Ethical Hacking:
Projet Botnet

Présenté par:
Charbel Farah
Hussein Kaddouh
Marie Joe Louca
Joseph Raji

Pour: Pr. Rida Khatoun

Sommaire

1. Introduction

a. Compréhension des Botnets

Les botnets représentent une menace persistante et évolutive dans le paysage de la sécurité informatique. Ce rapport vise à examiner en profondeur la nature, les mécanismes de fonctionnement et les implications des botnets sur la sécurité des systèmes informatiques.

i. Définition et Fonctionnement

- Un botnet est un réseau de dispositifs informatiques infectés par des logiciels malveillants, sous le contrôle d'un individu malveillant, appelé botmaster.
- Les botnets exploitent des vulnérabilités de sécurité pour infecter des dispositifs, créant ainsi des "bots" ou "zombies" contrôlés à distance.
- Les botmasters utilisent un serveur de commande et de contrôle (C&C) pour coordonner les activités des bots dans le botnet.

ii. Taille et Échelle

- Les botnets peuvent varier considérablement en taille, allant de quelques dizaines à des millions de dispositifs infectés.
- Leur échelle massive leur permet d'exécuter des attaques coordonnées et de grande envergure.

2. Fonctionnalités et Utilisations des Botnets

a. Types d'Activités Malveillantes

- Les botnets sont utilisés pour une gamme variée d'activités malveillantes telles que le vol d'informations, le spam, les attaques par déni de service distribué (DDoS), et le minage de cryptomonnaie.
- Leur flexibilité permet aux botmasters d'adapter les activités du botnet en fonction de leurs objectifs spécifiques.

b. Implications pour la Sécurité Informatique

- Les botnets représentent une menace sérieuse pour la confidentialité, l'intégrité et la disponibilité des données informatiques.
- Les attaques par botnets peuvent causer des dommages financiers significatifs, nuire à la réputation des entreprises et perturber les opérations en ligne.

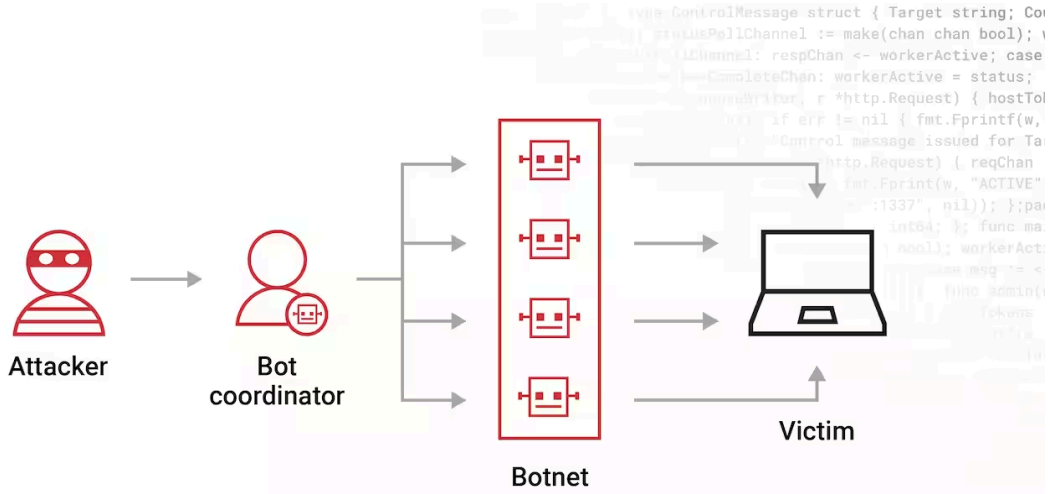
3. Méthodes de Protection et de Détection

a. Prévention des Infections

- Maintenir à jour les logiciels et les systèmes d'exploitation pour corriger les vulnérabilités de sécurité connues.
- Utiliser des solutions antivirus et pare-feu efficaces pour détecter et bloquer les logiciels malveillants.
- Sensibiliser les utilisateurs aux pratiques de sécurité en ligne, notamment en évitant les emails de phishing et les téléchargements de sources non fiables.

b. Détection et Mitigation des Attaques

- Surveiller le trafic réseau pour détecter les activités suspectes associées aux botnets.
- Mettre en œuvre des mécanismes de blocage pour limiter l'impact des attaques par botnets, comme la mise en quarantaine des dispositifs infectés.



