

PRÁCTICA EVALUABLE

MÓDULO 4:

RESPUESTA ANTE INCIDENTES



Escrito por:

Grupo 1B

/ Joan Navinés / Juan Carlos Vico / Ignacio Camacho / Jesica Julia Olortegui / Luis Delgado /

ÍNDICE

Introducción.....	3
¿Qué es un NIDS?.....	3
Sistema de Detección de Intrusos (IDS).....	3
Ejemplo práctico.....	3
Selección de un IDS para su implantación.....	4
Selección e Instalación del IDS.....	4
Configuración de la Máquina Virtual.....	4
Instalación de Snort.....	10
Montaje del Laboratorio con el IDS Escogido.....	13
Verificación de Conectividad entre Máquinas.....	14
Configuración del Firewall en Windows 10.....	14
Creación de Regla para Permitir ICMPv4 en Windows Defender.....	15
Configuración de la Dirección IP o Subred.....	17
Consulta de la Dirección IP en Windows 10.....	17
Verificación de Conectividad con Ping.....	18
Verificación de Conectividad con Ping (Parrot → Windows 10).....	18
Estado de la Conexión.....	18
Implementación de Casos de Uso para el IDS.....	19
Configuración Previa del Entorno para Gestionar Alertas.....	19
Creación de Reglas en Snort.....	20
Creación de una Alerta para Ping a la IP 8.8.8.8.....	20
Creación de una Alerta para Consultas Web (HTTP/HTTPS).....	23
Creación de una alerta para el uso de Dropbox.....	25
Creación de una Alerta para Conexiones Salientes SSH.....	27

Introducción

La siguiente práctica se ha realizado de forma grupal. El equipo está formado por los alumnos **Juan Carlos Vico, Ignacio, Jesica Julia Olortegui, Luis Delgado y Joan Navinés**.

¿Qué es un NIDS?

Sistema de Detección de Intrusos (IDS)

Para entender su concepto, primero debemos conocer qué es un **IDS** (*Intrusion Detection System*) o **Sistema de Detección de Intrusos**. Se trata de una herramienta de seguridad que supervisa el tráfico en una red o sistema para detectar **actividades sospechosas o maliciosas**.

Funciones principales

El IDS tiene como objetivo **alertar** al personal de seguridad sobre posibles amenazas, tales como: **Intentos de acceso no autorizados → Malware → Anomalías en el comportamiento del tráfico**

Un aspecto clave a tener en cuenta es que el **IDS solo detecta y avisa**, pero **no toma medidas automáticas** para bloquear acciones maliciosas.

NIDS: Sistema de Detección de Intrusos basado en la Red

Dentro de los **IDS**, encontramos los **NIDS** (*Network-based Intrusion Detection System*), o **Sistema de Detección de Intrusos basado en la Red**.

Esta herramienta se instala en **puntos estratégicos** de una red, como: **→ Router → Switch → Servidores**

Su objetivo es **analizar en tiempo real** el tráfico que circula por la red para **identificar patrones sospechosos** en la comunicación entre dispositivos y **alertar** al personal de seguridad antes de que un posible ataque afecte los sistemas internos.

Ejemplo práctico

Para ilustrar su funcionamiento, imaginemos el siguiente escenario:

*Una empresa decide implementar un **NIDS** en su red para monitorear todo el tráfico entrante y saliente. Si el sistema detecta que una **IP externa** está realizando múltiples intentos de conexión en poco tiempo—lo que podría indicar un **ataque de fuerza bruta**—el NIDS generará una alerta para que los administradores de seguridad investiguen y tomen medidas preventivas.*

Selección de un IDS para su implantación

Selección e Instalación del IDS

El IDS escogido ha sido **Snort**, el cual ejecutaremos desde una máquina virtual utilizando la distribución de **Linux Parrot OS**.

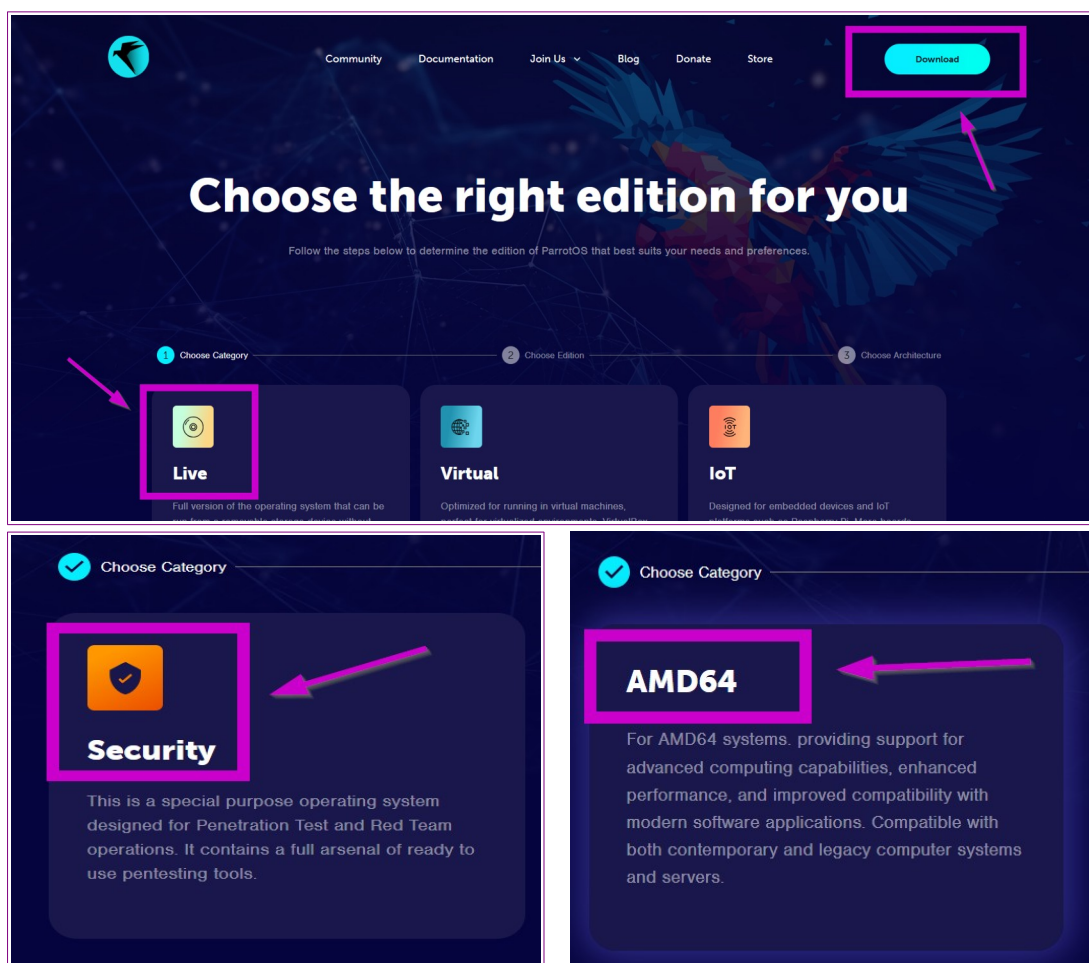
Configuración de la Máquina Virtual

Primero, debemos disponer de una máquina virtual; en nuestro caso, utilizaremos **VirtualBox**.

Para descargar la distribución, accedemos a la página oficial de <https://www.parrotsec.org/> y seguimos estos pasos:

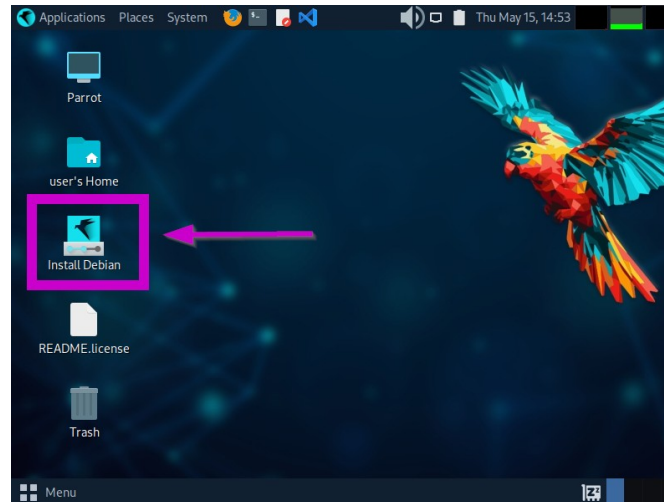
1. **Ingresamos** a la página proporcionada.
2. **Buscamos** la pestaña "Download".
3. **Hacemos clic** en la pestaña y nos desplazamos hasta las opciones de descarga.
4. **Seleccionamos:** *Live* → *Security* → *Amd64*.

Una vez descargada la imagen del sistema, podremos **levantar la máquina virtual** y proceder con la instalación de **Parrot OS**.



Instalación de Parrot OS

Una vez iniciamos nuestra máquina **Parrot OS**, en el escritorio aparece el icono "**Install Debian**". Procedemos con su instalación y configuración, ajustando los siguientes parámetros: ➔ **Idioma** ➔ **Teclado** ➔ **Zona horaria**.

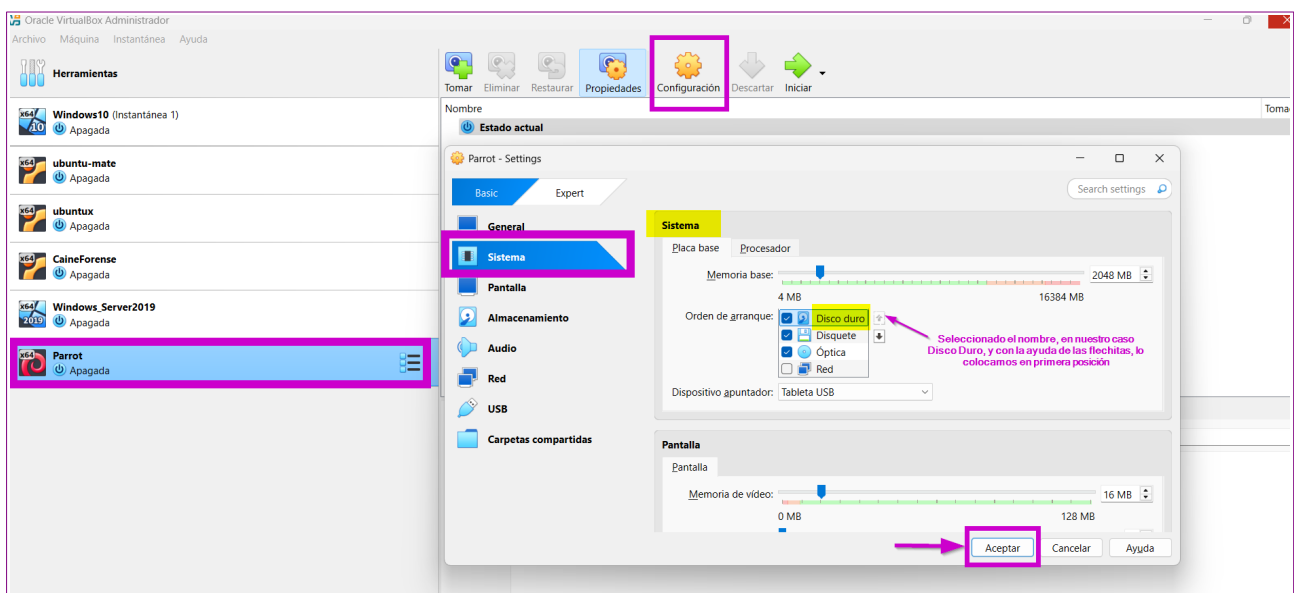


Finalización de la Instalación de Parrot OS

Una vez finalizada la instalación, el sistema nos solicita un **reinicio**, el cual debemos realizar.

Posteriormente, en **VirtualBox**, accedemos a la configuración de *Parrot OS* y seguimos estos pasos:
➔ En la pestaña **Sistema**, ajustamos el **orden de arranque**, colocando el **Disco Duro** en primer lugar. ➔ Marcamos la casilla **Aceptar** para guardar los cambios.

