



## SEGURIDAD INFORMÁTICA PRÁCTICA I (PARTE I)

Los objetivos de esta práctica son los siguientes:

### I. DEMONIOS EN LINUX

1. Estudiar los servicios (demonios) en Linux y el proceso de conexión cliente-servidor.
2. Instalar, arrancar y administrar dos tipos de demonios en Linux:
  - a. Los demonios dependientes y gestionados por el súper demonio de red o súper servidor. Es este súper demonio el que escucha y arranca los demonios al llegar una petición al puerto correspondiente.
  - b. Se elegirá para este caso el demonio telnetd, que deberá arrancarse mediante el súper demonio de red Inetd.
  - c. Los demonios *standalone*, llamados así porque funcionan de forma independiente del súper demonio de red. En este caso tendremos el Open SSH y el Proftpd



## **II. USO DEL WIRESHARK COMO SNIFFER**

1. Uso de un sniffer como el Wireshark para:
  - a. Identificar los paquetes del Three-way handshake
  - b. Identificar las características más importantes de los paquetes enviados: direcciones IP origen y destino, flags, número de secuencia y número ack (Leer artículo de la revista Hackxcrack “1-port\_scanning\_hxc.pdf”, pags. 59-61).
2. Capturar los paquetes con el nombre de usuario y contraseña en una sesión ftp y telnet (no cifrada) y en una sesión ssh y sftp (cifrada) para ver sus datos, en este caso los usuarios y contraseñas:
  - a. En texto claro en telnet y en ftp
  - b. Cifradas mediante claves pública y privada en SSH (ya explicaremos este método de cifrado en el curso más adelante)

## **III. USO DE NMAP PARA ESCANEAR PUERTOS**

1. Aprender a usar nmap para escanear puertos
2. Entender los diferentes tipos de escaneo que hay y el uso de los distintos flags en los paquetes TCP/IP



## ACTIVIDADES

### PARTE 1: Instalar servidor Telnet

#### 1. Instalar un servidor telnet en Linux.

- a. ¿Cómo funciona y cómo se instala?
  - i. Lo lanza el súper demonio de red llamado **inetd**.
  - ii. Este súper demonio se instala con la orden **sudo apt install openbsd-inetd**
  - iii. Después se instala el demonio servidor telnet con la orden **sudo apt install telnetd**
- b. Arrancar el servidor telnet (que en realidad es arrancar el súper demonio de red) con la orden **sudo /etc/init.d/openbsd-inetd start**
- c. ¿Cómo puedo saber si está arrancado o no?
  - i. Mediante la orden **systemctl sudo systemctl status openbsd-inetd.service**
  - ii. Mirando los procesos que están ejecutándose en el sistema mediante la orden **ps aux | grep inetd**
  - iii. Mediante nmap

```
(diego@kali)-[/etc]
$ nmap -p 23 172.20.10.2
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 16:04 CEST
Nmap scan report for 172.20.10.2
Host is up (0.00066s latency).

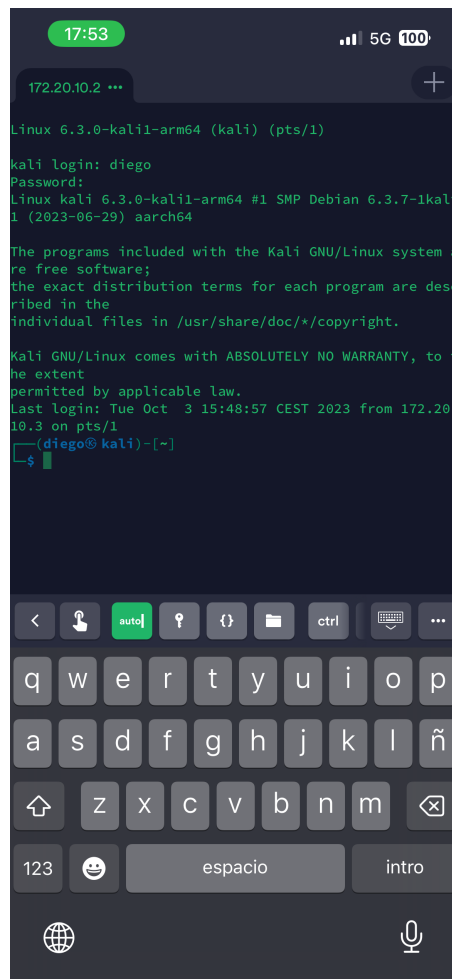
PORT      STATE SERVICE
23/tcp    open  telnet

Nmap done: 1 IP Address (1 host up) scanned in 0.06 seconds
```

- iv. Mediante cualquier otro método alternativo **telnet (ip del Kali) desde el cliente (Windows o mac)**
- d. ¿Qué puerto utiliza este servidor? **23**

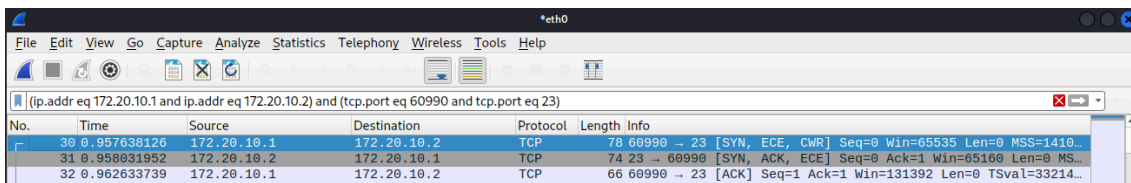
2. Instalar un cliente telnet en móvil, por ejemplo Termius, en iOS o en Android (también podríamos conectarnos mediante el CMD en Windows, aunque en este caso hay que habilitarlo previamente porque está deshabilitado por defecto).

