

Taller | Armando un laboratorio para detectar amenazas en la red



Dictado por: Sheyla Leacock

Objetivos del taller:

El objetivo de este taller es armar un laboratorio de detección de amenazas con herramientas OpenSource y simular ataques a nuestro entorno de red que nos permitan conocer cómo podemos mejorar nuestras defensas.

Se propone desplegar de forma automatizada el sistema de detección de intrusos Snort en un entorno Linux, así como realizar las configuraciones e implementar reglas que nos permitan detectar amenazas a nivel de red.

Se implementará también una arquitectura cliente servidor con Pytbull para generar tráfico malicioso en nuestra red a fin de que pueda ser detectado por el IDS.

Además desplegaremos herramientas que nos permitan monitorear las alertas generadas por el IDS y realizar correlación de eventos

Prerrequisitos:

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

The screenshot shows the official VirtualBox download page. At the top, it says "VirtualBox" and "Download VirtualBox". Below that, a note says "Here you will find links to VirtualBox binaries and its source code." Under "VirtualBox binaries", there's a note about accepting the license terms. It then lists "VirtualBox 6.1.18 platform packages" for various host operating systems: Windows hosts, OS X hosts, Linux distributions, Solaris hosts, and Solaris 11 IPS hosts. A note below says the binaries are released under the GPL version 2. There's also a link to the changelog and instructions for verifying checksums.

Descargar una máquina virtual con OpenSUSE 15.2 desde OSboxes:

<https://www.osboxes.org/opensuse/>

openSUSE 15.2 Leap

The screenshot shows the OSboxes download page for openSUSE 15.2 Leap. It has tabs for "VirtualBox", "VMware" (which is selected), and "Info". Below the tabs, it says "KDE Plasma Version". It lists a "VirtualBox (VDI) 64bit" option with a "Download" button, indicating a size of 1.6GB. A SHA256 checksum is provided: f7b58799e3f24b6b887a7808b94b1543a307b855d7f0d543846f. There are navigation arrows at the bottom of the list.

Descargar una máquina virtual con Fedora 33 desde OSboxes:

<https://www.osboxes.org/fedora/>

Fedora 33

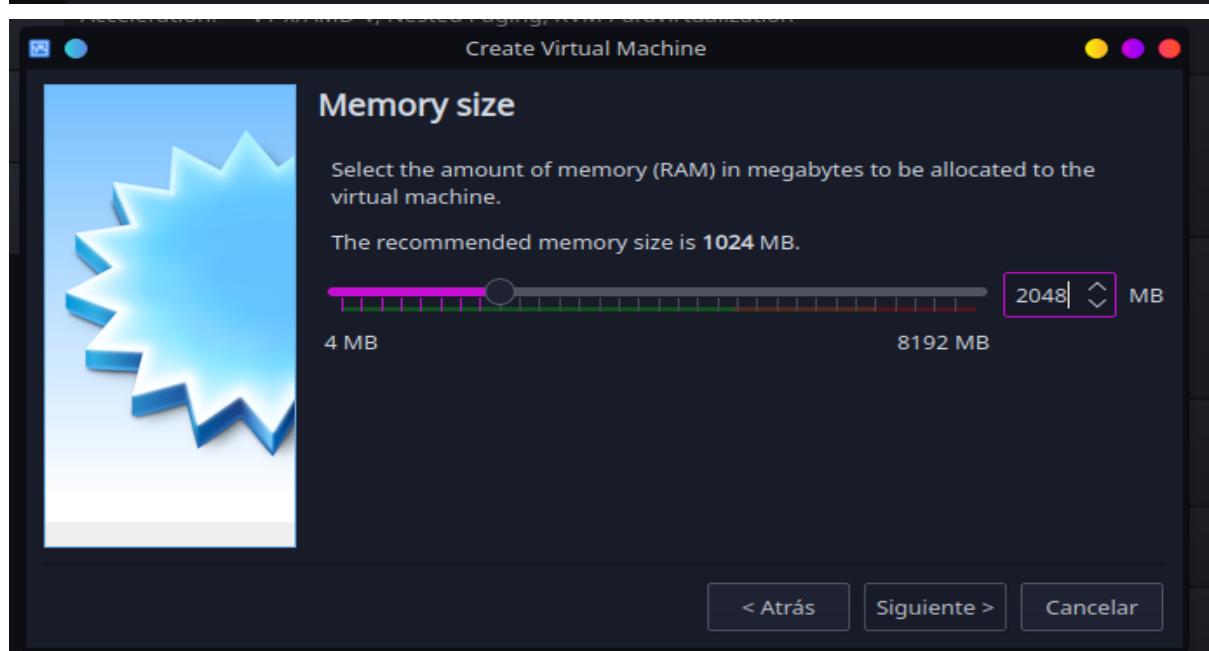
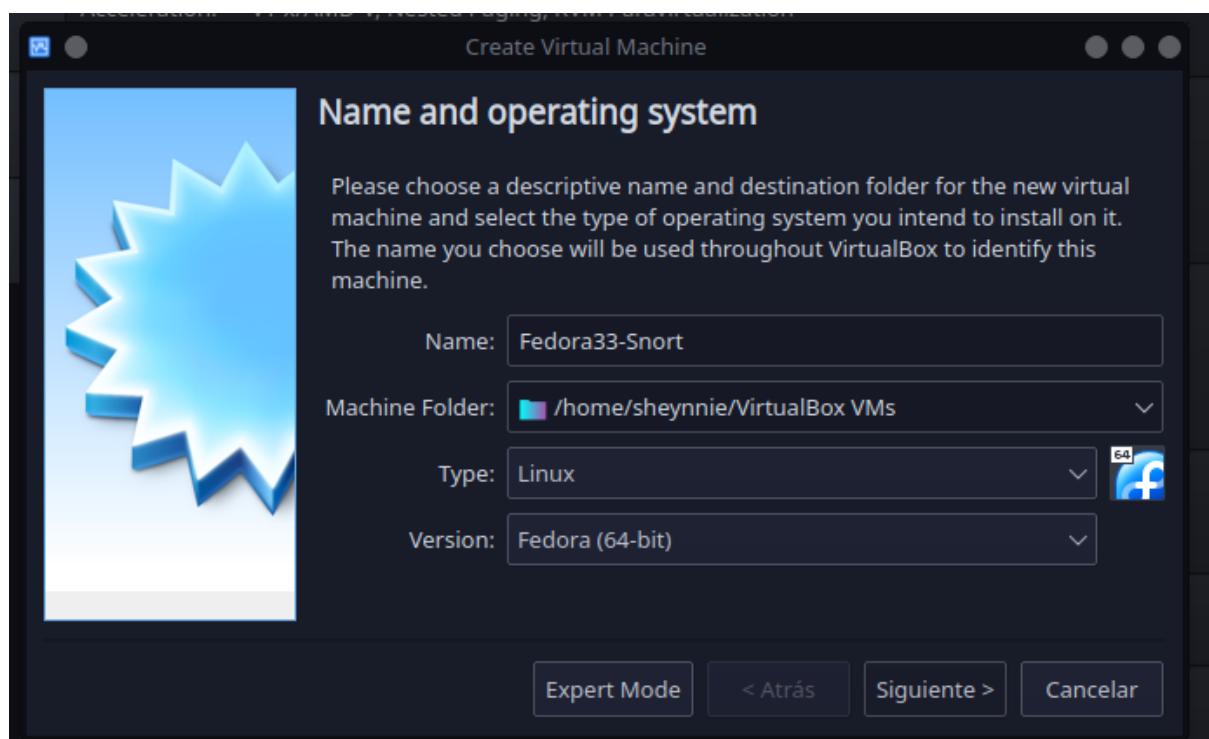
The screenshot shows the OSboxes download page for Fedora 33. It has tabs for "VirtualBox", "VMware" (selected), and "Info". Below the tabs, it says "Workstation". It lists a "VirtualBox (VDI) 64bit" option with a "Download" button, indicating a size of 2.1GB. A SHA256 checksum is provided: EE2C9CBDA274F30E8EC3ED49FECA89F0896DC15A617DBCD0DCA2. There are navigation arrows at the bottom of the list.

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**

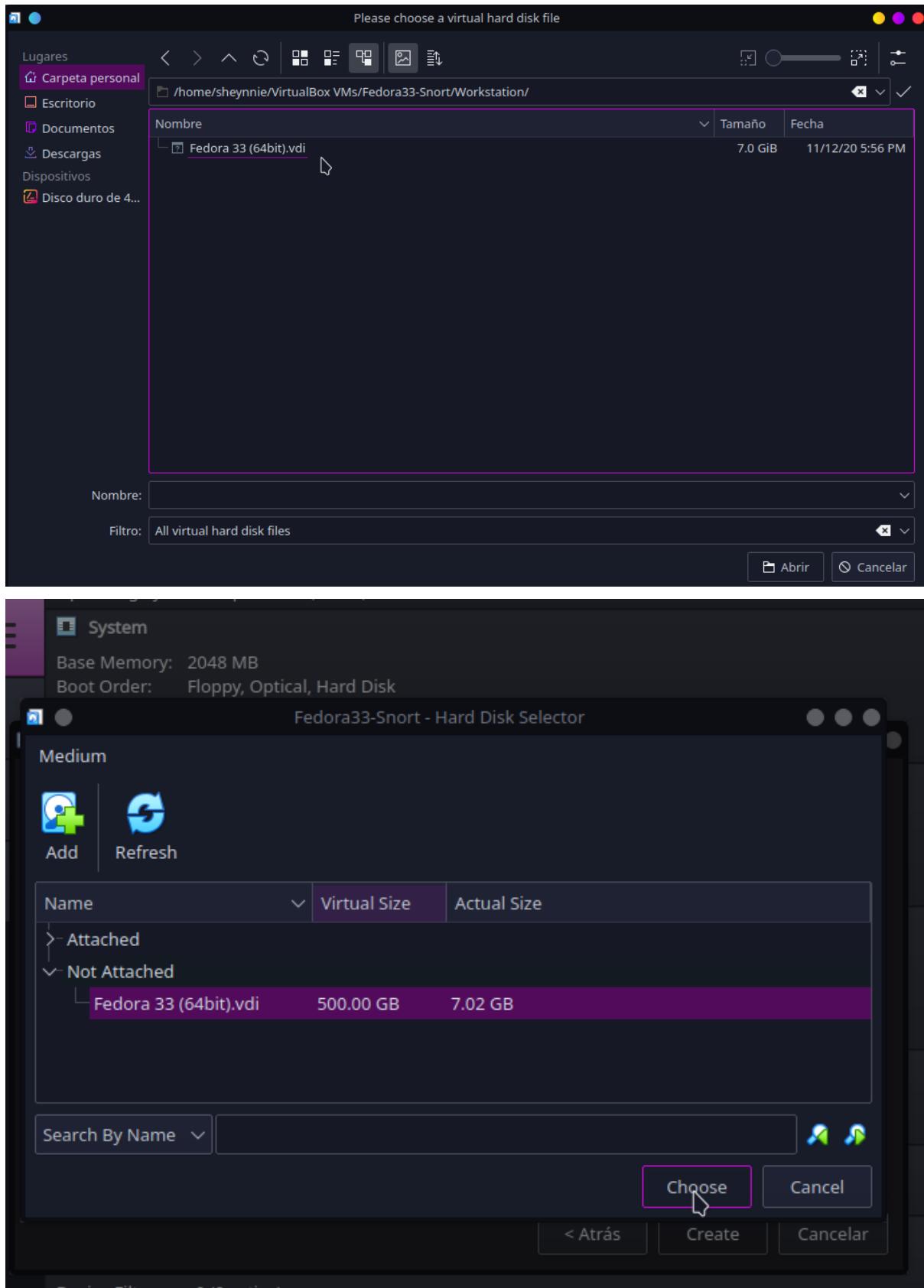
Una vez descargadas las máquinas virtuales de OSboxes, descomprimir cada una en un subdirectorio:

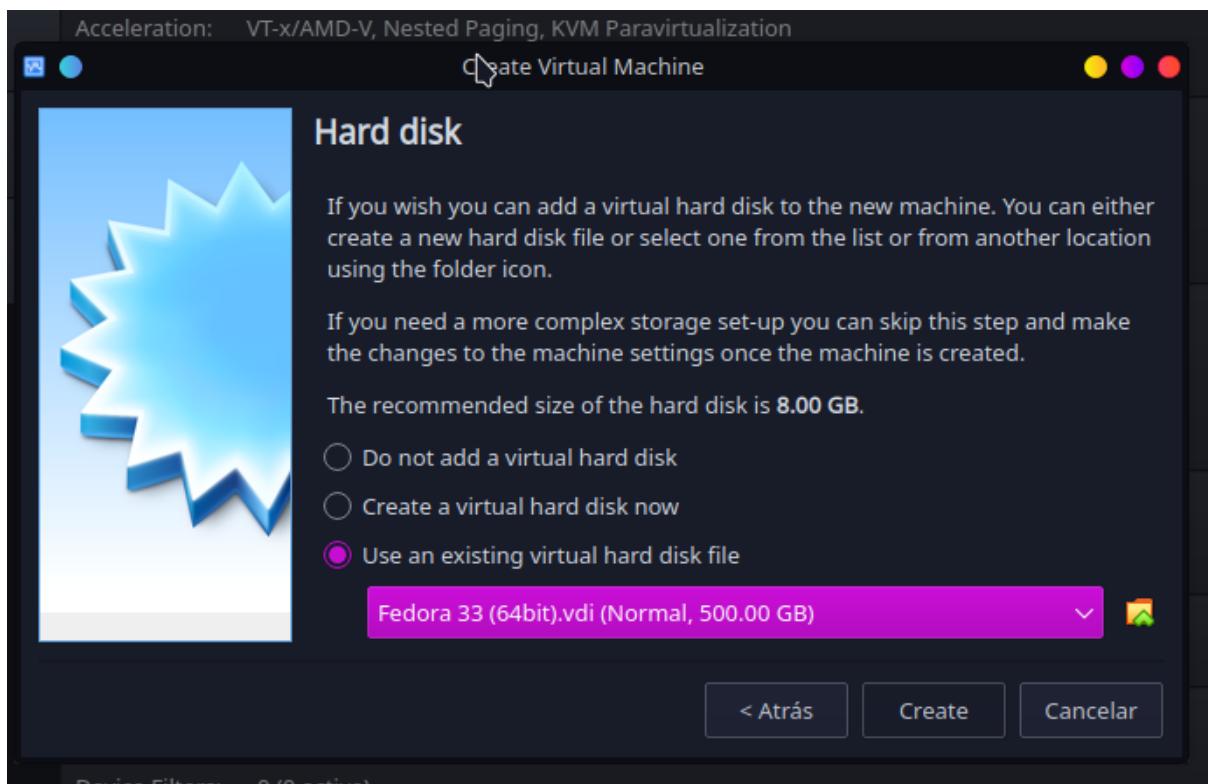
```
localhost:/home/sheynnie/Downloads # p7zip -d Fedora-33-Workstation-VB_64bit.7z
```

