SeguTech

Laboratorio MF0488_3: Configuración de Snort como H(IDS)

10/07/2025

Contenidos

0: Pasos previos

Snort 2.9 se puede instalar de forma nativa en Windows, aunque la configuración es bastante tediosa.

En este laboratorio, instalaremos y configuraremos Snort 2.9 en Ubuntu. El tutorial se puede seguir con Windows a través de WSL o de forma nativa.

0.1: Instalando WSL 2 en Windows 11

[Instalación de Linux en Windows con WSL - Microsoft]

Los pasos de instalación detallados, los agregamos en el Anexo 1: Instalación de WSL

Para comprobar nuestra versión de WSL, podemos ejecutar el siguiente comando:

```
wsl --version # wls -v
```

Deberíamos ver lo siguiente

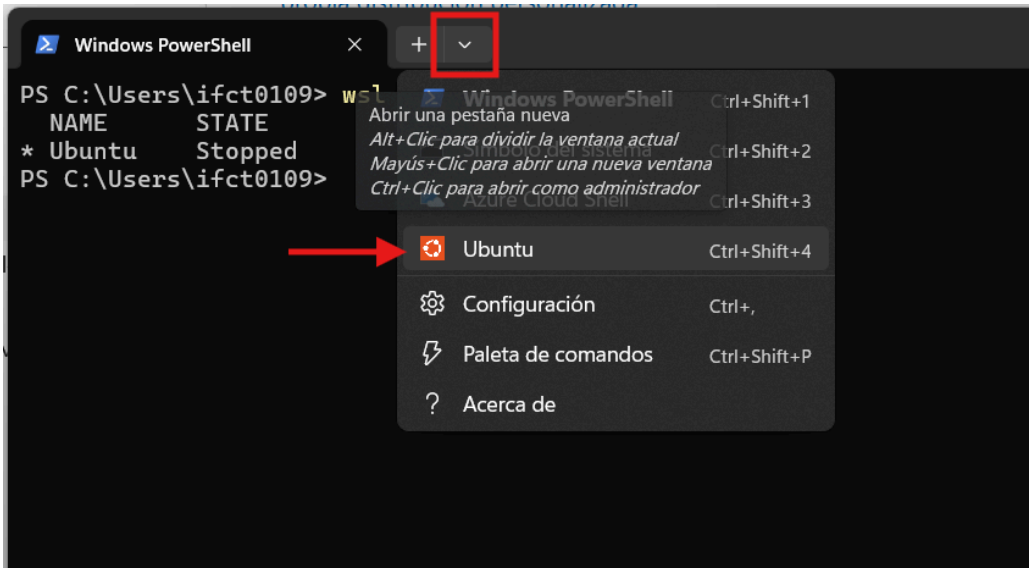
Versión de WSL: 2.5.7.0

0.2: Instalando Ubuntu en WSL 2

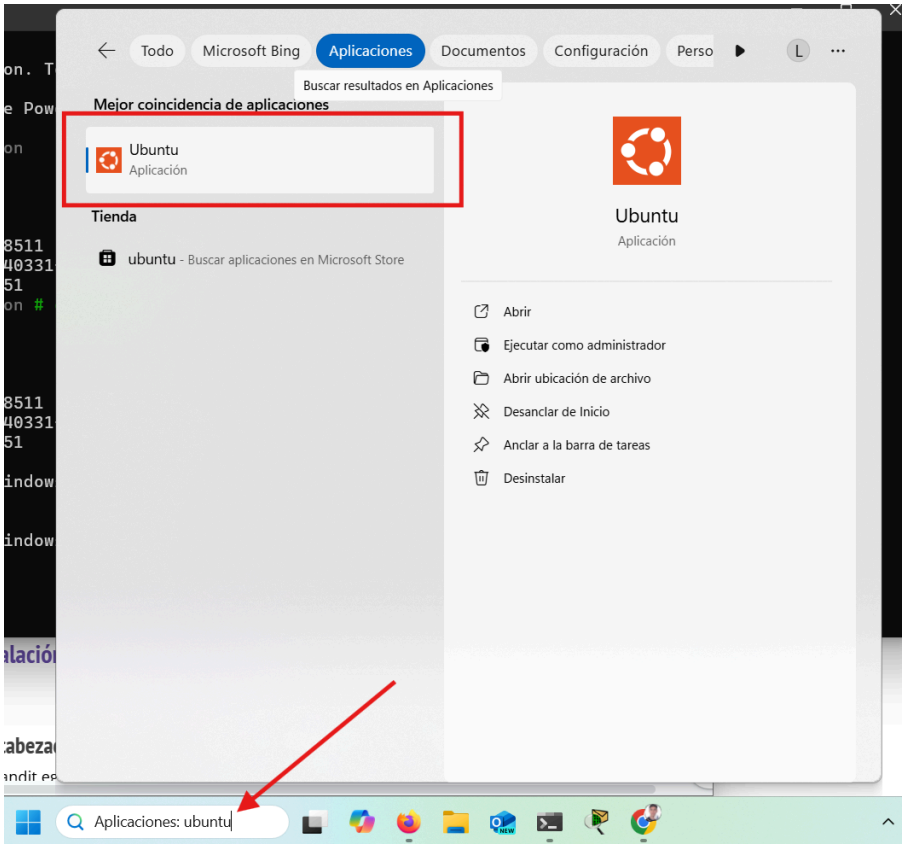
Con el siguiente comando, podemos comprobar las distribuciones de Linux instaladas

```
wsl --list # wsl -l
```

Si queremos abrir una nuevo terminal en Ubuntu, lo podemos hacer desde el menú desplegable de la aplicación de *Terminal* de Windows:

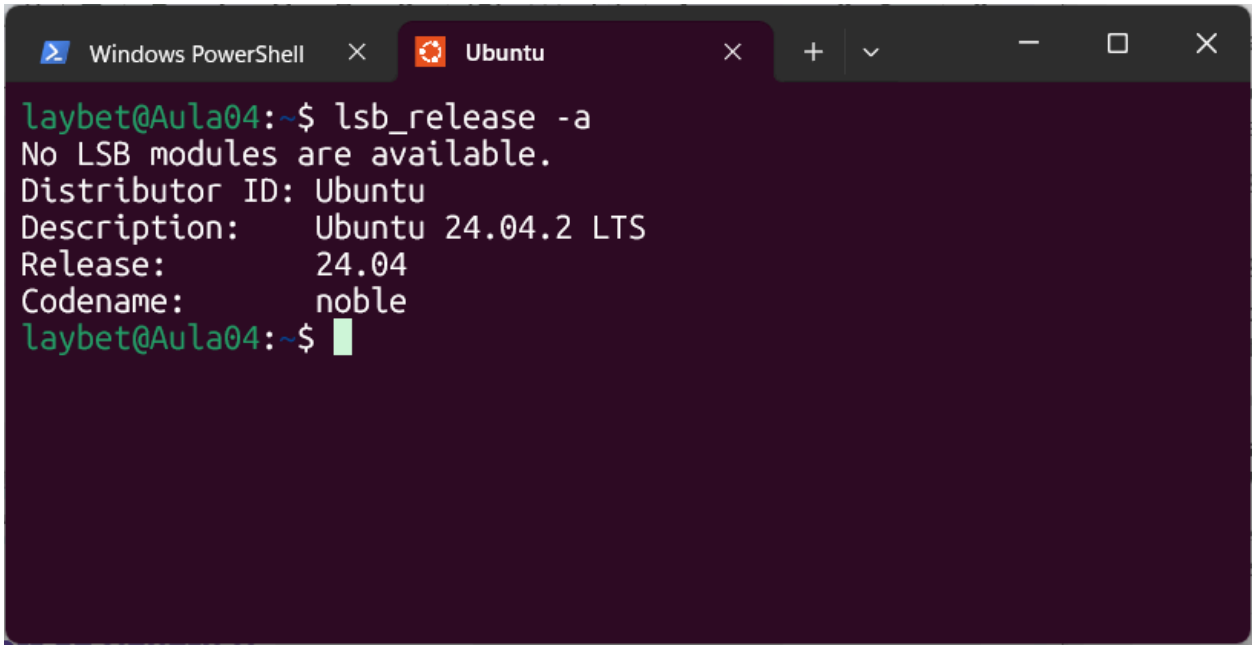


También, podemos lanzar un nuevo terminal de Ubuntu en WSL desde el menú de inicio de Windows. Simplemente debemos introducir “Ubuntu” en la barra de búsqueda, tal como se muestra en la siguiente captura:



Una vez tenemos abierto un terminal dentro de Ubuntu, podemos ejecutar el siguiente comando para comprobar nuestra distribución de Linux.

```
lsb_release -a
```

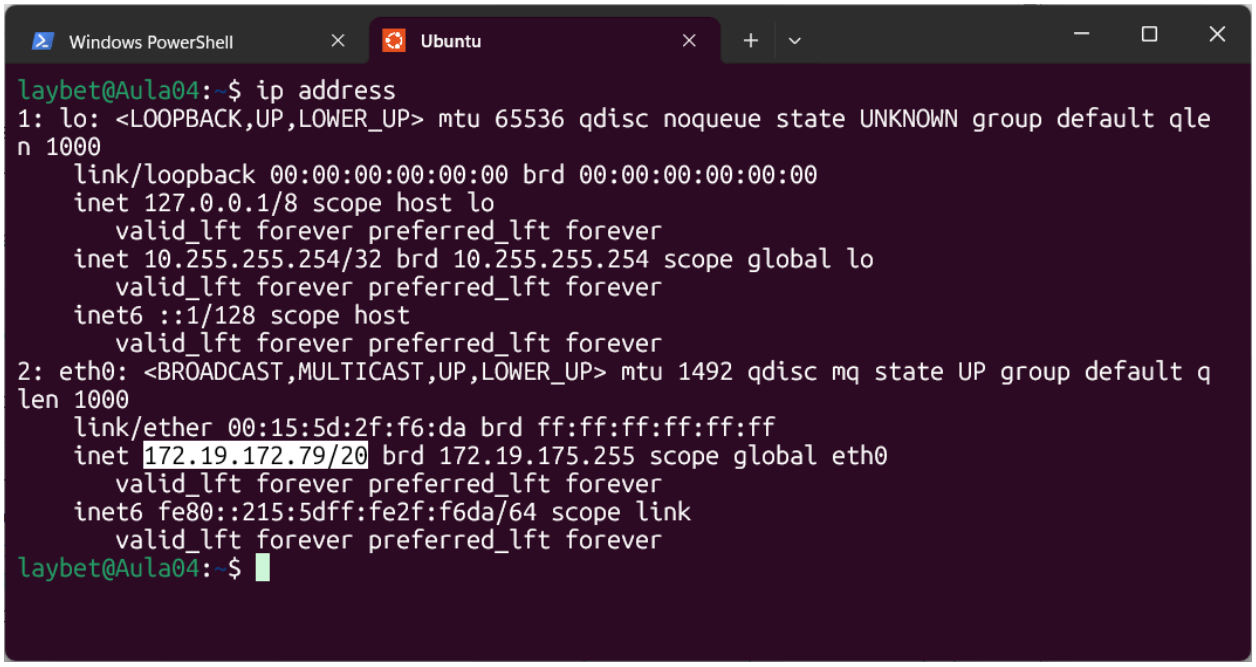


De la captura podemos comprobar que estamos ejecutando Ubuntu 24.04.2 LTS.

0.3: Comprobación de configuración de red en Ubuntu

En el terminal de Ubuntu, vamos a ejecutar el siguiente comando para comprobar nuestra IP local:

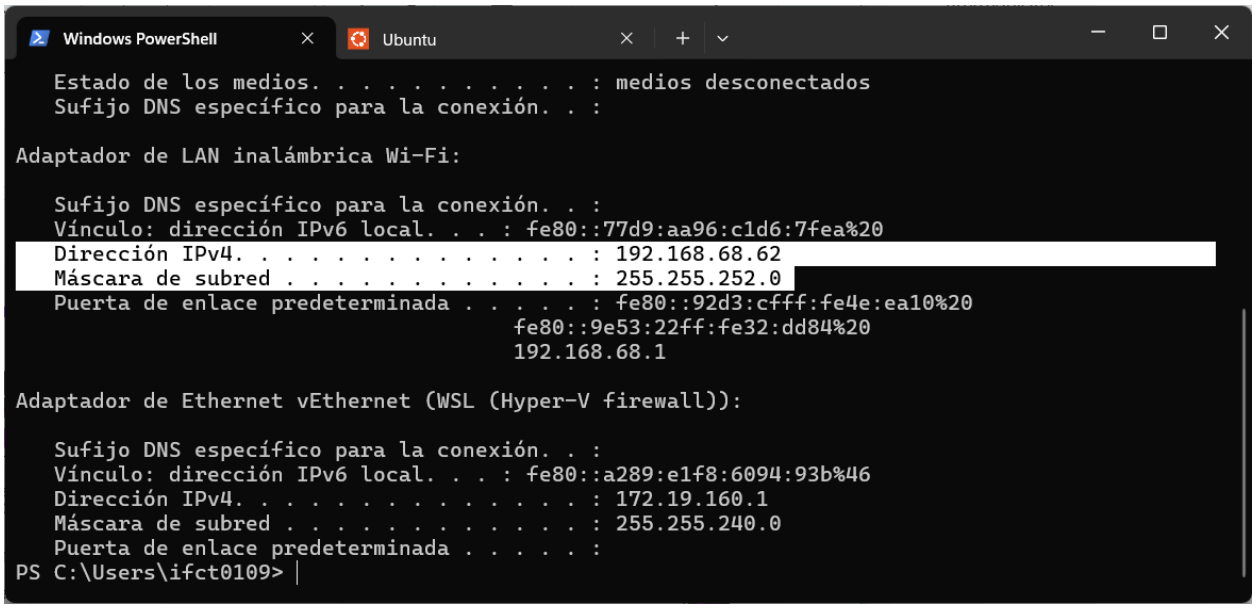
```
ip address
```



Podemos comprobar que tenemos una IP parecida a la siguiente: 172.19.172.79/20

Si ejecutamos comprobamos la IP de los adaptadores de red de Windows, veremos lo siguiente

ipconfig



La IP del adaptador WiFi es: `192.168.68.62/22`. Ésta no es la misma IP asignada a Ubuntu.

En el siguiente apartado veremos la razón.

La IP no es la misma ¿Qué está ocurriendo?

Por defecto, WSL 2 utiliza una arquitectura de red basada en NAT (Network Address Translation). Esto quiere decir que la dirección IP asignada a la interfaz de red de Ubuntu es la de un **adaptador Ethernet virtualizado**, no la IP real de Windows.

