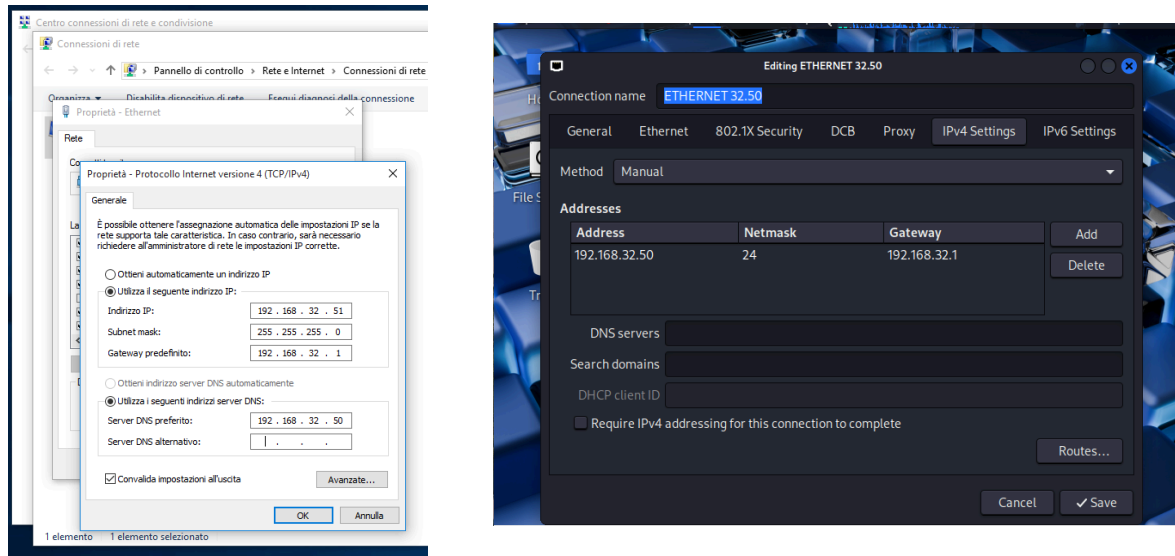


PROGETTO FINE MODULO M1

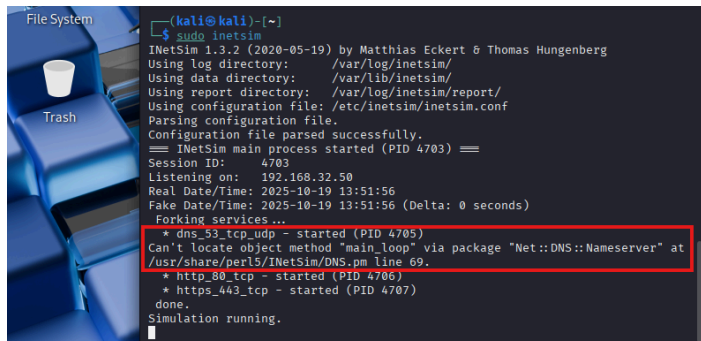
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.51 (Windows) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.50 (Kali).

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS. Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze se presenti.

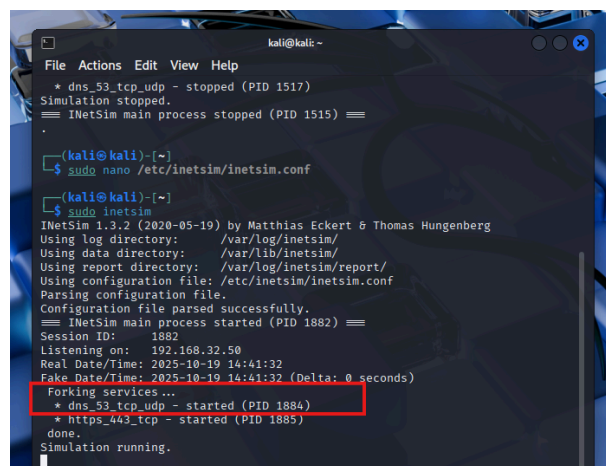
Sono partito impostando rete interna delle due macchine virtuali su VirtualBox, poi ho impostato indirizzi IP manualmente su KALI 192.168.32.50 e poi su Windows 192.168.32.51 e infine sempre su Windows ho inserito come DNS predefinito quello dato a KALI ovvero 192.168.32.50. Immagini Windows a sinistra e a destra KALI.



