Rapport de Projet
SYSTÈME
-SPYWARE-

Présenté par:

Guessoum Mohamed El fateh	171731073624
Djabri abdelkader chaker	171739003232
ZIANI Mohamed Riad	171731074153
Harouni djihane oumkeltoum	171731091986

1) Introduction:

Un malware ou logiciel malveillant est un terme générique qui décrit tous les codes et programmes qui peuvent menacer d'une manière ou une autre la sécurité ou le bon fonctionnement d'un système.

On peut distinguer plusieurs types de malwares selon la méthode de propagation, de déclenchement et l'impact sur la cible. Parmi les malwares les plus connus, nous citons :

Le cheval de Troie: (Trojan Horse) est un type de malware qui se présente sous forme d'un logiciel légitime et anodin, mais qui contient en même temps des fonctionnalités malveillantes; Son but est de les introduire dans le système et de les installer à l'insu de l'utilisateur.

Un ransomware: est un logiciel malveillant qui prend en otage les fichiers de la cible, et ce en chiffrant toutes les données sensibles. Afin de récupérer ces données, une somme d'argent est demandée à leur propriétaire en échange de la clé, qui permettra de les déchiffrer. Le ransomware peut aller jusqu'à bloquer l'accès de tous les utilisateurs d'une machine.

Un spyware: est un logiciel malveillant qui est injecté sur l'ordinateur ou l'appareil mobile de la cible. Ce type de logiciel permet d'espionner les personnes en allumant à titre d'exemple leurs caméras ou microphones et récupérer les enregistrements par la suite.

2) Problématique

Depuis le début de l’année 2020, les attaques basées sur les malwares ont vu une augmentation exponentielle, et le nombre de cas impactés avec plus ou moins de réussite pour les attaquants devient très inquiétant.

Cette menace évolue rapidement et met les grandes et petites entreprises face à un risque énorme. De ce fait, et dans le but d’améliorer le système des Antivirus, les professionnels optent pour une approche parfois offensive pour trouver les vulnérabilités qui sont potentiellement exploitables pour injecter les différents malwares et bypasser le système de détection développé par les Anti-virus; et les corriger ensuite.

Pour ce projet nous allons développer un Spyware qui prend le contrôle du microphone à l'insu de l'utilisateur, et le programme doit être exécuté discrètement en arrière-plan. De plus, ce dernier doit se relancer automatiquement avec le démarrage du système.

Afin de réaliser ce travail, nous avons suivi les étapes suivantes :

- Coder un programme en python qui prend le contrôle du microphone.
- S’assurer que l'exécution se fait en arrière-plan.
- Exécuter le programme au démarrage du système.
- Le cacher du gestionnaire des tâches.

3) Présentation de l'environnement de travail :

Windows 10 est un système d'exploitation(OS) développé par Microsoft, Il s'agit de la dernière version améliorée de la famille des systèmes Windows NT « New Technology ». Nous avons choisi de cibler Windows 10 par notre Spyware pour deux raisons essentielles: le nombre gigantesque d'utilisateurs et même des entreprises qui utilisent Windows 10 comme système d'exploitation pour leurs travaux quotidiens. De plus, les anciennes versions de Windows et qui partagent le même noyau et la même base d'architecture avec cet OS ont fait face à plusieurs types de malware.

4) Les Outils et les langages de programmation utilisés:

Dans ce projet nous avons utilisé deux langages de programmation, chacun pour un but précis, ainsi qu'un des services de stockage offert par Firebase.

Python: est un langage de programmation open source multi-paradigmes, car non seulement il offre la possibilité de coder d'une manière impérative classique, mais aussi de programmer de façon fonctionnelle tandis qu'en orientée-objet. Nous avons utilisé python pour la diversité des bibliothèques disponibles, et afin de développer le spyware qui prend le contrôle du microphone, ensuite charge les enregistrements sur Firebase. Nous avons également utilisé un jeu (Snake) qui est codé aussi sur python.

Firebase: Firebase est une plateforme Cloud (PaaS) fournie par Google et dédiée au développement des applications mobiles, web et de bureau. Elle met à la disposition des développeurs plusieurs outils et interfaces de programmation

(API), facilitant le développement et l'hébergement des applications. Parmi les services offerts par Firebase, nous citons : « RealtimeDatabase »,« FirebaseHosting » et « Cloud Storage ». Afin de récupérer l'audio enregistré par notre spyware, nous avons opté pour le service de stockage d'objets « Cloud Storage », et ce pour but de décentraliser l'accès à ces enregistrements, et afin de pouvoir y accéder en anonyme.

