**République Algérienne Démocratique et Populaire**
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**
**Université de Bordj Bou-Arreridj**

**Faculté des Mathématiques et d'Informatique**
**Département d'Informatique**


***Mémoire de fin d'études En vue de l'obtention du Diplôme de Master II en Informatique***

***Réseau et multimédia***

*Thème :*

# *Développement d'un logiciel malware*

***Réalisé par :***

*M. BENMAKHLOUF FAYÇAL*

**Devant le jury composé de :**

| | | |
|---|---|---|
| ATTIA Abd El-Ouahab | MCA à l'université de BBA | Président |
| BEN MALEK Mounir | MAA à l'université de BBA | Examinateur |
| LALAMI Fatih | Invité | Examinateur |
| DR. MOUSSAOUI Ali | MCB à l'université de BBA | Rapporteur |

**Année universitaire : 2019-2020**

## *DEDICACES*

*Je dédie ce modeste travail et ma profonde gratitude à :*

- ✓ *Ma mere :qui m'a transmis la vie, l'amour, le courage,*
- ✓ *Mon père à qui m'adresse au ciel les vœux les plus ardents pour la conservation de sa santé et de sa vie, pour l'éducation qu'ils m'ont prodigué ; avec tous les moyens et au prix de toutes les sacrifices qu'ils ont consentis à mon égard, pour le sens du devoir qu'ils mon enseigné depuis mon enfance.*
- ✓ *Ma femme*
- ✓ *Mes enfants*
- ✓ *Mes chères sœurs*
- ✓ *Mes chères frères*
- ✓ *Mes*
- ✓ *Et toute ma Famille*

*Je dédie aussi ce projet de fin d'études aux responsables du notre département, à tous mes enseignants surtout à mon encadreur, et sans oublier mes collègues de 2 ème année master informatique et mes chers amis.*

*Benmakhlouf*
*Fayçal*

# *REMERCIEMENTS*


- *Avant tout, le grand et le vrai merci à Allah qui nous a donné la volonté et le courage pour la réalisation de ce travail.*
- *Nous tenons à remercier tout particulièrement Mr Dr* MOUSSAOUI *ALI notre encadreur de mémoire pour son aide, son soutien, ces conseils, sa patience, sa générosité, con ouverture d'esprit, sa disponibilité et ces analyses pertinentes qui ont contribué à rendre cette étude agréable et enrichissante.*
- *Nous souhaitons remercier nos examinateurs qui seront désignés d'avoir accepté de participer au jury de ce mémoire :*
- *Nous nous n'oublierons pas de remercier tout le corps enseignements de notre faculté*
- *Enfin, nous tenons à remercier tous ceux qui ont contribué d'une façon ou d'une autre à la réalisation de ce mémoire.*

# Table des matières

# Introduction générale

Aujourd'hui la sécurité devient un enjeu majeur pour tout le monde parce que chacun de nous est devenu une victime potentielle pour ceux qui convoitent l'information personnelle ou industrielle ou autre. Sécuriser ces informations, son réseau son matériel est un besoin plus que important.

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridique et humains nécessaires à la mise en place de moyen visant à empêcher l'utilisation non autorisée, le mauvais usage, la modification ou le détournement du système d'information [1] .

La sécurité informatique vise les objectifs suivants (C.A.I.D) [2]:

- ✓ La **C**onfidentialité : seules les personnes autorisées peuvent avoir accès aux informations qui leur sont destinées. Tout accès indésirable doit être empêche.

- ✓ **A**uthentification : les utilisateurs doivent prouver leur identité par l'usage de code d'accès. Cela permet de gérer les droits d'accès aux ressources concernées et maintenir la confiance dans la relation d'échange.

- ✓ **I**ntégrité : les données doivent être celles que l'on attend, et ne doivent pas être altérés, illicite ou malveillante.

- ✓ **D**isponibilité : l'accès aux ressources du système d'information doit être permanent et sans faille. Les services et ressources sont accessibles rapidement et régulièrement.

L'avancée technologique et la numérisation de cette technologie à donner lieu à des activités cybercriminelles. Ces derniers utilisent des logiciels malveillants (également connu sous le nom malware) pour vol financier, l'espionnage, le sabotage, vol de la propriété intellectuelle, …ect

La détection des codes malveillants est un problème important pour le chargé de la sécurité informatique.

## **Problématique** :

Les auteurs de logiciels malveillants utilisent des techniques pour échapper à la détection par les défenses traditionnelles telles que les pare-feux, les antivirus et les passerelles qui utilisent généralement des techniques basées sur les signatures et ne peuvent pas détecter les codes malveillants.

Les éditeurs d'antivirus commerciaux ne sont pas en mesure d'offrir une protection immédiate pour les logiciels malveillants car ils doivent les analyser pour créer leurs signatures.

La détection basée sur la signature nécessite une mise à jour de la base de données des signatures et si la signature n'est pas présente dans ladite base de données, le code malveillant ne sera pas détecté comme malveillant.

L'analyse des logiciels malveillants, qui est une tâche essentielle dans la sécurité informatique, fournit la compréhension nécessaire pour concevoir des contres mesures efficaces et des stratégies d'atténuation contre les différents logiciels malveillants.

L'analyse des programmes malveillants est devenue une compétence indispensable pour lutter contre les logiciels malveillants avancés et les attaques ciblées. Elle nécessite une connaissance bien équilibrée de nombreuses compétences et sujets. Son apprentissage demande du temps et exige de la patience.

Notre modeste travail se résume à présenter les techniques et outils d'analyse des logiciels malveillants dits malwares et le développement d'un virus macro comme preuve de concept desdits logiciels malveillants

## Organisation dudit mémoire

Le présent mémoire est organisé comme suit :

**Chapitre I :** intitule généralité sur l'analyse des logiciels malveillants :décrit les concepts de base de l'analyse des logiciels malveillants
**Chapitre II :** intitule analyse statique des logiciels malveillants : décrit la notion de l'analyse statique et présente les différents outils utilisés dans ladite analyse
**Chapitre III :** intitule analyse dynamique : décrit la notion de l'analyse dynamique et les différents outils utilisés de ladite analyse
**Chapitre IV** : intitule développement de virus macros comme preuve de concept des logiciels malveillants. Il décrit le développement d'un virus macros excel
**Conclusion générale** : résume notre travail.

# Chapitre 1 : Généralité sur l'analyse des logiciels malveillants

Au cours de l'analyse des logiciels malveillants, un ensemble de concepts et terminologies seront utilisés. Ces derniers seront définit dans le présent chapitre. Ce dernier est organisé comme suit :

1. Qu'est-ce qu'un logiciel malveillant dit malware?
2. Qu'est-ce que l'analyse des logiciels malveillants?
3. Pourquoi l'analyse des logiciels malveillants?
4. Techniques d'analyse des logiciels malveillants
5. Configuration de l'environnement de laboratoire

## 1 Qu'est-ce qu'un logiciel malveillant dit malware?

Un logiciel malveillant est un code qui effectue des actions malveillantes. Il peut prendre la forme d'un exécutable, script, code ou tout autre logiciel. Les attaquants utilisent des logiciels malveillants pour voler des informations sensibles, espionner le système infecté ou prendre le contrôle du système. Ils pénètrent généralement dans le système sans le consentement de la victime et peuvent être livrés via différents canaux de communication tels que e-mail, Web ou clés USB, ..ect.[3]

Un logiciel malveillant ou maliciel, aussi dénommé logiciel nuisible ou programme malveillant ou pourriciel (de l'anglais malware ), est un programme développé dans le but de nuire à un système informatique, sans le consentement de l'utilisateur dont l'ordinateur est infecté. De nos jours, le terme « virus » est souvent employé, à tort, pour désigner toutes sortes de logiciels malveillants. En effet, les maliciels englobent les virus, les vers, les chevaux de Troie, ainsi que d'autres menaces. [3]

### 1.1 Actions malveillantes :

ci-après quelques actions malveillantes effectuées par les logiciels malveillants :[4]

- ✓ Perturber les opérations informatiques
- ✓ Voler des informations sensibles, y compris des données personnelles, professionnelles et financières
- ✓ Accès non autorisé au système de la victime
- ✓ Espionner les victimes
- ✓ Envoi de spams
- ✓ S'engager dans des attaques par déni de service distribué (DDOS)
- ✓ Verrouiller les fichiers sur l'ordinateur et les conserver contre une rançon
- ✓ Contaminer des fichiers sur l'ordinateur de la victime en leur ajoutant un code exécutable virale.
- ✓ Gêner ou empêcher certaines applications de fonctionner
- ✓ Supprimer ou crypter des fichiers crées par l'utilisateur pour obtenir un rançon
- ✓ Désactiver les logiciels de sécurité de la cible afin de l'infecter.
- ✓ Intercepter les comptes et mots de passe afin de les exploiter dans des activités illégale telque le vol financier sur le web.

✓Installer un relai sur l'ordinateur de la victime pour commettre des méfaits via cet ordinateur piraté et ainsi cacher son identité.

## 1.2 Type de logiciels malveillants :

Malware est un terme large qui désigne différents types de programmes malveillants tels que les Chevaux de Troie, Virus, Vers et Rootkits.. Lors de l'analyse des logiciels malveillants (malwares), divers types de logiciels malveillants a traiter; certains de ces programmes malveillants sont classés en fonction de leurs fonctionnalités et de leurs vecteurs d'attaque, comme mentionné ici:[3]

- ✓ **VIRUS** ou **WORM**: Malware qui est capable de se copier et de se propager à d'autres ordinateurs. **Un virus** a besoin intervention de l'utilisateur, alors qu'un ver peut se propager sans intervention de l'utilisateur.
- ✓ **TROJAN**: Les logiciels malveillants qui se déguise en un programme régulier pour tromper les utilisateurs à l'installer sur leurs systèmes. Une fois installé, il peut effectuer des actions malveillantes telles que le vol de données sensibles, le téléchargement de fichiers sur le serveur de l'attaquant, ou surveiller les webcams.
- ✓ **BACKDOOR / Remote Access Trojan (RAT):** Ce type de trojan permet à l'attaquant d'accéder et d'exécuter des commandes sur le système compromis.
- ✓ **ADWARE:** Malware qui présente des annonces indésirables à l'utilisateur. Il est livré via le téléchargement gratuit et peut installer forcement un logiciel sur le système de la victime.
- ✓ **BOTNET**: Un groupe d'ordinateurs infectés par le même logiciel malveillant (appelé bots), en attente de recevoir des instructions à partir du serveur de commande et de contrôle contrôlé par l'attaquant. L'attaquant peut envoyer une commande à ces bots, qui peut effectuer des activités malveillantes telles que les attaques DDOS ou l'envoi de spam e-mails.
- ✓ **INFORMATIONS STEALER**: Malware conçu pour voler des données sensibles telles que les informations bancaires ou les frappes dactylographiées du système infecté. key loggers, spyware, sniffers, et form grabbers sont des exemples dudit malware
- ✓ **RANSOMWARE**: Malware qui détient le système pour obtenir une rançon en bloquant les utilisateurs sur leur ordinateur ou en chiffrant leurs fichiers.
- ✓ **ROOTKIT**: Malware qui fournit l'attaquant un accès privilégié au système infecté et cache sa présence ou la présence d'autres logiciels.
- ✓ **DOWNLOADER** (telechargeur) ou **DROPPER**: Malware conçu pour télécharger ou installer des composants de logiciels malveillants

Une ressource pratique pour comprendre les terminologies des malwares et définitions est disponible via le lien :https://blog.malwarebytes.com/glossary

On peut trouver des échantillons de logiciels malveillants (ou échantillons similaires) en effectuant une recherche de logiciels malveillants référentiels différents. Voici quelques-unes des sources où nous pouvant obtenir des échantillons de logiciels malveillants. Certaines de ces sources vous permettent de télécharger gratuitement échantillons de logiciels malveillants

(ou après inscription gratuite), et certains nous demandent de contacter le propriétaire pour créer un compte, après quoi vous serai en mesure d'obtenir les échantillons:

- ✓ Analyse hybride: https://www.hybrid-analysis.com/
- ✓ KernelMode.info: http://www.kernelmode.info/forum/viewforum.php?f=16
- ✓ VirusBay: https://beta.virusbay.io/
- ✓ Contagio décharge des logiciels malveillants: http://contagiodump.blogspot.com/
- ✓ AVCaesar: https://avcaesar.malware.lu/
- ✓ Malwr: https://malwr.com/
- ✓ VirusShare: https://virusshare.com/
- ✓ le zoo: http://thezoo.morirt.com/

Il existe des liens vers d'autres sources de logiciels malveillants dans le blog de Lenny Zeltser https: //zeltser.com/malware-sample-sources/.

## 2 Classification des logiciels malveillants [3]

La classification des logiciels malveillants en fonction de leurs fonctionnalités n'est pas toujours possible car un seul malware peut contenir plusieurs fonctionnalités, qui peuvent appartenir à une variété de catégories mentionnées précédemment. Par exemple, les logiciels malveillants peuvent inclure un composant de worm qui scanne le réseau à la recherche de systèmes vulnérables et peut déposer un autre composant malware tel qu'une backdoor ou un ransomware après une exploitation réussie.

## 3 Qu'est-ce que l'analyse des logiciels malveillants

L'analyse des malwares est l'étude du comportement des malwares. L'objectif de l'analyse des malwares est de comprendre le fonctionnement des logiciels malveillants et comment les détecter et les éliminer. Ça implique analyser le binaire suspect dans un environnement sûr pour identifier ces caractéristiques et ces fonctionnalités afin que de meilleures défenses puissent être construites pour protéger le réseau d'une organisation.[3]

## 4 Pourquoi l'analyse des logiciels malveillants ? [3]

Le but de l'analyse des logiciels malveillants est généralement de fournir les informations dont vous avez besoin pour répondre à une intrusion réseau. L'objectifs sera généralement de déterminer exactement ce qui s'est passé, localisé tous les ordinateurs et fichiers infectés.et comment mesurer et contenir ces dommages. Une fois que les fichiers nécessitant une analyse complète sont identifiés, il est temps de développer des signatures pour détecter les infections de logiciels malveillants sur le réseau.

L'analyse des logiciels malveillants peut être utilisée pour développer des signatures basées sur l'hôte et sur le réseau. Les signatures ou indicateurs basés sur l'hôte sont utilisés pour détecter le code malveillant sur les ordinateurs des victimes. Ces indicateurs identifient souvent les fichiers créés ou modifiés par le malware ou des modifications spécifiques qu'il apporte au registre. Contrairement aux signatures antivirus, les indicateurs de malware se concentrent sur ce que le malware fait à un système, pas sur les caractéristiques du malware lui-même, ce qui les rend plus efficaces pour détecter malware qui change de forme ou qui a été supprimé du disque dur.

Les signatures réseau sont utilisées pour détecter le code malveillant en surveillant le réseau trafic. Les signatures réseau peuvent être créées sans analyse de malware, mais les signatures créés à l'aide de l'analyse des logiciels malveillants sont généralement beaucoup plus efficaces, offrant un taux de détection plus élevé et moins de faux positifs.

Après avoir obtenu les signatures, l'objectif final est de comprendre exactement comment les logiciels malveillants fonctionnent. C'est souvent la question la plus posée par la haute direction, qui veut une explication complète d'une intrusion majeure. Les techniques approfondies qui seront présentés dans le présent mémoire permettront de déterminer le but et les capacités des programmes malveillants.

## 5 Techniques d'analyse des logiciels malveillants [4]

Le plus souvent, lorsque vous effectuez une analyse de logiciels malveillants, vous ne disposez que de logiciel malveillants exécutable, qui ne sera pas lisible par l'homme. Pour en comprendre le sens, on utilise une variété d'outils et d'astuces, chacun révélant une petite quantité d'informations.

Il existe deux approches fondamentales de l'analyse des logiciels malveillants: statique et dynamique. L'analyse statique consiste à examiner le malware sans l'exécuter. Par contre l'analyse dynamique implique l'exécution du malware. Les deux techniques sont en outre classées comme basique ou avancé.

Ces techniques ainsi que d'autres techniques sont détaillés comme suit :

- ✓ **Analyse statique:** C'est le processus d'analyse d'un binaire sans l'exécuter. Il est plus facile à réaliser et vous permet d'extraire les métadonnées associées au binaire suspect. L'analyse statique pourrait ne pas révéler toutes les informations nécessaires, mais elle peut parfois fournir des informations intéressantes qui aide à déterminer où concentrer nos efforts d'analyse ultérieurs.
- ✓ **Analyse dynamique** (analyse comportementale): Elle consiste à l'exécution du binaire suspect dans un environnement isolé et le suivi de son comportement. Cette technique d'analyse est facile à réaliser et donne des indications précieuses sur l'activité du binaire au cours de son exécution. Cette technique d'analyse est utile, mais ne révèle pas toutes les fonctionnalités du programme hostile.
- ✓ **Analyse Code**: C'est une technique avancée qui se concentre sur l'analyse du code pour comprendre le fonctionnement interne du binaire. Cette technique révèle des informations qui ne sont pas possible de déterminer à partir de l'analyse statique et analyse dynamique. Analyse du code est divisé en analyse de code statique et analyse de code dynamique. Analyse de code statique implique désassemblage du binaire suspect en regardant le code pour comprendre le comportement du programme, alors que Analyse de code dynamique implique le débogage du binaire suspect d'une manière contrôlée pour comprendre sa fonctionnalité. L'analyse du code exige une compréhension des concepts de langage de programmation et le système d'exploitation.
- ✓ **Analyse de la mémoire**:
  C'est la technique d'analyse de la RAM de l'ordinateur. Son intégration dans l'analyse des programmes malveillants aidera à acquérir une compréhension du comportement

du malware après l'infection. L'analyse de la mémoire est particulièrement utile pour déterminer la furtivité et évasives capacités du logiciel malveillant.

L'intégration de différentes techniques d'analyses dans l'analyse des logiciels malveillants peut révéler une mine d'informations contextuelles, qui se révélera être utile dans la recherche des logiciels malveillants.

## 6 Configuration de l'environnement de laboratoire [3]

Analyse d'un programme nuisible nécessite un environnement de laboratoire sûr et sécurisé, pour ne pas infecter le système. Un laboratoire d'analyse malware peut être très simple ou complexe en fonction des ressources disponibles (matériel, logiciel de virtualisation, licence Windows, etc.). Avant de commencer la mise en place d'un laboratoire, on n'a besoin de quelques composants:

- ✓ un système physique exécutant un système d'exploitation de base comme Linux ou Windows ou Mac os
- ✓ Installé un logiciel de virtualisation (par exemple VMware ou VirtualBox).

L'avantage d'utiliser une machine virtuelle est que, après avoir terminé l'analyse des logiciels malveillants, nous pouvons revenir à un état propre. Afin d'avoir un environnement de laboratoire sûr, on doit prendre les précautions nécessaires pour éviter que les logiciels malveillants infecte l'environnement virtualisé ou infecte le système physique (hôte). Lors de la mise en place du laboratoire virtualisé:, retenir les éléments ci-après:

1. Gardez le logiciel de virtualisation à jour. Cela est nécessaire car le logiciel malveillant pourra exploiter une vulnérabilité dans le logiciel de virtualisation, échapper à l'environnement virtuel, et infecter le système hôte.
2. Installer une nouvelle copie du système d'exploitation à l'intérieur de la machine virtuelle (VM), et ne pas garder les informations sensibles dans la machine virtuelle.
3. Si l'analyse exige que le malware doit atteindre l'Internet, alors il faut envisager d'utiliser hôte uniquement mode de configuration réseau ou restreindre votre trafic réseau au sein de votre environnement de laboratoire utilisant des services simulés.
4. Ne connectez aucun autre support amovible qui pourraient ensuite être utilisés sur les machines physiques, tels que les lecteurs USB.
5. Étant donné que vous analysez les logiciels malveillants Windows (généralement Exécutable ou DLL), il est recommandé de choisir un système d'exploitation de base tels que Linux ou MacOS X pour votre machine hôte au lieu de Windows. En effet, même si une échappe de logiciels malveillants Windows à partir de la machine virtuelle, il ne sera pas toujours en mesure d'infecter votre machine hôte.

