INSTITUT NATIONAL SUPÉRIEUR D'INFORMATIQUE

Configuration d'un système de détection et de prevention avec Snort

MENTION : Informatique

PARCOURS : ARSB

ANNÉE UNIVERSITAIRE : Licence 2 – 2024–2025

RÉALISÉ PAR :
LAHINIRIKO Mara Sylvain
MBOLANIRINA Stephano Kevin

Introduction

Dans le domaine de la sécurité réseau, **Snort** est un outil incontournable pour détecter et analyser les intrusions en temps réel. Open source et largement utilisé, il permet de surveiller le trafic réseau et de repérer toute activité suspecte grâce à des règles personnalisées.

Dans ce premier guide, d'autres vont suivre, je vais vous montrer comment installer, configurer et utiliser **Snort** pour détecter les intrusions. Nous aborderons également la création de règles personnalisées et l'analyse des alertes générées pour agir rapidement en cas d'incident.

1 C'est quoi Snort ?

Snort est un logiciel de **détection d'intrusion réseau (NIDS)** reconnu pour sa capacité à surveiller et protéger les réseaux informatiques en temps réel.

Il fonctionne selon plusieurs modes :

— **Mode sniffer** : capture le trafic réseau en temps réel.
— **Mode enregistreur de paquets** : enregistre les paquets pour analyse ultérieure.
— **Mode IDS** : compare les paquets réseau à des règles prédéfinies et génère des alertes.

2 Un peu d'histoire

Depuis sa création en 1998 par **Martin Roesch**, Snort a évolué pour devenir l'un des outils NIDS les plus utilisés. Initialement simple capteur de paquets, il a intégré des fonctions avancées avec le temps.

En 2013, **Cisco Systems** a acquis Sourcefire, la société à l'origine de Snort, renforçant son développement et son intégration dans les systèmes professionnels.

3 Installation de Snort

Nous allons installer la version 3.5.0 de Snort sur une machine Debian 12(elle fonctionne également sur une Ubuntu 24.04), cette machine est ma machine de rebonds de mon homelab.

Étape 1 : Préparation de l'environnement

Commencez par mettre à jour votre système pour vous assurer que tous les paquets sont récents et compatibles avec l'installation.

```
sudo -i
apt update && apt upgrade -y
```

Ensuite, installez les dépendances essentielles pour compiler et exécuter Snort 3.5.0 :

```
apt install -y git wget build-essential libpcap-dev libpcre3-dev \
libdumbnet-dev zlib1g-dev libluajit-5.1-dev libssl-dev cmake
   libunwind-dev \
luajit hwloc bison flex liblzma-dev openssl pkg-config libhwloc-dev
   \
cpputest libsqlite3-dev uuid-dev libcmocka-dev libnetfilter-queue-
   dev \
libmnl-dev autotools-dev libfl-dev libgoogle-perftools-dev ethtool
```

Étape 2 : Installation de DAQ (Data Acquisition Library)

DAQ est une bibliothèque indispensable pour **Snort** puisqu'elle gère la capture de paquets. DAQ permet à **Snort** de fonctionner avec plusieurs moteurs de capture comme **pcap** ou **afpacket**.

1.Clonez le dépôt DAQ :

```
git clone https://github.com/snort3/libdaq.git
```

2.Configurez, compilez et installez DAQ :

```
cd libdaq
./bootstrap
./configure
make
sudo make install
ldconfig
```

DAQ est maintenant installé et prêt à être utilisé comme bibliothèque de capture pour **Snort**.

Étape 3 : Installation de Snort 3.5.0

1.Téléchargez Snort 3.5.0 depuis le site officiel :

```
cd ..
wget https://github.com/snort3/snort3/archive/refs/tags/3.5.0.0.tar.
   gz
```

