

Traccia e Dati:

In questo esercizio metteremo insieme le competenze acquisite finora.

Requisiti e servizi:

- Kali Linux: IP 192.168.32.100
- Windows: IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

Traccia:

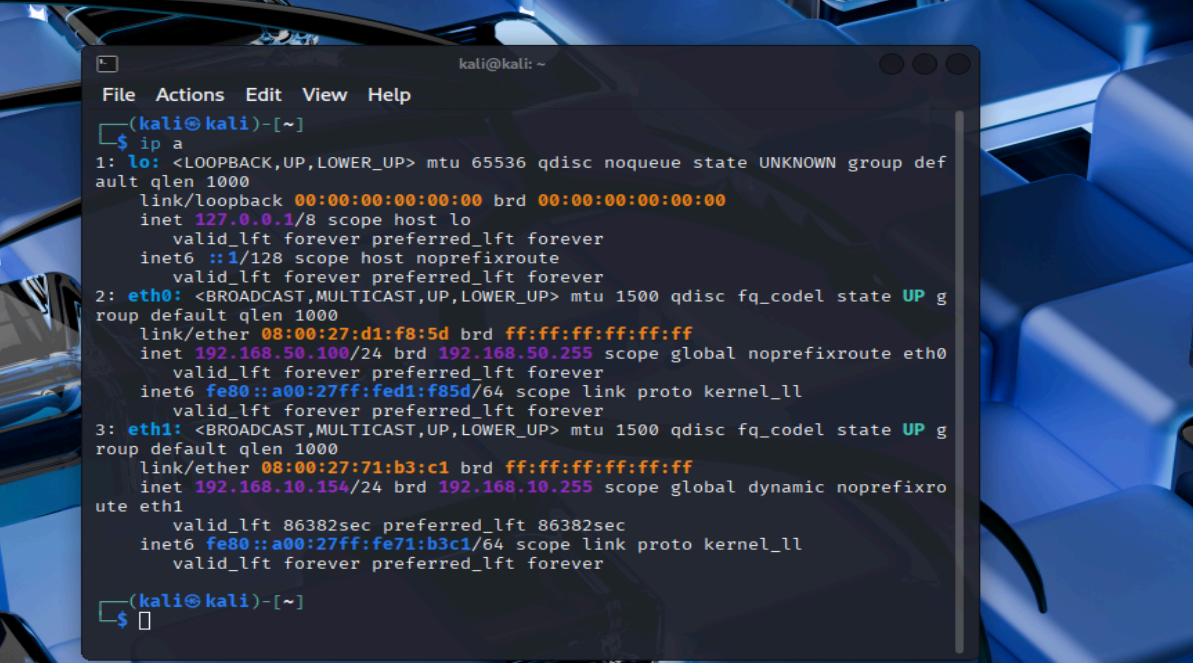
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows) richiede tramite web browser una risorsa all'hostname **epicode.internal** che risponde all'indirizzo 192.168.32.100 (Kali).

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:

CONFIGURAZIONE MACCHINE: PRIMA PARTE



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def  
ault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g  
roup default qlen 1000  
    link/ether 08:00:27:d1:f8:5d brd ff:ff:ff:ff:ff:ff  
    inet 192.168.50.100/24 brd 192.168.50.255 scope global noprefixroute eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fed1:f85d/64 scope link proto kernel_ll  
        valid_lft forever preferred_lft forever  
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g  
roup default qlen 1000  
    link/ether 08:00:27:71:b3:c1 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.10.154/24 brd 192.168.10.255 scope global dynamic noprefixro  
ute eth1  
        valid_lft 86382sec preferred_lft 86382sec  
    inet6 fe80::a00:27ff:fe71:b3c1/64 scope link proto kernel_ll  
        valid_lft forever preferred_lft forever  
(kali@kali)-[~]  
$
```

