ESERCITAZIONE FINE MODULO 1
BARRECA MONICA

Per poter svolgere l'esercizio richiesto come prima cosa bisogna configurare le macchine con i requisiti richiesti, quindi:

- Kali Linux IP 192.168.32.100
  Per impostare l'indirizzo ip statico richiesto, eseguo il comando **sudo nano /etc/nertwork/interfaces** (utilizzo sudo perché mi permette di salvare le modifiche effettuate) per aprire l'interfaccia di rete e impostare l'indirizzo IP richiesto, cambiando di conseguenza anche il gateway, così come in figura

```
File  Actions  Edit  View  Help
  GNU nano 7.2                                  /etc/network/interfaces
# 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 static
address 192.168.32.100/24
gateway 192.168.32.1
```

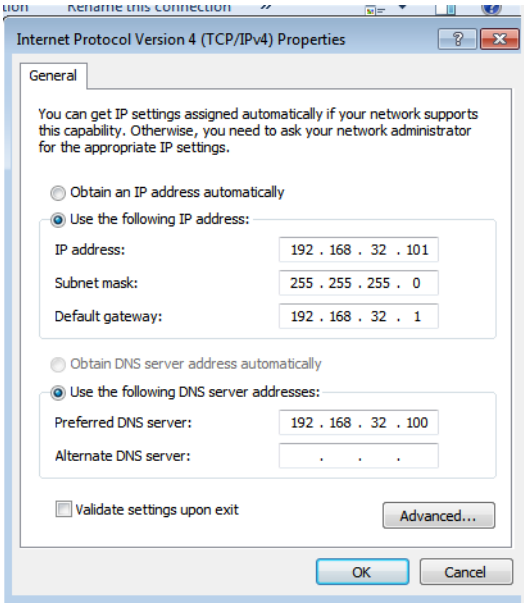
  Dopo di che, lancio il comando **sudo systemctl restart networking** per riavviare la rete e verificare ulteriormente col comando **ifconfig** che le impostazioni siano correttamente configurate.

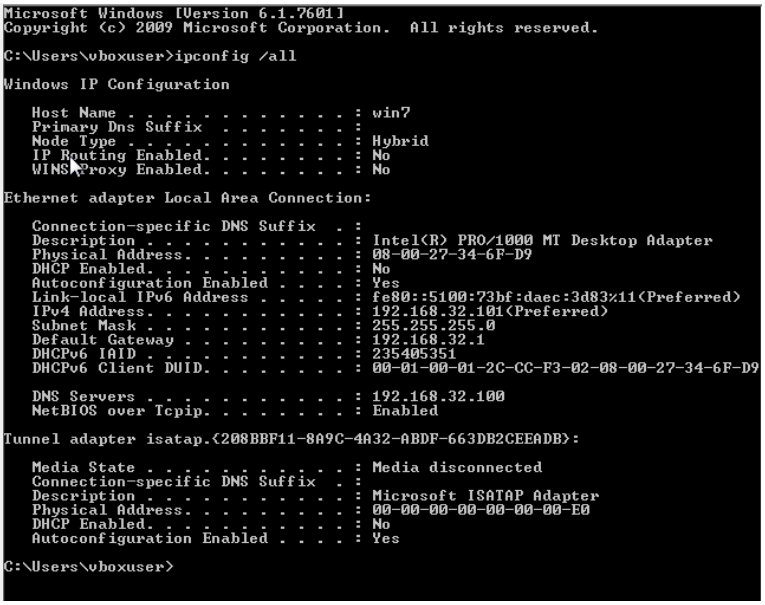
