	00	70	00	74	00	2E	00s.c.r.i.p.t...			
62	00	61	00	74	00	00	00	80	00	00	00	60	00	00	00	00	00	18	00	00	00	01	00	45	00	00	00	18	00	00	00	b.a.t...	€....`.....	E.....		
6E	65	74	20	75	73	65	72	20	61	64	6D	69	6E	20	53	33	63	72	33	74	21	20	2F	61	64	64	0D	0A	6E	65	74	net user admin S3cr3t! /add.net				
20	6C	6F	63	61	6C	67	72	6F	75	70	20	61	64	6D	69	6E	69	73	74	72	61	74	6F	72	73	20	61	64	6D	69	6E	localgroup administrators admin				
20	2F	61	64	64	20	2F	61	FF	FF	FF	FF	82	79	47	11	FF	FF	FF	FF	82	79	47	11	20	00	64	00	6F	00	63	00	/add /aÿÿÿÿ,yG.ÿÿÿÿ,yG.		.d.o.c.		
75	00	6D	00	65	00	6E	00	74	00	20	00	74	00	65	00	78	00	74	00	65	00	2E	00	74	00	78	00	74	00	00	00	u.m.e.n.t.		.t.e.x.t.e...t.x.t...		
80	00	00	00	18	00	00	00	00	00	18	00	00	00	01	00	00	00	00	00	18	00	00	00	FF	FF	FF	FF	82	79	47	11	€.....		ÿÿÿÿ,yG.		
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	07	00

Systemes de fichiers : NTFS

Et avec un ADS :

```
>echo "I am inside an ADS 0.0'" > script.bat:HiddenPlace
```

[illegible]

```
FILE0...K9P^.....8....Ä.....  
.....8_.....`.....  
.....H.....šöà-54Ú.È•$f>4Ú.  
È•$f>4Ú.Æã.*>4Ú.  
.....J.....úOP.....0...p..  
.....V.....2_.....šöà-54Ú.  
šöà-54Ú.šöà-54Ú.šöà-54Ú.....  
.....s.c.r.i.p.t..  
b.a.t...€...`.....E.....  
net user admin S3cr3t! /add..net  
localgroup administrators admin  
/add /a€...P.....0...  
H.i.d.d.e.n.P.l.a.c.e..."I am in  
side an ADS O.O'" .....ÿÿÿÿ,yG.
```

A vous de jouer !

Fichier : ~/Forensics/Cours/2_FileSystem/2_README.md

03

Artefacts propres à Windows

Artefacts propres à Windows

Chaque système comporte des artefacts qui lui sont propres.

Il est toujours pertinent de prendre le temps d'appréhender les formats de fichiers particuliers associés à ces artefacts afin d'être en mesure de les manipuler.

Sous Linux, il est globalement possible de s'en sortir en analysant des fichiers journaux qui sont des fichiers plats. En revanche, ceci est beaucoup moins vrai sous Windows.

Nous n'allons encore une fois pas passer l'ensemble des types de fichiers utiles en revue mais nous attarder sur les deux plus importants :

Les bases de registre et les journaux d'événements (EVTX)



Certains autres formats seront rencontrés tout au long des TPs, mais nous n'allons pas nous arrêter sur chacun d'eux en détail.

Artefacts propres à Windows : Registre

La base de registre Windows est l'endroit où sont stockées **l'ensemble des paramètres enregistrant la configuration du système et des applications**.

Plusieurs bases de registre existent, appelées **ruches**, chacune stockant des paramètres relatifs à des domaines différents:

h	Description
HKEY_CLASSES_ROOT	Les entrées de Registre subordonnées à cette clé définissent des types (ou des classes) de documents et les propriétés associées à ces types.
HKEY_CURRENT_USER	Les entrées de Registre subordonnées à cette clé définissent les préférences de l'utilisateur actuel. Ces préférences incluent les paramètres des variables d'environnement, les données sur les groupes de programmes, les couleurs, les imprimantes, les connexions réseau et les préférences d'application.
HKEY_LOCAL_MACHINE	Les entrées de Registre subordonnées à cette clé définissent tous les paramètres propres à la machine, mais également aux applications si les paramètres sont applicables à l'ensemble des utilisateurs de la machine.
HKEY_USERS	Les entrées de Registre subordonnées à cette clé définissent la configuration utilisateur par défaut pour les nouveaux utilisateurs sur l'ordinateur local.
Amcache.hve	Stocke les métadonnées des applications installées et exécutées pour assurer la compatibilité des applications.

Artefacts propres à Windows : Registre

Dans la ruche HKEY_LOCAL_MACHINE (HKLM), on trouve notamment les quatre ruches suivantes :

Ruche	Description
HKLM\SAM	<i>Security Account Manager</i> : Contient des informations sur les utilisateurs locaux (<i>hashs</i> des mots de passe, date de dernière modification, ...)
HKLM\SECURITY	Contient des données relatives à la sécurité, notamment les droits associés aux comptes, les <i>group policies</i> , ...
HKLM\SOFTWARE	Contient toutes les informations relatives aux applications installées, notamment les chemins d'installation, la version, l'éditeur, ...
HKLM\SYSTEM	Contient des informations sur le matériel et sa configuration, les drivers installés, les services activés, ...