2.Extrayez l'archive :

```
tar -xzvf 3.5.0.0.tar.gz
cd snort3-3.5.0.0
```

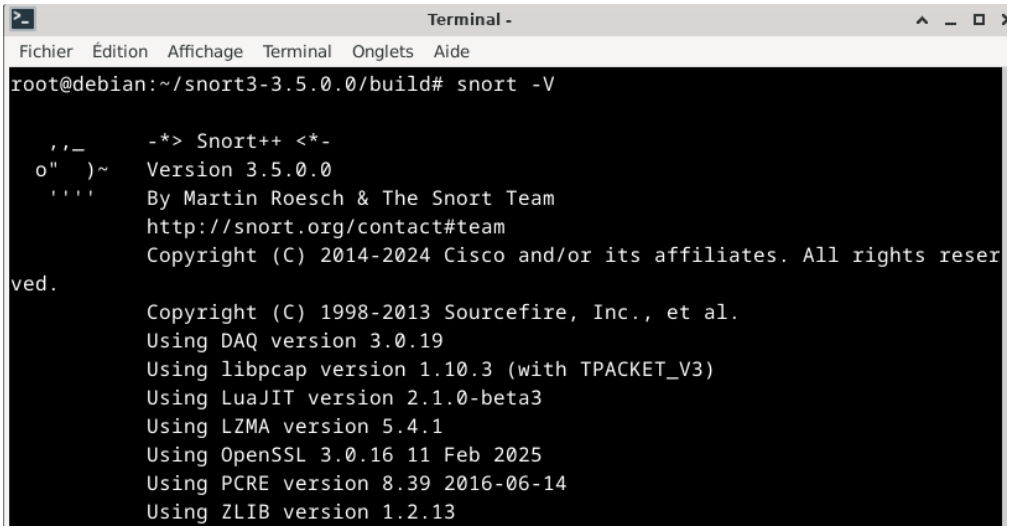
3.Configurez **Snort** en incluant les options pour **gperftools** et DAQ

```
./configure_cmake.sh --prefix=/usr/local --enable-tcmalloc
```

.Compilez et installez **Snort** :

```
cd build
make
make install
ldconfig
```

5.Vérifiez l'installation de **Snort** :



Snort est maintenant installé et prêt à être configuré et utilisé pour la détection d'intrusions réseau.

4 Configuration de l'interface réseau

Pour que Snort capture tous les paquets, l'interface réseau doit être placée en **mode promiscuous**.Cette configuration est essentielle pour un système de détection d'intrusion, car elle permet de capturer tous les paquets qui passent par l'interface, même ceux qui ne lui sont pas directement destinés. Voyons en détail ce qu'est le mode promisc et comment l'activer pour une interface réseau, ici 'enp1s0'.

Qu'est-ce que le mode PROMISC ?

En mode normal, une carte réseau traite seulement les paquets qui lui sont adressés. En mode **promiscuous**, elle capte tout le trafic transitant sur le réseau, même non destiné à sa propre adresse.

Configuration de l'interface

Sous Linux, il est simple d'activer le mode promisc pour une interface réseau en utilisant des commandes comme ip ou **ifconfig**. Voici les étapes pour activer ce mode pour l'interface 'enp1s0'. Remplacez le nom de l'interface réseau en fonction de votre configuration.

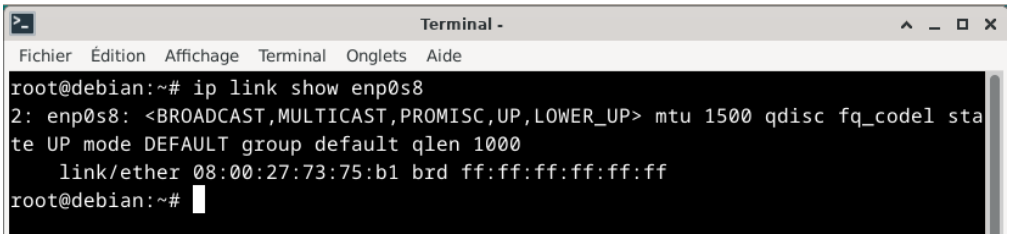
Par ailleurs, pour éviter que Snort ne tronque les paquets volumineux, nous devons désactiver certaines fonctions d'offloading au niveau de l'interface réseau. Ces fonctions sont souvent activées par défaut pour améliorer les performances du réseau, mais elles peuvent interférer avec la capture de paquets complète requise par Snort.

