



**APUNTES 06**

# **IDS/IPS SNORT**

**INCIDENTES DE CIBERSEGURIDAD**

**ALBA MOREJÓN GARCÍA**

**2024/2025**

**Ciberseguridad en Entornos de las Tecnologías de la Información**

## ÍNDICE

1. Prototipo de un SOC.
  - 1.1. Prevención de Intrusiones.
  - 1.2. Snort - El IDS/IPS de Código Abierto.
  - 1.3. Instalación y Configuración de Snort.
  - 1.4. Inicio, Arranque y Parada de Snort.
  - 1.5. Ficheros de Configuración Básica de Snort.
  - 1.6. Fichero de Registro de Alertas de Snort.
  - 1.7. Torre de Protocolos ISO-OSI.
  - 1.8. Detección Tráfico ICMP con Snort.
    - 1.8.1. Posición del Protocolo ICMP en la Torre de Comunicaciones.
    - 1.8.2. Construcción de Reglas para Snort.
    - 1.8.3. Ejemplo de Regla Snort.
    - 1.8.4. Configuración de Snort para Detección de Tráfico ICMP.
    - 1.8.5. Práctica de Detección.
  - 1.9. Detección Tráfico SSH con Snort.
    - 1.9.1. Posición del Protocolo SSH en la Torre de Comunicaciones.
    - 1.9.2. Detalle de la regla específica para detección SSH.
    - 1.9.3. Detección de Tráfico TCP/SSH.

## Detección y Prevención de Intrusiones

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.

## 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.

## 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.

### 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 múltiples 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



### 1.8.1.- POSICIÓN DEL PROTOCOLO ICMP EN LA TORRE DE COMUNICACIONES

### 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

### 1.8.3.- EJEMPLO DE REGLA SNORT.

Regla

```
alert tcp 192.168.1.23 any -> $HOME_NET any (msg:"Tráfico 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)

```
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;)
```

### 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
```

# 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=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=18 ttl=64 time=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=19 ttl=64 time=0.266 ms
64 bytes from 192.168.1.21: icmp_seq=20 ttl=64 time=0.266 ms
^C

```

