TP Suricata

Table des matières

[TP Suricata 1](#_Toc145320848)

[Présentation 2](#_Toc145320849)

[Configuration 3](#_Toc145320850)

[Mode IDS 4](#_Toc145320851)

[Mode service 5](#_Toc145320852)

[Simulation de parc 6](#_Toc145320853)

[Manipulation hping3 7](#_Toc145320854)

[Détection de flux UDP 8](#_Toc145320855)

[Blocus des flux SSH 9](#_Toc145320856)

[Mode IPS : 9](#_Toc145320857)

[Détection DOS 11](#_Toc145320858)

[Mode IDS : 11](#_Toc145320859)

[Mode IPS : 13](#_Toc145320860)

[Conclusion 14](#_Toc145320861)

## Présentation

Suricata est une sonde N.I.D.S / N.I.P.S facile d’installation et d’utilisation.

Ce TP a pour but de montrer comment manipuler cette sonde dans un premier temps en mode IDS puis en mode IPS.

Ce TP présentera les différents fichiers de configuration de la sonde et de comment les renseigner.

## Installation

#### VM

Dans un premier temps il faut créer un VM Debian avec une interface graphique.

Avec au minimum :

6 GB de RAM

20 GB

2 interfaces réseau.

1 connexion à internet.

#### Suricata

Pour télécharger Suricata :

sudo apt-get install suricata

Vérifiez que l’installation de Suricara s’est déroulée correctement :

Suricata – V

Vous devriez avoir le résultat suivant :

« This is Suricata version \*\*\* RELEASE »

## Configuration

Le fichier de configuration globale de Suricata est le suivant :

vi /etc/suricata/suricata.yaml

Ce fichier regroupe tous les paramètres de fonctionnement de Suricata.

Dans ce fichier, l’une des premières variables que l’on peut modifier est : HOME\_NET

Cette variable va permettre de renseigner plus facilement l’IP de l’équipement local de la VM.

Il faut donc renseigner cette variable de la plage d’IP local de la VM sur laquelle se trouve Suricata.

Exemple : HOME\_NET : « [10.13.0.0/26] »

Le dossier comportant les fichiers de règles de Suricata se trouve dans l’arborescence:

*/var/lib/suricata/rules/*

Ce dossier va contenir l’ensemble des fichiers de Suricata dans lesquels seront renseignés les règles de détection ou de filtrage.

Il y a déjà normalement, un certain nombre de fichier de règles importé par Suricata. Ce sont les fichiers de base permettant de lancer Suricata avec une grande variété de règles.

Si on lance Suricata sans préciser le fichier de règle, Suricata va utiliser le fichier suivant : */var/lib/suricata/rules/suricata.rules*

Vérifier qu’il existe sinon il faut le créer avec la commande :

vi /var/lib/suricata/rules/suricata.rules

Puis dans ce fichier on va écrire une règle simple pour détecter tous les pings :

Alert icmp any any <> any any (msg : « ICMP Ping Detected”; sid:1000001; rev:1;)

Décortiquons cette règle :

Alert : L’action à réaliser par Suricata lors de la détection du paquet définit par la règle. Ici Suricata va simplement générer une alerte dans son fichier /var/log/suricata/fast.log.

Icmp : Le protocole du paquet à surveiller. On renseigne ici ICMP car on souhaite détecter du ping.

Any any <> any any : Ces paramètres désignent l’IP et le port Source, le sens du paquet, puis l’IP et le port de Destination du paquet. Ici on ne souhaite pas préciser les ports ou les IPs car on souhaite détecter n’importe quel ping. Il en est de même en ce qui concerne le sens du paquet.

Msg : Le message compris dans le log que Suricata va emettre.

Sid : L’ID de la règle.

On peut à présent sauvegarder le fichier.

## Mode IDS

Maintenant avec notre fichier de règle, on peut lancer la commande suivante pour lancer Suricata :

suricata –c /etc/suricata/suricata.yaml –i \*\* -S /var/lib/suricata/suricata.rules

* -c : pour indiquer le fichier de configuration
* -i : pour indiquer l’interface sur laquelle Suricata va écouter
* -S : pour spécifier le fichier de règles pour Suricata (et ne pas récupérer les règles indiquées dans le fichier de configuration)

Deux fichiers vont se créer :

*/var/log/suricata/eve.json* : fichier structuré et très volumineux comprenant tous les agissements sur le réseau que Suricata va détecter.