Artefacts propres à Windows : Registre

Plusieurs fichiers sur le système sont en charge de stocker les ruches du registre:

Ruche	Fichiers de stockage
HKEY_CURRENT_CONFIG	C:\Windows\System32\config\System, System.alt, System.log, System.sav
HKEY_CURRENT_USER	C:\Users\<USER>\Ntuser.dat, Ntuser.dat.log
HKEY_CURRENT_USER\Software\Classes\	C:\Users\<USER>\AppData\Local\Microsoft\Windows\UsrClass.dat, UsrClass.dat.log, UsrClass.dat.sav
HKEY_LOCAL_MACHINE\SAM	C:\Windows\System32\config\Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	C:\Windows\System32\config\Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	C:\Windows\System32\config\Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	C:\Windows\System32\config\System, System.alt, System.log, System.sav
HKEY_USERS\.DEFAULT	C:\Windows\System32\config\Default, Default.log, Default.sav
HKEY_CURRENT_CONFIG	C:\Windows\System32\config\System, System.alt, System.log, System.sav
Amcache	C:\Windows\AppCompat\Programs\Amcache.hve



Les chemins de stockage peuvent être modifiés et sont définis dans
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist

Artefacts propres à Windows : Registre

Explications des différents types de fichiers:

Extension	Description
aucune	Ruche elle-même, contenant toutes les données.
.alt	Copie de sauvegarde de la ruche critique HKEY_LOCAL_MACHINE\System. Seule cette ruche possède un fichier .alt.
.log	Journal des transactions des modifications apportées aux clefs et aux valeurs de la ruche. Les journaux de transactions font office de tampon lorsqu'il n'est pas possible d'écrire dans la base de registre immédiatement. Les modifications sont d'abord enregistrées dans les journaux de transaction avant d'être appliquées aux ruches.
.sav	Copie de sauvegarde d'une ruche.



Lors de l'analyse d'une base de registre, il faut donc penser à appliquer les journaux de transactions à la base afin de bien avoir la liste des dernières modifications à jour ! Par ailleurs, le journal de transactions peut parfois servir d'historique et contenir des données qui ne sont plus présentes dans la ruche registre.

Artefacts propres à Windows : Registre

La structure du registre est composée de **ruches**, de **clefs** et de **valeurs**, organisées en **arborescence**.
Chaque clef peut contenir **une ou plusieurs sous-clefs et valeurs** (ou zéro : une clef peut tout à fait être vide).

Registry Explorer v2.0.0.0

File Tools Options Bookmarks (29/0) View Help

Registry hives (2) Available bookmarks (63/0)

Enter text to search... Find

Key name	# values	# subkeys
C:\Users\Romian\ntuser.dat	=	=
ROOT	0	1
AppEvents	0	
Console	48	
Control Panel	1	1
...		
User Shell Folders	20	
UserAssist	0	
{9E04CAB2-CC14-11DF-BB8C-A2F1DED720...}	1	
{A3D53349-6E61-4557-8FC7-0028EDCEE8...}	1	
{B267E3AD-A825-4A09-82B9-EEC22AA3B8...}	1	
{BCB48336-4DDD-48FF-BB0B-D3190DACB3...}	1	
{CAA59E3C-4792-41A5-9909-6A6A8D3249...}	1	
{CEBFF5CD-ACE2-4F4F-9178-9926F41749...}	1	
Count	134	
{F2A1CB5A-E3CC-4A2E-AF9D-505A7009D...}	1	

Values UserAssist

Drag a column header here to group by that column

Value Name
HRZR_PGYPHNPbhag:pgbe
Zvpebfbsg.Trfgfnegrq_8jrxlo3q8oojr!Ncc
HRZR_PGYFRFFVBA
Zvpebfbsg.JvaqbjsrrqonpxUho_8jrxlo3q8oojr!Ncc
Zvpebfbsg.JvaqbjsZncf_8jrxlo3q8oojr!Ncc
Zvpebfbsg.Crbcy_8jrxlo3q8oojr!k4p7n3o7ql2188l46q4ln362l19np5n5805r5k
Zvpebfbsg.ZvpebfbsgFgvpxlAbgrf_8jrxlo3q8oojr!Ncc
{1NP14R77-02R7-4R5Q-O744-2RO1NR5198O7}\FavccvatGbby.rkr
Zvpebfbsg.JvaqbjsPnyphngbe_8jrxlo3q8oojr!Ncc
{1NP14R77-02R7-4R5Q-O744-2RO1NR5198O7}\zfcnvag.rkr
Zvpebfbsg.JvaqbjsFurryRkcrevrprUbfg_pj5a1u2gklrj!Ncc
Zvpebfbsg.JvaqbjsFrnepu_pj5a1u2gklrj!PbegnanHV
NQ2S1837.zlUC_j10m8iwnt6xr6!Ncc
Zvpebfbsg.JvaqbjsFgnegZrahRkcrevrprUbfg_pj5a1u2gklrj!Ncc
{6Q809377-6NS0-4440-8957-N3773S02200R}\JC\Fher Pyvpx\freiref\OeQrfgbcPbafor.rkr
ZFRntr

Type viewer Slack viewer

Current offset: 0 (0x0) Bytes selected: 0 (0x0)

Key: Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count

Selected hive: ntuser.dat Last write: 2023-12-24 15:56:29 134 of 134 values shown (100,00 %) Load complete

Artefacts propres à Windows : Registre

Un mot sur les valeurs, dont le type est nécessairement l'un des suivants :

Type	Description
REG_BINARY	Données binaires sans restriction de format.
REG_DWORD	Nombre de 32bits (existe aussi en _LITTLE_ENDIAN et _BIG_ENDIAN)
REG_QWORD	Nombre de 64 bits (existe aussi en _LITTLE_ENDIAN et _BIG_ENDIAN)
REG_SZ	Chaîne de caractères UNICODE ou ANSI.
REG_EXPAND_SZ	Chaîne Unicode ou ANSI contenant des références non développées à des variables d'environnement (par exemple '%PATH%').
REG_LINK	Chaîne Unicode contenant un lien symbolique vers une clef de Registre.
REG_MULTI_SZ	Séquence de chaîne de caractères, toutes terminée par '\0'. La fin de la sequence est terminée par '\0\0'.
REG_NONE	Aucun type défini.
REG_RESOURCE_LIST	Liste de ressources matérielles utilisées par un driver ou l'un des périphériques qu'il contrôle.
REG_RESOURCE_REQUIREMENTS_LIST	Liste de ressources matérielles possible que peut utiliser un driver ou l'un des périphériques qu'il contrôle.
REG_FULL_RESOURCE_DESCRIPTOR	Liste de ressources matérielles qu'un appareil physique utilise.

