



# TITRE

## Sous-Titre

### PAGE DE SERVICE

**Référence :**

**Plan de classement :**

**Niveau de confidentialité :** public | corporate | confidential

**Mises à jour**

Version	Date	Auteur	Description du changement
23/01/2023		BARBIER PIERRE	Rédaction du contexte.
25/01/2023		BARBIER PIERRE	Rédaction cas 1.
26/01/2023		BARBIER PIERRE	Rédaction des prérequis.
28/01/2023		BARBIER PIERRE	Rédaction du cas d'attaque par empoisonnement DNS.
11/02/2023		BARBIER PIERRE	Rédaction de la partie détection des programmes malveillant.
1/03/2023		BARBIER PIERRE	Rédaction du contexte, introduction à la signature.
3/03/2023		BARBIER PIERRE	Modification de l'automate et résultat d'analyse
4/03/2023		BARBIER PIERRE	Test des outils en ligne de commandes.
5/03/2023		BARBIER PIERRE	Analyse de signature avec Process Explorer
18/03/2023		BARBIER PIERRE	Ecriture de la notion mémoire de masse, disque HDD et SSD + Préparation et autopsie d'un clef FAT et NTFS
19/03/2023		BARBIER PIERRE	Ecriture de la partie « boot séquence ».

**Validation**

Version	Date	Nom	Rôle
1	29/01/2023	JEROME VALENTI	ENSEIGNANT
2	19/02/2023	JEROM VALENTI	ENSEIGNANT
3	5/03/2023	JEROME VALENTI	ENSEIGNANT
4	19/03/2023	JEROME VALENTI	ENSEIGNANT

**Diffusion**

Version	Date	Nom	Cadre de la diffusion
1	29/01/2023	JEROME VALENTI	REMIS POUR EVALUATION
2	19/02/2023	JEROME VALENTI	REMIS POUR EVALUATION
3	5/03/2023	JEROME VALENTI	REMIS POUR EVALUATION
4	19/03/2023	JEROME VALENTI	REMIS POUR EVALUATION

## Table des matières

I)	Rappel du contexte	3
II)	Objectif	3
III)	Bibliographie	3
IV)	Les prérequis fondamentaux	3
1.1)	Les virus ou programme malveillant.	4
1.2)	Les notions de processus, thread et handles.	4
1.3)	Les processus.	5
1.4)	Les thread.	5
1.5)	Les handles.	6
1.6)	La mémoire de masse aussi appelée mémoire morte.	6
1.7)	Les attaques.	6
1.8)	Lien de corrélation.	7
V)	Contextualisation	8
1.1)	outil d'analyse	8
Process Explorer.		8
Virus Total		9
Process Monitor		9
VI)	Cas 1 : modification du home page du navigateur Firefox.	10
1.1)	Contextualisation.	10
1.2)	Pratique :	10
VII)	Cas 2 : Empoisonnement par DNS.	13
1.1)	contextualisation	13
1.2)	Le but recherché	13
1.3)	Méthode de détection	13
1.4)	Étude des paramètres du DNS	14
1.5)	La pose des filtres	14
IIIX)	la détection de programme malveillant.	17
1.1)	Comprendre le fonctionnement de virus total.	17
1.2)	Que penser de virustotal ?	19
IX)	Automatisation de la recherche de menaces.	20
1.1)	L'API.	20
1.2)	Le script.	20

1.3) Robot virus total.	20
1.4) Les fichiers de configurations.	23
1.5) Conclusion.	24
X) Une analyse de fichier approfondie grâce à la signature de code.	25
1.1) Contextualisation.	25
1.2) Introduction à la signature de code, le cas particulier des pe sous Windows.	25
1.3) La production et la lecture d'une signature numérique.	26
1.4) Une différence entre un exécutable pirate et légitime.	27
1.5) Les signatures de code standard et extended validation.	27
<b>1.6) Les outils en lignes de commandes.</b>	28
1) Sigcheck.	28
2) Sightool.	29
1.7) La détection de signature dans process Explorer.	29
1.8) Modification de l'automate.	30
1.9) Résultat d'analyse et vérification du bon fonctionnement des modifications.	33
1.10) Conclusion.	33
XI) Analyse de fichiers.	34
1.1) La mémoire de masse.	34
1.2) Les différentes mémoires de masse.	34
1) Le disque dur HDD ou disque dur mécanique :	34
2) Le disque dur SSD :	35
1.3) Les partitions.	36
1.4) Les différent types de partition.	37
1.4) Introduction aux tables de partitions.	37
1.5) Préparation et autopsie d'une clé USB fat.	38
1) Le FAT.	38
2) Les outils en lignes de commandes.	38
3) Phase d'exploration.	38
1.6) La considération de la sécurité informatique dans toutes les études et manipulations de données.	41
1.7) Préparation et autopsie d'une clé USB NTFS.	44
1) Le NTFS.	44
2) La notions d'ADS.	44
3) Les outils en ligne de commandes.	45
1.8) Le boot séquence.	47
1) L'amorçage de la machine.	47
1.9) Bios et UEFI.	48
1.10) Le démarrage logiciel	48
1) Le choix du support de démarrage.	48
2) Le démarrage de Windows.	49
XII) Table des illustrations	49

## I) RAPPEL DU CONTEXTE

---

Cette documentation est livrable sous le répertoire « doc » du *build*<sup>1</sup> du projet.

- Une entreprise de fabrication a récemment fait appel à NSI en matière de cybersécurité pour mener une analyse de sécurité sur les postes de travail de ses employés. Un consultant en cybersécurité se rendra dans les locaux de l'entreprise pour effectuer une analyse des postes de travail utilisés par les employés, ainsi que les ordinateurs de bureau et les ordinateurs portables. L'étude de ces postes passera par l'étude des logiciels installés sur ces machines, on notera ici qu'il est intéressant de regarder les processus actifs sur les machines afin d'identifier une ou plusieurs vulnérabilités potentielles et des logiciels malveillants. Pour l'étude de ces processus, l'utilisation de deux logiciels est recommandée : process explorer et process monitor. Une description détaillée du fonctionnement et de l'utilité de ces logiciels vous sera fournie lors des démonstrations d'utilisation.

## II) OBJECTIF

---

- Cette documentation vise à former les employés de la société NSI sur les différents types de menaces qui peuvent altérer les différents systèmes de Windows 10, vous pourrez trouver des méthodes pour les détecter.
- En effet Windows 10 est le système d'exploitation le plus utilisé pour les ordinateurs personnels et professionnels. Afin de comprendre au mieux les différentes menaces qu'il est possible de rencontrer, les concepts de processus et de l'organisation des programmes sur un système d'exploitation seront également abordés pour aider à comprendre comment différencier les processus malveillants des autres programmes. La plupart des ordinateurs sont protégés par un ou plusieurs pare-feux et un antivirus, cependant le risque 0 n'existe pas et il est important de noter que plus de 75 % des virus peuvent tout de même passer à travers ces protections. Il est donc nécessaire d'être capable de mener une analyse plus approfondie en plus de l'utilisation de ces outils de protection.

## III) BIBLIOGRAPHIE

---

La bibliographie de ce projet est accessible localement sous le répertoire « biblio » du *build* du projet et en ligne sur le site de veille technologique du projet.

## IV) LES PREREQUIS FONDAMENTAUX

---

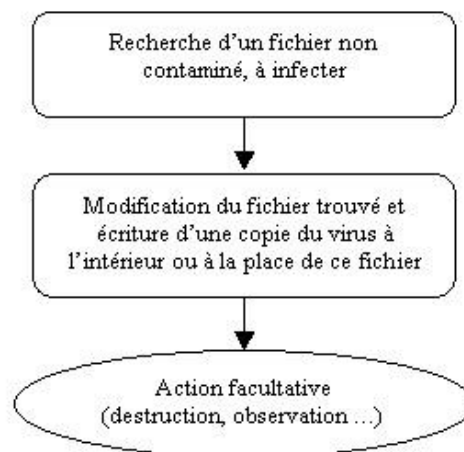
- Avant de se lancer en cybersécurité, il est nécessaire d'assimiler certaines connaissances qui vous seront indispensables pour avancer dans ce domaine et comprendre comment les programmes malveillants affectent les machines. Les connaissances dont vous aurez besoin toucheront les thèmes suivants :
  - ⇒ Le virus ou programme malveillant.
  - ⇒ Les notions de processus et de thread.
  - ⇒ La mémoire de masse aussi appelée mémoire morte.
  - ⇒ Les attaques.
- Nous verrons que ces quatre thèmes sont étroitement liés dans le monde de la cybersécurité.

---

<sup>1</sup> <http://www-igm.univ-mlv.fr/~dr/XPOSE2011/IntegrationContinue/build.html>

## 1.1) LES VIRUS OU PROGRAMME MALVEILLANT.

- Un programme malveillant aussi appelé virus est un programme, un code malveillant, qui peut vous causer du tort de différentes façons comme l'altération du système d'exploitation par la suppression de fichiers, la corruption de données ou l'altération des performances de la machine. Un virus peut se présenter sous différentes formes :
  - Le virus macro, celui-ci s'exécute à l'intérieur d'un document comme un fichier Word ou Excel, il se propagera via des macros, il peut causer des dommages importants aux fichiers.
  - Le virus boot, ce type de malware s'installe sur la zone de démarrage d'un disque, celui-ci s'active dès le démarrage de l'ordinateur.
  - Le cheval de Troie, ce sont des programmes malveillants qui se dissimulent dans des logiciels légitimes pour accéder à des informations privées ou pour contrôler à distance le poste infecté.
  - Le ver, il se propage automatiquement à d'autres ordinateurs sur un réseau ou sur internet.
  - Le rootkit, ils se dissimulent sur un système afin d'échapper à la détection des logiciels de sécurité.
- Il faut savoir qu'un virus doit contenir au moins deux parties pour pouvoir se reproduire : Un algorithme de recherche de fichier hôte à infecter et un algorithme de copie sur le fichier hôte, on peut éventuellement ajouter une troisième étape (étapes facultatives : destruction ou espionnage...). On peut donc parler d'une sorte d'organigramme du virus :



*Figure 1 : Structures d'un virus*

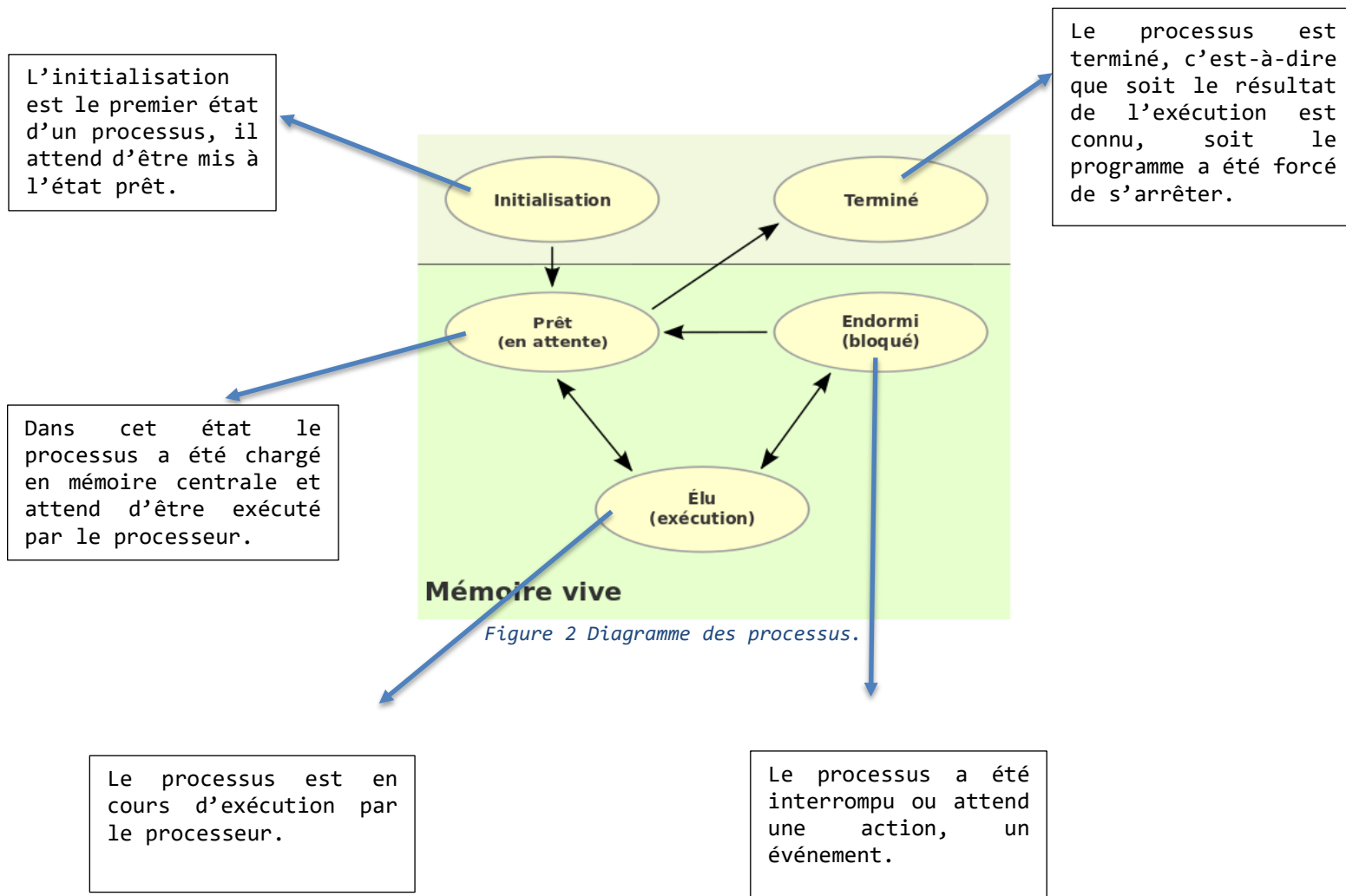
*Figure 1 Organigramme du virus.*

## 1.2) LES NOTIONS DE PROCESSUS, THREAD ET HANDLES.

- Un processus peut être vu comme l'instance d'un programme en cours d'exécution sur un poste. Pour faire simple à chaque fois qu'un programme est exécuté, un processus est créé, celui-ci sera géré par le système, on peut alors les considérer comme des tâches exécutées en arrière-plan ou en premier plan sur le poste. Un processus possède plusieurs attributs qui lui sont propres tel son ID de processus, la mémoire allouée et son état c'est-à-dire en cours d'exécution ou en sommeil. Les processus peuvent être gérés et listés à l'aide de différentes commandes par exemple la commande `Get-Process` sur l'invite PowerShell, celle-ci permet de lister tous les processus avec les exécutables.

### 1.3) LES PROCESSUS.

- Quand on parle de processus, on parle également des états successifs de celui-ci, on peut assimiler cela à un diagramme d'état :



- Les notions de « threads » et de « handles » sont également importantes pour comprendre les processus.

### 1.4) LES THREAD.

- Les threads sont des sous-unités d'un processus qui peuvent être exécutées de manière indépendante les unes des autres. Les threads ont un cycle de vie qui peut être représenté par le diagramme suivant :

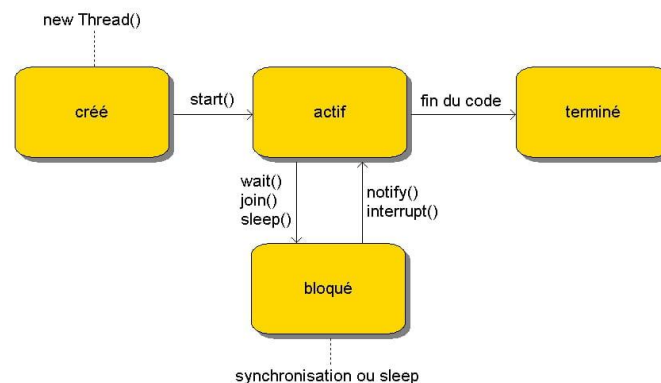


Figure 3 Diagramme des threads.

### 1.5) LES HANDLES.

- Tandis que les « handles » sont des références utilisées pour accéder aux ressources telles que les fichiers et les mémoires partagées.
- Afin de maximiser vos connaissances dans sur les notions de processus, handles et threads, voici une documentation Microsoft :

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

### 1.6) LA MEMOIRE DE MASSE AUSSI APPELEE MEMOIRE MORTE.

- La mémoire de masse est un type de mémoire permanente utilisé dans les ordinateurs pour stocker des données de manière permanente, même lorsque l'ordinateur est éteint. Elle est utilisée pour stocker des fichiers, des applications, des systèmes d'exploitation et d'autres données importantes. Il existe plusieurs types de mémoire de masse, notamment les disques durs (HDD), les disques durs à état solide (SSD), les cartes mémoire et les disques optiques (CD, DVD). La mémoire de masse est plus lente que la mémoire vive (RAM), mais elle est plus grande en taille et permet de stocker des données plus importantes de manière permanente. En gros, la mémoire de masse est l'endroit où sont stockés tous les fichiers, programmes et données de l'ordinateur à long terme.

