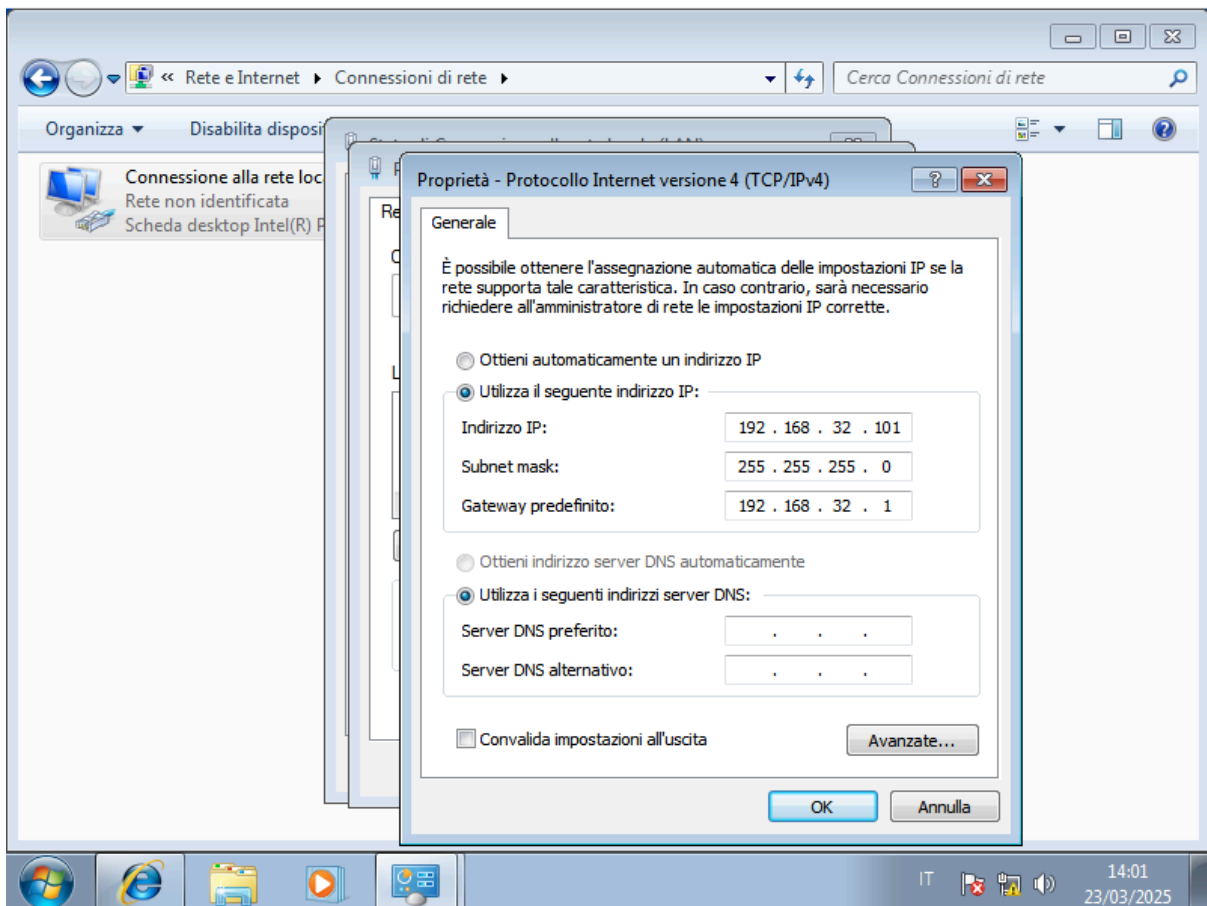


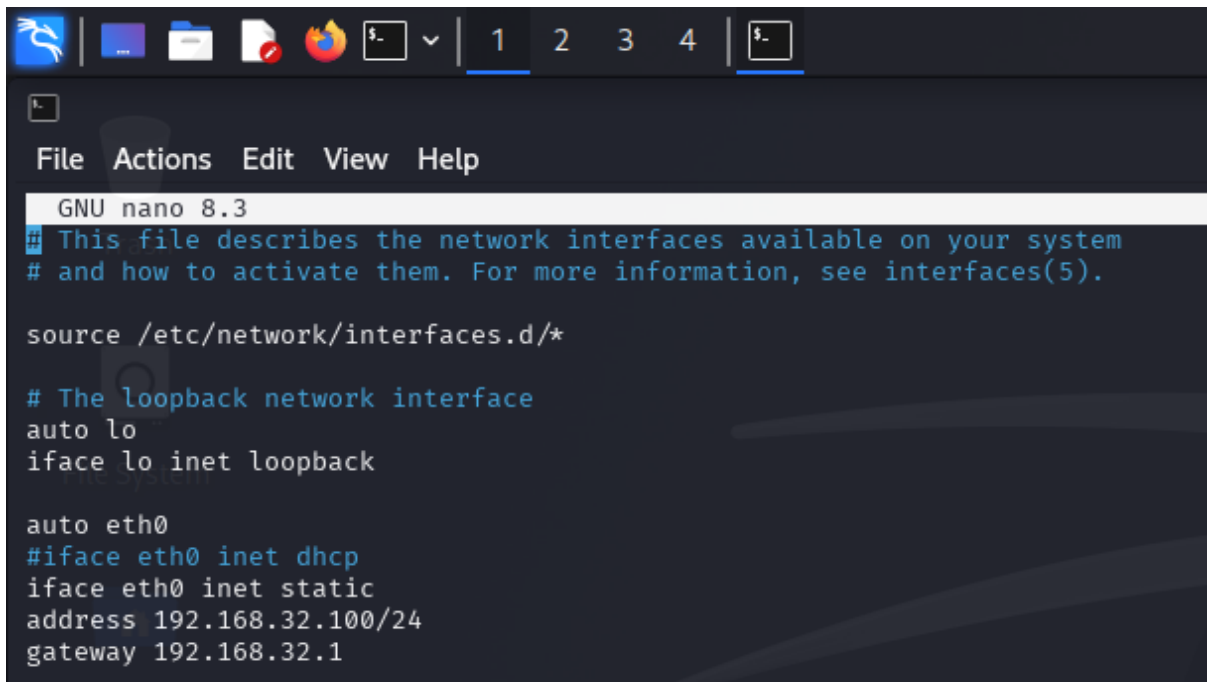
“REPORT” PROGETTO FINALE W4D4 (M1)

