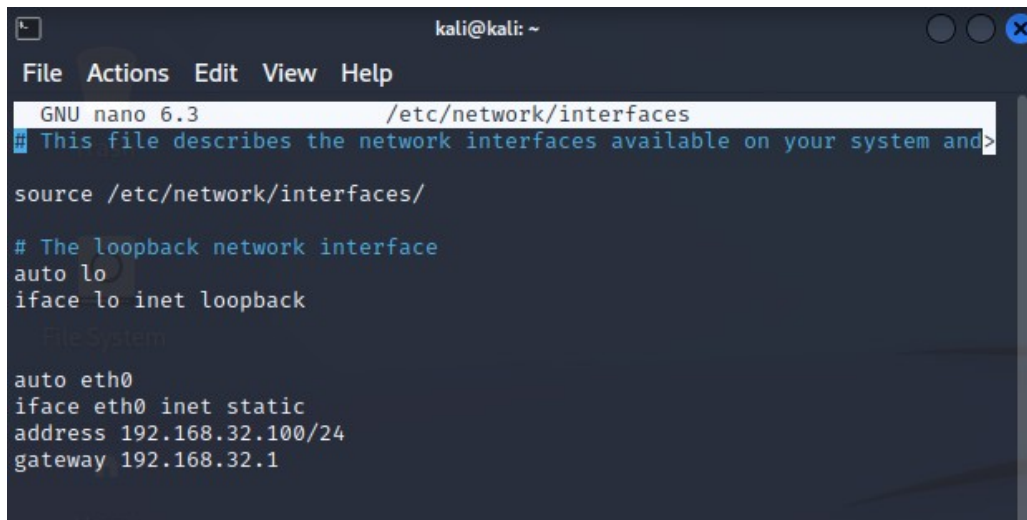


CONFIGURAZIONE DI 2 MACCHINE IN GRADO DI COMUNICARE TRA DI LORO E PROVA DI SNIFFING SULLA LORO COMUNICAZIONE

Per prima cosa sono andato a configurare prima la macchina avente Linux Kali come sistema operativo, dandole l'indirizzo IP indicato nell'esercizio (192.168.32.100):

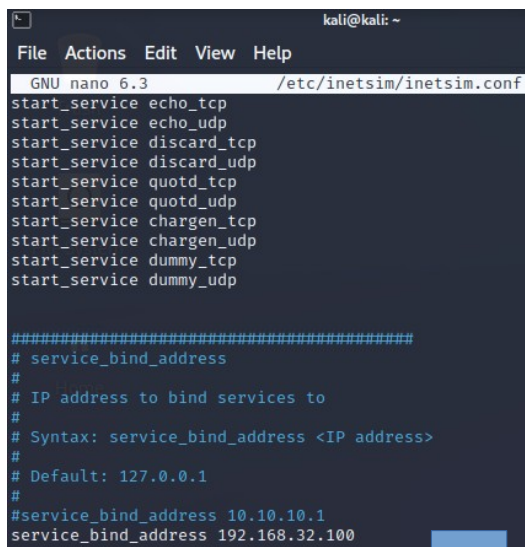


```
kali@kali: ~
File Actions Edit View Help
GNU nano 6.3 /etc/network/interfaces
# This file describes the network interfaces available on your system and
source /etc/network/interfaces/

# The loopback network interface
auto lo
iface lo inet loopback

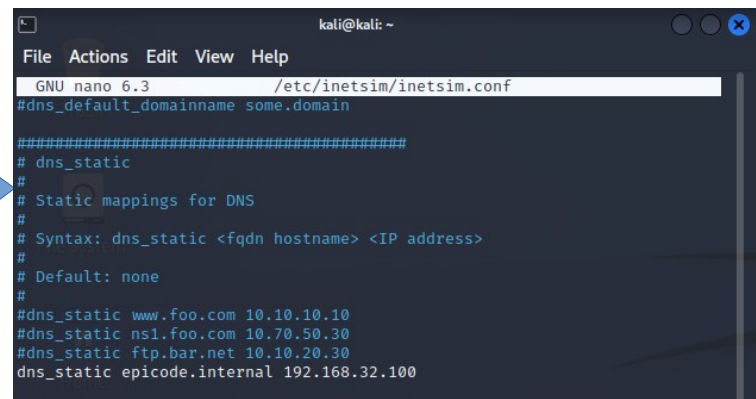
auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1
```