Puedes leer más al respecto en la documentación oficial: https://learn.microsoft.com/es-es/windows/wsl/networking#default-networking-mode-nat

La configuración por defecto podría ser apropiada para otras aplicaciones, pero en nuestro caso, nos interesa monitorizar el tráfico de red con un IDS. Es por ello que vamos a cambiar la configuración de red de WSL al modo espejo.

Nota: El uso de una distribución Linux dentro de WSL requiere consideraciones adicionales. Debemos tener en cuenta que, antes de llegar a la distribución de Linux, el tráfico de red pasará a través de:

1. El Firewall de Windows Defender
2. El Firewall de Hyper-V

0.4: Configuración de red WSL en modo espejo

En las máquinas que ejecutan Windows 11 22H2 y versiones posteriores, es posible establecer networkingMode=mirrored en [wsl2] el archivo .wslconfig para habilitar las redes en modo espejo.

Al habilitar esta opción, WSL cambia a una arquitectura de red cuya finalidad es "reflejar" las interfaces de red que están disponibles para Windows dentro de Linux.

En la powershell podemos ejecutar el siguiente comando:

```
notepad $env:USERPROFILE\.wslconfig
```

Debemos poner el siguiente contenido en el fichero de configuración que se abre:

```
[wsl2]
networkingMode=mirrored
```

Debemos asegurarnos de guardar. El atajo para Windows (en español) es Ctrl+G.

Guardamos, salimos de la terminal de ubuntu que tenemos abierta

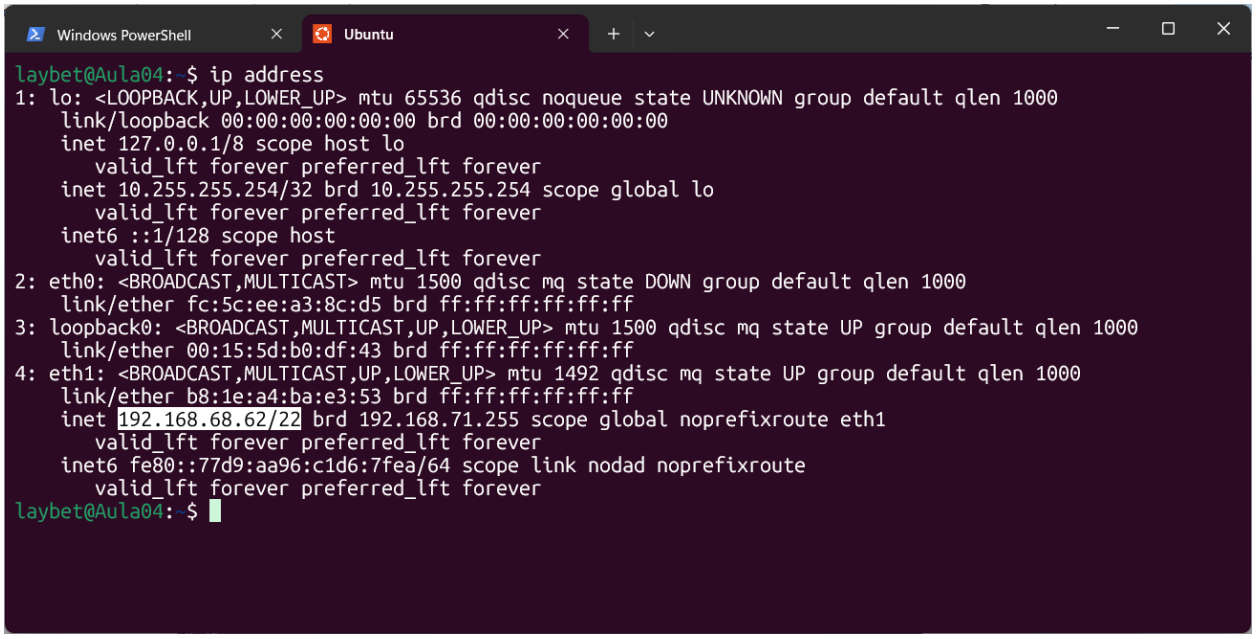
Ejecutamos el siguiente comando:

```
wsl --shutdown
```

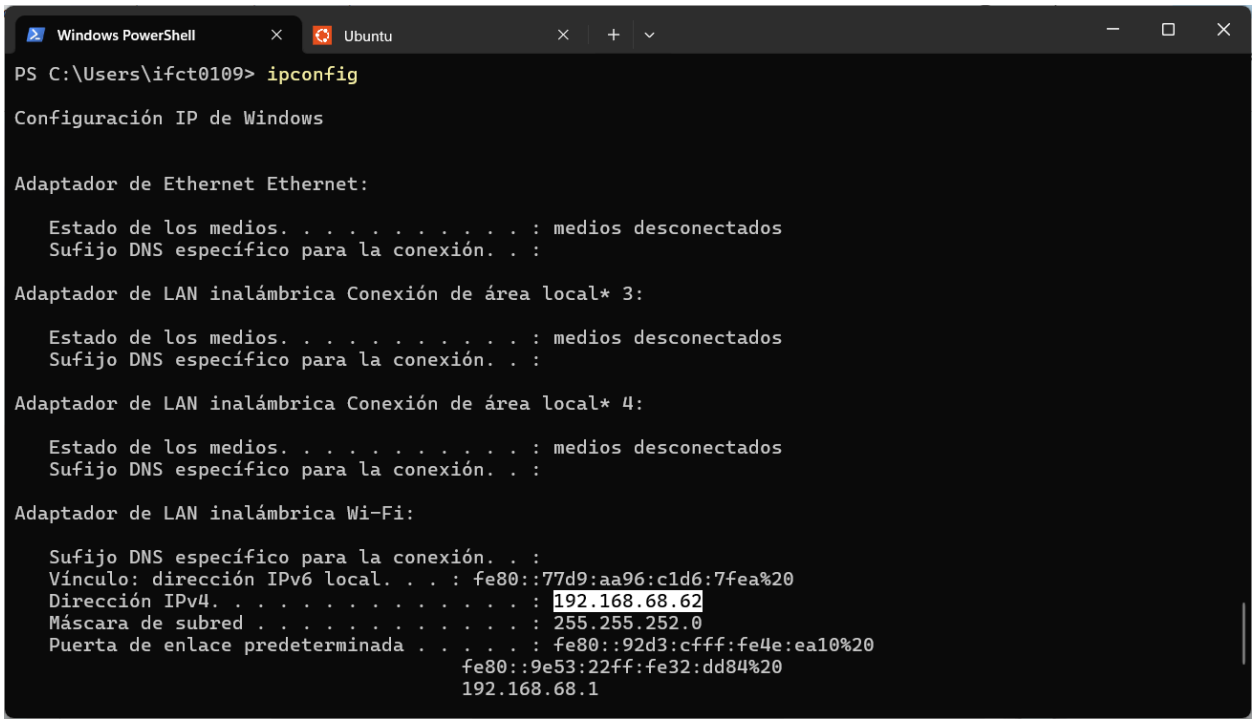
Al lanzar otra vez Ubuntu y ejecutar el siguiente comando:

```
ip address
```

Ubuntu tendrá acceso a la interfaces de red del equipo de forma transparente. Lo podemos comprobar al ver la dirección IP de la interfaz 4: eth1



La IP asignada en este caso es 192.168.68.62/22. Podemos comprobar que coincide con la asignada a Windows.



1: Instalación de Snort 2.9 en Ubuntu

Vamos a proceder a instalar Snort.

Lo primero es actualizar el listado de paquetes disponibles

```
sudo apt update
```

Si hay actualizaciones, descargarlas e instalarlas

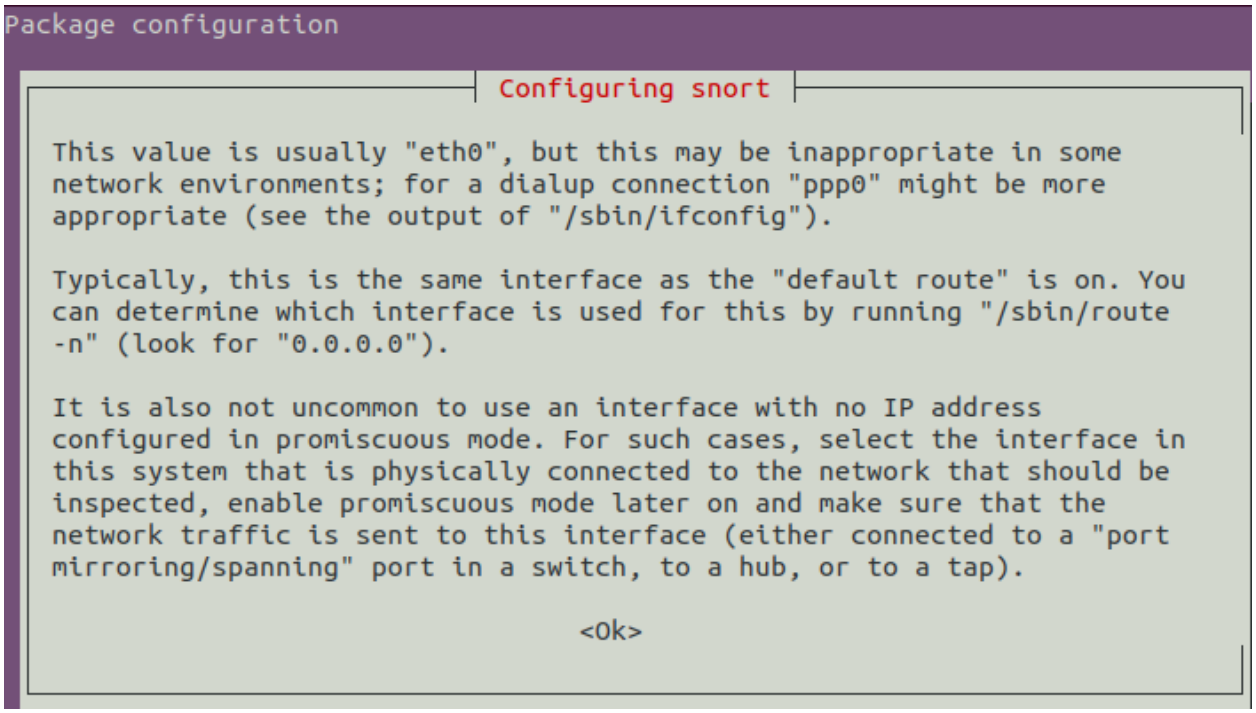
```
sudo apt upgrade
```

Ahora sí, instalamos snort con el siguiente comando:

```
sudo apt install snort
```

Nos solicitará confirmación. Introducimos: `Y` (*Yes:* Sí), para confirmar.

Después de que `apt` termine de descargar snort y sus dependencias, se nos presentará el asistente de configuración de Snort.

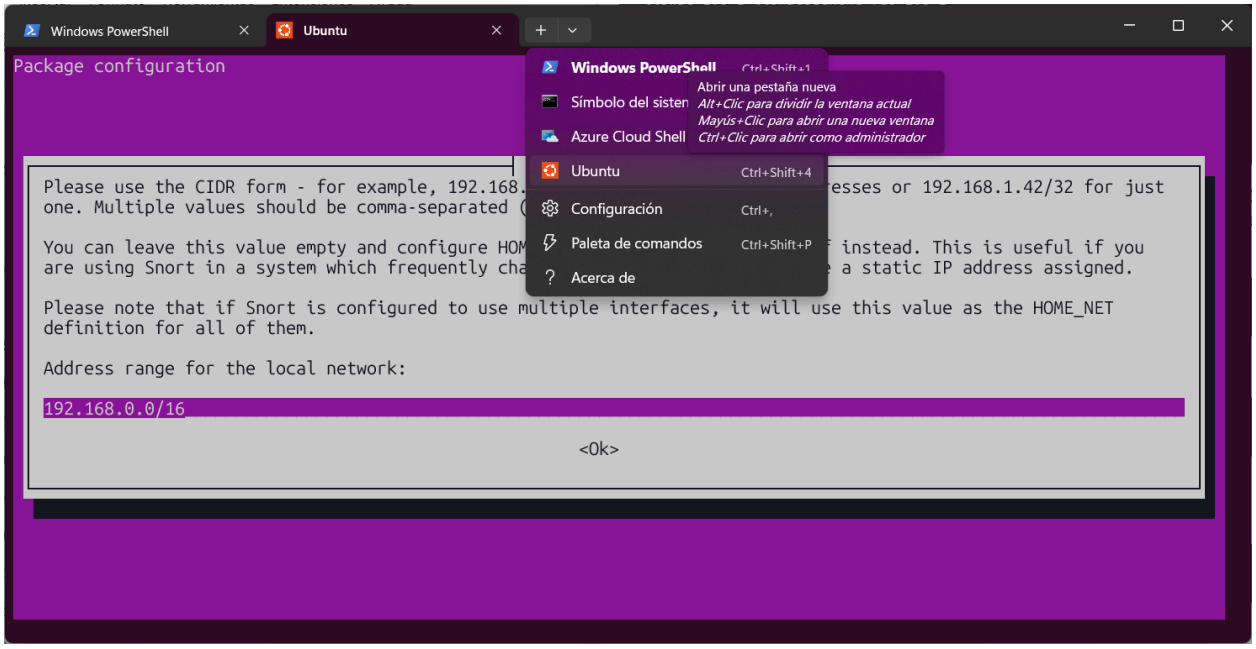


En el siguiente paso veremos cómo proceder con la configuración.

