INSTALLATION ET CONFIGURATION DE BASE D'UN IDS/IPS : CAS DE SURICATA

Présenté par :

Josias KONAN

Sommaire

INTRODUCTION

La sécurité des systèmes d'information et des réseaux est un enjeu crucial pour les entreprises et organisations, en raison de la croissance rapide des menaces cybernétiques. Les attaques informatiques, telles que les intrusions, les malwares et les attaques par déni de service (DDoS), sont de plus en plus fréquentes et sophistiquées, mettant en péril la confidentialité, l'intégrité et la disponibilité des données sensibles. Pour faire face à ces risques, il est indispensable de déployer des solutions de surveillance et de protection en temps réel, capables de détecter et de prévenir les intrusions sur les infrastructures informatiques.

Les systèmes de détection d'intrusions (IDS) et de prévention des intrusions (IPS) sont des outils essentiels pour garantir la sécurité des réseaux en permettant d'identifier et de réagir rapidement aux attaques. Ces systèmes fonctionnent principalement en analysant le trafic réseau et en comparant les paquets avec des signatures ou en détectant des anomalies qui pourraient indiquer une activité malveillante. L'outil **Suricata** se distingue parmi les solutions IDS/IPS open-source par ses performances, sa flexibilité et sa capacité à traiter des volumes de trafic réseau élevés, tout en offrant une analyse approfondie du contenu réseau.

Ce projet se concentre sur l'installation et la configuration de base de Suricata, en mettant l'accent sur les différentes étapes de déploiement et de paramétrage de l'outil pour en tirer le meilleur parti dans un environnement de production. En outre, ce projet explore les principales fonctionnalités de Suricata, telles que la détection de signatures, l'analyse en temps réel des flux réseau et la gestion des alertes. Une attention particulière est portée à la configuration des règles de sécurité et à la génération des logs, en vue de garantir une surveillance efficace et réactive des intrusions potentielles.

Ce rapport détaille ainsi l'ensemble du processus, depuis l'installation de Suricata jusqu'à la validation de son bon fonctionnement dans un réseau local, en passant par l'analyse des performances de l'outil face à des attaques simulées.

I. PREREQUIS DU PROJET

Environment:

- *2 Machine sous Ubuntu 22.04.5 LTS*
- *1 Machine sous Windows 10 Professionnel*
- *1 Machine sous Windows 11 Professionnel*
- *1 Machine sous Parrot Security 6.3.2*
- *VMWare Workstation Pro 17*

Prérequis du projet:

- *Aucun prérequis*

Adressage Réseau:

N°	HÔTE	RÔLE	ADRESSE IP	ENVIRONNEMENT
1	Wazuh	Serveur Wazuh	192.168.1.114/24	Ubuntu 22.04.5 LTS
2	Suricata	IDS / IPS	192.168.1.118/24	Ubuntu 22.04.5 LTS
3	Parrot	Machine Attaquante	192.168.1.67/24	Parrot Security
4	Client 1	Machine Cliente 1	192.168.1.65/24	Windows 11 Professionnel
5	Client 2	Machine Cliente 2	192.168.1.130/24	Windows 10 Professionnel

II. INSTALLATION ET CONFIGURATION DE BASE DU SERVEUR SURICATA

Dans la première partie de notre projet, nous allons procéder à l'installation basic de notre IPS et aux configurations de surveillance du réseau.

1. Installation de Suricata

- Nous allons nous connecter par ssh à notre serveur Suricata depuis notre ligne de commande Windows. La première chose à faire sera de mettre à jour le système Ubuntu et les paquets si cela n'est pas déjà fait, en tapant la commande « ***sudo apt update && sudo apt upgrade -y*** ». Après quoi, nous allons installer les paquets requis pour l'élaboration de notre projet. Dans un premier temps, nous allons ajouter la dernière version stable du repo de Suricata.

```
root@suricata-server:/home/adminsuricata# add-apt-repository ppa:oisf/suricata-stable
Repository: 'deb https://ppa.launchpadcontent.net/oisf/suricata-stable/ubuntu/ jammy ma
in'
Description:
Suricata IDS/IPS/NSM stable packages
https://suricata.io/
https://oisf.net/

Suricata IDS/IPS/NSM - Suricata is a high performance Intrusion Detection and Preventio
n System and Network Security Monitoring engine.

Open Source and owned by a community run non-profit foundation, the Open Information Se
curity Foundation (OISF). Suricata is developed by the OISF, its supporting vendors and
 the community.
```

