

# Programa de Fundamentos de Ciberseguridad

3° Edición

## Taller Módulo II Implementación del IDS Snort

By:

WoSEC Panamá

Comunidad DOJO



**COMUNIDAD  
DOJO**

# Objetivos del taller:

Armar un pequeño laboratorio de detección de amenazas en la red y probar su correcto funcionamiento utilizando herramientas OpenSource que nos permitan conocer cómo podemos identificar amenazas y mejorar nuestras defensas.

# Disclaimer:

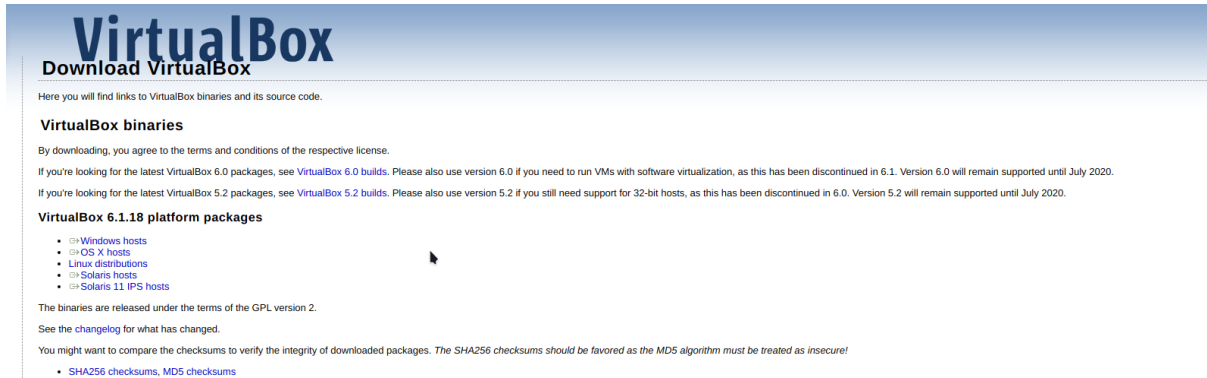
Este laboratorio se realiza sólomente con fines educativos y de aprendizaje, con el fin de brindar información que permita mejorar las defensas en ciberseguridad.

# Metodología:

1. Se desplegará el sistema de detección de intrusos (IDS) Snort en un entorno Linux utilizando docker y se realizarán las configuraciones necesarias para su funcionamiento.
2. Se configurarán reglas en el IDS que nos permitan alertar una vez se genere cierto tipo de tráfico en la red.
3. Se utilizará la herramienta Nmap desde un entorno Linux para realizar escaneos sobre la red a fin de que se puedan generar las alertas en el IDS.

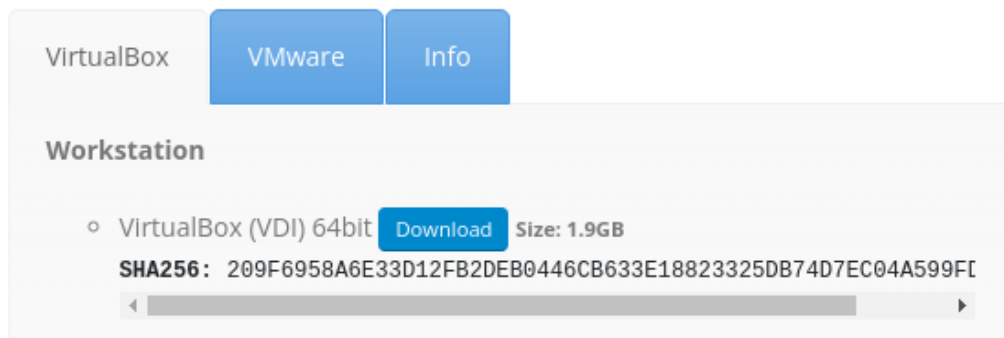
# Prerrequisitos:

Descargar e instalar VirtualBox: <https://www.virtualbox.org/wiki/Downloads>



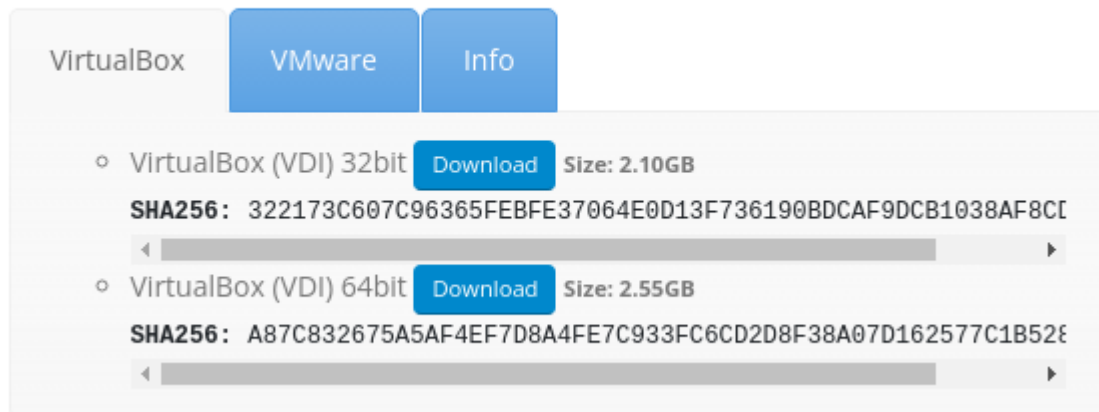
Descargar una máquina virtual con Fedora 34 la cual pueden conseguir desde OSboxes: <https://www.osboxes.org/fedora/>

## Fedora 34



Descargar una máquina virtual con Kali Linux la cual pueden conseguir desde OSboxes: <https://www.osboxes.org/kali-linux/>

## Kali Linux 2022.1 (All Tools)



Nota: La ventaja de las máquinas de OSBoxes es que ya se encuentran “Listas para usar”. Utilizan el siguiente usuario y contraseña por defecto: Usuario: **osboxes**  
Contraseña: **osboxes.org**