1.1 Asistente de configuración de Snort

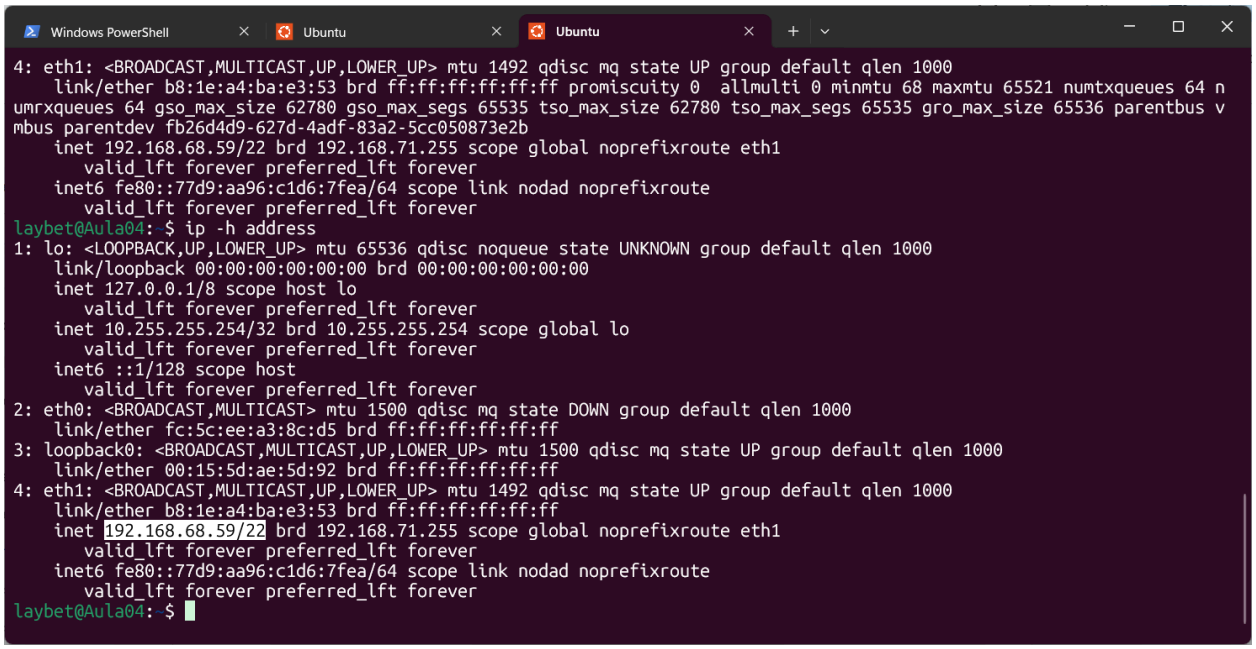
Llegados a este punto, se recomienda abrir otra pestaña con otro terminal de Ubuntu.

En esta otra pestaña podremos consultar la configuración de red de red. Así, podremos consultar la información que el asistente de configuración de Snort nos solicita.



Situados en esta otra pestaña, ejecutamos el siguiente comando para conocer nuestra configuración de red:

```
ip address
```



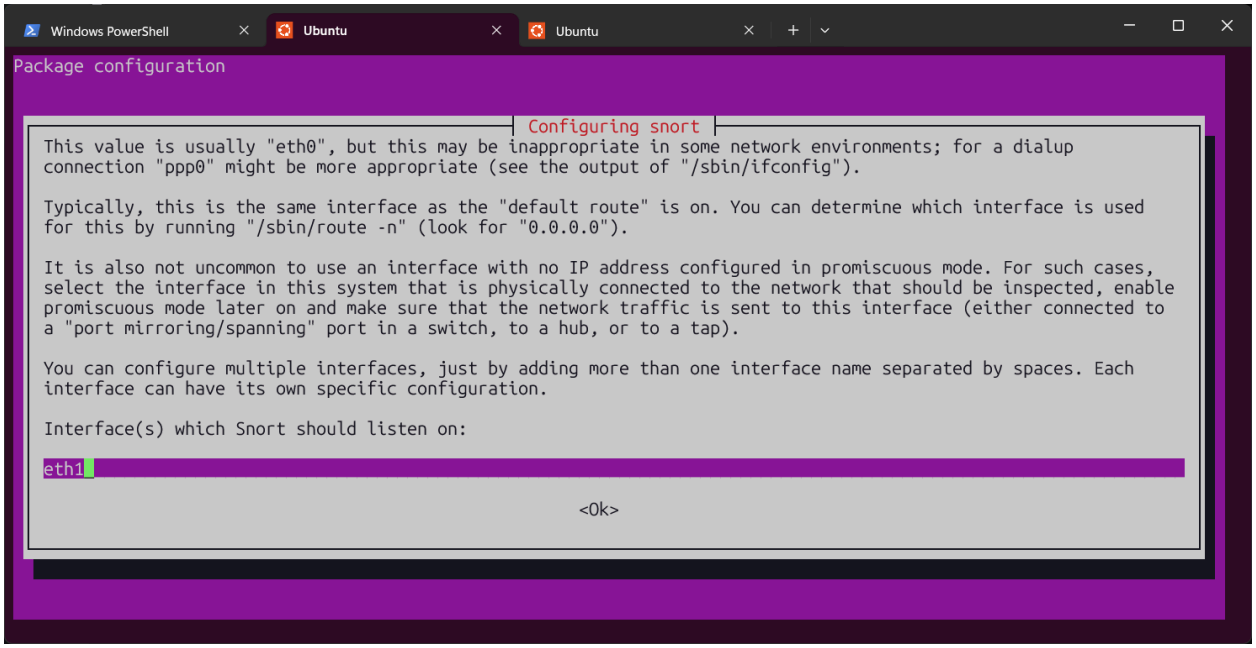
De aquí, lo importante será:

- El **nombre de la interfaz** a través de la que tenemos conexión de red, en este caso: `eth1`.
- La dirección IP local que tenemos asignada: `192.168.68.59/22`.

1.1 Interfaz de red (condicional)

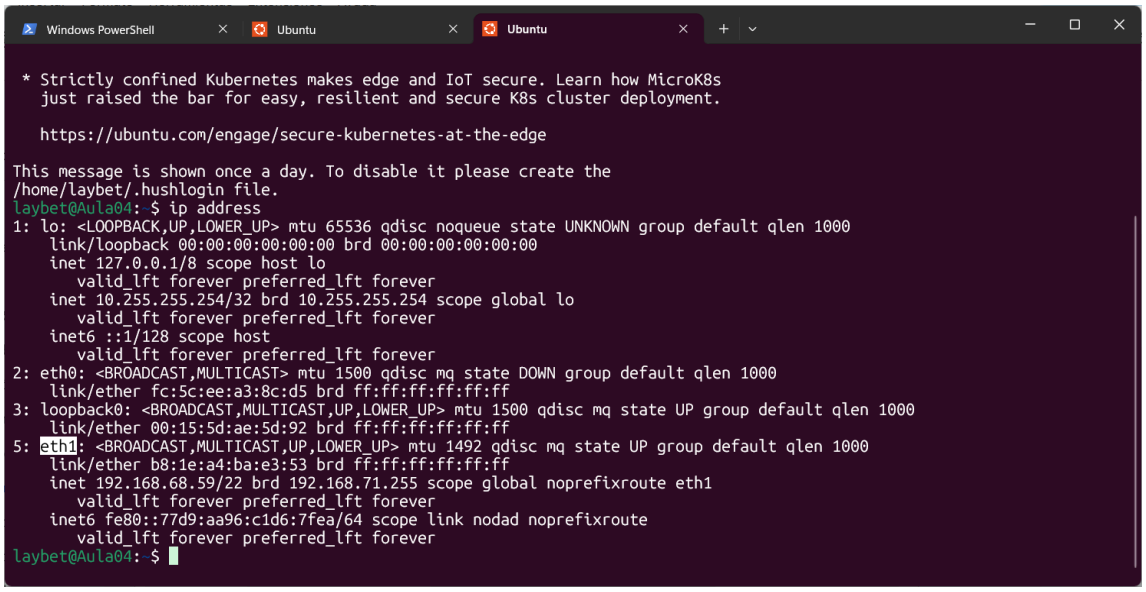
Si Snort detecta que tenemos varias interfaces de red, el asistente de configuración nos preguntará **en qué interfaces queremos que se capture tráfico de red**.

Este diálogo no aparecerá si solo tenemos una interfaz de red disponible. Sin embargo, se incluye en caso de que el diálogo se presente:



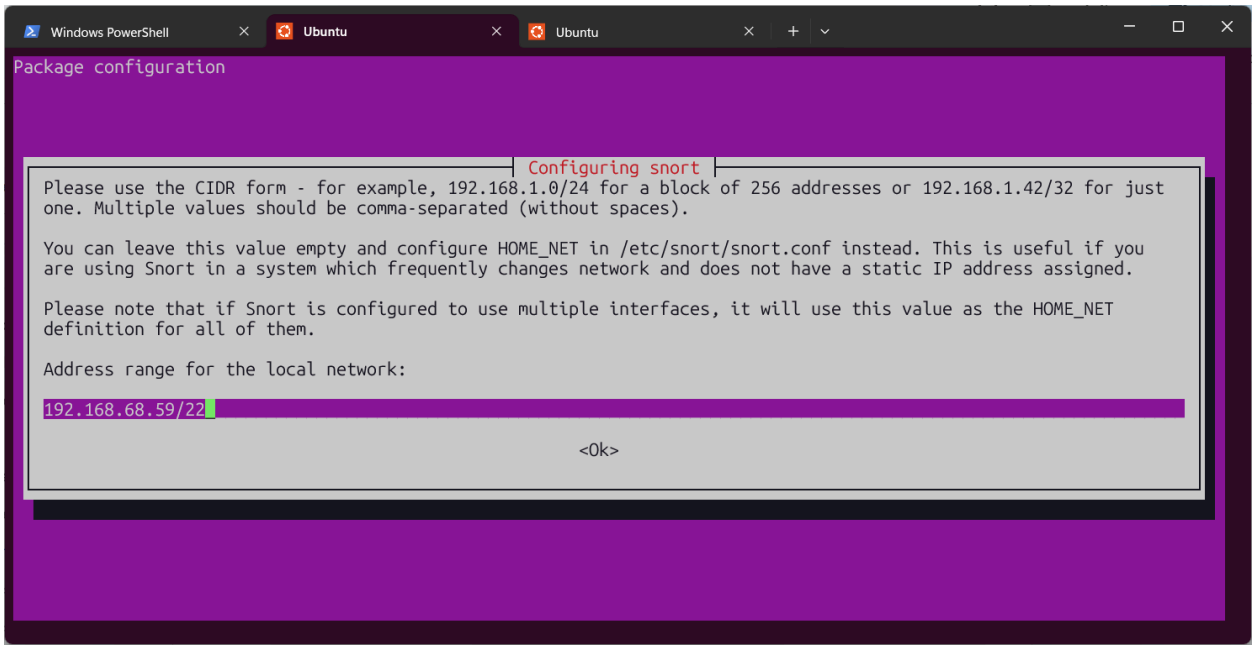
El valor por defecto, será `eth0`.

Tal como se ha explicado antes, con el comando `ip address` podemos determinar las interfaces de red disponibles. En este caso, nos interesa monitorizar el tráfico de la interfaz `eth1`.



1.2 Parámetro HOME_NET

En el diálogo que se nos presenta, se nos solicita **establecer el bloque de direcciones IP para el que Snort va a monitorizar tráfico**. Ésto se hace a través del parámetro de configuración de Snort: `HOME_NET`



El contenido del mensaje es el siguiente:

> Please use the CIDR form - for example, 192.168.1.0/24 for a block of 256 addresses or 192.168.1.42/32 for just one. Multiple values should be comma-separated (without spaces).

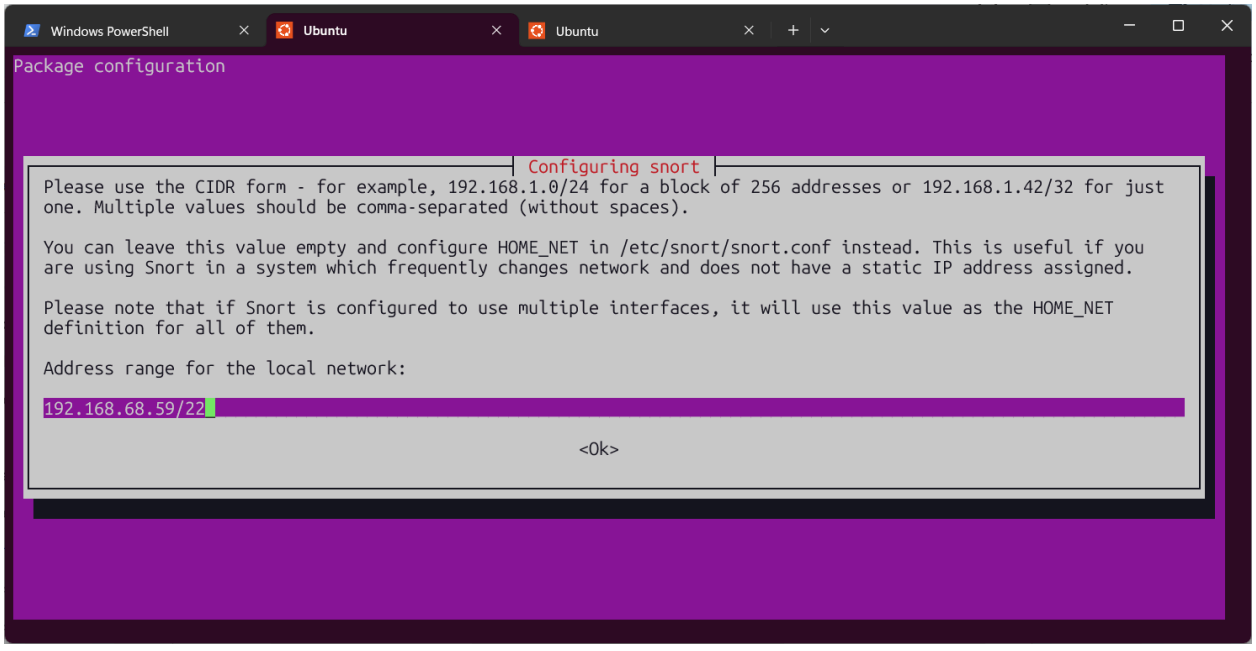
> You can leave this value empty and configure HOME_NET in /etc/snort/snort.conf instead. This is useful if you are using Snort in a system which frequently changes network and does not have a static IP address assigned.
>
> Please note that if Snort is configured to use multiple interfaces, it will use this value as the HOME_NET definition for all of them.

A continuación, una traducción automática de dicho mensaje:

> Utilice la forma CIDR: por ejemplo, 192.168.1.0/24 para un bloque de 256 direcciones o 192.168.1.42/32 para una sola. Los valores múltiples deben estar separados por comas (sin espacios).
>
> Puede dejar este valor vacío y configurar HOME_NET en /etc/snort/snort.conf en su lugar. Esto es útil si está utilizando Snort en un sistema que cambia frecuentemente de red y no tiene asignada una dirección IP estática.
>
> Tenga en cuenta que si Snort está configurado para usar múltiples interfaces, usará este valor como definición de HOME_NET para todas ellas.

En el diálogo presentado, se nos solicita el bloque CIDR (rango de direcciones IPs) que nos interesa que Snort analice.

Pretendemos monitorizar el tráfico de red transmitido a través de la única interfaz de red disponible para Ubuntu, así que podemos introducir tal cual el bloque CIDR que hemos recuperado al ejecutar `ip address`: **`192.168.68.59/22`**.