### 1.7) LES ATTAQUES.

- En informatique une attaque est une action malveillante qui vise à compromettre la sécurité d'un système informatique. Les attaques peuvent elles aussi se présenter sous différentes formes allant de la simple tentative de vol d'information à la destruction de données ou l'interruption de service. Il existe de nombreux types d'attaque tel que :
  - Les attaques par déni de service (Dos) qui visent à rendre un service ou un site Web inaccessible en surchargeant les serveurs avec des requêtes malveillantes.
  - Les attaques par injection SQL qui visent à injecter des commandes malveillantes dans une base de données en utilisant des failles de sécurité dans les applications Web.
  - Les attaques de phishing qui visent à voler des informations sensibles en incitant les utilisateurs à saisir leurs informations sur des sites Web malveillants qui ressemblent à des sites légitimes.
  - Les attaques par phishing qui visent à voler des informations sensibles en trompant l'utilisateur avec une fausse interface par exemple.
  - Les attaques rançongiciel qui visent à chiffrer les données d'un utilisateur en échange d'une somme d'argent pour les déchiffrer.
  - Les attaques de piratages qui visent à accéder à des informations ou à des systèmes protégés sans autorisation.
- Il est important de savoir que des attaques peuvent combiner plusieurs de ces attaques pour parvenir à leur fin. Par exemple en utilisant une technique de phishing pour voler des informations sensibles puis utiliser une injection SQL pour accéder à une base de données, et utiliser un rançongiciel pour chiffrer les fichiers en échange d'une rançon.

## 1.8) LIEN DE CORRELATION.

- En somme, les trois termes abordés dans les prérequis fondamentaux sont étroitement liés, car une attaque a besoin d'un processus actif sur une machine pour pouvoir fonctionner, et une attaque peut très bien être menée par un malware. Un malware c'est tout simplement un programme qui s'exécute sous le système. L'or de son exécution, un processus sera automatiquement généré. Il est donc possible d'identifier et de stopper une menace en analysant les processus grâce à un certain outil qui sera présenté dans les deux cas d'attaques.  
Pour conclure, une bonne application de la cybersécurité doit passer par la disponibilité, c'est-à-dire la continuité du fonctionnement des ressources ; l'intégrité des ressources ; l'authentications, c'est-à-dire vérifier l'identité des utilisateurs.



## V) CONTEXTUALISATION

- Après avoir consulté l'ensemble des notions fournies ci-dessus, ce qui vous a permis de comprendre le fonctionnement d'un programme, nous allons maintenant franchir une étape et introduire certains outils qui vous permettront d'établir une analyse complète afin de repérer des processus malveillants, ici les comportements des processus malveillants qui seront abordés concerneront : l'accès à des clés de registre et des fichiers de configurations.  
Passons maintenant à l'exploration des différentes possibilités fournies par les outils ProcessMonitor et ProcessExplorer.

### 1.1) OUTIL D'ANALYSE

#### PROCESS EXPLORER.

- Process Explorer est un outil d'analyse, celui-ci permet de lister tous les processus en cours d'exécution à un instant précis sur le système. La capture présentée ci-dessous présente la fenêtre principale de Process Explorer :

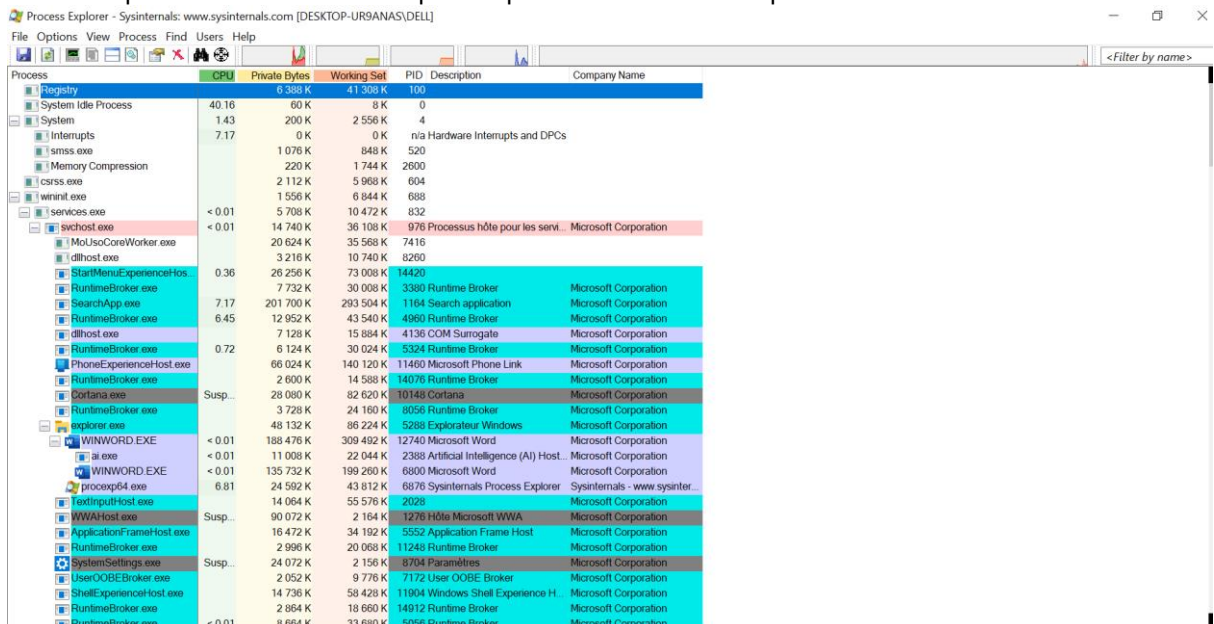


Figure 4 Fenêtre principale de Process Explorer.

- On peut relever la présence d'une arborescence qui présente les processus en cours d'exécution. On peut alors relever la présence de nombreux processus, ils peuvent être classés en deux catégories telles que les processus d'enfant qui dépendent directement des processus parents ou processus principaux. Pour illustrer mes propos, je m'appuierais ici sur l'exemple de l'explorateur de fichiers (explorer.exe), il sert de processus parents pour de nombreux programmes comme le montre la capture ci-dessous :

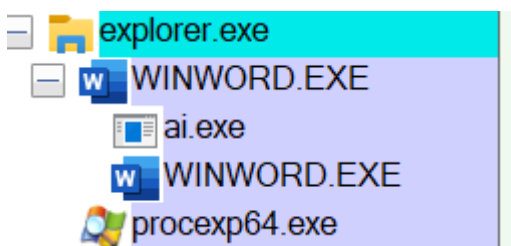


Figure 5 Les processus parents.

- Cependant certains programmes tels que Firefox sont indépendants :

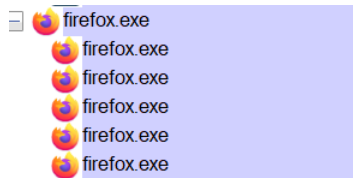


Figure 6 Les processus indépendants.

## VIRUS TOTAL

- Virus total est un service qui peut être utilisé en complément avec process Explorer, en effet cette combinaison permettra de scanner certains fichiers. Virus total scanne les fichiers avec plusieurs antivirus différents afin de maximiser la détection de code malveillant. Voici l'URL qui vous permettra d'accéder à un virus total par le biais d'un navigateur web :

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

- Cependant il existe un raccourci directement intégrer à process Explorer, en effet il suffit de faire un clic droit sur le fichier que l'on souhaite scanner et l'option de check avec virus total s'offre à vous :

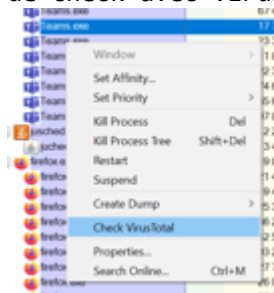


Figure 7 Check up avec virustotal.

- Après le scanne avec virus total on peut remarquer l'affichage d'un score sur 75 :

Teams.exe	89 744 K	115 800 K	13832 Microsoft Teams	Microsoft Corporation	0/75
Teams.exe	67 504 K	63 296 K	4204 Microsoft Teams	Microsoft Corporation	0/75
Teams.exe	17 776 K	22 892 K	10132 Microsoft Teams	Microsoft Corporation	0/75
Teams.exe	23 324 K	21 756 K	8996 Microsoft Teams	Microsoft Corporation	0/75
Teams.exe	11 856 K	15 260 K	14132 Microsoft Teams	Microsoft Corporation	0/75

Figure 8 Le score fourni après le scan avec virustotal.

- Celui-ci correspond au nombre d'antivirus qui ont interagi avec le fichier scanner et com= bien détermine ce fichier comme étant potentiellement nocif.

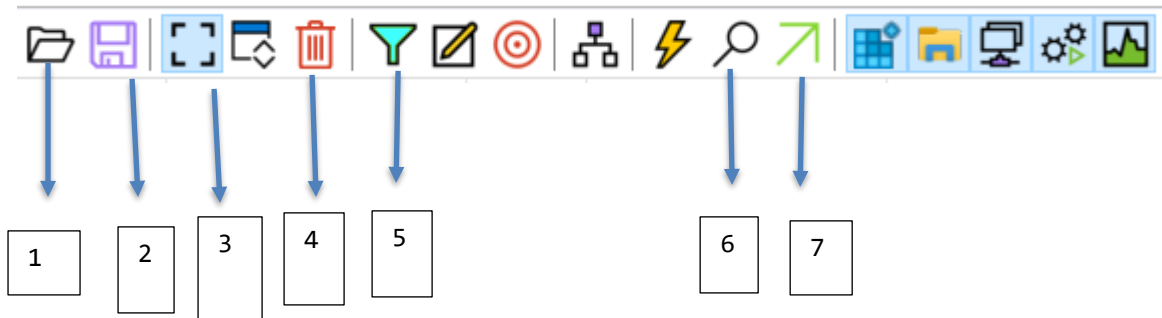
## PROCESS MONITOR

- En effet nous avons pu voir que Process Explorer permettait de lister les processus en cours d'exécutions à un poste, cependant il serait maintenant intéressant de pouvoir analyser le comportement de ces processus, et pouvoir interagir avec eux en cas d'urgence. Pour cela on utilisera ici Process Monitor, grâce à cet outil que nous pourrions voir quand et comment certains fichiers ont été modifiés par exemple. C'est donc un outil très intéressant, car il nous permettra de définir si un processus est malveillant ou non.
- Voici une capture montrant l'interface principal de process Monitor :



Figure 9 L'interface principale de Process Monitor.

- Voici maintenant une capture expliquant les différents outils qui seront amenés à être utilisés fréquemment dut la manipulation :



- ⇒ L'outil 1 permet de sélectionner un fichier pour une analyse.
  - ⇒ L'outil 2 permet d'enregistrer la capture d'événement.
  - ⇒ L'outil 3 permet de lancer une capture d'évènements.
  - ⇒ L'outil 4 permet de supprimer une capture d'évènements.
  - ⇒ L'outil 5 permet d'ajouter des filtres.
  - ⇒ L'outil 6 permet d'effectuer une recherche.
  - ⇒ L'outil 7 permet d'accéder à un objet (un élément du système).
- Il me paraît important d'aborder le sujet de l'application des filtres à fin que vous sachiez correctement les appliquer. En effet l'application des filtres vous permettra d'afficher seulement certains événements de la capture et non la capture complète, ce qui vous permettra une meilleure visibilité. Grâce au filtre il est possible de faire le tri entre les événements de lecture et les événements d'écriture, mais aussi de masquer certains événements.

## VI) CAS 1 : MODIFICATION DU HOME PAGE DU NAVIGATEUR FIREFOX.

### 1.1 CONTEXTUALISATION.

- Les employés de l'entreprise Vinci, qui est un client de NSI, se connectent à l'extranet en utilisant un nom d'utilisateur et un mot de passe via une page de connexion en ligne. Cependant, bien que la page de connexion semble normale, l'URL d'accès à cette page est légèrement différente et pourrait tromper certains employés de l'entreprise qui ne remarqueraient pas la différence. Une fois que les identifiants sont entrés, un pirate peut accéder à l'information à sa guise, et l'utilisateur est redirigé vers le vrai extranet de Vinci. Cette page de connexion a été définie comme étant la page par défaut pour le navigateur Web Mozilla Firefox.

### 1.2) PRATIQUE :

- Dans un premier temps la question à se poser est comment repérer le piège. Pour cela il suffit de se rendre dans les paramètres de Firefox, puis de vérifier que l'URL de la page d'accueil n'a pas été falsifiée en la comparant avec l'original :

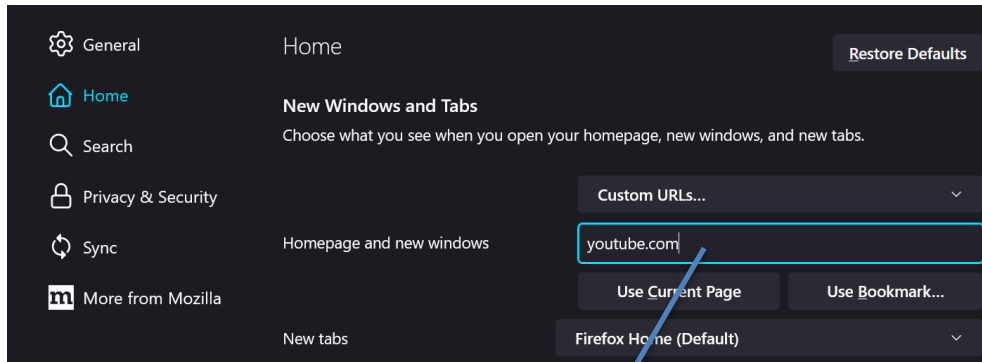


Figure 10 Modification du home page.

URL falsifiée.

- Si effectivement on remarque une anomalie, il faut alors intercepter la modification. L'utilisation de process Explorer permettra de capturer le processus responsable de ce changement.
- Dans un premier temps on passe par la réalisation d'une capture d'événements :

```

9368 CreateFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs.js
9368 CreateFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 CreateFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release
9368 CreateFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js
9368 WriteFile C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs-1.js

```

Figure 11 Capture d'événements.

- On peut voir ici que Firefox effectue des actions de création puis d'écritures dans un fichier « prefs.js », ce fichier correspond aux préférences, il se situe dans le répertoire du profile de l'utilisateur, et détermine-le profile Firefox par défaut.
- Maintenant que nous avons amassé suffisamment d'informations, nous pouvons passer à l'étape de création du « piège ». Nous avons vu que Firefox enregistre les préférences de l'utilisateur, le piège doit donc capturer le ou les processus susceptibles de modifier le fichier « pref.js ». Pour cela il suffit de changer quelque filtre tel que le filtre process Name ayant pour valeur firefox.exe, il nous faut le passer en exclusion, on ajoute en suite un filtre path avec comme condition contient et comme valeur prefs.js, en effet cela permettra de visualiser toute la modification apportée à prefs.js.
- La configuration des filtres devrait normalement ressembler à la figure ci-dessous :

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 Na...	is	firefox.exe	Exclude

Figure 12 Configuration des filtres.

- À présent on peut redémarrer la capture et modifier le fichier pref.js manuellement en guise de test. Pour cela il suffit de suivre le chemin suivant : C :