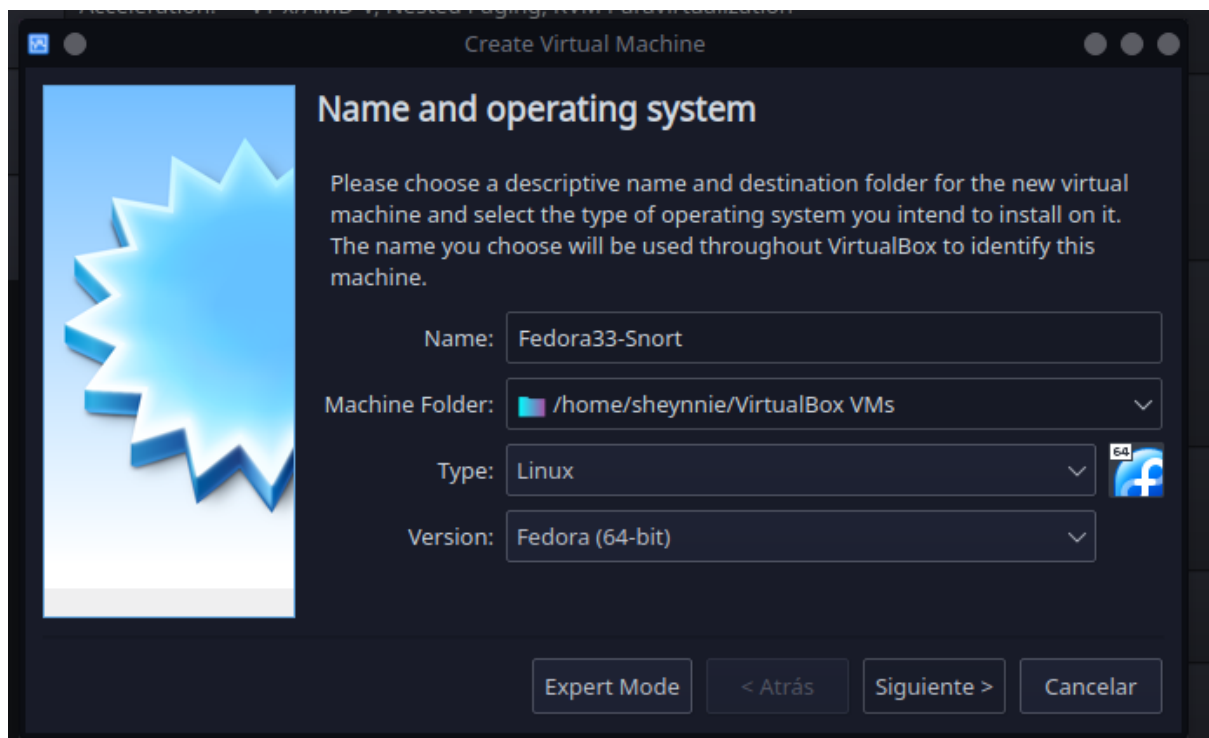
Una vez descargado el archivo de OSboxes, descomprimirlo en un subdirectorio:

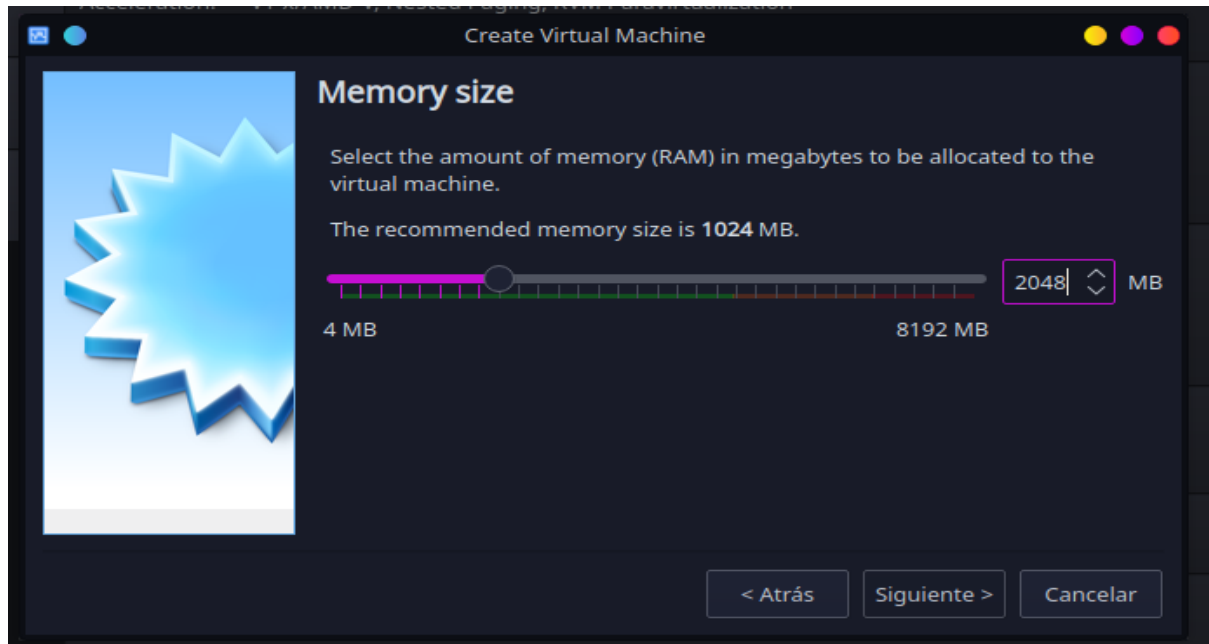
```
localhost:/home/sheynnie/Downloads # p7zip -d Fedora-33-Workstation-VB_64bit.7z
```

### Para la máquina Fedora:

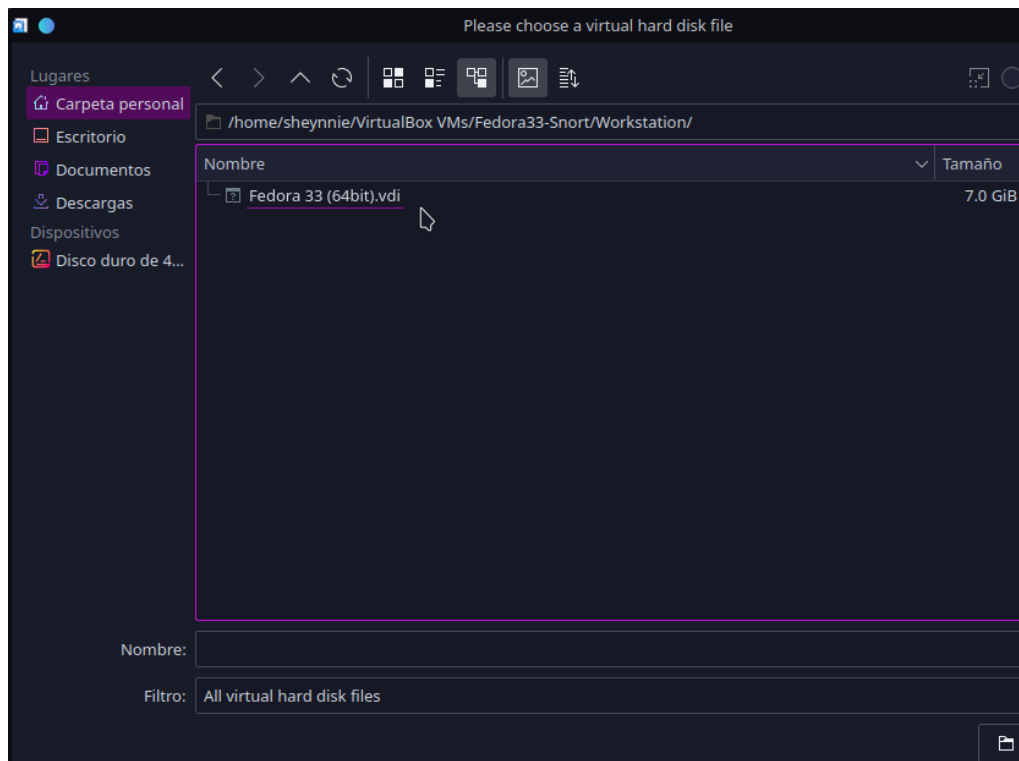
Abrir virtualbox y crear 1 máquina virtual con las siguientes características:

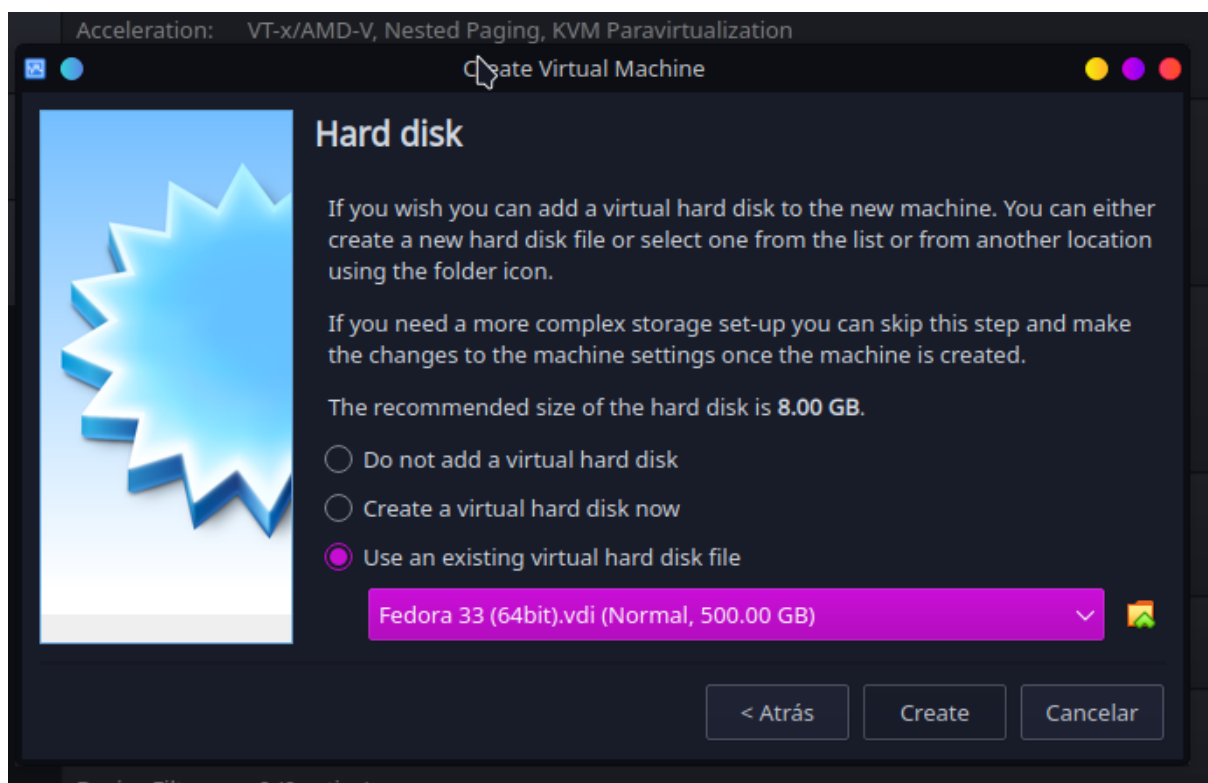
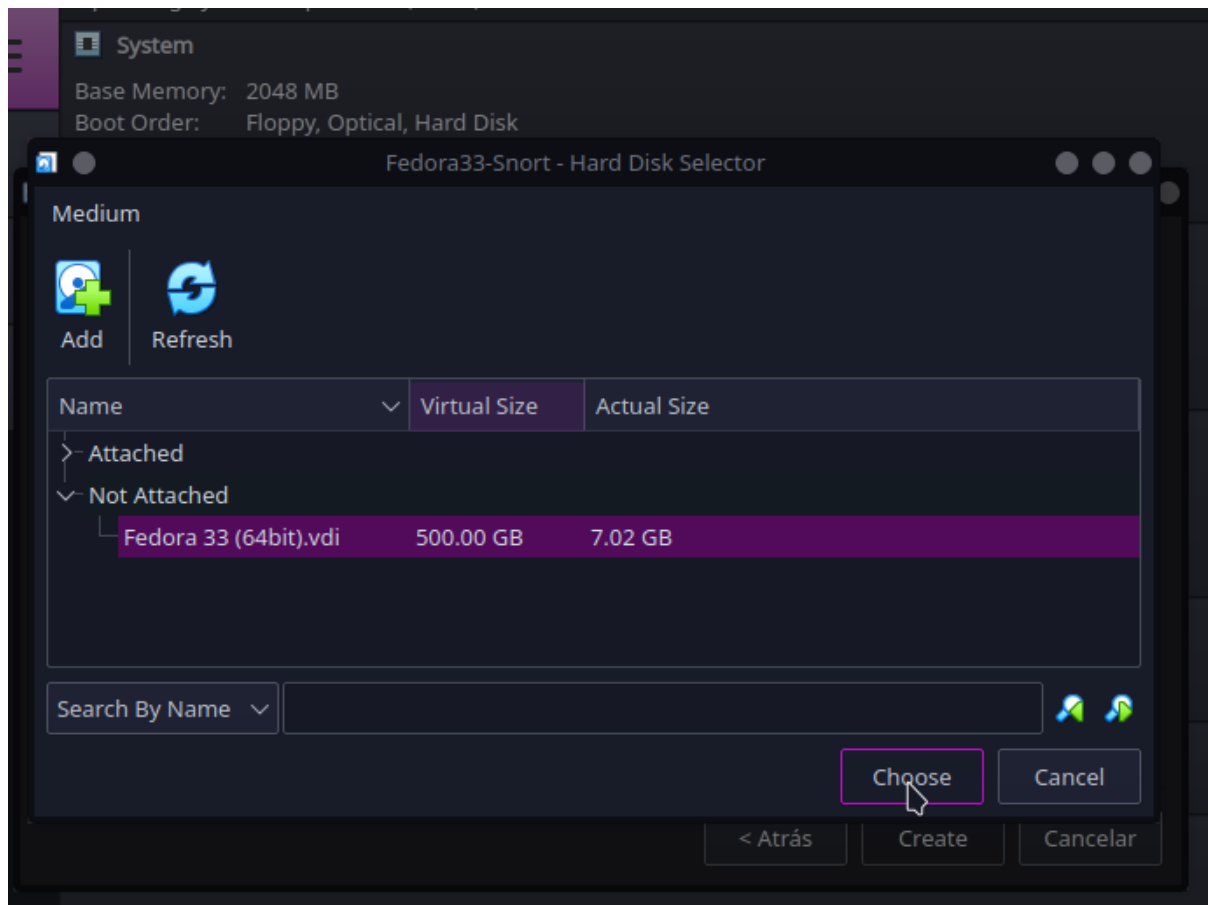
Mínimo 2048 mb de memoria