Abrir virtualbox y crear 2 maquinas virtuales con las siguientes características:
Mínimo 2048 mb de memoria



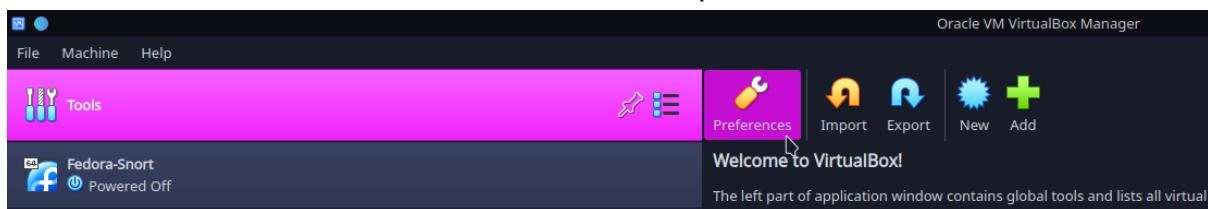
Asignar el archivo .vdi previamente extraído como disco duro existente de la máquina:



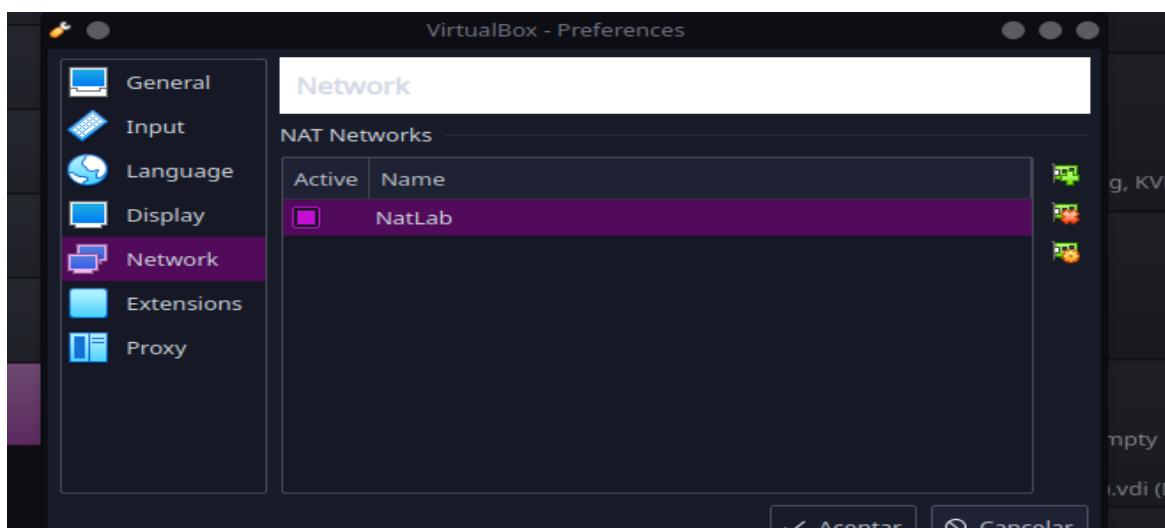


Dar click al botón crear para finalizar.

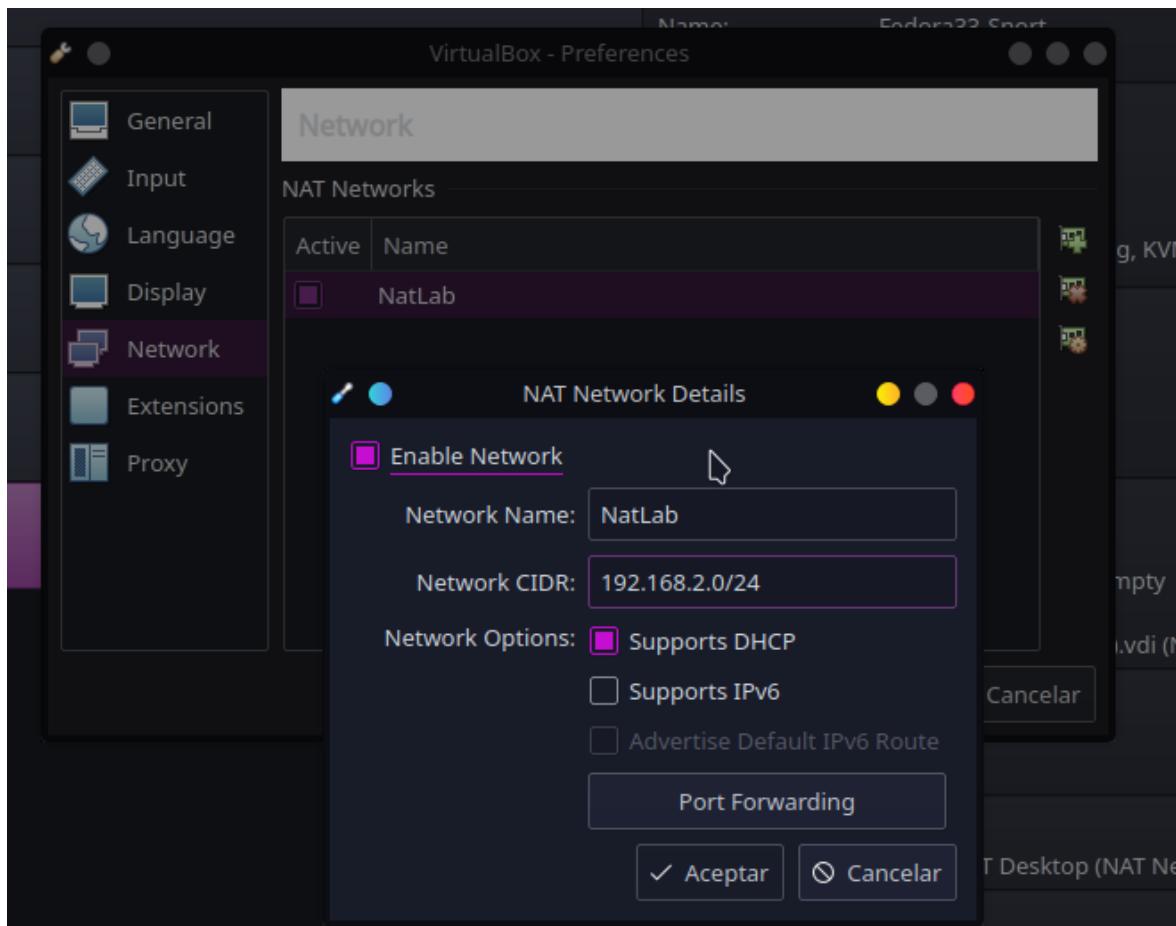
Una vez creada, desde virtualbox seleccione la opción herramientas -> Preferencias



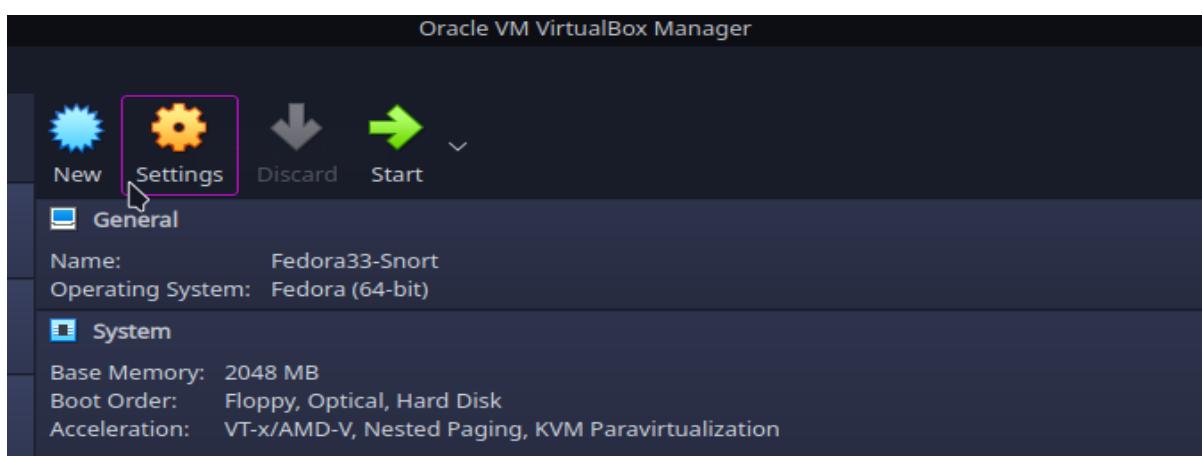
Seleccionar la opción “Red” y dar click al botón con símbolo “+” para añadir una nueva red.

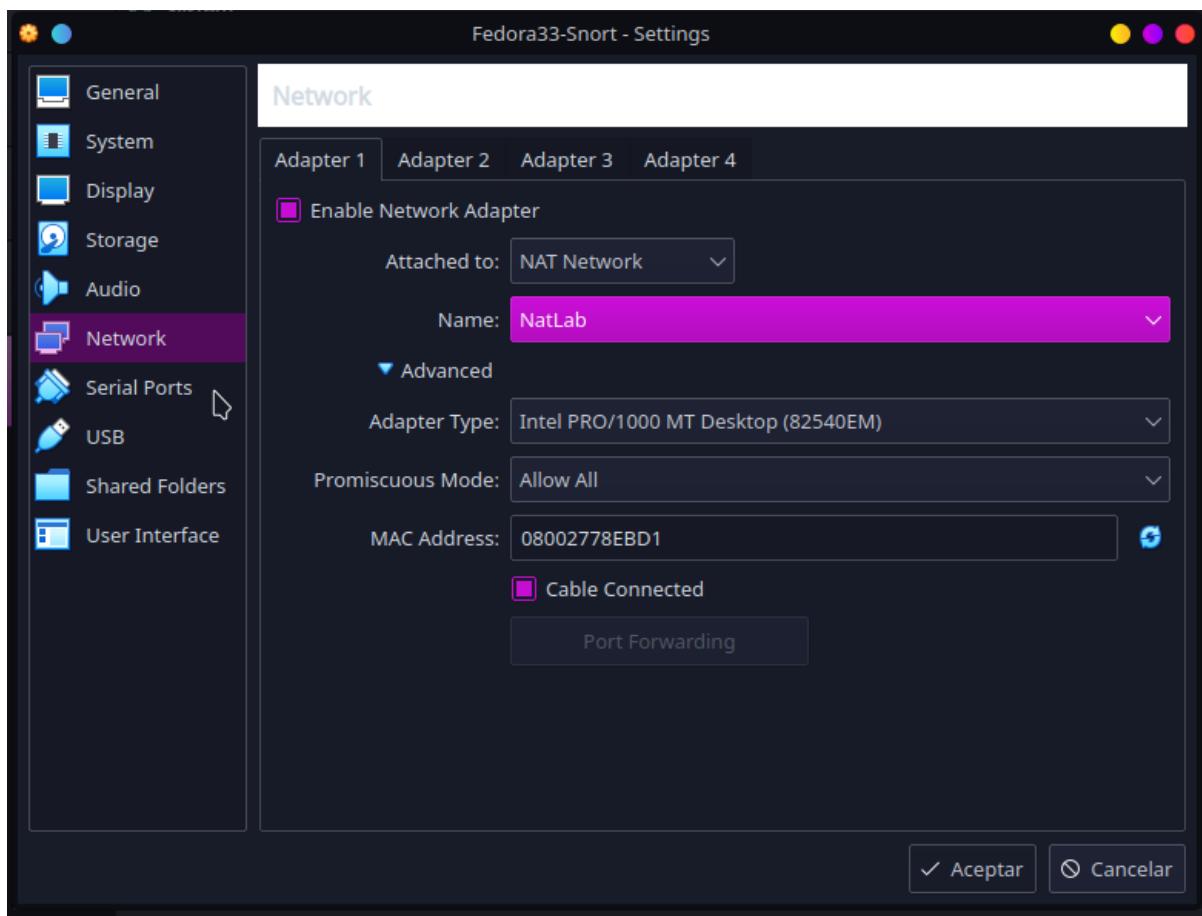


Nota: Si queremos ver los detalles como rango de IP con el cual se creó la red NAT podemos visualizarlo dandole click al botón con el símbolo de configuración (A pesar de que no es necesario editar ningún parámetro en este apartado se recomienda desmarcar el soporte para DHCP de forma que para pruebas posteriores no nos cambie la IP de las máquinas).



Una vez creada la red NAT, nos dirigimos a cada una de las máquinas creadas para enlazarlas a esta red. Seleccione la máquina y de click al botón configuración->red y en la sección del primer adaptador, asigne la “red NAT” creada previamente:





Con estas configuraciones ya podemos iniciar las máquinas.

Notas:

- Ambas máquinas deben quedar bajo la misma red NAT para que puedan visualizarse entre sí.
- 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 las máquinas:
 - Para actualizar Fedora, desde la terminal ingrese el comando: “**sudo yum update**”.
 - Para OpenSUSE, desde la terminal ingrese el comando “**sudo zypper update**”.

