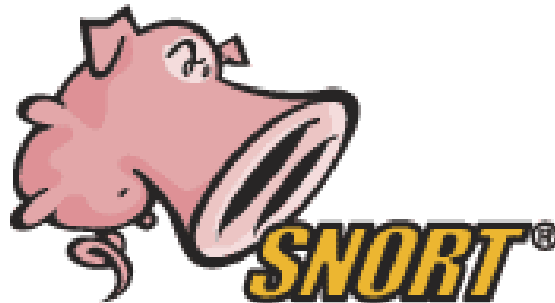


Configuración de un IDS (Intrusion detection system)

Práctica 4 - SNORT

[Aarón Escolano Candela](#)



Introducción	2
Objetivo de la práctica	2
Enunciado	3
Instalación y configuración	4
Configuración Servidor privado	4
Configuración en AWS	4
Configuración en la máquina	4
Rsyslog	4
Configuración Proxy/IDS	4
Configuración en AWS	5
Configuración en la máquina	5
Activación del bit de FORWARD	5
Configuración NAT	6
Instalación SNORT	6
Configuración SNORT	7
Reglas añadidas	8
Rsyslog	9
Puntos a realizar	10
1.- El IDS tiene que almacenar los generados en el equipo cliente/servidor a través de su servicio SYSLOG.	10
2.- Configurar una regla para detectar los paquetes icmp de salida y mostrar el mensaje. "Ping detectado". Tráfico desde nodo a internet.	10
3.- Configurar una regla que identifique cuándo se ha accedido a la web del marca.com, debe mostrar el mensaje "Acceso a marca.com". Tráfico desde nodo a internet	11
4.- Configura un servidor en el equipo cliente/servidor (nc -l 1000) y publica el puerto 1000 a través de iptables. Arranca snort como sniffer "snort -vde" y haz una petición a la ip NAT del router desde tu host físico ¿qué mensaje obtienes en relación con esa petición? ¿por qué? ¿Qué tendrías que configurar en snort para poder crear una alerta en referencia a esta petición?	11
5.- Desde nmap realiza un escaneo de puertos desde tu equipo al equipo Proxy. Mientras lanzas el escaneo lanza snort en modo sniffer. ¿Qué salida obtienes de snort? ¿Por qué? Ahora configura snort para detectar los escaneos de puertos y que muestre una alerta en caso de detección de este tipo de amenazas.	12
Referencias	13

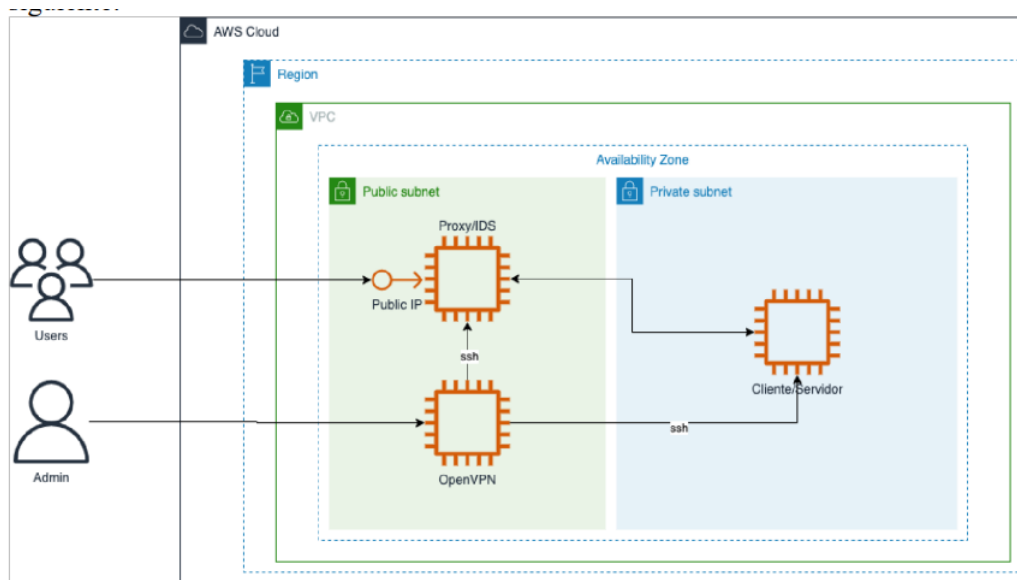
Introducción

Objetivo de la práctica

En la práctica 1 se trabajó con cierta profundidad la principal herramienta de la que disponen los administradores de redes a la hora de implantar seguridad perimetral: los cortafuegos. En esta práctica se estudia la herramienta principal, además de una buena configuración de los servicios, con la que los administradores pueden llevar a cabo seguridad interna: los Sistemas de Detección de Intrusos.

