

SEGURIDAD INFORMÁTICA
ORQUÍDEA SEIJAS

CONTENIDO

1.	Introducción: configuración previa.....	2
2.	Ejercicio 1	2
a.	acciones realizadas y resultados obtenidos.....	2
i.	TELNET	3
ii.	HTTP	4
iii.	SSH	5
b.	Conclusiones globales	8
3.	Ejercicio 2	8
a.	Acciones realizadas y resultados obtenidos	8
i.	Escaneo tipo <i>Ping</i>	8
ii.	Escaneo tipo <i>TCP connect</i>	9
iii.	Escaneo silencioso.....	11
iv.	Escaneo para obtener información del sistema operativo y la versión concreta de los servicios activados	12
v.	Interfaz gráfico para NMAP.....	13
b.	Conclusiones globales	13
	Referencias	14

1. INTRODUCCIÓN: CONFIGURACIÓN PREVIA

Para la realización del ejercicio 1 tanto como del ejercicio 2, fue necesario crear y configurar las máquinas virtuales asociadas a los mismos, de forma que se pudo simular una pequeña red. Estas máquinas virtuales son: interno1, con una dirección IP correspondiente a 192.168.100.11, interno2, con una dirección IP correspondiente a 192.168.100.22, y observador, con una dirección IP correspondiente a 192.168.100.33. Los equipos interno1 e interno2 eran las máquinas legítimas, conectadas a una red interna, y el observador era la máquina atacante, que trataba de espiar el tráfico de dicha red.

Tras los pasos de configuración necesarios, se comprobó que era posible comunicarse con cualquiera de las máquinas, haciendo *ping*, tal y como se puede ver en la figura a continuación:

```
interno2:~# ping 192.168.100.11
PING 192.168.100.11 (192.168.100.11) 56(84) bytes of data.
64 bytes from 192.168.100.11: icmp_seq=1 ttl=64 time=4.75 ms
64 bytes from 192.168.100.11: icmp_seq=2 ttl=64 time=0.253 ms
^C
--- 192.168.100.11 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.253/2.502/4.752/2.250 ms
interno2:~# ping 192.168.100.33
PING 192.168.100.33 (192.168.100.33) 56(84) bytes of data.
64 bytes from 192.168.100.33: icmp_seq=1 ttl=64 time=1.55 ms
64 bytes from 192.168.100.33: icmp_seq=2 ttl=64 time=0.249 ms
^C
--- 192.168.100.33 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1005ms
rtt min/avg/max/mdev = 0.249/0.900/1.552/0.652 ms
```

2. EJERCICIO 1

El primer ejercicio consistió en el uso de la herramienta WIRESHARK desde el equipo atacante, observador, para interceptar tráfico TELNET, HTTP y SSH entre los equipos legítimos. Esta herramienta es un *sniffer* y analizador de protocolos que recopila, analiza, extrae y presenta el contenido de los paquetes que fluyen por la red.

a. ACCIONES REALIZADAS Y RESULTADOS OBTENIDOS

En primer lugar, se inició el entorno gráfico y se arrancó WIRESHARK, desde observador. A continuación, se inició la escucha de red utilizando la opción *Capture* del menú y haciendo click en *Interfaces*. Así, se pudo acceder al menú que se ve en la imagen adjuntada a continuación:



Llegados a este punto, fue necesario iniciar la escucha, haciendo click en el botón *Start* asociado a la interfaz eth0. Se comprobó el funcionamiento de la interceptación con los protocolos citados anteriormente.

i. TELNET

En primer lugar, se interceptaron paquetes de TELNET, desde interno2 hacia interno1, tal y como se puede observar en la imagen adjuntada a continuación:

No.	Time	Source	Destination .	Protocol
4	0.001393	192.168.100.22	192.168.100.11	TELNET
9	0.114649	192.168.100.22	192.168.100.11	TELNET
11	0.114654	192.168.100.22	192.168.100.11	TELNET
13	0.114753	192.168.100.22	192.168.100.11	TELNET
15	0.115010	192.168.100.22	192.168.100.11	TELNET
17	0.117172	192.168.100.22	192.168.100.11	TELNET
19	0.117529	192.168.100.22	192.168.100.11	TELNET
22	5.144905	192.168.100.22	192.168.100.11	TELNET
29	7.560830	192.168.100.22	192.168.100.11	TELNET
36	13.448750	192.168.100.22	192.168.100.11	TELNET
45	19.248790	192.168.100.22	192.168.100.11	TELNET
6	0.114137	192.168.100.11	192.168.100.22	TELNET
8	0.114376	192.168.100.11	192.168.100.22	TELNET

▶	Frame 22 (75 bytes on wire, 75 bytes captured)
▶	Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11
▶	Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11
▶	Transmission Control Protocol, Src Port: 36707 (36707), Dst Port: telnet (23)
▼	Telnet
	Data: usuario1\n

Tal y como se puede ver en la imagen, no existe ningún tipo de encriptación o codificación que impida ver el contenido del mensaje, ya que se puede ver “Data: usuario1\n” sin ningún problema. Este protocolo no tiene protección de la información en ningún caso, lo que fue posible apreciar tanto al iniciar la conexión como al ejecutar comandos en la sesión TELNET, lo que se puede ver en la imagen siguiente:

29	7.560830	192.168.100.22	192.168.100.11	TELNET
36	13.448750	192.168.100.22	192.168.100.11	TELNET
45	19.248790	192.168.100.22	192.168.100.11	TELNET
6	0.114137	192.168.100.11	192.168.100.22	TELNET
8	0.114376	192.168.100.11	192.168.100.22	TELNET

▶	Frame 36 (69 bytes on wire, 69 bytes captured)
▶	Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11:11
▶	Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11 (192.168.100.11)
▶	Transmission Control Protocol, Src Port: 36707 (36707), Dst Port: telnet (23), Seq: 36707
▼	Telnet
	Data: ls\n

Por otra parte, desde la red interna, es decir: desde interno2 hacia interno1, se realizaron las siguientes acciones, reflejadas en las capturas previamente explicadas:

```
interno2:~# telnet 192.168.100.11
Trying 192.168.100.11...
Connected to 192.168.100.11.
Escape character is '^I'.

Linux 2.6.26-1-686 (::ffff:192.168.100.22) (pts/0)

interno1 nombre: usuario1
usuario1
Contraseña:usuario1

Último inicio de sesión:jue abr 23 20:50:47 CEST 2009de localhosten pts/0
Linux ligero 2.6.26-1-686 #1 SMP Sat Jan 10 18:29:31 UTC 2009 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
usuario1@interno1:~$ ls
ls
usuario1@interno1:~$ cd /
cd /
usuario1@interno1:/$ _
```

ii. HTTP

El segundo protocolo utilizado para realizar pruebas con WIRESHARK fue HTTP. Para ello fue necesario realizar una conexión web desde interno2 a interno1, utilizando un navegador en modo texto. Tal y como se puede ver en la imagen adjuntada a continuación, el mensaje no está ni cifrado ni encriptado, por lo que se puede leer claramente su contenido.

No.	Time	Source	Destination .	Protocol	Info
11	56.127051	192.168.100.22	192.168.100.11	HTTP	GET / HT
13	56.127905	192.168.100.11	192.168.100.22	HTTP	HTTP/1.1
21	76.746775	192.168.100.22	192.168.100.11	HTTP	GET / HT
23	76.747314	192.168.100.11	192.168.100.22	HTTP	HTTP/1.1
32	138.802558	192.168.100.22	192.168.100.11	HTTP	GET / HT
34	138.803155	192.168.100.11	192.168.100.22	HTTP	HTTP/1.1
45	169.521959	192.168.100.22	192.168.100.11	HTTP	GET / HT
47	169.522461	192.168.100.11	192.168.100.22	HTTP	HTTP/1.1

