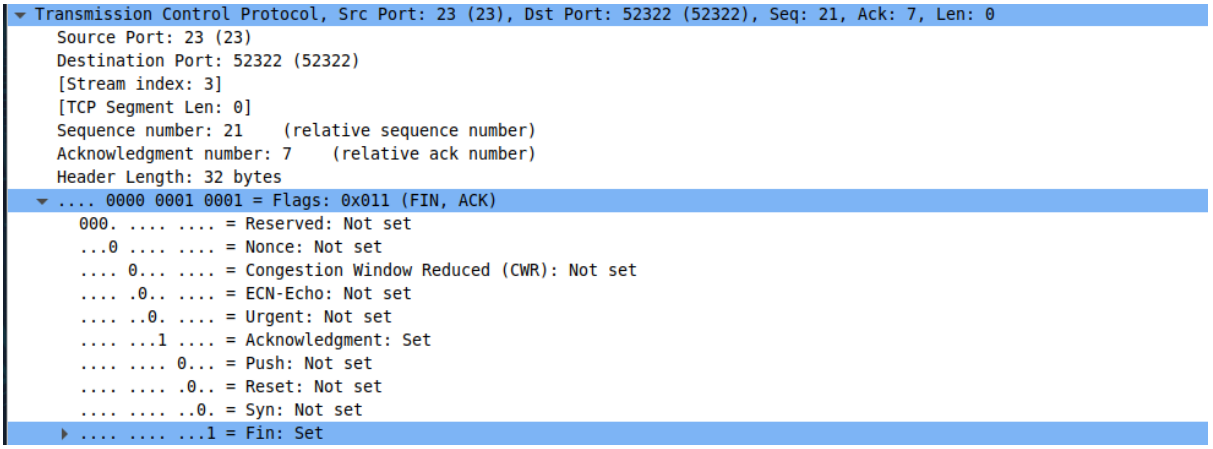
1.-
Quan volem identificar l'establiment d'una connexió TCP a Wireshark, podem fer servir el filtre tcp.flags.syn == 1 && tcp.flags.ack == 0, que ens mostra els paquets inicials on s'envia el bit SYN sense l'ACK, indicant el començament de la connexió. Això correspon al primer pas del handshake de tres vies. SYN, SYN-ACK i ACK, podem aplicar el filtre tcp.flags.syn == 1 || tcp.flags.ack == 1, cosa que ens permet veure tots els paquets relacionats i comprendre com s'estableix la connexió.



2.-
Una connexió TCP passa per diversos estats, començant amb l'establiment, continuant amb la transferència de dades i finalitzant el tancament. El procés s'inicia a l'estat CLOSED, on no hi ha connexió activa. Per establir la connexió, el client envia un paquet SYN i entra a SYN-SENT, mentre que el servidor respon amb un SYN-ACK, entrant a SYN-RECEIVED.