A vous de jouer !

Fichier : ~/Forensics/Cours/3_Registre/0_README.md

Artefacts propres à Windows : Journaux d'événements

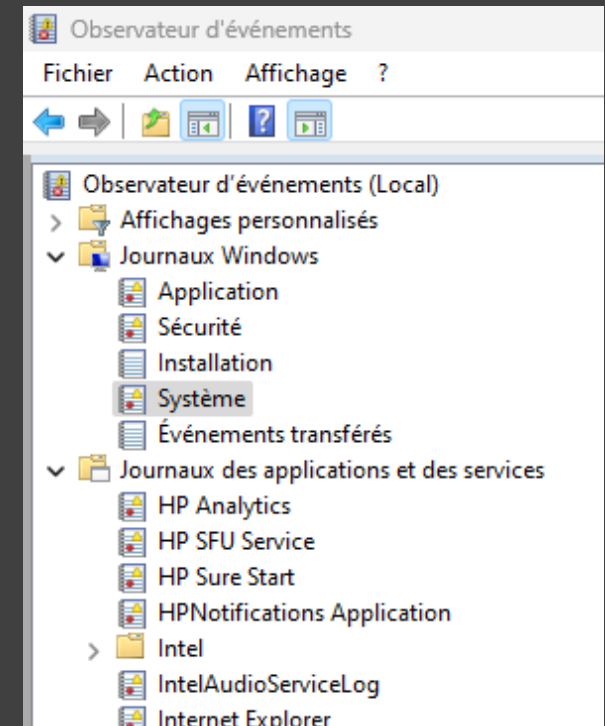
Les journaux d'événements sont un moyen mis à disposition par le système de **tracer les activités du système et des applications**. A la base, l'objectif était surtout de pouvoir retracer les éventuelles erreurs qui se produisent et d'en apprendre plus sur leurs causes, mais les journaux se sont petit à petit développés jusqu'à **enregistrer tout type d'activité**, permettant au final d'obtenir un **historique relativement précis de tout ce qu'il se passe** (ou presque) sur le système : authentications, installations d'application et de services, exécution de PowerShell, ...

Le format historique des journaux était le format EVT, remplacé depuis Vista par **EVTX** (on ne considèrera que ce dernier pendant ce cours).

Windows propose quatre journaux par défaut :

- Application : journal dédié aux applications et à leur fonctionnement
- Security : Événements de sécurité comme les authentications ou les accès aux fichiers
- Setup : Événements relatifs à l'installation de Windows
- System : Événements relatifs au système et à ses composants
- Forwarded Events : Événements transférés depuis d'autres machines du réseau

Les éditeurs d'application peuvent créer des journaux personnalisés s'ils le veulent pour enregistrer leurs propres événements et définir tous les paramètres du journal comme sa taille ou les droits d'accès.



Artefacts propres à Windows : Journaux d'événements

La liste des journaux ainsi que l'ensemble de leurs paramètres peuvent être consultés et modifiés dans le registre, à la clef `HKLM\CurrentControlSet\Services\EventLog\` :

Registry hives (3)Available bookmarks (61/0)

Enter text to search...Find

Key name	# values	# subkeys	Last write time
EventLog	16	18	2023-08-21
Application	8	208	2023-08-21
HardwareEvents	5	0	2022-05-01
HP Analytics	2	3	2023-08-21
HP SFU Service	0	2	2023-08-21
HP Sure Start	2	2	2023-08-21
HPNotifications Application	2	2	2023-08-21
IntelAudioServiceLog	2	2	2023-08-21
Internet Explorer	1	0	2023-08-21
Key Management Service	4	1	2022-05-01
McNeel	3	2	2023-08-21
OAlerts	5	1	2023-08-21
OneApp_IGCC	2	2	2023-08-21
Parameters	3	0	2022-05-01
Security	9	12	2023-08-21
State	2	0	2023-08-21
System	7	327	2023-12-01
Visual Studio	2	3	2023-08-21
Windows PowerShell	4	1	2022-05-01

