Une image contenant Graphique, clipart, logo, Police

Description générée automatiquement

|  |
| --- |
|  |
| **Windows Malware Hunter Handbook**  Implementation et concepts |
|  |

Page de service

**Référence :**

**Plan de classement :**

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

**Mises à jour**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Auteur | Description du changement |
| NSI-1 | 25/01 | EL GAAMOUS Anissa | Chasse aux virus et empoisonnement DNS |
| NSI-2 | 03/03 | EL GAAMOUS Anissa | Intégration de Python et Virus Total |
| NSI-3 | 17/03 | EL GAAMOUS Anissa | Signature de code |
|  |  |  |  |
|  |  |  |  |

**Validation**

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

**Diffusion**

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

Sommaire

[Page de service 0](#_Toc161416460)

[Sommaire 1](#_Toc161416461)

[1 Rappel du contexte 2](#_Toc161416462)

[2 Objectifs 2](#_Toc161416463)

[3 Chasse aux malware sous Windows 3](#_Toc161416464)

[4 Conception de processus 4](#_Toc161416465)

[5 Process Explorer 4](#_Toc161416466)

[6 Process Monitor 5](#_Toc161416467)

[6.1 Cas d’utilisation n°1 – Firefox homepage 5](#_Toc161416468)

[6.1.1 Recherche des évènements à cibler 5](#_Toc161416469)

[6.1.2 Création du filtre piège 7](#_Toc161416470)

[7 Virus Total 8](#_Toc161416471)

[7.1 Présentation de la plateforme virus total 8](#_Toc161416472)

[7.2 Ecriture d’un robot de défense 10](#_Toc161416473)

[8 Bibliographie 10](#_Toc161416474)

[9 Table des illustrations 10](#_Toc161416475)

[10 Conclusion 10](#_Toc161416476)

[11 Robot de surveillance 10](#_Toc161416477)

[12 Mise en production 11](#_Toc161416478)

[13 Signature de code 11](#_Toc161416479)

[14 Compréhension des certificats numériques et du code 11](#_Toc161416480)

[15 certificats et python 12](#_Toc161416481)

# Rappel du contexte

NetWorking Solutions Inc. (NSI) est une Société de Services du Numérique (ESN[[1]](#footnote-1)) spécialisée dans la conception, la réalisation et la maintenance des infrastructures matérielles et logicielles pour ses clients.

Une entreprise a récemment sollicité les services de NSI en matière de cybersécurité pour réaliser une analyse de sécurité sur les postes de travail de ses employés.

L’entreprise fait appel aux services d’un expert en cybersécurité, qui se déplacera dans les locaux de l'entreprise afin d'effectuer une évaluation des postes de travail utilisés par le personnel, comprenant les ordinateurs de bureau et les ordinateurs portables. L'examen de ces postes impliquera l'analyse des logiciels installés, mettant particulièrement l'accent sur l'observation des processus actifs pour détecter d'éventuelles vulnérabilités et la présence de logiciels malveillants.

Pour cette analyse des processus, il est recommandé d'utiliser deux outils spécifiques : **Process** **Explorer** et **Process** **Monitor**. L'utilisation de ces logiciels permettra une inspection approfondie des activités en cours sur les machines, offrant ainsi la possibilité d'identifier d'éventuelles failles de sécurité[[2]](#footnote-2) et la présence de programmes malveillants. Une description détaillée du fonctionnement et de l'importance de ces outils vous sera fournie lors des démonstrations d'utilisation.

@More : Le lecteur qui souhaite découvrir plus d’information sur la spécialité numérique et sciences informatiques peut consulter le MOOC**[[3]](#footnote-3)** mis à disposition avec l’établissement Inria[[4]](#footnote-4).

Lien vers le MOOC : <https://www.fun-mooc.fr/fr/cours/numerique-et-sciences-informatiques-les-fondamentaux/>

# Objectifs

À l'heure actuelle, il est impératif de maintenir des machines sécurisées en utilisant à la fois un antivirus[[5]](#footnote-5) et un pare-feu[[6]](#footnote-6) 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[[7]](#footnote-7). En 2023 on estime que les attaques par malwares ont augmenté avec une hausse trimestrielle de 110 %[[8]](#footnote-8). 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[[9]](#footnote-9) supplémentaires telles que la sensibilisation des utilisateurs, les mises à jour régulières du système et la surveillance vigilante des menaces.

Ce document offre une exploration approfondie des concepts clés et de leurs applications techniques au sein de l'écosystème du système d'exploitation Windows. Nous illustrons ces concepts à travers des exemples concrets tels que les processus sous Windows, la manipulation d'outils conduisant à l'analyse et à la mise en place de solutions techniques.

Il est crucial de souligner que bien que les démonstrations se concentrent principalement sur Windows en raison de sa popularité mondiale, notamment avec Windows 11, cela n'implique pas que les principes fondamentaux ne s'appliquent qu'à cet environnement. Au contraire, l'objectif est de mettre en lumière l'intrinsèque universalité de ces principes, leur applicabilité à tous les systèmes d'exploitation, avec une mise en œuvre technique généralement similaire d'une plateforme à l'autre.

Les encadrés annotés avec "@ToDo" signalent des tâches cruciales dans le code ou la documentation, tandis que ceux annotés avec "@More" indiquent des aspects nécessitant une attention particulière ou des développements supplémentaires au sein du projet.

*Les éléments annotés avec "@ToDo" constituent des livrables essentiels à fournir.*

*Les sections marquées avec "@More" fournissent des indications pour ceux désirant approfondir leurs connaissances.*

Cette documentation est livrable sous le répertoire ‘doc’ du build du projet.

L’archive des livrables respecte le format suivant :

* build
  + biblio

Dédié au stockage de documents au format PDF.

* + data

Comprends les scripts SQL essentiels à la création et à l'alimentation de la base de données du projet, ainsi que les fichiers de données utilisés comme jeu d'essai le cas échéant.

* + doc

Concerne la documentation du projet, en particulier un document conforme au format de nommage spécifié.

* + lib

Regroupe les bibliothèques additionnelles utilisées par le projet.

* + src

Englobe les codes sources du projet, regroupant ainsi tous les éléments tels que les scripts PowerShell, batch, Python, ainsi que les codes Java, et tout autre code écrit et livré dans le cadre du projet.

* + tools

Regroupe l'ensemble des outils associés au build.