- a. Conectarse desde el móvil al servidor telnet



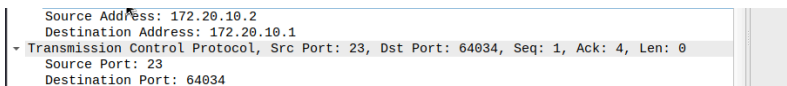
### 3. Monitorización con un sniffer

- a. Una vez que se ha logrado la conexión cliente-servidor, capturar mediante el wireshark los siguientes paquetes:
  - i. Three-way handshake – Tres paquetes con los que se establece la conexión cliente-servidor



No.	Time	Source	Destination	Protocol	Length	Info
30	0.957638126	172.20.10.1	172.20.10.2	TCP	78	60990 → 23 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1410
31	0.958831952	172.20.10.2	172.20.10.1	TCP	74	23 → 60990 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MS...
32	0.962633739	172.20.10.1	172.20.10.2	TCP	66	60990 → 23 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=33214...

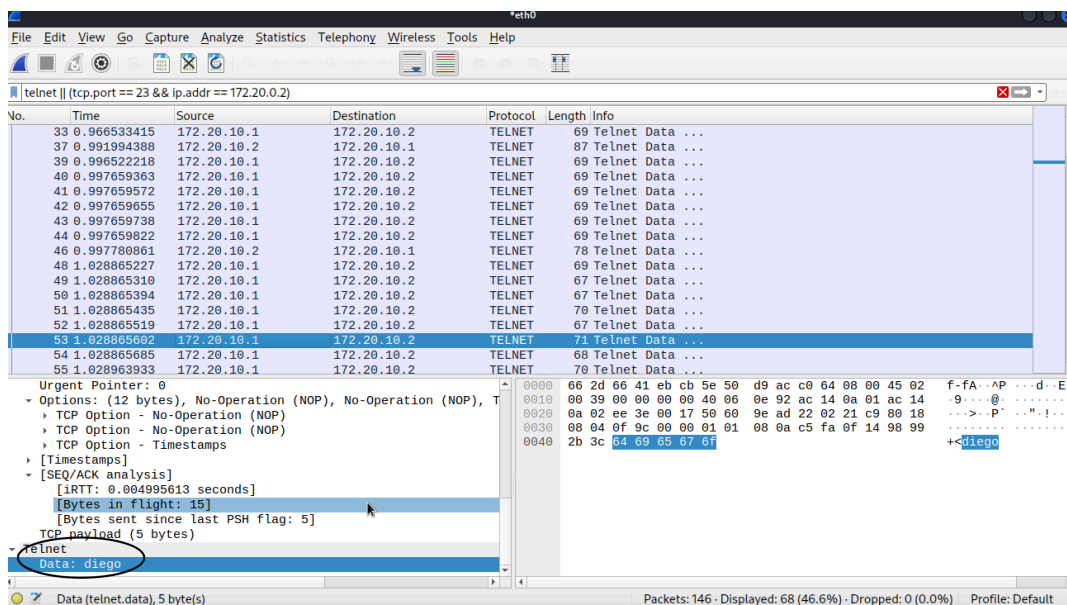
ii. Donde aparezcan los puertos del cliente y el servidor, junto con sus direcciones IP.



Source Address: 172.20.10.2
Destination Address: 172.20.10.1
Transmission Control Protocol, Src Port: 23, Dst Port: 64034, Seq: 1, Ack: 4, Len: 0
Source Port: 23
Destination Port: 64034

iii. USUARIO Y CONTRASEÑA

## 1. Usuario:



No.	Time	Source	Destination	Protocol	Length	Info
33	0.966533415	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
37	0.991994388	172.20.10.2	172.20.10.1	TELNET	87	Telnet Data ...
39	0.996522218	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
40	0.997659363	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
41	0.997659572	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
42	0.997659655	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
43	0.997659738	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
44	0.997659822	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
46	0.997780861	172.20.10.2	172.20.10.1	TELNET	78	Telnet Data ...
48	1.028865227	172.20.10.1	172.20.10.2	TELNET	69	Telnet Data ...
49	1.028865310	172.20.10.1	172.20.10.2	TELNET	67	Telnet Data ...
50	1.028865394	172.20.10.1	172.20.10.2	TELNET	67	Telnet Data ...
51	1.028865435	172.20.10.1	172.20.10.2	TELNET	70	Telnet Data ...
52	1.028865519	172.20.10.1	172.20.10.2	TELNET	67	Telnet Data ...
53	1.028865602	172.20.10.1	172.20.10.2	TELNET	71	Telnet Data ...
54	1.028865685	172.20.10.1	172.20.10.2	TELNET	68	Telnet Data ...
55	1.028963933	172.20.10.1	172.20.10.2	TELNET	70	Telnet Data ...

