



Detección y Prevención de Intrusiones



[Snort](#). Logo Snort (CC0)

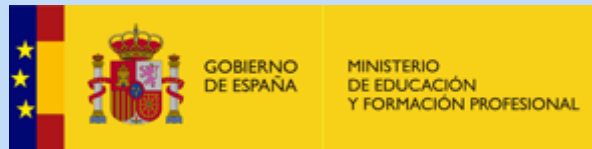
Como se ha visto, las reglas apócrifas de una buena política de ciberseguridad son las siguientes:

- ✓ **Protegerse de todo lo conocido**, instalando las últimas versiones de SW, utilizando aplicaciones antimalware y activando todos los escudos posibles.
- ✓ **Alertar de lo desconocido**, disponiendo de un *Sistema de Alerta Temprana* que informe de cualquier actividad fuera de lo común, para que se analice rápidamente y se averigüe si se trata de una amenaza real o potencial
- ✓ **Asumir la vulnerabilidad**, considerando que a pesar de aplicar sistemáticamente las dos reglas anteriores, los incidentes siempre se pueden llegar a dar, por lo que será necesario dotar planes de respuesta, planes de acción, análisis forense, políticas consistentes de respaldos y, en caso extremo, instalaciones gemelas listas para entrar en acción en cualquier momento (*hot stand by*).

Este módulo profesional está relacionado con las tres reglas, pero con énfasis en la segunda de ellas, esto es, en la **detección y análisis rápido de los incidentes de seguridad**, la **extracción de conclusiones de dicho análisis**, y la **aplicación de estas conclusiones a las políticas de prevención de incidentes**. Estas labores son las que se realizan fundamentalmente en un Centro de Operaciones de Seguridad (SOC).

Un SOC está compuesto por una plataforma hardware, un software de detección y análisis, y un equipo de expertos que estudian cada caso con objeto de reforzar la estrategia ante los incidentes de seguridad, efectuando prevención activa de incidentes. Por lo que respecta al software, está constituido por agentes de detección/prevención de intrusiones (IDS), que se sitúan en cualquier máquina que se considere crítica y/o vulnerable, y por un sistema

centralizado de correlación de incidentes y extracción de información refinada acerca de los mismos.



[Ministerio de Educación y Formación Profesional](#) (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

1.- Prototipo de un SOC.

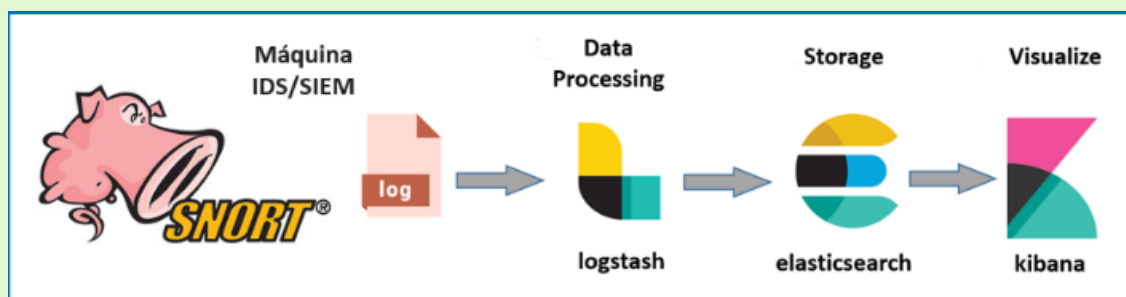


Caso práctico

Los módulos 6 y 7 del presente programa formativo tienen por objeto el **desarrollo de un prototipo real y operativo de un SOC**, empezando por sus fundamentos, esto es, la *Detección de Intrusiones* y la posterior *Gestión de la Información y los Incidentes de Seguridad*.

Este prototipo quedará instalado sobre una única máquina en primera instancia, si bien se incluirán las instrucciones detalladas para el despliegue de agentes de detección en las máquinas perimetrales de la Zona Desmilitarizada, o en las máquinas clave de la empresa (por ejemplo, si se trata de una factoría, en MES, SCADA, PLC, etc.).

En el módulo 6 se desarrollará el procedimiento de instalación y configuración del IDS/IPS más extendido, la aplicación Snort. Por otra parte, en el módulo 7 se hará lo propio con la solución SIEM más utilizada en estos momentos, el Stack ELK de Elasticsearch.