# Chasse aux malware sous Windows

Avant d'engager la lutte contre les logiciels malveillants[[10]](#footnote-10) au sein d'un système, il est crucial d'acquérir une compréhension approfondie de l'environnement et des concepts clés liés au système d'exploitation[[11]](#footnote-11), en particulier dans une implémentation Windows. Dans ce chapitre, le lecteur est guidé à travers la découverte des deux outils spécifiques de SysInternals[[12]](#footnote-12), Process Explorer et Process Monitor. Par la suite, nous explorerons les solutions à mettre en œuvre pour détecter et éliminer les logiciels malveillants.

Les concepts clés qui seront abordé sont les suivantes :

* Processeur – processus | thread
* Mémoire de masse – fichiers
* Mémoire vive

@More : Ces concepts clés sont complexes sous Windows ; il est conseillé au lecteur d'approfondir sa compréhension pour une meilleure maîtrise.

Vous pouvez consulter l’ouvrage : Windows Internals Part 1\_7th Edition.pdf[[13]](#footnote-13) & Windows Internals Part 2\_7th Edition.pdf[[14]](#footnote-14).

SysInternals, développé initialement par Mark Russinovich[[15]](#footnote-15) et Bryce Cogswell avant son acquisition par Microsoft, constitue une suite d'outils logiciels système reconnue, visant à faciliter le diagnostic, la surveillance, et la gestion des performances avancées du système d'exploitation Windows.

# Conception de processus

Lorsque nous abordons le sujet des logiciels malveillants, ou malware[[16]](#footnote-16), il est important de comprendre que nous faisons en réalité référence à des processus malveillants. Dans cette perspective, il est légitime de se poser les questions suivantes : qu'est-ce qu'un processus et à quel moment un processus devient-il malveillant ?

Tout d'abord, un processus[[17]](#footnote-17) est défini comme un ensemble d'instructions programmées exécutées par un ordinateur. Ces instructions sont traitées par le processeur[[18]](#footnote-18), et le programme peut résider dans la mémoire morte[[19]](#footnote-19), mais il est principalement stocké dans la mémoire de masse, puis transféré vers la mémoire vive (RAM)[[20]](#footnote-20) pour son exécution. La durée d'exécution d'un processus est définie, débutant à un moment précis et parfois se terminant également. Un processus peut être lancé soit par l'utilisateur via un périphérique[[21]](#footnote-21), soit par un autre processus.

Le caractère malveillant d'un processus se manifeste généralement par des intentions hostiles. Il peut être introduit par un programme malveillant visant à compromettre la sécurité du système, voler des données sensibles ou causer des dommages. La détection des processus malveillants nécessite souvent l'utilisation d'outils spécialisés en cybersécurité pour identifier et neutraliser les menaces potentielles.

# Process Explorer

Process Explorer, un utilitaire de gestion de processus avancé de SysInternals, va au-delà du Gestionnaire des tâches classique de Windows, fournissant un aperçu détaillé des processus, y compris leur icône, ligne de commande, chemin d'accès complet, statistiques de mémoire, compte utilisateur, attributs de sécurité, et plus encore. En zoomant sur un processus spécifique, il est possible de répertorier les DLL chargées ou les descripteurs de ressources du système d'exploitation qu'il a ouverts.

L'interface de Process Explorer comporte deux sous-fenêtres. La partie supérieure affiche une liste des processus actifs avec les noms de leurs propriétaires, tandis que la fenêtre inférieure, pouvant être fermée, dépend du mode de Process Explorer. En mode handle, les poignées ouvertes par le processus sélectionné dans la fenêtre supérieure sont visualisées. En mode DLL, les DLL et fichiers mappés en mémoire chargés par le processus sont explorés.

Process Explorer intègre également une fonction de recherche puissante pour identifier rapidement les processus avec des descripteurs particuliers ou des DLL chargées. Ces fonctionnalités uniques en font un outil précieux pour détecter les problèmes de version de DLL, gérer les fuites, et offrir un aperçu détaillé du fonctionnement de Windows et des applications.

Il est important de noter que Process Explorer n'exige pas de privilèges administratifs pour son exécution et est compatible avec les clients sous Windows XP et versions ultérieures (y compris IA64), ainsi qu'avec les serveurs sous Windows Server 2003 et versions ultérieures (y compris IA64). Les outils Handle et ListDLL, équivalents en ligne de commande, sont disponibles sur le site Web Sysinternals.

# Process Monitor

## Cas d’utilisation n°1 – Firefox homepage

Le helpdesk niv.1 du client Vinci escalade plusieurs tickets vers vous. Ces tickets signalent tous la modification, sans intervention de l’utilisateur, de la page par défaut du navigateur Firefox. Pire, il apparait que cette page par défaut est une page de phishing qui imite quasi parfaitement la page de connexion de l’extranet Vinci.

### Recherche des évènements à cibler

Explication des procédés

Dans un premier temps, il faut ajouter le « Process Name » dans la colonne. Pour se faire il faut faire défiler la barre au niveau de l’architecture puis sectionner celui qui est demandé.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Figure 1 : Process Monitor Filter

Commentaire de la capture :

Cette capture montre un outil de filtrage pour le suivi des processus, spécifiquement pour le processus « firefox.exe » et pour les événements de type « File System », « Registry » et « Profiling » de catégorie "Write ».

Process Monitor Filter est utilisé pour filtrer et afficher les entrées de suivi de processus qui correspondent à des conditions spécifiques.

Il y a plusieurs sections qui permettent de définir les critères de filtrage :

- Architecture : Cette section permet de filtrer les processus en fonction de leur architecture (par exemple, 32 bits ou 64 bits). Dans ce cas, le filtre est défini sur "Reset", ce qui signifie qu'il n'y a pas de filtrage spécifique basé sur l'architecture.

- Process Name : Cette section permet de filtrer les processus en fonction de leur nom. Le filtre est défini sur "firefox.exe", ce qui signifie que seules les entrées liées au processus "firefox.exe" seront incluses dans les résultats.

- Event Class : Cette section permet de filtrer les événements de suivi de processus en fonction de leur classe. Dans cette capture, il y a plusieurs filtres d'événements définis, tels que "File System", "Registry" et "Profiling". Cela signifie que seuls les événements de ces classes seront inclus dans les résultats.

- Category : Cette section permet de filtrer les événements de suivi de processus en fonction de leur catégorie. Dans cette capture, le filtre est défini sur "Write", ce qui signifie que seuls les événements d'écriture seront inclus dans les résultats.

Dans un second temps, il faut se rendre sur le firefox et capturer les trames, les voici :

Une image contenant texte, logiciel, capture d’écran, Page web

Description générée automatiquement

Figure 2 : Process Monitor

Cette capture de processus montre une liste d'événements liés à l'utilisation de Firefox.

On y retrouve des détails sur les opérations effectuées, les chemins d'accès des fichiers, les résultats des opérations (succès ou échec) et d'autres informations pertinentes.

