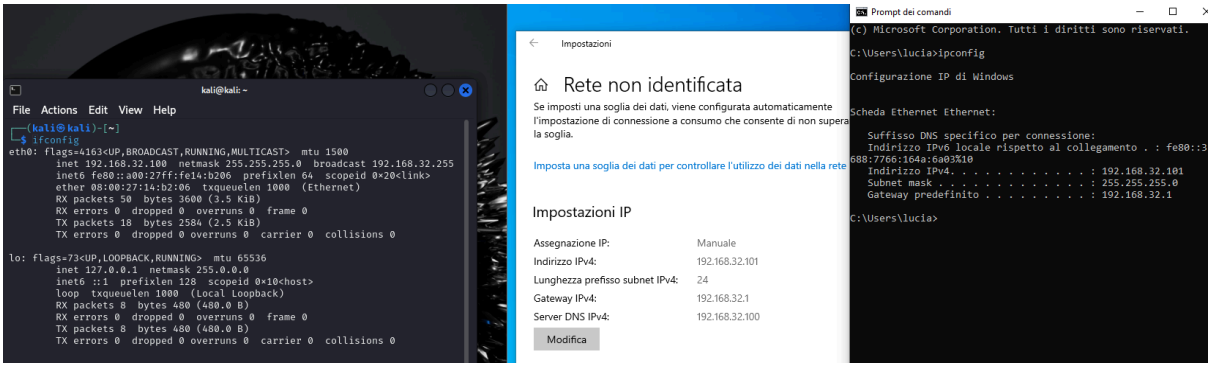
REPORT di Pandolfi Luciano W4 Esercizio Fine Modulo

TRACCIA:

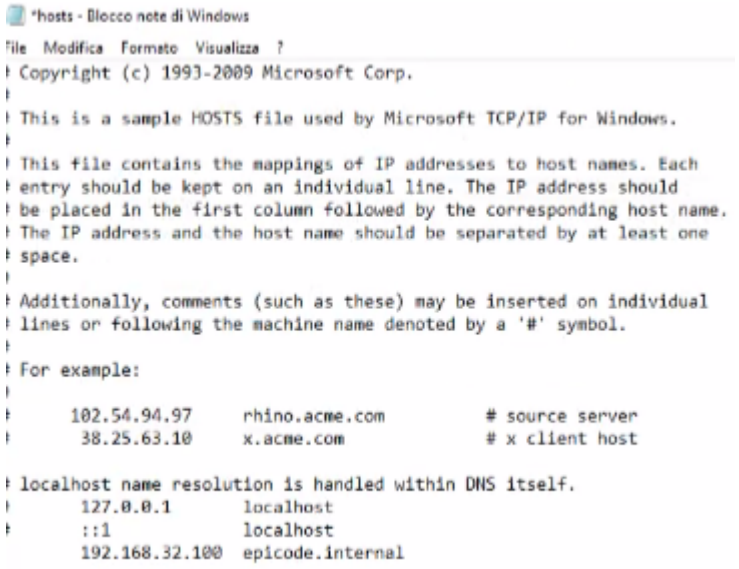
Simulare in un ambiente virtuale un'architettura client server in cui un client con IP statico 192.168.32.101 richiede tramite web browser una risorsa all'hostname **epicode.internal** che risponde all'indirizzo IP statico 192.168.32.100 (funge da server DNS - Kali Linux).
Si intercetti la comunicazione tramite Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS e HTTP motivando e spiegando le differenze tra i due servizi.

ESERCIZIO:

Come prima cosa ho configurato gli indirizzi IP statici rispettivamente al client (Windows10) e server (Kali) nel primo caso aggiungendo il Server DNS da utilizzare per accedere alla risorsa, nel secondo caso la configurazione tramite Terminale (CLI) con il comando "sudo nano /etc/network/interfaces".



In seguito ho creato una policy che permettesse al Firewall di Windows che la comunicazione in entrata ed uscita con Kali Linux (IP 192.168.32.100) avvenisse.
Successivamente accedendo sul Disco C del client, ho modificato da amministratore il File "hosts" aggiungendo il dominio **epicode.internal** ed il Server DNS (Kali).



