

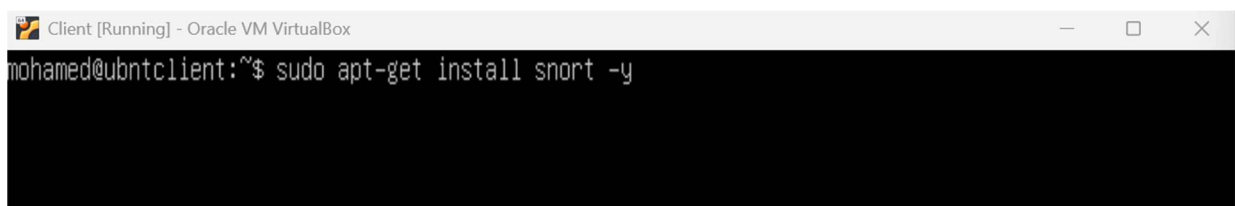
IDS/IPS : Snort

Objectif

Le but de ce laboratoire est d'installer et de configurer Snort comme système de détection d'intrusion(IDS), de tester sa configuration et de générer des alertes sur des attaques simulées

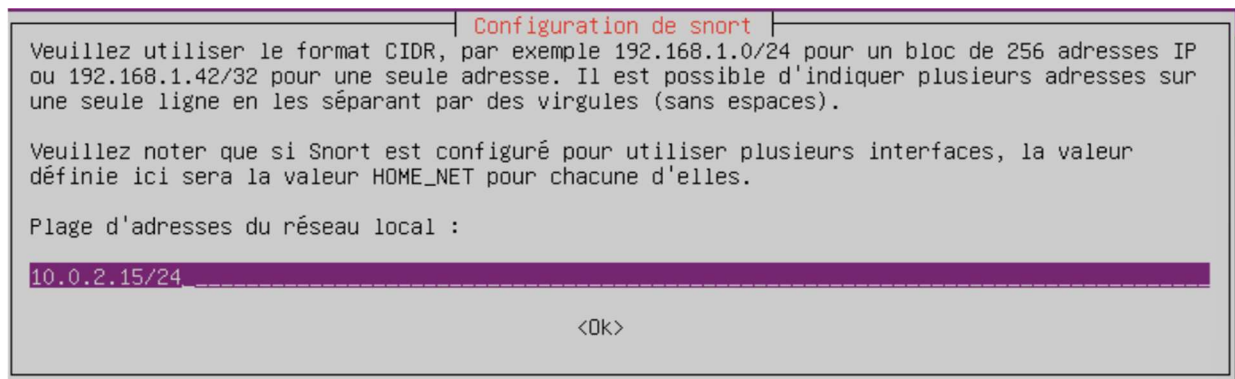
Étape 1 : Installer Snort

La première chose est de faire la mise à jour du système. Ensuite comme vous pouvez le voir sur l'image ci-dessous, nous avons exécuter la commande pour installer Snort.

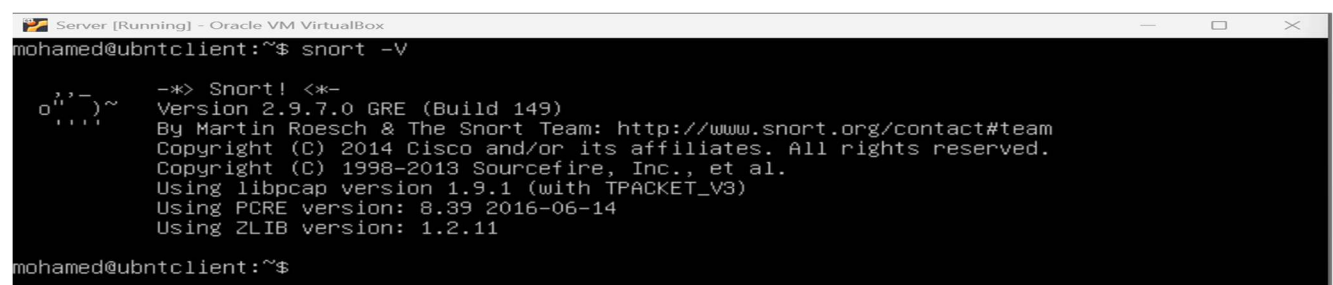


```
Client [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ sudo apt-get install snort -y
```

Ensuite on me demande l'adresse sur laquelle Snort sera appliqué, dans notre cas on va mettre l'IP de notre serveur ubuntu.