Poi sono passato alla configurazione di inetsim, per avere così un server virtuale HTTPS attivo:



```
kali@kali: ~
File Actions Edit View Help
GNU nano 6.3 /etc/inetsim/inetsim.conf
start_service echo_tcp
start_service echo_udp
start_service discard_tcp
start_service discard_udp
start_service quotd_tcp
start_service quotd_udp
start_service chargen_tcp
start_service chargen_udp
start_service dummy_tcp
start_service dummy_udp

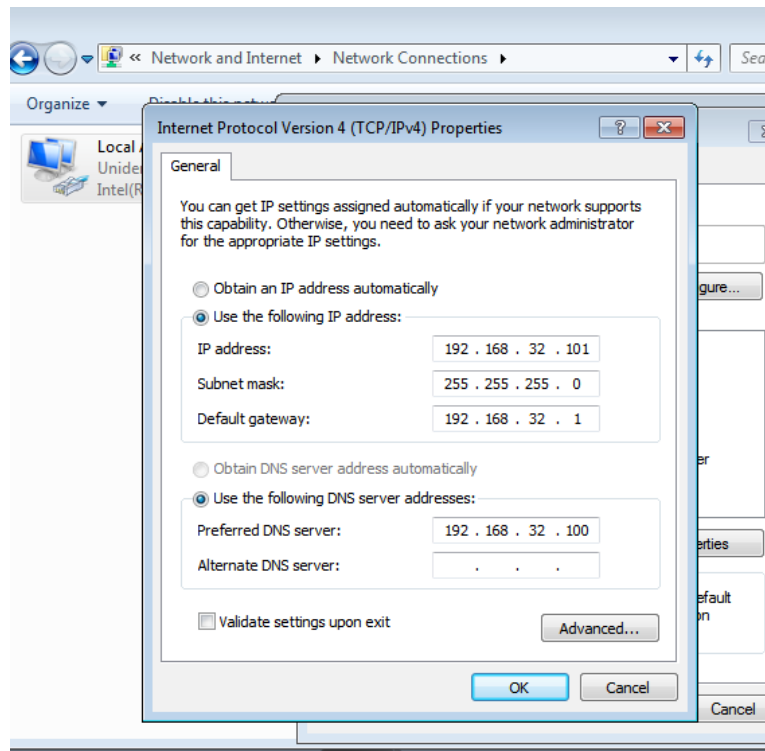
#####
# service_bind_address
#
# IP address to bind services to
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#service_bind_address 10.10.10.1
service_bind_address 192.168.32.100
```



```
kali@kali: ~
File Actions Edit View Help
GNU nano 6.3 /etc/inetsim/inetsim.conf
#dns_default_domainname some.domain

#####
# 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
```

Ho attivato anche i servizi DNS per la risoluzione dei nomi di dominio. Finito momentaneamente il mio lavoro su Kali sono passato alla configurazione della macchina virtuale con sistema operativo Windows 7, impostando prima l'indirizzo IP richiesto nell'esercizio (192.168.32.101):



Per assicurarmi che le macchine fossero correttamente collegate tra loro ho fatto 2 prove.

Nella prima ho lanciato il comando ping inserendo l'indirizzo IP del server

```
C:\Windows\system32\cmd.exe
C:\Users\admin>ping 192.163.32.100

Pinging 192.163.32.100 with 32 bytes of data:
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.
PING: transmit failed. General failure.

Ping statistics for 192.163.32.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\admin>ping 192.168.32.100

Pinging 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 = 0ms, Maximum = 1ms, Average = 0ms
```

```
C:\Windows\system32\cmd.exe
Pinging 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 = 0ms, Maximum = 1ms, Average = 0ms

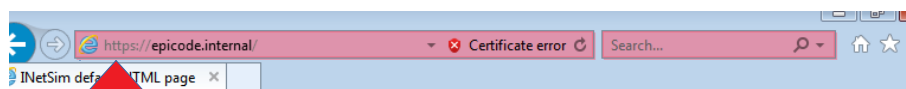
C:\Users\admin>ping epicode.internal

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 = 0ms, Maximum = 0ms, Average = 0ms
```

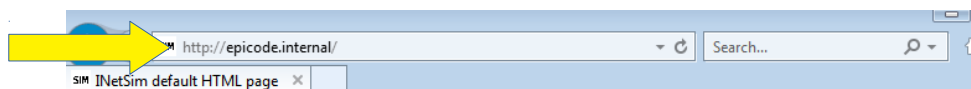
Nella seconda ho lanciato sempre il comando ping ma inserendo l'hostname

Avendo avuto esito positivo in entrambe le prove sono passato alla fase successiva: ho richiesto tramite web browser su Windows 7 una risorsa alla macchina Kali, prima tramite protocollo HTTPS, poi HTTP e i risultati anche questa volta sono stati positivi come dimostrano le immagini sottostanti:



This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.