Étape 1 : Activer le mode promiscuous

1. Pour activer le mode promiscuous sur l'interface réseau 'enp0s8', exécutez la commande suivante :

```
sudo ip link set enp1s0 promisc on
```

Pour vérifier l'état :



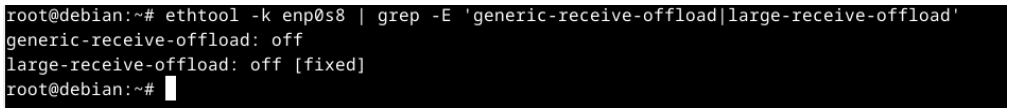
Étape 2 : Désactiver l'Interface Offloading

L'Interface **Offloading** est une série d'optimisations matérielles utilisées pour décharger le traitement des paquets vers la carte réseau, ce qui peut réduire la charge CPU. Cependant, ces options peuvent causer des problèmes avec **Snort**, notamment en tronquant les paquets de plus de 1518 octets, ce qui empêche **Snort** de capturer certaines menaces de manière complète.

Pour éviter les paquets tronqués, désactiver GRO et LRO :

```
ethtool -K enp1s0 gro off
ethtool -K enp1s0 lro off
```

Vérification :



Cela garantit que Snort capture les paquets dans leur intégralité, même au-delà de 1518 octets.

Étape 3 : Rendre les changements permanents

Le but est créer deux scripts, un pour l'IDS et un pour l'IPS mais aussi les services corréspondants.

1-Configuration de l' IDS

Voici le script qui fait le lancement du service IDS. Ce script doit être stocké dans le répertoire **/usr/local/bin/** et nommé **run_snort_ids.sh**

```
#!/bin/bash

# Vérifier les droits root
if [ "$EUID" -ne 0 ]; then
  echo "Ce script doit être exécuté en tant que root."
  exit 1
fi

echo "[*] Démarrage de Snort en mode IDS..."

# Répertoire de log
LOG_DIR="/var/log/snort"
mkdir -p "$LOG_DIR"
chown snort:snort "$LOG_DIR"

# Lancer Snort en mode IDS (écoute passive)
snort -c /usr/local/etc/snort/snort.lua \
  -R /usr/local/etc/rules/snort3-community-rules/snort3-community.rules \
  -i enp0s8 \
  -A fast \
  -s 65535 \
  -k none \
  -l "$LOG_DIR"
```

Ce script doit être éxecuté en tant que**root**.

- **LOGDIR="/var/log/snort** : c'est dans ce repertoire ques les logs de capture de paquets seront enregistré
- la dérnière section permet de lancer snort en mode IDS

Création du service IDS : Pour que le scrit soit lancer automatiquement à chaque redémarrage, nous avons de créer un service permettant de le faire. Ce service doit impérativement être stocké dans le répertoire : **/etc/sytemd/system/**, on va le nommé : **snort-ids.service**. Le contenu du fichier **service** est affiché ci-dessous

```
 GNU nano 7.2                                  /etc/systemd/system/snort-ids.service
[Unit]
Description=Snort IDS
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/run_snort_ids.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

2-Configuration de l' IPS

Voici le script qui fait le lancement du service IPS. Ce script doit être stocké dans le répertoire **/usr/local/bin/** et nommé **run_snort_ips.sh**

```
 GNU nano 7.2                                  /usr/local/bin/run_snort_ips.sh
#!/bin/bash

# Vérifier les droits root
if [ "$EUID" -ne 0 ]; then
  echo "Ce script doit être exécuté en tant que root."
  exit 1
fi

echo "[*] Démarrage de Snort en mode IPS (inline)..."

# Répertoire de log
LOG_DIR="/var/log/snort"
mkdir -p "$LOG_DIR"
chown snort:snort "$LOG_DIR"

# Réinitialiser les règles iptables et activer NFQUEUE
iptables -F
iptables -I INPUT -j NFQUEUE --queue-num 0
iptables -I OUTPUT -j NFQUEUE --queue-num 0
iptables -I FORWARD -j NFQUEUE --queue-num 0
```

```
# Réinitialiser les règles iptables et activer NFQUEUE
iptables -F
iptables -I INPUT -j NFQUEUE --queue-num 0
iptables -I OUTPUT -j NFQUEUE --queue-num 0
iptables -I FORWARD -j NFQUEUE --queue-num 0