```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.32.100  netmask 255.255.255.0  broadcast 192.168.32.255
        inet6 fe80::a00:27ff:fecb:7ef5  prefixlen 64  scopeid 0×20<link>
        ether 08:00:27:cb:7e:f5  txqueuelen 1000  (Ethernet)
        RX packets 73  bytes 5588 (5.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 56  bytes 5950 (5.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

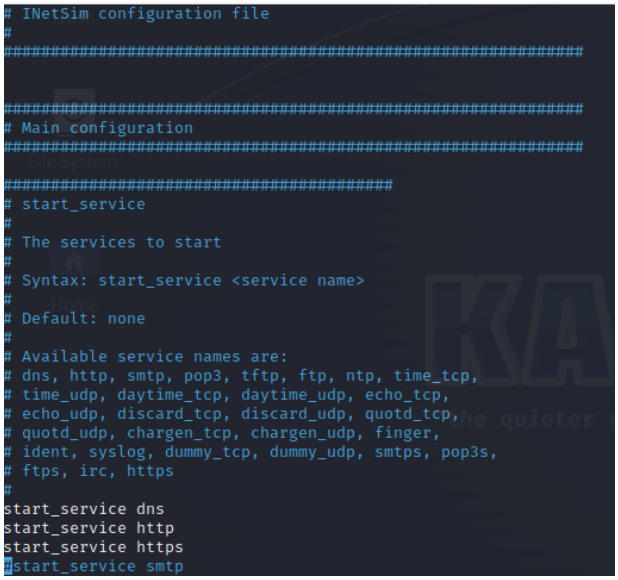
- Windows 7 IP 192.168.32.101
  Per modificare l'indirizzo IP della scheda di rete su Win7 bisogna accedere da **Control Panel -> Network and Internet -> Network and Sharing Center -> Change adapter settings** poi sulla scheda di rete tasto destro e seleziono **properties**, doppio clic su **Internet Protocol Version 4** e modifico manualmente l'indirizzo IP, già che ci sono imposto anche l'indirizzo IP del server DNS con l'indirizzo IP di Kali, per verificare che tutto sia configurato correttamente lancio il comando **ipconfig /all**

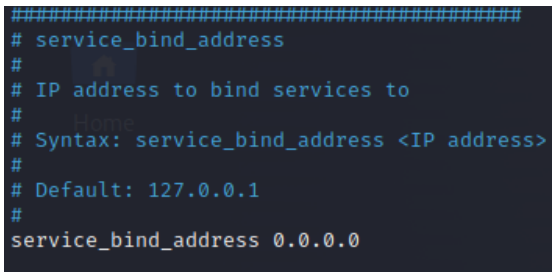
ESERCITAZIONE FINE MODULO 1
BARRECA MONICA

