

M1 - PROGETTO FINALE

Traccia: Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.50.102 (Windows) richiede tramite web browser una risorsa all' hostname epicode.internal che risponde all'indirizzo 192.168.50.100 (Kali).

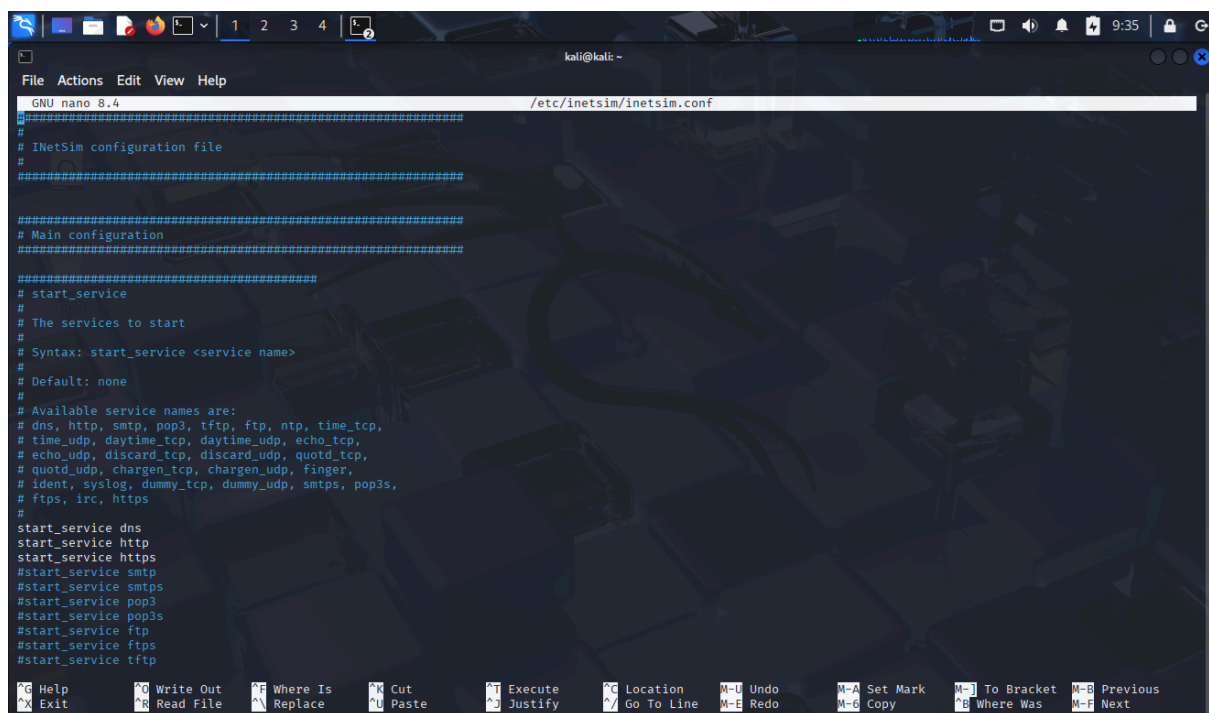
Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

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.

Svolgimento: Per simulare un ambiente virtuale client server, è necessario predisporre due macchine virtuali, la prima con il ruolo di server e la seconda con il ruolo di client.

Sulla macchina server, ho utilizzato il tool INetSim per emulare diversi servizi di rete. In particolare, ho attivato i servizi DNS, HTTP e HTTPS modificando il file di configurazione rimuovendo il carattere di commento “#” dalle relative voci.