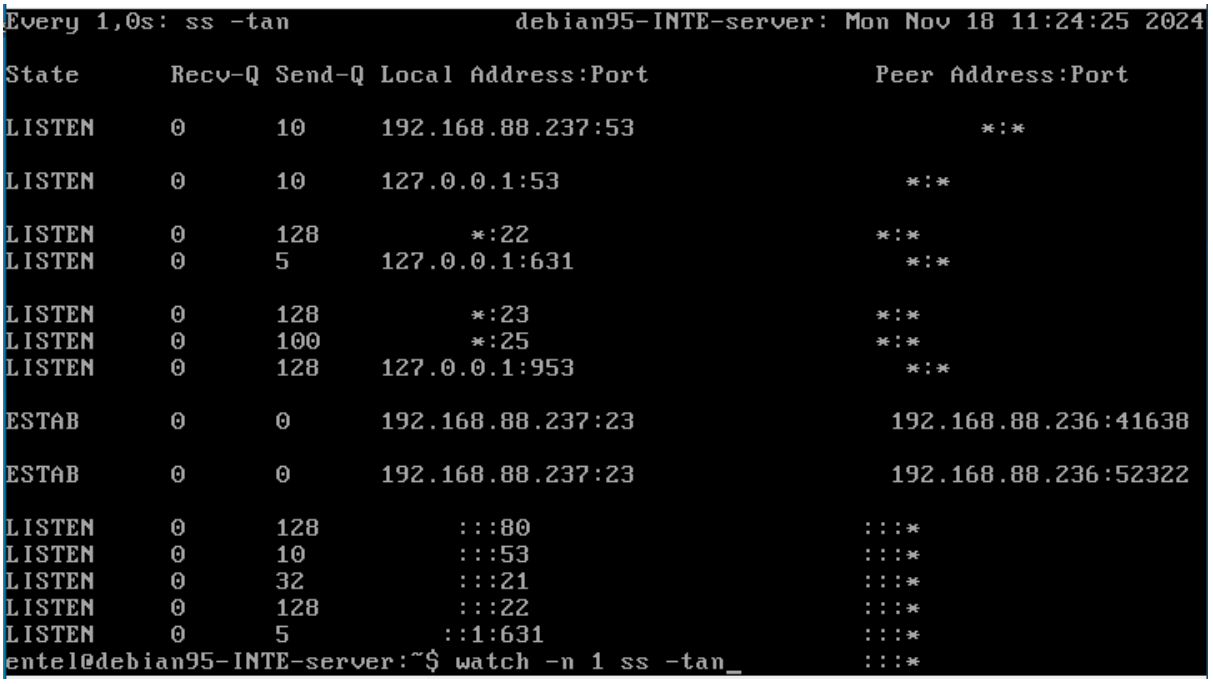
Quan el client confirma amb un ACK, tots dos costats passen a l'estat ESTABLISHED, que és on es fa la transferència de dades. Quan una de les parts decideix tancar la connexió, inicia el procés enviant un paquet FIN, entrant a FIN-WAIT-1. L'altra part respon amb un ACK, passant a CLOSE-WAIT, i eventualment envia el seu propi FIN. Això porta al primer costat a FIN-WAIT-2 i, en rebre l'ACK final, entra a TIME-WAIT per assegurar-se que no hi hagi paquets endarrerits. Finalment, ambdues parts arriben a l'estat CLOSED, completant el tancament de la connexió de manera segura.

3.-
Sí, el servidor pot acceptar connexions de múltiples clients simultàniament, i això s'aconsegueix gràcies a com el protocol TCP gestiona els sockets i les connexions a nivell de ports.

El servidor escolta en un port fix (el 23 per a Telnet) i roman en estat de “escolta” (LISTEN). Quan un client es connecta, el servidor accepta aquesta connexió i crea un socket únic per gestionar-la. Aquest socket únic està identificat per quatre valors: adreça IP i port del client, adreça IP i port del servidor.

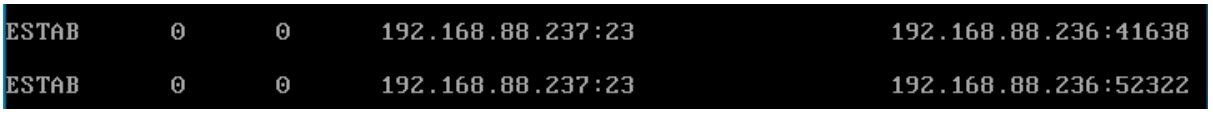
Això permet que múltiples clients es connectin simultàniament al mateix port del servidor (23 per Telnet), ja que cada connexió té una combinació única d'adreça IP i port del client. El port del servidor continua constant, però els ports d'origen dels clients són diferents.

L'eina ss mostra aquesta relació en temps real. Si executem una ordre com ss -tna, podem observar:



El port d'escolta del servidor per a Telnet (LISTEN 0 128 *:23).

Les connexions actives on veurem múltiples entrades amb el port del servidor fix (23), però diferents ports d'origen per a cada client.



Aquí, el servidor (192.168.88.237:23) té dues connexions establertes amb clients a diferents ports d'origen (41638, 52322). Podem veure com TCP utilitza diferents combinacions de ports i adreces per gestionar múltiples connexions simultànies sobre un mateix port de servidor.