# Chapitre II : L'analyse statique

L'analyse statique est la technique d'analyse du fichier suspect sans l'exécuter. C'est une méthode d'analyse initiale qui consiste à extraire des informations utiles du binaire suspect pour prendre une décision éclairée sur la façon de le classer ou de l'analyser et sur où concentrer les efforts d'analyse ultérieurs. Divers outils et techniques seront présentés dans le présent chapitre. Ce dernier est organisé ainsi :

- Identifier l'architecture cible du malware
- Empreinte digitale du malware
- Analyse du binaire suspect avec des moteurs antivirus
- Extraction de chaînes, de fonctions et de métadonnées associées au fichier
- Identifier les techniques d'obfuscation utilisées pour contrecarrer l'analyse
- Classer et comparer les échantillons de logiciels malveillants

Ces techniques peuvent révéler différentes informations sur le fichier. Il n'est pas nécessaire de suivre toutes ces techniques, et il n'est pas nécessaire de les suivre dans l'ordre présenté. Le choix de la technique à utiliser dépend de l'objectif de l'analyseur et du contexte entourant le fichier suspect.

## 1 Déterminer le type de fichier suspect

Au cours de l'analyse, déterminer le type de fichier binaire d'un suspect aidera à identifier le système d'exploitation cible du malware (Windows, Linux, etc.) -et l'architecture (plates formes 32 bits ou 64 bits). Par exemple, si le binaire suspect a un type de fichier Portable Executable (PE), qui est le format de fichier des fichiers exécutables Windows (.exe, .dll, .sys, .drv, .com,.ocx, et ainsi de suite), en déduit que le fichier est conçu pour cibler le système Windows . La plupart des logiciels malveillants Windows sont des fichiers exécutables se terminant par des extensions telles que .exe,.dll, .sys, etc. Mais il n'est pas recommandé de se fier uniquement aux extensions de fichiers. L'extension fichier n'est pas le seul indicateur du type de fichier. Les attaquants utilisent différentes astuces pour cacher leur fichier en modifiant l'extension du fichier et en changeant son apparence afin d'inciter les utilisateurs à l'exécuter. Au lieu de s'appuyer sur l'extension de fichier, la signature de fichier peut être utilisée pour déterminer le type de fichier.

Une signature de fichier est une séquence unique d'octets qui est écrite dans l'en-tête du fichier. Les différents fichiers ont des signatures différentes, qui peuvent être utilisées pour identifier le type de fichier. Les fichiers windows exécutables, également appelés fichiers PE (tels que les fichiers se terminant par .exe, .dll, .com, .drv,.sys, et ainsi de suite), ont une signature de fichier de **MZ** ou de caractères hexadécimaux **4D 5A** dans le premier deux octets du fichier. Le lien http://www.filesignatures.net/. présente les signatures des différents fichiers en fonction de leur extension.

### 1.1 Identification du type de fichier à l'aide de la méthode manuelle

La méthode manuelle pour déterminer le type de fichier est la rechercher de signature du fichier en l'ouvrant dans un éditeur hexadécimal. Un éditeur hexadécimal est un outil qui

permet à un examinateur d'inspecter chaque octet du fichier; la plupart des éditeurs hexagonaux offrent de nombreuses fonctionnalités qui aident à l'analyse d'un fichier. La capture d'écran ci-dessous montre la signature de fichier MZ dans les deux premiers octets lorsqu'un fichier exécutable est ouverte avec le HxD éditeur hexadécimal http://www.filesig ( https://mh-nexus.de/en/hxd/):

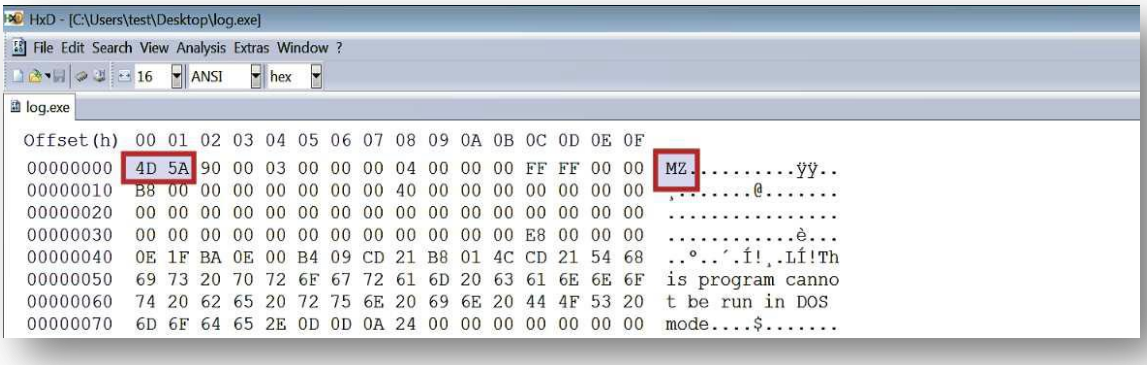



Figure 1

Vous avez de nombreuses options pour choisir les éditeurs hexadécimaux windows. Ces éditeurs hexadécimaux offrent différentes fonctionnalités. Pour les consulter, voir le lien https://en.wikipedia.org/wiki/Comparison_of_hex_editors.

## 1.2 Utilisation des outils pour identification type de fichier

L'autre méthode pratique pour déterminer le type de fichier est d'utiliser des outils d'identification des fichiers.

Sur les systèmes Linux, cela peut être réalisé en utilisant le fichier utilitaire. Dans l'exemple suivant, la commande de fichier a été exécutée sur deux fichiers différents. On peut voir que le premier fichier n'a pas d'extension, mais il est détecté comme un fichier exécutable 32 bits ( PE32) et le second fichier est un 64-bit ( PE32 +) exécutable:

$ **file mini**

mini: **PE32** executable (GUI) Intel 80386, for MS Windows

$ **file notepad.exe**

notepad.exe: **PE32+** executable (GUI) x86-64, for MS Windows

Sous Windows, CFF Explorer, partie de Explorer Suite ( http://www.ntcore.com/exsuite.php), peut être utilisé pour déterminer le type de fichier; il ne se limite pas à déterminer le type de fichier. Il est également un excellent outil pour le contrôle des fichiers exécutables (32 bits et 64 bits) et permet d'examiner la structure interne de PE, modifier les champs et les ressources extrait. Pour démontrer l'utilisation de la détection du type de fichier, prenons un exemple de fichier qui a été créé pour ressembler à un document Word en modifiant l'extension de .exe en .doc.exe. Dans ce cas, les attaquants ont profité du fait que, par défaut, "Masquer l'extension pour les types de fichiers connus" est activé dans les "Options d'affichage des dossiers Windows"; cette option empêche l'extension de fichier d'être afficher à l'utilisateur. La capture d'écran suivante montre l'apparence du fichier avec "Masquer l'extension pour les types de fichiers connus" activé:

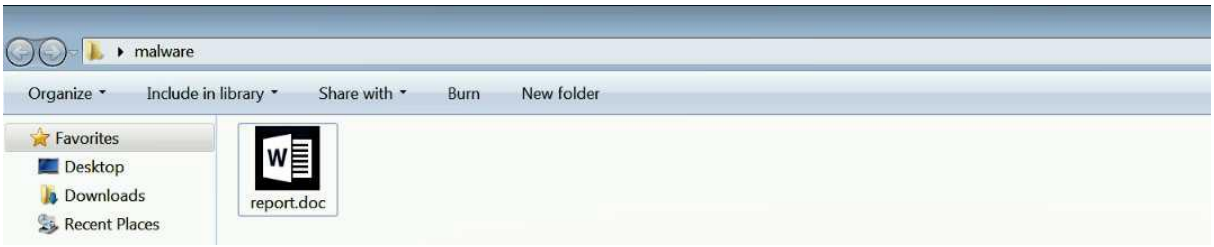



**Figure 2**

L'ouverture du fichier dans l'explorateur CFF révèle qu'il s'agit d'un fichier exécutable 32 bits et non d'un document, comme indiqué ici:

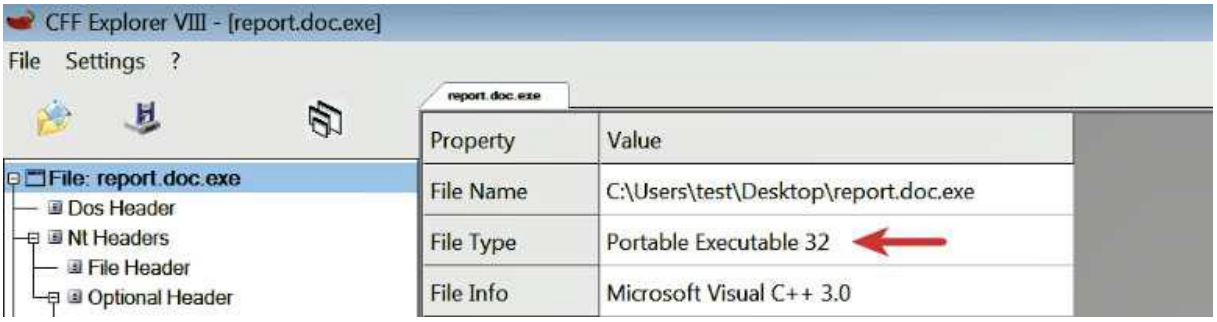



**Figure 3**

## 2 Empreinte digitale du logiciel malveillant

L'empreinte digitale implique la génération des valeurs de hachage cryptographiques pour le binaire suspect en fonction de son contenu. Les algorithmes de hachage cryptographique tels que MD5, SHA1 ou SHA256 sont considérés comme la norme de facto pour générer des hachages de fichiers pour le spécimen malware. La liste suivante décrit l'utilisation des hachages cryptographiques:

- ✓ Identifier un spécimen de malware basé sur le nom de fichier est inefficace car le même échantillon de malware peut utiliser des noms de fichiers différents, mais le hachage cryptographique qui est calculé en fonction du contenu du fichier restera le même. Par conséquent, le hachage cryptographique de votre fichier suspect sert d'identifiant unique au cours de l'analyse.
- ✓ Lors de l'analyse dynamique, lorsque le logiciel malveillant est exécuté, il peut se copier dans un emplacement différent ou déposez un autre logiciel malveillant. Avoir la cryptographie le hachage de l'échantillon peut aider à identifier si le fichier nouvellement déposé / copié l'échantillon est le même que l'échantillon d'origine ou un autre. Ces informations peuvent vous aider à décider si l'analyse doit être effectuée sur un seul échantillon ou plusieurs échantillons.
- ✓ Le hachage de fichier est fréquemment utilisé comme indicateur pour partager avec d'autres services de sécurité chercheurs pour les aider à identifier l'échantillon.
- ✓ Le hachage de fichier peut être utilisé pour déterminer si l'échantillon a déjà été détecté en recherchant en ligne ou en recherchant la base de données de plusieurs antivirus service d'analyse comme VirusTotal

Sur un système Linux, les hachages de fichiers peuvent être générés à l'aide de md5sum, sha256sum et Utilitaires sha1sum:

$ **md5sum log.exe**
6e4e030fbd2ee786e1b6b758d5897316 log.exe
$ **sha256sum log.exe**
01636faaae739655bf88b39d21834b7dac923386d2b52efb4142cb278061f97f log.exe
$ **sha1sum log.exe**
625644bacf83a889038e4a283d29204edc0e9b65 log.exe

**Figure 4**

Pour Windows, divers outils de génération de hachages de fichiers sont disponibles en ligne comme HashMyFiles (http: //www. microsoft.net/utils/hash_ my_ files.html). C'est l'un de ces outils qui génère valeurs de hachage pour un ou plusieurs fichiers, et il met également en évidence les hachages identiques avec la même couleurs. Dans la capture d'écran suivante, on peut voir que log.exe et bunny.exe sont les mêmes échantillons en fonction de leurs valeurs de hachage:


**Figure 5**

Le lien https: //en.wikipedia.org/wiki/Comparison_of_file_verification_software permet d'obtenir une liste et une comparaison des différents outils de hachage.

## 3 Analyse antivirus multiple

L'analyse du binaire suspect avec plusieurs scanners antivirus aide à déterminer si des signatures de code malveillant existent pour le fichier suspect. Le nom de signature pour un fichier particulier peut fournir des informations supplémentaires sur le fichier et ces capacités. La visite des sites web des fournisseurs d'antivirus ou la recherche de la signature dans la recherche moteur, donne plus de détails sur le fichier suspect. Ces informations peuvent aider dans l'enquête ultérieure et peut réduire le temps d'analyse.

### 3.1 Analyse du binaire suspect avec VirusTotal

VirusTotal (http://www.virustotal.com) est un service Web d'analyse de logiciels malveillants populaires. Il permet de télécharger un fichier, qui est ensuite analysé avec divers scanners antivirus et les résultats de l'analyse sont présentés en temps réel sur la page Web. En plus de télécharger des fichiers pour l'analyse, l'interface Web de VirusTotal permet d'effectuer des recherches dans leur base de données en utilisant un hachage, une URL, un domaine ou une adresse IP.

VirusTotal offre une autre fonctionnalité utile appelée VirusTotal Graph, construit au-dessus de l'ensemble de données VirusTotal. À l'aide de VirusTotal Graph, on visualise la relation entre le fichier soumit et ces indicateurs associés tels que les domaines, les adresses IP et les URL. Il permet également de pivoter et de naviguer sur chaque indicateur. Cette fonction est extrêmement utile si vous souhaitez déterminer rapidement les indicateurs associé à un binaire malveillant. La capture d'écran suivante montre les noms de détection d'un binaire de malware, et il peut être vu que le binaire a été scanné avec 67 moteurs antivirus; 60 d'entre eux l'ont détecté binaire comme malveillant.

Si on souhaite utiliser le graphique VirusTotal sur le binaire pour visualiser ces indicateurs, il faut cliquez simplement sur l'icône VirusTotal Graph et la connexion avec le compte VirusTotal :

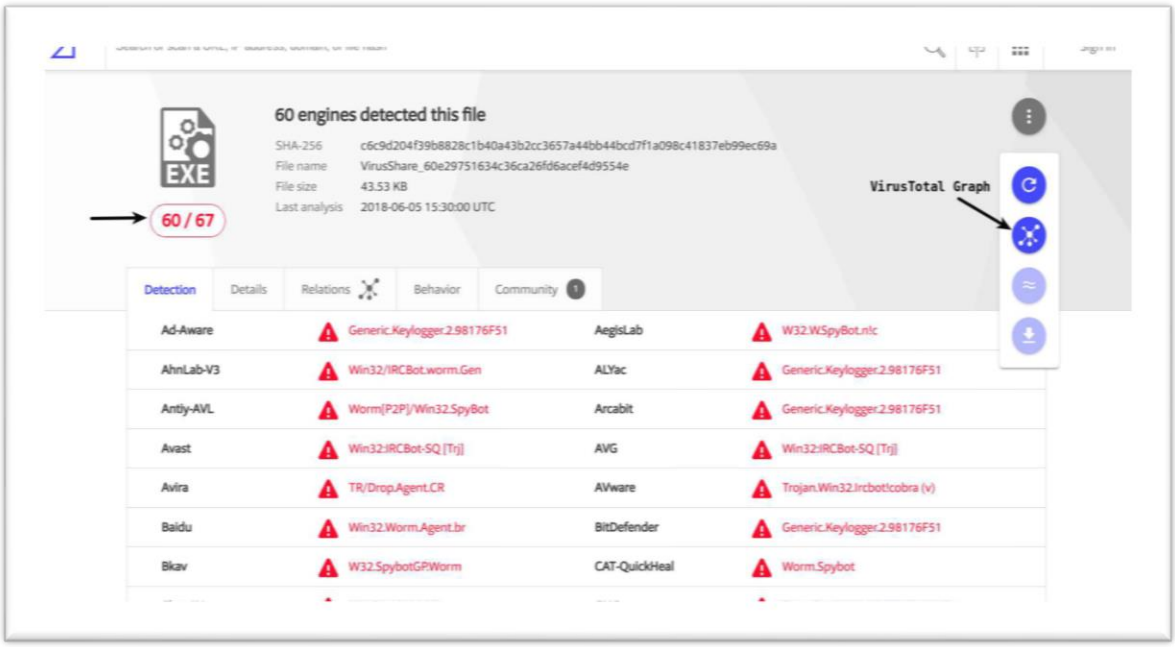



**Figure 6**

VirusTotal propose différents services privés (payants) qui permettent de rechercher des menaces et de télécharger des échantillons soumis à lui.

## 3.2 Interrogation des valeurs de hachage à l'aide de VirusTotal API publique

VirusTotal fournit également des fonctionnalités de script via son API publique. Il permet d'automatiser le fichier soumission, récupérer des rapports d'analyse de fichiers / URL et récupérer des rapports de domaine / IP.

Un script Python illustre l'utilisation de l'API publique de VirusTotal. Ce script prend la valeur de hachage ( MD5 / SHA1 / SHA256) comme entrée et les requêtes de la base de données VirusTotal.

L'autre alternative est d'utiliser des outils d'analyse tels que PE **PESTUDIO** ( https://www.winitor. com /) ou **PPEE** ( https://www.mzrst.com/). Lors du chargement du binaire, la valeur de hachage du binaire est interrogé automatiquement à partir de la base de

données VirusTotal et les résultats sont affichés, comme représenté sur la capture d'écran suivante:

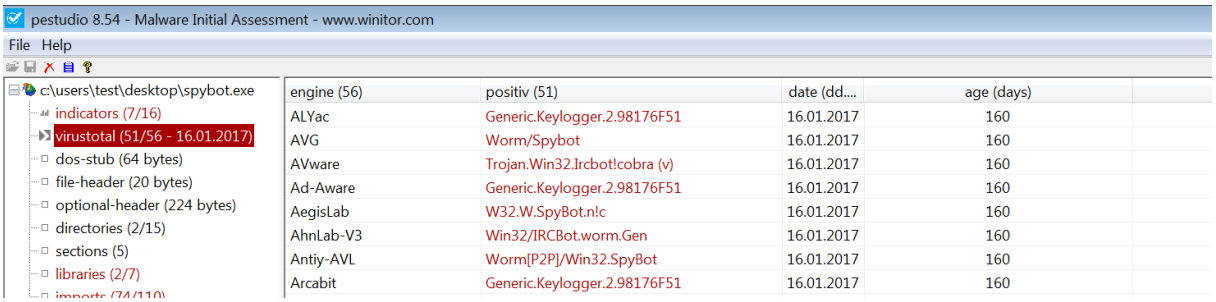



**Figure 7**

Si un binaire suspect ne soit pas détecté par les moteurs d'analyse anti-virus, cela ne signifie pas nécessairement que le binaire suspect est sûr. Ces moteurs anti-virus reposent sur les signatures et heuristiques pour détecter les fichiers malveillants. Les auteurs de logiciels malveillants peuvent facilement modifier leurs techniques d'obfuscation de code et d'utiliser pour contourner ces détections, à cause duquel certains des moteurs anti-virus pourrait ne pas détecter le binaire comme malveillant.

Lorsque vous téléchargez un fichier binaire à un site public, le binaire que vous soumettez peuvent être partagées avec des tiers et des fournisseurs. Le binaire suspect peut contenir des informations sensibles, personnelles ou propriétaire spécifique à votre organisation, il est donc conseillé de ne pas soumettre un binaire qui fait partie d'une enquête confidentielle aux services d'analyse anti-virus du public. La plupart des services d'analyse anti-virus sur le Web vous permettent de rechercher leur base de données existante de fichiers numérisés à l'aide des valeurs de hachage cryptographique (MD5, SHA1 ou SHA256); donc une alternative à soumettre le binaire est à la recherche basée sur le hachage cryptographique du binaire.

Lorsque vous soumettez un binaire aux moteurs d'analyse antivirus en ligne, les résultats d'analyse sont stockés dans leur base de données, et la plupart des données numérisées sont accessibles au public et peuvent être interrogés plus tard. Les pirates peuvent utiliser la fonction de recherche pour interroger le hachage de leur échantillon pour vérifier si leur binaire a été détectée. La détection de leur échantillon peut entraîner les attaquants de changer leurs tactiques pour éviter la détection.

## 4 Extraction de chaînes

Les chaînes sont des séquences de caractères imprimables ASCII et Unicode incorporées dans un fichier. L'extraction de chaînes peut donner des indices sur la fonctionnalité du programme et les indicateurs associés avec un binaire suspect. Par exemple, si un logiciel malveillant crée un fichier, le nom de fichier est stocké en tant que string dans le binaire. Ou, si un malware résout un nom de domaine contrôlé par l'attaquant, alors le nom de domaine est stocké sous forme de chaîne. Les chaînes extraites du binaire peuvent contenir références aux noms de fichiers, URL, noms de domaine, adresses IP, commandes d'attaque, registre clés, et ainsi de suite. Bien que les chaînes ne donnent pas une image claire de l'objectif et de la capacité d'un fichier, ils peuvent donner une idée de ce que les logiciels malveillants sont capables de faire.