Il progetto di “fine modulo” richiede inizialmente la configurazione delle nostre macchine virtuali (VM), quali Kali Linux e Windows 7, con i loro corrispettivi IP (Kali Linux: 192.168.32.100 e Windows 7: 192.168.32.101). Oltre ciò bisogna abilitare i seguenti servizi: HTTPS e DNS e in seguito anche HTTP per avere un confronto tra le due tipologie. Infine attraverso “wireshark” verificare il traffico dei pacchetti , trovando anche il MAC address sorgente e destinazione. Procediamo con ordine:

Prima di tutto impostiamo gli IP stabiliti dall’esercitazione, iniziamo con windows: andando su “centro connessione di rete>modifica impostazioni scheda e selezionare il nostro dispositivo, poi proprietà>protocollo internet versione ipv4” e avremo una schermata come in figura, inserendo i nostri valori (IP, subnet mask, gateway) e salvando le modifiche su “ok”



Invece su Kali Linux tramite “Terminal Emulator” (il quadrato nero in alto a sinistra) non sarebbe altro che il prompt dei comandi di windows (cmd) e useremo il comando: “sudo nano /etc/network/interfaces” e una volta inserita la password “kali” avremo questa schermata:

A screenshot of the nano text editor interface in Kali Linux. The top bar shows the GNU nano 8.3 version and a menu with File, Actions, Edit, View, and Help. The main text area displays the contents of the /etc/network/interfaces file. The file starts with a comment explaining its purpose. Below the comment, there are two network interface configurations: 'lo' (loopback) and 'eth0' (ethernet). The 'lo' interface is configured with 'auto lo' and 'iface lo inet loopback'. The 'eth0' interface is configured with 'auto eth0', '#iface eth0 inet dhcp' (commented out), 'iface eth0 inet static', 'address 192.168.32.100/24', and 'gateway 192.168.32.1'. The editor has a dark theme with a light blue cursor at the end of the last line.

```
GNU nano 8.3
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

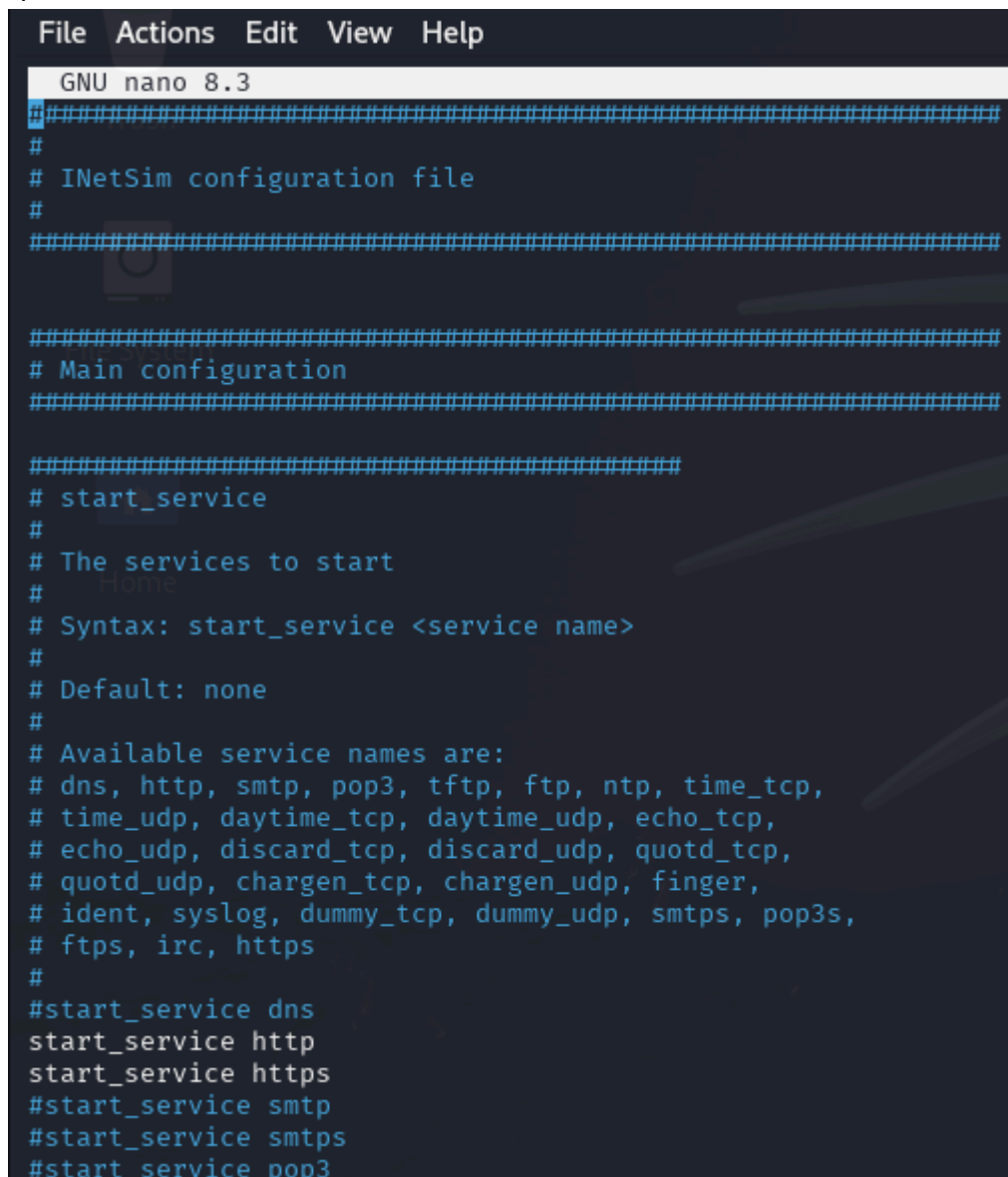
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
#iface eth0 inet dhcp
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1
```

Dovremo avere appunto l'ip corrispettivo (oltre il gateway e il subnet mask; ATTENZIONE, in questo caso corrisponde a /24 il subnet mask, non è raffigurato come su windows ma è sostanzialmente uguale la logica) , e per far avvenire le seguenti modifiche bisogna togliere “#” all'inizio della stringa altrimenti non funzionerà come vorremmo, ma soprattutto impostare “static” e non “dhcp”, in questo caso nonostante sia scritto comunque sia l'uno che l'altro, ho messo “#” così che non venisse “commentato” e che quindi usasse il mio ip statico. Per salvare tutto bisogna fare “Ctrl+x>y>invio”