4.-
Les comandes s'envien en text clar des del client cap a la consola del servidor. Això significa que les dades, incloses les credencials d'inici de sessió i les ordres executades, es transmeten sense xifrar a través de la xarxa.


```
▸ Frame 107: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface 0
▸ Ethernet II, Src: CadmusCo_99:2c:e5 (08:00:27:99:2c:e5), Dst: 6c:4b:90:c1:68:68 (6c:4b:90:c1:68:68)
▸ Internet Protocol Version 4, Src: 192.168.88.237 (192.168.88.237), Dst: 192.168.88.236 (192.168.88.236)
▾ Transmission Control Protocol, Src Port: 23 (23), Dst Port: 52322 (52322), Seq: 177, Ack: 26, Len: 1
    Source Port: 23 (23)
    Destination Port: 52322 (52322)
    [Stream index: 1]
    [TCP Segment Len: 1]
    Sequence number: 177    (relative sequence number)
    [Next sequence number: 178    (relative sequence number)]
    Acknowledgment number: 26    (relative ack number)
    Header Length: 32 bytes
  ▸ .... 0000 0001 1000 = Flags: 0x018 (PSH, ACK)
    Window size value: 905
    [Calculated window size: 905]
    [Window size scaling factor: -1 (unknown)]
  ▸ Checksum: 0xd6e2 [validation disabled]
    Urgent pointer: 0
  ▸ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  ▸ [SEQ/ACK analysis]
▾ Telnet
    Data: h
```

7.-
FTP utilitza dues connexions TCP: una per al control i una altra per a la transferència de dades La connexió de control s'estableix al port 21 i es manté sempre activa durant tota la sessió, ja que s'encarrega de gestionar les ordres entre el client i el servidor En canvi, la connexió de dades s'utilitza només quan cal transferir fitxers o directoris Depenent del mode (actiu o passiu), el client o el servidor estableix aquesta connexió en un port diferent. sempre activa, ja que només s'obre quan cal transferir informació i es tanca després de completar la transferència.

```
Every 1,0s: ss -tan              debian95-INTE-server: Mon Nov 18 11:42:00 2024

State      Recv-Q Send-Q Local Address:Port               Peer Address:Port

LISTEN     0      10     192.168.88.237:53                      *:*

LISTEN     0      10     127.0.0.1:53                        *:*

LISTEN     0      128          *:22                        *:*
LISTEN     0      5      127.0.0.1:631                       *:*

LISTEN     0      128          *:23                        *:*
LISTEN     0      100          *:25                        *:*
LISTEN     0      128    127.0.0.1:953                       *:*

LISTEN     0      128         :::80                       :::*
LISTEN     0      32          :::21                       :::*
LISTEN     0      10          :::53                       :::*
LISTEN     0      128         :::22                       :::*
LISTEN     0      5          ::1:631                      :::*
LISTEN     0      100         :::25                       :::*
LISTEN     0      128        ::1:953                      :::*
TIME-WAIT  0      0        ::ffff:192.168.88.237:20                  ::ffff:192
.168.88.236:60991
ESTAB      0      0        ::ffff:192.168.88.237:21                  ::ffff:192
```

8.-
A FTP, la responsabilitat d'establir la connexió de dades varia segons el mode d'operació. En mode actiu, el servidor és qui estableix la connexió de dades, ja que el client us indica el port on esteu escoltant i el servidor es connecta a aquest port des del seu port 20. En mode passiu, la situació canvia: el servidor obre un port aleatori i comunica al client a quin port s'ha de connectar i el client inicia la connexió de dades cap al port especificat pel servidor.

```
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
```

A la imatge anterior podem observar que el mètode d'operació és mode passiu.

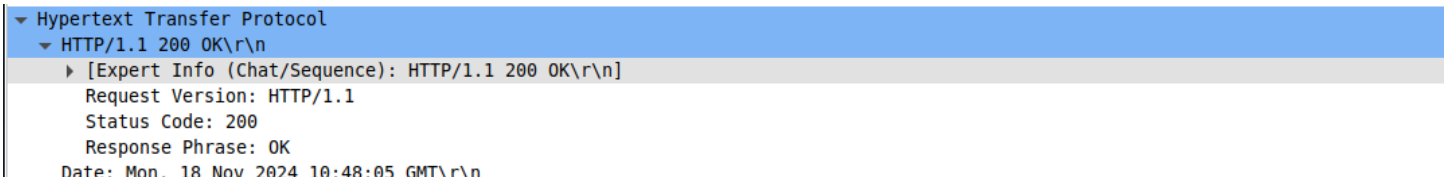
9.-

Sí, es poden fer diverses peticions HTTP en una mateixa connexió TCP. Amb HTTP/1.1, gràcies a Keep-Alive, la connexió es manté oberta per fer múltiples sol·licituds consecutives. HTTP/2 permet fer múltiples peticions simultànies amb multiplexing, millorant la velocitat i reduint la latència. Això redueix la sobrecàrrega, millora l'eficiència i disminueix la latència, sempre que client i servidor ho suportin.