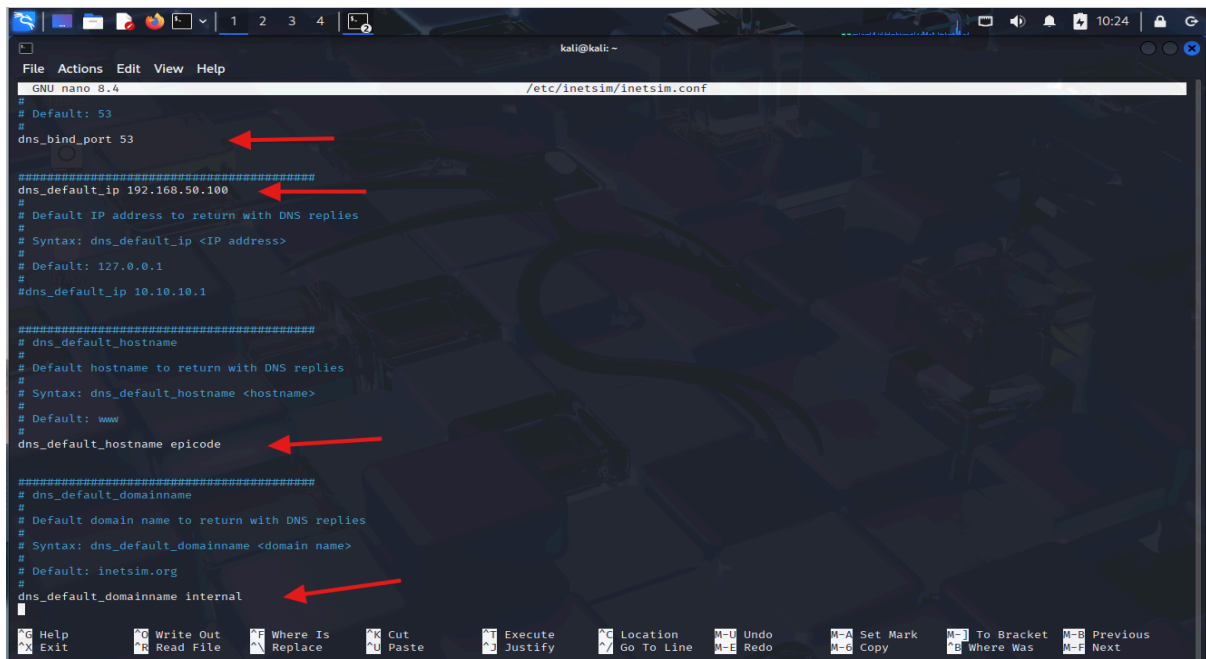
```
GNU nano 8.4 /etc/inetsim/inetsim.conf
#####
#
# INetSim configuration file
#
#####

# Main configuration
#####

# start_service
#
# The services to start
#
# Syntax: start_service <service name>
#
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
#start_service smtp
#start_service smtps
#start_service pop3
#start_service pop3s
#start_service ftp
#start_service ftps
#start_service tftp
```

Successivamente, scendendo nel file di configurazione fino alla sezione dei servizi DNS, ho:

- attivato la porta del DNS, ovvero la n. 53
- impostato l'indirizzo IP del DNS 192.168.50.100, ovvero l'indirizzo IP della macchina Kali
- impostato l'hostname "epicode" e il dominio "internal"



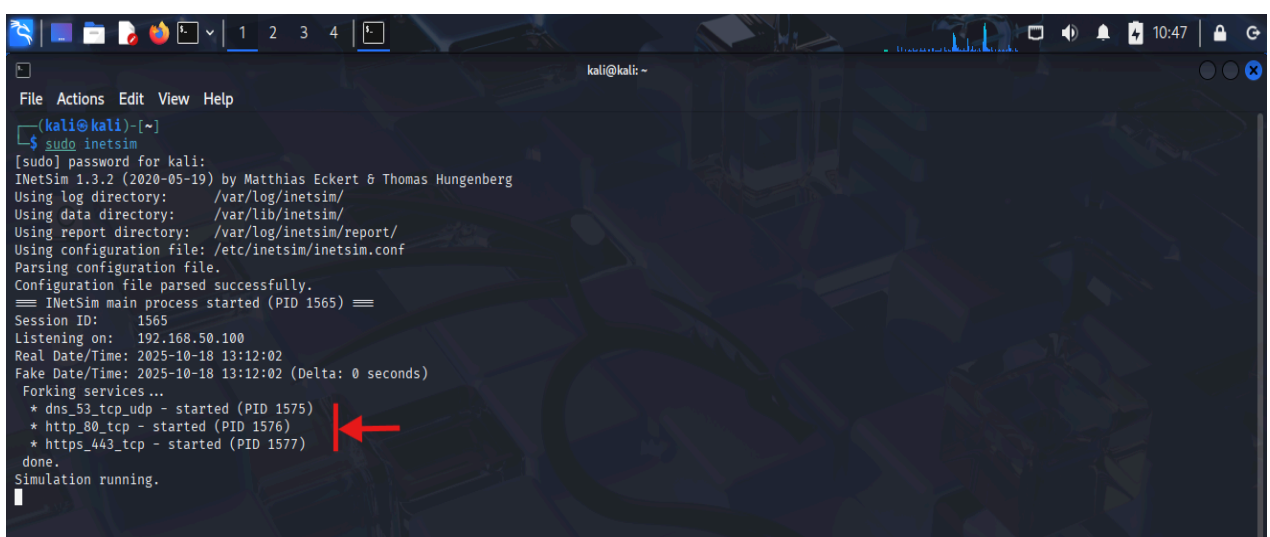
```
GNU nano 8.4 /etc/inetsim/inetsim.conf
#
# Default: 53
dns_bind_port 53

#####
dns_default_ip 192.168.50.100
#
# Default IP address to return with DNS replies
# Syntax: dns_default_ip <IP address>
# Default: 127.0.0.1
#dns_default_ip 10.10.10.1

#####
# dns_default_hostname
#
# Default hostname to return with DNS replies
# Syntax: dns_default_hostname <hostname>
# Default: www
dns_default_hostname epicode

#####
# dns_default_domainname
#
# Default domain name to return with DNS replies
# Syntax: dns_default_domainname <domain name>
# Default: inetsim.org
dns_default_domainname internal
```

