

SNORT

Travaux Pratiques — Guide de Laboratoire

3 Labs Progressifs : Installation • Règles • Simulation d'Attaques

Echahbouni Issam • Mohamed Reda Qiyaoui • Programme CASI 2025–2026

Vue d'Ensemble des Travaux Pratiques :

Lab	Intitulé	Objectif principal
TP 1	Installation & Configuration	Installer Snort 3 et valider la configuration
TP 2	Écriture de Règles Personnalisées	Créer des règles de détection locales
TP 3	Simulation d'Attaques & Analyse	Simuler des attaques et analyser les alertes

NOTE	Chaque TP est indépendant mais complémentaire. Il est recommandé de les réaliser dans l'ordre pour progresser de manière optimale.
------	--

**LAB
1/3**

TP 1 · Installation & Configuration de Snort :

Étapes d'installation :

1. Installation de Snort 3 :

```
sudo apt update && sudo apt install snort -y
```

2. Vérifier la version installée :

```
snort --version
```

⇒ Vous devriez voir: Snort version 3.x.

3. Paramètres clés :

- Ouvrez le fichier de configuration suivant :

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

- Définis le Réseau :

HOME NET = 'IP adresse de votre Réseau >/Préfixe'

```
-- 1. configure defaults

-- HOME_NET and EXTERNAL_NET must be set now
-- setup the network addresses you are protecting
HOME_NET = '192.168.0.0/24'
-- set up the external network addresses.
```

- Indiquer le chemin des règles :

```
include = ["/etc/snort/rules/local.rules"]
```

```
-- 5. configure detection group default qlen 1000
-- linktrained 22.0.0.0/24 brd 22.0.0.0
-- Trash inet 192.168.17.129/24 brd 192.168.17.255
references = default_references,
classifications = default_classifications,
ips = {
    -- use this to enable decoder and inspector alerts
    --enable_builtin_rules = true,
    -- default via 192.168.17.2 dev eth0 proto dhcp
    -- use include for rules files; be sure to set your path local scope
    -- note that rules files can include other rules files
    -- (see also related path vars at the top of snort_defaults.lua)
variables = default_variables,
include = [[/etc/snort/rules/local.rules]]
}
```

❖ Tapez : **'Ctrl + O' → 'ENTRER' → 'Ctrl + X'**



 Vous devez insérer une virgule après la commande variable.

4. Tester la configuration :

```
sudo snort -c /etc/snort/snort.lua --daq-dir /usr/lib/daq
```

```
Unable to open directory "/usr/lib/daq"  
pcap DAQ configured to passive.  
  
Snort successfully validated the configuration (with 0 warnings).  
o")~ Snort exiting
```

5. Lancer en mode IDS :

```
sudo snort -c /etc/snort/snort.lua -i eth0 -A alert_fast
```



Si Snort démarre sans erreur et affiche les statistiques réseau, l'installation est réussie.

LAB
2/3

TP 2 · Écriture de Règles Personnalisées et simulation des attaques

🔗 **Objectif :** Utiliser le mode IDS et Rédiger des règles de détection personnalisées

✓ **Règles à copier :**

- Ouvrez :

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

❖ **Détecter tout Ping entrant :**

```
alert icmp any any -> $HOME_NET any (msg:"Ping ICMP détecté"; itype:8; sid:9000001; rev:1;)
```

❖ **Détecter connexions Telnet :**

```
alert tcp any any -> $HOME_NET 23 (msg:"Tentative Telnet"; sid:9000002; rev:1;)
```

❖ **Détecter Brute Force SSH avec Hydra :**

- Vous devez activer le SSH

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"Possible SSH Brute Force Attempt"; flags:S; detection_filter:track by_src, count 5, seconds 60; sid:1000005; rev:1;)
```

❖ **Détecter le scan Nmap :**

```
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Nmap SYN Scan Detected"; flags:S; threshold:type threshold, track by_src, count 20, seconds 10; sid:1000002; rev:1;)
```

- Lancer Snort avec vos règles :

```
sudo snort -c /etc/snort/snort.lua -i eth0 -A alert_fast -l /var/log/snort/
```

✓ **Attaques à simuler :****1. Nmap SYN Scan :**

```
nmap -sS [IP ADDRESS DU VICTIME]
```

⇒ Détecte les ports ouverts en envoyant des paquets TCP SYN.

2. Ping Flood :

```
ping -f [IP ADDRESS DU VICTIME]
```

3. Brute Force SSH avec Hydra :

```
hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt -t 4 ssh://[IP ADDRESS DU VICTIME]
```

⇒ Tente de deviner le mot de passe SSH par force brute.

4. Scan de versions :

```
nmap -sV [IP ADDRESS DU VICTIME]
```

Identifie les services et versions en cours d'exécution.

✓ Analyse des alertes Snort :

Exemple d'une alerte générée lors du scan Nmap :