- 
- \Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release.  
Une fois le fichier pref.js retrouver je l'ouvrirais dans mon cas avec notepad++.
- La figure ci-dessous montre un aperçu de la contenu du fichier :

```

1 // Mozilla User Preferences
2
3 // DO NOT EDIT THIS FILE.
4 // If you make changes to this file while the application is running,
5 // the changes will be overwritten when the application exits.
6 //
7 // To change a preference value, you can either:
8 // - modify it via the UI (e.g. via about:config in the browser); or
9 // - set it within a user.js file in your profile.
10
11
12 user_pref("app.installation.timestamp", "133189644726376517");
13 user_pref("app.normandy.first_run", false);
14 user_pref("app.normandy.migrateToNewVersion", {});
15 user_pref("app.normandy.user_id", "f0f7f175-0a31-44a2-bf91-1ea7caa93937");
16 user_pref("app.update.background.lastInstalledVersion", {});
17 user_pref("app.update.background.lastUpdateAttempt", {});
18 user_pref("app.update.background.lastUpdateTime", {});
19 user_pref("app.update.background.previousRelease", {});
20 user_pref("app.update.background.updateAttempt", {});
21 user_pref("app.update.background.updateTime", {});
22 user_pref("app.update.background.updateTime.addon-background-update-time", {});
23 user_pref("app.update.background.updateTime.addon-update-time", {});
24 user_pref("app.update.background.updateTime.browser-update-time", {});
25 user_pref("app.update.background.updateTime.extension-update-time", {});
26 user_pref("app.update.background.updateTime.plugin-update-time", {});
27 user_pref("app.update.background.updateTime.update-time", {});
28 user_pref("app.update.background.updateTime.update-time", {});
29 user_pref("app.update.background.updateTime.update-time", {});
30 user_pref("app.update.background.updateTime.update-time", {});
31 user_pref("app.update.background.updateTime.update-time", {});
32 user_pref("app.update.background.updateTime.update-time", {});
33 user_pref("app.update.background.updateTime.update-time", {});
34 user_pref("app.update.background.updateTime.update-time", {});
35 user_pref("app.update.background.updateTime.update-time", {});
36 user_pref("app.update.background.updateTime.update-time", {});
37 user_pref("app.update.background.updateTime.update-time", {});
38 user_pref("app.update.background.updateTime.update-time", {});
39 user_pref("app.update.background.updateTime.update-time", {});
40 user_pref("app.update.background.updateTime.update-time", {});
41 user_pref("app.update.background.updateTime.update-time", {});
42 user_pref("app.update.background.updateTime.update-time", {});
43 user_pref("app.update.background.updateTime.update-time", {});
44 user_pref("app.update.background.updateTime.update-time", {});
45 user_pref("app.update.background.updateTime.update-time", {});
46 user_pref("app.update.background.updateTime.update-time", {});
47 user_pref("app.update.background.updateTime.update-time", {});
48 user_pref("app.update.background.updateTime.update-time", {});
49 user_pref("app.update.background.updateTime.update-time", {});
50 user_pref("app.update.background.updateTime.update-time", {});
51 user_pref("app.update.background.updateTime.update-time", {});
52 user_pref("app.update.background.updateTime.update-time", {});
53 user_pref("app.update.background.updateTime.update-time", {});
54 user_pref("app.update.background.updateTime.update-time", {});
55 user_pref("app.update.background.updateTime.update-time", {});
56 user_pref("app.update.background.updateTime.update-time", {});
57 user_pref("app.update.background.updateTime.update-time", {});
58 user_pref("app.update.background.updateTime.update-time", {});
59 user_pref("app.update.background.updateTime.update-time", {});
60 user_pref("app.update.background.updateTime.update-time", {});
61 user_pref("app.update.background.updateTime.update-time", {});
62 user_pref("app.update.background.updateTime.update-time", {});
63 user_pref("app.update.background.updateTime.update-time", {});
64 user_pref("app.update.background.updateTime.update-time", {});
65 user_pref("app.update.background.updateTime.update-time", {});
66 user_pref("app.update.background.updateTime.update-time", {});
67 user_pref("app.update.background.updateTime.update-time", {});
68 user_pref("app.update.background.updateTime.update-time", {});
69 user_pref("app.update.background.updateTime.update-time", {});
70 user_pref("app.update.background.updateTime.update-time", {});
71 user_pref("app.update.background.updateTime.update-time", {});
72 user_pref("app.update.background.updateTime.update-time", {});
73 user_pref("app.update.background.updateTime.update-time", {});
74 user_pref("app.update.background.updateTime.update-time", {});
75 user_pref("app.update.background.updateTime.update-time", {});
76 user_pref("app.update.background.updateTime.update-time", {});
77 user_pref("app.update.background.updateTime.update-time", {});
78 user_pref("app.update.background.updateTime.update-time", {});
79 user_pref("app.update.background.updateTime.update-time", {});
80 user_pref("app.update.background.updateTime.update-time", {});
81 user_pref("app.update.background.updateTime.update-time", {});
82 user_pref("app.update.background.updateTime.update-time", {});
83 user_pref("app.update.background.updateTime.update-time", {});
84 user_pref("app.update.background.updateTime.update-time", {});
85 user_pref("app.update.background.updateTime.update-time", {});
86 user_pref("app.update.background.updateTime.update-time", {});
87 user_pref("app.update.background.updateTime.update-time", {});
88 user_pref("app.update.background.updateTime.update-time", {});
89 user_pref("app.update.background.updateTime.update-time", {});
90 user_pref("app.update.background.updateTime.update-time", {});
91 user_pref("app.update.background.updateTime.update-time", {});
92 user_pref("app.update.background.updateTime.update-time", {});
93 user_pref("app.update.background.updateTime.update-time", {});
94 user_pref("app.update.background.updateTime.update-time", {});
95 user_pref("app.update.background.updateTime.update-time", {});
96 user_pref("app.update.background.updateTime.update-time", {});
97 user_pref("app.update.background.updateTime.update-time", {});
98 user_pref("app.update.background.updateTime.update-time", {});
99 user_pref("app.update.background.updateTime.update-time", {});
100 user_pref("app.update.background.updateTime.update-time", {});

```

Figure 13 Contenu du fichier pref.js.

- Une fois la modification enregistrée et effectuée, on peut voir que de nouveaux événements sont affichés sur la capture :

PID	Operation	Path
2084	CreateFile	C:\Users\DELL\AppData\Roaming\Notepad++\backup\prefs.js@2023-02-10_160443
2084	WriteFile	C:\Users\DELL\AppData\Roaming\Notepad++\backup\prefs.js@2023-02-10_160443
2084	WriteFile	C:\Users\DELL\AppData\Roaming\Notepad++\backup\prefs.js@2023-02-10_160443
2084	CreateFile	C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs.js
2084	WriteFile	C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs.js
2084	WriteFile	C:\Users\DELL\AppData\Roaming\Mozilla\Firefox\Profiles\cb2awp5a.default-release\prefs.js
2084	SetDisposition...	C:\Users\DELL\AppData\Roaming\Notepad++\backup\prefs.js@2023-02-10_160443

Figure 14 Capture des nouveaux évènements.

- Effectivement cet événement correspond bien à une modification dans le fichier prefs.js avec le logiciel notepad++, on peut donc en déduire que notre piège fonctionne.

## VII) CAS 2 : EMPOISONNEMENT PAR DNS.

---

### 1.1) CONTEXTUALISATION

- L'équipe du service d'assistance du premier niveau aussi appelé helpdesk 1 a reçu une demande de support enregistré sous forme de billet d'incident. En effet un client de Vinci a signalé une plainte concernant la modification de la page de connexion à l'extranet de l'entreprise. Il s'agit en réalité d'un cas d'empoisonnement DNS.

### 1.2) LE BUT RECHERCHE

- Ici il peut être intéressant de se glisser dans la peau de l'attaquant pour se demander quel est le but de l'attaque.  
Le but principal d'une attaque par empoisonnement DNS est de rediriger les utilisateurs vers des sites Web malveillants ou non fiables en falsifiant les informations de la base de données DNS. Cela peut se produire lorsqu'un attaquant parvient à modifier les entrées DNS pour faire correspondre une adresse IP légitime à un nom de domaine malveillant. Les utilisateurs qui cherchent à accéder au site Web légitime peuvent être redirigés vers un site dangereux qui peut voler leurs informations personnelles, diffuser des logiciels malveillants ou simplement afficher du contenu inapproprié. C'est pourquoi il est important de maintenir la sécurité de votre système DNS et de surveiller les modifications apportées à votre base de données DNS pour prévenir ce type d'attaque.

### 1.3) METHODE DE DETECTION

- Il faut maintenant se poser la question « comment repérer un DNS empoisonné ? » Tout d'abord un DNS empoisonné se caractérise par une IP différente de l'IP du DNS souhaité. Dans un premier temps, il faut donc réussir à déterminer l'emplacement de sauvegarde des paramètres DNS de Windows, on notera que ces paramètres DNS correspondent à une carte réseau spécifique. Un autre point à prendre en compte est que ces paramètres peuvent être stockés sous différentes formes tels que des clés de registre ou un fichier de configuration.
- Afin de déterminer l'emplacement de ces paramètres, nous utiliserons Process Explorer.

## 1.4) ÉTUDE DES PARAMETRES DU DNS

- Pour débiter l'étude de ces paramètres, nous devons d'abord trouver l'emplacement de ceux-ci, pour cela les filtres proposés sur process Explorer nous seront d'une grande utilité. Ici étant donné que les paramètres DNS sont des paramètres de bas niveau, on peut en déduire, qu'ils seront stockés sous forme de valeur dans une clé de registre.

## 1.5) LA POSE DES FILTRES

- Dans un premier temps l'application de deux filtres sera nécessaire
  - ⇒ Filtre 1 : type catégorie/valeur write.
  - ⇒ Filtre 2 : Operation/ Valeur RegSetValue.
- La capture ci-dessous démontre la configuration des filtres nécessaire :

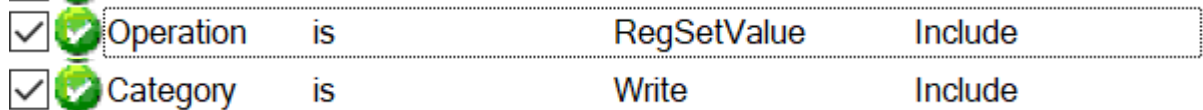


Figure 15 Modification de La configuration des filtres.

- Dans notre exemple, afin que des modifications soient enregistrées, on effectuera des modifications sur nos propres paramètres DNS. Pour cela, se rendre le panneau de configuration, puis modifier les paramètres de la carte, sélectionner le protocole « internet version 4 (TCP/IPv4), les figures ci-dessous vous montre le bon chemin à suivre :

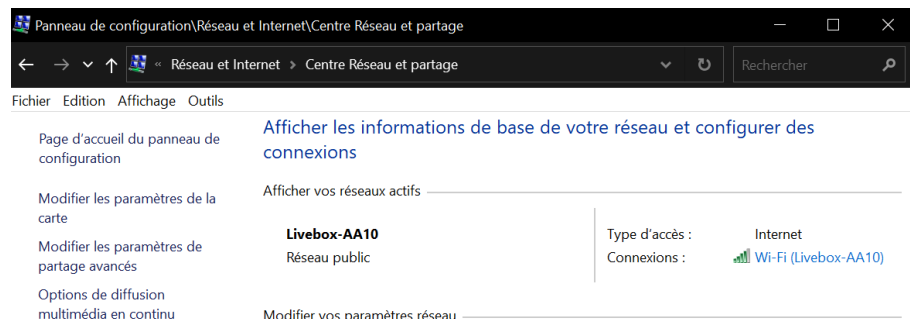


Figure 16 Le chemin à suivre pour la modification des paramètres DNS.

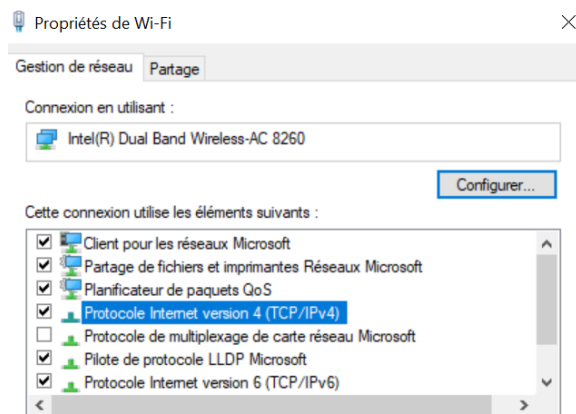


Figure 17 Propriété Wifi.



- Nous avons maintenant accès aux paramètres DNS et nous pouvons les modifier avant toute chose, lancer votre capture d'événements sur Process Monitor, puis prenez soin de cocher l'option "utiliser l'adresse de serveur DNS suivante" puis "ok" :

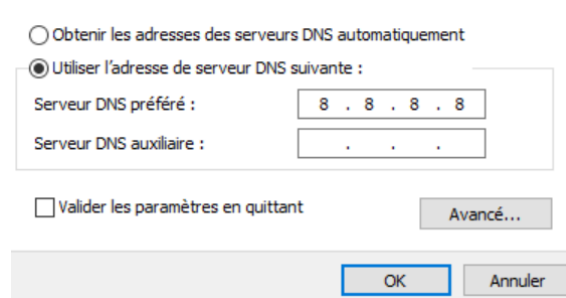




Figure 18 Insertion d'une nouvelle adresse.

- Revenez maintenant sur process Monitor et stoppé votre capture d'événements, celle-ci devrait normalement ressembler à la figure ci-dessous :

Time o...	Process Name	PID	Operation	Path
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\ACPI\PNP0C14\0_0-[05901221-D566-11d1-B2F0-00A0C9062910]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\system32\kernelbase.dll[MofResourceName]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\system32\fr-FR\kernelbase.dll.mu[MofResourceName]
17:11:0...	ctfmon.exe	11460	RegSetValue	HKCU\SOFTWARE\Microsoft\Input\Typing\Insights\Insights
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\ACPI.sys[ACPI\MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\fr-FR\ACPI.sys.mu[ACPI\MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\system32\drivers\indis.sys[MofResourceName]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\system32\drivers\fr-FR\indis.sys.mu[MofResourceName]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\system32\drivers\battc.sys[BATT\WMI]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\msmbios.sys[MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\fr-FR\msmbios.sys.mu[MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\HDAudio.sys[HDAudio\MofName]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\processr.sys[PROCESSOR\WMI]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\fr-FR\processr.sys.mu[PROCESSOR\WMI]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\wmiaapl.sys[MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\portcls.sys[Ports\Mof]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\lthpan.sys[Ndis\MofResource]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\DriverStore\FileRepository\hcd\inf_amd64_a54e540558404ee5\...
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\lthhenum.sys[MofResourceName]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\monitor.sys[Monitor\WMI]
17:11:0...	wmiprvse.exe	1108	RegSetValue	HKLM\SOFTWARE\Microsoft\Wbem\WDM\C:\Windows\System32\drivers\esif_if.sys[Esif\WMI]
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\TraySearchBoxVisible
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Search\TraySearchBoxVisibleOnAnyMonitor
17:11:0...	ctfmon.exe	11460	RegSetValue	HKCU\SOFTWARE\Microsoft\Input\Typing\Insights\Insights
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ActivityDataModel\ReaderRevisionInfo\7BC4F56B-9D63-0EB4-E793-C6574...
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\ActivityDataModel\ReaderRevisionInfo\7BC4F56B-9D63-0EB4-E793-C6574...
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\FeatureUsage\AppSwitched\Microsoft.Windows.ControlPanel
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count...
17:11:0...	Explorer.EXE	12076	RegSetValue	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count...
17:11:0...	ctfmon.exe	11460	RegSetValue	HKCU\SOFTWARE\Microsoft\Input\Typing\Insights\Insights

Figure 19 Capture d'événements après modification du DNS.

- Grâce à l'outil de recherche :  et l'outil permettant de n'afficher que les événements en rapport avec le registre :  nous pouvons écarter un grand nombre d'affichages inutiles, dans l'outil de recherche, il faut saisir NameServer :

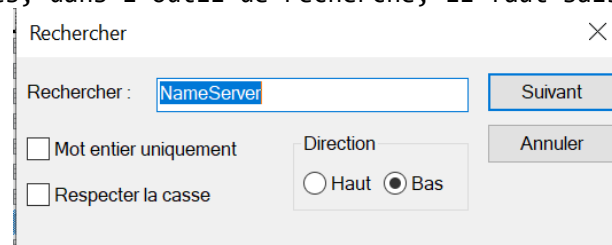


Figure 20 Recherche NameServer.

- Pourquoi cette recherche ? C'est tout simplement, car la valeur ServerName contient les adresses des DNS paramétrés pour chaque interface. Maintenant afin de savoir si une des valeurs du registre a été modifiée, il suffit de regarder dans la colonne détail et d'y trouver la mention data, on vérifie si cette mention contient bien la valeur que nous avons entrée précédemment :



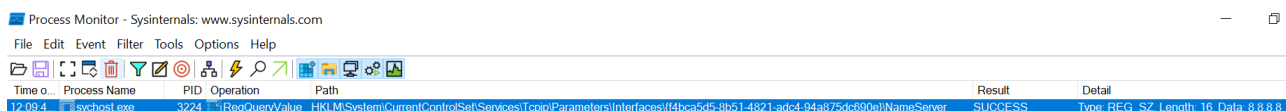


Figure 21 Vérification de La mention DATA.

- Comme le démontre cette figure, on retrouve bien la valeur 8.8.8.8, nous pouvons en déduire que nous avons trouvé la bonne capture.
- En conclusion pour ce cas d'attaque, Process Monitor permet d'établir des filtres, et ces filtres permettent d'identifier les processus à l'origine de la modification. À partir de là, il est possible de nettoyer la machine.

## IIX) LA DETECTION DE PROGRAMME MALVEILLANT.

- Dans cette partie, nous nous intéresserons aux différentes méthodes de détection de virus. Dans cette partie, nous nous servirons de l'outil virus total, celui-ci est directement accessible par le lien suivant : <https://www.virustotal.com/gui/home/upload>.

### 1.1) COMPRENDRE LE FONCTIONNEMENT DE VIRUS TOTAL.

- Afin de pouvoir utiliser convenablement le virus total, il est nécessaire de comprendre son fonctionnement. Cet outil utilise une combinaison de plusieurs méthodes de détection telles que :
  - L'analyse de signature => cette méthode utilise des bases de données de signatures de virus pour rechercher des correspondances dans les fichiers soumis.
  - L'analyse heuristique => cette méthode permet d'analyser et d'examiner le comportement d'un fichier pour déterminer si oui ou non il est potentiellement malveillant.
  - Sandboxing => cette méthode consiste en une exécution de tous les fichiers soumis dans un environnement isolé pour observer leur comportement et identifier les activités malveillantes.
  - Machine learning => cette méthode utilise des algorithmes d'apprentissage automatique pour classer les fichiers en fonction de leur sécurité.
- En définitive, vous l'aurez compris l'outil virus total de cette base sur une analyse multimoteur, c'est-à-dire que plusieurs moteurs de détection sont utilisés en parallèle pour examiner un fichier ou un logiciel et permet de fournir une analyse complète de sa sécurité. Le fait d'utiliser une analyse multimoteur améliore la précision de détection des logiciels malveillants et minimise les faux positifs. Cette approche permet également de détecter des menaces qui pourraient être manquées par un moteur de détection individuel.
- La figure ci-dessous présente la page d'accueil de virus total :



Figure 22 Page d'accueil de virustotal.