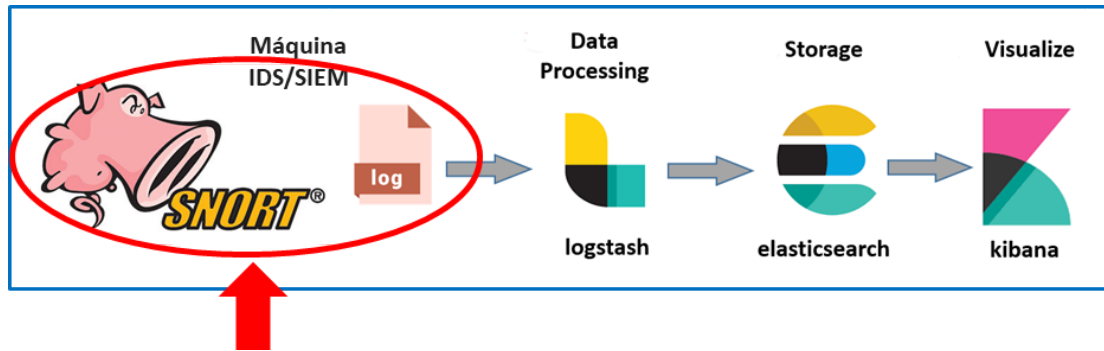


[Francisco Artés - Elaboración Propia](#). IDS Snort & SIEM ELK ([CC0](#))

Detección y Prevención de Intrusiones

- ✓ **IDS, Intrusion Detection System.** Sistema de Detección de Intrusiones. Este sistema permite analizar y supervisar el tráfico de una red, en busca de señales que indiquen que los atacantes están utilizando una amenaza conocida para infiltrarse o robar datos de su red. Por lo general, además de instalar un IDS las empresas suelen instalar y mantener una base de datos de amenazas conocidas y comparar la actividad actual de la red con dichas amenazas para detectar diferentes tipos de comportamientos, tales como violaciones de políticas de seguridad, malware y escaneo de puertos.
- ✓ **IPS, Intrusion Prevention System.** Sistema de Prevención de Intrusiones. La ubicación habitual de este sistema es la misma área de red en la que está situado el cortafuegos, esto es, entre la red externa y la red interna. A diferencia del IDS, que sólo es un monitor, el IPS bloquea proactivamente el tráfico en función de las reglas

establecidas en el perfil de seguridad, siempre y cuando el paquete en cuestión pueda suponer una amenaza conocida para la seguridad del entorno.



[Francisco Artés - Elaboración Propia](#). *Snort en el SOC* ([CC0](#))

1.1.- Prevención de Intrusiones.

Para activar Snort como un IPS, se deberá disponer de una máquina con dos interfaces de red (eth0 y eth1).

Normalmente, en una máquina de este tipo, se encontrará configurado un bridge entre las dos interfaces para transferir paquetes de forma transparente entre las dos redes unidas por la máquina, que habrá que desactivar antes de arrancar Snort en modo inline.

Una vez configuradas las opciones necesarias para arrancar Snort en modo IPS, las reglas de rechazo de tráfico serán de la siguiente forma:

```
drop tcp 192.168.1.52 any -> $HOME_NET any (msg:"ATAQUE SSH";sid:3000003)
```

Con esta instrucción se descartarán los paquetes TCP procedentes de cualquier Puerto de la IP indicada que se dirijan a la red interna, registrando el evento en el log de alertas con el texto indicado.

Existen multiples posibilidades de tratamiento, que se pueden estudiar en el manual de Snort.

1.2.- Snort - El IDS/IPS de Código Abierto.

Snort es el **sistema de Detección y Prevención de intrusiones de código abierto más utilizado**, tanto en el ámbito privado como en el empresarial.

Contiene un **motor de reglas** que permite definir las actividades a detectar, sean maliciosas o no, para después identificar posibles paquetes que cualifiquen con dichas reglas, generando a continuación **alertas para los usuarios de la red**.

Snort tiene pues **tres funciones principales**:

- ✓ **Rastreo** de paquetes
- ✓ **Registro** de paquetes para notificación online y análisis offline
- ✓ **Prevención** de intrusiones en la red con base en las amenazas conocidas



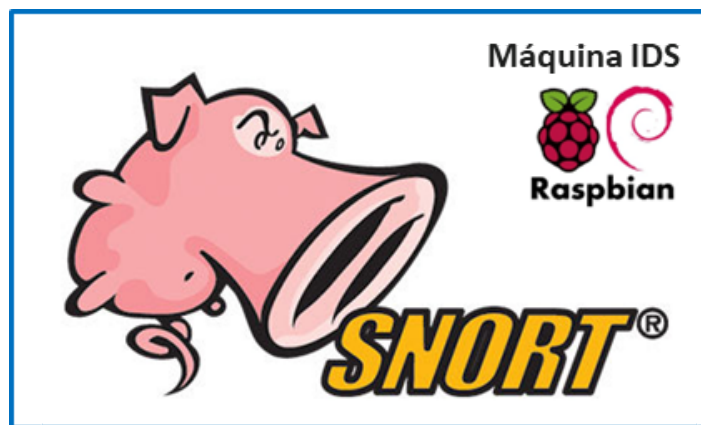
[Snort](#). Logotipo de Snort ([CC0](#))

1.3.- Instalación y Configuración de Snort.

Procedemos a instalar Snort sobre Raspbian ejecutando la secuencia clásica de comandos:

```
sudo apt-get update  
  
sudo apt-get upgrade  
  
sudo apt install snort
```

Durante la instalación informaremos al programa instalador que utilizaremos una configuración base para un rango de 256 direcciones de red (24 bits): 192.168.1.0/24 (el programa preguntará expresamente por esto).



