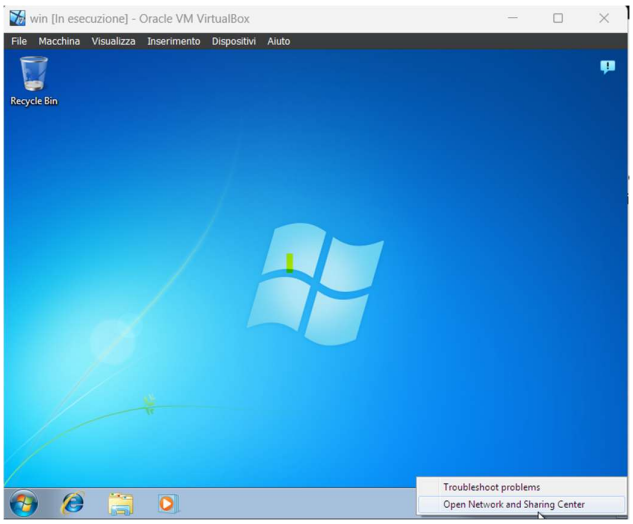
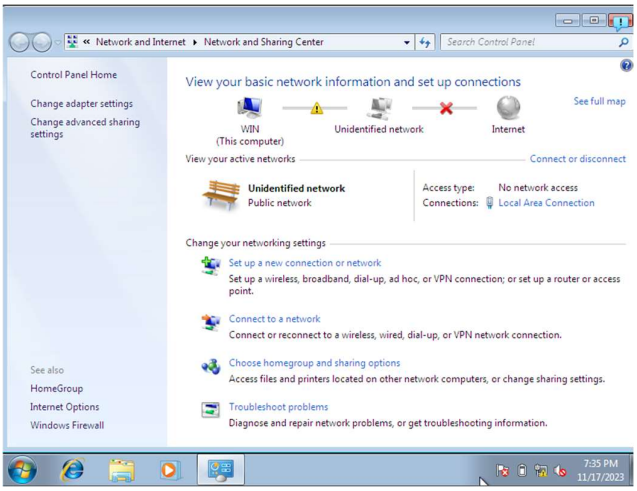
Esercizio W4D4

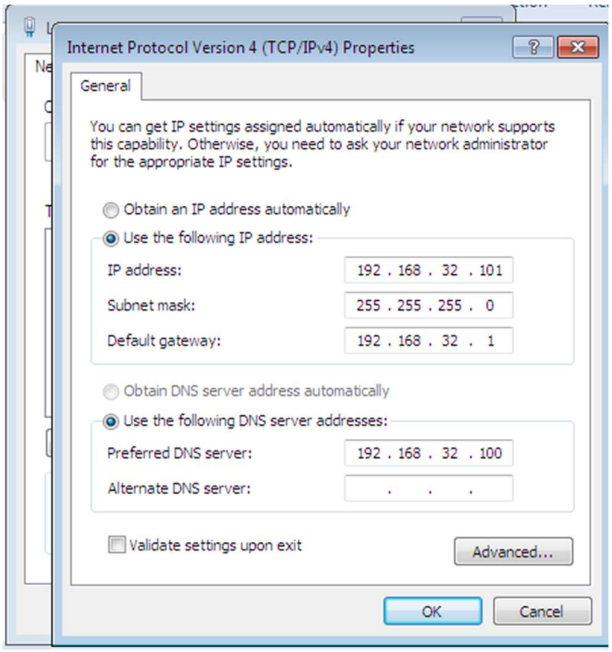
Per lo svolgimento di questo esercizio come primo passaggio provvedo alla configurazione delle schede di rete delle due macchine.