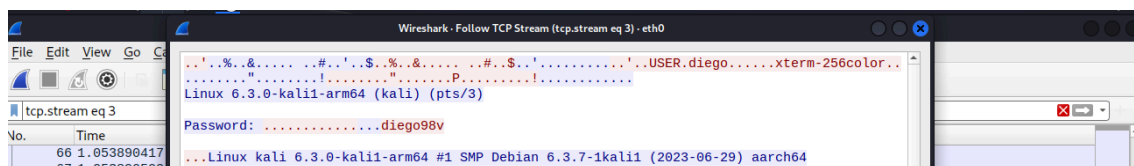
Urgent Pointer: 0

- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), T...
- TCP Option - No-Operation (NOP)
- TCP Option - No-Operation (NOP)
- TCP Option - Timestamps
- [Timestamps]
- [SEQ/ACK analysis]
- [RTT: 0.004995613 seconds]
- [Bytes in flight: 15]
- [Bytes sent since last PSH flag: 5]
- TCP payload (5 bytes)
- Telnet
- Data: diego

Data (telnet.data), 5 byte(s)

Packets: 146 - Displayed: 68 (46.6%) - Dropped: 0 (0.0%) - Profile: Default

## 2. Contraseña



No.	Time	Source	Destination	Protocol	Length	Info
66	1.053890417	172.20.10.1	172.20.10.2	TELNET	71	Telnet Data ...
67	1.053890500	172.20.10.1	172.20.10.2	TELNET	71	Telnet Data ...

Urgent Pointer: 0

- Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), T...
- TCP Option - No-Operation (NOP)
- TCP Option - No-Operation (NOP)
- TCP Option - Timestamps
- [Timestamps]
- [SEQ/ACK analysis]
- [RTT: 0.004995613 seconds]
- [Bytes in flight: 15]
- [Bytes sent since last PSH flag: 5]
- TCP payload (5 bytes)
- Telnet
- Data: diego98v

Data (telnet.data), 5 byte(s)

Packets: 146 - Displayed: 68 (46.6%) - Dropped: 0 (0.0%) - Profile: Default

La contraseña aparece como una letra por cada paquete, lo que hemos hecho es ir al primer paquete y darle a la opción follow TCP stream para poder mostrar la contraseña por completo

iv. Capturar los paquetes con los que se cierra la conexión

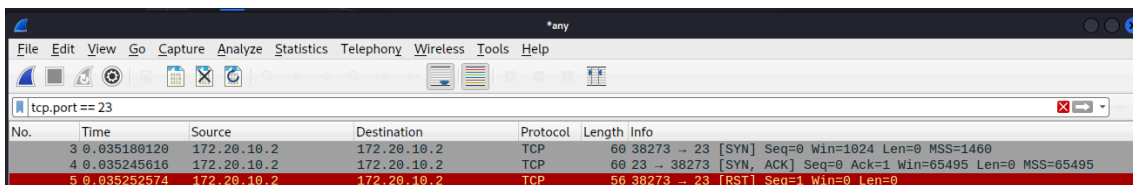
139	6.272169056	172.20.10.2	172.20.10.1	TCP	66 23 → 60990 [FIN, ACK] Seq=1237 Ack=126 Win=65280 Len=0 TS...
140	6.273772650	172.20.10.1	172.20.10.2	TCP	66 60990 → 23 [ACK] Seq=126 Ack=1237 Win=131008 Len=0 TSval=...
141	6.295279865	172.20.10.1	172.20.10.2	TCP	66 60990 → 23 [ACK] Seq=126 Ack=1238 Win=131072 Len=0 TSval=...
142	6.295280157	172.20.10.1	172.20.10.2	TCP	66 60990 → 23 [FIN, ACK] Seq=126 Ack=1238 Win=131072 Len=0 T...
143	6.295349572	172.20.10.2	172.20.10.1	TCP	66 23 → 60990 [ACK] Seq=1238 Ack=127 Win=65280 Len=0 TSval=2...

**NOTA:** No serán válidas capturas que no estén bien filtradas de forma que se puedan encontrar de forma más o menos sencilla esos paquetes. Destacar esos paquetes con un trazo rojo ayuda, pero eso no es un filtro de Wireshark.

#### 4. Escaneo de los puertos mediante nmap

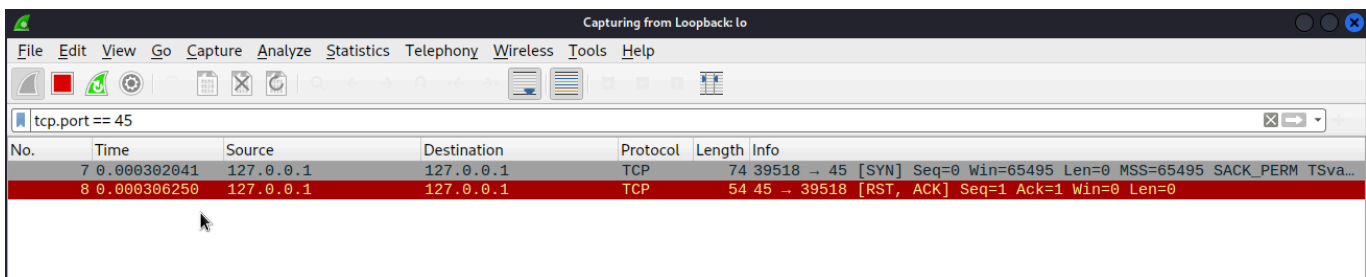
- Hacer un escaneo **HALF SCAN** al servidor telnet mediante nmap
- Identificar mediante wireshark y los filtros necesarios los paquetes mandados en ese escaneo a ese puerto en concreto. Para ello, debe aparecer:

- Un escaneo filtrado con éxito (a un puerto abierto)



No.	Time	Source	Destination	Protocol	Length	Info
3	0.035180120	172.20.10.2	172.20.10.2	TCP	60	38273 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
4	0.035245616	172.20.10.2	172.20.10.2	TCP	60	23 → 38273 [SYN, ACK] Seq=0 Ack=1 Win=65495 Len=0 MSS=65495
5	0.035252574	172.20.10.2	172.20.10.2	TCP	56	38273 → 23 [RST] Seq=1 Win=0 Len=0

- Un escaneo filtrado a un puerto cerrado



No.	Time	Source	Destination	Protocol	Length	Info
7	0.000302041	127.0.0.1	127.0.0.1	TCP	74	39518 → 45 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSva...
8	0.000306250	127.0.0.1	127.0.0.1	TCP	54	45 → 39518 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

**NOTA:** El resultado de estos pantallazos deben reflejar las diapositivas que hay en la teoría donde se explica este tipo de escaneo.

## PARTE 2: PROFTPD

5. Repetir los pasos 1 al 4 para el servidor Proftpd y conectarse al servidor desde el móvil (mediante la app FTP Manager o cualquier cliente gratuito). También nos podríamos conectar desde el CMD de Windows.

### 1. Instalar un servidor ftp en Linux.

- a. ¿Cómo funciona y cómo se instala?
  - i. Lo lanza el súper demonio de red llamado **inetd**.
  - ii. Este súper demonio se instala con la orden **sudo apt install openbsd-inetd**
  - iii. Después se instala el demonio servidor FTP con la orden **sudo apt install proftpd-core**
- b. Arrancar el servidor FTP (que en realidad es arrancar el súper demonio de red) con la orden **sudo /etc/init.d/proftpd restart**
- c. ¿Cómo puedo saber si está arrancado o no?
  - i. Mediante la orden systemctl **sudo systemctl status proftpd**
  - ii. Mirando los procesos que están ejecutándose en el sistema mediante la orden **ps aux | grep ftp**
  - iii. Mediante nmap

```
(diego@kali)-[/etc]
$ nmap -p 21 172.20.10.2
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-03 16:14 CEST
Nmap scan report for 172.20.10.2
Host is up (0.00043s latency).

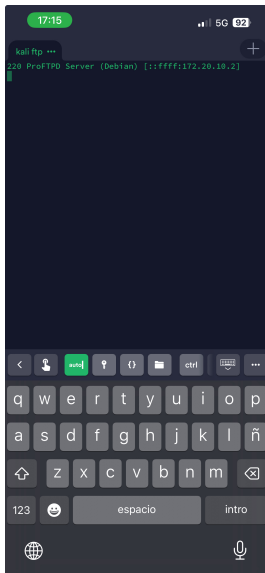
PORT      STATE SERVICE
21/tcp    open  ftp

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

- iv. Mediante cualquier otro método alternativo **desde el terminal mac: ftp (ip del Kali linx)**
- d. ¿Qué puerto utiliza este servidor? **21**

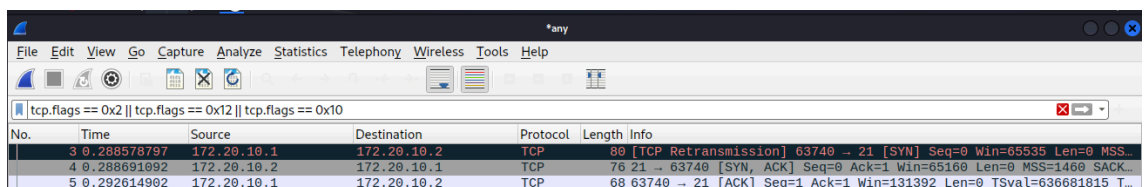
2. Instalar un cliente ftp en móvil, por ejemplo Termius, en iOS o en Android (también podríamos conectarnos mediante el CMD en Windows, aunque en este caso hay que habilitarlo previamente porque está deshabilitado por defecto).

