Configuration Complète de Snort et IPTables pour la Détection et le Blocage des Attaques

Résumé du travail effectué sur Snort et IPTables

Date : 24 avril 2025

Auteur : ZALLE T. Abdel Razak

Introduction

Ce document résume l'ensemble du travail effectué pour installer, configurer et utiliser Snort et IPTables sur une machine virtuelle Ubuntu (IP : `192.168.1.92`) afin de détecter et bloquer les attaques réseau provenant d'une machine virtuelle Kali (IP : `192.168.1.90`). Le processus inclut l'installation des prérequis (DAQ et Snort), la configuration des règles, les tests de détection, et la mise en place d'IPTables pour journaliser et bloquer le trafic malveillant. Les alertes de Snort sont affichées dans la console.

1 Installation des Prérequis

1.1 Mise à jour du système

Avant d'installer les outils, le système Ubuntu a été mis à jour pour s'assurer que tous les paquets sont à jour.

```
sudo apt update
sudo apt upgrade -y
```

- `sudo apt update` : Met à jour la liste des paquets disponibles.
- `sudo apt upgrade -y` : Met à jour tous les paquets installés.

1.2 Installation des dépendances

Les dépendances nécessaires pour DAQ et Snort ont été installées.

```
sudo apt install -y build-essential libpcre3-dev libdumbnet-dev
   bison flex zlib1g-dev liblzma-dev openssl libssl-dev
   pkg-config libhwloc-dev cmake libnetfilter-queue-dev
   libnfnetlink-dev libmnl-dev
```

- `build-essential` : Inclut les outils de compilation (gcc, make, etc.).
- `libpcre3-dev`, `libdumbnet-dev`, etc. : Bibliothèques nécessaires pour DAQ et Snort.

- `bison`, `flex` : Outils pour générer des analyseurs syntaxiques (utilisés lors de la compilation).

1.3 Installation de DAQ

DAQ (Data Acquisition Library) est une dépendance clé pour Snort, permettant la capture de paquets réseau.

- Téléchargement de DAQ (version 2.0.7, par exemple) :

```
wget https://www.snort.org/downloads/snort/daq-2.0.7.tar.gz
tar -xvzf daq-2.0.7.tar.gz
cd daq-2.0.7
```

  - `wget` : Télécharge l'archive DAQ.
  - `tar -xvzf` : Extrait l'archive.
  - `cd daq-2.0.7` : Accède au répertoire extrait.

- Compilation et installation de DAQ :

```
./configure
make
sudo make install
```

  - `./configure` : Configure le processus de compilation.
  - `make` : Compile DAQ.
  - `sudo make install` : Installe DAQ sur le système.

1.4 Installation de Snort

Snort a été installé à partir des sources pour s'assurer d'avoir la dernière version compatible.

- Téléchargement de Snort (version 2.9.20, par exemple) :

```
wget https://www.snort.org/downloads/snort/snort-2.9.20.tar.gz
tar -xvzf snort-2.9.20.tar.gz
cd snort-2.9.20
```

- Compilation et installation de Snort :

```
./configure --enable-sourcefire
make
sudo make install
```

  - `-enable-sourcefire` : Active des fonctionnalités avancées de Snort.
  - `make` et `sudo make install` : Compile et installe Snort.

- Mise à jour des bibliothèques partagées :

```
sudo ldconfig
```

  - `ldconfig` : Met à jour les liens des bibliothèques partagées.

1.5 Création des répertoires et configuration de Snort

Les répertoires nécessaires pour Snort ont été créés, et les fichiers de configuration ont été mis en place.

- Création des répertoires :

```
sudo mkdir -p /etc/snort
sudo mkdir -p /etc/snort/rules
sudo mkdir -p /var/log/snort
```

- Copie des fichiers de configuration par défaut :

```
sudo cp ~/snort-2.9.20/etc/*.conf* /etc/snort
sudo cp ~/snort-2.9.20/etc/*.map /etc/snort
```

- Création d'un fichier de règles vide :

```
sudo touch /etc/snort/rules/local.rules
```

1.6 Installation des règles communautaires

Les règles communautaires de Snort ont été téléchargées pour enrichir la détection.