4. Présentation du scénario

a. SYN Flooding Attaque

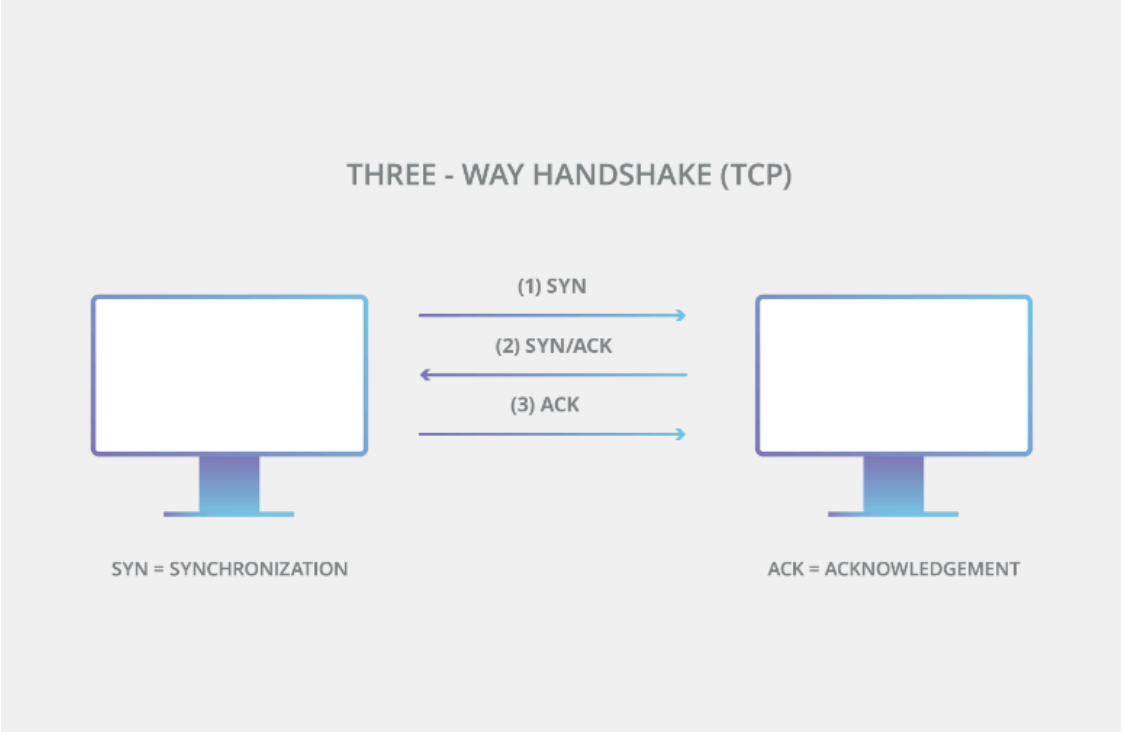
i. Qu'est-ce qu'une attaque SYN flood ?

Une attaque SYN flood (attaque semi-ouverte) est un type d'attaque par déni de service (DDoS) qui vise à rendre un serveur indisponible pour le trafic légitime en consommant toutes les ressources serveur disponibles. En envoyant à plusieurs reprises des paquets de demande de connexion initiale (SYN), le pirate est en mesure de submerger tous les ports disponibles sur une machine serveur ciblée, ce qui oblige l'appareil ciblé à répondre lentement au trafic légitime, ou l'empêche totalement de répondre.

ii. Comment l'attaque SYN Flood se déroule?

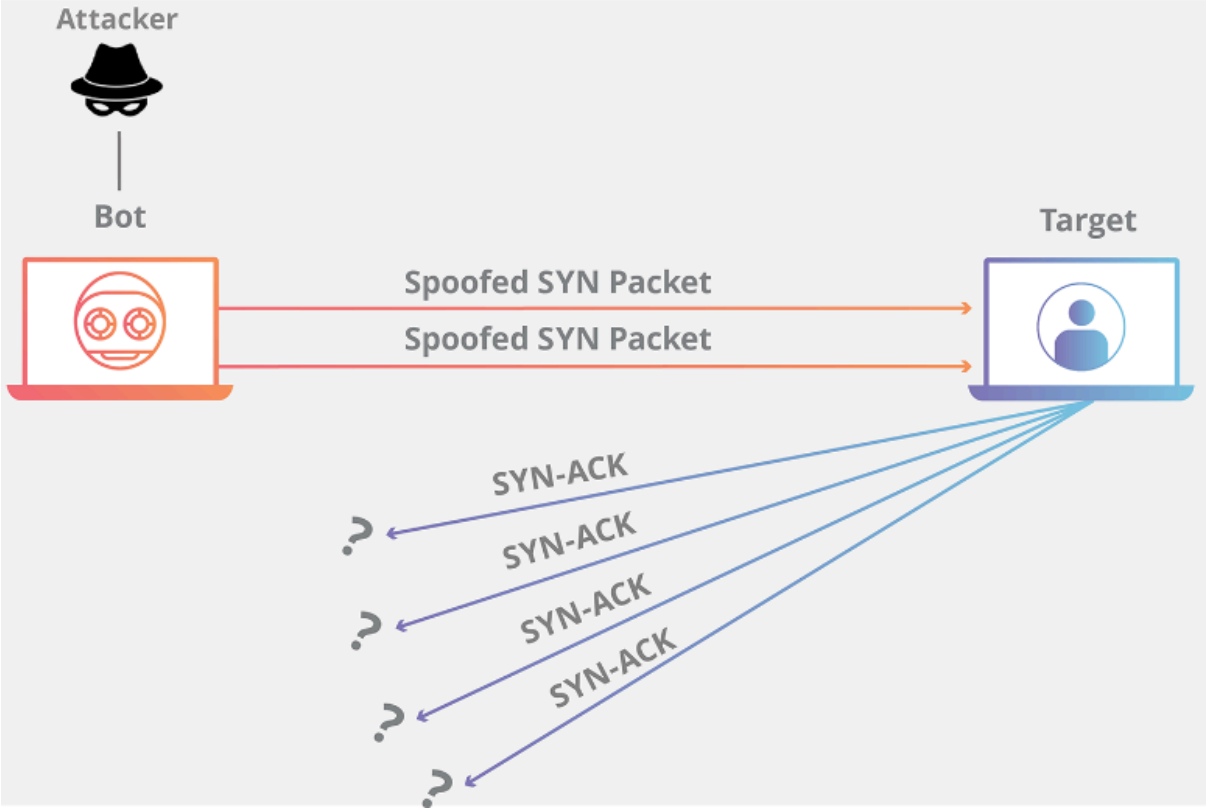
Les attaques SYN flood fonctionnent en exploitant le processus d'établissement de liaison d'une connexion TCP. En temps normal, la connexion TCP présente trois processus distincts pour établir une connexion.

1. Tout d'abord, le client envoie un paquet SYN au serveur afin d'établir la connexion.
2. Le serveur répond ensuite à ce paquet initial avec un paquet SYN/ACK, afin d'accuser réception de la communication.
3. Enfin, le client renvoie un paquet ACK pour accuser réception du paquet provenant du serveur. Après avoir terminé cette séquence d'envoi et de réception de paquets, la connexion TCP est ouverte et capable d'envoyer et de recevoir des données.