- a. Conectarse desde el móvil al servidor FTP



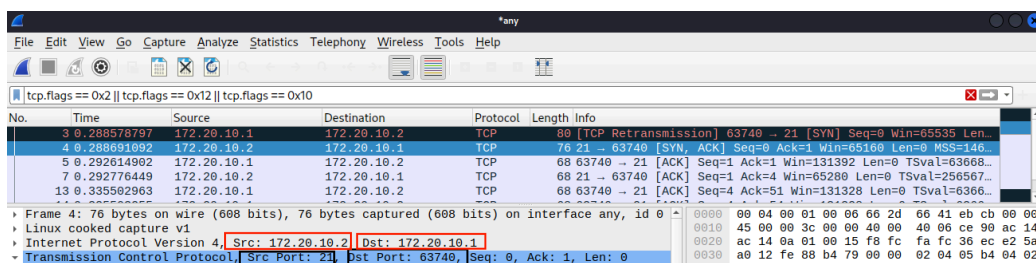
### 3. Monitorización con un sniffer

- a. Una vez que se ha logrado la conexión cliente-servidor, capturar mediante el wireshark los siguientes paquetes:
  - i. Three-way handshake – Tres paquetes con los que se establece la conexión cliente-servidor



No.	Time	Source	Destination	Protocol	Length	Info
3	0.288578797	172.20.10.1	172.20.10.2	TCP	80	[TCP Retransmission] 63740 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=...
4	0.288691892	172.20.10.2	172.20.10.1	TCP	76	21 → 63740 [SYN, ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK...
5	0.292614902	172.20.10.1	172.20.10.2	TCP	68	63740 → 21 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=636681815 T...

- ii. donde aparezcan los puertos del cliente y el servidor, junto con sus direcciones IP.



No.	Time	Source	Destination	Protocol	Length	Info
3	0.288578797	172.20.10.1	172.20.10.2	TCP	80	[TCP Retransmission] 63740 → 21 [SYN] Seq=0 Win=65535 Len=0 MSS=...
4	0.288691892	172.20.10.2	172.20.10.1	TCP	76	21 → 63740 [SYN, ACK] Seq=1 Ack=1 Win=65160 Len=0 MSS=1460 SACK...
5	0.292614902	172.20.10.1	172.20.10.2	TCP	68	63740 → 21 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=636681815 T...
7	0.292776449	172.20.10.2	172.20.10.1	TCP	68	21 → 63740 [ACK] Seq=1 Ack=4 Win=65280 Len=0 TSval=256567...
13	0.335562963	172.20.10.1	172.20.10.2	TCP	68	63740 → 21 [ACK] Seq=4 Ack=51 Win=131328 Len=0 TSval=6366...

Frame 4: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0

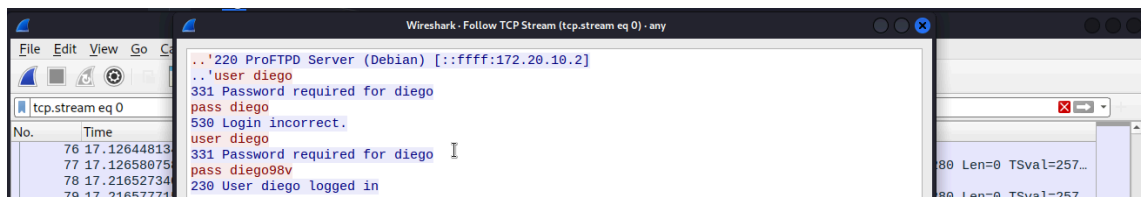
Ethernet II, Src: Linux cooked capture v1, Dst: 172.20.10.1

Internet Protocol Version 4, Src: 172.20.10.2, Dst: 172.20.10.1

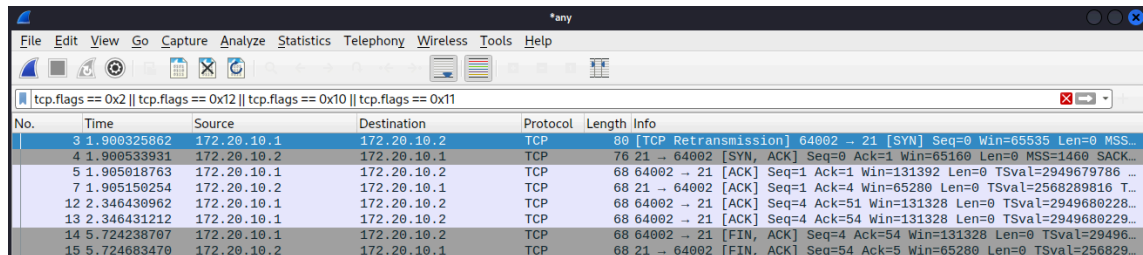
Transmission Control Protocol, Src Port: 21, Dst Port: 63740, Seq: 0, Ack: 1, Len: 0



### iii. USUARIO Y CONTRASEÑA



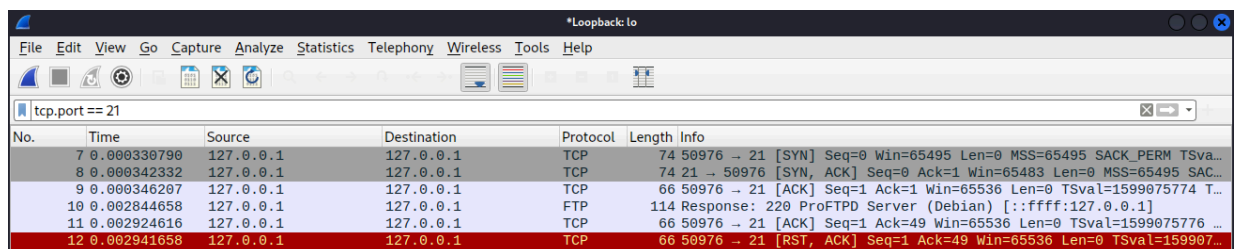
### iv. Capturar los paquetes con los que se cierra la conexión



**NOTA:** No serán válidas capturas que no estén bien filtradas de forma que se puedan encontrar de forma más o menos sencilla esos paquetes. Destacar esos paquetes con un trazo rojo ayuda, pero eso no es un filtro de Wireshark.

## 4. Escaneo de los puertos mediante nmap

- Hacer un escaneo **FULL SCAN** al servidor ftp mediante nmap
- Identificar mediante wireshark y los filtros necesarios los paquetes mandados en ese escaneo a ese puerto en concreto. Para ello, debe aparecer:
  - Un escaneo filtrado con éxito (a un puerto abierto)



- Un escaneo filtrado a un puerto cerrado