<div>GET / HTTP/1.0\r\n</div> <div>Request Method: GET</div> <div>Request URI: /</div> <div>Request Version: HTTP/1.0</div> <div>Host: 192.168.100.11\r\n</div> <div>Accept: text/html, text/plain, text/css, text/sgml, */*;q=0.01\r\n</div> <div>Accept-Encoding: gzip, compress, bzip2\r\n</div>

iii. SSH

Finalmente, se observaron los paquetes asociados a una conexión SSH. Esta conexión es segura, es decir: está encriptada. Es por ello que los pasos a realizar para obtener esta conexión serán distintos a los de los protocolos explicados previamente. En primer lugar, fue necesario realizar la conexión, lo que se hizo desde el equipo interno1 al equipo interno2. Los mensajes intercambiados fueron los siguientes y se explicarán con más detalle a continuación:

Time	Source	Destination	Protocol	Info
57 400.834638	192.168.100.11	192.168.100.22	SSHv2	Server Protocol: SSH-2.0-OpenSSH_5.1p1 Debian-5\r\n
59 400.834844	192.168.100.22	192.168.100.11	SSHv2	Client Protocol: SSH-2.0-OpenSSH_5.1p1 Debian-5\r\n
61 400.835087	192.168.100.22	192.168.100.11	SSHv2	Client: Key Exchange Init
63 400.835720	192.168.100.11	192.168.100.22	SSHv2	Server: Key Exchange Init
64 400.835927	192.168.100.22	192.168.100.11	SSHv2	Client: Diffie-Hellman GEX Request
65 400.858539	192.168.100.11	192.168.100.22	SSHv2	Server: Diffie-Hellman Key Exchange
66 400.860084	192.168.100.22	192.168.100.11	SSHv2	Client: Diffie-Hellman GEX Init
67 400.870969	192.168.100.11	192.168.100.22	SSHv2	Server: Diffie-Hellman GEX Reply
69 405.946167	192.168.100.22	192.168.100.11	SSHv2	Client: New Keys
71 405.985913	192.168.100.22	192.168.100.11	SSHv2	Encrypted request packet len=46
73 405.986136	192.168.100.11	192.168.100.22	SSHv2	Encrypted response packet len=46

En primer lugar, se estableció el protocolo SSH a utilizar, en este caso: SSH-2.0-OpenSSH_5.1p1 Debian -5 por parte del servidor y del cliente, tal y como se puede ver en las imágenes adjuntadas a continuación:

<p>▶ Frame 57 (98 bytes on wire, 98 bytes captured)</p> <p>▶ Ethernet II, Src: CadmusCo_11:11:11 (08:00:27:11:11:11), Dst: CadmusCo_22:22:22</p> <p>▶ Internet Protocol, Src: 192.168.100.11 (192.168.100.11), Dst: 192.168.100.22 (192.168.100.22)</p> <p>▶ Transmission Control Protocol, Src Port: ssh (22), Dst Port: 47214 (47214), Seq: 3042158400, Win: 0, Len: 0</p> <p>▼ SSH Protocol</p> <p>Protocol: SSH-2.0-OpenSSH_5.1p1 Debian-5\r\n</p>
--

Ilustración 1 Servidor (interno1)

<p>▶ Frame 59 (98 bytes on wire, 98 bytes captured)</p> <p>▶ Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11:11</p> <p>▶ Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11 (192.168.100.11)</p> <p>▶ Transmission Control Protocol, Src Port: 47214 (47214), Dst Port: ssh (22), Seq: 3042158400, Win: 0, Len: 0</p> <p>▼ SSH Protocol</p> <p>Protocol: SSH-2.0-OpenSSH_5.1p1 Debian-5\r\n</p>
--

Ilustración 2 Cliente (interno2)

A continuación, tanto el cliente como el servidor inician el intercambio de claves. El contenido del mensaje en el cliente es el siguiente:

```

SSH Protocol
  SSH Version 2
    Packet Length: 788
    Padding Length: 8
    Key Exchange
      Msg code: Key Exchange Init (20)
      Algorithms
        Cookie: 4F52DE9DED8988FFF71183AA97A67193
        kex_algorithms length: 126
        kex_algorithms string: diffie-hellman-group-exchange-sha256,diffie-hellman-group-exchange-sh
        server_host_key_algorithms length: 15
        server_host_key_algorithms string: ssh-rsa,ssh-dss
        encryption_algorithms_client_to_server length: 157
        encryption_algorithms_client_to_server string: aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,
        encryption_algorithms_server_to_client length: 157
        encryption_algorithms_server_to_client string: aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc

```

Para intercambiar claves, se utiliza Diffie-Hellman, tal y como se puede ver en la imagen de inicio de intercambio de claves en *kex_algorithms string*. En primer lugar, el cliente procede a realizar una petición de intercambio de grupo de Diffie-Hellman. Esto se puede ver en la imagen adjunta a continuación:

```

Frame 64 (90 bytes on wire, 90 bytes captured)
Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11:11 (08:00:27:11:11:11)
Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11 (192.168.100.11)
Transmission Control Protocol, Src Port: 47214 (47214), Dst Port: ssh (22), Seq: 825, Ack: 817, Len
SSH Protocol
  SSH Version 2
    Packet Length: 20
    Padding Length: 6
    Key Exchange
      Msg code: Diffie-Hellman GEX Request (34)
      Payload: 0000040000000040000002000
      Padding String:

```

A continuación, el servidor responde a la petición:

```

Frame 65 (218 bytes on wire, 218 bytes captured)
Ethernet II, Src: CadmusCo_11:11:11 (08:00:27:11:11:11), Dst: CadmusCo_22:22:22 (08:00:27:22:22:22)
Internet Protocol, Src: 192.168.100.11 (192.168.100.11), Dst: 192.168.100.22 (192.168.100.22)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 47214 (47214), Seq: 817, Ack: 8
SSH Protocol
  SSH Version 2
    Packet Length: 148
    Padding Length: 8
    Key Exchange
      Msg code: Diffie-Hellman Key Exchange Reply (31)
      Payload: 0000008100DE49FC9069994C379D2B6563EFD37EFAE6785E...
      Padding String:

```

A continuación, se produce un mensaje de inicio intercambio de grupo Diffie-Hellman por parte del cliente, para generar las llaves definitivas para la comunicación:

```

▶ Frame 66 (210 bytes on wire, 210 bytes captured)
▶ Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11:11 (08:00:27:11:11:11)
▶ Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11 (192.168.100.11)
▶ Transmission Control Protocol, Src Port: 47214 (47214), Dst Port: ssh (22), Seq: 849, Ack: 969,
▼ SSH Protocol
  ▼ SSH Version 2
    Packet Length: 140
    Padding Length: 5
    ▼ Key Exchange
      Msg code: Diffie-Hellman GEX Init (32)
      Payload: 0000008100D86649EC182EC459612E69F3588F266B99585A...
      Padding String:

```

0000	08 00 27 11 11 08 00	27 22 22 22 08 00 45 00	..'. '""..E.
0010	00 c4 5e ec 40 00 06	91 d5 c0 a8 64 16 c0 a8	..^.@.@.d...
0020	64 0b b8 6e 00 16 b7 65	24 66 b6 ed b6 67 80 18	d..n...e \$f...g..
0030	00 8d e4 e0 00 01 01	08 0a 00 07 80 02 00 07

```

eth0: <live capture in progress> Fil... Packets: 376 Displayed: 171 Marked: 0 Profile: Default

```

Así pues, se produce una respuesta por parte del servidor y se generan las llaves definitivas a utilizar en la comunicación:

```

Frame 67 (786 bytes on wire, 786 bytes captured)
Ethernet II, Src: CadmusCo_11:11:11 (08:00:27:11:11:11), Dst: CadmusCo_22:22:22 (08:00:27:22:22:22)
Internet Protocol, Src: 192.168.100.11 (192.168.100.11), Dst: 192.168.100.22 (192.168.100.22)
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 47214 (47214), Seq: 969, Ack: 993, L
SSH Protocol
  ▼ SSH Version 2
    Packet Length: 700
    Padding Length: 10
    ▼ Key Exchange
      Msg code: Diffie-Hellman GEX Reply (33)
      Payload: 000001150000000077373682D727361000000012300000101...
      Padding String:
      MAC String:

```

Finalmente, el cliente envía un mensaje asociado a las nuevas llaves:

```

▶ Frame 69 (82 bytes on wire, 82 bytes captured)
▶ Ethernet II, Src: CadmusCo_22:22:22 (08:00:27:22:22:22), Dst: CadmusCo_11:11:11
▶ Internet Protocol, Src: 192.168.100.22 (192.168.100.22), Dst: 192.168.100.11 (192.168.100.11)
▶ Transmission Control Protocol, Src Port: 47214 (47214), Dst Port: ssh (22), Seq:
▼ SSH Protocol
  ▼ SSH Version 2
    Packet Length: 12
    Padding Length: 10
    ▼ Key Exchange
      Msg code: New Keys (21)
      Padding String:

```


Llegados a este punto, todos los mensajes enviados a continuación están cifrados y, por lo tanto, es imposible conocer su contenido a simple vista.

b. CONCLUSIONES GLOBALES

Así pues, tras realizar pruebas con los tres protocolos, es necesario hacer notar que tanto Telnet como HTTP no tienen mensajes encriptados. Esto provoca que sean vulnerables al uso de analizadores de tráfico de red. Es por ello que solo se debería utilizar SSH y la versión segura de HTTP (HTTPS) para comunicaciones, puesto que estos protocolos sí están encriptados y por lo tanto, no son vulnerables al uso de analizadores de tráfico de red.

3. EJERCICIO 2

Este ejercicio consistió en el uso de la herramienta de escaneo de puertos NMAP para obtener información de los equipos y servicios de la red.

a. ACCIONES REALIZADAS Y RESULTADOS OBTENIDOS

i. Escaneo tipo *Ping*

En primer lugar, se realizó un escaneo tipo *Ping*, con la opción *-sP*, para identificar las máquinas que componían la red. Tal y como se puede ver a continuación, las máquinas que componen la red son interno1, interno2 y observador.

```
observador:~# nmap -sP 192.168.100.0/24

Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 11:30 CET
Host 192.168.100.11 appears to be up.
MAC Address: 08:00:27:11:11:11 (Cadmus Computer Systems)
Host 192.168.100.22 appears to be up.
MAC Address: 08:00:27:22:22:22 (Cadmus Computer Systems)
Host 192.168.100.33 appears to be up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 31.229 seconds
```

El escaneo no deja rastro en ninguno de los elementos escaneados, así que iniciamos el servidor inetutils-inetd para ello. Así pues, tal y como se puede ver en la imagen adjuntada, los últimos cambios realizados en el fichero de log son del día de la realización de la práctica (24 de diciembre):

```
interno2:~# tail /var/log/syslog
Dec 24 11:30:38 ligero dhclient: No DHCP OFFERS received.
Dec 24 11:30:38 ligero dhclient: No working leases in persistent database - sleeping.
Dec 24 11:33:33 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
Dec 24 11:33:36 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
Dec 24 11:33:44 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 10
Dec 24 11:33:54 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 21
Dec 24 11:34:15 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
Dec 24 11:34:23 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 11
Dec 24 11:34:34 ligero dhclient: No DHCP OFFERS received.
Dec 24 11:34:34 ligero dhclient: No working leases in persistent database - sleeping.
```

```

interno1:~# tail /var/log/syslog
Dec 24 11:29:23 ligero dhclient: No DHCPOFFERS received.
Dec 24 11:29:23 ligero dhclient: No working leases in persistent database - sleeping.
Dec 24 11:32:32 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
Dec 24 11:32:36 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 4
Dec 24 11:32:40 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 8
Dec 24 11:32:48 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 15
Dec 24 11:33:03 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 20
Dec 24 11:33:23 ligero dhclient: DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 10
Dec 24 11:33:33 ligero dhclient: No DHCPOFFERS received.
Dec 24 11:33:33 ligero dhclient: No working leases in persistent database - sleeping.