Une fois l'installation terminée, on peut utiliser la commande snort -V pour vérifier la version qui a été installé. Dans notre cas la version installée est le 2.9.7

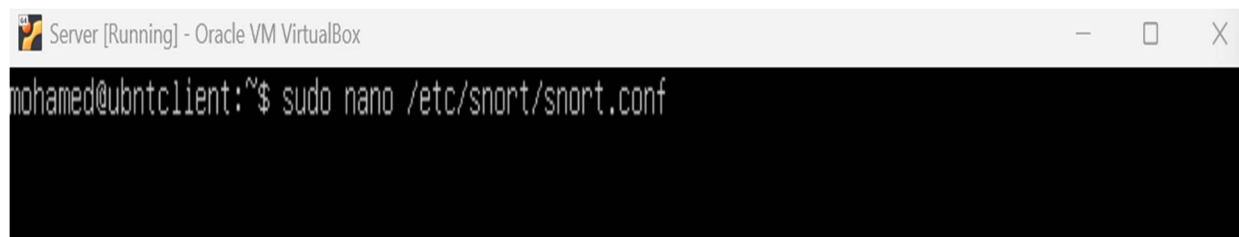


```
Server [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ snort -V
o^_^~
  |||~
  ~~~~

-*> Snort! <*-
Version 2.9.7.0 GRE (Build 149)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11
mohamed@ubntclient:~$
```

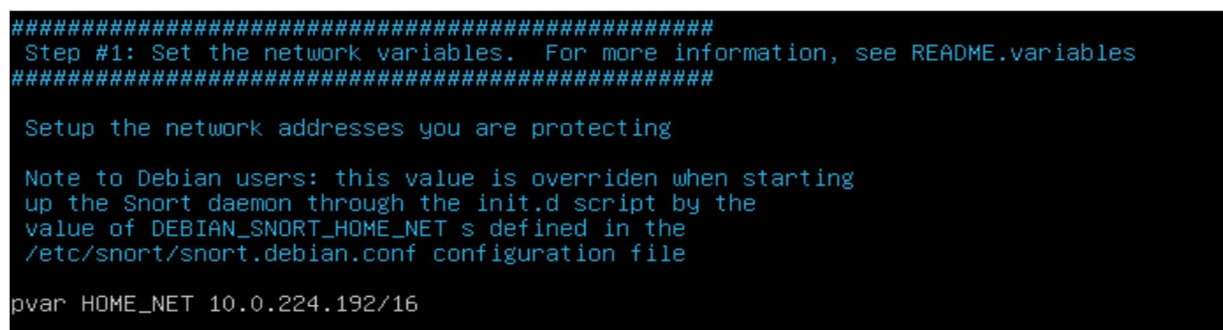
Étape 2 : Configuration Snort

Ensuite pour faire la configuration de notre IDS, on doit entrer dans le fichier de configuration qui est le snort.conf, on va l'ouvrir avec notre éditeur nano.



