

ANALISI PACCHETTI

HTTPS

Wireshark packet capture showing the first three-way handshake on interface eth0. The selected packet is an Ethernet II frame from 192.168.32.101 to 192.168.32.100, containing an IPv4 packet and a TCP SYN packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_9b:cd:1d	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000019977	PcsCompu_c7:e1:36	PcsCompu_9b:cd:1d	ARP	42	192.168.32.100 is at 08:00:27:c7:e1:36
3	0.000191729	192.168.32.101	192.168.32.100	TCP	66	49221 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000232556	192.168.32.100	192.168.32.101	TCP	66	443 → 49221 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.000350267	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.000681487	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
7	0.000708843	192.168.32.100	192.168.32.101	TCP	54	443 → 49221 [ACK] Seq=1 Ack=162 Win=64128 Len=0
8	0.033578664	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
9	0.038578072	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.039139453	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
11	0.041213534	192.168.32.101	192.168.32.100	TLSv1	395	Application Data
12	0.051913116	192.168.32.100	192.168.32.101	TLSv1	235	Application Data
13	0.053715433	192.168.32.100	192.168.32.101	TLSv1	384	Application Data, Encrypted Alert
14	0.053867332	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [ACK] Seq=637 Ack=1891 Win=65700 Len=0
15	0.053923990	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [FIN, ACK] Seq=637 Ack=1891 Win=65700 Len=0
16	0.053938430	192.168.32.100	192.168.32.101	TCP	54	443 → 49221 [ACK] Seq=1891 Ack=638 Win=64128 Len=0
17	5.055611964	PcsCompu_c7:e1:36	PcsCompu_9b:cd:1d	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
18	5.055803956	PcsCompu_9b:cd:1d	PcsCompu_c7:e1:36	ARP	60	192.168.32.101 is at 08:00:27:9b:cd:1d

Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d), Dst: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)

Destination: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)

Source: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 52

Identification: 0x028d (653)

010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 128

Protocol: TCP (6)

Header Checksum: 0x361d [validation disabled]

[Header checksum status: Unverified]

Source Address: 192.168.32.101

Destination Address: 192.168.32.100

Transmission Control Protocol, Src Port: 49221, Dst Port: 443, Seq: 0, Len: 0

Ethernet (eth), 14 bytes

Packets: 18 · Displayed: 18 (100.0%) · Dropped: 0 (0.0%) · Profile: Default

Dall'immagine precedente si può notare che:

- il MAC address sorgente è quello del client (windows 7) ovvero "08:00:27:9b:cd:1d"
- il MAC address destinazione è quello del server (kali linux) ovvero "08:00:27:c7:e1:36"

Nelle schermate successive tali MAC address si alternano tra source address e destination address in quanto sta avvenendo il cosiddetto "three way handshake" al fine di poter stabilire una connessione con protocollo TCP.

Wireshark packet capture showing the second three-way handshake on interface eth0. The selected packet is an Ethernet II frame from 192.168.32.100 to 192.168.32.101, containing an IPv4 packet and a TCP ACK packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_9b:cd:1d	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000019977	PcsCompu_c7:e1:36	PcsCompu_9b:cd:1d	ARP	42	192.168.32.100 is at 08:00:27:c7:e1:36
3	0.000191729	192.168.32.101	192.168.32.100	TCP	66	49221 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000232556	192.168.32.100	192.168.32.101	TCP	66	443 → 49221 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.000350267	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.000681487	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
7	0.000708843	192.168.32.100	192.168.32.101	TCP	54	443 → 49221 [ACK] Seq=1 Ack=162 Win=64128 Len=0
8	0.033578664	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
9	0.038578072	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.039139453	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
11	0.041213534	192.168.32.101	192.168.32.100	TLSv1	395	Application Data
12	0.051913116	192.168.32.100	192.168.32.101	TLSv1	235	Application Data
13	0.053715433	192.168.32.100	192.168.32.101	TLSv1	384	Application Data, Encrypted Alert
14	0.053867332	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [ACK] Seq=637 Ack=1891 Win=65700 Len=0
15	0.053923990	192.168.32.101	192.168.32.100	TCP	60	49221 → 443 [FIN, ACK] Seq=637 Ack=1891 Win=65700 Len=0
16	0.053938430	192.168.32.100	192.168.32.101	TCP	54	443 → 49221 [ACK] Seq=1891 Ack=638 Win=64128 Len=0
17	5.055611964	PcsCompu_c7:e1:36	PcsCompu_9b:cd:1d	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
18	5.055803956	PcsCompu_9b:cd:1d	PcsCompu_c7:e1:36	ARP	60	192.168.32.101 is at 08:00:27:9b:cd:1d

Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36), Dst: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)

Destination: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)

Source: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

Total Length: 52

Identification: 0x0000 (0)

010. = Flags: 0x2, Don't fragment

...0 0000 0000 0000 = Fragment Offset: 0

Time to Live: 64

Protocol: TCP (6)

Header Checksum: 0x78aa [validation disabled]

[Header checksum status: Unverified]

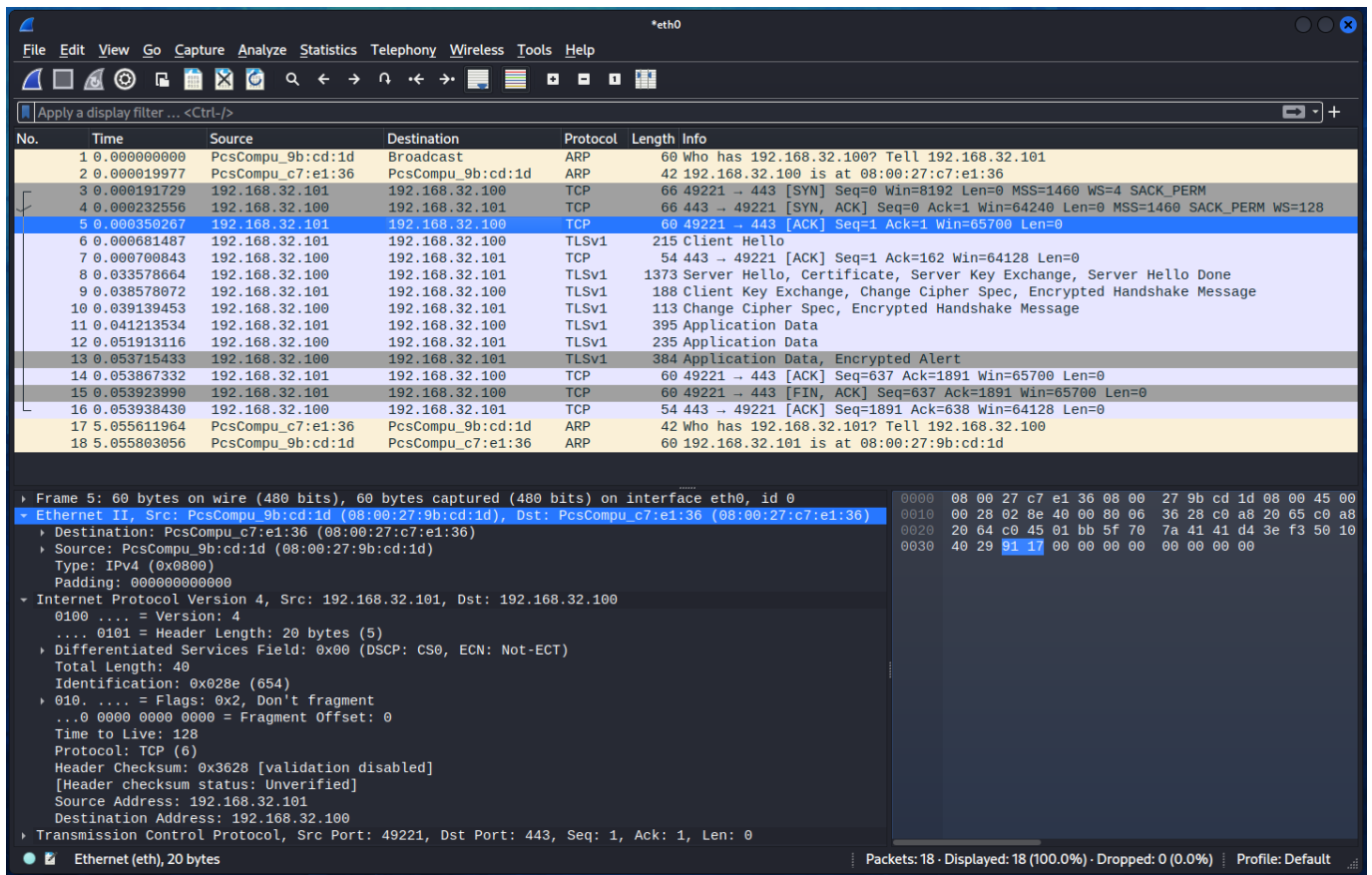
Source Address: 192.168.32.100

Destination Address: 192.168.32.101

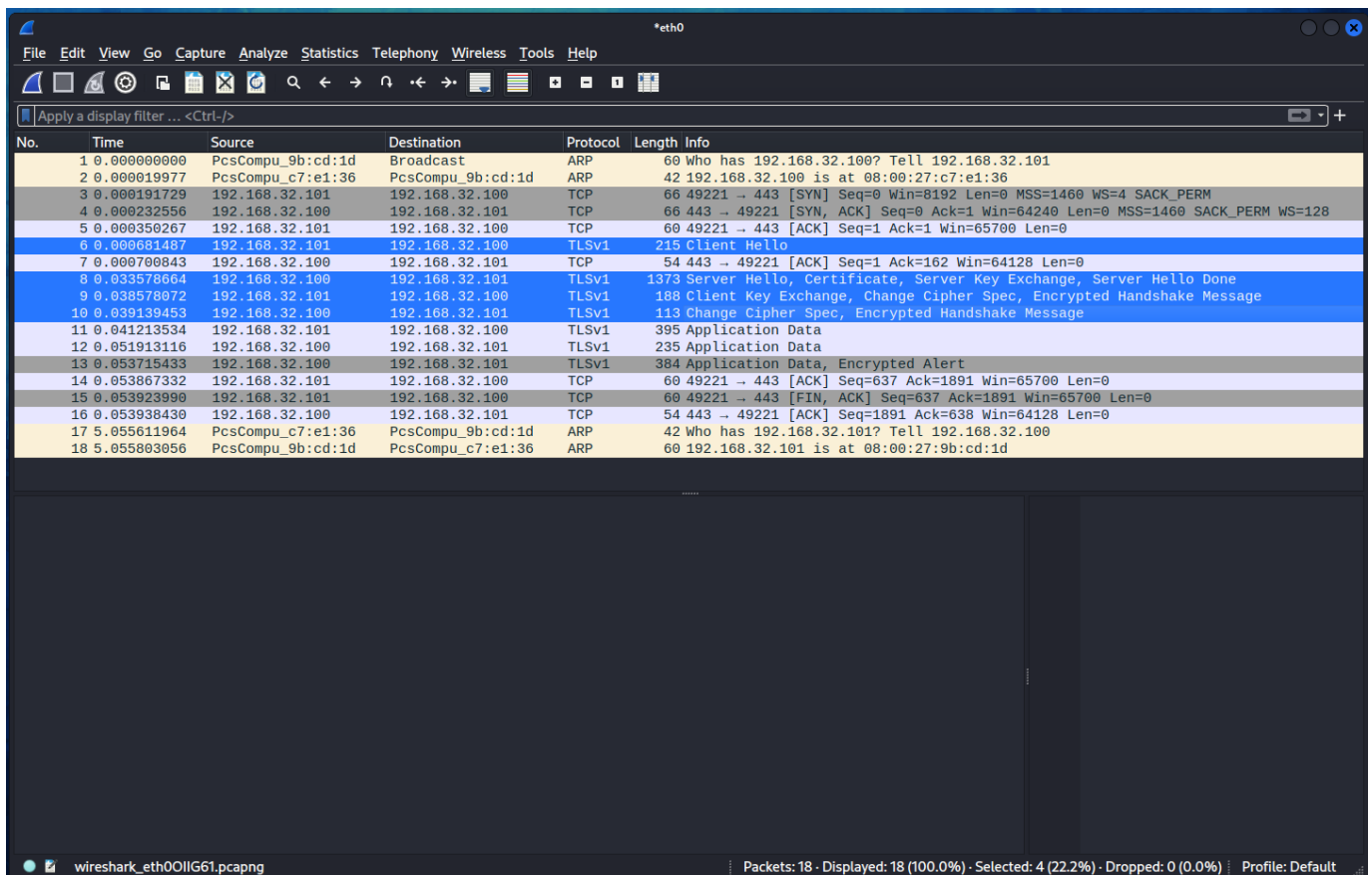
Transmission Control Protocol, Src Port: 443, Dst Port: 49221, Seq: 0, Ack: 1, Len: 0