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 45

No.	Time	Source	Destination	Protocol	Length	Info
7	0.000302041	127.0.0.1	127.0.0.1	TCP	74	39518 → 45 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSva...
8	0.000306250	127.0.0.1	127.0.0.1	TCP	54	45 → 39518 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

**NOTA:** La única diferencia es que hay que hacer un escaneo HALF SCAN para el servidor TELNET y ver los paquetes (en especial los flags que están activados) que se mandan en este caso.

### PARTE 3: SSH

5. Realizar el paso 3 (captura del Three-way handshake, fin de conexión y usuario y contraseña) para el servidor Open SSH, con sus correspondientes pantallazos.

#### 1. Monitorización con un sniffer (SSH)

a. Una vez que se ha logrado la conexión cliente-servidor, capturar mediante el wireshark los siguientes paquetes:

i. Three-way handshake – Tres paquetes con los que se establece la conexión cliente-servidor

\*any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags == 0x2 || tcp.flags == 0x12 || tcp.flags == 0x10

No.	Time	Source	Destination	Protocol	Length	Info
5	1.863175942	172.20.10.1	172.20.10.2	TCP	80	[TCP Retransmission] 63990 → 22 [SYN] Seq=0 Win=65535 Len=...
6	1.863523931	172.20.10.2	172.20.10.1	TCP	76	22 → 63990 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=146...
7	1.881946560	172.20.10.1	172.20.10.2	TCP	68	63990 → 22 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=29656...

ii. donde aparezcan los puertos del cliente y el servidor, junto con sus direcciones IP.

63	2.446050924	172.20.10.1	172.20.10.2	TCP	68	63970 → 22 [ACK] Seq=2644 Ack=4321 Win=131072 Len=0 TSval=...
9	2.083507711	172.20.10.1	172.20.10.2	TCP	68	63970 → 22 [ACK] Seq=28 Ack=33 Win=131328 Len=0 TSval=342...
11	2.083507919	172.20.10.1	172.20.10.2	TCP	1466	63970 → 22 [ACK] Seq=28 Ack=33 Win=131328 Len=1398 TSval=...
71	3.945372580	172.20.10.1	172.20.10.2	TCP	68	63970 → 22 [ACK] Seq=2820 Ack=4529 Win=130816 Len=0 TSval=...

Frame 4: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface any, id 0

Ethernet II, Src: Linux cooked capture v1 (00:00:00:00:00:00), Dst: 172.20.10.2

Internet Protocol Version 4, Src: 172.20.10.2, Dst: 172.20.10.1

Transmission Control Protocol, Src Port: 22, Dst Port: 63970, Seq: 0, Ack: 1, Len: 0

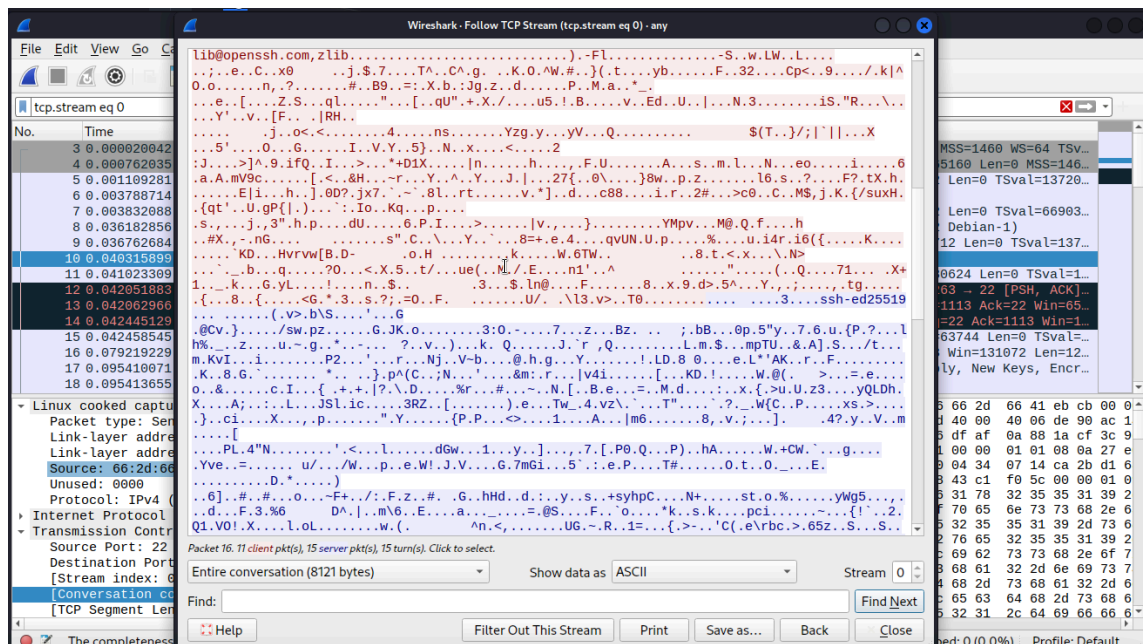
Source Port: 22

Destination Port: 63970

[Stream index: 0]

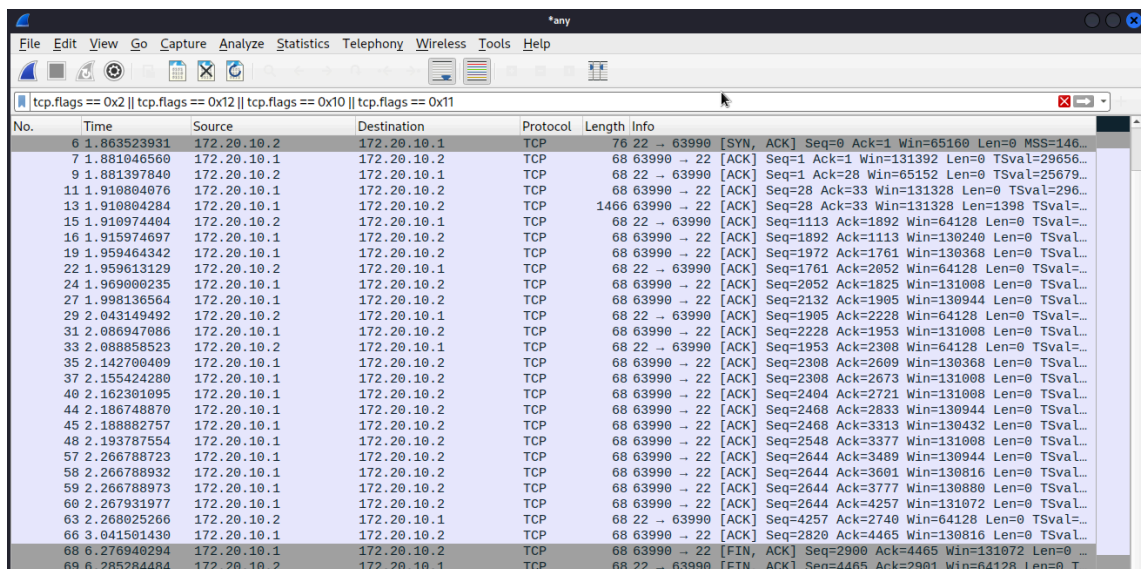


### iii. USUARIO Y CONTRASEÑA



Podemos observar que el paquete está completamente cifrado, lo que significa que ni la contraseña ni el nombre de usuario son visibles.