Ho attivato per poter effettuare le diverse simulazioni, i servizi DNS/HTTP ed DNS/HTTPS in momenti differenti su Kali Linux tramite InetSim specificando il “service_bind_address” ed il “dns_default_ip” inserendo anche il dominio e l’IP da raggiungere per accedere alla risorsa.

```
# ftps, irc, https
#
start_service dns
#start_service http
start_service https
#start_service smtp
#start_service smtps
```

```
#
dns_default_ip 192.168.32.100
```

```
#
service_bind_address 192.168.32.100
```

```
#
start_service dns
start_service http
#start_service https
```

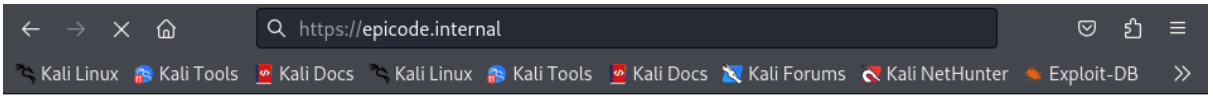
```
#
dns_static epicode.internal 192.168.32.100
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
```

Ho poi avviato InetSim con i servizi attivi sopra esposti usufruendo dei servizi DNS/HTTP (porta predefinita 80) ed in seguito DNS/HTTPS (porta predefinita 443) evidenziando i protocolli TCP/UDP.

```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/inetsim/inetsim.conf

┌──(kali㉿kali)-[~]
└─$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 10376) ===
Session ID:     10376
Listening on:   192.168.32.100
Real Date/Time: 2024-09-28 06:49:48
Fake Date/Time: 2024-09-28 06:49:48 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 10386)
deprecated method; prefer start_server() at /usr/share/perl5/INetSim/DNS.pm l
ine 69.
Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INe
tSim/DNS.pm line 69.
  * http_80_tcp - started (PID 10387)
 done.
Simulation running.
```

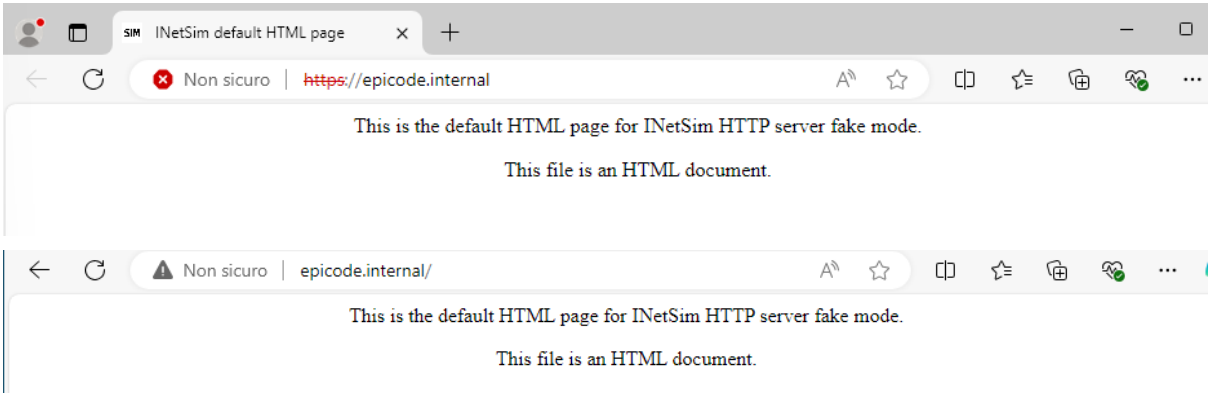
```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/inetsim/inetsim.conf

┌──(kali㉿kali)-[~]
└─$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 9884) ===
Session ID:     9884
Listening on:   192.168.32.100
Real Date/Time: 2024-09-28 04:36:43
Fake Date/Time: 2024-09-28 04:36:43 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 9888)
deprecated method; prefer start_server() at /usr/share/perl5/INetSim/DNS.pm l
ine 69.
Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INe
tSim/DNS.pm line 69.
  * https_443_tcp - started (PID 9892)
 done.
Simulation running.
```