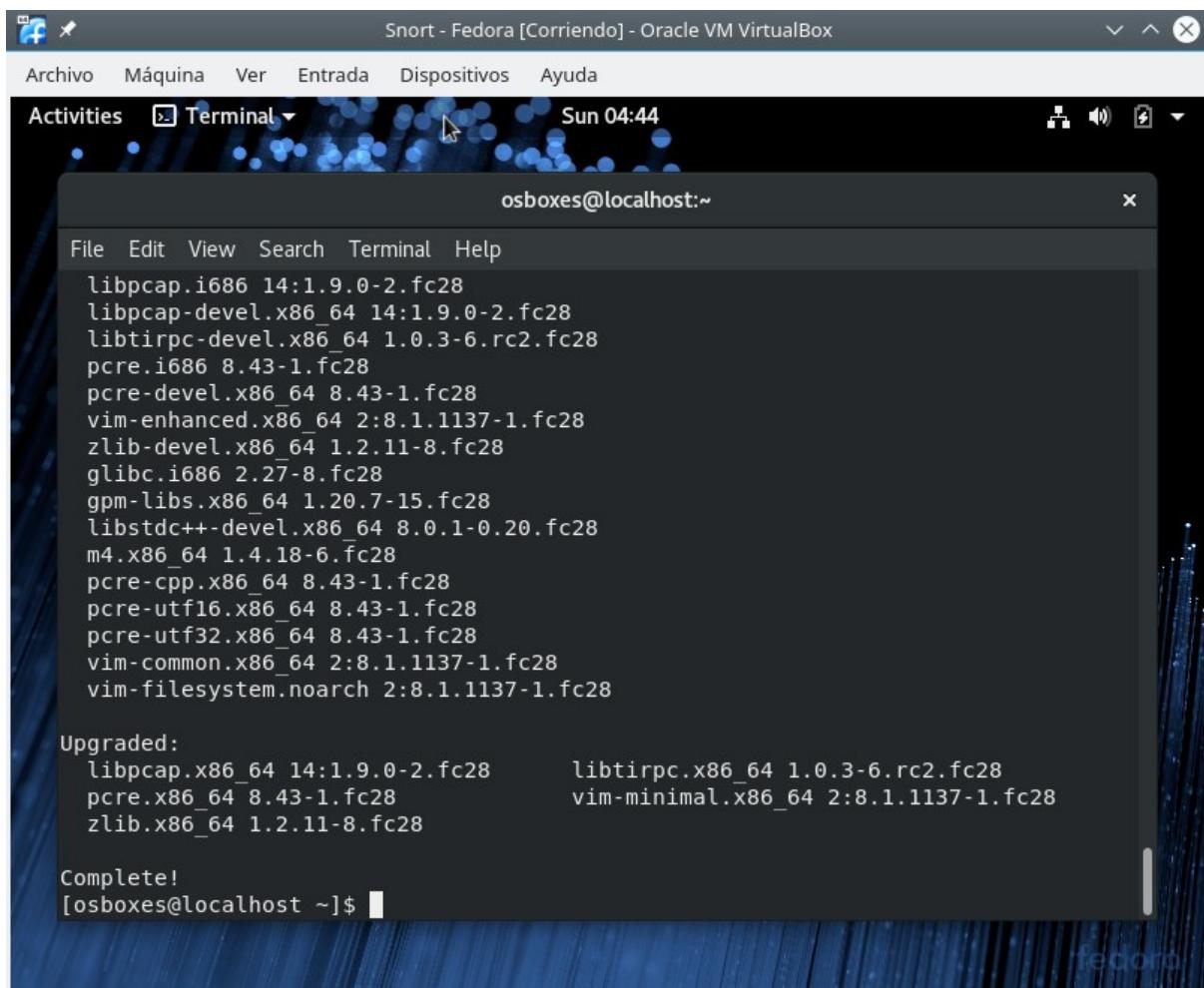
Parte 1 - Configuración del IDS Snort en la máquina Fedora.

Instalar las librerías necesarias desde la terminal:

Aplicaciones → buscador→terminal

Ingresamos lo siguiente para instalar las librerías:

```
sudo yum install pcre.* gcc git vim flex bison libpcap.* libdnet.* zlib.* libtirpc.* luajit autoconf openssl libtool perl lwp.* perl-libwww-perl perl-GD perl-Crypt-SSLeay perl-LWP-Protocol-https tcpdump zlib-devel libpcap-devel pcre-devel libdnet-devel libtirpc-devel
```



```
Snort - Fedora [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Activities Terminal Sun 04:44
osboxes@localhost:~ x
File Edit View Search Terminal Help
libpcap.i686 14:1.9.0-2.fc28
libpcap-devel.x86_64 14:1.9.0-2.fc28
libtirpc-devel.x86_64 1.0.3-6.rc2.fc28
pcre.i686 8.43-1.fc28
pcre-devel.x86_64 8.43-1.fc28
vim-enhanced.x86_64 2:8.1.1137-1.fc28
zlib-devel.x86_64 1.2.11-8.fc28
glibc.i686 2.27-8.fc28
gpm-libs.x86_64 1.20.7-15.fc28
libstdc++-devel.x86_64 8.0.1-0.20.fc28
m4.x86_64 1.4.18-6.fc28
pcre-cpp.x86_64 8.43-1.fc28
pcre-utf16.x86_64 8.43-1.fc28
pcre-utf32.x86_64 8.43-1.fc28
vim-common.x86_64 2:8.1.1137-1.fc28
vim-filesystem.noarch 2:8.1.1137-1.fc28

Upgraded:
libpcap.x86_64 14:1.9.0-2.fc28      libtirpc.x86_64 1.0.3-6.rc2.fc28
pcre.x86_64 8.43-1.fc28            vim-minimal.x86_64 2:8.1.1137-1.fc28
zlib.x86_64 1.2.11-8.fc28

Complete!
[osboxes@localhost ~]$
```

Nota: Los entornos linux son Case sensitive por lo que se deben respetar las mayúsculas y minúsculas.

Ahora, realizaremos el clon del repositorio github de Snorter en nuestro entorno local con el comando: **git clone https://github.com/joanbono/Snorter.git**

```
[osboxes@localhost ~]$ git clone https://github.com/joanbono/Snorter.git
Cloning into 'Snorter'...
remote: Enumerating objects: 526, done.
remote: Total 526 (delta 0), reused 0 (delta 0), pack-reused 526
Receiving objects: 100% (526/526), 6.52 MiB | 6.63 MiB/s, done.
Resolving deltas: 100% (214/214), done.
[osboxes@localhost ~]$ cd Snorter
[osboxes@localhost Snorter]$ ls
config.yml  doc  img  LICENSE  package-lock.json  README.md  src
[osboxes@localhost Snorter]$ cd src
[osboxes@localhost src]$ ls
SnorterDock  Snorter.sh  Snorter_Ubuntu-14.04.sh
[osboxes@localhost src]$
```

Posicionarse en la carpeta Snorter/src y editar la línea 53 del archivo snorter.sh con el comando ***sudo vim Snorter.sh***. Una vez en el editor agregaremos el parámetro ***--disable-open-appid*** como se muestra a continuación:

```
File Edit View Search Terminal Help
mv $HOME/snort_src/daq-* $HOME/snort_src/daq
cd $HOME/snort_src/daq
./configure && make && sudo make install
echo -ne "\n\t${GREEN}[+] INFO:${NOCOLOR} ${BOLD}$DAQ${NOCOLOR} installed successfully.\n\n"
#Installing SNORT
cd $HOME/snort_src
echo -ne "\n\t${CYAN}[i] INFO:${NOCOLOR} Installing ${BOLD}$SNORT${NOCOLOR}\n".
tar xvzf $SNORT.tar.gz > /dev/null 2>&1
rm -r *.tar.gz > /dev/null 2>&1
mv snort-* snort
cd snort
./configure --enable-sourcefire --disable-open-appid && make && sudo make install
echo -ne "\n\t${GREEN}[+] INFO:${NOCOLOR} ${BOLD}$SNORT${NOCOLOR} installed successfully.\n\n"
cd ..
sudo ldconfig
sudo ln -s /usr/local/bin/snort /usr/sbin/snort
-- INSERT --
53,54-61 6%
```

Para salir del editor presionamos la tecla escape ***ESC*** seguido de ***:wq!***

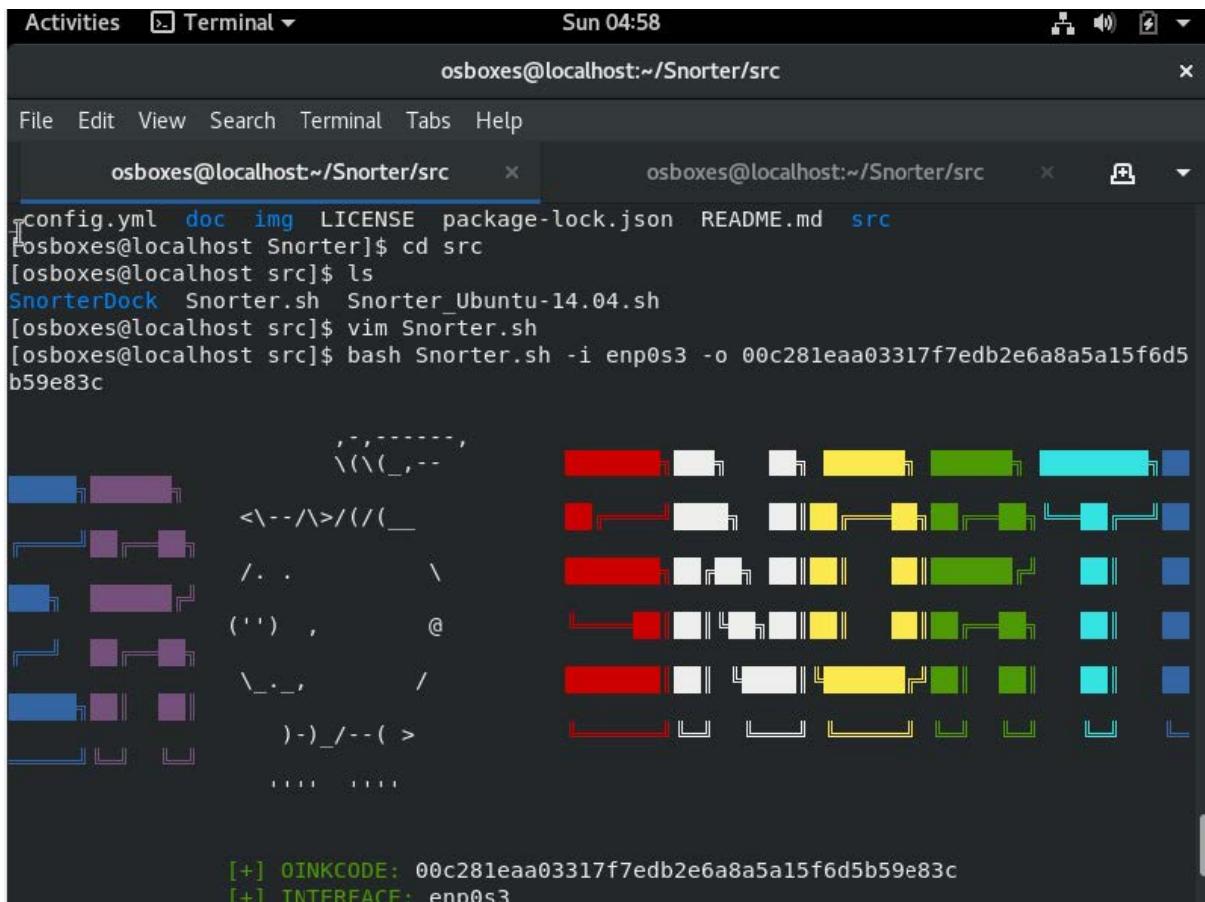
Luego, ejecutamos los siguientes comandos para asociar los comandos de instalación de Fedora en el script de snorter:

```
sed -i 's/apt-get/yum/g' Snorter.sh
sed -i 's/--force-yes//g' Snorter.sh
```

```
[osboxes@localhost ~]$ cd Snorter/src
[osboxes@localhost src]$ ls
SnorterDock  Snorter.sh  Snorter_Ubuntu-14.04.sh
[osboxes@localhost src]$ sed -i 's/apt-get/yum/g' Snorter.sh
[osboxes@localhost src]$
```

Una vez hecho esto ejecutamos snorter:

sudo bash Snorter.sh -o <'oinkcode'> -i 'interfaz de red'



```

Activities Terminal Sun 04:58
osboxes@localhost:~/Snorter/src
File Edit View Search Terminal Tabs Help
osboxes@localhost:~/Snorter/src osboxes@localhost:~/Snorter/src
config.yml doc img LICENSE package-lock.json README.md src
[osboxes@localhost Snorter]$ cd src
[osboxes@localhost src]$ ls
SnorterDock Snorter.sh Snorter_Ubuntu-14.04.sh
[osboxes@localhost src]$ vim Snorter.sh
[osboxes@localhost src]$ bash Snorter.sh -i enp0s3 -o 00c281eaa03317f7edb2e6a8a5a15f6d5
b59e83c

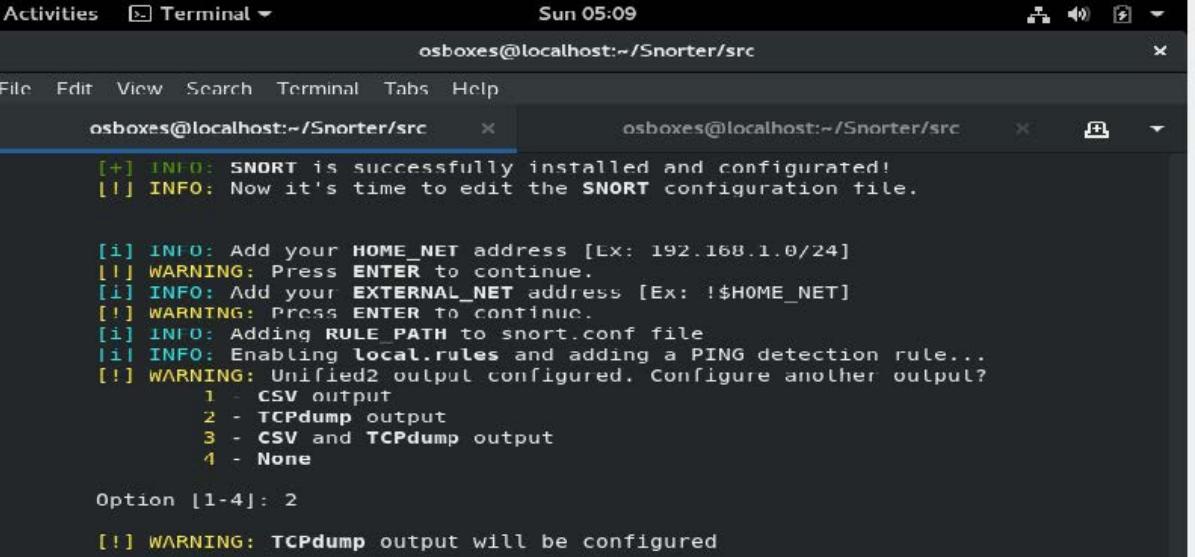
[+/-] OINKCODE: 00c281eaa03317f7edb2e6a8a5a15f6d5b59e83c
[+/-] INTERFACE: enp0s3

```

Nota:

- El comando `-o` indica que ingresaremos el oinkcode asociado, en caso de no tenerlo, ejecutemos Snorter sin este comando.
- Es recomendable ejecutar snorter con un oinkcode generado al registrarnos en Snort: https://snort.org/users/sign_up. El oinkcode es una llave asociada a nuestro registro que funciona como api para descargar las reglas de snort. Para el taller no es necesario el oinkcode, se utilizarán las reglas comunitarias.
- Con el comando `-i` indicamos cuál será la interfaz de red sobre la cual queremos que Snort realice el monitoreo (Por ejemplo: `eth1`, `enp0s3`). Puedes ver tus interfaces de red desde la terminal ejecutando el comando ***ip address***.