Al finalizar la sesión dispondremos del conocimiento necesario para configurar adecuadamente el sistema de detección de intrusos más empleado en el mundo: snort. La práctica que nos ocupa se dedica al análisis del archivo de configuración de snort, denominado snort.conf.

Enunciado



Después de haber desplegado la plataforma se debe configurar Snort en el equipo IDS. Este equipo realizará las funciones de NAT transversal para permitir el tráfico desde el interior al exterior, y a su vez realizará las labores de NATP para publicar un servicio en el equipo cliente/servidor que se explicará en el resto del enunciado. El otro elemento existente será la OpenVPN a la cual nos conectaremos para la administración de los elementos. El tráfico SSH está prohibido en el servidor de proxy y en cliente/servidor directamente desde internet a ningún equipo de la plataforma. El equipo cliente/servidor sólo puede tener direccionamiento privado, por lo que para administrar el sistema será necesario cerrar el túnel OpenVPN y desde ahí se podrá conectar a todos los equipos. Finalmente, el equipo cliente/servidor será desde donde se generará el tráfico y se recibirá desde el exterior según las peticiones que se van a realizar a continuación.

- 1.- El IDS tiene que almacenar los logs generados en el equipo cliente/servidor a través de su servicio SYSLOG.
- 2.- Configurar una regla para detectar los paquetes icmp de salida y mostrar el mensaje. "Ping detectado". Tráfico desde nodo a internet.
- 3.- Configurar una regla que identifique cuándo se ha accedido a la web del marca.com, debe mostrar el mensaje "Acceso a marca.com". Tráfico desde nodo a internet
- 4.- Configura un servidor en el equipo cliente/servidor (nc -l 1000) y publica el puerto 1000 a través de iptables. Arranca snort como sniffer "snort -vde" y haz una petición a la ip NAT del router desde tu host físico ¿qué mensaje obtienes en relación con esa petición? ¿por qué? ¿qué tendrías que configurar en snort para poder crear una alerta en referencia a esta petición?
- 5.- Desde mediante nmap realiza un escaneo de puertos desde tu equipo al equipo Proxy. Mientras lanzas el escaneo lanza snort en modo sniffer. ¿Qué salida obtienes de snort? ¿Por qué? Ahora configura snort para detectar los escaneos de puertos y que muestre una alerta en caso de detección de este tipo de amenazas

Instalación y configuración

Configuración Servidor privado

Configuración en AWS

Modificamos las rutas de la subred privada de la maquina privada y añadimos la regla de enviar todos los paquetes a la instancia IDS.

Destino	Destino	Estado
pl-63a5400a	vpce-0b68bc47d8e2ee072	✓ Activo
0.0.0.0/0	eni-03dd5ca19441ff980 🔗	✓ Activo
10.1.0.0/16	local	✓ Activo

Configuración en la máquina

Rsyslog

Modificamos el archivo /etc/rsyslog.conf y activamos el servidor syslog para recibir paquetes UDP y TCP por el puerto 514.

```
#####
#### MODULES ####
#####

module(load="imuxsock") # provides support for local system logging
module(load="immark") # provides --MARK-- message capability

# provides UDP syslog reception
module(load="imudp")
input(type="imudp" port="514")

# provides TCP syslog reception
module(load="imtcp")
input(type="imtcp" port="514")
```

Reiniciamos el servicio

```
sudo systemctl restart rsyslog
```

Configuración Proxy/IDS


La máquina IDS funcionará como un router NAT para dotar de conectividad a internet a la máquina en la subred privada. Por otro lado, instalaremos snort en esta máquina y enviaremos los logs mediante syslog al servidor privado.