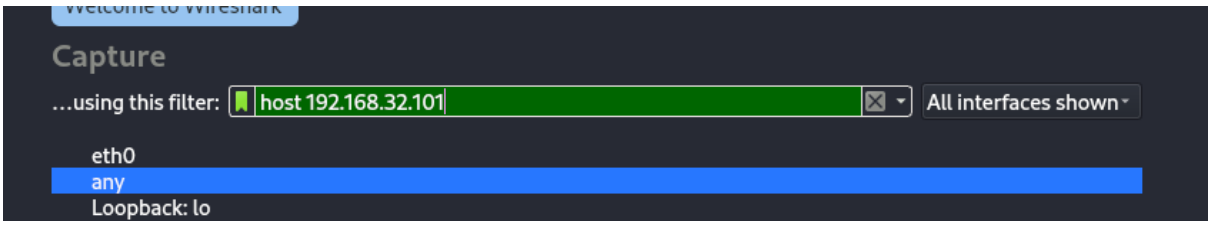
Infine, contemporaneamente all'avvio dei servizi impostati su InetSim ho intercettato la comunicazione tramite Wireshark , evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS e HTTP.

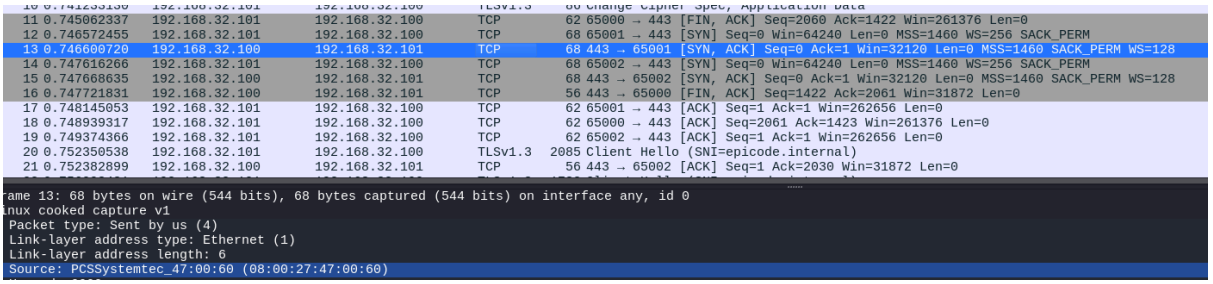


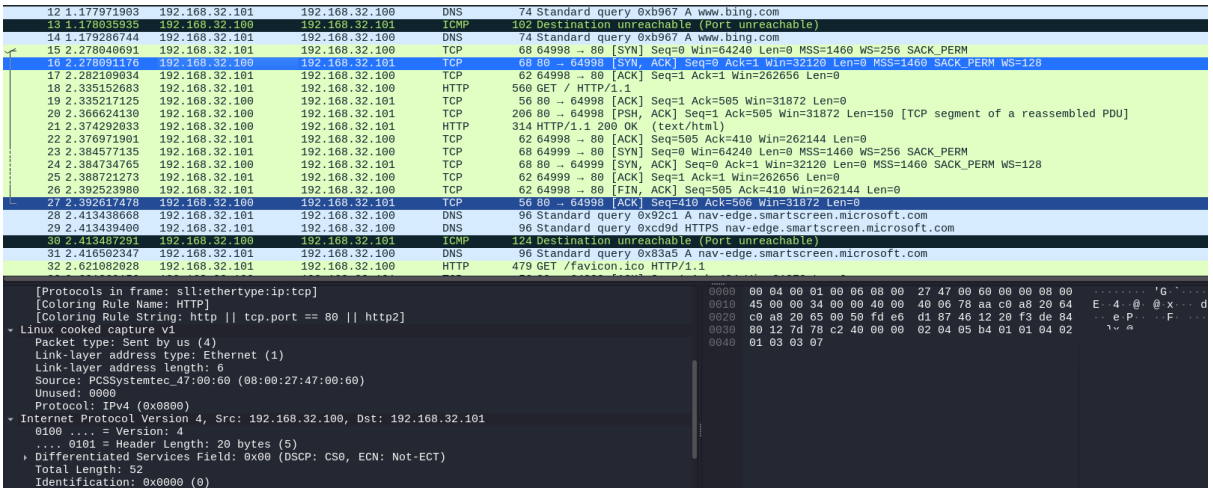
This is the default HTML page for INetSim HTTP server fake mode.