```
C:\Users\user>ipconfig

Configurazione IP di Windows

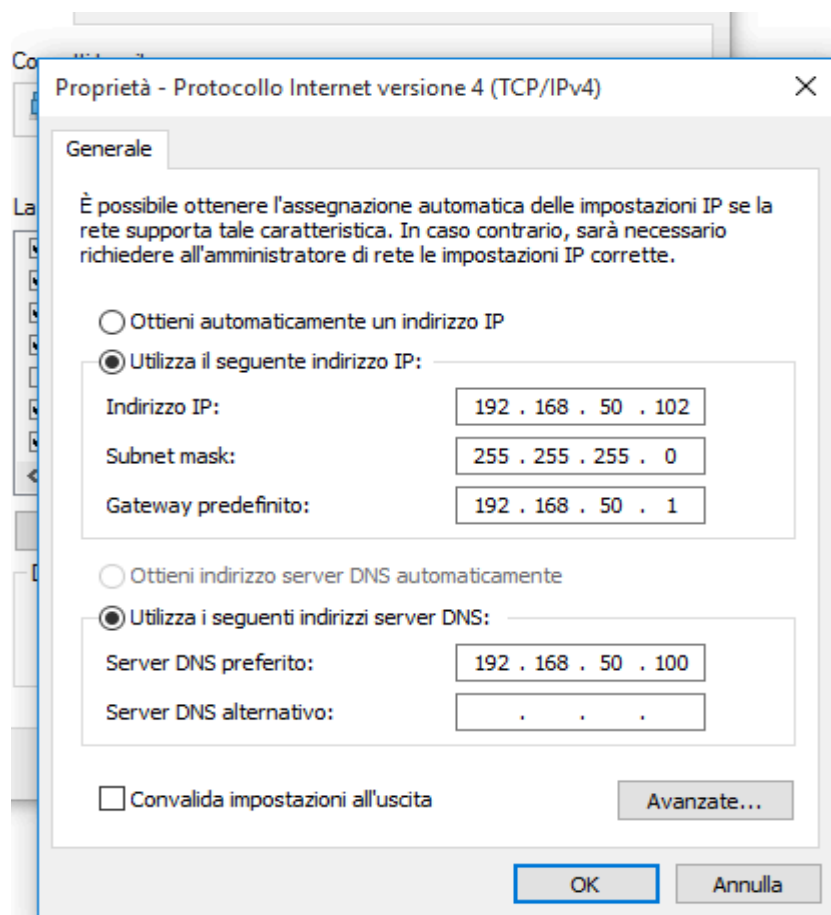
Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv4. . . . . : 192.168.50.102
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.50.1