Pour créer un déni de service, un pirate exploite le fait qu'après réception d'un paquet SYN initial, le serveur répond avec un ou plusieurs paquets SYN/ACK et attend l'étape finale du handshake. Voici comment cela fonctionne :

1. Le pirate envoie un volume élevé de paquets SYN au serveur ciblé, souvent avec des adresses IP usurpées.
2. Le serveur répond ensuite à chacune des demandes de connexion et laisse un port ouvert prêt à recevoir la réponse.
3. Pendant que le serveur attend le dernier paquet ACK, qui n'arrive jamais, le pirate continue d'envoyer plus de paquets SYN. L'arrivée de chaque nouveau paquet SYN oblige le serveur à maintenir temporairement une nouvelle connexion de port ouverte pendant un certain temps, et une fois que tous les ports disponibles ont été utilisés, le serveur ne peut plus fonctionner normalement.



Source[1]

En réseau, lorsqu'un serveur laisse une connexion ouverte mais que la machine de l'autre côté de la connexion ne l'est pas, la connexion est considérée comme étant à moitié ouverte. Dans ce type d'attaque DDoS, le serveur ciblé laisse en permanence des connexions ouvertes et attend que chaque connexion expire avant que les ports ne redeviennent disponibles. Le résultat est que ce type d'attaque peut être considéré comme une « attaque semi-ouverte ».[1]

b. Attaque DDoS de type Ping (ICMP) flood

i. Qu'est-ce qu'une attaque de type Ping (ICMP) flood ?

Un Ping flood est une attaque par déni de service au cours de laquelle l'attaquant tente de submerger un appareil ciblé avec des paquets de demande d'écho ICMP, rendant la cible inaccessible au trafic normal. Lorsque le trafic d'attaque provient de plusieurs appareils, l'attaque devient une attaque DDoS ou déni de service distribuée.

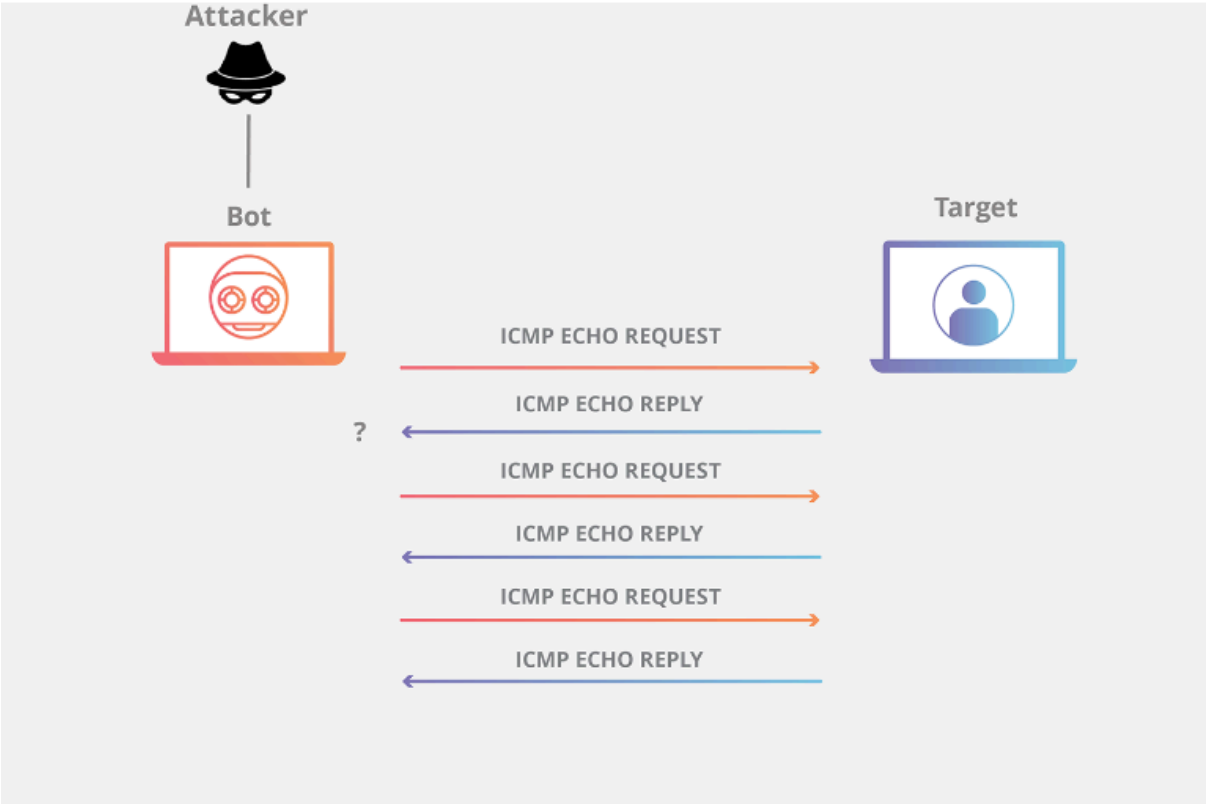
ii. Comment fonctionne une attaque de type Ping flood ?

Le protocole ICMP (Internet Control Message Protocol), qui est utilisé dans une attaque Ping Flood, est un protocole de couche Internet utilisé dont les dispositifs de réseau se servent pour communiquer. Les outils de diagnostic du réseau, traceroute et ping s'exécutent tous deux avec ICMP. Généralement, les messages ICMP de demande et de réponse par écho sont utilisés pour envoyer un ping à un interface réseau afin de diagnostiquer la santé et la connectivité de l'appareil et la connexion entre l'expéditeur et ledit appareil.