Avendo configurato le nostre VM, ora bisogna andare su Kali Linux e attivare il tool già preinstallato ovvero “inetsim”, è un software gratuito che simula servizi internet comuni, come in questo caso bisogna attivare i servizi richiesti prima: DNS,HTTP,HTTPS. Per fare ciò bisogna andare nel “Terminal Emulator” e dare questo specifico comando: “sudo nano /etc/inetsim/inetsim.conf” e avremo una schermata del genere come nella figura riportata sotto:



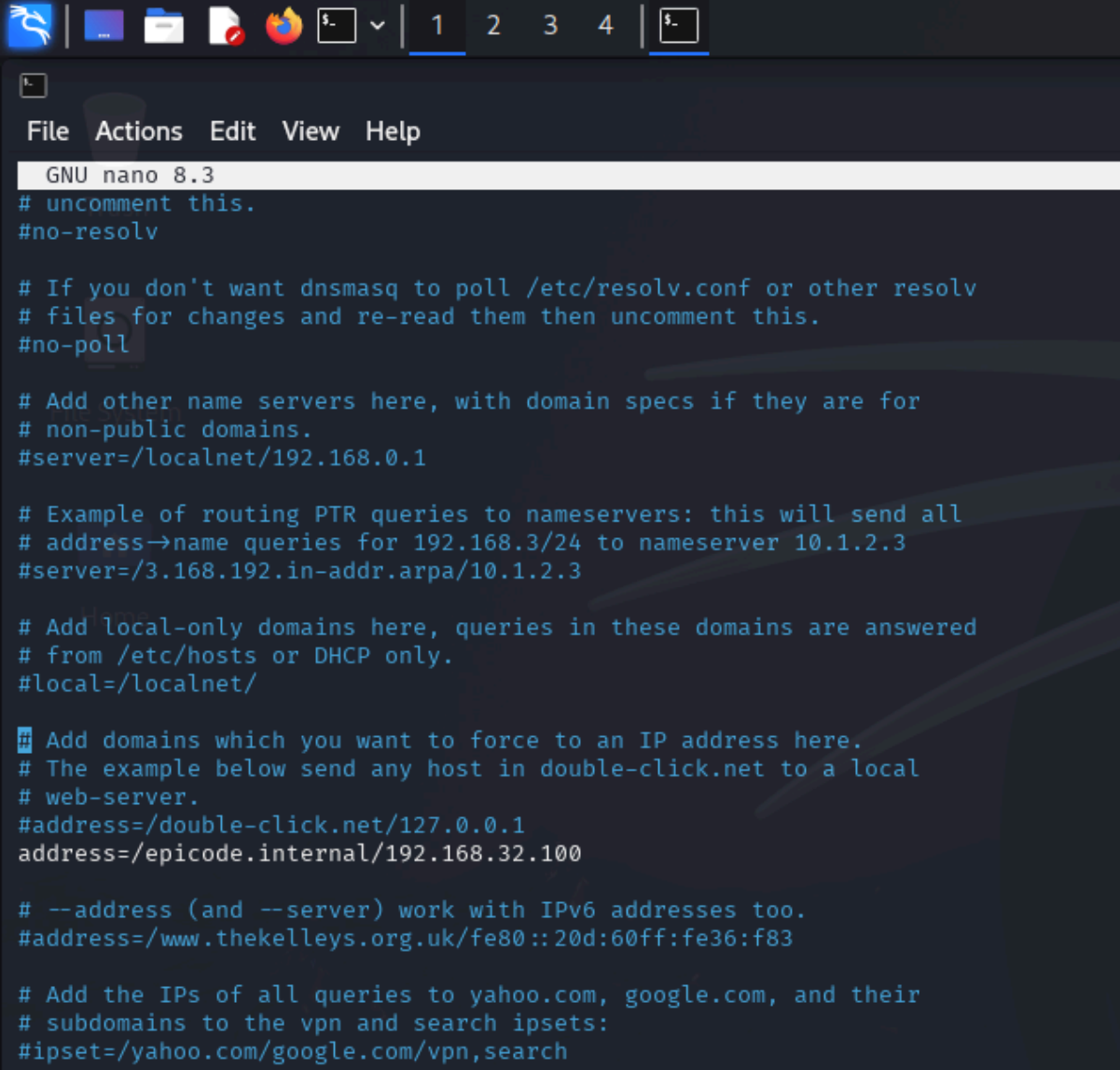
```
File Actions Edit View Help
GNU nano 8.3
#####
#
# INetSim configuration file
#
#####

#####
# 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
```

Notiamo in questo caso come il servizio HTTP e HTTPS siano già abilitati e il resto no, Perché il DNS non è abilitato se serve per la risoluzione dell’esercizio? Perché la versione da me utilizzata ha riscontrato problemi con il servizio “DNS” portandomi a procedere in maniera diversamente ma con gli stessi risultati. Per far funzionare il servizio , avremo bisogno di un aiuto esterno ovvero l’installazione di “dnsmasq” (Dnsmasq è un semplice server DNS che può essere utilizzato per reti locali) e dovremo collegare a internet la nostra macchina virtuale impostando sulla schede di rete NAT o Bridge anziché rete interna per il momento, per poter poi usare il comando: sudo apt update>invio>sudo apt install dnsmasq.

Una volta installato “dnsmasq” possiamo iniziare a configurarlo usando il comando: “sudo nano etc/dnsmasq.conf” avremo questa schermata come in figura riportata sotto



```
GNU nano 8.3
# uncomment this.
#no-resolv

# If you don't want dnsmasq to poll /etc/resolv.conf or other resolv
# files for changes and re-read them then uncomment this.
#no-poll

# Add other name servers here, with domain specs if they are for
# non-public domains.
#server=/localnet/192.168.0.1

# Example of routing PTR queries to nameservers: this will send all
# address→name queries for 192.168.3/24 to nameserver 10.1.2.3
#server=/3.168.192.in-addr.arpa/10.1.2.3

# Add local-only domains here, queries in these domains are answered
# from /etc/hosts or DHCP only.
#local=/localnet/

# Add domains which you want to force to an IP address here.
# The example below send any host in double-click.net to a local
# web-server.
#address=/double-click.net/127.0.0.1
address=/epicode.internal/192.168.32.100

# --address (and --server) work with IPv6 addresses too.
#address=/www.thekelleys.org.uk/fe80::20d:60ff:fe36:f83

# Add the IPs of all queries to yahoo.com, google.com, and their
# subdomains to the vpn and search ipsets:
#ipset=/yahoo.com/google.com/vpn,search
```

Per visualizzare ciò che si vede in figura basterà andare con le frecce della tastiera verso in basso (oppure ctrl+frecchia in basso per fare più veloce) e trovare questa stringa, in modo tale da aggiungere il nome “epicode.internal” associato al suo IP attraverso il comando riportato in “bianco”. Dopodiché una volta usciti, confermando la modifica con “Ctrl+x>y>invio”, dobbiamo verificare che il tutto sia abilitato così da essere sicuri di non avere problemi dopo.