- On peut voir ici que virus total nous offre 3 possibilités de scan :

- ⇒ Le sans de fichiers : en effet ici il est possible d'envoyer directement le fichier souhaité a virus total pour que celui-ci le scan, il vous suffira de cliquer sur file puis sur chose file pour sélectionner le fichier désiré.
- Voici la présentation d'une courte démonstration avec un fichier nommé test.txt contenant l'empreinte EICAR. On notera ici que l'emprunte EICAR de son nom complet European Institute for computer antivirus research est utilisé pour vérifier la capacité d'un logiciel antivirus à détecter les logiciels malveillants, il s'agit donc d'un fichier inoffensif conçu pour être détecté comme un virus par les logiciels antivirus.
  - Après avoir scanné le fichier, voici le résultat :

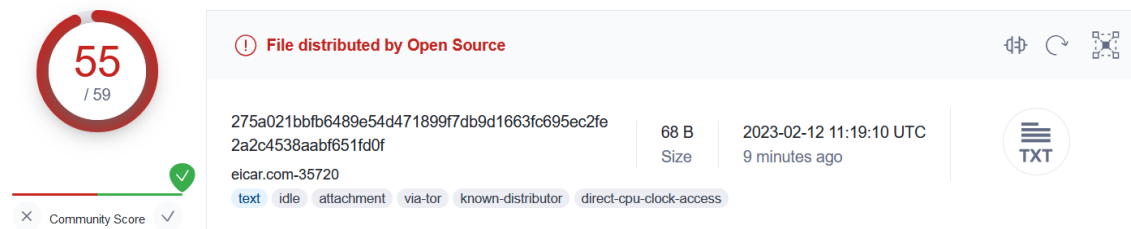


Figure 23 Résultat du scan de fichier.

- On peut voir ici que plusieurs des moteurs ont détecté l'empreinte EICAR comme le montre la figure ci-dessous :

AhnLab-V3	⚠ Virus/EICAR_Test_File	Alibaba	⚠ Trojan:MacOS/eicar.com
ALYac	⚠ Misc.Eicar-Test-File	Antiy-AVL	⚠ TestFile/Win32.EICAR
Arcabit	⚠ EICAR-Test-File (not A Virus)	Avast-Mobile	⚠ Eicar
Avira (no cloud)	⚠ Eicar-Test-Signature	BitDefender	⚠ EICAR-Test-File (not A Virus)
BitDefenderTheta	⚠ EICAR-Test-File (not A Virus)	Bkav Pro	⚠ W32.EicarTest.Trojan

Figure 24 Détection de L'empreinte EICAR.

- On constate donc que le virus total est réceptif à notre test.
- ⇒ Le scan de site internet : En effet nous pouvons également scanner un site web, pour cela il suffit de sélectionner l'option "URL" et de renseigner l'URL correspondant au site souhaité.
- Voici un test illustrant mes propos, ici je vais scanner l'URL du site virus total <https://www.virustotal.com/gui/url/analysis> et voici le résultat :

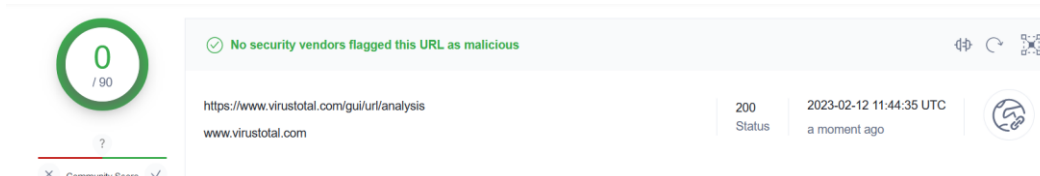


Figure 25 Résultat du scan de site web.

- Ici on peut voir qu'aucune menace n'a été détectée, l'ite est sûre.
- ⇒ La fonctionnalité de recherche de virus total propose d'effectuer une recherche permettant de vérifier si un programme ou un fichier a déjà été répertorié dans la base de données de virus total.

## 1.2) QUE PENSER DE VIRUSTOTAL ?

- En somme, virus total est un outil puissant et gratuit qui s'avère être très utile pour la détection de logiciels malveillants et l'analyse de la sécurité des fichiers et des URL. Il utilise plusieurs moteurs de détection pour offrir une analyse complète et améliorer la précision de la détection des logiciels malveillants. Cependant, il est important de noter que les résultats de l'analyse peuvent varier en fonction des différents moteurs de détection utilisés et il est toujours recommandé de consulter plusieurs sources pour obtenir une image complète de la sécurité d'un fichier ou d'une URL. De plus, il est essentiel de maintenir à jour votre logiciel antivirus et de prendre des mesures de sécurité supplémentaires pour protéger votre ordinateur et vos informations personnelles. En utilisant VirusTotal en combinaison avec d'autres outils et pratiques de sécurité, vous pouvez améliorer la protection de votre ordinateur contre les logiciels malveillants et les autres menaces en ligne.

## IX) AUTOMATISATION DE LA RECHERCHE DE MENACES.

---

- En effet, pour un professionnel, il serait très intéressant de pouvoir automatiser la recherche de menaces. Pour cela, nous aurons besoin d'un script et d'une API.

### 1.1) L'API.

- Pour commencer, une API ou interface de programmation d'applications est un ensemble de règles et de spécifications qui définissent comment deux systèmes informatiques peuvent interagir entre eux. Elle décrit les fonctionnalités et les services proposés par un logiciel et la manière de les utiliser. Une API permet à un développeur de créer des applications qui s'intègrent à un autre logiciel existant en utilisant les fonctionnalités proposées par celle-ci. Par exemple une API peut permettre à un développeur de créer une application qui accède aux données d'un service en ligne. En général, un API est utilisé pour faciliter la communication et l'intégration entre différents systèmes et applications.
- Il existe de nombreux types d'API, ici celle qui nous intéressera sera l'API REST. C'est un style d'architecture pour les services web qui utilise le protocole HTTP afin de communiquer et échanger des données. Les APIs rest utilisent souvent le format json (Java Script), celui-ci est plus léger et facile à manipuler. Les apis rest permettent aux développeurs de créer des services web évolutifs, flexibles et facilement accessibles à partir de différents types de clients, tels que des applications web, mobiles ou IoT (Internet of Things).

### 1.2) LE SCRIPT.

- En Informatique un script représente un ensemble d'instructions qui permettent d'automatiser une ou plusieurs tâches. Pour la création de scripts, il existe des langages de programmation différents tels que le bash ou le python, ceux qui sont deux des langages les plus couramment utilisés.
- Ici nous porterons un intérêt particulier pour les scripts, car ils nous permettront d'effectuer de manière répétée une suite d'instructions, cela se révélera être utile pour l'automatisation de la recherche de menaces.

### 1.3) ROBOT VIRUS TOTAL.

- Après avoir abordé toutes les notions nécessaires ci-dessus, nous pouvons maintenant commencer à réfléchir à un moyen d'automatiser la recherche de menaces sur le service virus total. En cherchant bien, on se rend compte que Virus total propose une API rest. Vous pourrez trouver une documentation explicite en rapport avec cette API grâce à l'URL suivante : <https://developers.virustotal.com/reference/overview>.
- Maintenant que nous connaissons l'existence de cette API, il nous faut un programme nous permettant de dialoguer avec cette API. Il ne nous reste donc plus qu'à nous renseigner sur les bibliothèques de langages disponibles sur virustotal. Ici nous choisirons Python pour coder notre robot. Il nous faut donc installer l'extension python, pour cela consultez l'URL suivante : <https://www.python.org/downloads/> Puis télécharger la dernière version pour Windows.
- Vous pouvez maintenant passer à l'installation de la bibliothèque python vt-py. Cette bibliothèque permettra de fournir une interface pour interagir avec l'API de virus total afin d'effectuer les scans voulus. Pour installer cette bibliothèque, il vous faut saisir la commande suivante "py - m pip install vt-py" dans une invite de commande en mode administrateur. Maintenant afin de lier votre script avec virustotal, il vous faut générer une clef API, pour cela rendez-vous sur le site

virus total, créez un compte en cliquant sur le bouton **Sign up** . Une fois que cette action est finalisée, cliquez sur votre profil : pierre barbier puis sur **API key** .

- Il ne vous reste maintenant plus qu'à copier votre clef API :

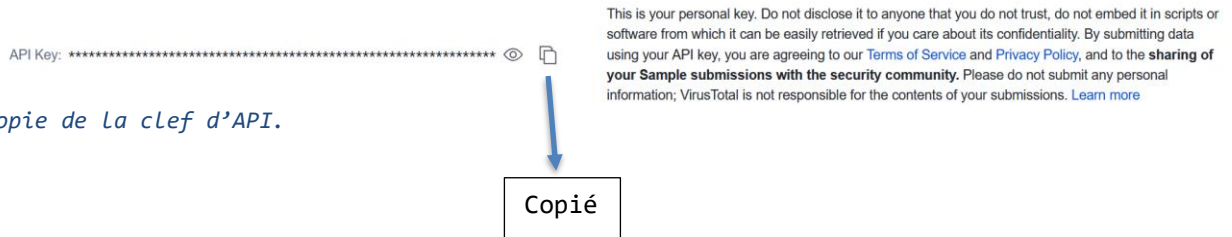


Figure 26 Copie de la clef d'API.

- Maintenant que vous possédez tous les outils nécessaires, nous pouvons maintenant passer à la conception du script du robot. Ici je choisis d'exécuter mon script dans l'environnement python de mon terminal, pour cela il me suffit de lancer une invite de commande en mode privilégié puis de saisir « py ». La figure ci-dessous caractérise un robot permettant le scan d'un fichier grâce à virus total et l'affichage du résultat du scan de manière ordonné afin de faciliter la lecture pour l'utilisateur :

```
>>> import vt
>>> import os
>>> import time
>>> import json
>>>
>>> API_KEY = '0196010fd366852a1e939a0fe72a0ffdb96fcc08683a2b99755afbf864aeb62d'
>>> folder_path = 'C:\\Users\\DELL\\Desktop\\test'
>>> def scan_file(file_path):
...     with open(file_path, 'rb') as f:
...         client = vt.Client(API_KEY)
...         analysis = client.scan_file(f, wait_for_completion=True)
...         analysis = client.get_object("/analyses/{}", analysis.id)
...         sorted_results = json.dumps(analysis.results, sort_keys=True, indent=4)
...         print(sorted_results)
...         return analysis
...
>>> for file_name in os.listdir(folder_path):
...     file_path = os.path.join(folder_path, file_name)
...     if os.path.isfile(file_path):
...         print(f"Scanning {file_name}")
...         analysis = scan_file(file_path)
...         time.sleep(15)
```

Figure 27 Code du robot VirusTotal.

- Voici un pseudo-code illustrant le script ci-dessus :
  - Importer les modules nécessaires : vt, os, time, json.
  - Définir la clé d'API VirusTotal et le chemin du dossier contenant les fichiers à analyser.
  - Créer une fonction pour analyser un fichier :
  - Ouvrir le fichier en mode binaire.
  - Créer un client VirusTotal.
  - Soumettre le fichier pour analyse et attendre que l'analyse soit terminée.
  - Récupérer les résultats de l'analyse.
  - Trier les résultats par clé et formatez-les en JSON.
  - Afficher les résultats.
  - Renvoyer l'objet d'analyse.
  - Parcourir tous les fichiers dans le dossier spécifié :
  - Créer le chemin complet du fichier.
  - Vérifier si le chemin correspond à un fichier.
  - Afficher le nom du fichier en cours d'analyse.
  - Analyser le fichier.
  - Attendre 15 secondes avant de passer au fichier suivant
- La figure suivante présente un aperçu du résultat du scan :

```
{
  "ALYac": {
    "category": "undetected",
    "engine_name": "ALYac",
    "engine_update": "20230214",
    "engine_version": "1.1.3.1",
    "method": "blacklist",
    "result": null
  },
  "APEX": {
    "category": "type-unsupported",
    "engine_name": "APEX",
    "engine_update": "20230213",
    "engine_version": "6.387",
    "method": "blacklist",
    "result": null
  },
  "AVG": {
    "category": "undetected",
    "engine_name": "AVG",
    "engine_update": "20230214",
    "engine_version": "22.11.7701.0",
    "method": "blacklist",
    "result": null
  },
  "Acronis": {
    "category": "undetected",
    "engine_name": "Acronis",
    "engine_update": "20221114",
    "engine_version": "1.2.0.113",
    "method": "blacklist",
    "result": null
  },
  "AhnLab-V3": {
    "category": "undetected",
    "engine_name": "AhnLab-V3",
    "engine_update": "20230214",
    "engine_version": "3.23.1.10344",
    "method": "blacklist",
    "result": null
  }
}
```

Figure 28 Résultat du scan du robot.

- On peut voir ici qu'il y a différents rendus d'analyse, à travers les différents moteurs qu'utilise virustotal pour la détection des fichiers Vireaux. En effet un rendu d'analyse se présente sous la forme suivante :
  - ⇒ La « category » présente si le fichier scanner présente une menace ou non.
  - ⇒ « Engine\_name » présente le nom du moteur qui a effectué le scan.
  - ⇒ « Engine\_update » permet de fixer la date de mise à jour du moteur de recherche.
  - ⇒ « Engine\_version » permet de fixer la version du moteur.
  - ⇒ « Method » permet de fixer la méthode qui a été utilisée pour le scan.

⇒ « Result » permet de fixer si une menace a été trouvée.

- Il vous est possible de trouver ce script dans le dossier SRC présent dans le build, il se présentera sous le nom de « robot\_virustotal »

#### 1.4) LES FICHIERS DE CONFIGURATIONS.

- Dans cette partie, nous parlerons du fichier de configuration. Celui-ci est au format. YAML. Un fichier de configuration est un fichier qui contient des données de configurations pour une application ou un programme. Dans notre cas, c'est-à-dire notre robot virus total spécialisé pour le scan de fichier, ce fichier peut contenir des informations telles que les paramètres de connexion à l'API de l'outil de scan, les chemins d'accès pour les fichiers à scanner et enfin les répertoires à ignorer lors du scan et même la fréquence de numérisation.
- On notera que l'utilisation d'un fichier de configuration a toute son importance, car cela permet de séparer les données de configurations du code de l'application, ce qui a pour effet de rendre le code modulaire et plus facile à maintenir. En somme les fichiers.yaml sont faciles à lire et à éditer, et cela même pour les utilisateurs non techniques. Ainsi en utilisant le fichier de configuration, les utilisateurs peuvent très facilement changer le comportement du robot de scan sans avoir à modifier le code de celui-ci.
- Dans le cadre de notre robot, la figure ci-dessous montre un exemple de fichier de configuration qui pourrait convenir :



```
! conf.yaml X
C: > Users > DELL > Desktop > BARBIER_PIERRE_V2 > src > ! conf.yaml
1  api_key: '0196010fd366852a1e939a0fe72a0ffdb96fcc08683a2b99755afbf864aeb62d'
2  folder_path: 'C:\\Users\\DELL\\Desktop\\BARBIER_PIERRE_V2\\src\\test'
3  sleep_time: 15
4  file_types:
5      txt
6      docx
7      pdf
8  ignore_dirs:
9      C:\\Users\\DELL\\Desktop\\ BARBIER_PIERRE_V2\\src\\test\\ignore
10     C:\\Users\\DELL\\Desktop\\BARBIER_PIERRE_V2\\src\\test\\ignore2
11 recursive: true
```

Figure 29 Contenu du fichier de configuration.

- Voici l'explication de chaque paramètre de configurations :
- ⇒ « Api\_key » : La clé d'API à utiliser pour se connecter à l'API de l'outil de scan.
- ⇒ « folder-path » : Le chemin d'accès au répertoire contenant les fichiers à scanner.
- ⇒ « sleep\_time » : Le temps en secondes à attendre entre chaque analyse de fichiers.
- ⇒ « files\_types » : Les types de fichiers à scanner, seuls les fichiers avec les extensions précisées dans le fichier de configuration seront pris en compte.
- ⇒ « ignore\_dirs » : Les répertoires à ignorer lors de la numérisation, tous les fichiers dans ce répertoire seront ignorés.
- ⇒ « récursive » : Indique si la numérisation doit être récursive, c'est-à-dire si les sous-répertoires doivent être également analysés.



- Une fois que le fichier de configuration est correctement rédigé, le code peut être modifié pour changer les valeurs à partir de ce fichier de configuration en utilisant la bibliothèque « PyYAML » pour lire les données YAML. Pour importer cette bibliothèque dans votre environnement python, voici la commande à saisir « pip install pyyaml ». La figure ci-dessous démontre que l'installation s'est correctement effectuée :

```
C:\Users\DELL>pip install pyyaml
Requirement already satisfied: pyyaml in c:\users\dell\appdata\local\packages\pythonsoftwarefoundat
[notice] A new release of pip available: 22.3.1 -> 23.0.1
[notice] To update, run: C:\Users\DELL\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation
```

Figure 30 Installation de La bibliothèque YAML.

- Il nous faut maintenant lier notre fichier de configuration a notre code python, pour cela il suffit de glisser le fichier YAML dans le même répertoire que notre script. Une fois ces actions terminées, il nous faut ajouter certaines choses à notre robot afin de terminer la liaison entre celui-ci est le fichier de configuration, la figure ci-dessous montre les éléments ajoutés au code :

```
import yaml

with open('conf.yaml', 'r') as f:
    config = yaml.safe_load(f)
```

Figure 31 Ajout du code nécessaire à La liaison du fichier de configuration et du script.

- Voici une explication de ces lignes de code :
  - ⇒ « import yaml » sert à faire appel à la bibliothèque « pyAML » installé précédemment.
  - ⇒ « with open » cette partie sert tous simplement à ouvrir puis lire la contenue du fichier de configuration et le charger dans le script python.
- La liaison entre le script et le fichier de configuration est maintenant terminée.
- Le fichier de configuration se présentera sous le nom de config. YAML et sera accessible dans le répertoire data du build.
- En somme un fichier yaml se construit avec des clés et des valeurs. Il est important de noter qu'une clé définit un paramètre, et sa valeur peut être vue comme un réglage.