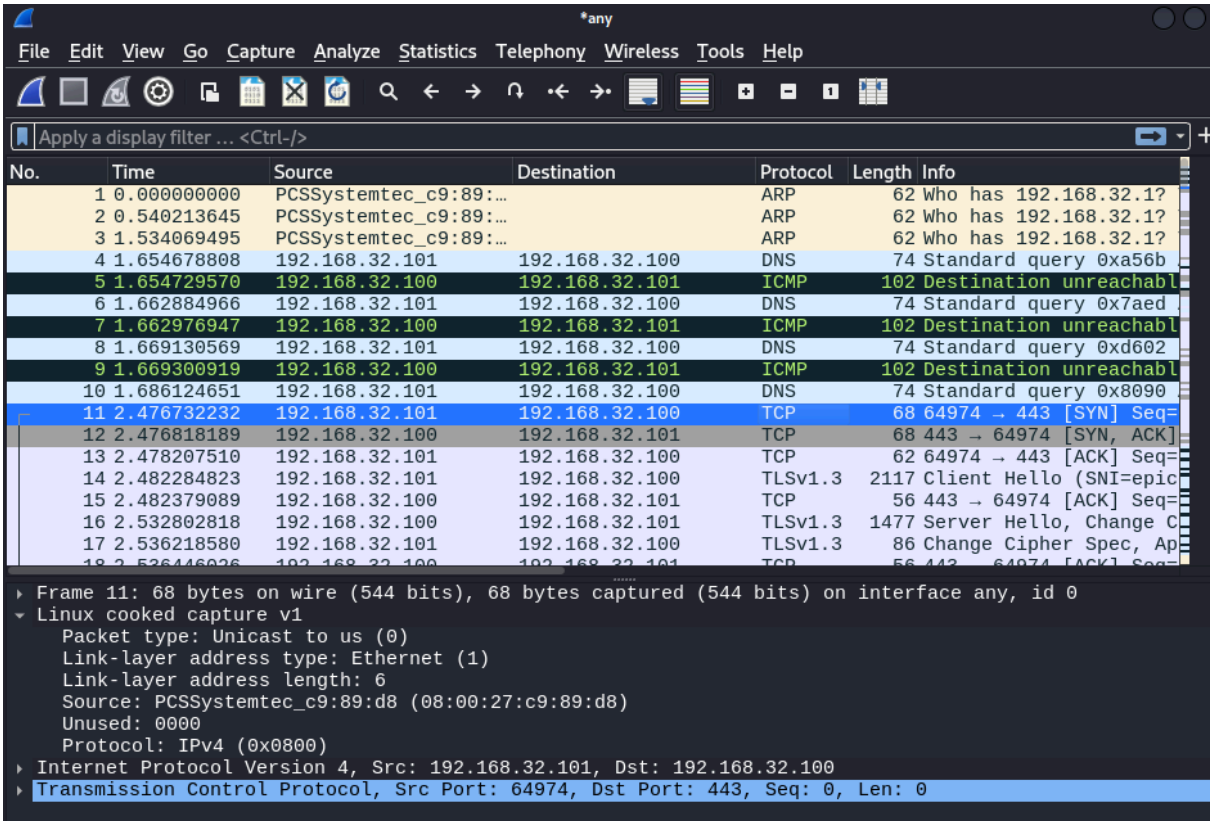
This file is an HTML document.