## 4.1 Extraction de chaînes à l'aide d'outils

Pour extraire des chaînes d'un binaire suspect, vous pouvez utiliser l'utilitaire de chaînes sur les systèmes Linux. La commande **strings**, par défaut, extrait les chaînes ASCII d'au moins quatre caractères longue. Avec l'option **-a**, il est possible d'extraire des chaînes du fichier entier.

Sous Windows, **PESTUDIO** (https://www.winitor.com) est un outil pratique qui affiche à la fois l'ASCII et chaînes Unicode. PESTUDIO est un excellent outil d'analyse PE pour effectuer évaluation des logiciels malveillants d'un binaire suspect, et est conçu pour récupérer divers éléments utiles d' informations provenant d'un exécutable PE. La capture d'écran suivante montre certaines des chaînes ASCII et Unicode répertoriées par PESTUDIO; il nous aide en mettant en évidence certaines des chaînes notables de la colonne de la liste noire, qui nous permet de nous concentrer sur les chaînes intéressantes du binaire:

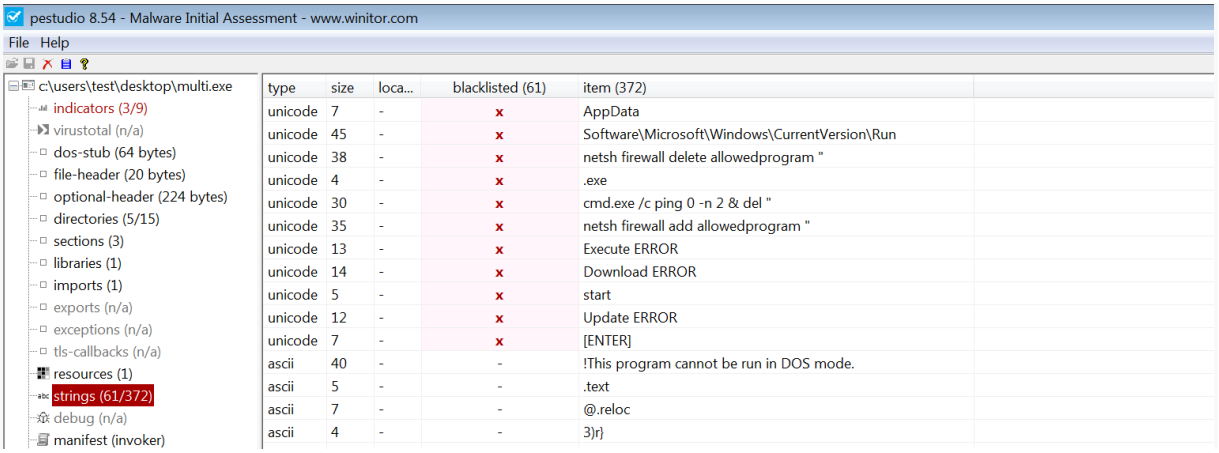



**Figure 8**

L'utilitaire de chaînes porté sur Windows par Mark Russinovich (https: //technet.microsoft.com/ en-us/sysinternals/strings.aspx) et PPEE (https://www.mzrst.com/) sont quelques-uns des autres outils qui peuvent être utilisés pour extraire les chaînes ASCII et Unicode.

## 4.2 Décodage de chaînes masquées à l'aide de FLOSS

La plupart du temps, les auteurs de logiciels malveillants utilisent des techniques simples d'obfuscation de chaîne pour éviter sa détection. Dans de tels cas, ces chaînes masquées n'apparaîtront pas dans l'utilitaire de chaînes et d'autres outils d'extraction de chaines. FireEye Labs Obfuscated String Solver (**FLOSS**) est un outil conçu pour identifier et extraire automatiquement les chaînes masquées des logiciels malveillants. Il permet de déterminer les chaînes que les auteurs de logiciels malveillants souhaitent masquer . Ledit outil peut également être utilisé tout comme l'utilitaire de chaînes pour extraire des chaînes lisibles par l'homme (ASCII et Unicode). Pour le télécharger sous Windows ou Linux , accéder au lien : https://github.com/fireeye/flare-floss.

# 5 Détermination de l'obfuscation des fichiers

Même si l'extraction de chaines de caractères est une excellente technique pour récolter des informations précieuses, souvent les auteurs de malwares obscurcissent leur

binaire de malwares. L'obfuscation est utilisée par les auteurs de logiciels malveillants pour protéger le fonctionnement interne des logiciels malveillants des chercheurs en sécurité, analystes de logiciels malveillants et rétro-ingénieurs. Ces techniques d'obscurcissement rendent difficile la détection/analyse du binaire. L'extraction des chaînes de ce type binaire entraîne très peu de chaînes et la plupart des chaînes sont masquées. Les auteurs de malwares utilisent souvent des programmes tels en tant que packers et crypteurs pour masquer leur fichier pour échapper à la détection des produits de sécurité tels que comme anti-virus et pour contrecarrer l'analyse.

## 5.1 Packers and Cryptors

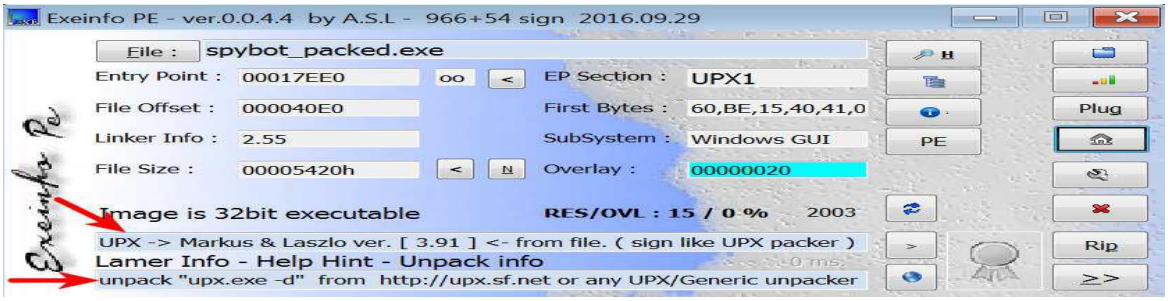
Un Packer est un programme qui prend l'exécutable comme entrée et utilise la compression pour obscurcir le contenu de l'exécutable. Ce contenu obscurci est ensuite stocké dans la structure d'un nouveau fichier exécutable; le résultat est un nouveau fichier exécutable (programme compressé) avec contenu obscurci sur le disque. Lors de l'exécution du programme compressé, il exécute une routine de décompression, qui extrait le binaire d'origine en mémoire pendant l'exécution et déclenche l'exécution.

Un Cryptor est similaire à un Packer, mais au lieu d'utiliser la compression, il utilise le cryptage pour obscurcir le contenu de l'exécutable et le contenu chiffré est stocké dans le nouveau fichier exécutable. Lors de l'exécution du programme chiffré, il exécute une routine de déchiffrement pour extraire le binaire d'origine dans la mémoire puis déclenche l'exécution.

## 5.2 Détection de l'obfuscation de fichiers à l'aide D'EXEINFO PE

La plupart des exécutables légitimes n'obscurcissent pas le contenu, mais certains exécutables peuvent le faire pour empêcher les autres d'examiner leur code. Lorsque on rencontre un échantillon emballé, il y a de fortes chances qu'il soit malveillant. Pour détecter les packers sous Windows, on peut utiliser un outil gratuit tel QU'EXEINFO PE (http://exeinfo.atwebpages.com/). Il a une utilisation facile GUI.

En plus de détecter les Packers, une autre fonctionnalité intéressante D'EXEINFO PE est qu'il donne des informations/références sur la façon de décompresser l'échantillon. Le chargement de l'exemple de malware Spybot emballé dans EXEINFO PE montre qu'il contient UPX, et il donne également un indice sur la commande à utiliser pour décompresser le fichier masqué; cela peut rendre l'analyse beaucoup plus facile:



## Figure 9

# 6 Inspection des informations d'en-tête PE

Les exécutables Windows doivent être conformes au **PE/COFF** (**P**ortable **E**xecutable/ **C**ommon **O**bject **F**ile **F**ormat). Le format de fichier PE est utilisé par les fichiers exécutables Windows (tels que .exe, .dll,.sys, .ocx et .drv) et ces fichiers sont généralement appelés fichiers PE (Portable Executable). Le fichier PE est une série de structures et de sous-composants contenant les informations requises par le système d'exploitation pour le charger en mémoire. Lorsqu'un exécutable est compilé, il inclut un en-tête (en-tête PE), qui décrit sa structure. Lorsque le binaire est exécuté, le chargeur du système d'exploitation lit les informations à partir de l'en-tête PE, puis charge le contenu binaire du fichier dans la mémoire. L'en-tête du PE contient des informations telles que l'endroit où l'exécutable doit être chargé en mémoire, l'adresse de début de l'exécution, la liste des bibliothèques/fonctions sur lesquelles l'application s'appuie sur et les ressources utilisées par le binaire. Examen des rendements d'en-tête PE une mine d'informations sur le binaire et ces fonctionnalités.

On peut obtenir une compréhension claire du format de fichier PE en chargeant un fichier suspect dans PE outils d'analyse. Voici quelques-uns des outils qui permettent d'examiner et de modifier la structure PE et ces sous-composants:

- *CFF Explorer:* http:/www.ntcore.com/exsuite.php
- *PE Internals:* http://www.andreybazhan.com/pe-internals.html
- *PPEE(puppy):* https://www.mzrst.com/
- *PEBrowse Professional:* http://www.smidgeonsoft.prohosting.com/pebrowsepro-file-viewer.html

## 6.1 Inspection des dépendances de fichiers et des importations

Habituellement, les logiciels malveillants interagissent avec le fichier, le registre, le réseau, etc. Pour effectuer une telle interaction, les logiciels malveillants dépendent fréquemment des fonctions exposées par le système.

Windows exporte la plupart de ces fonctions, appelées interfaces de programmation d'application (API), requis pour ces interactions dans les fichiers Dynamic Link Libary (DLL).

Importation des exécutables c'est l'appelle de ces fonctions généralement à partir de diverses DLL qui fournissent des fonctionnalités différentes. Les fonctions qu'un exécutable importe à partir d'autres fichiers (principalement des DLL) sont appelées fonctions importées (ou importations). Par exemple, si un exécutable de malware veut créer un fichier sur le disque, sous Windows, il peut utiliser une API CreateFile (), qui est exportée dans kernel32.dll. Pour appeler l'API, il faut d'abord chargez kernel32.dll dans sa mémoire, puis appelez la fonction CreateFile ().

Inspecter les DLL sur lesquelles un malware s'appuie et les fonctions API à partir desquelles il importe les DLL peuvent donner une idée de la fonctionnalité et de la capacité des logiciels malveillants et de ce qu'il faut anticiper lors de son exécution. Les dépendances de fichiers dans les exécutables Windows sont stockées dans la table d'importation de la structure de fichier PE. Dans l'exemple suivant, l'exemple de spybot a été chargé dans PESTUDIO. En cliquant sur le bouton des bibliothèques dans PESTUDIO affiche tous les fichiers DLL dont dépend l'exécutable et le nombre de fonctions importées de chaque DLL. Ce sont les fichiers DLL qui seront chargé en mémoire lors de l'exécution du programme:

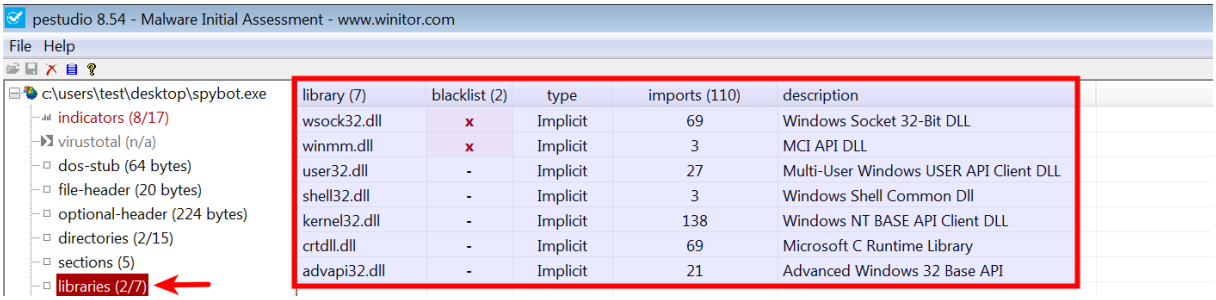



**Figure 10**

En cliquant sur le bouton importations dans PESTUDIO, les fonctions API importées depuis ces DLL. Dans la capture d'écran suivante, le malware importe les fonctions d'API liées au réseau (comme connecter, socket, écouter, envoyer, etc.) à partir de wsock32.dll, indiquant que le logiciel malveillant, lors de son exécution, se connectera très probablement à Internet ou effectuera activité réseau. PESTUDIO met en évidence les fonctions API fréquemment utilisées par malwares dans la colonne de la liste noire.

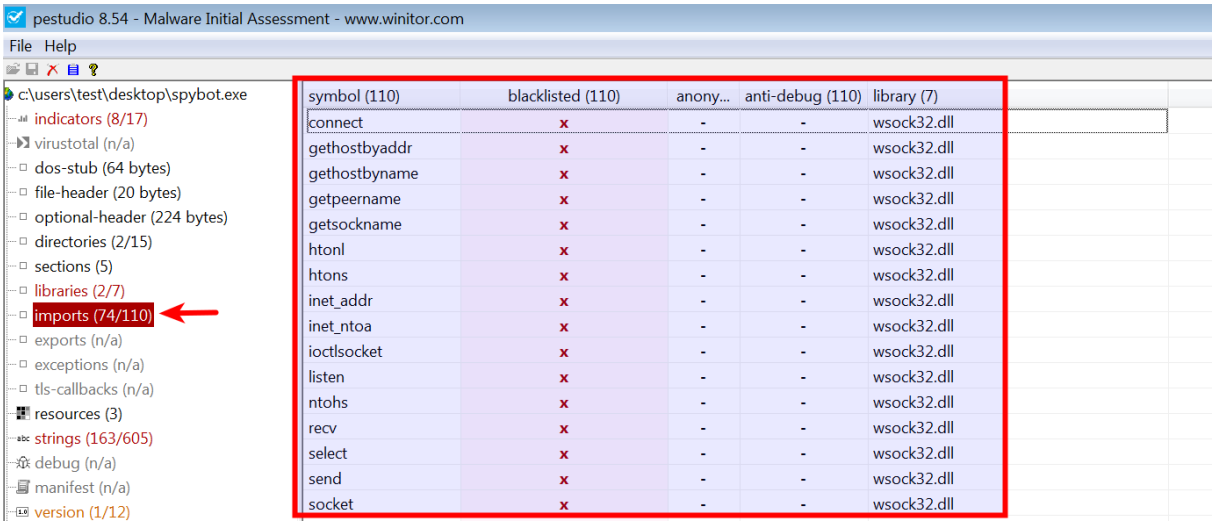



**Figure 11**

Parfois, les logiciels malveillants peuvent charger une DLL explicitement pendant l'exécution à l'aide d'appels d'API tels que comme LoadLibrary () ou LdrLoadDLL (), et il peut résoudre l'adresse de la fonction en utilisant l'API GetProcessAdress (). Les informations sur les DLL chargées pendant l'exécution ne sera pas présent dans la table d'import du fichier PE et ne sera donc pas affiché par ledit outils. En plus de déterminer la fonctionnalité des logiciels malveillants, les importations peuvent nous aider à détecter si un échantillon de logiciel malveillant est masqué. Si on rencontre un malware avec très peu d'importations, alors c'est une forte indication d'un binaire compressé. Pour démontrer cela, comparons les

importations entre l'échantillon décompressé de spybot et l'échantillon de spybot emballé. La capture d'écran suivante montre 110 importations dans le décompressé Échantillon de spybot:

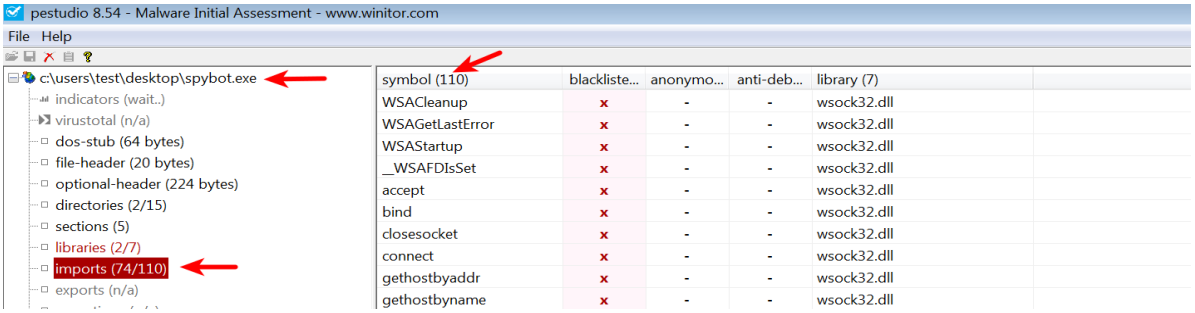



**Figure 12**

D'autre part, l'échantillon emballé de spybot ne montre que 12 importations:

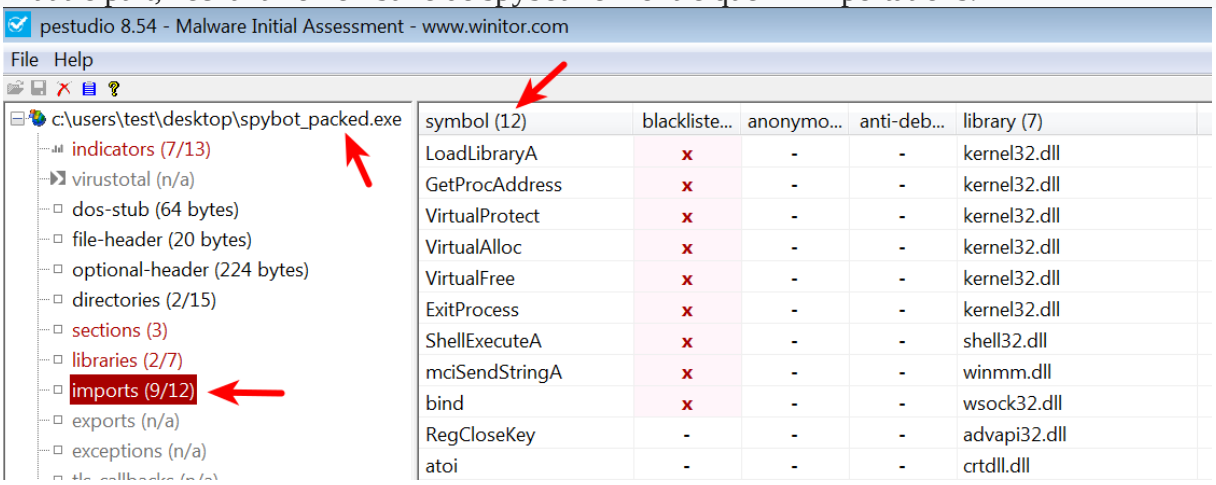



**Figure 13**

## 6.2 Inspection des exportations

L'exécutable et la DLL peuvent exporter des fonctions, qui peuvent être utilisées par d'autres programmes. En règle générale, une DLL exporte les fonctions (exportations) qui sont importées par l'exécutable. Une DLL ne peut pas s'exécuter seul et dépend d'un processus hôte pour exécuter son code. Un attaquant crée souvent une DLL qui exporte des fonctions contenant des fonctionnalités malveillantes. Pour exécuter les fonctions malveillantes de la DLL, il est en quelque sorte chargé par un processus qui appelle ces fonctions malveillantes. Les DLL peuvent également importer des fonctions d'autres bibliothèques (DLL) pour effectuer des opérations système. L'inspection des fonctions exportées peut donner une idée rapide des capacités de la DLL.

Dans l'exemple suivant, chargement d'une DLL associée à un logiciel malveillant appelé Ramnit dans PESTUDIO affiche ces fonctions exportées, donnant une indication de ces capacités. Lorsqu'un processus charge cette DLL, à un moment donné, ces fonctions seront appelées pour effectuer des activités:

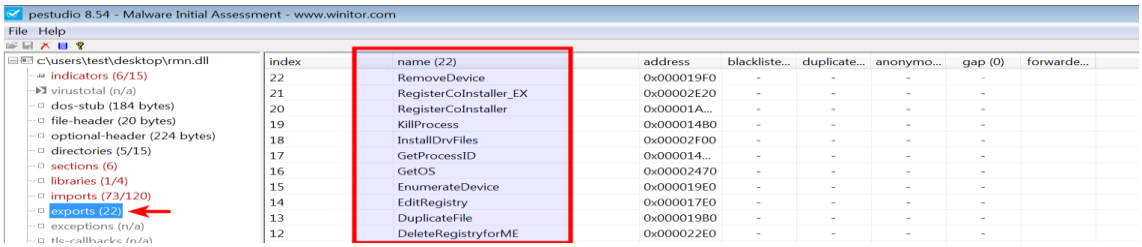



**Figure 14**

Les noms de fonction d'exportation ne donnent pas toujours une idée de la capacités. Un attaquant peut utiliser des noms d'exportation aléatoires ou faux pour induire en erreur notre analyse ou pour nous égarer.