- Téléchargement des règles communautaires :

```
wget
   https://www.snort.org/downloads/community/community-rules.tar.gz
tar -xvzf community-rules.tar.gz
sudo cp community-rules/*.rules /etc/snort/rules/
```

- Mise à jour de `snort.conf` pour inclure les règles communautaires :

```
sudo nano /etc/snort/snort.conf
```

  Ajout ou vérification de la ligne suivante dans `snort.conf` :

```
include $RULE_PATH/community.rules
```

1.7 Configuration de `snort.conf`

Le fichier `snort.conf` a été modifié pour définir les variables réseau et les chemins.

```
sudo nano /etc/snort/snort.conf
```

Modifications effectuées :

- Définition de $\texttt{HOME}_{NET}$ et $\texttt{EXTERNAL}_{NET}$:

2 Configuration de Snort pour la Détection

2.1 Vérification de l'exécution de Snort

Snort a été lancé pour surveiller le trafic réseau sur l'interface enp0s3 et afficher les alertes dans la console.

```
sudo snort -i enp0s3 -A console -c /etc/snort/snort.conf
```

- -i enp0s3 : Spécifie l'interface réseau à surveiller.
- -A console : Affiche les alertes dans la console.
- -c /etc/snort/snort.conf : Utilise le fichier de configuration.

Vérification que Snort est en cours d'exécution :

```
ps aux | grep snort
```

Sortie attendue :

```
root     20854  1.3 29.3 700380 597500 pts/0   Sl+  05:40
   0:04 snort -i enp0s3 -A console -c /etc/snort/snort.conf
```

2.2 Configuration des règles personnalisées

Des règles personnalisées ont été ajoutées dans /etc/snort/rules/local.rules.

```
sudo nano /etc/snort/rules/local.rules
```

Contenu ajouté :

```
alert tcp any any -> $HOME_NET any (msg:"SYN Scan Detected";
   flags:S; sid:1000001; rev:1;)
alert tcp any any -> $HOME_NET 12345 (msg:"Connection attempt
   to port 12345"; sid:1000002; rev:1;)
alert tcp any any -> $HOME_NET 80 (msg:"Potential SQL
   Injection Attempt"; content:"OR 1=1"; http_uri;
   sid:1000003; rev:1;)
alert tcp $HOME_NET 12345 -> any any (msg:"RST Response
   Detected"; flags:R; sid:1000004; rev:1;)
```

- flags:S : Détecte les paquets SYN (utilisés dans les scans).
- port 12345 : Cible les connexions au port 12345.
- content:"OR 1=1" : Détecte les tentatives d'injection SQL.
- flags:R : Détecte les réponses RST (port fermé).

Vérification des règles :

```
cat /etc/snort/rules/local.rules
```

2.3 Tests de détection avec Snort

Des scans et attaques ont été lancés depuis Kali pour tester Snort.

Scan SYN sur le port 12345 :

```
nmap -sS -sV -Pn 192.168.1.92 -p 12345
```

Alerte Snort attendue :

```
[**] [1:1000001:1] SYN Scan Detected [**]
[Priority: 0]
{TCP} 192.168.1.90:port -> 192.168.1.92:12345
```

Tentative d'injection SQL sur le port 80 :

```
curl "http://192.168.1.92/?id=1%20OR%201=1"
```

Alerte Snort attendue :

```
[**] [1:1000003:1] Potential SQL Injection Attempt [**]
[Priority: 0]
{TCP} 192.168.1.90:port -> 192.168.1.92:80
```

2.4 Ouverture d'un port pour tester

Le port 12345 a été ouvert pour tester les connexions sur un port ouvert.

```
sudo apt install -y netcat
nc -l 12345 &
```