[Francisco Artés - Elaboración Propia](#). Snort IDS en Raspbian (CC0)

The image is a screenshot of a terminal window. The title bar shows 'pi@DMZ1: ~'. The terminal content shows the command 'ps -ef|grep snort' being executed. The output lists two processes: 'snort' (PID 22296) and 'pi' (PID 23091). The 'snort' process is running as root (UID 0) and is configured with various options including '-m 027', '-D', '-d -l /var/log/snort', '-u snort', '-g snort', '-c /etc/snort/snort.conf', and '-S HOME_NET=[192.168.1.0/24] -i eth0'. The 'pi' process is running 'grep' to search for 'snort' in the command list.

```
pi@DMZ1:~ $ ps -ef|grep snort  
snort    22296      1  0 08:40 ?        00:00:00 /usr/sbin/snort -m 027 -D -d -l  
/var/log/snort -u snort -g snort -c /etc/snort/snort.conf -S HOME_NET=[192.168.1  
.0/24] -i eth0  
pi       23091  23041  0 09:00 pts/0    00:00:00 grep --color=auto snort  
pi@DMZ1:~ $
```

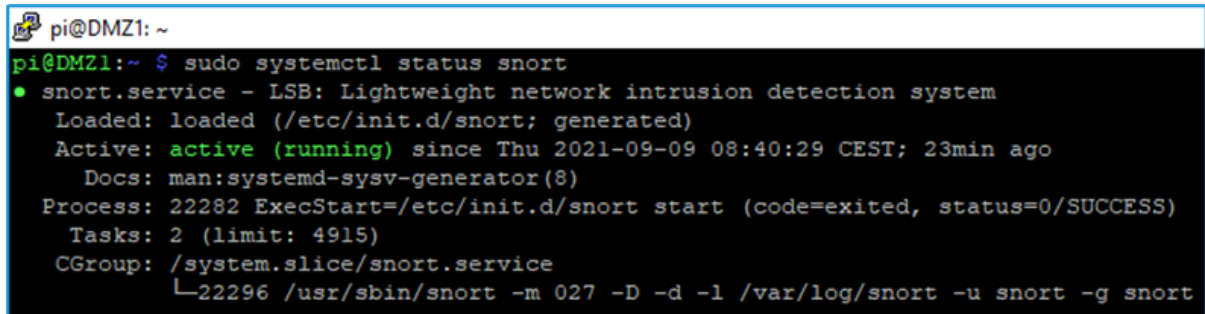
[Francisco Artés - Elaboración Propia](#). Captura de Pantalla del proceso Snort (CC0)

1.4.- Inicio, Arranque y Parada de Snort.

Al finalizar el proceso anterior, Snort quedará instalado como un servicio del sistema, por lo que se podrá controlar su funcionamiento con el comando **systemctl** con las opciones siguientes:

- ✓ **enable**. Habilita el arranque de Snort en el inicio de la máquina.
- ✓ **disable**. Inhabilita el arranque de Snort en el inicio de la máquina.
- ✓ **start**. Arranca Snort en cualquier momento.
- ✓ **stop**. Detiene Snort en cualquier momento.
- ✓ **status**. Muestra el estado de ejecución de Snort.

```
sudo systemctl [opción] snort
```



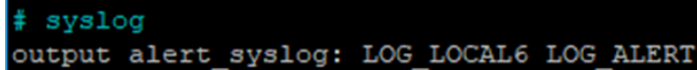
```
pi@DMZ1: ~
pi@DMZ1:~ $ sudo systemctl status snort
● snort.service - LSB: Lightweight network intrusion detection system
   Loaded: loaded (/etc/init.d/snort; generated)
   Active: active (running) since Thu 2021-09-09 08:40:29 CEST; 23min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 22282 ExecStart=/etc/init.d/snort start (code=exited, status=0/SUCCESS)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/snort.service
           └─22296 /usr/sbin/snort -m 027 -D -d -l /var/log/snort -u snort -g snort
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con el Status de Snort ([CC0](#))

1.5.- Ficheros de Configuración Básica de Snort.

Editamos el fichero de configuración de Snort, localizamos el apartado “syslog” y cambiamos las facilidades de alerta como se indica en la captura adjunta:

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

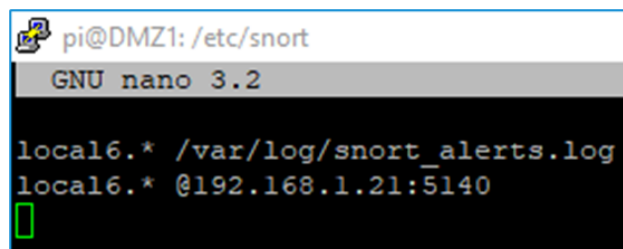


```
# syslog
output alert_syslog: LOG_LOCAL6 LOG_ALERT
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con las Facilidades de Snort (CC0)

A continuación escribimos un fichero de configuración de Rsyslog para grabar logs de Snort local y remotamente, especificando la facilidad “local6” que hemos dado de alta en la configuración de Snort y detallando la dirección IP de la interfaz eth0 de nuestro host:

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



```
pi@DMZ1: /etc/snort
GNU nano 3.2

local6.* /var/log/snort_alerts.log
local6.* @192.168.1.21:5140
█
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Configuración de Rsyslog para Snort (CC0)

Rearrancamos Rsyslog y Snort:

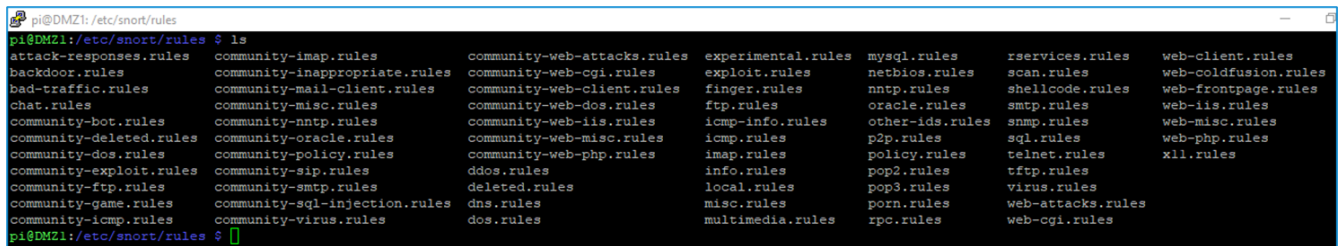
```
sudo systemctl restart rsyslog
```

```
sudo systemctl restart snort
```

1.6.- Fichero de Registro de Alertas de Snort.

Desde el momento en el que arranca, Snort empieza a grabar alertas en su fichero de log `/var/log/snort_alerts.log` puesto que, aunque todavía no hayamos programado reglas locales para nuestro laboratorio (en el fichero `local.rules`), el programa arrancará además con todas las reglas recomendadas por la Comunidad Snort, con objeto de detectar todos los ataques conocidos hasta el momento.

Los ficheros que contienen dichas reglas adicionales se encuentran en el mismo directorio que nuestro fichero `local.rules`, como se puede ver en la captura adjunta:



```
pi@DMZ1: /etc/snort/rules
pi@DMZ1:/etc/snort/rules $ ls
attack-responses.rules  community-imap.rules      community-web-attacks.rules  experimental.rules  mysql.rules      rservices.rules  web-client.rules
backdoor.rules          community-inappropriate.rules  community-web-cgi.rules     exploit.rules        netbios.rules    scan.rules        web-coldfusion.rules
bad-traffic.rules       community-mail-client.rules    community-web-client.rules  finger.rules         nntp.rules       shellcode.rules   web-frontpage.rules
chat.rules              community-misc.rules           community-web-dos.rules     ftp.rules            oracle.rules      smtp.rules        web-iis.rules
community-bot.rules     community-nntp.rules          community-web-iis.rules     icmp-info.rules      other-ids.rules  snmp.rules        web-misc.rules
community-deleted.rules community-oracle.rules         community-web-misc.rules    icmp.rules           p2p.rules        sql.rules         web-php.rules
community-dos.rules     community-policy.rules        community-web-php.rules     imap.rules           policy.rules      telnet.rules      x11.rules
community-exploit.rules community-sip.rules            ddos.rules                 info.rules           pop2.rules        tftp.rules
community-ftp.rules     community-smtp.rules          deleted.rules               local.rules          pop3.rules        virus.rules
community-game.rules    community-sql-injection.rules  dns.rules                  misc.rules           porn.rules        web-attacks.rules
community-icmp.rules    community-virus.rules          dos.rules                  multimedia.rules     rpc.rules         web-cgi.rules
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con los ficheros de Reglas de Snort ([CC0](#))

1.7.- Torre de Protocolos ISO-OSI.

Se trata de un modelo que define una torre o pila de protocolos (Protocol Stack) multicapa, que se estructura en siete niveles:

- ✓ **Capa Física.** Define el HW de conexión física.
- ✓ **Capa de Enlace de Datos.** Controla la transferencia de datos en la red, mediante protocolos de bajo nivel.
- ✓ **Capa de Red.** Introduce el direccionamiento y la comunicación entre diferentes redes.
- ✓ **Capa de Transporte.** Introduce el concepto de Integridad, asegurando que los datos no se deterioran durante la transferencia.
- ✓ **Capa de Sesión.** Incorpora el concepto de sesión, esto es inicio, desarrollo y fin de la transmisión.
- ✓ **Capa de Presentación.** En esta capa se asegura que la información se transfiera de forma comprensible para el sistema.
- ✓ **Capa de Aplicación.** Se compone de los servicios y aplicaciones de comunicación estándar a disposición de cualquier usuario.