```
Server [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ sudo nano /etc/snort/snort.conf
```

Ensuite Sur la ligne HOME_NET, on met l'IP de notre Ubuntu, car c'est seulement lui qu'on va surveiller.



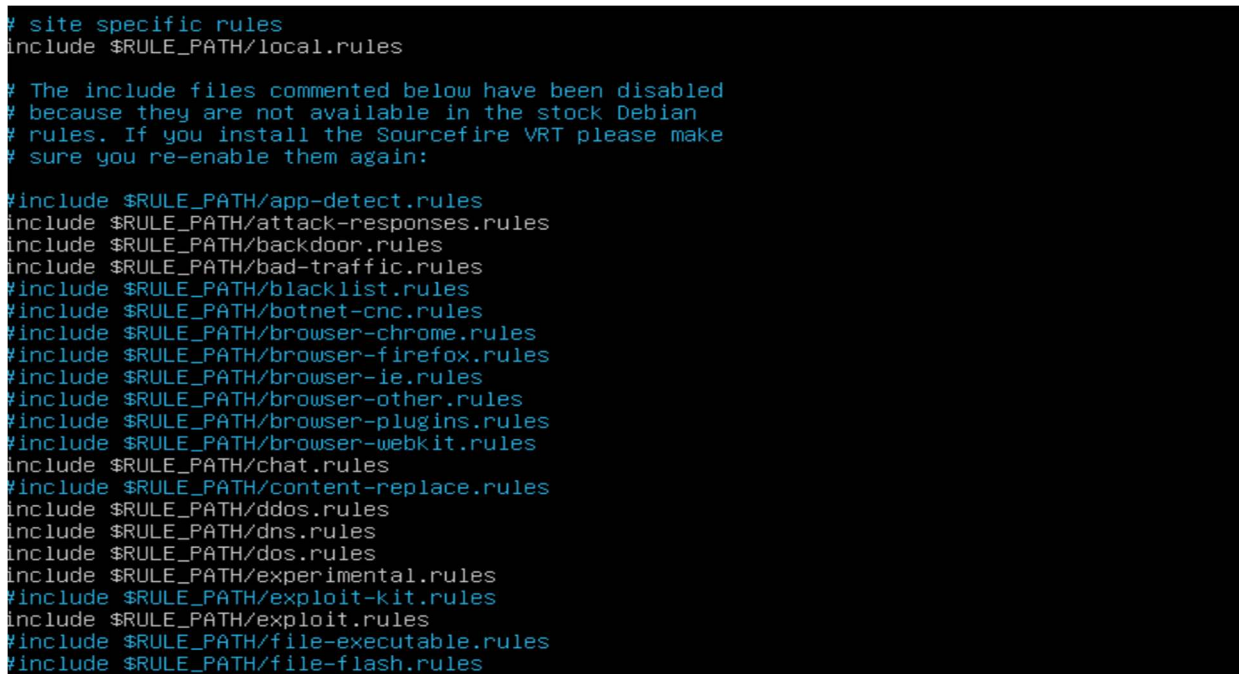
```
#####
Step #1: Set the network variables.  For more information, see README.variables
#####

Setup the network addresses you are protecting

Note to Debian users: this value is overridden when starting
up the Snort daemon through the init.d script by the
value of DEBIAN_SNORT_HOME_NET s defined in the
/etc/snort/snort.debian.conf configuration file

pvar HOME_NET 10.0.224.192/16
```

Fait que là, nous avons la première ligne qui contient les règles locales, c'est là-bas, que nous pouvons écrire nos propres règles. Ensuite on a des règles qui viennent par défaut avec l'installation de snort.



```
# site specific rules
include $RULE_PATH/local.rules

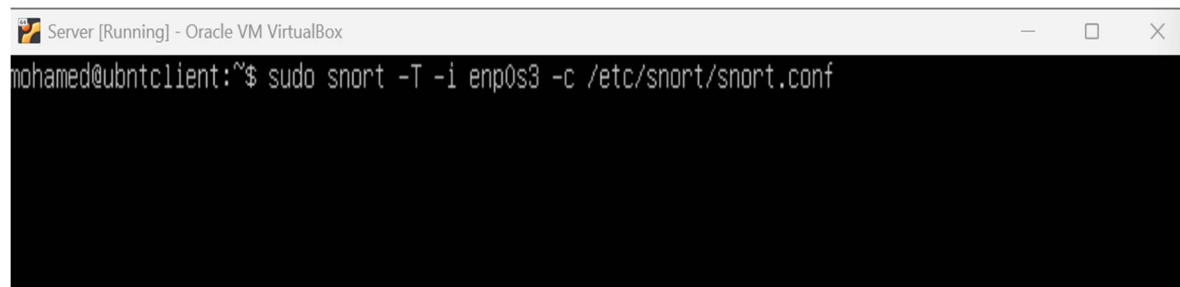
# The include files commented below have been disabled
# because they are not available in the stock Debian
# rules. If you install the Sourcefire VRT please make
# sure you re-enable them again:

#include $RULE_PATH/app-detect.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
#include $RULE_PATH/browser-firefox.rules
#include $RULE_PATH/browser-ie.rules
#include $RULE_PATH/browser-other.rules
#include $RULE_PATH/browser-plugins.rules
#include $RULE_PATH/browser-webkit.rules
include $RULE_PATH/chat.rules
#include $RULE_PATH/content-replace.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/experimental.rules
#include $RULE_PATH/exploit-kit.rules
include $RULE_PATH/exploit.rules
#include $RULE_PATH/file-executable.rules
#include $RULE_PATH/file-flash.rules
#include $RULE_PATH/file-javascript.rules
```

Ensuite on ajoute les deux lignes suivantes sur la section 6 pour enregistrer les logs sur un fichier CSV et PCAP.

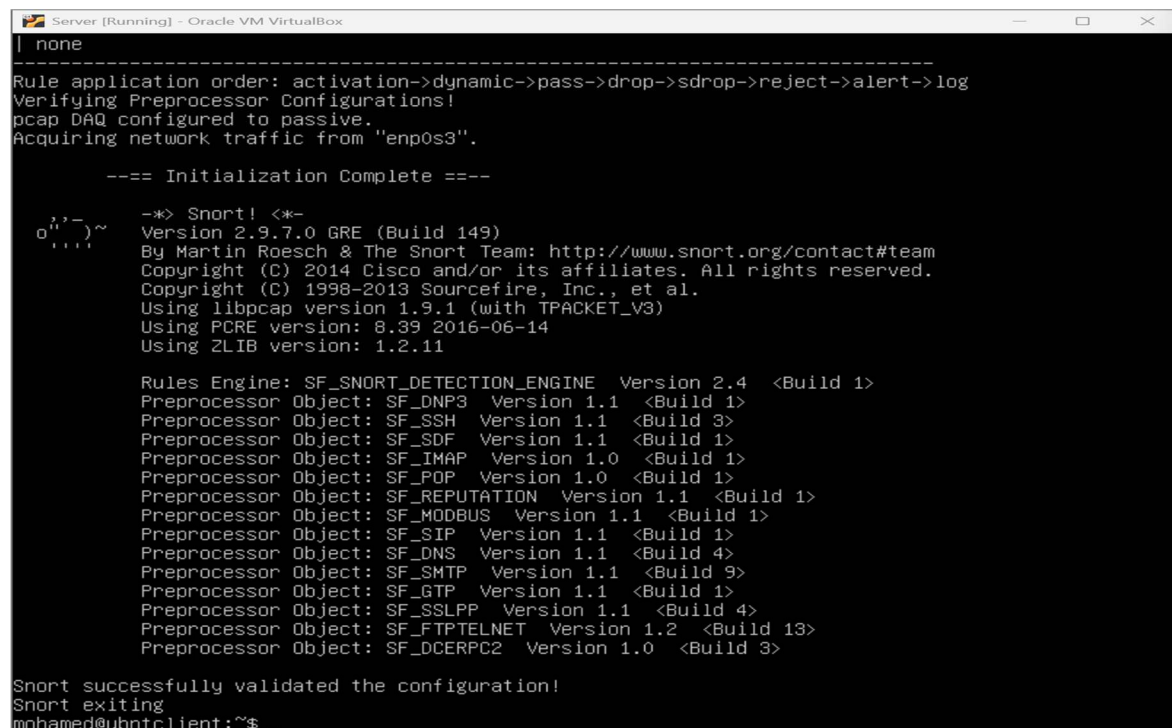
```
#####
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
#####
#csv
output alert_csv: /var/log/snort/alert.csv default
#pcap_
output log_tcpdump: /var/log/snort/tcpdump.log
```

Ensuite on tape la commande ci-dessous pour tester notre fichier de configuration et s'assurer qu'il n'y'a pas d'erreur.



```
Server [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ sudo snort -T -i enp0s3 -c /etc/snort/snort.conf
```

Comme vous pouvez le voir, la configuration a été validé avec succès.



```
Server [Running] - Oracle VM VirtualBox
| none
-----
Rule application order: activation->dynamic->pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".