Une requête ICMP nécessite des ressources de serveur pour traiter chaque requête et pour envoyer une réponse. La requête nécessite également de la bande passante à la fois pour le message entrant (demande d'écho) et pour la réponse sortante (réponse d'écho). L'attaque Ping Flood vise à submerger la capacité du dispositif ciblé à répondre au nombre élevé de demandes et/ou à surcharger la connexion réseau avec du trafic fictif. Si le pirate fait en sorte que plusieurs dispositifs dans un ciblent la même propriété Internet ou le même composant d'infrastructure avec des requêtes ICMP, le trafic d'attaque est considérablement augmenté, ce qui peut entraîner une perturbation de l'activité normale du réseau. Historiquement, les attaquants usurpent souvent une fausse adresse IP afin de masquer le périphérique émetteur. Avec les attaques de botnets modernes, les pirates voient rarement la nécessité de masquer l'adresse IP du bot, et s'appuient plutôt sur un vaste réseau de bots non piratés pour saturer la capacité d'une cible.

La forme DDoS d'un Ping (ICMP) flood peut être décomposée en étapes répétitives

1. Le pirate envoie de nombreux paquets de requêtes d'écho ICMP au serveur ciblé via plusieurs appareils.
2. Le serveur ciblé envoie ensuite un paquet d'écho de réponse ICMP à l'adresse IP de chaque appareil demandeur en guise de réponse.



Source[2]

L'effet dommageable d'une Ping Flood est directement proportionnel au nombre de requêtes adressées au serveur visé. Contrairement aux attaques DDoS basées sur la réflexion comme l'amplification NTP et l'amplification DNS, le trafic d'une attaque Ping Flood est symétrique ; la quantité de bande passante que le dispositif ciblé reçoit est simplement la somme du trafic total envoyé par chaque bot.[2]

5. Resources used for the project

a. Bot machines:

- **Type**: Virtual machine on VirtualBox
- **Operating System**: Ubuntu version 20.04
- **Number of Bots Needed**: At most 6
- **Requirements**:
    - Python3
    - XAMPP version 8.2.12

b. C2 server

- **Type**: Virtual machine on VirtualBox
- **Operating System**: Kali Linux
- **Number of Machines Used**: 1
- **Requirements**:
    - Python3

c. Target machine

- **Type**: Virtual machine on VirtualBox
- **Operating System**: Ubuntu 20.04
- **Number of Machines**: 1
- **Requirements**:
    - Apache Server 2.4.58

6. Configuration

a. Setting up the bot machines

Il n'y a pas de moyen idéal pour infecter un hôte afin qu'il devienne un bot ou un zombie ; la méthode choisie par nous était à travers une simple application web que nous avons écrite. L'application web était hébergée sur un serveur Apache téléchargé sur XAMPP sur Ubuntu 20.04. L'application web était un simple téléchargement de fichiers, où un utilisateur téléchargerait un fichier en elle. La principale vulnérabilité ciblée dans cette application web est le téléchargement de fichiers et le manque de vérification de l'extension du fichier, donc nous sommes capables de télécharger un fichier PHP qui est stocké sur le serveur de la machine Web et de l'exécuter en allant au chemin correct. Pour des améliorations sur le serveur, il devrait bien sûr y avoir une vérification de l'extension de fichier et une fois qu'un fichier est téléchargé, changer son nom en un nom aléatoire afin qu'il ne puisse pas être appelé côté client.

Pour configurer le serveur Apache sur la machine bot, suivez ces étapes :

- Installez la dernière version de XAMPP depuis https://www.apachefriends.org/
- Après l'installation, vous devez aller à /opt/lampp pour configurer le serveur. Par défaut, le serveur Apache fonctionnera sur le port 80, vous pouvez le changer dans /opt/lampp/etc/httpd.conf et changer cette ligne : Listen 80, à Listen <numéro_de_port_de_votre_choix> (des privilèges root sont nécessaires pour modifier ce fichier)
- Maintenant, allez dans le fichier php.ini situé à /opt/lampp/etc/php.ini et assurez-vous d'avoir ces lignes avec ces valeurs mises à jour pour vous assurer que le téléchargement de fichiers fonctionne (des privilèges root sont nécessaires pour modifier ce fichier) :
    - output_buffering=4096
    - max_execution_time=120
    - max_input_time=60
    - memory_limit=512M
    - post_max_size=1024M
    - file_uploads=On
    - upload_max_filesize=1024M
- Insérez le fichier **index.php** trouvé dans le github dans **/opt/lampp/etc/htdocs/vulnerable_app** (créez vulnerable_app) afin d'avoir l'application web vulnérable. Vous pouvez y accéder via **localhost:80/vulnerable_app/index.php.**
- Dans le même répertoire, créez un répertoire appelé "uploads" pour stocker les fichiers téléchargés. C'est là que le client pourra exécuter son code PHP téléchargé en appelant l'URL **<ip_bot_machine>:80/vulnerable_app/uploads/<nom_fichier_PHP.php>**

- Changez le propriétaire du répertoire **/opt/lampp/etc/htdocs/vulnerable_app** de root à daemon avec la commande **sudo chown daemon:daemon -R /opt/lampp/etc/htdocs/vulnerable_app**. Cela est nécessaire pour que l'application web ait un accès en écriture au fichier "uploads" et insère les fichiers téléchargés.
- Pour que l'hôte devienne un bot, une connexion doit être établie avec le serveur C2 qui est réalisée à travers un fichier Python qui est téléchargé. Le fichier Python sera exécuté par un fichier PHP qui est également téléchargé du côté client. Certaines commandes dans le script Python utilisé nécessitent des permissions sudo, comme celles impliquant scapy. Comme le but du projet n'était pas l'escalade des privilèges, nous avons trouvé une solution rapide pour exécuter la commande sudo en tant qu'utilisateur daemon. Configurez **/etc/sudoers** pour permettre l'exécution de commandes spécifiques sans mot de passe avec les instructions suivantes :
    - Ouvrez le fichier sudoers avec **sudo visudo.**
    - Ajoutez la ligne suivante pour accorder à **daemon** la permission d'exécuter n'importe quelle commande sudo sans mot de passe :
      daemon ALL=(ALL) NOPASSWD: ALL