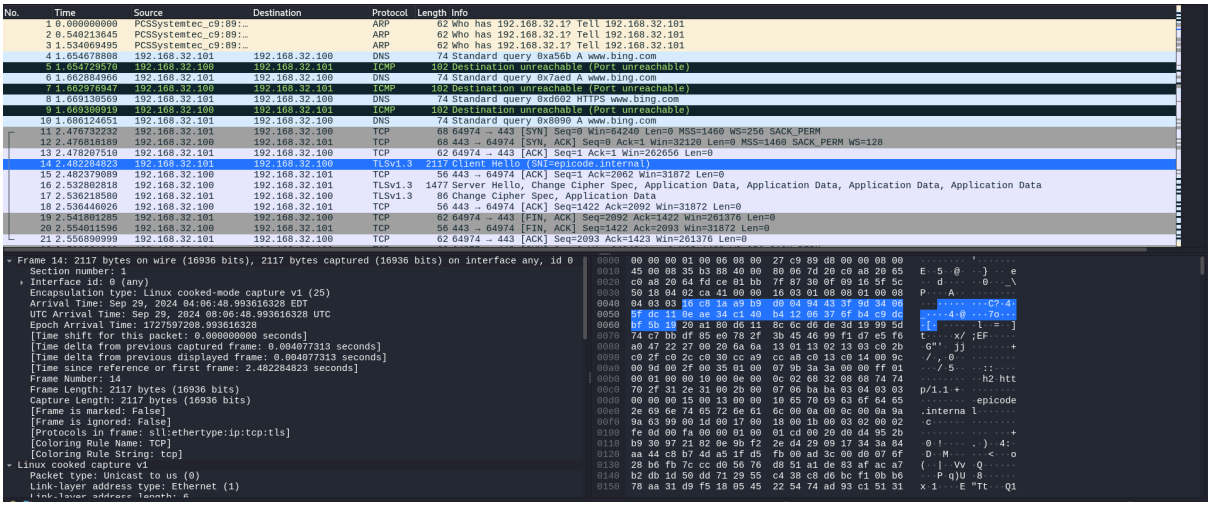


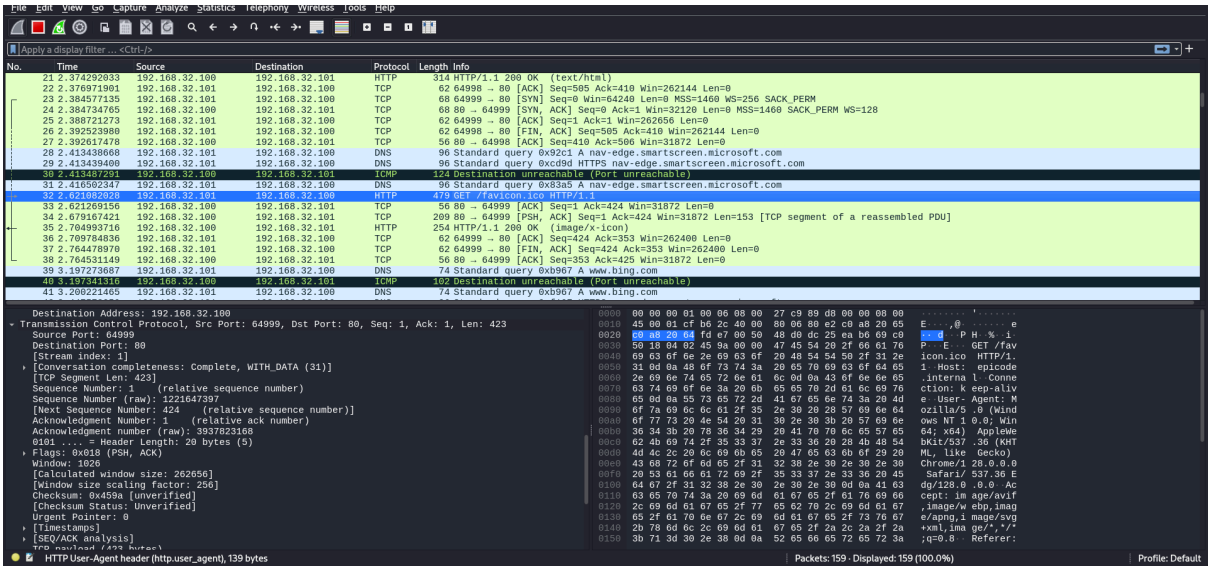

Capture
...using this filter:
host 192.168.32.101
All interfaces shown
eth0
any
Loopback: lo