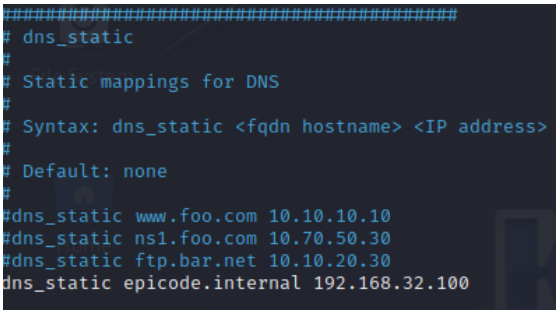




- Servizio HTTPS e DNS attivo
  Per poter attivare i servizi HTTPS e DNS utilizziamo il software INetSim già installato su Kali Linux, che ci permette di simulare i servizi tipici di Internet
  Per accedere e modificare le configurazioni di INetSim, entriamo nella directory con il comando **cd /etc/inetsim** e poi **ls** che ci permette di visualizzare i nomi dei file all'interno della directory, apriamo quindi il file con **sudo nano inetsim.conf** così possiamo commentare tutto quello che non ci serve e lasciare attivi i servizi DNS – HTTPS – HTTP (per il confronto successivo) e modificare anche:
  - l'indirizzo IP del service_bind_address in 0.0.0.0
  - nel DNS statico inserire dns_static epicode.internal 192.168.32.100 così da creare un record A e associare il nome "epicode.internal" all'indirizzo IP richiesto, cioè a Kali

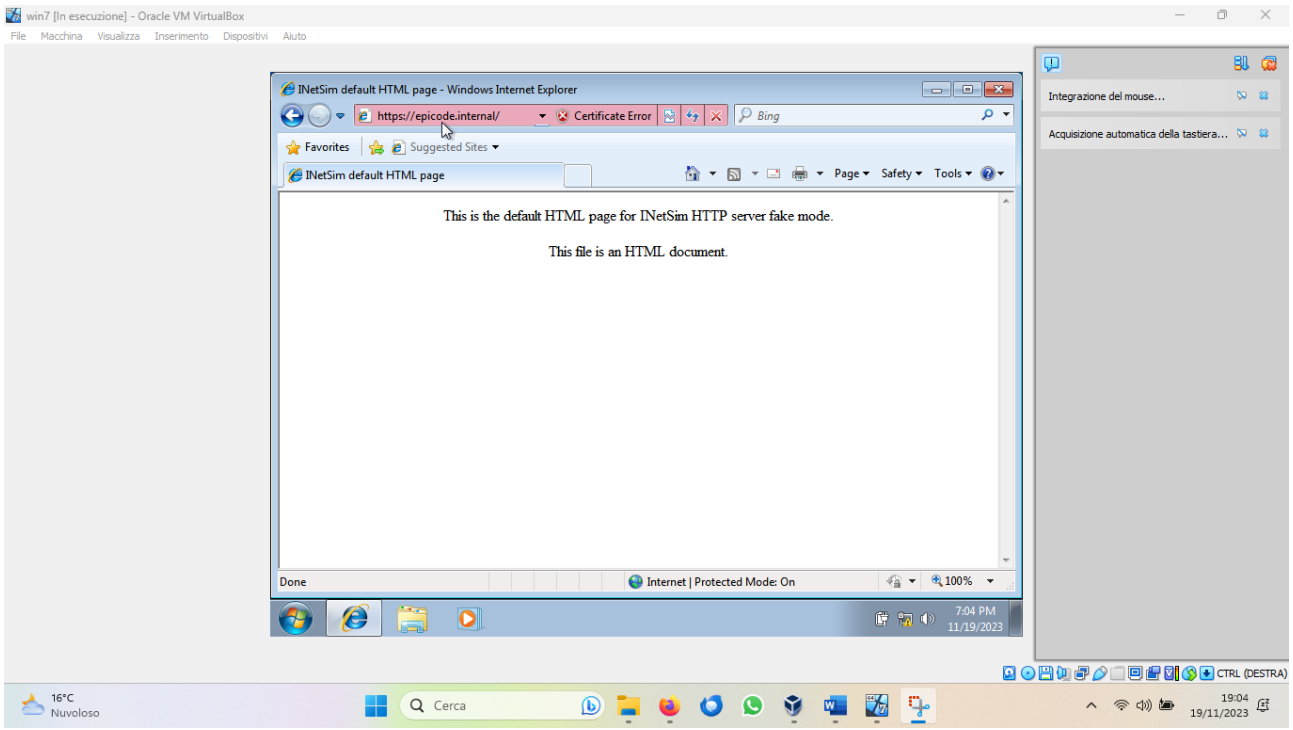




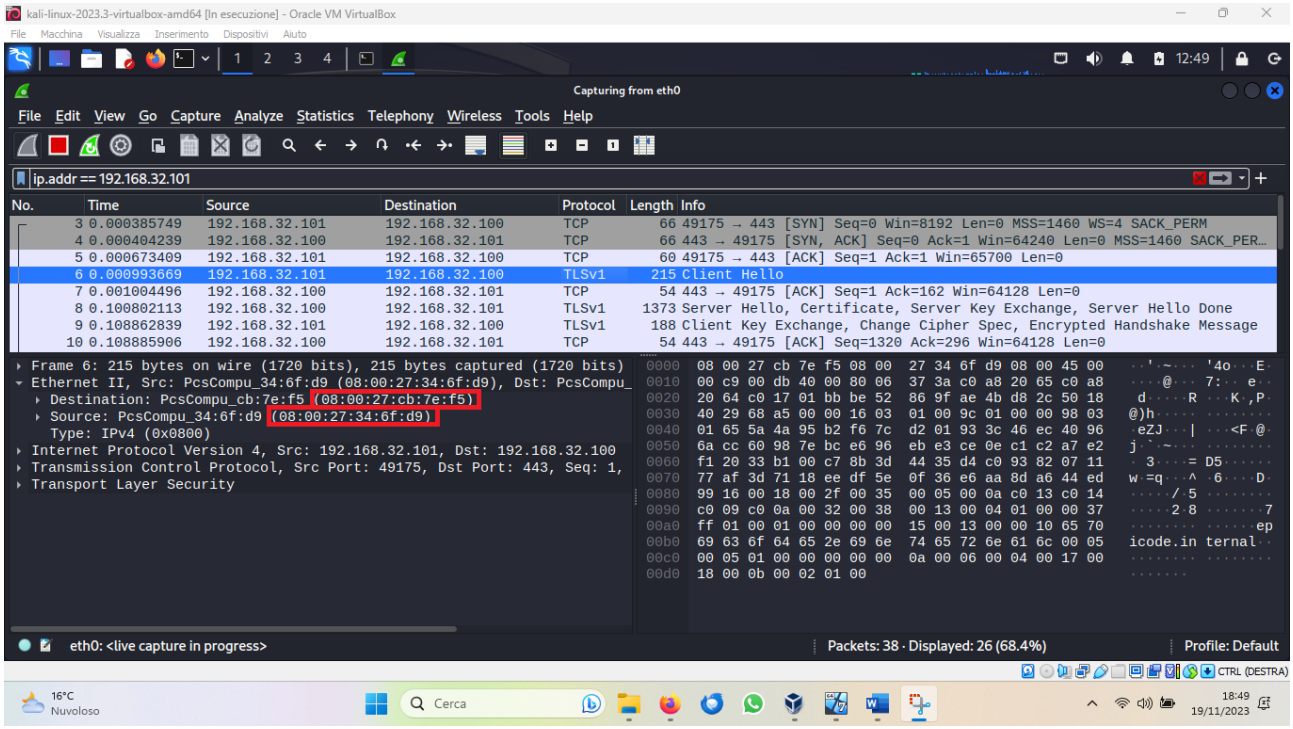