Maintenant, vous êtes prêt du côté bot.

b. Setting up the C2 server

Il n'y a pas grand-chose à faire ici, il suffit d'installer les dépendances Python trouvées sur GitHub dans le fichier **requirements.txt** avec **pip install -r requirements.txt**.
Avant de démarrer l'application, créez un fichier .env contenant ce qui suit :

- PASSWORD="admin"
- SECRET_KEY=<INSERT A SECRET KEY>
- SQLALCHEMY_DATABASE_URI="sqlite:///schedule.db"
- SERVERIP=<YOUR MACHINE IP>
- SERVERPORT=<PORT YOU WANT THE C2 SERVER TO OPEN>

Ensuite, exécutez simplement **python3 app.py**, l'interface web sera disponible sur **localhost:5000** et le serveur C2 écoutera sur l'adresse IP de l'appareil et sur le port 50000.

Maintenant, vous êtes prêt du côté C2.

c. Setting up the target host

Afin de réaliser les attaques de type ICMP et TCP flooding, nous devons configurer la bande passante de la machine cible. Une bande passante plus basse signifie un serveur plus vulnérable à ce type d'attaques. Comme nous travaillons sur une machine virtuelle, il offre une manière de manipuler la bande passante de la machine avec VBoxManager. Pour y parvenir, ajoutez VBoxManage à votre chemin d'accès. Sur Windows, vous pouvez le trouver ici : **C:\Program Files\Oracle\VirtualBox\VBoxManage.exe.** Accédez à votre terminal et exécutez les commandes suivantes :

- VBoxManage bandwidthctl "VM name" add Limit --type network --limit 0.5m
  (0.5m represents 0.5 Megabits per second)

- VBoxManage modifyvm "VM name" --nicbandwidthgroup1 Limit

Documentation links:

- https://download.virtualbox.org/virtualbox/7.0.6/UserManual.pdf
- https://docs.oracle.com/en/virtualization/virtualbox/6.0/user/vboxmanage-bandwidthctl.html

Particulièrement pour les attaques de type TCP flooding, vous devez configurer la manière dont le tampon TCP et le traitement fonctionnent sous Ubuntu, car de nombreuses configurations sont mises en place pour se protéger contre les attaques de type TCP flooding. Un travail minimal a été effectué sur le serveur Apache car il ne gère pas les paquets au niveau de la couche 4, mais manipule les paquets au niveau de la couche 7 après qu'une connexion a été établie. Il est donc préférable de configurer le serveur Apache pour les attaques de type slowLoris et HTTP flooding.

Accédez au répertoire **/proc/sys/net/ipv4** et modifiez les fichiers suivants (les privilèges root sont nécessaires pour modifier ces fichiers) :

- **tcp_syncookies:**Ce paramètre active ou désactive le mécanisme des cookies SYN, qui est utilisé pour se protéger contre les attaques de type SYN flood. Désactiver les cookies SYN rend le système plus vulnérable. Réglez cette valeur sur 0.
- **tcp_max_syn_backlog:** Ce paramètre contrôle le nombre maximum de demandes de connexion en attente qui n'ont pas encore reçu d'accusé de réception du client connectant. Augmenter cette valeur permet plus de connexions demi-ouvertes, ce qui peut rendre votre système plus susceptible aux attaques de type SYN flood s'il est réglé trop haut sans surveillance appropriée. Réglez cette valeur sur **256**.
- **tcp_synack_retries:** Ce paramètre définit le nombre de fois que le système va retransmettre un SYN-ACK avant de décider que la connexion a échoué. Diminuer cette valeur pourrait potentiellement libérer des ressources plus rapidement, mais cela pourrait également abandonner des connexions légitimes dans des conditions réseau médiocres. Réglez cette valeur sur **3**.
- **tcp_abort_on_overflow:** Lorsqu'il est réglé sur 1, le système abandonnera les connexions si la file d'attente de socket est pleine. Mettre cette valeur sur 0 peut faire en sorte que votre système tente de gérer les connexions même sous stress, aggravant potentiellement l'impact d'une attaque de type SYN flood. Réglez cette valeur sur **0**.

Allez dans **/proc/sys/net/core** et modifiez les fichiers suivants (des privilèges root sont nécessaires pour modifier ce fichier):

- **somaxconn:** Ce paramètre ajuste le nombre maximum de connexions pouvant être mises en attente pour être acceptées par votre application. L'augmenter permet plus de connexions en attente, ce qui peut aggraver les effets d'une attaque de type SYN flood. Réglez cette valeur sur 256.