Configuración en AWS


Lanzaremos una instancia ec2 con una Ubuntu AMI y la instalaremos en una subred pública. También le autoasignaremos una IP pública.

Una vez lanzada, deberemos desactivar el source/destination check para habilitar el reenvío de paquetes a la instancia privada.

Change Source / destination check ✕

The source / destination check ensures that the instance is the source or destination of all the traffic it sends and receives. Each EC2 instance performs source and destination checks by default. [Learn more](#) 

Instance ID
 [i-0959166cd2bdcf2c2](#) (IDS)

Network interface
 [eni-03dd5ca19441ff980](#)

Source / destination checking
Stop to allow your instance to send and receive traffic when the source or destination is not itself.
☒ Stop

Cancel Save

La tabla de rutas de la subred pública será esta:

Destination	Target
10.1.0.0/16	local
0.0.0.0/0	igw-038c7abf0aea33975

Configuración en la máquina

Activación del bit de FORWARD

Para que la máquina pueda reenviar paquetes que no van dirigidos a ella debemos activar el redireccionamiento de paquetes. Deberemos modificar el paquete `sysctl.conf` y añadir esta línea: `net.ipv4.ip_forward=1`

```
sudo nano /etc/sysctl.conf
```

Configuración NAT

Utilizaremos iptables. Primero instalamos iptables:

```
sudo apt install iptables
```

También instalaremos iptables-persistent para mantener las reglas entre reinicios de la máquina.

```
sudo apt install iptables-persistent
```

Nuestra máquina no tiene ip pública, por lo que deberemos hacer un SOURCE NAT (DINÁMICO).

SOURCE NAT

Esta regla permitirá la comunicación desde el servidor privado con internet. Cuando la máquina privada envíe un paquete a internet esta regla modificará la ip de origen y seleccionará la ip pública que tenga el proxy en ese momento. El parámetro -j masquerade es el que se encarga de hacer esta traducción.

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j masquerade
```

Si tuviéramos una ip pública estática podríamos hacer NAT ESTÁTICO. El comando sería así:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to <IP PÚBLICA>
```

Port Forwarding o Destination NAT (DNAT)

El port forwarding nos permitirá publicar un puerto del servidor privado y que sea accesible desde internet. Para esto, abriremos un puerto en el proxy y redirigiremos todas las conexiones que vengan a ese puerto a la máquina privada.

```
sudo iptables -t nat -A PREROUTING -p tcp --dport 1000 -j DNAT --to 10.1.132.97:1000
```

Este comando redirige todos los paquetes tcp(-p TCP) que lleguen al puerto 1000 (--dport 1000) en el proxy al servidor privado(--to 10.1.132.97:1000).

Ahora guardamos las reglas.

```
sudo iptables-save > /etc/iptables/rules.v4
```

Instalación SNORT

SNORT es un sistema de prevención/detección de intrusiones (IDS/IPS).

Tiene la capacidad de realizar análisis de tráfico en tiempo real y registro de paquetes en redes de Protocolo de Internet (IP). Snort realiza análisis de protocolo, búsqueda de contenido y coincidencia.

El programa también se puede utilizar para detectar sondeos o ataques, intentos de toma de huellas dactilares del sistema operativo, ataques de URL semánticos, desbordamientos de búfer, sondeos de bloqueo de mensajes del servidor y escaneos de puertos sigilosos.

Snort se puede configurar en tres modos principales:

1. sniffer. Detecta todos los paquetes y los muestra por pantalla
2. logger. Escribe los logs en disco.
3. IDS. Utiliza las reglas establecidas para detectar paquetes y tomar las acciones especificadas.

En esta práctica utilizaremos el modo sniffer y el modo IDS.

Configuración SNORT

El archivo de configuración se encuentra en la carpeta `/etc/snort`.

El archivo de configuración es `snort.conf`. Vamos a hablar de los cambios realizados para esta práctica.

