

GOURAUD Elie

KOWAL Jérémy

DUT RT Chalons en Champagne

Tuteur du projet : Melcior Michel

Année universitaire 2020-2021

Rapport de projet : sécurité offensive

Remerciements

Merci à Madame Gobillard Delphine et Monsieur Melcior Michel.

Mots clés

Botnet : « Botnet » est une contraction des termes « robot » et « network » (réseau). Les cybercriminels utilisent des chevaux de Troie spéciaux pour violer la sécurité des ordinateurs de différents utilisateurs, prendre le contrôle de chacun de ces ordinateurs infectés et les regrouper au sein d'un réseau de « bots » gérables à distance.

Bot : Hôte infecté et faisant partie du botnet.

Botnet Master : Entité de contrôle d’un botnet

Obfuscation : Méthode de camouflage de code et de dissimulation d’action.

Payload : « Charge utile » élément permettant un piratage.

Exploit : Désigne l’exploitation d’une faille de sécurité.

Combolist : C’est une liste regroupant login et mot de passe sous la forme login : password

ESX : serveur VMware pour faire de la virtualisation.

IRC : Internet Relay Chat, protocole de communication textuel sur Internet.

Résumé du projet

Le projet vise à développer un botnet à usage éducatif dans le contexte d’une exploitation d’un réseau d’entreprise. Un fichier corrompu par un malware qui a été placé sur l’ordinateur d’un salarié.

L’agent infectieux se propage dans le réseau via les partages de fichiers entre groupes et utilisateurs dans l’Active Directory. Le but est d’infiltrer et d’infecter un réseau de manière invisible pour obtenir des informations sensibles. Des modules sont implémentés dans l’agent, comme un module créant un Reverse Shell, donnant un accès distant via un terminal sur l’hôte à l’attaquant.

Sommaire

**Introduction**

**1.Etude de cas**

- Analyse d’un squelette

- Environnement de déploiement

**2. Master et serveur C&C**

- Infrastructure

- Charge utile

- Détection et camouflage

- Modules

**3. Scénario et infection**

- Infrastructure de test

- Environnement de déploiement

Introduction

Le sujet vise à développer un Botnet à usage éducatif, dans le but d’occuper un réseau type entreprise, de recueillir des informations sensibles, ainsi que d’asservir les hôtes à différentes tâches.

Dans le bus d’étudier l’utilité ainsi que les méthodes utilisées actuellement par les pirates pour comprendre et envisager des contre-mesures.

Le sujet répond donc à la probatique :

**Peut-on privilégier l’attaque pour mieux défendre ?** ou l’étude et le développement d’un botnet pour en étudier les contre-mesures à adopter.

1. **Etude de cas**

**A - Analyse d’un squelette**

**Mirai Botnet**

Le Botnet Mirai est le plus connus des botnets, car il a sévèrement sévi sur internet, réalisant des attaques DDoS sur des infrastructures web et serveurs, notamment pour avoir DDoS le site web d’un journaliste ayant critiqué le botnet dans un article. Il est également connu pour être l’architecture de base des botnets actuels. (Carte de prolifération : **Annexe A1**)

Le Botnet Mirai possédait une méthodologie d’auto-propagation en utilisant des failles de sécurités dans les caméras de surveillance autour du monde, ces caméras sont connectées sur internet et possèdent donc une adresse IP. Le but principal du botnet Mirai était l’asservissement de petits objets connectés afin d’en avoir le plus grand nombre. Cette méthode de ciblage permettait d’obtenir un très grand nombre d’hôtes. Les ordres étaient envoyés via la méthode Command & Control (C&C), ces commandes installaient des crypto-mineurs et des scripts de DDoS.

Infographie méthode de propagation : **Annexe A3**

Ce botnet possédait des caractéristiques propres comme un tableau de Blacklist, listant les adresses IP à ne pas attaquer, notamment les adresses IP des sites gouvernementaux. Il était également équipé d’une table de combolist avec les identifiants et les mots de passes par défaut des fabriquant des caméras. Le corps de l’agent se caractérise par du multiprocessing, de la persistance et des modules pouvant être uploadé.