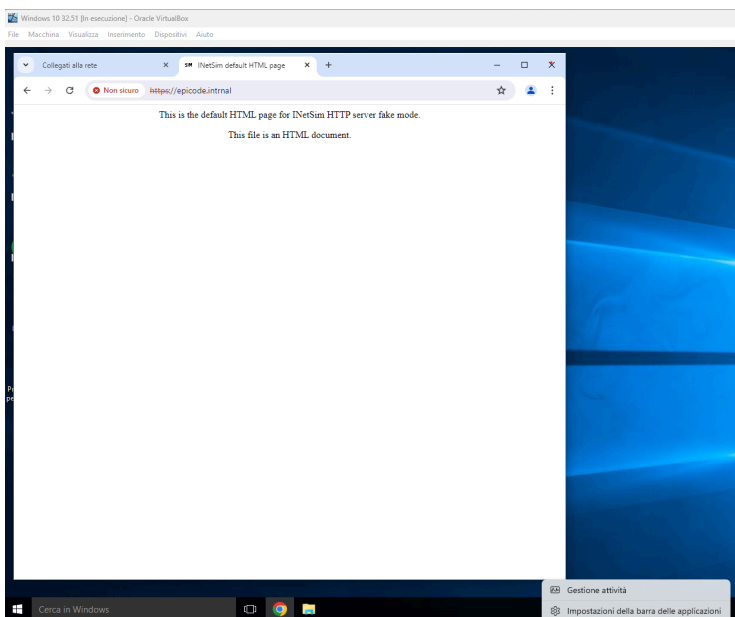
Dopo aver attribuito gli indirizzi IP alle macchine virtuali e aver messo un DNS di riferimento a Windows, su kali ho modificato le impostazioni di Inetsim rendendo attivi i servizi DNS, HTTP e HTTPS per poter simulare dei servizi web. Ho inserito su Inetsim nella riga dns_default.ip e nella riga service_bind_address l'indirizzo 192.168.32.50. Poi ho inserito nella riga dns_default_name epicode, mentre nella riga dns_default_domainname internal. ho cancellato i cancelletti ha inizio riga per poter rendere tutti i servizi necessari attivi. Dopodiché ho salvato le modifiche e sembrava tutto pronto ma una volta provato ad avviare Inetsim il DNS dava un errore:



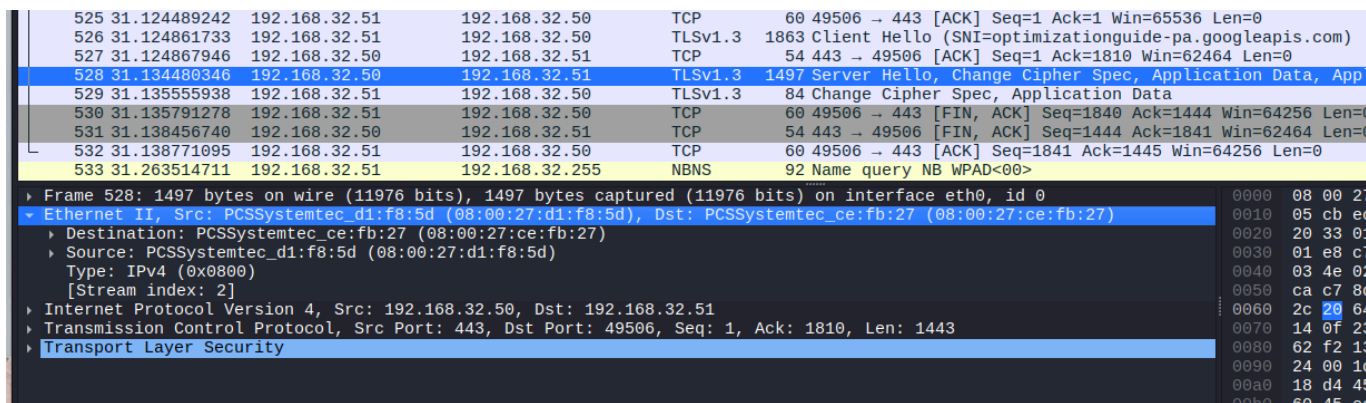
Ho risolto l'errore copiandolo e incollandolo su google, sono giunto ad una pagina di discussioni dove veniva spiegato che per far funzionare quel servizio di Inetsim bisognava installare una versione più vecchia del servizio DNS, nella conversazione ho trovato il link per il download e tutti i comandi utili alla sua installazione, eccetto il comando che serviva per decomprimere il file scaricato, il quale mi ha fatto sudare per essere trovato. Dunque ho installato Kali ad internet per poter scaricare il file e una volta eseguiti tutti i comandi per la sua installazione il servizio DNS di Inetsim ha iniziato a funzionare.