Values

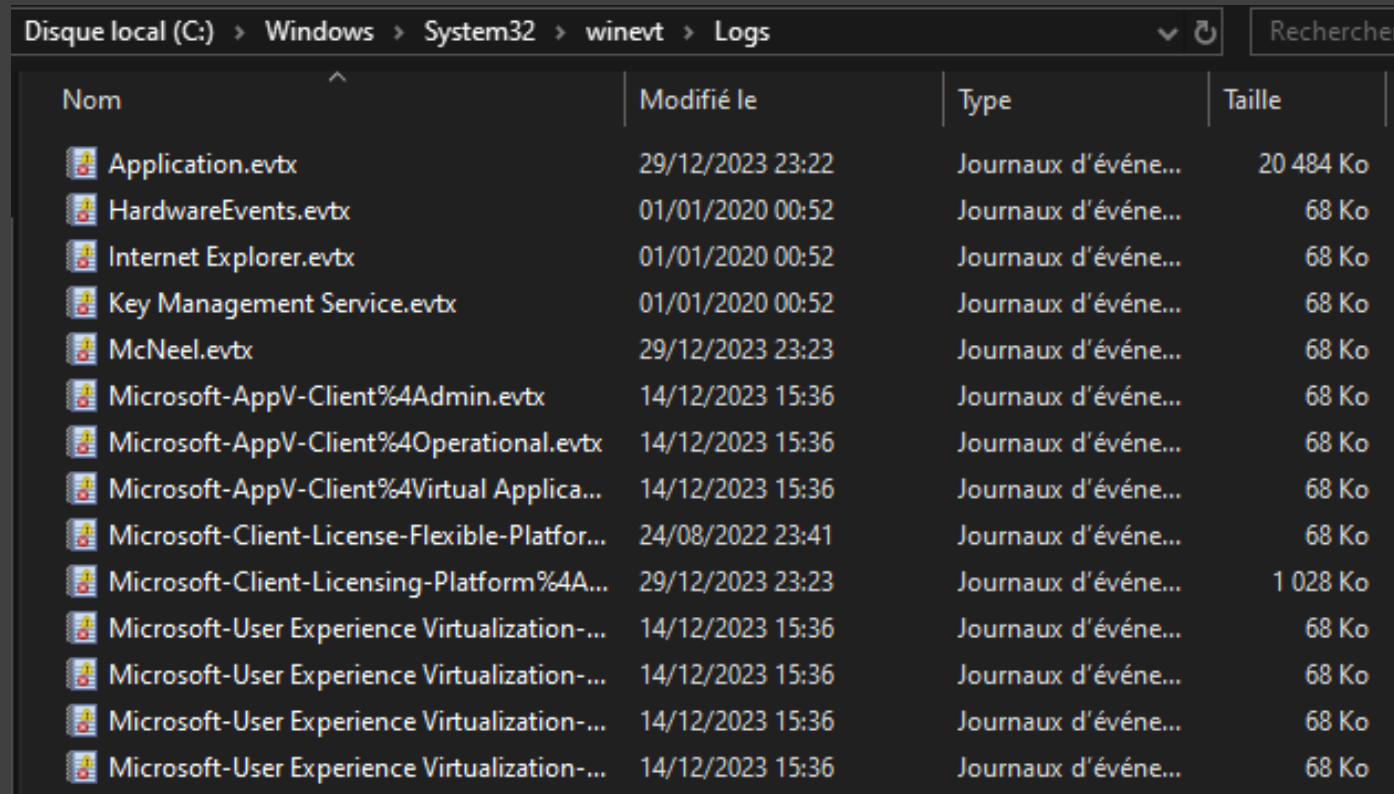
Drag a column header here to group by that column















Value Name	Value Type	Data
DisplayNameFile	RegExpandSz	%SystemRoot%\system32\wevt...
DisplayNameID	RegDword	256
File	RegExpandSz	%SystemRoot%\system32\wine...
MaxSize	RegDword	20971520
PrimaryModule	RegSz	Application
Retention	RegDword	0
AutoBackupLogFiles	RegDword	0
RestrictGuestAccess	RegDword	1

Type viewerBinary viewer

Artefacts propres à Windows : Journaux d'événements

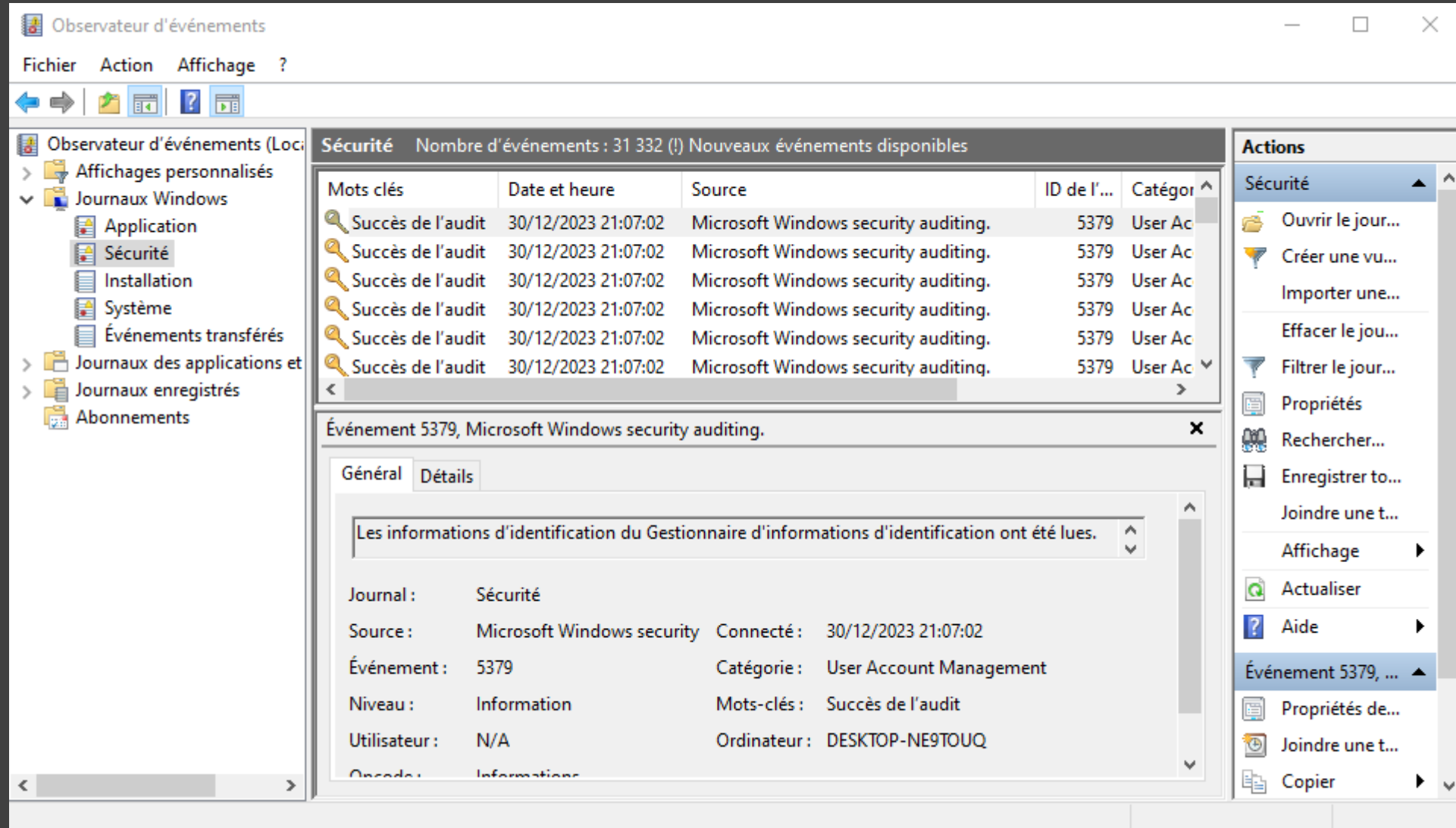
Le répertoire de stockage des fichiers .evtx est par défaut `C:\Windows\System32\winevt\Logs` :



