3: IDS y Pentesting
Parte I

2. Detección de intrusos

La máquina ids1 va a ejecutar una herramienta IDS, snort, que detecta accesos potencialmente maliciosos y los registra en un fichero de log. Esta herramienta no está instalada en las máquinas, para instalarla, ejecuta lo siguiente:
ids1:~/ dpkg -i *.deb
Al instalarla, preguntará la red a explorar dentro de snort: 100.<ID>.0.0/8 que incluye las máquinas que están conectadas al hub1 y hub2. La configuración por defecto te devolverá un error debido a las restricciones de memoria que tenemos en NetGUI-NG. A continuación realizaremos la configuración manual.
Vamos a utilizar una configuración simplificada de snort ya que es una herramienta pesada que necesita más memoria de la que tienen asignadas las máquinas de NetGUI-NG. Por eso, en la instalación por defecto te muestra un error con la configuración que trae por defecto el paquete.
Copia el fichero de configuración de snort que te proporcionamos a la carpeta de configuración de snort:
ids1:~/ cp snort.conf /etc/snort
El fichero de configuración es complejo y sólo tiene activadas la detección de cierto tipo de vulnerabilidades. Cuando snort descubra tráfico potencialmente malicioso escribirá una alerta en un fichero de logs y almacenará el tráfico malicioso en un fichero de captura. Estos ficheros se encontrarán en la carpeta /var/log/snort.
En la carpeta /etc/snort/rules/ se describen reglas predefinidas en snort para la inspección de tráfico. Dependiendo de la configuración del fichero /etc/snort/snort.conf se podrán cargar las reglas que se desean aplicar al tráfico que el IDS examine.
A continuación responde a las siguientes preguntas en la memoria de la práctica:

1. Mira el apartado ”Step #7: Customize your rule set” en el fichero /etc/snort/snort.conf e indica cuáles son los ficheros con reglas que se están cargando.

```
##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
##################################################

# Note for Debian users: The rules preinstalled in the system
# can be *very* out of date. For more information please read
# the /usr/share/doc/snort-rules-default/README.Debian file

# site specific rules
include $RULE_PATH/local.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
```

include $RULE_PATH/dos.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/info.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/scan.rules
Note: These rules are disable by default as they are
too coarse grained. Enabling them causes a large
performance impact
include $RULE_PATH/virus.rules

2. Mira el contenido del fichero /etc/snort/rules/icmp.rules. Las líneas que comienzan por # son comentarios, las reglas están escritas cada una en una única línea. Incluye la última regla de ese fichero en la memoria y explica el contenido.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Large ICMP Packet"; dsize:>800; reference:arachnids,246; classtype:bad-unknown; sid:499; rev:4;)

Para la regla previa, los campos de la cabecera de la regla serían: acción a aplicar alert (guarda el paquete en un fichero de captura y escribe un mensaje en el fichero log), el protocolo icmp, $EXTERNAL_NET any, puerto1 any, sentido de la comunicación -> (desde la direcciónIP1 y puerto1 dirigido a la direcciónIP2, puerto2), direcciónIP2 $HOME_NET y puerto2 any.
Las opciones se escriben separadas por ';', donde cada opción queda definida por el formato: nombre_opción:valor_opción. La opción msg indica el mensaje que se va a guardar en el fichero de log ("ICMP Large ICMP Packet").

Entre las opciones se encuentra su clasificación y prioridad, con el nombre de opción classtype se asocia un valor cuyo significado y el valor de prioridad de la alerta se pueden consultar en el fichero /etc/snort/classification.config. Números bajos de prioridad significa, prioridad muy alta.
La opción sid es el identificador Snort de la regla, se usa para herramientas de búsqueda de reglas en la configuración. La opción rev hace referencia al número de versión de esa regla.
config classification: bad-unknown,Potentially Bad Traffic, 2 → prioridad media

3. Busca en el fichero icmp.rules la regla que es una alerta que escribe el mensaje "ICMP PING NMAP", inclúyela en la memoria y explica todo lo que puedes saber de su contenido.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)

Para la regla previa, los campos de la cabecera de la regla serían: acción a aplicar alert (guarda el paquete en un fichero de captura y escribe un mensaje en el fichero log), el protocolo icmp, $EXTERNAL_NET any, puerto1 any, sentido de la comunicación -> (desde la direcciónIP1 y puerto1 dirigido a la direcciónIP2, puerto2), direcciónIP2 $HOME_NET y puerto2 any.
Las opciones se escriben separadas por ';', donde cada opción queda definida por el formato: nombre_opción:valor_opción. La opción msg indica el mensaje que se va a guardar en el fichero de log ("ICMP PING NMAP").