Ceci peut être utile pour comprendre les actions spécifiques effectuées par Firefox et pour diagnostiquer d'éventuels problèmes.

L’onglet « Time » (1ère colonne) représente l’heure à laquelle les opérations ont été effectuées sur le système.

L’onglet « Process name » concerne les noms des processus ayant effectué les opérations.

Le « PID » est l’identifiant unique associé aux différents processus du système.

L’onglet « Opération » inclut le type d’opération effectuée par un processus, tel que CreateFile ou WriteFile.

L’onglet « Path » précise l’emplacement où une opération est réalisée.

L’onglet « Result » fournit des informations sur l’état final d’une opération.

L’onglet « Détails » fournit des informations sur les opérations, la taille et le type de data.

Par la suite il est nécessaire de se rendre dans l’« event properties » dans la catégorie « event » :

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Figure 3 : Event repertoire

Ci-dessus, nous pouvons constater qu'il y a eu une tentative de création d'un fichier avec un nom en collision dans le répertoire spécifié. Ceci pourrait indiquer un problème potentiel lors de la création du fichier dans le contexte de l'application ou du système d'exploitation.

Pour continuer, nous devons nous rendre dans le fichier par défaut :

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Figure 4 : Dossier fichier Prefs

Puis sélectionner le fichier nommé « prefs »

Voici ce qu’il contient :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 5 : Ouverture du fichier PREF.JS

Nous pouvons ici, retrouvé notre [www.lapin.fr](http://www.lapin.fr). C’est ce qui sera lancé au démarrage de Firefox.

Le fichier PREF.JS de Firefox stocke les préférences utilisateur, y compris les paramètres de sécurité, d'interface utilisateur, de confidentialité et de performances, ainsi que des informations telles que la page d'accueil, les favoris et les liens. Ces données, (chaînes de caractères) définissent les valeurs par défaut ou personnalisées pour ces éléments dans le navigateur. Cependant, un pirate qui accède et modifie le contenu de ce fichier pourrait potentiellement l'utiliser à des fins d'empoisonnement SEO, comme rediriger l'utilisateur vers des sites malveillants ou ajouter des favoris ou des liens vers de tels sites.

### Création du filtre piège

Afin de créer un filtre piège, il faut analyser le Keylogger.

Un keylogger c’est un type de logiciel malveillant qui est concu pour enregistrer et surveiller les frappes clavier d'un utilisateur sans son consentement. Ceci va permettre à un attaquant d'intercepter et de collecter des informations sensibles (mots de passe, numéros de carte de crédit et autres données confidentielles).

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Figure 6 : Analyse KeyLoger

Voici les fichiers proposés :

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

Figure 7 : Programme KeyLogger

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

## Cas d’utilisation N°2-DNS empoisonné

Le service d'assistance de niveau 1 du client CNBC a soumis plusieurs tickets signalant une modification non sollicitée du DNS par défaut associé aux cartes réseau des utilisateurs, sans intervention de leur part.

L'expression "empoisonnement DNS" désigne une attaque malveillante visant à altérer les données stockées dans un serveur DNS (Domain Name System) afin de rediriger les utilisateurs vers des destinations non désirées. Le DNS est le système qui traduit les noms de domaine en adresses IP, facilitant l'accès aux sites web en utilisant des noms compréhensibles. L'ICANN[[22]](#footnote-22) est à l'origine du protocole DNS.

Pour obtenir l'adresse IP à partir d'un nom de domaine, on peut utiliser un DNS Resolver, accessible via les commandes "**nslookup**" ou "**dig**" dans un terminal ou une invite de commande (CLI).

La modification du DNS peut être effectuée dans les paramètres des adaptateurs réseau via le programme **ncpa.cpl**, géré par le système (svchost.exe).

### Identification de l’évènement à capturer

Pour détecter quel outil modifie le DNS sous Windows, on filtre les changements effectués par "svchosts.exe" afin de localiser le fichier dans le registre et l'interface où le DNS est modifié.

Les filtres appliqués sont : Process Name – contains - svchost.exe – Include Category – is – Write - Include

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Figure 8 : Filtres Process Name - svchost.exe - Include

Nous allons maintenant modifier le DNS en utilisant l’outil ncpa.cpl, pour cela, faire Windows + R et ajouter « ncpa.cpl » :

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 9 : Run - ncpa.cpl - DNS

Il suffit ensuite de cliquer sur l’adapteur réseau qui est connecté au réseau, puis de se rendre dans les paramètres du Protocole IPv4 :

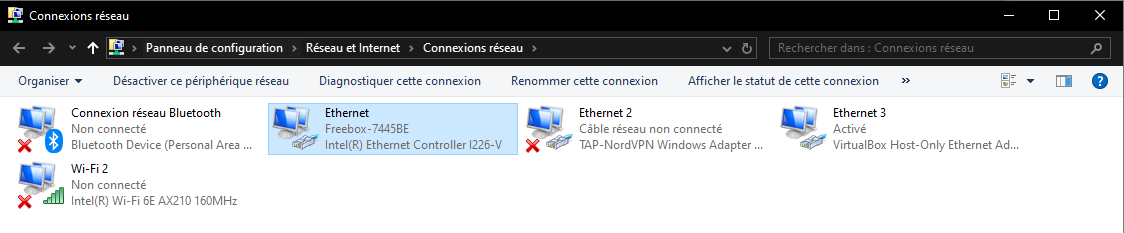


Figure 10 : Connexion réseau - ncpa.cpl - DNS

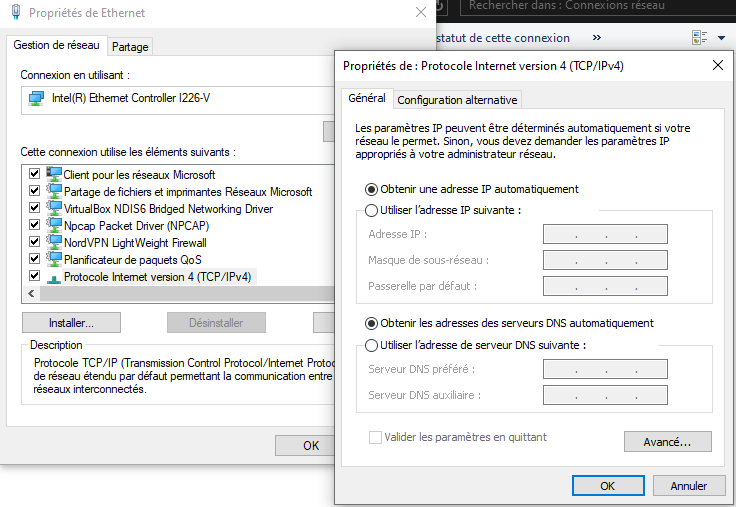


Figure 11 : Protocole IPv4 - panneau de configuration

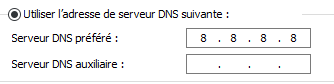


Figure 12 : Protocole Ipv4 - panneau de configuration - modification DNS

Sur Process Monitor nous pouvons observer les opérations qui ont été réalisées, nous avons la possibilité de rechercher « **Interfaces** » avec l’outil de recherche,  pour trouver plus facilement la ligne correspondant à la modification du fichier par le registre. L’opération est de type « **RegSetValue** » :