NIVEL	CAPA	FUNCIÓN	DATOS TÉCNICOS Y PROTOCOLOS DE COMUNICACIONES
7	Aplicación	Del Proceso de Red a la Aplicación	DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6	Presentación	Representación y Cifrado de Datos	Reconocimiento de Datos: HTML, DOC, JPEG, MP3, AVI, Sockets
5	Sesión	Comunicación entre hosts	Establecimiento de Sesión en TCP, SIP, RTP, RPC-Named pipes
4	Transporte	Conexiones Extremo a Extremo y Fiabilidad	TCP, UDP, SCTP, SSL, TLS
3	Red	Determinación de la Ruta y Direccionamiento Lógico	IP, ARP, Ipsec, ICMP, IGMP, OSPF
2	Enlace de Datos	Direccionamiento Físico	Ethernet, 802.11, MAC/LLC, VALN, ATM, HDP, Fibre Channel, Frame Relay, HDLC, PPP, Q.921, Token Ring
1	Física	Medios, Señales y Transmisión Binaria	RS-232, RJ45, V.34, 100-BASE-TX, SDH, DSL, 802.11

[Francisco Artés - Elaboración Propia.](#) Capas de la Torre de Comunicaciones (CC0)

1.8.- Detección Tráfico ICMP con Snort.

En esta sección se verá cómo detectar tráfico ICMP con el IDS Snort.

Este tráfico corresponde al protocolo que soporta el comando ping, especialmente importante tanto para detectar problemas de alcanzabilidad de máquinas, como para descubrir su existencia.

1.8.1.- Posición del Protocolo ICMP en la Torre de Comunicaciones.

NIVEL	CAPA	FUNCIÓN	DATOS TÉCNICOS Y PROTOCOLOS DE COMUNICACIONES
7	Aplicación	Del Proceso de Red a la Aplicación	DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, FTP
6	Presentación	Representación y Cifrado de Datos	Reconocimiento de Datos: HTML, DOC, JPEG, MP3, AVI, Sockets
5	Sesión	Comunicación entre hosts	Establecimiento de Sesión en TCP, SIP, RTP, RPC-Named pipes
4	Transporte	Conexiones Extremo a Extremo y Fiabilidad	TCP, UDP, SCTP, SSL, TLS
3	Red	Determinación de la Ruta y Direccionamiento Lógico	IP, ARP, Ipsec, ICMP , GMP, OSPF
2	Enlace de Datos	Direccionamiento Físico	Ethernet, 802.11, MAC/LLC, VALN, ATM, HDP, Fibre Channel, Frame Relay, HDLC, PPP, Q.921, Token Ring
1	Física	Medios, Señales y Transmisión Binaria	RS-232, RJ45, V.34, 100-BAS, SDH, DSL, 802.11

- Posición del Protocolo ICMP en la Torre de Protocolos

[Francisco Artés - Elaboración Propia](#). Zoom de Stack de Comunicaciones para ICMP (CC0)

1.8.2.- Construcción de Reglas para Snort.

Una regla Snort se compone de la forma siguiente:

Header

- ✓ Acción de la regla
- ✓ Protocolo
- ✓ Dirección IP origen
- ✓ Puerto IP origen
- ✓ Dirección de la operación
- ✓ Dirección IP destino
- ✓ Puerto IP destino

Trailer

- ✓ Mensaje
- ✓ Opciones



[Snort](#). Logo Snort (CC0)



Para saber más

Web oficial de Snort:

www.snort.org

Manual de Snort:

<https://www.snort.org/documents/snort-users-manual>

1.8.3.- Ejemplo de Regla Snort.

Regla

```
alert tcp 192.168.1.23 any -> $HOME_NET any (msg:"Trafico TCP desde Claudia"; sid:666003;)
```

Header

- ✓ Acción de la regla: Alerta
- ✓ Protocolo: TCP
- ✓ Dirección IP origen: 192.168.1.23
- ✓ Puerto IP origen: Cualquiera
- ✓ Dirección de la operación: De izquierda a derecha (->)
- ✓ Dirección IP destino: 192.168.1.0 (cualquier dirección de la red local)
- ✓ Puerto IP destino: Cualquiera

Trailer

- ✓ Mensaje: "Tráfico TCP desde Claudia"
- ✓ Opciones: Identificador del mensaje que cualifica con la regla (666003)

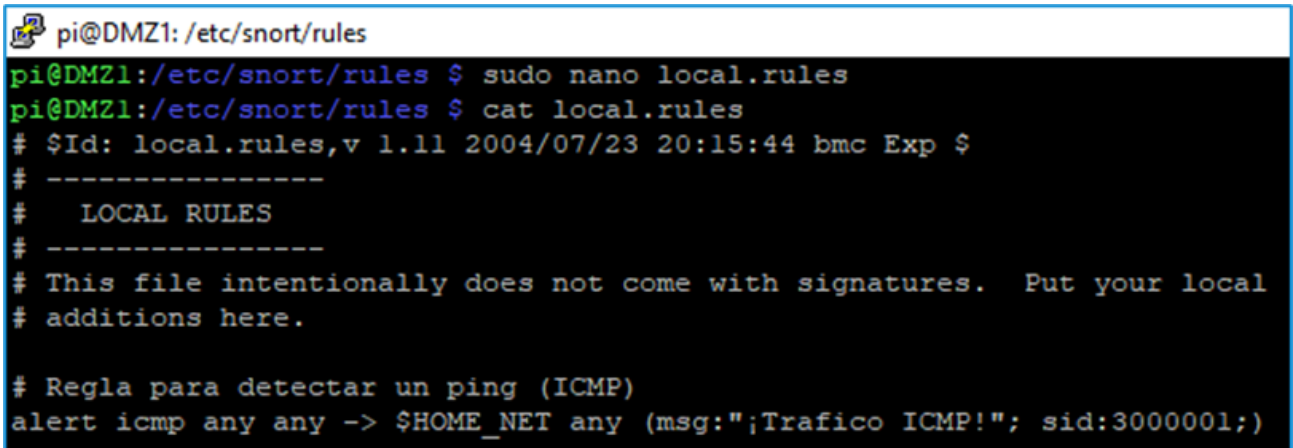
1.8.4.- Configuración de Snort para Detección de Tráfico ICMP.