- Une fois les upgrades terminées, nous allons maintenant pouvoir passer à l'installation de Suricata en exécutant cette commande : « ***apt install suricata -y*** »

```
root@suricata-server:/home/adminsuricata# apt install suricata -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libevent-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhtp2 libhyperscan5
  libluajit-5.1-2 libluajit-5.1-common liblzma-dev libnet1 libnetfilter-queue1
Suggested packages:
  liblzma-doc
The following NEW packages will be installed:
  libevent-2.1-7 libevent-pthreads-2.1-7 libhiredis0.14 libhtp2 libhyperscan5
  libluajit-5.1-2 libluajit-5.1-common liblzma-dev libnet1 libnetfilter-queue1
  suricata
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 6,426 kB of archives.
After this operation, 29.0 MB of additional disk space will be used.
```

- A présent, nous allons vérifier nous assurer que notre installation s'est bien déroulée et que notre service ***Suricata*** tourne correctement. Pour se faire, nous allons exécuter successivement les commandes suivante :

suricata --build-info
systemctl status suricata

```
root@suricata-server:/home/adminsuricata# suricata --build-info
This is Suricata version 7.0.10 RELEASE
Features: NFQ PCAP_SET_BUFF AF_PACKET HAVE_PACKET_FANOUT LIBCAP_NG LIBNET1.1 HAVE_HTP_U
RI_NORMALIZE_HOOK PCRE_JIT HAVE_NSS HTTP2_DECOMPRESSION HAVE_LUA HAVE_JA3 HAVE_JA4 HAVE
_LUAJIT HAVE_LIBJANSSON TLS TLS_C11 MAGIC RUST POPCNT64
SIMD support: SSE_2
Atomic intrinsics: 1 2 4 8 byte(s)
64-bits, Little-endian architecture
GCC version 11.4.0, C version 201112
compiled with _FORTIFY_SOURCE=2
L1 cache line size (CLS)=64
thread local storage method: _Thread_local
compiled with LibHTP v0.5.50, linked against LibHTP v0.5.50

Suricata Configuration:
  AF_PACKET support:                       yes
  AF_XDP support:                          no
  DPDK support:                            no
  eBPF support:                            no
  XDP support:
  PF_RING support:
  NFQueue support:
  NFLOG support:
  IPFW support:
  Netmap support:
  DAG enabled:
  Napatech enabled:
  WinDivert enabled:
```

```
root@suricata-server:/home/adminsuricata# systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: active (exited) since Thu 2025-03-27 17:30:35 UTC; 53s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2841 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
        CPU: 878ms

Mar 27 17:30:35 suricata-server systemd[1]: Starting LSB: Next Generation IDS/IPS...
Mar 27 17:30:35 suricata-server suricata[2841]: Starting suricata in IDS (af-packet) m>
Mar 27 17:30:35 suricata-server systemd[1]: Started LSB: Next Generation IDS/IPS.
lines 1-10/10 (END)
```

2. Configuration de base de Suricata

- Ensuite, il s'agira de paramétrer l'interface qui sera utilisée pour le monitoring, afin de contrôler le trafic réseau. Pour se faire, nous allons d'abord vérifier les détails de l'interface réseau concernée avec cette commande « ***ip addr*** »

```
root@suricata-server:/home/adminsuricata# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group defa
ult qlen 1000
    link/ether 00:0c:29:b5:93:8c brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.118/24 metric 100 brd 192.168.1.255 scope global dynamic ens33
       valid_lft 70923sec preferred_lft 70923sec
    inet6 fe80::20c:29ff:feb5:938c/64 scope link
       valid_lft forever preferred_lft forever
```

- Maintenant que nous avons pris connaissance des détails de l'interface qui sera écoutée, nous allons devoir les intégrer dans les configurations de ***Suricata.*** Alors, nous allons modifier le fichier de configuration de Suricata localisé dans « ***/etc/suricata/suricata.yaml*** » en faisant la commande « ***nano /etc/suricata/suricata.yaml*** »

```
  GNU nano 6.2                  /etc/suricata/suricata.yaml *
%YAML 1.1
---

# Suricata configuration file. In addition to the comments describing all
# options in this file, full documentation can be found at:
# https://docs.suricata.io/en/latest/configuration/suricata-yaml.html

# This configuration file generated by Suricata 7.0.10.
suricata-version: "7.0"

##
## Step 1: Inform Suricata about your network
##

vars:
  # more specific is better for alert accuracy and performance
  address-groups:
    #HOME_NET: "[192.168.0.0/16,10.0.0.0/8,172.16.0.0/12]"
    HOME_NET: "[192.168.1.118/24]"
    #HOME_NET: "[10.0.0.0/8]"
```

```
# Linux high speed capture support
af-packet:
  - interface: ens33
    # Number of receive threads. "auto" uses the number of cores
    #threads: auto
    # Default clusterid. AF_PACKET will load balance packets based on flow.
```

Après avoir fait ces modifications, pour les enregistrer, ous allons faire « ***Ctrl + W > Y > Entrer*** ».