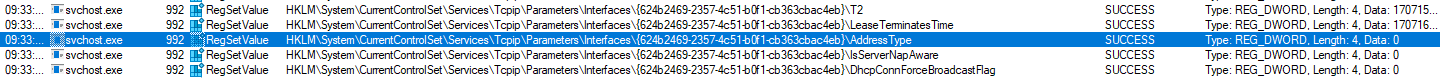


Figure 13 : Capture - path - modification DNS par le svchost

De cette manière nous savons que le fichier est inclus dans ce répertoire accessible via le registre de Windows :

**HKLM\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{5c620fb1-e343-483f-be61-1b0225468f4a}**

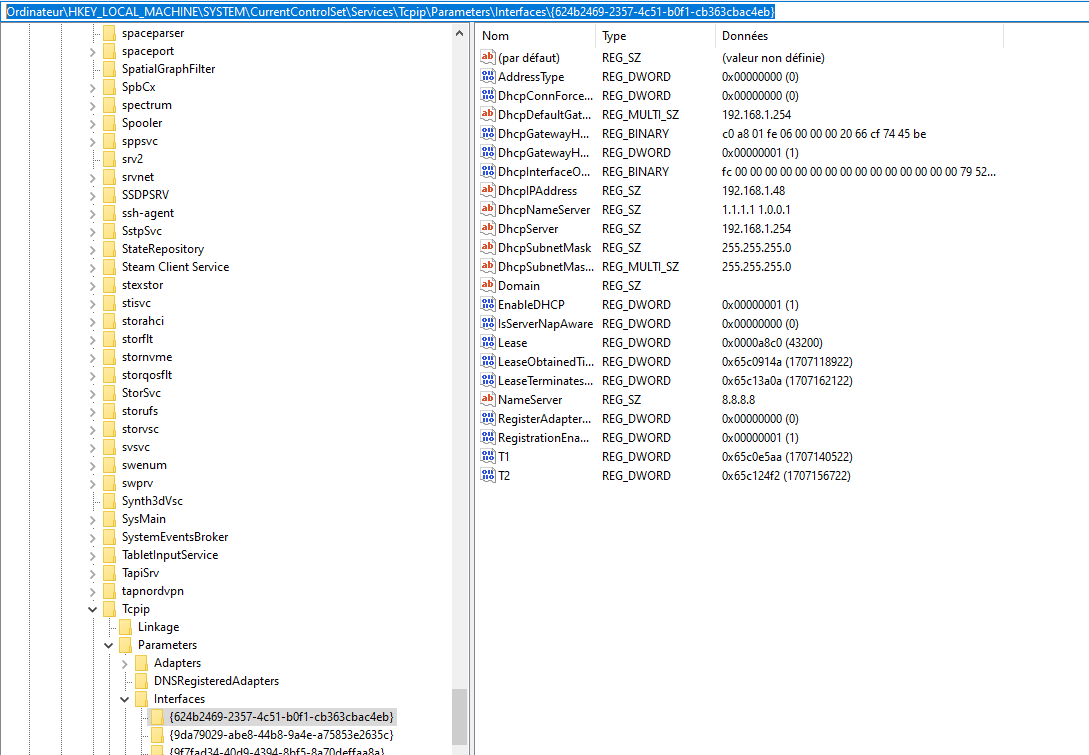


Figure 14 : répertoires du path - configuration du DNS - registre

# Virus Total

## Présentation de la plateforme virus total

Virus Total est une plateforme de service en ligne, qui analyse des fichiers et URL suspects, afin de détecter des logiciels frauduleux et du contenu malveillant à l’aide de moteur antivirus et de scanner de sites Web.

VirusTotal a été lancé en 2004 par Hispasec Sistemas. Son objectif est de fournir un service gratuit permettant aux utilisateurs de télécharger des fichiers suspects ou des URL et de les analyser avec plusieurs moteurs antivirus et outils de détection de logiciels malveillants.

L'agrégation de résultats de nombreux moteurs antivirus en un seul endroit permet aux chercheurs en sécurité, analystes de menaces et administrateurs système d'avoir une vue d'ensemble rapide sur la nature et la dangerosité des fichiers ou des URL suspectes.

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 base de données.

Son API publique permet aux développeurs d'intégrer ses fonctionnalités dans leurs propres outils et systèmes, ce qui renforce la sécurité des réseaux et des systèmes informatiques.

Voici à quoi ressemble la plateforme :

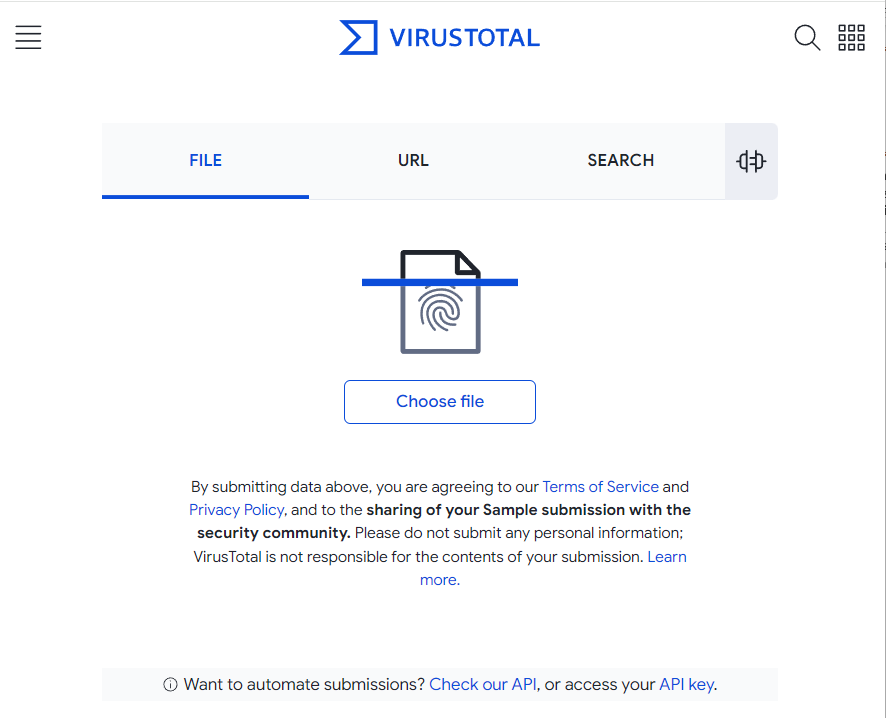


Figure 15 : site de VirusTotal

On peut ici y déposer des fichiers afin de les analyser

Par la suite, il a fallu créer un code, le voici :