11 0.745062337 192.168.32.101 192.168.32.100 TCP 62 65000 → 443 [FIN, ACK] Seq=2060 Ack=1422 Win=261376 Len=0
12 0.746572455 192.168.32.101 192.168.32.100 TCP 68 65001 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
13 0.746600720 192.168.32.100 192.168.32.101 TCP 68 443 → 65001 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
14 0.747616266 192.168.32.101 192.168.32.100 TCP 68 65002 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
15 0.747668635 192.168.32.100 192.168.32.101 TCP 68 443 → 65002 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
16 0.747721831 192.168.32.100 192.168.32.101 TCP 56 443 → 65000 [FIN, ACK] Seq=1422 Ack=2061 Win=31872 Len=0
17 0.748145053 192.168.32.101 192.168.32.100 TCP 62 65001 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
18 0.748939317 192.168.32.101 192.168.32.100 TCP 62 65000 → 443 [ACK] Seq=2061 Ack=1423 Win=261376 Len=0
19 0.749374366 192.168.32.101 192.168.32.100 TCP 62 65002 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
20 0.752350538 192.168.32.101 192.168.32.100 TLSv1.3 2085 Client Hello (SNI=epicode.internal)
21 0.752382899 192.168.32.100 192.168.32.101 TCP 56 443 → 65002 [ACK] Seq=1 Ack=2030 Win=31872 Len=0
ame 13: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0
inux cooked capture v1
Packet type: Sent by us (4)
Link-layer address type: Ethernet (1)
Link-layer address length: 6
Source: PCSSystemtec_47:00:60 (08:00:27:47:00:60)


12 1.177971903 192.168.32.101 192.168.32.100 DNS 74 Standard query 0xb967 A www.bing.com
13 1.178035935 192.168.32.100 192.168.32.101 ICMP 102 Destination unreachable (Port unreachable)
14 1.179286744 192.168.32.101 192.168.32.100 DNS 74 Standard query 0xb967 A www.bing.com
15 2.278040691 192.168.32.101 192.168.32.100 TCP 68 64998 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
16 2.278091176 192.168.32.100 192.168.32.101 TCP 68 80 → 64998 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
17 2.282109034 192.168.32.101 192.168.32.100 TCP 62 64998 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
18 2.335152683 192.168.32.101 192.168.32.100 HTTP 560 GET / HTTP/1.1
19 2.335217125 192.168.32.100 192.168.32.101 TCP 56 80 → 64998 [ACK] Seq=1 Ack=505 Win=31872 Len=0
20 2.366624130 192.168.32.100 192.168.32.101 TCP 206 80 → 64998 [PSH, ACK] Seq=1 Ack=505 Win=31872 Len=150 [TCP segment of a reassembled PDU]
21 2.374292033 192.168.32.100 192.168.32.101 HTTP 314 HTTP/1.1 200 OK (text/html)
22 2.376971901 192.168.32.101 192.168.32.100 TCP 62 64998 → 80 [ACK] Seq=505 Ack=410 Win=262144 Len=0
23 2.384577135 192.168.32.101 192.168.32.100 TCP 68 64999 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
24 2.384734765 192.168.32.100 192.168.32.101 TCP 68 80 → 64999 [SYN, ACK] Seq=0 Ack=1 Win=32120 Len=0 MSS=1460 SACK_PERM WS=128
25 2.388721273 192.168.32.101 192.168.32.100 TCP 62 64999 → 80 [ACK] Seq=1 Ack=1 Win=262656 Len=0
26 2.392523980 192.168.32.101 192.168.32.100 TCP 62 64998 → 80 [FIN, ACK] Seq=505 Ack=410 Win=262144 Len=0
27 2.392617478 192.168.32.100 192.168.32.101 TCP 56 80 → 64998 [ACK] Seq=410 Ack=506 Win=31872 Len=0
28 2.413438668 192.168.32.101 192.168.32.100 DNS 96 Standard query 0x92c1 A nav-edge.smartscreen.microsoft.com
29 2.413439400 192.168.32.101 192.168.32.100 DNS 96 Standard query 0xcd9d HTTPS nav-edge.smartscreen.microsoft.com
30 2.413487291 192.168.32.100 192.168.32.101 ICMP 124 Destination unreachable (Port unreachable)
31 2.416502347 192.168.32.101 192.168.32.100 DNS 96 Standard query 0x83a5 A nav-edge.smartscreen.microsoft.com
32 2.621082028 192.168.32.101 192.168.32.100 HTTP 479 GET /favicon.ico HTTP/1.1
[Protocols in frame: sll:ethertype:ip:tcp]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80 || http2]
Linux cooked capture v1
Packet type: Sent by us (4)
Link-layer address type: Ethernet (1)
Link-layer address length: 6
Source: PCSSystemtec_47:00:60 (08:00:27:47:00:60)
Unused: 0000
Protocol: 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)
0000 00 04 00 01 00 06 08 00 27 47 00 60 00 00 08 00
0010 45 00 00 34 00 00 40 00 40 06 78 aa c0 a8 20 64
0020 c0 a8 20 65 00 50 fd e6 d1 87 46 12 20 f3 de 84
0030 80 12 7d 78 c2 40 00 00 02 04 05 b4 01 01 04 02
0040 01 03 03 07