Nota: La dirección IP obtenida depende del entorno en el que los encontremos y puede cambiar en el futuro. Tras una desconexión y reconexión, el router de la red nos podría asignar otra IP (a menos que tengamos configurada una IP estática).

Consideración adicional: Configuración de un HIDS

En este laboratorio, por las limitaciones que existen en el entorno disponible, **solo vamos a poder monitorizar el tráfico generado por nuestra propia instalación Ubuntu**. Las limitaciones que tenemos son las siguientes:

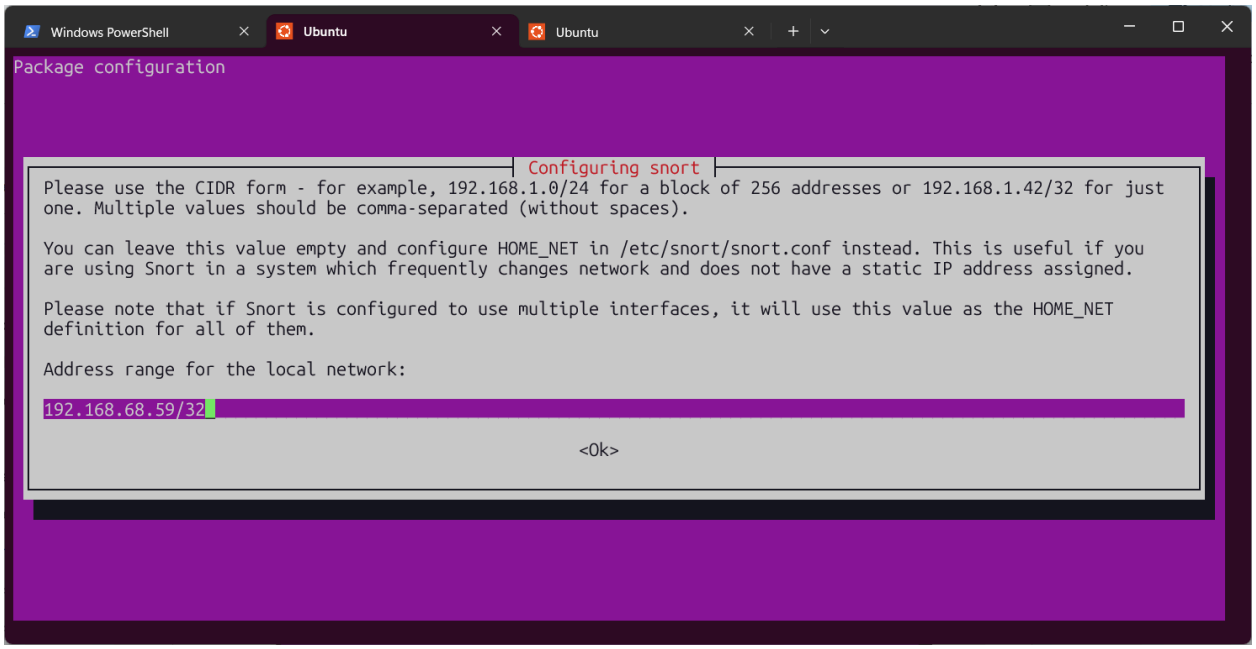
- Ubuntu se ejecuta dentro de WSL (Windows Subsystem for Linux)
- Los adaptadores WiFi de la mayoría de ordenadores no permiten capturar tráfico inalámbrico que no esté dirigido a ellas mismas.
  Es decir: Que la dirección MAC destino, sea su propia dirección MAC o una dirección de *Broadcast*.
- El tráfico de red de una red WiFi normalmente está encriptado. Aunque capturemos el tráfico de red, no seríamos capaces de analizar IPs de origen o destino, o protocolos de capas superiores (por ejemplo: TCP, UDP de la capa de transporte).

Es por esto que el tipo de IDS que vamos a desplegar sería un HIDS (*HostIDS*): Un IDS en el Host.

Esto contrasta con un NIDS (*NetworkIDS*): Un IDS basado en red, monitorizando todos los segmentos de la red.

Como estamos configurando un HIDS (IDS basado en host), el tráfico de red que vamos monitorizar es el de nuestro propio equipo. En la configuración de Snort, esto significa que **el rango de redes que vamos a analizar es una única IP: La nuestra**.

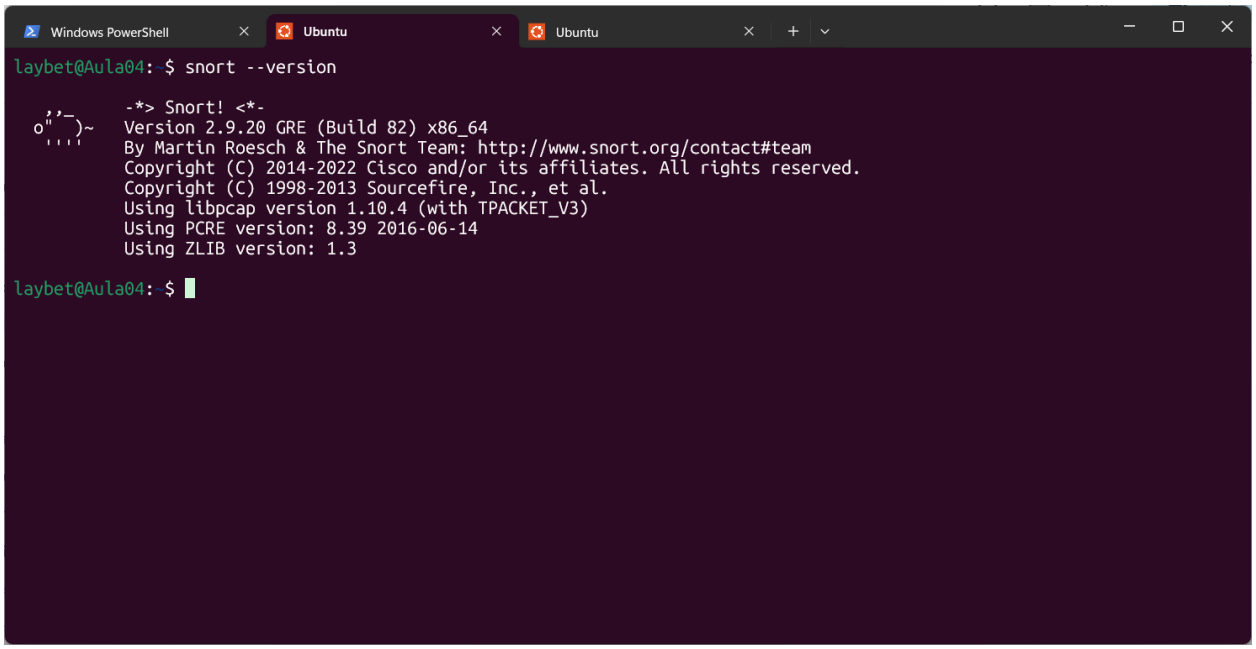
La forma de representar una sola IP en formato CIDR es poniendo `/32` como máscara de red. Por tanto, pondríamos el siguiente bloque CIDR en este parámetro de configuración: `192.168.68.59/32`.



1.3 Comprobación de la instalación

Una vez ha finalizado el asistente de instalación, para comprobar la instalación de snort, ejecutamos el siguiente comando:

```
snort --version
```



También podemos usar:

```
snort -V
```

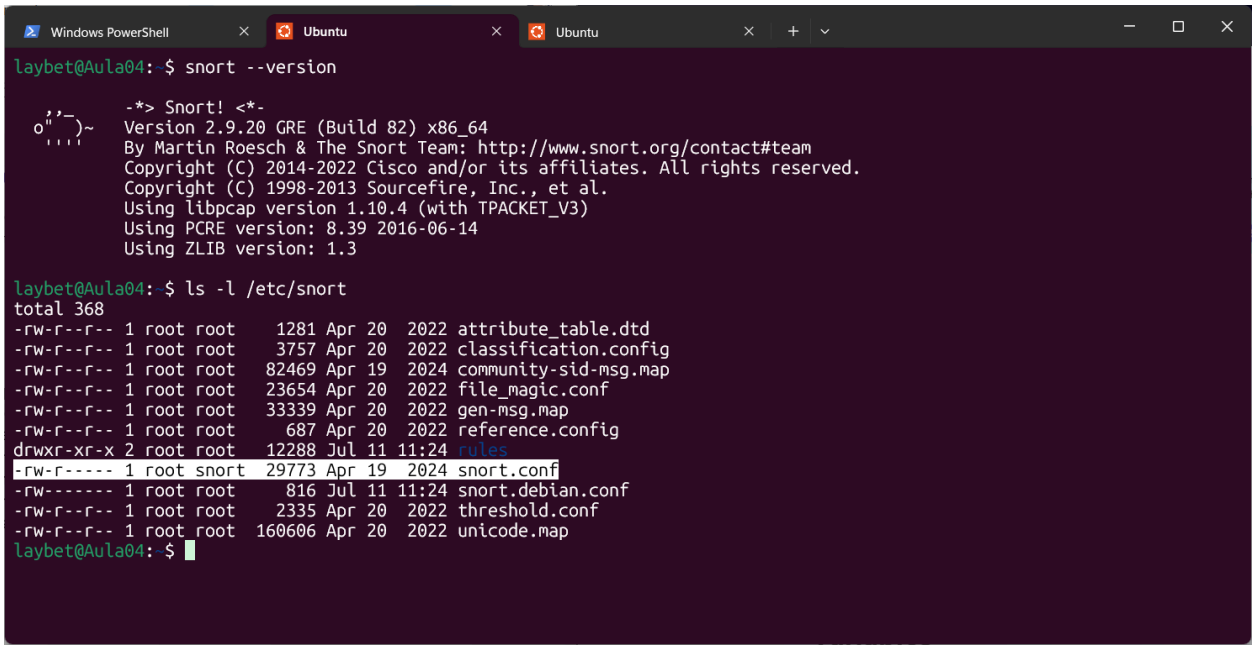
2: Configuración de Snort

La configuración de Snort se establece en el fichero de texto `snort.conf`. Éste fichero se encuentra en la ruta: `/etc/snort`.

Con el comando `ls` (list directory) podemos consultar el contenido de dicho directorio:

```
ls -l /etc/snort
```

Veremos que en el listado se incluye el fichero **`snort.conf`**.



2.1 Eligiendo un editor de texto

Editar ficheros de configuración es algo muy habitual al desplegar software en Linux. La forma más sencilla de hacerlo es con editores de texto en la terminal como `nano` o `vim`.

Por ejemplo, para editar el fichero snort.conf con nano, debemos usar siguiente comando:

```
sudo nano /etc/snort/snort.conf
```

Aún así, si nos intimida utilizar un editor de texto desde la terminal, WSL nos permite ejecutar aplicaciones con interfaz gráfica desde Linux. Un editor de texto con interfaz gráfica que podemos usar es gedit.

Procedemos a instalarlo con los siguientes comandos:

```
sudo apt update && sudo apt upgrade
sudo apt install gedit
```

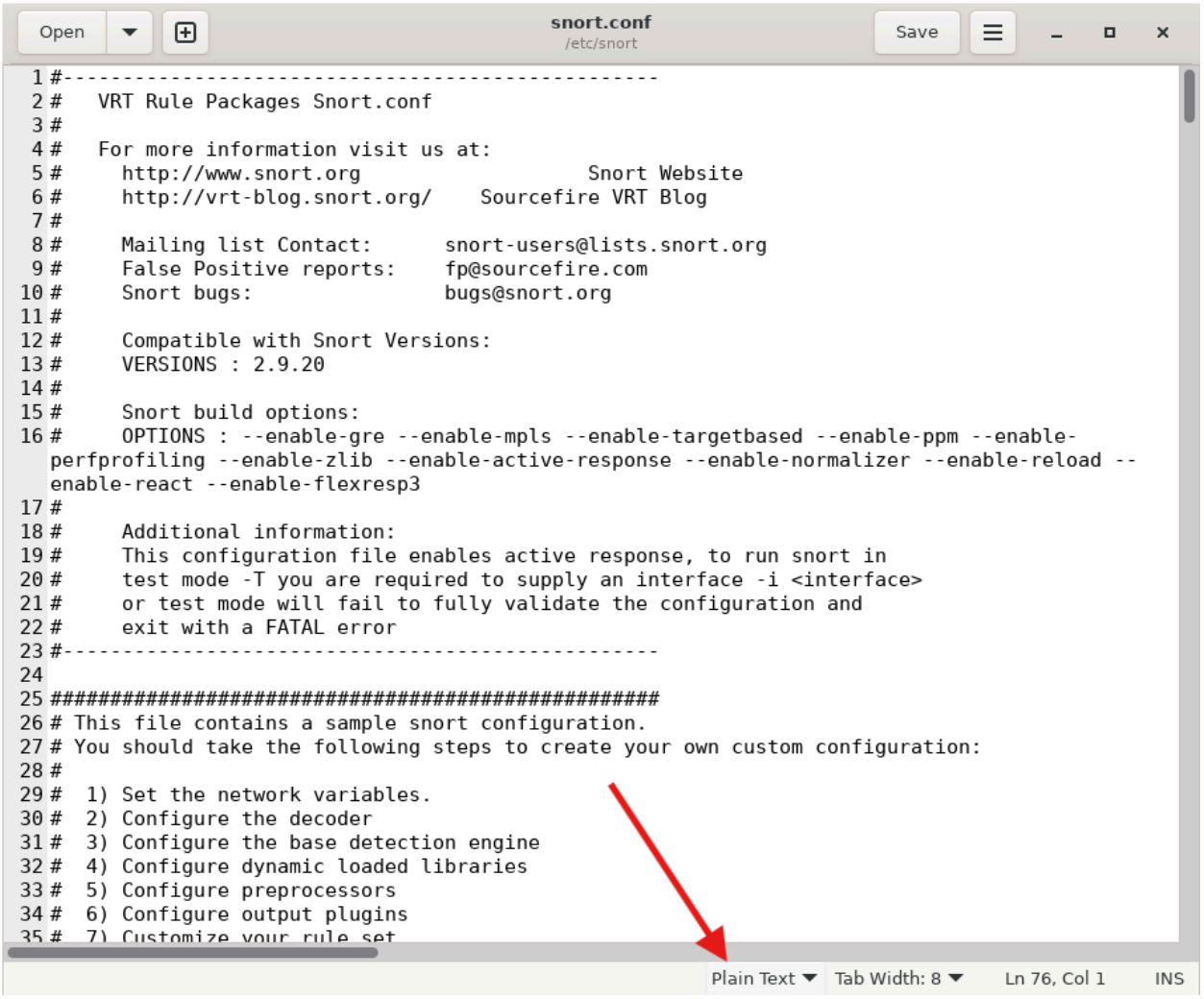
Una vez tengamos gedit instalado, para editar `snort.conf` con este editor ejecutamos el siguiente comando:

```
sudo gedit /etc/snort/snort.conf
```