import os

import time

import json

import vt

def lancer\_analyse():

global dossier\_analyse

API\_KEY = '91f9e38e83754bedaf43cd1fd31c881755cffc2ddec040cfeb7e82a35bca2817'

folder\_path = r'C:\Users\anissa\Desktop\BTS\_SIO' # Chemin du dossier à analyser

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

def scan\_directory(folder\_path):

for root, \_, files in os.walk(folder\_path):

for file\_name in files:

file\_path = os.path.join(root, file\_name)

print(f"Scanning {file\_path}")

analysis = scan\_file(file\_path)

time.sleep(4)

scan\_directory(folder\_path)

lancer\_analyse()

## Creation d’un robot de surveillance

Dans ce chapitre, on écrit un script python d’un robot de surveillance des fichiers contenus dans une liste de répertoire. Cette liste est configurable par l’administrateur et l’utilisateur autorisé, dans un fichier texte virus-hunter.csg. Le script planifié parcourt le fichier de configuration et pour chaque fichier contenu dans les répertoires à surveiller, le script appel l’API virus total, analyse la réponse de l’API et alerte l’administrateur en cas d’infection en plaçant le fichier infecté en quarantaine.

Le script appel l’API

L’analyse

On trace dans un fichier log

Nous avons utilisé Process Monitor pour suivre les processus, en appliquant un filtre que nous avons sauvegardé et déployé sur tous les ordinateurs de Vinci via une stratégie de groupe (GPO) lors de leur arrêt. De plus, un script au démarrage a ajouté une nouvelle entrée dans le registre nommée "Test", avec le chemin complet de l'exécutable "procmon64.exe". Enfin, le script a exporté la configuration de Procmon dans un dossier spécifique. L'objectif initial était de cibler la détection potentielle des ordinateurs infectés par des logiciels malveillants pour isoler une partie du problème.

Passons maintenant à la prochaine étape : l'utilisation d'un robot de surveillance programmé pour détecter les signatures malveillantes des fichiers suspects via l'API VirusTotal. Dès qu'une signature est repérée, le fichier suspect est mis en quarantaine, et toutes les informations importantes, telles que son emplacement et son nom, sont enregistrées dans un fichier journal. Enfin, un résumé est envoyé à l'administrateur.

Voici les instructions pour développer un prototype de script Python nommé VirusHunter.py :

1. Lecture des répertoires à inspecter : Le script extrait la liste des répertoires à vérifier à partir d'un fichier de configuration nommé "virus-hunter.cfg". Ce fichier est localisé à .\build\data\virus-hunter.cfg et répertorie les chemins des répertoires à examiner, un par ligne.

2. Exploration des répertoires : Le prototype parcourt minutieusement chaque répertoire désigné, inspectant les fichiers contenus ainsi que ceux des sous-répertoires pour assurer une analyse complète de l'environnement.

3. Analyse des fichiers via l'API VirusTotal : À chaque fichier, le prototype interroge l'API VirusTotal pour évaluer les risques potentiels. Les résultats de cette analyse sont enregistrés dans un fichier journal nommé "virus-hunter.log", disponible dans le répertoire « .\build\data ».

4. Lecture de la réponse de l'API : Après envoi d'un fichier à l'API VirusTotal pour analyse, le prototype examine la réponse. Les attributs sont encadrés par des guillemets simples (') au lieu des guillemets doubles habituels dans le format JSON, nécessitant une manipulation spécifique pour extraire les informations.

5. Gestion et traitement des alertes en cas de détection d'infection : En cas de détection d'une menace lors de l'analyse, notre script consigne les détails de cette alerte dans un fichier journal pour suivre les fichiers infectés et les informations cruciales sur les menaces.

6. Notification par SMS à l'administrateur : À la fin du processus, notre script enverra un SMS résumant les résultats de l'analyse à l'administrateur. L'utilisation d'une API de messagerie, comme celle de Vonage, facilite cette tâche en automatisant l'envoi du SMS, permettant à l'administrateur de réagir rapidement en cas de menace détectée.

## Analyse de la conception

Une analyse de conception est cruciale pour cerner avec précision les objectifs de chaque étape du processus de création du Robot de surveillance. Voici une analyse détaillée pour chacune des étapes à accomplir.

Lecture du fichier de configuration :

* Développons un script Python pour parcourir le fichier de configuration ligne par ligne, en extrayant les informations pertinentes.
* Utilisons un programme Python pour analyser le fichier de configuration et en extraire les données nécessaires.

Parcours des répertoires à scanner :

* Écrivons un script Python pour explorer les répertoires à scanner.
* Pour chaque répertoire configuré, répertorions les fichiers à scanner, y compris ceux des sous-répertoires.
* Utilisons le module Python os pour naviguer dans le système de fichiers et lister les fichiers.

Scan de chaque fichier :

* Concevons un script Python pour analyser chaque fichier en utilisant l'API VirusTotal.
* Utilisons le module Python vt-py pour interagir avec l'API VirusTotal et envoyer les fichiers à analyser.

Lecture de la réponse de l'API :

* Rédigeons un script Python pour interpréter la réponse de l'API VirusTotal.
* Comme la réponse de l'API n'est pas entièrement en JSON, utilisons le module Python ast pour analyser la réponse et extraire les informations pertinentes.

Traitement des alarmes :

* En cas de détection d'une infection, consignons les anomalies dans un fichier de log.
* Mettons également le fichier infecté en quarantaine pour éviter toute propagation de la menace.

Envoi d'un SMS récapitulatif à l'administrateur :

* Pour clore le processus, élaborons un script Python pour envoyer un SMS récapitulatif à l'administrateur.
* Nous pouvons intégrer l'API de Vonage pour cette fonctionnalité, facilitant ainsi la notification rapide des événements critiques.

Après la finalisation de tous les scripts, nous entamerons la création du fichier VirusHunter.py. Ce fichier agira comme un automate, exécutant séquentiellement chacun des scripts. En procédant ainsi, nous nous assurerons que chaque étape du processus est exécutée de manière ordonnée et efficace. Ensuite, nous effectuerons des tests unitaires approfondis pour vérifier les traces d'exécution et garantir le bon fonctionnement de l'ensemble du système.