### 1.5) CONCLUSION.

- En somme, on peut voir ici que le robot permet un gain de temps énorme, car celui-ci permet de scanner des fichiers de manières automatiques. Le script rédigé fournit un résultat cohérent para port aux fichiers scanner, on peut donc en conclure que tous fonctionnent bien.

## X) UNE ANALYSE DE FICHIER APPROFONDIE GRACE A LA SIGNATURE DE CODE.

---

### 1.1) CONTEXTUALISATION.

- Comme vu précédemment, il est possible de vérifier en se servant de VirusTotal, si un programme est malveillant ou non en analysant son contenu afin de déterminer si le code qu'il contient est malveillant ou non.
- Cependant, il existe un autre paramètre permettant d'identifier la provenance d'un fichier ainsi que sa nature, ce qui permet de se forger un avis sur la dangerosité ou non du programme. Pour ce faire, il nous faudra vérifier si le programme possède une signature numérique.

### 1.2) INTRODUCTION A LA SIGNATURE DE CODE, LE CAS PARTICULIER DES PE SOUS WINDOWS.

- Dans le cadre de notre automate python conçue précédemment pour le scan de fichier, il est important de noter que la détection de signature de code est une étape importante. En effet la signature de code est un processus de sécurité informatique qui consiste à ajouter une signature numérique à un programme exécutable tel qu'un logiciel ou une application. Cela permettra de garantir son intégrité et son authenticité. Quand on parle de garantir l'intégrité d'un fichier exécutable par la signature de code, cela signifie que la signature numérique permet d'assurer que le fichier n'a pas été modifié ou altéré depuis sa création.
- Un fichier PE appelé aussi « portable exécutable » est en fait une structure de fichier utilisée par d'autres fichiers tels que les .exe, ou DLL par exemple. Cette structure permet tout simplement le fonctionnement de ces fichiers sur n'importe quel poste utilisant le noyau Windows. On notera que le format PE possède plusieurs parties, voici les plus importantes :
  - L'en-tête MS-DOS permet de reconnaître le fichier comme étant un exécutable valide dans le cas où le fichier serait lancé dans un environnement MS-DOS.
  - Le segment DOS est exécuté si Windows ne reconnaît pas le fichier comme étant au format PE valide.
  - L'en-tête PE contient toutes les informations importantes en rapport avec le fichier, tel que la « signature » permettant d'identifier le fichier ou le « FileHeader », contenant les informations en rapport avec la structure du fichier.
  - La table des sections contient toutes les informations en rapport avec les différents binaires constituant le fichier, devant être chargée en mémoire.
  - Les sections contiennent différents types de données du programme, telles que le code à exécuter, les variables initialisées ou les ressources du fichier.

- La figure ci-dessous dénonce un schéma relatif à l'intégrité que permet la signature de code :

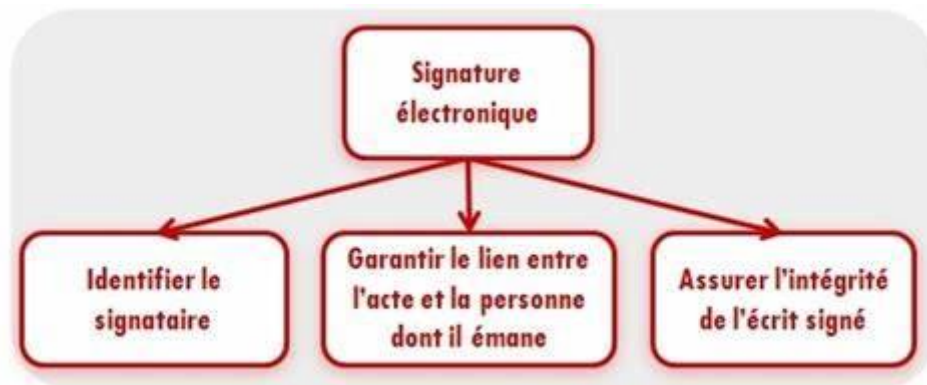


Figure 32 L'intégrité que permet la signature de code.

### 1.3) LA PRODUCTION ET LA LECTURE D'UNE SIGNATURE NUMERIQUE.

- En effet, la production d'une signature de code découle de la contenu du fichier exécutable en question, l'utilisation d'une fonction de hachage cryptographique permettra de générer une empreinte numérique unique. Cette empreinte sera par la suite chiffrée à l'aide d'une clé privée associée à un certificat numérique de l'éditeur ou du développeur.
- Il sera possible pour l'utilisateur, après le téléchargement de l'exécutable, de vérifier la signature numérique à l'aide d'une clé publique associée au certificat numérique. Si la signature numérique est valide, cela indique que le fichier exécutable n'a pas été modifié depuis sa création. La figure ci-dessous dénonce un schéma expliquant le calcul du hachage, le cryptage et le décryptage :

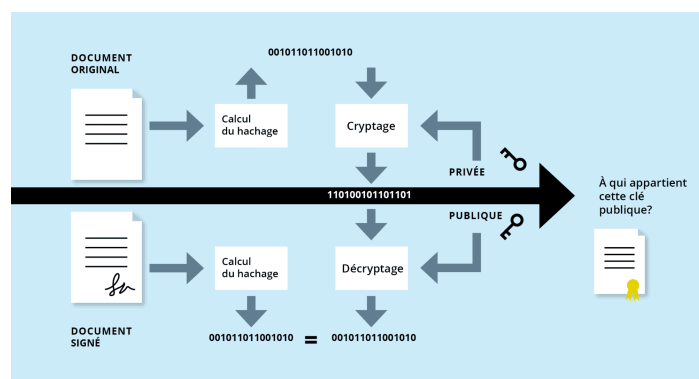


Figure 33, Le cryptage et Le décryptage des clefs.

- Pour plus d'informations relatives, consulter la documentation Microsoft sur le décryptage de signature de code accessible par l'URL suivante :

<https://docs.microsoft.com/fr-fr/windows-hardware/drivers/install/digital-signatures>

#### 1.4) UNE DIFFERENCE ENTRE UN EXECUTABLE PIRATE ET LEGITIME.

- En effet nous avons vu précédemment qu'une signature de code est un mécanisme de sécurité permettant de vérifier l'intégrité d'un fichier exécutable. Ce mécanisme peut s'avérer très utile pour différencier un exécutable pirate d'un exécutable légitime. Voici une explication complète du fonctionnement de processus :
  - ⇒ Le développeur crée un fichier exécutable (par exemple, une application ou un programme) et génère une signature numérique unique à partir de ce fichier. Cette signature est créée en utilisant une clé privée associée à un certificat numérique.
  - ⇒ Le certificat numérique est émis par une autorité de certification (CA), qui est une entité de confiance qui vérifie l'identité du développeur et garantit que la clé privée utilisée pour signer le fichier n'a pas été compromise.
  - ⇒ Lorsqu'un utilisateur télécharge le fichier exécutable, le système d'exploitation vérifie la signature numérique en utilisant la clé publique associée au certificat numérique. Si la signature est valide, le système d'exploitation affiche une confirmation que le fichier provient d'une source de confiance.
  - ⇒ Si la signature est invalide ou si le certificat numérique est expiré ou révoqué, le système d'exploitation avertit l'utilisateur que le fichier pourrait être dangereux et lui demande s'il souhaite l'exécuter.
- En somme, si l'exécutable possède une signature de code, il provient d'une source fiable et peut être considéré comme non viral.

#### 1.5) LES SIGNATURES DE CODE STANDARD ET EXTENDED VALIDATION.

- Bien entendu, il existe différentes signatures telles qu'une signature standard et une signature de code EV (extended validation). La différence entre ces deux signatures réside dans le niveau de vérification et la confiance associée à chaque type de signature.
- On notera qu'une signature de code standard utilise un certificat numérique émis par une autorité de certification (CA) pour vérifier l'identité de l'auteur du code. Cependant, le processus de vérification est relativement simple et ne nécessite généralement que des informations de base sur l'auteur du code.
- En revanche, une signature de code EV implique un processus de vérification plus approfondi pour garantir l'identité de l'auteur du code. Les autorités de certification qui délivrent des certificats EV doivent se conformer à des normes de vérification strictes définies par les normes du CA/Browser Forum. Ces normes exigent que l'autorité de certification vérifie l'identité de l'auteur du code en utilisant plusieurs sources d'information, notamment des registres publics, des bases de données gouvernementales et des contrôles de solvabilité.
- Les signatures de code EV sont donc considérées comme plus fiables et plus dignes de confiance que les signatures de code standard, car elles offrent un niveau de vérification plus élevé de l'identité de l'auteur du code. Cela est particulièrement important pour les logiciels critiques, tels que les logiciels de sécurité et les applications bancaires en ligne, qui nécessitent une assurance maximale que le code qu'ils téléchargent n'a pas été altéré ou falsifié.

## 1.6) LES OUTILS EN LIGNES DE COMMANDES.

### 1) SIGCHECK.

- Sigcheck du Sysinternals :  
Sigcheck est un outil en ligne de commande développé par Sysinternals (maintenant une partie de Microsoft) qui permet de vérifier l'authenticité et l'intégrité des fichiers PE (EXE, DLL, etc.). Sigcheck utilise la signature numérique du fichier pour vérifier son authenticité et peut également vérifier si le fichier a été modifié depuis sa création.
- Sigcheck peut être utilisé pour vérifier les fichiers système Windows, les fichiers exécutables téléchargés à partir d'Internet et les fichiers exécutables créés par les développeurs de logiciels. Il peut également être utilisé pour afficher les informations de version et les dépendances de fichier d'un fichier PE.
- Voici la procédure d'installation en ligne de commande :
  - 1) Powershell – command « Invoke-WebRequest  
<https://download.sysinternals.com/files/Sigcheck.zip> – OutFile Sigcheck.zip »
  - ⇒ Cette commande permet l'installation de sigchek
  - 2) Expand-Archive Sigcheck.zip – DestinationPath.
  - ⇒ Cette commande permet l'extraction des fichiers de l'archive compressés.
  - 3) .\shigcheck
  - ⇒ Cette commande permet le lancement de l'outil, vous devriez arriver sur cette interface. :

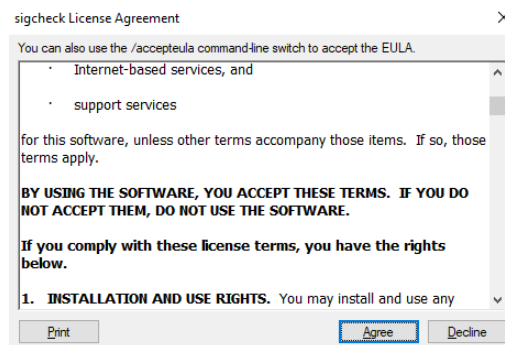


Figure 34 Condition d'utilisation de sigcheck.

- Une fois les conditions d'utilisation acceptées pour vérifier un exécutable on procède de la façon suivante : .\sigcheck -a [Le chemin d'accès de l'exécutable]
- La figure ci-dessous montre le résultat de l'analyse :  

```
PS C:\Users\DELL> .\sigcheck -a [C:\Users\DELL\Desktop\BARBIER_PIERRE_V2\src\test\notepad.exe]

Sigcheck v2.90 - File version and signature viewer
Copyright (C) 2004-2022 Mark Russinovich
Sysinternals - www.sysinternals.com
```

Figure 35 Résultat d'analyse avec sigcheck.

- On peut voir ici que l'exécutable est signé, on peut même connaître le nom du développeur (Mark Russinovich).

## 2) SIGHTOOL.

- Tous comme sigcheck, signtool est un utilitaire de ligne de commande fournie par Microsoft pour signer numériquement des fichiers exécutables, des bibliothèques de liens dynamiques (DLL), des contrôles ActiveX, des packages de déploiement d'applications (MSI) et d'autres types de fichiers Windows. La signature numérique ajoute une couche de sécurité supplémentaire pour garantir que les fichiers n'ont pas été modifiés ou altérés depuis leur création initiale.
- Pour utiliser SignTool, vous devez d'abord créer un certificat numérique à l'aide d'un outil tel que MakeCert ou un autre fournisseur tiers. Ce certificat est ensuite utilisé pour signer le fichier avec SignTool. Le processus de signature ajoute une signature numérique au fichier, qui peut être vérifiée à l'aide de l'outil de vérification de signature de Microsoft ou d'autres outils tiers.
- En plus de signer des fichiers individuels, SignTool peut également être utilisé pour créer et vérifier des catalogues de fichiers (des fichiers .cat) qui contiennent des informations de signature pour un ensemble de fichiers. Les catalogues de fichiers peuvent être utilisés pour signer de grandes quantités de fichiers en une seule opération, ce qui peut être utile dans les environnements de déploiement d'applications.
- En somme signtool est un utilitaire permettant l'ajout d'une couche de sécurité sur les fichiers Windows en les signant numériquement.

### 1.7) LA DETECTION DE SIGNATURE DANS PROCESS EXPLORER.

- En effet nous avons déjà parlé de process Explorer précédemment. Cet exécutable permet également de vérifier si un exécutable est signé. Il permet de vérifier tous les exécutables à l'origine des processus en cours d'exécution sur la machine. Pour cela il suffit de faire apparaître la colonne « verified signer » il faut effectuer une clique droite sur les colonnes puis sélectionner « select columns ». Il ne reste plus qu'à cocher la case vérifier signer dans la section « process image » comme le montre la figure si dessous :

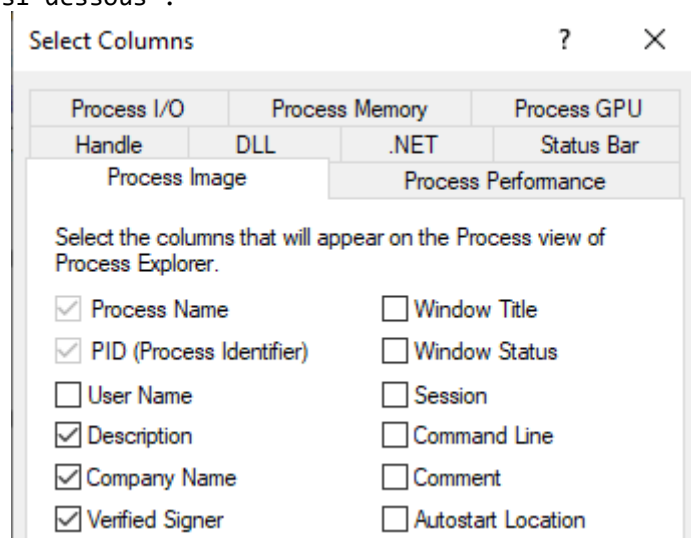


Figure 36 Activations de La vérification de signature dans process Explorer.

- Maintenant on se dirige dans les options, puis on sélectionne « verify image signatures » comme le montre la capture ci-dessous :

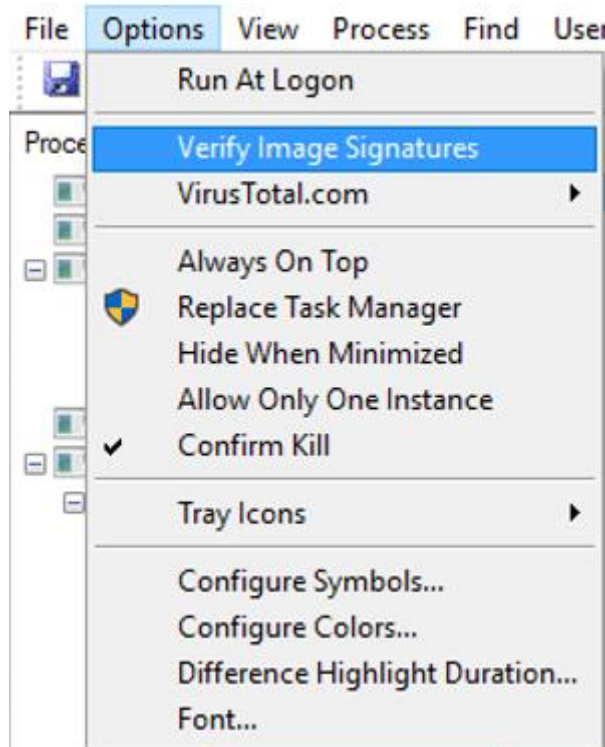


Figure 37 Vérification des signatures avec Process Explorer.

- On peut maintenant voir qu'une colonne « verified signer » est apparue, et qu'elle contient le nom du signataire pour chaque exécutable comme le montre la figure ci-dessous :

Process Explorer - Sysinternals: www.sysinternals.com [DESKTOP-UR9ANAS\DELL]

Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	VirusTotal	Verified Signer
ApplicationFrameHost.exe		15 692 K	33 484 K	12764	Application Frame Host	Microsoft Corporation		(Verified) Microsoft Corporation
WINWORD.EXE	< 0.01	263 156 K	410 748 K	10720	Microsoft Word	Microsoft Corporation		(Verified) Microsoft Corporation
pwa-helper.exe		2 384 K	12 924 K	7700	PwaHelper executable for Id...	Microsoft Corporation		(Verified) Microsoft Corporation
procexp64.exe	4.54	31 924 K	53 756 K	5156	Sysinternals Process Explorer	Sysinternals - www.sysinter...		(Verified) Microsoft Corporation
PresentationFontCache.exe		24 744 K	16 068 K	5452	PresentationFontCache.exe	Microsoft Corporation		(Verified) Microsoft Corporation
OneDrive.exe		29 712 K	99 980 K	8756	Microsoft OneDrive	Microsoft Corporation		(Verified) Microsoft Corporation
OfficeClickToRun.exe	< 0.01	41 120 K	72 228 K	4292	Microsoft Office Click-to-Run...	Microsoft Corporation		(Verified) Microsoft Corporation

Figure 38 résultats d'analyses de signature avec Process Explorer.