Per verificare il corretto funzionamento del server e dei relativi servizi, ho avviato il tool INetSim dal terminale di kali con il comando "sudo inetsim".

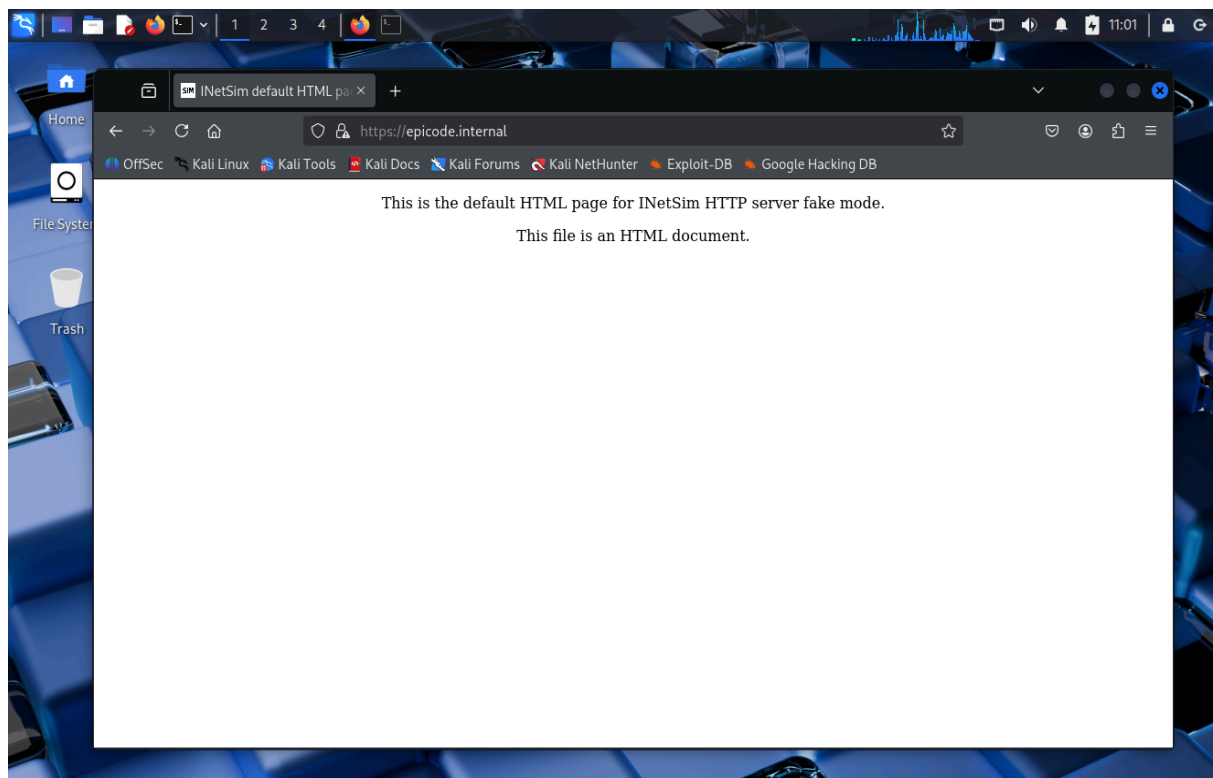


```
(kali@kali)-[~]
$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 1565) ==
Session ID: 1565
Listening on: 192.168.50.100
Real Date/Time: 2025-10-18 13:12:02
Fake Date/Time: 2025-10-18 13:12:02 (Delta: 0 seconds)
Forking services ...
* dns_53_tcp_udp - started (PID 1575)
* http_80_tcp - started (PID 1576)
* https_443_tcp - started (PID 1577)
done.
Simulation running.
```