```
pi@DMZ1:~$ tail -f /var/log/snort_alerts.log
Sep  9 10:21:27 DMZ1 snort[26217]: [1:366:7] ICMP PING *WIX [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] ,Tráfico 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] ,Tráfico 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 *WIX [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] ,Tráfico 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] ,Tráfico 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

```

## 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.



- c. Para implementar las primitivas de mantenimiento remoto.
  - d. Para ninguna opción de las anteriores.
8. ¿En qué posición de una regla Snort se sitúa el "Mensaje"?:
- a. Trailer.
  - b. No Aplica.
  - c. Header.
9. ¿En qué posición de una regla Snort se sitúa el "Protocolo"?:
- a. Trailer.
  - b. Header.
  - c. No Aplica.
10. ¿En qué posición de una regla Snort se sitúa la "Dirección IP Origen"?:
- a. No Aplica.
  - b. Header.
  - c. Trailer.

## TEST II

1. ¿Qué capa de la Torre ISO-OSI controla la transferencia de datos en la red?:
- a. Capa Presentación.
  - b. Capa Sesión.
  - c. Capa Red.
  - d. Capa Aplicación.
  - e. Capa Enlace de Datos.
  - f. Capa Física.
  - g. Capa Transporte.
2. ¿Qué capa de la Torre ISO-OSI define el hardware de conexión?:
- a. Capa Presentación.
  - b. Capa Transporte.
  - c. Capa Red.
  - d. Capa Física.
  - e. Capa Sesión.
  - f. Capa Aplicación.
  - g. Capa Enlace de Datos.
3. ¿En qué capa de la Torre ISO-OSI se sitúa el protocolo ICMP?:
- a. Capa Sesión.
  - b. Capa Enlace de Datos.
  - c. Capa Transporte.
  - d. Capa Red.
4. ¿Qué entidad técnica utiliza Snort para enviar la información de logging a una máquina remota?
- a. Un Socket.
  - b. Una Linux Facility.
  - c. Un Linux Pipe.
5. ¿En qué posición de una regla Snort se sitúa el "Puerto IP Destino"?:
- a. Header.
  - b. Trailer.
  - c. No Aplica.
6. ¿En qué posición de una regla Snort se sitúa la "Acción de la Regla"?:
- a. No Aplica.
  - b. Trailer.
  - c. Header.
7. ¿Que estrategia permite estar preparado ante cualquier incidente?:
- a. Las instalaciones gemelas que pueden entrar en acción en cualquier momento.
  - b. Los planes de respuesta.
  - c. Los planes de acción.
  - d. Las políticas consistentes de respaldos.
  - e. Todas las anteriores.
  - f. El análisis forense.
8. ¿Qué capa de la Torre ISO-OSI se compone de los servicios de comunicación estándar a disposición de cualquier usuario?:
- a. Capa Red.
  - b. Capa Transporte.
  - c. Capa Sesión.
  - d. Capa Física.
  - e. Capa Aplicación.

- f. Capa Enlace de Datos.
  - g. Capa Presentación.
9. ¿Cuál es la misión de Snort en el SOC?:
- a. Detección y Prevención de Intrusiones.
  - b. Monitorización de la información.
  - c. Almacenamiento de la información.
  - d. Filtrado de la información de los logs.
10. ¿Qué funcionalidad tiene Snort?:
- a. Es un IDS/IPS totalmente funcional.
  - b. Es un IDS con algunas funciones de IPS.