Per Win7 procediamo nel seguente modo:

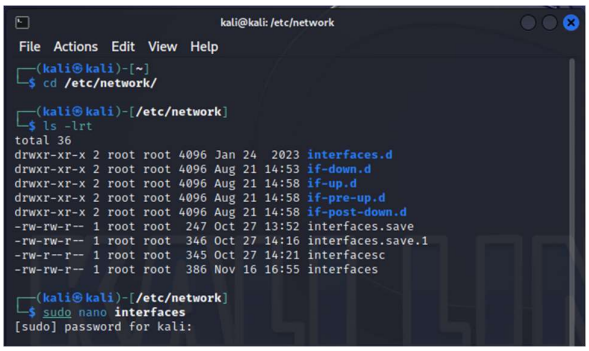


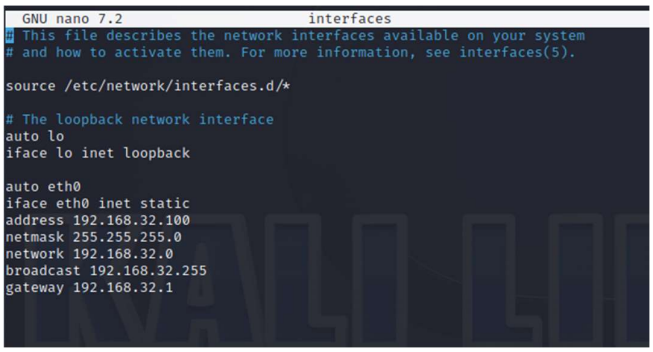


Imposto indirizzo IP, Subnet mask, gateway e il DNS relativo all'indirizzo IP della macchina Kali sul quale appunto oltre ai servizi di https e http avvieremo anche il servizio DNS.

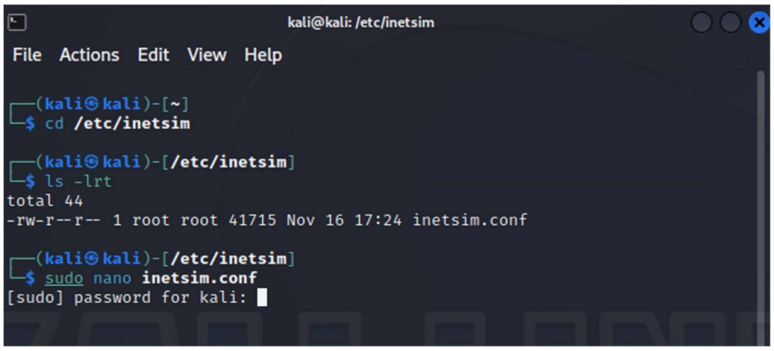


Per quanto riguarda Kali, apriremo da terminale il file interfaces per configurare la scheda di rete:

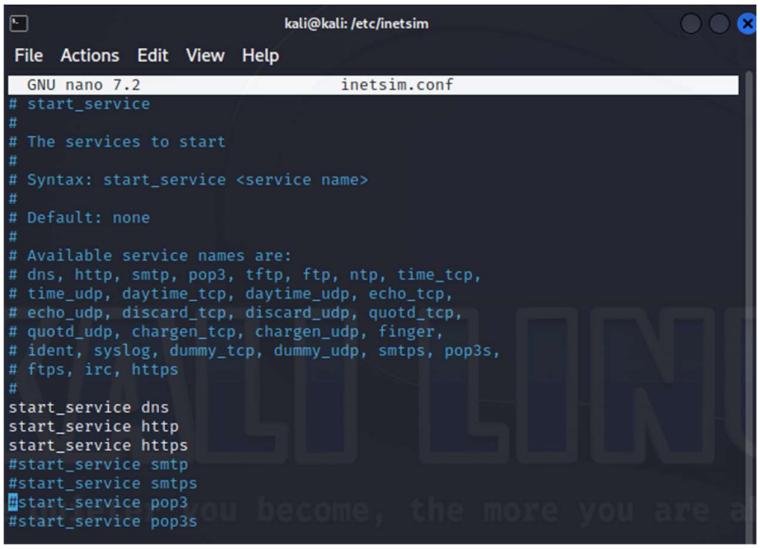




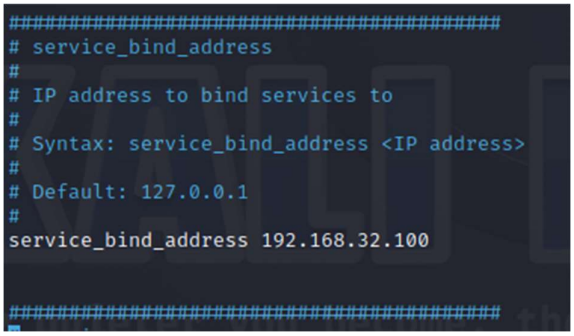
Procediamo poi con la configurazione di inetsim strumento progettato per simulare diversi servizi di rete tra cui appunto quelli richiesti dalla traccia.