- Nous allons maintenant redémarrer le service suricata afin d'appliquer les configuration effectuées. On va donc exécuter successivement ces commandes :

systemctl restart suricata
systemctl status suricata

```
root@suricata-server:/home/adminsuricata# systemctl restart suricata
root@suricata-server:/home/adminsuricata# systemctl status suricata
● suricata.service - LSB: Next Generation IDS/IPS
     Loaded: loaded (/etc/init.d/suricata; generated)
     Active: active (running) since Thu 2025-03-27 17:34:43 UTC; 6s ago
       Docs: man:systemd-sysv-generator(8)
    Process: 2964 ExecStart=/etc/init.d/suricata start (code=exited, status=0/SUCCESS)
      Tasks: 8 (limit: 2101)
     Memory: 36.9M
        CPU: 778ms
     CGroup: /system.slice/suricata.service
             └─2973 /usr/bin/suricata -c /etc/suricata/suricata.yaml --pidfile /var/ru>

Mar 27 17:34:43 suricata-server systemd[1]: Starting LSB: Next Generation IDS/IPS...
Mar 27 17:34:43 suricata-server suricata[2964]: Likely stale PID 2857 with /var/run/su>
Mar 27 17:34:43 suricata-server suricata[2964]: Removing stale PID file /var/run/suric>
Mar 27 17:34:43 suricata-server suricata[2964]: Starting suricata in IDS (af-packet) m>
Mar 27 17:34:43 suricata-server systemd[1]: Started LSB: Next Generation IDS/IPS.
lines 1-16/16 (END)
```

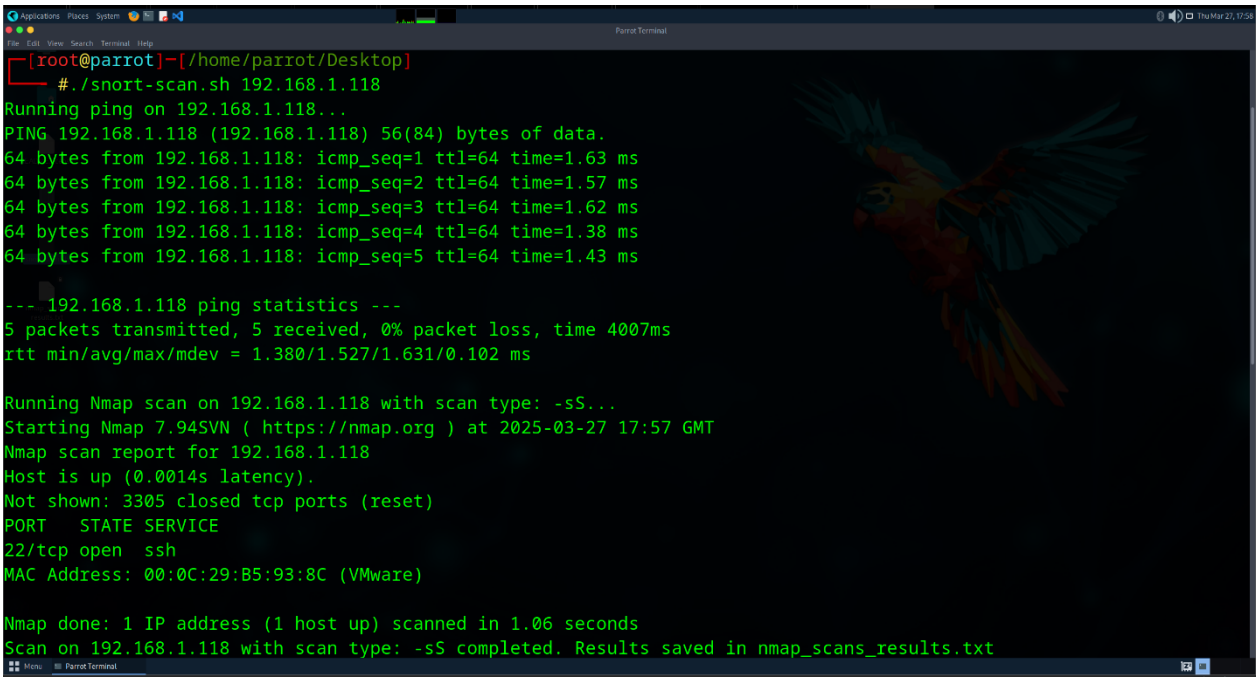
- Une fois rassuré, nous allons mettre à jour suricata pour utiliser les signatures et télécharger les règles par défaut de suricata. Nous exécuterons donc cette commande : « ***suricata-update*** ». Après cela, nous exécuterons la commande « ***tail /var/log/suricata/suricata.log*** » pour s'assurer que suricata tourne correctement

```
root@suricata-server:/home/adminsuricata# tail /var/log/suricata/suricata.log
[2973 - Suricata-Main] 2025-03-27 17:34:44 Warning: af-packet: ens33: AF_PACKET tpacket
-v3 is recommended for non-inline operation
[2973 - Suricata-Main] 2025-03-27 17:34:44 Info: runmodes: ens33: creating 2 threads
[2973 - Suricata-Main] 2025-03-27 17:34:44 Config: flow-manager: using 1 flow manager t
hreads
[2973 - Suricata-Main] 2025-03-27 17:34:44 Config: flow-manager: using 1 flow recycler
threads
[2973 - Suricata-Main] 2025-03-27 17:34:44 Info: unix-manager: unix socket '/var/run/su
ricata/suricata-command.socket'
[2974 - W#01-ens33] 2025-03-27 17:34:44 Config: af-packet: ens33: defrag enabled, setti
ng snaplen to 9216
[2974 - W#01-ens33] 2025-03-27 17:34:44 Perf: af-packet: ens33: rx ring: block_size=131
072 block_nr=74 frame_size=9296 frame_nr=1036
[2975 - W#02-ens33] 2025-03-27 17:34:44 Config: af-packet: ens33: defrag enabled, setti
ng snaplen to 9216
[2975 - W#02-ens33] 2025-03-27 17:34:44 Perf: af-packet: ens33: rx ring: block_size=131
072 block_nr=74 frame_size=9296 frame_nr=1036
[2973 - Suricata-Main] 2025-03-27 17:34:44 Notice: threads: Threads created -> W: 2 FM:
 1 FR: 1   Engine started.
root@suricata-server:/home/adminsuricata#
```