Exemple de table Combolist : **Annexe A4**

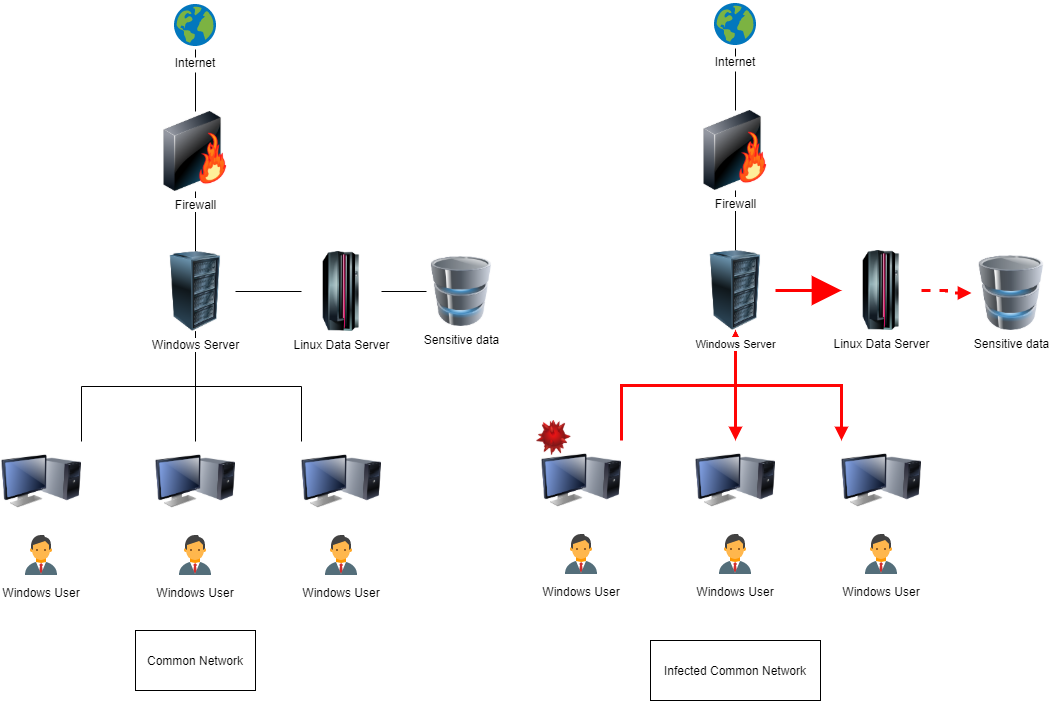
**Rustock Botnet**

Rustock est connu pour avoir été l’un des premiers botnets démantelé par Microsoft et le FBI, il reste aujourd’hui le botnet avec la vitesse de propagation la plus élevé de l’histoire avec 12 millions de spams par minutes, soit 17.28 milliards de spams par jours. L’agent infectieux avait la spécificité de ne pas être persistent

Carte de prolifération : **Annexe A2**

**B - Environnement de déploiement**

Dans le monde des entreprises les infrastructures sont semblables à celles de l’annexe **B1**, dans le contexte de notre projet l’infrastructure réseau sera semblable à l’annexe suivante.

