

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

User: ftp Password: ftp

7. ¿Qué tipo de sistema es el servidor?

En este caso el cliente escribe el comando SYST para solicitar información al servidor sobre el tipo de sistema que es.

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 UNIX Type: L8

8. ¿Qué comando(s) ha ejecutado el cliente en el servidor?

SYST, FEAT, PWD, EPSV, LIST, TYPE, SIZE, RETR, MDTM, CWD, STOR, MKD, SITE, QUIT

Ejercicio 2.1

106	23.717543	127.0.0.1	127.0.0.1	TLSv1.3	573	Client Hello
108	23.717988	127.0.0.1	127.0.0.1	TLSv1.3	573	Client Hello
110	23.724237	127.0.0.1	127.0.0.1	TLSv1.3	1487	Server Hello, Change Cipher Spec, Application Data, Application Data, App...

9. ¿Cuándo (de qué trama a qué trama) se procede con el proceso de handshake (sesión SSL), tal y como se ha explicado en teoría?

De la trama 106 a la 110

10. En esta conexión se utiliza TLS1.3. ¿Dónde se negocia exactamente la versión de TLS que se utiliza?

Se negocia en el intercambio de mensajes *Client Hello* y *Server Hello*.

- El cliente inicia la conexión enviando un mensaje "Client Hello" al servidor cuando quiere establecer una conexión segura. Envía las versiones que soporta de TLS, junto con otros elementos tales como extensiones que admite y las suites de cifrado preferidas o su clave pública o *nonce* para la posterior creación de la clave de sesión.

- El servidor selecciona la versión de TLS y responderá con un mensaje "Server Hello" confirmando la versión y puede aceptar o ajustar las preferencias del cliente en términos de suites de cifrado y otras configuraciones.

NOTA

Es importante destacar que en TLS 1.3, la versión de TLS se especifica en el campo **legacy_version** dentro de los mensajes *ClientHello* y *ServerHello*. Aunque el nombre del campo incluye la palabra "legacy", este es el campo utilizado para indicar la versión de TLS en esta versión del protocolo.

Por ejemplo, el campo **legacy_version** contendrá **0x0304** para indicar TLS 1.3. El valor **0x03** indica TLS y **0x04** indica la versión específica, en este caso, TLS 1.3.

**11. En la parte del cliente, ¿en qué trama se puede ver las suites de cifrado que soporta el cliente?
¿Cuáles son?**

En la misma trama 108 por ejemplo.

108	23.717988	127.0.0.1	127.0.0.1	TLSv1.3	573 Client Hello
110	23.724237	127.0.0.1	127.0.0.1	TLSv1.3	1487 Server Hello, Change Cipher Spec, Appli

▼ Cipher Suites (16 suites)

Cipher Suite: Reserved (GREASE) (0x6a6a)
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02f)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc030)
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca8)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
Compression Methods Length: 1

12. ¿Qué suite de cifrado se acepta finalmente para el proceso de conexión?

110	23.724237	127.0.0.1	127.0.0.1	TLSv1.3	1487 Server Hello, Change Cipher Spec, Appli
113	23.725450	127.0.0.1	127.0.0.1	TLSv1.3	74 Change Cipher Spec, Application Data

Length: 122

▼ Handshake Protocol: Server Hello

Handshake Type: Server Hello (2)
Length: 118
Version: TLS 1.2 (0x0303)
Random: 784c299677edc89d797884b6a90d10be1e331584cbfff31119cbf4799e1b4081
Session ID Length: 32
Session ID: 2f386282d59ca378f582957d162e50f8b46736fa77c952cbb7328d108df893db
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
Compression Method: null (0)

13. En TLS1.3, no es posible ver la trama en la que se envía el certificado digital del servidor. ¿Por qué ocurre eso?

Porque el servidor responde en “Server Hello” con un mensaje *Certificate* que contiene el certificado digital del servidor, pero cifrado con la clave pública del cliente.

• **Adicionalmente, de forma opcional: ¿Sería posible inferir cuál es la trama en la que el servidor envía al cliente su certificado?**

En TLS 1.3 el mensaje *Certificate* normalmente se envía después del mensaje *Server Hello* y antes del intercambio de claves *ChangeCipherSpec*.

Ejercicio 2.2

14. ¿Cuándo (de qué trama a qué trama) se procede con el proceso de handshake (sesión SSL), tal y como se ha explicado en teoría?