Ethernet (eth), 14 bytes

Packets: 18 · Displayed: 18 (100.0%) · Dropped: 0 (0.0%) · Profile: Default

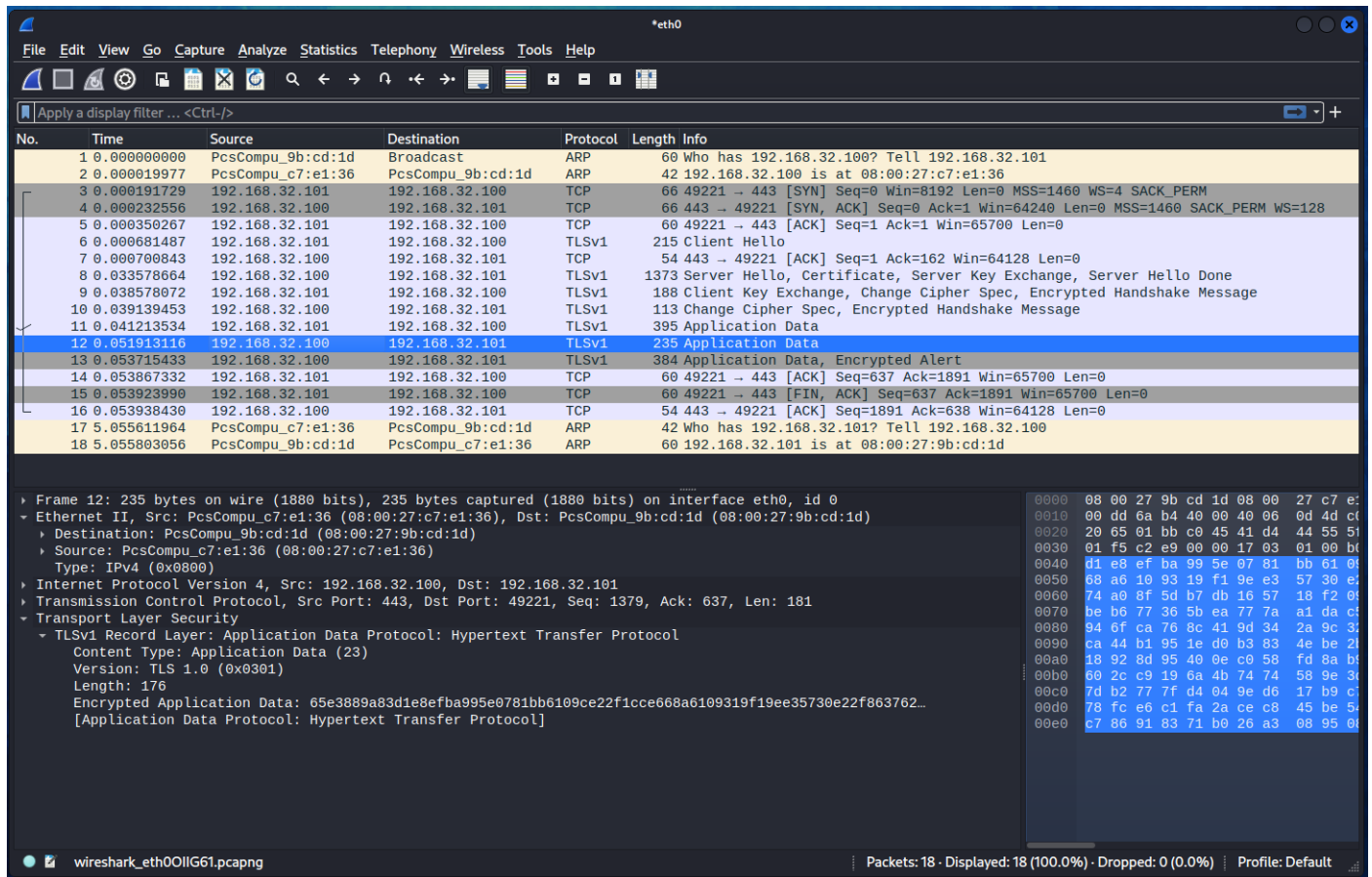


Nelle schermate precedenti possiamo notare che sono stati catturati pacchetti con protocollo TLSv1. Questi pacchetti sono presenti poiché la pagina web richiesta utilizza il protocollo “HTTPS” perciò viene utilizzato il protocollo di crittografia dei dati detto TLS (Transport Layer Security).