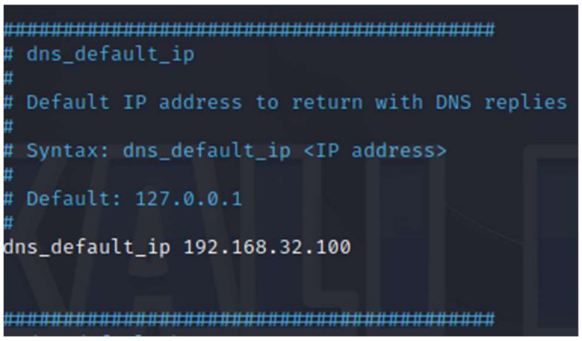


Successivamente procedo a commentare tutti i servizi di rete non richiesti lasciando attivi solo quello di https, http e dns.



Imposto indirizzo di bind e configuro il dns:





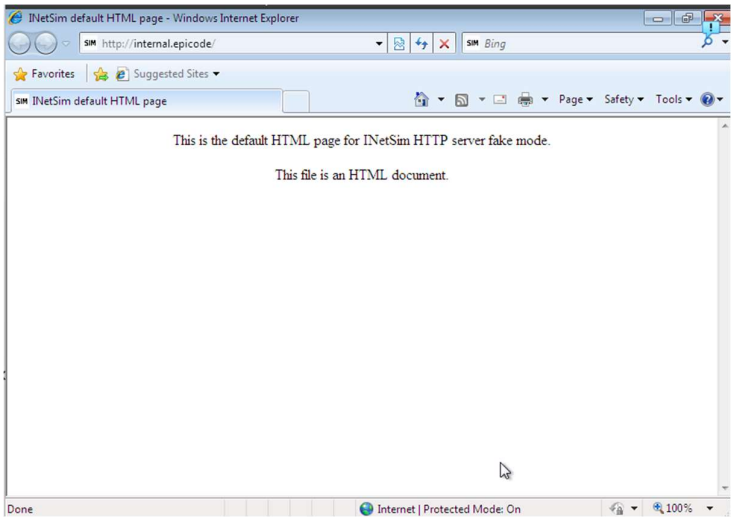
```
##########################################
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static epicode.internal 192.168.32.100


##########################################
```

Avvio inetsim:

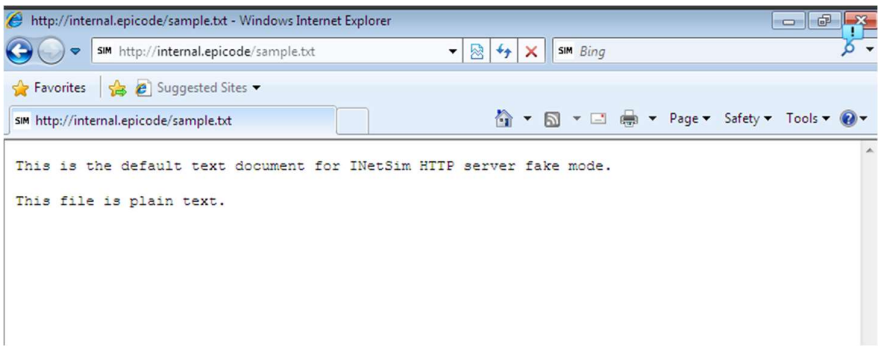
```
┌──(kali㉿kali)-[~]
└─$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
=== INetSim main process started (PID 17154) ===
Session ID:     17154
Listening on:   192.168.32.100
Real Date/Time: 2023-11-17 13:57:51
Fake Date/Time: 2023-11-17 13:57:51 (Delta: 0 seconds)
 Forking services...
  * dns_53_tcp_udp - started (PID 17164)
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
  * http_80_tcp - started (PID 17165)
  * https_443_tcp - started (PID 17166)
 done.
Simulation running.
```