  - c. Es sólo un IDS.

#### RESPUESTAS

TEST I: 1 ), 2 ), 3 ), 4 ), 5 ), 6 ), 7 ), 8 ), 9 ), 10 )

TEST II: 1 ), 2 ), 3 ), 4 ), 5 ), 6 ), 7 ), 8 ), 9 ), 10 )

#### La Detección Multipunto de Incidentes

En la Unidad 6 hemos estudiado cómo instalar y configurar el IDS Snort, situándolo en la misma máquina en la que estará el SIEM que procesará su información una vez filtrada y almacenada.

Sin embargo, aunque esta configuración es habitual en los laboratorios, no es la corriente en las instalaciones reales. En cualquier entorno productivo suele haber una sonda Snort en cada una de las máquinas perimetrales, comprometidas, vulnerables, etc., cuya información de logging se ha de redirigir hacia una única máquina en la que estará instalado el SIEM (Unidad 7).

En esta tarea abordaremos el registro de logs en los diferentes agentes IDS en tiempo real utilizando la aplicación SNORT.

Para el desarrollo de la práctica nos centraremos en la red DMZ, en concreto sobre el SNORT situado en dicha zona y una de las máquinas, la cual, tiene instalado los servicios SSH, HTTP y MySQL. Esta última máquina se proporciona en esta tarea (WebServer).

#### Apartado 1: Configurar las máquinas virtuales para que tengan comunicación completa.

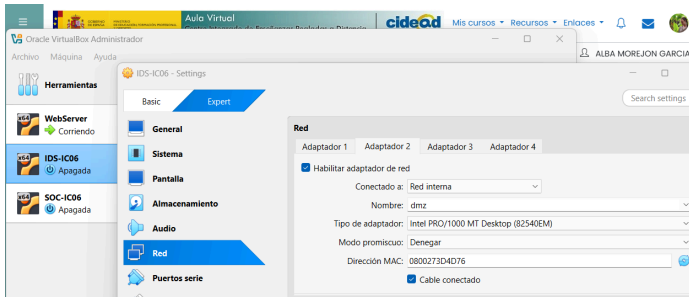
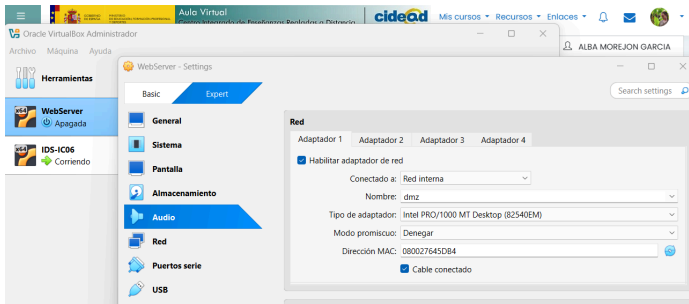
Deberás efectuar las siguientes tareas:

- Crear la máquina IDS (Snort) con dos interfaces de red y configurarla para que permita la comunicación completa entre ambas interfaces. Una tomará el rol de adaptador puente con la red externa y la otra interfaz sería la puerta de enlace predeterminada de la red DMZ. Se debe mostrar el fichero de configuración de las interfaces de red. Se recomienda el uso de Ubuntu SERVER o DEBIAN.
- Configurar la máquina IDS para que las máquinas de la red interna DMZ (WebServer) se puedan comunicar correctamente con el exterior. Se debe conseguir acceso a internet y a la red externa.
- Crear una máquina virtual que se denomine SOC, la cual esté conectada a la red externa (adaptador puente). Esta máquina debe tener comunicación con el WebServer. La máquina SOC debe tener interfaz gráfica, por lo que se recomienda la instalación de Ubuntu Desktop.

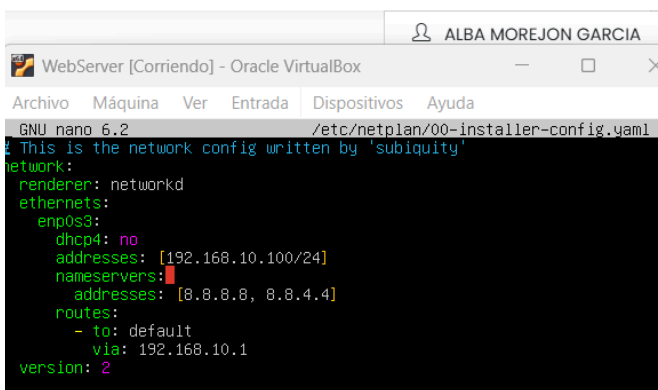
NOMBRE	SISTEMA OPERATIVO	ADAPTADOR	IP
IDS-IC06	Ubuntu Server 24.04.2	Adaptador puente 08:00:27:F7:6F:F7 enp0s3	dhcp 192.168.0.36/24
		Red interna (dmz) 08:00:27:7E:1C:F2 enp0s8	192.168.10.1/24
SOC-IC06	Ubuntu Desktop 24.04.1	Adaptador puente 08:00:27:4B:B1:F1 enp0s3	dhcp 192.168.0.41/24
WebServer-IC06	Ubuntu 64bits	Red interna (dmz) 08:00:27:64:5D:B4 enp0s3	192.168.10.100/24

Importamos el archivo .ova facilitado en el enunciado de esta misma práctica y comprobamos su configuración.

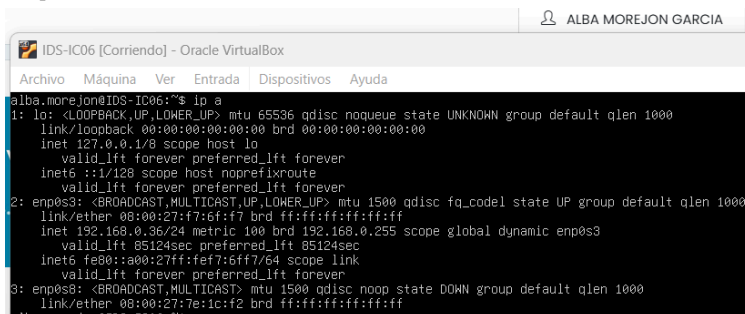




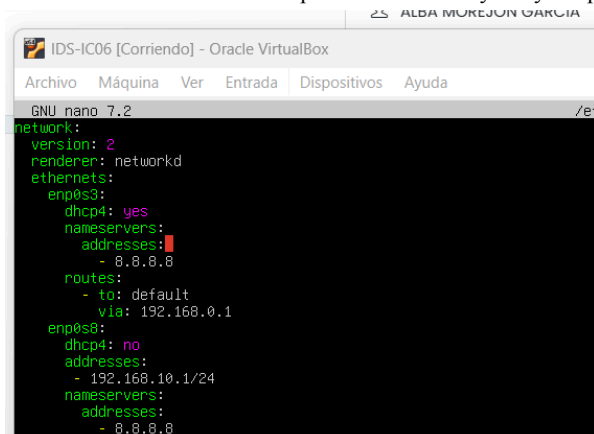
## Configuración red WebServer



## Maquina IDS:



modificamos el fichero /etc/netplan/50-cloud-init.yaml y lo aplicamos con `sudo netplan apply`