## 1.8) MODIFICATION DE L'AUTOMATE.

- Nous avons vu précédemment qu'il était possible de vérifier la signature d'un exécutable via des outils fournis par le sysinternal et Microsoft. Cependant, il peut être intéressant d'automatiser cette tâche. Pour cela, nous allons intégrer une fonction à notre automate, permettant d'identifier si l'application possède un certificat ou à contrario s'il n'en possède pas. Pour cela, nous utiliserons le module python pefile. Pour procéder à son installation, on saisit la commande « `pip install pefile` » dans un terminal ouvert au préalable en mode administrateur.



- Voici le résultat de l'installation :

```
C:\Windows\system32>pip install pefile
Collecting pefile
  Downloading pefile-2023.2.7-py3-none-any.whl (71 kB)
----- 71.8/71.8 kB 654.8 kB/s eta 0:00:00
Installing collected packages: pefile
Successfully installed pefile-2023.2.7
```

Figure 39 Installation du module pefile.

- Maintenant nous pouvons passer à la modification du code pour que le robot puisse répondre à nos attentes. Dans un premier temps, on commence par l'importation du module installé ci-dessus, dans la liste des modules précédemment importer on ajoute pefile. :

```
import vt
import os
import time
import json
import pefile
```

Figure 40 Importation du module pefile dans le code.

- On peut maintenant ajouter une fonction permettant de détecter si l'exécutable possède un certificat :

```
def is_signed(file_path):
    pe = pefile.PE(file_path)
    for section in pe.sections :
        characteristics = getattr(section, 'Characteristics')
        if characteristics & 0x00000020 > 0 or characteristics & 0x20000000 > 0:
            return True
    return False

def scan_file (file_path) :
    with open(file_path, 'rb') as f:
        client = vt.Client(API_KEY)
        analysis = client.scan_file(f, wait_for_completion=True)
        analysis = client.get_object('/analyses/{}'.format(analysis.id))

    if os.path.isfile(file_path) and os.access(file_path, os.X_OK) :
        if is_signed(file_path) :
            print('cette application possède un certificat.')
        else :
            print('Cette application ne possède pas de certificat.')

    sorted_results = json.dumps(analysis.results, sort_keys=True, indent=4)
    print(sorted_results)
    return analysis
```



- Les captures ci-dessous présentent une explication détaillée du code :

```
def is_signed(file_path):  
    pe = pefile.PE(file_path)  
    for section in pe.sections :  
        characteristics = getattr(section, 'Characteristics')  
        if characteristics & 0x00000020 > 0 or characteristics & 0x20000000 > 0:  
            return True  
    return False  
  
if os.path.isfile(file_path) and os.access(file_path, os.X_OK) :  
    if is_signed(file_path) :  
        print("cette application possède un certificat.")  
    else :  
        print("Cette application ne possède pas de certificat.")  
  
sorted_results = json.dumps(analysis.results, sort_keys=True, indent=4)  
print(sorted_results)  
return analysis
```

Figure 41, Explication détaillée du code.

- ⇒ La fonction Python « is\_signed » est définie, prenant en entrée un chemin de fichier.
- ⇒ La fonction utilise la bibliothèque « pefile » pour ouvrir le fichier PE spécifié par le chemin de fichier.
- ⇒ Pour chaque section du fichier PE, la fonction vérifie si les indicateurs de caractéristiques « IMAGE\_SCN\_MEM\_EXECUTE » ou « IMAGE\_SCN\_MEM\_IMAGE » sont définis.
- ⇒ Si l'un de ces indicateurs est défini, la fonction retourne True pour indiquer que le fichier est signé.
- ⇒ Si aucun de ces indicateurs n'est défini pour toutes les sections du fichier PE, la fonction retourne False pour indiquer que le fichier n'est pas signé.
- ⇒ Si le fichier spécifié par le chemin de fichier existe et est exécutable, la fonction vérifie si le fichier est signé en appelant la fonction « is\_signed ».
- ⇒ Si le fichier est signé, la fonction affiche un message indiquant que l'application possède un certificat.
- ⇒ Sinon, la fonction affiche un message indiquant que l'application ne possède pas de certificat.
- ⇒ Les résultats d'une analyse (dont le code n'est pas fourni) sont convertis en format JSON, triés par ordre alphabétique des clés et indentés avec 4 espaces.
- ⇒ Les résultats sont affichés à l'aide de la fonction print.
- ⇒ La fonction retourne le résultat de l'analyse (qui n'est pas clair sans plus d'informations sur le reste du code).

## 1.9) RÉSULTAT D'ANALYSE ET VÉRIFICATION DU BON FONCTIONNEMENT DES MODIFICATIONS.

- Ici, toujours dans le même principe de fonctionnement, nous allons maintenant tester notre automate. Pour, placez un fichier exécutable dans un répertoire accessible par votre robot, ici j'utiliserai l'exécutable notepad.exe. La figure ci-dessous montre le résultat d'analyse :

```
Scanning notepad.exe
cette application possède un certificat.
{
  "ALYac": {
    "category": "undetected",
    "engine_name": "ALYac",
    "engine_update": "20230303",
    "engine_version": "1.1.3.1",
    "method": "blacklist",
    "result": null
  },
  "APEX": {
    "category": "undetected",
    "engine_name": "APEX",
    "engine_update": "20230301",
    "engine_version": "6.393",
    "method": "blacklist",
    "result": null
  },
  "AVG": {
    "category": "undetected",
    "engine_name": "AVG",
    "engine_update": "20230303",
    "engine_version": "22.11.7701.0",
    "method": "blacklist",
    "result": null
  },
  "Acronis": {
    "category": "undetected",
    "engine_name": "Acronis",
    "engine_update": "20230219",
    "engine_version": "1.2.0.114",
    "method": "blacklist",
    "result": null
  }
}
```

Figure 42 Résultat d'analyse de l'automate.

- On peut voir ici que notre script fonctionne, en effet le programme détecte que Notepad++ est un exécutable possédant un certificat et donc qu'il possède une signature de code. On peut donc en déduire que notre automate est opérationnel.

## 1.10) CONCLUSION.

- En somme, la signature de code est une technique essentielle pour garantir l'intégrité et l'authenticité des logiciels. Le fait de signer votre code permettra de prouver que celui-ci provient d'une source fiable et qu'il n'a pas été altéré depuis sa création. Cette technique de sécurité est particulièrement importante pour les applications qui gèrent des données sensibles ou critique, comme les systèmes bancaires, les applications médicales ou les systèmes de contrôle industriel. En utilisant des certificats de signature émis par des autorités de confiance, les développeurs peuvent renforcer la confiance des utilisateurs dans leurs applications et réduire le risque de compromission ou d'attaque malveillante.

## **XI) ANALYSE DE FICHIERS.**

---

- L'organisation et l'accès aux données stockées sur des dispositifs de mémoire de masse est une fonctionnalité vitale de tout système informatique. Cela est réalisé par le biais d'un ensemble de logiciels connu sous le nom de système de gestion de fichiers. Le système de gestion de fichiers permet aux utilisateurs de classer et de localiser leurs fichiers avec facilité, tout en optimisant l'utilisation de l'espace de stockage en créant des partitions et des volumes. Dans les sections suivantes, nous examinerons plus en détail les différents éléments du système de gestion de fichiers, ainsi que la manière dont les fichiers sont stockés, organisés et accessibles via les applications et le système d'exploitation. Nous discuterons également de la sécurité des données et de la fragmentation des fichiers, des concepts qui jouent un rôle crucial dans la gestion des fichiers.

### **1.1) LA MEMOIRE DE MASSE.**

- La mémoire de masse est un élément crucial pour tout système informatique car elle permet de stocker des données à long terme. Les dispositifs de stockage les plus communs incluent le disque dur, la clé USB, le SSD et le CD/DVD, qui offrent tous une grande capacité de stockage à un coût abordable. En plus de stocker des fichiers, des programmes et des systèmes d'exploitation, la mémoire de masse est également utilisée pour stocker des données de sauvegarde et des archives de long terme. Cependant, il est important de faire la distinction entre la mémoire de masse et la mémoire cache, qui est une forme de mémoire vive utilisée pour stocker temporairement des données fréquemment utilisées pour améliorer les performances du système. Dans cette section, nous allons examiner les deux types de dispositifs de stockage les plus utilisés dans les ordinateurs de bureau et portables, tout en explorant leurs différences et leurs avantages.

### **1.2) LES DIFFERENTES MEMOIRES DE MASSE.**

#### **1) LE DISQUE DUR HDD OU DISQUE DUR MECANIQUE :**

- Dans les systèmes informatiques, l'une des formes de stockage de données les plus fréquentes est le disque dur. Celui-ci est constitué d'un assemblage de disques rigides empilés les uns sur les autres, revêtus d'un matériau magnétique et lus/écrits à l'aide d'une tête de lecture/écriture. Les disques durs sont généralement divisés en secteurs de 512 ou 4096 octets, selon le standard utilisé. Lorsqu'un fichier est sauvegardé sur un disque dur, le système d'exploitation le découpe en plusieurs fragments, qu'il enregistre ensuite sur le disque à différents emplacements. Cette technique, appelée fragmentation, permet d'optimiser l'utilisation de l'espace de stockage. Cependant, si le fichier est trop volumineux, les fragments supplémentaires seront stockés ailleurs sur le disque, ce qui peut entraîner des temps de lecture/écriture plus longs et donc des performances réduites.

- La capture ci-dessous présente les différents composants d'un disque dur mécanique :

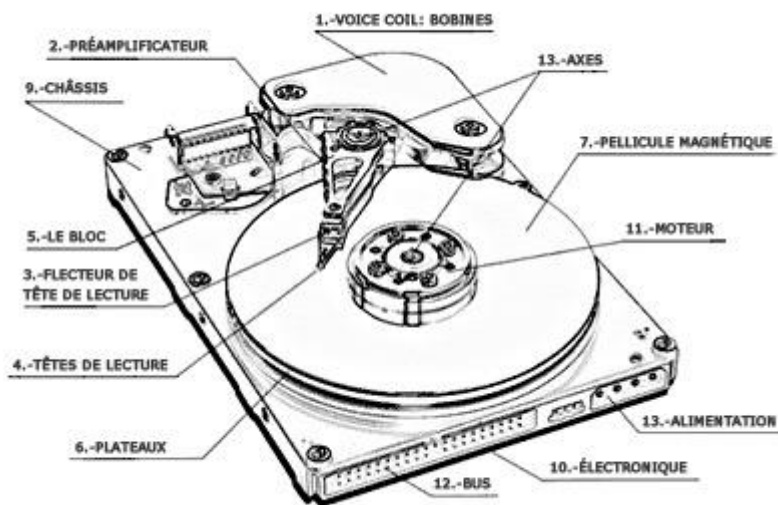


Figure 43 Les différents composants d'un disque mécanique.

- Lorsque l'accès à un fichier est demandé, le système d'exploitation recherche les fragments de données qui le composent sur le disque dur et les regroupe pour reconstituer le fichier. Le processus de lecture des données est effectué par une tête mobile qui se déplace sur le disque dur à une vitesse variable en fonction de la rotation du disque et de l'emplacement des fragments de données.
- Il faut savoir que les disques HDD peuvent fournir une capacité de stockage importante, cependant ils sont extrêmement sensibles aux chocs et à la perturbation magnétique. Leur vitesse de lecture est limitée par la vitesse de rotation des disquettes.

## 2) LE DISQUE DUR SSD :

- Les mémoires SSD (Solid State Drive) sont un type de stockage de données qui diffèrent des disques durs. En effet, ce type de mémoire utilise des circuits électroniques pour stocker les données. Les SSD ne contiennent pas de pièces permettant des actions mécaniques et stockent les données sous forme de signaux électriques dans des puces de mémoire flash.
- Les SSD sont composés de plusieurs puces de mémoire flash qui sont organisées en unités de stockage plus grandes appelées blocs, et en unités de stockage plus petites appelées pages. Lorsqu'un fichier est stocké sur un SSD, il est divisé en blocs qui sont ensuite stockés sur différentes pages.
- Pour lire ou écrire des données sur un SSD, le contrôleur de stockage envoie des signaux électriques aux puces de mémoire flash. Les données sont ensuite lues ou écrites sur les pages de mémoire flash en utilisant des transistors comme des interrupteurs pour stocker ou libérer des charges électriques.

- La figure ci-dessous présente une illustration de mes propos :

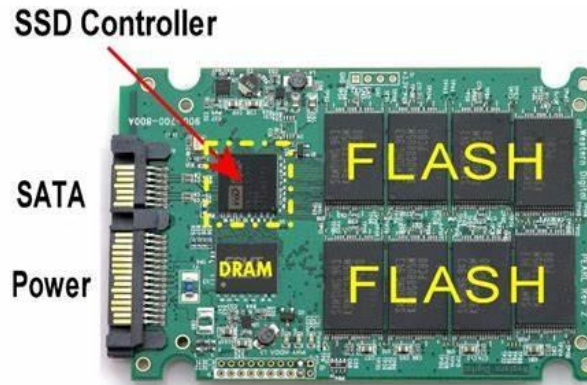


Figure 44 Présentation d'un disque SSD.

### 1.3) LES PARTITIONS.

- Une partition sur un disque dur est une section de ce disque qui est délimitée et traitée comme une unité logique distincte. Lorsqu'un disque dur est partitionné, il est divisé en plusieurs sections distinctes, chacune d'entre elles agissant comme si elle était un disque dur distinct.
- Les partitions sont généralement créées pour permettre l'installation de différents systèmes d'exploitation sur un seul disque dur, pour organiser les fichiers et les données en fonction de leur utilisation, ou pour améliorer les performances en séparant les fichiers et les programmes.
- Chaque partition a une table de partition qui enregistre les informations sur la façon dont la partition est formatée et organisée, comme le système de fichiers utilisé et l'emplacement de chaque fichier. Les partitions on peut également étiquetées ces partitions avec des noms pour faciliter leur identification et leur utilisation.
- Les partitions peuvent être créées et gérées directement avec le système d'exploitation. Par exemple Windows 10 possède un utilitaire de gestion des disques intégré. Cependant il existe aussi des applications tierces comme Disk Management sur Windows ou Disk Utility sur MacOS.
- La figure ci-dessous présente un schéma explicatif du partitionnement des disques :

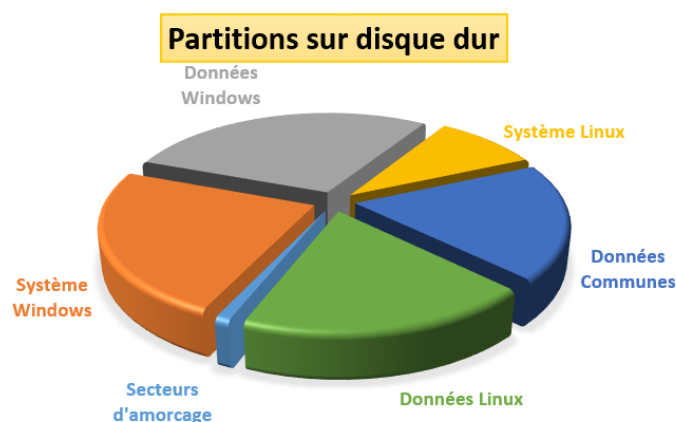


Figure 45 Les partitions sur un disque.

- Sur un disque dur, chaque partition est traitée comme une unité logique distincte, avec sa propre table de partitionnement et son système de fichiers.
- Par exemple Lors du formatage d'une partition, le système de fichiers de cette partition est configuré pour organiser et stocker les données de manière cohérente. Les fichiers sont stockés dans des emplacements spécifiques sur la partition, et la table de partitionnement enregistre les informations sur l'emplacement de chaque fichier sur le disque.

#### 1.4) LES DIFFERENT TYPES DE PARTITION.

- Il existe en effet différents types de partitions de disque spécifiques notamment :
  - ⇒ La partition physique : il s'agit d'une division du disque dur qui crée une zone de stockage séparée, généralement pour installer un système d'exploitation ou stocker des données. Chaque partition physique est traitée comme une unité de stockage distincte, et peut contenir un système de fichiers distinct.
  - ⇒ La partition logique : il s'agit d'une partition virtuelle créée à l'intérieur d'une partition physique existante. Elle permet de diviser une partition physique en plusieurs sections logiques, chacune agissant comme une unité de stockage distincte. Les partitions logiques sont souvent utilisées pour contourner les limites imposées par le système d'exploitation, qui limite le nombre de partitions physiques possibles.
  - ⇒ La partition étendue : elle est une partition spéciale qui est utilisée pour héberger des partitions logiques supplémentaires. En d'autres termes, elle permet de créer plusieurs partitions logiques à l'intérieur d'une seule partition physique. Les partitions étendues sont souvent utilisées lorsque le nombre de partitions nécessaires dépasse la limite autorisée par le système d'exploitation.
- Il est important de noter que quand une partition est créée, celle-ci est stockée dans la table des partitions du disque dur. La table des partitions est une notion que nous aborderons un peu plus tard.