Application des modifications :

Pour appliquer ces modifications, vous pouvez les ajouter à votre /etc/sysctl.conf. Voici comment modifier sysctl.conf :

- Sudo nano **/etc/sysctl.conf**

- Ajoutez ou modifiez les paramètres avec les valeurs que vous choisissez en fonction des descriptions ci-dessus.
- Appliquez les modifications en exécutant : **sudo sysctl -p**

Maintenant, vous êtes prêt du côté de la cible.

d. Network set up

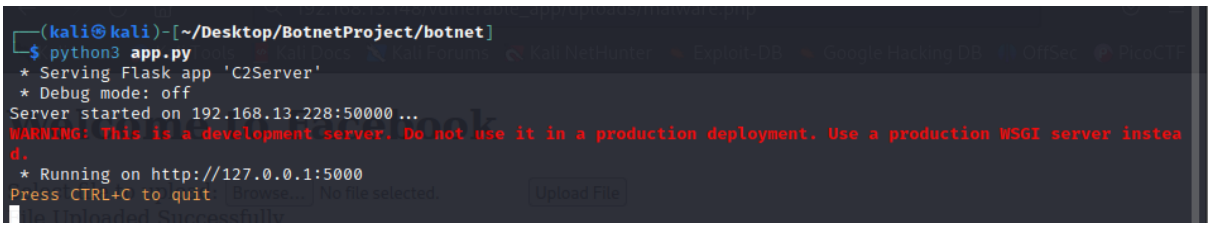
Pour que les machines virtuelles puissent communiquer entre elles, vous avez plusieurs configurations réseau parmi lesquelles choisir :

- Bridged adapter si vous souhaitez travailler sur plusieurs ordinateurs portables.
- NAT network
- Host-only adapter

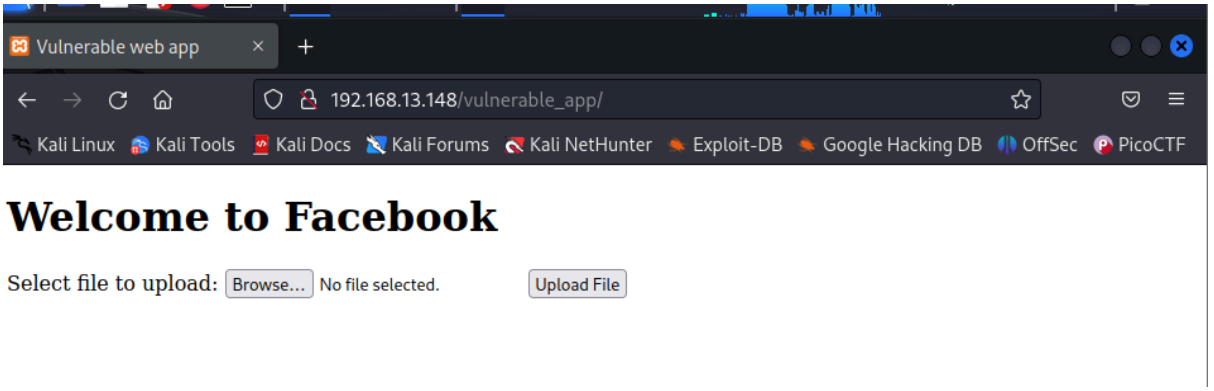
Toutes ces options sont efficaces pour créer un réseau dans VirtualBox afin que les différentes machines virtuelles puissent communiquer entre elles. Nous avons travaillé avec l'option d'adaptateur en mode Pont, car cela permet de communiquer entre différents ordinateurs portables sur le même réseau.
Vous devrez peut-être attribuer des adresses IP statiques en fonction de vos besoins.

7. Démarche

- Exécutez votre serveur C2 sur votre machine C2 avec la commande suivante : `**python3 app.py**`

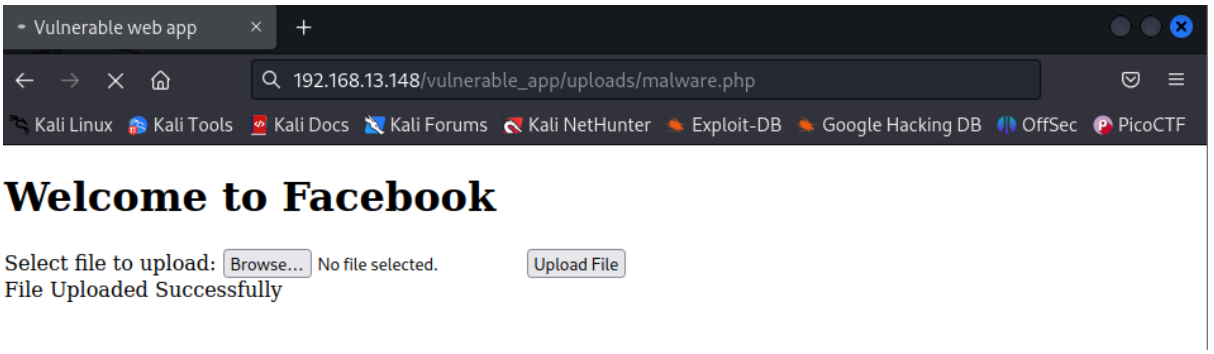


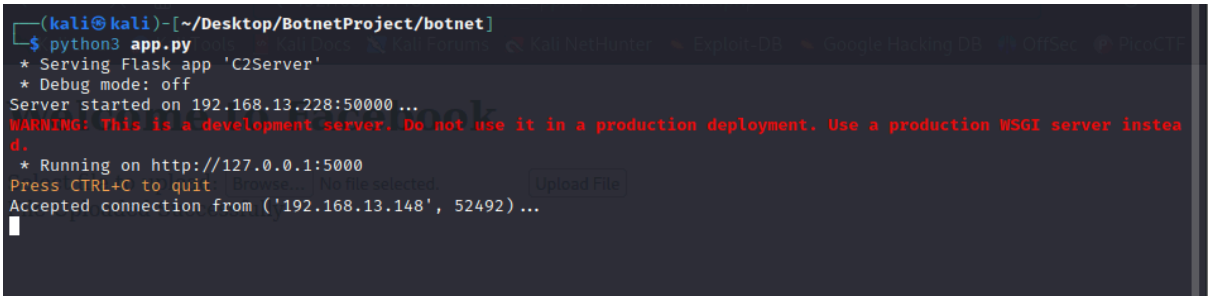
- Accédez à votre machine cible et démarrez le serveur XAMPP avec la commande suivante : `**sudo /opt/lampp/lampp start**`
- Sur votre serveur C2, à partir de Firefox, saisissez l'URL suivante : `**<bot_ip>/vulnerable_app**`



Et téléchargez les fichiers **threaded_client.py** et **malware.php** disponibles sur GitHub.

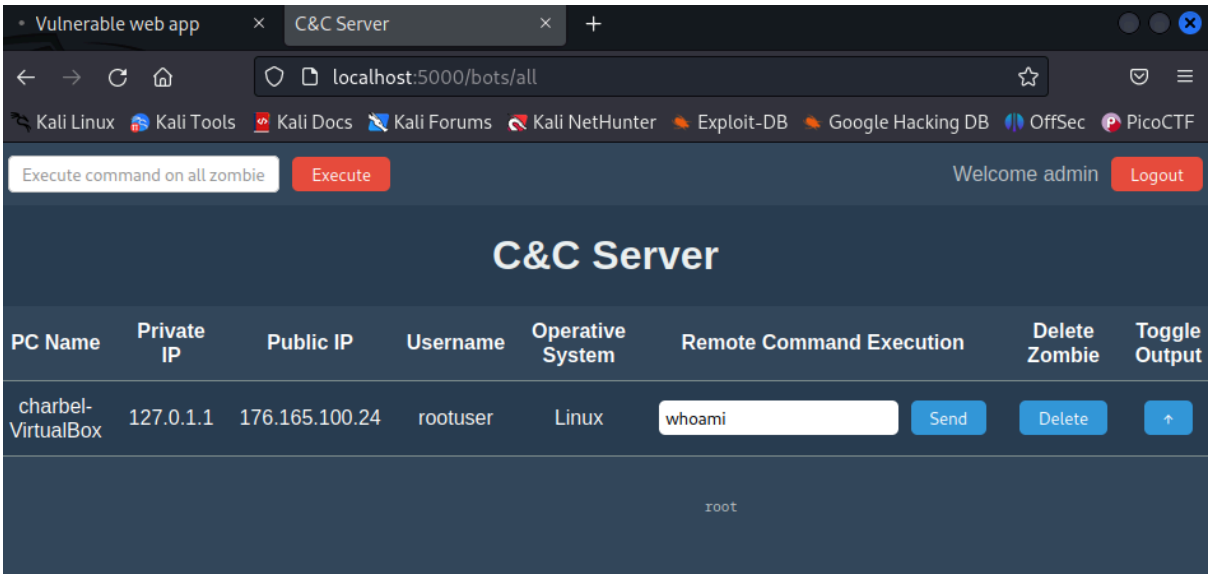
- Accédez à l'URL `**<bot_ip>/vulnerable_app/uploads/malware.php**` pour exécuter le fichier PHP qui lancera le fichier Python afin de se connecter au serveur C2.