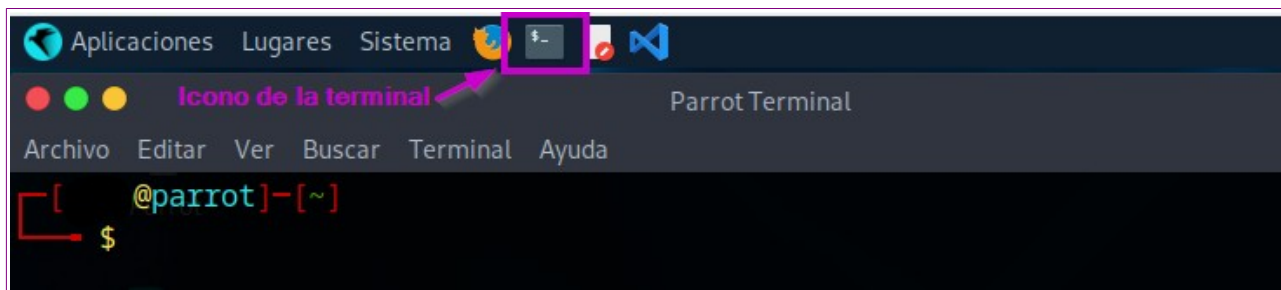
Ahora, cada vez que iniciemos **Parrot OS**, el sistema arrancará directamente desde el **disco duro**, sin necesidad de utilizar el **modo Live**. Es decir, estaremos utilizando la instalación completa en nuestra máquina virtual, sin necesidad de volver a cargarla.



Inicio de Parrot OS e Instalación de Snort

Una vez hemos **instalado y configurado** *Parrot OS* en nuestra máquina virtual, lo iniciamos.

El siguiente paso será **proceder con la instalación de Snort**. Para ello: ➔ **Abrimos la consola** (*Terminal* o *cmd*) dentro de *Parrot OS*. ➔ **Ejecutamos** una serie de **comandos** para su instalación.

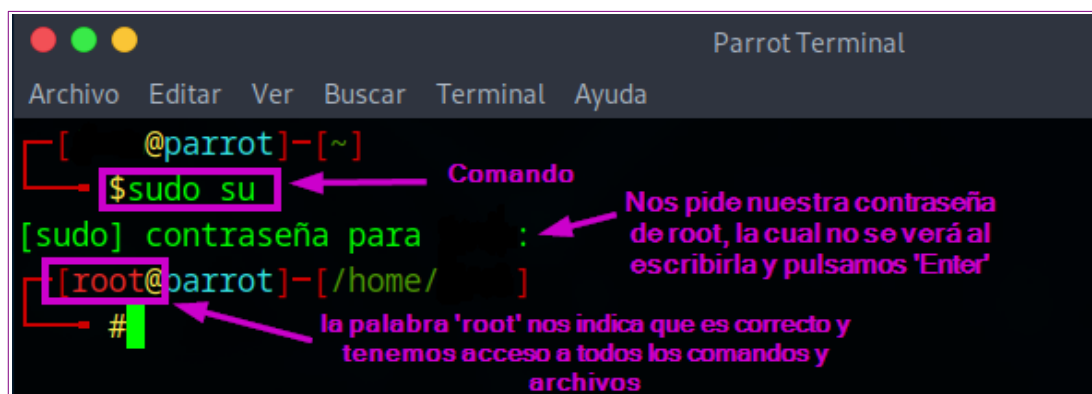


Elevación de Privilegios

Para evitar problemas de **administrador** durante la instalación, ejecutamos el siguiente comando:

```
sudo su
```

Esto nos permitirá obtener **privilegios de superusuario** y ejecutar los comandos posteriores de manera más cómoda.



Instalación de Snort y Configuración de Repositorios

El primer paso fue intentar la instalación de **Snort** con el siguiente comando:

```
sudo apt-get install snort
```

Sin embargo, el paquete **no estaba disponible**.

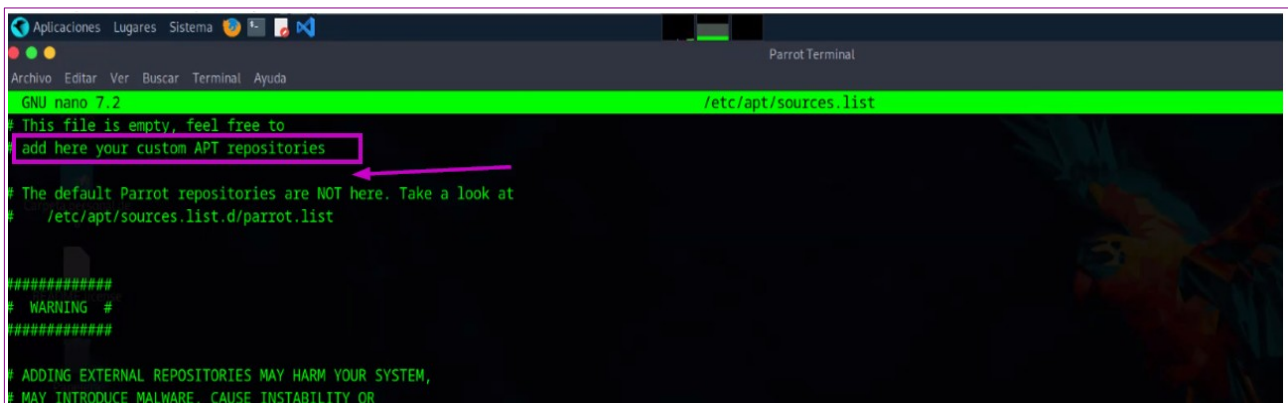
Solución

Para acceder a versiones actualizadas de **Snort**, añadimos los **repositorios de Ubuntu Focal (20.04)** al archivo de fuentes del sistema.

Para editar el archivo de repositorios, ejecutamos:

```
sudo nano /etc/apt/sources.list
```

Esto abrirá el documento de configuración, donde podremos añadir las fuentes necesarias.



```
GNU nano 7.2 /etc/apt/sources.list
# This file is empty, feel free to
# add here your custom APT repositories

# The default Parrot repositories are NOT here. Take a look at
# /etc/apt/sources.list.d/parrot.list

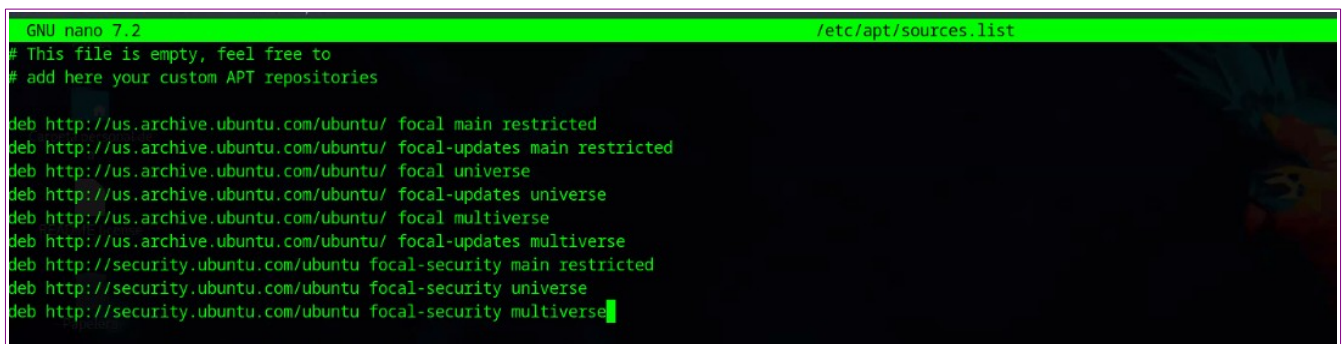
#####
# WARNING #
#####

# ADDING EXTERNAL REPOSITORIES MAY HARM YOUR SYSTEM,
# MAY INTRODUCE MALWARE, CAUSE INSTABILITY OR
```

Adición de Repositorios de Ubuntu

Justo debajo de la línea que indica *"add here your custom APT repositories"*, añadimos los siguientes **repositorios**, cada uno en una línea distinta:

```
deb http://us.archive.ubuntu.com/ubuntu/ focal main restricted
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted
deb http://us.archive.ubuntu.com/ubuntu/ focal universe
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates universe
deb http://us.archive.ubuntu.com/ubuntu/ focal multiverse
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates multiverse
deb http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
```



```
GNU nano 7.2 /etc/apt/sources.list
# This file is empty, feel free to
# add here your custom APT repositories

deb http://us.archive.ubuntu.com/ubuntu/ focal main restricted
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates main restricted
deb http://us.archive.ubuntu.com/ubuntu/ focal universe
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates universe
deb http://us.archive.ubuntu.com/ubuntu/ focal multiverse
deb http://us.archive.ubuntu.com/ubuntu/ focal-updates multiverse
deb http://security.ubuntu.com/ubuntu focal-security main restricted
deb http://security.ubuntu.com/ubuntu focal-security universe
deb http://security.ubuntu.com/ubuntu focal-security multiverse
```

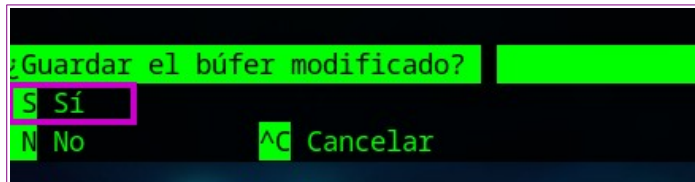
Guardado de Configuración

Para salir del editor, **presionamos**:

CTRL + X

Esto mostrará un mensaje confirmando si deseamos **guardar los cambios realizados**.

Pulsamos la tecla S para confirmar y guardar la configuración.



Validación de Paquetes e Importación de Claves GPG

Para validar los **paquetes** procedentes de los repositorios añadidos, es necesario **importar las claves GPG** correspondientes.

Estas claves garantizan que los paquetes de software sean **auténticos**, evitando la instalación de programas modificados o maliciosos.

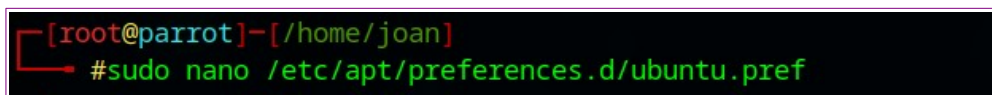
Ejecutamos los siguientes comandos **una a una**:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C
sudo apt-key export 3B4FE6ACC0B21F32 | sudo gpg --dearmor -o
/etc/apt/trusted.gpg.d/ubuntu-keyring-3B4FE6ACC0B21F32.gpg
sudo apt-key export 871920D1991BC93C | sudo gpg --dearmor -o
/etc/apt/trusted.gpg.d/ubuntu-keyring-871920D1991BC93C.gpg
```

Creación del Archivo de Preferencias

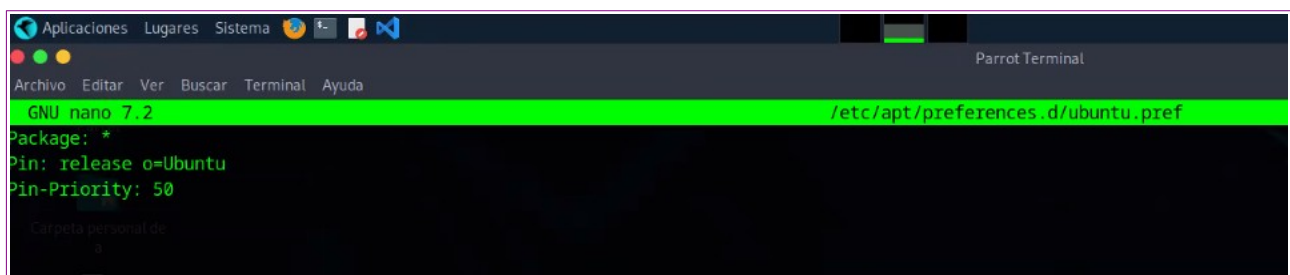
Para evitar que los **paquetes de Ubuntu** sobrescriban los propios de **Debian** en futuras actualizaciones, se crea un **archivo de preferencias** con el siguiente comando:

```
sudo nano /etc/apt/preferences.d/ubuntu.pref
```



Esto abrirá un documento en el cual debemos escribir el siguiente contenido:

```
Package: *
Pin: release o=Ubuntu
Pin-Priority: 50
```



Guardado de Configuración y Actualización del Sistema

Para salir del editor, **presionamos**:

CTRL + X

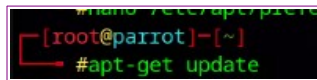
Esto nos mostrará un mensaje preguntando si queremos **guardar los cambios realizados**.

➔ **Pulsamos la tecla S** para confirmar y guardar la configuración.

Actualización de Paquetes

Una vez configurado el sistema, ejecutamos el siguiente comando para **actualizar la lista de paquetes disponibles** desde los repositorios:

`apt-get update`



Este proceso garantiza que nuestro sistema tenga acceso a las **versiones más recientes** de los paquetes.

Lista de Paquetes Actualizados

```
Des:5 http://security.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Des:6 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Des:7 http://us.archive.ubuntu.com/ubuntu focal/main amd64 Packages [970 kB]
Des:8 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [3,549 kB]
Des:9 http://us.archive.ubuntu.com/ubuntu focal/main Translation-en [506 kB]
Des:10 http://us.archive.ubuntu.com/ubuntu focal/main Translation-es [342 kB]
Des:11 http://us.archive.ubuntu.com/ubuntu focal/restricted amd64 Packages [22.0 kB]
Des:12 http://us.archive.ubuntu.com/ubuntu focal/restricted Translation-es [2,152 B]
Des:13 http://us.archive.ubuntu.com/ubuntu focal/restricted Translation-en [6,212 B]
Des:14 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8,628 kB]
Des:15 http://us.archive.ubuntu.com/ubuntu focal/universe Translation-es [1,326 kB]
Des:16 http://us.archive.ubuntu.com/ubuntu focal/universe Translation-en [5,124 kB]
Des:17 http://us.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Des:18 http://us.archive.ubuntu.com/ubuntu focal/multiverse Translation-es [70.0 kB]
Des:19 http://us.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Des:20 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,941 kB]
Des:21 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [516 kB]
Des:22 http://security.ubuntu.com/ubuntu focal-security/restricted amd64 Packages [3,745 kB]
Des:23 http://us.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [598 kB]
Des:24 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [3,901 kB]
Des:25 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [545 kB]
Des:26 http://us.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1,262 kB]
Des:27 http://us.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [303 kB]
Des:28 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [29.7 kB]
Des:29 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [8,316 B]
Des:30 http://security.ubuntu.com/ubuntu focal-security/restricted Translation-en [524 kB]
Des:31 http://security.ubuntu.com/ubuntu focal-security/universe amd64 Packages [1,040 kB]
Des:32 http://security.ubuntu.com/ubuntu focal-security/universe Translation-en [221 kB]
Des:33 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 Packages [26.6 kB]
Des:34 http://security.ubuntu.com/ubuntu focal-security/multiverse Translation-en [6,448 B]
Descargados 38.0 MB en 11s (3,419 kB/s)
Leyendo lista de paquetes... Hecho
```

Instalación de Snort

Seguidamente, ejecutamos el siguiente comando para instalar **Snort**, un IDS open source:

```
apt-get install snort
```

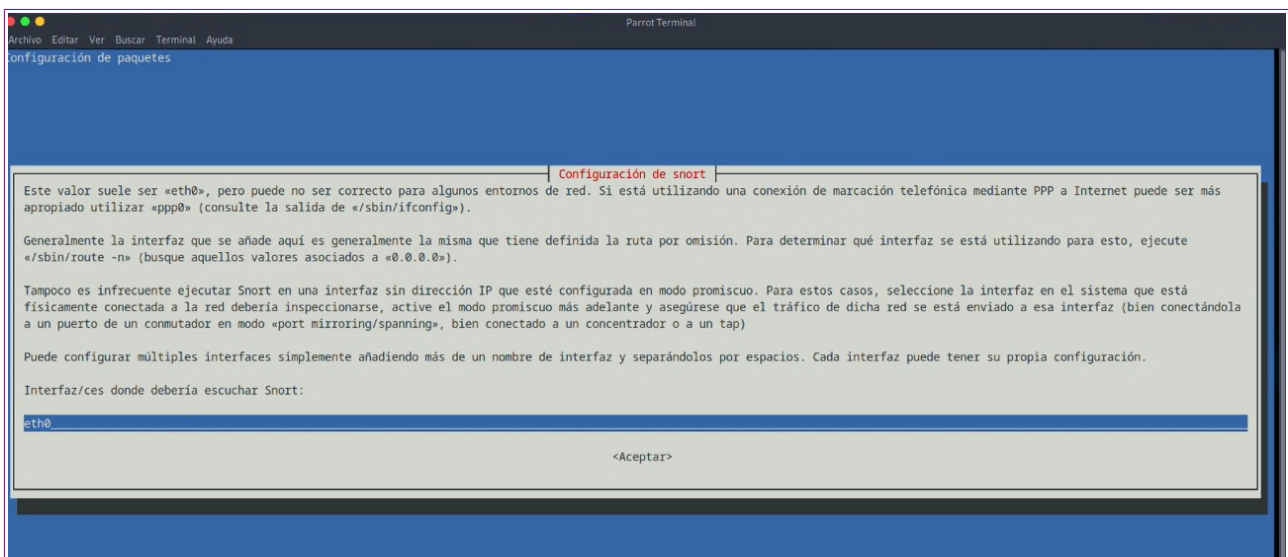
Este proceso instalará los **paquetes necesarios**.

Al finalizar la instalación, el sistema preguntará si deseamos continuar. ➡ Para confirmar, pulsamos la tecla **S**.

```
[root@parrot:~/home/ian]
#sudo apt-get install snort
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  golang-1.22-go golang-1.22-src libglapi-mesa lp-solve
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libdaq2 libestr0 libfastjson4 liblognorm5 oinkmaster rsyslog snort-common
  snort-common-libraries snort-rules-default
Paquetes sugeridos:
  rsyslog-mysql | rsyslog-pgsql rsyslog-mongodb rsyslog-doc rsyslog-openssl
  | rsyslog-gnutls rsyslog-gssapi rsyslog-relp snort-doc
Se instalarán los siguientes paquetes NUEVOS:
  libdaq2 libestr0 libfastjson4 liblognorm5 oinkmaster rsyslog snort
  snort-common snort-common-libraries snort-rules-default
0 actualizados, 10 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.247 kB de archivos.
Se utilizarán 9.636 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Selección de Interfaz de Red

Durante el proceso de instalación, el sistema **solicita la interfaz de red** para la configuración de **Snort**.



Por defecto, el sistema muestra la interfaz <eth0>, pero debemos **buscar** y utilizar la interfaz de red que usa nuestra maquina.

Para averiguarla, **abrimos otra terminal** y ejecutamos uno de los siguientes comandos:

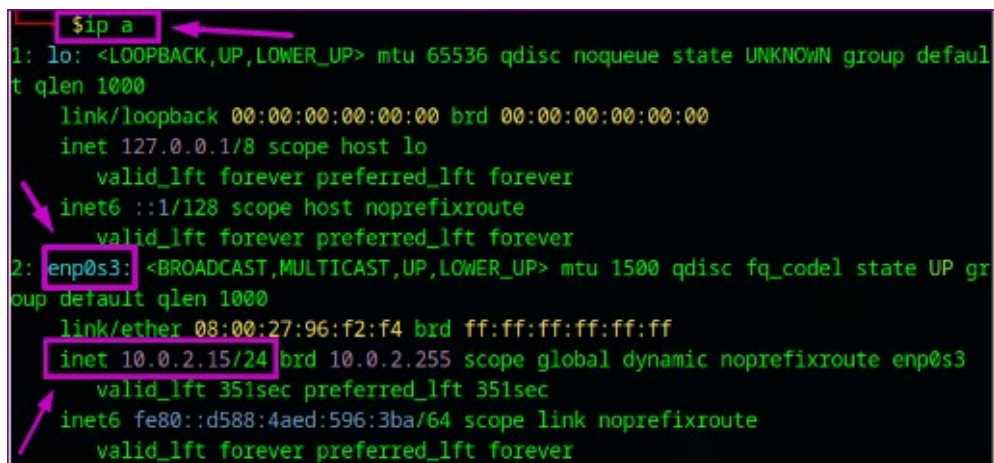
```
ip a  
ip addr
```

Esto nos proporcionará la información necesaria para seleccionar la interfaz correcta.

Nuestra interfaz es <enp0s3>, por lo que la escribimos en el cuadro correspondiente donde se nos solicita la interfaz en la que **Snort** escuchará el tráfico de red.

➔ **Pulsamos la tecla Tabulador** para mover la selección a **Aceptar**. ➔ **Presionamos Enter** para confirmar.

Acto seguido, se abrirá otra ventana donde se nos pedirá **el intervalo de direcciones de la red local**.

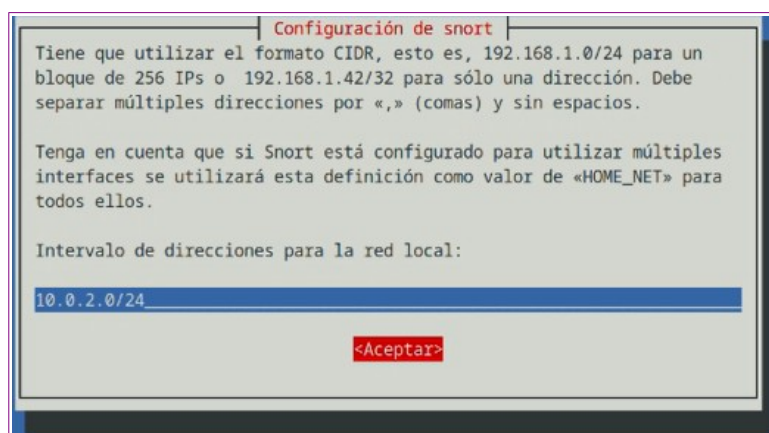
A terminal window showing the output of the 'ip a' command. The output lists network interfaces. Interface 'lo' (loopback) is shown first. Interface 'enp0s3' (Ethernet) is shown second and is highlighted with a pink box. Within the 'enp0s3' details, the line 'inet 10.0.2.15/24' is also highlighted with a pink box. Arrows point from the text in the surrounding paragraphs to these specific elements in the terminal output.

```
$ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    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 noprefixroute  
        valid_lft forever preferred_lft forever  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:96:f2:f4 brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 351sec preferred_lft 351sec  
    inet6 fe80::d588:4aed:596:3ba/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Esto nos mostrará nuestra dirección IP actual y para escuchar en **todo el segmento de la red**, ingresamos la siguiente dirección:

```
10.0.2.0/24
```

Pulsamos la tecla Tabulador para que la selección se marque en **rojo** sobre **Aceptar**. ➔ **Presionamos Enter** para confirmar la configuración.

A window titled 'Configuración de snort'. It contains instructions about CIDR notation and the HOME_NET variable. Below the text, there is a label 'Intervalo de direcciones para la red local:' followed by a text input field containing '10.0.2.0/24'. At the bottom right of the input area is a red button labeled '<Aceptar>'.

Configuración de snort

Tiene que utilizar el formato CIDR, esto es, 192.168.1.0/24 para un bloque de 256 IPs o 192.168.1.42/32 para sólo una dirección. Debe separar múltiples direcciones por «,» (comas) y sin espacios.

Tenga en cuenta que si Snort está configurado para utilizar múltiples interfaces se utilizará esta definición como valor de «HOME_NET» para todos ellos.

Intervalo de direcciones para la red local:

10.0.2.0/24

<Aceptar>

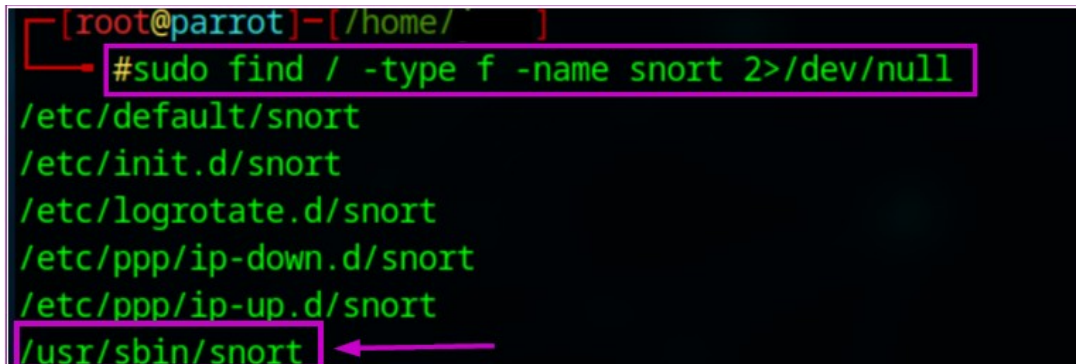
Verificación de la Instalación de Snort

Para verificar que **Snort** se ha instalado correctamente, ejecutamos el siguiente comando:

```
sudo find / -type f -name snort 2>/dev/null
```

Este comando nos permitirá **confirmar la ubicación del ejecutable**, que debería encontrarse en:

```
/usr/sbin/snort
```



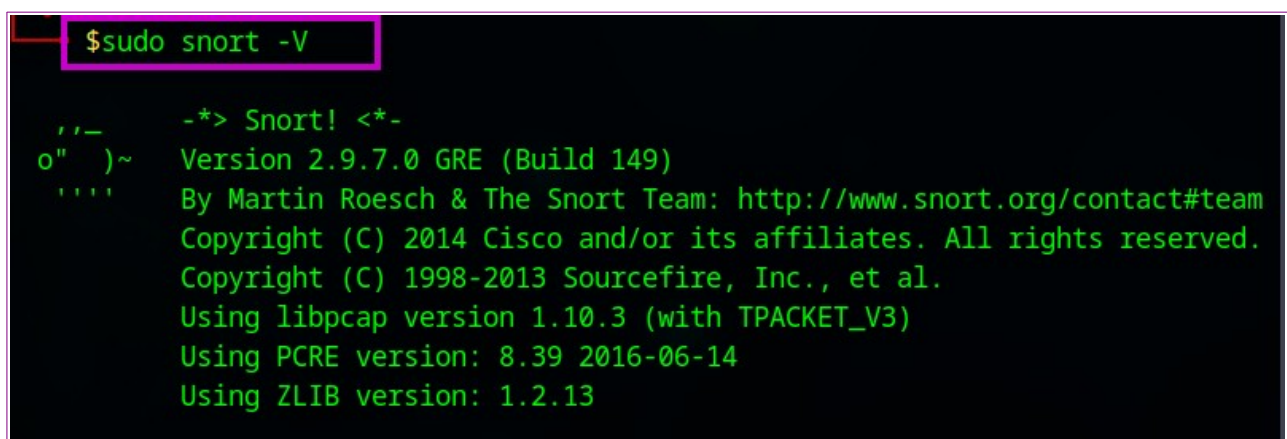
```
[root@parrot]~# sudo find / -type f -name snort 2>/dev/null
/etc/default/snort
/etc/init.d/snort
/etc/logrotate.d/snort
/etc/ppp/ip-down.d/snort
/etc/ppp/ip-up.d/snort
/usr/sbin/snort
```

Consulta de la Versión de Snort

Para obtener más información sobre **Snort**, ejecutamos el siguiente comando:

```
snort -V
```

Esto nos proporcionará la **versión instalada** de Snort en el sistema.



```
$sudo snort -V
_*> Snort! <*-
''_
o" )~ 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 1.10.3 (with TPACKET_V3)
      Using PCRE version: 8.39 2016-06-14
      Using ZLIB version: 1.2.13
```

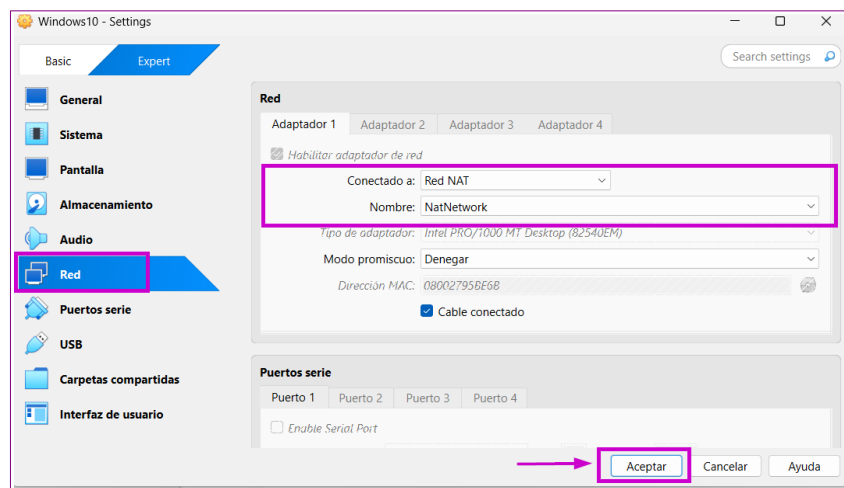
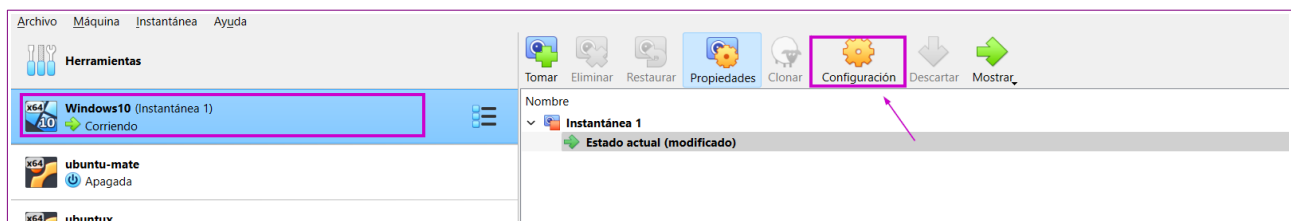
Montaje del Laboratorio con el IDS Escogido

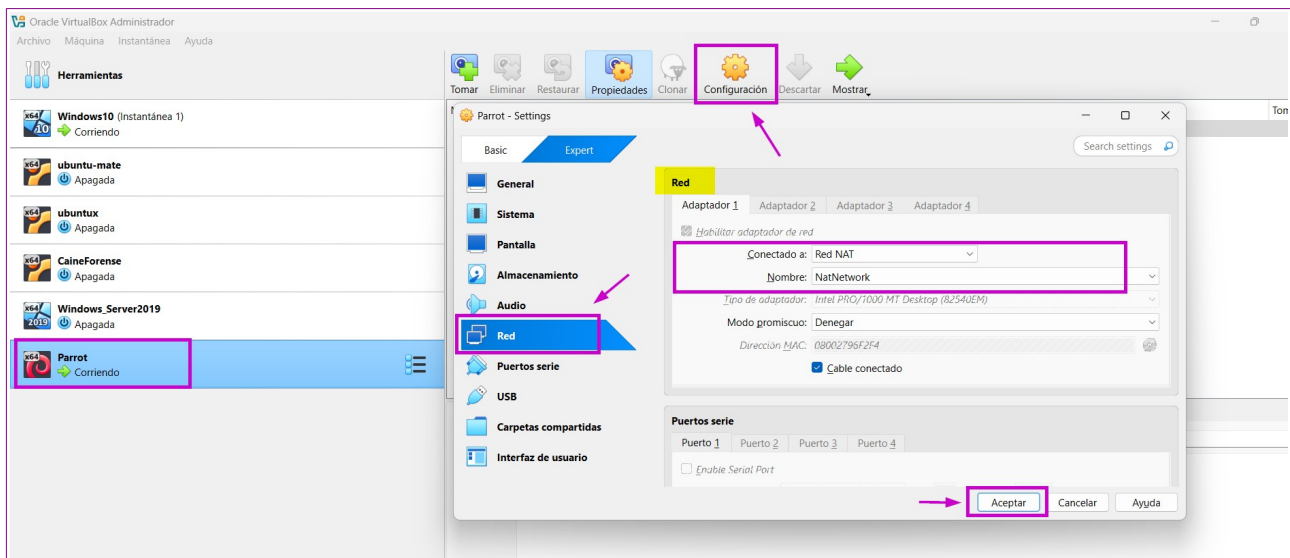
Para configurar nuestro **laboratorio**, utilizaremos **dos máquinas virtuales**: ➔ Una con **Parrot OS**. ➔ Otra con **Windows 10**.

Para que ambas máquinas **puedan comunicarse**, debemos asegurarnos de que están en la **misma red**.

Configuración de red:

1. Accedemos al apartado **Configuración** de cada máquina virtual.
2. Navegamos a la sección **Red**.
3. En *Conectado a*: seleccionamos **Red NAT**.
4. Elegimos el nombre de la red NAT (en nuestro caso, **NatNetwork**).
5. **Pulsamos Aceptar** para aplicar los cambios





Verificación de Conectividad entre Máquinas

Para afinar la comunicación entre las máquinas virtuales, realizamos un **ping** de **Parrot OS** a **Windows 10**.

Resultado: ➔ **El ping NO es correcto.** ➔ **Hay pérdidas de paquetes.**

Esto puede deberse a que el **Firewall de Windows Defender** está bloqueando los paquetes **ICMP**.

Solución

Para permitir la comunicación, debemos **crear una regla de entrada** que acepte estos paquetes y permita que el **ping funcione correctamente**.

