

Practica 6: Análisis de Protocolos y TLS

1. ¿Cuál es la dirección IP del cliente y cuál es la del servidor?

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.12.1	192.168.12.2	TCP	58	34906 → 23 [SYN]
2	0.003264	192.168.12.2	192.168.12.1	TCP	58	23 → 34906 [SYN,
3	0.009006	192.168.12.1	192.168.12.2	TCP	54	34906 → 23 [ACK]
4	0.012542	192.168.12.2	192.168.12.1	TELNET	66	Telnet Data ...
5	0.013252	192.168.12.1	192.168.12.2	TELNET	66	Telnet Data ...
6	0.013318	192.168.12.1	192.168.12.2	TCP	54	[TCP Dup ACK 3#1]
7	0.014439	192.168.12.2	192.168.12.1	TELNET	96	Telnet Data ...
8	0.016232	192.168.12.2	192.168.12.1	TELNET	57	Telnet Data ...

El cliente inicia la conexión (192.168.12.1)

El servidor será el destino (192.168.12.2)

2. ¿Qué credenciales se han utilizado para acceder al servidor?

- PISTA: En esta captura TELNET, el cliente TELNET envía un solo carácter por mensaje en la mayoría de las tramas.

Filtramos por telnet y le damos a Follow stream en cualquier paquete:

Follow TCP: tcp.stream eq 0 in telnet-client-server.pcapng

Show only this stream | Filter out this stream

```
.....
.....!

User Access Verification
Password:
..
..!
..!...
...
...
...P...
...
..
C
i
s
c
o

R2>
e
e
x
x
```

La contraseña es: cisco

3. ¿Qué tipo de sistema es el servidor?
4. ¿Qué comando(s) ha ejecutado el cliente en el servidor?

El prompt R2 nos indica q es dispositivo basado en Cisco IOS como un enrutador o switch.
Ejecuta el comando 'exit'

Follow TCP: tcp.stream eq 0 in telnet-client-server.pcapng
[Show only this stream](#) | [Filter out this stream](#)



Entire Conversation ▾ ☒ ASCII ☐ Hex Dump ☒ Wrap long lines

⇌ Ladder Diagram Open in new window Done

también se puede filtrar y mirar por el cliente:

telnet && ip.src == 192.168.12.1 ✓ Apply Clear F

El mismo adversario ha capturado también una trama de una comunicación entre un cliente y un servidor FTP

<https://www.cloudshark.org/captures/abdc8742488f>

Se pide responder a las mismas preguntas que se planteaban para el protocolo anterior. Intenta entender qué ha hecho el cliente.

1. ¿Cuál es la dirección IP del cliente y cuál es la del servidor?

Podemos mirarlo en el follow tcp stream, los clientes usaran los comandos USER,ETC, y el servidor se verá reflejado:

Follow TCP: tcp.stream eq 0 in ftp.pcap

Show only this stream | Filter out this stream

```
220 ProFTPD 1.3.0a Server (ProFTPD Anonymous Server) [192.168.1.231]
USER ftp
331 Anonymous login ok, send your complete email address as your password.
PASS ftp
230 Anonymous access granted, restrictions apply.
SYST
215 UNIX Type: L8
FEAT
211-Features:
  MDTM
  REST STREAM
  SIZE
211 End
PWD
257 "/" is current directory.
EPSV
229 Entering Extended Passive Mode (|||58612|)
```

server: 192.168.1.231

cliente: 192.168.1.182

No.	Time	Source	Destination	Protocol	Length	Info
4	0.040555	192.168.1.231	192.168.1.182	FTP	136	Response: 220
6	2.522199	192.168.1.182	192.168.1.231	FTP	76	Request: USER
8	2.641952	192.168.1.231	192.168.1.182	FTP	142	Response: 331
10	4.098171	192.168.1.182	192.168.1.231	FTP	76	Request: PASS
12	6.070595	192.168.1.231	192.168.1.182	FTP	117	Response: 230
14	6.071130	192.168.1.182	192.168.1.231	FTP	72	Request: SYST
16	6.071624	192.168.1.231	192.168.1.182	FTP	85	Response: 215
18	6.071927	192.168.1.182	192.168.1.231	FTP	72	Request: FEAT
19	6.094682	192.168.1.231	192.168.1.182	FTP	106	Response: 211
20	6.094752	192.168.1.231	192.168.1.182	FTP	75	Response: 211
23	6.095187	192.168.1.182	192.168.1.231	FTP	71	Request: PWD
25	6.150683	192.168.1.231	192.168.1.182	FTP	97	Response: 257
27	9.394257	192.168.1.182	192.168.1.231	FTP	72	Request: EPSV
29	9.395002	192.168.1.231	192.168.1.182	FTP	114	Response: 229
31	9.395696	192.168.1.182	192.168.1.231	FTP	72	Request: LIST

Frame 4: 136 bytes on wire (1088 bits) : 136 bytes captured (1088 bits)

2. ¿Qué credenciales se han utilizado para acceder al servidor?

- a. PISTA: En esta captura TELNET, el cliente TELNET envía un solo carácter por mensaje en la mayoría de las tramas.

user: ftp

password: ftp

Follow TCP: tcp.stream eq 0 in ftp.pcap

Show only this stream | Filter out this stream

```
220 ProFTPD 1.3.0a Server (ProFTPD Anonymous Server) [192.168.1.231]
USER ftp
331 Anonymous login ok, send your complete email address as your password.
PASS ftp
230 Anonymous access granted, restrictions apply.
```

3. ¿Qué tipo de sistema es el servidor?
4. ¿Qué comando(s) ha ejecutado el cliente en el servidor?