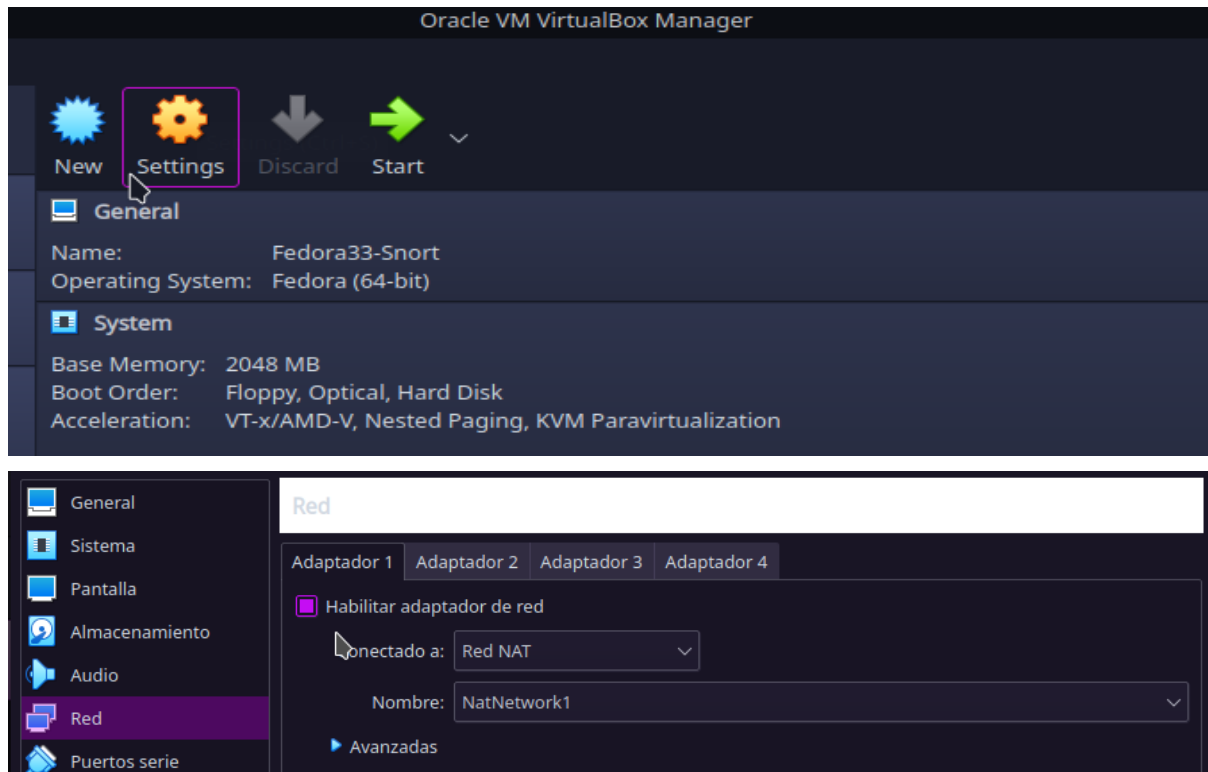
Asignar el archivo .vdi previamente extraído como disco duro existente de la máquina:





Dar click al botón crear para finalizar.

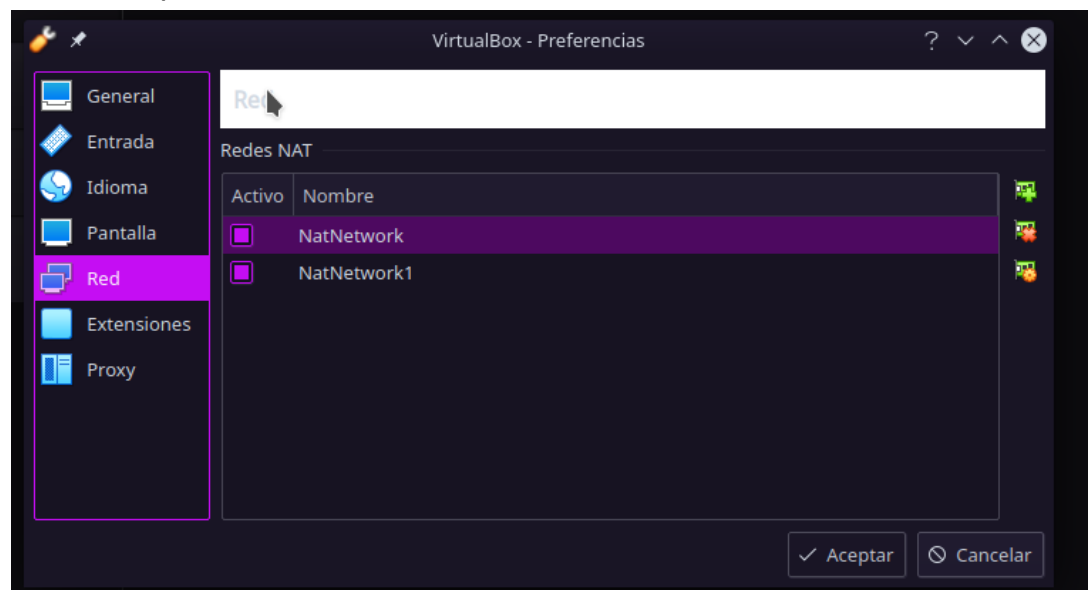
Posteriormente, nos dirigimos a la máquina creada y damos click al botón configuración->red y en la sección del primer adaptador, asigne la opción “Red NAT” y seleccione una red NAT previamente creada.



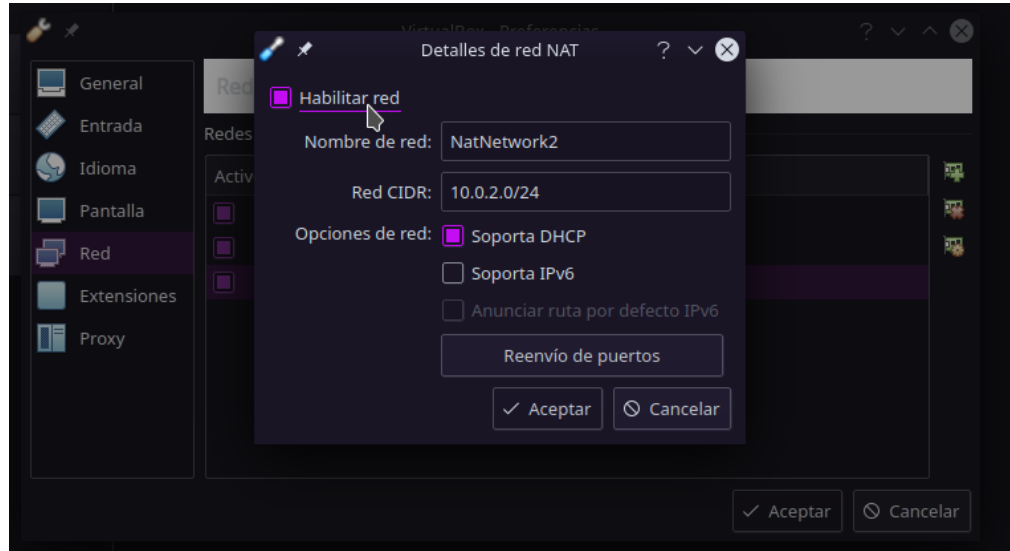
Con esta configuración ya podemos iniciar la máquina.

Notas:

- Si no tiene una red NAT creada previamente, puede realizarlo de la siguiente forma:
  - Desde la opción “**Archivo->Preferencias->red**” de VirtualBox



- Donde, con el botón **+** puede agregar una nueva red y con el botón de **configuración** puede cambiar el nombre, rango de red y opciones como DHCP, IPV6 y reenvío de puertos. Para fines de este laboratorio solo necesitamos especificar el rango de red y la opción de DHCP para la asignación de IP automática.