# Mise en production

Pour planifier le script de surveillance, on peut utiliser « one shot », l’outil graphique de planification des jobs.

On peut également faire la même chose en ligne de commande, avec la commande « AT » ou la commande « schtasas /create » ou la commande « batch ».

J’ai pour ma part opté pour Visual Code Studio

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Figure 16 : Script Python

# Conclusion

En synthèse, l'utilisation de VirusTotal permet une analyse efficace des fichiers suspects, intégrée à un robot de surveillance automatisé. Ce dernier parcourt les répertoires désignés, analyse les fichiers, traite les alarmes et notifie l'administrateur en cas de menace. La mise en production peut se faire de manière automatisée via des outils de planification de tâches, garantissant une surveillance continue de la sécurité du système.

# Signature de code

La signature de code est la signature numérique de fichier exécutable et de script. Elle permet d’authentifier l’auteur du code.

La signature de code c’est un processus de sécurisation des logiciels qui associe une empreinte numérique unique à un code source. Cette empreinte (généralement sous forme de certificat numérique) permet d'authentifier l'origine du code et de garantir son intégrité.

En d'autres termes, la signature de code assure aux utilisateurs que le logiciel n'a pas été altéré depuis sa publication par l'auteur légitime. Cela renforce la confiance des utilisateurs dans les logiciels téléchargés, réduisant ainsi les risques de manipulation malveillante ou de compromission.

# Compréhension des certificats numériques et du code

Un certificat numérique de code lie une clé publique à l'identité d'une entité, comme un développeur ou une organisation, et est émis par une autorité de certification (AC) de confiance après vérification. Les AC peuvent être commerciales, gouvernementales ou internes aux organisations.

Ces certificats sont utilisés pour authentifier les développeurs ou organisations avant le téléchargement ou l'exécution de logiciels, garantissant l'intégrité des données et assurant la confidentialité des communications. Ils spécifient également les privilèges d'accès, facilitant la gestion des autorisations et sécurisant les transactions en ligne.

Dans le contexte du SSL/TLS, les certificats numériques sont essentiels pour sécuriser les communications sur Internet. Lorsqu'un utilisateur accède à un site sécurisé, le serveur envoie un certificat contenant sa clé publique et des informations sur le site. Si le certificat est valide, une connexion chiffrée est établie, protégeant les données échangées contre l'interception.

En bref, les certificats numériques sont cruciaux pour assurer l'authenticité, l'intégrité et la confidentialité des échanges sur Internet via SSL/TLS.

# certificats et python

Les fichiers Python sont des scripts ou des programmes écrits dans le langage Python, sans lien intrinsèque avec les certificats numériques. À l'inverse, un certificat numérique est un fichier électronique qui atteste de l'identité d'une entité, telle qu'une personne, une organisation ou un serveur.

Les certificats numériques sont principalement utilisés en sécurité informatique pour assurer l'authenticité et l'intégrité des communications électroniques. Dans le cas des fichiers exécutables comme les ".exe" sur Windows, un certificat numérique peut être intégré lorsqu'un développeur signe son code avant la compilation.

La signature numérique, issue du certificat, est ensuite ajoutée au code de l'exécutable. Au lancement, le système d'exploitation vérifie cette signature avec le certificat correspondant pour garantir l'origine légitime du fichier et son intégrité depuis la signature, empêchant ainsi toute altération par des tiers.

En résumé, bien que les fichiers Python ne soient pas accompagnés de certificats, il est possible d'en intégrer dans les fichiers exécutables pour garantir leur authenticité et leur intégrité. Ces certificats permettent de vérifier l'identité du signataire du code et d'assurer qu'il n'a pas été altéré depuis sa signature.

Voici une liste de principaux organismes de certification reconnus dans le domaine de la certification numérique :

Voici une liste concise des principaux organismes de certification reconnus dans le domaine de la certification numérique :

1. PKI (Public Key Infrastructure) : Infrastructure à clé publique

2. ICAO (International Civil Aviation Organization) : Organisation de l'aviation civile internationale

3. ETSI (European Telecommunications Standards Institute) : Institut européen des normes de télécommunications

4. AATL (Adobe Approved Trust List) : Liste de confiance approuvée par Adobe

5. WebTrust : Programme de certification de la sécurité et de la confidentialité des sites web

6. ISO (International Organization for Standardization) : Organisation internationale de normalisation

7. ANSI (American National Standards Institute) : Institut national américain de normalisation

8. CEN (European Committee for Standardization) : Comité européen de normalisation

9. DIN (Deutsches Institut für Normung) : Institut allemand de normalisation

10. EEMA (European Association for e-Identity and Security) : Association européenne pour l'e-identité et la sécurité- Geotrust

# Stockage du fichier exécutable

Dans le fichier exécutable (".exe"), le code du certificat numérique est intégré en tant que partie du processus de signature numérique. Cette pratique est couramment utilisée sur les systèmes Windows pour garantir l'authenticité et l'intégrité des logiciels exécutables. L'intégration du code du certificat permet au système d'exploitation de vérifier la signature numérique lors du lancement de l'exécutable, assurant ainsi que le fichier n'a pas été altéré depuis sa signature et que l'entité qui l'a signé est légitime.

# Vérification de certificat

La vérification de certificat pour un fichier exécutable (exe) est cruciale en cybersécurité. Cela sert à confirmer l'authenticité du fichier via un certificat numérique émis par une autorité de certification reconnue. Voici le processus :

1. Structure de l'EXE :

- Header DOS : affiche un message si exécuté sous DOS.

- Header PE : détails sur la structure PE, incluant tailles et adresses.

- Sections : contiennent le code exécutable, les données, et les ressources.

2. Vérification du certificat :

- Data Directory : section de l'en-tête PE indiquant l'emplacement du certificat.

- Certificat numérique : stocké dans une section spécifique (.rsrc ou .data), souvent chiffré.

3. Processus de vérification :

- Extraction du certificat : le certificat est extrait de son emplacement.

- Validation du certificat : vérification de la signature avec la clé publique de l'émetteur.

- Vérification de l'émetteur : confirmation de la fiabilité de l'autorité de certification.

4. Réaction du système :

- Selon le résultat, le système peut autoriser l'exécution, émettre un avertissement ou bloquer le fichier.

Globalement, cette procédure vérifie l'intégrité et l'authenticité d'un fichier exe avant son exécution, en s’assurant qu'il provient d'une source fiable et qu'il n'a pas été corrompu.

