Ransomware WEB

Analyse Statique

1. Installer le logiciel theZoo

theZoo est une base de donné des malwares avec des exemples qu'on peut les utiliser

Toute utilisation hors ce cadre peut soumettre l'utilisateur à des poursuites judicaire

Pour plus d'info voir le lien GitHub - ytisf/theZoo: A repository of LIVE malwares for your own joy and pleasure. theZoo is a project created to make the possibility of malware analysis open and available to the public.

Ce programme est conçu pour Python 2.7.

Si vous avez une version ultérieure, veuillez suivre ce lien How to change from default to alternative Python version on Debian Linux - Linux Tutorials - Learn Linux Configuration

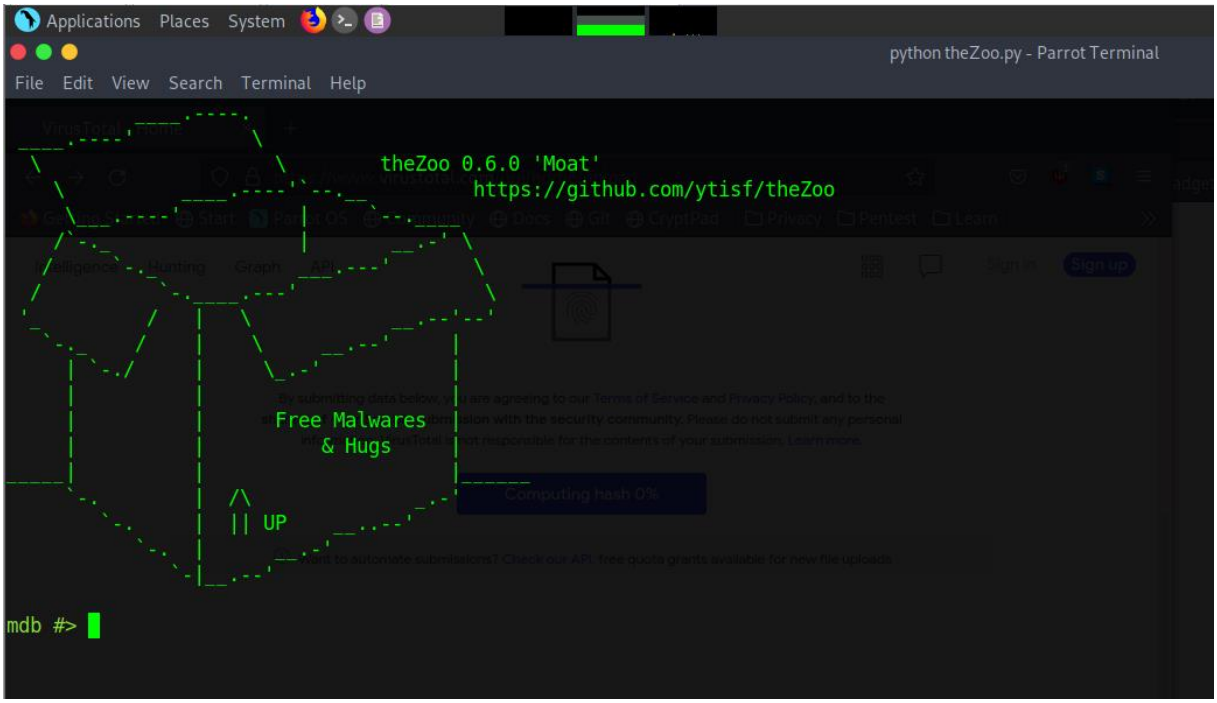
Etapes d'installation:

- git clone https://www.github.com/ytisf/theZoo
- cd theZoo
- pip install --user -r requirements.txt