1. He deshabilitado todas las reglas por defecto comentando todas las reglas activas.

```
#include $RULE_PATH/icmp-info.rules
#include $RULE_PATH/icmp.rules
#include $RULE_PATH/imap.rules
#include $RULE_PATH/indicator-compromise.rules
#include $RULE_PATH/indicator-obfuscation.rules
#include $RULE_PATH/indicator-shellcode.rules
#include $RULE_PATH/info.rules
#include $RULE_PATH/malware-backdoor.rules
#include $RULE_PATH/malware-cnc.rules
#include $RULE_PATH/malware-other.rules
#include $RULE_PATH/malware-tools.rules
#include $RULE_PATH/misc.rules
#include $RULE_PATH/multimedia.rules
#include $RULE_PATH/mysql.rules
#include $RULE_PATH/netbios.rules
#include $RULE_PATH/nntp.rules
#include $RULE_PATH/oracle.rules
#include $RULE_PATH/os-linux.rules
#include $RULE_PATH/os-other.rules
#include $RULE_PATH/os-solaris.rules
#include $RULE_PATH/os-windows.rules
#include $RULE_PATH/other-ids.rules
#include $RULE_PATH/p2p.rules
#include $RULE_PATH/phishing-spam.rules
#include $RULE_PATH/policy-multimedia.rules
#include $RULE_PATH/policy-other.rules
#include $RULE_PATH/policy.rules
#include $RULE_PATH/policy-social.rules
#include $RULE_PATH/policy-spam.rules
#include $RULE_PATH/pop2.rules
#include $RULE_PATH/pop3.rules
#include $RULE_PATH/protocol-finger.rules
#include $RULE_PATH/protocol-ftp.rules
```

2. He incluido las reglas con nombre local.rules

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

3. He cambiado la ruta de la carpeta de reglas.(Este paso no es necesario. Puedes añadir tu archivo de reglas a la carpeta por defecto de reglas.)

```
# Path to your rules files (this can be a relative path)
# Note for Windows users: You are advised to make this an absolute path,
# such as: c:\snort\rules
var RULE_PATH /etc/snort/aaronrules
```

4. He añadido el envío de mensajes por rsyslog.

```
# syslog
output alert_syslog: LOG_AUTH LOG_ALERT
```

Reglas añadidas

He creado la carpeta aaronrules en la ruta /etc/snort/. En la carpeta aaronrules he añadido el archivo local.rules con las siguientes rutas:

```
root@ip-10-1-15-11:/etc/snort/aaronrules# cat local.rules
alert icmp 10.1.128.0/20 any -> any any (msg: "ICMP PING Detectado"; sid:10000001;)
alert tcp 10.1.128.0/20 any -> any any (content:"marca.com";msg:"Acceso a marca.com"; sid:10000002;rev:1;)
alert tcp any any -> 10.1.128.0/20 1000 (msg:"Acceso a máquina privada"; sid:10000003;)
alert tcp any any -> $HOME_NET any (flags:S; msg:"TCP port scan detected"; threshold: type both, track by_src, count 5, seconds 10; sid:10000001; rev:1;)
```

- alert icmp 10.1.128.0/20 any -> any any (msg: "ICMP PING Detectado"; sid:10000001;)

detecta los pings desde la máquina privada hacia el exterior.

- alert tcp 10.1.128.0/20 any -> any any (content:"marca.com";msg:"Acceso a marca.com"; sid:10000002;rev:1;)

detecta los accesos al servidor marca.com desde la máquina privada.

- alert tcp any any -> 10.1.128.0/20 1000 (msg:"Acceso a máquina privada"; sid:10000003;)

detecta los accesos al puerto 1000 de la máquina privada con el port forwarding desde la máquina IDS.

- alert tcp any any -> \$HOME_NET any (flags:S; msg:"TCP port scan detected"; threshold: type both, track by_src, count 5, seconds 10; sid:10000001; rev:1;)

detecta los escaneos de puertos a la máquina IDS.

Rsyslog

Esta máquina debe enviar los logs hacia el servidor syslog en la máquina privada. Para conseguir esto debemos modificar el archivo /etc/rsyslog.conf y añadir la siguiente línea al final del fichero:

```
*.*@10.1.132.97:514
```

En el fichero /etc/snort/snort.conf nos iremos al apartado 6 y descomentamos la siguiente línea:

```
#####
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
#####

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
output unified2: filename snort.log, limit 128, nostamp, mpls_event_types, vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
output alert syslog: LOG AUTH LOG ALERT
```

Puntos a realizar

1.- El IDS tiene que almacenar los generados en el equipo cliente/servidor a través de su servicio SYSLOG.