## 1. Examen de la section table et des sections

Le contenu réel du fichier PE est divisé en sections. Les sections sont immédiatement suivi de l'en-tête PE. Ces sections représentent soit du code, soit des données et elles ont en mémoire attributs tels que lecture/écriture. La section représentant le code contient des instructions qui sera exécuté par le processeur, alors que la section contenant des données peut représenter différents types de données, telles que des données de programme en lecture/écriture (variables globales), import/export tables, ressources, etc. Chaque section a un nom distinct qui exprime le but de la section. Par exemple, une section avec le nom .text indique le code et a un attribut de lecture-exécution; une section avec le nom .data indique des données globales et a un attribut de lire/ écrire.

Lors de la compilation de l'exécutable, des noms de section cohérents sont ajoutés par le compilateurs. Le tableau suivant présente certaines des sections courantes dans un fichier PE:

| SECTION NAME | DESCRIPTION |
|---|---|
| .text or CODE | Contient exécutable code. |
| .data or DATA | Contient lire/ecrire data et variable globale |
| .rdata | Contient lire uniquement data. Parfois contient importation/exportation d'informations |
| .idata | Si elle est presente, elle contient table d'importation Sinon l'information importe est charge dans la section .rdata section. |
| .edata | Si elle est présente, elle contient exportation informations sinon l'informations exportés se trouvent dans la section .rdata section. |
| .rsrc | Cette section contient les ressources utilisé par l'executable comme les icones , fenêtre de dialogues, les menus, …ect |

Ces noms de section sont principalement destinés aux humains et ne sont pas utilisés par le système d'exploitation, ce qui signifie qu'il est possible pour un attaquant ou un logiciel d'obscurcissement de créer des sections avec noms différents. Si vous rencontrez des noms de section qui ne sont pas courants, vous devriez les traitez avec suspicion et une analyse plus approfondie est nécessaire pour confirmer la malveillance.

Le tableau des sections de l'en-tête PE ci-après présente des informations sur ces sections (telles que le nom de la section, où trouver la section et ces caractéristiques)

| Champ | Description |
|---|---|
| Names | Affiche les noms de sections. Dans ce cas l'executable contient quatre sections (.text, .data, .rdata and .rsrc). |
| Virtual-Size | Indique la taille de la section lorsqu'elle est chargé en mémoire |
| Virtual-Address | C'est l'adresse relatif (C'est offset de l'adresse de base de l'executable) lorque la section est en mémoire |
| Raw-size | Indique la taille de la section sur le disk . |
| Raw-data | Indique le offset dans le fichier lorsque la section peut être trouvé . |
| Entry-point | C'est le RVA (relatif Virtuelle Addresse) quand le code commence a s'exécuté. Dans ce cas, c'est le point d'entrée dans la section .txt |

Lorsque vous chargez un exécutable dans PESTUDIO et cliquez sur ces sections, il affiche la section informations extraites de la table de section et ces attributs (lecture / écriture, etc.). La capture d'écran suivante de PESTUDIO montre les informations de section pour un exécutable, et certains les champs pertinents de la capture d'écran sont expliqués ici:

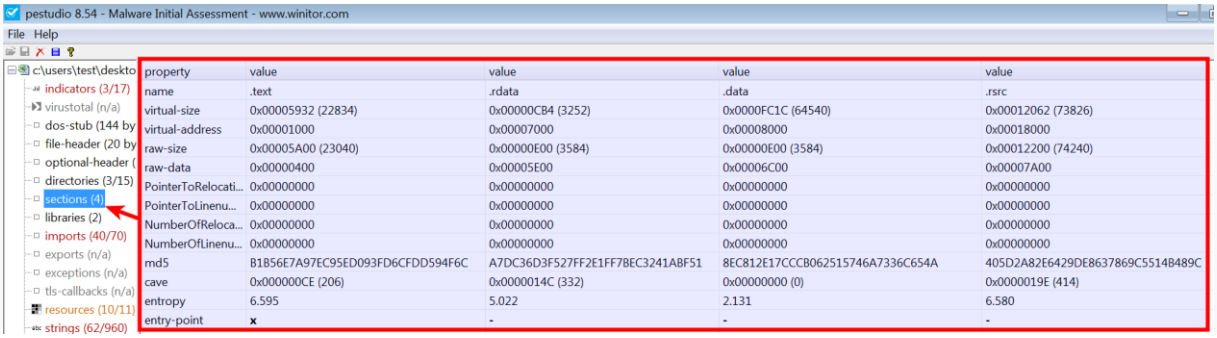



**Figure 15**

L'examen de la table de section peut également aider à identifier toute anomalie dans le fichier PE. La capture d'écran suivante montre les noms de section d'un malware emballé avec UPX; le malware contient les écarts suivants:

- Les noms de section ne contiennent pas de sections communes ajoutées par le compilateur (comme .text, .data, etc.) mais contiennent les noms de section UPX0 et UPX1.
- Le point d'entrée se trouve dans la section UPX1, indiquant que l'exécution commencera dans cette section (routine de décompression).
- En règle générale, la taille brute et la taille virtuelle doivent être presque égales, mais petites

Les différences sont normales en raison de l'alignement des sections. Dans ce cas, la taille brute est 0,indiquant que cette section ne prendra pas d'espace sur le disque, mais de taille virtuelle précise que, en mémoire, il prend plus de place (environ 127 ko). C'est un indication forte d'un binaire compressé. La raison de cet écart est que lorsque un binaire compressé est exécuté, la routine de décompression du packer copiera décompressé les données ou les instructions dans la mémoire pendant l'exécution.

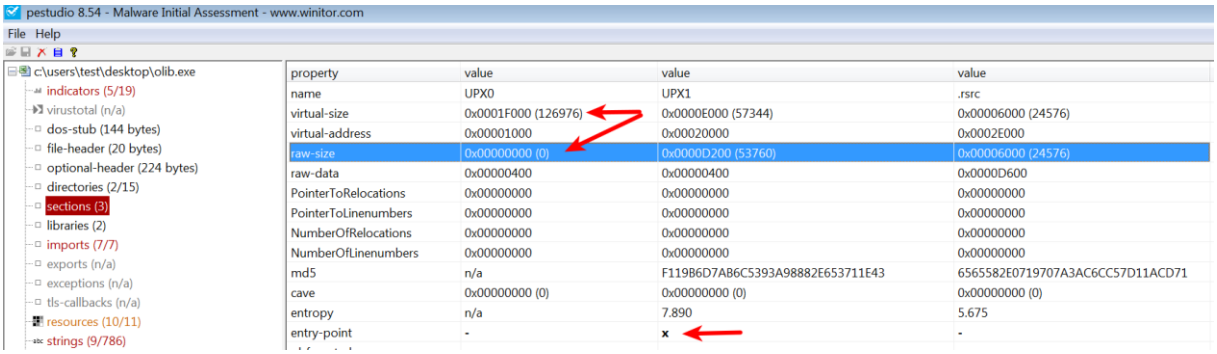



Figure 16

## 7 Examen de l'horodatage de la compilation

L'en-tête PE contient des informations qui spécifient quand le binaire a été compilé; l'examen de ce champ peut donner une idée de la date de création du malware. Les informations peuvent être utiles pour construire une chronologie de la campagne d'attaque. C'est aussi possible qu'un attaquant modifie l'horodatage pour empêcher un analyste de connaître le réel horodatage. Un horodatage de compilation peut parfois être utilisé pour classer les échantillons suspects. L'exemple suivant montre un binaire de malware dont l'horodatage a été modifié à un futur date en 2020. Dans ce cas, même si l'horodatage réel de la compilation n'a pas pu être détecté, ces caractéristiques peuvent vous aider à identifier un comportement anormal:

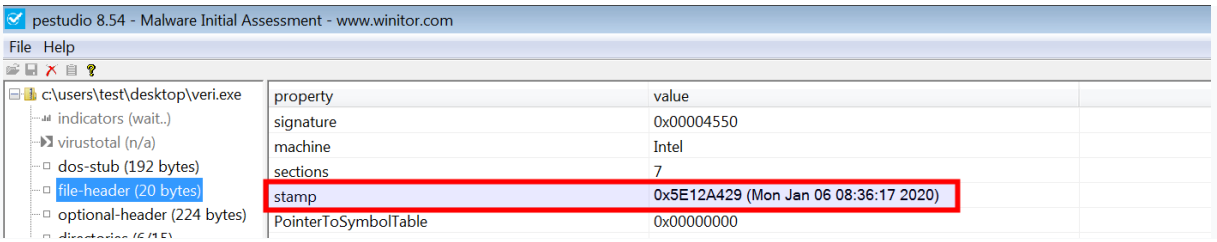



**Figure 17**

## 8 Examen des ressources PE

Les ressources requises par le fichier exécutable telles que les icônes, le menu, la boîte de dialogue et les chaînes sont stocké dans la section ressource (.rsrc) d'un fichier exécutable. Souvent, les attaquants stockent des informations telles que des binaires supplémentaires, des documents pièges et des données de configuration dans le la section de ressources, ainsi l'examen de la ressource peut révéler des informations précieuses sur un binaire. La section des ressources contient également des informations sur la version qui peuvent révéler des informations sur l'origine, le nom de l'entreprise, les détails de l'auteur du programme et les informations de copyright.

Resource Hacker (http: / / www. angusj. com/ resourcehacker/ ) est un excellent outil pour examiner, les masque (view) et extraire la ressource d'un binaire suspect. Un exemple de binaire qui ressemble à un fichier Excel sur le disque (notez comment l'extension de fichier est changée en .xls.exe),comme montré ici:


Figure 18

Le chargement d'un binaire malveillant dans le pirate de ressources affiche trois ressources (Icône, Binaire et Icône Groupe). Le spécimen de malware utilise l'icône de Microsoft Excel (pour donner l'apparence d'une feuille Excel):

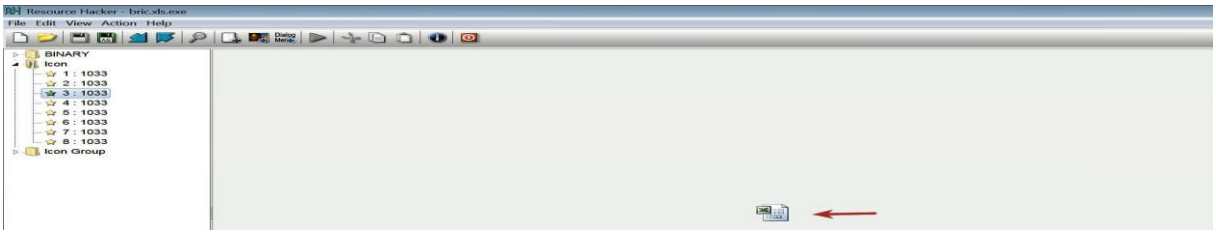



Figure 19

L'exécutable contient également des données binaires; l'un d'eux a une signature de fichier D0 CF 11 E0 A1 B1 1A E1. Cette séquence d'octets représente la signature de fichier pour un Microsoft Office fichier de document. Les attaquants, dans ce cas, ont stocké une feuille Excel piège dans la section des ressources. Lors de l'exécution, le malware est exécuté en arrière-plan, et cette feuille Excel piège est affiché à l'utilisateur comme un renvoi:

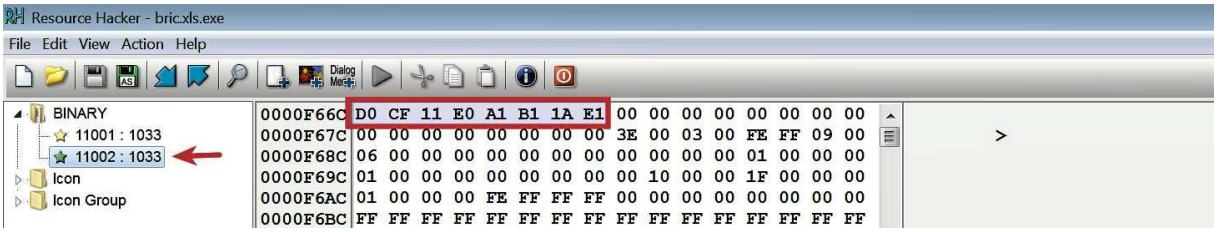
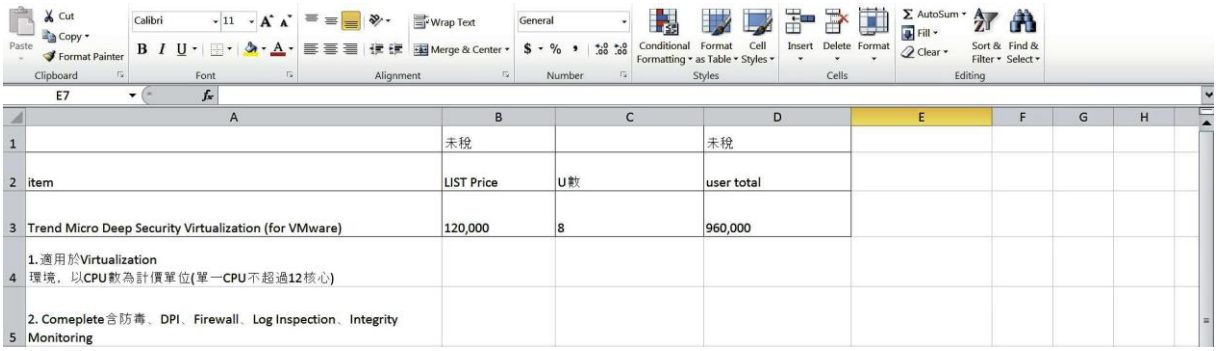



Figure 20

Pour enregistrer le binaire sur le disque, cliquez droit sur la ressource que vous souhaitez extraire et cliquez sur Enregistrez la ressource dans un fichier * .bin. Dans ce cas, la ressource a été enregistrée sous le nom sample.xls..

La capture d'écran suivante montre la feuille Excel piège qui sera affichée à l'utilisateur:



| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| 1 | | 未稅 | | 未稅 | | | | |
| 2 | item | LIST Price | U數 | user total | | | | |
| 3 | Trend Micro Deep Security Virtualization (for VMware) | 120,000 | 8 | 960,000 | | | | |
| 4 | 1.適用於Virtualization<br>環境，以CPU數為計價單位(單一CPU不超過12核心) | | | | | | | |
| 5 | 2. Comeplete含防毒、DPI、Firewall、Log Inspection、Integrity Monitoring | | | | | | | |

**Figure 21**

En explorant simplement le contenu de la section des ressources, on peut en apprendre beaucoup sur la caractéristiques des logiciels malveillants.

# Chapitre III. Analyse dynamique de base

L'analyse dynamique (analyse comportementale) consiste à analyser un échantillon en l'exécutant dans un environnement isolé et surveillance de ces activités, de son interaction et de son effet sur le système. [1]

Dans le présent chapitre, nous appuierons sur ces informations pour explorer davantage la nature, le but et la fonctionnalité du suspect binaire en utilisant l'analyse dynamique.[1]

Les points ci-après seront présentés :

- Outils d'analyse dynamique et leurs fonctionnalités
- Simulation de services Internet
- Les étapes de l'analyse dynamique
- Surveiller l'activité du malware et comprendre son comportement

## 1 Présentation de l'environnement de laboratoire

Lors de l'analyse dynamique, pour exécuter le spécimen de malware, on a besoin d'un environnement de laboratoire sûr et sécurisé pour protéger le système d'exploitation d'être infecté. L'environnement de laboratoire isolé qui a été présenté dans le premier chapitre du présent mémoire est exigé

## 2 Surveillance du système et du réseau

Lorsque le logiciel malveillant est exécuté, il peut interagir avec un système de différentes manières et effectuer différentes activités. Par exemple, lorsqu'il est exécuté, un malware peut engendrer un processus enfant, déposer des fichiers supplémentaires sur le système de fichiers, créer des clés de registre et des valeurs pour sa persistance et téléchargez d'autres composants ou prenez des commandes du serveur de commande et de contrôle.

Surveiller l'interaction d'un logiciel malveillant avec le système et le réseau permet d'obtenir une meilleure compréhension de la nature et du but du malware.

Lors de l'analyse dynamique, lorsque le malware est exécuté, on effectue divers activités de surveillance. L'objectif est de collecter des données en temps réel liées au comportement des malwares et son impact sur le système. La liste suivante décrit les différents types de surveillance effectuée lors de l'analyse dynamique:

- Surveillance du processus: implique la surveillance de l'activité du processus et l'examen des propriétés de résultat lors de l'exécution du malware
- Surveillance du système de fichiers: comprend la surveillance de l'activité du système de fichiers en temps réel lors de l'exécution de logiciels malveillants.
- Surveillance du registre: implique la surveillance des clés de registre (accédées/modifiées) et les données de registre qui sont lues/écrites par le binaire malveillant.
- Surveillance du réseau: implique la surveillance du trafic en direct vers et depuis le système lors de l'exécution de logiciels malveillants.

Les activités de suivi expliquées dans les points précédents aideront à rassembler les informations réseau liées au comportement du malware. Les prochaines sections couvriront l'utilisation pratique de ces activités. La section suivante présente les différents outils qui peuvent être utilisées pour effectuer ces activités de surveillance.

# 3 Outils d'analyse dynamique (surveillance)

Avant d'effectuer une analyse dynamique, il est essentiel de comprendre les outils a utiliser pour surveiller le comportement du malware. Si l'environnement de laboratoire est configuré d'une façons sécurisé, les outils d'analyse peuvent être téléchargés puis transférer / installer ces outils sur les machines virtuelles.