Entre las opciones se encuentra su clasificación y prioridad, con el nombre de opción classtype se

asocia un valor cuyo significado y el valor de prioridad de la alerta se pueden consultar en el fichero /etc/snort/classification.config. Números bajos de prioridad significa, prioridad muy alta.
La opción sid (469) es el identificador Snort de la regla, se usa para herramientas de búsqueda de reglas en la configuración. La opción rev hace referencia al número de versión de esa regla.
config classification: attempted-recon,Attempted Information Leak,2 → media

4. Busca en el fichero icmp-info.rules la regla que es una alerta que escribe el mensaje ICMP PING *NIX",inclúyela en la memoria y explica todo lo que puedes saber de su contenido.

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING *NIX"; itype:8; content:"|10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F|"; depth:32; classtype:misc-activity; sid:366; rev:7;)

Para la regla previa, los campos de la cabecera de la regla serían: acción a aplicar alert (guarda el paquete en un fichero de captura y escribe un mensaje en el fichero log), el protocolo icmp, direcciónIP1 $EXTERNAL_NET, puerto1 any, sentido de la comunicación -> (desde la direcciónIP1 y puerto1 dirigido a la direcciónIP2, puerto2), direcciónIP2 $HOME_NET y puerto2 any.
La opción msg indica el mensaje que se va a guardar en el fichero de log ("ICMP PING *NIX").

Viendo el fichero para la prioridad a través del campo classtype:
config classification: misc-activity,Misc activity,3 → prioridad baja

5. Explica cuál es la diferencia entre ambas reglas y el nivel de prioridad de cada una de ellas. ¿Por qué una tiene mayor prioridad que otra?

Según pone en el enunciado, cuanto menor sea el valor, mayor será su prioridad, en los casos que se han buscado, ninguno tenía la prioridad más alta, esta sería cundo el mensaje sufre algún altercado. La prioridad más baja sería justo al contrario.

6. Lanza snort en la máquina ids1 (snort -A console -c /etc/snort/snort.conf) para que comience a detectar tráfico potencialmente peligroso y déjalo lanzado para que te vaya mostrando las alertas que detecte en los apartados sucesivos.

Se lanza snort.

3. Pentesting (Penetration Testing)

Vamos a realizar algunas pruebas sencillas de penetración para ver el comportamiento de los protocolos y analizar algunas vulnerabilidades. Para este apartado no configuraremos las reglas del firewall fw1/fw2. En la siguiente parte de la práctica se observará como al configurar correctamente fw1/fw2 se pueden evitar los ataques más simples.

3.1. DoS en TCP

Vamos a realizar una prueba sencilla de DoS contra un servidor en la máquina server1, no podemos dejar ejecutándose mucho tiempo esta prueba porque se genera demasiado tráfico y NetGUI-NG no puede procesar tantos paquetes.

1. Con la herramienta nc lanza un servidor TCP en la máquina server1 y puerto 2222, ejecútalo en segundo plano para poder utilizar la consola de server1. Con la herramienta netstat puedes explorar las conexiones que hay activas en una determinada máquina. Ejecuta netstat -ant4 en server1 . Explica toda la información que te muestra la herramienta netstat. A continuación, ejecuta periódicamente netstat, ayúdate de la herramienta watch:
watch -n 0.5 netstat -ant

```
root@server1:~# nc -l -p 2222 &
[1] 2384
root@server1:~# netstat -ant4
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN

Every 0.5s: netstat -ant                     Fri Apr 28 22:21:45 2017

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
tcp6       0      0 :::2222                 :::*                    LISTEN
```

Netstat muestra un listado de las conexiones activas de una computadora, tanto entrantes como salientes.