# Lancer Snort en inline avec le DAQ nfq
snort -Q \
  --daq nfq \
  --daq-var queue=0 \
  -c /usr/local/etc/snort/snort.lua \
  -R /usr/local/etc/rules/snort3-community-rules/snort3-community.rules \
  -l "$LOG_DIR"
```

Ce script doit être éxecuté en tant que **root**.

— **LOGDIR=”/var/log/snort** : c’est dans ce repertoire ques les logs de capture de paquets seront enregistré
— les règles iptables doivent être réinitialisées et activées l’outil NFQUEU pour que snort prevenir lorsqu’il y a des attaques.
— le dernière section permet snort en mode IPS.

Création du service IDS : Pour que le script de l’IPS soit lancer automatiquement aussi à chaque redémarrage, nous allons crée un service permettant de le faire. Ce service doit impérativement être aussi stocké dans le répertoire : **/etc/sytemd/system/**, on va le nommé : **snort-ips.service**. Le contenu du fichier **service** est affiché ci-dessous

```
  GNU nano 7.2                                  /etc/systemd/system/snort-ips.service
[Unit]
Description=Snort IDS
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/run_snort_ips.sh
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Activez ensuite les services pour qu’il démarre automatiquement :

Activation de service IDS

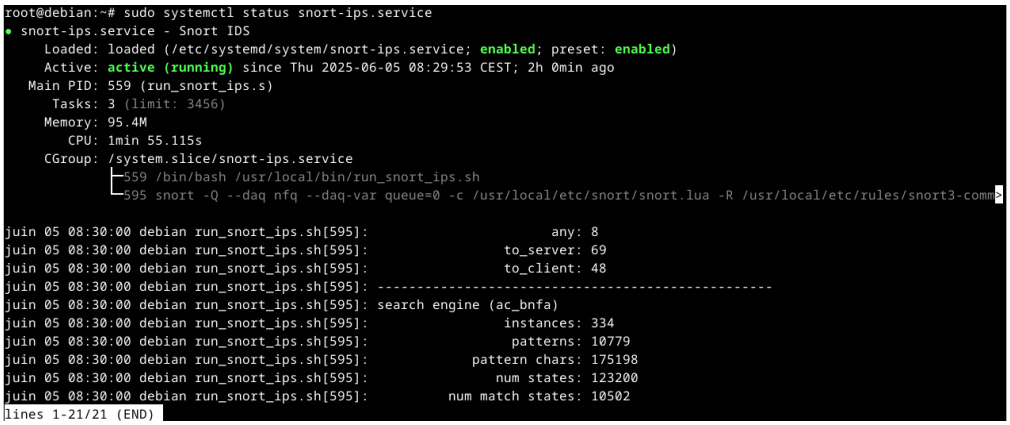
```
sudo systemctl enable snort-ids.service
sudo systemctl start snort-ids.service
```

```
root@debian:~# sudo systemctl status snort-ids.service
● snort-ids.service - Snort IDS
     Loaded: loaded (/etc/systemd/system/snort-ids.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-06-05 08:29:53 CEST; 2h 1min ago
   Main PID: 558 (run_snort_ids.s)
      Tasks: 3 (limit: 3456)
     Memory: 116.9M
        CPU: 2min 850ms
     CGroup: /system.slice/snort-ids.service
             ├─558 /bin/bash /usr/local/bin/run_snort_ids.sh
             └─563 snort -c /usr/local/etc/snort/snort.lua -R /usr/local/etc/rules/snort3-community-rules/snort3-community.ru>

juin 05 08:30:00 debian run_snort_ids.sh[563]:                     any: 8
juin 05 08:30:00 debian run_snort_ids.sh[563]:               to_server: 69
juin 05 08:30:00 debian run_snort_ids.sh[563]:               to_client: 48
juin 05 08:30:00 debian run_snort_ids.sh[563]: ------------------------------------------------
juin 05 08:30:00 debian run_snort_ids.sh[563]: search engine (ac_bnfa)
juin 05 08:30:00 debian run_snort_ids.sh[563]:                instances: 334
juin 05 08:30:00 debian run_snort_ids.sh[563]:                 patterns: 10779
juin 05 08:30:00 debian run_snort_ids.sh[563]:           pattern chars: 175198
juin 05 08:30:00 debian run_snort_ids.sh[563]:              num states: 123200
juin 05 08:30:00 debian run_snort_ids.sh[563]:         num match states: 10502
lines 1-21/21 (END)
```