```

ALBA MOREJON GARCIA
IDS-IC06 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alba.morejon@IDS-IC06:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.36 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::a00:27ff:fe77:6ff7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:f7:6f:f7 txqueuelen 1000 (Ethernet)
    RX packets 2801 bytes 1675837 (1.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 406 bytes 30390 (30.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.1 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::a00:27ff:fe7e:1cf2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:7e:1c:f2 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19 bytes 1336 (1.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 107 bytes 9098 (9.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 107 bytes 9098 (9.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

### Hay conectividad entre las máquinas IDS y WebService

```

ALBA MOREJON GARCIA
WebServer-IC06 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
webuser@webserver:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=2.54 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=2.25 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=2.39 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=1.94 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=2.81 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=2.36 ms
64 bytes from 192.168.10.1: icmp_seq=7 ttl=64 time=1.65 ms
^C
--- 192.168.10.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6008ms
rtt min/avg/max/mdev = 1.653/2.277/2.812/0.354 ms

ALBA MOREJON GARCIA
IDS-IC06 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
alba.morejon@IDS-IC06:~$ ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=64 time=1.24 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=64 time=2.57 ms
64 bytes from 192.168.10.100: icmp_seq=4 ttl=64 time=2.88 ms
64 bytes from 192.168.10.100: icmp_seq=5 ttl=64 time=2.12 ms
64 bytes from 192.168.10.100: icmp_seq=6 ttl=64 time=1.39 ms
^C
--- 192.168.10.100 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5005ms
rtt min/avg/max/mdev = 1.241/1.924/2.882/0.639 ms

```

Vamos a hacer que ids enrute para que la máquina webserver tenga salida a internet.

En el fichero /etc/sysctl.conf descomentamos la línea siguiente y aplicamos la configuración con el comando `sudo sysctl -p` (activar el enrutamiento ip)



```

ALBA MOREJON GARCIA
IDS-IC06 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
GNU nano 7.2 /etc/sysctl.conf
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
# Uncomment the next line to enable packet forwarding for IPv6

```

A continuación en caso de no estar instalado con el comando `sudo apt-get iptables` instalamos la herramienta, configuraremos el nat para permitir el tráfico.

```

alba.morejon@IDS-IC06:~$ sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE
alba.morejon@IDS-IC06:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s3 -j ACCEPT
alba.morejon@IDS-IC06:~$ sudo iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT
alba.morejon@IDS-IC06:~$

```

Instalamos la herramienta para hacer fijas las reglas tras cada reinicio

```

alba.morejon@IDS-IC06:~$ sudo apt-get install iptables-persistent
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
iptables-persistent is already the newest version (1.0.20).
0 upgraded, 0 newly installed, 0 to remove and 28 not upgraded.

```

```

alba.morejon@IDS-IC06:~$ sudo netfilter-persistent save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables save
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables save
alba.morejon@IDS-IC06:~$ sudo netfilter-persistent reload
run-parts: executing /usr/share/netfilter-persistent/plugins.d/15-ip4tables start
run-parts: executing /usr/share/netfilter-persistent/plugins.d/25-ip6tables start
alba.morejon@IDS-IC06:~$

```

Mostramos reglas creadas

```

alba.morejon@IDS-IC06:~$ sudo iptables -v -n
Chain INPUT (policy ACCEPT 4 packets, 144 bytes)
pkts bytes target prot opt in out source destination
Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
1 76 ACCEPT 0 -- enp0s8 enp0s3 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
1 76 ACCEPT 0 -- enp0s3 enp0s8 0.0.0.0/0 0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
alba.morejon@IDS-IC06:~$
alba.morejon@IDS-IC06:~$ sudo iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 1 packets, 76 bytes)
pkts bytes target prot opt in out source destination
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
1 76 MASQUERADE 0 -- * * enp0s3 0.0.0.0/0 0.0.0.0/0
alba.morejon@IDS-IC06:~$