Disque local (C:) > Windows > System32 > winevt > Logs					Rechercher
Nom	Modifié le	Type	Taille		
 Application.evtx	29/12/2023 23:22	Journaux d'événe...	20 484 Ko		
 HardwareEvents.evtx	01/01/2020 00:52	Journaux d'événe...	68 Ko		
 Internet Explorer.evtx	01/01/2020 00:52	Journaux d'événe...	68 Ko		
 Key Management Service.evtx	01/01/2020 00:52	Journaux d'événe...	68 Ko		
 McNeel.evtx	29/12/2023 23:23	Journaux d'événe...	68 Ko		
 Microsoft-AppV-Client%4Admin.evtx	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-AppV-Client%4Operational.evtx	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-AppV-Client%4Virtual Applica...	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-Client-License-Flexible-Platfor...	24/08/2022 23:41	Journaux d'événe...	68 Ko		
 Microsoft-Client-Licensing-Platform%4A...	29/12/2023 23:23	Journaux d'événe...	1 028 Ko		
 Microsoft-User Experience Virtualization-...	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-User Experience Virtualization-...	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-User Experience Virtualization-...	14/12/2023 15:36	Journaux d'événe...	68 Ko		
 Microsoft-User Experience Virtualization-...	14/12/2023 15:36	Journaux d'événe...	68 Ko		

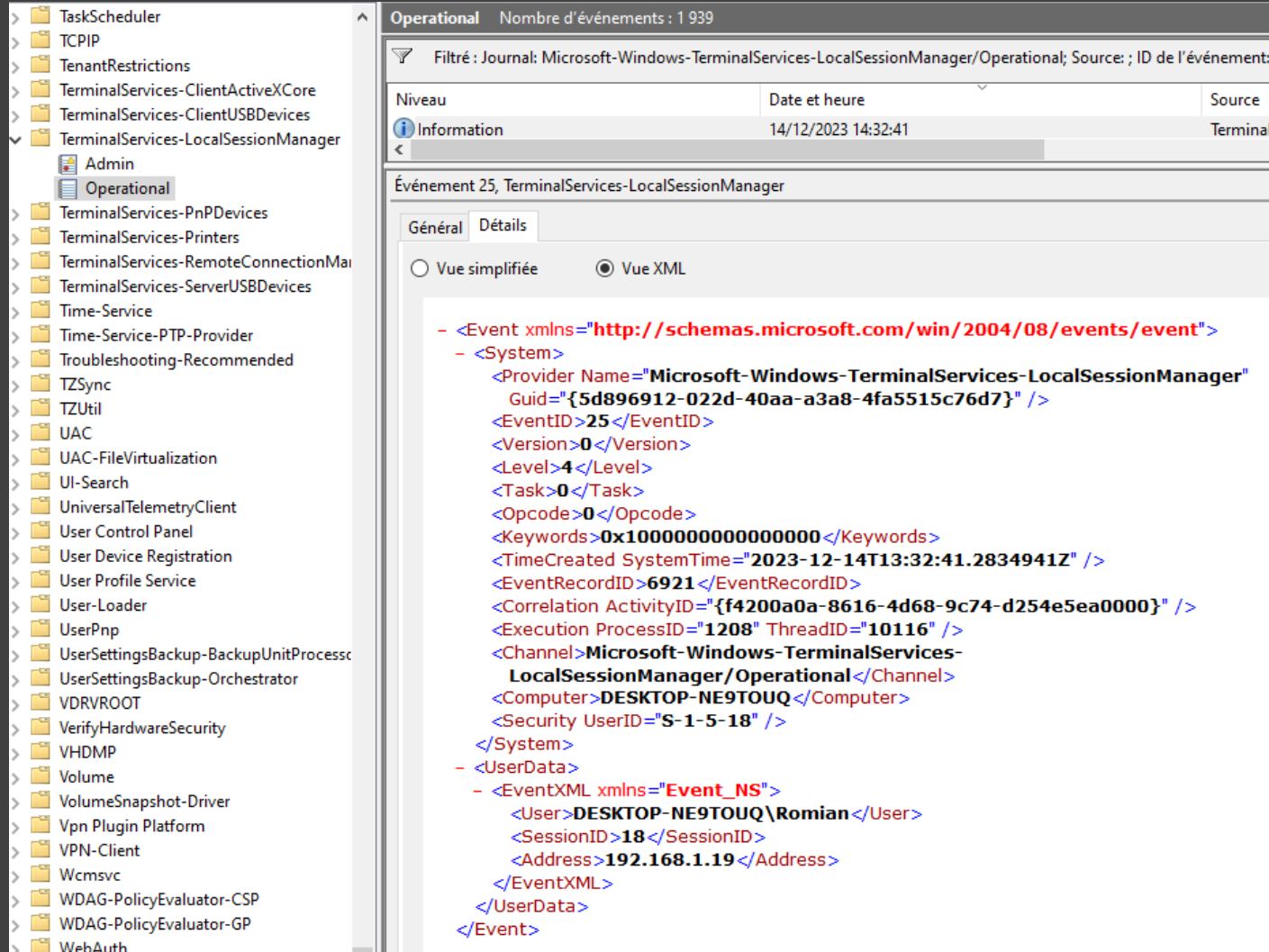
Artefacts propres à Windows : Journaux d'événements