I pacchetti evidenziati in blu nell’immagine precedente rappresentano lo scambio di informazioni tra client e server al fine di poter instaurare la sessione crittografata. Il client ha inviato al server un messaggio “Client Hello” che riporta la

Arriviamo infine ai pacchetti contenenti i dati della pagina web che, come possiamo vedere nella penultima stringa del dettaglio del pacchetto selezionato, risultano crittografati.



HTTP

The image shows a Wireshark packet capture on interface eth0. The packet list displays 14 packets. The first packet is a broadcast ARP request. The second packet is an ARP response. The third packet is a DNS query. The fourth packet is a DNS response. The fifth packet is a TCP SYN packet. The sixth packet is a TCP ACK packet. The seventh packet is a TCP ACK packet. The eighth packet is a TCP ACK packet. The ninth packet is a TCP ACK packet. The tenth packet is a TCP ACK packet. The eleventh packet is a TCP ACK packet. The twelfth packet is a TCP ACK packet. The thirteenth packet is a TCP ACK packet. The fourteenth packet is a TCP ACK packet.

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
Ethernet II, Src: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Destination: Broadcast (ff:ff:ff:ff:ff:ff)
Source: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)
Type: ARP (0x0806)
Padding: 00000000000000000000000000000000
Address Resolution Protocol (request)
Hardware type: Ethernet (1)
Protocol type: IPv4 (0x0800)
Hardware size: 6
Protocol size: 4
Opcode: request (1)
Sender MAC address: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)
Sender IP address: 192.168.32.101
Target MAC address: 00:00:00:00:00:00 (00:00:00:00:00:00)
Target IP address: 192.168.32.100

Nell'immagine precedente possiamo vedere i pacchetti che sono stati scambiati tra il client e il server HTTP. Rispetto al server HTTPS rimangono invariati i tre pacchetti TCP sia all'inizio della connessione (three way handshake) che alla fine con la differenza che la porta utilizzata è diversa ovvero con HTTP viene utilizzata la porta 80 mentre con HTTPS viene utilizzata la porta 443.

The image shows a Wireshark packet capture on interface eth0. The packet list displays 14 packets. The first packet is a broadcast ARP request. The second packet is an ARP response. The third packet is a DNS query. The fourth packet is a DNS response. The fifth packet is a TCP SYN packet. The sixth packet is a TCP ACK packet. The seventh packet is a TCP ACK packet. The eighth packet is a TCP ACK packet. The ninth packet is a TCP ACK packet. The tenth packet is a TCP ACK packet. The eleventh packet is a TCP ACK packet. The twelfth packet is a TCP ACK packet. The thirteenth packet is a TCP ACK packet. The fourteenth packet is a TCP ACK packet.

Frame 11: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface eth0, id 0
Ethernet II, Src: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36), Dst: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)
Destination: PcsCompu_9b:cd:1d (08:00:27:9b:cd:1d)
Source: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
Transmission Control Protocol, Src Port: 80, Dst Port: 49239, Seq: 151, Ack: 419, Len: 258
[2 Reassembled TCP Segments (408 bytes): #10(150), #11(258)]
Hypertext Transfer Protocol
Line-based text data: text/html (10 lines)
<html>\n<head>\n<title>INetSim default HTML page</title>\n</head>\n<body>\n<p>\n<p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>\n<p align="center">This file is an HTML document.</p>\n</body>\n</html>\n

Un'altra sostanziale differenza è nel tipo di pacchetti in quanto non vediamo più dei pacchetti di tipo TLS ma solo pacchetti di tipo HTTP in quanto questi ultimi non sono più crittografati infatti, come mostrato nella schermata precedente si può notare che riusciamo a leggere il codice HTML della pagina visitata dal client.

Nella cattura dei pacchetti possiamo vedere anche i quelli di tipo DNS che dimostrano che è stata fatta richiesta al server DNS per la risoluzione del nome di dominio "epicode.internal".