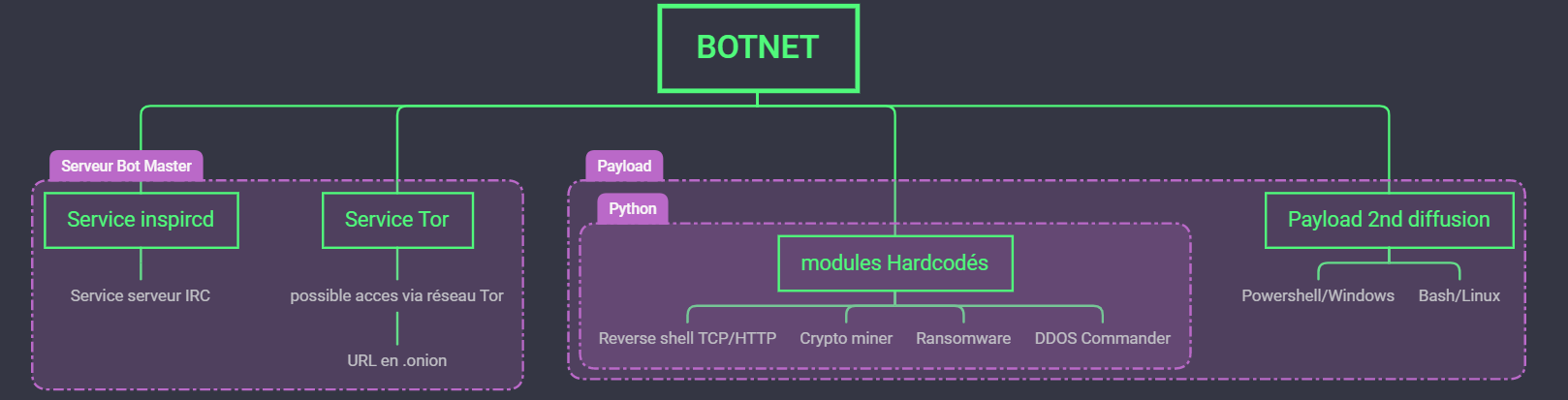


L’environnement se compose de plusieurs postes Windows destinés à des employés administratifs. Un serveur Windows 2016 possédant une mauvaise configuration dû à un administrateur non consciencieux. L’agent infectieux va se déployer sur la machine où il est exécuté, puis va chercher la présence d’un partage de fichier sur l’hôte, une fois trouvé il va infecter les fichiers présents dans le partage. Quand les autres utilisateurs iront chercher les fichiers partagés ils deviendront eux aussi des infecté et rejoindrons le botnet. Dans le cas ou un administrateur viendrait à naviguer dans les fichiers partagés, lui aussi deviendrait infecté.

1. **Master & serveur C&C**

**A – Infrastructure**

Le botnet sera basé sur une infrastructure Command & Control. Cette méthode de commandement est simple et efficace à petite échelle. L’agent est développé en langage python et le serveur est un service de serveur IRC sous Linux nommé Inspircd. L’agent communique avec le serveur via le réseau Tor, lui permettant de devenir anonyme et d’encrypter les informations envoyées.



**B – Charge utile**

Les charges utiles sont composées de scripts en PowerShell et Python, permettant une exécution précise et conforme avec Windows afin d’éviter les erreurs lors du déploiement du payload sur l’hôte infecté. Les payload sont hardcodés dans le code de l’agent.

Les payloads les plus courants qui constituent l’armement dit « létaux » sont du type : Ransomewares, Vers, DDoS… Le projet vise la discrétion du déploiement, les payloads n’ont pour but que le recel d’information et l’infection.

Les charges utiles présentent dans le projet donnent des avantages comme des inconvénients. En effet l’agent nécessite de nombreuses dépendances indispensables, qui sont pour la plupart, composés de beaucoup de fichiers. L’optimisation de code entre en jeu à ce moment-là.

Le payload nommé « Windows Key Retriever » nécessite lui une dépendance python nommé « Subprocess » lui permettant d’exécuter des commandes Powershell, qui sont très utiles dans ce domaine, cela permet de faire du « Remote Code Execution », c’est-à-dire d’exécuter des codes malicieux sur l’infecté, la faille « Remote Code Execution » constitue l’une des faiblisses les plus critique.

Le payload « OS ID » lui, nécessite la dépendance « Platform », lui permettant d’identifier le système d’exploitation sur lequel il se trouve.

Le payload « IP Retriever » utilise la dépendance « request » pour obtenir son adresse IP.

Le payload « PDFworld » a besoin de la dépendance « PyPDF2 », il permet de lire et modifier des fichiers type PDF. Ce qui est très utile dans une optique de recel d’information.

**C – Détection et camouflage**

Les réseaux d’entreprises sont obligatoirement équipés d’antivirus, de ce fait des recherches en obfuscations et usurpations de propriétés ont été faites. L’agent est le code ou l’obfuscation est présente.

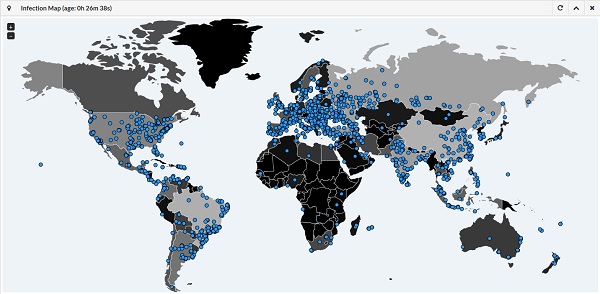
Le code à été modifier pour le rendre illisible par les antivirus communs comme Windows Defender, Avast, McAfee, … Nous avons utilisé le site Virustotal.com pour effectuer des tests de détection sur plus de 70 antivirus.