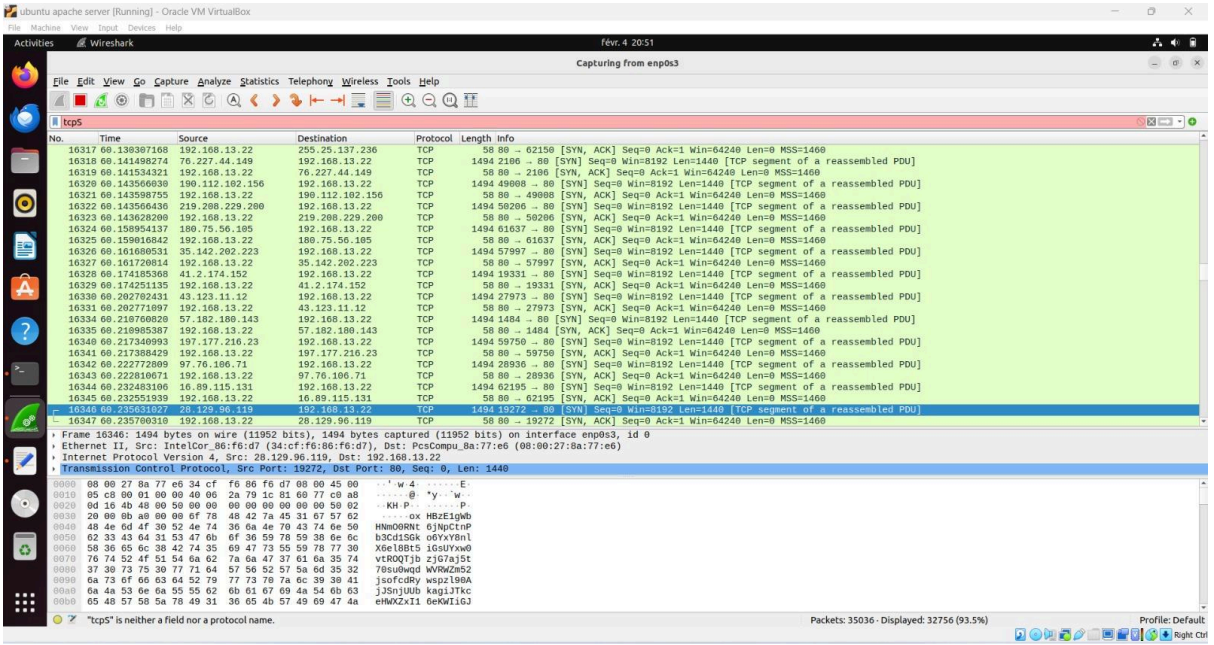


- Connectez-vous à votre C2 avec le nom d'utilisateur "admin" et le mot de passe

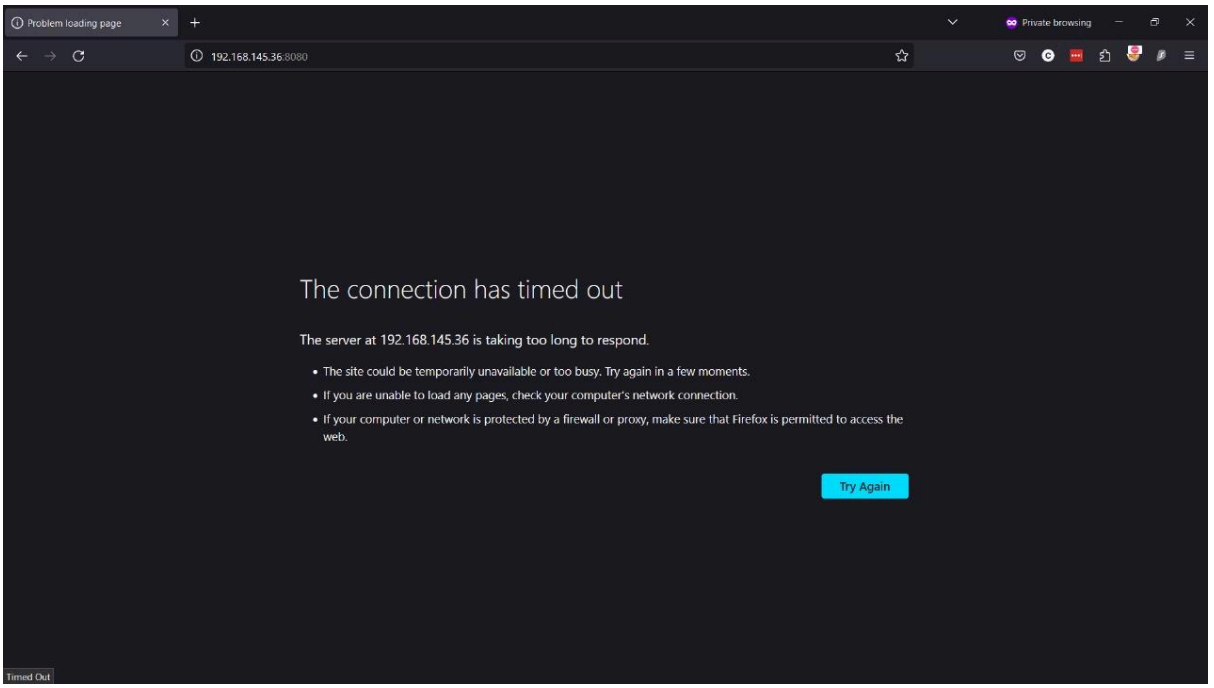
"admin" pour vérifier les connexions actives. Le serveur n'est pas dynamique en ce qui concerne la connexion ou la suppression de connexions ou l'exécution de commandes ; vous devez rafraîchir la page.



- Pour exécuter l'attaque de type SYN flooding, exécutez la commande suivante : **syn target_ip target_port number_of_packets packet_zie number_of_threads attack_duration attack_rate**
  Ce qui a fonctionné pour nous était le suivant :
  syn target_ip target_port 1000000 1440 30000 6000 10000
- Pour exécuter l'attaque de type ICMP flooding, exécutez la commande suivante : **icmp target_ip number_of_packets packet_zie number_of_threads attack_duration attack_rate**
  Ce qui a fonctionné pour nous était le suivant :
  icmp target_ip target_port 1000000 1440 30000 6000 10000
- Sur la cible, nous pouvons voir les paquets suivants provenant de l'attaque de type SYN flood :



Les adresses IP et les ports sources sont générés de manière aléatoire et nous pouvons rencontrer un serveur non réactif lorsque nous essayons de nous connecter.



8. Conclusion

En conclusion, il est essentiel de faire face aux menaces posées par les attaques SYN flood et ICMP flood pour maintenir l'intégrité et la disponibilité des services réseau. Pour les attaques SYN flood, des stratégies telles que l'augmentation de la file d'attente de backlog, le recyclage de la connexion TCP semi-ouverte la plus ancienne et la mise en œuvre des cookies SYN peuvent aider à atténuer l'impact. Cependant, il est crucial de trouver un équilibre attentif pour éviter des répercussions négatives sur les performances du système.

En ce qui concerne les attaques ICMP flood, la désactivation de la fonctionnalité ICMP sur les routeurs, les ordinateurs ou les appareils cibles offre une solution simple. Les administrateurs réseau peuvent également renforcer la sécurité en configurant les pare-feu pour bloquer les pings ICMP, en ajoutant des filtres pour détecter et rejeter les paquets suspects, en mettant en œuvre un filtrage de sortie pour les paquets usurpés, et en utilisant des logiciels de surveillance réseau pour identifier les modèles de trafic anormaux. La numérisation régulière du réseau à la recherche de ports ouverts en dehors de la ligne de base peut également renforcer la posture globale de sécurité.

La mise en œuvre d'une combinaison de ces stratégies d'atténuation, adaptée à l'environnement réseau spécifique, est cruciale pour créer une défense robuste contre d'éventuelles attaques DDoS. La vigilance constante, la surveillance proactive du réseau et une infrastructure de sécurité bien configurée sont des composants essentiels pour se prémunir contre l'évolution constante du paysage des menaces cybernétiques.

9. Références:

[1] (*Attaque DDoS SYN Flood*, n.d.)
https://www.cloudflare.com/fr-fr/learning/ddos/syn-flood-ddos-attack/

[2] (*Attaque DDoS De Type Ping (ICMP) Flood*, n.d.)
https://www.cloudflare.com/fr-fr/learning/ddos/ping-icmp-flood-ddos-attack/

Github: https://github.com/CharbelFarah057/C2-Server.git