- Para la máquina Fedora se debe habilitar en esta misma sección el modo promiscuo en: “permitir todo” para que pueda visualizar todo el tráfico de la red ya que en esta se instalará el IDS Snort.
- Se recomienda actualizar la máquina:
  - Desde la terminal ingrese el comando: “***sudo dnf update***”.

### Para la máquina Kali:

Repetimos el mismo procedimiento utilizado para la máquina de Fedora.

### Notas:

- Ambas máquinas deben quedar bajo la misma red NAT para que puedan visualizarse entre sí.
- Se recomienda actualizar la máquina:
  - Desde la terminal ingrese el comando “***sudo apt-get update***”.



# Parte I - Configuración del IDS Snort en la máquina Fedora.

Ingresamos el siguiente comando para instalar los prerequisites necesarios: ***sudo dnf install docker***

```
osboxes@fedora:~ — sudo dnf install docker
[osboxes@fedora ~]$ sudo dnf install docker
[sudo] password for osboxes:
Last metadata expiration check: 0:02:51 ago on Wed 06 Apr 2022 11:26:55 PM EDT.
Dependencies resolved.
=====
Package                Architecture Version           Repository        Size
=====
Installing:
moby-engine             x86_64          20.10.12-1.fc34  updates          33 M
Installing dependencies:
containerd              x86_64          1.6.1-1.fc34     updates          39 M
runc                    x86_64          2:1.1.0-1.fc34   updates          3.1 M
=====
Transaction Summary
=====
Install 3 Packages

Total download size: 75 M
Installed size: 317 M
Is this ok [y/N]: y
Downloading Packages:
(1/3): runc-1.1.0-1.fc34.x86_64.rpm          921 kB/s | 3.1 MB   00:03
(2/3): moby-engine-20.10.12-1.fc34.x86_64.rpm 3.4 MB/s | 33 MB   00:09
(3/3): containerd-1.6.1-1.fc34.x86_64.rpm    3.7 MB/s | 39 MB   00:10
```

Iniciamos el servicio de docker

***sudo systemctl start docker***

```
[osboxes@fedora ~]$ sudo systemctl start docker
```

Revisamos que se encuentre en ejecución:

***sudo systemctl status docker***

```

osboxes@fedora:~ — sudo systemctl status docker

[osboxes@fedora ~]$ sudo systemctl start docker
[osboxes@fedora ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor p
   Active: active (running) since Wed 2022-04-06 23:31:39 EDT; 26s ago
   TriggeredBy: ● docker.socket
     Docs: https://docs.docker.com
    Main PID: 41476 (dockerd)
      Tasks: 15 (limit: 2329)
     Memory: 102.2M
        CPU: 300ms
    CGroup: /system.slice/docker.service
            └─41476 /usr/bin/dockerd --host=fd:// --exec-opt native.cgroupdriv
              └─41481 containerd --config /var/run/docker/containerd/containerd.

```

Ahora, pasaremos a ejecutar Snort desde docker:

**`sudo docker run -it --name snort --net=host --cap-add=NET_ADMIN  
linton/docker-snort /bin/bash`**

```

osboxes@fedora:~ — sudo docker run -it --name snort --net=host --cap-add=NET_AD...

osboxes@fedora:~ — sudo ... x osboxes@fedora:~ x osboxes@fedora:~ x
[osboxes@fedora ~]$ sudo docker run -it --name snort --net=host --cap-add=NET_AD
MIN linton/docker-snort /bin/bash
Unable to find image 'linton/docker-snort:latest' locally
latest: Pulling from linton/docker-snort
6c953ac5d795: Downloading 15.61MB/65.7MB
3eed5ff20a90: Download complete
f8419ea7c1b5: Download complete
51900bc9e720: Download complete
a3ed95caeb02: Download complete
ab939ba3f43e: Downloading 531.7kB/125.2MB
7ae92921c441: Waiting
9e5d3e1ddbda: Waiting
807808ccf631: Waiting
0372b3540a39: Waiting
5c49542d91ef: Waiting
9d194e7d0569: Waiting
bd2d51f11e87: Waiting

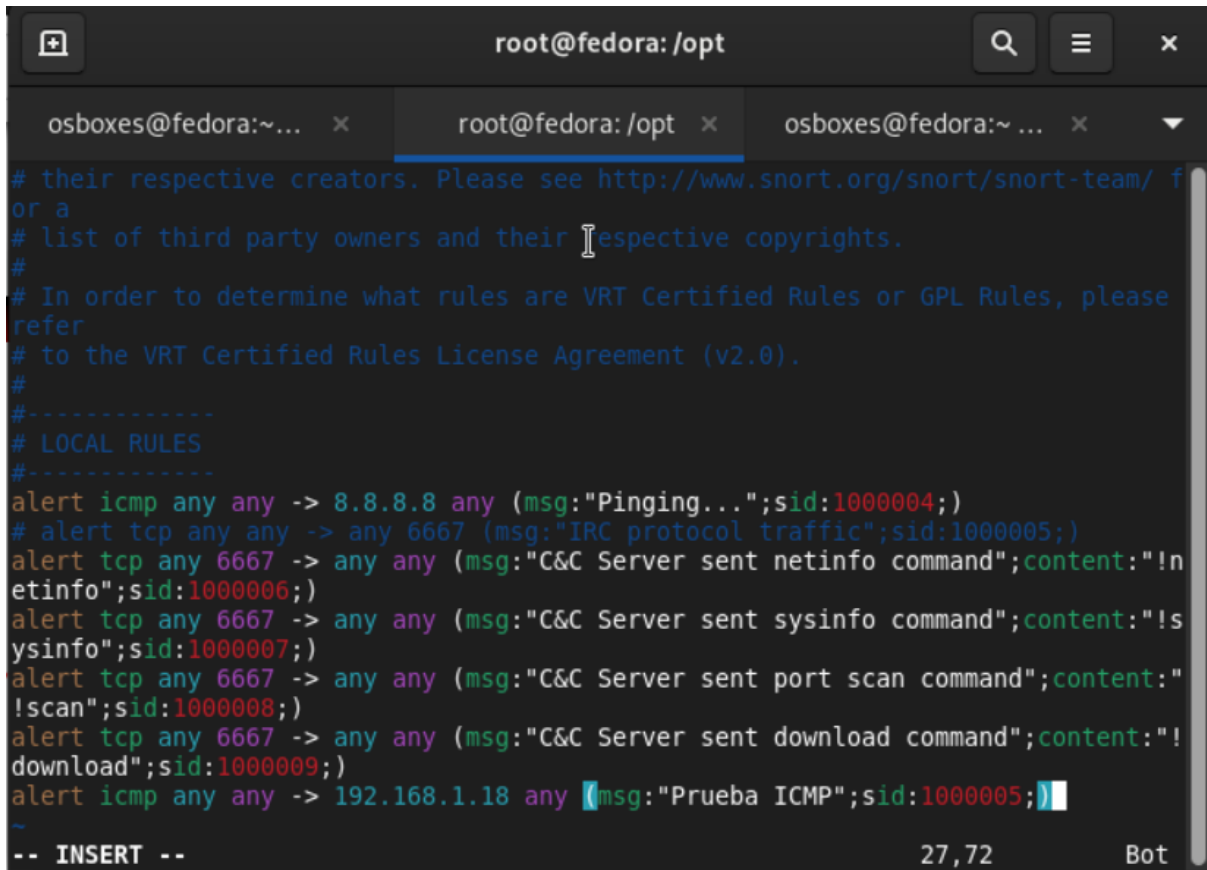
```