Grabaremos una regla para detectar tráfico ICMP (ping) en el fichero de configuración de Snort:

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

Tras rearrancar la aplicación, podremos comprobar la grabación de tráfico en tiempo real mediante el comando:

```
tail -f /var/log/snort_alerts.log
```



```
pi@DMZ1: /etc/snort/rules
pi@DMZ1:/etc/snort/rules $ sudo nano local.rules
pi@DMZ1:/etc/snort/rules $ cat 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.

# Regla para detectar un ping (ICMP)
alert icmp any any -> $HOME_NET any (msg:"¡Tráfico ICMP!"; sid:3000001;)
```

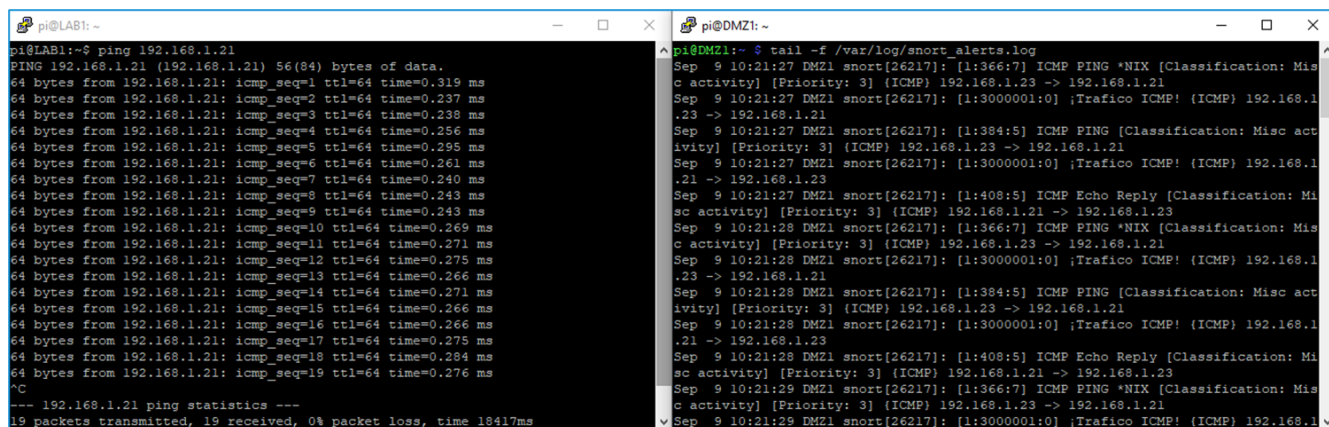
[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con Grabación de Tráfico en Tiempo Real ([CC0](#))

```
# Regla para detectar un ping (ICMP)

alert icmp any any -> $HOME_NET any (msg:"¡Tráfico ICMP!"; sid:3000001;)
```

1.8.5.- Práctica de Detección.