## 3.1 Inspection de processus avec Process Hacker

Process Hacker (http://processhacker.sourceforge.net/) est un logiciel open source polyvalent. C'est un outil qui aide à surveiller les ressources du système. C'est un excellent outil pour examiner les processus en cours d'exécution sur le système et pour inspecter les attributs de processus. Il peut également être utilisé pour explorez les services, les connexions réseau, l'activité du disque, etc.[1]

Une fois le spécimen de malware exécuté, cet outil peut vous aider à identifier le processus malveillant (son nom de processus et son ID de processus), et en cliquant avec le bouton droit sur un processus nom et en sélectionnant propriétés, vous serez en mesure d'examiner divers attributs de processus. On peut également cliquer avec le bouton droit sur un processus et y mettre fin. La capture d'écran suivante montre Process Hacker répertoriant tous les processus en cours d'exécution sur le system et les propriétés de wininit.exe: .[1]

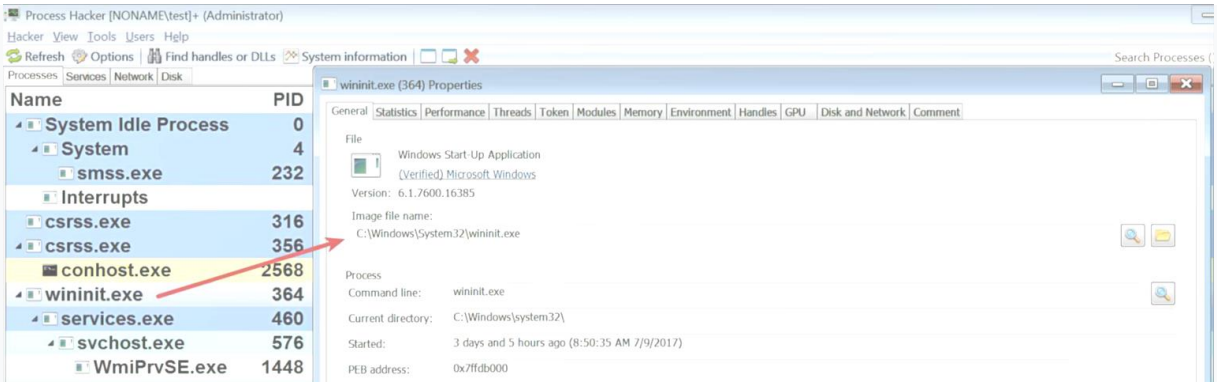



**Figure 22**

## 3.2 Détermination de l'interaction du système avec le processus Moniteur

Moniteur de processus (https://technet.microsoft.com/en-us/sysinternals/processmonitor.aspx) est un outil de surveillance avancé qui montre l'interaction en temps réel des processus avec le système de fichiers, le registre et l'activité des processus / threads. Lorsque vous exécutez cet outil (exécuté en tant qu'administrateur), vous remarquerez immédiatement qu'il capture tous les événements système, comme illustré dans la capture d'écran suivante. Pour arrêter de capturer les événements, vous pouvez appuyer sur Ctrl + E, et pour effacer tous les événements, vous pouvez appuyer sur Ctrl + X. La capture d'écran montre les activités capturées par Process Monitor sur un système propre.[1]:

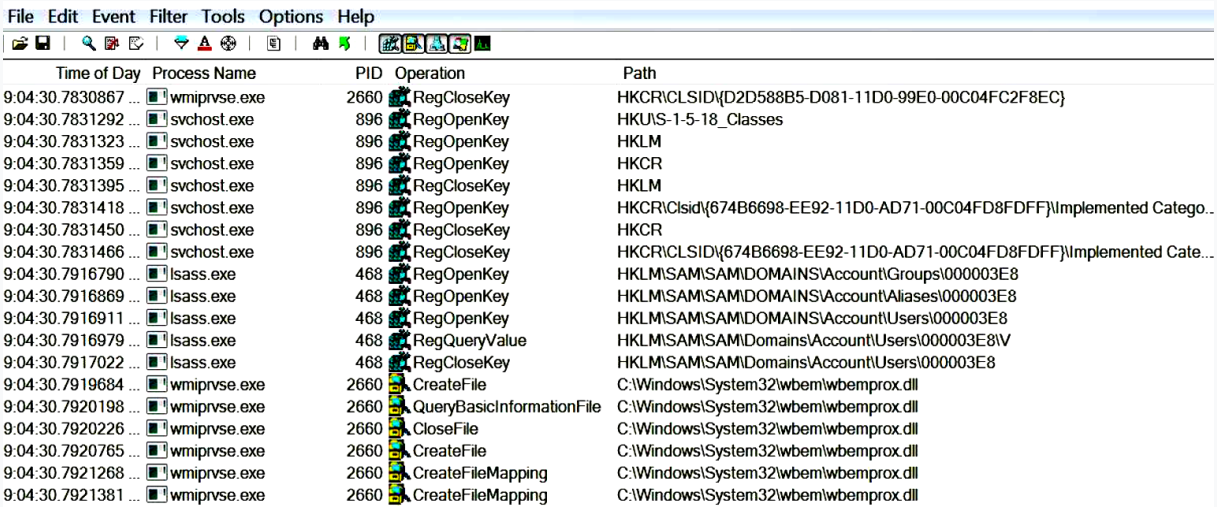



**Figure 23**

À partir des événements capturés par le moniteur de processus, vous pouvez voir que de nombreuses activités généré sur un système propre. Lors de l'analyse des logiciels malveillants, vous ne serez intéressé par les activités produites par le malware. Pour réduire le bruit, vous pouvez utiliser les fonctionnalités de filtrage qui masquent les entrées indésirables et vous permettent de filtrer les attributs. Pour accéder à cette fonction, sélectionnez le menu Filtre puis cliquez sur Filtre (ou appuyez sur Ctrl + L). Dans la capture d'écran suivante, le filtre est configuré pour afficher uniquement les événements liés au processus, svchost.exe.[1]:

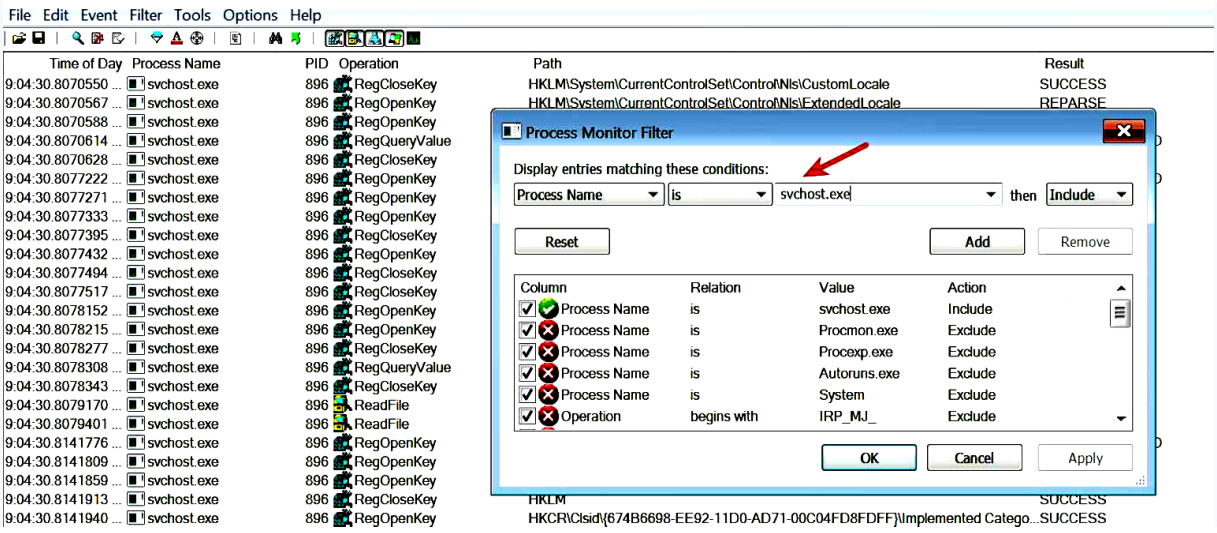



**Figure 24**

## 4 Journalisation des activités du système à l'aide de Noriben

Même si Process Monitor est un excellent outil pour surveiller l'interaction d'un malware avec le système, il peut être très bruyant et un effort manuel est nécessaire pour filtrer le bruit.

Noriben (https: / / github. com / Rurik / Noriben) est un script Python qui fonctionne en conjonction avec Process Monitor et aide à collecter, analyzer et rapporter les indicateurs d'exécution du malware.

L'avantage d'utiliser Noriben est qu'il est livré avec des filtres prédéfinis qui aider à réduire le bruit et vous permettre de vous concentrer sur les événements liés aux logiciels malveillants.

Pour utiliser Noriben, télécharger-le sur votre machine virtuelle Windows, extraire-le dans un dossier et copiez Process Surveillez (Procmon.exe) dans le même dossier avant d'exécuter Noriben.Python script, comme illustré dans la capture d'écran suivante.[1]:

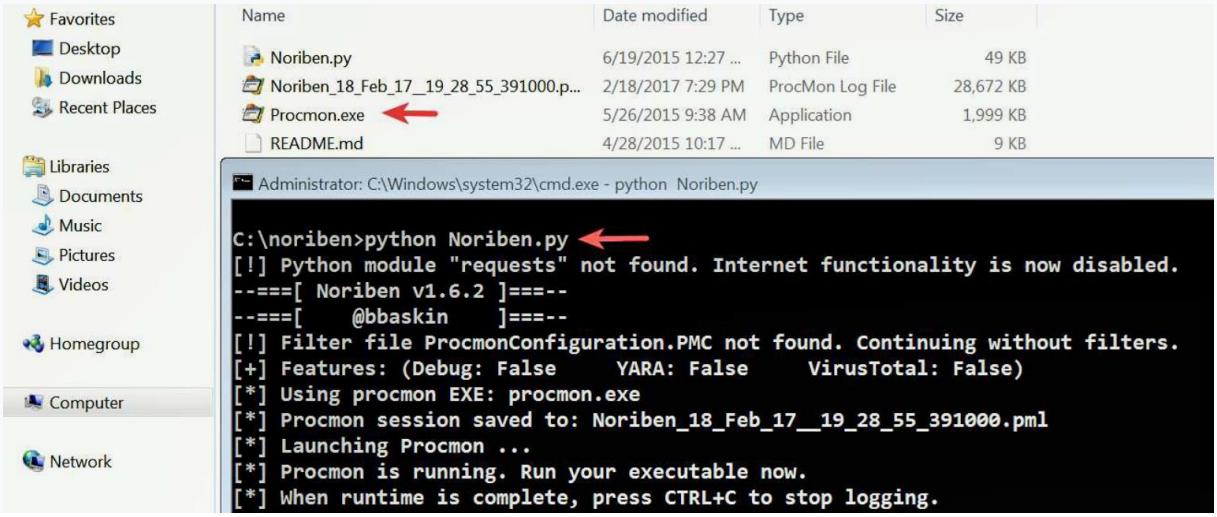



**Figure 25**

Pour exécuter Noriben, il lance Process Monitor. Une fois que vous avez terminé la surveillance, vous pouvez arrêter Noriben en appuyant sur Ctrl + C, ce qui mettra fin à Process Monitor. Une fois que terminé, Noriben stocke les résultats dans un fichier texte (.txt) et un fichier CSV (.csv) dans le même annuaire. Le fichier texte contient des événements séparés en fonction des catégories (comme processus, fichier, registre et activité réseau) dans des sections distinctes, comme illustré dans la capture d'écran suivante. Notez également que le nombre d'événements est bien moindre car il a appliqué des filtres prédéfinis qui réduit la plupart des bruits indésirables:


**Figure 26**

Le fichier CSV contient tous les événements (processus, fichier, registre et activité réseau) triés par la chronologie (l'ordre dans lequel les événements se sont produits), comme illustré dans la capture d'écran suivante:

Nonben_09_Jul_17__10_16_13_078000_timeline.csv

```
1  10:16:23,Registry,RegDeleteValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass
2  10:16:23,Registry,RegDeleteValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName
3  10:16:23,Registry,RegSetValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet, = 0
4  10:16:23,Registry,RegSetValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect, = 1
5  10:16:23,Registry,RegDeleteValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass
6  10:16:23,Registry,RegDeleteValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName
7  10:16:23,Registry,RegSetValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet, = 0
8  10:16:23,Registry,RegSetValue,notepad++.exe,3884,HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect, = 1
9  10:16:23,Process,CreateProcess,notepad++.exe,3884,%ProgramFiles%\Notepad++\updater\gup.exe -v7.32,3752
```

**Figure 27**

Le fichier texte et le fichier CSV peuvent donner des perspectives différentes. Consulter le fichier texte , si vous êtes intéressé par le résumé des événements en fonction de la catégorie, si vous êtes intéressé par la séquence des événements dans l'ordre dans lequel ils se sont produits, afficher le Fichier CSV.

## 5 Capture du trafic réseau avec Wireshark

Lorsque le malware est exécuté, vous voudrez capturer le trafic réseau généré en tant que résultat de l'exécution du malware; cela vous aidera à comprendre le canal de communication utilisé par le malware et aidera également à déterminer les indicateurs basés sur le réseau. Wireshark (https://www.wireshark.org/) est un renifleur de paquets qui vous permet de capturer le réseau trafic. Pour appeler Wireshark sous Linux, exécutez la commande $ sudo WireShark. Pour commencer à capturer le trafic sur une interface réseau, cliquez sur Capturer | Options (ou appuyez sur Ctrl + K), sélectionnez l'interface réseau et cliquez sur Démarrer .[1]:

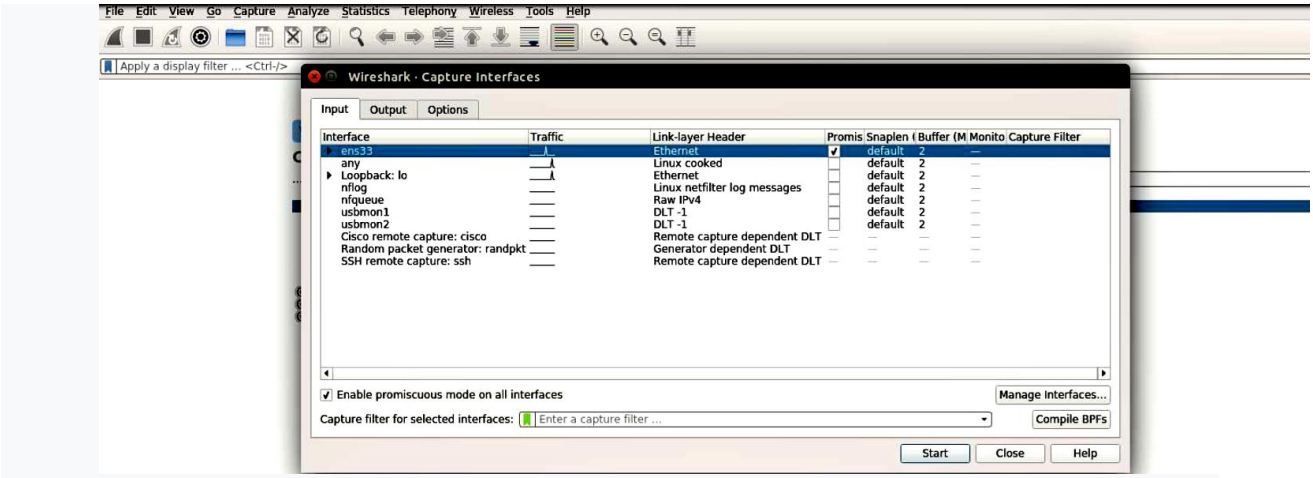



**Figure 28**

## 6 Simulation de services avec INetSim

La plupart des logiciels malveillants, lorsqu'ils sont exécutés, atteignent Internet (serveur de commande et de contrôle), et ce n'est pas une bonne idée de permettre au malware de se connecter à son serveur C2, et aussi parfois ces serveurs peuvent être indisponibles. Lors de l'analyse des logiciels malveillants, vous devez déterminer le comportement du malware sans lui permettre de contacter la commande et le contrôle réels (C2) serveur, mais en même temps, vous devez fournir tous les services requis par le malware afin qu'il puisse continuer son fonctionnement.

INetSim est une suite logicielle gratuite basée sur Linux pour simuler des services Internet standard (tels que comme DNS, HTTP / HTTPS, etc.).Une fois INetSim lancé, il simule divers services.

Simulation running.

Outre la simulation de services, INetSim peut enregistrer les communications, et il peut également être configuré pour répondre aux requêtes HTTP/HTTPS et renvoyer tous les fichiers en fonction de l' extension. Par exemple, si un logiciel malveillant demande un fichier exécutable (.exe) au serveur C2,INetSim peut renvoyer un fichier exécutable factice au malware. De cette façon, tu apprends à connaître ce que le logiciel malveillant fait avec le fichier exécutable après l'avoir téléchargé à partir du serveur C2.

L'exemple suivant illustre l'utilisation d'INetSim. Dans cet exemple, un malware sample a été exécuté sur la machine virtuelle Windows et le trafic réseau a été capturé à l'aide de Wireshark sur la machine virtuelle Linux sans appeler INetSim. La capture d'écran suivante affiche le trafic capturé par Wireshark. Il montre que le système Windows infecté (192.168.1.50) essaie de communiquer avec le serveur C2 en résolvant d'abord le domaine C2, mais parce que notre machine virtuelle Linux n'a pas de serveur DNS en cours d'exécution, ce domaine n'a pas pu être résolu (car indiqué par le message Port inaccessible) .[1]:

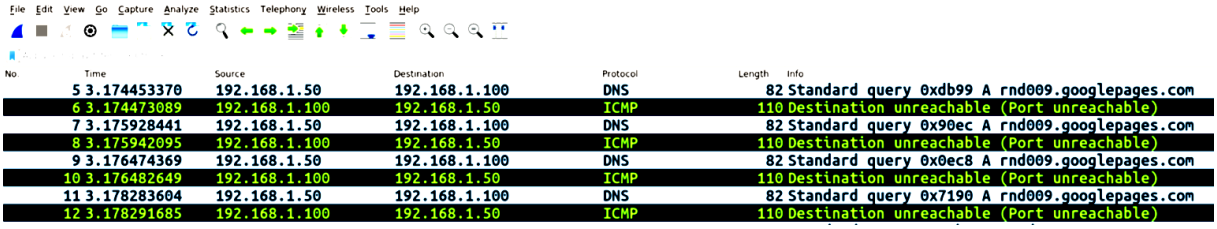



**Figure 29**

Cette fois, le malware a été exécuté et le trafic réseau a été capturé sur le Linux VM avec INetSim en cours d'exécution (simulation de services). À partir de la capture d'écran suivante, il peut être vu que le malware résout d'abord le domaine C2, qui est résolu sur l'adresse IP de la machine virtuelle Linux adresse 192.168.1.100. Une fois résolu, il établit ensuite une communication HTTP vers télécharger un fichier (settings.ini) .[1]:

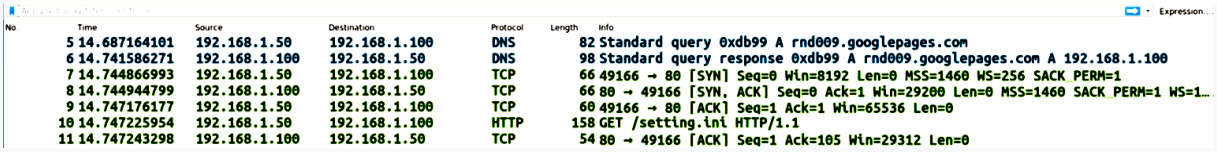



**Figure 30**

Dans la capture d'écran suivante, on peut voir que la réponse HTTP a été donnée par le Serveur HTTP simulé par INetSim. Dans ce cas, le champ User-Agent dans la requête http suggère que le navigateur standard n'a pas initié la communication et qu'une telle l'indicateur peut être utilisé pour créer des signatures réseau.[1]:

```
GET /setting.ini HTTP/1.1
User-Agent: AutoIt
Host: rnd009.googlepages.com
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Tue, 11 Jul 2017 05:18:16 GMT
Content-Length: 258
Content-Type: text/html
Connection: Close
Server: INetSim HTTP Server
```

**Figure 31**

En simulant les services, il était possible de déterminer que le malware télécharge un fichier depuis le serveur C2 après exécution. Un outil tel qu'INetSim permet à un analyste de sécurité de déterminer rapidement le comportement du malware et capturer son trafic réseau sans avoir à configurer manuellement tous les services (tels que DNS, HTTP, etc.).

Une autre alternative à INetSim est FakeNet-NG (https: // github.com/fireeye /flare-fakenet-ng), qui permet d'intercepter et de rediriger trafic réseau complet ou spécifique en simulant des services réseau.

## 7 Les étapes de l'analyse dynamique

Lors de l'analyse dynamique (analyse comportementale), vous suivrez une séquence d'étapes pour déterminer la fonctionnalité du malware. La liste suivante décrit les étapes impliquées dans ladite analyse.[1] :

- Revenir à l'instantané propre: cela inclut la restauration de vos machines virtuelles à un état propre.
- -Exécution des outils de surveillance / d'analyse dynamique: Dans cette étape, vous exécuterez les outils de surveillance avant d'exécuter le spécimen de malware. Pour tirer le meilleur parti des outils de surveillance abordés dans la section précédente, vous devez les exécuter avec les privilèges d'administrateur.
- Exécution de l'échantillon de malware: dans cette étape, vous exécuterez l'exemple de malware avec des privilèges d'administrateur.
- Arrêt des outils de surveillance: cela implique l'arrêt des outils de surveillance après l'exécution du binaire du malware pendant une durée spécifiée.
- Analyse des résultats: cela implique la collecte des données / rapports des outils de surveillance et de les analyser pour déterminer le comportement du malware et Fonctionnalité.

## 8 Assembler tout cela:

Analyser un malware exécutable Une fois que vous avez compris les outils d'analyse dynamique et les étapes impliquées dans ladite analyse , ces outils peuvent être utilisés ensemble pour glaner un maximum d'informations échantillon de malware. Dans cette section, nous effectuerons des analyses statiques et dynamiques pour déterminer les caractéristiques et le comportement d'un échantillon de malware (sales.exe) [1].

## 8.1 Analyse statique de l'échantillon

Commençons l'examen de l'échantillon de malware par une analyse statique. En analyse statique, l'exemple de logiciel malveillant n'est pas exécuté, il peut être effectué sur la machine virtuelle Linux ou la machine virtuelle Windows, en utilisant les outils et techniques décrits dans le chapitre 2 intitule l'analyse statique. On commencera par déterminer le type de fichier et le hachage cryptographique. basé sur ce qui suit :

- Sortie, le binaire du malware est un fichier exécutable 32 bits:

```
$ fichier sales.exe
sales.exe: exécutable PE32 (GUI) Intel 80386, pour MS Windows
$ md5sum sales.exe
51d9e2993d203bd43a502a2b1e1193da sales.exe
```

Les chaînes ASCII extraites du binaire à l'aide de l'utilitaire de chaînes contiennent des références à un ensemble de commandes par lots, qui ressemble à une commande pour supprimer des fichiers. Les chaînes montrent également une référence à un fichier batch (_melt.bat), qui indique que lors de l'exécution, le malware crée probablement un fichier batch (.bat) et exécute ces commandes batch.