---== Initialization Complete ===--

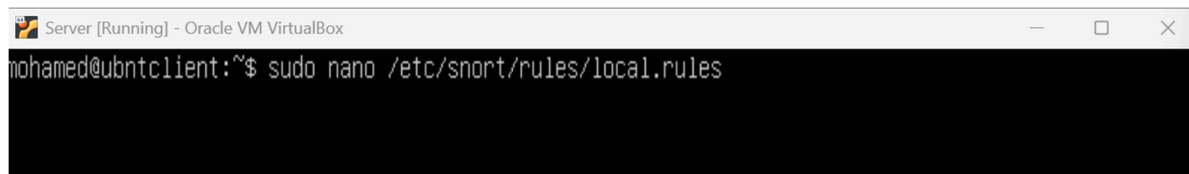
o'-'~
o''~
o''~
o''~

-*> Snort! <*-
Version 2.9.7.0 GRE (Build 149)
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>

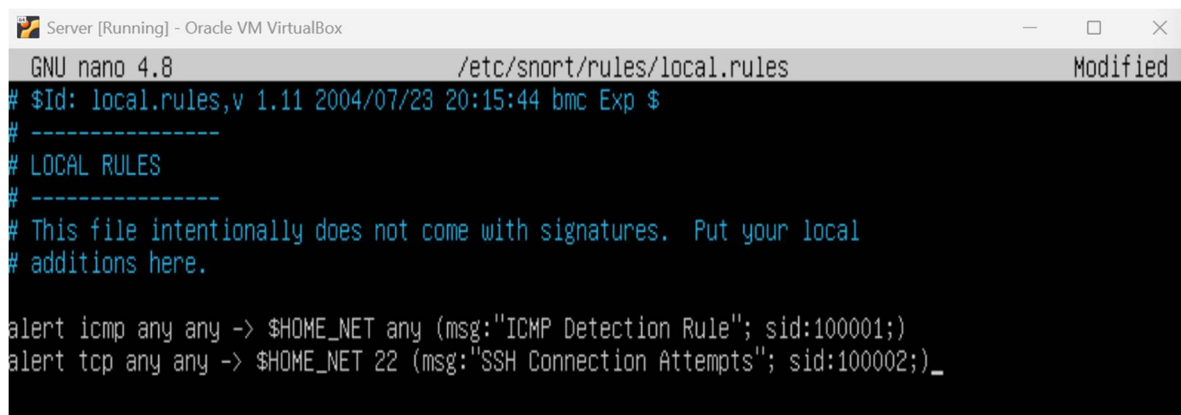
Snort successfully validated the configuration!
Snort exiting
mohamed@ubntclient:~$ _
```

Ensuite nous allons ouvrir notre fichier local.rules qui va nous permettre d'écrire nos propres réglés locales.



```
Server [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ sudo nano /etc/snort/rules/local.rules
```

Ensuite on va ajouter les deux lignes que vous pouvez voir sur l'image ci-dessous. La première ligne est la règle qui va nous permettre de détecter toute connexion icmp. Ce qui veut dire si une autre machine essaye de ping ma machine Ubuntu, IDS est censé le détecter. Puis la deuxième ligne est pour détecter les connexions SSH.

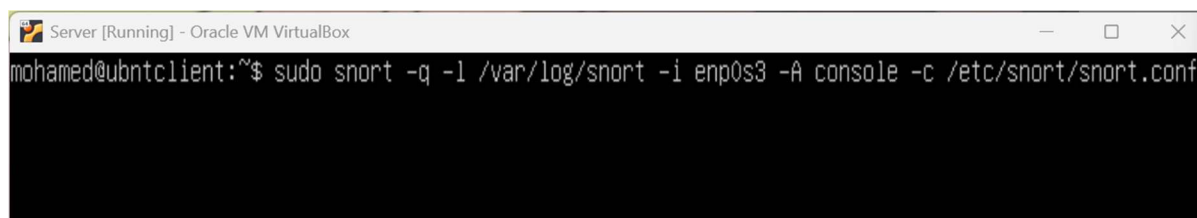


```
Server [Running] - Oracle VM VirtualBox
GNU nano 4.8 /etc/snort/rules/local.rules Modified
# $Id: local.rules,v 1.11 2004/07/23 20:15:44 bmc Exp $
# -----
# LOCAL RULES
# -----
# This file intentionally does not come with signatures. Put your local
# additions here.

alert icmp any any -> $HOME_NET any (msg:"ICMP Detection Rule"; sid:100001;)
alert tcp any any -> $HOME_NET 22 (msg:"SSH Connection Attempts"; sid:100002;)_
```

Étape 3 : Test

Ensuite pour pouvoir voir en temps réel la détection d'intrusion de snort, on peut utiliser la commande ci-dessous pour écouter directement sur la ligne de commande.