### iv. Capturar los paquetes con los que se cierra la conexión



## 2. Contesta a las siguientes preguntas:

- Explica en pocas líneas qué es el servicio SSH y para qué sirve

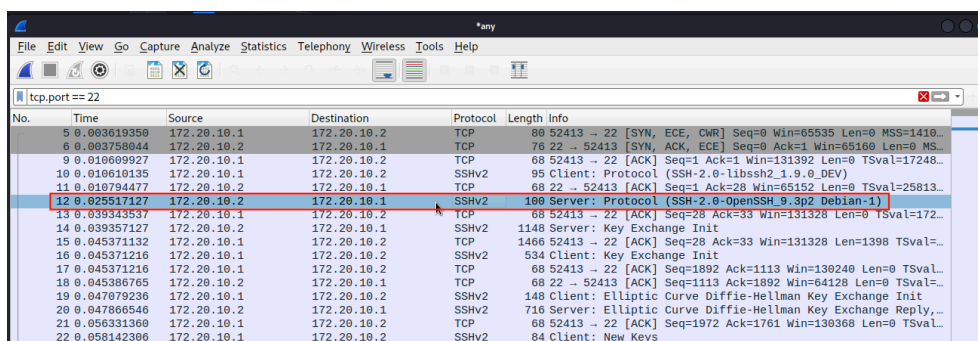
**SSH (Secure Shell) es un protocolo de red utilizado para establecer conexiones seguras y cifradas entre dispositivos a través de una red.**

Su principal objetivo es permitir la administración remota segura de sistemas y transferencia de datos de manera protegida. SSH se utiliza comúnmente para acceder y gestionar servidores y dispositivos de forma remota, así como para transferir archivos de manera segura mediante SCP (Secure Copy) o SFTP (SSH File Transfer Protocol). Proporciona autenticación y cifrado para proteger la confidencialidad e integridad de la información transmitida a través de la red.

- b. ¿Qué puerto utiliza el servidor? ¿Qué puerto has utilizado tú como cliente? Usa el puerto 22. Nosotros hemos usado como cliente el puerto 63990
- c. ¿Cómo es su autenticación y cómo viajan los datos que se intercambian entre el cliente y el servidor? Explica en unas pocas líneas y con tus palabras qué método utilizamos para autenticarnos y qué algoritmo de cifrado de utiliza para ello.

SSH utiliza un par de claves para autenticación: una clave privada en el cliente y una clave pública en el servidor. Cuando un cliente se conecta, el servidor verifica la autenticidad solicitando una firma digital con la clave privada. El cifrado de datos utiliza algoritmos simétricos como AES, y el intercambio de claves se hace con algoritmos asimétricos como RSA o DSA para garantizar la seguridad de la comunicación. Esto asegura una conexión segura en SSH.

- d. ¿Has podido ver la contraseña en Wireshark? ¿Qué paquetes son los que definen la autenticación? Señálalos en tu captura.



No.	Time	Source	Destination	Protocol	Length	Info
5	0.003619350	172.28.10.1	172.28.10.2	TCP	60	52413 → 22 [SYN, ECE, CWR] Seq=0 Win=65535 Len=0 MSS=1418...
6	0.003758044	172.28.10.2	172.28.10.1	TCP	76	22 → 52413 [SYN, ACK, ECE] Seq=0 Ack=1 Win=65160 Len=0 MS...
9	0.016009927	172.28.10.1	172.28.10.2	TCP	68	52413 → 22 [ACK] Seq=1 Ack=1 Win=131392 Len=0 TSval=17248...
10	0.016610135	172.28.10.1	172.28.10.2	SSHv2	95	Client: Protocol (SSH-2.0-libssh2.1.9.0_DEV)
11	0.016794477	172.28.10.2	172.28.10.1	TCP	68	22 → 52413 [ACK] Seq=1 Ack=28 Win=65152 Len=0 TSval=25813...
12	0.025517127	172.28.10.2	172.28.10.1	SSHv2	108	Server: Protocol (SSH-2.0-OpenSSH_9.3p2 Debian-1)