3.- UNIX Type: L8

Follow TCP: tcp.stream eq 0 in ftp.pcap

[Show only this stream](#) | [Filter out this stream](#)

```
220 ProFTPD 1.3.0a Server (ProFTPD Anonymous Server) [192.168.1.231]
USER ftp
331 Anonymous login ok, send your complete email address as your password.
PASS ftp
230 Anonymous access granted, restrictions apply.
SYST
215 UNIX Type: L8
FEAT
211-Features:
  MDTM
  REST STREAM
  SIZE
211 End
PWD
257 "/" is current directory.
EPSV
229 Entering Extended Passive Mode (|||58612|)
LIST
150 Opening ASCII mode data connection for file list
226 Transfer complete
```

4.-Miramos los comandos rojos:

USER, PASS, SYST, FEAT, PWD, EPSV, LIST, TYPE, SIZE, RETR, MDTM, CWD, PWD, STOR, MKD

EJERCICIO 2: El protocolo TLS

En el Campus Virtual tienes a tu disposición un código fuente de Python para crear un servidor HTTPS (HTTP sobre TLS 1.3²) en tu equipo. Este servidor utilizará un certificado autofirmado creado en el programa XCA, visto en las prácticas anteriores. Dicho certificado debe estar habilitado para funcionar como servidor TLS/SSL:

- Sujeto: Datos del alumno/a,
- Plantilla: TLS (o SSL) server,
- Uso de la clave con opciones: Digital Signature, Non-Repudiation, Key Encipherment, Key Agreement, y TLS (o SSL) Web Server Authentication,
- Opciones Netscape: SSL Server.

Ejercicio 2.1. Dado el código fuente con los certificados creados por el alumno/a, capturar la comunicación con el servidor usando Wireshark, y contestar a las siguientes preguntas:

1. ¿Cuándo (de qué trama a qué trama) se procede con el proceso de handshake (sesión SSL), tal y como se ha explicado en teoría?
2. En esta conexión se utiliza TLS1.3. ¿Dónde se negocia exactamente la versión de TLS que se utiliza?
3. En la parte del cliente, ¿en qué trama se puede ver las suites de cifrado que soporta el cliente? ¿Cuáles son?
4. ¿Qué suite de cifrado se acepta finalmente para el proceso de conexión?
5. En TLS1.3, no es posible ver la trama en la que se envía el certificado digital del servidor. ¿Por qué ocurre eso?
 - Adicionalmente, de forma opcional: ¿Sería posible inferir cuál es la trama en la que el servidor envía al cliente su certificado?

Para dar respuesta a estas preguntas, pueden utilizarse tanto las transparencias de teoría como la web <https://tls13.xargs.org/>

EN TLS 1.3:

1.-Handshake:

-Se busca el *Client Hello* y se rastrea el intercambio de mensajes hasta que se complete el *handshake* (aparezca un mensaje de tipo *Finished*).

-Tramas:

Client Hello

Server Hello

--Mensajes de configuración de claves (*Key Share*, *Finished*, ...)

2.-La versión de TLS se negocia en la trama del Server Hello.

En esta trama se especifica que se usará TLS1.3

3.-Se examina la trama del Client Hello, en el campo Cipher Suites podemos ver los detalles de la trama.

4.-La Suite de cifrado aceptada se encuentra en el Server Hello, en esa trama se busca el Cipher Suite.

5.-En TLS 1.3 el certificado se cifra utilizando las claves derivadas del handshake inicial, por eso no se pueden ver sus tramas.

TLS 1.2:

```
import http.server
import ssl

# Create an HTTP server instance in port 4443 (access it through https://localhost)
server_address = ('localhost', 4443)
httpd = http.server.HTTPServer(server_address, http.server.SimpleHTTPRequestHandler)

# Wrap the socket with the latest TLS encryption (ssl.PROTOCOL_TLS_SERVER) and use
context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
context.load_cert_chain('server.crt', 'key.pem')
httpd.socket = context.wrap_socket(httpd.socket, server_side=True)

# Start the HTTPS server and keep it forever until finishing the process
httpd.serve_forever()
```

1.-Al igual que en TLS 1.3, aunque en este se pueden incluir más pasos y mensajes como: Client Hello, Server Hello, Envío del certificado digital del servidor, Server Key Exchange,....

2.- La negociación se puede ver en Server Hello

3.-Las Suites de cifrado del cliente están en Client Hello como en el TLS1.3 en cipher suite

4.- en server hello, cipher suite

5. ¿En qué trama se envía el certificado digital del servidor? En esa trama, ¿Dónde se encuentra vuestro nombre (el “common name” cuando creasteis el certificado)? ¿Cuál es la clave pública del servidor?
6. ¿El servidor se autentica al cliente? ¿Y el cliente al servidor?

5.-en donde ponga el mensaje Certificate, el nombre común(cn) esta en el campo subject
la clave pública del servidor en el campo public key

6.- normalmente puede autenticar el servidor al cliente mediante su certificado digital
-si la configuración exige autenticación mutua el cliente debe enviar su propio certificado
mensaje: certificate(Client)