1. (IDS) sudo snort -c /etc/snort/snort.conf -s -A console

2. (Private computer) `tail -f /var/log/auth.log`
3. (Otra máquina desde internet) `sudo nmap <IP Pública IDS> -F`

```
May 2 15:57:49 ip-10-1-15-11 snort[559]: [1:1000001:1] ICMP port scan detected [ICMP] 77.224.188.168:53582 -> 10.1.15.11:135
May 2 15:58:32 ip-10-1-15-11 snort[559]: [1:1000001:1] TCP port scan detected [TCP] 77.224.188.168:54618 -> 10.1.15.11:139
May 2 15:58:32 ip-10-1-15-11 snort[559]: [1:1000001:1] TCP port scan detected [TCP] 77.224.188.168:54618 -> 10.1.15.11:139
May 2 15:59:03 ip-10-1-132-97 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/systemctl status rsyslog
May 2 15:59:03 ip-10-1-132-97 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=1000)
May 2 15:59:03 ip-10-1-132-97 sudo: pam_unix(sudo:session): session closed for user root
May 2 15:59:21 ip-10-1-132-97 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/nano /etc/rsyslog.d/50-default.conf
May 2 15:59:21 ip-10-1-132-97 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=1000)
May 2 15:59:42 ip-10-1-132-97 sudo: pam_unix(sudo:session): session closed for user root
May 2 15:59:51 ip-10-1-132-97 sudo: ubuntu : TTY=pts/0 ; PWD=/var/log ; USER=root ; COMMAND=/usr/bin/systemctl restart rsyslog
May 2 15:59:51 ip-10-1-132-97 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=1000)
May 2 15:59:51 ip-10-1-132-97 sudo: pam_unix(sudo:session): session closed for user root
May 2 15:59:57 ip-10-1-15-11 snort[559]: [1:1000001:1] TCP port scan detected [TCP] 77.224.188.168:36670 -> 10.1.15.11:587
May 2 15:59:57 ip-10-1-15-11 snort[559]: [1:1000001:1] TCP port scan detected [TCP] 77.224.188.168:36670 -> 10.1.15.11:587
May 2 16:00:50 ip-10-1-15-11 sudo: pam_unix(sudo:session): session closed for user root
May 2 16:02:21 ip-10-1-15-11 sudo: ubuntu : TTY=pts/0 ; PWD=/etc/rsyslog.d ; USER=root ; COMMAND=/usr/bin/su
May 2 16:02:21 ip-10-1-15-11 sudo: pam_unix(sudo:session): session opened for user root(uid=0) by ubuntu(uid=1000)
May 2 16:02:21 ip-10-1-15-11 su: (to root) root on pts/1
May 2 16:02:21 ip-10-1-15-11 su: pam_unix(su:session): session opened for user root(uid=0) by ubuntu(uid=0)
```

2.- Configurar una regla para detectar los paquetes icmp de salida y mostrar el mensaje. “Ping detectado”. Tráfico desde el nodo a internet.

Pasos:

Activamos el snort en modo IDS.

1. (IDS) `sudo snort -c /etc/snort/snort.conf -A console -q`

El parámetro `-c` especifica el archivo de configuración a utilizar. `-A` especifica el output Mode, en nuestro caso `-A console` envía “fast-style” alerts to the console (screen). `-q` es el modo quiet (omite imprimir por pantalla el inicio de snort)

2. (SERVIDOR PRIVADO) `ping google.com -c 1`

```
ubuntu@ip-10-1-132-97:~$ ping google.com -c 1
PING google.com (172.253.63.101) 56(84) bytes of data:
64 bytes from bi-in-f101.1e100.net (172.253.63.101): icmp_seq=1 ttl=95 time=2.63 ms

--- google.com ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 2.634/2.634/2.634/0.000 ms
ubuntu@ip-10-1-132-97:~$
```

3. Vemos la salida por pantalla en el IDS

```
ubuntu@ip-10-1-15-11:~$ sudo snort -c /etc/snort/snort.conf -A console -q
05/01-15:59:36.830968  [**] [1:1000001:0] ICMP PING Detectado [**] [Priority: 0] {ICMP} 10.1.132.97 -> 172.253.63.138
```