Una volta correttamente configurato il laboratorio, su Kali avviamo INetSim con il comando **sudo inetsim**, poi apriamo anche Wireshark in modo da catturare i pacchetti della scheda di rete eth0, inseriamo anche il filtro ip.addr == 192.168.32.101, così possiamo vedere lo scambio che avviene quando da win7 cerchiamo sul browser l'indirizzo https://epicode.internal

ESERCITAZIONE FINE MODULO 1
BARRECA MONICA

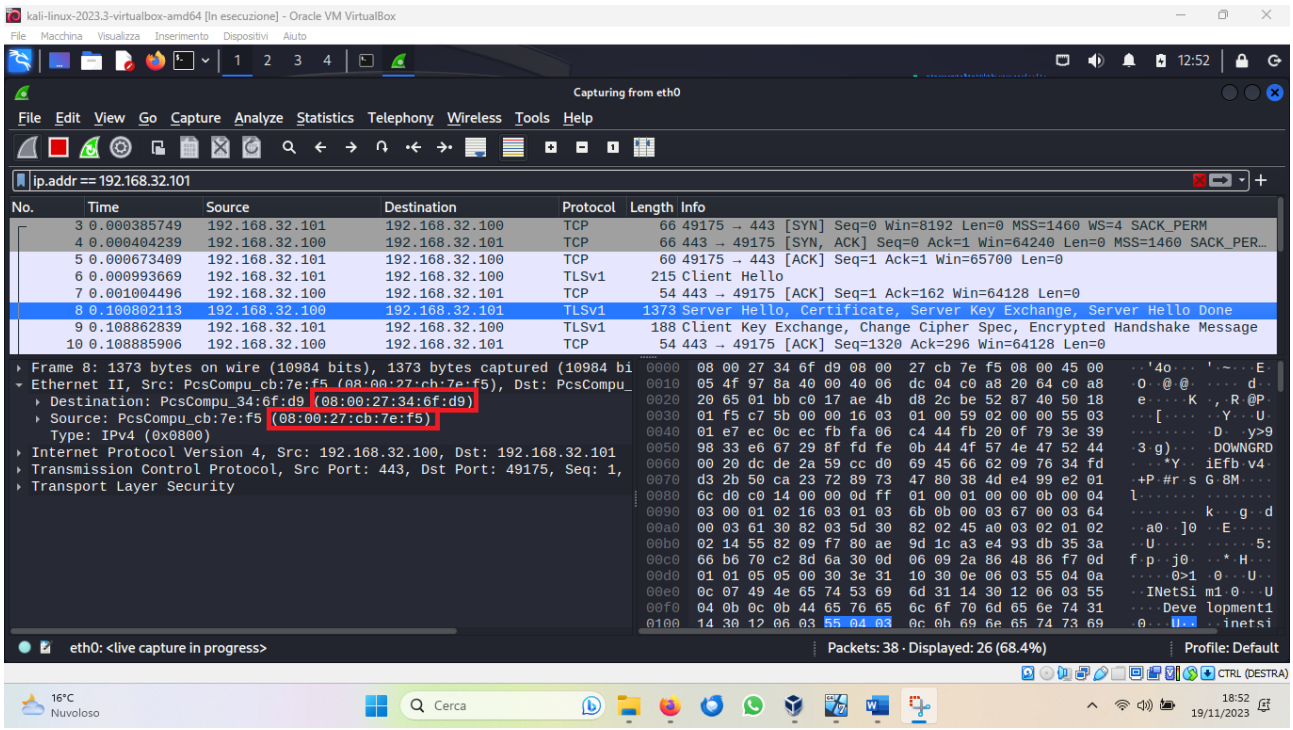


Questa è la pagina che restituisce il browser di windows 7 cercando l'indirizzo https://epicode.internal



Da questa immagine sopra possiamo invece vedere attraverso wireshark come inizialmente avviene la stretta di mano tra il client e il server: [SYN] -> [SYN, ACK] -> [ACK], quindi una volta stabilita una comunicazione stabile il client (Win7) fa una richiesta al server che poi risponde (immagine sotto)

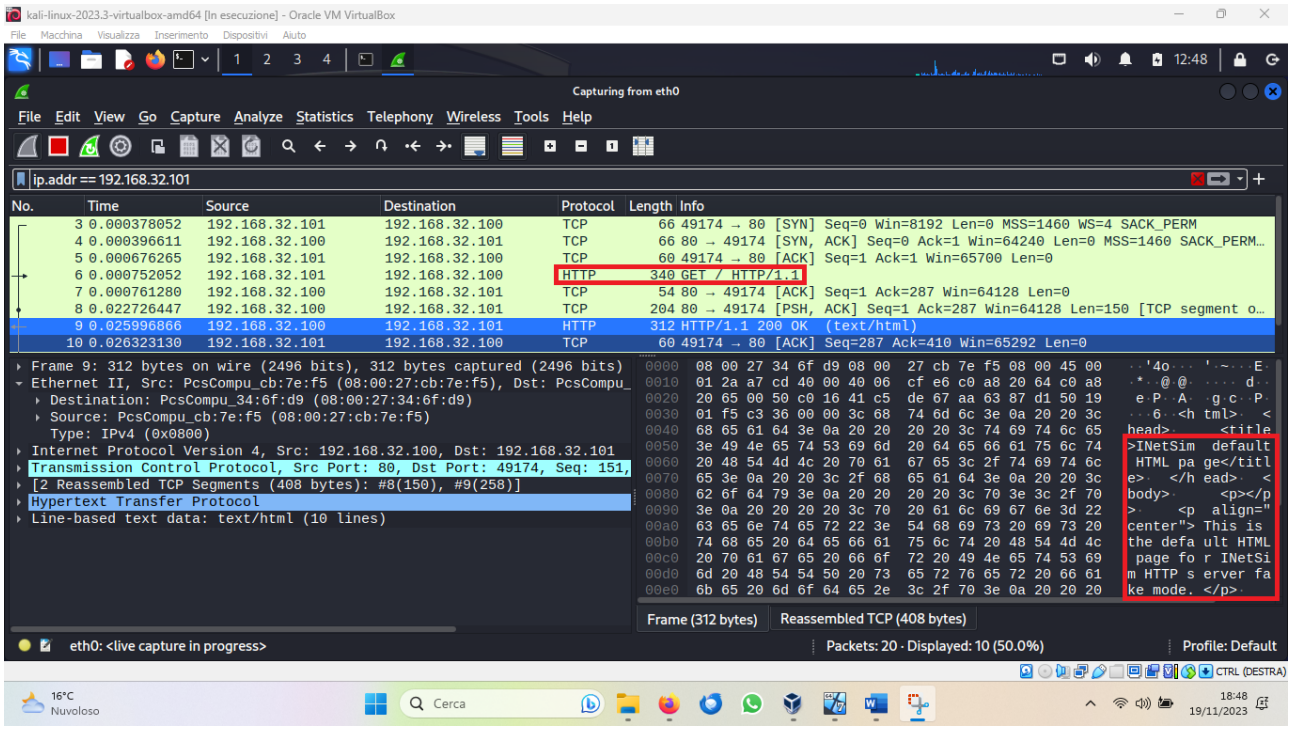
ESERCITAZIONE FINE MODULO 1
BARRECA MONICA



Evidenziati in rosso in entrambe le immagini vi sono gli indirizzi MAC Address della sorgente e del destinatario dai quali è possibile proprio dedurre l'interazione tra i due.

Oltre a questo, in entrambe le immagini si può notare come sia client che server sfruttino il protocollo TLS per cifrare la request e la response e concordare una chiave di lettura per accedere alle informazioni scambiate.

Di seguito abbiamo invece un pacchetto catturato quando dal browser win7 si cerca l'indirizzo http://epicode.internal



Da questa immagine possiamo notare invece come sia la request del client che la response del server siano chiare, visibili e leggibili.

Riassumendo quindi è possibile affermare che vi è una sostanziale differenza tra il servizio HTTPS e HTTP:

- HTTPS: sia la richiesta che la risposta vengono cifrate attraverso protocolli TLS (o SSL) rendendo lo scambio molto più sicuro, dato che esternamente non un ipotetico man in the middle non riuscirebbe a carpire alcuna informazione.
- HTTP: questo servizio non è sicuro in quanto sia la richiesta che la risposta sono “in chiaro”, ed è possibile vedere tutto quello che avviene durante lo scambio, dalle informazioni delle macchine, al tipo di richiesta, fino alla risposta.