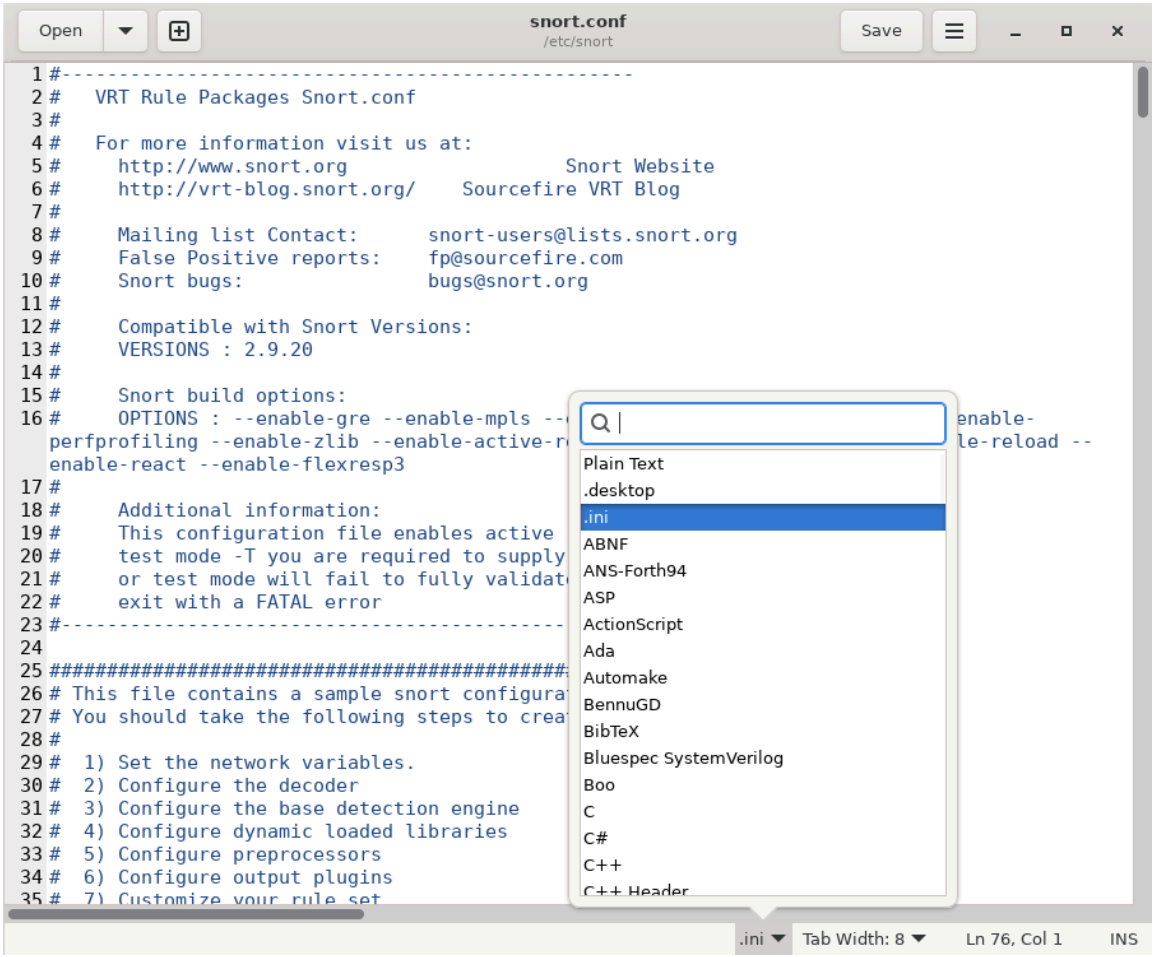
Se debería abrir en una ventana el editor.

El fichero de texto se presenta como texto plano. Para facilitarnos un poco las cosas, vamos a visualizar el texto con color, de forma que podamos interpretar más fácilmente el fichero de configuración.

Para esto, en la barra inferior, hacemos clic en el desplegable que pone *Plain Text*.

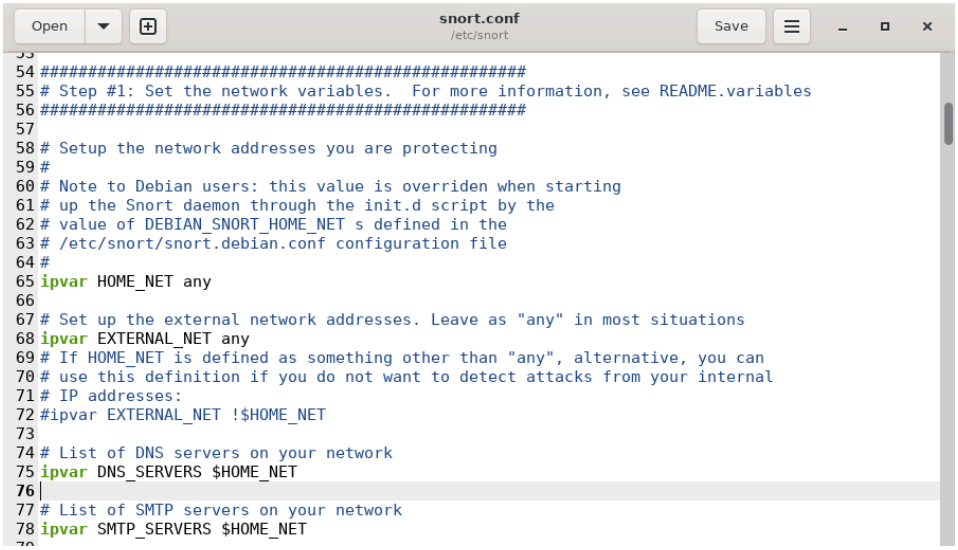


De este desplegable, podemos seleccionar `.ini` para visualizar el texto como si se tratase de un fichero `.ini`.



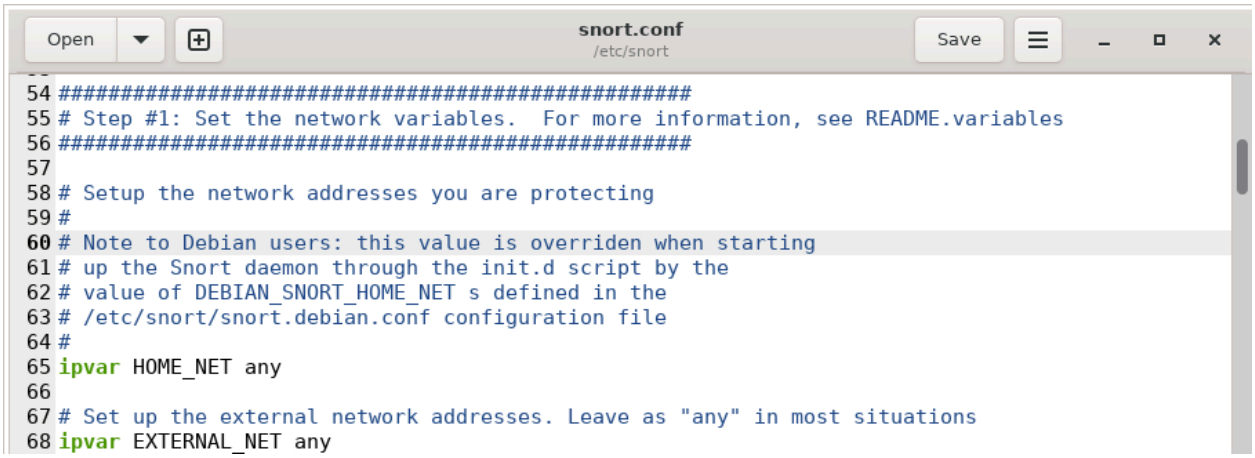
Tras hacer esto, deberíamos diferenciar bien:

- Los comentarios: Líneas que comienzan con “#“ y cuyo contenido es ignorado.
- Variables de configuración con sus valores correspondientes.



2.2 Comprobando el fichero `snort.debian.conf`

En la primera sección del fichero `snort.conf`, podemos leer la siguiente nota:



En ese comentario se incluye una aclaración para los usuarios de Debian (y Ubuntu es una distribución de Linux basada en Debian): Las variables de red de esta sección quedan sobreescritas por el fichero `/etc/snort/snort.debian.conf`.

En otra terminal de Ubuntu, podemos consultar el contenido de este fichero:

```
gedit /etc/snort/snort.debian.conf
```

Omitimos `sudo` porque solo vamos a leer el fichero, no vamos a modificarlo. Por tanto, no necesitamos permisos de superusuario.

El contenido de `snort.debian.conf` es el siguiente:

En este fichero deberíamos ver lo mismo que hayamos configurado a través del asistente de configuración.

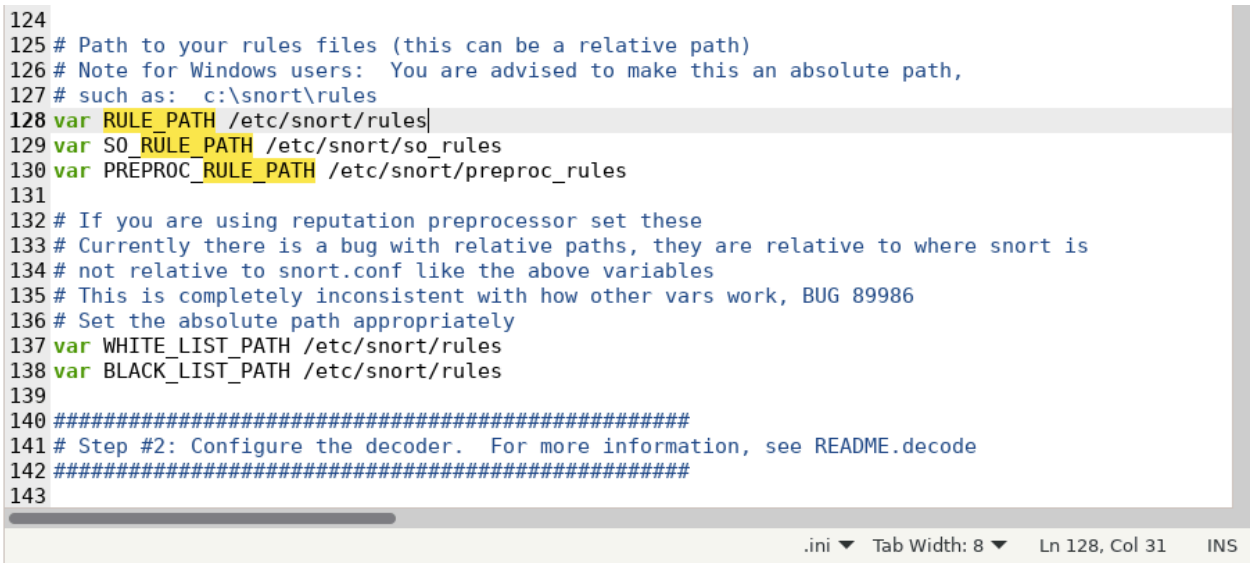
2.3 Editando el fichero `snort.conf`: Desactivando las reglas por defecto

Deberíamos tener abierto el fichero `snort.conf` al haber ejecutado el comando:

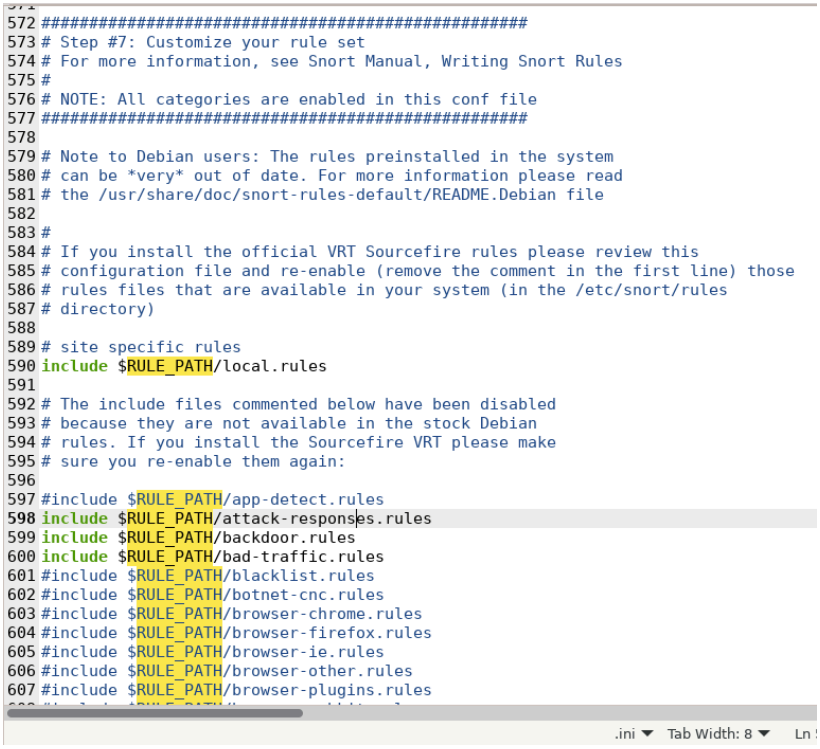
```
sudo gedit /etc/snort/snort.conf
```

Si nos desplazamos un poco, en la **línea 128** encontraremos la **variable `RULE_PATH`**.

`RULE_PATH` define la ruta en la se encuentran las reglas de detección de Snort. La ruta en este caso es: `/etc/snort/rules`.



Mucho más abajo, en la sección *`Step #7: Customize your rule set`*, se establece qué reglas aplicará Snort.