```
[joan@parrot]~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
^C
--- 10.0.2.4 ping statistics ---
66 packets transmitted, 0 received, 100% packet loss, time 66426ms
```

¡ALERTA!
HAY PÉRDIDA DE PAQUETES, NO FUNCIONA EL PING

Configuración del Firewall en Windows 10

Prueba de conexión: Windows 10 ➔ Parrot OS *El ping es correcto, los paquetes se transmiten sin problemas.*

Sin embargo, el **Firewall de Windows Defender** puede bloquear los paquetes **ICMP entrantes**, impidiendo la comunicación cuando se envía un ping hacia **Windows 10**.

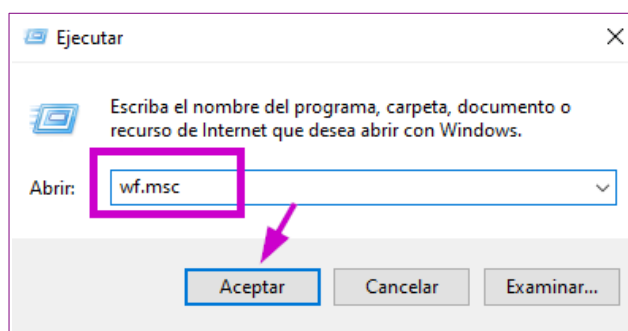
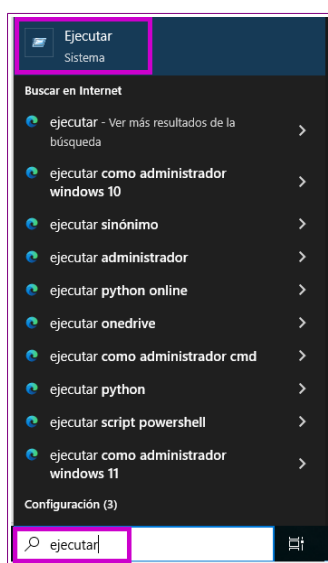
Solución

Para permitir estos paquetes, debemos **crear una regla de entrada** en **Windows Defender** siguiendo estos pasos:

1. **Abrimos el menú de búsqueda en Windows 10.**
2. **Escribimos "Ejecutar"** y hacemos clic sobre la aplicación.
3. **En el cuadro de texto, ingresamos:**

`wf.msc`

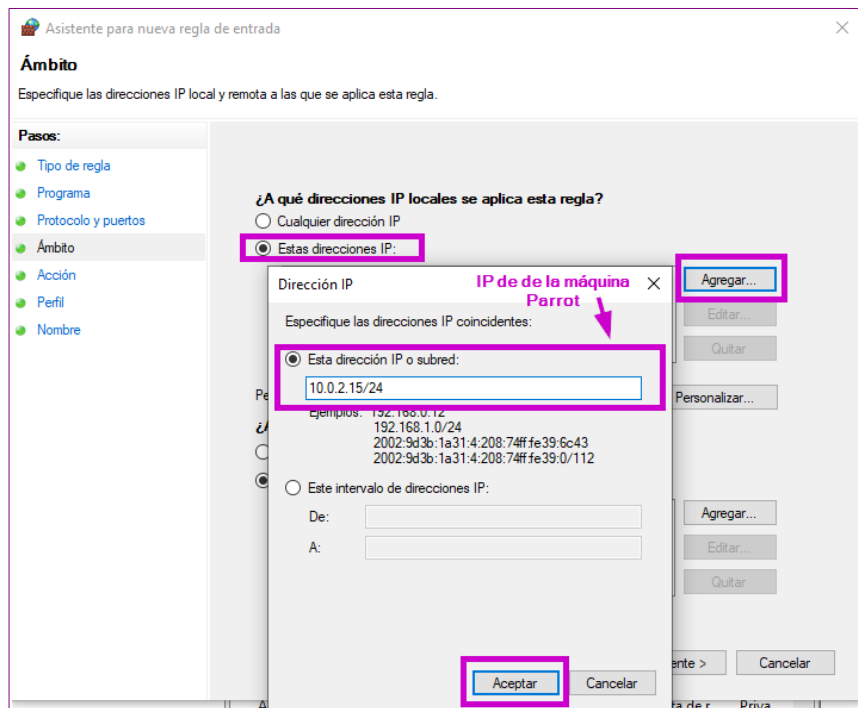
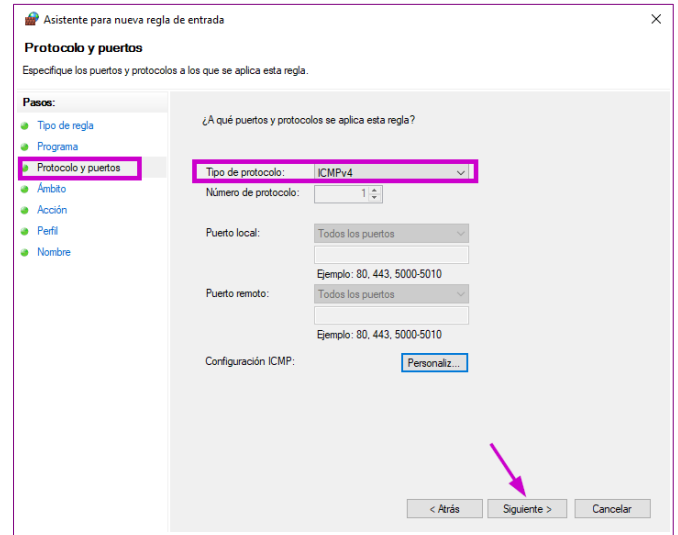
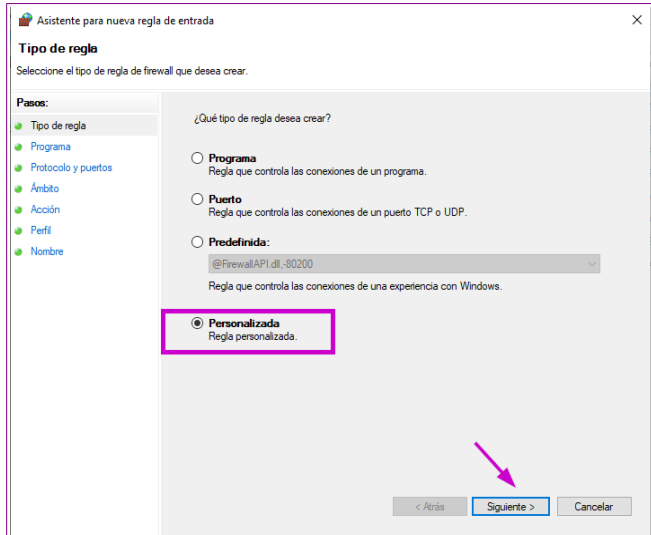
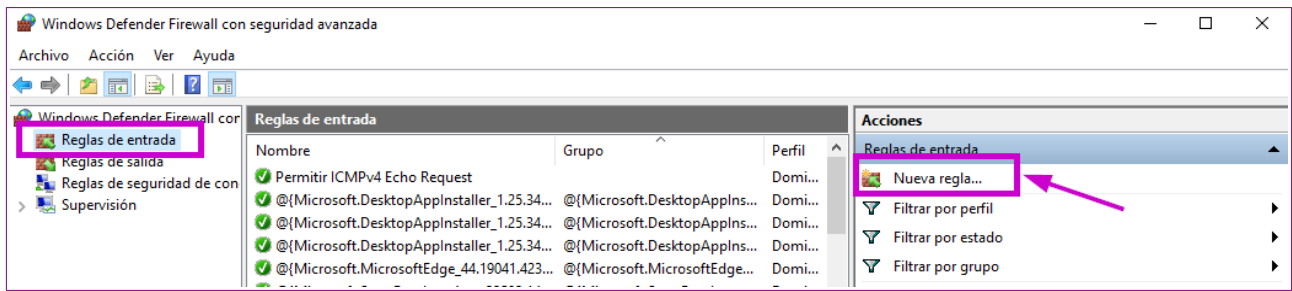
4. **Presionamos Aceptar** para acceder a la configuración del firewall.



Creación de Regla para Permitir ICMPv4 en Windows Defender

Una vez dentro de la configuración del **Firewall de Windows Defender**, seguimos estos pasos:

1. En el menú lateral izquierdo, **seleccionamos Reglas de entrada**.
2. En el panel derecho, **hacemos clic en Añadir nueva regla**.
3. **Configuramos** la regla para permitir **ICMPv4**, asegurando que Windows acepte el **ping** desde nuestra máquina **Parrot OS**.
4. Podemos establecer la regla para que afecte: → **Un rango de direcciones IP** → **Una IP específica** correspondiente a la máquina que deseamos comunicar.



Configuración de la Dirección IP o Subred

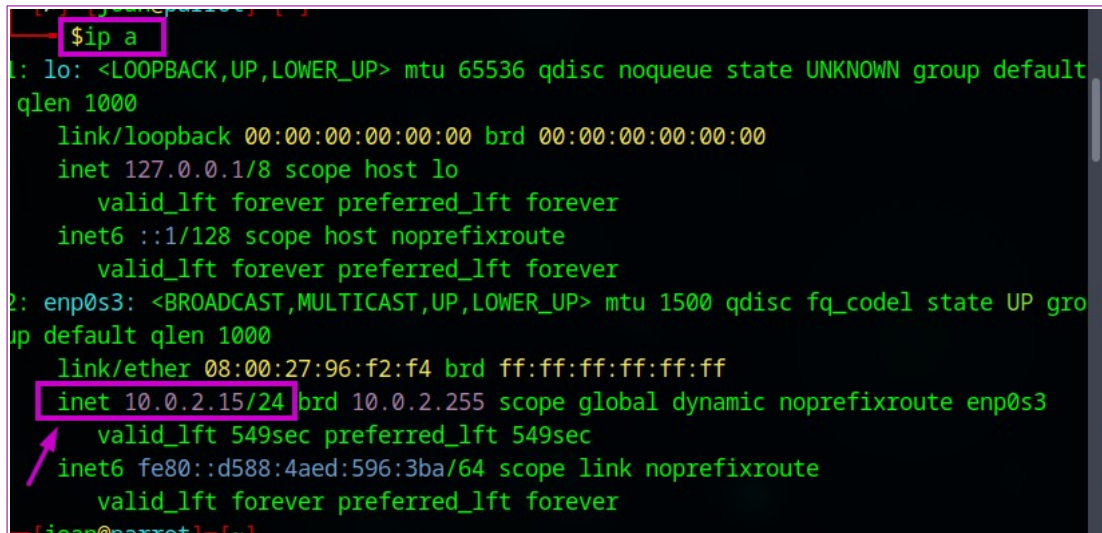
En el campo de **dirección IP o subred**, ingresamos la correspondiente a nuestra máquina **Parrot OS**:

10.0.2.15/24

Podemos obtener esta información desde la **consola de Parrot**, ejecutando el siguiente comando:

ip a

Esto mostrará los detalles de la configuración de red, como se observa en la imagen de referencia.



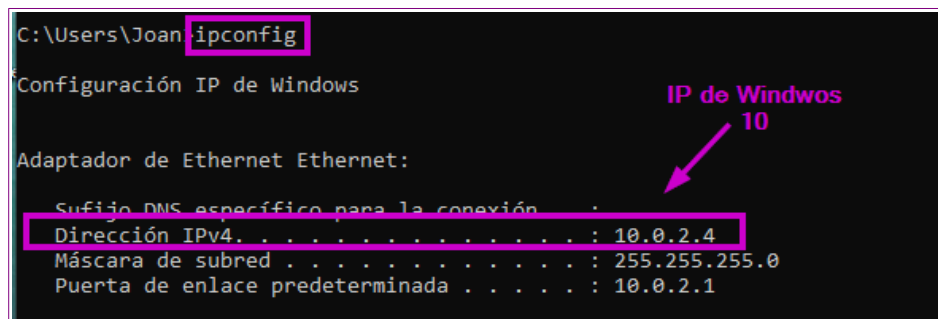
```
$ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:96:f2:f4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 549sec preferred_lft 549sec
    inet6 fe80::d588:4aed:596:3ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Consulta de la Dirección IP en Windows 10

Para conocer la **dirección IP** de nuestra máquina **Windows 10**, abrimos la **Consola de comandos (cmd)** y ejecutamos:

ipconfig

Este comando mostrará la información de la configuración de red, incluyendo la dirección IP asignada.



```
C:\Users\Joan>ipconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión . . . . . : 
    Dirección IPv4. . . . . : 10.0.2.4
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 10.0.2.1
```

Verificación de Conectividad con Ping

Volvemos a ejecutar las pruebas de **ping**: ➔ Desde Windows 10 hacia Parrot OS. ➔ Observamos los resultados para verificar la conectividad entre las máquinas.

```
C:\Users\Joan>ping 10.0.2.15

Haciendo ping a 10.0.2.15 con 32 bytes de datos:
Respuesta desde 10.0.2.15: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.0.2.15: bytes=32 tiempo=18ms TTL=64
Respuesta desde 10.0.2.15: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.0.2.15: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 10.0.2.15:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 18ms, Media = 4ms
```

OK, no se pierden paquetes

Verificación de Conectividad con Ping (Parrot → Windows 10)

Ahora realizamos la prueba de **ping** en sentido contrario: Desde Parrot OS hacia Windows 10. ➔ Observamos los resultados para verificar la correcta comunicación entre ambas máquinas.

```
[joan@parrot]~$ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=128 time=0.639 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=128 time=0.379 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=128 time=0.337 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=128 time=0.373 ms
64 bytes from 10.0.2.4: icmp_seq=5 ttl=128 time=2.55 ms
64 bytes from 10.0.2.4: icmp_seq=6 ttl=128 time=0.555 ms
64 bytes from 10.0.2.4: icmp_seq=7 ttl=128 time=0.371 ms
```

```
--- 10.0.2.4 ping statistics ---
123 packets transmitted, 123 received, 0% packet loss, time 124281ms
rtt min/avg/max/mdev = 0.279/0.842/7.051/1.011 ms
```

OK, no hay pérdida de paquetes

Estado de la Conexión

Ahora **no observamos pérdida de paquetes**. Los **tiempos de respuesta** (*rtt min/avg/max/mdev*) son bastante bajos.