Una vez se realice la revisión y descarga de librerías se presentará la interfaz para configuración:



```

Activities Terminal ▾ Sun 05:09
osboxes@localhost:~/Snorter/src

File Edit View Search Terminal Tabs Help
osboxes@localhost:~/Snorter/src x osboxes@localhost:~/Snorter/src x

[+] INFO: SNORT is successfully installed and configurated!
[!] INFO: Now it's time to edit the SNORT configuration file.

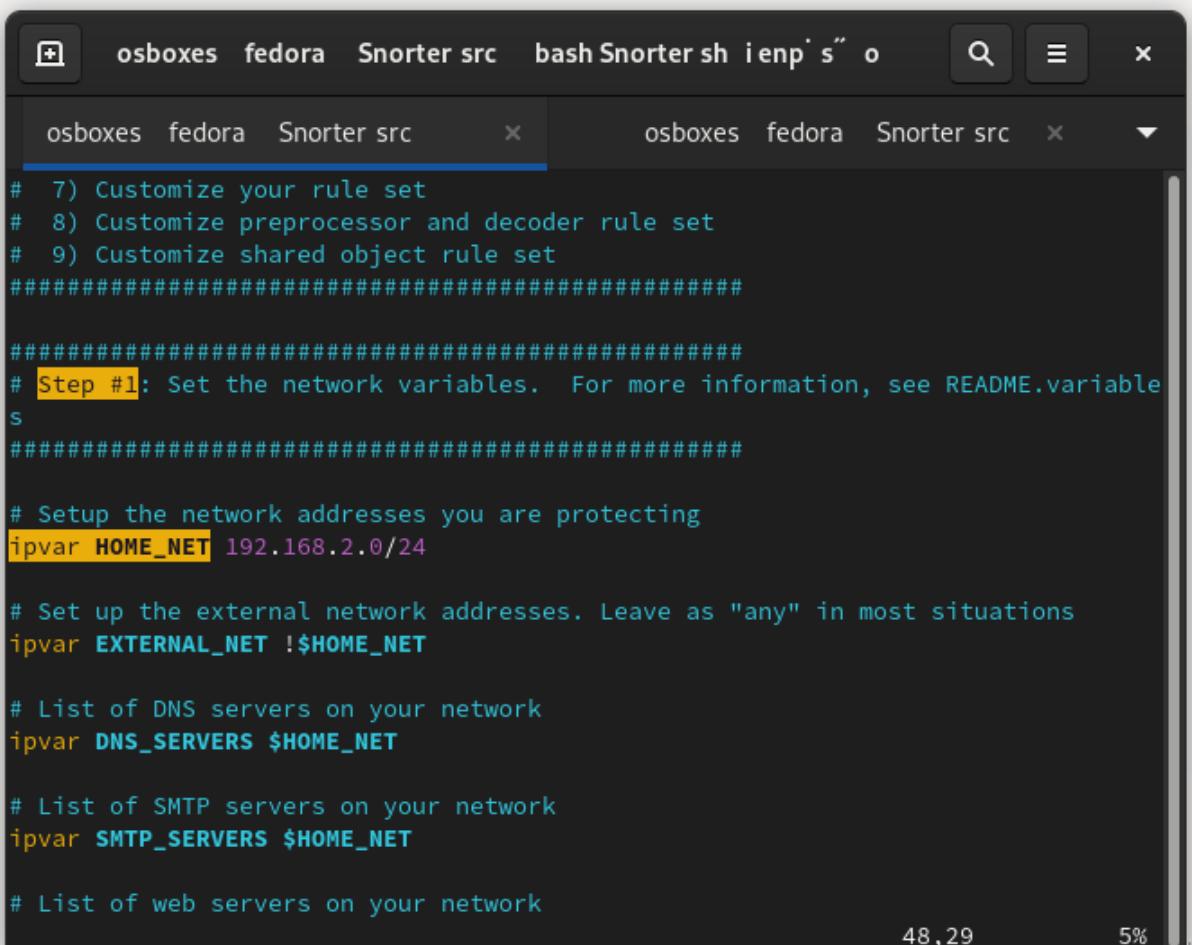
[i] INFO: Add your HOME_NET address [Ex: 192.168.1.0/24]
[!] WARNING: Press ENTER to continue.
[i] INFO: Add your EXTERNAL_NET address [Ex: !$HOME_NET]
[!] WARNING: Press ENTER to continue.
[i] INFO: Adding RULE_PATH to snort.conf file
[i] INFO: Enabling local.rules and adding a PING detection rule...
[!] WARNING: Unified2 output configured. Configure another output?
    1 - CSV output
    2 - TCPdump output
    3 - CSV and TCPdump output
    4 - None

Option [1-4]: 2

[!] WARNING: TCPdump output will be configured

```

Primero, para la configuración de snort se deben modificar los valores de la red interna (la que indicamos al crear la red NAT de virtualbox) y la red externa (Todo aquello que no sea la red interna).



```

osboxes fedora Snorter src bash Snorter.sh i enp1s0
osboxes fedora Snorter src

# 7) Customize your rule set
# 8) Customize preprocessor and decoder rule set
# 9) Customize shared object rule set
#####
# Step #1: Set the network variables. For more information, see README.variable
#
#####

# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.2.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET

# List of web servers on your network

```

48,29

5%

Recuerda salir del editor con la tecla escape **ESC** seguido de :wq!

Para la segunda pregunta de la configuración elegimos la opción **2** (cualquiera de las 3 primeras es válida).

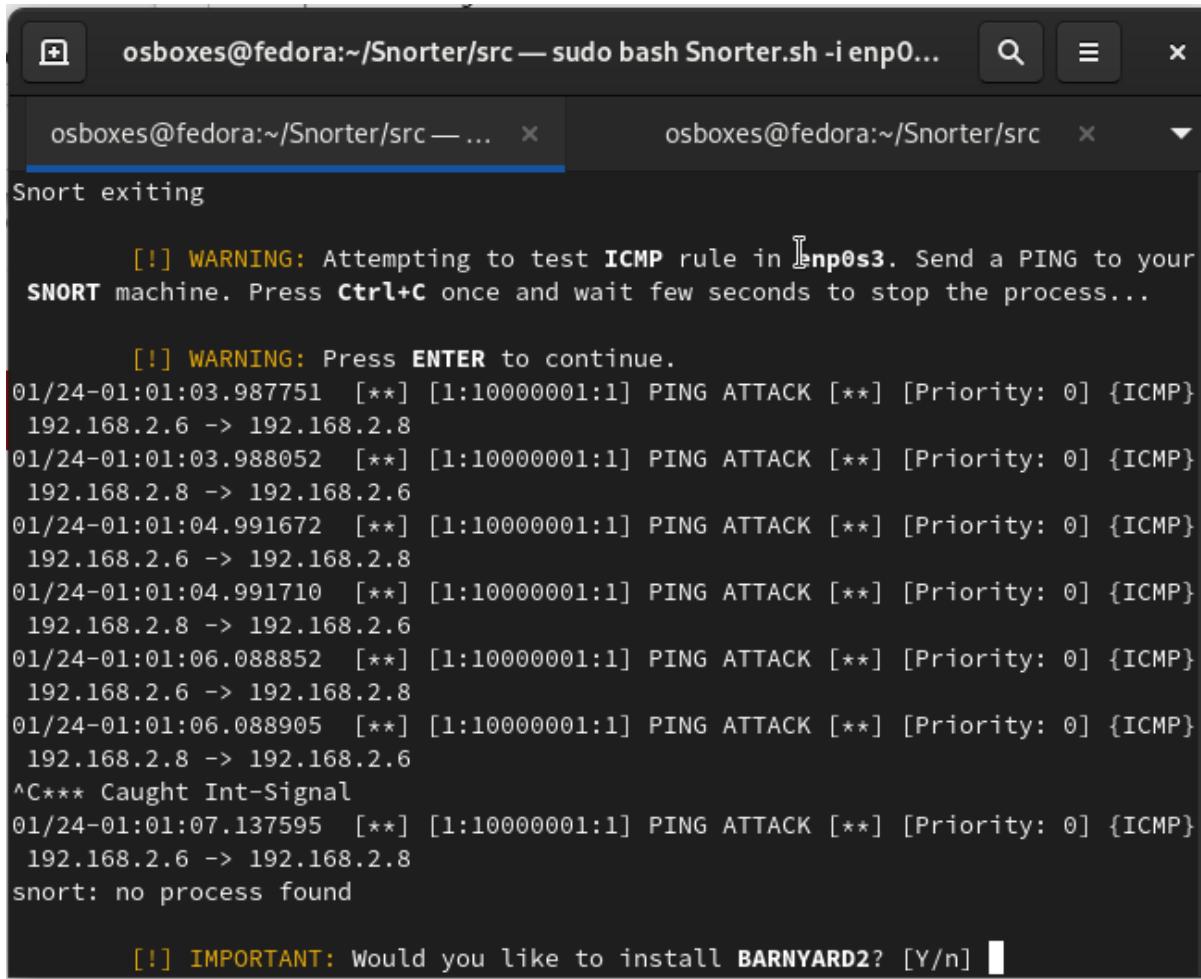
```
[!] INFO: Enabling local.rules and adding a PING detection rule...
[!] WARNING: Unified2 output configured. Configure another output?
  1 - CSV output
  2 - TCPdump output
  3 - CSV and TCPdump output
  4 - None

Option [1-4]: 2

[!] WARNING: TCPdump output will be configured
```

Después de esto, se ejecutará Snort en modo de prueba para validar la configuración.

Anteriormente Snorter escribió una regla en el archivo de reglas locales que alertará cualquier tráfico ICMP generado, la que usará en el proceso de prueba de snort.



The screenshot shows a terminal window with two tabs. The active tab displays Snort test results:

```
osboxes@fedora:~/Snorter/src — sudo bash Snorter.sh -i enp0... osboxes@fedora:~/Snorter/src — ... x osboxes@fedora:~/Snorter/src — ... x
```

Snort exiting

```
[!] WARNING: Attempting to test ICMP rule in [enp0s3]. Send a PING to your SNORT machine. Press Ctrl+C once and wait few seconds to stop the process...

[!] WARNING: Press ENTER to continue.
01/24/01:01:03.987751  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.6 -> 192.168.2.8
01/24/01:01:03.988052  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.8 -> 192.168.2.6
01/24/01:01:04.991672  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.6 -> 192.168.2.8
01/24/01:01:04.991710  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.8 -> 192.168.2.6
01/24/01:01:06.088852  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.6 -> 192.168.2.8
01/24/01:01:06.088905  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.8 -> 192.168.2.6
^C*** Caught Int-Signal
01/24/01:01:07.137595  [**] [1:10000001:1] PING ATTACK [**] [Priority: 0] {ICMP}
  192.168.2.6 -> 192.168.2.8
snort: no process found
```

[!] IMPORTANT: Would you like to install BARNYARD2? [Y/n]

Nota: En este apartado, podemos probar snort lanzando un comando ping desde otra máquina que esté dentro de la red NAT hacia la máquina con snort.

Presionamos enter y veremos las alertas generadas en tiempo real. Luego podemos salir de este modo presionando **Ctrl+C** para continuar con las configuraciones.

Ingresamos la opción **n** para no instalar Barnyard2 (fuera de alcance del taller ya que requiere otras configuraciones y dependencias).

Ingresamos la opción **y** para instalar Pulledpork.

```
[!] IMPORTANT: Would you like to install BARNYARD2? [Y/n] n
[i] INFO: BARNYARD2 won't be installed.

[!] IMPORTANT: Would you like to install PULLEDPOORK? [Y/n] y
[i] INFO: Installing dependencies.

usage: yum install [-c [config file]] [-q] [-v] [--version]
                   [--installroot [path]] [--nодocs] [--nopugins]
                   [--enableplugin [plugin]] [--disableplugin [plugin]]
                   [--releasever RELEASEVER] [--setopt SETOPTS]
                   [--skip-broken] [-h] [--allowerasing] [-b | --nobest] [-C]
                   [-R [minutes]] [-d [debug level]] [--debugsolver]
                   [--showduplicates] [-e ERRORLEVEL] [--obsoletes]
                   [--rpmverbosity [debug level name]] [-y] [--assumeno]
                   [--enablerepo [repo]] [--disablerepo [repo] | --repo
                   [repo]] [--enable | --disable] [-x [package]]
                   [--disableexcludes [repo]] [--repofrompath [repo, path]]
                   [--noautoremove] [--nogpgcheck] [--color COLOR] [--refresh]
                   [-4] [-6] [--destdir DESTDIR] [--downloadonly]
```

Seleccionamos la opción **n** para no habilitar las reglas de Emerging Threats y la opción **y** para la creación del servicio snort y también para la opción de descargar nuevas reglas utilizando Pulledpork.

```
[i] INFO: Editing pulledpork.conf settings...
[!] IMPORTANT: Would you like to create a service snort? [Y/n] y
Created symlink /etc/systemd/system/multi-user.target.wants/snort.service → /usr
/lib/systemd/system/snort.service.
Failed to enable unit: Unit file barnyard2.service does not exist.

[i] INFO: Now you can run sudo systemctl {start|stop|status} snort .

[!] IMPORTANT: Would you like to download new rules using PULLEDPOORK? [Y
/n] y
Option H requires an argument

https://github.com/shirkdog/pulledpork
`----, \   )
`---=\\ / PulledPork v0.8.0 - The only positive thing to come out of 2
```

Instalamos websnort con la opción **y**

```
Rule Stats...
    New:-----41791
    Deleted:---0
    Enabled Rules:----9899
    Dropped Rules:----0
    Disabled Rules:---31892
    Total Rules:-----41791
IP Blocklist Stats...
    Total IPs:----791

Done
Please review /var/log/sid_changes.log for additional details
Fly Piggy Fly!

[!] IMPORTANT: Would you like to install WEBSNORT for PCAP Analysis? [Y/n] ■
```

Habilitamos la descarga de las reglas comunitarias y luego reiniciamos la máquina virtual para aplicar todos los cambios con la opción **y**.