2. Inicia una captura de tráfico en r1(eth0) almacenando el contenido en el fichero analisis-01.cap. Desde pc1 realiza un ataque SYN Flood hacia server1, a la aplicación ejecutándose en el puerto 2222, con hping3 utilizando --rand_source y déjalo funcionando hasta que veas que netstat te muestra varias conexiones. Después interrumpe hping3 inmediatamente con Ctrl+C. Interrumpe también la captura en r1(eth0). Fíjate como después de haber interrumpido hping3 todavía server1 tiene información sobre las conexiones ¿por qué? ¿en qué estado se encuentran? ¿Crees que este ataque le hace daño a server1?

```
root@pc1:~# hping3 -S -p 2222 --flood --rand-source 100.10.1.10
HPING 100.10.1.10 (eth0 100.10.1.10): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 100.10.1.10 hping statistic ---
11988 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

server1:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN
tcp        0      0 100.10.1.10:2222        157.73.33.33:1803       SYN_RECV
tcp        0      0 100.10.1.10:2222        160.122.159.213:1776    SYN_RECV
tcp        0      0 100.10.1.10:2222        162.53.82.137:1777      SYN_RECV
tcp        0      0 100.10.1.10:2222        160.38.115.167:1838     SYN_RECV
tcp        0      0 100.10.1.10:2222        155.117.82.140:1742     SYN_RECV
...
```

Sigue teniendo información porque se envían peticiones syn, desde unas IP's y puertos que físicamente no existen, por lo que nunca devolverá un ACK. El servidor intenta retransmitir el paquete, pero como no lo consigue, la potencia aumenta y al final queda bloquedo y se netstat -ant se queda como en la configuración inicial.

3. Fíjate en la cantidad de paquetes capturados en tan poco tiempo y explica el contenido de la captura.

En la captura observamos que el destino en muchas de ellas es erróneo, se ha generado aleatoriamente. Se observa que los paquetes están fuera de tiempo (out-of-order).

4. Observa en la máquina IDS si se ha generado algún mensaje de alerta en el terminal.

```
04/28-22:25:59.047768  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787422  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787424  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787426  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787427  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787429  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
04/28-22:26:27.787431  [**] [1:449:6] ICMP Time-To-Live Exceeded in Transit [**]
[Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.1.10
```

5. Busca en el fichero /etc/snort/rules/icmp-info.rules la alerta que ha generado el mensaje y explícala.

alert icmp $HOME_NET any -> $EXTERNAL_NET any (msg:"ICMP Time-To-Live Exceeded in Transit"; icode:0; itype:11; classtype:misc-activity; sid:449; rev:6;)

Para la regla previa, los campos de la cabecera de la regla serían: acción a aplicar alert, el protocolo icmp, el cual ha obtenido el mensaje Time-To-Live Exceeded in Transit, direcciónIP1 $HOME_NET, puerto1 any, direcciónIP2 $EXTERNAL_NET y puerto2 any.

La opción msg indica el mensaje que se va a guardar en el fichero de log ("ICMP Time-To-Live Exceeded in Transit").

Viendo el fichero para la prioridad a través del campo classtype:
config classification: misc-activity,Misc activity,3 → prioridad baja

6. Comprueba el fichero de captura que ha dejado snort con los paquetes que han provocado la alerta. Se encuentra en /var/log/snort/tcpdump.log.* . Examina el fichero y explica por qué esos paquetes cumplen la regla que has encontrado en /etc/snort/rules/icmp-info.rules.

En el contenido de una de las capturas de tcpdump.log, se muestra la información de Time-To-Live Exceeded, que es justo la alerta que nos aparecía anteriormente.

7. ¿Por qué crees que snort no considera que los mensajes TCP SYN enviados desde pc1 sean una alerta de seguridad?

Porque el sistema ha funcionado sin problemas, sin sobrepasar el número de conexiones, aunque no se haya enviando ningún ACK.
No es capaz de distinguir si es un cliente malicioso o no.