Activation de service IPS

```
sudo systemctl enable snort-ips.service
sudo systemctl start snort-ips.service
```



5 Configuration de base de Snort

Après avoir installé **Snort 3.5.0** avec toutes les dépendances, la prochaine étape est la configuration initiale pour que **Snort** puisse analyser le trafic réseau efficacement. Le fichier de configuration principal de **Snort** est *snort.lua* (au lieu de snort.conf dans les versions précédentes). Nous allons voir comment configurer **Snort** pour surveiller une interface spécifique, ici **enp0s8** et comment définir les paramètres de base.

5.1 Étape 1 : Ouvrir et éditer le fichier de configuration

Le fichier de configuration par défaut pour **Snort 3** est un fichier Lua généralement situé dans /usr/local/etc/snort/snort.lua. Commencez par ouvrir ce fichier dans un éditeur de texte :

```
sudo nano /usr/local/etc/snort/snort.lua
```

5.2 Étape 2 : Configurer l'interface réseau

Dans le fichier de configuration, spécifiez l'interface réseau à surveiller. Dans ce cas, nous utiliserons **enp0s8** comme interface pour capturer le trafic.

5.2.1 Localisez la section qui définit l'interface réseau et modifiez-la pour qu'elle pointe vers enp0s8 :

```
##interface a surveiller
interface = "enp0s8"
```

5.2.2 Cette interface sera maintenant surveillée en continu par Snort, permettant de capturer tout le trafic entrant et sortant sur cette interface.

5.3 Étape 3 : Définir les réseaux surveillés et externes

Pour que Snort puisse distinguer le trafic local du trafic externe, il est essentiel de définir les variables HOME_NET et EXTERNAL_NET dans le fichier snort.lua. Cela aide Snort à générer des alertes spécifiques au réseau surveillé.

5.3.1 Dans la section des variables réseau, configurez les valeurs pour HOME_NET et EXTERNAL_NET. Par exemple :

```
HOME_NET = '192.168.63.0/24'
EXTERNAL_NET = 'any'
```

HOME_NET : définit le sous-réseau local (remplacez 192.168.63.0/24 par votre réseau local).

EXTERNAL_NET : any signifie que tout ce qui n'est pas dans HOME_NET sera considéré comme externe.

5.4 Étape 6 : Activer les règles de détection

Les règles de détection permettent à Snort d'identifier les menaces. En fonction de votre environnement, vous pouvez activer des règles spécifiques dans le fichier snort.lua :

5.4.1 Incluez les fichiers de règles nécessaires pour analyser les menaces. Si vous avez téléchargé des règles, comme les règles de base ou les règles communautaires de Snort, assurez-vous de spécifier leur emplacement dans snort.lua :

```
ips =
{
-- use this to enable decoder and inspector alerts
enable_builtin_rules = true,
include = RULE_PATH .. "/local.rules",
-- use include for rules files; be sure to set your path
-- note that rules files can include other rules files
-- (see also related path vars at the top of snort_defaults.lua)
variables = default_variables
}
```

Ici le fichier des règles se trouve sur le repertoire /usr/local/etc/rules/ c'est qui est remplacé par RULE_PATH

ET pour les règles, on a téléchargé de fichier rules dans le site officielles de snort et on a copier sur le fichier local.rules qui est notre fichier personnelle.