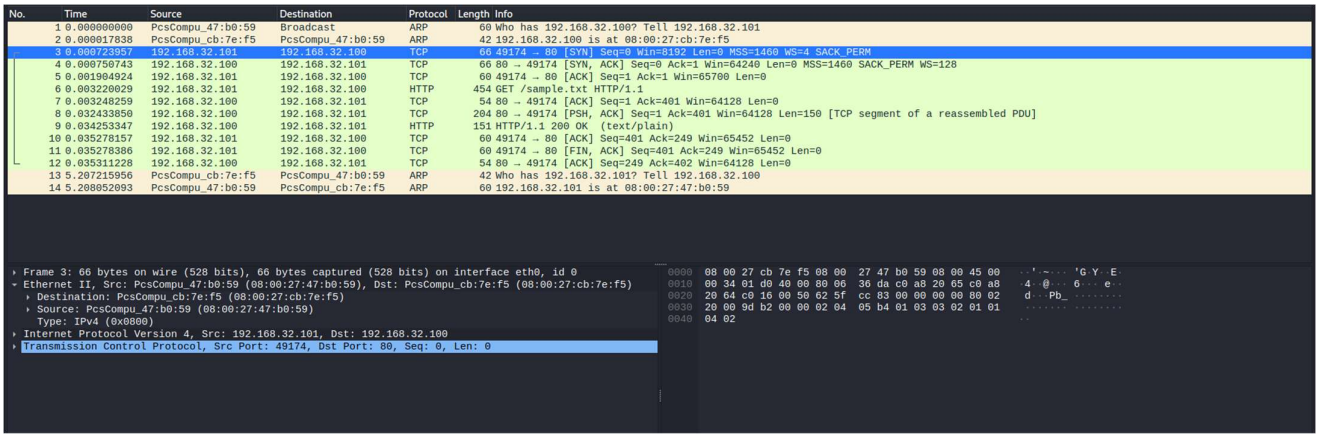
Da Win7 Provo ad preventivamente ad accedere alla risorsa tramite ping e poi con il web browser:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\vboxuser>ping epicode.internal