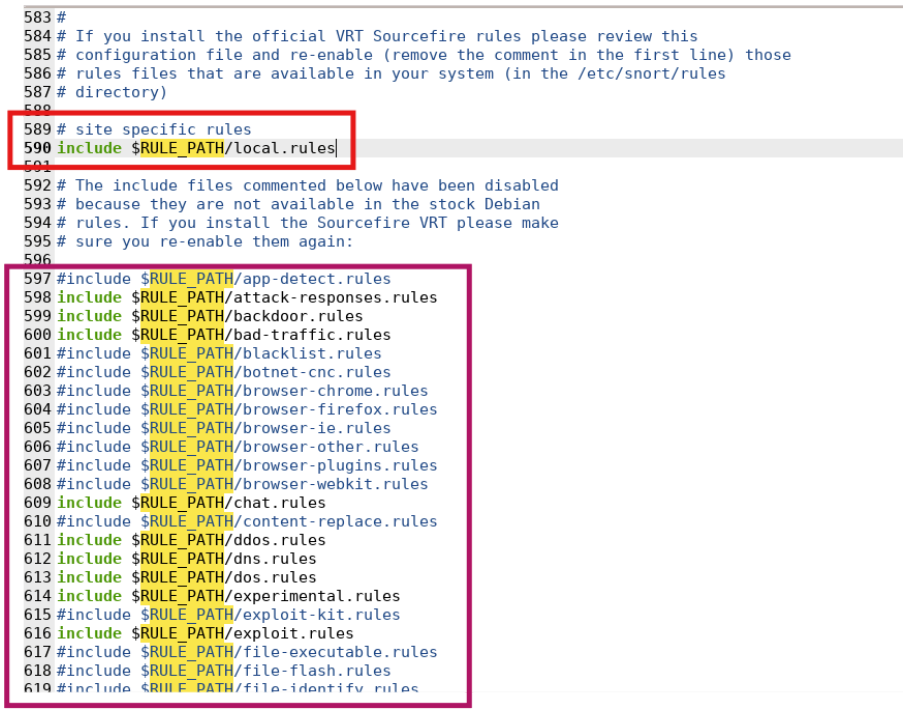
```
572 ##################################################
573 # Step #7: Customize your rule set
574 # For more information, see Snort Manual, Writing Snort Rules
575 #
576 # NOTE: All categories are enabled in this conf file
577 ##################################################
578
579 # Note to Debian users: The rules preinstalled in the system
580 # can be *very* out of date. For more information please read
581 # the /usr/share/doc/snort-rules-default/README.Debian file
582
583 #
584 # If you install the official VRT Sourcefire rules please review this
585 # configuration file and re-enable (remove the comment in the first line) those
586 # rules files that are available in your system (in the /etc/snort/rules
587 # directory)
588
589 # site specific rules
590 include $RULE_PATH/local.rules
591
592 # The include files commented below have been disabled
593 # because they are not available in the stock Debian
594 # rules. If you install the Sourcefire VRT please make
595 # sure you re-enable them again:
596
597 #include $RULE_PATH/app-detect.rules
598 include $RULE_PATH/attack-responses.rules
599 include $RULE_PATH/backdoor.rules
600 include $RULE_PATH/bad-traffic.rules
601 #include $RULE_PATH/blacklist.rules
602 #include $RULE_PATH/botnet-cnc.rules
603 #include $RULE_PATH/browser-chrome.rules
604 #include $RULE_PATH/browser-firefox.rules
605 #include $RULE_PATH/browser-ie.rules
606 #include $RULE_PATH/browser-other.rules
607 #include $RULE_PATH/browser-plugins.rules
                                    .ini  Tab Width: 8  Ln
```

En la **línea 590** tenemos se establece el fichero de reglas personalizadas:

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

Si queremos definir nuestras propias reglas, es en este fichero donde debemos incluirlas: `/etc/snort/rules/`**`local.rules`**

Más abajo, a partir de la línea 598, tenemos un listado de reglas predeterminadas que incluye Snort:



```
583 #
584 # If you install the official VRT Sourcefire rules please review this
585 # configuration file and re-enable (remove the comment in the first line) those
586 # rules files that are available in your system (in the /etc/snort/rules
587 # directory)
588
589 # site specific rules
590 include $RULE_PATH/local.rules
591
592 # The include files commented below have been disabled
593 # because they are not available in the stock Debian
594 # rules. If you install the Sourcefire VRT please make
595 # sure you re-enable them again:
596
597 #include $RULE_PATH/app-detect.rules
598 include $RULE_PATH/attack-responses.rules
599 include $RULE_PATH/backdoor.rules
600 include $RULE_PATH/bad-traffic.rules
601 #include $RULE_PATH/blacklist.rules
602 #include $RULE_PATH/botnet-cnc.rules
603 #include $RULE_PATH/browser-chrome.rules
604 #include $RULE_PATH/browser-firefox.rules
605 #include $RULE_PATH/browser-ie.rules
606 #include $RULE_PATH/browser-other.rules
607 #include $RULE_PATH/browser-plugins.rules
608 #include $RULE_PATH/browser-webkit.rules
609 include $RULE_PATH/chat.rules
610 #include $RULE_PATH/content-replace.rules
611 include $RULE_PATH/ddos.rules
612 include $RULE_PATH/dns.rules
613 include $RULE_PATH/dos.rules
614 include $RULE_PATH/experimental.rules
615 #include $RULE_PATH/exploit-kit.rules
616 include $RULE_PATH/exploit.rules
617 #include $RULE_PATH/file-executable.rules
618 #include $RULE_PATH/file-flash.rules
619 #include $RULE_PATH/file-identify.rules
```

Si queremos desactivar alguna de estas reglas, la podemos comentar **añadiendo un "#" al principio de la línea**.

Para comentar una regla,

```
include $RULE_PATH/attack-responses.rules
```

añadimos "#" al principio de la línea:

```
# include $RULE_PATH/attack-responses.rules
```

Con esto, la regla quedaría desactivada.

Desactivando las reglas por defecto

A continuación, vamos a **comentar todas las reglas del listado**. Nuestro objetivo es empezar a familiarizarnos con el funcionamiento de Snort.

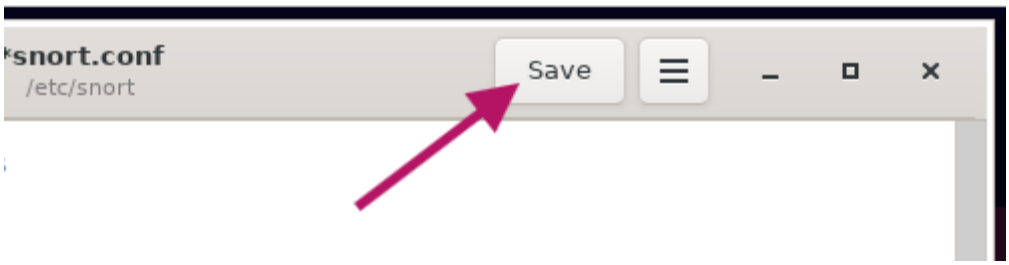
Podemos mantener el fichero de reglas personalizadas:

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

Pero el **resto de reglas por defecto, las desactivamos**. Se encuentran **entre las líneas 597 y 717**.

```
Open                                    *snort.conf
                                        /etc/snort
689 #include $RULE_PATH/specific-threats.rules
690 #include $RULE_PATH/spyware-put.rules
691 #include $RULE_PATH/sql.rules
692 #include $RULE_PATH/telnet.rules
693 #include $RULE_PATH/tftp.rules
694 #include $RULE_PATH/virus.rules
695 #include $RULE_PATH/voip.rules
696 #include $RULE_PATH/web-activex.rules
697 #include $RULE_PATH/web-attacks.rules
698 #include $RULE_PATH/web-cgi.rules
699 #include $RULE_PATH/web-client.rules
700 #include $RULE_PATH/web-coldfusion.rules
701 #include $RULE_PATH/web-frontpage.rules
702 #include $RULE_PATH/web-iis.rules
703 #include $RULE_PATH/web-misc.rules
704 #include $RULE_PATH/web-php.rules
705 #include $RULE_PATH/x11.rules
706 #include $RULE_PATH/community-sql-injection.rules
707 #include $RULE_PATH/community-web-client.rules
708 #include $RULE_PATH/community-web-dos.rules
709 #include $RULE_PATH/community-web-iis.rules
710 #include $RULE_PATH/community-web-misc.rules
711 #include $RULE_PATH/community-web-php.rules
712 #include $RULE_PATH/community-sql-injection.rules
713 #include $RULE_PATH/community-web-client.rules
714 #include $RULE_PATH/community-web-dos.rules
715 #include $RULE_PATH/community-web-iis.rules
716 #include $RULE_PATH/community-web-misc.rules
717 #include $RULE_PATH/community-web-php.rules
```

Tras haber hecho los cambios, guardamos:



2.4 Validando la configuración del fichero `snort.conf`

Con el siguiente comando, ejecutamos Snort en modo prueba:

```
sudo snort -T -c /etc/snort/snort.conf
```

Ésto nos permite comprobar que nuestro fichero de configuración está escrito correctamente.

Deberíamos ver un mensaje como el siguiente si todo ha ido bien:

```
Total snort Fixed Memory Cost - MaxRss:48000
Snort successfully validated the configuration!
Snort exiting
laybet@Aula04:~$
```

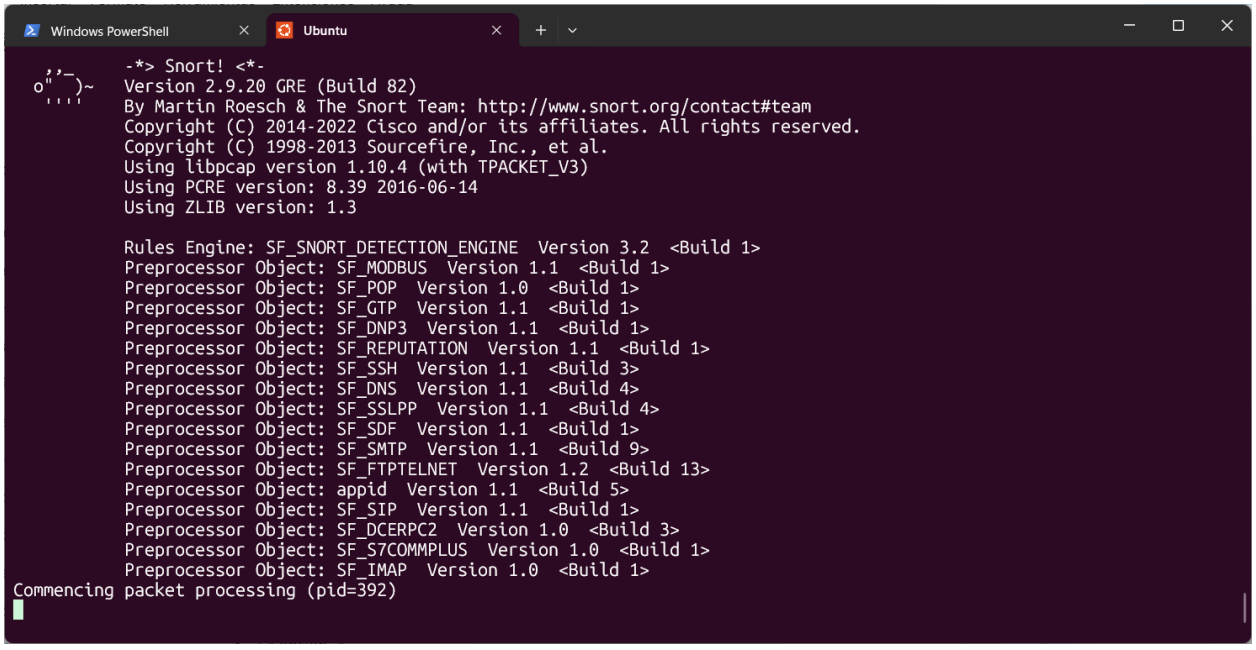
3: Ejecutando Snort

Para lanzar Snort con el fichero de reglas que acabamos de configurar, hacemos lo siguiente:

```
sudo snort -i eth1 -c /etc/snort/snort.conf
```

Las opciones que estamos usando en este caso son las siguientes:

- `-i`: Especificamos la interfaz de red que vamos a monitorizar. En este caso, `eth1`
- `-c`: Fichero reglas a utilizar, en este caso `/etc/snort/snort.conf`.



Para **detener la ejecución de Snort, presionamos las teclas `Ctrl+C` en el terminal**. Tras unos segundos, deberíamos tener acceso a nuestra consola.

Snort imprime bastante información en el terminal. Si queremos omitir este tipo de mensajes iniciales, podemos usar la opción `-q` (*quiet*, silencioso):

```
sudo snort -q -i eth1 -c /etc/snort/snort.conf
```

Aunque no veamos ningún mensaje al iniciar, Snort se está ejecutando.

Presionamos las teclas **`Ctrl+C` en el terminal para detener Snort**.

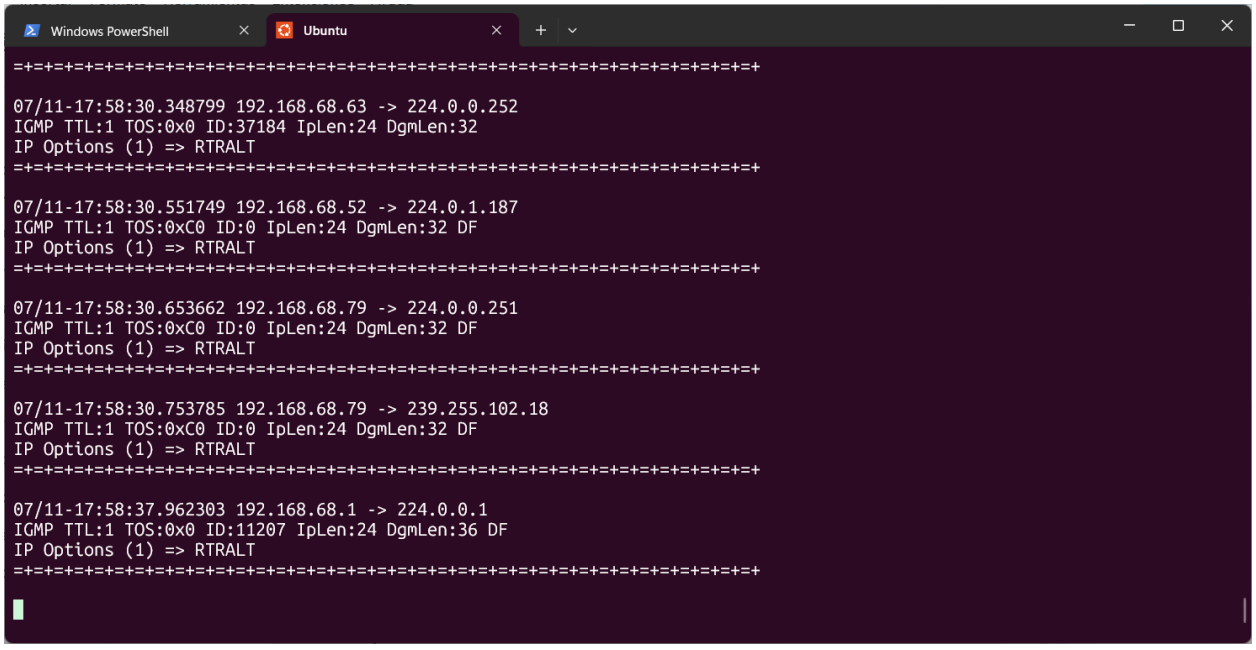
3.1 Snort en modo escucha (*sniffer*)

Snort puede ejecutarse en modo escucha, imprimiendo todos los paquetes TCP/IP que capture. Para esto, usamos la **opción `-v`**:

```
sudo snort -q -v -i eth1 -c /etc/snort/snort.conf
```

- `-q`: Modo "silencioso", omitiendo los mensajes de inicio
- `-v`: Modo *sniffer*, escucha.