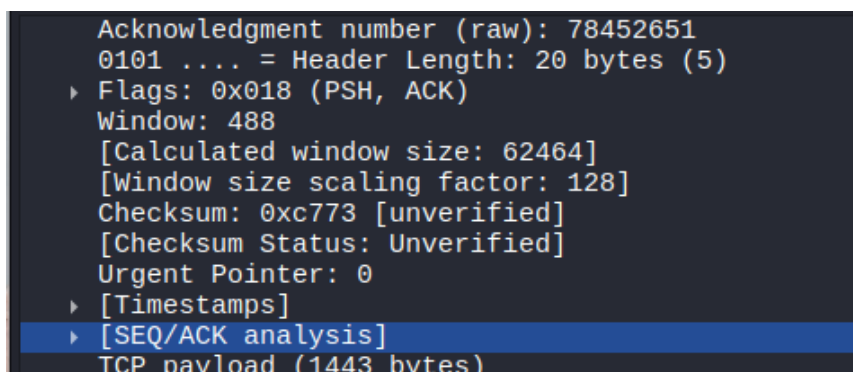
Una volta avviato Inetsim senza errori ho tentato la ricerca da il Browser di Windows, e la ricerca è andata in porto:



Visto che tutto è andato come dovrebbe ho iniziato a osservare i pacchetti con wireshark, iniziando ad osservare una connessione HTTPS:



Questo è lo screenshot della risposta del server al client, si possono trovare gli indirizzi MAC ed attribuirli agli indirizzi IP, perché sapendo che è la risposta del server il source IP è 192.268.32.50 e nella riga evidenziata troviamo il source mac address che è 08:00:27:d1:f8:5d e troviamo anche il destination address quindi IP 192.168.32.51 e il suo MAC address 08:00:27:ce:fb:27 che sono gli indirizzi del client. Da questa foto si possono anche rilevare il numero di bytes che sono passati nel cavo.



OSSERVAZIONE PACCHETTI HTTPS CON WIRESHARK

Visto che tutto è andato come dovrebbe ho iniziato a osservare i pacchetti con wireshark, iniziando ad osservare una connessione HTTPS:

525	31.124489242	192.168.32.51	192.168.32.50	TCP	60 49506 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0
526	31.124861733	192.168.32.51	192.168.32.50	TLSv1.3	1863 Client Hello (SNI=optimizationguide-pa.googleapis.com)
527	31.124867946	192.168.32.50	192.168.32.51	TCP	54 443 → 49506 [ACK] Seq=1 Ack=1810 Win=62464 Len=0
528	31.134480346	192.168.32.50	192.168.32.51	TLSv1.3	1497 Server Hello, Change Cipher Spec, Application Data, App...
529	31.135555938	192.168.32.51	192.168.32.50	TLSv1.3	84 Change Cipher Spec, Application Data
530	31.135791278	192.168.32.51	192.168.32.50	TCP	60 49506 → 443 [FIN, ACK] Seq=1840 Ack=1444 Win=64256 Len=0
531	31.138456740	192.168.32.50	192.168.32.51	TCP	54 443 → 49506 [FIN, ACK] Seq=1444 Ack=1841 Win=62464 Len=0
532	31.138771095	192.168.32.51	192.168.32.50	TCP	60 49506 → 443 [ACK] Seq=1841 Ack=1445 Win=64256 Len=0
533	31.263514711	192.168.32.51	192.168.32.255	NBNS	92 Name query NB WPAD<00>