Vérification que le port est ouvert :

```
sudo netstat -tuln | grep 12345
```

Sortie attendue :

```
tcp        0      0 0.0.0.0:12345           0.0.0.0:*
                  LISTEN
```

3 Configuration d'IPTables

3.1 Vérification des règles actuelles

Les règles IPTables ont été vérifiées avant modification :

```
sudo iptables -L -v -n
```

Sortie :

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
                   destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
                   destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source
                   destination
```

Sauvegarde des règles actuelles :

```
sudo iptables-save > iptables-backup.txt
```

3.2 Création d'une chaîne personnalisée

Une chaîne $\texttt{SNORT}_BLOCKatcrepourgrerletraficmalveillant$:

```
sudo iptables -N SNORT_BLOCK
sudo iptables -A INPUT -j SNORT_BLOCK
```

— -N $\texttt{SNORT}_BLOCK$: $Creunenouvellechane.$-A INPUT -j $\texttt{SNORT}_BLOCK$: $Redirigetoutletraficentran$

3.3 Journalisation et blocage du trafic

Des règles ont été ajoutées pour journaliser et bloquer le trafic TCP provenant de Kali (192.168.1.90) :

Journalisation :

```
sudo iptables -A SNORT_BLOCK -p tcp -s 192.168.1.90 -m limit
   --limit 3/min -j LOG --log-prefix "SNORT_BLOCK: "
   --log-level 4
```

- -p tcp : Cible le trafic TCP.
- -s 192.168.1.90 : Cible l'IP source de Kali.
- -m limit -limit 3/min : Limite la journalisation à 3 messages par minute.
- -j LOG : Journalise le trafic.

Blocage :

```
sudo iptables -A SNORT_BLOCK -p tcp -s 192.168.1.90 -j DROP
```

- -j DROP : Bloque le trafic sans réponse.

3.4 Journalisation et blocage des injections SQL

Des règles spécifiques ont été ajoutées pour les tentatives d'injection SQL sur le port 80 :

Journalisation :

```
sudo iptables -A SNORT_BLOCK -p tcp -s 192.168.1.90 --dport
   80 -m string --string "OR 1=1" --algo bm -m limit --limit
   3/min -j LOG --log-prefix "SNORT_SQL_INJECT: " --log-level
   4
```

- -dport 80 : Cible le port 80 (HTTP).
- -m string -string "OR 1=1" : Recherche la chaîne "OR 1=1".
- -algo bm : Utilise l'algorithme Boyer-Moore.

Blocage :

```
sudo iptables -A SNORT_BLOCK -p tcp -s 192.168.1.90 --dport
   80 -m string --string "OR 1=1" --algo bm -j DROP
```

3.5 Vérification des logs IPTables

Les logs ont été vérifiés pour confirmer que le trafic est journalisé :

```
sudo cat /var/log/kern.log | grep SNORT_BLOCK
sudo cat /var/log/kern.log | grep SNORT_SQL_INJECT
```

3.6 Sauvegarde des règles IPTables

Les règles ont été sauvegardées pour persister après un redémarrage :

```
sudo iptables-save > /etc/iptables/rules.v4
sudo apt install -y iptables-persistent
```

- iptables-persistent : Charge automatiquement les règles au démarrage.

4 Conclusion

Le travail a couvert l'installation de DAQ et Snort, la configuration des règles communautaires et personnalisées, la détection des scans et attaques (SYN scans, injections SQL), et la mise en place d'IPTables pour journaliser et bloquer le trafic malveillant provenant de 192.168.1.90. Les alertes de Snort sont affichées dans la console.

5 Prochaines étapes

- Tester des attaques plus complexes en installant une application vulnérable comme DVWA sur le serveur Apache.
- Automatiser le blocage en extrayant les adresses IP des alertes Snort pour les ajouter dynamiquement à IPTables.