L'examen des importations montre des références aux appels d'API liés au système de fichiers et au registre, indiquant la capacité du malware à effectuer des opérations de système de fichiers et de registre, comme mis en évidence dans la sortie suivante. La présence d'API appelle WinExec et ShellExecuteA, suggère la capacité du malware à invoquer d'autres programmes (créer un nouveau processus):

L'interrogation de la valeur de hachage dans la base de données VirusTotal affiche 58 détections antivirus et les noms de signature suggèrent que nous avons probablement affaire à un échantillon de malware appelé Sumac vénéneux.

Pour effectuer la recherche par hachage à partir de VirusTotal, vous avez besoin d'un accès Internet et si vous souhaitez utiliser l'API publique VirusTotal, vous avez alors besoin d'une clé API, qui peut être obtenue par inscription à un compte VirusTotal:

Les résultats des outils de suivi ont été collectés et examinés pour comprendre le comportement du malware déterminé à partir de différents outils de surveillance:

- Lors de l'exécution de l'exemple de logiciel malveillant (sales.exe), un nouveau process, iexplorer.exe, a été créé avec un ID de processus de 1272. Le processus l'exécutable se trouve dans le répertoire % Appdata%. La capture d'écran suivante est la sortie de Process Hacker montrant le processus nouvellement créé:

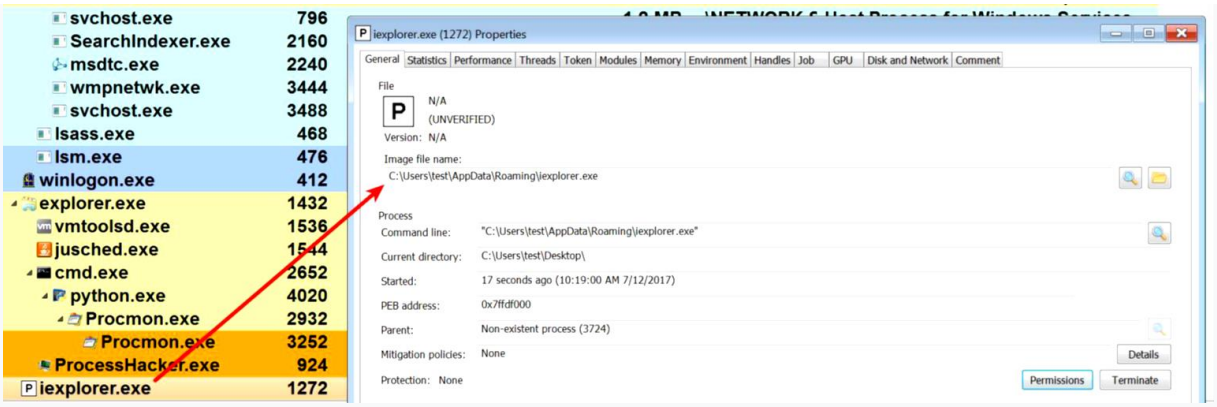



**Figure 32**

- En examinant les journaux Noriben, il est possible de déterminer que le logiciel malveillant a abandonné un fichier appelé iexplorer.exe dans le répertoire % AppData%. Le nom du fichier (iexplorer.exe) est similaire au nom de fichier d'Internet Explorer (iexplore.exe) navigateur. Cette technique est une tentative délibérée de l'attaquant pour que le binaire malveillant ressemble à un exécutable légitime:
- Après avoir déposé le fichier, le logiciel malveillant a exécuté le fichier déposé. En conséquence, un nouveau Le processus iexplorer.exe a été créé. C'était le processus affiché par le Hacker de processus:
- À partir du trafic réseau capturé par Wireshark, on peut voir que le malware résout le domaine C2 et établit une connexion sur le port 80:

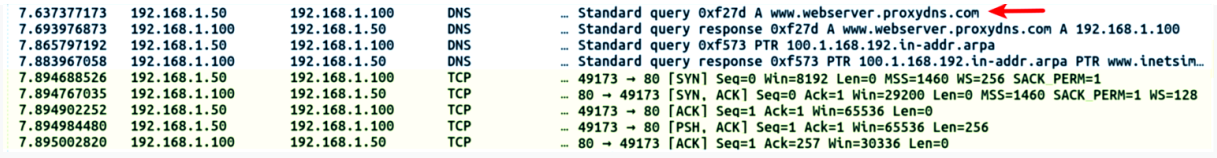



**Figure 33**

Le flux TCP de la communication du port 80, comme illustré dans la capture d'écran suivante, n'est pas trafic HTTP standard; cela suggère que le malware utilise probablement un protocole personnalisé ou communication cryptée. Dans la plupart des cas, le logiciel malveillant utilise un protocole personnalisé ou chiffre. Son trafic réseau pour contourner les signatures réseau. Vous devez effectuer une analyse de code de binaires malveillants pour déterminer la nature du trafic réseau.

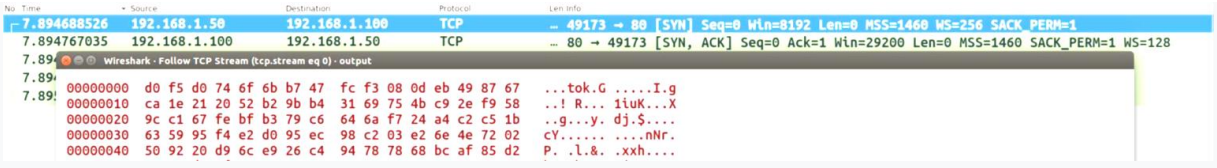



**Figure 34**

Comparaison du hachage cryptographique de l'échantillon déposé (iexplorer.exe) et du binaire d'origine (sales.exe) montre qu'ils sont identiques:

```
$ md5sum sales.exe iexplorer.exe
51d9e2993d203bd43a502a2b1e1193da sales.exe
51d9e2993d203bd43a502a2b1e1193da iexplorer.exe
```

Pour résumer, lorsque le malware est exécuté, il se copie dans le répertoire% AppData% comme iexplorer.exe puis supprime un script batch dont le travail consiste à supprimer le binaire d'origine et lui-même.

Le logiciel malveillant ajoute ensuite une entrée dans la clé de registre afin qu'il puisse démarrer chaque heure à laquelle le système démarre. Le binaire malveillant crypte éventuellement son trafic réseau et communique avec le serveur de commande et de contrôle (C2) sur le port 80 en utilisant un protocole.

En combinant l'analyse statique et dynamique, il a été possible de déterminer les caractéristiques et le comportement du binaire malveillant. Ces techniques d'analyse aidé à identifier le réseau et les indicateurs basés sur l'hôte associés au malware échantillon.

Les équipes de réponse aux incidents utilisent les indicateurs déterminés à partir du malware analyse pour créer le réseau et les signatures basées sur l'hôte pour détecter infections supplémentaires sur le réseau.

# 9 Analyse de la bibliothèque de liens dynamiques (DLL)

Une bibliothèque de liens dynamiques (DLL) est un module qui contient des fonctions (appelées fonctions exportées ou exportations) qui peuvent être utilisées par un autre programme (tel qu'un exécutable ou une DLL). Un exécutable peut utiliser les fonctions implémentées dans une DLL en l'important depuis la DLL.[1].

Le système d'exploitation Windows contient de nombreuses DLL qui exportent diverses fonctions appelées Interfaces de programmation d'applications (API). Les fonctions contenues dans ces DLL sont utilisées par les processus pour interagir avec le système de fichiers, le processus, le registre, le réseau et l'interface utilisateur graphique (GUI).

Pour afficher les fonctions exportées dans l'outil CFF Explorer, chargez le fichier PE qui exporte les fonctions et cliquez sur Export Directory. La capture d'écran suivante montre certaines des fonctions exporté par Kernel32.dll (il s'agit d'une DLL du système d'exploitation et se trouve dans le répertoire C: \ Windows \ System32). L'une des fonctions exportées par Kernel32.dll est CreateFile; cette fonction API est utilisée pour créer ou ouvrir un fichier:

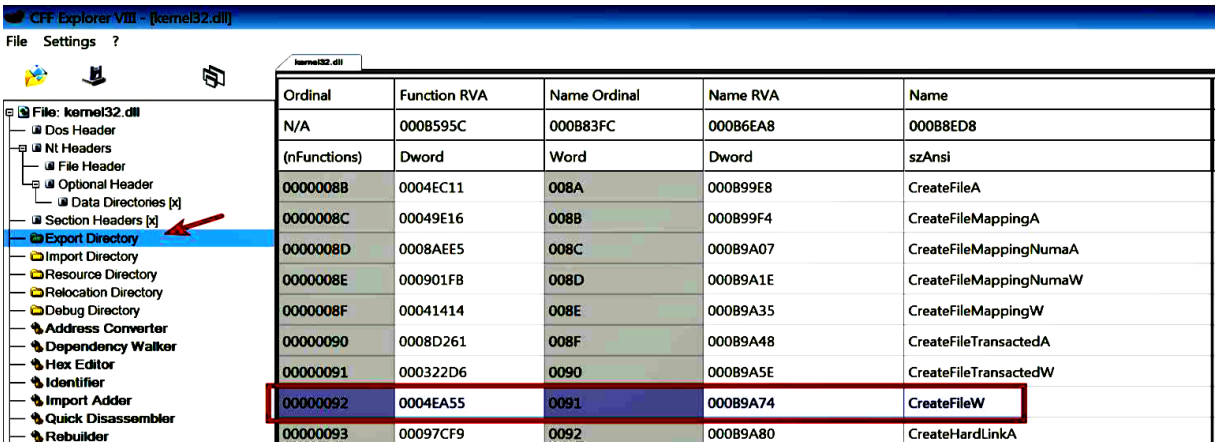



Figure 35

Dans la capture d'écran suivante, on peut voir que notepad.exe importe certaines des fonctions exporté par kernel32.dll, y compris la fonction CreateFile. Lorsque vous ouvrez ou créez un fichier avec Notepad, il appelle l'API CreateFile implémentée dans Kernel32.dll:

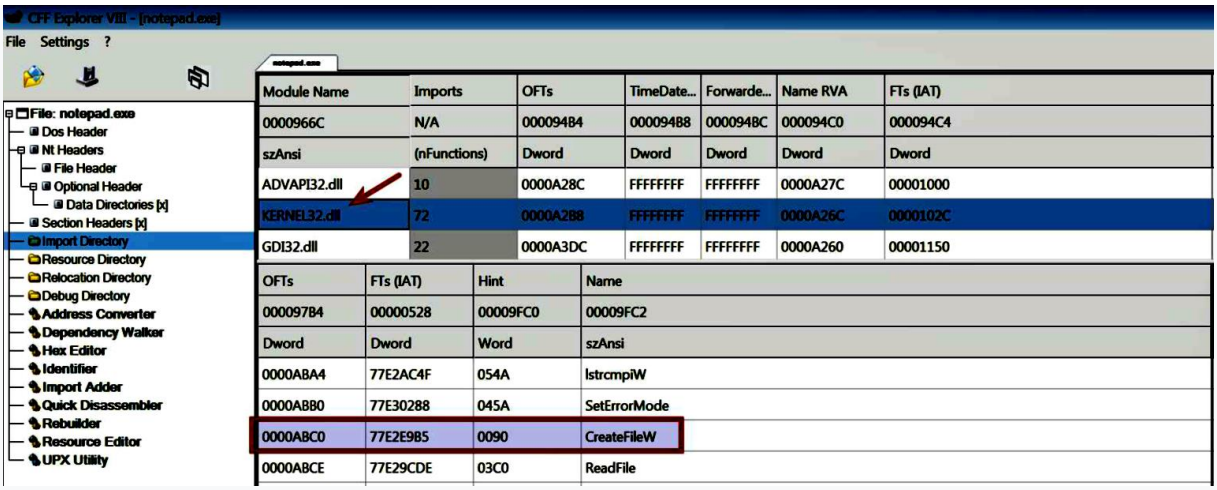



**Figure 36**

Cette fois, le malware a été exécuté et le trafic réseau a été capturé sur le Linux VM avec INetSim en cours d'exécution (simulation de services).

Pour en savoir plus sur les bibliothèques de liens dynamiques, accèder au liens ci-après : https:/ / support. microsoft. com/ en- us/ help/815065/what- is- a- dll
et https: // msdn. microsoft. com/ en- us/bibliothèque / windows / desktop / ms681914 (v = vs. 85) . aspx.

## 9.1 Pourquoi les attaquants utilisent des DLL

Vous verrez souvent des auteurs de logiciels malveillants distribuer leur code malveillant sous forme de DLL au lieu de fichiers exécutables. La liste suivante présente certaines des raisons pour lesquelles les attaquants implémentent leur code malveillant sous forme de DLL:

- Une DLL ne peut pas être exécutée en double-cliquant; DLL a besoin d'un processus hôte pour s'exécuter.
- En distribuant le code malveillant sous forme de DLL, un auteur de malware peut charger son DLL dans n'importe quel processus, y compris un processus

légitime tel que Explorer.exe, winlogon.exe et ainsi de suite. Cette technique donne à l'attaquant la capacité de masquer les actions d'un malware et toutes les activités malveillantes effectuées par le les logiciels malveillants semblent provenir du processus hôte.

- L'injection d'une DLL dans un processus déjà en cours fournit à l'attaquant le capacité à persister sur le système. Lorsqu'une DLL est chargée par un processus dans son espace mémoire, la DLL aura accès à l'ensemble de l'espace mémoire du processus, lui donnant ainsi la possibilité de manipuler la fonctionnalité du processus. Par exemple, un attaquant peut injecter une DLL dans un processus de navigateur et volez les informations d'identification en redirigeant sa fonction API.
- L'analyse d'une DLL n'est pas simple et peut être délicate par rapport à l'analyse de l'exécutable. La plupart des échantillons de logiciels malveillants déposent ou téléchargent une DLL, puis chargent la DLL dans l'espace mémoire d'un autre processus. Après avoir chargé la DLL, le composant dropper / loader supprime lui-même. Par conséquent, lors d'une enquête sur les logiciels malveillants, vous ne trouverez que la DLL.

## 9.2 Analyse de la DLL à l'aide de rundll32.exe

Pour déterminer le comportement du malware et surveiller son activité à l'aide d'une analyse dynamique, il est essentiel pour comprendre comment exécuter la DLL.

Comme mentionné précédemment, une DLL a besoin d'un processus à exécuter. Sous Windows, rundll32.exe peut être utilisé pour lancer une DLL et appeler fonctions exportées depuis la DLL. Voici une syntaxe pour lancer une DLL et appeler une fonction d'exportation utilisant rundll32.exe:

rundll32.exe <chemin complet vers dll>, <fonction d'exportation> <arguments facultatifs>

Les paramètres associés à rundll32.exe sont expliqués comme suit:

-Chemin d'accès complet à la DLL: spécifie le chemin d'accès complet à la DLL, et ce chemin ne peut pas contenir espaces ou caractères spéciaux.

-Fonction d'exportation: il s'agit d'une fonction de la DLL qui sera appelée une fois la DLL chargé.

-Arguments facultatifs: les arguments sont facultatifs et, s'ils sont fournis, les arguments seront passés à la fonction d'exportation lors de son appel.

La virgule: elle est placée entre le chemin complet de la DLL et la fonction d'exportation. La fonction d'exportation est requise pour que la syntaxe soit correcte.

## 9.3 Fonctionnement de rundll32.exe

Il est important de comprendre le fonctionnement de rundll32.exe pour éviter toute erreur. Lorsque vous lancez rundll32.exe à l'aide des arguments de ligne de commande mentionné précédemment, les étapes suivantes sont effectuées par rundll32.exe:

1. Les arguments de ligne de commande passés à rundll32.exe sont d'abord validés; si la la syntaxe est incorrecte, rundll32.exe se termine.

2. Si la syntaxe est correcte, il charge la DLL fournie. À la suite du chargement de la DLL, la fonction de point d'entrée de la DLL est exécutée (qui à son tour appelle la DLLMain une fonction). La plupart des logiciels malveillants implémentent leur code malveillant dans la DLL une fonction.
3. Après avoir chargé la DLL, il obtient l'adresse de la fonction d'exportation et appelle la fonction. Si l'adresse de la fonction ne peut pas être déterminée, alors rundll32.exe s'arrête.
4. Si les arguments facultatifs sont fournis, les arguments facultatifs sont fournis à la fonction d'exportation lors de son appel.

Des informations détaillées sur l'interface rundll32 et son fonctionnement sont expliqué dans cet article: https:/ / support. microsoft. com/ en- in/ help/164787 / info- windows- rundll- et- rundll32- interface.

## 9.4 Lancement de la DLL à l'aide de rundll32.exe

Au cours de l'enquête sur les logiciels malveillants, vous rencontrerez différentes variantes de DLL. Comprendre comment les reconnaître et les analyser est essentiel pour déterminer leur actions malveillantes. Les exemples suivants couvrent différents scénarios impliquant des DLL.

Exemple 1 :Analyse d'une DLL sans exportation

Chaque fois qu'une DLL est chargée, sa fonction de point d'entrée est appelée (qui à son tour appelle son DLLMain). Un attaquant peut implémenter des fonctionnalités malveillantes (comme le keylogging, vol d'informations, etc.) dans la fonction DLLMain sans exporter aucune fonction.

Dans l'exemple suivant, la DLL malveillante (aa.dll) ne contient aucune exportation, qui vous indique que, toutes les fonctionnalités malveillantes peuvent être implémentées dans sa fonction DLLmain, qui sera exécuté (appelé à partir du point d'entrée de la DLL) lorsque la DLL sera chargée.