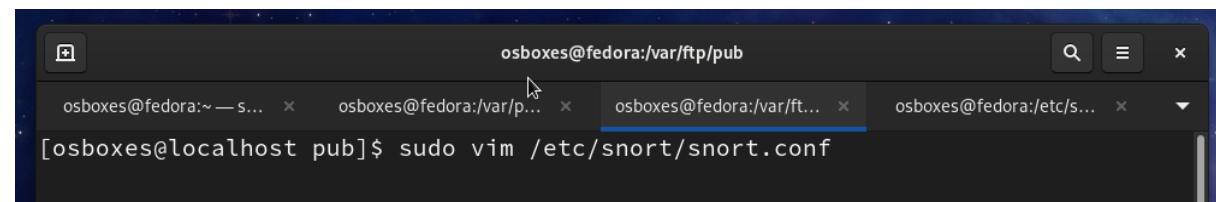
```
[!] IMPORTANT: Would you like to enable Emerging Threats and Community rules for detection? [Y/n] y
ls: cannot access '/etc/snort/rules/emerging-*.rules': No such file or directory
Failed to restart barnyard2.service: Unit barnyard2.service not found.

[+] SUCCESS: Emerging Threats and Community rules enabled

[!] IMPORTANT: Would you like to REBOOT now? [Y/n]
```

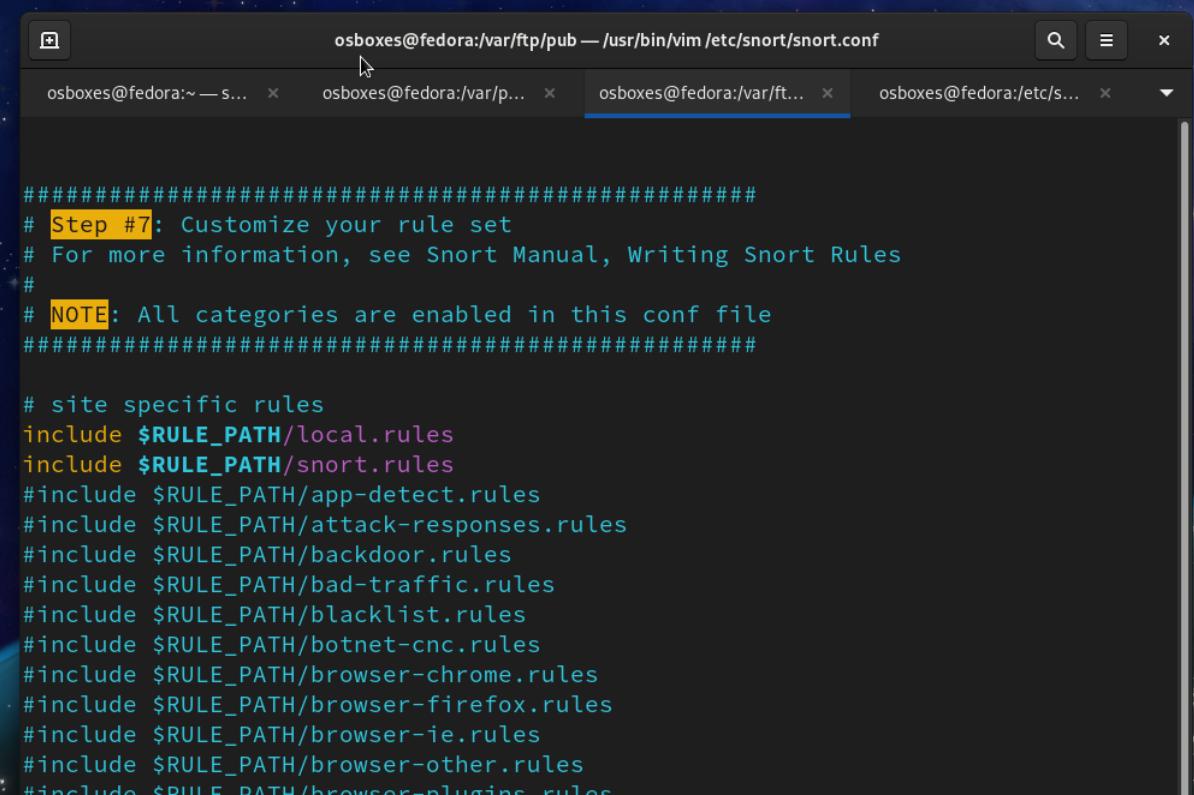
Una vez reiniciamos, nos dirigimos al archivo de configuración de snort para incluir la ruta al archivo de reglas de snort:

sudo vim /etc/snort/snort.conf



```
osboxes@fedora:~— s... x osboxes@fedora:/var/p... x osboxes@fedora:/var/ft... x osboxes@fedora:/etc/s... x
[osboxes@localhost pub]$ sudo vim /etc/snort/snort.conf
```

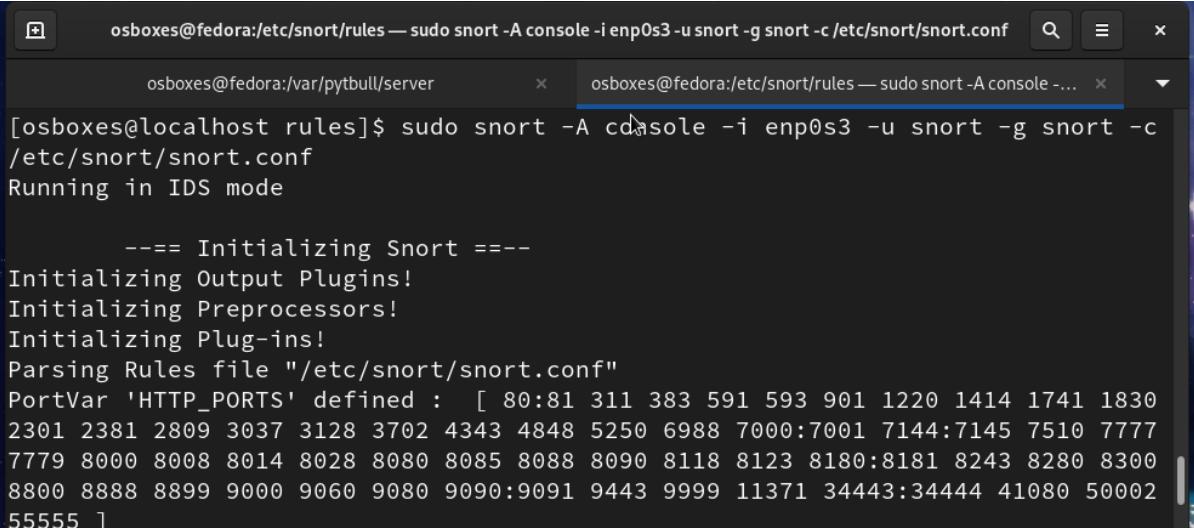
y debajo de la ruta de reglas locales (línea 547) agregamos la linea **include \$RULE_PATH/snort.rules** y bajo la línea 690 agregamos la inclusión de las reglas comunitarias: **include \$RULE_PATH/community.rules**



```
#####
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
#####

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

Ahora si, podemos ejecutar snort y dejarlo en modo escucha para ver las alertas en la consola con el comando: **sudo snort -A console -i enp0s3 -u snort -g snort -c /etc/snort/snort.conf**



```
osboxes@fedora:/etc/snort/rules — sudo snort -A console -i enp0s3 -u snort -g snort -c /etc/snort/snort.conf
```

```
[osboxes@localhost rules]$ sudo snort -A console -i enp0s3 -u snort -g snort -c /etc/snort/snort.conf
Running in IDS mode

     === Initializing Snort ===
Initializing Output Plugins!
Initializing Preprocessors!
Initializing Plug-ins!
Parsing Rules file "/etc/snort/snort.conf"
PortVar 'HTTP_PORTS' defined : [ 80:81 311 383 591 593 901 1220 1414 1741 1830
2301 2381 2809 3037 3128 3702 4343 4848 5250 6988 7000:7001 7144:7145 7510 7777
7779 8000 8008 8014 8028 8080 8085 8088 8090 8118 8123 8180:8181 8243 8280 8300
8800 8888 8899 9000 9060 9080 9090:9091 9443 9999 11371 34443:34444 41080 50002
55555 ]
```

```
Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_S7COMMPLUS Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Commencing packet processing (pid=8535)
```

Nota: los archivos de reglas de snort los encuentras en la ruta: /etc/snort/rules.



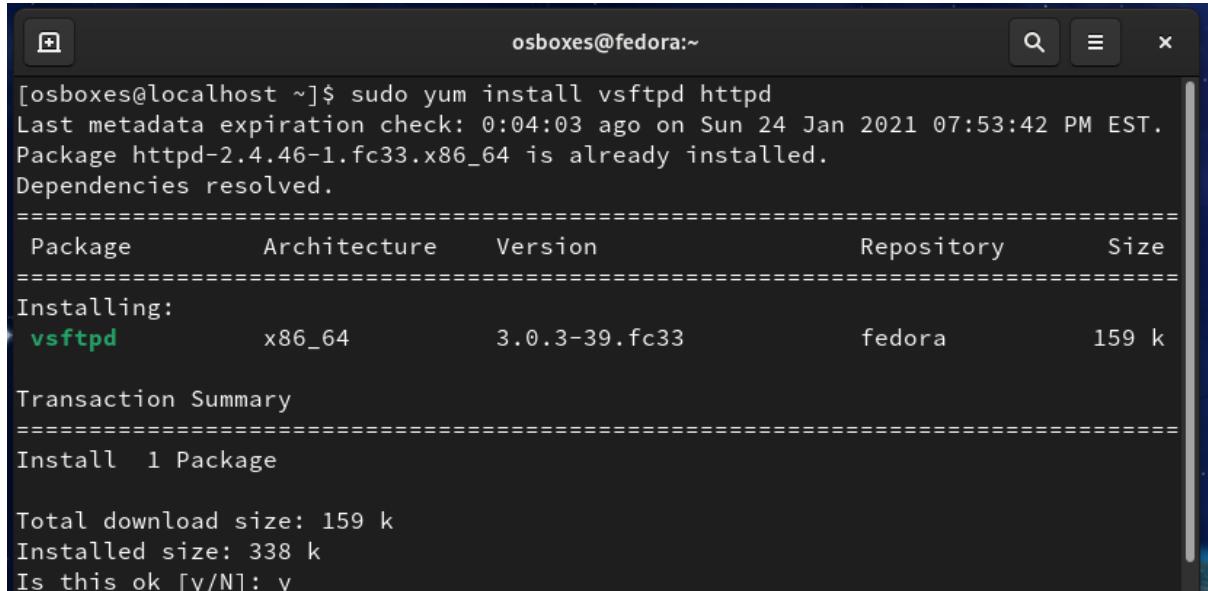
```
osboxes@fedora:var/pytbull/server
osboxes@fedora:/etc/snort/rules — /usr/bin/vim community.rules

alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"FILE-IDENTIFY JPEG file download request"; flow:to_server,established; content:".jpg"; fast_pattern:only; http_uri; pcre:"/\x2ejpg([\?\x5c\x2f]|$)/smiU"; flowbits:set,file.jpeg; flowbits:noalert; metadata:policy balanced-ips alert, policy max-detect-ips alert, policy security-ips alert, ruleset community, service http; reference:url,en.wikipedia.org/wiki/Jpg; classtype:misc-activity; sid:16406; rev:20;)
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"FILE-IDENTIFY JPEG file download request"; flow:to_server,established; content:".jpeg"; fast_pattern:only; http_uri; pcre:"/\x2ejpeg([\?\x5c\x2f]|$)/smiU"; flowbits:set,file.jpeg; flowbits:noalert; metadata:policy balanced-ips alert, policy max-detect-ips alert, policy security-ips alert, ruleset community, service http; reference:url,en.wikipedia.org/wiki/Jpg; classtype:misc-activity; sid:16407; rev:20;)
```

Parte 2 - Configuración del servidor Pytbull en la máquina Fedora.

Nuevamente desde la consola instalamos las librerías que necesitaremos ahora para el servidor Pytbull:

sudo yum install vsftpd httpd python2



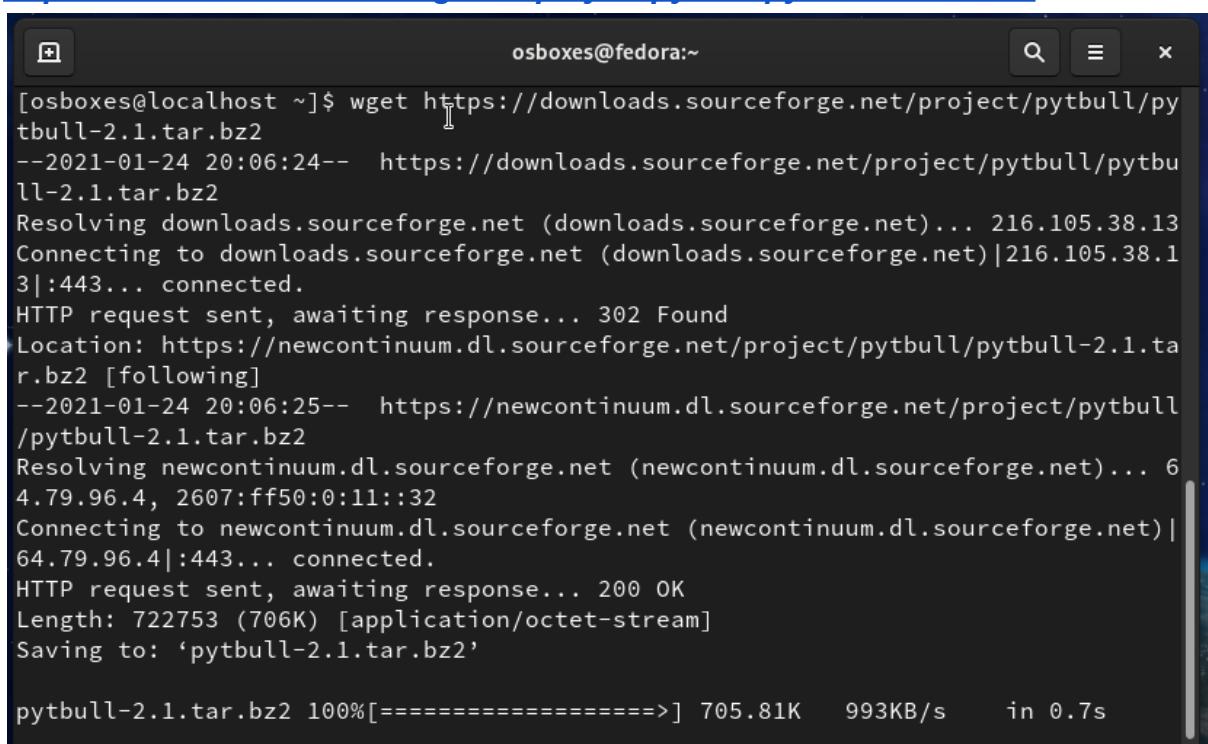
```
[osboxes@localhost ~]$ sudo yum install vsftpd httpd
Last metadata expiration check: 0:04:03 ago on Sun 24 Jan 2021 07:53:42 PM EST.
Package httpd-2.4.46-1.fc33.x86_64 is already installed.
Dependencies resolved.
=====
 Package           Architecture   Version        Repository      Size
 =====
 Installing:
 vsftpd            x86_64       3.0.3-39.fc33    fedora          159 k