La figura successiva riporta gli “stati” dei nostri “servizi” i quali ci servono per arrivare alla soluzione dell’esercizio. Il comando in questione è: “sudo systemctl status inetsim/dnsmasq (a seconda di quale ci serve sapere)”. Possiamo vedere come siano entrambi “inattivi” (dead).

```
(kali@kali)-[~]
$ sudo systemctl status inetsim
o inetsim.service - LSB: start and stop the internet simulation
   Loaded: loaded (/etc/init.d/inetsim; generated)
   Active: inactive (dead)
     Docs: man:systemd-sysv-generator(8)

(kali@kali)-[~]
$ sudo systemctl status dnsmasq
o dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:dnsmasq(8)

(kali@kali)-[~]
$
```

Quindi il passo successivo è “attivarli” e per farlo bisogna usare il comando: “sudo systemctl start inetsim” e poi lo stesso con dnsmasq.

Dopo averli attivati e verificato il loro “stato” dovremo avere una schermata di questo tipo:

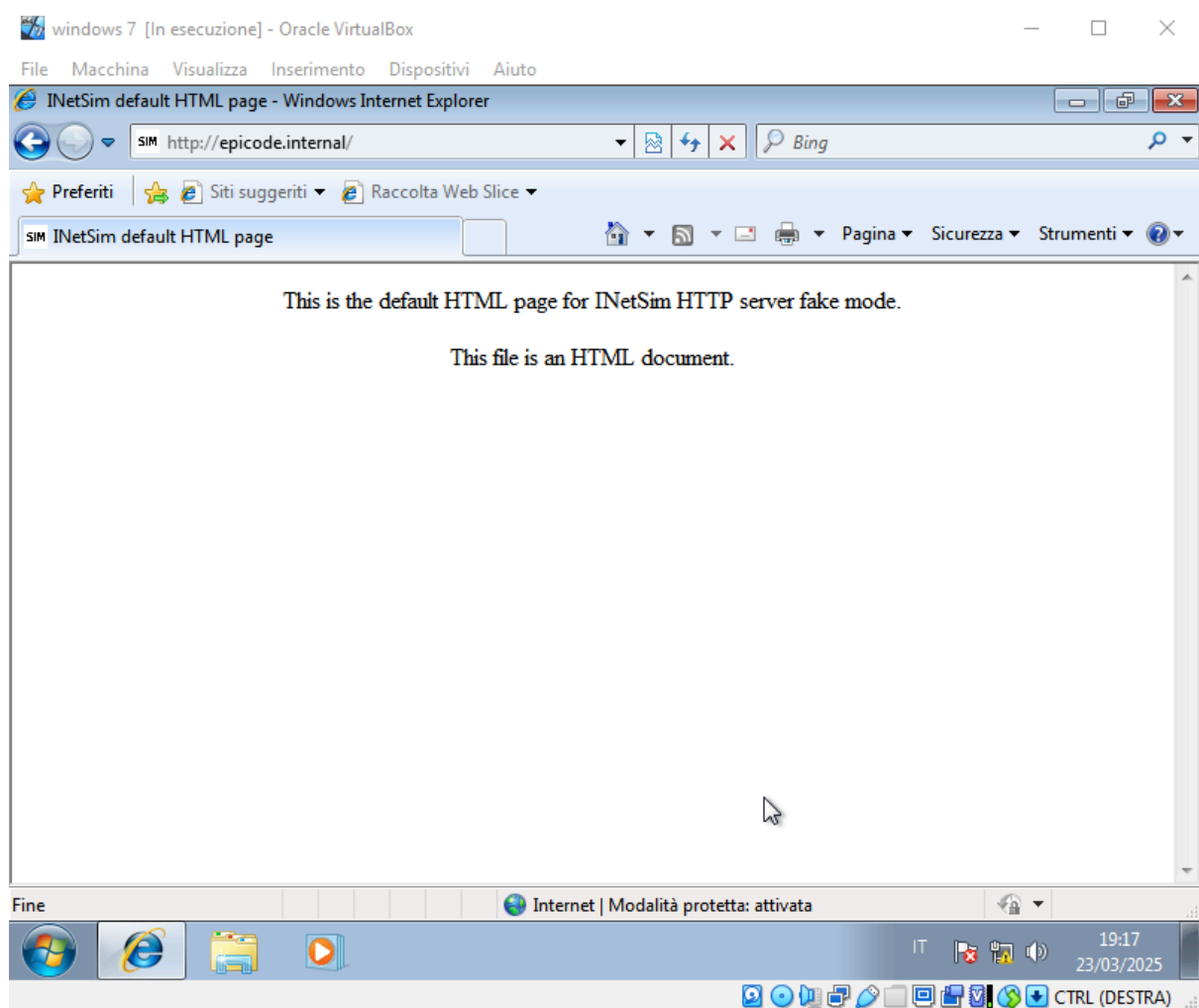
```
(kali@kali)-[~]
$ sudo systemctl status inetsim
● inetsim.service - LSB: start and stop the internet simulation
   Loaded: loaded (/etc/init.d/inetsim; generated)
   Active: active (running) since Sun 2025-03-23 14:03:25 EDT; 6s ago
     Invocation: cb08fef80676430bb3be1cb57132b174
       Docs: man:systemd-sysv-generator(8)
   Process: 11877 ExecStart=/etc/init.d/inetsim start (code=exited, status=0/SUCCESS)
     Tasks: 3 (limit: 2210)
    Memory: 38.8M (peak: 39.4M)
       CPU: 325ms
    CGroup: /system.slice/inetsim.service
            └─11887 inetsim_main
              └─11890 inetsim_http_80_tcp
                └─11891 inetsim_https_443_tcp

Mar 23 14:03:23 kali systemd[1]: Starting inetsim.service - LSB: start and stop the internet simulation...
Mar 23 14:03:25 kali inetsim[11877]: Starting Internet Service Simulation Suite: inetsim.
Mar 23 14:03:25 kali systemd[1]: Started inetsim.service - LSB: start and stop the internet simulation.

(kali@kali)-[~]
$ sudo systemctl status dnsmasq
● dnsmasq.service - dnsmasq - A lightweight DHCP and caching DNS server
   Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service; disabled; preset: disabled)
   Active: active (running) since Sun 2025-03-23 14:01:47 EDT; 1min 59s ago
     Invocation: 36885030c1de40bb9d465e859f9cc0ed
       Docs: man:dnsmasq(8)
   Process: 10952 ExecStartPre=/usr/share/dnsmasq/systemd-helper checkconfig (code=exited, status=0/SUCCESS)
   Process: 10957 ExecStart=/usr/share/dnsmasq/systemd-helper exec (code=exited, status=0/SUCCESS)
   Process: 10964 ExecStartPost=/usr/share/dnsmasq/systemd-helper start-resolvconf (code=exited, status=0/SUCCESS)
    Main PID: 10963 (dnsmasq)
       Tasks: 1 (limit: 2210)
    Memory: 2.2M (peak: 3.7M)
       CPU: 35ms
    CGroup: /system.slice/dnsmasq.service
            └─10963 /usr/sbin/dnsmasq -x /run/dnsmasq/dnsmasq.pid -u dnsmasq -7 /etc/dnsmasq.d,.dpkg-dist,.dpkg-old,
```