# Vérification par explorateur de fichier

L'Explorateur de fichiers Windows utilise les API CNG, le magasin de certificats, et TLS pour sécuriser les accès en ligne et vérifier les certificats SSL/TLS. Windows Defender SmartScreen alerte sur les téléchargements risqués. Un outil Windows spécifique permet l'analyse directe des certificats.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquementDans l’onglet « Signature », les détails du certificat sont apparents, mais on peut aussi les retrouver dans l’onglet « Détails ».

Figure 17 : Propriétés de Signature

Une image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquement

Figure 18 : Détails de Signature

Afin de vérifier le certificat il est primordial d’avoir les informations nécessaires. Ces dernières sont les suivantes :

-Nom

-Adresse de messagerie

-Date de signature

Il y a également les contre-signatures, indiquant le nom du signataire, l’adresse et l’horodateur, c’est-à-dire la date.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Figure 19 : Signature en détails

Dans l’avancé des propriétés nous pouvons

retrouver les détails de la signature :

# Vérification par python

Dans cette partie j’ai fait une fonction pour vérifier la légitimée des programmes suspectés par l’entreprise Vinci.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 20 : Morceau de code python

Grâce à ce code nous pouvons vérifier si le fichier .exe contient un certificat ou non. Si ce dernier est valide on aura accès au jeton et à d’autres informations.

# Vérification automatique de certificat

Afin de faciliter l’intégration de mon script python au programme principal, je l’ai créé afin qu’il soit utilisé en tant que fonction prédéfinie. Cela va permettre d’alléger le contenu du programme principal.

Dans le code suivant, nous voyons le processus d’analyse de certificat avec certutil.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Figure 21 : Execution du programme

Afin d’exécuter le programme, il faut spécifier le chemin d’accès du fichier en question et l’action que l’on souhaite utiliser. L’avantage est que cela se compile en une seule ligne de code.

En intégrant les fonctionnalités évoquées au programme Virus\_Hunter, on développe un système qui, d'une part, utilise VirusTotal pour détecter les signatures et codes malveillants, et d'autre part, assure une vérification de certificat pour confirmer la validité et la signature de chaque programme. Le processus consiste à analyser les fichiers dans "jeu\_dessais" selon les paramètres du fichier "cfg", à valider leur signature numérique, puis à isoler les programmes suspects tout en notifiant le technicien.

Une image contenant texte, capture d’écran, Police, menu

Description générée automatiquement

Figure 22 : Analyse du jeu d'essai

# Signtool et son fonctionnement

Signtool est un utilitaire de ligne de commande fourni par Microsoft dans le cadre du kit de développement logiciel (SDK) Windows. Son rôle principal est de permettre la signature numérique des fichiers exécutables (comme les fichiers .exe) et des bibliothèques de liens dynamiques (DLL) sous Windows. Cette signature numérique fournit une preuve de l'origine et de l'intégrité du fichier signé, ce qui est crucial pour la sécurité des logiciels distribués.

Voici comment il fonctionne :

1. Certificat numérique : Avant de signer un fichier, vous devez disposer d'un certificat numérique valide délivré par une autorité de certification (CA) de confiance. Ce certificat est associé à une clé privée qui permet de générer des signatures numériques uniques. Le certificat garantit votre identité en tant que signataire légitime.

2. Sélection du certificat : Lorsque vous utilisez Signtool, vous spécifiez le chemin vers le certificat que vous souhaitez utiliser pour la signature. Vous pouvez avoir plusieurs certificats sur votre système, chacun associé à une clé privée distincte.

3. Calcul du hachage : Avant de signer le fichier, Signtool calcule un hachage cryptographique (ou empreinte numérique) du contenu du fichier. Ce hachage est une valeur unique qui représente le contenu du fichier de manière sécurisée.

4. Création de la signature : Une fois le hachage calculé, Signtool utilise votre clé privée associée au certificat pour créer une signature numérique du hachage. Cette opération se fait en utilisant des algorithmes de cryptographie asymétrique, ce qui garantit que la signature est unique et liée à votre identité en tant que détenteur du certificat.

5. Incorporation de la signature : La signature numérique générée est ensuite incorporée dans le fichier signé. Elle est généralement ajoutée dans un en-tête ou un champ spécifique réservé aux signatures numériques, de manière qu'elle puisse être extraite et vérifiée ultérieurement.

6. Vérification de la signature : Les utilisateurs peuvent vérifier la signature d'un fichier en utilisant des outils fournis par Windows ou des bibliothèques logicielles appropriées. Lorsque le fichier est ouvert ou exécuté, le système d'exploitation peut recalculer le hachage du contenu du fichier et le comparer à celui stocké dans la signature. Si les hachages correspondent et que le certificat de signature est valide, le fichier est considéré comme authentique et non modifié depuis la signature.

En résumé, Signtool permet de garantir l'authenticité et l'intégrité des fichiers exécutables et des DLL sous Windows en utilisant des certificats numériques pour créer et vérifier des signatures numériques. Cela offre une couche de sécurité essentielle pour les logiciels distribués, en permettant aux utilisateurs de vérifier l'origine et l'intégrité des fichiers qu'ils téléchargent et exécutent.

# Vérification par Process Explorer

Process Explorer est un outil avancé de gestion des processus pour Windows qui montre les processus en cours d'exécution sur un ordinateur. Il fournit plus d'informations que le gestionnaire de tâches standard de Windows, notamment les détails des descripteurs et les DLL chargés.

Le logiciel est organisé en deux panneaux : le panneau supérieur affiche une liste des processus en cours, y compris des informations sur le nom du processus, l'ID du processus (PID), la description et le nom de l'entreprise. Le panneau inférieur, qui semble être masqué dans cette capture, montre les descripteurs et les DLL chargés par le processus sélectionné dans le panneau supérieur.

Une image contenant texte, capture d’écran, nombre, logiciel

Description générée automatiquement

Figure 23 : Analyse de Process Explorer

Ci-dessus, le processus `explorer.exe` indique que l'utilisateur (moi dans ce cas là) s'apprête à effectuer une action sur ce processus, comme le terminer ou en créer un vidage.

On peut aussi remarquer plusieurs instances de `chrome.exe`, ce qui suggère que le navigateur Google Chrome est ouvert avec de nombreux onglets ou extensions actifs, chacun étant traité comme un processus distinct pour l'isolation et la sécurité.

Il y a aussi d'autres processus liés à des logiciels et services spécifiques comme `AnyDesk` et `AMD Software`.