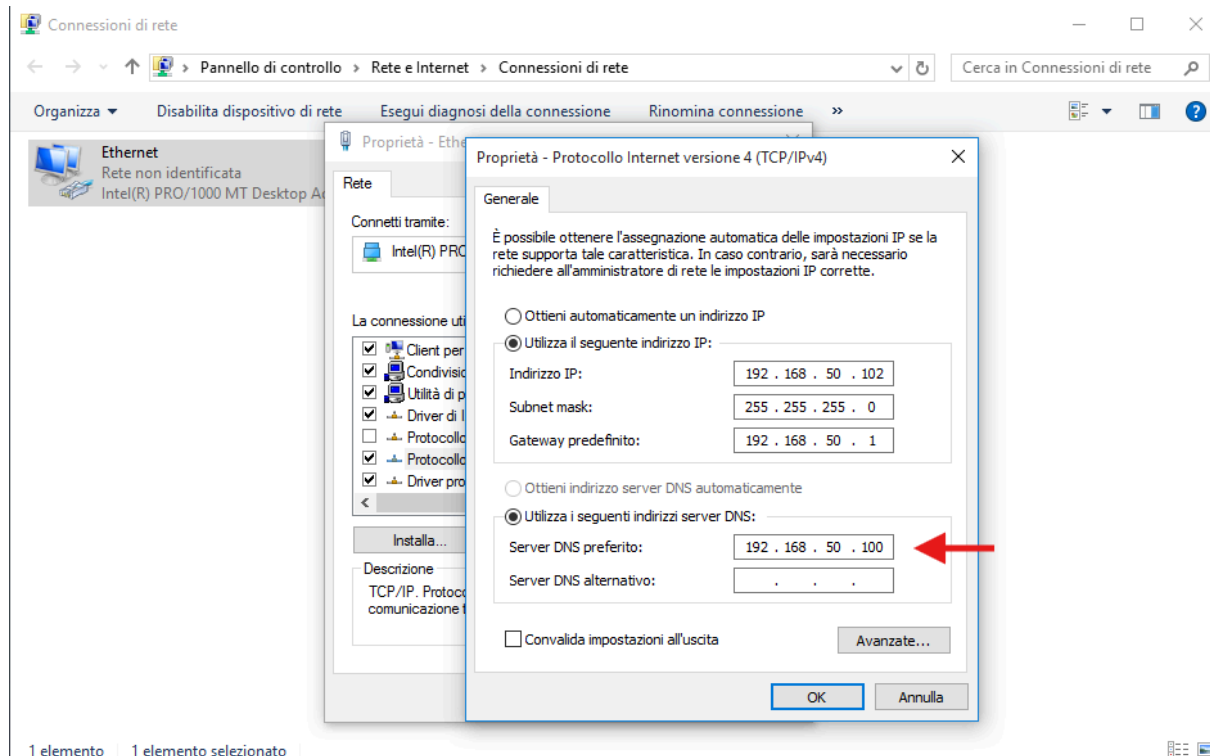
Come si può osservare dalla foto, i servizi DNS, HTTP e HTTPS risultano essere attivi.

In un primo momento, dopo aver configurato i servizi su INetSim, non era possibile accedere all'indirizzo `epicode.internal` tramite browser. Dopo diverse ricerche è emerso che il problema era legato alla versione del servizio DNS e per questo motivo ho dovuto effettuare un downgrade alla versione 1.37.