 Transaction Summary
 =====
 Install 1 Package

 Total download size: 159 k
 Installed size: 338 k
 Is this ok [y/N]: y
```

Descargar pytbull con el comando: ***wget***

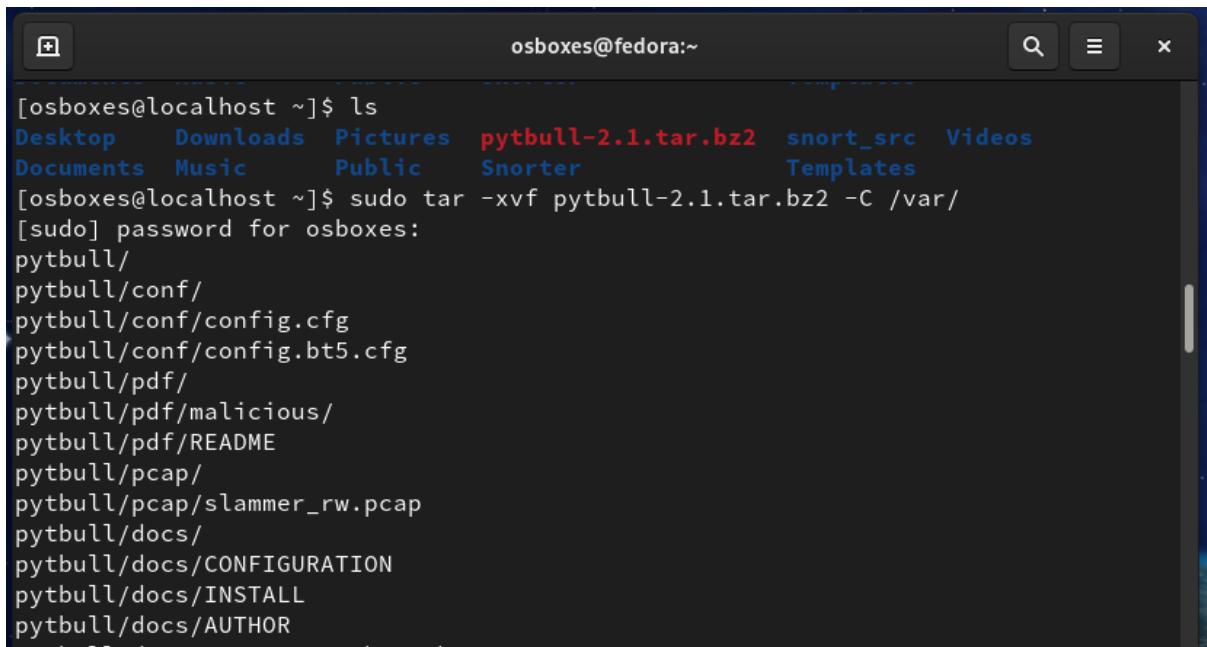
<https://downloads.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2>



```
[osboxes@localhost ~]$ wget https://downloads.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2
--2021-01-24 20:06:24-- https://downloads.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2
Resolving downloads.sourceforge.net (downloads.sourceforge.net)... 216.105.38.13
Connecting to downloads.sourceforge.net (downloads.sourceforge.net)|216.105.38.13|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://newcontinuum.dl.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2 [following]
--2021-01-24 20:06:25-- https://newcontinuum.dl.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2
Resolving newcontinuum.dl.sourceforge.net (newcontinuum.dl.sourceforge.net)... 64.79.96.4, 2607:ff50:0:11::32
Connecting to newcontinuum.dl.sourceforge.net (newcontinuum.dl.sourceforge.net)|64.79.96.4|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 722753 (706K) [application/octet-stream]
Saving to: 'pytbull-2.1.tar.bz2'

pytbull-2.1.tar.bz2 100%[=====] 705.81K  993KB/s   in 0.7s
```

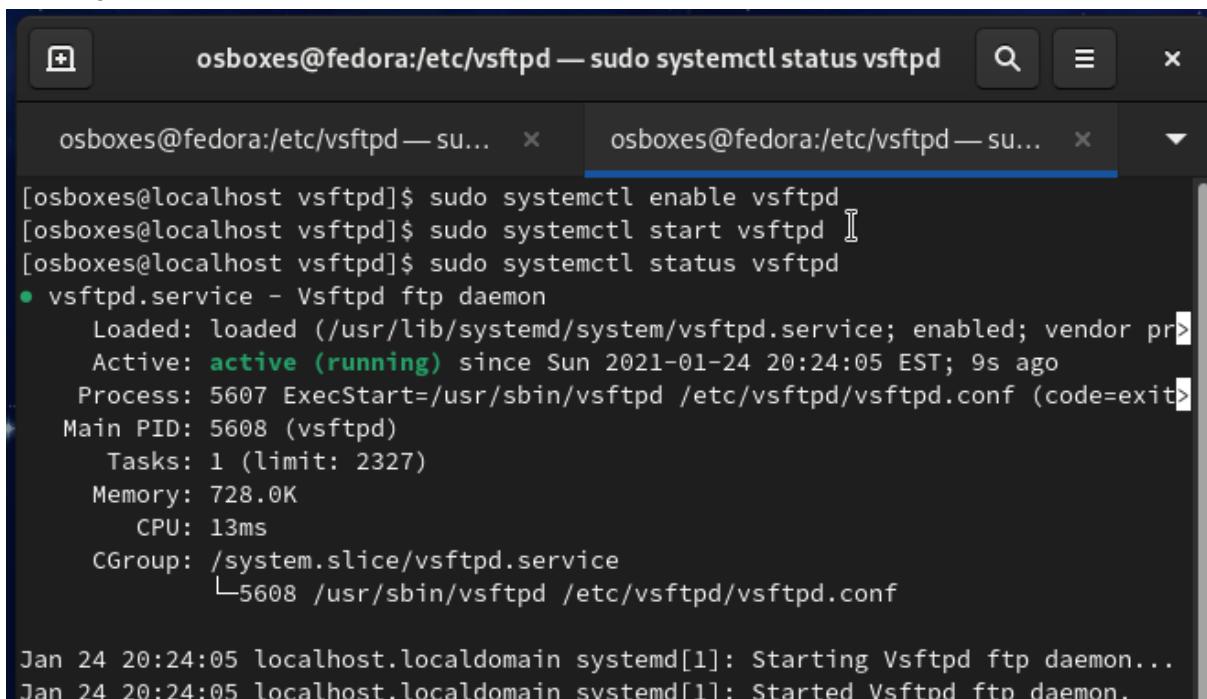
Extraer los archivos a la carpeta var: ***tar -xvf pytbull-2.1.tar.bz2 -C /var/***



```
[osboxes@localhost ~]$ ls
Desktop  Downloads  Pictures  pytbull-2.1.tar.bz2  snort_src  Videos
Documents  Music  Public  Snorter  Templates
[osboxes@localhost ~]$ sudo tar -xvf pytbull-2.1.tar.bz2 -C /var/
[sudo] password for osboxes:
pytbull/
pytbull/conf/
pytbull/conf/config.cfg
pytbull/conf/config.bt5.cfg
pytbull/pdf/
pytbull/pdf/malicious/
pytbull/pdf/README
pytbull pcap/
pytbull/pcap/slammer_rw.pcap
pytbull/docs/
pytbull/docs/CONFIGURATION
pytbull/docs/INSTALL
pytbull/docs/AUTHOR
```

Habilitar e iniciar los servicios para los servidores web (httpd), ftp (vsftpd) y ssh (sshd):

```
sudo systemctl enable vsftpd
sudo systemctl start vsftpd
sudo systemctl enable sshd
sudo systemctl start sshd
sudo systemctl enable httpd
sudo systemctl start httpd
```



```
[osboxes@localhost vsftpd]$ sudo systemctl enable vsftpd
[osboxes@localhost vsftpd]$ sudo systemctl start vsftpd
[osboxes@localhost vsftpd]$ sudo systemctl status vsftpd
● vsftpd.service - Vsftpd ftp daemon
   Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; vendor pr>
   Active: active (running) since Sun 2021-01-24 20:24:05 EST; 9s ago
     Process: 5607 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exit>
    Main PID: 5608 (vsftpd)
      Tasks: 1 (limit: 2327)
     Memory: 728.0K
        CPU: 13ms
       CGroup: /system.slice/vsftpd.service
               └─5608 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

Jan 24 20:24:05 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
Jan 24 20:24:05 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.
```

Nota: podemos verificar el status de los servicios (deben quedar activos) por ejemplo: ***sudo systemctl status httpd***

Ahora, añadimos las reglas al firewall para permitir la comunicación de los servicios:

```
sudo firewall-cmd --permanent --add-service=ftp  
sudo firewall-cmd --permanent --add-service=ssh  
sudo firewall-cmd --permanent --add-service=http  
sudo firewall-cmd --permanent --add-port=21/tcp  
sudo firewall-cmd --permanent --add-port=22/tcp  
sudo firewall-cmd --permanent --add-port=80/tcp
```

Luego reiniciamos el firewall:

```
sudo firewall-cmd --reload
```

The screenshot shows a terminal window titled "osboxes@fedora:/etc/vsftpd". It contains two tabs: "osboxes@fedora:/etc/vsftpd — su..." and "osboxes@fedora:/etc/vsftpd". The bottom tab is active. The terminal output is as follows:

```
[osboxes@localhost vsftpd]$ sudo firewall-cmd --permanent --add-service=ftp  
success  
[osboxes@localhost vsftpd]$ sudo firewall-cmd --reload  
success  
[osboxes@localhost vsftpd]$
```

Nos dirigimos a /var/ftp/pub y creamos un archivo de alerta, el cual llamaremos desde el cliente pytbull:

```
cd /var/ftp/pub  
sudo touch snort.alert
```

Nos dirigimos a la ruta /var/pytbull/server y ejecutamos el servidor de pytbull y lo dejamos en modo escucha:

```
cd /var/pytbull/server  
sudo python2 pytbull-server.py
```

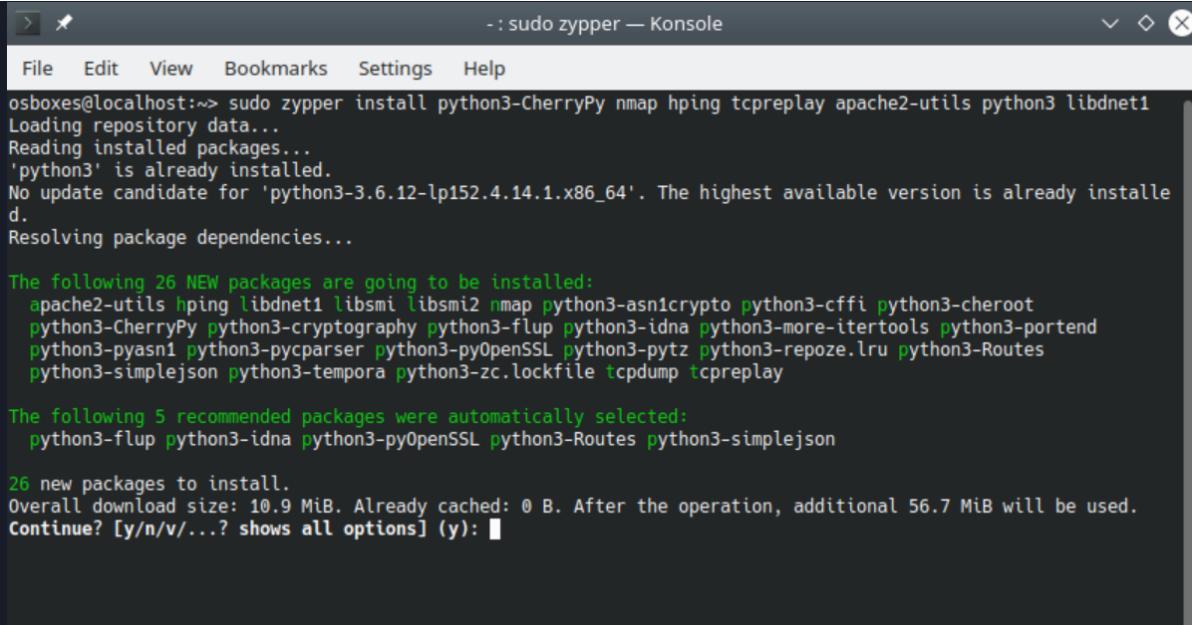
The screenshot shows a terminal window titled "osboxes@fedora:/var/pytbull/server — sudo p...". It contains two tabs: "osboxes@fedora:/var/pytbull/server — sudo p..." and "osboxes@fedora:/etc/snort/rules — sudo snort...". The bottom tab is active. The terminal output is as follows:

```
[osboxes@localhost server]$ sudo python2 pytbull-server.py  
  
 _ _ _ _ _  
| | | | | | |
| | ) | | |  
| | | | | | |  
| .--/ \__,_| \__| .--/ \__,_| \__|  
|_| | | /  
Sebastien Damase, aldeid.com  
  
Checking root privileges..... [ OK ]  
Checking port to use..... [ OK ]  
  
Server started on port: 12345  
Listening...
```

Parte 3 - Configuración del cliente Pytbull en la máquina OpenSUSE.

Instalamos las librerías necesarias:

```
sudo zypper install python2 python2-pip nmap hping tcpreplay apache2-utils  
python3 libdnet1 libssl45, libcrypto43 y libssh-devel python2-feedparser
```



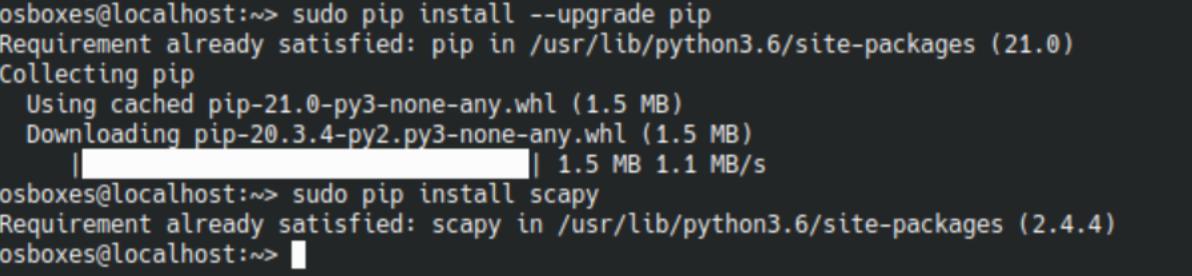
```
- : sudo zypper — Konsole  
File Edit View Bookmarks Settings Help  
osboxes@localhost:~> sudo zypper install python3-CherryPy nmap hping tcpreplay apache2-utils python3 libdnet1  
Loading repository data...  
Reading installed packages...  
'python3' is already installed.  
No update candidate for 'python3-3.6.12-lp152.4.14.1.x86_64'. The highest available version is already installed.  
Resolving package dependencies...  
  