```

Ambas máquinas tienen visibilidad y tienen salida a internet

```

WebServer-IC06:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.98 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=2.16 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=1.98 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=1.18 ms
^C
--- 192.168.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.179/1.673/2.156/0.404 ms
WebServer-IC06:~$
WebServer-IC06:~$ ping google.com
PING google.com (142.250.201.78) 56(84) bytes of data.
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=1 ttl=117 time=18.5 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=2 ttl=117 time=122 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=3 ttl=117 time=18.3 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=4 ttl=117 time=75.9 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.162/1.487/1.769/0.283 ms

IDS-IC06:~$ ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=64 time=1.77 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=64 time=1.77 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=64 time=1.16 ms
64 bytes from 192.168.10.100: icmp_seq=4 ttl=64 time=1.25 ms
^C
--- 192.168.10.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.162/1.487/1.769/0.283 ms
IDS-IC06:~$
IDS-IC06:~$ ping google.com
PING google.com (142.250.201.78) 56(84) bytes of data.
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=1 ttl=118 time=70.2 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=2 ttl=118 time=13.3 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=3 ttl=118 time=70.0 ms
64 bytes from mad07s25-in-f14.1e100.net (142.250.201.78): icmp_seq=4 ttl=118 time=13.3 ms
^C
--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 1.162/1.487/1.769/0.283 ms

```

Máquina SOC

```

alba.morejon@SOCIC06:~$ cat /etc/netplan/50-cloud-init.yaml
network:
  version: 2
  ethernet:
  - enp0s3:
      dhcp: true
      routes:
        - to: 192.168.10.0/24
          via: 192.168.0.36
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4

```

```

alba.morejon@SOCIC06:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.0.41 netmask 255.255.255.0 broadcast 192.168.0.255
inet6 fe80::a00:27ff:fe4b:b1f1 prefixlen 64 scopeid 0x20<link>
ether 08:00:27:4b:b1:f1 txqueuelen 1000 (Ethernet)
RX packets 3310 bytes 2022571 (2.0 MB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 1077 bytes 119518 (119.5 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 444 bytes 42224 (42.2 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 444 bytes 42224 (42.2 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Comprobamos que tenga conexión con las otras máquinas

```

alba.morejon@SOCIC06:~$ ping 192.168.0.41
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=64 time=0.633 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=64 time=0.027 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=64 time=0.042 ms
^C
--- 192.168.0.41 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2061ms
rtt min/avg/max/mdev = 0.027/0.234/0.633/0.282 ms
alba.morejon@SOCIC06:~$ ping 192.168.10.100
PING 192.168.10.100 (192.168.10.100) 56(84) bytes of data.
64 bytes from 192.168.10.100: icmp_seq=1 ttl=63 time=7.17 ms
64 bytes from 192.168.10.100: icmp_seq=2 ttl=63 time=3.35 ms
64 bytes from 192.168.10.100: icmp_seq=3 ttl=63 time=4.33 ms
^C
--- 192.168.10.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 3.348/4.950/7.174/1.622 ms

```

## Apartado 2: Configuración del IDS para registrar el tráfico de red. (SNORT)

- Instalar y configurar SNORT en el IDS para poder escuchar y guardar todo el tráfico de la red DMZ.

```

alba.morejon@IDS-IC06:~$ sudo apt-get install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauthen-sasl-perl libclone-perl libdaq2t64 libdata-dump-perl libdumbnet1 libencode-
  libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-
  libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl lib
  liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl lib
  libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-opens
Suggested packages:
  libdigest-hmac-perl libgssapi-perl libio-compress-brotli-perl libcrypt-ssleay-perl lib
  libauthen-ntlm-perl debhelper snort-doc
The following NEW packages will be installed:
  libauthen-sasl-perl libclone-perl libdaq2t64 libdata-dump-perl libdumbnet1 libencode-
  libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libhttp-
  libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl lib
  liblwp-protocol-https-perl libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl lib
  libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl oinkmaster perl-opens
snort-rules-default
0 upgraded, 41 newly installed, 0 to remove and 28 not upgraded.

```

```

Please use the CIDR form - for example, 192.168.1.0/24 for a block of 256 addresses or 192.168.1.0/24,192.168.2.0/24 for two blocks of 256 addresses.

You can leave this value empty and configure HOME_NET in /etc/snort/snort.conf instead. This is 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 default.