VBScript: (diminutif de Microsoft Visual Basic Scripting Edition) est un langage interprété utilisé en tant que langage de script d'usage général. Comme tout langage de script. Il ne nécessite pas de compilation avant d'être exécuté, et l'interpréteur est disponible sur toutes les machines Windows. VBScript nous a était nécessaire pour déplacer d'une manière automatique le spyware et le mettre dans le fichier démarrage du système windows de la cible.

5) Méthode d'implémentation:

Pour le processus d'installation, et le fonctionnement de notre programme nous avons mis à votre disposition une vidéo qui les explique tout en montrant les résultats.

6) Développement de la solution:

6.1) Programme de prise de contrôle du microphone:

Tout d'abord la première étape du script et de tester la connexion internet, pour éviter toute sorte d'erreur qui puisse déclencher une exception, et maintenir l'exécution discrète du script. Dans le cas où la cible est connectée à l'internet, le script fait appel à la fonction *"get_audio()"* qui permet d'allumer le microphone et d'enregistrer l'audio pour une durée de notre choix. Le prototype de la fonction est comme suit:

```
# Function that opens microphone and start recording
def get_audio():
```

6.2) Charger les enregistrements sur Firebase:

Pour charger les enregistrements sur firebase nous avons utilisé le module “pyrbase” de python. Après avoir enregistré l’audio à l’aide de la fonction “*get_audio()*” le script fait appel à une autre fonction *“Upload_ToCloud(FileName)”* qui prend le nom du fichier audio en argument ensuite le charge sur le répertoir “Recording/FileName” de firebase. Chaque fichier audio est identifié par la date de début de son enregistrement; le nom est sous le format suivant "*Voice_Record"+Current_DateTime+".wav*" où “*Current_DateTime*” est une variable contenant la date et l’heure actuelle. Le prototype de la fonction est comme suit:

```
# Function that  upload the file that exist in local path to firebase's storage service
def Upload_ToCloud(FileName):
```

6.3) Exécution en arrière-plan:

Nous nous sommes assurés pendant le développement du script que le fonctionnement du programme se fait à l'arrière-plan, et à l’aide de générateur des exécutables “pyinstaller” nous avons caché la fenêtre du ligne de commande qui s’affiche pendant l'exécution.

6.4) Exécution au démarrage du système :

Sur les systèmes windows il y a plusieurs façons de lancer un programme avec le démarrage de l’ordinateur, entre autres l’utilisation des services, ou carrément mettre le raccourci du

programme dans le répertoire “startup”. C’est cette dernière méthode que nous avons utilisé. Afin d’automatiser le processus, nous avons coder deux scripts vbs qui se chargent de cette opération (mettre le spyware dans “programs” et le raccourci dans le répertoire “startup”), et il suffit de lancer le jeu une fois seulement pour executer ces scripts, et ainsi mettre le malware et son raccourci en place. Les scripts VBS sont en pièces jointes.

6.5) cacher le programme du gestionnaire des tâches:

Vu que la méthode pour cacher réellement un processus du gestionnaire des tâches est d’une complexité importante et nécessite des techniques un peu avancées, nous avons opté pour une approche différente basée sur l'ingénierie sociale. Cette méthode consiste à changer les noms des fichiers malveillants avec des noms des programmes légitimes et qui se trouvent de manière continue sur le gestionnaire des tâches, et ces fichiers seront dans le même répertoire que les autres fichiers de dépendance du jeu.

7) Conclusion :

Au cours de la réalisation de ce projet nous avons appris de nouvelles notions et des concepts, entre autres les méthodes d'injection, d'exécution en arrière-plan ainsi que la méthode de lancement automatique d’un programme lors du démarrage d’un système.

Au début, nous avons essayé une méthode qui s’appelle RunPE injection [1] pour cacher l'exécutable du malware dans un fichier légitime tout en gardant la même signature mais en changeant son contenu en mémoire. Nous avons ensuite découvert un souci de compatibilité entre le programme exécutable du spyware et le

programme qui implémente le RunPE développé avec C#. Nous avons ensuite pensé à une autre façon qui est basée sur l’exploitation des erreurs humaines pour infecter le système de notre cible.

Pour conclure les systèmes de détection des malwares offert par les différents antivirus peut sembler sûr et efficace en utilisant de nouvelles technologies de détection basés Intelligence Artificielle, avec un taux de faux négatif très réduit; mais cela pourrait être irrémédiablement anéantis si un utilisateur dans une entreprise ou chez lui ne traite pas tous les fichiers et liens qu’il reçoit avec prudence, il risque de lancer des programmes malveillants involontairement, et d'être la cause d’une cyberattaque.

[1] : **RunPE**: ou bien Process hollowing ,Cette technique consiste à lancer un processus système en pause, puis de remplacer son contenu mémoire avec l'image mémoire de l'exécutable (spyware) et enfin de relancer le processus système. Cela permet d'exécuter un exécutable sans avoir besoin à le déposer sur le disque.