3.2. smurf attack

root@r1:~# tcpdump -i eth0 -s 0 -w
/hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/analisis-02.cap

root@fw1:~# ping -c 1 100.10.0.1
root@fw1:~# hping3 -c 1 --icmp --spoof 100.10.0.1 100.10.0.0

1. Explica el contenido de la captura. Los paquete pueden aparecer en un orden extraño, no te preocupes por eso, trata de ver qué paquetes se han generado y sus respuestas.

Cuando se ejecuta la herramienta de ping, un paquete de solicitud de eco de ICMP se envía al equipo de destino. Si el equipo de destino recibe el paquete de TCP, contesta para confirmar la solicitud ping. En el caso de un ataque smurf de negación de servicio, se crea la dirección IP devuelta del paquete de ping con el IP del equipo de destino. El ping se emite a la dirección IP entera de difusión. Esta técnica hace que cada equipo responda a los paquetes de ping falsos y conteste al equipo de destino, que lo satura.

En la captura, con hping3 no consigue realizar el request.

2. ¿Por qué crees que un ataque de este tipo puede hacer daño a la máquina víctima?

Porque satura la red.

3. Fíjate que snort ha detectado unas alertas ICMP, en particular la alerta ICMP PING NMAP, la que habías examinado en el apartado 2.3. Fíjate si los campos de esa alerta se corresponden con la captura realizada y explícalos.

(ping)
06/10-10:17:36.422912 [**] [1:368:6] ICMP PING BSDtype [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.0.1
06/10-10:17:36.422912 [**] [1:366:7] ICMP PING *NIX [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.0.1
06/10-10:17:36.422912 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.0.1
06/10-10:17:36.422934 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.1 -> 100.10.0.11

(hping3)
06/10-10:18:14.247166 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.11 -> 100.10.0.1
06/10-10:18:14.247172 [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 100.10.0.1 -> 100.10.0.0
06/10-10:18:14.247172 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.1 -> 100.10.0.0
06/10-10:18:14.247483 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.13 -> 100.10.0.1
06/10-10:18:14.247589 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.0.12 -> 100.10.0.1

(Del ejercicio 2.3) → el mensaje es tiene el mismo msg y prioridad.
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:3;)

Para la regla previa de este caso en concreto, los campos de la cabecera de la regla serían: acción a aplicar alert, el protocolo icmp, 100.10.0.1, puerto1 any -> 100.10.0.0 y puerto2 any.
La opción msg indica el mensaje que se va a guardar en el fichero de log ("ICMP PING NMAP").
config classification: attempted-recon,Attempted Information Leak,2 → media

Esta alerta concuerda con lo que se vió en el apartado 2.3.

4. ¿Por qué el primer paquete ICMP Echo Request que envió fw1 no generó la alerta ICMP PING NMAP?

Porque se genero como un paquete normal, realizo el “ping” de la dirección origen (la real) a la destino.
También, para que se genere esta alerta, el campo dst_size = 0, y el campo de ICMP ECHO Request = 8.

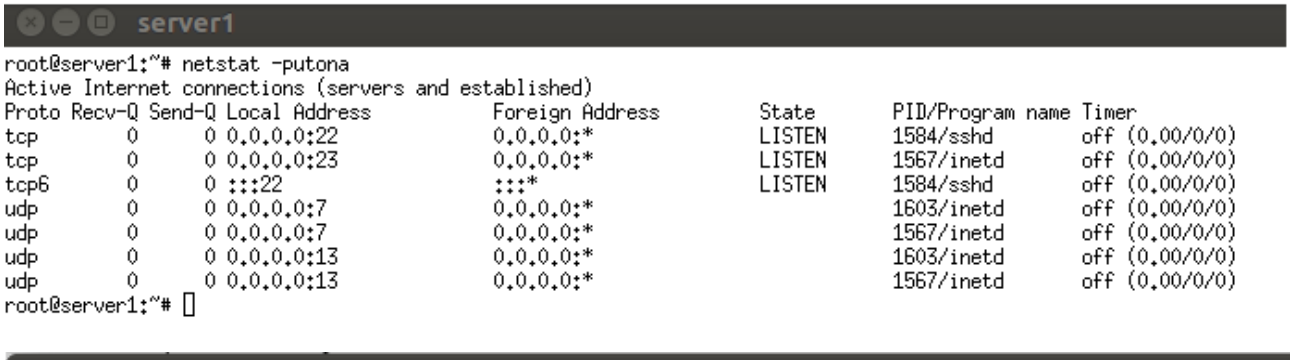
3. Firewall, IDS, Pentesting
Parte II

1. nmap

root@server1:/# cp /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/inetd.conf /etc/
root@server1:/# /usr/sbin/inetd