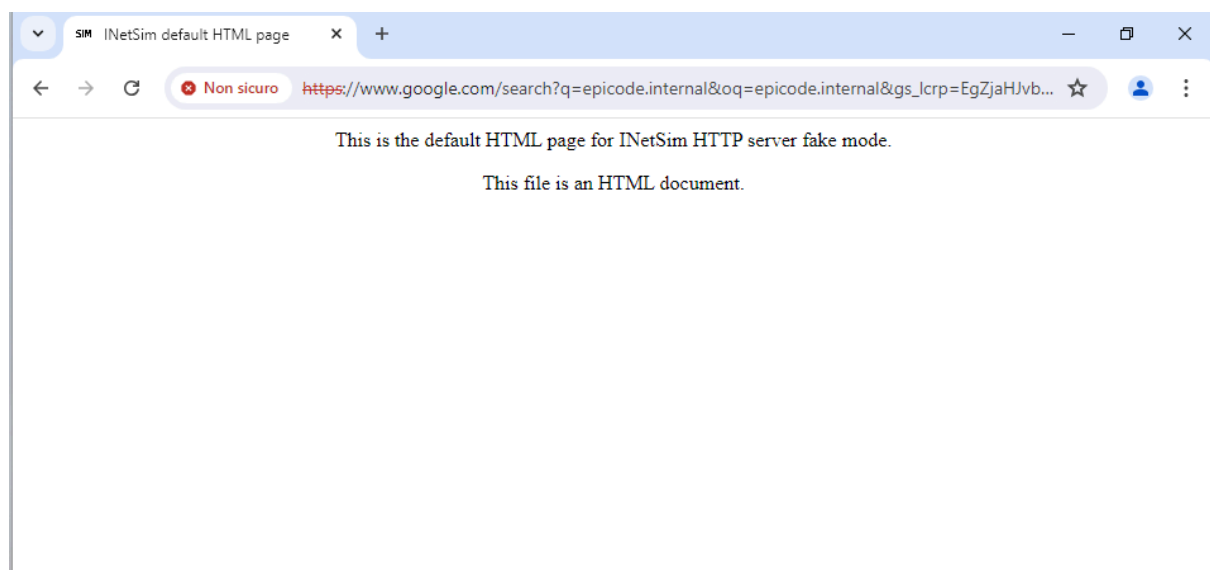
Così facendo, ho successivamente aperto il browser sulla macchina server e inserito nella barra degli indirizzi "`https://epicode.internal`". Il browser ha correttamente risposto, collegandosi al sito, confermando così che le impostazioni e i relativi parametri di configurazione sono stati applicati correttamente.



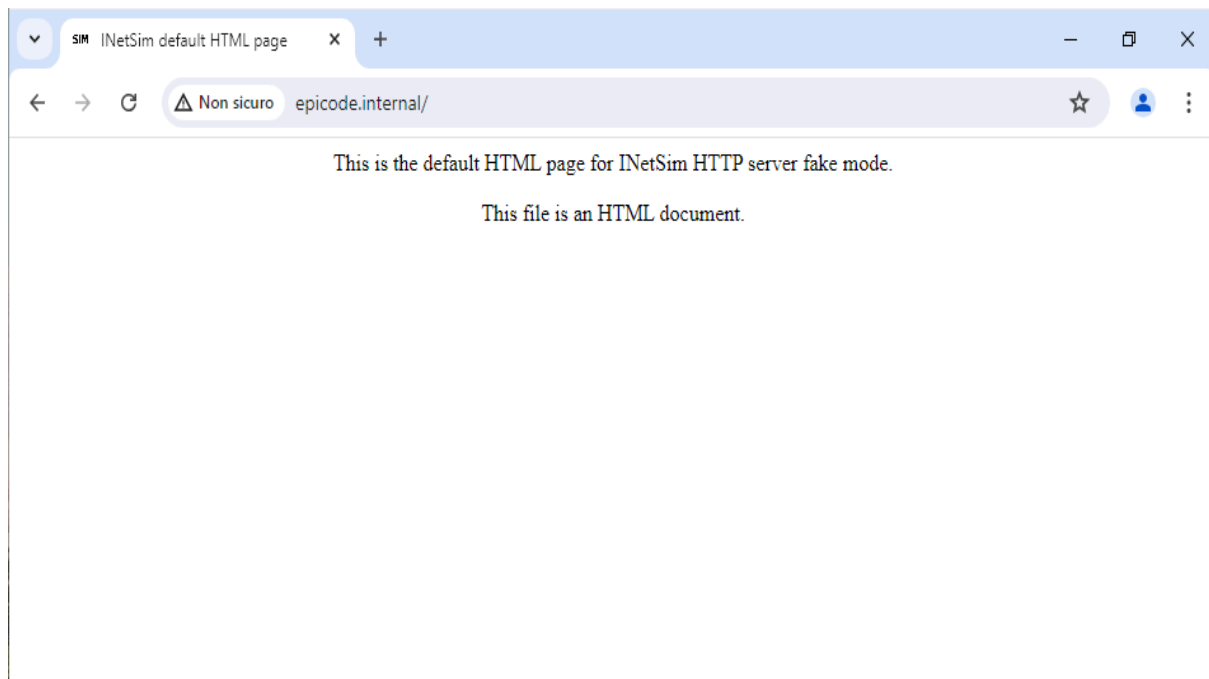
Successivamente, mi sono spostato sulla macchina client (Windows) e per prima cosa, nelle impostazioni della scheda di rete, più precisamente nelle proprietà del “Protocollo Internet versione 4 (TCP/IPv4)”, ho inserito l’indirizzo DNS 192.168.50.100 (server Kali).