```
[*] [1:1000010:1] NMAP SYN Scan [*]  
Classification: Attempted Information Leak Priority: 2  
04/05-14:22:13.441230 192.168.1.100:45612 -> 192.168.1.10:22  
TCP TTL:64 TOS:0x0 ID:44234 IpLen:20 DgmLen:44 DF  
***S*** Seq: 0x1A2B3C4D Ack: 0x0 Win: 0x1000 TcpLen: 24
```

Chaque alerte contient :

- La règle déclenchée (SID + message)
- La classification et la priorité
- L'horodatage précis de l'événement
- Les adresses IP source et destination avec ports
- Les flags TCP et informations du paquet



Analysez les logs avec Wireshark : ouvrez `/var/log/snort/*.pcap` pour une vue graphique complète du trafic capturé.

LAB
3/3

TP 3 · Règles IPS Actives — drop & reject

🕒 **Objectif** : Passer du mode IDS passif au mode IPS actif avec des règles drop et reject, puis vérifier le blocage en temps réel.

✓ **Prérequis — Activer le mode IPS :**

Sur une machine avec une seule interface réseau, Snort doit être lancé en mode inline avec le flag -Q et utiliser le module NFQ pour intercepter le trafic.

1. Rediriger le trafic vers Snort avec iptables :

Exécutez ces commandes pour envoyer le trafic réseau dans la file d'attente numéro 0 :

```
sudo iptables -I INPUT -j NFQUEUE --queue-num 0
sudo iptables -I OUTPUT -j NFQUEUE --queue-num 0
sudo iptables -I FORWARD -j NFQUEUE --queue-num 0
```

2. Lancer Snort en mode inline avec le DAQ NFQ :

```
sudo snort -c /etc/snort/snort.lua -Q \
--daq nfq --daq-var queue=0 -A alert_fast
```

Avertissement

Le flag -Q active le mode inline. Sans lui, les règles drop ne bloquent rien. Attention : en modifiant iptables, votre VM perdra son accès réseau normal si Snort n'est pas en cours d'exécution.

❖ **Bloquer tout ping (drop) :**

drop bloque silencieusement. Ajoutez la règle dans local.rules, puis testez depuis la VM attaquante :

```
# -- Règle : local.rules --
drop icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Ping bloqué"; itype:8; sid:9000010; rev:1;)

# -- Test : VM attaquante (doit afficher 100% packet loss) --
ping -c 4 [IP ADDRESS DU VICTIME]
```

❖ **Rejeter Telnet (reject) :**

reject bloque et renvoie un TCP RST à l'expéditeur :

```
# -- Règle : local.rules --
reject tcp $EXTERNAL_NET any -> $HOME_NET 23 (msg:"Telnet rejeté"; flow:to_server; sid:9000011; rev:1;)

# -- Test : VM attaquante (connexion refusée immédiatement) --
telnet [IP ADDRESS DU VICTIME]
```

❖ **Bloquer scan Nmap SYN :**

```
# -- Règle : local.rules --
drop tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Nmap SYN bloqué"; flags:S; flow:stateless; detection_filter: track by_src, count 5, seconds 2; sid:9000012; rev:2;)

# -- Test : VM attaquante --
nmap -sS [IP ADDRESS DU VICTIME]
```

❖ **Bloquer injection SQL HTTP avec PCRE :**

```
# -- Règle : local.rules --
drop tcp $EXTERNAL_NET any -> $HTTP_SERVERS 80 (msg:"SQLi bloquée"; flow:established,to_server; http_uri;
content:"UNION", nocase; pcre:"/UNION.{1,20}SELECT/i"; sid:9000013; rev:2;)

# -- VM victime : démarrer un serveur HTTP --
sudo python3 -m http.server 80

# -- Test : VM attaquante --
curl 'http://[IP ADDRESS DU VICTIME]/page?id=1+UNION+SELECT+1,2,3--'
```

❖ **Nettoyage Post-TP (Obligatoire) :**

Une fois vos tests terminés pour ce lab, videz les règles de pare-feu pour restaurer la connectivité réseau de votre VM :

```
sudo iptables -F
```

✓ **IDS vs IPS — Récapitulatif :**

Critère	IDS (alert/log)	IPS (drop/reject)
Action	Génère une alerte	Bloque le paquet
Impact réseau	Aucun (passif)	Interrompt la connexion
Faux positifs	Peu risqués	Peuvent bloquer du légitime
Commande Snort	alert / log	drop / reject / react
Mode lancement	Sans -Q	Avec -Q (inline)

Conseil Vérifiez les alertes dans /var/log/snort/alert_fast.txt — même en mode IPS, chaque paquet bloqué est journalisé.

Bonne chance dans vos travaux pratiques !

Echahbouni Issam • Mohamed Reda • Programme CASI 2025–2026