root@server2:~# /etc/init.d/apache2 start

Si queremos una información completa de las conexiones de nuestra red, en vez de ir una por una, podemos seguir la sintaxis que nos proporciona el propio programa, agrupar algunas variables y obtenemos el comando:
netstat -putona



Se puede observar que en el server1 tenemos los puertos de tcp en el 22 y 23, mientras que en udp estan en el 7 y 13. Para el protocolo tcp6 también usa el puerto 22.
Mirando en el fichero /etc/services vemos los puertos para que se utilizan.

```
echo            7/udp
daytime         13/udp
ssh             22/tcp          # SSH Remote Login Protocol
telnet          23/tcp
```

En el server2, usa el protocolo tcp6 en el puerto 80.

```
http            80/tcp      www         # WorldWideWeb HTTP
```

Aunque no es el protocolo tcp6, vemos el de tcp.

1.1. Descubrimiento de equipos

1.1.1. Sondeo de equipos Ping Scan

root@fw1:~# tcpdump -i eth0 -s 0 -w
/hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-01.cap

root@pc1:~# nmap -sn 200.10.0.20 100.10.1.20
Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 10:51 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 200.10.0.20
Host is up (0.00016s latency).
MAC Address: D2:02:CC:F9:B5:F8 (Unknown)
Nmap scan report for 100.10.1.20
Host is up (0.00061s latency).
Nmap done: 2 IP addresses (2 hosts up) scanned in 0.07 seconds

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta.

Pc1 envía un SYN y luego un ACK para establecer la conexión con server2, y este le responde con un RST, ACK y luego un RST.
En pc2 al estar en la misma subred se comprueba la conexión en los paquetes ARP.

2. ¿Cuál es la diferencia que encuentras entre el sondeo a la máquina pc2 y a la máquina server2?

Como pc2 (ip→ 200.10.0.20) se encuentra en la misma subred, no hay mensajes de icmp/tcp, como ocurre con la máquina server2. Para comprobar la conexión con pc2 se miran los paquetes de ARP.

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)
Ethernet II, Src: 52:6e:e2:34:bc:1b (52:6e:e2:34:bc:1b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: 52:6e:e2:34:bc:1b (52:6e:e2:34:bc:1b)
Sender IP address: 200.10.0.10
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)
Target IP address: 200.10.0.20

3. Explica si snort ha detectado alertas e indica cuáles y por qué.

06/10-10:43:42.195069 [**] [1:453:5] ICMP Timestamp Request [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.20
06/10-10:43:42.195342 [**] [1:451:5] ICMP Timestamp Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.20 -> 200.10.0.10

Se han levantado dos alertas, la primera es para la conexión de pc1 → server2, y la otra, para el reply de server2 → pc1.
Las dos alertas muestran la misma prioridad.

La alerta salta porque en el paquete 7 se observa un timestamp request y en el paquete 10 un timestamp reply.

1.1.2. Sondeo de protocolos

root@r1:~# tcpdump -i eth1 -s 0 -w /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-02.cap
root@pc1:~# nmap -sn -PO 100.10.1.10

Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 11:07 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 100.10.1.10
Host is up (0.0014s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta.

Pc1 envía 3 paquetes a server1, el primero tiene protocolo ICMP, el cual es un echo request, el segundo tiene el protocolo IGMPv1, con información Membership Query, y el tercero es un IPv4 sin información.
Después de estos paquetes, server1 intenta contestar, primero con un echo reply y luego, en el siguiente paquete se observa que el destino es inalcanzable.