Le score suivant a été obtenu par l’agent infectieux (**Annexe C1**)

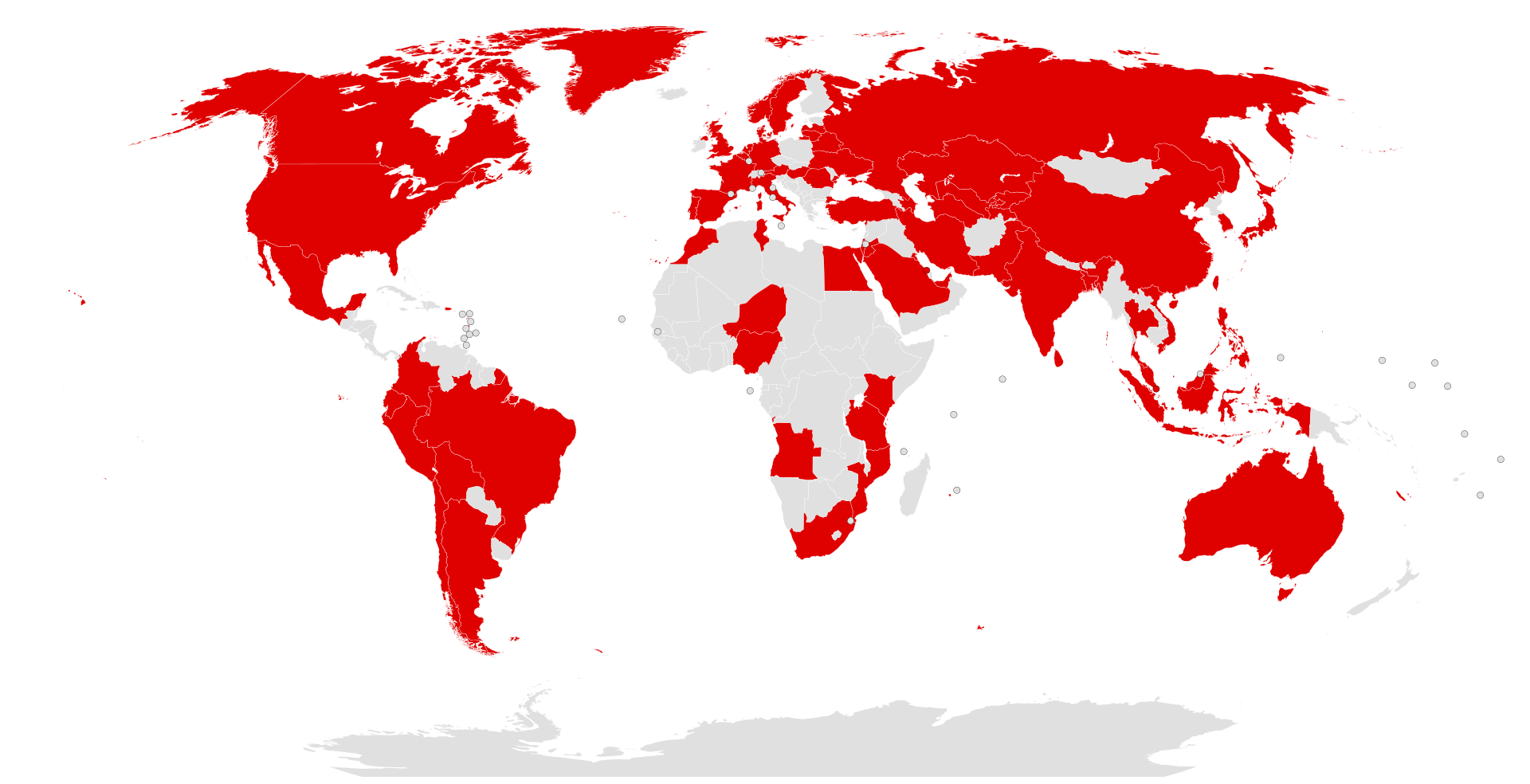
**D – Scénario & tests**

**ANNEXES**

**A1**



**A2**

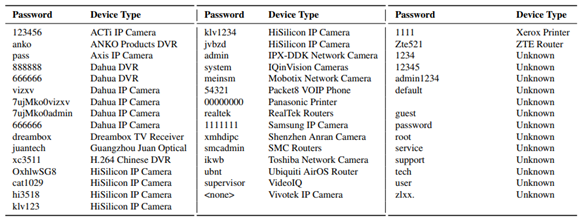


**A3**

Diagram

Description automatically generated

**A4**



**B1**

Timeline

Description automatically generated