3.- Configurar una regla que identifique cuándo se ha accedido a la web del marca.com, debe mostrar el mensaje “Acceso a marca.com”. Tráfico desde nodo a internet

1. (IDS) `sudo snort -c /etc/snort/snort.conf -A console -q`
2. (SERVIDOR PRIVADO) `lynx marca.com`

```
ubuntu@ip-10-1-15-11: ~
ubuntu@ip-10-1-132-97: ~
MARCA - Diario online líder en información deportiva (p1 of 44)
#alternate alternate alternate alternate RSS Portada - Marca.com RSS Portada - Marca.com
REFRESH(900 sec): https://www.marca.com/
IFRAME:
https://5214106.fl.s.doubleclick.net/activityi;src=5214106;type=corp;cat=regis00;dc_lat=;dc_rdid=;tag_for_child_directed
_treatment=;ord=1?

* Es noticia:
* Valladolid - Atletico
* Cronica GP Azerbaiyan
* Carrera F1 GP Azerbaiyan
* Espanyol - Getafe
* Valladolid - Atletico donde ver
* Carrera MotoGP Jerez
* Alcaraz - Dimitrov
* Sprint MotoGP Jerez
* Alonso Sainz
* Alcaraz - Dimitrov TV
* GP Azerbaiyan F1 fechas
* Movistar 2024
* Horario Mutua Madrid
* Partidos Mutua hoy
* Joana Sanz
* Tu Cara Me Suenas
* Frases Trabajo

-- press space for next page --
Arrow keys: Up and Down to move. Right to follow a link; Left to go back.
H)elp O)ptions P)rint G)o M)ain screen Q)uit /=search [delete]=history list
```

3. Salida por pantalla en el IDS

```
05/01-16:01:51.019739  [**] [1:10000002:1] Acceso a marca.com [**] [Priority: 0] {TCP} 10.1.132.97:33988 -> 34.147.120.111:80
05/01-16:01:53.214922  [**] [1:10000002:1] Acceso a marca.com [**] [Priority: 0] {TCP} 10.1.132.97:46160 -> 34.147.120.111:443
05/01-16:01:55.496034  [**] [1:10000002:1] Acceso a marca.com [**] [Priority: 0] {TCP} 10.1.132.97:33424 -> 146.75.33.50:443
```

4.- Configura un servidor en el equipo cliente/servidor (nc -l 1000) y publica el puerto 1000 a través de iptables. Arranca snort como sniffer “snort -vde” y haz una petición a la ip NAT del router desde tu host físico ¿qué mensaje obtienes en relación con esa petición? ¿por qué? ¿Qué tendrías que configurar en snort para poder crear una alerta en referencia a esta petición?

Modo sniffer

```
(snort_decoder) WARNING: IP dgm len > captured len
WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
05/01-16:18:27.163467 0E:D6:E2:5E:E4:A9 -> 0E:E0:05:9C:2D:0F type:0x800 len:0x42
77.224.188.168:58293 -> 10.1.15.11:1000 TCP TTL:103 TOS:0x0 ID:28717 IpLen:20 DgmLen:52 DF
*****S* Seq: 0xD92EC084 Ack: 0x0 Win: 0xFAF0 TcpLen: 32
TCP Options (6) => MSS: 1460 NOP WS: 8 NOP NOP SackOK

=====
WARNING: No preprocessors configured for policy 0.
05/01-16:18:27.163503 0E:E0:05:9C:2D:0F -> 0E:D6:E2:5E:E4:A9 type:0x800 len:0x42
10.1.15.11:58293 -> 10.1.132.97:1000 TCP TTL:102 TOS:0x0 ID:28717 IpLen:20 DgmLen:52 DF
*****S* Seq: 0xD92EC084 Ack: 0x0 Win: 0xFAF0 TcpLen: 32
TCP Options (6) => MSS: 1460 NOP WS: 8 NOP NOP SackOK

=====
WARNING: No preprocessors configured for policy 0.
05/01-16:18:27.164027 0E:D6:E2:5E:E4:A9 -> 0E:E0:05:9C:2D:0F type:0x800 len:0x36
10.1.132.97:1000 -> 10.1.15.11:58293 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0 Ack: 0xD92EC085 Win: 0x0 TcpLen: 20

=====
WARNING: No preprocessors configured for policy 0.
05/01-16:18:27.164041 0E:E0:05:9C:2D:0F -> 0E:D6:E2:5E:E4:A9 type:0x800 len:0x36
10.1.15.11:1000 -> 77.224.188.168:58293 TCP TTL:63 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF
***A*R** Seq: 0x0 Ack: 0xD92EC085 Win: 0x0 TcpLen: 20
```

Comprobación de que el IDS detecta las peticiones al servidor privado.