2. Explica si snort ha detectado alertas e indica cuáles y por qué.

06/10-11:04:46.408739 [**] [1:469:3] ICMP PING NMAP [**] [Classification: Attempted Information Leak] [Priority: 2] {ICMP} 200.10.0.10 -> 100.10.1.10
06/10-11:04:46.408739 [**] [1:384:5] ICMP PING [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.10
06/10-11:04:46.409235 [**] [1:408:5] ICMP Echo Reply [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
06/10-11:04:46.409237 [**] [1:404:6] ICMP Destination Unreachable Protocol Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10

Los dos primeros los genera con la conexión de pc1 a server1, si vemos la captura, la primera alerta es generada por el protocolo IGMPv1, y la segunda por IPv4.

Estas dos últimas las ha generado el server1 al intentar responder a pc1.
La tercera alerta se ha producido en respuesta al ping.
La cuarta alerta se ha producido porque el destino es inalcanzable.

1.1.3. Sondeo TCP SYN

root@r1:~# tcpdump -i eth1 -s 0 -w /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-03.cap

root@pc1:~# nmap -sn -PS80-82 100.10.1.20
Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 11:50 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 100.10.1.20
Host is up (0.00071s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.03 seconds

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

Se envían 3 paquetes sync y cada uno de ellos recibe respuesta, se puede ver en la parte de info, como el puerto va del 80 al 82.

2. Explica si snort ha detectado alertas e indica cuáles y por qué.

No ha detectado ninguna alerta, porque los paquetes son de SYNC y todos han recibido su request y reply.

1.1.4. Sondeo UDP

root@r1:~# tcpdump -i eth1 -s 0 -w /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-04.cap

root@pc1:~# nmap -sn -PU7-13 100.10.1.10
Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 11:54 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 100.10.1.10
Host is up (0.00061s latency).
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

Hay dos paquetes que reciben respuesta, el de echo y daytime, que son los que tienen los puertos abiertos según comprobamos con netstat en los apartados anteriores. Todos los demás paquetes tienen como respuesta Destination unrechable.

2. Explica si coincide con los servicios UDP que hay arrancados en server1 y que examinaste con netstat.

Si, coincide, ya que son justo esos dos puertos los que estaban abiertos (el 7 y 13)

3. Explica si snort ha detectado alertas e indica cuáles y por qué.

04/29-11:54:53.933648 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-11:54:53.933651 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-11:54:53.933771 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.10
04/29-11:54:53.934046 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-11:54:53.934576 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-11:54:53.935357 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-11:54:53.937101 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.10

Están las alertas de redes inalcanzables que se han lanzado por las peticiones de conexión a puertos en los que no hay un servicio arrancado.

1.2. Escaneo de puertos

1.2.1. Sondeo TCP SYN

root@r1:~# tcpdump -i eth1 -s 0 -w /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-05.cap

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating SYN Stealth Scan at 12:41
Scanning 100.10.1.10 [6 ports]
Discovered open port 22/tcp on 100.10.1.10
Discovered open port 23/tcp on 100.10.1.10
Completed SYN Stealth Scan at 12:41, 0.02s elapsed (6 total ports)
Nmap scan report for 100.10.1.10
Host is up (0.0011s latency).
PORT STATE SERVICE
20/tcp closed ftp-data
21/tcp closed ftp
22/tcp open ssh

23/tcp open telnet
24/tcp closed priv-mail
25/tcp closed smtp

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
        Raw packets sent: 6 (264B) | Rcvd: 6 (248B)

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

Pc1 envía peticiones de conexión a server1, en la captura se observa que se asientes los puertos 22 y 23, que son los que están abiertos. En estos puertos, Pc1 asiente el reset de la secuencia de conexión.

2. Fíjate en la salida de nmap y la diferencia con el apartado 2.1.3 donde sólo se deseaba conocer si la máquina estaba activa. Con esta prueba se desean conocer los servicios TCP. Incluye esta salida en la memoria.

Discovered open port 22/tcp on 100.10.1.10
Discovered open port 23/tcp on 100.10.1.10

(La salida entera está puesta al principio del enunciado, donde he ido poniendo todas las salidas de nmap).

3. Explica si snort ha detectado alertas e indica cuáles y por qué.

No ha detectado ninguna alerta.