Pinging epicode.internal [192.168.32.100] with 32 bytes of data:
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.32.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\vboxuser>_
```



Su Win7 procedo a richiedere il file sample.txt tramite http e intercetto la comunicazione con Wireshark su Kali sulla eth0:

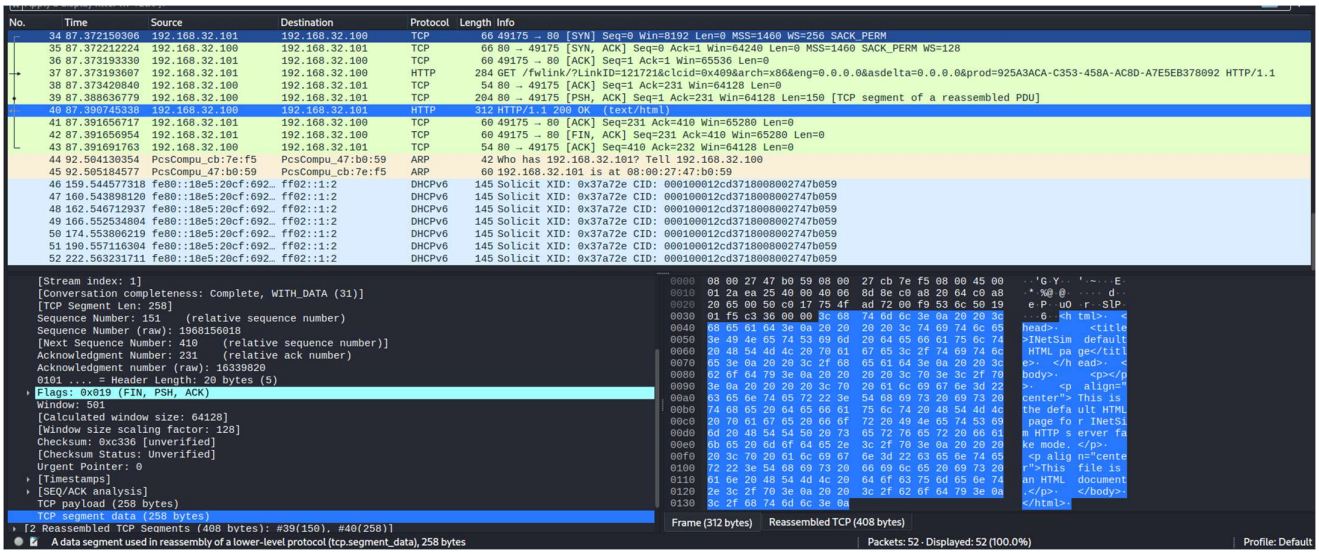




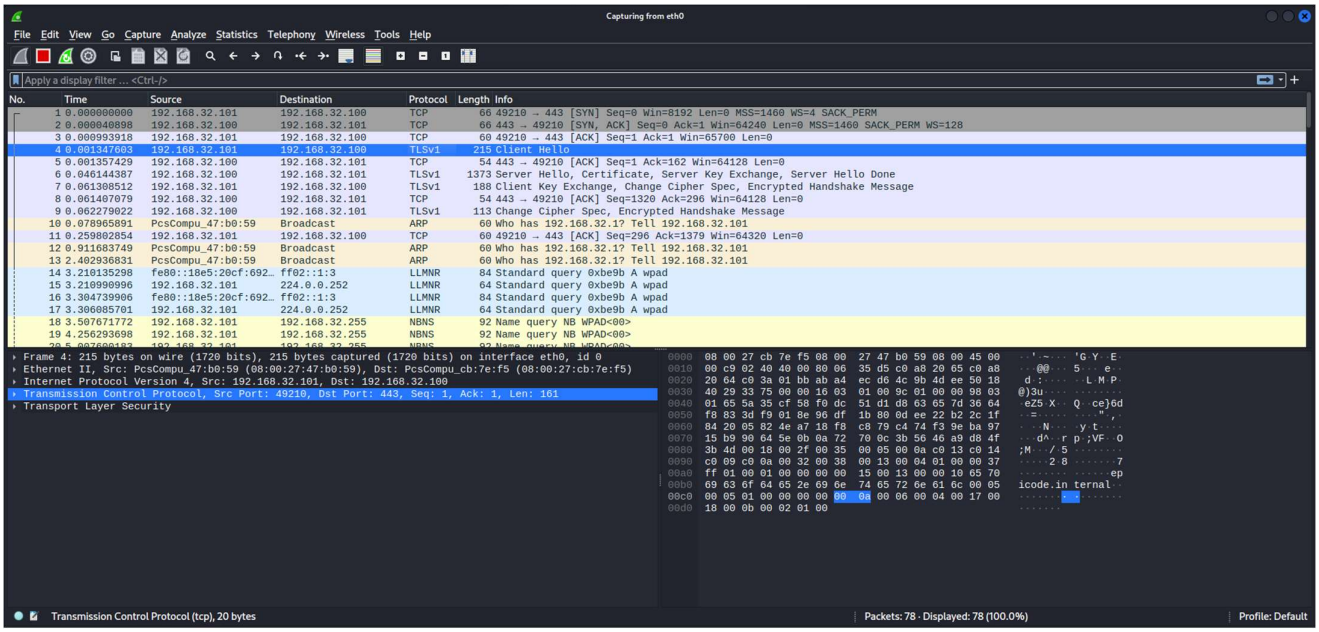
Dalla schermata sopra possiamo vedere il protocollo ARP, il processo di scambio di pacchetti SYN, SYN-ACK e ACK (three way handshake) , fondamentale nel protocollo TCP/IP per stabilire una connessione tra due dispositivi su una rete e l'utilizzo della porta 80 (http). Possiamo successivamente evincere gli indirizzi MAC sorgente e destinatario:

- Source: 08:00:27:47:b0:59
- Destination: 08:00:27:cb:7e:f5

Ed infine essendo una richiesta http possiamo vedere in chiaro il contenuto del file richiesto:



Ora procedo ad analizzare lo scambio di pacchetti per richiedere lo stesso file tramite https:



Dalla schermata possiamo evincere l'utilizzo della porta 443 (HTTPS) e l'handshake SSL/TLS che includono informazioni sui certificati, le chiavi crittografiche e altri parametri di sicurezza. Inoltre non abbiamo modo di poter visualizzare in chiaro come prima il file sample.txt dato che stiamo utilizzando un protocollo che crittografa i dati durante la trasmissione.