Tous les journaux sont accessibles et peuvent être parcourus avec l'*EventViewer* (observateur d'événements) :



Artefacts propres à Windows : Journaux d'événements

Le format des événements est une sorte d'XML binaire grâce auquel sont renseignées toutes les informations utiles à l'interprétation de l'événement :



The screenshot shows the Windows Event Viewer interface. On the left, a tree view lists various system logs, with 'Operational' under 'TerminalServices-LocalSessionManager' selected. The main pane displays a table of events, with one event selected. Below the table, the 'Details' tab shows the XML structure of the event. The XML is a binary format with various fields like Provider Name, Guid, EventID, Version, Level, Task, Opcode, Keywords, TimeCreated, EventRecordID, Correlation ActivityID, Execution ProcessID, ThreadID, Channel, Computer, Security UserID, User, SessionID, and Address.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
  <Provider Name="Microsoft-Windows-TerminalServices-LocalSessionManager"
    Guid="{5d896912-022d-40aa-a3a8-4fa5515c76d7}" />
  <EventID>25</EventID>
  <Version>0</Version>
  <Level>4</Level>
  <Task>0</Task>
  <Opcode>0</Opcode>
  <Keywords>0x10000000000000000</Keywords>
  <TimeCreated SystemTime="2023-12-14T13:32:41.2834941Z" />
  <EventRecordID>6921</EventRecordID>
  <Correlation ActivityID="{f4200a0a-8616-4d68-9c74-d254e5ea0000}" />
  <Execution ProcessID="1208" ThreadID="10116" />
  <Channel>Microsoft-Windows-TerminalServices-LocalSessionManager/Operational</Channel>
  <Computer>DESKTOP-NE9TOUQ</Computer>
  <Security UserID="S-1-5-18" />
</System>
- <UserData>
  - <EventXML xmlns="Event_NS">
    <User>DESKTOP-NE9TOUQ\Romian</User>
    <SessionID>18</SessionID>
    <Address>192.168.1.19</Address>
  </EventXML>
</UserData>
</Event>
```

- Une partie des données est toujours présente et renseigne sur l'événement lui-même (balise **System**).
- Une autre partie contient les informations enregistrées par l'événement (balises **UserData** ou **EventData** dans la grande majorité des cas).



La spécification précise du format EVT_X n'a pas été documentée par Microsoft. Les chercheurs en sécurité ont donc dû la déduire à partir des exemples à dispositions afin de pouvoir proposer des outils de *parsing* des EVT_X.

Artefacts propres à Windows : Journaux d'événements

Éléments les plus importants de la balise *System* :

- **Provider Name** : il s'agit du fournisseur de l'événements, aussi appelé **Source**. C'est le journal auquel est associé l'événement. Il porte en général le nom de l'application ou du composant du système concerné.
- **EventID** : l'identifiant de l'événement. A chaque identifiant correspond une activité spécifique. C'est l'information qui, combinée à la source, permet d'identifier rapidement un événement et de savoir à quoi il correspond.
- **TimeCreated** : timestamp de création de l'événement (et donc à peu de choses près de l'activité enregistrée).
- **EventRecordId** : nombre incrémental identifiant l'événement au sein du journal consulté .
- **CorrelationActivityId** : GUID qui permet d'associer plusieurs événements entre eux.
- **Computer** : Hostname de la machine sur laquelle l'événement a été enregistré
- **UserID** : Utilisateur qui a réalisé l'action enregistrée

```
<System>
  <Provider Name="Microsoft-Windows-TerminalServices-LocalSessionManager"
    Guid="{5d896912-022d-40aa-a3a8-4fa5515c76d7}" />
  <EventID>25</EventID>
  <Version>0</Version>
  <Level>4</Level>
  <Task>0</Task>
  <Opcode>0</Opcode>
  <Keywords>0x1000000000000000</Keywords>
  <TimeCreated SystemTime="2023-12-14T13:33:10.9777788Z" />
  <EventRecordID>6927</EventRecordID>
  <Correlation ActivityID="{61a55000-55e5-1017-0000-000000000000}" />
  <Execution ProcessID="1208" ThreadID="2424" />
  <Channel>Microsoft-Windows-TerminalServices-
    LocalSessionManager/Operational</Channel>
  <Computer>DESKTOP-NE9TOUQ</Computer>
  <Security UserID="S-1-5-18" />
```

Artefacts propres à Windows : Journaux d'événements

Concernant **EventData** et **UserData**, les formats sont légèrement différents :

- **EventData** : Les données sont toutes contenues dans des balises Data avec un attribut Name précisant la donnée renseignée