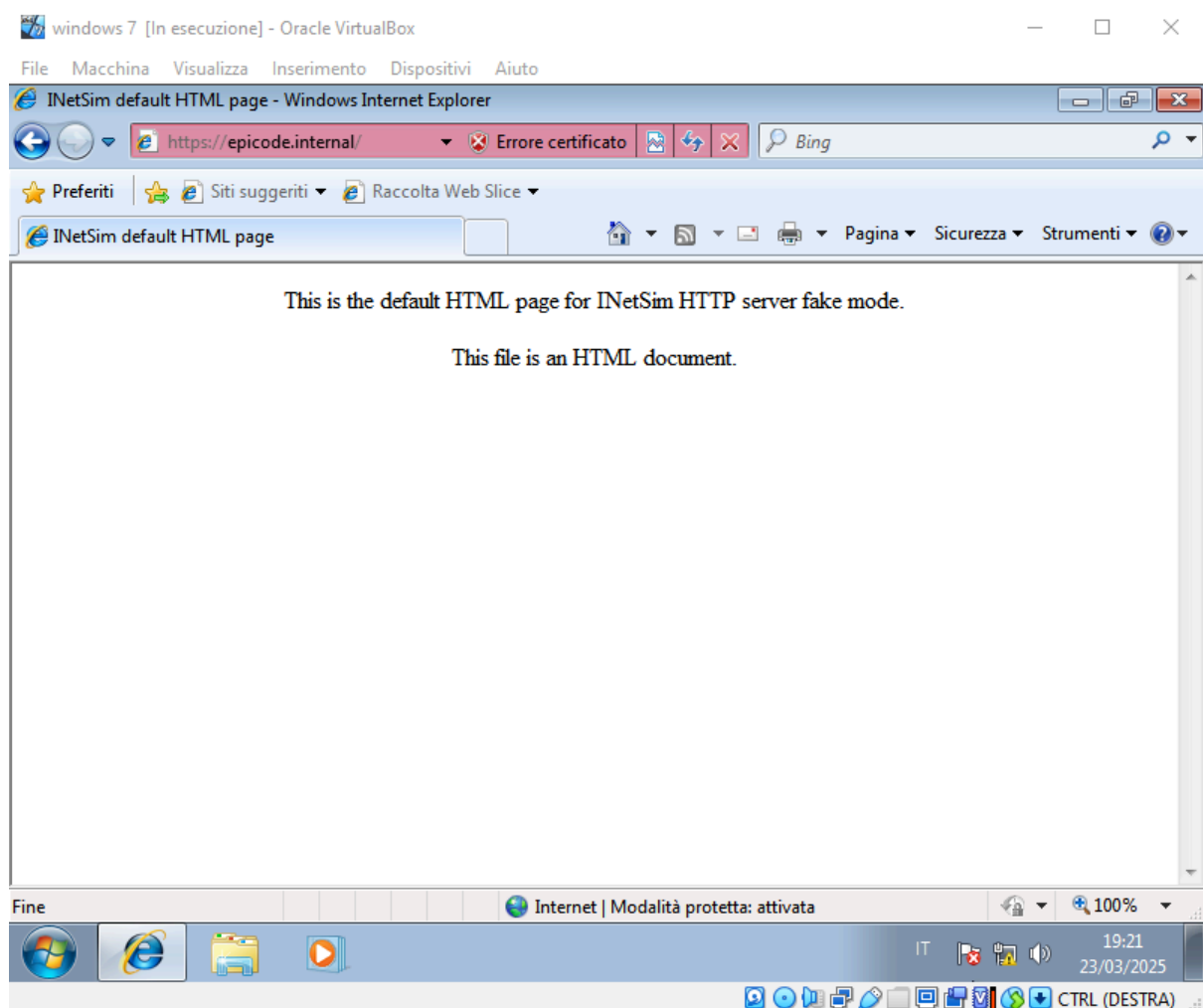
noteremo come i “servizi” (DNS,HTTPS,HTTP) siano “attivi” e pronti. active (running)

Ora torniamo su Windows 7 e apriamo internet explorer e mettiamo nella barra di ricerca (URL) il seguente indirizzo: <http://epicode.internal>



Questo è ciò che vedremo se tutto è andato bene, vuol dire che il DNS sta facendo il suo lavoro, ovvero associa il nome al suo indirizzo.

