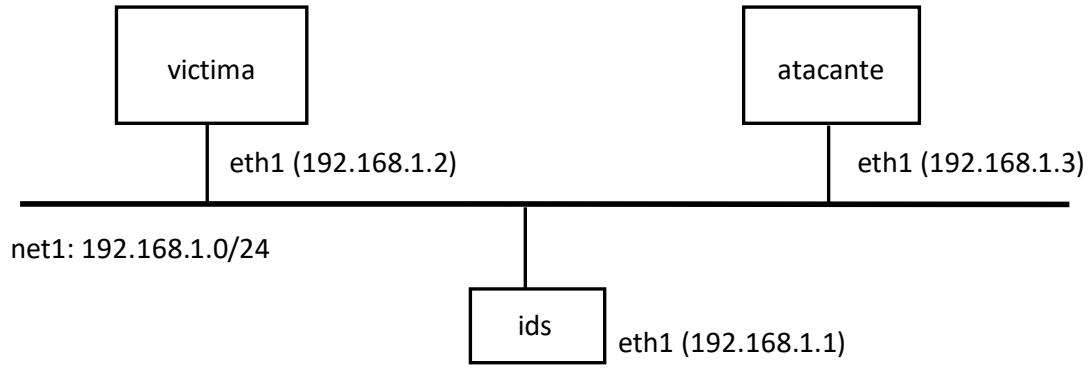
Seguridad en Redes

Práctica 3.4. Snort

Preparación del entorno

Vamos a usar 3 MVs (`ids`, `victima` y `atacante`) conectadas a una red interna (`net1`):



Importa una MV, haz dos clonaciones enlazadas y añade un interfaz de red a cada MV.

Para simular que `ids` se conecta a un puerto espejo (*mirror* o SPAN), en la configuración del interfaz añadido selecciona la opción *"Permitir MVs"* en el parámetro *"Modo promiscuo"*.

Configura `ids`:

```
sudo ifconfig eth1 192.168.1.1/24 up
sudo apt-get update
sudo apt-get install snort
sudo dpkg-reconfigure snort
```

En el último paso, indica que se inicie en el arranque, que escuche por `eth1`, que la red local sea 192.168.1.0/24, que no se deshabilite el modo promiscuo, ninguna opción adicional y que no se envíen resúmenes por correo electrónico.

Configura `victima`:

```
sudo ifconfig eth1 192.168.1.2/24 up
```

Configura `atacante`:

```
sudo ifconfig eth1 192.168.1.3/24 up
sudo apt-get update
sudo apt-get install nmap dsniff
```

Desconecta el adaptador 1 (`eth0`) de todas las MVs.

Documentación y configuración

Consulta la página de manual de Snort y su documentación (http://manual.snort.org).

Revisa la configuración (`/etc/snort/snort.conf`, `/etc/snort/snort.debian.conf` y `/etc/snort/rules/`).

Sniffer y *packet logger*

Inicia Snort en modo *sniffer*, para imprimir los encabezados TCP/IP en pantalla:

```
$ sudo snort -v -i eth1
```

Genera tráfico de red (por ejemplo, haciendo un `ping` de `atacante` a `victima`) y comprueba que Snort lo detecta.

Prueba otras opciones del modo *sniffer* (`-d`, `-e`...) y averigua cómo iniciar Snort en modo *packet logger*, lo cual puede ser muy útil para realizar un análisis forense tras haberse detectado un ataque.