13	0.039343537	172.28.10.1	172.28.10.2	TCP	68	52413 → 22 [ACK] Seq=28 Ack=33 Win=131328 Len=0 TSval=172...
14	0.039357127	172.28.10.2	172.28.10.1	SSHv2	1148	Server: Key Exchange Init
15	0.045371132	172.28.10.1	172.28.10.2	TCP	1466	52413 → 22 [ACK] Seq=28 Ack=33 Win=131328 Len=1398 TSval=...
16	0.045371216	172.28.10.1	172.28.10.2	SSHv2	534	Client: Key Exchange Init
17	0.045371216	172.28.10.1	172.28.10.2	TCP	68	52413 → 22 [ACK] Seq=1892 Ack=1113 Win=130240 Len=0 TSval=...
18	0.045386765	172.28.10.2	172.28.10.1	TCP	68	22 → 52413 [ACK] Seq=1113 Ack=1892 Win=64128 Len=0 TSval=...
19	0.047879236	172.28.10.1	172.28.10.2	SSHv2	148	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
20	0.047866546	172.28.10.2	172.28.10.1	SSHv2	716	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply,...
21	0.056331369	172.28.10.1	172.28.10.2	TCP	68	52413 → 22 [ACK] Seq=1972 Ack=1761 Win=130368 Len=0 TSval=...
22	0.058142306	172.28.10.1	172.28.10.2	SSHv2	84	Client: New Keys



e. ¿Hay un servicio análogo para el servicio ftp basado en SSH?

**Sí, existe un servicio similar al FTP llamado SFTP (SSH File Transfer Protocol) que utiliza SSH para una transferencia de archivos segura, cifrando toda la comunicación, incluyendo las credenciales y los datos transferidos, a diferencia del FTP estándar que transmite datos en texto plano. SFTP es ampliamente utilizado para garantizar la seguridad en la transferencia de archivos.**

**IMPORTANTE: Si se detecta que este último punto está copiado la práctica será calificada con un 0. Tenéis que leerlo, entenderlo y explicarlo con vuestras palabras.**

## **INSTRUCCIONES**

- Entrega:
  - **Un archivo PDF** a partir de este documento de Word modificado con las respuestas escritas (las que están señaladas en rojo) y los pantallazos pedidos.
- Los ejercicios **SÓLO** podrán realizarse en grupos de dos alumnos como máximo. Si hay un grupo de tres se debe escribir un correo al profesor para notificárselo. **No se permiten entregas de prácticas por grupos de tres o más alumnos que no hayan sido notificadas en fecha al profesor.**
- El nombre del fichero entregado serán los apellidos de los alumnos separados por guion.
- Se deberán usar al menos dos equipos diferentes (cliente y servidor) o realizarlo mediante máquinas virtuales.
- **La fecha límite de entrega será el miércoles 11 de octubre a las 23 horas.**
- No se recogerán memorias entregadas fuera de fecha o por otro medio distinto de los indicados (como por ejemplo el mail). Debe entregarse en el apartado correspondiente en el campus virtual.