*/var/log/suricata/fast.log* : C’est le fichier qui va contenir toutes les alertes liées aux règles qu’on a indiquées à Suricata dans le fichier suricata.rules.

Pour vérifier si Suricata s’est correctement lancée, on peut ouvrir un second terminal et taper :

sudo suricata –T –c /etc/suricata/suricata.yaml

Cette commande affiche les instances de Suricata en cours d’exécution.

Si la commande indique que Suricata fonctionne correctement, on peut tester notre règle :

Toujours avec ce second terminal essayer de ping vers google.

ping 8.8.8.8

On peut ensuite vérifier dans notre fichier d’alerte :

cat /var/log/suricata/fast.log

Et on retrouve bien les alertes liées à nos pings.

Si à cette commande :

suricata –c /etc/suricata/suricata.yaml –i \*\* -S /var/lib/suricata/suricata.rules

On ajoute l’argument « -D », cela permet de lancer Suricata en Deamon et de continuer à utiliser la CLI.

## Mode service

On peut ajouter Suricata en tant que service afin de le lancer ou même voir son statut plus facilement.

Pour cela on doit créer le fichier : */etc/systemd/system/suricata.service*

Puis le compléter comme ci-dessous :

[Unit]

Description=Suricata IDPS Daemon

Wants=network.target syslog.target

After=network.target syslog.target

Restart=on-failure

[Service]

Type=forking

PIDFile=/var/run/suricata.pid

ExecStart=/usr/bin/suricata -c /etc/suricata/suricata.yaml -i enp2s0 -i enp3s0 -S /var/lib/suricata/rules/suricata.rules -D

ExecReload=/bin/kill -HUP $MAINPID

ExecStopPost=/bin/kill $MAINPID

Restart=on-failure

[Install]

WantedBy=multi-user.target

**Attention :** La valeur ExecStart correspond à la commande que le system service va lancer avec la commande « start ». Il convient donc de modifier cette commande en fonction de votre environnement (interface, commande, rules, etc)

A présent, pour activer le service :



Enfin pour le démarrer :



On peut également le redémarrer :



L’arrêter :



Ou vérifier son statut :



## Simulation de parc

A présent on va générer une nouvelle VM qu’on va utiliser pour émettre des flux entres les deux.

On va créer une autre VM Debian : « PC Utilisateur »

3 Gb de RAM

20 Gb

1 interface réseau

Il faut à présent configurer cette VM en réseaux local avec la VM Suricata avec sa seconde interface réseau.

N’importe quelle plage ip pour ce réseau local fonctionne par exemple : 192.168.10.0/24

On va à présent lancer Suricata pour qu’il surveille le trafic sur l’interface connectée à la VM « PC Utilisateur ».

Attention si on lance Suricata via le système de service, il faut modifier la ligne « ExecStart » du fichier */etc/systemd/system/suricata.service*

Réaliser plusieurs manipulations de ping entre les VMs et vérifier si Suricata parvient à détecter les pings.

A présent dans le fichier suricata.rules, écrire une règle qui va générer une alerte si Suricata détecte une connexion SSH entre la VM Suricata et la VM PC Utilisateur.

Ci-dessous un lien vers la documentation officielle de Suricata :

[6. Suricata Rules — Suricata 6.0.0 documentation](https://suricata.readthedocs.io/en/suricata-6.0.0/rules/index.html)

Essayer de faire une connexion SSH entre « PC Utilisateur » et "Suricata”.

(Il est possible qu’il soit nécessaire d’installer SSH sur les VMs ):

apt-get install ssh

Normalement cela fonctionne et une alerte est générée à cause de la règle que vous avez rédigée plus tôt.

## Manipulation hping3

On va connecter temporairement la VM “PC Utilisateur” à internet afin d’installer hping3.

hping3 est un outil réseau capable d'envoyer des paquets TCP/IP sur commande et d'afficher les réponses de la cible comme le programme ping le fait avec les réponses ICMP.

sudo apt-get update

sudo apt-get install hping3

On reconnecte “PC Utilisateur” à “Suricata” puis on relance Suricata sur la bonne interface.