The following 26 NEW packages are going to be installed:  
apache2-utils hping libdnet1 libsmi libsmi2 nmap python3-asn1crypto python3-cffi python3-cheroot  
python3-CherryPy python3-cryptography python3-flup python3-idna python3-more-itertools python3-portend  
python3-pasn1 python3-pycparser python3-pyOpenSSL python3-ptz python3-repoze.lru python3-Routes  
python3-simplejson python3-tempora python3-zc.lockfile tcpdump tcpreplay  
  
The following 5 recommended packages were automatically selected:  
python3-flup python3-idna python3-pyOpenSSL python3-Routes python3-simplejson  
  
26 new packages to install.  
Overall download size: 10.9 MiB. Already cached: 0 B. After the operation, additional 56.7 MiB will be used.  
Continue? [y/n/v/...? shows all options] (y): ■
```

Ahora haremos un upgrade al comando pip para luego instalar las librerías scapy y cherrypy:

```
sudo pip install --upgrade pip
```

```
sudo pip install scapy
```

```
sudo pip install cherrypy
```



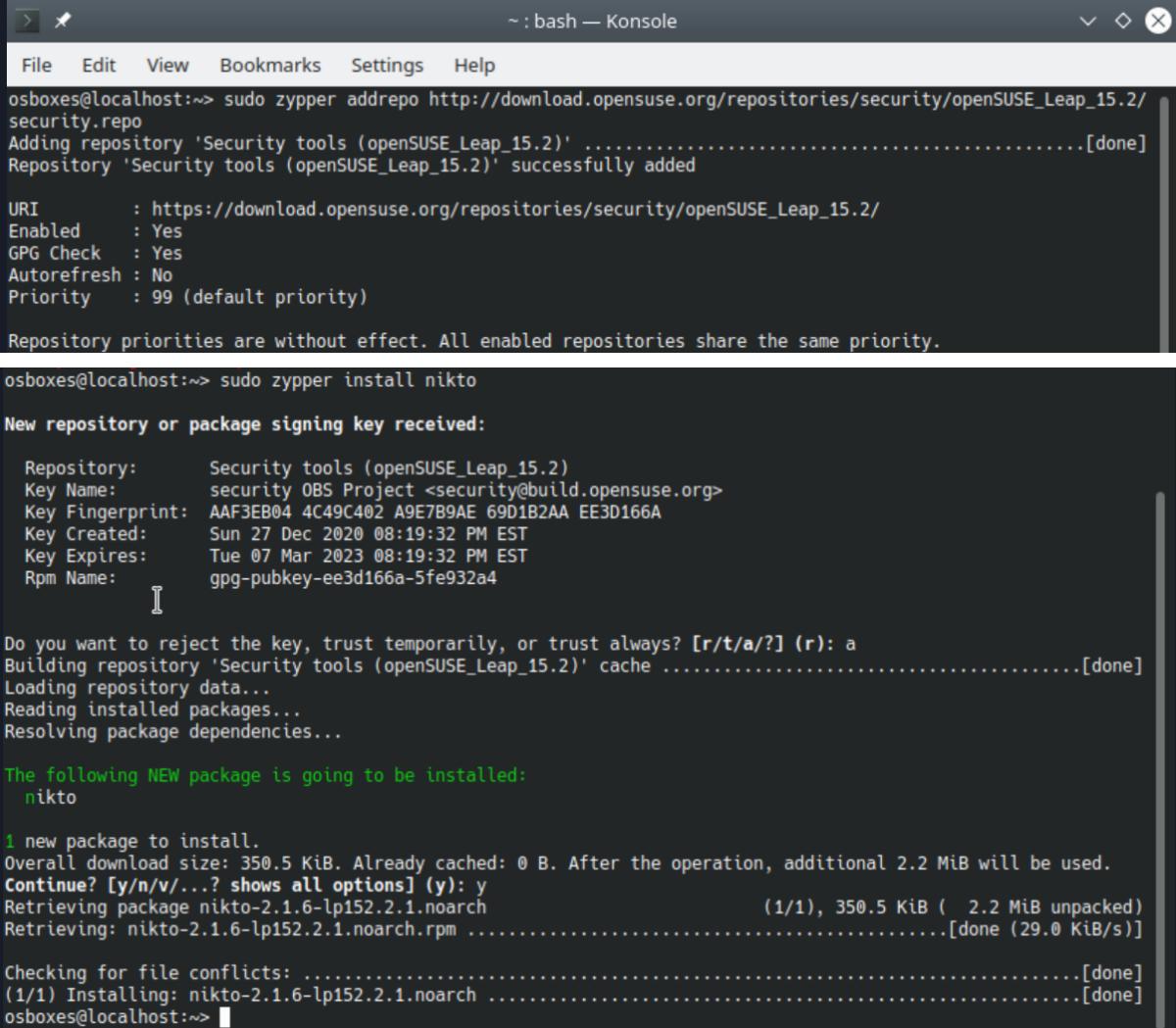
```
osboxes@localhost:~> sudo pip install --upgrade pip  
Requirement already satisfied: pip in /usr/lib/python3.6/site-packages (21.0)  
Collecting pip  
  Using cached pip-21.0-py3-none-any.whl (1.5 MB)  
  Downloading pip-20.3.4-py2.py3-none-any.whl (1.5 MB)  
   |██████████| 1.5 MB 1.1 MB/s  
osboxes@localhost:~> sudo pip install scapy  
Requirement already satisfied: scapy in /usr/lib/python3.6/site-packages (2.4.4)  
osboxes@localhost:~> ■
```

Agregamos el repositorio de seguridad que nos permitirá descargar nikto:

```
zypper addrepo
```

```
https://download.opensuse.org/repositories/security/openSUSE\_Leap\_15.2/security.repo
```

sudo zypper install nikto



The screenshot shows a terminal window titled "bash — Konsole". The user has run the command "sudo zypper install nikto". The output shows the addition of a new repository and the successful download and installation of the nikto package. A GPG key was received and trusted.

```
osboxes@localhost:~> sudo zypper addrepo http://download.opensuse.org/repositories/security/openSUSE_Leap_15.2/security.repo
Adding repository 'Security tools (openSUSE_Leap_15.2)' .....[done]
Repository 'Security tools (openSUSE_Leap_15.2)' successfully added

URI      : https://download.opensuse.org/repositories/security/openSUSE_Leap_15.2/
Enabled   : Yes
GPG Check : Yes
Autorefresh : No
Priority  : 99 (default priority)

Repository priorities are without effect. All enabled repositories share the same priority.

osboxes@localhost:~> sudo zypper install nikto

New repository or package signing key received:

Repository: Security tools (openSUSE_Leap_15.2)
Key Name: security OBS Project <security@build.opensuse.org>
Key Fingerprint: AAF3EB04 4C49C402 A9E7B9AE 69D1B2AA EE3D166A
Key Created: Sun 27 Dec 2020 08:19:32 PM EST
Key Expires: Tue 07 Mar 2023 08:19:32 PM EST
Rpm Name: gpg-pubkey-ee3d166a-5fe932a4

Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r): a
Building repository 'Security tools (openSUSE_Leap_15.2)' cache .....[done]
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following NEW package is going to be installed:
  nikto

1 new package to install.
Overall download size: 350.5 KiB. Already cached: 0 B. After the operation, additional 2.2 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): y
Retrieving package nikto-2.1.6-lp152.2.1.noarch          (1/1), 350.5 KiB ( 2.2 MiB unpacked)
Retrieving: nikto-2.1.6-lp152.2.1.noarch.rpm .....[done (29.0 KiB/s)]

Checking for file conflicts: .....[done]
(1/1) Installing: nikto-2.1.6-lp152.2.1.noarch .....[done]
osboxes@localhost:~>
```

Ahora agregaremos el repositorio packman para descargar ncrack:

zypper ar -cfp 90

'https://ftp.gwdg.de/pub/linux/misc/packman/suse/openSUSE_Leap_\$releasever/' packman

```
- : sudo zypper — Konsole
File Edit View Bookmarks Settings Help
osboxes@localhost:~> sudo zypper addrepo -cfp 90 'https://ftp.gwdg.de/pub/linux/misc/packman/suse/openSUSE_Leap
$_releasever/' packman
[sudo] password for root:
Adding repository 'packman' .....[done]
Repository 'packman' successfully added

URI      : https://ftp.gwdg.de/pub/linux/misc/packman/suse/openSUSE_Leap_15.2/
Enabled   : Yes
GPG Check : Yes
Autorefresh : Yes
Priority   : 90 (raised priority)

Repository priorities in effect: (See 'zypper lr -P' for details)
  90 (raised priority) : 1 repository
  99 (default priority) : 6 repositories
osboxes@localhost:~> sudo zypper dup --from packman --allow-vendor-change
Retrieving repository 'packman' metadata -----[|]

New repository or package signing key received:

Repository: packman
Key Name: PackMan Project (signing key) <packman@links2linux.de>
Key Fingerprint: F8875B88 0D518B6B 8C530D13 45A1D067 1ABD1AFB
Key Created: Mon 15 Sep 2014 06:18:00 PM EDT
Key Expires: Thu 12 Sep 2024 06:17:21 PM EDT
Rpm Name: gpg-pubkey-1abd1afb-54176598

Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r): a
Retrieving repository 'packman' metadata -----[|]
```

Indicamos la opción **y** para continuar

```
- : sudo zypper — Konsole
File Edit View Bookmarks Settings Help
Reading installed packages...
Computing distribution upgrade...

The following 6 NEW packages are going to be installed:
  libopencore-amrnb0 libopencore-amrwb0 libvidstab1_1 libx264-161 libx265-192 libxvidcore4

The following 17 packages are going to be upgraded:
  libavcodec57 libavfilter6 libavformat57 libavresample3 libavutil55 libgstphotography-1_0-0 libpostproc54
  libswresample2 libswscale4 libvlc5 libvlccore9 vlc vlc-codec-gstreamer vlc-lang vlc-noX vlc-qt vlc-vdpau

The following 17 packages are going to change vendor:
  libavcodec57          openSUSE -> http://packman.links2linux.de
  libavfilter6          openSUSE -> http://packman.links2linux.de
  libavformat57         openSUSE -> http://packman.links2linux.de
  libavresample3        openSUSE -> http://packman.links2linux.de
  libavutil55           openSUSE -> http://packman.links2linux.de
  libgstphotography-1_0-0 openSUSE -> http://packman.links2linux.de
  libpostproc54          openSUSE -> http://packman.links2linux.de
  libswresample2         openSUSE -> http://packman.links2linux.de
  libswscale4            openSUSE -> http://packman.links2linux.de
  libvlc5                openSUSE -> http://packman.links2linux.de
  libvlccore9            openSUSE -> http://packman.links2linux.de
  vlc                   openSUSE -> http://packman.links2linux.de
  vlc-codec-gstreamer    openSUSE -> http://packman.links2linux.de
  vlc-lang               openSUSE -> http://packman.links2linux.de
  vlc-noX                openSUSE -> http://packman.links2linux.de
  vlc-qt                 openSUSE -> http://packman.links2linux.de
  vlc-vdpau              openSUSE -> http://packman.links2linux.de

17 packages to upgrade, 6 new, 17 to change vendor.
Overall download size: 18.6 MiB. Already cached: 0 B. After the operation, additional 24.7 MiB will be used.
Continue? [y/n/v/...? shows all options] (y): y
```

Procedemos a instalar ncrack con: ***sudo zypper install ncrack***

```
The following NEW package is going to be installed:
[+] ncrack
1 new package to install.
Overall download size: 630.6 KiB. Already cached: 0 B. After the operation, additional 1.6 MiB will be used.
Continue? [y/n/v/? shows all options] (y): y
Retrieving package ncrack-0.6.0+git.20180620-pm152.2.7.x86_64          (1/1), 630.6 KiB ( 1.6 MiB unpacked)
Retrieving: ncrack-0.6.0+git.20180620-pm152.2.7.x86_64.rpm .....[done (207.6 KiB/s)]
Checking for file conflicts: .....[done]
(1/1) Installing: ncrack-0.6.0+git.20180620-pm152.2.7.x86_64 .....[done]
osboxes@localhost:~> [REDACTED]
```

Descargar pytbull con el comando: **wget**

<https://downloads.sourceforge.net/project/pytbull/pytbull-2.1.tar.bz2>

Extraemos la carpeta y la movemos al directorio /opt/ con: **sudo tar -xvf**

pytbull-2.1.tar.bz2 -C /opt/

```
Downloads : bash — Konsole
File Edit View Bookmarks Settings Help
osboxes@localhost:~/Downloads> sudo tar -xvf pytbull-2.1.tar.bz2 -C /opt/
pytbull/
pytbull/conf/
pytbull/conf/config.cfg
pytbull/conf/config_bt5.cfg
```

Desde el script de pytbull, agregaremos dos líneas para importar la librería datetime la cual es utilizada por varios de los tests a ejecutar:

sudo vim /opt/pytbull/pytbull

```
osboxes@localhost:/opt/pytbull> sudo vim /opt/pytbull/pytbull
```

Ya dentro del archivo bajamos a la línea 368 y agregamos una nueva línea con lo siguiente:

import datetime

Repetimos este paso en la línea 482 del archivo y guardamos los cambios con **ESC :wq!**

```
except:
    print "[ Failed ]"
    print "\n***ERROR: Please setup reverse shell on remote server first or
use --mode=gateway!"
    sys.exit(0)
print "[ OK ]"

for payload in payloads:
    # Perform test & write report
    str = "TEST #%s - %s" % (self.testnum, payload[0])
    print str[:62].ljust(65, '.')
    import datetime
    test_dt_start = datetime.datetime.now()

    if self._mode=="standalone":
        # Send cmd to execute on server side (wget file)
        s.send("wget %s" % os.path.join(self.config.get('PATHS', 'urlpdf'), payload[0]))
        # Issue 3450032 - Synchronisation issue. The server has to instruct
        # the client that the file has been successfully downloaded before it
        # goes to next file
        s.recv(1024)
    else:
        self.downloadFile(self.config.get('PATHS', 'urlpdf'), payload[0])

    # Sleep before getting alerts
    time.sleep(int(self.config.get('TIMING', 'sleepbeforegetalerts')))
```

-- INSERT --

Ahora nos ubicamos en la siguiente ruta:

cd /opt/pytbull/conf y editamos el archivo de configuración config.cfg para ir ingresando los datos:

sudo vim config.cfg

```
[CLIENT]
ipaddr          = 192.168.2.9
iface           = eth1
useproxy        = 0
proxyhost       =
proxyport       =
proxyuser       =
proxypass      =

[PATHS]
db              = data/pytbull.db
urldf           = https://github.com/sebastiendamaye/public/raw/master/infected/
pdfdir          = pdf/malicious
pcapdir         = pcap
tempfile        = /tmp/pytbull.tmp
alertsfile      = /var/ftp/pub/snort.alert
#alertsfile     = /var/log/suricata/fast.log
```

En la estructura del archivo, la cual está dividida por secciones, ingresaremos la información correspondiente:

- En la sección client, solo ingresaremos la ipaddr en la cual tenemos instalado nuestro cliente pytbull y la iface que indica la interfaz de salida que usará pytbull para enviar los payloads.
- En la sección PATHS, en las variables alertfiles indicamos la dirección donde se almacenan nuestros archivos de alerta, en este caso indicaremos la siguiente ruta: /var/ftp/pub/snort.alert
Es importante mantener solo una ruta alertfiles activa, la otra se comenta con un #.
Nota: Para efectos del taller se creó anteriormente el archivo snort.alert para evitar el error de pytbull client al intentar acceder al archivo de alerta, ya que dicha configuración no está dentro del alcance del taller. su objetivo es mostrar en el reporte de pytbull las pruebas que el IDS detectó correctamente. Para ello se debe configurar el acceso FTP a la ruta de logs de snort con los permisos correspondientes.

En la sección FTP agregamos el protocolo FTP, puerto 21 y las credenciales de nuestra máquina osboxes para establecer la conexión:

```
[FTP]
ftppproto      = ftp
ftppport        = 21
ftpuser          = osboxes
ftppasswd        = osboxes.org
```

Las secciones Timing y Server no requieren modificación.

En la sección ENV, modificamos las rutas por defecto y agregamos las siguientes para nikto (**/usr/bin/nikto**), niktoconf (**/etc/nikto.conf**) y ncrack (**/usr/bin/ncrack**) como se muestra en la imagen:

```
[ENV]
sudo          = /usr/bin/sudo
nmap          = /usr/bin/nmap
nikto         = /usr/bin/nikto
niktoconf     = /etc/nikto.conf
hping3        = /usr/sbin/hping3
tcpreplay     = /usr/bin/tcpreplay
ab            = /usr/bin/ab
ping          = /bin/ping
ncrack        = /usr/bin/ncrack
ncrackusers   = data/ncrack-users.txt
ncrackpasswords = data/ncrack-passwords.txt
-- INSERT -- W10: Warning: Changing a readonly file
```

16,53

Top

El apartado de TESTS puede modificarse para habilitar (valor 1) o deshabilitar (valor 0) los tests que queramos realizar. Para este caso deshabilitamos los testRules, denialOfService y ipReputation.