#### 1.4) INTRODUCTION AUX TABLES DE PARTITIONS.

- Une table de partitions est une structure de données utilisée par un système d'exploitation pour identifier et gérer les partitions présentes sur un disque dur ou un autre périphérique de stockage de données. La table de partition contient des informations sur la disposition des partitions sur le disque, telles que leur taille, leur emplacement et leur système de fichiers.
- On notera qu'il existe de type de table des partitions : la table de partition de disque maître (MBR) et la table de partition guidée (GPT). La table MBR a été introduite avec les premiers disques durs et est limitée à une taille maximale de 2 To pour les disques durs. En revanche, la table GPT est utilisée pour les disques durs de plus grande capacité et peut prendre en charge des partitions plus grandes.
- En somme une table des partitions est utilisé par le système d'exploitation pour gérer l'espace des disques ou bien créer, supprimer, redimensionner ainsi que pour installer et démarrer des systèmes d'exploitation. On notera qu'il est très important de comprendre la structure d'une table de partitions avant d'effectuer des opérations avancé sur le disque dur comme la gestion des partitions, la récupération de données ou le clonage de disque.

## 1.5) PREPARATION ET AUTOPSIE D'UNE CLE USB FAT.

- Avant de commencer notre exploration, il faut savoir qu'il existe plusieurs système de fichiers dont deux sont les plus important : le FAT et le NTFS.

### 1) LE FAT.

- ⇒ FAT (File Allocation Table) est un système de fichiers populaire utilisé sur les disques durs et les supports de stockage tels que les disquettes, les cartes mémoire et les clés USB. Ce système de fichiers a été développé par Microsoft et était largement utilisé dans les premières versions de Windows, jusqu'à Windows 95/98.
- Le système de fichiers FAT organise les données sur un disque dur ou un autre support de stockage en utilisant une table d'allocation de fichiers. Cette table indique où sont stockées les données de chaque fichier sur le disque. La table d'allocation des fichiers se divise en plusieurs sections, dont la première est la table FAT. Cette table contient une liste des clusters de données sur le disque et indique si un cluster donné est utilisé ou libre. Les fichiers sont stockés sur le disque en divisant leur contenu en clusters de taille fixe, qui peuvent varier de 512 octets à plusieurs kilo-octets, en fonction de la taille du disque et des options de formatage.
- Maintenant que nous connaissons l'existence de ce système de fichier, nous pouvons commencer l'étude de celui-ci. Commençons par préparer notre clé au format FAT.

### 2) LES OUTILS EN LIGNES DE COMMANDES.

- Pour préparer notre disque nous pouvons utiliser des outils tel que « diskpart » en ligne de commandes. La suite d'instruction suivante permet e formater notre disque au format fat :
- ⇒ Diskpart
  - Permet de lancer l'outil disk part.
- ⇒ list disk
  - Permet de lister tous les disponible.
- ⇒ select disk <numéro du disque>
  - Permet de sélectionner le disque voulu.
- ⇒ Clean
  - Permet le nettoyage du disque.
- ⇒ create partition primary
  - Permet de créer une nouvelle partition.
- ⇒ select partition 1
  - Permet de sélectionner la partitions nouvellement crée
- ⇒ format fs=fat32 quick
  - Permet de formater cette partition au format fat32.

### 3) PHASE D'EXPLORATION.

- Maintenant que notre support est prêt, nous pouvons commencer son exploration. Nous allons ici utiliser l'outil « HxD », c'est tout simplement un éditeur en hexa décimal. Ici de faciliter notre lecture des données, on privilégiera l'hexadécimal au binaire. La logiciel « HxD est téléchargeable via l'url suivante :

<https://mh-nexus.de/en/downloads.php?product=HxD20>

- La capture ci-dessous présente l'interface du logiciel HxD :

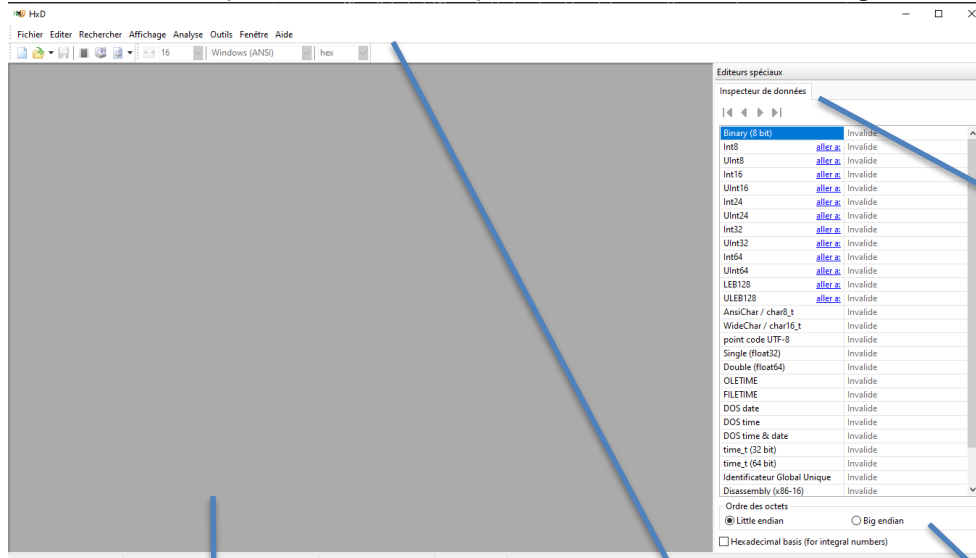


Figure 46 Présentation de L'éditeur hexadécimal HxD.

Inspection des données.

Emplacement des données en cour d'analyse.

Barre des tâches.

Choix du sens de lecture.



- Pour ouvrir notre support dans le logiciel, il suffit de cliquer sur « outils », puis « ouvrir disque » comme le montre la figure ci-dessous :

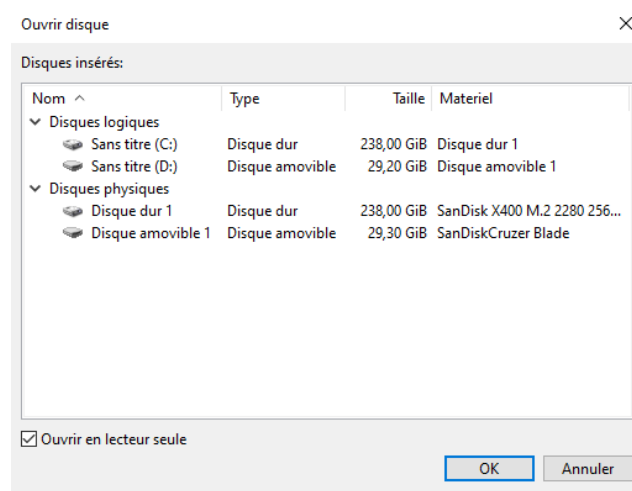


Figure 47 Ouverture du support test dans HxD.

- On peut voir deux types de disques : le disque physique et le disque logique.
- Le disque physique représente le support de stockage réel, tel que la clé USB ou le disque dur. Il est identifié par une lettre de lecteur, telle que C:, D:, E:, etc. Le disque physique contient une ou plusieurs partitions, qui sont des zones de stockage délimitées sur le disque.
- Le disque logique est une vue virtuelle des partitions du disque physique. Il est créé par le système d'exploitation pour permettre aux utilisateurs d'accéder aux données stockées sur le disque. Chaque partition est représentée par une lettre de lecteur et peut contenir un système de fichiers, tel que NTFS ou FAT32.
- Et on sélectionne « Disque amovible 1 » qui représente la clef formatée précédemment qui nous sert de test. Voici un aperçu du résultat à l'ouverture du disque :

Disque amovible 1																	
Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Texte Décodé
00000000	B3	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00	ŠaZD4.  žAž04.  ž.
00000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00	Š. .uówPh. .Ež. .
00000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10	Š4.Š. . . . .fŠ.
00000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00	ŠnI. .V.UEF. .EF. .
00000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	ŠaŠUI.  r. .žUŠu.
00000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74	ŠA. .t.ŠF.Š.Š.Š.Š.Š.
00000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	Šfh. . . . .fšv.Š.Š.Š.
00000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	Šh.Š.Š.Š.Š.Š.Š.Š.Š.
00000080	9F	83	C4	10	9E	EB	14	B8	01	02	BB	00	7C	8A	56	00	ŠfŠ.Š.Š.Š.Š.Š.Š.Š.Š.
00000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	Šv.ŠN.Šn.Š.Š.Š.Š.Š.
000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	Šu.Š.Š.Š.Š.Š.Š.Š.Š.
000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	Š2aŠV.Š.Š.Š.Š.Š.Š.Š.
000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	Šunšv.Š.Š.Š.Š.Š.Š.Š.
000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75	Šf.Š.Š.Š.Š.Š.Š.Š.Š.
000000E0	00	FB	B8	00	BB	CD	1A	66	23	C0	75	3B	66	81	FB	54	Š.Š.Š.Š.Š.Š.Š.Š.Š.
000000F0	43	50	41	75	32	81	F9	02	01	72	C2	66	68	07	BB	00	ŠPAu2.Š.Š.Š.Š.Š.Š.Š.
00000100	00	66	68	00	02	00	00	66	68	08	00	00	00	66	53	66	Šfh.Š.Š.Š.Š.Š.Š.Š.Š.
00000110	53	66	55	66	68	00	00	00	00	66	68	00	7C	00	00	66	Šfufh.Š.Š.Š.Š.Š.Š.Š.
00000120	61	68	00	00	07	CD	1A	5A	32	F6	EA	00	7C	00	00	CD	Šah.Š.Š.Š.Š.Š.Š.Š.Š.
00000130	18	A0	B7	07	EB	08	A0	B6	07	EB	03	A0	B5	07	32	E4	Š.Š.Š.Š.Š.Š.Š.Š.Š.
00000140	05	00	07	8B	F0	AC	3C	00	74	09	BB	07	00	B4	0E	CD	Š.Š.Š.Š.Š.Š.Š.Š.Š.
00000150	10	EB	F2	F4	EB	FD	2B	C9	E4	64	EB	00	24	02	E0	F8	Š.Š.Š.Š.Š.Š.Š.Š.Š.
00000160	24	02	C3	49	6E	76	61	6C	69	64	20	70	61	72	74	69	Š.Š.Invalid parti
00000170	74	69	6F	6E	20	74	61	62	6C	65	00	45	72	72	6F	72	tion table.Error
00000180	20	6C	6F	61	64	69	6E	67	20	6F	70	65	72	61	74	69	loading operati
00000190	6E	67	20	73	79	73	74	65	6D	00	4D	69	73	73	69	6E	ng system.Missin
000001A0	67	20	6F	70	65	72	61	74	69	6E	67	20	73	79	73	74	g operating syst
000001B0	65	6D	00	00	00	63	7B	9A	96	73	82	F0	00	00	00	20	em.Š.Š.Š.Š.Š.Š.Š.
000001C0	21	00	0C	FE	FF	FF	00	08	00	00	00	F8	A7	03	00	00	!Š.Š.Š.Š.Š.Š.Š.Š.
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	Š.Š.Š.Š.Š.Š.Š.Š.
00000200	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.
00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	Š.Š.Š.Š.Š.Š.Š.Š.

Figure 48 Résultat de L'ouverture.

- On peut apercevoir Ici le premier secteur soit le secteur d'amorçage présenté dans le schéma ci-dessus.
- Maintenant, il serait intéressant de trouver la table d'allocation, pour cela nous allons créer un fichier texte nommé « test.txt » avec comme contenu « ceci est un test ». Puis à l'aide de l'outil de recherche, nous essaierons de retrouver le fichier dans la table d'allocation. On sélectionne donc « rechercher dans la barre des tâches du logiciel, puis on entre le nom du fichier comme le montre la figure-ci-dessous :

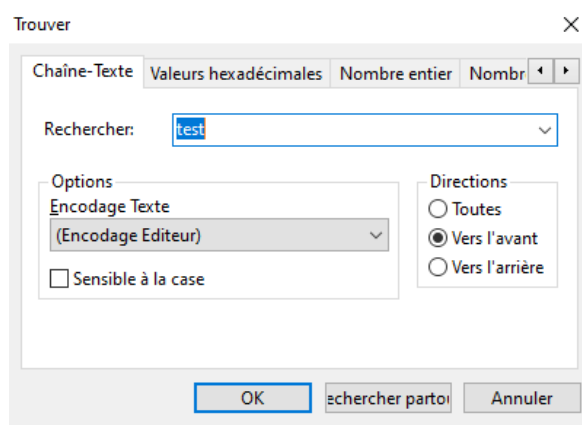


Figure 49 Recherche de La table d'allocation dans HxD.

- En validant, on peut voir que le fichier a bien été retrouvé dans la table d'allocation :

00110C000 63 65 63 69 20 65 73 74 20 75 6E 20 74 65 73 74 ceci est un test Secteur 34 912

Figure 50 Résultat de La recherche.

## 1.6) LA CONSIDERATION DE LA SECURITE INFORMATIQUE DANS TOUTES LES ETUDES ET MANIPULATIONS DE DONNEES.

- Après avoir abordé la plupart des concepts fondamentaux dans le stockage de données, en matière d'étude de fichiers, il est également important de considérer

la sécurité informatique pour protéger les données contre les attaques malveillantes, telles que la dissimulation de payload dans les fichiers. La dissimulation de payload dans les fichiers consiste à ajouter des instructions malveillantes dans un fichier apparemment inoffensif, tel qu'un document Word ou un fichier image. Cela peut permettre aux attaquants de prendre le contrôle de l'ordinateur de la victime, de voler des informations sensibles ou d'exécuter d'autres activités malveillantes. Dans le cadre de l'étude de fichiers avec le logiciel HxD, il est important de prendre en compte la sécurité informatique et de comprendre comment repérer les signes de dissimulation de payload dans les fichiers.

- Un payload peut se traduire par l'insertion de données derrière un fichier comme un charge utile inconnue du logiciel et accédant aux données. Nous allons voir qu'il n'est pas très compliqué d'ajouter de la payload. Avec le logiciel HxD on commence par ouvrir notre disque en décochant l'option « ouvrir en lecture seul » puis on ouvre le disque logique qui nous sert de test. On utilise ensuite l'outil de recherche comme précédemment pour trouver le fichier test. Puis il ne reste plus qu'à se placer à la fin du dernier secteur utilisé par le fichier test et d'inscrire ce que vous souhaitez puis d'enregistrer les changements.
- Et rappelle les propriétés de mon fichier sont les suivantes :

Emplacement :	D:\
Taille :	16 octet(s) (16 octets)
Sur disque :	16,0 Ko (16 384 octets)

Figure 51 Propriété du fichier créé.

- Voici un le résultat après la modification :

```

00100C000 63 65 63 69 20 65 73 74 20 75 6E 20 74 65 73 74 ceci est un test Secteur 32 864
00100C010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C0F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00100C1F0 00 00 00 00 00 00 00 00 50 41 59 4C 4F 41 44 00 .....PAYLOAD.

```

Figure 52 Dissimulation de la charge utile (payload).

- Pourtant en ouvrant le fichier, on n'aperçoit aucun changement :

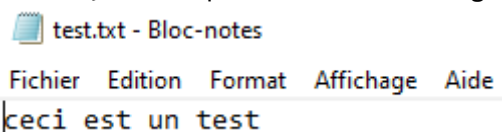


Figure 53 Vérification de la transparence de la charge utile.

- Ni même en consultant les propriétés de ce fichier :

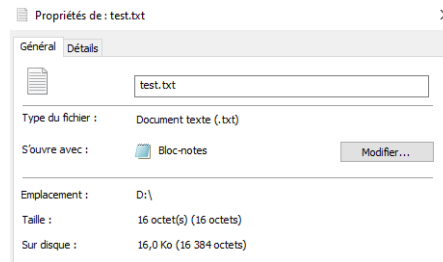


Figure 54 Les propriétés du fichier modifié.

- On remarque la taille du fichier et toujours la même. Nous pouvons donc en conclure que nous avons réussi notre test, c'est-à-dire ajouter des données en fin de fichiers de manière qu'elles ne soient pas détectées par le système d'exploitation ou le logiciel. C'est ici que l'on peut s'apercevoir du potentiel de cette manœuvre. En effet on pourrait très bien dissimuler du code malveillant plus tôt qu'une simple chaîne de caractère.

## 1.7) PREPARATION ET AUTOPSIE D'UNE CLE USB NTFS.

### 1) LE NTFS.

- ⇒ NTFS (New Technology File System) est un système de fichiers propriétaire développé par Microsoft pour les systèmes d'exploitation Windows NT, Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 8 et Windows 10. Il est destiné à remplacer le système de fichiers FAT et offre de nombreuses fonctionnalités avancées pour la gestion des fichiers et des disques durs.
- Le système de fichiers NTFS utilise une table d'allocation de fichiers similaire à celle du système de fichiers FAT, mais il est organisé différemment et offre plusieurs avantages par rapport à son prédécesseur. Il prend en charge les volumes de grande capacité allant jusqu'à plusieurs téraoctets et les fichiers de grande taille, avec une limite théorique de 16 exaoctets (soit environ 16 milliards de gigaoctets). Il permet également des noms de fichiers et de dossiers plus longs, jusqu'à 255 caractères, avec une prise en charge de la casse et des caractères spéciaux.
  - Le système de fichiers NTFS offre également une sécurité accrue grâce à la prise en charge de l'ACL (Access Control List), qui permet de définir des autorisations d'accès pour les fichiers et les dossiers, ainsi qu'un journal des transactions pour les opérations de lecture/écriture. Il prend également en charge la compression de fichiers et la cryptographie de fichiers individuels.
  - Maintenant que nous connaissons l'existence de ce système de fichier, nous pouvons commencer l'étude de celui-ci. Commençons par préparer notre clé au format FAT.
  - L'outil diskpart utilisé précédemment peut également permettre le formatage de disque au format « NTFS ». Il suffit de remplacer la commande « format fs=fat32 quick » par « format fs=ntfs quick ».