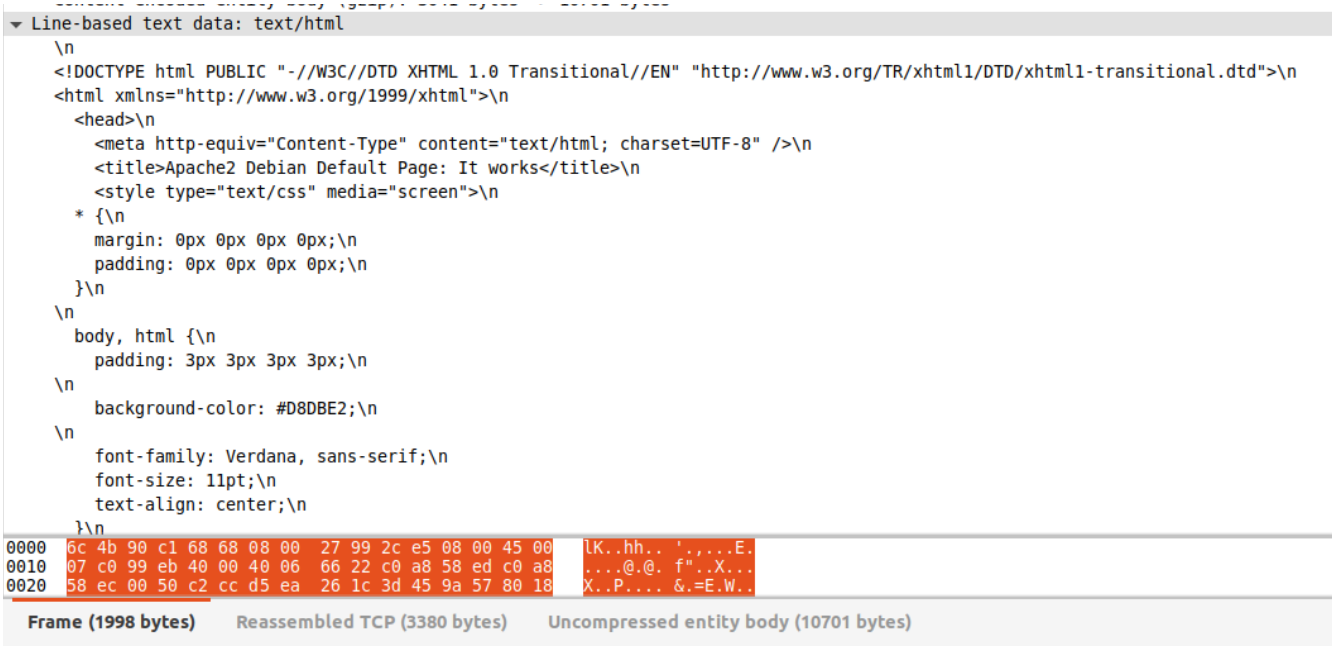
10.-
Es fa servir la versio HTPP/1.1

Els mètodes utilitzats son:
-GET: Per sol·licitar recursos.
-POST: Per enviar dades.
Existeixen altres mètodes menys utilitzats, com PUT, DELETE, HEAD o OPTIONS.

Els codis de resposta HTTP més comuns que pots observar són el 200 OK i el 204 No Content per a èxits, el 301 Moved Permanently, 302 Found i 304 Not Modified per a redireccions, el 400 Bad Request i 404 Not Found per a errors del client, i el 500 Internal Server Error i 503 Service Unavailable per a errors del servidor.