Deberíamos ver cómo los distintos paquetes de tráfico de red se van imprimiendo en consola:



En otra pestaña del terminal de Ubuntu, podemos ejecutar un ping a la IP 8.8.8.8:

```
ping 8.8.8.8
```

En el terminal veremos las distintas peticiones de ping, seguidas de su respuesta.

- IP **origen** -> **destino**
- Protocolo **ICMP** (usado por *ping*)

```
07/11-18:00:15.394823 192.168.68.59 -> 8.8.8.8
ICMP TTL:64 TOS:0x0 ID:57163 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:430  Seq:9  ECHO
```

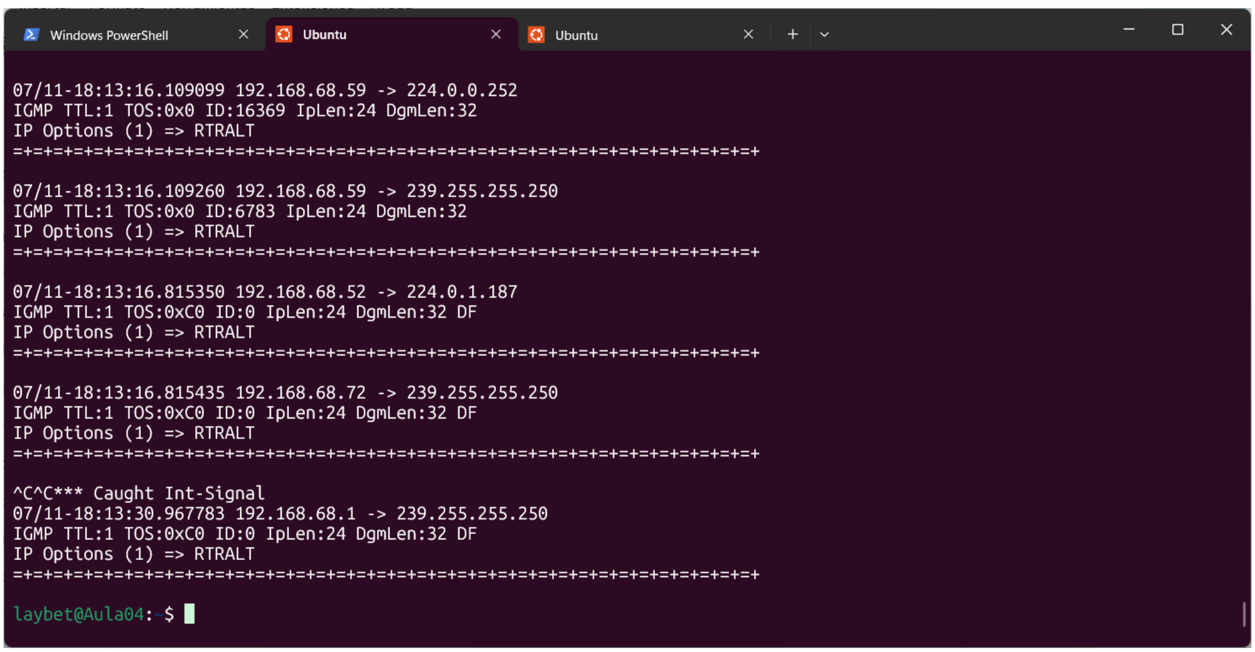
Si seguimos examinando la salida de Snort, podemos fijarnos en algunas cosas más:

- A cada petición de ECHO, le corresponde una ECHO REPLY
- Las direcciones IP origen -> destino, se invierten en ECHO REPLY
- Tenemos un número de secuencia `Seq` que permite hacer un seguimiento del número de *pings* hechos:

```
07/11-18:00:15.394823 192.168.68.59 -> 8.8.8.8
ICMP TTL:64 TOS:0x0 ID:57163 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:430   Seq:9  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

07/11-18:00:15.413314 8.8.8.8 -> 192.168.68.59
ICMP TTL:116 TOS:0xE0 ID:0 IpLen:20 DgmLen:84
Type:0  Code:0  ID:430  Seq:9  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Procedemos a detener la ejecución de Snort en modo escucha:



3.2 Snort mostrando alertas en consola

Si queremos ejecutar Snort, de modo que muestre alertas en la consola para el tráfico interceptado, podemos usar el siguiente comando:

```
sudo snort -q -A console -i eth1 -c /etc/snort/snort.conf
```

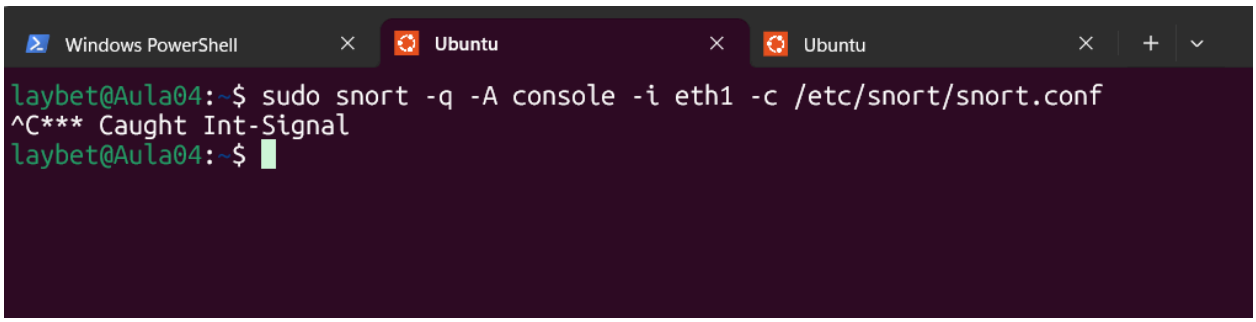
- `-A`: Modalidad de las alertas, en este caso **`console`**.

- Las opciones son: `fast`, `full`, `console`, `test` y `none`.

Sin embargo, no vemos ninguna alerta...

Recordemos que **hemos desactivado todos los ficheros de reglas en `snort.conf`**, salvo el de reglas personalizadas `/etc/snort/rules/`**`local.rules`**.

Presionamos las teclas **`Ctrl+C` en el terminal para detener Snort** nuevamente:



3.3 Primera regla

Vamos a añadir nuestra primera regla al fichero `/etc/snort/rules/`**`local.rules`**.

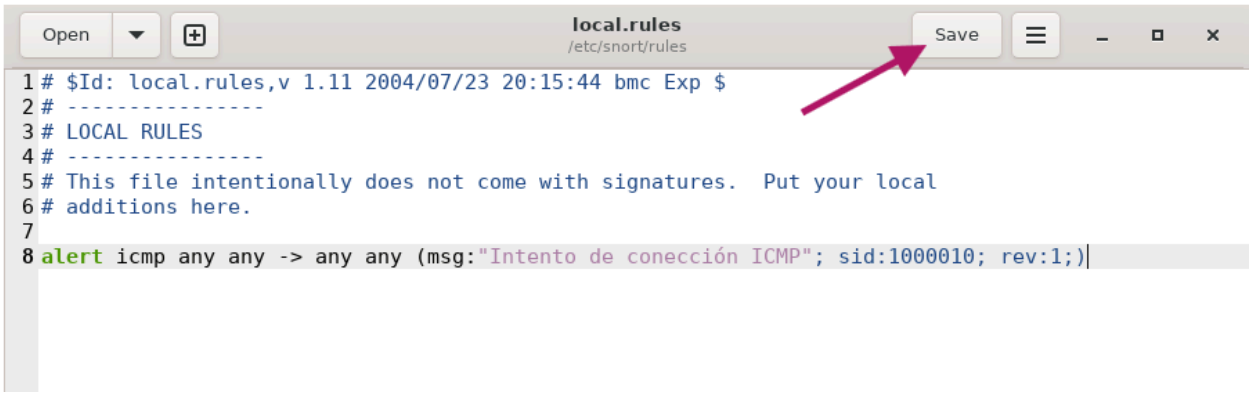
Tal como hemos explicado antes, podemos modificar el fichero `local.rules` con un editor de texto gráfico como `gedit` o con uno de terminal como `nano` o `vim`.

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

Al final del fichero, incluimos la siguiente regla:

```
alert icmp any any -> any any (msg:"Intento de conexion ICMP"; sid:1000010; rev:1;)
```

Tras editar el fichero de reglas, debemos **recordar guardar**.



Esta regla sencilla cualquier petición de ping, sin importar su dirección IP origen o destino.

1. Protocolo: `icmp`
2. Direcciones IP origen y puerto -> IP destino y puerto: `any any -> any any`

3. Mensaje personalizado en la alerta: `msg:"Intento de conexion ICMP"`

Procedemos a ejecutar Snort con el siguiente comando:

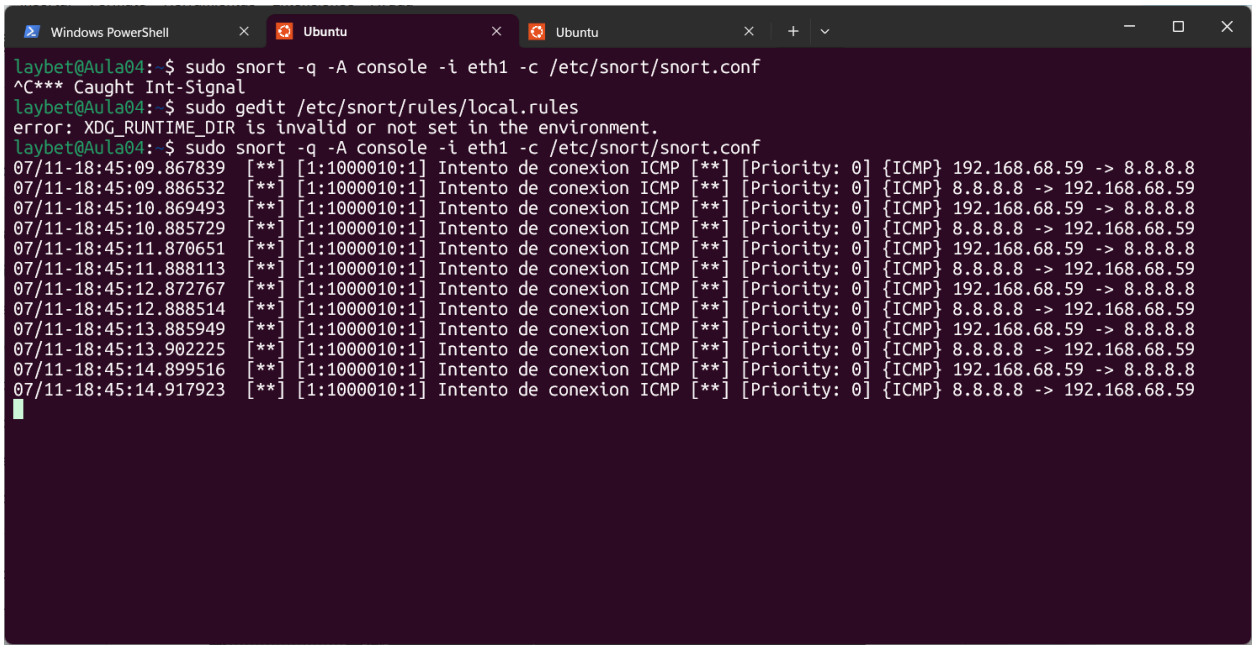
```
sudo snort -q -A console -i eth1 -c /etc/snort/snort.conf
```

Recordemos que al cargar el fichero `snort.conf`, incluye también el fichero de reglas `/etc/snort/rules/local.rules`.

Con Snort ejecutándose, en otro terminal de Ubuntu hagamos ping a la IP 8.8.8.8:

```
ping 8.8.8.8
```

Si hemos hecho todo correctamente, nuestra regla debería lanzarse e imprimir nuestro mensaje personalizado, tal como se muestra en la siguiente captura:



¡Felicitaciones! Has logrado definir y probar tu primera regla en Snort.

Puedes usar **`Ctrl+C` en el terminal** para detener Snort.

4: Definiendo reglas Snort

En la sección anterior hemos:

1. Definido nuestra primera regla en `/etc/snort/rules/local.rules`
2. Hemos puesto Snort en ejecución
3. Hemos comprobado que la regla generaba una alerta al detectar el tráfico especificado

En esta sección explicaremos con más profundidad la sintaxis de las reglas Snort.

Como referencia, recordemos la regla anterior:

```
alert icmp any any -> any any (msg:"Intento de conexion ICMP"; sid:1000010;
rev:1;)
```

Estructura de las reglas Snort

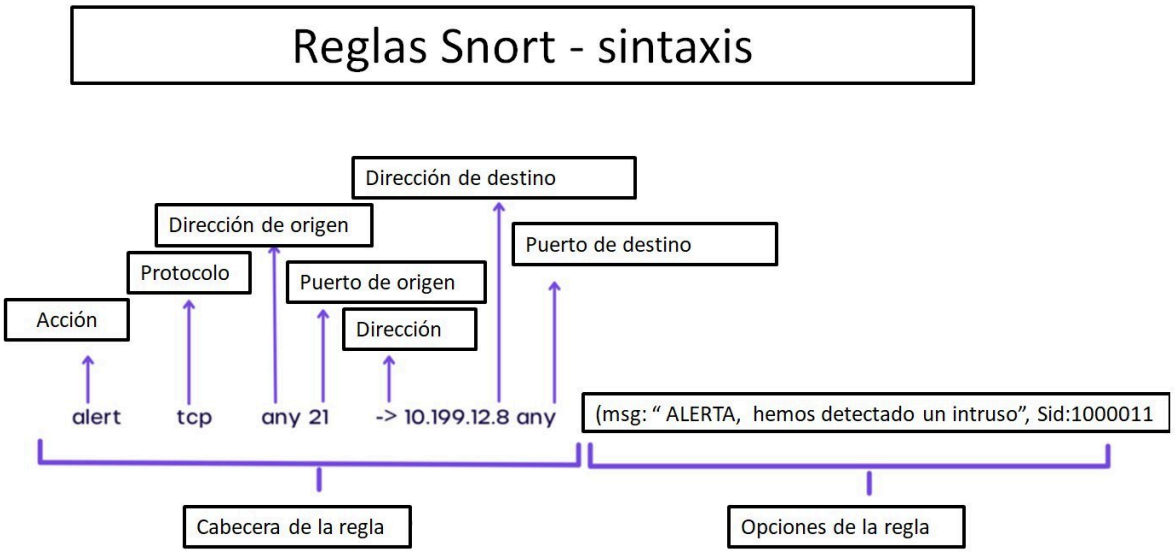
La estructura de las reglas Snort es la siguiente:

```
action protocol sourceIP sourceport -> destinationIP destinationport
([Rule options])
```

En español:

```
acción protocolo IPOrigen puertoOrigen -> IPDestino puertoDestino
([Opciones de la regla])
```

En el siguiente esquema se presenta de forma más visual:



Una regla tiene **2 partes**:

1. **Cabecera**
2. **Opciones** de la regla

Veremos ahora qué criterios podemos establecer en cada parte de la regla.

Cabecera

De nuestra primera regla, la **cabecera** era la siguiente:

```
alert icmp any any -> any any
acción protocolo IPOrigen puertoOrigen -> IPDestino puertoDestino
```

1. ***Acción*** (¿Qué hacer cuando el tráfico coincide con la regla?): **`alert`**, `log`, **`drop`**.
   1.1. `alert`: Genera una alerta
   1.2. `log`: Genera un registro
   1.3. `drop`: Bloquea el tráfico y genera un log
2. ***Protocolo***: `tcp`, `udp`, `icmp`, etc.
3. ***Dirección IP origen***, que ser un bloque de IPs en formato CIDR (por ejemplo: `192.168.1.0/24`)
   3.1. seguido del ***puerto origen***.
4. `->`
5. ***Dirección IP destino***, que ser un bloque de IPs en formato CIDR (por ejemplo: `192.168.1.0/24`)
   5.1. seguido del ***puerto destino***.

Referencia: Rules Headers (documentación de Snort 2.9)

Opciones

Las opciones permiten **refinar las condiciones** de la regla (aplicando algún patrón al contenido del paquete de red) y además **incluir metadatos** para la regla.

Las opciones se especifican entre paréntesis al final de la regla, cada una separada con "`;`".

```
(opcion1:valor; opcion2:valor; ... ;)
```

De nuestra primera regla, las **opciones** de la regla eran las siguientes:

```
(msg:"Intento de conexion ICMP"; sid:1000010; rev:1;)
```

- `msg`: El mensaje asociado a la regla
- `sid`: Identificador de la regla.
    - Al tener un listado extenso de reglas, es útil establecer un **identificador único** para cada regla. Es para ello que se usa la opción `sid`.
    - Utilidad: Al examinar los logs, podríamos filtrar fácilmente las veces que una regla concreta se ha lanzado.
- `rev`: Número de "versión" de la regla.
    - Tiene como propósito mantener un **control de las revisiones** que se le van haciendo a cada regla.
    - Utilidad: En caso tener Snort desplegado, generando logs, puede ocurrir que queremos refinar alguna regla. Al hacer una actualización, deberíamos incrementar el número `rev`, de forma que en los logs quede constancia de qué revisión de la regla se ha lanzado (por ejemplo, para comparar con logs en el pasado)

Las reglas tienen una **lista muy extensa de opciones**, que se dividen en **4 categorías**. En este laboratorio nos centramos solo en algunas concretas.

Entre las opciones más útiles, tenemos la opción:

- `content`: Fragmento (o patrón) contenido tráfico de red.

En la siguiente regla, tendríamos un ejemplo:

```
alert tcp any any -> any 80 (content:"GET";)
```

- Siendo tráfico TCP (capa de transporte)
- De cualquier origen (IP y puerto)
- Con destino cualquier IP y puerto **80** (servicio HTTP)
- **Si el paquete contiene la palabra GET** en cualquier parte, la regla se lanza.

Referencia: Rule Options (documentación de Snort 2.9)

Conociendo la estructura y sintaxis de las reglas, es momento de pasar a la práctica.

4.1 Alerta de ping desde IPs concretas

Las direcciones `8.8.8.8` y `8.8.4.4` son servidores DNS de Google.

Se pide en este caso alertar ante una respuesta de ping proveniente de estos servidores.

A continuación, especifica **2 reglas** (una para cada dirección IP concreta) que cumplan lo siguiente:

- Alerta ante una **respuesta** Ping (para ello, se utiliza la opción `itype`)
- En el mensaje de la alerta: Tus iniciales en mayúscula, seguido de una explicación del propósito de la regla.
- Un identificador único para cada regla, con el formato (numérico): **<año><mes>**`00X`.
- Un número de revisión

Como referencia, puedes utilizar la siguiente regla:

```
alert icmp any any -> any any (msg:"..."; itype:1; sid:20201200X; rev:1;)
```

El campo `itype` de ICMP establece el propósito del paquete ICMP. Por ejemplo:

- `itype:8` corresponde a Echo
- `itype:30` corresponde a Traceroute
- `itype:0` corresponde a Echo Reply

Incluye tu respuesta a continuación:

Solución

```
alert icmp 8.8.8.8 any -> any any (msg:"LCZ Respuesta ping de DNS 1
de Google"; itype:0; sid:202507001; rev:1;)

alert icmp 8.8.4.4 any -> any any (msg:"LCZ Respuesta ping de DNS 2
de Google"; itype:0; sid:202507002; rev:1;)
```

Ahora, procedemos a editar el fichero `/etc/snort/rules/local.rules`:

1. Comentamos la primera regla que habíamos definido

```
# alert icmp any any -> any any (msg:"Intento de conexion ICMP";
sid:1000010; rev:1;)
```

2. Incluimos nuestras reglas personalizadas
3. Guardamos
4. **Iniciamos la ejecución de Snort** con el siguiente comando:

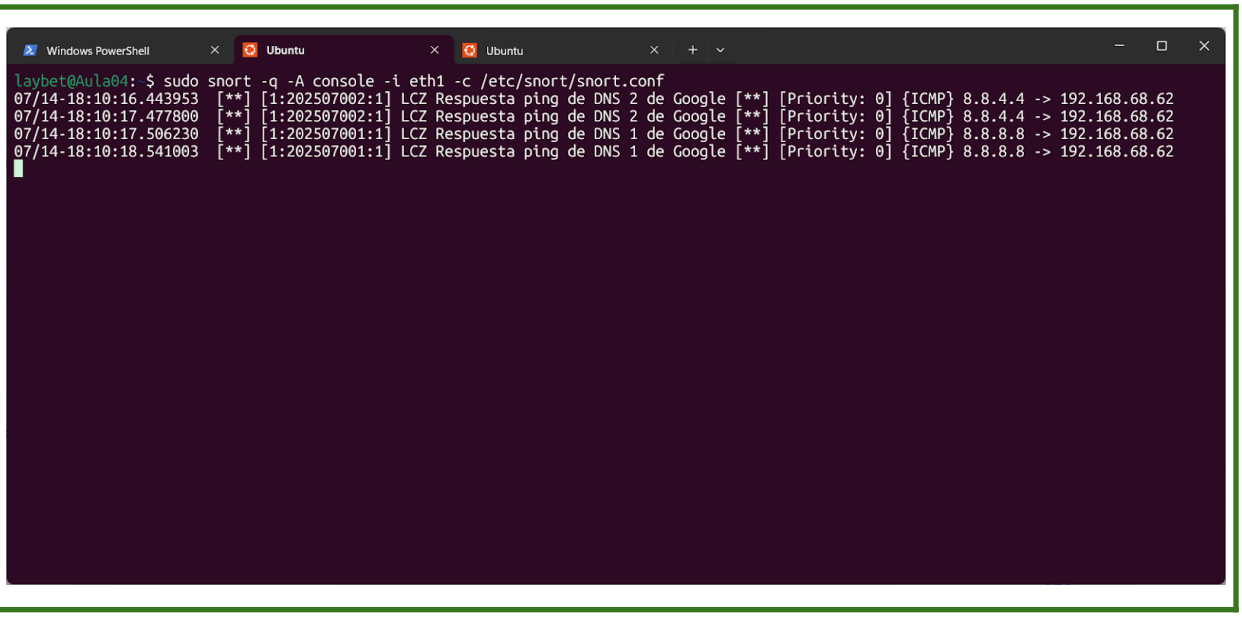
```
sudo snort -q -A console -i eth1 -c /etc/snort/snort.conf
```

Para comprobar que la regla se lanza correctamente, ejecute los siguientes comandos de ping:

```
ping 1.1.1.1 -c 2
ping 8.8.4.4 -c 2
ping 8.8.8.8 -c 2
```

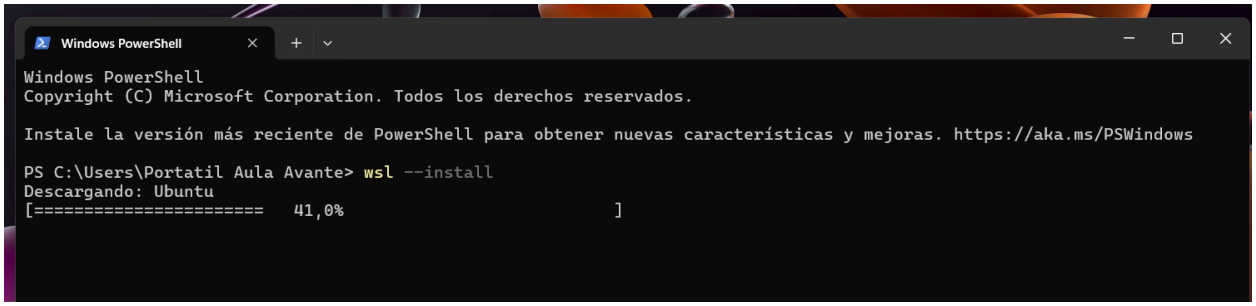
Incluya a continuación, una captura de la **salida de Snort con las alertas generadas**:

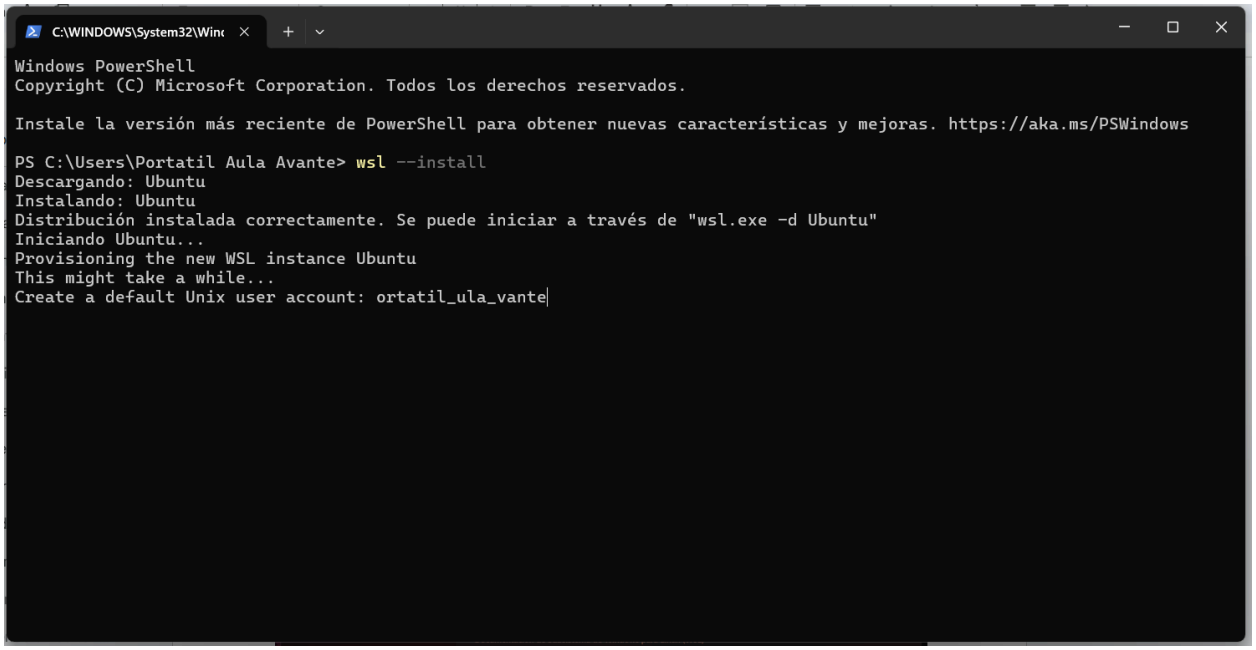
Solución

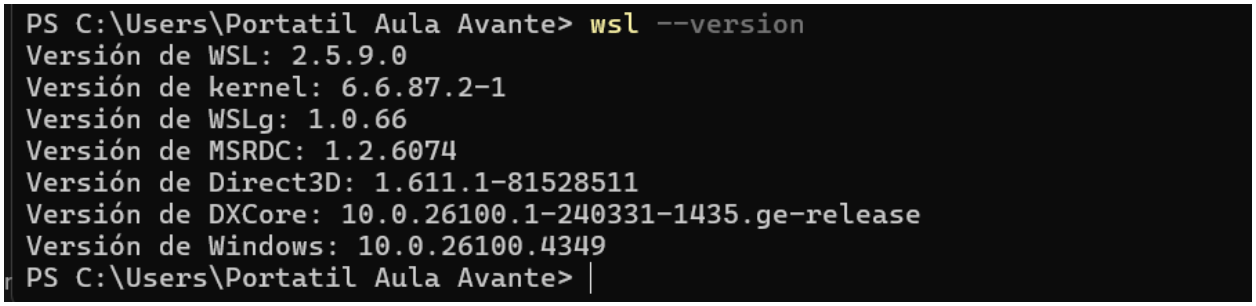


Anexo 1: Instalación de WSL

Estas son algunas de las capturas de pantalla que vamos a recolectar durante la actividad:








Le damos la bienvenida al Subsistema de Windows para Linux
General
Trabajando en sistemas de archivos
Aplicaciones de GUI
Aceleración de la GPU
Integración de redes
Administración de distribuciones
Integración de Docker Desktop
Integración de VS Code
Configuración
Te damos la bienvenida a WSL
El Subsistema de Windows para Linux (WSL) le permite ejecutar sus herramientas, utilidades, aplicaciones y flujos de trabajo favoritos de Linux directamente en Windows.
Dedique un momento a obtener una vista previa de algunas de las características favoritas de la comunidad o vea nuestra documentación completa.
Documentación de Subsistema de Windows para Linux (WSL)
Procedimientos recomendados para la configuración
Introducción a Linux