```
<EventData>
  <Data Name="ServiceName">LGHUB Updater Service</Data>
  <Data Name="ImagePath">"C:\Program Files\LGHUB\lghub_updater.exe" --run-as-
    service</Data>
  <Data Name="ServiceType">service en mode utilisateur</Data>
  <Data Name="StartType">Démarrage automatique</Data>
  <Data Name="AccountName">LocalSystem</Data>
</EventData>
```

- **UserData** : Les données sont contenues dans des balises portant le nom de la donnée renseignée

```
- <UserData>
  - <EventXML xmlns="Event_NS">
    <User>DESKTOP-NE9TOUQ\Romian</User>
    <SessionID>18</SessionID>
    <Address>LOCAL</Address>
  </EventXML>
</UserData>
```



Il n'est pas toujours évident de savoir à quoi correspondent les données enregistrées. Certains événements enregistrent par exemple des données avec comme nom « Data » ou « param1 ».

A vous de jouer !

Fichier : ~/Forensics/Cours/4_EventLogs/0_README.md

Et si on analysait tout en même temps ?

Fichier : ~/Forensics/Cours/5_Timeline/0_README.md

04

Comportements recherchés
lors d'une analyse de système

Comportements recherchés lors d'une analyse de système

Les principaux comportements recherchés sont les suivants :

- Accès initiaux à l'infrastructure
- Etablissement de points de persistance
- Mouvements latéraux
- Exécution de programmes et de commandes
- Accès et manipulation de fichiers
- Navigation sur internet
- Utilisation de périphériques externes



Encore une fois, en fonction du contexte, il est tout à fait possible de rechercher d'autres types de comportements ou des comportements très précis qui correspondent aux besoins de l'investigation.

Les parties suivantes donnent **des exemples d'artefacts** ou d'éléments associés à ces comportements qu'il peut être utile d'analyser.

Si vous ne comprenez pas de quoi il s'agit, c'est normal ! Vous aurez l'occasion de vous familiariser avec certains de ces artefacts au cours des TP.



Les listes qui suivent ne sont pas exhaustives ! Elles reprennent les éléments les plus couramment utilisés.

Comportements recherchés lors d'une analyse de système

Accès initiaux à l'infrastructure :

On recherche ici tous les éléments permettant d'expliquer de quelle façon l'attaquant a pu obtenir un accès au SI. Il est ainsi pertinent de :

- Lister les moyens d'accès légitime au SI mis en place par la victime (passerelle VPN, Citrix, RDP ouvert, ...) et analyser les journaux des équipements associés
- Etudier la possibilité d'une exploitation sur une application ou un équipement exposé et analyser les journaux idoines pour vérifier si une exploitation a eu lieu
- Analyser les journaux des équipements réseau périmétriques à la recherche de toute connexion anormale (entrante ou sortante)
- Interroger les utilisateurs sur la réception d'e-mail suspects, de comportements inhabituels sur leurs postes, ou tout autre élément qui pourrait indiquer une compromission d'un poste
- En fonction du périmètre, envisager tous les scénarios possibles et analyser les éléments qui permettront de lever le doute sur chaque des hypothèses

Comportements recherchés lors d'une analyse de système

Etablissement de points de persistance :

Il est commun que les attaquants tentent de maintenir un accès distant au SI, et ce même après un redémarrage du système. Voici un certain nombre d'éléments qu'il peut être pertinent d'analyser afin d'identifier la mise en place de tels points de persistance :

- Journaux Windows : installation de **service**, de **tâche planifiée** et **d'application** d'accès distant
- Registre : **services**, **Run**, **RunOnce**, **BootExecute**, **ApplInit_DLLs**, **UserInit**, **Shell**, ...
- Répertoires **Startup**
- Fichiers de **tâches planifiées**
- Persistance **WMI**
- **MBR / UEFI**
- **DLL hijacking**
- Pour linux : **cron**, clef **SSH**, **systemd** et **timers**, modification des **.bashrc**, **.profile**, **syslog**...
- ...

Comportements recherchés lors d'une analyse de système

Mouvements latéraux :

Une fois présent sur l'infrastructure, les attaquants rebondissent en général sur d'autres serveurs du périmètre. Il est donc nécessaire de tracer ces activités afin de lister les serveurs touchés par l'attaque et éventuellement lancer des analyses de ces serveurs. Pour cela, il est possible de s'aider des éléments suivants :

- Journaux Security : **authentications** (4624, 4625) et **authentications explicites** (4648), événements **Kerberos**
- Journaux **RDP** (client et serveur)
- **User Access Logging**
- Créations de **service** (PsExec, Meterpreter, ...)
- **Exécution de binaire** type wmiexec, psexec, smbexec, ...
- Utilisation **WinRM**, **Remote Powershell**, **DCOM/RPC**, ...
- Linux : **audit.log**, **ssh.log**, **auth.log**, **syslog**, **last** / **who**, **utmp**, **wtmp**, **btmpt**
- ...

Comportements recherchés lors d'une analyse de système

Exécution de programmes et de commandes :

Les éléments suivants peuvent permettre de déterminer qu'un programme a été exécuté sur la machine :

- Journaux Windows des process ([Security – 4688, 4689](#))
- Journaux [Powershell \(400, 600, 4104\)](#) et fichiers [ConsoleHost_history.txt](#)