Frame 528: 1497 bytes on wire (11976 bits), 1497 bytes captured (11976 bits) on interface eth0, id 0

Ethernet II, Src: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d), Dst: PCSSystemtec_ce:fb:27 (08:00:27:ce:fb:27)

Destination: PCSSystemtec_ce:fb:27 (08:00:27:ce:fb:27)

Source: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d)

Type: IPv4 (0x0800)

[Stream index: 2]

Internet Protocol Version 4, Src: 192.168.32.50, Dst: 192.168.32.51

Transmission Control Protocol, Src Port: 443, Dst Port: 49506, Seq: 1, Ack: 1810, Len: 1443

Transport Layer Security

Questo è lo screenshot della risposta del server al client, si possono trovare gli indirizzi MAC ed attribuirli agli indirizzi IP, perché sapendo che è la risposta del server il source IP è 192.268.32.50 e nella riga evidenziata troviamo il source mac address che è 08:00:27:d1:f8:5d e troviamo anche il destination address quindi IP 192.168.32.51 e il suo MAC address 08:00:27:ce:fb:27 che sono gli indirizzi del client. Da questa foto si possono anche rilevare il numero di bytes che sono passati nel cavo.

Acknowledgment number (raw): 78452651
0101 = Header Length: 20 bytes (5)
Flags: 0x018 (PSH, ACK)
Window: 488
[Calculated window size: 62464]
[Window size scaling factor: 128]
Checksum: 0xc773 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
[Timestamps]
[SEQ/ACK analysis]
TCP payload (1443 bytes)

gli unici altri dati che si riescono a carpire su HTTPS sono la lunghezza dell'header e del payload in bytes il resto dei dati è tutto cifrato e impossibile da leggere.

528	31.134480346	192.168.32.50	192.168.32.51	TLSv1.3	1497 Server Hello, Change Cipher Spec, Application Data, Application Data, App...
529	31.135555938	192.168.32.51	192.168.32.50	TLSv1.3	84 Change Cipher Spec, Application Data
530	31.135791278	192.168.32.51	192.168.32.50	TCP	60 49506 → 443 [FIN, ACK] Seq=1840 Ack=1444 Win=64256 Len=0
531	31.138456740	192.168.32.50	192.168.32.51	TCP	54 443 → 49506 [FIN, ACK] Seq=1444 Ack=1841 Win=62464 Len=0
532	31.138771095	192.168.32.51	192.168.32.50	TCP	60 49506 → 443 [ACK] Seq=1841 Ack=1445 Win=64256 Len=0
533	31.263514711	192.168.32.51	192.168.32.255	NBNS	92 Name query NB WPAD<00>

Frame 528: 1497 bytes on wire (11976 bits), 1497 bytes captured (11976 bits) on interface eth0, id 0

Ethernet II, Src: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d), Dst: PCSSystemtec_ce:fb:27 (08:00:27:ce:fb:27)

Internet Protocol Version 4, Src: 192.168.32.50, Dst: 192.168.32.51

Transmission Control Protocol, Src Port: 443, Dst Port: 49506, Seq: 1, Ack: 1810, Len: 1443