### 2) LA NOTIONS D'ADS.

- Les ADS, ou Alternate Data Streams (flux de données alternatifs en français), sont une fonctionnalité du système de fichiers NTFS (New Technology File System) de Windows. Les ADS permettent d'associer des données supplémentaires à un fichier existant, sans modifier les données d'origine du fichier. Cela permet de stocker des informations telles que des métadonnées, des informations de sécurité ou des données supplémentaires pour une application.
- Les ADS peuvent être créés et manipulés à l'aide d'outils en ligne de commande tels que "streams.exe" de Sysinternals ou d'autres outils tiers. Les ADS peuvent être utiles pour certaines applications, mais peuvent également poser un risque pour la sécurité car ils peuvent être utilisés pour cacher des fichiers malveillants ou pour contourner les mesures de sécurité.
- Les Alternate Data Streams (ADS) fonctionnent de manière simple dans un système de fichiers NTFS. Chaque fichier dans le système est représenté par une entrée dans la Master File Table (MFT), contenant des informations telles que le nom du fichier, sa taille, ses dates de création et de modification, ainsi que ses autorisations d'accès. Pour ajouter des données supplémentaires à un fichier, il suffit de créer un nouveau flux de données à l'intérieur de l'entrée de ce fichier dans la MFT, appelé un ADS.

- Concernant la creation d'un ADS, on utilisera ici la redirection grâce à l'invite de commandes. Voici une illustration de mes propos :
- Pour utiliser la redirection avec l'invite de commande on saisie la commande suivantes : « echo test payload > test.txt :data ».
- ⇒ La commande "echo test payload > test.txt :data" permet de créer un fichier nommé "test.txt" contenant une charge utile ou des données supplémentaires appelées "data" dans un flux de données alternatif (ADS) à l'aide de la ligne de commande Windows.  
Le contenu "test payload" est transmis à la commande "echo", qui l'écrit dans le fichier "test.txt". Le symbole ">", utilisé ici, redirige la sortie de la commande vers le fichier spécifié. Le texte ":data" après le nom de fichier est le nom du flux de données alternatif (ADS) qui sera créé pour stocker la charge utile.
- En ouvrant le fichier en question, avec l'éditeur hexa décimal on peut voir que celui-ci est vierge comme le montre la capture ci-dessous :

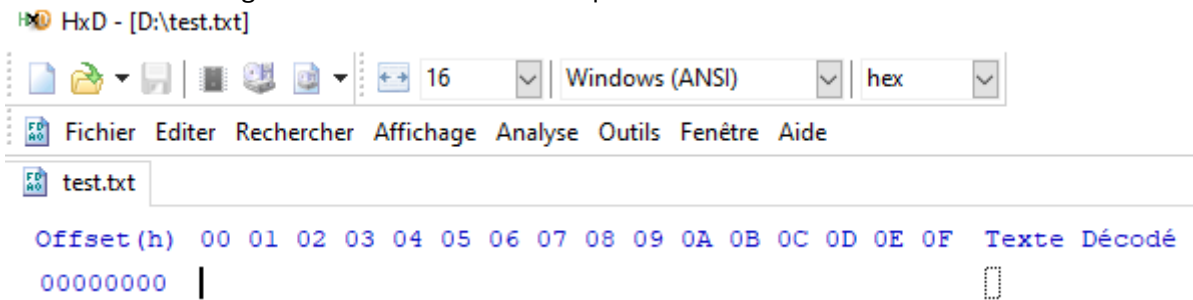


Figure 55 Ouverture du fichier dans L'éditeur HxD.

### 3) LES OUTILS EN LIGNE DE COMMANDES.

- Pour retrouver la chaine de caractère que nous avons entré au paravent, il faut chercher dans l'ADS nommé « data » lié au fichier en question. Pour cela on utilise la commande suivante « more < test.txt :data », Voici le résultat de la commande.

```
C:\Users\DELL\Desktop>more < test.txt:data
echo test payload
```

Figure 56 Recherche dans L'ADS.

- Cependant, il existe un autre moyen de consulter les ADS. Avec un logiciel tiers appelé streams. Des outils nous ont déjà été fourni don streams dans la suite sysinternal. Il nous suffit donc de l'appeler en précisant sont chemin d'accès, puis de préciser le fichier souhaité et voici le résultat :

```
C:\Users\DELL\Desktop>C:\Users\DELL\Desktop\cours\TP_NSI\virus-hunter\tools\SysinternalsSuite\streams.exe test.txt
streams v1.60 - Reveal NTFS alternate streams.
Copyright (C) 2005-2016 Mark Russinovich
Sysinternals - www.sysinternals.com

C:\Users\DELL\Desktop\test.txt:
:data:$DATA 20
```

Figure 57 Streams pour consulter Les ADS.

- On aperçoit alors notre ADS ainsi que sa taille.
- En sommes, les flux de données alternatifs (ADS) sont une fonctionnalité du système de fichiers NTFS de Windows qui permettent de stocker des données supplémentaires dans un fichier. Les ADS permettent de stocker des informations telles que des métadonnées, des commentaires ou des charges utiles dans un fichier existant, sans affecter le contenu principal du fichier. Les ADS peuvent être

créés et lus à l'aide d'outils en ligne de commande tels que "more" ou "type", ainsi qu'à l'aide de programmes tiers tel que streams.

## 1.8) LE BOOT SEQUENCE.

- L'INTRODUCTION DE LA SEQUENCE DE DEMARRAGE EST IMPORTANTE DANS LA RECHERCHE DE MALWARE CAR ELLE PERMET D'IDENTIFIER LES ACTIVITES SUSPECTES QUI SE PRODUISENT DES LE DEMARRAGE DU SYSTEME D'EXPLOITATION.
- LA SEQUENCE DE DEMARRAGE, EGALEMENT APPELEE "BOOT SEQUENCE", EST LA SEQUENCE D'OPERATIONS QUI SE PRODUISENT LORSQUE VOUS ALLUMEZ VOTRE ORDINATEUR. ELLE INCLUT LE CHARGEMENT DU BIOS (BASIC INPUT/OUTPUT SYSTEM), L'INITIALISATION DU MATERIEL, LE CHARGEMENT DU SYSTEME D'EXPLOITATION ET LE LANCEMENT DES PROGRAMMES ET DES SERVICES.
- LES MALWARES PEUVENT INFECTER LE SYSTEME D'EXPLOITATION DES LE DEMARRAGE EN SE CACHANT DANS LA SEQUENCE DE DEMARRAGE. ILS PEUVENT MODIFIER LES FICHIERS DE CONFIGURATION, LES PARAMETRES DU SYSTEME OU LES FICHIERS D'AMORÇAGE POUR ASSURER LEUR PERSISTANCE ET LEUR SURVIE MEME APRES UN REDEMARRAGE DU SYSTEME. DANS CETTE PARTIE DU DOCUMENT, NOUS ALLONS EXAMINER LA SEQUENCE DE DEMARRAGE D'UNE MACHINE SOUS WINDOWS.

### 1) L'AMORÇAGE DE LA MACHINE.

- Le boot de la machine, également appelé démarrage ou amorçage de la machine, est le processus qui se produit lorsque vous allumez votre ordinateur ou votre appareil électronique.
- Pendant le processus de boot, le système effectue une série d'opérations pour s'initialiser et se préparer à exécuter les programmes et les tâches demandées par l'utilisateur. Voici les étapes typiques du processus de boot d'un ordinateur :
  - ⇒ Le BIOS (Basic Input/Output System) est chargé. Le BIOS est un petit programme stocké dans une puce sur la carte mère de l'ordinateur. Il est responsable de l'initialisation du matériel et de la configuration du système.
  - ⇒ Le BIOS effectue une série de tests sur le matériel de l'ordinateur, tels que la mémoire vive (RAM), le processeur, les disques durs, les périphériques d'entrée/sortie, etc. Si le BIOS détecte une erreur matérielle, il affiche généralement un message d'erreur à l'écran.
  - ⇒ Le BIOS recherche le système d'exploitation (par exemple, Windows, Linux ou macOS) et charge les fichiers de démarrage correspondants à partir du disque dur ou d'un autre support de stockage amovible tel qu'une clé USB ou un CD/DVD.
  - ⇒ Le système d'exploitation est chargé en mémoire vive (RAM) et démarre. À partir de là, le système d'exploitation est prêt à exécuter les programmes et les tâches demandées par l'utilisateur.
- Le démarrage se poursuit avec l'alimentation de la puce flash celle-ci contenant le microprogramme du BIOS/UEFI. Celle-ci commencera alors à exécuter le programme qu'elle contient.



## 1.9) BIOS ET UEFI.

- Le BIOS (Basic Input/Output System) et l'UEFI (Unified Extensible Firmware Interface) sont deux types de micrologiciels (firmware) utilisés pour initialiser et configurer le matériel d'un ordinateur lors du processus de démarrage.
- Le BIOS est un firmware ancien qui a été utilisé sur la plupart des ordinateurs personnels jusqu'à la fin des années 1990. Il utilise une interface texte simple et un mode de partitionnement de disque dur appelé Master Boot Record (MBR). Il a quelques limitations, notamment la capacité de reconnaître et d'amorcer les disques durs de plus de 2 To.
- L'UEFI, quant à lui, est un firmware plus récent et plus avancé. Il utilise une interface graphique, offre une meilleure compatibilité avec les disques durs de grande capacité, utilise un mode de partitionnement de disque dur appelé GUID Partition Table (GPT) et prend en charge les systèmes d'exploitation 64 bits.
- En outre, l'UEFI peut prendre en charge des fonctionnalités avancées telles que le Secure Boot, qui aide à prévenir l'exécution de logiciels malveillants au démarrage, et le démarrage réseau PXE (Preboot Execution Environment), qui permet de démarrer un ordinateur à partir d'un serveur distant.
- En somme, le BIOS et l'UEFI remplissent la même fonction bien que l'UEFI est plus avancé et offre des fonctionnalités plus poussées pour les ordinateurs modernes. Le lien suivant présente des informations plus poussées sur la différence de démarrage entre le BIOS et l'UEFI :

<https://www.malekal.com/processus-demarrage-windows-mbr/>

## 1.10) LE DEMARRAGE LOGICIEL

- Le démarrage logiciel est la suite de la séquence de démarrage après la phase d'initialisation matérielle effectuée par le BIOS ou l'UEFI. Après avoir initialisé le matériel, le micrologiciel (firmware) cherche et charge le chargeur d'amorçage (bootloader) du système d'exploitation à partir d'un disque dur ou d'un autre périphérique de stockage.
- Le chargeur d'amorçage est un petit programme qui se charge en mémoire et qui est responsable de charger le système d'exploitation en mémoire et de le démarrer. En général, le chargeur d'amorçage est situé dans la partition de démarrage du disque dur et est souvent lié au système d'exploitation installé sur la machine.

### 1) LE CHOIX DU SUPPORT DE DEMARRAGE.

- Le choix du périphérique de démarrage est une fonctionnalité qui permet à l'utilisateur de spécifier à partir de quel périphérique le micrologiciel (firmware) doit démarrer l'ordinateur. En général, l'ordinateur démarre à partir du disque dur interne qui contient le système d'exploitation installé, mais il peut également démarrer à partir d'autres périphériques tels que des disques externes, des CD/DVD, des clés USB ou même d'un réseau.
- Pour sélectionner le périphérique de démarrage, l'utilisateur doit accéder au menu de configuration du BIOS ou de l'UEFI. Dans ce menu, il peut spécifier l'ordre de priorité des périphériques de démarrage. Par exemple, s'il souhaite démarrer à partir d'un CD, il peut spécifier que le CD est le premier périphérique de démarrage. Si le CD n'est pas présent, l'ordinateur tente de démarrer à partir du deuxième périphérique de démarrage spécifié, et ainsi de suite.
- Le choix du périphérique de démarrage peut être utile dans de nombreuses situations, notamment lorsque l'utilisateur souhaite installer un nouveau système d'exploitation à partir d'un disque externe ou d'une clé USB, ou lorsqu'il souhaite effectuer une

récupération système à partir d'un CD de secours. Il peut également être utilisé pour démarrer à partir d'un disque dur différent ou pour tester un nouveau système d'exploitation sans affecter le système d'exploitation actuellement installé sur l'ordinateur.

## 2) LE DEMARRAGE DE WINDOWS.

- Une fois que le chargeur d'amorçage a été chargé en mémoire et qu'il a démarré le noyau du système d'exploitation, le processus de démarrage de Windows commence. Le démarrage de Windows est un processus complexe qui implique plusieurs étapes. Voici une vue d'ensemble des principales étapes du démarrage de Windows :
  - ⇒ Le noyau du système d'exploitation Windows est chargé en mémoire.
  - ⇒ Le noyau initialisé, il charge les pilotes de périphériques requis pour le matériel présent sur l'ordinateur.
  - ⇒ Windows crée le processus système (System Process) et le processus de session (Session Manager Process) qui sont responsables du démarrage de tous les autres processus.
  - ⇒ Le processus de session charge les fichiers de configuration système (System Registry) et les fichiers de configuration de l'utilisateur (User Registry).
  - ⇒ Windows charge les programmes de démarrage automatique (Startup Programs) qui ont été configurés pour s'exécuter au démarrage.
  - ⇒ Le Bureau de Windows est chargé et l'utilisateur peut se connecter.

## XII) TABLE DES ILLUSTRATIONS

Figure 1 Organigramme du virus. ....	4
Figure 2 Diagramme des processus.....	5
Figure 3 Diagramme des threads. ....	5
Figure 4 Fenêtre principale de Process Explorer.....	8
Figure 5 les processus parents. ....	8
Figure 6 les processus indépendants.....	9
Figure 7 Check up avec virustotal.....	9
Figure 8 le score fourni après le scan avec virustotal. ....	9
Figure 9 L'interface principal de Process Monitor.....	9
Figure 10 Modification du home page.....	11
Figure 11 Capture d'événements. ....	11
Figure 12 Configuration des filtres.....	11
Figure 13 Contenu du fichier pref.js.....	12
Figure 14 Capture des nouveaux évènements.....	12
Figure 15 Modification de la configuration des filtres.....	14
Figure 16 le chemin à suivre pour la modification des paramètres DNS. ....	14
Figure 17 Propriété Wifi. ....	14
Figure 18 Insertion d'une nouvelle adresse.....	15
Figure 19 Capture d'événements après modification du DNS. ....	15
Figure 20 Recherche NameServer. ....	15
Figure 21 Vérification de la mention DATA.....	16
Figure 22 Page d'accueil de virustotal.....	17
Figure 23 Résultat du scan de fichier.....	18
Figure 24 Détection de l'empreinte EICAR.....	18
Figure 25 Résultat du scan de site web.....	18
Figure 26 Copie de la clef d'API.....	21
Figure 27 Code du robot VirusTotal.....	21

Figure 28 Résultat du scan du robot.....	22
Figure 29 Contenu du fichier de configuration.....	23
Figure 30 Installation de la bibliothèque YAML.....	24
Figure 31 Ajout du code nécessaire à la liaison du fichier de configuration et du script..	24
Figure 32 l'intégrité que permet la signature de code.....	26
Figure 33, le cryptage et le décryptage des clefs.....	26
Figure 34 Condition d'utilisation de sigcheck.....	28
Figure 35 Résultat d'analyse avec sigcheck.....	28
Figure 36 Activations de la vérification de signature dans process Explorer. ....	29
Figure 37 Vérification des signatures avec Process Explorer. ....	30
Figure 38 résultats d'analyses de signature avec Process Explorer. ....	30
Figure 39 Installation du module pefile.....	31
Figure 40 Importation du module pefile dans le code.....	31
Figure 41, Explication détaillée du code.....	32
Figure 42 Résultat d'analyse de l'automate.....	33
Figure 43 Les différents composant d'un disque mécanique. ....	35
Figure 44 Présentation d'un disque SSD.....	36
Figure 45 Les partitions sur un disque.....	36
Figure 46 Présentation de l'éditeur hexadécimal HxD. ....	39
Figure 47 Ouverture du support test dans HxD.....	40
Figure 48 Résultat de l'ouverture.....	41
Figure 49 Recherche de la table d'allocation dans HxD. ....	41
Figure 50 Résultat de la recherche.....	41
Figure 51 Propriété du fichier créer.....	42
Figure 52 Dissimulation de la charge utile (payload).....	42
Figure 53 Vérification de la transparence de la charge utile. ....	42
Figure 54 Les propriété du fichier modifié.....	43
Figure 55 Ouverture du fichier dans l'éditeur HxD.....	45
Figure 56 Recherche dans l'ADS. ....	45
Figure 57 Streams pour consulter les ADS.....	45