Una vez completado, escribimos una regla en el archivo de reglas locales que alertará cualquier tráfico ICMP generado, la que usará en el proceso de prueba de snort. Para ello ingresamos el comando **`sudo vi /etc/snort/rules/local.rules`**

```
root@fedora:/opt# sudo vi /etc/snort/rules/local.rules
```

luego ingresamos el comando **`i`** y agregamos lo siguiente, donde **`xxx.xxx.x.xx`** será la ip de nuestra máquina:

**`alert icmp any any -> xxx.xxx.x.xx any (msg:"Prueba ICMP";sid:1000005;)`**

A terminal window titled 'root@fedora: /opt' with three tabs: 'osboxes@fedora:~...', 'root@fedora: /opt', and 'osboxes@fedora:~ ...'. The terminal displays Snort rule configuration text. It includes a header with a URL to the Snort team's website, followed by a section for 'LOCAL RULES'. Several rules are defined for alerting on specific traffic patterns, such as ICMP pings to 8.8.8.8, TCP connections to port 6667, and C&C server commands. The last rule is an ICMP rule for 192.168.1.18 with a message 'Prueba ICMP'. The terminal ends with '-- INSERT --', a line count of '27,72', and the text 'Bot'.

Y guardamos los cambios con **esc :wq!**

Después de esto, ejecutamos Snort para validar la configuración, donde **interfaz** será la interfaz de red de nuestra máquina: **sudo snort -i interfaz -c /etc/snort/etc/snort.conf -A console**

```
root@fedora: /opt

osboxes@fedora:~ x osboxes@fedora:~ x root@fedora: /opt x

root@fedora:/opt# sudo snort -i enp0s3 -c /etc/snort/etc/snort.conf -A console
sudo: unable to resolve host fedora
Running in IDS mode

--== Initializing Snort ==--
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/etc/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 36 80:90 311 383 555 591 593 631 801 808 818 9
01 972 1158 1220 1414 1533 1741 1830 1942 2231 2301 2381 2578 2809 2980 3029 303
7 3057 3128 3443 3702 4000 4343 4848 5000 5117 5250 5600 6080 6173 6988 7000:700
1 7071 7144:7145 7510 7770 7777:7779 8000 8008 8014 8028 8080:8082 8085 8088 809
0 8118 8123 8180:8181 8222 8243 8280 8300 8333 8344 8500 8509 8800 8888 8899 898
3 9000 9060 9080 9090:9091 9111 9290 9443 9999:10000 11371 12601 13014 15489 299
91 33300 34412 34443:34444 41080 44449 50000 50002 51423 53331 55252 55555 56712
]
PortVar 'SHELLCODE_PORTS' defined : [ 0:79 81:65535 ]
PortVar 'ORACLE_PORTS' defined : [ 1024:65535 ]
PortVar 'SSH_PORTS' defined : [ 22 ]
PortVar 'FTP_PORTS' defined : [ 21 2100 3535 ]
PortVar 'SIP_PORTS' defined : [ 5060:5061 5600 ]
PortVar 'FILE_DATA_PORTS' defined : [ 36 80:90 110 143 311 383 555 591 593 631
...

```

```
root@fedora: /opt

osboxes@fedora:~ x osboxes@fedora:~ x root@fedora: /opt x

Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.5.3
Using PCRE version: 8.31 2012-07-06
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT DETECTION ENGINE Version 2.6 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Commencing packet processing (pid=19)
```

Una vez se encuentra en ejecución, vamos a lanzar un ping desde la máquina Kali que se encuentra bajo la misma red NAT hacia nuestra máquina actual:

```

osboxes@osboxes: ~ x  osboxes@osboxes: /var/www/html x  osboxes@osboxes: ~ x
(osboxes@osboxes)-[~]
$ ip address
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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:ce:0c:15 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute eth0
        valid_lft 582sec preferred_lft 582sec
    inet6 fe80::a00:27ff:fece:c15/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:74:3c:2d:b8 brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

(osboxes@osboxes)-[~]
$ ping 192.168.1.18
PING 192.168.1.18 (192.168.1.18) 56(84) bytes of data.
64 bytes from 192.168.1.18: icmp_seq=1 ttl=64 time=0.551 ms
64 bytes from 192.168.1.18: icmp_seq=2 ttl=64 time=0.271 ms
64 bytes from 192.168.1.18: icmp_seq=3 ttl=64 time=0.388 ms

```

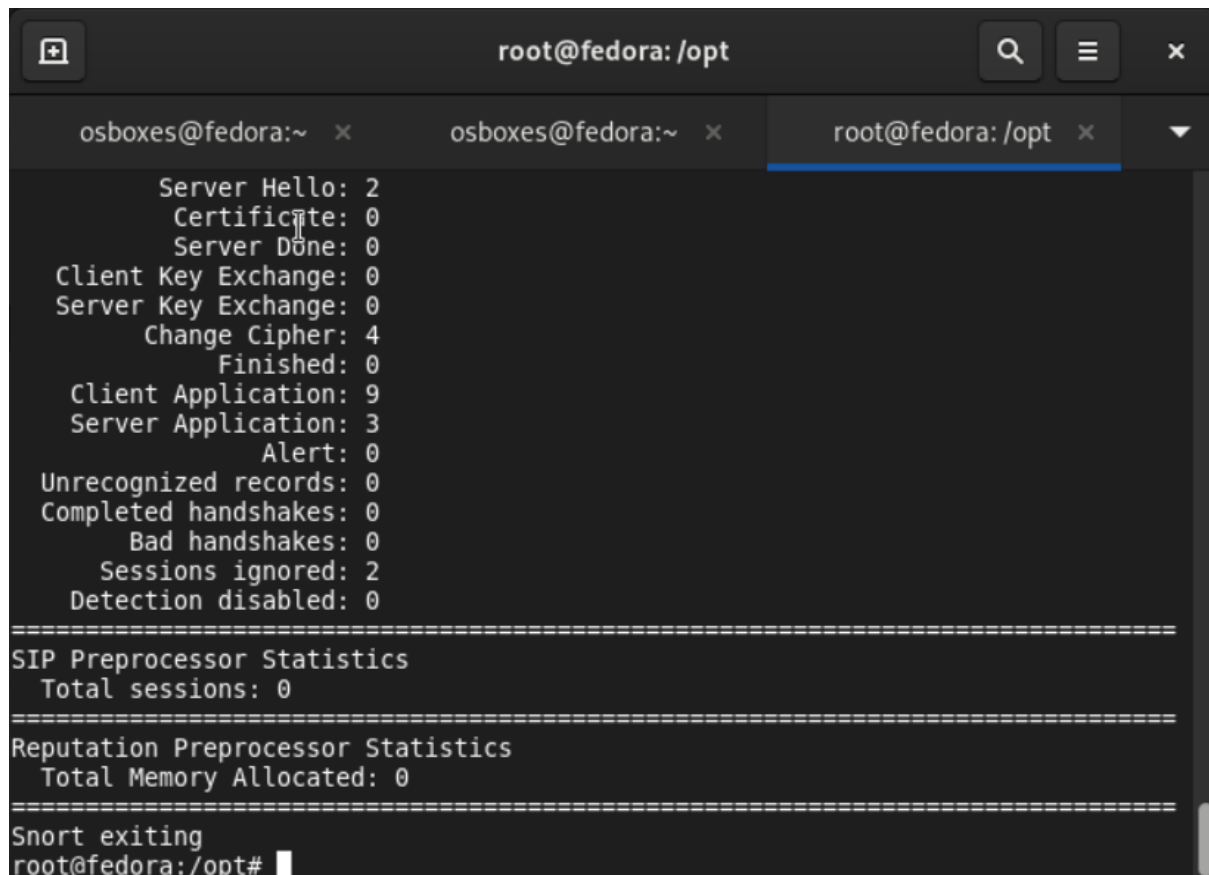
Y desde la máquina actual con Snort vamos a ver reflejada la alerta desde la consola:

```

root@fedora: /opt
osboxes@fedora:~ x  osboxes@fedora:~ x  root@fedora: /opt x
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Commencing packet processing (pid=19)
04/07-05:31:14.289558  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:15.310823  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:16.355572  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:17.400210  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:18.444841  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:19.489430  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:20.535027  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:21.559426  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:22.591194  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:23.636261  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}
192.168.1.11 -> 192.168.1.18
04/07-05:31:24.657845  [**] [1:10000005:0] Prueba ICMP [**] [Priority: 0] {ICMP}