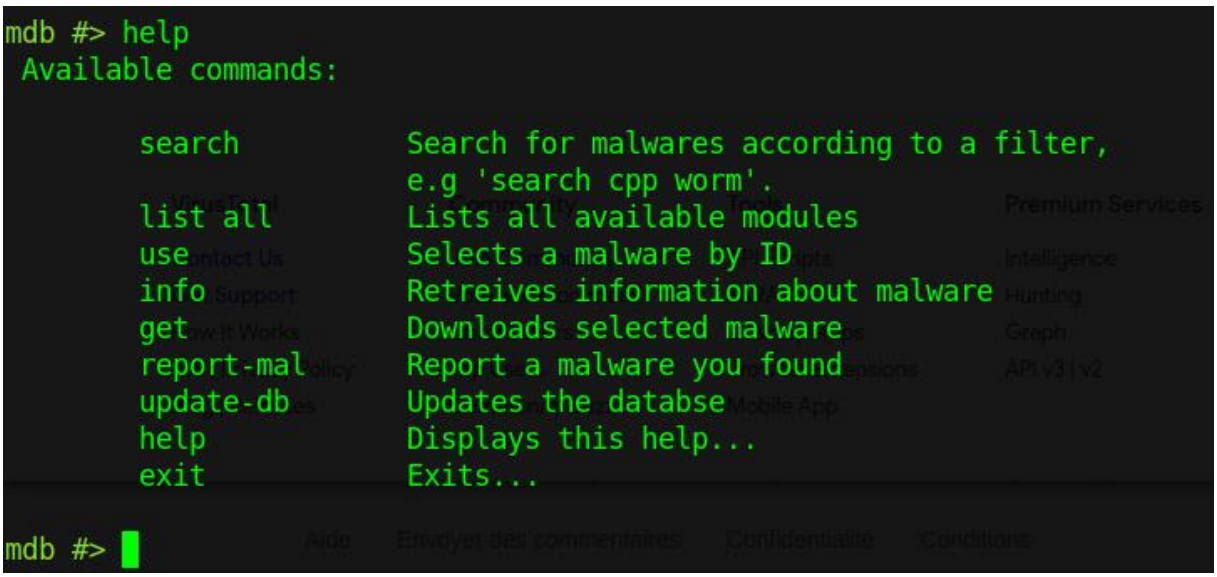
2. Récupérer le ransomware

Une fois vous avez installé theZoo, on exécute :