Uguale sarà per HTTPS:

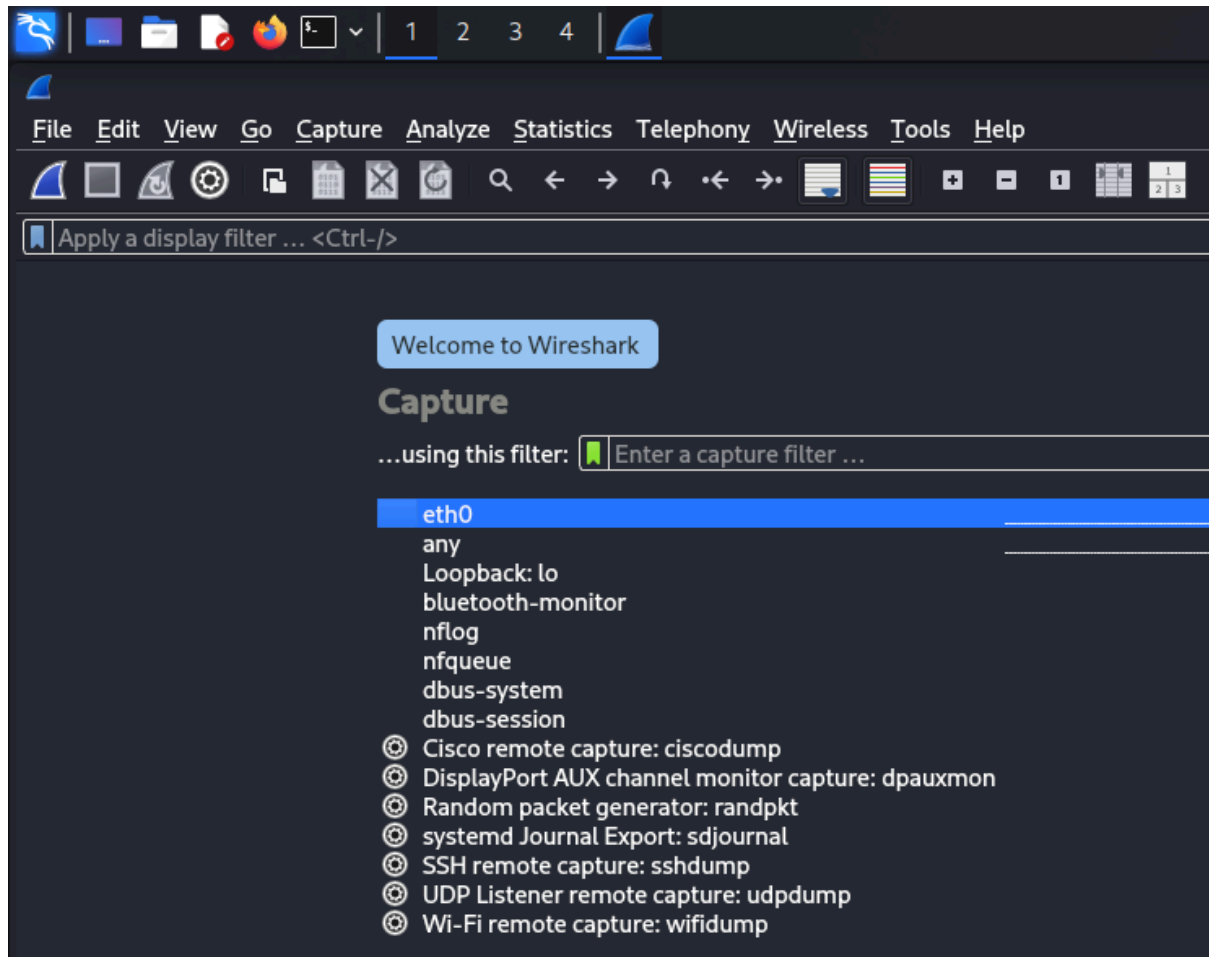


Se si nota bene in alto, ci dà l'errore del certificato, questo perché il servizio HTTPS ha una funzione diversa dall' HTTP, ovvero fornisce più sicurezza su quel determinato sito. In che modo? Attraverso dei certificati SSL validi e configurati correttamente emessi da un'autorità di certificazione attendibile, proteggendo i dati sensibili e garantendo che il sito sia "sicuro" e non una truffa, ma in questo caso non essendoci, windows ci avvisa che il sito "potrebbe" non essere sicuro per la mancanza del certificato stesso.

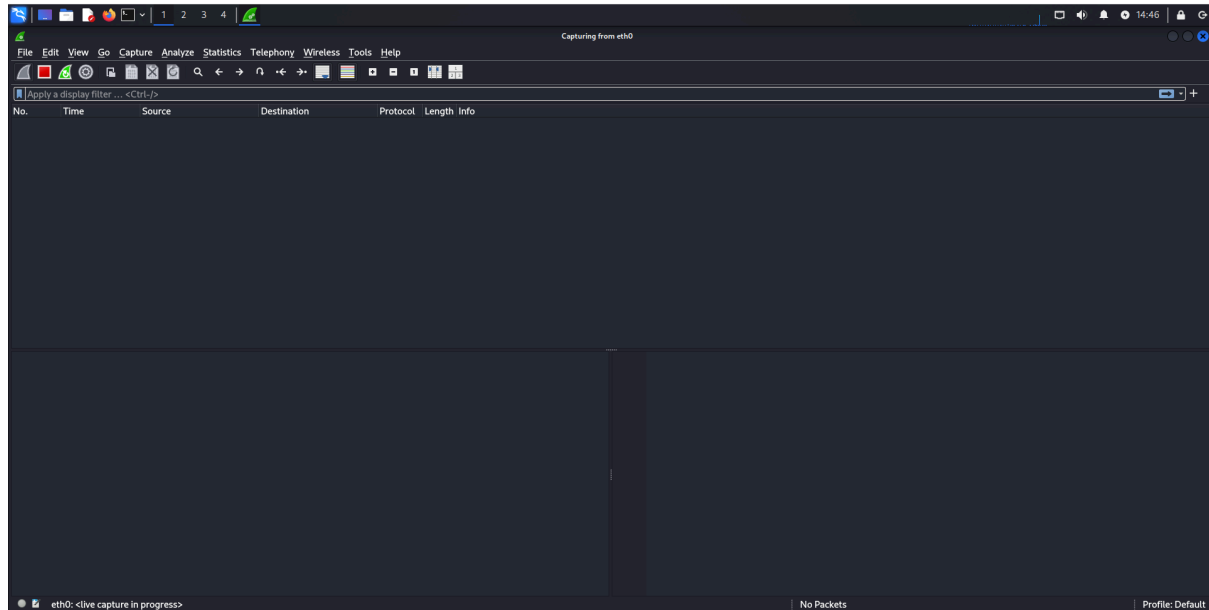
Come ultimo svolgimento dell'esercizio avremo bisogno del tool "wireshark" già presente su kali linux perchè preinstallato, andando in alto a sinistra sulla barra di ricerca e digitare il nome stesso o cercarlo tra i vari tool.