Esto sugiere que la conexión es **estable y eficiente**.

Implementación de Casos de Uso para el IDS

Configuración Previa del Entorno para Gestionar Alertas

Según lo que hemos investigado sobre **Snort**, este puede generar **logs** de manera inmediata en la misma consola de comandos o guardarlos en un **archivo específico** para su posterior revisión.

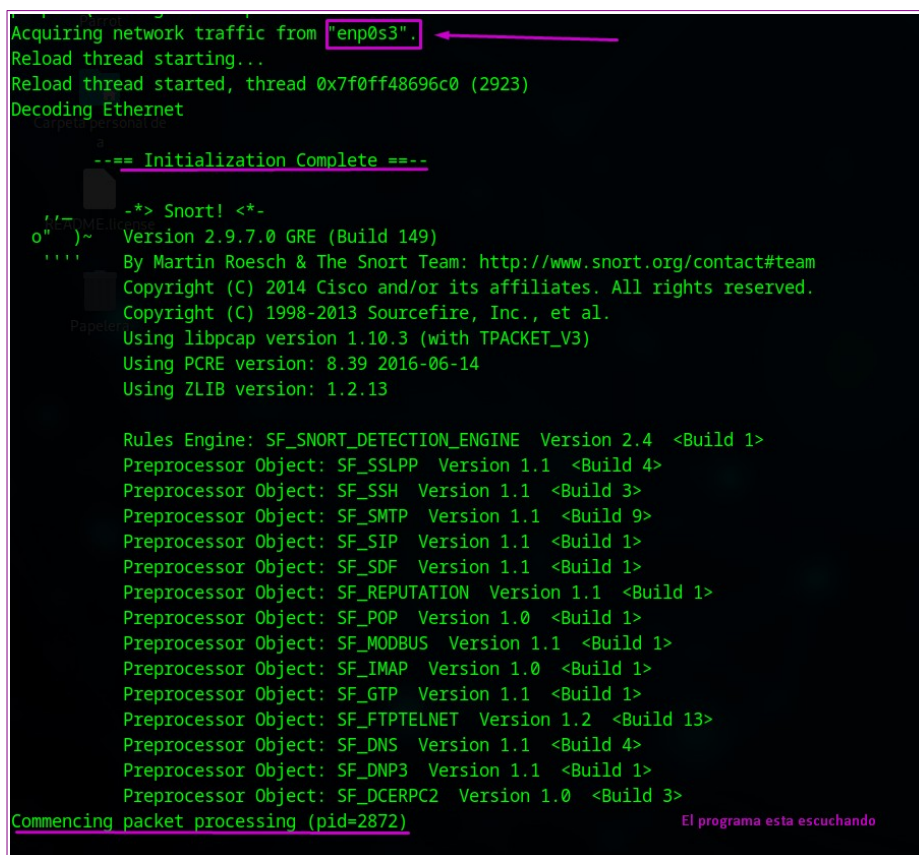
Para habilitar esta función, utilizamos el siguiente comando de configuración:

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Explicación del comando:

- i enp0s3 → Ejecuta Snort en la interfaz de red enp0s3.
- c /etc/snort/snort.conf → Usa las reglas definidas en el archivo principal de configuración snort.conf.
- l /var/log/snort → Guarda las alertas en la ruta /var/log/snort.
- A fast → Utiliza el formato **simple** para almacenar las alertas.

Una vez introducido el comando, **Snort se ejecutará y comenzará a monitorear la red en la interfaz enp0s3.**



```
Acquiring network traffic from "enp0s3".
Reload thread starting...
Reload thread started, thread 0x7f0ff48696c0 (2923)
Decoding Ethernet
--== Initialization Complete ==--
-*> 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 1.10.3 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.13

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Commencing packet processing (pid=2872)
```

Creación de Reglas en Snort

En este punto, podemos comenzar a **crear las reglas** que requiere el documento.

Las reglas se guardarán en el siguiente archivo:

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

Este archivo permitirá definir **las reglas de detección** para el **IDS Snort**, asegurando una correcta configuración.



```
[root@parrot]-[/home/a]  
#nano /etc/snort/rules/local.rules
```

Creación de una Alerta para Ping a la IP 8.8.8.8

Para configurar la **alerta**, accedemos al archivo de reglas en Snort:

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

Dentro del archivo, escribimos la siguiente regla:

```
alert icmp any any -> 8.8.8.8 any (msg:"Ping a 8.8.8.8 detectado"; sid:1000001;  
rev:1;)
```

Explicación de la regla:

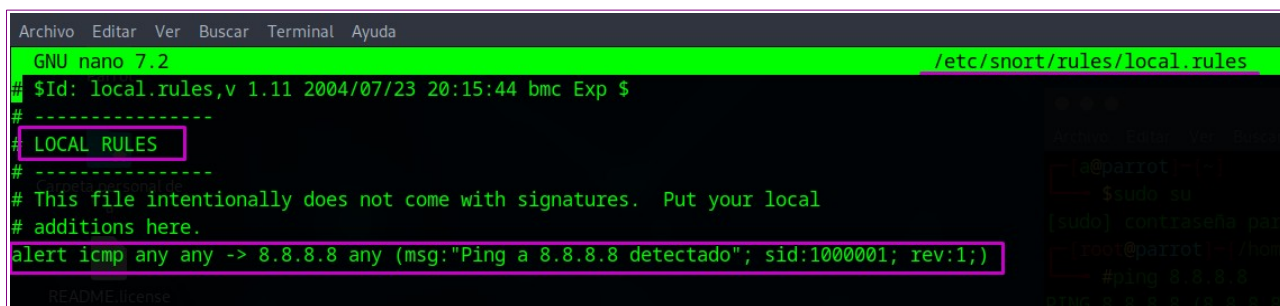
`alert icmp any any -> 8.8.8.8 any` → Activa una alerta cuando se detecte tráfico **ICMP** dirigido a 8.8.8.8.

`msg:"Ping a 8.8.8.8 detectado"` → Mensaje de alerta que aparecerá en los registros.

`sid:1000001` → Identificador único de la regla.

`rev:1` → Número de versión de la regla.

Esta configuración permitirá que **Snort detecte y registre cualquier intento de ping a la dirección 8.8.8.8**.



```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda  
GNU nano 7.2 /etc/snort/rules/local.rules  
# $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 -> 8.8.8.8 any (msg:"Ping a 8.8.8.8 detectado"; sid:1000001; rev:1;)  
README: https://www.snort.org/docs/snort-docs/local_rules.php
```

Ejecución de Snort

Con la configuración completada, procedemos a ejecutar **Snort** con el siguiente comando:

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Este comando iniciará **Snort** en la interfaz `enp0s3`, aplicando las reglas definidas en el archivo de configuración `snort.conf` y almacenando las alertas en `/var/log/snort` con el formato **rápido** (`-A fast`).

```
==== Initialization Complete ====

-*> 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 1.10.3 (with TPACKET_V3)
Using PCRE version: 8.39 2016-06-14
Using ZLIB version: 1.2.13

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.4 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Commencing packet processing (pid=2946)
```

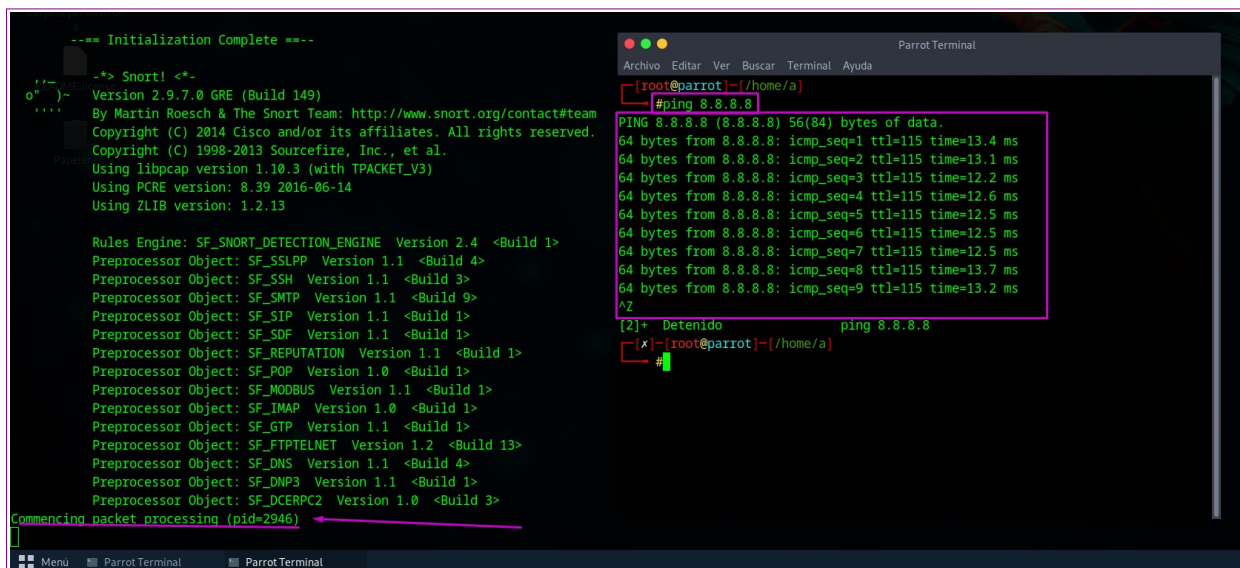
Inicio de Snort y Aplicación de Reglas

El siguiente comando inicia **Snort** en la interfaz **enp0s3**:

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Acciones realizadas:

- Se aplican las reglas definidas en el archivo de configuración `snort.conf`.
- Las alertas generadas se almacenan en `/var/log/snort`.
- Se utiliza el formato **rápido** (`-A fast`) para la captura de alertas.



The screenshot shows a Parrot Terminal window with two panes. The left pane displays the Snort initialization output, including version information (2.9.7.0 GRE, Build 149) and a list of preprocessor objects (SF_SSH, SF_SMTP, SF_SIP, SF_SDF, SF_REPUTATION, SF_POP, SF_MODBUS, SF_IMAP, SF_GTP, SF_FTPTELNET, SF_DNS, SF_DNP3, SF_DCERPC2). The right pane shows a root user at the parrot machine in the /home/a directory, running a ping command to 8.8.8.8. The output shows 9 successful ping requests with varying times (13.1 ms to 13.7 ms). The terminal window has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'.

Verificación de Alertas en Snort

Una vez finalizada la ejecución de **Snort**, lo apagamos con:

CTRL + Z

Luego, comprobamos que **Snort** está guardando las alertas en el archivo indicado:

`/var/log/snort/alert`

Opciones para visualizar las alertas:

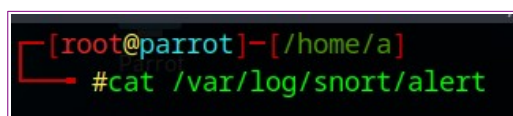
1. Imprimir el contenido en la terminal con el siguiente comando:

`cat /var/log/snort/alert`

2. Abrir el archivo y leerlo directamente utilizando un editor de texto como:

`nano /var/log/snort/alert`

Dependiendo de nuestras preferencias, utilizamos **uno u otro** para inspeccionar las alertas registradas por Snort.



A terminal window showing the command `#cat /var/log/snort/alert` being entered at the prompt `[root@parrot]-[/home/a]`.

Abrir el Archivo de Alertas de Snort

Para visualizar las alertas registradas por **Snort**, utilizamos el siguiente comando:

`nano /var/log/snort/alert`

Esto abrirá el archivo en el editor de texto **Nano**, permitiéndonos **leer y analizar** las alertas generadas.



A terminal window showing the command `#nano /var/log/snort/alert` being entered at the prompt `[root@parrot]-[/home/a]`.