Address range for the local network:
192.168.10.0/24

```

Establecemos las variables HOME\_NET y EXTERNAL\_NET

```

#####
# Step #0: (Debian specific) Create a configuration
# for a specific interface
#####
# If you want to run Snort in Debian using different
# instances each handling a different interface and
# a different configuration you can copy this file to
# /etc/snort/snort.<interface>.conf (where '<interface>' is the name of your
# network interface) and adjust the value there.
#
# The Debian init.d script is defined in such a way
# that you can run multiple instances.
#####
# Step #1: Set the network variables. For more information, see README.variables
#####
# Setup the network addresses you are protecting
#
# Note to Debian users: this value is overridden when starting
# up the Snort daemon through the init.d script by the
# value of DEBIAN_SNORT_HOME_NET as defined in the
# /etc/snort/snort.debian.conf configuration file
#
ipvar HOME_NET 192.168.10.0/24

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET 192.168.10.0/24
# If HOME_NET is defined as something other than "any", alternative, you can
# use this definition if you do not want to detect attacks from your internal

```

- **Configurar las reglas de detección de Snort, cada una de ellas debe recoger un mensaje indicando el tipo de conexión que se establece y un identificador único. Las alertas a generar son:**
  - **Ping (Request) desde la red interna (DMZ) hacia el exterior. Se debe registrar únicamente el Request de la interna y no la respuesta de la externa.**

```

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.

#ping request de la DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; icmp_type:8;)

```

- **Ping (Request) desde el exterior hacia la DMZ. Se debe registrar únicamente el Request de la externa y no la respuesta de la DMZ.**

```

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.

#ping desde DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; icmp_type:8;)

#ping desde el exterior a DMZ
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Solicitud ping desde el exterior a la DMZ"; sid:1000002; rev:1; icmp_type:8;)

```

- **Intentos de las conexiones SSH hacia WebServer. Solamente registra el primer paquete de sincronización de este intento de conexión.**

```

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.

#ping desde DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; icmp_type:8;)

#ping desde el exterior a DMZ
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Solicitud ping desde el exterior a la DMZ"; sid:1000002; rev:1; icmp_type:8;)

#Conexiones SSH al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Intento conexion SSH al WebServer"; sid:1000003; rev:1; flags:S;)

```

- **Intentos de las conexiones HTTP hacia WebServer. Solamente registra el primer paquete de sincronización de este intento de conexión.**

```

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.

#ping desde DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; icmp_type:8;)

#ping desde el exterior a DMZ
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Solicitud ping desde el exterior a la DMZ"; sid:1000002; rev:1; icmp_type:8;)

#Conexiones SSH al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Intento conexion SSH al WebServer"; sid:1000003; rev:1; flags:S;)

#Conexiones HTTP al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Intento de conexion HTTP al WebServer"; sid:1000004; rev:1; flags:S;)

```

- **Intentos de las conexiones a phpMyAdmin hacia WebServer. La ruta hacia la base de datos es <http://ip-de-WebServer/phpmyadmin>.**

```

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.

#ping desde DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; icmp_type:8;)

#ping desde el exterior a DMZ
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Solicitud ping desde el exterior a la DMZ"; sid:1000002; rev:1; icmp_type:8;)

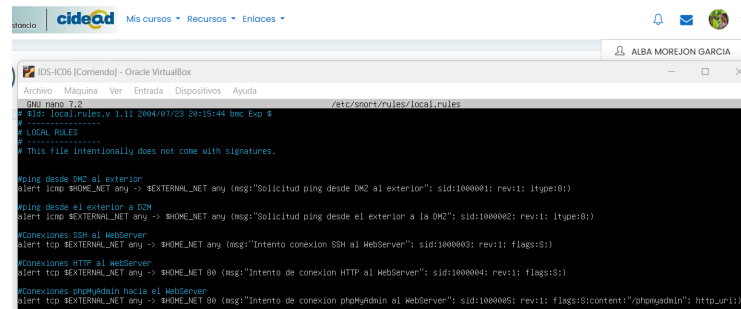
#Conexiones SSH al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Intento conexion SSH al WebServer"; sid:1000003; rev:1; flags:S;)

#Conexiones HTTP al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Intento de conexion HTTP al WebServer"; sid:1000004; rev:1; flags:S;)

#Conexiones phpMyAdmin hacia el WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Intento de conexion phpMyAdmin al WebServer"; sid:1000005; rev:1; flags:S;content:"/phpmyadmin"; http_uri:);

```

Tuvimos que cambiar el icmp\_type de las dos primeras líneas por itype



```

/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.