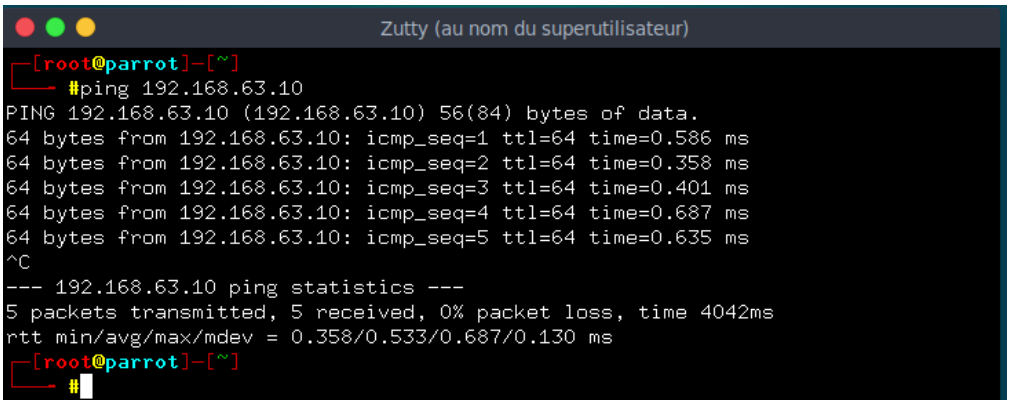
6 Test simple

Une fois que Snort est installé et configuré, il est essentiel de tester son bon fonctionnement pour s'assurer qu'il détecte bien les événements réseau. Un test simple consiste à utiliser la commande ping depuis une autre machine pour générer un type de trafic réseau courant. Cette opération permettra de vérifier que Snort capture les paquets ICMP (Internet Control Message Protocol) et génère une alerte en conséquence.

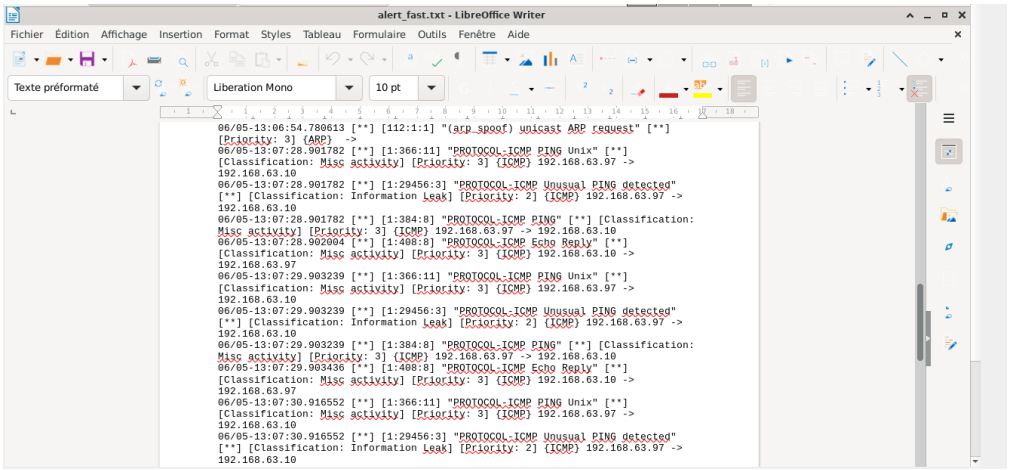
Dans cette teste, on va faire une teste de surveiller le trafic ICMP et à générer des alertes lorsqu'il détecte un ping.

On a trois machines : le premier machiene est le system de defence lequelle snort est installé , le deuxième machine est la machine cible pour simuler une attaque et le dernier machine est une machine hôte parce que le réseau que nous allons surveiller est le réseau d'une téléphone.

on va lance une ping sur le PC hôte vers le PC cible



On va verifié le fichier log depuis le machine system de défence et On voit les messages de ping avec protocole ICMP suivant dans le fichier alert_fast.txt :



Conclusion

Ce guide a présenté les étapes essentielles pour la mise en place d'un système de détection et de prévention d'intrusion (IDS/IPS) à l'aide de Snort 3.5.0 sur une distribution Debian 12. En partant de l'installation des dépendances jusqu'à la configuration du service et de la capture de trafic via NFQUEUE, nous avons mis en œuvre un système capable d'analyser les paquets qui circulent sur l'interface réseau.

Cette configuration constitue une base solide pour surveiller et protéger un réseau contre des menaces connues. Toutefois, afin d'assurer une sécurité optimale, il est recommandé d'enrichir continuellement les règles Snort, de maintenir le système à jour, et d'intégrer cette solution à un ensemble d'outils de sécurité plus large dans le cadre d'une politique de défense en profondeur.