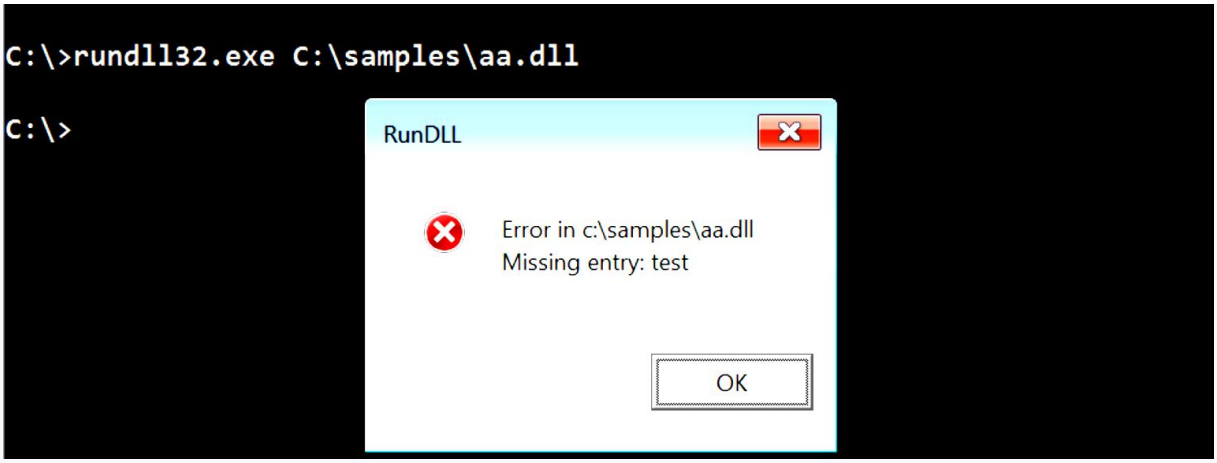
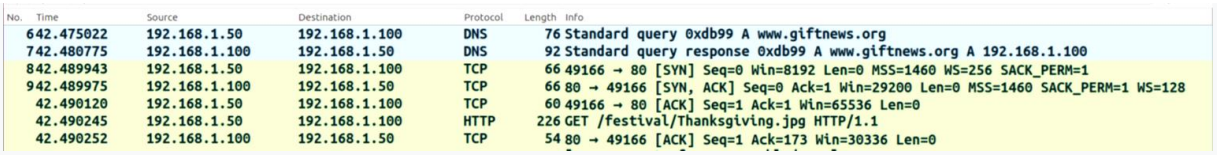



Figure 37

Lors de l'exécution, le malware établit une connexion HTTP avec le domaine C2 et télécharge un fichier (Thanksgiving.jpg), comme indiqué dans la sortie Wireshark suivante:



| No. | Time | Source | Destination | Protocol | Length | Info |
|---|---|---|---|---|---|---|
| | 642.475022 | 192.168.1.50 | 192.168.1.100 | DNS | 76 | Standard query 0xdb99 A www.giftnews.org |
| | 742.480775 | 192.168.1.100 | 192.168.1.50 | DNS | 92 | Standard query response 0xdb99 A www.giftnews.org A 192.168.1.100 |
| | 842.489943 | 192.168.1.50 | 192.168.1.100 | TCP | 66 | 49166 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| | 942.489975 | 192.168.1.100 | 192.168.1.50 | TCP | 66 | 80 → 49166 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| | 42.490120 | 192.168.1.50 | 192.168.1.100 | TCP | 60 | 49166 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 |
| | 42.490245 | 192.168.1.50 | 192.168.1.100 | HTTP | 226 | GET /festival/Thanksgiving.jpg HTTP/1.1 |
| | 42.490252 | 192.168.1.100 | 192.168.1.50 | TCP | 54 | 80 → 49166 [ACK] Seq=1 Ack=173 Win=30336 Len=0 |

**Figure 38**

Exemple 2 Analyse d'une DLL contenant des exportations

Dans cet exemple, nous examinerons une autre DLL malveillante (obe.dll). La capture d'écran montre deux fonctions (DllRegisterServer et DllUnRegisterServer) exportés par la DLL:

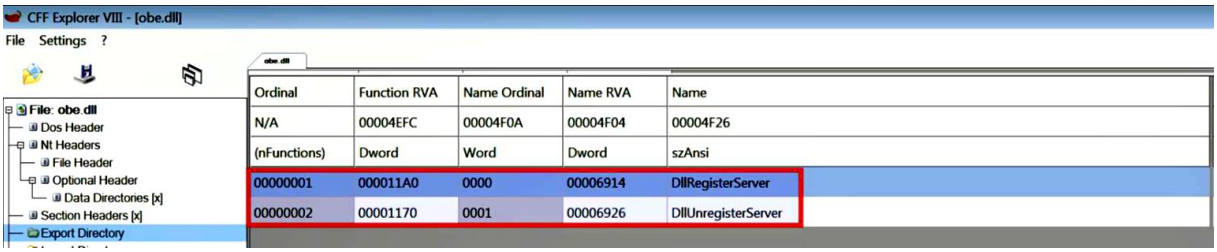



**Figure 39**

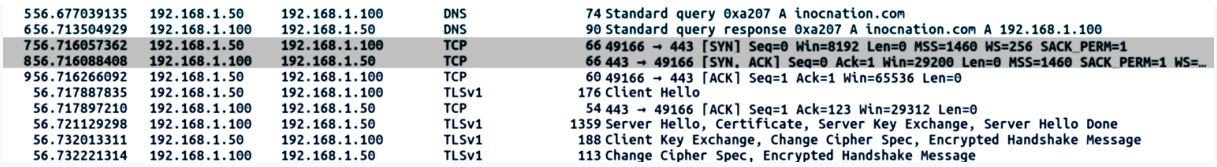
L'exemple DLL a été exécuté avec la commande suivante. Même si obe.dll a été chargé dans la mémoire de rundll32.exe, il n'a déclenché aucun comportement. C'est parce que les DLL la fonction de point d'entrée n'implémente aucune fonctionnalité:

C: \> rundll32.exe c: \ samples \ obe.dll, test

D'autre part, exécuter l'exemple avec la fonction DllRegisterServer comme indiqué ci-dessous, a déclenché une communication HTTPS vers le serveur C2. De là, on peut en déduire que DLLRegisterServer implémente la fonctionnalité réseau:

C: \> rundll32.exe c: \ samples \ obe.dll, DllRegisterServer

La capture d'écran suivante montre le trafic réseau capturé par Wireshark:



| | | | | | | |
|---|---|---|---|---|---|---|
| 556.677039135 | 192.168.1.50 | 192.168.1.100 | DNS | 74 | Standard query 0xa207 A inocnation.com |
| 656.713504929 | 192.168.1.100 | 192.168.1.50 | DNS | 90 | Standard query response 0xa207 A inocnation.com A 192.168.1.100 |
| 756.716057362 | 192.168.1.50 | 192.168.1.100 | TCP | 66 | 49166 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 856.716088408 | 192.168.1.100 | 192.168.1.50 | TCP | 66 | 443 → 49166 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=… |
| 956.716266092 | 192.168.1.50 | 192.168.1.100 | TCP | 60 | 49166 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 |
| 56.717887835 | 192.168.1.50 | 192.168.1.100 | TLSv1 | 176 | Client Hello |
| 56.717897210 | 192.168.1.100 | 192.168.1.50 | TCP | 54 | 443 → 49166 [ACK] Seq=1 Ack=123 Win=29312 Len=0 |
| 56.721129298 | 192.168.1.100 | 192.168.1.50 | TLSv1 | 1359 | Server Hello, Certificate, Server Key Exchange, Server Hello Done |
| 56.732013311 | 192.168.1.50 | 192.168.1.100 | TLSv1 | 188 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 56.732221314 | 192.168.1.100 | 192.168.1.50 | TLSv1 | 113 | Change Cipher Spec, Encrypted Handshake Message |

**Figure 40**

Vous pouvez écrire un script pour déterminer toutes les fonctions exportées (comme indiqué dans le chapitre 2, Analyse statique) dans une DLL et appelez-les en séquence tout en exécuter les outils de surveillance. Cette technique peut aider à comprendre la fonctionnalité de chaque fonction exportée. DLLRunner (https: // github.com/ Neo23x0 / DLLRunner) est un script Python qui exécute tous les fonctions dans une DLL.

Exemple 3 Analyse d'une DLL acceptant les arguments d'exportation

L'exemple suivant montre comment vous pouvez analyser une DLL qui accepte les arguments d'exportation. La DLL utilisée dans cet exemple a été fournie via PowerPoint, comme décrit dans ce lien: https: //securingtomorrow. mcafee. com/ mcafee- labs/ threat- actors- useencrypted- office- binary- format- evade- detection /

La DLL (SearchCache.dll) se compose d'une fonction d'exportation, _flushfile @ 16, dont la fonctionnalité consiste à supprimer un fichier. Cette fonction d'exportation accepte un argument, qui est le fichier à effacer:

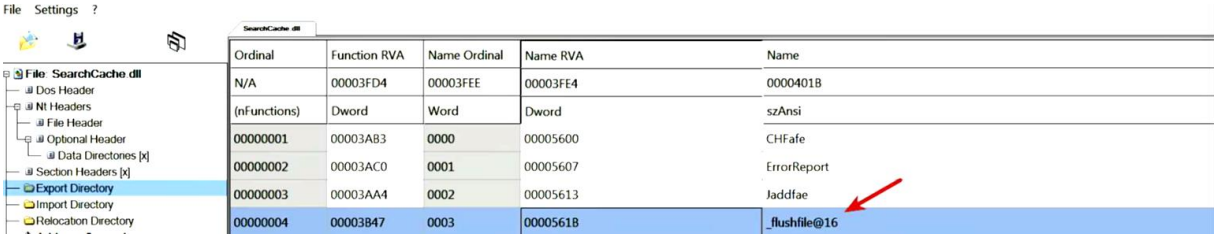



Figure 41

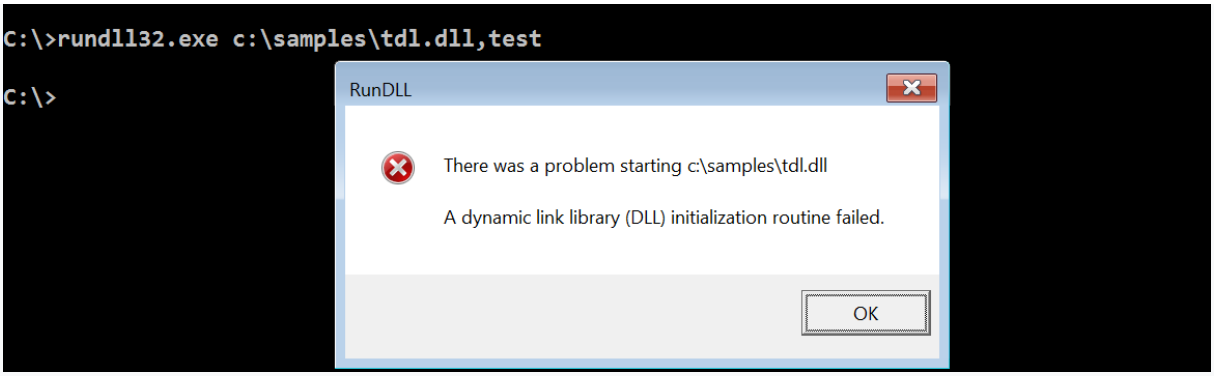
Pour démontrer la fonctionnalité de suppression, un fichier de test (file_to_delete.txt) a été créé et les outils de suivi ont été lancés. Le fichier de test a reçu un argument pour l'exportation function _flushfile @ 16 à l'aide de la commande suivante. Après avoir exécuté ce qui suit: commande, le fichier de test a été supprimé du disque:
rundll32.exe c: \ samples \ SearchCache.dll, _flushfile @ 16
C: \ samples \ file_to_delete.txt

## 9.5 Analyse d'une DLL avec des contrôles de processus

La plupart du temps, le lancement d'une DLL avec rundll32.exe fonctionnera correctement, mais certaines DLL vérifier s'ils s'exécutent sous un processus particulier (tel que explorer.exe ou iexplore.exe) et pourraient changer leur comportement ou se terminer s'ils sont s'exécutant sous tout autre processus (y compris rundll32.exe). Dans de tels cas, vous aurez pour injecter la DLL dans le processus spécifique pour déclencher le comportement.

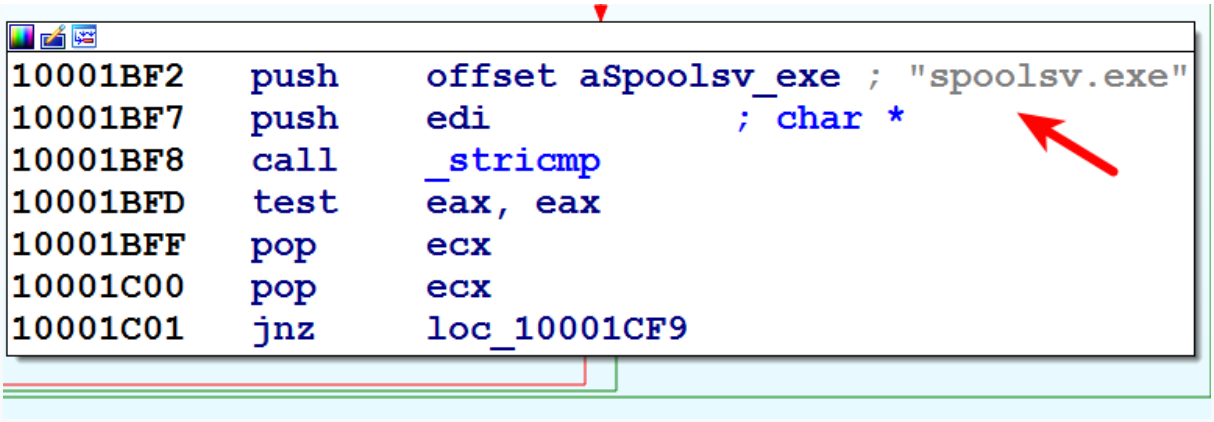
Un outil tel que RemoteDLL (http: / / securityxploded. com/ remotedll. php) vous permet d'injecter une DLL dans tout processus en cours d'exécution sur le système. Il vous permet d'injecter une DLL en utilisant trois méthodes différentes; ceci est utile car si une méthode échoue, vous pouvez en essayer une autre méthode.

La DLL (tdl.dll) utilisée dans l'exemple suivant est un composant de TDSS Rootkit. Cette DLL ne contient aucune exportation; tous les comportements malveillants sont implémentés dans la Fonction de point d'entrée de la DLL. Exécution de la DLL à l'aide de la commande suivante générée une erreur indiquant que la routine d'initialisation de la DLL a échoué, cela indique que la DLL la fonction de point d'entrée n'a pas été exécutée avec succès:



## Figure 42

Pour comprendre la condition qui a déclenché l'erreur, une analyse de code statique (ingénierie) a été réalisée. Après avoir analysé le code, il a été constaté que la DLL, dans sa fonction de point d'entrée, a effectué une vérification pour déterminer si elle s'exécute sous spoolsv.exe (le service de spouleur d'impression). S'il s'exécute sous un autre processus, l'initialisation de la DLL échoue:

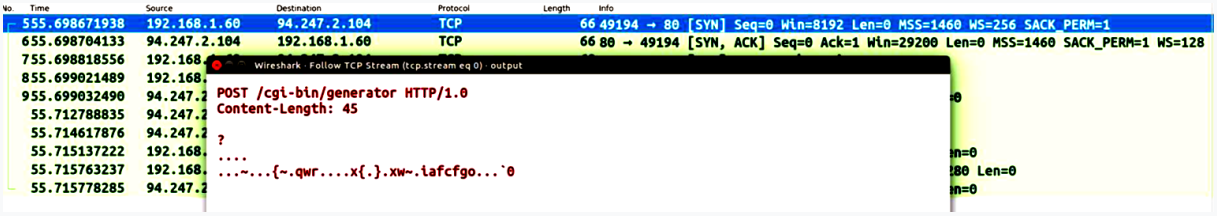



## Figure 43

Pour déclencher le comportement, une DLL malveillante a dû être injectée dans le processus spoolsv.exe à l'aide de l'outil RemoteDLL. Après avoir injecté la DLL dans spoolsv.exe, les éléments suivants:

- Les activités ont été capturées par les outils de suivi. Le malware a créé un dossier (recyclé) et un fichier autorun.inf sur le lecteur C: \. Il a ensuite déposé un fichier boot.com dans le nouveau dossier créé C: \ resycled:
- Le malware a ajouté les entrées de registre suivantes; à partir des entrées ajoutées, vous pouvez dire que le malware stocke des données chiffrées ou de configuration dans le registre:

La capture d'écran suivante montre la communication C2 du malware sur le port 80:




## Figure 44

Lors de l'enquête sur les logiciels malveillants, vous pouvez rencontrer une DLL qui s'exécutera uniquement lorsqu'il est chargé en tant que service. Ce type de DLL s'appelle un service DLL.

## Résumé

L'analyse dynamique est une excellente technique pour comprendre le comportement des logiciels malveillants et déterminer son réseau et ces indicateurs basés sur l'hôte. Utiliser l'analyse dynamique pour valider résultats obtenus lors de l'analyse statique.

Combiner analyse statique et analyse dynamique, nous aide à mieux comprendre le binaire du malware. L'analyse dynamique de base a ces limites, et pour mieux comprendre le fonctionnement du binaire malveillant, on devra effectuer une analyse de code (rétro-ingénierie).

# Chapitre VI : Développement de virus macro comme preuve de concept des logiciels malveillants

## 1 Introduction

Ce chapitre présent la description de notre application développée, cette description est divisée en deux parties : La conception et l'implémentation, la première présente quelques aspects relatifs au concept de base et la conception de l'application, tandis que la deuxième traite la phase d'implémentation ainsi que les outils de développement.

## 2 La conception

Notre objectif est de mettre en œuvre un virus macro qui se propage via e-mail. Pour mieux décrire notre application on va présenter la notion de preuve de concept ainsi que la notion de virus macro, les raisons pour lesquelles les auteurs de virus aiment les virus de macro, méthodes de leurs propagation, les actions des virus macros, technique de développement des virus macros et moyens de détection desdits virus.

### 2.1 Preuve de concept

Une preuve de concept (de l'anglais : proof of concept P.O.C) ou démonstration de faisabilité, est une réalisation ayant pour vocation de montrer la faisabilité d'un procède ou d'une innovation.[3]

La preuve de concept peut être considérée comme une étape importante sur la voie d'une véritable prototype.

En sécurité informatique, une preuve de concept est la démonstration de l'existence d'une faille logicielle par un programme qui le met en évidence.

Dans notre cas, la preuve de concept se résume au développement d'un virus macros pour les fichiers excel. afin de montrer l'existence des logiciels malveillants et que l'analyse desdites logiciels est une exigence dans le domaine de sécurité

### 2.2 Qu'est-ce qu'un virus de macro?

En termes informatiques, un virus macro est un virus écrit dans un langage macro qui est un langage de programmation intégré à une application logicielle. Certaines applications telles que Microsoft Office, Excel et PowerPoint permettent d'intégrer des macros dans des documents afin que les macros soient automatiquement exécutées lorsque le document est ouvert, ce qui fournit un mécanisme distinct par lequel des instructions informatiques nuisibles peuvent être publiées. [3]

### 2.3 Pourquoi les auteurs de virus aiment les virus de macro [4]

- Facile à écrire.
- Tout le monde échange des documents et des données et, ce faisant, les macro-virus peuvent infecter plus de personnes que leurs homologues plus complexes.
- Peut être multiplateforme et multiculturelle, infectant tout ordinateur capable d'exécuter Office

- Internet Explorer peut télécharger automatiquement des documents Office à partir du Web ou à partir d'e-mails sans demander à l'utilisateur de confirmer le téléchargement.

## 2.4 Comment les virus de macro se propagent

À quelques exceptions près, les virus de macro se propagent lorsqu'un utilisateur ouvre ou ferme un document infecté. Les documents sont répartis entre les utilisateurs des manières suivantes: courrier électronique, disquette, Internet et support externe tel que falsh disk, CD, DVD, Carte mémoire [4]…ect.

## 2.5 Ce qu'un virus de macro peut faire

Un auteur de virus de macro peut programmer sa création pour faire presque tout ce qui est possible avec un PC. Il peut corrompre des données, créer de nouveaux fichiers, déplacer du texte, faire clignoter des couleurs, insérer des images, envoyer des fichiers sur Internet et formater des disques durs. Non seulement limités aux commandes déjà puissantes du langage macro, les virus macro sont de plus en plus utilisés comme mécanismes de transport pour éliminer des bogues encore plus méchants. Les virus de macro peuvent utiliser la commande VBA *SHELL* ou utiliser l'API du noyau du système d'exploitation pour exécuter toute commande externe de leur choix. La commande VBA *KILL* peut être utilisée pour supprimer des fichiers. Les virus de macro modifient les registres, utilisent le courrier électronique pour transmettre des copies d'eux-mêmes à d'autres personnes, recherchent des mots de passe, copient des documents et infectent d'autres programmes. Les virus de macro peuvent causer de nombreux dommages de différentes manières.[4]

## 2.6 Techniques générales de macro-virus

Les progrès de la technologie antivirus et les changements de sécurité de Microsoft ont obligé les auteurs de macros virus à apprendre de nouvelles astuces[4].

### 2.6.1 Virus de messagerie

Malheureusement, en utilisant VBA, il est trop facile pour un virus de s'envoyer à d'autres victimes par courrier électronique. VBA permet à un rédacteur de virus d'interroger le système pour obtenir toutes les informations nécessaires (nom de l'application de messagerie, nom d'utilisateur et mot de passe de messagerie) et d'envoyer une pièce jointe par courrier électronique. MAPI , ou Messaging Application Programming Interface , est la norme de facto pour les programmes de messagerie Windows. Il peut être utilisé par de nombreux langages informatiques pour envoyer des e-mails depuis le poste de travail d'un utilisateur vers un autre utilisateur. C'est cette technique que nous avons utilisé pour développer notre malware.

### 2.6.2 Virus de macro furtifs

À mesure que les virus de macro sont devenus plus populaires, Microsoft a développé différentes méthodes de notification qui devraient alerter l'utilisateur que quelque chose ne va pas. Malheureusement, toutes ces notifications sont faciles à désactiver pour les virus de macro, et même lorsqu'elles ne le sont pas, la plupart des utilisateurs finaux ne comprennent pas ce que les avertissements tentent de communiquer. Avec Office 97 et 2000, les avertissements de virus de macro sont écrits un peu plus clairement.Les virus de macro ont une poignée de façons de se cacher de l'inspection par défaut de l'utilisateur final, bien que la plupart des routines furtives ne se déroulent qu'après que l'utilisateur a ignoré les avertissements d'origine et accepté le virus en premier. Un virus de macro ne peut pas

désactiver les invites d'avertissement et les paramètres prédéfinis lors de sa première activation. Le paramètre le plus courant vous avertit simplement de tout document contenant une macro, que la macro soit ou non malveillante.