```

ii. Escaneo tipo *TCP connect*

A continuación, se realiza un escaneo tipo *TCP connect*, con el fin de determinar qué puertos están abiertos. Para interno1, se obtiene la siguiente información:

```

Scanning 192.168.100.11 [1 port]
Completed ARP Ping Scan at 11:38, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:38
Completed Parallel DNS resolution of 1 host. at 11:38, 13.00s elapsed
Initiating Connect Scan at 11:38
Scanning 192.168.100.11 [1715 ports]
Discovered open port 22/tcp on 192.168.100.11
Discovered open port 23/tcp on 192.168.100.11
Discovered open port 80/tcp on 192.168.100.11
Completed Connect Scan at 11:38, 1.16s elapsed (1715 total ports)
Host 192.168.100.11 appears to be up ... good.
Interesting ports on 192.168.100.11:
Not shown: 1712 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
MAC Address: 08:00:27:11:11:11 (Cadmus Computer Systems)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 14.255 seconds
Raw packets sent: 1 (42B) | Rcvd: 1 (42B)

```

Tal y como se puede ver, esta máquina tiene abiertos tres puertos: 22, 23 y 80.

El rastro que queda en el log de interno1 es el siguiente. Tal y como se puede ver, se realiza un cambio a las 11:38, hora a la que se realizó el análisis.

```
Dec 24 11:33:33 ligero dhclient: No working leases in persistent database - sleeping.  
Dec 24 11:38:27 ligero telnetd[2142]: getpeername: Transport endpoint is not connected
```

Para interno2, realizando el mismo tipo de escaneo, se obtiene la siguiente información:

```
observador:~# nmap -sT -v 192.168.100.22  
  
Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 12:04 CET  
Initiating ARP Ping Scan at 12:04  
Scanning 192.168.100.22 [1 port]  
Completed ARP Ping Scan at 12:04, 0.00s elapsed (1 total hosts)  
Initiating Parallel DNS resolution of 1 host. at 12:04  
Completed Parallel DNS resolution of 1 host. at 12:04, 13.00s elapsed  
Initiating Connect Scan at 12:04  
Scanning 192.168.100.22 [1715 ports]  
Discovered open port 22/tcp on 192.168.100.22  
Completed Connect Scan at 12:04, 0.14s elapsed (1715 total ports)  
Host 192.168.100.22 appears to be up ... good.  
Interesting ports on 192.168.100.22:  
Not shown: 1714 closed ports  
PORT      STATE SERVICE  
22/tcp    open  ssh  
MAC Address: 08:00:27:22:22:22 (Cadmus Computer Systems)  
  
Read data files from: /usr/share/nmap  
Nmap done: 1 IP address (1 host up) scanned in 13.240 seconds  
Raw packets sent: 1 (42B) | Rcvd: 1 (42B)
```

Tal y como se puede ver, en este caso solo hay un puerto abierto. Lo que es consistente ya que de interno1 se habían abierto los puertos asociados a TELNET, HTTP y SSH, mientras que de interno2 solo estaba abierto SSH. El cambio asociado en el backlog a este escaneo, en interno2, es el siguiente:

```
Dec 24 12:04:48 ligero dhclient: No DHCP OFFERS received.  
Dec 24 12:04:48 ligero dhclient: No working leases in persistent database - sleeping.
```

iii. Escaneo silencioso

A continuación, se repitió el escaneo de puertos, realizando un escaneo silencioso. Para interno1 se obtuvieron los siguientes resultados:

```
Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 12:06 CET
Initiating ARP Ping Scan at 12:06
Scanning 192.168.100.11 [1 port]
Completed ARP Ping Scan at 12:06, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:06
Completed Parallel DNS resolution of 1 host. at 12:06, 13.00s elapsed
Initiating SYN Stealth Scan at 12:06
Scanning 192.168.100.11 [1715 ports]
Discovered open port 23/tcp on 192.168.100.11
Discovered open port 22/tcp on 192.168.100.11
Discovered open port 80/tcp on 192.168.100.11
Completed SYN Stealth Scan at 12:06, 0.14s elapsed (1715 total ports)
Host 192.168.100.11 appears to be up ... good.
Interesting ports on 192.168.100.11:
Not shown: 1712 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
MAC Address: 08:00:27:11:11:11 (Cadmus Computer Systems)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 13.238 seconds
Raw packets sent: 1716 (75.502KB) | Rcvd: 1716 (78.932KB)
```