*any
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 PCSSystemtec_c9:89:... ARP 62 Who has 192.168.32.1?
2 0.540213645 PCSSystemtec_c9:89:... ARP 62 Who has 192.168.32.1?
3 1.534069495 PCSSystemtec_c9:89:... ARP 62 Who has 192.168.32.1?
4 1.654678808 192.168.32.101 192.168.32.100 DNS 74 Standard query 0xa56b
5 1.654729570 192.168.32.100 192.168.32.101 ICMP 102 Destination unreachabl
6 1.662884966 192.168.32.101 192.168.32.100 DNS 74 Standard query 0x7aed
7 1.662976947 192.168.32.100 192.168.32.101 ICMP 102 Destination unreachabl
8 1.669130569 192.168.32.101 192.168.32.100 DNS 74 Standard query 0xd602
9 1.669300919 192.168.32.100 192.168.32.101 ICMP 102 Destination unreachabl
10 1.686124651 192.168.32.101 192.168.32.100 DNS 74 Standard query 0x8090
11 2.476732232 192.168.32.101 192.168.32.100 TCP 68 64974 → 443 [SYN] Seq=
12 2.476818189 192.168.32.100 192.168.32.101 TCP 68 443 → 64974 [SYN, ACK]
13 2.478207510 192.168.32.101 192.168.32.100 TCP 62 64974 → 443 [ACK] Seq=
14 2.482284823 192.168.32.101 192.168.32.100 TLSv1.3 2117 Client Hello (SNI=epic
15 2.482379089 192.168.32.100 192.168.32.101 TCP 56 443 → 64974 [ACK] Seq=
16 2.532802818 192.168.32.100 192.168.32.101 TLSv1.3 1477 Server Hello, Change C
17 2.536218580 192.168.32.101 192.168.32.100 TLSv1.3 86 Change Cipher Spec, Ap
Frame 11: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0
Linux cooked capture v1
Packet type: Unicast to us (0)
Link-layer address type: Ethernet (1)
Link-layer address length: 6
Source: PCSSystemtec_c9:89:d8 (08:00:27:c9:89:d8)
Unused: 0000
Protocol: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
Transmission Control Protocol, Src Port: 64974, Dst Port: 443, Seq: 0, Len: 0





Questo mi ha permesso di identificare le differenze dei servizi HTTP/HTTPS potendo così metterli a paragone.

servizio HTTP:

Mostra **in chiaro** la richiesta (GET) del client, l'Header e la risposta con i dati richiesti leggibile.

La comunicazione non avviene tramite un processo di handshake SSL, quindi più diretta rendendolo più vulnerabile ad attacchi come per esempio il man-in-the-middle.

servizio HTTPS:

Il contenuto è **criptato**, la comunicazione è più sicura ed avviene tramite il processo three-way handshake (SYN/ACK) con scambi di chiavi e certificati crittografati, proteggendo l'integrità e la riservatezza dei dati trasmessi.