Rust Remote.Access.Trojan

Antoine MARTIN, Wesley EDE, Amad MOHAMMAD, Denis REMACLE

29 novembre 2021

1 Objectifs

1.1 Besoin

Nous souhaitons mieux comprendre le fonctionnement d'un Remote Access Trojan, l'écriture d'une relation client-serveur ainsi qu'apprendre le Rust.
Et pour cela rien de mieux que de mettre en place un tel programme.

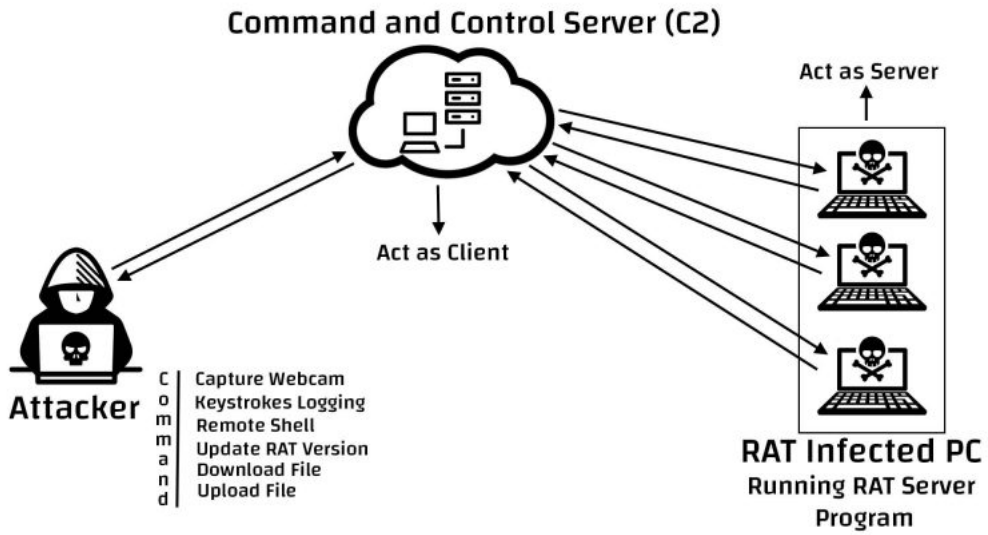
Ce projet commun sera également une vitrine de notre compétence et pourrait nous permettre d'acquerir une plus grande crédibilité sur le long terme.
La solution sera donc publié en tant que logiciel libre une fois la soutenance passée et par conséquent nous devrions signaler le répository github aux grands éditeurs de solutions cybersecurité.

1.2 Notre Approche

Nous allons procéder par équipes de deux : Amad / Denis (Serveur et payloads) et Wesley / Antoine (Client Windows et payloads).
Cela est dû à la disparité de niveau en algorithmique et en programmation dans notre équipe et cela permettra aux plus faibles de suivre le mouvement et de suggérer du code sans casser le code actuel.

Ça permettra également l'écriture progressive de la documentation technique de la solution.
Nous allons également réunir les deux équipes au moins une fois par semaine pour faire un état de l'avancement du projet.

Nous allons donc mettre en place un architechture similaire à celle si :



2 Planning

Langage privilégié : Rust (Programme)
Langages scripts infection : Powershell

En février nous aurons conceptualisé la relation client-serveur.
Le client compatible Windows pourra
- "Contacter" le serveur (Connexion, extinction, heartbeat toute les heures)
- Possibilité d'exécuter un reverse shell envoyé depuis le serveur

et le Serveur pourra :
- Écouter sur un port et récupération de la liste des clients connectés
- Gérer les differents clients (nommer, grouper, supprimer le client à distance)
- Envoyer d'instruction au client (reverse shell)

Ensuite en avril, on aura ajouté différents payloads, en voici une liste qui seront mises en place :
- Keylogger
- Enregistrement du micro
- Enregistrement de la caméra
- Récupération des mots de passe sur navigateur

La connexion client serveur sera également chiffrée afin d'essayer de bypass certains firewalls.

Puis en Mai nous aurons commencer à travailler sur la mise en place d'une interface web pour la gestion du serveur avec les deux équipes en parallèle.

3 Un état des lieux sur les solutions existantes

Un RAT (Remote Administration Tool - Outil d'administration à distance) est un programme permettant la prise de contrôle totale, à distance, d'un ordinateur depuis un autre ordinateur.
Il est constitué de deux parties : le "client" et le "serveur". Le client est installé sur l'ordinateur de celui qui prend le contrôle et le serveur est installé sur l'ordinateur contrôlé.

Il en existe de tout à fait légitime comme teamviewer ou le take control de N-Able RMM.
Mais il en existe aussi des malveillants comme nous allons le voir par la suite.

Il existe divers Trojan RAT, un des plus anciens est BlackShades, le plus utilisé est Darkcomet ou NanoCore.
Ces chevaux de troie sont vendus mais on peut trouver des versions crackés, des tutoriels (dont des vidéos sur youtube) existent à foison.
Dès lors, n'importe qui, qui a très peu de connaissances peut se créer son propre Botnet (réseaux de PC infectés).

Les RAT fonctionnent en Client/Serveur ;

Vous faites tourner un serveur, soit vous louez une machine pour cela, soit la plupart le font tourner sur leur ordinateur personnel.
Le serveur se présente sous la forme d'une interface qui permet de piloter les clients (machine infectée).
Et une partie cliente qui se connecte au serveur : Le but est d'arriver à faire exécuter la partie cliente à l'insu de l'utilisateur du PC afin de pouvoir prendre le contrôle de la machine, ce qui est en général assez simple, via dû le social engineering, puisque les personnes visées ne sont en général pas très férues d'informatique.