- **Entrega:** Explica para qué sirven las distintas opciones del modo *sniffer*. Explica también cómo se puede iniciar Snort en modo *packet logger.*

NIDS basado en reglas

Al instalar el paquete, se inicia Snort en modo NIDS. Compruébalo:

```
$ sudo service snort status
Status of snort daemon(s):  eth1  OK.
```

Si no, se arrancaría con:

```
$ sudo service snort start
```

Al hacer *ping* a `victima` desde `atacante`, en `/var/log/snort/alert` debe aparecer algo como:

```
[**] [1:366:7] ICMP PING *NIX [**]
[Classification: Misc activity] [Priority: 3]
01/19-18:46:54.159821 192.168.1.3 -> 192.168.1.2
ICMP TTL:64 TOS:0x0 ID:469 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:2352   Seq:1  ECHO

[**] [1:384:5] ICMP PING [**]
[Classification: Misc activity] [Priority: 3]
01/19-18:46:54.159821 192.168.1.3 -> 192.168.1.2
ICMP TTL:64 TOS:0x0 ID:469 IpLen:20 DgmLen:84 DF
```

```
Type:8  Code:0  ID:2352   Seq:1  ECHO

[**] [1:408:5] ICMP Echo Reply [**]
[Classification: Misc activity] [Priority: 3]
01/19-18:46:54.159834 192.168.1.2 -> 192.168.1.3
ICMP TTL:64 TOS:0x0 ID:39046 IpLen:20 DgmLen:84
Type:0  Code:0  ID:2352  Seq:1  ECHO REPLY
```

Cada alerta se identifica con tres números entre corchetes:

- El primer número es el *Generator ID* (GID), que indica qué componente de Snort generó la alerta. El fichero `gen-msg.map` contiene una lista de GIDs. En este caso, el GID igual a 1 indica que la alerta corresponde a *"snort general alert"*.
- El segundo número es el *Snort ID* o *Signature ID* (SID). Los SID de cada regla se indican directamente en la regla mediante la opción `sid`. Los SIDs de los pre-procesadores se indican el fichero `gen-msg.map`.
- El tercer número es el número de revisión. Cada nueva versión de una regla debe incrementar este número con la opción `rev`.

Consulta algunas de las reglas predefinidas en `/etc/snort/rules`. Como verás, la mayoría de ellas detectan ataques que explotan vulnerabilidades del sistema o del *software*.

Busca las reglas que generaron las alertas anteriores.

- **Entrega:** Copia las reglas que generaron las alertas anteriores e indica en qué archivo las has encontrado.

Definición de nuevas reglas

Añade la siguiente regla a `/etc/snort/rules/local.rules`:

```
alert icmp any any -> any any  (msg:"Special ping"; ttl:100; dsize:200; sid:10000001; rev:1;)
```

Reinicia Snort:

```
$ sudo service snort restart
```

Intenta activar la alerta.

- **Entrega:** Copia la alerta generada.

Preprocesadores

Para generar automáticamente alertas asociadas a los eventos generados por los preprocesadores, añade la siguiente línea al fichero `/etc/snort/snort.conf`:

```
config autogenerate_preprocessor_decoder_rules
```

El preprocesador `sfportscan` (GID=122) detecta ataques de exploración de puertos. Descomenta la línea de ejemplo en `/etc/snort/snort.conf` y recarga la configuración.

Comprueba si se detecta este tipo de ataques ejecutando en `atacante`:

```
$ sudo nmap 192.168.1.2
```

En `/var/log/snort/alert` debe aparecer algo como:

```
[**] [122:1:0] (portscan) TCP Portscan [**]
[Priority: 3]
01/21-18:29:01.916797 192.168.1.1 -> 192.168.1.2
PROTO:255 TTL:57 TOS:0x0 ID:24507 IpLen:20 DgmLen:161
```

- **Entrega**: Copia las alertas generadas.

El preprocesador `arpspoof` (GID=112) detecta ataques de ARP *spoofing*. Descomenta las líneas de ejemplo en `/etc/snort/snort.conf`, modifícalas para asociar IPs con MACs y recarga la configuración.

Comprueba si se detectan este tipo de ataques ejecutando lo siguiente en `atacante`:

```
$ sudo arpspoof -i eth1 -r -t 192.168.1.1 192.168.1.2
```

En `/var/log/snort/alert` debe aparecer algo como:

```
[**] [112:4:1] (spp_arpspoof) Attempted ARP cache overwrite
attack [**]
01/21-18:07:17.157078
```

- **Entrega**: Copia las alertas generadas.