```
Server [Running] - Oracle VM VirtualBox
mohamed@ubntclient:~$ sudo snort -q -l /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
```

Ensuite j'ai utilisé une machine qui est sur le même réseau que ma Vm Ubuntu. J'ai lancé une requête ping vers ma Vm Ubuntu. Comme vous pouvez le voir sur l'image ci-dessous, on peut sur la ligne de commande de ma VM Ubuntu, que snort a généré directement des alertes de la requête icmp.

```
Server [Running] - Oracle VM VirtualBox
mohamed@ubuntuclient:~$ sudo snort -q -i /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
05/23-13:37:21.028857 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.71.161 -> 10.0.224.192
05/23-13:37:21.028887 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.224.192 -> 10.0.71.161
05/23-13:37:22.032657 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.71.161 -> 10.0.224.192
05/23-13:37:22.032712 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.224.192 -> 10.0.71.161
05/23-13:37:23.037747 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.71.161 -> 10.0.224.192
05/23-13:37:23.037777 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.224.192 -> 10.0.71.161
05/23-13:37:24.045093 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.71.161 -> 10.0.224.192
05/23-13:37:24.045126 [**] [1:100001:0] ICMP Detection Rule [**] (Priority: 0) {ICMP} 10.0.224.192 -> 10.0.71.161

Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>ping 10.0.224.192

Pinging 10.0.224.192 with 32 bytes of data:
Reply from 10.0.224.192: bytes=32 time=2ms TTL=64
Reply from 10.0.224.192: bytes=32 time=1ms TTL=64
Reply from 10.0.224.192: bytes=32 time<1ms TTL=64
Reply from 10.0.224.192: bytes=32 time=1ms TTL=64

Ping statistics for 10.0.224.192:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 2ms, Average = 1ms

C:\Users\Admin>
C:\Users\Admin>
```

Ensuite J'ai essayé aussi une connexion SSH, comme vous pouvez le voir, Snort l'a détecté avec succès.

```
Server [Running] - Oracle VM VirtualBox
mohamed@ubuntuclient:~$ sudo snort -q -i /var/log/snort -i enp0s3 -A console -c /etc/snort/snort.conf
05/23-13:39:18.806419 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.807668 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.814395 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.828541 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.882498 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.833490 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:18.882498 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:21.753351 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:21.753351 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:21.753351 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:21.755069 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:21.806065 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:24.489897 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:24.502228 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:25.633825 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:25.634794 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:25.650471 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:25.651926 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22
05/23-13:39:25.756573 [**] [1:100002:0] SSH Connection Attempts [**] (Priority: 0) {TCP} 10.0.71.161:1:57187 -> 10.0.224.192:22

C:\Users\Admin>ssh mohamed@10.0.224.192
The authenticity of host '10.0.224.192 (10.0.224.192)' can't be established.
ED25519 key fingerprint is SHA256:584xpHYMTSwU6/iPudJMenSW7J4U4m+XOqh6H3wcWJU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.224.192' (ED25519) to the list of known hosts.
mohamed@10.0.224.192's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-216-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of ven. 23 mai 2025 13:39:24 UTC

System load:  0.0          Processes:           128
Usage of /:   42.2% of 11.21GB  Users logged in:    1
Memory usage: 22%          IPv4 address for enp0s3: 10.0.224.192
Swap usage:   0%

 * Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet
connection or proxy settings

Last login: Fri May 23 13:32:02 2025
mohamed@ubuntuclient:~$
```

Rôle de Snort comme IDS

Snort est un système de détection d'intrusion. Son rôle en tant que IDS, et qu'il nous permet de surveiller toute activité d'intrusion sur notre réseau. Il peut être configuré non seulement pour surveiller tout un réseau mais aussi pour surveiller juste une seule machine comme nous l'avons testé sur ce laboratoire. Il peut nous informer en temps réel, de toute tentative de connexion, et aussi enregistrer les logs de toutes intrusion sur un fichier, qu'on peut consulter plus tard.

Conclusion

En fin, grâce à ce laboratoire, j'ai appris à installer et configurer Snort comme système de détection d'intrusion(IDS), de tester sa configuration et de générer des alertes sur des attaques simulées comme une connexion ssh par exemple.