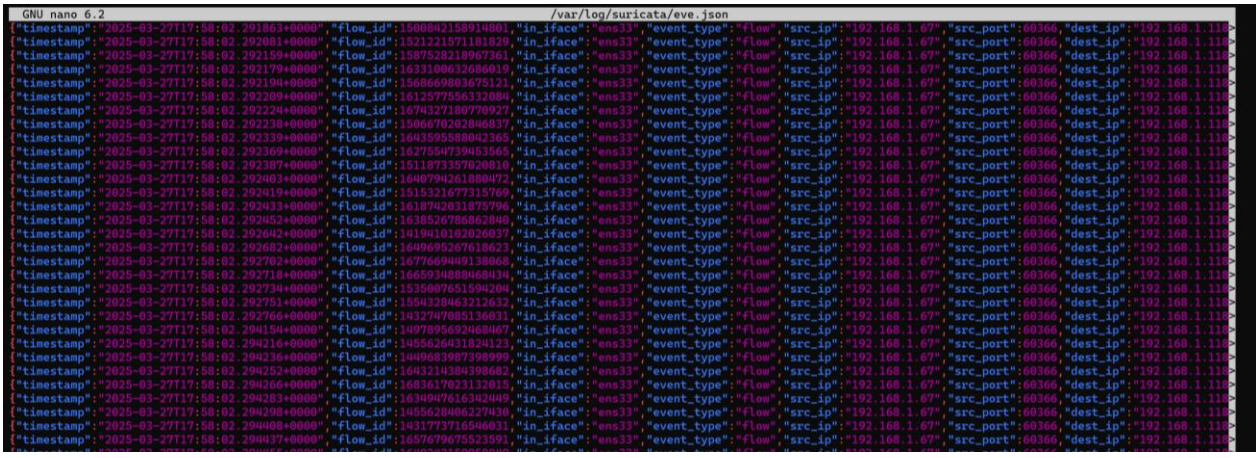
III. TEST DE SIMULATION

Pour les tests, nous allons utiliser une machine attaquante sous « ***parrot security*** », ainsi qu'un script de scan.

- Une fois sur notre machine attaquante, nous allons exécuter notre script avec cette commande : « ***./snort-scan.sh 192.168.1.118*** »



- Maintenant que nous avons exécuter le script, nous allons observer les captures au niveau de notre IPS en exécutant la commande suivante : « ***less /var/log/suricata/eve.json*** »



Nous pouvons observer ici toutes les informations sur le trafic que notre script nous a généré.

CONCLUSION

L’installation et la configuration de Suricata en tant qu'IDS/IPS ont permis de déployer une solution efficace pour la détection et la prévention des intrusions réseau. Cette configuration offre une surveillance en temps réel des menaces, assurant une sécurité proactive du réseau. Cependant, pour améliorer la gestion des alertes et la corrélation des événements, l’intégration de Suricata avec un SIEM comme **Wazuh** est une étape essentielle. Cela permettrait une centralisation des événements de sécurité pour une analyse plus approfondie. De plus, l’analyse des fichiers suspects via **Virus Total** et l'adaptation des règles de Suricata à l’aide de **Suricata Rules** renforceront la capacité de détection des menaces, en assurant une réponse rapide aux attaques. Ainsi, cette première étape ouvre la voie à une amélioration continue de la sécurité réseau.