No se produce ningún cambio, el último mensaje, tras ejecutar *tail*, es el mismo que había previamente:

```
Dec 24 12:03:14 ligero telnetd[23041]: ttloop: read: Connection reset by peer
interno1:~#
```

Para interno 2 se obtienen los mismos resultados

```
observador:~# nmap -sS -v 192.168.100.22

Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 12:08 CET
Initiating ARP Ping Scan at 12:08
Scanning 192.168.100.22 [1 port]
Completed ARP Ping Scan at 12:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:08
Completed Parallel DNS resolution of 1 host. at 12:08, 13.00s elapsed
Initiating SYN Stealth Scan at 12:08
Scanning 192.168.100.22 [1715 ports]
Discovered open port 22/tcp on 192.168.100.22
Completed SYN Stealth Scan at 12:08, 0.14s elapsed (1715 total ports)
Host 192.168.100.22 appears to be up ... good.
Interesting ports on 192.168.100.22:
Not shown: 1714 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 08:00:27:22:22:22 (Cadmus Computer Systems)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 13.238 seconds
Raw packets sent: 1716 (75.502KB) | Rcvd: 1716 (78.932KB)
```

```
Dec 24 12:04:48 ligero dhclient: No DHCP OFFERS received.
Dec 24 12:04:48 ligero dhclient: No working leases in persistent database - sleeping.
interno2:~#
```

- iv. Escaneo para obtener información del sistema operativo y la versión concreta de los servicios activados

Para interno1 se obtienen las versiones concretas de los protocolos cuyos puertos tiene abiertos y además, se obtiene el sistema operativo, tal y como se puede ver a continuación:

```
observador:~# nmap -sT -O -sV 192.168.100.11

Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 12:10 CET
Interesting ports on 192.168.100.11:
Not shown: 1712 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
23/tcp    open  telnet   BSD-derived telnetd
80/tcp    open  http     Apache httpd 2.2.9 ((Debian))
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port22-TCP:V=4.62%I=7%D=12/24%Time=5A3F8B39%P=i686-pc-linux-gnu%r(NULL,
SF:20,"SSH-2.0-OpenSSH_5\1p1x20Debian-5\r\n");
MAC Address: 08:00:27:11:11:11 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.24
Uptime: 0.100 days (since Sun Dec 24 09:46:48 2017)
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.653 seconds
```

Esto es peligroso ya que cabe la posibilidad de que haya vulnerabilidades conocidas asociadas a una versión concreta de software y, si es posible ver detalladamente la versión de cada uno de los protocolos o sistema operativo, cabe la posibilidad de que esta se explote.

Para interno2 se obtuvo el mismo tipo de información:

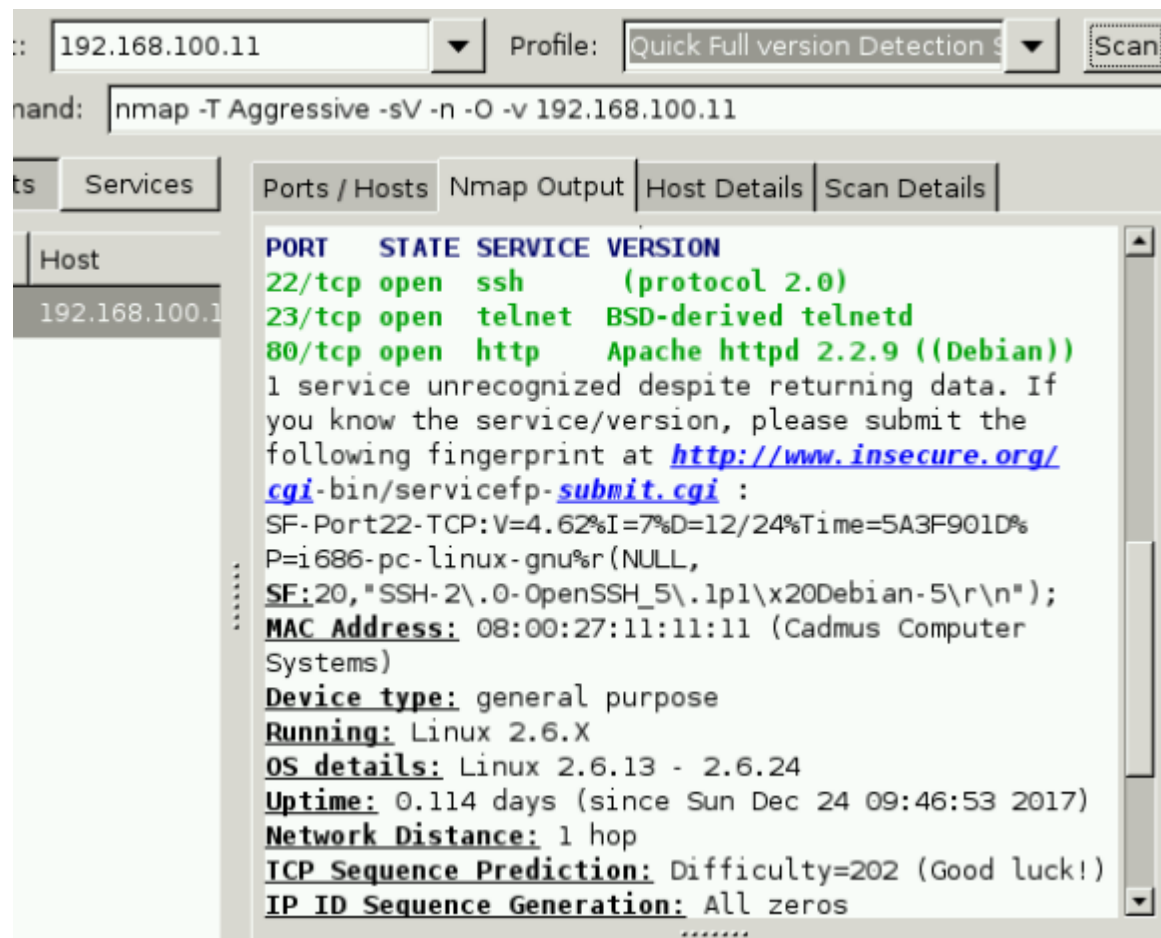
```
observador:~# nmap -sT -O -sV 192.168.100.22

Starting Nmap 4.62 ( http://nmap.org ) at 2017-12-24 12:15 CET
Interesting ports on 192.168.100.22:
Not shown: 1714 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      (protocol 2.0)
1 service unrecognized despite returning data. If you know the service/version,
please submit the following fingerprint at http://www.insecure.org/cgi-bin/servicefp-submit.cgi :
SF-Port22-TCP:V=4.62%I=7%D=12/24%Time=5A3F8C47%P=i686-pc-linux-gnu%r(NULL,
SF:20,"SSH-2.0-OpenSSH_5\1p1x20Debian-5\r\n");
MAC Address: 08:00:27:22:22:22 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.13 - 2.6.24
Uptime: 0.104 days (since Sun Dec 24 09:45:38 2017)
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.574 seconds
observador:~#
```

v. Interfaz gráfico para NMAP

La GUI permite obtener la misma información que a través de línea de comandos, pero presentada de forma más cómoda y legible.



b. CONCLUSIONES GLOBALES

En primer lugar, es necesario hacer notar que no se deberían abrir más puertos de los necesarios. Al realizar el análisis de puertos es posible observar el estado y, si un puerto que no debería estar abierto, lo está, se puede producir un fallo de seguridad. Así pues, como medida de seguridad básica, no se debería abrir ningún puerto que no sea necesario.

Como medida de seguridad más avanzada, es recomendable establecer un cortafuegos para evitar que se ejecute el comando sobre la red. Sin embargo, si el cortafuegos presenta vulnerabilidades, existe la posibilidad de que un usuario experimentado utilice Nmap para evadir estos cortafuegos, ya que Nmap tiene herramientas para ello.

REFERENCIAS

- NMAP.ORG, <https://nmap.org>. [Visitado por última vez: 24/12/2017]