1.2.2. Sondeo UDP

root@r1:~# tcpdump -i eth1 -s 0 -w /hosthome/Escritorio/SEGURIDAD/p6_IDS_Pentesting/nmap-06.cap

root@pc1:~# nmap -Pn -sU -p 7-13 100.10.1.10
Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 12:51 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 100.10.1.10
Host is up (0.00070s latency).
PORT STATE SERVICE
7/udp open echo
8/udp closed unknown
9/udp closed discard
10/udp closed unknown
11/udp closed systat
12/udp closed unknown
13/udp open daytime

Nmap done: 1 IP address (1 host up) scanned in 1.22 seconds

1. Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

En la captura se observa que solo recibe respuesta de los servicios echo y daytime, que son los que su estado estaba abierto (puerto 7 y 13). Todos los demás tienen como respuesta destino inalcanzable.

2. Fíjate en la salida de nmap y la diferencia con el apartado 2.1.4 donde sólo se deseaba conocer si la máquina estaba activa. Con esta prueba se desean conocer los servicios UDP. Incluye esta salida en la memoria.

7/udp open   echo
8/udp closed unknown
9/udp closed discard
10/udp closed unknown
11/udp closed systat
12/udp closed unknown
13/udp open   daytime

(La salida entera está puesta al principio del enunciado, donde he ido poniendo todas las salidas de nmap).

3. Explica si snort ha detectado alertas e indica cuáles y por qué.

04/29-12:51:04.777868 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-12:51:04.778567 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.10
04/29-12:51:04.778569 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 200.10.0.10 -> 100.10.1.10
04/29-12:51:05.895038 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-12:51:05.895043 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-12:51:05.895184 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10
04/29-12:51:05.896006 [**] [1:402:7] ICMP Destination Unreachable Port Unreachable [**] [Classification: Misc activity] [Priority: 3] {ICMP} 100.10.1.10 -> 200.10.0.10

En la captura se observan 7 paquetes en los cuales tienen Destination Unreachable Port Unreachable, que son justamente las 7 alarmas que han saltado, excluyendo a dos de ellos que se produce cuando pc1 intenta llegar a server1, los otros 5 son a la inversa, server1 intenta conectar con pc1.
El destino es inalcanzable a causa de que los puertos a los que se intentan conectar no están activos.

1.2.3. Sondeo TCP Null, FIN, Xmas

1. Realiza una captura en r1(eth1) y guarda el contenido en un fichero nmap-07.cap. Utiliza nmap: nmap -Pn -sN -p 20-25 -v <dirIPServer1>
Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

```
root@pc1:~# nmap -Pn -sN -p 20-25 -v 100.10.1.10

Starting Nmap 6.00 ( http://nmap.org ) at 2017-04-29 13:11 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating NULL Scan at 13:11
Scanning 100.10.1.10 [6 ports]
Completed NULL Scan at 13:11, 2.35s elapsed (6 total ports)
Nmap scan report for 100.10.1.10
Host is up (0.00071s latency).
PORT   STATE         SERVICE
20/tcp closed        ftp-data
21/tcp closed        ftp
22/tcp open|filtered ssh
23/tcp open|filtered telnet
24/tcp closed        priv-mail
25/tcp closed        smtp

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.45 seconds
           Raw packets sent: 12 (480B) | Rcvd: 4 (160B)
```

Pc1 envía peticiones de conexión a server1. El servidor manda una instrucción de interrupción a los puertos que están cerrados. Pc1 reenvia los paquetes de los puertos 22 y 23 ya que no ha recibido respuesta.

2. Explica si snort ha detectado alertas e indica cuáles y por qué.

No ha detectado alertas

3. Realiza una captura en r1(eth1) y guarda el contenido en un fichero nmap-08.cap. Utiliza nmap: nmap -Pn -sF -p 20-25 -v <dirIPServer1>
Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

```
root@pc1:~# nmap -Pn -sF -p 20-25 -v 100.10.1.10

Starting Nmap 6.00 ( http://nmap.org ) at 2017-04-29 13:17 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
```

mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating FIN Scan at 13:17
Scanning 100.10.1.10 [6 ports]
Completed FIN Scan at 13:17, 2.36s elapsed (6 total ports)
Nmap scan report for 100.10.1.10
Host is up (0.00068s latency).
PORT STATE SERVICE
20/tcp closed ftp-data
21/tcp closed ftp
22/tcp open|filtered ssh
23/tcp open|filtered telnet
24/tcp closed priv-mail
25/tcp closed smtp

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.42 seconds
Raw packets sent: 11 (440B) | Rcvd: 4 (160B)

Pc1 envía peticiones de conexión a server1. El servidor manda una instrucción de interrupción a los puertos que están cerrados. Pc1 reenvia los paquetes de los puertos 22 y 23 para interrumpir la conexión.