Così facendo, inserendo nella barra di ricerca del browser <https://epicode.internal> e <http://epicode.internal>, la macchina client invierà la richiesta collegandosi all’hostname che risponde all’ indirizzo 192.168.50.100.



https



http

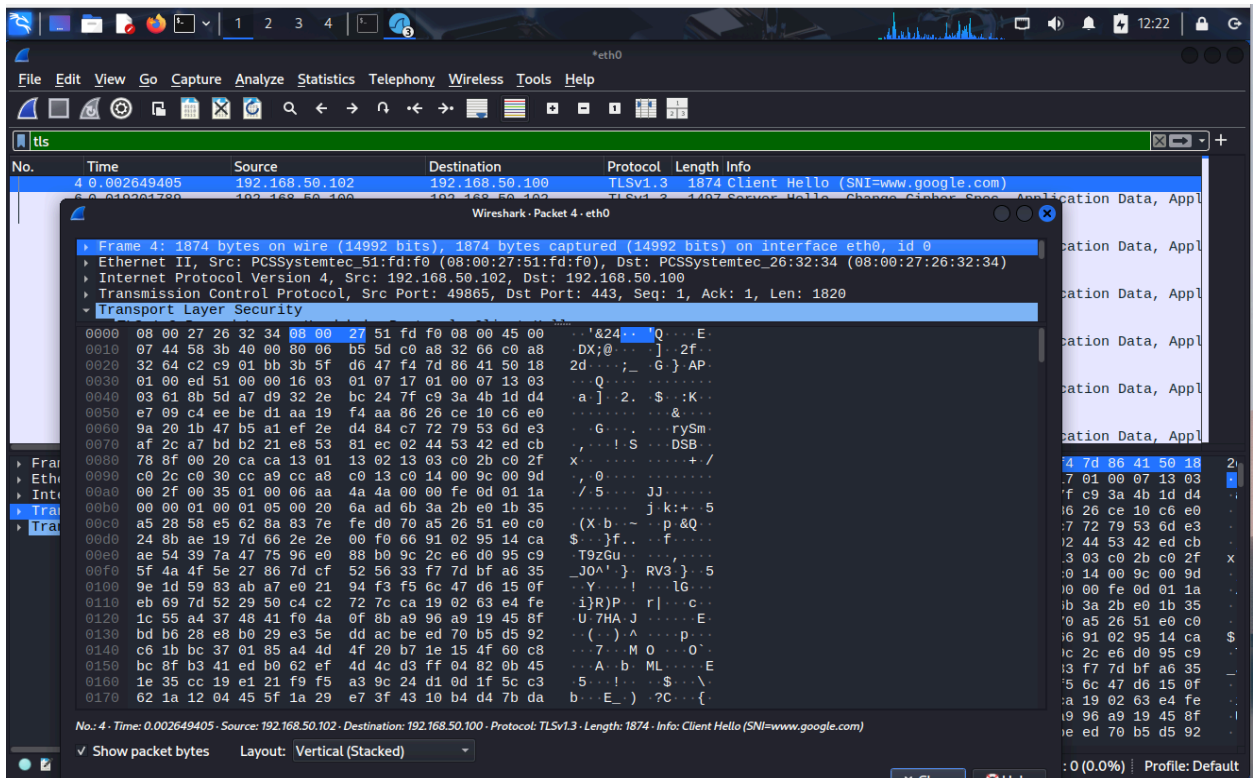
Infine, tornando sulla macchina server Kali, attraverso il tool Wireshark, ho intercettato e analizzato il traffico di rete sia in HTTP che in HTTPS andando a filtrare i pacchetti evidenziando solamente quelli che utilizzano i suddetti protocolli. Nelle foto che seguono possiamo osservare le caratteristiche di un pacchetto HTTPS e di uno HTTP.