Desde las tramas 70 hasta 138

70	7.188094	127.0.0.1	127.0.0.1	TLSv1.2	561 Client Hello
72	7.190644	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key Exchange, Server Hello Done
77	7.191700	127.0.0.1	127.0.0.1	TLSv1.2	573 Client Hello
79	7.191850	127.0.0.1	127.0.0.1	TLSv1.2	51 Alert (Level: Fatal, Description: Certificate Unknown)
85	7.193512	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key Exchange, Server Hello Done
87	7.193730	127.0.0.1	127.0.0.1	TLSv1.2	51 Alert (Level: Fatal, Description: Certificate Unknown)
98	7.443982	127.0.0.1	127.0.0.1	TLSv1.2	561 Client Hello
100	7.445356	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key Exchange, Server Hello Done
102	7.445602	127.0.0.1	127.0.0.1	TLSv1.2	51 Alert (Level: Fatal, Description: Certificate Unknown)
113	7.500527	127.0.0.1	127.0.0.1	TLSv1.2	573 Client Hello
115	7.501913	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key Exchange, Server Hello Done
117	7.502437	127.0.0.1	127.0.0.1	TLSv1.2	137 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
119	7.507211	127.0.0.1	127.0.0.1	TLSv1.2	286 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
121	7.507636	127.0.0.1	127.0.0.1	TLSv1.2	752 Application Data
123	7.532784	127.0.0.1	127.0.0.1	TLSv1.2	228 Application Data
125	7.532976	127.0.0.1	127.0.0.1	TLSv1.2	897 Application Data
134	7.748055	127.0.0.1	127.0.0.1	TLSv1.2	813 Client Hello
136	7.748424	127.0.0.1	127.0.0.1	TLSv1.2	185 Server Hello, Change Cipher Spec, Encrypted Handshake Message
138	7.749745	127.0.0.1	127.0.0.1	TLSv1.2	95 Change Cipher Spec, Encrypted Handshake Message
144	10.675086	127.0.0.1	127.0.0.1	TLSv1.2	177 Application Data
146	10.675158	127.0.0.1	127.0.0.1	TLSv1.2	177 Application Data
147	10.675162	127.0.0.1	127.0.0.1	TLSv1.2	177 Application Data
150	10.675690	127.0.0.1	127.0.0.1	TLSv1.2	257 Application Data
152	10.676591	127.0.0.1	127.0.0.1	TLSv1.2	414 Application Data

15. En esta conexión se utiliza TLS1.2. ¿Dónde se negocia exactamente la versión de TLS que se utiliza?

Igual que en TLS1.3, en los mensajes *Client Hello* y *Server Hello*.

16. En la parte del cliente, ¿en qué trama se puede ver las suites de cifrado que soporta el cliente? ¿Cuáles son?

70	7.188094	127.0.0.1	127.0.0.1	TLSv1.2	561 Client Hello
72	7.190644	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate,
77	7.191700	127.0.0.1	127.0.0.1	TLSv1.2	573 Client Hello
Session ID: c15e48ecabacab249be002f85e6493ae6a13ee5d2317f58d73b7862927f2957					
Cipher Suites Length: 32					
Cipher Suites (16 suites)					
Cipher Suite: Reserved (GREASE) (0x5a5a)					
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)					
Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)					
Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)					
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)					
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)					
Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)					
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)					
Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0a9)					
Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0aa)					
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)					
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)					
Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)					
Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)					
Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)					
Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)					

17. ¿Qué suite de cifrado se acepta finalmente para el proceso de conexión?

115	7.501913	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key E
117	7.502437	127.0.0.1	127.0.0.1	TLSv1.2	137 Client Key Exchange, Change Cipher Spec,
119	7.507211	127.0.0.1	127.0.0.1	TLSv1.2	286 New Session Ticket, Change Cipher Spec,
121	7.507636	127.0.0.1	127.0.0.1	TLSv1.2	752 Application Data
123	7.532784	127.0.0.1	127.0.0.1	TLSv1.2	228 Application Data
125	7.532976	127.0.0.1	127.0.0.1	TLSv1.2	897 Application Data
127	7.540055	127.0.0.1	127.0.0.1	TLSv1.2	813 Client Hello
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)					

18. ¿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?

72	7.190644	127.0.0.1	127.0.0.1	TLSv1.2	1330 Server Hello, Certificate, Server Key E
77	7.191700	127.0.0.1	127.0.0.1	TLSv1.2	573 Client Hello
79	7.191850	127.0.0.1	127.0.0.1	TLSv1.2	513 Client Hello
Length: 897					
Handshake Protocol: Certificate					
Handshake Type: Certificate (11)					
Length: 893					
Certificates Length: 890					
Certificates (890 bytes)					
Certificate Length: 887					
Certificate: 308203733082025ba003020102020809b8dc0d1ede20c4300d06092a864886f70d01010b... (i					
signedCertificate					
version: v3 (2)					
serialNumber: 0x09b8dc0d1ede20c4					
signature (sha256WithRSAEncryption)					
issuer: rdnSequence (0)					
rdnSequence: 1 item (id-at-commonName=Santiago Ponce Arrocha)					
RelativeDistinguishedName item (id-at-commonName=Santiago Ponce Arrocha)					
Object Id: 2.5.4.3 (id-at-commonName)					
DirectoryString: printableString (1)					
printableString: Santiago Ponce Arrocha					
DirectoryString: printableString (1)					
printableString: Santiago Ponce Arrocha					
validity					
subject: rdnSequence (0)					
subjectPublicKeyInfo					
algorithm (rsaEncryption)					
subjectPublicKey: 3082010a02010100c3fb91b2fb936a7c6723e09c0fbbf9ac26b3525e2c0ece1582a3f5...					
modulus: 0x00c3fb91b2fb936a7c6723e09c0fbbf9ac26b3525e2c0ece1582a3f5172bcb230482fbb3...					
publicExponent: 65537					

19. ¿El servidor se autentica al cliente? ¿Y el cliente al servidor?

Nuestro servidor en todo momento envía su certificado para autenticarse. En el caso del cliente en ningún momento envía un certificado.