On va utiliser hping3 afin de générer du flux vers notre VM avec Suricata pour voir la réaction de la sonde.

La commande suivante permet de générer du flux :

sudo hping3 <adresse IP de la machine virtuelle de Suricata> -S -p 80 -c 10

Le –S permet de préciser que ce sont des paquets TCP SYN que l’on envoie.

Le –p permet de préciser le port de destination, ici 80

Le –c permet de choisir le nombre de flux que l’on souhaite envoyer, ici 10.

On peut à présent vérifier si Suricata a détecté le trafic :

sudo tail -f /var/log/suricata/fast.log

Rien n’apparaît car aucune règle dans notre fichier suricata.rules permet d’identifier ce genre de flux. Cependant il y a un second fichier dans lequel Suricata génère des logs :

***/var/log/suricata/eve.json***

Ce fichier va lister tous les paquets que Suricata va détecter au format json. Regroupant ainsi les IPs sources et les IPs de destinations, les ports sources et les ports de destinations ainsi que le protocole du paquet.

hping3 est un outil très utile avec de nombreux paramètres. Ci-dessous un lien permettant de connaître les paramètres possibles :

[https://manpages.ubuntu.com/manpages/bionic/fr/man8/hping3.8.html#:~:text=hping3 est un outil réseau,fait avec les réponses ICMP.)](https://manpages.ubuntu.com/manpages/bionic/fr/man8/hping3.8.html#:~:text=hping3%20est%20un%20outil%20r%C3%A9seau,fait%20avec%20les%20r%C3%A9ponses%20ICMP.))

## Détection de flux UDP

On va maintenant utiliser Suricata afin de détecter les flux UDP.

Générer une règle permettant de détecter les flux UDP provenant de la machine « PC Utilisateur » avec le port 85 en source et le port 75 en destination.

Maintenant que notre règle est enregistrée dans le fichier suricata.rules, on peut relancer Suricata.

Puis sur « PC Utilisateur » avec hping3 générer le flux correspondant.

Ensuite on vérifie que Suricata détecte correctement et génère une alerte.

Si ce n’est pas le cas, il se peut que ce soit à cause d’une erreur dans la règle ou bien la commande hping3.

A présent on relance la commande hping3 mais cette fois-ci en ajoutant l’option : « -a --spoof hostname »

En théorie Suricata ne parvient plus à détecter les paquets envoyés par hping3.

Il vous faut modifier la règle Suricata pour parvenir à détecter ces paquets.

(Utiliser l’argument any de partout serait trop facile, on se rappelle que même si Suricata ne génère pas d’alerte, le fichier */var/log/suricata/eve.json* renseigne tous les paquets transmis sur Suricata et que vous avez de nombreuses option dans le fichier */etc/suricata/suricata.yaml* )

## Blocus des flux SSH

### Mode IPS :

Il se peut que l’on souhaite configurer Suricata pour agir en tant que IPS (Intervention Prevention System). C’est-à-dire que Suricata ne sera plus seulement capable de détecter les packets parcourant le réseau, mais également de bloquer, rejeter ou droper certains packets, potentiellement malveillant.

Avec un IPS on ajoute une couche de protection active au sein de son réseau, cependant cela peut également bloquer ou ralentir la production au sein de l’entreprise si l’IPS n’a pas été configuré avec soin car certains packets primordiaux pourraient être bloqués par erreur. C’est pour cela que je conseille d’être méticuleux et attentifs si vous décidez d’utiliser un IPS.

Suricata utilise NFQueue pour interagir comme IPS. NFQ est un filtreur réseau (netfilter) qui marche avec des tables ip (iptables).

Dans un premier temps, vérifier que Suricata a été configuré et installé avec NFQ.