```

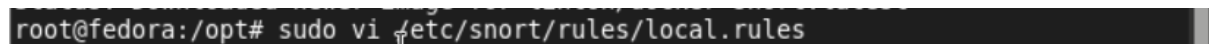
Podemos parar la ejecución del ping y de Snort con **ctrl C**



The screenshot shows a terminal window with the title bar 'root@fedora: /opt'. The terminal content displays various statistics for Snort, including 'Server Hello: 2', 'Certificate: 0', 'Server Done: 0', 'Client Key Exchange: 0', 'Server Key Exchange: 0', 'Change Cipher: 4', 'Finished: 0', 'Client Application: 9', 'Server Application: 3', 'Alert: 0', 'Unrecognized records: 0', 'Completed handshakes: 0', 'Bad handshakes: 0', 'Sessions ignored: 2', and 'Detection disabled: 0'. Below these are sections for 'SIP Preprocessor Statistics' (Total sessions: 0) and 'Reputation Preprocessor Statistics' (Total Memory Allocated: 0). The terminal ends with the message 'Snort exiting' and the prompt 'root@fedora:/opt#'.

```
root@fedora: /opt
osboxes@fedora:~ x osboxes@fedora:~ x root@fedora: /opt x
Server Hello: 2
Certificate: 0
Server Done: 0
Client Key Exchange: 0
Server Key Exchange: 0
Change Cipher: 4
Finished: 0
Client Application: 9
Server Application: 3
Alert: 0
Unrecognized records: 0
Completed handshakes: 0
Bad handshakes: 0
Sessions ignored: 2
Detection disabled: 0
=====
SIP Preprocessor Statistics
Total sessions: 0
=====
Reputation Preprocessor Statistics
Total Memory Allocated: 0
=====
Snort exiting
root@fedora:/opt#
```

Agregamos una segunda regla en el archivo de reglas locales que alertará cualquier tráfico TCP generado, la que usará en el proceso de prueba con Nmap. Para ello ingresamos el comando **sudo vi /etc/snort/rules/local.rules**



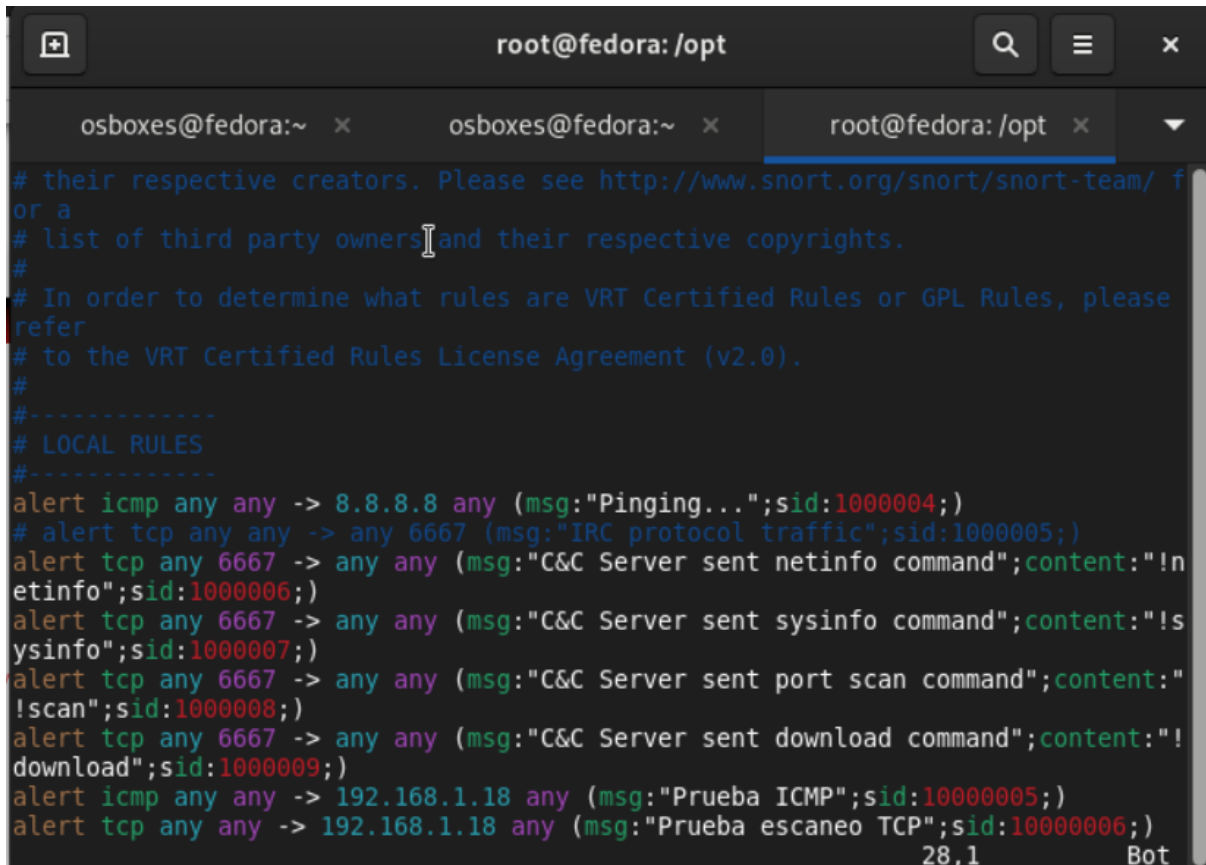
The screenshot shows a terminal window with the command 'sudo vi /etc/snort/rules/local.rules' entered at the prompt 'root@fedora:/opt#'.

```
root@fedora:/opt# sudo vi /etc/snort/rules/local.rules
```

luego ingresamos el comando **i** y agregamos lo siguiente, donde **xxx.xxx.x.xx** será la ip de nuestra máquina:

**alert tcp any any -> xxx.xxx.x.xx any (msg:"Prueba escaneo TCP";sid:1000006;)**



A terminal window titled 'root@fedora: /opt' with three tabs: 'osboxes@fedora:~', 'osboxes@fedora:~', and 'root@fedora: /opt'. The active tab shows the Snort rule configuration file. The text is as follows:

```
# their respective creators. Please see http://www.snort.org/snort/snort-team/ f
or a
# list of third party owners and their respective copyrights.
#
# In order to determine what rules are VRT Certified Rules or GPL Rules, please
refer
# to the VRT Certified Rules License Agreement (v2.0).
#
#-----
# LOCAL RULES
#-----
alert icmp any any -> 8.8.8.8 any (msg:"Pinging...";sid:1000004;)
# alert tcp any any -> any 6667 (msg:"IRC protocol traffic";sid:1000005;)
alert tcp any 6667 -> any any (msg:"C&C Server sent netinfo command";content:"!n
etinfo";sid:1000006;)
alert tcp any 6667 -> any any (msg:"C&C Server sent sysinfo command";content:"!s
ysinfo";sid:1000007;)
alert tcp any 6667 -> any any (msg:"C&C Server sent port scan command";content:"
!scan";sid:1000008;)
alert tcp any 6667 -> any any (msg:"C&C Server sent download command";content:"!
download";sid:1000009;)
alert icmp any any -> 192.168.1.18 any (msg:"Prueba ICMP";sid:10000005;)
alert tcp any any -> 192.168.1.18 any (msg:"Prueba escaneo TCP";sid:10000006;)
28,1 Bot
```

luego guardamos los cambios con **esc :wq!**

Dejamos nuevamente Snort en ejecución: **sudo snort -i interfaz -c**

**/etc/snort/etc/snort.conf -A console**

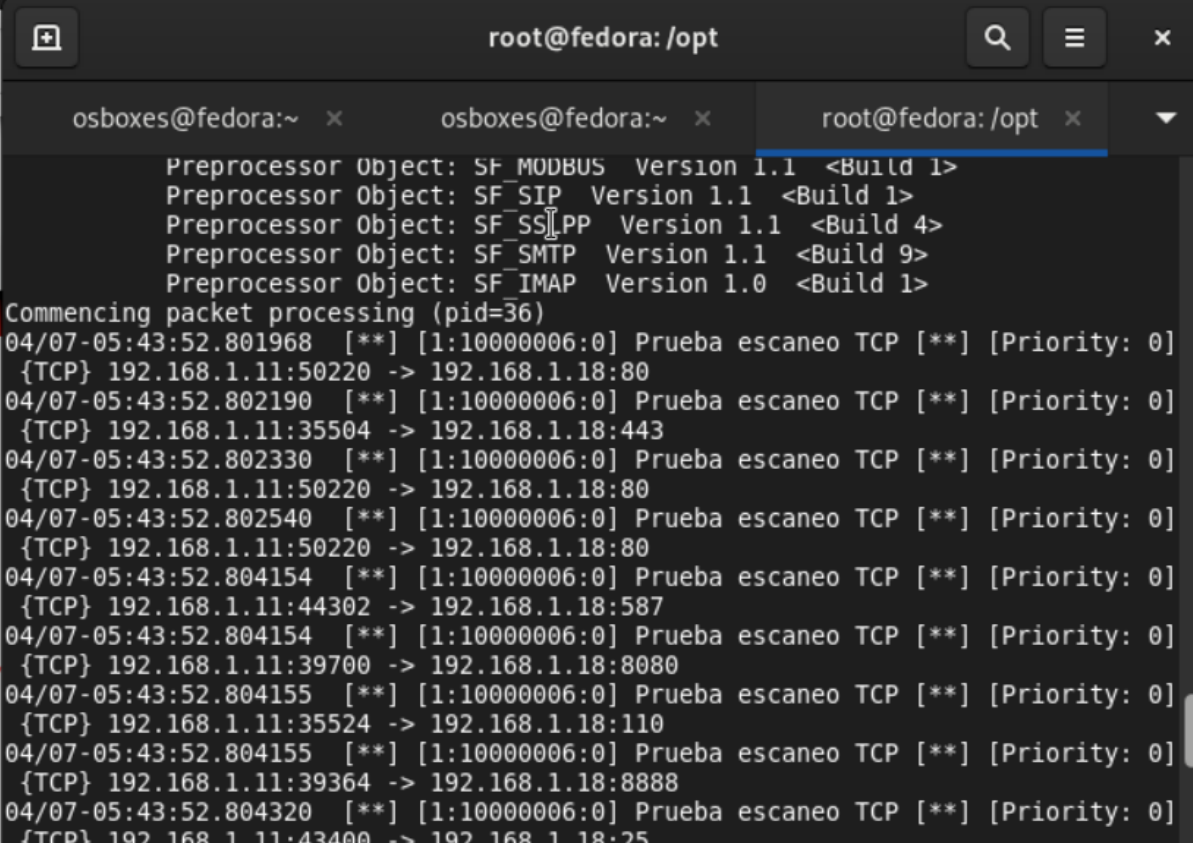
## Parte II - Lanzamiento de escaneos de red con Nmap.

Ingresamos el comando `nmap -T4 -F` seguido de la dirección IP a escanear, seguido del parámetro `-v`, la cual será nuestra máquina con Snort.

```
(osboxes@osboxes)-[~]  
$ nmap -T4 -F 192.168.1.18 -v
```

Con las opciones `-T4 -F` indicamos que realizaremos un escaneo rápido y agresivo.

Damos enter y luego de unos segundos, veremos que las alertas se empiezan a reflejar en la máquina con Snort.



```
root@fedora: /opt  
osboxes@fedora:~ x osboxes@fedora:~ x root@fedora: /opt x  
Preprocessor Object: SF_MQDBUS Version 1.1 <Build 1>  
Preprocessor Object: SF_SIP Version 1.1 <Build 1>  
Preprocessor Object: SF_SSIPP Version 1.1 <Build 4>  
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>  
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>  
Commencing packet processing (pid=36)  
04/07-05:43:52.801968  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:50220 -> 192.168.1.18:80  
04/07-05:43:52.802190  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:35504 -> 192.168.1.18:443  
04/07-05:43:52.802330  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:50220 -> 192.168.1.18:80  
04/07-05:43:52.802540  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:50220 -> 192.168.1.18:80  
04/07-05:43:52.804154  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:44302 -> 192.168.1.18:587  
04/07-05:43:52.804154  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:39700 -> 192.168.1.18:8080  
04/07-05:43:52.804155  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:35524 -> 192.168.1.18:110  
04/07-05:43:52.804155  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:39364 -> 192.168.1.18:8888  
04/07-05:43:52.804320  [**] [1:10000006:0] Prueba escaneo TCP [**] [Priority: 0]  
{TCP} 192.168.1.11:43400 -> 192.168.1.18:25
```