"Wireshark" è un network sniffer e packet analyzer, ovvero analizza qualsiasi pacchetto, flusso di traffico o connessione che passa attraverso la scheda di rete del pc.

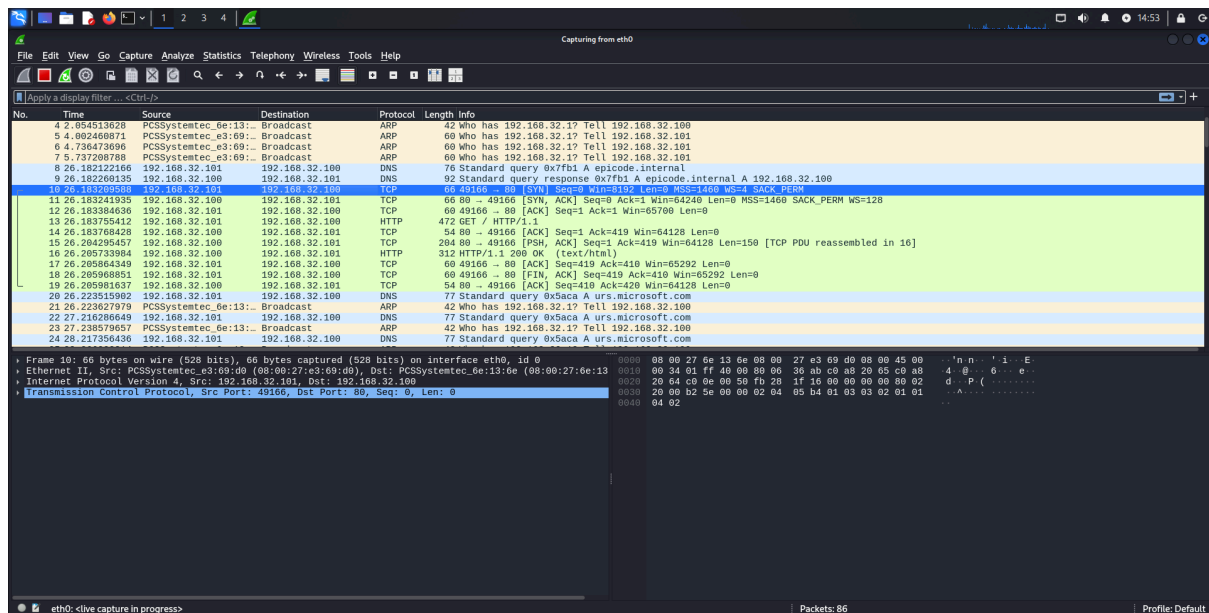
Qui sotto in figura, è la schermata che vedremo una volta aperto il programma e selezioniamo "eth0", la nostra NIC (Network Interface Card) (comunica con le altre VM)



Se la schermata risulterà come in figura vorrà dire che, windows non ha ancora provato a raggiungere il sito “epicode.internal”. Quindi abilitando DNS,HTTP e HTTPS e proveremo a raggiungere il sito in questione, noteremo un cambiamento di questa schermata.

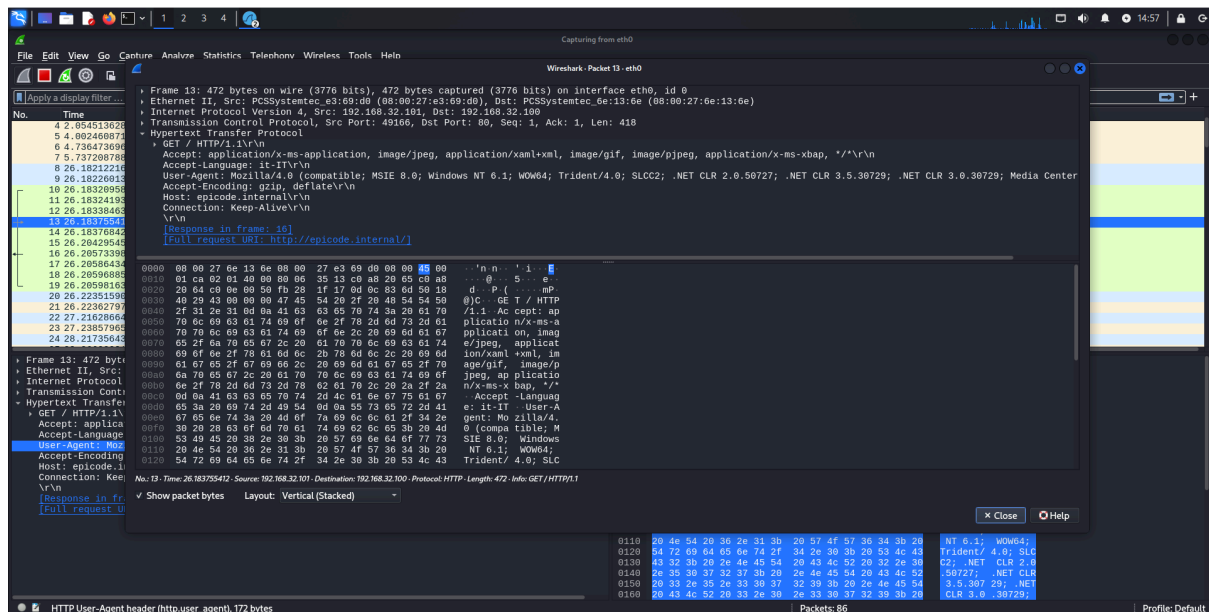


Una volta aver cercato <http://epicode.internal> su windows, vedremo come questa serie di numeri o parole non sono altro che il flusso del traffico dei pacchetti tra kali e windows