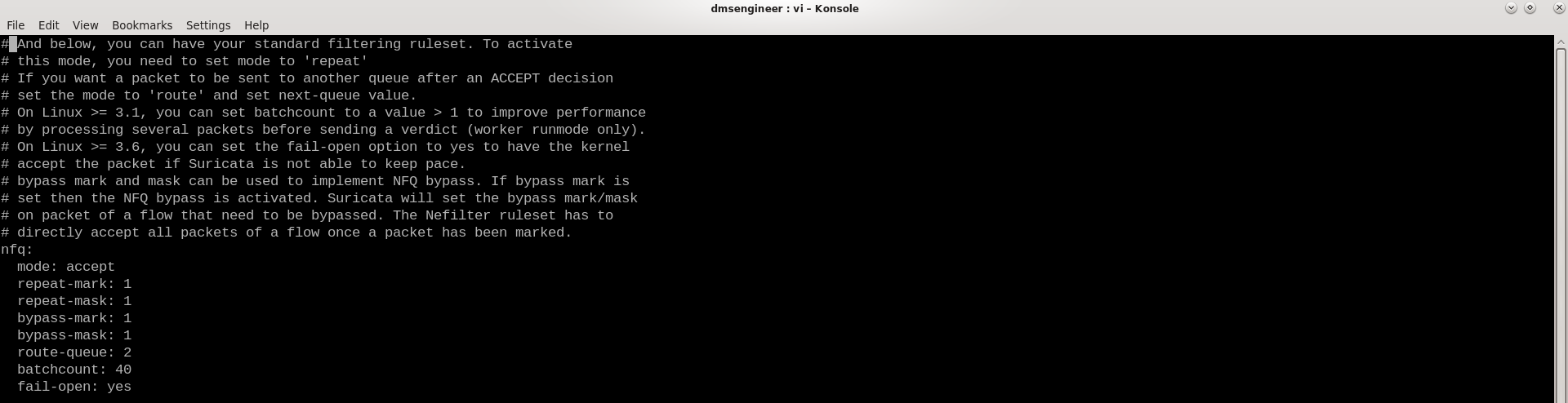
Taper la commande :

suricata –build-info

Vérifier ensuite sous la catégorie « Suricata Configuration » que le « NFQueue support » est coché « yes ».

Une fois le NFQueue validé Suricata, il faut légèrement modifier le fichier : */etc/suricata/suricata.yaml*:

Trouver la ligne « nfq » et décommenter là. Également les lignes doivent ressembler à ceci :



Pour cela il faut commencer par configurer des iptables NFQueue. Voici un exemple :

sudo iptables -I INPUT -p tcp -j NFQUEUE

sudo iptables -I OUTPUT -p tcp -j NFQUEUE

Ces commandes vont créer des iptables qui vont superviser tous les paquets entrant et sortant en TCP. NFQueue va capturer tous les paquets qu’il va détecter, il va les stocker dans une « queue » qui va ensuite les transmettre à Suricata qui va devoir les filtrer en fonction de ces règles configurées.

De base c’est la queue 0 qui sera sélectionné. On peut préciser celle que souhaitée avec l’argument : --queue-num \*. (\* Étant le nombre désiré).

**Attention** : Tant que Suricata n’a pas autorisé les paquets, NFQueue empêche leur transmission. Par exemple si on ne lance pas Suricata avec alors que ces iptables sont actives, NFQueue bloquera toutes les communications en TCP car il restera en attente de la validation de Suricata, même si ce dernier n’est pas activé.

Cependant il existe un paramètre avec NFQueue qui permet de lui indiquer de laisser les paquets si Suricata ne réponds pas : --queue-bypass***.*** Cela permets d’empêcher de bloquer tout votre réseau si jamais Suricata rencontre un problème. Cependant cela signifie également que si jamais quelqu’un parvient à éteindre Suricata, votre réseau ne sera plus filtré.

On peut également indiquer à NFQueue que l’on souhaite filtrer un protocole en particulier par exemple :

sudo iptables -I INPUT -p tcp --sport 80 -j NFQUEUE

A présent on va configurer Suricata pour fonctionner en tant que sonde IPS. Bloquant ainsi le trafic entre « Suricata » et « PC Utilisateur ».

Vérifier que la règle SSH qui alerte lorsqu’il y a une connexion SSH entre « PC Utilisateur » et « Suricata » soit activée et configurée correctement.