```
[TESTS]
clientSideAttacks      = 1
testRules               = 0
badTraffic              = 1
fragmentedPackets       = 1
bruteForce               = 1
evasionTechniques       = 1
shellCodes              = 1
denialOfService          = 0
pcapReplay               = 1
normalUsage              = 1
ipReputation             = 0
```

Una vez realizadas las configuraciones, procedemos a ejecutar las pruebas desde pytbull client: **sudo python2 ./pytbull -t 192.168.2.8**

```
osboxes@localhost:/opt/pytbull> sudo python2 ./pytbull -t 192.168.2.8
```



```
What would you like to do?
1. Run a new campaign (will erase previous results)
2. View results from previous campaign
3. Exit
Choose an option: 1
```

```
(standalone mode)
```

```
+-----+
| pytbull will set off IDS/IPS alarms and/or other security devices
| and security monitoring software. The user is aware that malicious
| content will be downloaded and that the user should have been
| authorized before running the tool.
+-----+
```

```
Do you accept (y/n)? y
```

```
- : sudo python2 — Konsole
File Edit View Bookmarks Settings Help
-----
Client Side Attacks.... [ yes ]
Test Rules..... [ yes ]
Bad Traffic..... [ yes ]
Fragmented Packets..... [ yes ]
Brute Force..... [ yes ]
Evasion Techniques..... [ yes ]
ShellCodes..... [ yes ]
Denial of Service..... [ yes ]
Pcap Replay..... [ yes ]
Normal Usage..... [ yes ]
IP Reputation..... [ yes ]

CLIENT SIDE ATTACKS
-----
TEST #1 - 001e2710555613a82e94156d3ed9c289..... [ done ]
TEST #2 - 7b9e1c1b479447506cc046a5d8219eca..... [ done ]
TEST #3 - 004e74d54dcf79c641d5cf8a615488a0..... [ done ]
TEST #4 - 7d6e9af1018c10f1b7dfa5169a35d941..... [ done ]
TEST #5 - 0106fb569e87e02fc88d496064abdf19..... [ done ]
TEST #6 - 7f73dd439572409a64bc4dd0d603aacf..... [ done ]
TEST #7 - 02bfe34bea55e327cfdead9cff215f33..... [ done ]
TEST #8 - 7f7413bd2a4a0f001efd0305f4f56acf..... [ done ]
TEST #9 - 030423da29e1e6f4a527518126de4aeb..... [ done ]
TEST #10 - 80202a9c51d8544bac7ac273428dd97c..... [ done ]
TEST #11 - 03042cc3786dafdb941019488d4cad3e..... [ done ]
TEST #12 - 80f20af63314be2e8c79d8ca99eeb713..... [ done ]
TEST #13 - 03546e59967af0c2dbf609013934cd07..... [ done ]
TEST #14 - 82a5f96d1834411a3b5af9c21ffb14a8..... [ done ]
TEST #15 - 04095314d51057a13e21908de1266fc1..... [ done ]
TEST #16 - 82a7c8fdacc91b1bd0fdc2407674f50..... [ done ]
```

```
- : sudo python2 — Konsole
File Edit View Bookmarks Settings Help
-----
TEST #81 - win32_bind_dllinject - EXITFUNC=seh DLL=c:\ LPORT=4.... [ done ]
TEST #82 - win32_bind - EXITFUNC=seh LPORT=4444 Size=709 Encod... [ done ]
TEST #83 - db "cmd.exe" /c net user USERNAME PASSWORD /ADD && n... [ done ]
TEST #84 - Cisco: Creates a new VTY, allocates a password then... [ done ]
TEST #85 - Rothenburg Shellcode..... [ done ]
TEST #86 - Mainz/Bielefeld Shellcode..... [ done ]

PCAP REPLAY
-----
TEST #87 - slammer worm..... [ done ]

NORMAL USAGE
-----
apr_sockaddr_info_get() for %target%: Name or service not known (670002)
TEST #88 - ApacheBench 10 requests..... [ done ]
TEST #89 - Standard ping..... [ done ]

-----
DONE. Check the report.

-----
Webserver started at http://127.0.0.1:8080
(use ^C to stop)
```

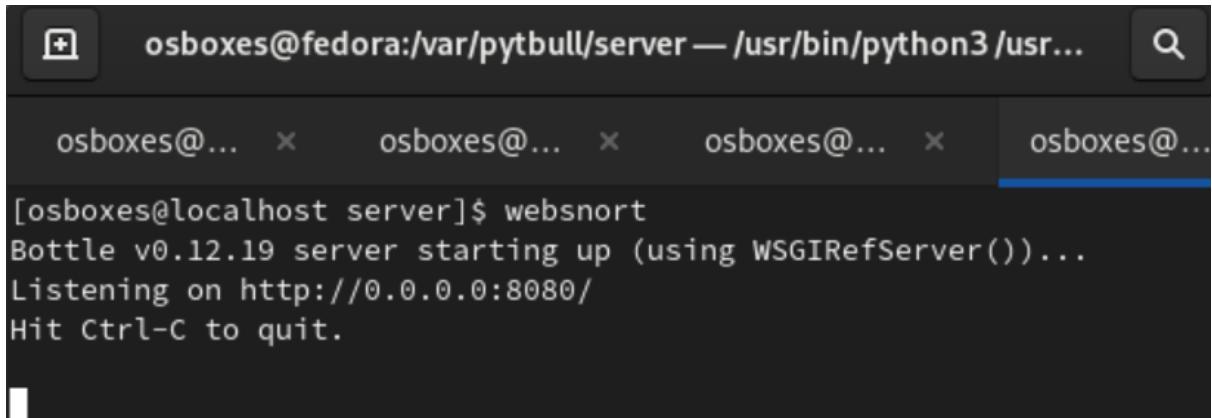
Luego de terminada la ejecución de los tests de pytbull, podemos regresar a la máquina snort a visualizar las alertas generadas utilizando websnort:

Asignamos los permisos correspondientes a la carpeta de logs de snort para que pueda utilizarlos websnort:

```
sudo chmod -R 775 /var/log/snort
```

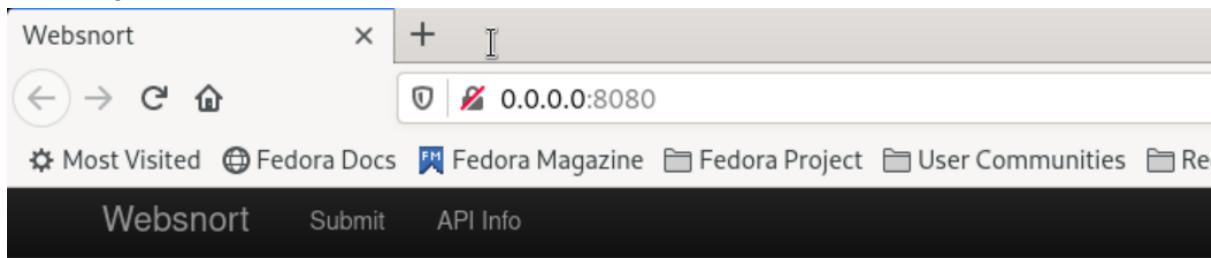
```
[osboxes@localhost server]$ sudo chmod -R 775 /var/log/snort
[sudo] password for osboxes:
[osboxes@localhost server]$
```

Ejecutamos el comando **websnort** desde la consola:

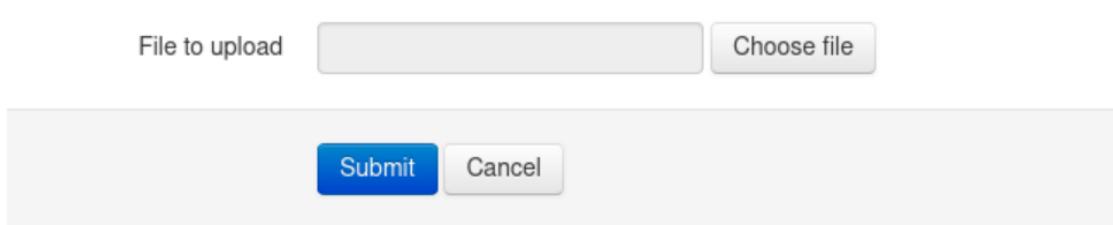


```
osboxes@fedora:/var/pytbull/server — /usr/bin/python3 /usr...
osboxes@... ✘ osboxes@... ✘ osboxes@... ✘ osboxes@...
[osboxes@localhost server]$ websnort
Bottle v0.12.19 server starting up (using WSGIRefServer())...
Listening on http://0.0.0.0:8080/
Hit Ctrl-C to quit.
```

Y nos dirigimos a la ruta indicada: <http://0.0.0.0:8080> donde seleccionamos “choose file” para cargar un archivo de alerta:

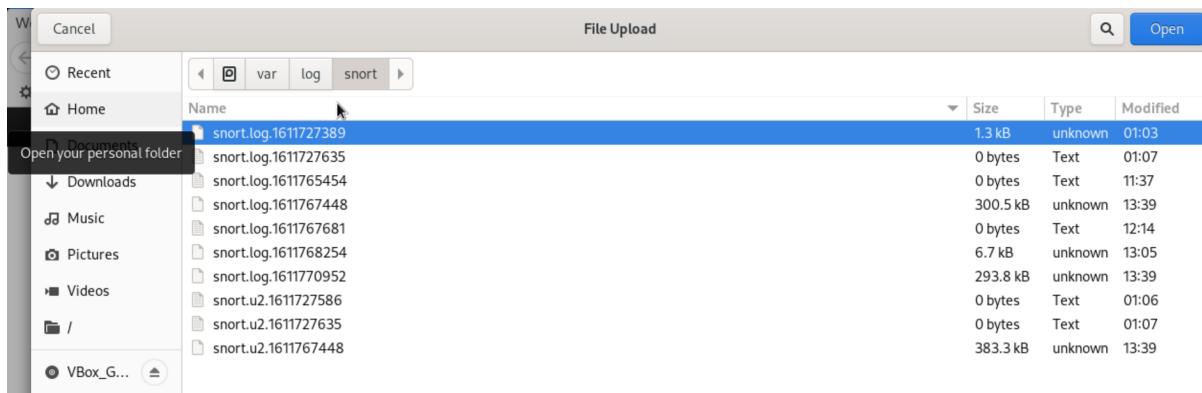


Submit PCAP



The screenshot shows a "Submit PCAP" form. At the top, there is a label "File to upload" next to a file input field and a "Choose file" button. Below this, there is a large input field for the PCAP file. At the bottom of the input field, there are two buttons: a blue "Submit" button and a white "Cancel" button.

Seleccionamos el archivo de log a visualizar:



Seleccionamos submit y luego podremos visualizar las alertas correspondientes:

Submit PCAP

File to upload

Websnort	Submit	API Info						
2021-01-27 13:15:19.866625	10000001			PING ATTACK	10.0.2.4	10.0.2.5	ICMP	
2021-01-27 13:15:23.384462	10000001		0	PING ATTACK	10.0.2.4	10.0.2.5	ICMP	
2021-01-27 13:15:24.286744	2010936	Potentially Bad Traffic	2	ET SCAN Suspicious inbound to Oracle SQL port 1521	192.168.100.1:59631	10.0.2.4:1521	TCP	
2021-01-27 13:15:24.286869	2010936	Potentially Bad Traffic	2	ET SCAN Suspicious inbound to Oracle SQL port 1521	192.168.100.2:59631	10.0.2.4:1521	TCP	
2021-01-27 13:15:24.286870	2010936	Potentially Bad Traffic	2	ET SCAN Suspicious inbound to Oracle SQL port 1521	192.168.100.3:59631	10.0.2.4:1521	TCP	
2021-01-27 13:15:24.286964	2010936	Potentially Bad Traffic	2	ET SCAN Suspicious inbound to Oracle SQL port 1521	192.168.100.4:59631	10.0.2.4:1521	TCP	
2021-01-27 13:15:24.286964	2010936	Potentially Bad Traffic	2	ET SCAN Suspicious inbound to Oracle SQL port 1521	192.168.100.5:59631	10.0.2.4:1521	TCP	
2021-01-27 13:15:56.993710	2018489	Attempted Information Leak	2	ET SCAN NMAP OS Detection Probe	192.168.100.1:37488	10.0.2.4:38555	UDP	
2021-01-27 13:15:56.993711	2018489	Attempted Information Leak	2	ET SCAN NMAP OS Detection Probe	192.168.100.2:37488	10.0.2.4:38555	UDP	
2021-01-27 13:15:56.993711	2018489	Attempted Information Leak	2	ET SCAN NMAP OS Detection Probe	192.168.100.3:37488	10.0.2.4:38555	UDP	
2021-01-27 13:15:56.993711	2018489	Attempted Information Leak	2	ET SCAN NMAP OS Detection Probe	192.168.100.4:37488	10.0.2.4:38555	UDP	
2021-01-27 13:15:56.993712	2018489	Attempted Information Leak	2	ET SCAN NMAP OS Detection Probe	192.168.100.5:37488	10.0.2.4:38555	UDP	