Comprobación de Alertas Registradas por Snort

Hemos utilizado el siguiente comando para **visualizar las alertas dentro del archivo**:

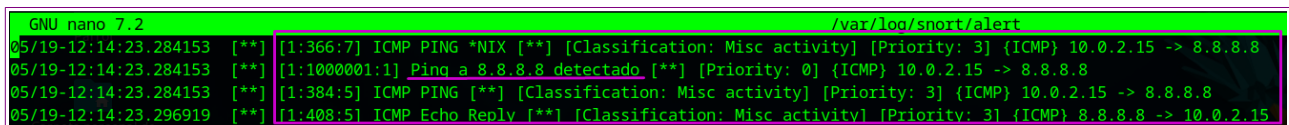
```
nano /var/log/snort/alert
```

Al acceder al archivo, confirmamos que **Snort ha registrado correctamente los pings realizados**, verificando así su funcionamiento.

Confirmación del Funcionamiento de la Regla

Tras la comprobación de alertas, podemos afirmar que **la regla funciona correctamente**.

Ahora podemos **proceder con la configuración de la siguiente regla** para continuar con la implementación.



```
GNU nano 7.2 /var/log/snort/alert
05/19-12:14:23.284153  [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 10.0.2.15 -> 8.8.8.8
05/19-12:14:23.284153  [**] [1:1000001:1] Ping a 8.8.8.8 detectado. [**] [Priority: 0] {ICMP} 10.0.2.15 -> 8.8.8.8
05/19-12:14:23.284153  [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 10.0.2.15 -> 8.8.8.8
05/19-12:14:23.296919  [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 8.8.8.8 -> 10.0.2.15
```

Creación de una Alerta para Consultas Web (HTTP/HTTPS)

Para generar una alerta cuando se realice una **consulta web**, creamos una nueva regla dentro del archivo de Snort:

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

Dentro del archivo, agregamos la siguiente regla:

```
alert tcp any any -> any [80,443,8080,8443] (msg:"Conexión web detectada
(HTTP/HTTPS)"; sid:1000002; rev:1;)
```

Explicación de la regla:

alert tcp any any -> any [80,443,8080,8443] → Detecta tráfico **TCP** en los puertos típicos de navegación web (80, 443, 8080, 8443).

msg:"Conexión web detectada (HTTP/HTTPS)" → Mensaje de alerta que aparecerá en los registros.

sid:1000002 → Identificador único de la regla.

rev:1 → Número de versión de la regla.

Con esta configuración, **Snort generará una alerta cada vez que se realice una solicitud HTTP o HTTPS**.

```
GNU nano 7.2 /etc/snort/rules/local.rules
# $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 -> 8.8.8.8 any (msg:"Ping a 8.8.8.8 detectado"; sid:1000001; rev:1;)
alert tcp any any -> any [80,443,8080,8443] (msg:"Conexion web detectada (HTTP/HTTPS)"; sid:1000002; rev:1;)
#-----
#addsnort security2
```

Comprobación del Funcionamiento de la Regla de Consultas Web

Iniciar Snort:

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Generar tráfico web:

- Abrimos el **navegador web** en nuestra máquina **Parrot OS**.
- Navegamos por **diferentes sitios de internet** para generar solicitudes web mientras Snort está activo.

Comprobar el archivo de logs:

- Accedemos al archivo de alertas de Snort:

```
cat /var/log/snort/alert
```
- Verificamos que **la regla ha generado las alertas correctamente**, registrando las conexiones web según nuestras especificaciones.

```
05/20-07:25:20.671730  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:38534 -> 142.250.200.202:443
05/20-07:25:46.183246  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:48046 -> 172.217.17.14:443
05/20-07:25:46.210144  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:48058 -> 172.217.17.14:443
05/20-07:25:46.217252  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:48064 -> 172.217.17.14:443
05/20-07:25:46.233917  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:48018 -> 172.217.17.14:443
05/20-07:25:46.293972  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:48030 -> 172.217.17.14:443
05/20-07:25:50.953475  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:36686 -> 142.250.200.202:443
05/20-07:25:53.484641  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:34578 -> 142.250.200.74:443
05/20-07:25:58.967027  ** [1:1000002:1] Conexion web detectada (HTTP/HTTPS) ** [Priority: 0] (TCP) 10.0.2.15:41760 -> 69.43.111.82:80
```

Verificación de Alertas de Consultas Web

Al visualizar el archivo de alertas, podemos comprobar que la regla **se cumple correctamente** y que Snort **registra nuestras peticiones web**.

Además, es posible determinar **si la conexión es HTTP o HTTPS**, observando el **puerto por el que se establece**:

- **HTTP** → Puerto 80
- **HTTPS** → Puerto 443

Esto nos permite **identificar el tipo de tráfico web** que ha sido detectado por Snort.

Creación de una alerta para el uso de Dropbox

Para registrar la conexión web al servicio **Dropbox**, creamos la siguiente regla dentro del archivo de Snort:

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

Dentro del archivo, agregamos la siguiente regla:

```
alert tcp any any -> any any (msg:"ALERTA: Uso de Dropbox detectado";  
content:"dropbox"; nocase; sid:1000003; rev:1;)
```

Explicación de la regla:

`alert tcp any any -> any any` → Genera una alerta para cualquier conexión **TCP**.

- `content:"dropbox"` → Busca la palabra "dropbox" en los paquetes de datos para detectar actividad relacionada con el servicio.
- `nocase` → Ignora mayúsculas y minúsculas en la búsqueda del contenido.
- `sid:1000003` → Identificador único de la regla.
- `rev:1` → Número de versión de la regla.

Esta configuración permitirá que **Snort detecte y registre cualquier intento de conexión al servicio Dropbox**.

```
05/21-14:01:53.647709  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36442  
05/21-14:01:53.647764  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36470  
05/21-14:01:53.650298  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36480  
05/21-14:01:53.650772  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36456  
05/21-14:01:53.651413  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36462  
05/21-14:01:53.651601  [**] [1:1000001:1] ALERTA: Uso de Dropbox detectado [**] [Priority: 0] (TCP) 104.16.100.29:443 -> 10.0.2.15:36478
```

Detección de Conexiones a Dropbox en Snort

Inicialmente, se utilizó una regla simple basada en la coincidencia de contenido "Dropbox":

```
alert tcp any any -> any any (msg:"ALERTA: Uso de Dropbox detectado";  
content:"dropbox"; nocase; sid:1000003; rev:1;)
```

Sin embargo, se detectaron **problemas de funcionamiento en algunos dispositivos**, lo que llevó a buscar alternativas más **efectivas y consistentes**.

Segunda Regla: Inspección de TLS SNI

Para mejorar la detección, se diseñó una nueva regla:

```
alert tcp any any -> any 443 (msg:"Uso de Dropbox detectado"; tls_sni;  
pcre:"/dropbox.com|dropboxapi.com|dl.dropboxusercontent.com/i"; sid:1000003;  
rev:1;)
```

Características de la regla:

- Filtra conexiones a **cualquier IP** con tráfico HTTPS (443).
- Inspecciona el **handshake TLS**, analizando el campo **SNI** para identificar el **nombre del dominio**.
- Busca coincidencias con los dominios `dropbox.com`, `dropboxapi.com`, `dl.dropboxusercontent.com`.

⚠ **Limitación:** Snort no admite **TLS SNI** en versiones anteriores a la 3.x. Como el entorno de trabajo usa la versión 2.9.7, fue necesario buscar otra alternativa. ⚠

Tercera Regla: Detección por Rango de IP

Para garantizar la funcionalidad, se optó por una regla basada en la **detección de direcciones IP** de Dropbox:

```
alert ip any any -> 162.125.0.0/16 any (msg:"ALERTA: Conexión IP a Dropbox";  
sid:1000003; rev:1;)
```

Características de la regla:

- Se activa si la conexión está dirigida al **rango de IPs 162.125.0.0/16**, asignado a **Dropbox**.
- No depende de inspección TLS ni del contenido de los paquetes.

Esta última regla garantiza **compatibilidad total**, proporcionando una detección más fiable en el entorno de trabajo.

```
05/23-15:28:05.938862  [**] [1:1000003:1] ALERTA: Conexión IP a Dropbox [**] [Priority: 0] (ICMP) 10.0.2.15 -> 162.125.68.18
```

Posible Inconveniente

Si bien la regla cumple su función y **Snort genera la alerta esperada**, existe un **riesgo potencial**: ⚠
Si Dropbox modifica el rango de IP que utiliza, la regla dejará de ser efectiva, lo que afectaría su fiabilidad a largo plazo. ⚠

Conclusión

Dado este escenario, la **solución óptima** sería **actualizar Snort a la versión 3 o superior**, lo que permitiría **garantizar esta funcionalidad y minimizar posibles fallos en conexiones similares**.

Sin embargo, hemos decidido **adaptarnos a la versión actual (2.9.7)** y explorar **alternativas** para seguir trabajando en un **entorno estable** sin necesidad de actualizar.

Creación de una Alerta para Conexiones Salientes SSH

Objetivo del ejercicio: Desarrollar una regla que **detecte y genere una alerta** ante una conexión saliente SSH.

Regla creada:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 22 (msg:"Conexión saliente SSH detectada"; sid:1000004; rev:1;)
```

Explicación de la regla:

- `alert tcp` → Genera una alerta para tráfico **TCP**.
- `$HOME_NET any` → Detecta conexiones desde la **red interna** (`$HOME_NET`).
- `$EXTERNAL_NET 22` → Registra tráfico dirigido a **cualquier dirección externa** (`$EXTERNAL_NET`) en el **puerto 22**, utilizado en conexiones **SSH**.
- `msg:"Conexión saliente SSH detectada"` → Mensaje que se mostrará en las alertas.
- `sid:1000004` → Identificador único de la regla.
- `rev:1` → Número de versión de la regla.

Esta configuración permite que **Snort detecte y registre cualquier intento de conexión SSH saliente**.

```
[root@parrot]~[/home/joan]
#sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
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 183
8085 8088 8090 8118 8123 8180:8181 8243 8280 8300 8800 8888 8899 9000 9060 908
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 ]
```

Guardado de la Regla en Snort

Como en los pasos anteriores, adjuntamos la regla en el archivo correspondiente de Snort:

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

```
GNU nano 7.2 /etc/snort/rules/local.rules
# $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 -> 8.8.8.8 any (msg:"Ping a 8.8.8.8 detectado"; sid:1000001; rev:1;)
alert tcp any any -> any [80,443,8080,8443] (msg:"Conexion web detectada (HTTP/HTTPS)"; sid:1000002; rev:1;)
alert tcp $HOME_NET any -> $EXTERNAL_NET 22 (msg:"Conexion saliente SSH detectada"; sid:1000004; rev:1;)
```

Ejecución de Snort y Configuración para Captura de SSH

Iniciar Snort

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Verificamos que **Snort está funcionando correctamente** antes de proceder con la conexión SSH.

⚠ **IMPORTANTE:** Antes de realizar la conexión SSH, es necesario una **configuración previa**. Si no la realizamos, **Snort no capturará la conexión**, y aparecerá el error de “**conexión rechazada**”. ⚠

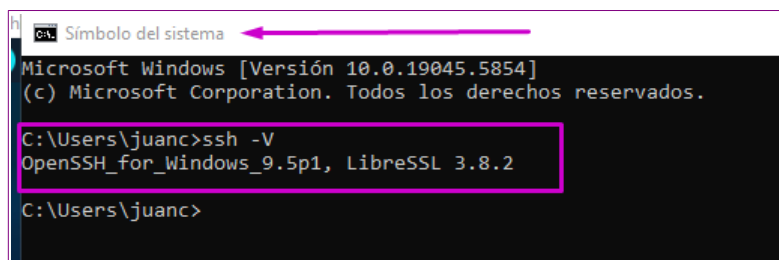
Pasos para Configurar SSH en Windows

Comprobar el estado del puerto 22 y el servicio SSH:

- Desde la **consola CMD de Windows**, ejecutamos:

```
ssh -V
```

Este comando nos mostrará la **versión de SSH**, asegurando que el servicio está habilitado.