(Attention il ne faut pas qu’elle détecte toutes les connexions SSH, il faut préciser les IPs.)

A présent on va activer Suricata en mode IPS :

On commence par configurer les iptables pour que les paquets TCP transit vers Suricata.

Ensuite on modifie la règle Suricata pour qu’il n’alerte plus mais bloque cette connexion SSH.

Enfin on relance Suricata en précisant la queue configurée avec iptables (0 par défaut)

On tente de se connecter en SSH à « PC Utilisateur » depuis "Suricata”, on devrait avoir une erreur.

Et si l’on vérifie dans le fichier */etc/var/log/fast.log* Suricata a correctement bloqué la requête SSH.

Si Suricata ne bloque pas la connexion SSH, on vérifie si ce dernier est correctement configuré en IPS et que la règle est correcte.

## Détection DOS

### Mode IDS :

A présent nous allons simuler une situation de DOS entre nos équipements et configurer Suricata afin qu’il puisse détecter ces attaques.

*DOS ( Denial Of Service) est une attaque réseau qui empêche l'utilisation des ressources d'un serveur en surchargeant celui-ci de paquets.*

*Disposant de ressources limitées ; puissance de calcul ou mémoire, les ordinateurs administrent ces ressources afin de les répartir entre les différents programmes pour s’assurer du bon fonctionnement.*

*Cependant lorsque ces ressources atteignent leurs limites, le programme peut se bloquer ou planter, ce qui le rend indisponible. Une attaque DoS consiste à saturer ces ressources et faire en sorte qu'un serveur ou un réseau ne soit plus disponible pour ses utilisateurs légitimes, ou au moins pour faire fonctionner plus lentement.*

Dans le fichier de règles, ajoutez ces deux règles :

alert tcp $EXTERNAL\_NET any -> $HOME\_NET any (msg:"LOCAL DOS SYN packet flood inbound, Potential DOS"; flow:to\_server; flags: S,12; threshold: type both, track by\_dst, count 5000, seconds 5; classtype:misc-activity; sid:5;)

alert tcp $HOME\_NET any -> $EXTERNAL\_NET any (msg:"LOCAL DOS SYN packet flood outbound, Potential DOS"; flow:to\_server; flags: S,12; threshold: type both, track by\_dst, count 5000, seconds 5; classtype:misc-activity; sid:6;)

* flags contient S pour SYN paquet.
* threshold paramètre qui comporte le minimum de message qu’il doit recevoir avant d’envoyer une alerte

(Si les variables $EXTERNAL\_NET et $HOME\_NET ne sont pas renseignés dans le fichier */etc/suricata.con*, on peut les remplacer par IPs de “Suricata” et “PC utilisateur”).

Ces règles permettent de générer une alerte lorsque Suricata va détecter 5000 paquets SYN en moins de 5 secondes.

On relance Suricata.

Sur “PC Utilisateur”, lancez la commande :

sudo hping3 -S —flood -V -p 80 <IP “Suricata”>

* **S**: Spécifie qu’on utilise des SYN paquets.
* **–flood**: on ignore les réponses et on envoie le plus de messages possibles.
* **V**: Verbeux
* **p 80**: le port de destination

Normalement si on retourne sur “Suricata”, elle va cesser de fonctionner correctement et sera impossible à manipuler.

On peut maintenant arrêter l’attaque DDOS en retournant sur “PC Utilisateur” et un CTRL + C

On vérifie tout de même que Suricata a correctement détecté ces flux.

Si jamais rien ne se passe, cela peut-être dû aux iptables.

On peut effacer les iptables avec la commande iptables –flush.

### Mode IPS :

Maintenant modifier les règles afin que Suricata bloque ces flux.

Configurer également correctement iptables puis relancez Suricata en mode IPS.

Relancer hping3 via “PC Utilisateur” et vérifier si “Suricata” reste stable.

En toute logique ces flux seront bloqués, empêchant ainsi le mal fonctionnement de la VM. Suricata loggera les flux bloqués dans le fichier */var/log/suricata/fast.log.*

# Conclusion

Le TP est ainsi terminé. Ce permet de comprendre et de se familiariser avec Suricata et d’avoir une base solide dans le monde des sondes IDS et IPS.