Possiamo evidenziare il source MAC address, quello del client Windows (08:00:27:51:FD:F0) e il destination MAC address, quello del server Kali (08:00:27:26:32:34), i quali rimangono invariati in entrambi i casi.

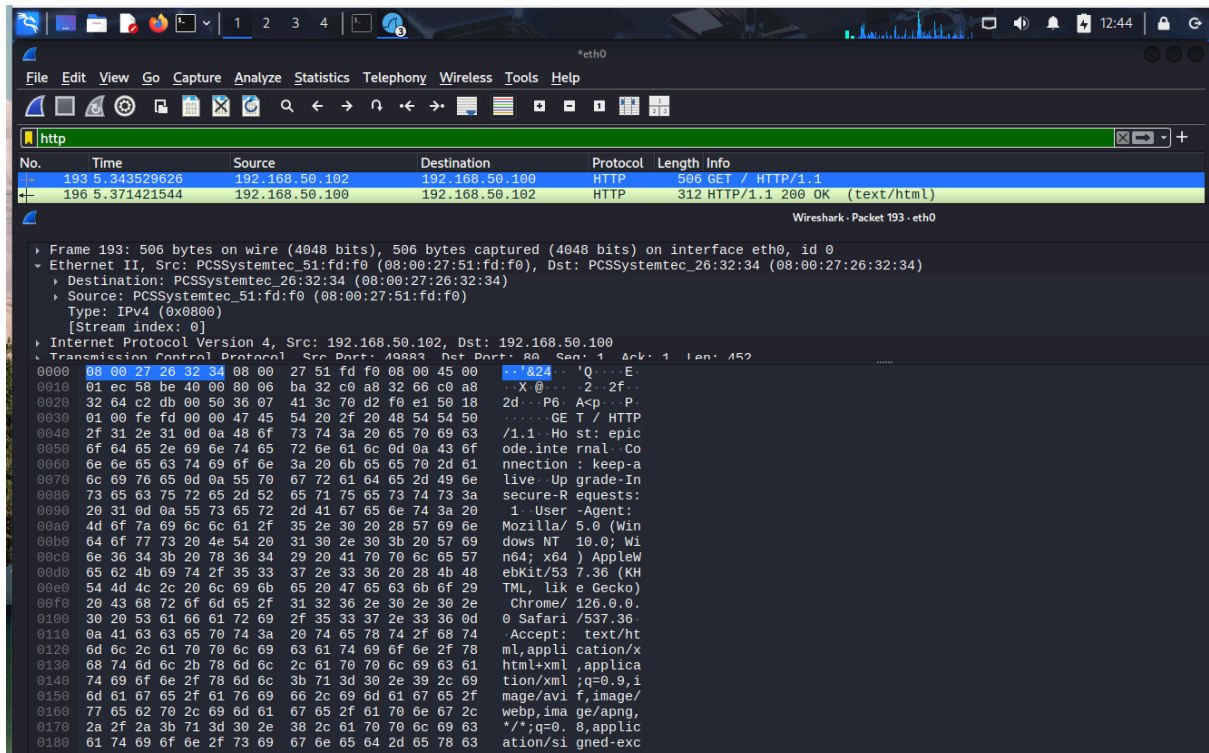
La differenza sostanziale tra questi due tipi di comunicazione sta nel contenuto della richiesta, il quale, possiamo vedere “cifrato” all’interno del pacchetto HTTPS.

Quando parliamo di pacchetti HTTPS, ci riferiamo a dati crittografati che vengono trasmessi in modo sicuro, in modo tale da non poter essere letti da chi non è autorizzato, mentre, in HTTP, header e il payload sono in chiaro.

Un’ulteriore caratteristica che possiamo osservare analizzando il traffico di rete riguarda le porte utilizzate: la porta 80 per il protocollo HTTP e la porta 443 per HTTPS.



https



http

Roma, 20/10/2025
Riccardo Ricci