#ping desde DMZ al exterior
alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"Solicitud ping desde DMZ al exterior"; sid:1000001; rev:1; itype:8;)

#ping desde el exterior a DMZ
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"Solicitud ping desde el exterior a la DMZ"; sid:1000002; rev:1; itype:8;)

#Conexiones SSH al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"Intento conexión SSH al WebServer"; sid:1000003; rev:1; flags:S;)

#Conexiones HTTP al WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Intento de conexión HTTP al WebServer"; sid:1000004; rev:1; flags:S;)

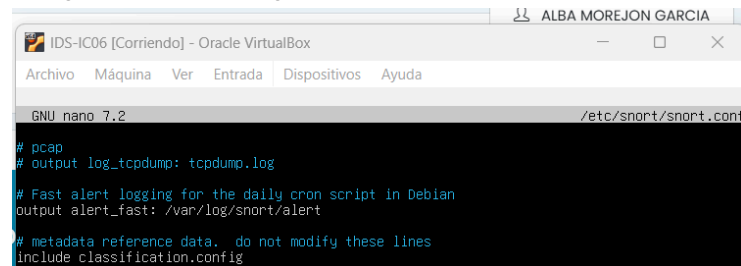
#Conexiones phpmyadmin hacia el WebServer
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"Intento de conexión phpmyadmin al WebServer"; sid:1000005; rev:1; flags:S;content:"/phpmyadmin/"; http:uri;)

```

Añadimos una línea como adicional para evitar en tráfico dhcp

```
#Ignorar tráfico DHCP
pass udp any 68 -> any 67 (msg:"Ignorar tráfico dhcp"; sid:1000006; rev:1;)
```

Configuramos la ruta del log



```

GNU nano 7.2 /etc/snort/snort.conf

# pcap
# output_log_tcpdump: tcpdump.log

# Fast alert logging for the daily cron script in Debian
output alert_fast: /var/log/snort/alert

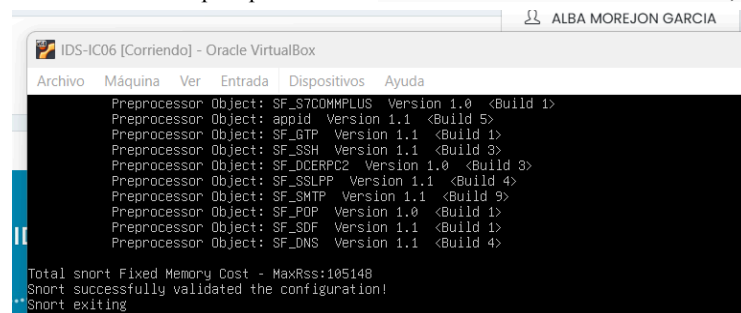
# metadata reference data. do not modify these lines
include classification.config

```

### Apartado 3: Pruebas de las alertas generadas.

- Realizar las pruebas pertinentes donde se demuestren las diferentes alertas generadas en el apartado 2.
- Para cada una de estas alertas se debe recoger pantallazo con las acciones realizadas y un volcado final del archivo de log resultante tras todas las pruebas.

Al usar el comando para probar snort: `sudo snort -T -c /etc/snort/snort.conf`, se consigue un resultado favorable:



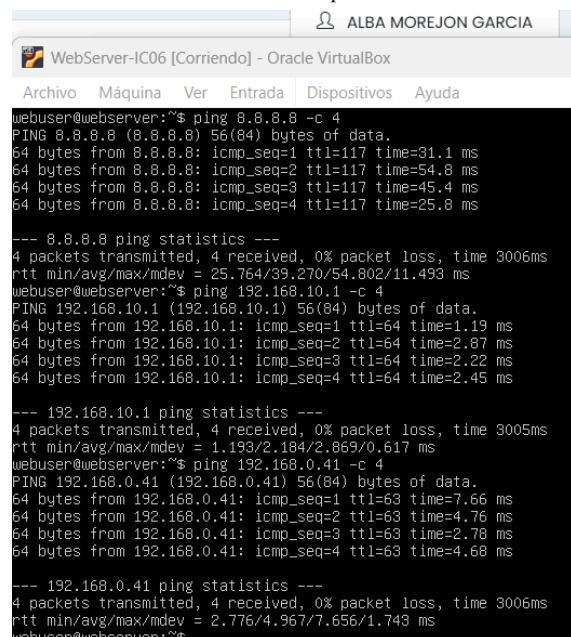
```