> Malheureusement, les documents ne contenant aucune macros peuvent faire apparaître l'avertissement de macro. Les documents avec des raccourcis clavier, des redéfinitions de menus ou de boutons, ou même des documents qui contenaient des macros mais qui n'en contiennent pas actuellement, peuvent déclencher l'avertissement de macro.

Les virus peuvent modifier les paramètres du registre pour empêcher Office de notifier l'utilisateur de toute macros. D'autres paramètres de sécurité peuvent être désactivés dans VBA en écrivant la commande de macro appropriée dans un modèle infecté. Une autre technique furtive courante consiste à désactiver l' option de menu Outils -> Macro afin que les macros en cours d'exécution ne puissent pas être inspectées. Les virus de macro peuvent désactiver l'invite afin qu'Office convertisse le document sans demander de réponse à l'utilisateur. Même si un virus désactive l'invite de conversion, si l'utilisateur final recherche, Office affiche généralement le nom de la macro en cours de conversion dans la barre d'état pendant le processus de conversion. La plupart des utilisateurs ne le remarquent pas.VBA permet aux virus de macro de se «verrouiller» et ne peut être visualisé que si l'utilisateur connaît le mot de passe correct. Cependant, si le projet VBA est protégé par mot de passe, aucun module ne peut être copié pour lui.

## 2.7 Virus de macro cryptés et polymorphes

Comme leurs homologues exécutables, de nombreux virus de macro changent d'apparence pour éviter la détection par analyse. Des routines de chiffrement aléatoire sont utilisées pour masquer le code du virus, mais les routines de chiffrement ont tendance à être plus faibles que leurs homologues de virus exécutables. Certains virus renomment de manière aléatoire les noms de macro et les variables de mémoire. D'autres créent leurs macros à la volée. Ils le font en stockant la plupart des macros sous forme de texte brut dans le document et en appelant un générateur de macros intégré. Le constructeur de macros crée ensuite les macros et les exécute.

## 2.8 Plus de manipulation externe avec VBA

VBA contient de nombreuses fonctionnalités permettant aux virus de macro d'interagir avec le PC en dehors du champ d'application de l'application. Voici quelques exemples:

- ·La commande VBA *KILL* permet de supprimer n'importe quel fichier sur le disque dur local. Il prend en charge l'utilisation de symboles génériques (* ou?),
- Les virus de macro peuvent supprimer des sous-répertoires avec la commande *RMDIR* .
- La commande *SHELL* est la commande la plus puissante et permet d'exécuter n'importe quelle commande externe.
- Mieux encore, pour les auteurs de code malveillant, il a un paramètre, *vbHide* , qui permet à la commande externe d'être exécutée dans une fenêtre cachée.

Ces quatre exemples de commandes peuvent rendre n'importe quel PC vulnérable à de nombreux types d'attaques.

## 2.9 Détection des virus de macro

### 2.9.1 Avertissements de macro

La plupart des versions plus récentes d'Office (97 et versions ultérieures) vous avertiront si un document, un classeur ou un fichier de données contient des macros avec le message suivant:

C: \ <chemin> \ <filename> contient des macros. Les macros peuvent contenir des virus. Il est toujours prudent de désactiver les macros, mais si les macros sont légitimes, vous risquez de perdre certaines fonctionnalités.

### 2.9.2 Suppression des virus de macro et réparation des dommages

- Essayez un antivirus
- Obtenez une application propre
- Contourner les automacros : Maintenir la touche Maj enfoncée lors de l'ouverture de Word ou Excel, ou lors de l'ouverture d'un document, classeur ou modèle désactivera automatiquement toutes les automacros présentes. La touche Shift peut être maintenue enfoncée tout en quittant pour désactiver les macros de fermeture automatique .
- Inspecter les données et supprimer les macros malveillantes: Ouvrez votre document de macro suspecté, en veillant à désactiver les macros. Vous disposez de trois outils de macro au sein d'Office: éditeur de macros, organisateur et Visual Basic Editor. On utilise généralement les trois pour s'assurer que tout est nettoyé.
- Restaurer à partir d'une sauvegarde
- Prévention des virus de macro : Les virus de macro sont le premier type de code mobile malveillant. Voici quelques recommandations pour les empêcher d'attaquer votre environnement.
    - Désactiver les macros dans les documents
    - Définir la sécurité du bureau sur Élevée

# 3 Implémentation du virus macros excel

## 3.1 Les étapes d'implementation

Les étapes d'implémentation de développement de notre virus se résumes dans les points suivant :

- Création d'un fichier excel avec une macro qui se déclenche automatiquement à l'ouverture dudit fichier. Ladite macro affiche un message qui invite l'utilisateur à saisir son e-mail en cliquant sur le bouton pour gagner un cheque de 1000.000 $
- Une fois l'utilisateur saisie l'adresse e-mail, le logicielle de messagerie est activer automatiquement (dans notre cas c'est outlook.com ) avec génération

d'un fichier excel comme piece jointe et le destinateur du message est l'adresse e-mail de l'utilisateur avec possibilite de saisir d'autres adresse par l'utilisateur.

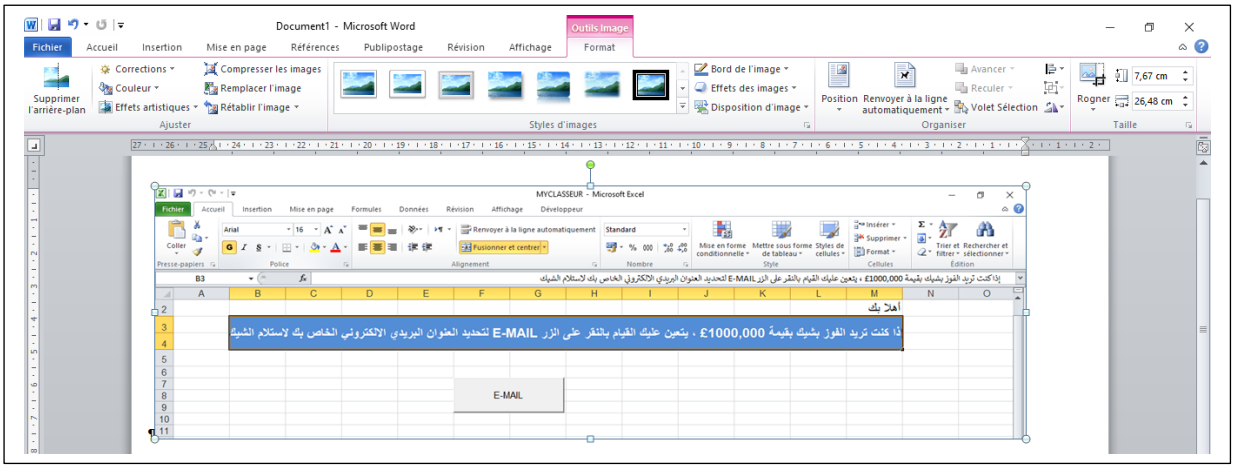
- Une fois le message est envoyer, le recepteur de message à l'ouverture du fichier excel envoyer comme pièce jointe, le virus peut effectuer n'importe quelle action de la victime qui a ouvert la pièce jointe. Pour notre cas, notre pièce jointe, une fois ouverte elle créera un dossier "projet taha1" et déplacera tout les fichiers documents dans le present dossier.

## 3.2 Les outils de développement

Les outils de développement de notre virus sont :

- Système d'exploitation windows (windows 10)
- L'application Excel 2010
- Le langage VBA dédie à la suite Office
- Internet
- Logiciel de messagerie (pour notre cas c'est outlook.com)

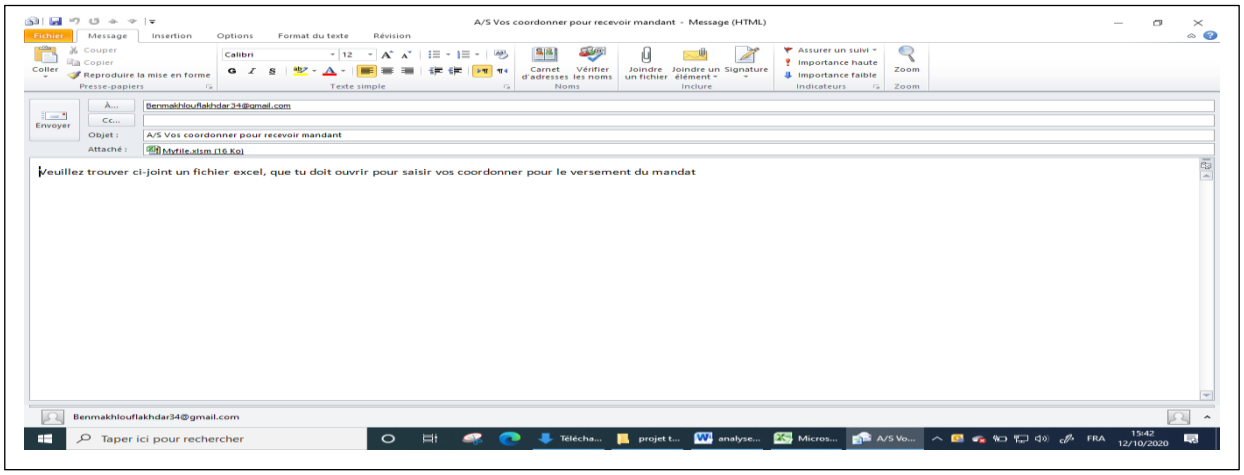
## 3.3 Aperçus de l'application virus maros



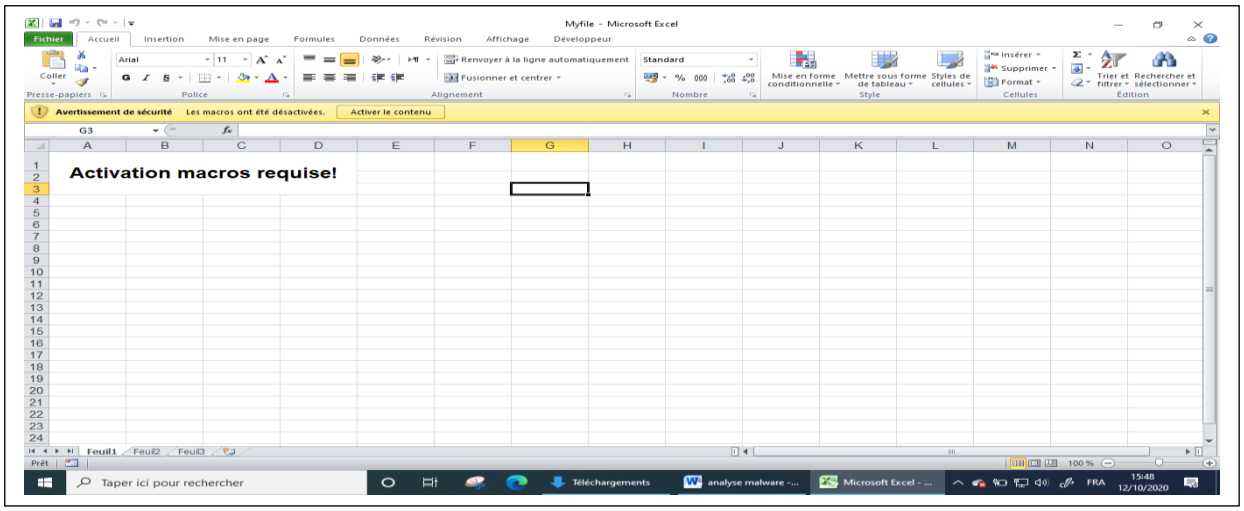

- **Ouvrir le fichier excel dont le nom est "MyClasseur",**


- **Cliquer sue le bouton "E-Mail" , un masque de saisie est affiché**

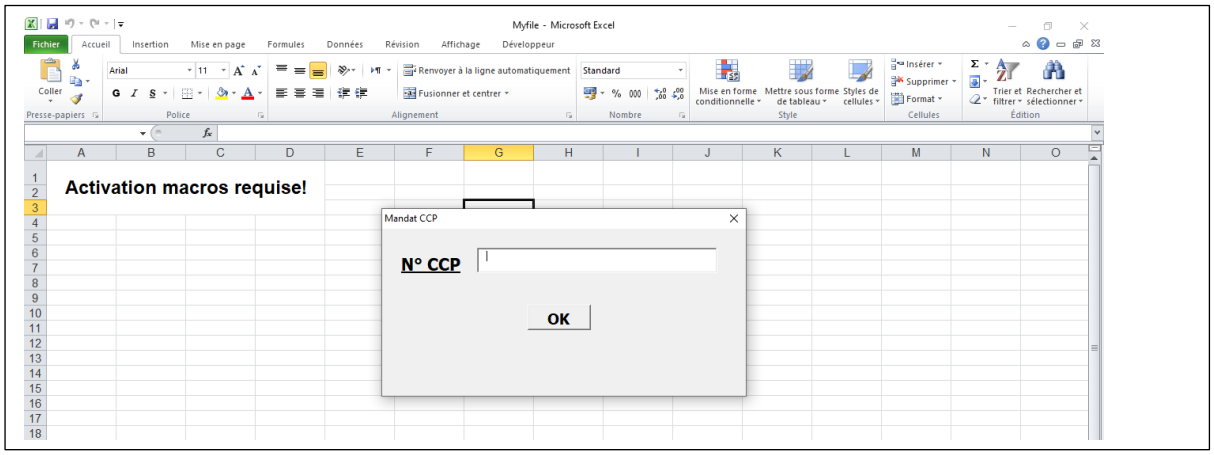
Remarque : Il possible de générer une routine pour vérifier la validité de l'adresse e-mail mais ce n'est notre but




- En cliquant sur le bouton ok, la boite de messagerie s'affiche automatiquement
- Cliquer sur le bouton Envoyer de l'application messagerie pour envoyer le fichier excel générer automatiquement comme pièce jointe.

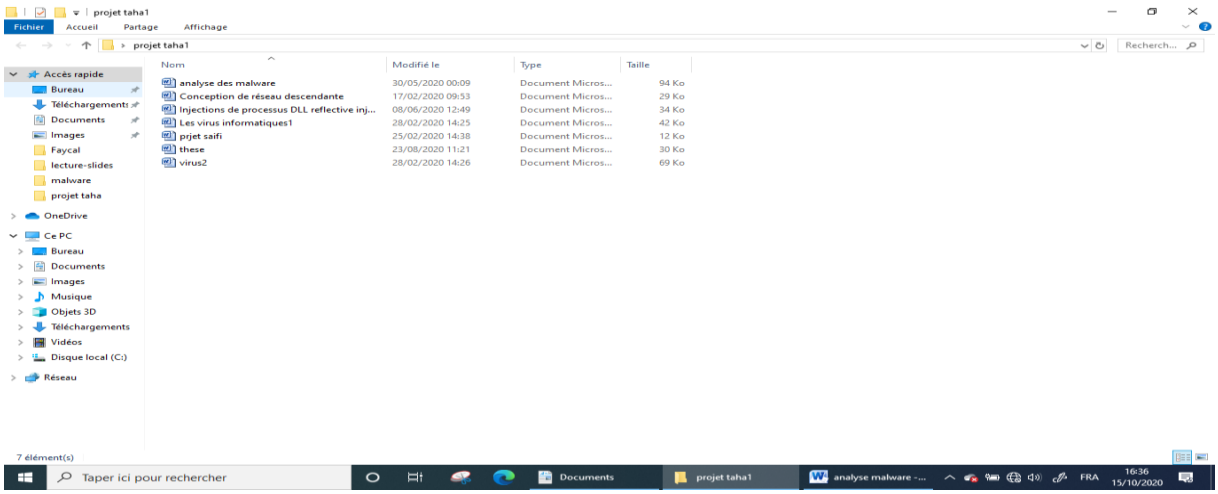



- Le fichier excel générer automatiquement porte le nom Myfile, le destinataire l'ouvre un message d'avertissement est affiché pour demander l'activation des macros




- Une fois la macro activé, un masque de saisie est affiché pour demander à l'utilisateur la saisie le N° CCP
- Une fois l'utilisateur clique sur le bouton OK, la macros représentant les actions malveillantes est déclenchés automatiquement .

**Remarque1 :**Pour notre cas et à titre d'illustration notre macro creera un nouveau dossier sur le bureau et déplacera tous les fichiers documents du dossier documents du windows vers le nouveau dossier créer




**Remarque2**:Notez bien, on peut activer n'importe quelle actions malveillantes comme kill qui détruit les fichiers sans récupération ou crypter des fichiers …ect.

# Conclusion générale

Le travail présenté dans ce mémoire tourne autour de développement d'un virus macro comme preuve de concept des logiciels malveillants. L'objectif était de générer une macro représentant des actions malveillantes. Ladite macro est enregistré dans un fichier excel qui fera objet d'une pièce jointe envoyer via email au destinataire(s) choisis par l'expéditeur.

Finalement, on a réussi à implémenter cette application d'une manière simple et compréhensible, pour montrer le concept des logiciels malveillants. Le but de notre travail est d'inviter les étudiants spécialiste dans la sécurité informatique a s'intéresse a la notion d'analyse des logiciels malveillants pour les raisons suivantes :

- L'analyse des logiciels malveillants est la source de développement des différents moyens de protection des systèmes (antivirus, methode de protection, ..ect
- Pour sécuriser le système et développer une stratigie de protection, ils faut être un expaire dans l'analyse

Ce modeste travail constitue une bonne expérience pour nous, et represente un bon complément pour notre formation de base, Il nous a permis d'enrichir nos connaissances théoriques et pratiques, et constitue la base de départ pour des futurs travaux dans le domaine de sécurité.

Nos perspectives étant de continuer dans le domaine de la sécurité , de bâtir une base solide pour pouvoir développer les méthodes d'analyses et des applications plus consistantes, et plus complètes dans le domaine de sécurite.

Enfin, on invite les futures étudiants de développé les autres methodes d'analyse telque le reverse engeniring qui leurs permettres de mieux sécurirer n'importe qu'elle système informatique.

# *Bibliographie et webographie*


[1]: Livre intitulé **"Learning Malware Analysis"** Copyright © 2018 Packt Publishing by MONNAPA K A

[2]:Livre intitule **Practical Malware Analysis**" Copyright © 2012 by Michael Sikorski and Andrew Honig.

[3] Wikpedia

[4] Article pdf " **Des macros Excel pour exporter et importer des modules de code VBA**" par *Jean-Baptiste Duclos*[1]

[5] :https://searchsecurity.techtarget.co/definition/macro-virus

**[6]** :https://blog.malwarebytes.com/glossary

**[7]** Analyse hybride: https://www.hybrid-analysis.com/

**[8].KernelMode.info:**

**http://www.kernelmode.info/forum/viewforum.php?f=16**

**[9] . VirusBay: https://beta.virusbay.io/**

**[10] Contagio décharge des logiciels malveillants: http://contagiodump.blogspot.com/**

**[11]. AVCaesar: https://avcaesar.malware.lu/**

**[12] . Malwr: https://malwr.com/**

**[13].VirusShare: https://virusshare.com/**

**[14]. le zoo: http://thezoo.morirt.com/**

[15] http://www.filesignatures.net/.

[16] https://mh-nexus.de/en/hxd/:

## *LISTE DES FIGURES*

## Résumé

Notre travail consiste à développer un logiciel malware comme preuve de concept des malware en démontrant que la meilleure solution pour sécuriser le système informatique est l'analyse. Cette analyse qui est considéré comme une compétence se base sur l'analyse statique et l'analyse dynamique. Chaque analyse a ces propres outils. Ces derniers sont présentés dans le but de donner une idée sur l'information récolte sur le malware. Ces informations permettent d''identifier le malware et d'adopter des méthodes de sécurisation de notre système contre ce type de malware.

**Abstract**

Our job is to develop malware software as a proof of concept for malware by demonstrating that the best solution to secure the computer system is analysis. This analysis, which is considered a skill, is based on static analysis and dynamic analysis. Each analysis has its own tools. These are presented in order to give an idea of the information gathered on the malware. This information is used to identify the malware and adopt methods of securing our system against this type of malware.

ملخص:

مهمتنا هي تطوير برنامج خبيث كدليل على مفهوم البرمجيات الخبيثة من خلال إثبات أن أفضل حل لتأمين نظام الكمبيوتر هو التحليل. هذا التحليل ، الذي يعتبر مهارة ، يعتمد على التحليل الثابت والتحليل الديناميكي. كل تحليل له أدواته الخاصة. يتم تقديم هذه من أجل إعطاء فكرة عن المعلومات التي تم جمعها حول البرامج الضارة. تُستخدم هذه المعلومات لتحديد البرامج الضارة واعتماد طرق لتأمين نظامنا ضد هذا النوع من البرامج الضارة.