Transport Layer Security

TLSv1.3 Record Layer: Handshake Protocol: Server Hello

TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec

TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

TLSv1.3 Record Layer: Application Data Protocol: Hypertext Transfer Protocol

OSSERVAZIONE PACCHETTI CON WIRESHARK SU HTTP

```
470 18.718699384 192.168.32.51 192.168.32.50 TCP 60 49819 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
471 18.720675020 192.168.32.51 192.168.32.50 HTTP 516 GET /sample.txt HTTP/1.1
472 18.720692173 192.168.32.50 192.168.32.51 TCP 54 80 → 49819 [ACK] Seq=1 Ack=463 Win=64128 Len=0
473 18.724254265 192.168.32.50 192.168.32.51 DNS 76 Standard query response 0x4421 Not implemented HTTPS epicode.internal
474 18.724666114 192.168.32.51 192.168.32.50 ICMP 104 Destination unreachable (Port unreachable)
475 18.737062952 192.168.32.50 192.168.32.51 TCP 204 80 → 49819 [PSH, ACK] Seq=1 Ack=463 Win=64128 Len=150 [TCP PDU reassembled]
476 18.738673703 192.168.32.50 192.168.32.51 HTTP 151 HTTP/1.1 200 OK (text/plain)
477 18.739312675 192.168.32.51 192.168.32.50 TCP 60 49819 → 80 [ACK] Seq=463 Ack=249 Win=65280 Len=0
478 18.739312820 192.168.32.51 192.168.32.50 TCP 60 49819 → 80 [FIN, ACK] Seq=463 Ack=249 Win=65280 Len=0

Hypertext Transfer Protocol
  GET /sample.txt HTTP/1.1\r\n
  Host: epicode.internal\r\n
  Connection: keep-alive\r\n
  Upgrade-Insecure-Requests: 1\r\n
  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.0.0 Safari/537.36\r\n
  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7\r\n
  Accept-Encoding: gzip, deflate\r\n
  Accept-Language: it-IT,it;q=0.9,en-US;q=0.8,en;q=0.7\r\n
  \r\n
  [Response in frame: 476]
  [Full request URI: http://epicode.internal/sample.txt]
```

Qui sopra abbiamo una richiesta del client ad un server in HTTP, si possono vedere in chiaro il tipo di richiesta, il formato del file richiesto, tipo di connessione, il nome dell'host, da quale programma sta effettuando la richiesta e in che lingua è stata effettuata la richiesta. e poi in fondo c'è anche l'URI completo

```
475 18.737062952 192.168.32.50 192.168.32.51 TCP 204 80 → 49819 [PSH, ACK] Seq=1 Ack=463 Win=64128 Len=150 [TCP PDU reassembled]
476 18.738673703 192.168.32.50 192.168.32.51 HTTP 151 HTTP/1.1 200 OK (text/plain)
477 18.739312675 192.168.32.51 192.168.32.50 TCP 60 49819 → 80 [ACK] Seq=463 Ack=249 Win=65280 Len=0
478 18.739312820 192.168.32.51 192.168.32.50 TCP 60 49819 → 80 [FIN, ACK] Seq=463 Ack=249 Win=65280 Len=0

[2 Reassembled TCP Segments (247 bytes): #475(150), #476(97)]
Hypertext Transfer Protocol
  HTTP/1.1 200 OK\r\n
  Server: INetSim HTTP Server\r\n
  Date: Sun, 19 Oct 2025 18:59:14 GMT\r\n
  Connection: Close\r\n
  Content-Length: 97\r\n
  Content-Type: text/plain\r\n
  \r\n
  [Request in frame: 471]
  [Time since request: 0.017998683 seconds]
  [Request URI: /sample.txt]
```

Anche dalla risposta del server possiamo carpire molte informazioni: ad esempio che la risposta è stata positiva, informazioni sul server, la data con giorno e orario e la lunghezza del contenuto in bytes.

CONCLUSIONE

Le principali differenze tra HTTP e HTTPS sono che utilizzano numeri di porta differenti HTTP (80) e HTTPS (403), e soprattutto i dati su HTTP possono essere visti in chiaro se intercettati da qualcuno, mentre su HTTPS i pacchetti vengono cifrati da TLS cosicché se venissero intercettati da qualcuno questi siano incomprensibili garantendo la privacy dei client che intendono usare i servizi di un sito web.