IDS-IC06 [Corriendo] - Oracle VirtualBox
Preprocessor Object: SF_S7COMPLUS Version 1.0 <Build 1>
Preprocessor Object: appid Version 1.1 <Build 5>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DCEPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_SSLFP Version 1.1 <Build 4>
Preprocessor Object: SF_SFTP Version 1.1 <Build 9>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>

Total snort Fixed Memory Cost - MaxRss:105148
Snort successfully validated the configuration!
**Snort exiting

```

Tenian conectividad entre las máquinas



```

WebServer-IC06 [Corriendo] - Oracle VirtualBox
webuser@webserver:~$ ping 8.8.8.8 -c 4
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=31.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=54.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=45.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=25.8 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 25.764/39.270/54.802/11.493 ms
webuser@webserver:~$ ping 192.168.10.1 -c 4
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=2.87 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=2.22 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=2.45 ms

--- 192.168.10.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.193/2.184/2.869/0.617 ms
webuser@webserver:~$ ping 192.168.0.41 -c 4
PING 192.168.0.41 (192.168.0.41) 56(84) bytes of data.
64 bytes from 192.168.0.41: icmp_seq=1 ttl=63 time=7.66 ms
64 bytes from 192.168.0.41: icmp_seq=2 ttl=63 time=4.76 ms
64 bytes from 192.168.0.41: icmp_seq=3 ttl=63 time=2.78 ms
64 bytes from 192.168.0.41: icmp_seq=4 ttl=63 time=4.68 ms

--- 192.168.0.41 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3006ms
rtt min/avg/max/mdev = 2.776/4.967/7.656/1.743 ms
webuser@webserver:~$

```

## Primera prueba, trafico fuera de la red

```

webuser@webserver:~$ ping 8.8.8.8 -c 4
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=117 time=81.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=17.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=17.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=14.3 ms

--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 14.326/32.858/81.431/28.080 ms

```

## Mostramos el log

```

aliba.morejon@IDS-IC06:~$ sudo cat /var/log/snort/alert
[1] Stopped
aliba.morejon@IDS-IC06:~$ sudo snort -c /etc/snort/snort.conf -i enp0s8
03/10-22:15:29.998644 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} ::
03/10-22:15:29.998644 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} ::
03/10-22:15:30.000426 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2] {IPV6-ICMP} ::
03/10-22:41:52.912015 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:53.913778 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:54.915050 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:55.917539 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
aliba.morejon@IDS-IC06:~$

```

## Conexión desde http a WebServer

WEB SERVER - SECURITY SYSTEM

Ubuntu

Please, be good.  
Don't hack me!

This is the default welcome page used to test the correct operation of the Apache2 server after installation. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you read this page, it means that the Apache HTTP server installed at this site is working properly. You should see the file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means it is the first time you have visited this site. It is the default page of the Apache HTTP server installed at this site. It is the default page of the Apache HTTP server installed at this site. It is the default page of the Apache HTTP server installed at this site.

## Mostramos el log

```

aliba.morejon@IDS-IC06:~$ sudo cat /var/log/snort/alert
[1] Stopped
aliba.morejon@IDS-IC06:~$ sudo snort -c /etc/snort/snort.conf -i enp0s8
03/10-22:15:29.998644 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2]
03/10-22:15:29.998644 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2]
03/10-22:15:30.000426 00000000: [1:527:8] BAD-TRAFFIC same SRC/DST 00000000: [Classification: Potentially Bad Traffic] [Priority: 2]
03/10-22:41:52.912015 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:53.913778 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:54.915050 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:41:55.917539 00000001: [1:1000001:1] Sollicitud ping desde DM2 al exterior 00000000: [Priority: 0] {ICMP} 192.168.10.100 -> 8.8.8.8
03/10-22:47:59.849996 00000004: [1:1000004:1] Intento de conexion HTTP al WebServer 00000000: [Priority: 0] {TCP} 192.168.0.41:60508 -> 192.168.10.100:80
03/10-22:47:59.849996 00000003: [1:1000003:1] Intento conexion SSH al WebServer 00000000: [Priority: 0] {TCP} 192.168.0.41:60508 -> 192.168.10.100:22
aliba.morejon@IDS-IC06:~$

```