Si visualizamos en vivo con el fichero de log y lanzamos un ping desde otra máquina, veremos que se detecta perfectamente el ping, indicando además desde qué dirección se está efectuando (se verán los pings de todas las máquinas a este host, incluido el router).



```
pi@LAB1:~$ ping 192.168.1.21
PING 192.168.1.21 (192.168.1.21) 56(84) bytes of data.
64 bytes from 192.168.1.21: icmp_seq=1 ttl=64 time=0.319 ms
64 bytes from 192.168.1.21: icmp_seq=2 ttl=64 time=0.237 ms
64 bytes from 192.168.1.21: icmp_seq=3 ttl=64 time=0.238 ms
64 bytes from 192.168.1.21: icmp_seq=4 ttl=64 time=0.256 ms
64 bytes from 192.168.1.21: icmp_seq=5 ttl=64 time=0.295 ms
64 bytes from 192.168.1.21: icmp_seq=6 ttl=64 time=0.261 ms
64 bytes from 192.168.1.21: icmp_seq=7 ttl=64 time=0.240 ms
64 bytes from 192.168.1.21: icmp_seq=8 ttl=64 time=0.243 ms
64 bytes from 192.168.1.21: icmp_seq=9 ttl=64 time=0.243 ms
64 bytes from 192.168.1.21: icmp_seq=10 ttl=64 time=0.269 ms
64 bytes from 192.168.1.21: icmp_seq=11 ttl=64 time=0.271 ms
64 bytes from 192.168.1.21: icmp_seq=12 ttl=64 time=0.275 ms
64 bytes from 192.168.1.21: icmp_seq=13 ttl=64 time=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=14 ttl=64 time=0.271 ms
64 bytes from 192.168.1.21: icmp_seq=15 ttl=64 time=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=16 ttl=64 time=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=17 ttl=64 time=0.275 ms
64 bytes from 192.168.1.21: icmp_seq=18 ttl=64 time=0.284 ms
64 bytes from 192.168.1.21: icmp_seq=19 ttl=64 time=0.276 ms
^C
--- 192.168.1.21 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18417ms
```

```
pi@DMZ1:~$ tail -f /var/log/snort_alerts.log
Sep  9 10:21:27 DMZ1 snort[26217]: [1:366:7] ICMP PING *NIX [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:27 DMZ1 snort[26217]: [1:3000001:0] ;Trafico ICMP! (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:27 DMZ1 snort[26217]: [1:384:5] ICMP PING [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:27 DMZ1 snort[26217]: [1:3000001:0] ;Trafico ICMP! (ICMP) 192.168.1.21 -> 192.168.1.23
Sep  9 10:21:27 DMZ1 snort[26217]: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.21 -> 192.168.1.23
Sep  9 10:21:28 DMZ1 snort[26217]: [1:366:7] ICMP PING *NIX [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:28 DMZ1 snort[26217]: [1:3000001:0] ;Trafico ICMP! (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:28 DMZ1 snort[26217]: [1:384:5] ICMP PING [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:28 DMZ1 snort[26217]: [1:3000001:0] ;Trafico ICMP! (ICMP) 192.168.1.21 -> 192.168.1.23
Sep  9 10:21:28 DMZ1 snort[26217]: [1:408:5] ICMP Echo Reply [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.21 -> 192.168.1.23
Sep  9 10:21:29 DMZ1 snort[26217]: [1:366:7] ICMP PING *NIX [Classification: Misc activity] [Priority: 3] (ICMP) 192.168.1.23 -> 192.168.1.21
Sep  9 10:21:29 DMZ1 snort[26217]: [1:3000001:0] ;Trafico ICMP! (ICMP) 192.168.1.21 -> 192.168.1.23
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con Detección de Ping (CC0)

1.9.- Detección Tráfico SSH con Snort.

SSH significa “Secure Shell”, esto es, sesión segura. Es el protocolo que se usa en la actualidad en lugar del protocolo telnet, que es obsoleto e inseguro. Ocurre lo mismo con SFTP (Secure File Transfer Protocol), que es el sustituto del antiguo protocolo ftp.

Este es el protocolo que se utiliza para abrir sesiones en máquinas remotas, por lo que normalmente también es el empleado para los Ataques con Fuerza Bruta o con Diccionario.

A continuación incluiremos en el fichero de configuración de Snort una regla detectora de SSH, rearrancaremos la aplicación, y lanzaremos solicitudes SSH desde otra máquina para comprobar que se detectan correctamente.

1.9.1.- Posición del Protocolo SSH en la Torre de Comunicaciones.

Zoom de la torre de comunicaciones con detalle para el protocolo SSH:

NIVEL	CAPA	FUNCIÓN	DATOS TÉCNICOS Y PROTOCOLOS DE COMUNICACIONES
7	Aplicación	Del Proceso de Red a la Aplicación	DNS, WWW/HTTP, P2P, EMAIL/POP, SMTP, Telnet, SSH , POP
6	Presentación	Representación y Cifrado de Datos	Reconocimiento de Datos: HTML, DOC, JPEG, MP3, AVI, Sockets
5	Sesión	Comunicación entre hosts	Establecimiento de Sesión: TCP , SP, RTP, RPC-N, named pipes
4	Transporte	Conexiones Extremo a Extremo y Fiabilidad	TCP, UDP, SCTP, SSL, TLS
3	Red	Determinación de la Ruta y Direccionamiento Lógico	IP, ICMP, Ipsec, ICMP, IGMP, OSPF
2	Enlace de Datos	Direccionamiento Físico	Ethernet, 802.11, MAC/LLC, VLAN, HDLC, Fibre Channel, Frame Relay, HDLC, PPP, Q.921, Token Ring
1	Física	Medios, Señales y Transmisión Binaria	RS-485, V.34, 100-BASE-TX, S, 802.11

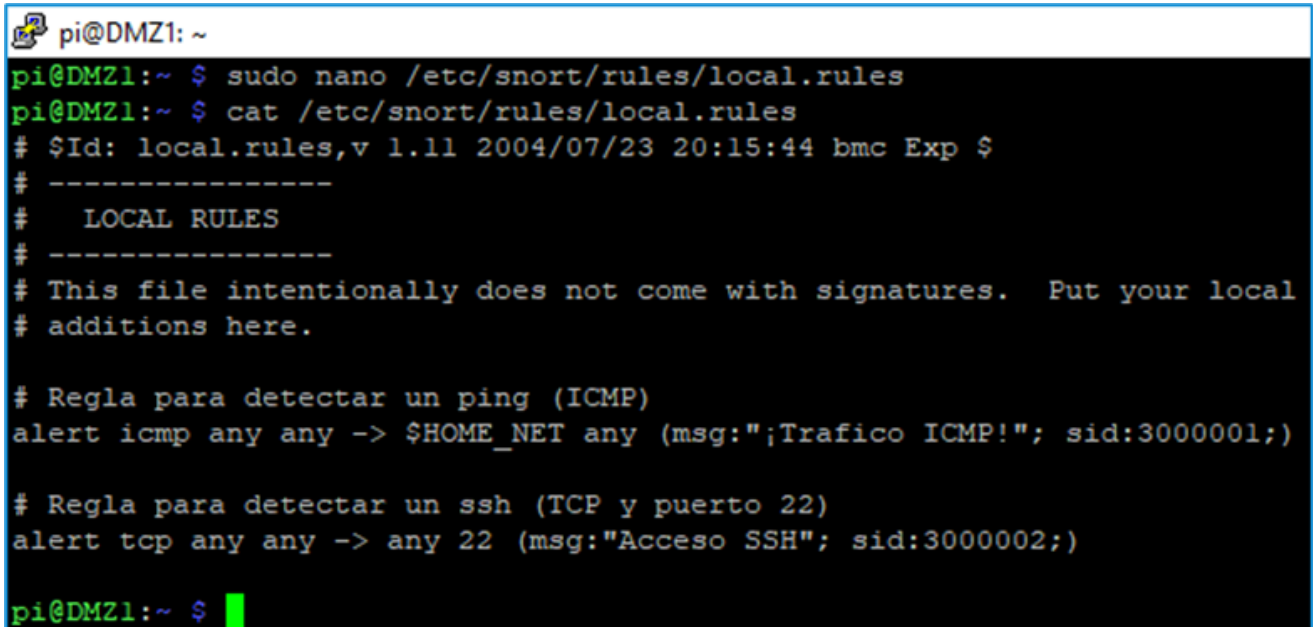
• Posición del Protocolo TCP en la Torre de Protocolos

• Posición del Protocolo SSH en la Torre de Protocolos

[Francisco Artés - Elaboración Propia](#). Posición del Protocolo SSH en la Torre de Comunicaciones (CC0)

1.9.2.- Detalle de la regla específica para detección SSH.

Adición de la regla de detección de tráfico TCP para apertura de sesiones por SSH (tras la inclusión de la nueva regla, se deberá rearrancar Snort mediante systemctl):