Le fait que l'utilisateur ait ouvert Process Explorer peut indiquer qu'il surveille les performances du système, recherche des processus suspects ou essaie de résoudre un problème.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

Figure 24 : Elements de sécurité du processus

Nous pouvons ici voir la fenêtre des propriétés pour une instance du processus `chrome.exe` dans Process Explorer. La fenêtre est ouverte sur l'onglet "Security" qui affiche les détails de sécurité du processus sélectionné.

Voici les détails notables:

- User : L'utilisateur sous lequel le processus s'exécute est indiqué par le SID (Security Identifier), qui est une chaîne unique de numéros utilisée pour identifier un utilisateur ou un groupe de sécurité. Il indique également la session et la session de connexion.

- Virtualized : Cela indique si le processus est virtualisé, ce qui signifie qu'il est isolé du système d'exploitation pour éviter qu'il ne modifie les fichiers du système ou d'autres processus.

- Groups : Cette section liste les groupes de sécurité auxquels appartient le processus. On y voit des références à des groupes tels que les administrateurs locaux, les utilisateurs authentifiés et d'autres.

- Privilèges : Plusieurs privilèges sont listés avec des drapeaux indiquant s'ils sont activés ou désactivés. Ces privilèges déterminent les opérations que le processus peut effectuer sur le système. Par exemple, le privilège `SeChangeNotifyPrivilege` est normalement utilisé pour recevoir des notifications de modification de fichiers ou de dossiers et est activé par défaut.

L'adresse mail affichée est la mienne, connecté au processus Chrome, ce qui pourrait suggérer qu'elle a été utilisée pour se connecter à un service Google ou une fonctionnalité de synchronisation dans Chrome.

C'est un outil puissant pour les professionnels de la cybersécurité et les administrateurs système, car il permet d'auditer les privilèges et les droits de sécurité des processus en cours, ce qui peut être crucial pour maintenir la sécurité d'un système.

# Table des illustrations

[Figure 1 : Process Monitor Filter 5](#_Toc161670899)

[Figure 2 : Process Monitor 6](#_Toc161670900)

[Figure 3 : Event repertoire 7](#_Toc161670901)

[Figure 4 : Dossier fichier Prefs 7](#_Toc161670902)

[Figure 5 : Ouverture du fichier PREF.JS 8](#_Toc161670903)

[Figure 6 : Analyse KeyLoger 8](#_Toc161670904)

[Figure 7 : Programme KeyLogger 9](#_Toc161670905)

[Figure 8 : Filtres Process Name - svchost.exe - Include 9](#_Toc161670906)

[Figure 9 : Run - ncpa.cpl - DNS 10](#_Toc161670907)

[Figure 10 : Connexion réseau - ncpa.cpl - DNS 10](#_Toc161670908)

[Figure 11 : Protocole IPv4 - panneau de configuration 10](#_Toc161670909)

[Figure 12 : Protocole Ipv4 - panneau de configuration - modification DNS 10](#_Toc161670910)

[Figure 13 : Capture - path - modification DNS par le svchost 11](#_Toc161670911)

[Figure 14 : répertoires du path - configuration du DNS - registre 11](#_Toc161670912)

[Figure 15 : site de VirusTotal 12](#_Toc161670913)

[Figure 16 : Script Python 15](#_Toc161670914)

[Figure 17 : Propriétés de Signature 18](#_Toc161670915)

[Figure 18 : Détails de Signature 18](#_Toc161670916)

[Figure 19 : Signature en détails 18](#_Toc161670917)

[Figure 20 : Morceau de code python 19](#_Toc161670918)

[Figure 21 : Execution du programme 19](#_Toc161670919)

[Figure 22 : Analyse du jeu d'essai 20](#_Toc161670920)

[Figure 23 : Analyse de Process Explorer 22](#_Toc161670921)

[Figure 24 : Elements de sécurité du processus 23](#_Toc161670922)

# Conclusion

1. <https://fr.wikipedia.org/wiki/Entreprise_de_services_du_num%C3%A9rique> [↑](#footnote-ref-1)
2. <https://www.next-decision.fr/wiki/failles-securite-informatique> [↑](#footnote-ref-2)
3. <https://www.fun-mooc.fr/fr/> [↑](#footnote-ref-3)
4. <https://www.inria.fr/fr> [↑](#footnote-ref-4)
5. <https://fr.wikipedia.org/wiki/Logiciel_antivirus> [↑](#footnote-ref-5)
6. <https://www.cisco.com/c/fr_fr/products/security/firewalls/what-is-a-firewall.html> [↑](#footnote-ref-6)
7. <https://blog.advancia-itsystem.com/pourquoi-les-antivirus-ne-sont-plus-suffisants/>

   <https://www.cert-ist.com/public/fr/SO_detail?code=201006_antivirus> [↑](#footnote-ref-7)
8. <https://www.vadesecure.com/fr/blog/rapport-sur-le-phishing-et-les-malwares-t3-2023> [↑](#footnote-ref-8)
9. <https://www.cybermalveillance.gouv.fr/tous-nos-contenus/bonnes-pratiques/10-mesures-essentielles-assurer-securite-numerique> [↑](#footnote-ref-9)
10. <https://www.microsoft.com/fr-fr/security/business/security-101/what-is-malware> [↑](#footnote-ref-10)
11. <https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation> [↑](#footnote-ref-11)
12. <https://en.wikipedia.org/wiki/Sysinternals> [↑](#footnote-ref-12)
13. <https://www.octawian.ro/fisiere/situri/asor/build/html/_downloads/122f95f9a032396603a837c53b125bb8/Russinovich_M_WinInternals_part1_7th_ed.pdf> [↑](#footnote-ref-13)
14. <https://ptgmedia.pearsoncmg.com/images/9780135462409/samplepages/9780135462409_Sample.pdf> [↑](#footnote-ref-14)
15. <https://fr.wikipedia.org/wiki/Mark_Russinovich> [↑](#footnote-ref-15)
16. <https://guardia.school/boite-a-outils/quest-ce-quun-malware.html> [↑](#footnote-ref-16)
17. <https://fr.wikipedia.org/wiki/Processus_(informatique)> [↑](#footnote-ref-17)
18. <https://fr.wikipedia.org/wiki/Processeur> [↑](#footnote-ref-18)
19. <https://fr.wikipedia.org/wiki/M%C3%A9moire_morte> [↑](#footnote-ref-19)
20. <https://fr.wikipedia.org/wiki/M%C3%A9moire_vive> [↑](#footnote-ref-20)
21. <https://fr.wikipedia.org/wiki/P%C3%A9riph%C3%A9rique_informatique> [↑](#footnote-ref-21)
22. <https://www.icann.org/resources/pages/what-2012-02-25-fr> [↑](#footnote-ref-22)