- python theZoo.py

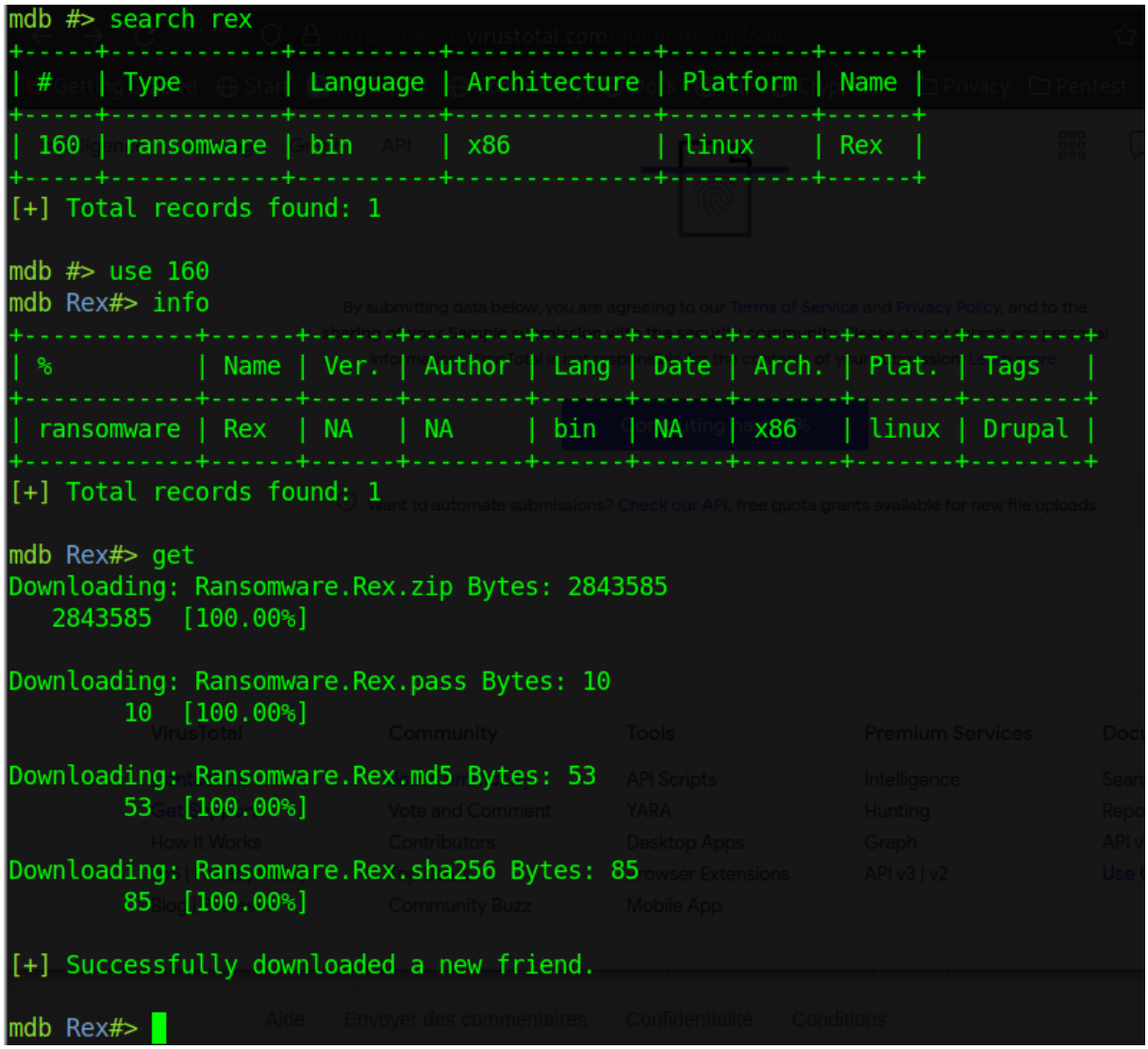


On exécute la commande **help** pour savoir les commandes qu'on peut utiliser



On exécute ces commandes pour télécharger le Ransomware REX

- search rex
- use 160
- info
- get



Maintenant, en utilisant le mot de passe dans le fichier **Ransomware.Rex.pass** et la commande **unzip**, on déchiffre le zip file **Ransomware.Rex.zip**

3. Installer l'anti-malware clamav

Pour faire l'analyse, on a besoin d'installer l'anti-spam clamav

- sudo apt install clamav

4. Analyse Statique

Veuillez répondre à ses questions sans exécuter le ransomware sauf pour la question 7

Quesrtion 7 est une introduction à l'analyse dynamique

1. À l'aide de l'outil **file,** identifier les informations suivantes :
    - Plateforme d'exécution
    - Nature du fichier (binaire / librairie)
    - Architecture
2. À l'aide de l'outil strings, récupérer le Template du mail de notif utilisé pour envoyer la demande de la rançon à la victime
3. À l'aide de l'outil strings, Déterminer la liste des CMS qui peuvent être victime à ce ransomware ?
4. À l'aide de l'outil clamav, scanner le fichier ransomware
    - clamscan -v –debug ransomware_file
5. À l'aide de l'antivirus en ligne VirusTotal - Home, scanner le fichier ransomware
6. Selon le résultat du virusTotal, pourquoi il y a certain antivirus/anti-malware n'ont pas détecté ce ransomware
7. En exécutant ces commande , déterminer la première action du ransomware en désactivant la carte réseau de la VM
    - chmod u+x ransomware_file
    - ./ransomware_file -debug -log.http -log.dht

5. Analyse Dynamique

Seul le noyau peut effectuer des modifications en dehors de l'espace mémoire du processus. Le processus doit demander au noyau d'effectuer des tâches telles que la création de fichiers ou l'écriture d'une sortie. C'est là que les appels système entrent en jeu.