This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.

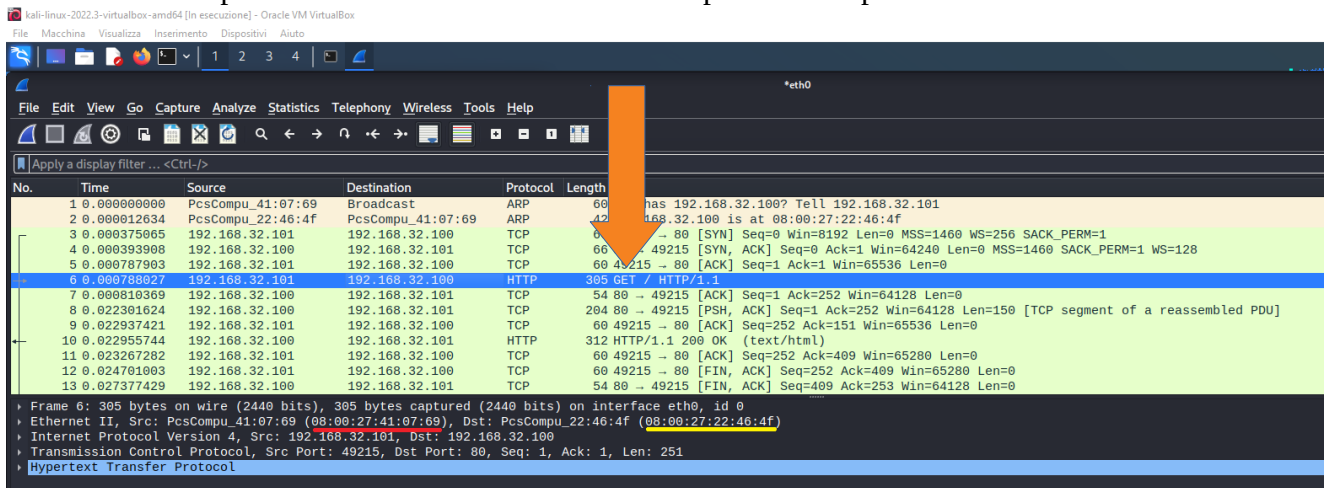
Sono quindi passato quindi alla seconda parte dell'esercizio: usare wireshark per fare sniffing sulla comunicazione tra le 2 macchine cercando gli indirizzi MAC.

A screenshot of the Wireshark network protocol analyzer. The main pane shows a list of captured packets. The first three packets are highlighted in blue, indicating a three-way handshake (SYN, SYN-ACK, ACK). The source and destination IP addresses are 192.168.32.101 and 192.168.32.100. The source and destination MAC addresses are 08:00:27:41:07:69 and 08:00:27:22:46:4f. The details pane on the left shows the 'Network Connection Details' for the selected packet, with the physical address 08:00:27:41:07:69 highlighted in red. The details pane on the right shows the 'Ethernet II' details, with the destination MAC address 08:00:27:22:46:4f highlighted in yellow. The packet list shows the following details for the first three packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49277 → 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.000000000	192.168.32.100	192.168.32.101	TCP	66	443 → 49277 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000000000	192.168.32.101	192.168.32.100	TCP	60	49277 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0

Possiamo vedere nelle 3 immagini qua sopra in rosso la corrispondenza tra il MAC sorgente e il MAC di Windows 7 e in giallo invece quella tra il MAC destinatario ed il MAC di Kali. Mentre in arancione possiamo vedere il three-way handshake tra le due macchine. Possiamo notare inoltre che wireshark non è capace di vedere qual'è la request ed il suo contenuto in quanto il protocollo HTTPS è criptato.

Situazione completamente diversa avviene invece se passiamo al protocollo HTTP:



dall'immagine qui sopra possiamo vedere non solo come wireshark è stato capace di vedere qual'è il contenuto della richiesta (in questo caso GET / HTTP/ 1.1) ma scendendo ed andando ad analizzare la riga 10, abbiamo anche la possibilità di andare a leggere il messaggio contenuto, come dimostra l'immagine qui sotto:

