
WINDOWS MALWARE HUNTER HANDBOOK

Page de service

Référence : N/A

Plan de classement : Cybersécurité, Malwares, Microsoft Sysinternals suite

Niveau de confidentialité : Public

Mises à jour :

VERSION	DATE	AUTEUR	DESCRIPTION
1.0	21/01/2023	Thomas PRADEAU	Création du document

Validations :

VERSION	DATE	NOM	RÔLE
1.0	29/01/2023	Jérôme VALENTI	Enseignant

Diffusions :

VERSION	DATE	NOM	CADRE DE LA DIFFUSION
1.0	29/01/2023	Jérôme VALENTI	Remise pour correction

Sommaire

Page de service	1
Sommaire	2
Rappel du contexte	3
Objectifs	3
Les concepts fondamentaux	4
Le processeur et les processus	5
Le processeur	5
Les processus	5
Les threads	5
Priorité et affinité	6
Handle	7
La mémoire vive	8
La mémoire de masse	9
Outils d'analyse	10
Contexte	10
Registres Windows	10
Process Explorer	11
Détail des processus	12
Virus Total	13
Le mode d'exécution	14
Process Monitor	15
Détail des évènements	16
Les filtres	17
Cas 1 – Modification de la page d'accueil du navigateur	19
Contexte	19
But de l'attaque	19
Comment repérer la supercherie ?	19
Intercepter la modification	20
Création du piège	21
Conclusion	23
Cas 2 – DNS Empoisonné	24
Contexte	24
But de l'attaque	24
Comment repérer un DNS empoisonné ?	24
Identifier les paramètres du DNS	25
Capture des résultats sous Process Monitor	27
Conclusion	28
Table des illustrations	29

Rappel du contexte

NSI (Networking Solution Incorporated) est une entreprise de services du numérique (ESN)¹ qui se charge de la réalisation et de la maintenance des infrastructures matérielles et logicielles de ces clients.

Les ESN sont spécialisées dans les nouvelles technologies et englobent en général plusieurs métiers. Celles-ci ont comme objectif principal d'accompagner leurs clients dans la réalisation de leurs projets.

Objectifs

Le présent document a pour objectifs de former et introduire aux collaborateurs de l'entreprise NSI les différentes menaces opérantes sous le système d'exploitation **Windows 10**², ainsi que différents moyens de les détecter.

Windows 10 est présentement le système d'exploitation le plus utilisé aussi bien chez les particuliers que les professionnels, que ce soit sur les postes ou même sur des serveurs avec la famille à part entière Windows Server³.

Les notions de processus et comment un système d'exploitation organise les exécutions des programmes sur une machine seront également abordés afin de mieux comprendre comment il est possible de distinguer un processus malveillant des autres programmes. La très grande majorité des postes sont protégés derrière un ou plusieurs pare-feux, en plus d'un antivirus, qu'il soit intégré au système ou non. Or il s'avère que plus de 75 % des virus passent au travers de ces protections. Un travail d'analyse plus approfondie est donc nécessaire en plus de l'utilisation de ces outils.

¹ https://fr.wikipedia.org/wiki/Entreprise_de_services_du_num%C3%A9rique

² https://fr.wikipedia.org/wiki/Windows_10

³ https://fr.wikipedia.org/wiki/Windows_Server

Les concepts fondamentaux

Afin d'être en mesure de comprendre comment les programmes malveillants affectent les machines, il nous faut dans un premier temps comprendre les concepts fondamentaux, à savoir :

- Notion de processeur, de processus et thread
- La mémoire vive
- La mémoire de masse (mémoire morte)

Ces notions sont communes pour toutes les machines et systèmes d'exploitation. Bien que ceux-ci fonctionnent différemment en interne, ils gardent tous certains points communs. Tels que la gestion des processus et l'utilisation de mémoire vive pour le traitement et la mémoire de masse pour le stockage à plus ou moins long terme.

Le processeur et les processus

Le processeur

Commençons par quelques définitions, le **processeur** est le composant physique dans un ordinateur qui va effectuer différents traitements à une certaine fréquence, exprimé en hertz. Un processeur est constitué de cœurs physiques, ce sont ces cœurs qui effectuent les calculs et traitements.

La figure ci-dessous détaille l'anatomie d'un processeur, ainsi que ces différentes parties :

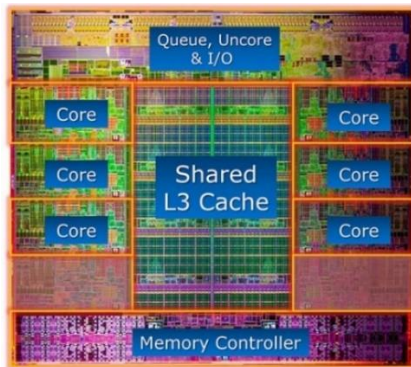


Figure 1 Anatomie du processeur

On peut constater les différents cœurs « **core** ». Mais également la mémoire de traitement interne au processeur, appelée mémoire cache « **Shared L3 Cache** ».

Le contrôleur mémoire « **memory controller** » est le composant qui assure la liaison entre le processeur et la mémoire vive.

Ainsi que le module de gestion des entrées sorties entre le processeur et le chipset, noter « **Queue, Uncore & I/O** ».

Le fonctionnement beaucoup plus détaillé est disponible sur l'article suivant <https://www.freecodecamp.org/news/how-does-a-cpu-work/>. Ce qu'il faut retenir principalement est que le processeur s'occupe de l'exécution des programmes sur la machine.

Les processus

Le **processus** est un programme en cours d'exécution. Un programme peut être composé de plusieurs processus. À ne pas confondre avec le thread, qui lui est une unité de base à laquelle le système d'exploitation alloue du temps processeur. Autrement dit, un processus est composé d'un ou plusieurs threads. Un thread est ensuite traité par le processeur pendant un certain temps, avant de passer à un autre thread.

Les threads

Le thread lui, ou fil d'exécution est un ensemble de code machine (Assembleur), qui ensuite exécuté par le processeur durant la période qui lui est accordée. Le thread est le fruit du code du programme (Java, Python, C++)⁴, qui a été compilé afin d'être compréhensible par le processeur.

En somme, l'exécution de code par le processeur peut être schématisée comme il suit :

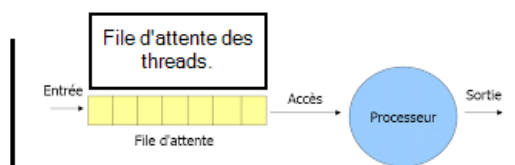


Figure 2 Attente et exécution des threads par le processeur

⁴ https://fr.wikipedia.org/wiki/Langage_de_programmation

Chaque thread est exécuté chacun son tour. L'ordre de passage et le temps d'exécution sont gérés par le système d'exploitation. Ce mécanisme est aussi dénommé « Ordonnanceur »⁵.

Priorité et affinité

L'ordonnanceur organise l'ordre d'exécution des threads en fonction de leurs affinités et de leurs priorités.

- L'affinité indique les cœurs physiques sur lesquels sont exécutés les threads. Par exemple, un thread possédant une affinité de 0-1 sera exécuté sur les cœurs 1 et 2 du processeur. Il est ainsi possible de modifier cette affinité pour par exemple réserver un ou plusieurs cœurs du processeur aux programmes les plus gourmands. Il est cependant recommandé de ne pas modifier cette affinité au risque de rendre le système plus lent ou instable.
- La priorité d'un thread influe sur son ordre de passage dans l'ordonnanceur. En effet, plus la priorité est haute, plus on accordera de temps un processeur à ce thread, et vice versa. Par défaut, les processus système, comme les **drivers**⁶, possèdent une priorité plus élevée que les programmes standards.

Ces paramètres sont facilement modifiables via le **gestionnaire des tâches** intégré à Windows. Dans celui-ci, sélectionnez l'onglet « détail », puis la liste détaillée des processus en cours d'exécution s'affiche.

La figure suivante montre la modification de l'affinité et la priorité d'un processus par le biais du gestionnaire des tâches :

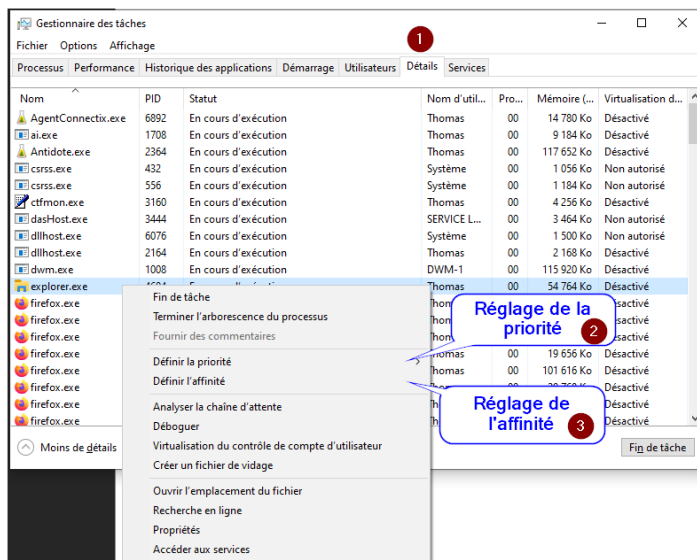


Figure 3 Définition de la priorité et de l'affinité d'un processus

⁵ [https://fr.wikipedia.org/wiki/Ordonnancement dans les syst%C3%A8mes d%27exploitation](https://fr.wikipedia.org/wiki/Ordonnancement_dans_les_syst%C3%A8mes_d%27exploitation)

⁶ <https://www.malekal.com/pilotes-drivers-windows/>

Handle

La dernière notion liée aux processus est les handle⁷. Un handle, qui peut se traduire par « poignée ». Un handle est une référence abstraite à une ressource mémoire ou un fichier.

Il s'agit d'un pointeur qui permet à un thread d'accéder rapidement et facilement à une portion de code contenu dans la mémoire de traitement.

Chaque thread crée un nombre variable de handles, il en crée autant que celui-ci en a besoin pour fonctionner.

Afin d'approfondir les connaissances sur les processus, threads et handles, il est recommandé de lire cette documentation technique fournie par Microsoft :

<https://learn.microsoft.com/fr-fr/windows/win32/procthread/processes-and-threads>

À retenir :

Le processeur exécute les instructions. Un programme en cours d'exécution est représenté par un ou plusieurs processus, qui eux-mêmes possèdent plusieurs threads. Un thread ou fil d'exécution contient le code compilé exécuté par le processeur. Les threads sont organisés et triés en fonction de leurs priorités et de leurs affinités. Cela est effectué par le système d'exploitation, plus précisément l'ordonnanceur.

⁷ <https://stackoverflow.com/questions/902967/what-is-a-windows-handle>

La mémoire vive

La **mémoire vive** ou plus couramment appelée « **RAM** », qui signifie « Random Access Memory », est un type de mémoire utilisé pour l'exécution des programmes. Cette mémoire est directement reliée au processeur et est beaucoup plus rapide en termes de bande passante et latence que la mémoire de stockage.

La RAM stocke les données utilisées par le processeur. À l'exécution d'un programme, les données du programme lui-même, ainsi que ses dépendances toutes deux stockées sur mémoire de masse (Disque dur), sont chargées en mémoire vive. Le code du programme est ainsi compilé puis exécuté par le processeur sous forme de threads comme expliqués précédemment.

Étant donné que la mémoire vive stocke les données utiles à l'exécution des programmes, il s'agit d'un endroit critique et est possiblement une porte d'entrée pour les programmes malveillants. En effet il serait théoriquement possible d'altérer cette mémoire pour produire des effets indésirables. Dans la pratique les programmes sont isolés dans la mémoire et ne peuvent pas modifier la mémoire en dehors de l'espace qui leur est alloué.

On se repère dans la mémoire à l'aide d'adresses, noté en hexadécimal (base 16). Ces adresses ressemblent à ceci :

➤ 0x00034ab1

Ce sont ces adresses qui permettent de communiquer avec la mémoire afin d'y stocker les données voulues.

Afin de faciliter l'accès aux données les plus utilisées par le processeur, on trouve également de la mémoire cache, présentée sur la [figure 1](#), qui est une mémoire encore plus rapide, mais en quantité limitée.

La figure suivante représente les interactions entre le processeur et la mémoire vive et la mémoire cache :

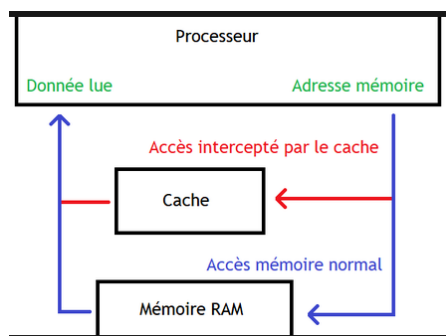


Figure 4 Interactions entre le processeur et les mémoires de traitement

À retenir :

La mémoire cache et la mémoire vive sont des mémoires très rapides et à faibles latences utilisées par le processeur pour l'exécution des programmes. Le processeur accède et interagit avec ces mémoires par le biais d'adresses notées en hexadécimal.

La mémoire de masse

La mémoire de masse, à l'inverse de la mémoire vive, sert à stocker des données de manière persistante. La mémoire de masse existe sous plusieurs formes, telles que le disque dur, le SSD ou le SSHD qui combine les deux premiers.

C'est bien en se penchant sur la mémoire de masse que l'on comprend l'intérêt des mémoires de traitements vues dans la [partie précédente](#). En effet, la mémoire de masse est beaucoup plus lente que la mémoire vive.

À titre de comparaison, le temps de latence moyen d'un disque dur est de 15 millisecondes, celui d'un SSD est de 0,2 milliseconde, et celui de la mémoire vive est de l'ordre de la nanoseconde, soit mille fois plus rapides⁸.

Le temps de latence représente le temps que met la mémoire pour répondre à une demande du système.

Alors la mémoire de masse ne sert qu'à retenir les données lorsque la machine n'est plus sous tension, ce qui n'est pas le cas de la RAM, puisqu'il s'agit de mémoire vive. A contrario de la mémoire de masse, aussi dénommée mémoire morte.

Lors de l'exécution d'un programme, toutes les données nécessaires à l'exécution sont copiées de la mémoire de stockage à la mémoire vive. C'est cette partie que l'on appelle le temps de chargement. L'exécution du programme se poursuit alors avec les concepts présentés dans la partie concernant [le processeur](#) et [la mémoire vive](#).

À retenir :

La mémoire de masse s'occupe de la rétention des données à court et long termes. Contrairement à la mémoire vive, la mémoire de masse est beaucoup plus lente, mais permet de retenir les données hors tension et accepte de plus grandes capacités de stockage.

⁸ [https://fr.wikipedia.org/wiki/Temps_de_r%C3%A9ponse_\(informatique\)](https://fr.wikipedia.org/wiki/Temps_de_r%C3%A9ponse_(informatique))

Outils d'analyse

Contexte

Après avoir fait un tour d'horizon des notions fondamentales nécessaires afin de comprendre la vie d'un programme, de son lancement à son arrêt. Nous allons à présent voir comment utiliser certains outils d'analyse afin de repérer des processus malveillants et ses comportements suspects, tels que l'accès à des clés de registre ou fichiers de configuration par exemple. Nous allons explorer les différentes possibilités disponibles avec les outils **ProcessMonitor** et **ProcessExplorer**.

Ces outils sont disponibles dans la suite **Sysinternals**, proposée par Microsoft à ce lien : <https://download.sysinternals.com/files/SysinternalsSuite.zip>

Bien qu'ils puissent sembler similaires, ces outils sont complémentaires.

Registres Windows

Les registres Windows permettent de stocker les paramètres de bas niveau du système. Les registres sont utilisés par le système et peuvent l'être par les applications, ceux-ci permettent en effet d'y stocker des valeurs comme la version ou la licence rattachée à un logiciel. Le registre permet d'enregistrer des paramètres sur la mémoire de masse.

Il s'agit donc d'un point critique qui peut être affecté par certains processus malveillants.

Ce registre s'organise en clés de registres, dans lesquelles l'on trouve les valeurs. Chacune de ces valeurs possède un type, tel que DWORD ou REG_SZ. Le type définit la valeur qui est stockée. La valeur quant à elle peut être un entier ou une chaîne de caractères.

Les clés de registres sont-elles organisées sous forme d'arborescence ? La figure ci-dessous montre à quoi ressemble une partie de la structure des registres sous Windows 11 :

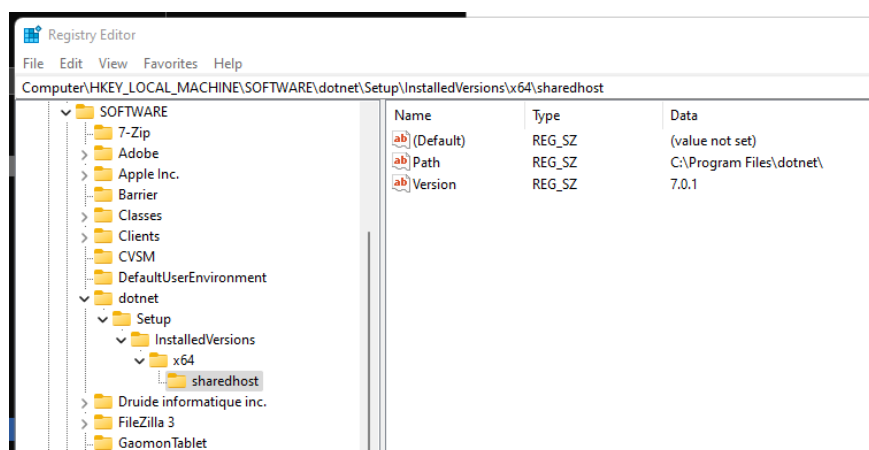


Figure 5 : Structure du registre sous Windows 11

Il est possible d'accéder et modifier ce registre à l'aide d'un outil graphique intégré à Windows, qui se nomme l'éditeur de registres (Regedit.exe).

Process Explorer

Process Explorer est un outil qui permet de lister tous les processus en cours d'exécution à un instant T sur le système.

La figure suivante montre la fenêtre principale de Process Explorer :

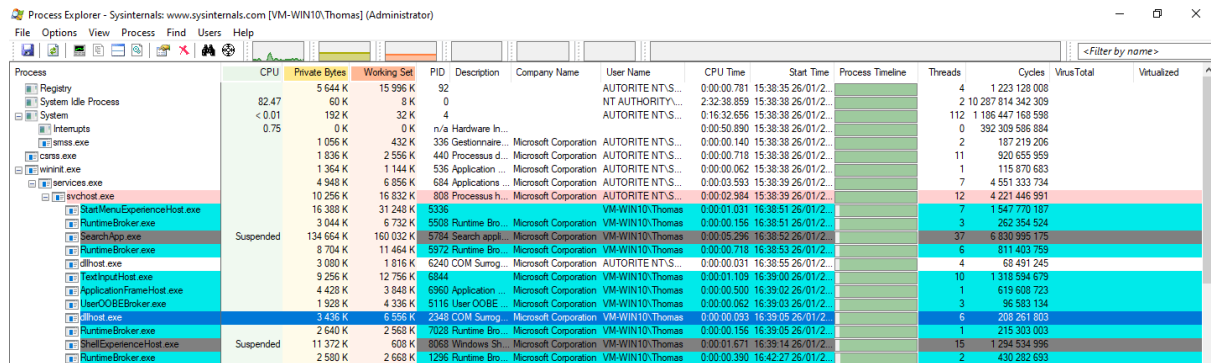


Figure 6 : Fenêtre principale de Process Explorer

Les processus en cours d'exécution sont présentés sous forme d'arborescence. On constate alors que de très nombreux processus sont en réalité des processus d'enfant, qui dépendent de leurs processus parents. On remarque quatre processus parents principaux :

- System
- Wininit.exe⁹
- Winlogon.exe¹⁰
- Explorer.exe

Explorer.exe est le processus qui permet la gestion du bureau et de l'interface utilisateur, il sert également de processus parent à de nombreux programmes, telle le montre la figure suivante :

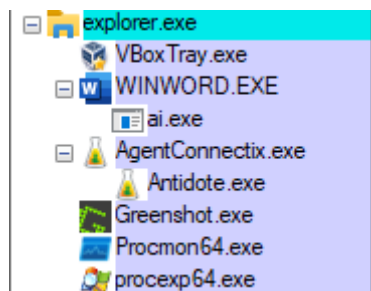
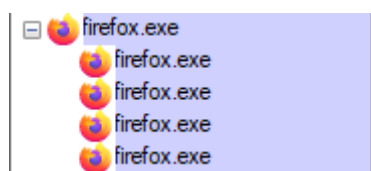


Figure 7 : Processus enfants d'Explorer.exe

Au contraire, d'autres processus comme firefox.exe sont eux-mêmes leurs processus parents, tel le montre la figure suivante :

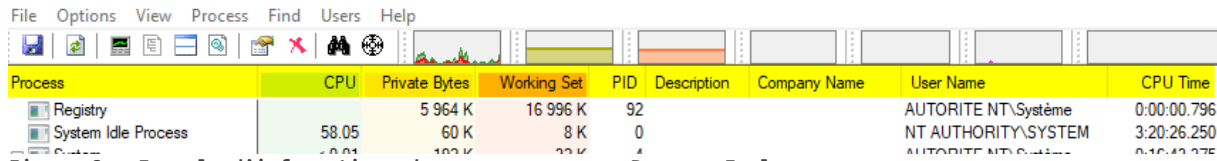


⁹ <https://social.technet.microsoft.com/Forums/ie/en-US/df6f5eeb-cbb9-404f-9414-320ea02b4a60/wininitexe-what-is-is-and-why-is-it-constantly-running?forum=win10itprosecurity>

¹⁰ <https://www.malekal.com/winlogon-exe/>

Détail des processus

Par rapport au gestionnaire des tâches inclus dans Windows, Process Explorer présente beaucoup plus de détails en rapport avec les processus. Ces informations sont organisées sous forme de colonnes, tel le montre la figure suivante :



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	User Name	CPU Time
Registry		5 964 K	16 996 K	92			AUTORITE NT\Système	0:00:00.796
System Idle Process	58.05	60 K	8 K	0			NT AUTHORITY\SYSTEM	3:20:26.250
System	0.04	100 K	32 K	4			AUTORITE NT\Système	0:00:00.000

Figure 8 : Exemple d'informations des processus sous Process Explorer

Cette figure n'est pas du tout exhaustive qu'en aux informations disponibles sur les processus, voici donc les principales :

Concernant les performances, la colonne « **CPU** » indique en pour cent l'utilisation du processeur, la colonne « **Private Bytes** » indique le nombre d'octets réservés en mémoire RAM avant l'exécution d'une application¹¹. La colonne « **Working Set** » indique en octets la taille de l'ensemble de travail, il s'agit des éléments en mémoire récemment utilisés par les threads du processus en question.

Concernant les informations en rapport avec la provenance du processus, la colonne « **PID** » indique l'identifiant du processus, soit un identifiant unique du processus. La colonne « **Description** » offre la description contenue dans la signature de l'exécutable à l'origine du processus. La colonne « **Company Name** » nous donne le nom de l'entreprise à l'origine de l'exécutable à l'origine du processus. Et enfin la colonne « **User Name** » indique le nom de l'utilisateur ainsi que son domaine¹² qui est à l'origine du processus.

Il est possible d'afficher ou non des colonnes supplémentaires un faisant un clic droit sur celles-ci, on clique ensuite sur « Select columns », la figure suivante montre les colonnes disponibles :

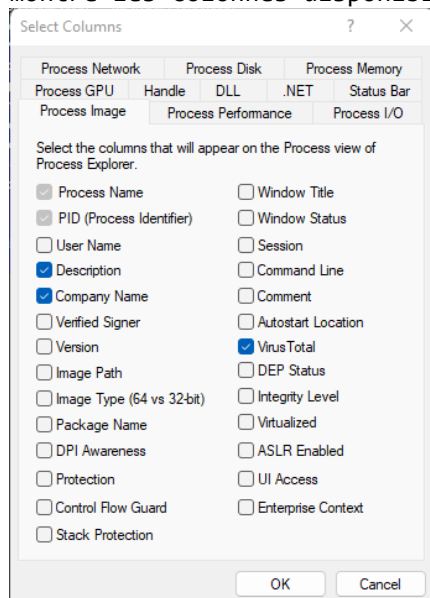


Figure 9 : Étendue des informations disponibles sous Process Explorer

Afin de comprendre l'étendue des possibilités offertes par cet outil, il est recommandé d'aller visionner cette vidéo :

https://www.youtube.com/watch?v=svLwLwB_How

Réalisée par **Mark Russinovich**, celle-ci rentre en profondeur dans les différentes fonctionnalités les plus poussées de Process Explorer.

¹¹ <https://stackoverflow.com/questions/1984186/what-is-private-bytes-virtual-bytes-working-set>

¹² <https://learn.microsoft.com/fr-fr/windows-server/identity/ad-fs/deployment/join-a-computer-to-a-domain>

Virus Total

Process Explorer permet de scanner certains fichiers à l'aide du service Virus total. Celui-ci scanne les fichiers avec plusieurs antivirus différents, afin de maximiser la détection de code malveillant.

Process Explorer possède un raccourci vers cet outil, qui est en temps normal, accessible directement par le biais d'un navigateur Web, à cette adresse :

<https://www.virustotal.com/gui/home/upload>

Sous Process Explorer, il suffit de faire un clic droit sur le processus que l'on souhaite scanner, puis dans le menu contextuel, cliquez sur « Check Virustotal.com », comme le montre la figure suivante :

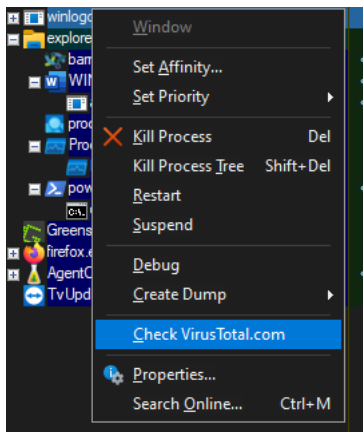


Figure 10 : Menu contextuel Virus total

Une fois scanné, un score sur 74 apparaîtra dans la colonne Virus Total. Ce score représente le nombre d'antivirus qui considèrent cet exécutable comme malveillant. Si la colonne Virus Total n'est pas présente, il suffit de l'afficher en répétant les opérations précédentes afin de modifier les colonnes affichées par défaut.

La figure ci-dessous montre un exemple de résultat Virus total :



Figure 11 : Exemple de résultat Virus total

Remarque :

Un score non nul ne signifie pas obligatoirement qu'un exécutable est dangereux pour la machine. Cela peut signifier que la somme de contrôle¹³ de l'exécutable n'est pas connue de Virus Total.

¹³ <https://academy.bit2me.com/fr/que-es-hash/>

Le mode d'exécution

Sur les postes exécutant le système d'exploitation Windows, il existe deux modes d'exécution.

- Le mode utilisateur
- Le mode noyau

Ces deux modes sont importants à comprendre puisque ceux-ci définissent les restrictions en rapport avec l'espace d'adressage d'une application. Dit plus simplement, le mode d'exécution vient modifier les autorisations qu'a un processus vis-à-vis de la mémoire de traitement, la RAM.

Tel qu'expliqué dans [la partie](#) sur la mémoire de traitement. La RAM contient toutes les données nécessaires aux processus en cours d'exécution pour fonctionner correctement. En outre, la mémoire vive contient aussi bien les données du système d'exploitation que des programmes ou des drivers.

Il est donc nécessaire d'introduire une isolation entre ces processus. Interviennent alors les modes d'exécution¹⁴. La différence entre ces deux modes est très simple.

- **Le mode utilisateur** concerne tous les programmes exécutés par l'utilisateur, ainsi que certains drivers. En mode utilisateur, l'espace d'adressage est propre à chaque application. Autrement dit, un processus en mode utilisateur ne peut pas aller modifier la mémoire allouée à un autre processus que lui-même. Ainsi, une application mal conçue ou qui serait prône à des plantages n'affecterait pas les autres applications ou le système.
- **Le mode noyau** est l'opposé du mode utilisateur. En mode noyau, les processus s'exécutent dans une plage de mémoire commune. Tous les processus qui s'exécutent en mode noyau sont autorisés à altérer les adresses mémoires qui font partie de la plage d'adresses allouée à ce mode d'exécution. Sans isolation, un processus pourrait affecter les données du système d'exploitation lui-même est causé des instabilités.

¹⁴ <https://learn.microsoft.com/fr-fr/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode>

Process Monitor

Là où Process Explorer ne fait que lister de manière détaillée les processus en cours d'exécution sur la machine. Process Monitor analyse le comportement de ces processus.

Si Process Explorer répond à la question « Qui ? », Process Monitor répond à la question « Fait quoi ? ».

Sur un ordinateur, le système d'exploitation fait le lien entre la couche applicative et la couche matérielle. Les applications effectuent donc des appels au système afin de lui demander des ressources, telles qu'un emplacement dans la mémoire vive, des fichiers sur le disque dur, ou l'accès à un périphérique tel qu'une imprimante par exemple.

Process Monitor permet donc de visualiser tous ces appels ! Il permet de voir tout ce que font les processus. Par exemple, quels fichiers ils modifient, quelles valeurs du registre ils lisent ou même les accès réseau des processus.

Il s'agit donc d'un outil très puissant dans le domaine de la cybersécurité. Il donne accès à tout un panel d'informations qui permettent de déterminer si un processus est malveillant ou non.

La figure ci-dessous montre la fenêtre principale de Process Monitor :

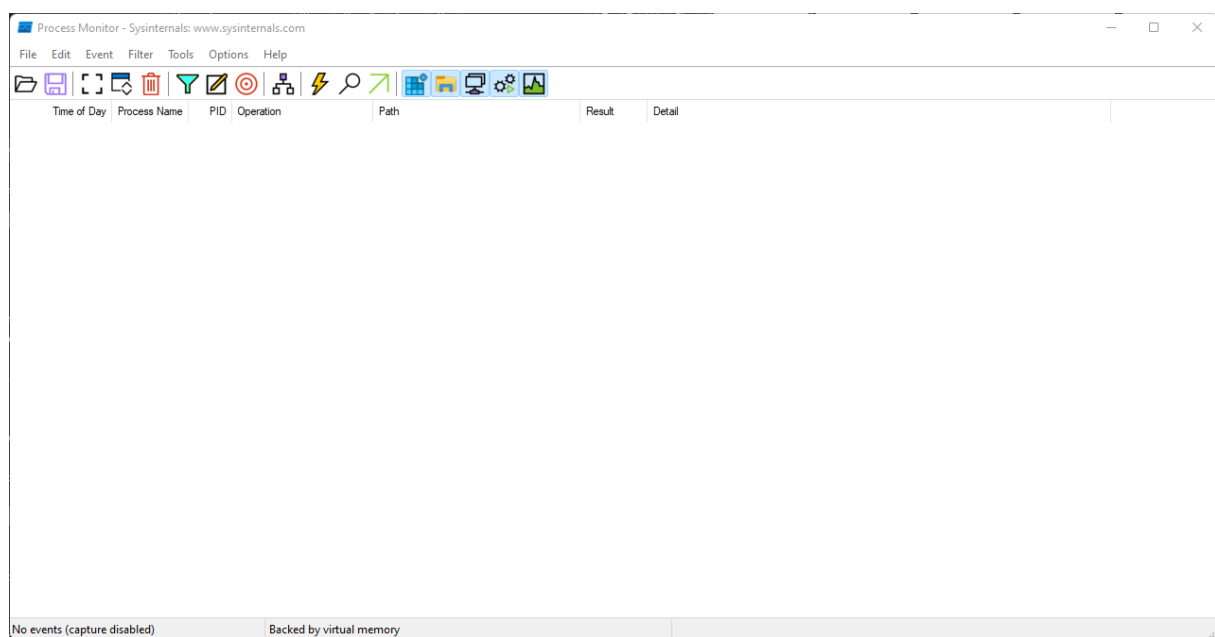


Figure 12 : Fenêtre principale de Process Monitor

Tous les événements capturés par Process Monitor apparaissent sous forme de liste. Où il est détaillé certaines informations en rapport avec ces événements.

Afin de tirer le meilleur parti de cet outil, commençons par analyser ces fonctionnalités principales. La figure ci-dessous montre l'accès aux fonctionnalités les plus couramment utilisées :

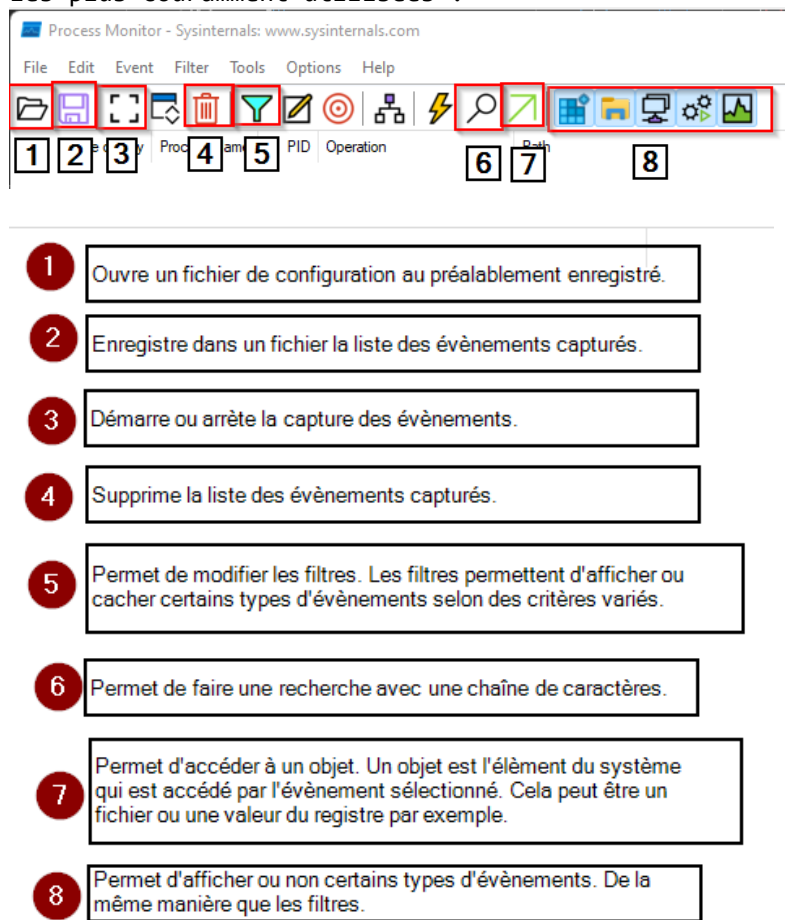


Figure 13 : Fonctionnalités principales de Process Monitor

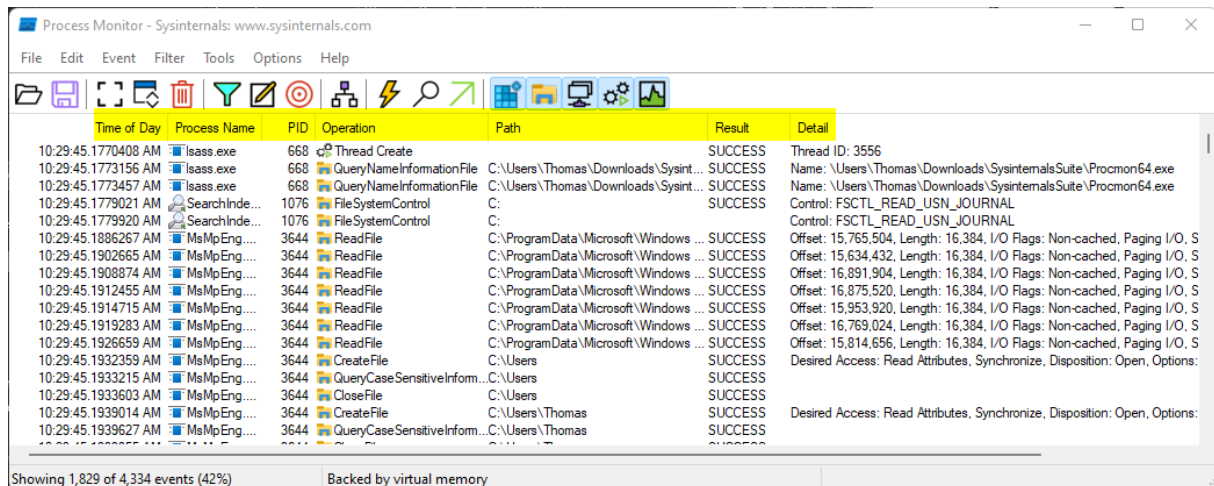
Cette liste n'est pas exhaustive concernant les fonctionnalités de Process Monitor, cependant celle-ci regroupe toutes les fonctionnalités utilisées le plus souvent.

Détail des événements

Commençons par capturer des événements, pour cela, cliquez sur le bouton noté « 3 » sur [cette figure](#) pour commencer la capture. Attendez quelques secondes, puis recliquez sur le même bouton pour stopper la capture.

Process Monitor a pendant ces quelques secondes capturé tous les événements système non filtrés. C'est-à-dire tous puisqu'il n'y a aucun filtre pour l'instant.

La figure ci-dessous montre un exemple de résultat obtenu :



Time of Day	Process Name	PID	Operation	Path	Result	Detail
10:29:45.1770408 AM	lsass.exe	668	Thread Create		SUCCESS	Thread ID: 3556
10:29:45.1773156 AM	lsass.exe	668	QueryNameInformationFile	C:\Users\Thomas\Downloads\Sysint...	SUCCESS	Name: \Users\Thomas\Downloads\SysinternalsSuite\Procmon64.exe
10:29:45.1773457 AM	lsass.exe	668	QueryNameInformationFile	C:\Users\Thomas\Downloads\Sysint...	SUCCESS	Name: \Users\Thomas\Downloads\SysinternalsSuite\Procmon64.exe
10:29:45.1779021 AM	SearchIndexing.exe	1076	FileSystemControl	C:	SUCCESS	Control: FSCTL_READ_USN_JOURNAL
10:29:45.1779920 AM	SearchIndexing.exe	1076	FileSystemControl	C:	SUCCESS	Control: FSCTL_READ_USN_JOURNAL
10:29:45.1886267 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 15,765,504, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1902665 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 15,634,432, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1908874 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 16,891,904, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1912455 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 16,875,520, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1914715 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 15,953,920, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1919283 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 16,769,024, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1926659 AM	MsMpEng.exe	3644	ReadFile	C:\ProgramData\Microsoft\Windows ...	SUCCESS	Offset: 15,814,656, Length: 16,384, I/O Flags: Non-cached, Paging I/O, S
10:29:45.1932359 AM	MsMpEng.exe	3644	CreateFile	C:\Users	SUCCESS	Desired Access: Read Attributes, Synchronize, Disposition: Open, Options:
10:29:45.1933215 AM	MsMpEng.exe	3644	QueryCaseSensitiveInform...	C:\Users	SUCCESS	
10:29:45.1933603 AM	MsMpEng.exe	3644	CloseFile	C:\Users	SUCCESS	
10:29:45.1939014 AM	MsMpEng.exe	3644	CreateFile	C:\Users\Thomas	SUCCESS	Desired Access: Read Attributes, Synchronize, Disposition: Open, Options:
10:29:45.1939627 AM	MsMpEng.exe	3644	QueryCaseSensitiveInform...	C:\Users\Thomas	SUCCESS	

Figure 14 : Exemple de résultat de capture

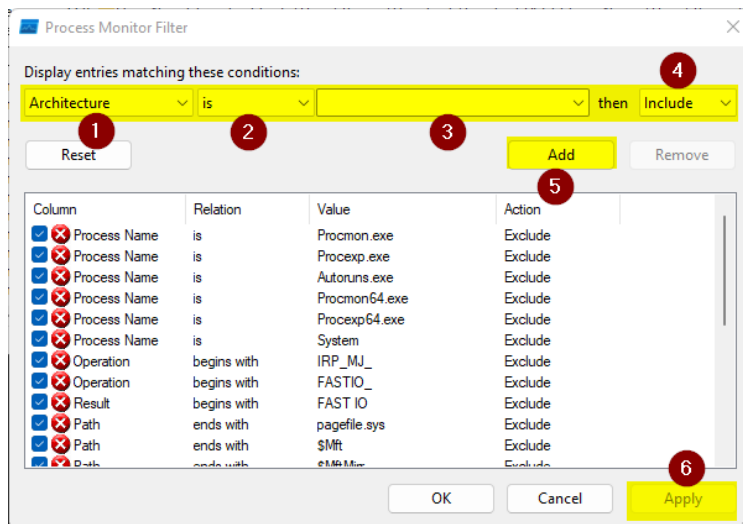
On retrouve parmi les colonnes affichées par défaut les informations suivantes :

- **Time of Day:** l'heure de la journée à laquelle s'est produit l'évènement. Cette information est donnée à la microseconde près.
- **Process Name:** indique le nom du processus à l'origine de l'évènement. Plus précisément, celui de son exécutable.
- **PID :** Process identifier, il s'agit d'un identifiant unique à chaque processus qui permet de facilement les différencier.
- **Operation:** il s'agit de la nature de l'évènement. Plus précisément ce que le processus a essayé de faire. L'opération peut se traduire par un verbe d'action.
- **Path:** le chemin physique sur la mémoire de masse relative à l'évènement.
- **Result:** le résultat de l'opération décrite dans la colonne « Operation ».
- **Detail:** Détails concernant l'évènement.

Les filtres

Les filtres permettent de masquer ou d'afficher certains évènements. Il est possible de n'afficher que les évènements qui possèdent l'opération « ReadFile » par exemple. À l'inverse il est aussi possible de masquer certains évènements qui obstruent la lecture des résultats.

Pour paramétrer les évènements, l'on clique sur le bouton noté « 5 » sur [cette figure](#), la fenêtre de configuration des filtres apparaît alors comme le montre la figure suivante :



La procédure pour ajouter un filtre est la suivante :

1. Sélectionner le type de filtre, le type est identique aux noms des colonnes présentées plus tôt.
2. Sélectionner la condition, telle que « est », ou « contient ».
3. Indiquer la valeur à vérifier. Cette valeur sera comparée selon la condition choisie.
4. Définir si le filtre doit inclure ou exclure les événements qui respectent la condition définie.
5. Cliquer sur ajouter pour ajouter le filtre à la liste des filtres actifs.
6. Appliquez les changements à la liste des événements capturés.

Cas 1 — Modification de la page d'accueil du navigateur

Contexte

Les utilisateurs de l'entreprise Vinci, cliente de NSI, se connectent à l'extranet de l'entreprise avec un identifiant et un mot de passe, via un portail de connexion Web.

Or, bien que la page de connexion n'ait rien d'anormal, l'URL d'accès à cette page n'est pas tout à fait la même, et pourrait tromper certains salariés de l'entreprise qui ne verraient pas la supercherie. Une fois les identifiants entrés, le pirate peut y accéder comme bon lui semble, et l'utilisateur se retrouve redirigé vers le vrai extranet de Vinci.

Cette page de connexion a été définie comme étant la page par défaut du navigateur Web Mozilla Firefox.

But de l'attaque

Le but de cette attaque est de récupérer les identifiants de connexion des utilisateurs de l'extranet de Vinci afin de pouvoir y accéder par le biais de comptes utilisateurs compromis. Les identifiants sont dérobés par le biais d'un faux portail de connexion, hébergé par le pirate. Contrairement au vrai portail de connexion qui lui est hébergé sur le serveur Web de l'entreprise.

La fausse page de connexion est très similaire à la page originale, seule l'URL peut différer et tromper les salariés.

Comment repérer la supercherie ?

Dans un premier temps, il faut vérifier si la page d'accueil du navigateur a été altérée ou non. Pour cela, allez dans les paramètres de la page d'accueil de Firefox. Ces paramètres sont accessibles par le petit engrenage situé en haut à droite de la page d'accueil du navigateur. Puis cliquez sur « Gérer plus de paramètres ».

Les figures suivantes détaillent la procédure :

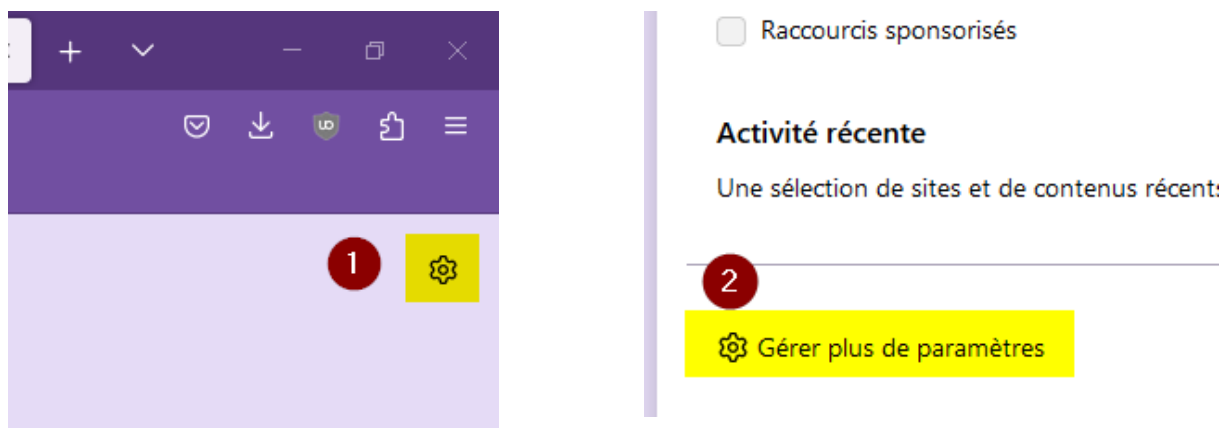


Figure 15 : Procédure d'accès aux paramètres de Firefox.

Les paramètres de gestion de la page d'accueil apparaissent :

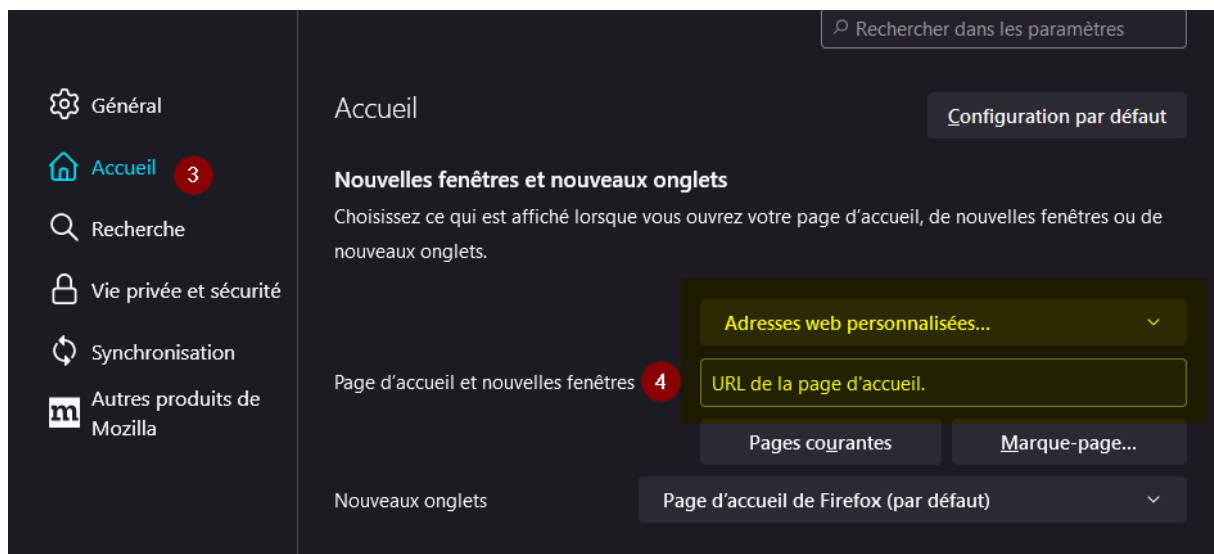


Figure 16 : Paramètres de gestion de la page d'accueil de Firefox

Il suffit ensuite de vérifier si l'URL indiquée dans le champ 4 indiqué ci-dessus correspond à l'URL du portail Web authentique.

Intercepter la modification

La prochaine étape est de capturer le processus responsable de ce changement. Pour commencer, démarrez Process Monitor. Gardez sur une moitié de l'écran le navigateur Web, et sur l'autre moitié Process Monitor.

Si des événements sont déjà capturés, videz le cache de Process Explorer en cliquant que le bouton noté 4 sur [cette figure](#). Supprimez également les filtres en place s'il y en a. Pour cela, cliquez sur le bouton noté 5 sur [cette figure](#) et cliquez sur le bouton « Reset », ce qui réinitialisera les filtres.

Ensuite, démarrez la capture en cliquant sur le bouton noté 3 sur [cette figure](#).

Après avoir démarré la capture, retournez sous Firefox et modifiez manuellement la page d'accueil en suivant les étapes décrites plus tôt pour accéder aux paramètres du navigateur.

Après avoir modifié la page d'accueil, arrêtez la capture.

Le but ici est de voir où sont enregistrées les préférences de l'utilisateur. Il nous faut donc trouver l'événement sous Process Explorer qui témoigne de la modification de cette page d'accueil.

Afin de trouver le bon événement, on utilisera deux filtres :

- Un filtre d'inclusion de catégorie « Write ».
- Un filtre d'inclusion de nom de processus « firefox.exe ».

Pour savoir comment ajouter des filtres, reportez-vous à [cette partie](#) du document concernant les filtres de Process Monitor.

Après avoir été filtrés, on remarque dans les résultats de la capture que Firefox semble réaliser des opérations de création et d'écriture dans un fichier préfixé « prefs », qui correspondrait à « preferences », tel en témoigne la figure suivante :

PM	firefox.exe	6744	CreateFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs.js
PM	firefox.exe	6744	CreateFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	CreateFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	CreateFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	WriteFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	WriteFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	WriteFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	WriteFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	WriteFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js
PM	firefox.exe	6744	SetRenameInformationFile	C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\{u4c8ypq8.default-release}\prefs-1.js

Figure 17 : Partie des résultats de la capture de Firefox.

Ce fichier « prefs.js » se situe dans le répertoire du profil de l'utilisateur, ici « u4c8ypq8.default-release ». Il s'agit du profil Firefox par défaut.

La nature du fichier concorde donc bien avec nos attentes.

Création du piège

Désormais nous savons où Firefox enregistre les préférences de l'utilisateur. Nous allons maintenant créer un piège pour capturer le processus qui serait susceptible de modifier la page d'accueil, soit modifier ce fichier « prefs.js ».

Dans Process Monitor, nous allons modifier quelques filtres :

- Modifiez le filtre d'inclusion de nom de processus ayant comme valeur « firefox.exe », en passant ce filtre en mode exclusion. Puisque nous cherchons un processus autre que Firefox.
- Ajoutez un filtre d'inclusion de type « Path », ayant comme condition « contains », et comme valeur « prefs.js ». Grâce à ce filtre, nous ne
- verrons uniquement les modifications apportées à ce fichier.

La figure suivante montre les filtres finaux :

Column	Relation	Value	Action
<input checked="" type="checkbox"/> Path	contains	prefs.js	Include
<input checked="" type="checkbox"/> Category	is	Write	Include
<input checked="" type="checkbox"/> Process Name	is	firefox.exe	Exclude

Figure 18 : Filtres du piège de capture Firefox

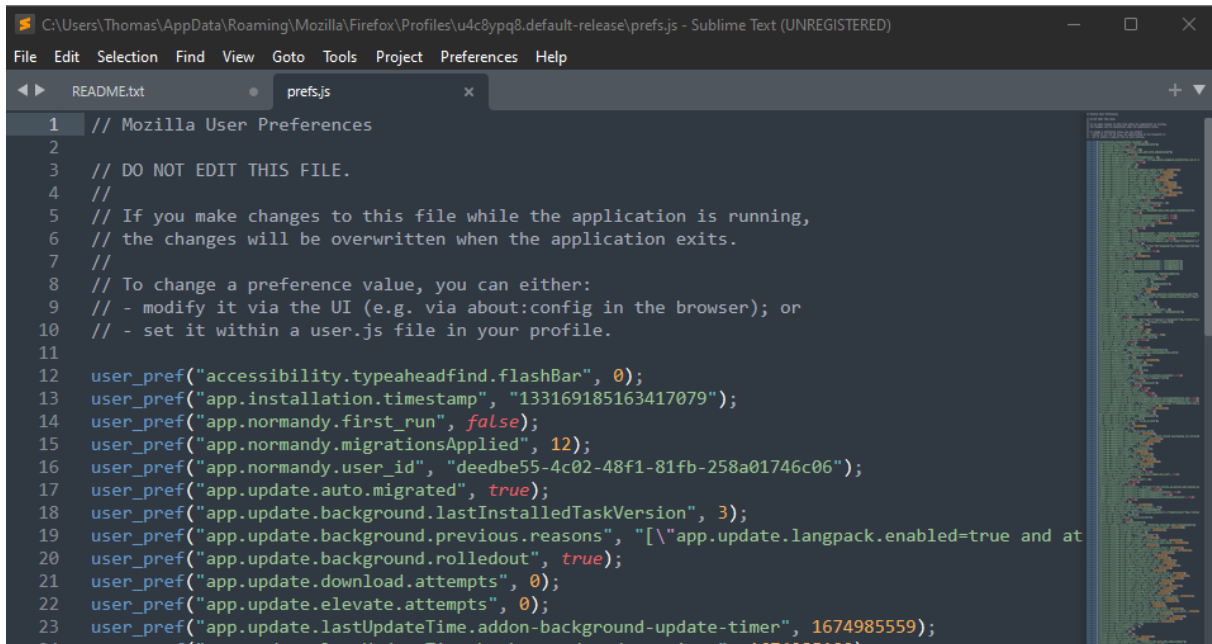
À présent, on peut redémarrer la capture. Tous les événements qui s'afficheront dans la liste seront des processus tiers qui modifient le fichier « prefs.js ».

On peut faire le test en modifiant manuellement ce fichier. Pour cela, rendez-vous dans le répertoire « % appdata %\Mozilla\Firefox\Profiles », puis ouvrez le dossier correspondant à votre profil Firefox¹⁵.

Enfin dans ce répertoire, cherchez le fichier « prefs.js », puis ouvrez-le avec n'importe quel éditeur de texte, essayons avec Sublimetext.

¹⁵ <https://support.mozilla.org/fr/kb/gestionnaire-profils-creer-supprimer-changer-profils-firefox>

Ci-dessous, la figure montre un aperçu du contenu du fichier « prefs.js » ouvert avec Sublimetext :



```

1 // Mozilla User Preferences
2
3 // DO NOT EDIT THIS FILE.
4 //
5 // If you make changes to this file while the application is running,
6 // the changes will be overwritten when the application exits.
7 //
8 // To change a preference value, you can either:
9 // - modify it via the UI (e.g. via about:config in the browser); or
10 // - set it within a user.js file in your profile.
11
12 user_pref("accessibility.typeaheadfind.flashBar", 0);
13 user_pref("app.installation.timestamp", "133169185163417079");
14 user_pref("app.normandy.first_run", false);
15 user_pref("app.normandy.migrationsApplied", 12);
16 user_pref("app.normandy.user_id", "deedbe55-4c02-48f1-81fb-258a01746c06");
17 user_pref("app.update.auto.migrated", true);
18 user_pref("app.update.background.lastInstalledTaskVersion", 3);
19 user_pref("app.update.background.previous.reasons", "[\"app.update.langpack.enabled=true and at
20 user_pref("app.update.background.rolledout", true);
21 user_pref("app.update.download.attempts", 0);
22 user_pref("app.update.elevate.attempts", 0);
23 user_pref("app.update.lastUpdateTime.addon-background-update-timer", 1674985559);
24 user_pref("app.update.lastUpdateTime.background-update-timer", 1674985100);

```

Figure 19 : Aperçu du contenu du fichier « prefs.js ».

Remarque :

Avant de modifier le fichier, vérifier que la capture sous Process Monitor est bien active !

Dès lors que nous modifions et enregistrons le fichier, un évènement s'affiche dans Process Monitor, tel que le montre la figure suivante :

1:36:3... sublime_text.exe 5008 WriteFile C:\Users\Thomas\AppData\Roaming\Mozilla\Firefox\Profiles\u4c8ypq8.default-release\prefs.js SUCCESS Offset: 0, Length: 1...

Figure 20 : Évènement de modification du fichier « prefs.js » par « sublime_text.exe ».

Notre piège fonctionne donc parfaitement ! N'importe quel processus qui apportera une modification au fichier « prefs.js » sera capturé par Process Monitor.

Dans ce cadre de test, le processus responsable de la modification est bien évidemment notre éditeur de texte favori. Mais dans un cas réel, il peut s'agir d'un tout autre processus.

Process Monitor nous indique l'exécutable, ainsi que le PID. On a alors utilisé Process Explorer pour enquêter sur ce processus et ainsi procéder au nettoyage et à la désinfection de machines.

Les différents filtres pour la mise en place du piège sont disponibles dans le sous-dossier « src » du build, portant le nom « [firefox homepage trap filters.pmc](#) ». Les résultats de ce piège sont également disponibles sous le fichier « [firefox homepage trap results.PML](#) ».

Conclusion

Ce jeu de filtres nous a effectivement permis de déceler quel processus est à l'origine de la modification de la page d'accueil de Firefox.

Dans le cadre où l'analyse est à porter sur un parc de machines, alors il serait possible d'utiliser un script PowerShell afin d'automatiser le processus. Le programme Process Explorer peut en effet être démarré de cette manière, et on peut y passer en paramètres un fichier de configuration.

Il ne reste plus qu'à redirigé les résultats de sortie dans un fichier texte, puis de laisser le script effectuer son travail. Lors de l'analyse des résultats, si l'on remarque qu'un processus X est à l'origine du changement sur plusieurs machines différentes. Alors il est possible de mener un travail de recherche post-analyse afin de procéder au nettoyage du parc informatique et à la restauration des pages d'accueil de Firefox.

Cas 2 — DNS Empoisonné

Contexte

L’helpdesk de niveau 1 reçoit un ticket d’incident. Un utilisateur de chez Vinci se plaint du changement de la page de connexion à l’extranet de l’entreprise. De ce cas présent, nous sommes en présence d’un empoisonnement de DNS (Domain Name System).

But de l’attaque

Ici le but de cette attaque est de mener l’utilisateur vers une page externe qui reproduit très fidèlement des services internes à l’entreprise, tels que la page de connexion comme décrite précédemment.

La prochaine étape est d’utiliser un serveur DNS pour rediriger l’utilisateur vers la page piégée. Le but du DNS est de lier une adresse IP à un nom de domaine. Il suffit alors de lier l’adresse IP du serveur Web sur lequel se trouve la page piégée à un nom de domaine très proche du nom de domaine utilisé en temps normal, en voici un exemple concret :

Cas	URL	Adresse du serveur Web de destination
DNS Sein	https://vinci.com	102.178.87.19
DNS Empoisonne	https://vinci.com	134.15.7.90

Le DNS empoisonné redirige la machine vers un tout autre serveur Web.

Comment repérer un DNS empoisonné ?

Un DNS empoisonné se définit comme étant un DNS non voulu par l’administrateur du système. Il s’agit donc de toute adresse IP différente de l’adresse IP du DNS souhaité.

À présent il faut repérer où Windows enregistre les paramètres DNS spécifiques à une carte réseau. Puisque ces paramètres doivent être restaurés au démarrage de la machine, ceux-ci sont stockés sur mémoire de masse. Cependant ils peuvent être stockés sous différentes formes, telles qu’une clé de registre ou un fichier de configuration par exemple.

Nous allons donc utiliser l’outil Process Explorer pour repérer où sont stockés ces paramètres et qui les modifient.

Ce procédé se scinde en quatre étapes principales :

1. Modification manuelle des paramètres
2. Identification de l’emplacement des paramètres
3. Création d’un piège de capture
4. Capture du processus à l’origine de la modification

Identifier les paramètres du DNS

Pour commencer, il faut trouver où et comment Windows enregistre les paramètres du DNS. Pour cela, on utilisera les filtres de Process Explorer pour capturer toutes les modifications des paramètres DNS.

Étant donné que les paramètres DNS sont des paramètres de bas niveau, propres à chaque interface, alors l'on peut en déduire qu'il est très probable que les paramètres DNS soient stockés sous forme de valeur, à l'intérieur d'une clé de registre.

On peut alors commencer par poser deux filtres :

- Le premier sera de type « Category » et prendra la valeur « WRITE ». Puisque l'édition du registre implique forcément des écritures sur la mémoire de masse.
- Le second sera de type « Operation » et prendra la valeur « RegSetValue ». Cette opération définit l'édition d'une valeur du registre.

La figure ci-dessous montre les filtres désormais configurés :



Column	Relation	Value	Action
<input checked="" type="checkbox"/>  Operation	is	RegSetValue	Include
<input checked="" type="checkbox"/>  Category	is	WRITE	Include

Figure 21 : Filtres de capture du DNS Empoisonné

Remarque :

Le jeu de filtres utilisé est disponible dans le fichier de configuration nommé « [poisoned_dns_filters.pmc](#) », qui se situe dans le répertoire « data » du build.

Afin de percevoir des éditions dans le registre, il faut modifier manuellement les valeurs du serveur DNS préféré. Pour cela, tapez dans le menu démarrer « Connexions Réseau ». La figure suivante montre la fenêtre des connexions réseau :

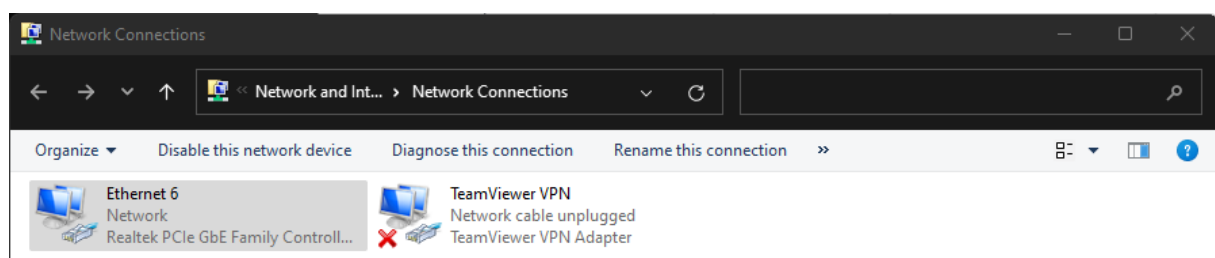


Figure 22 : Fenêtre des connexions réseau

Puis faites un clic droit sur la connexion de votre choix, dans notre cas, le choix de la connexion choisie n'importe pas sur les résultats.

Comme le montre la figure suivante, dans le menu contextuel, cliquez sur propriétés :

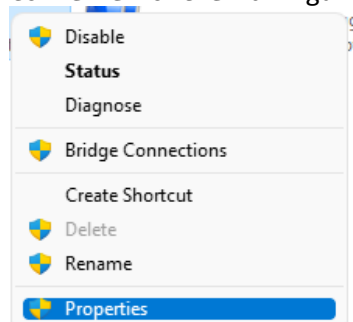
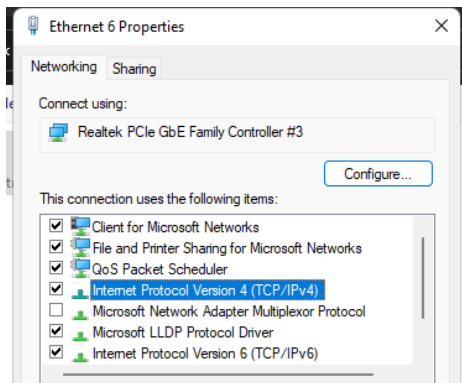


Figure 23 : Menu contextuel de la connexion

Les propriétés de l'interface apparaissent alors, double cliquez sur « Protocole Internet Version 4 » comme le montre la figure suivante :



Les paramètres IPv4 de l'interface apparaissent. Dans la nouvelle fenêtre, sélectionnez « Utiliser les paramètres DNS suivants », puis renseignez l'adresse IP d'un serveur DNS au choix. Prenons par exemple le serveur DNS de Google, tel que « 8.8.8.8 ». La figure suivante indique les étapes de configuration :

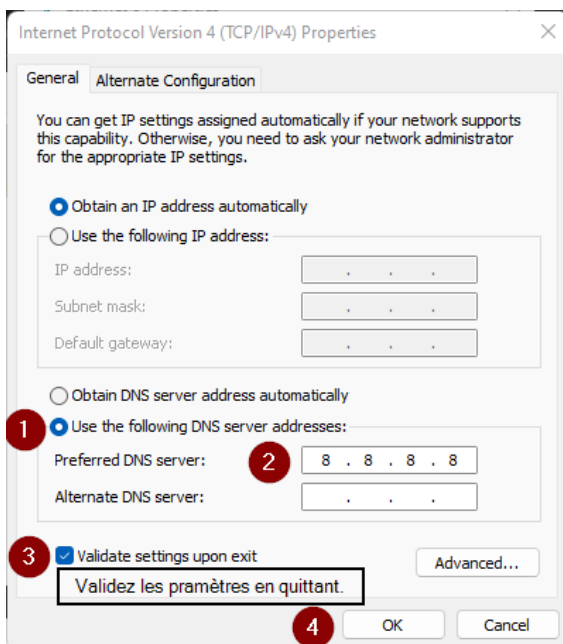


Figure 24 : Étapes de configuration manuelle du serveur DNS préféré.

Attention :

Le paramétrage d'un DNS erroné sur l'interface active résultera en l'incapacité d'accéder à Internet.

Avant de cliquer sur « OK », vérifiez que la capture sur Process Monitor est active !

Capture des résultats sous Process Monitor

Après ce paramétrage, cliquez sur « OK » sur chaque fenêtre pour valider les changements. Une fois validés, retournez sur Process Monitor et arrêtez la capture. Si les filtres ont correctement été définis, des événements devraient être apparus sous Process Monitor, tel que le montre la figure suivante :

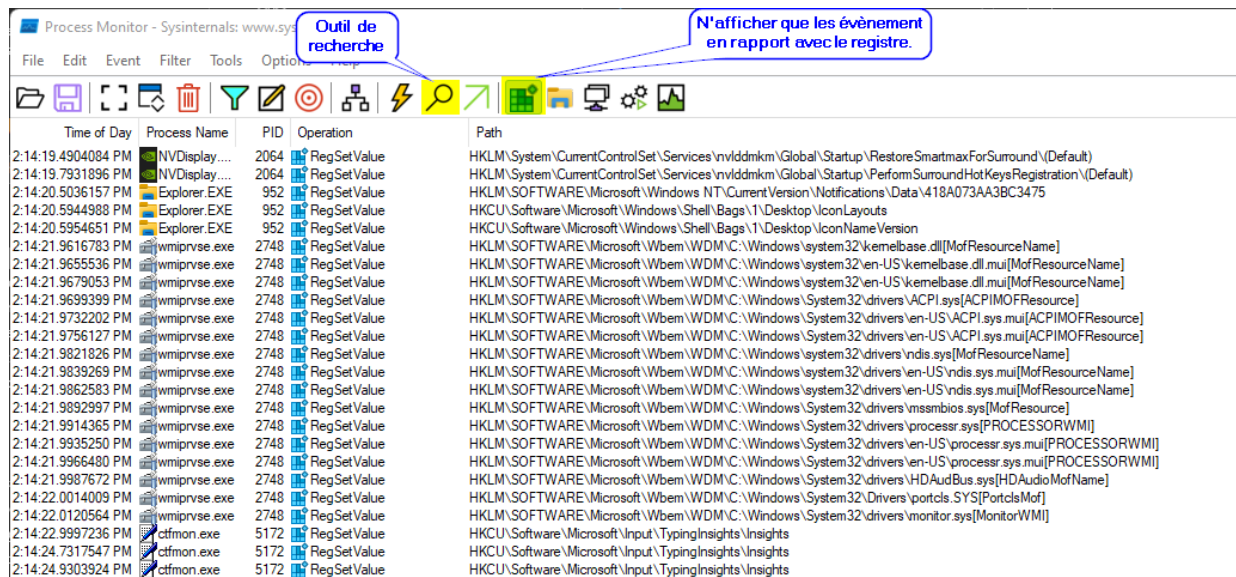


Figure 25 : Exemples d'événements capturés sous Process Monitor

Cependant un très grand nombre de ces événements ne nous sont d'aucune utilité. Nous allons alors utiliser l'outil recherché, symbolisé par l'icône de loupe, mise en valeur sur la figure ci-dessus.

Il est également possible de n'afficher uniquement les événements en lien avec le registre en ne sélectionnant uniquement l'icône bleue mise en valeur sur la figure ci-dessus.

L'outil de recherche permet d'effectuer une recherche par chaîne de caractères, attention, la recherche est sensible à la casse. La figure ci-dessous montre la fenêtre de l'outil de recherche :

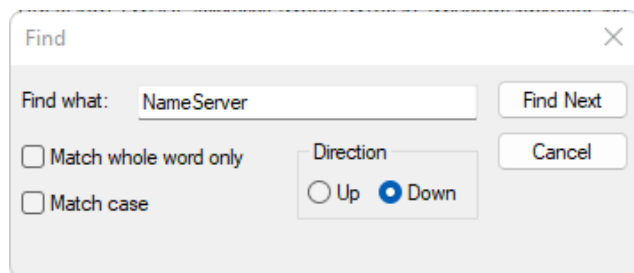


Figure 26 : Outil de recherche de Process Monitor

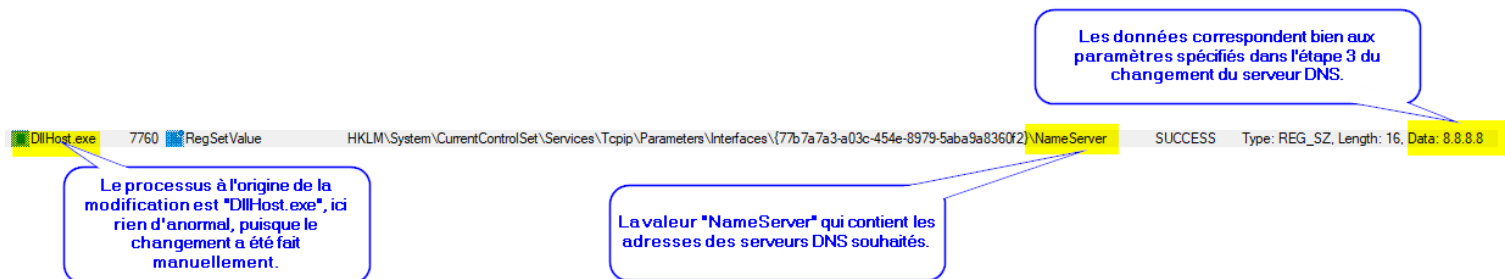
Dans le registre, la valeur « NameServer » contient la ou les adresses des DNS paramétrés pour chaque interface¹⁶. Il nous suffit alors de réaliser une recherche avec cette chaîne de caractères, tel le montre la figure ci-dessus.

Si plusieurs résultats ressortent, effectuer la même recherche affichera le prochain résultat.

¹⁶ <https://serverfault.com/questions/856244/where-does-windows-store-dns-entry-list>

Dans le cas où une valeur du registre est modifiée. Process Monitor affiche les nouvelles données dans la colonne « Detail », au côté de la mention « Data : ». Après avoir validé la recherche, il suffit alors de vérifier la colonne « Detail » et d'y trouver la mention « Data : ». L'adresse du DNS entrée plus tôt devrait y être affichée. Si tel est le cas, alors vous avez trouvé la bonne capture.

La figure ci-dessous montre à quoi devrait ressembler la capture indiquant le changement de la valeur « NameServer » du registre par la valeur du nouveau serveur DNS :



Le processus à l'origine de la modification est "DllHost.exe", ici rien d'anormal, puisque le changement a été fait manuellement.

La valeur "NameServer" qui contient les adresses des serveurs DNS souhaités.

Les données correspondent bien aux paramètres spécifiés dans l'étape 3 du changement du serveur DNS.

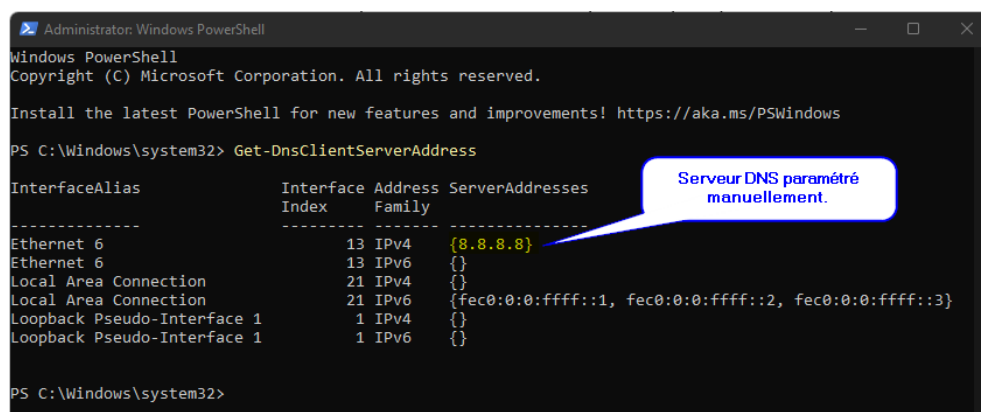
On remarque d'ailleurs que la valeur « NameServer » est contenue dans la clé de registre portant comme nom un UUID¹⁷ (Universal Unique Identifier). Cet UUID représente la connexion réseau sélectionnée dans le panneau de configuration. En outre, les paramètres sont donc propres à chaque interface réseau de l'ordinateur.

Conclusion

Dans le cas où la machine serait saine, les filtres proposés plus tôt permettent comme nous l'avons pu le démontrer de capturer quel processus précis est à l'origine du changement de DNS. À partir de là, il suffirait alors d'analyser l'exécutable à l'origine du processus et mener le nettoyage et la désinfection de la machine.

Cependant, si le DNS est déjà modifié, alors il ne sera pas possible de savoir quel processus le modifie via Process Explorer. Il faudrait alors vérifier manuellement ou via un outil tel que **PowerShell**¹⁸, la valeur actuelle du DNS paramétré.

Avec PowerShell par exemple, la commande « Get-DnsClientServerAddress » permet de lister les serveurs DNS paramétrés pour chacune des interfaces de la machine. La figure ci-dessous montre l'exécution et le résultat de cette commande :



```

PS C:\Windows\system32> Get-DnsClientServerAddress

InterfaceAlias      Interface Index Address Family ServerAddresses
-----
Ethernet 6          13          IPv4      {8.8.8.8}
Ethernet 6          13          IPv6      {}
Local Area Connection 21          IPv4      {}
Local Area Connection 21          IPv6      {fec0:0:0:ffff::1, fec0:0:0:ffff::2, fec0:0:0:ffff::3}
Loopback Pseudo-Interface 1 1          IPv4      {}
Loopback Pseudo-Interface 1 1          IPv6      {}
  
```

Serveur DNS paramétré manuellement.

¹⁷ https://fr.wikipedia.org/wiki/Universally_unique_identifier

¹⁸ <https://learn.microsoft.com/fr-fr/powershell/scripting/overview?view=powershell-7.3>

Table des illustrations

Figure 1 Anatomie du processeur	5
Figure 2 Attente et exécution des threads par le processeur	5
Figure 3 Définition de la priorité et de l'affinité d'un processus	6
Figure 4 Interactions entre le processeur et les mémoires de traitement	8
Figure 5 : Structure du registre sous Windows 11	10
Figure 6 : Fenêtre principale de Process Explorer	11
Figure 7 : Processus enfants d'Explorer.exe	11
Figure 8 : Exemple d'informations des processus sous Process Explorer	12
Figure 9 : Étendue des informations disponibles sous Process Explorer	12
Figure 10 : Menu contextuel Virus total	13
Figure 11 : Exemple de résultat Virus total	13
Figure 12 : Fenêtre principale de Process Monitor	15
Figure 13 : Fonctionnalités principales de Process Monitor	16
Figure 14 : Exemple de résultat de capture	17
Figure 15 : Procédure d'accès aux paramètres de Firefox.	19
Figure 16 : Paramètres de gestion de la page d'accueil de Firefox	20
Figure 17 : Partie des résultats de la capture de Firefox.	21
Figure 18 : Filtres du piège de capture Firefox	21
Figure 19 : Aperçu du contenu du fichier « prefs.js »	22
Figure 20 : Évènement de modification du fichier « prefs.js » par « sublime_text.exe »	22
Figure 21 : Filtres de capture du DNS Empoisonné	25
Figure 22 : Fenêtre des connexions réseau	25
Figure 23 : Menu contextuel de la connexion	25
Figure 24 : Étapes de configuration manuelle du serveur DNS préféré.	26
Figure 25 : Exemples d'évènements capturés sous Process Monitor	27
Figure 26 : Outil de recherche de Process Monitor	27