```
pi@DMZ1: ~  
pi@DMZ1:~ $ sudo nano /etc/snort/rules/local.rules  
pi@DMZ1:~ $ cat /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.  
  
# Regla para detectar un ping (ICMP)  
alert icmp any any -> $HOME_NET any (msg:";Tráfico ICMP!"; sid:3000001;)  
  
# Regla para detectar un ssh (TCP y puerto 22)  
alert tcp any any -> any 22 (msg:"Acceso SSH"; sid:3000002;)  
  
pi@DMZ1:~ $
```

[Francisco Artés - Elaboración Propia](#). Captura de Pantalla con la Regla de Detección SSH ([CC0](#))

```
# Regla para detectar un ssh (TCP y puerto 22)  
  
alert tcp any any -> any 22 (msg:"Acceso SSH"; sid:3000002;)
```

1.9.3.- Detección de Tráfico TCP/SSH.

Visualizamos de nuevo el fichero snort_alerts.log y comprobaremos que se registran los intentos de establecimiento de sesión SSH por parte de la máquina LAB2, con dirección IP 192.168.1.24:

```
pi@DMZ1:~  
pi@LAB2:~$ ssh 192.168.1.21  
pi@192.168.1.21's password:  
Linux DMZ1 5.10.17-v7l+ #1421 SMP Thu May 27 14:00:13 BST 2021 armv7l  
  
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Thu Sep  9 10:48:01 2021 from 192.168.1.24  
  
Wi-Fi is currently blocked by rfkill.  
Use raspi-config to set the country before use.  
  
pi@DMZ1:~$ ls -l  
total 480700  
drwxr-xr-x 2 pi pi      4096 jun 18 08:32 arduino-cli  
-rw-r--r-- 1 pi pi       88 jun 7 20:54 ARRANCAR_MINISOCS  
-rw-r--r-- 2 pi pi     4096 may 7 16:52 Bookshelf  
-rw-r--r-- 1 pi pi       33 sep 7 11:18 desactivar_vlan  
drwxr-xr-x 2 pi pi     4096 may 7 17:07 Desktop  
drwxr-xr-x 2 pi pi     4096 may 7 17:07 Documents  
drwxr-xr-x 2 pi pi     4096 may 7 17:07 Downloads
```

Francisco Artés - Elaboración Propia. Captura de Pantalla con Detección de Tráfico SSH (CC0)

2.- Bibliografía.

[Bibliografía](#) (pdf - 60498 B)