05/01-16:07:16.623876	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58125	->	10.1.132.97:1000
05/01-16:07:16.877858	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58127	->	10.1.132.97:1000
05/01-16:07:17.270427	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58125	->	10.1.132.97:1000
05/01-16:07:17.488140	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58127	->	10.1.132.97:1000
05/01-16:07:17.890695	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58125	->	10.1.132.97:1000
05/01-16:07:18.097076	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58127	->	10.1.132.97:1000
05/01-16:07:18.509449	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58125	->	10.1.132.97:1000
05/01-16:07:18.710830	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58127	->	10.1.132.97:1000
05/01-16:07:19.134736	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58125	->	10.1.132.97:1000
05/01-16:07:19.320837	[[**]]	[1:10000003:0]	Acceso a máquina privada	[[**]]	[Priority: 0]	{TCP}	10.1.15.11:58127	->	10.1.132.97:1000

```
WARNING: No preprocessors configured for policy 0.
05/01-16:26:48.245979 0E:D6:E2:5E:E4:A9 -> 0E:E0:05:9C:2D:0F type:0x800 len:0x4A
77.224.188.168:37178 -> 10.1.15.11:1666 TCP TTL:40 TOS:0x0 ID:30113 IpLen:20 DgmLen:60 DF
*****S* Seq: 0x5DDDC80 Ack: 0x0 Win: 0xFAF0 TcpLen: 40
TCP Options (5) => MSS: 1460 SackOK TS: 813890472 0 NOP WS: 7 Scanner port 1666
```

```
WARNING: No preprocessors configured for policy 0.  
05/01-16:26:48.246014 0E:E0:05:9C:2D:0F -> 0E:D6:E2:5E:E4:A9 type:0x800 len:0x36  
10.1.15.11:1666 -> 77.224.188.168:37178 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF  
***A*R** Seq: 0x0 Ack: 0x5DDDC81 Win: 0x0 TcpLen: 20
```

```
WARNING: No preprocessors configured for policy 0.  
05/01-16:26:48.281726 0E:D6:E2:5E:E4:A9 -> 0E:E0:05:9C:2D:0F type:0x800 len:0x4A  
77.224.188.168:55338 -> 10.1.15.11:3826 TCP TTL:40 TOS:0x0 ID:4530 IpLen:20 DgmLen:60 DF  
*****S* Seq: 0x48033EBD Ack: 0x0 Win: 0xFAF0 TcpLen: 40  
TCP Options (5) => MSS: 1460 SackOK TS: 813890512 0 NOP WS: 7
```

[illegible]

```
WARNING: No preprocessors configured for policy 0.  
05/01-16:26:48.281755 0E:E0:05:9C:2D:0F -> 0E:D6:E2:5E:E4:A9 type:0x800 len:0x36  
10.1.15.11:3826 -> 77.224.188.168:55338 TCP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:40 DF  
***A*R** Seq: 0x0 Ack: 0x48033EBE Win: 0x0 TcpLen: 20
```

```
nmap 35.168.9.1 -F
```

```
05/02-16:24:06.690210  [**] [1:1000001:1] TCP port scan detected [**] [Priority: 0] {TCP} 77.224.188.168:60494 -> 10.1.15.11:110
```

Referencias

1. <https://albertomolina.wordpress.com/2009/01/09/nat-con-iptables/>
2. <https://www.linode.com/docs/guides/linux-router-and-ip-forwarding/>
3. <https://wiki.centos.org/HowTos/Network/IPTables>
4. [Configuring snort rules](#)
5. [Snort usage example](#)
6. <http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node1.html>