- Registre : [UserAssist](#), [AmCache](#), [MRU](#), [SRUM](#), [BAM](#), [RecentApps](#)...
- [AppCompatCache](#)
- Fichiers [JumpList](#) et [Recents](#)
- [Prefetch](#)
- Linux : [audit.log](#), [shell.log](#), [syslog](#), [history](#) et [.bash_history](#), derniers [exécutables accédés](#) (find), ...
- ...

Comportements recherchés lors d'une analyse de système

Accès et manipulation de fichiers :

La liste des fichiers créés, modifiés et accédés par un attaquant est une source souvent importante d'informations permettant de mieux comprendre son mode opératoire et ses objectifs. Les artefacts suivants peuvent permettre d'identifier ce type de comportement :

- \$MFT et \$USNJournal
- Répertoires parcourus : clefs ShellBags du registre
- Fichiers récemment consultés : Jumplist, Recents, clefs OpenSaveMRU et LastVisitedMRU, clefs RecentApps et RecentDocs
- \$RecycleBin et VSS pour identifier les fichiers supprimés (et éventuellement *carving* du disque)
- TypedURLs pour les fichiers / répertoires accédés directement depuis l'explorateur / le navigateur
- Linux : lister les fichiers et leurs MACB, lsof, ...
- ...

Comportements recherchés lors d'une analyse de système

Navigation sur internet :

Les navigateurs sont souvent utilisés par les attaquants pour télécharger des outils, exfiltrer des données ou parfois accéder à des fichiers ou des applications (vCenter par exemple). Dans d'autres cas, il est aussi possible d'identifier l'exploitation de vulnérabilité ou d'identifier un point d'entrée potentiel en analysant ce qui a été consulté / installé (extensions malveillante, téléchargement de cracks, ...). L'analyse des traces laissées par ces navigateurs est donc souvent une bonne source d'informations. Les principaux éléments à consulter sont les suivants :

- **Historique** de navigation et de téléchargement
- Fichiers placés en **cache** (peut notamment permettre de récupérer des fichiers qui auraient par ailleurs été supprimés : binaire, fichiers de configuration, etc)
- **Extensions** installées
- **Marque-pages** enregistrés
- **Cookies** stockés
- Fichiers de **restauration de session**
- ...



Chaque navigateur stocke ces informations à sa manière et selon des formats différents.

Comportements recherchés lors d'une analyse de système

Utilisation de périphériques externes :

Plus rare dans les cas de réponse à incident en entreprise, l'utilisation de périphérique externe reste néanmoins intéressante à étudier dans de nombreux autres cas, que ce soit pour expliquer l'origine de la compromission ou montrer que des données sensibles ont été volées. Les éléments suivants peuvent être récupérés sur les systèmes :

- Cles registre **USB** et **USBSTOR** : **identification** de **tous les périphériques** branchés (fabricant, numéro de série, capacité de stockage, ...), date de **première et dernière insertion** ainsi que la date du **dernier retrait** et l'**utilisateur** associé (ces données sont stockées dans les bases NTUSER)
- Fichier C:\Windows**setupapi.log** pour les heures de branchement des périphériques
- Evtx : **événement NTFS** enregistrant le montage des partitions
- Fichiers de raccourcis, ShellBags, et globalement **tous les artefacts liés aux fichiers** qui pointent vers un autre lecteur que C:
- Linux : **syslog**, **kern.log**, **messages** / **dmesg**
- ...

05

Outils d'analyse Windows

Outils d'analyse Windows



De très nombreux outils existent, avec chacun leurs avantages et inconvénients. La liste qui suit ne contient que quelques exemples qui suffisent pour le cours.

Parsing d'artefacts de manière unitaire:

- Suite d'outil d'Eric Zimmermann (Windows) :
 - <https://ericzimmerman.github.io/#!index.md>
 - Des parsers pour globalement tout ce dont vous aurez besoin, notamment : AmCache, Registre, LNK, Evtx, JumpList, ShellBags, SRUM, MFT, ...
- RegRipper (Windows et linux) :
 - <https://github.com/keydet89/RegRipper3.0>
 - Parsing de la base de registre
- Fred / registryspy, pour analyser le registre sous Linux
 - <https://www.penguin.lu/fred>
- EVTX-DUMP sous Linux et Windows :
 - <https://github.com/omerbenamram/evtx>
- Mft2bodyfile pour parser la \$MFT :
 - <https://github.com/janstarke/mft2bodyfile>
- USN-Journal-Parser :
 - <https://github.com/PoorBillionaire/USN-Journal-Parser>

Outils de parsing plus complets :

- Ces outils parsent de nombreux types d'artefacts et ont pour objectif de simplifier l'analyse en regroupant les informations. Attention cependant, ils sont nécessairement moins spécialisés et il n'est pas rare de trouver des erreurs de parsing ou des imprécisions.
 - Plaso (log2timeline) : outil en ligne de commande générant une timeline unique. Très puissant et complet, permet d'avoir une vision globale de tout ce qu'il s'est passé sur le système
 - Autopsy : programme en GUI qui parse aussi beaucoup d'artefacts et fonctionne avec des modules pour étendre ses fonctionnalités



Le bon outil est un outil fiable qui répond à votre besoin. L'idéal est de toujours bien comprendre comment ils fonctionnent, quelles sont leurs limites, et de les choisir en connaissance de cause.



Des questions ?