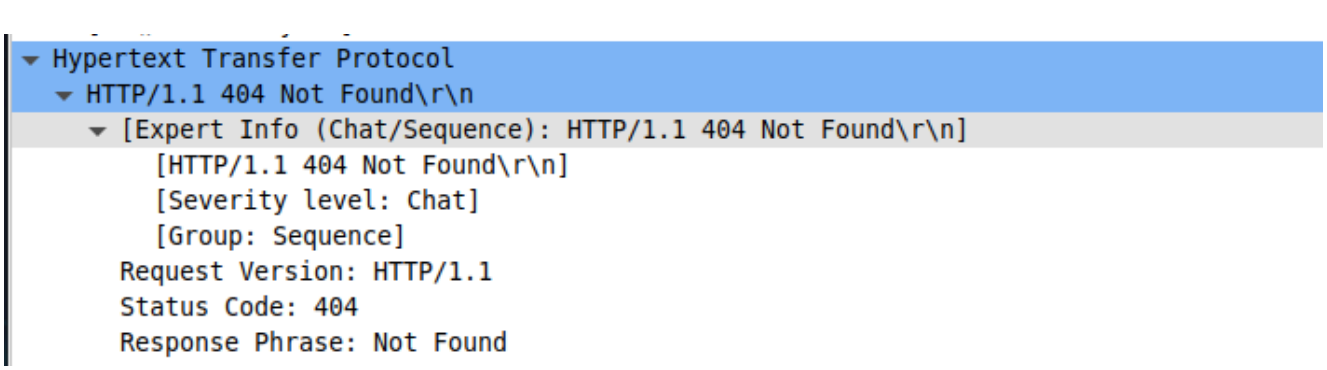
11.-



La pàgina per defecte del servidor Apache, esta composta per alguns recursos bàsics com HTML, imatges, fulls d'estil (CSS), links…

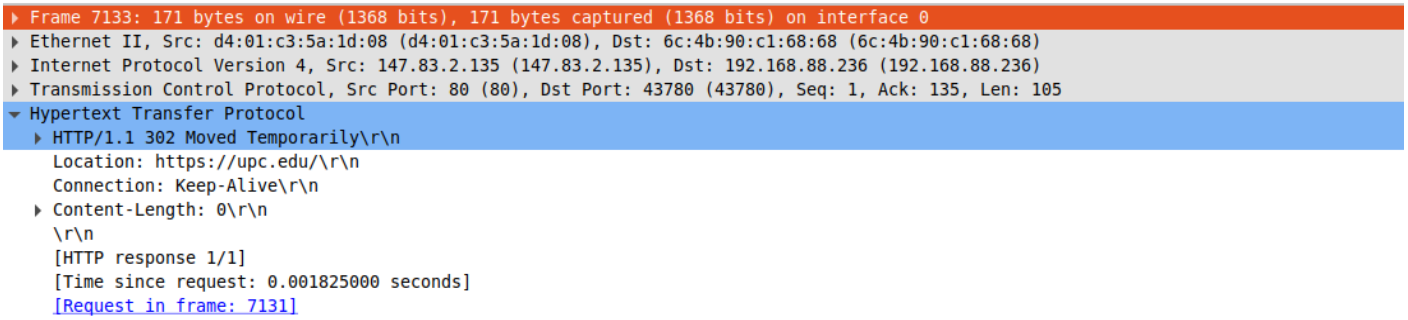
12.-

Quan es demana un recurs que no existeix al servidor, aquest retorna un error 404 Not Found. Això vol dir que el servidor ha processat la sol·licitud, però no ha trobat el fitxer o la pàgina que s'ha demanat.



13.-

Quan el servidor web d'Atenea ens passa d'una connexió HTTP a una HTTPS, utilitza el codi d'estat 302, conegut com "Moved Temporarily". Aquest codi vol dir que el recurs que hem demanat es troba temporalment en una altra adreça, que es detalla a l'encapçalament Location. Això passa quan volem accedir a una pàgina http, i ens redirigeix a la https.



14.-

Per veure que la connexió és segura a Wireshark, cal filtrar per tls o ssl. A continuació, es pot observar el procés de negociació TLS, on apareixen els missatges ClientHello i ServerHello, que indiquen l'establiment de la connexió segura. Un cop establerta la connexió, el trànsit entre el client i el servidor es xifra, i els paquets mostraran "Application Data". A més, podem veure el certificat SSL del servidor i la llista de ciphersuites que s'utilitzaran per al xifratge de la comunicació.