Les appels système sont l'interface utilisée par l'application pour demander des services au noyau. Les appels système sont généralement invoqués via des wrappers glibc et non directement vers le noyau en raison de la portabilité. Les appels système de bas niveau diffèrent entre les architectures, c'est pourquoi la glibc gère ces différences à la place du développeur.

Les appels système sont une interface que le malware utilise afin de causer un dommage réel au système. L'analyse des appels système peut nous aider à comprendre comment le malware interagit avec le système et comment il fonctionne dans les coulisses.

strace est un outil puissant pour tracer les appels système d'un fichier. Chaque ligne de la sortie strace est un appel système, et le premier appel système sera **execve** qui signifie exécuter le programme. Chaque appel système a une valeur de retour qui varie entre les appels.

La valeur du retour peut être un :

- Descripteur de fichier (entier)
- 0 en cas de succès
- -1 en cas d'erreur
- etc.

Exemple d'appels systèmes :

- open/openat – ouvrir et éventuellement créer un fichier.
- read – lire à partir d'un descripteur de fichier.
- access - vérifier les autorisations de l'utilisateur pour un fichier.
- write – écrire dans un descripteur de fichier.
- mkdir/mkdirat – créer des répertoires.
- connect – initier une connexion sur un socket.
- socket - crée un point de terminaison pour la communication.
- execve – exécuter le programme.

1. Exécutez ***strace whoami*** sur votre machine virtuelle Linux et examinez la sortie.
2. Identifier comment la commande whoami a pu récupérer le résultat depuis l'analyse de l'output

Scénario

Le ransomware REX possède plusieurs fonctionalités :

- Attaque Ransomware
- Bitcoin mining
- Attaque DDoS

On va focaliser sur la fonctionatlité « Attaque Ransomware ».

Le hacker utilise des robots (bots) afin de scanner le réseau et identifier les sites webs Drupal infecté par la faille CVE-2014-3704 (une injection SQL sur Drupal 7).

Le bot ajoute un nouveau compte administrateur, verrouille tous les articles de blog avec des notes, télécharge et exécute Rex.

Comme on a vu dans la question 7 du partie analyse statique, Rex envoi des requêtes vers des web services en ligne de géolocalisation par @ IP

Rex ne commence à être malveillant que s'il réceptionne une réponse sous une format bien déterminé depuis ses web services

Si pas de réponse selon le format souhaité donc il ne s'exécute pas et donc pas d'actions malveillantes

Afin de simuler tout ça, on a besoin de :

1. Un Lab avec serveur Web et MySQL et Drupal
2. Configurer Drupal afin de le rendre vulnérable au faille SQL injection
3. Exécuter REX **avec un accès internet** pour qu'il réceptionne une réponse depuis ces services web de géolocalisation
4. Attendre le résultat de la découverte et du scan du réseau pour qu'il identifie le site web Drupal infecté

Vu que ceci est difficile de le réaliser dans le TP, on va analyser le fonctionnement du fichier log correspondant au début d'exécution du ransomware Rex dans un environnement similaire (strace .txt).


strace.txt

En analysant ce fichier log :

1. Créer une liste des appels systèmes utilisé par REX et donner la définition de chacune
2. Pourquoi REX a cherché ce fichier :

   openat(AT_FDCWD, "/proc/sys/net/core/somaxconn", O_RDONLY|O_LARGEFILE|O_CLOEXEC) = 3

3. Expliquer cette ligne:

   connect(8, {sa_family=AF_INET, sin_port=htons(5099), sin_addr=inet_addr("83.241.220.100")}, 16) = -1 EINPROGRESS (Operation now in progress)