```
Microsoft Windows [Versión 10.0.19045.5854]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\juanc>ssh -V
OpenSSH_for_Windows_9.5p1, LibreSSL 3.8.2

C:\Users\juanc>
```

Instalación y Configuración del Servicio SSH

Si el servicio **SSH** no está instalado en la máquina, lo instalamos con los siguientes comandos:

```
sudo apt update
sudo apt install ssh
```

Verificación del estado de SSH: Para comprobar si el servicio está en ejecución, utilizamos:

```
sudo systemctl status ssh
```

Activación del servicio SSH si no está corriendo: Si el servicio **SSH** está instalado pero no activo, lo iniciamos con los siguientes comandos:

```
sudo systemctl enable ssh
sudo systemctl start ssh
```

Apertura del Puerto 22 en el Firewall

⚠ **Por defecto, el firewall bloquea el puerto 22**, impidiendo las conexiones SSH. ⚠

Para permitir la conexión, ejecutamos:

```
sudo ufw allow 22
```

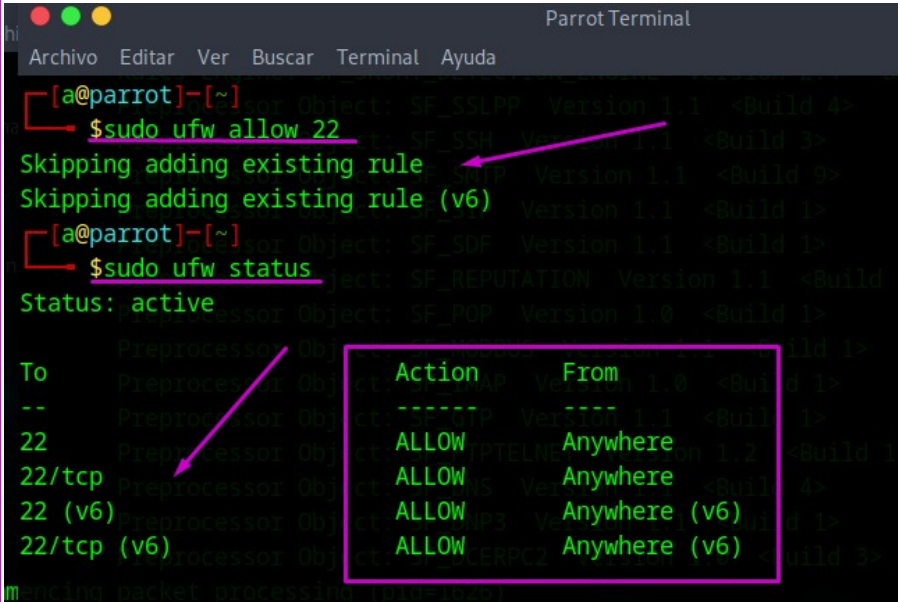
Verificación del estado del firewall

```
sudo ufw status
```

Esto nos mostrará la configuración actual del firewall, permitiéndonos verificar que el puerto **22** ha sido correctamente habilitado.

En caso de no tenerlo los instalaríamos con los comandos: 'sudo apt update' y 'sudo apt install ssh', en ese orden. Una vez instalado podemos ver si está funcionando con 'sudo systemctl status ssh'. En el caso de que no esté funcionando, pero si instalado, lo arrancaremos con los comandos 'sudo systemctl enable ssh' seguido del comando 'sudo systemctl start ssh'.

Para observar el estado del puerto 22, el cual el firewall lo tenía cerrado, lo hemos abierto mediante la ejecución del comando 'sudo ufw allow 22'. Seguidamente lo comprobamos con el 'sudo ufw status' y observamos la siguiente información:



```
Parrot Terminal
Archivo Editar Ver Buscar Terminal Ayuda

[a@parrot]-[~]
$ sudo ufw allow 22
Skipping adding existing rule
Skipping adding existing rule (v6)
[a@parrot]-[~]
$ sudo ufw status
Status: active

To: Preprocessor Object: SF_SSH Version 1.1 <Build 3>
-- Preprocessor Object: SF_SSH Version 1.1 <Build 3>
22 Preprocessor Object: SF_SSH Version 1.1 <Build 3>
22/tcp Preprocessor Object: SF_SSH Version 1.1 <Build 3>
22 (v6) Preprocessor Object: SF_SSH Version 1.1 <Build 3>
22/tcp (v6) Preprocessor Object: SF_SSH Version 1.1 <Build 3>
```

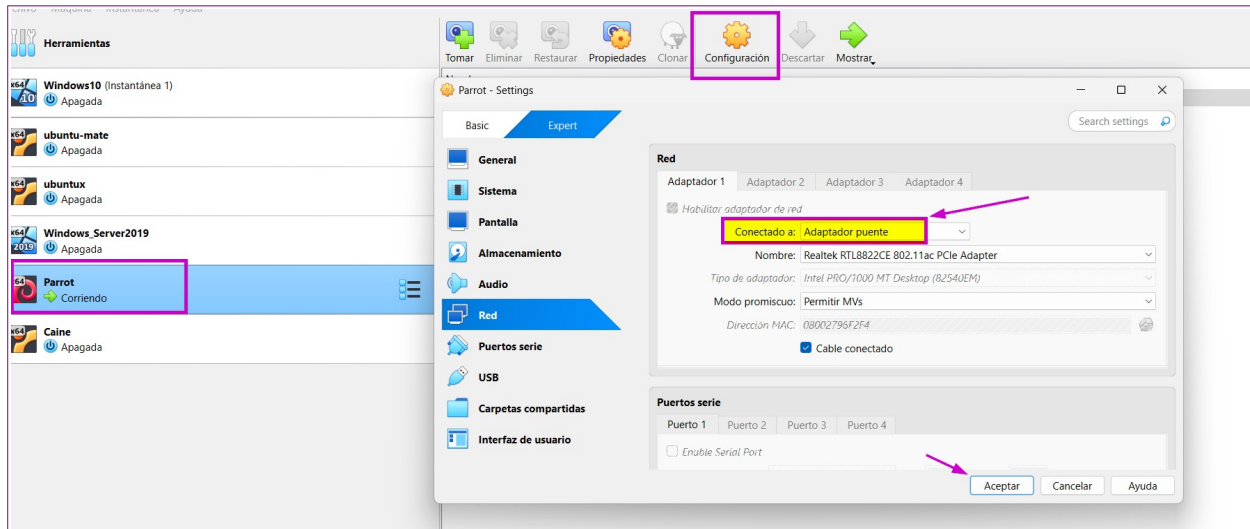
Action	From
ALLOW	Anywhere
ALLOW	Anywhere (v6)
ALLOW	Anywhere (v6)
ALLOW	Anywhere (v6)

Cambio de Configuración de Red en Parrot OS

Con **todo preparado**, pasamos al siguiente paso:

1. **Apagar la máquina virtual Parrot OS.**
2. **Acceder a la configuración de red** de la máquina virtual.
3. **Modificar el ajuste de red**, estableciendo el modo **adaptador puente**.
4. **Guardar los cambios y reiniciar la máquina virtual** para aplicar la nueva configuración.

En la **imagen adjunta**, se muestra el procedimiento exacto.



Verificación de la Dirección IP en Parrot OS

Encendemos la máquina virtual Parrot OS nuevamente.

Accedemos a la **terminal** y ejecutamos el siguiente comando:

`ip a`

Esto nos mostrará la **configuración de la interfaz de red**, incluida la **nueva dirección IP** asignada tras el cambio a **modo adaptador puente**.

Este dato será necesario en el siguiente paso para continuar con la configuración de la conexión.

```
$ip a
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 noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:96:f2:f4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.49/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
        valid_lft 42747sec preferred_lft 42747sec
    inet6 fe80::d588:4aed:596:3ba/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ejecución de Snort y Establecimiento de Conexión SSH

Iniciamos Snort en la máquina Parrot OS con el siguiente comando:

```
sudo snort -i enp0s3 -c /etc/snort/snort.conf -l /var/log/snort -A fast
```

Esto permite que Snort monitoree la actividad de red y detecte conexiones SSH salientes.

Realizamos la petición SSH desde nuestra máquina host utilizando la siguiente sintaxis:

```
ssh nombre_de_usuario_de_parrot@dirección_ip_de_parrot
```

Ejemplo práctico: Si el usuario en Parrot OS se llama joan y la IP asignada es 192.168.1.49, entonces el comando sería:

```
ssh joan@192.168.1.49
```

```
Microsoft Windows [Versión 10.0.26100.4061]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\navij>ssh joan@192.168.1.49
The authenticity of host '192.168.1.49 (192.168.1.49)' can't be established.
ED25519 key fingerprint is SHA256:ZkXI0G6yRqFhkI7yq/01YrhTRrehUnvFcSg49jgv50.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.49' (ED25519) to the list of known hosts.
joan@192.168.1.49's password:
Linux parrot 6.12.12-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.12-1parrot1 (2025-02-27) x86_64

[ASCII Art Logo]

The programs included with the Parrot GNU/Linux are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Parrot GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue May 20 23:51:45 2025 from 10.0.2.15
joan@parrot:~$
```

Annotations in the image:

- Realizamos la conexión SSH a la ip de nuestra máquina Parrot (points to the command).
- Acceptamos que se siga conectando escribiendo 'Yes' (points to the 'yes' response).
- Escribimos la contraseña del root de Parrot (points to the password prompt).
- FUNCIONA CORRECTAMENTE. No sale ningún error y por el menú que observamos vemos que nos hemos conectado a la máquina de Parrot (points to the successful login prompt).

Proceso de Autenticación y Verificación de la Conexión SSH

Al ejecutar la conexión SSH, la terminal solicitará algunas comprobaciones de seguridad:

1. Confirmación de confianza en la dirección IP:

- Se nos pedirá **aceptar o denegar** la asociación de la IP al fichero de **usuarios conocidos**.
- En nuestro caso, seleccionamos **"yes"** para continuar.

2. Ingreso de credenciales:

- La terminal nos solicitará la **contraseña del usuario de Parrot OS**.
- Debemos ingresarla correctamente para completar la autenticación.

3. Acceso exitoso a la consola de Parrot OS:

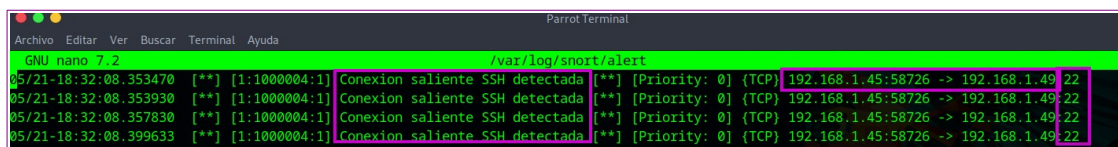
- Si todo ha sido configurado correctamente, podremos visualizar la interfaz de consola de **Parrot OS**, confirmando que la conexión SSH se ha establecido satisfactoriamente.

Verificación de la Regla de Snort

Como paso final, accedemos al **archivo de alertas de Snort** para comprobar que la detección ha funcionado:

```
cat /var/log/snort/alert
```

En este archivo observaremos que **Snort ha generado la alerta correctamente**, registrando tanto las **direcciones IP como el puerto utilizado (22)** según lo establecido en la regla.



```
GNU nano 7.2 /var/log/snort/alert
05/21-18:32:08.353470  [**] [1:1000004:1] Conexión saliente SSH detectada [**] [Priority: 0] (TCP) 192.168.1.45:58726 -> 192.168.1.49:22
05/21-18:32:08.353930  [**] [1:1000004:1] Conexión saliente SSH detectada [**] [Priority: 0] (TCP) 192.168.1.45:58726 -> 192.168.1.49:22
05/21-18:32:08.357830  [**] [1:1000004:1] Conexión saliente SSH detectada [**] [Priority: 0] (TCP) 192.168.1.45:58726 -> 192.168.1.49:22
05/21-18:32:08.399633  [**] [1:1000004:1] Conexión saliente SSH detectada [**] [Priority: 0] (TCP) 192.168.1.45:58726 -> 192.168.1.49:22
```