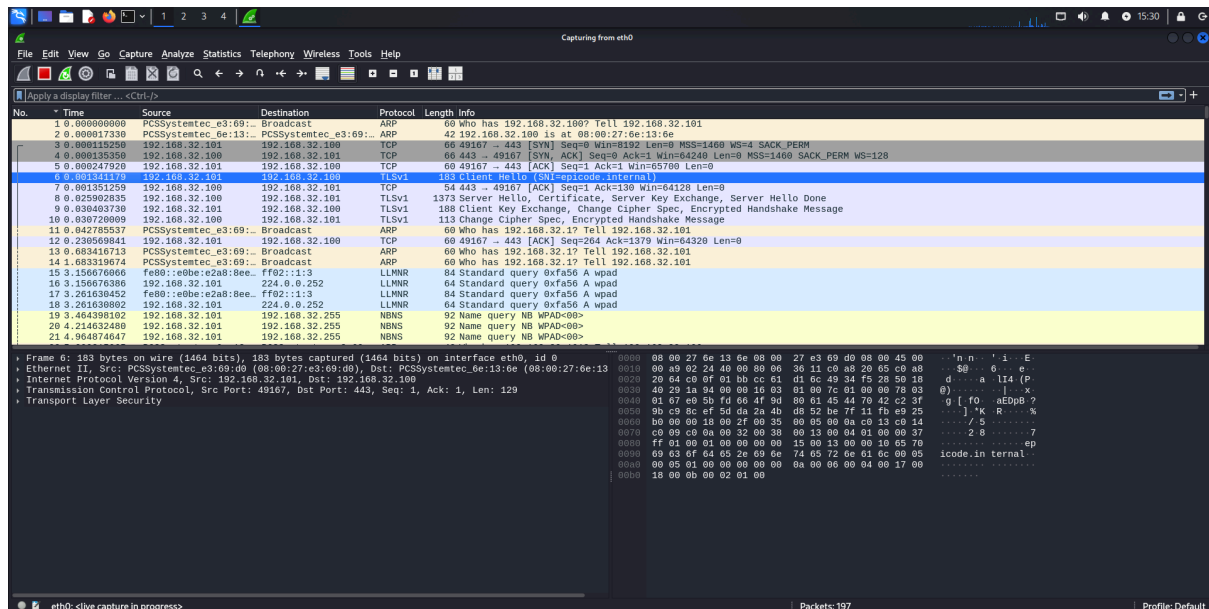


Notiamo subito all'occhio i vari protocolli come: ARP,DNS,TCP,HTTP e la stretta di mano a 3 step (SYN, SYN ACK, ACK) e gli IP delle VM.

Prendendo il pacchetto “13” notiamo subito anche il MAC address delle nostre due macchine virtuali , sorgente (source/src) è windows in questo caso e il MAC corrisponde a: (08:00:27:e3:69:d0) mentre destinatario (destination/dst) è kali linux e il suo MAC corrisponde a: (08:00:27:6e:13:6e) , e possiamo anche vedere le informazioni del pacchetto in questione sulla colonna di destra. Durante una connessione HTTP, il client ed il server si scambiano messaggi. I messaggi inviati dal client prendono il nome di “request”, mentre quelli inviati in risposta dal server in “response”. GET: viene utilizzato quando si richiede una risorsa web e sotto notiamo il resto della richiesta.

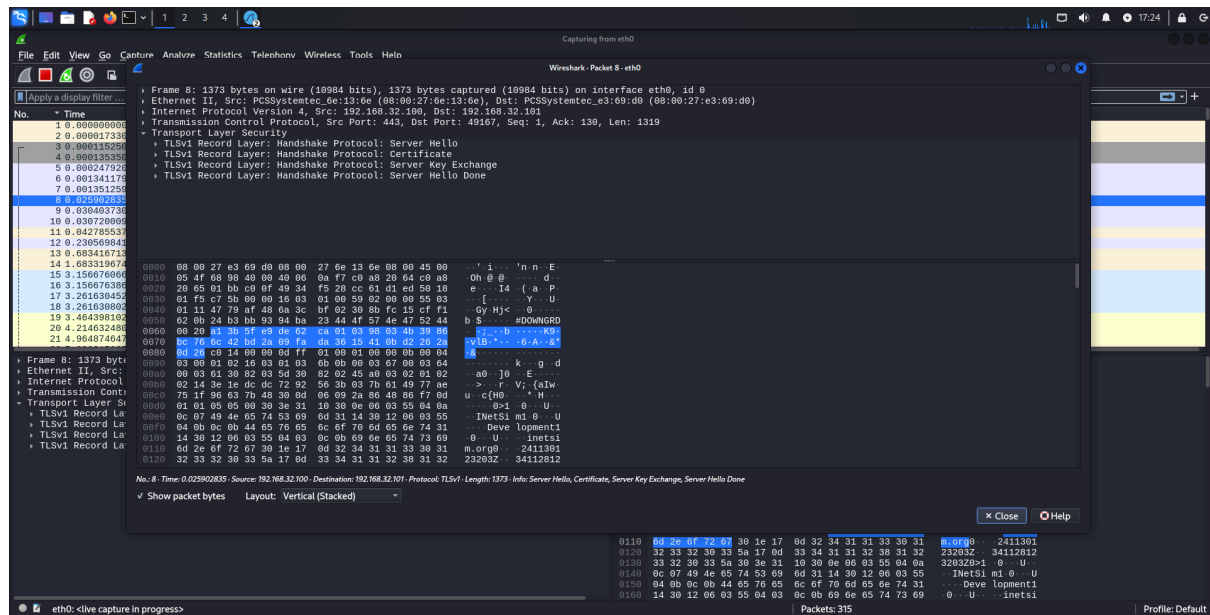


Ora proviamo a raggiungere il sito con <https://epicode.internal> e vediamo se cambia qualcosa.



Il protocollo HTTP per l'appunto non c'è. Un SSL sta per “Secure Socket Layer”, TLS (TLSv1) per “Transport Layer Security”. Entrambi sono protocolli di crittografia per il livello di trasporto di internet. Il loro compito è criptare i flussi di dati tra client e server.

Analizzando i pacchetti, sulla colonna di destra non riusciremo a ricavarne nulla, solo lettere e simboli casuali. Il MAC address è sempre lo stesso per windows e kali.



In conclusione: abbiamo impostato i vari IP delle VM e abilitato i servizi come DNS,HTTP,HTTPS, verificando tramite Windows il corretto raggiungimento del sito “epicode.internal”. Invece con Wireshark su Kali abbiamo notato i MAC address di Kali e Windows , a seguito anche delle differenze tra HTTP e HTTPS nella lettura dei pacchetti.