4. Explica si snort ha detectado alertas e indica cuáles y por qué.

04/29-13:17:50.595506 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:23
04/29-13:17:50.595511 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:22
04/29-13:17:50.595516 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:25
04/29-13:17:50.596192 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:21
04/29-13:17:50.596193 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:20
04/29-13:17:50.596195 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33526 -> 100.10.1.10:24
04/29-13:17:51.701555 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33527 -> 100.10.1.10:25
04/29-13:17:51.701785 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33527 -> 100.10.1.10:22
04/29-13:17:51.701790 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33527 -> 100.10.1.10:23
04/29-13:17:52.819393 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33528 -> 100.10.1.10:23
04/29-13:17:52.819403 [**] [1:621:7] SCAN FIN [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:33528 -> 100.10.1.10:22

Hay alertas porque se esta intentando visualizar que puertos están abiertos y cuales no.

5. Realiza una captura en r1(eth1) y guarda el contenido en un fichero nmap-09.cap. Utiliza nmap: nmap -Pn -sX -p 20-25 -v <dirIPServer1>
Interrumpe la captura y observa qué paquetes se han enviado y explica cuáles tienen respuesta y el tipo de respuesta obtenida.

root@pc1:~# nmap -Pn -sX -p 20-25 -v 100.10.1.10

Starting Nmap 6.00 (http://nmap.org) at 2017-04-29 13:27 CEST
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Initiating XMAS Scan at 13:27
Scanning 100.10.1.10 [6 ports]
Completed XMAS Scan at 13:27, 2.35s elapsed (6 total ports)
Nmap scan report for 100.10.1.10
Host is up (0.0012s latency).
PORT STATE SERVICE
20/tcp closed ftp-data
21/tcp closed ftp
22/tcp open|filtered ssh
23/tcp open|filtered telnet
24/tcp closed priv-mail
25/tcp closed smtp

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 2.44 seconds
Raw packets sent: 11 (440B) | Rcvd: 4 (160B)

Pc1 envía peticiones de terminar la conexión urgentemente a server1. El servidor manda una instrucción de interrupción a los puertos que están cerrados. Pc1 reenvia los paquetes de los puertos 22 y 23 para interrumpir la conexión urgentemente.

6. Explica si snort ha detectado alertas e indica cuáles y por qué.

04/29-13:27:47.317274 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:22
04/29-13:27:47.317277 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:21
04/29-13:27:47.317611 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:23
04/29-13:27:47.317612 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:25
04/29-13:27:47.317613 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:24
04/29-13:27:47.317614 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35535 -> 100.10.1.10:20
04/29-13:27:48.427513 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35536 -> 100.10.1.10:23

04/29-13:27:48.427752 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35536 -> 100.10.1.10:21
04/29-13:27:48.427756 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35536 -> 100.10.1.10:22
04/29-13:27:49.532886 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35537 -> 100.10.1.10:22
04/29-13:27:49.532895 [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priority: 2] {TCP} 200.10.0.10:35537 -> 100.10.1.10:23

Hay alertas porque se esta intentando visualizar que puertos estan abiertos y cuales no.

2.firewall

Teoría de clase, ya que está mal configurada esta parte:

Solo dejo pasar trafico con el puerto 80 o al 22, con una solo pasa al puerto 80 le van a llegar los ataques, y al puerto 22, que no lleva lo de sync, no le va a llegar. Solo se permite meter un trafico al puerto 22 si va de sync, pero no le afecta ninguno de los otros ataques, porque el firewall tiene una regla de que ninguno de los otros ataques le va a afectar.