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:
```

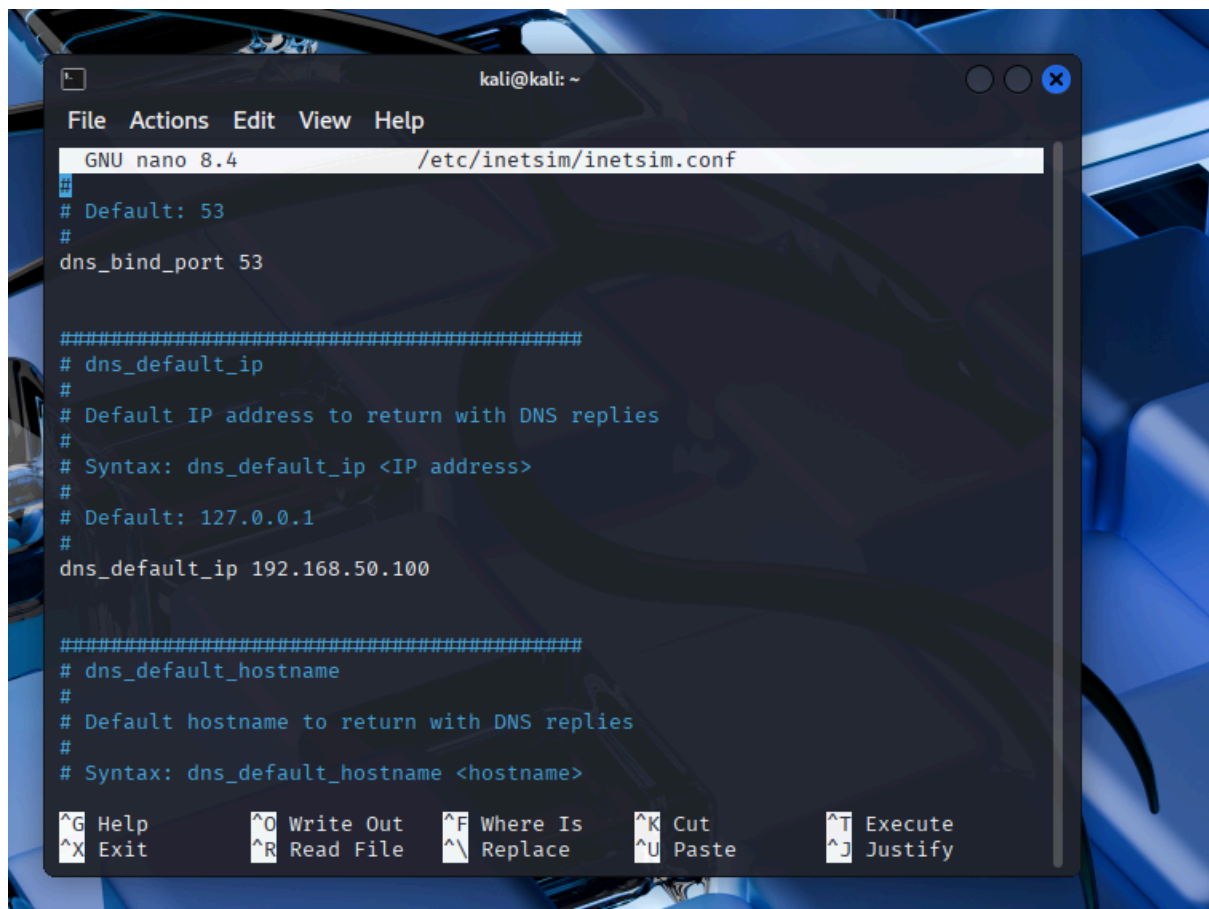
```
C:\Users\user>
```



nento 1 elemento selezionato

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf  
# 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  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf  
#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 192.168.50.100  
  
#####  
# service_run_as_user  
#  
# User to run services  
#  
# Syntax: service_run_as_user <username>  
#  
# Default: inetsim  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

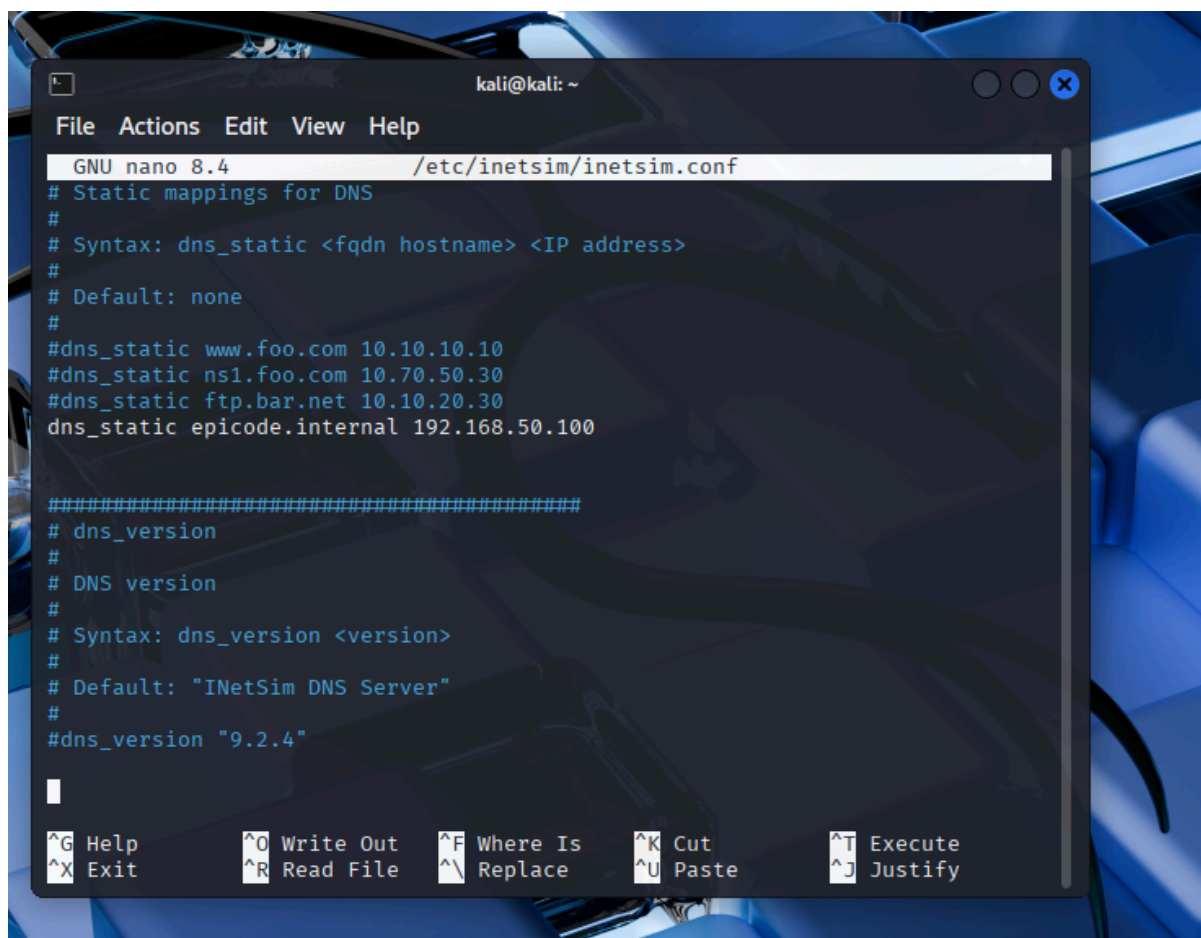


```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.4 /etc/inetsim/inetsim.conf
#
# Default: 53
#
dns_bind_port 53

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

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

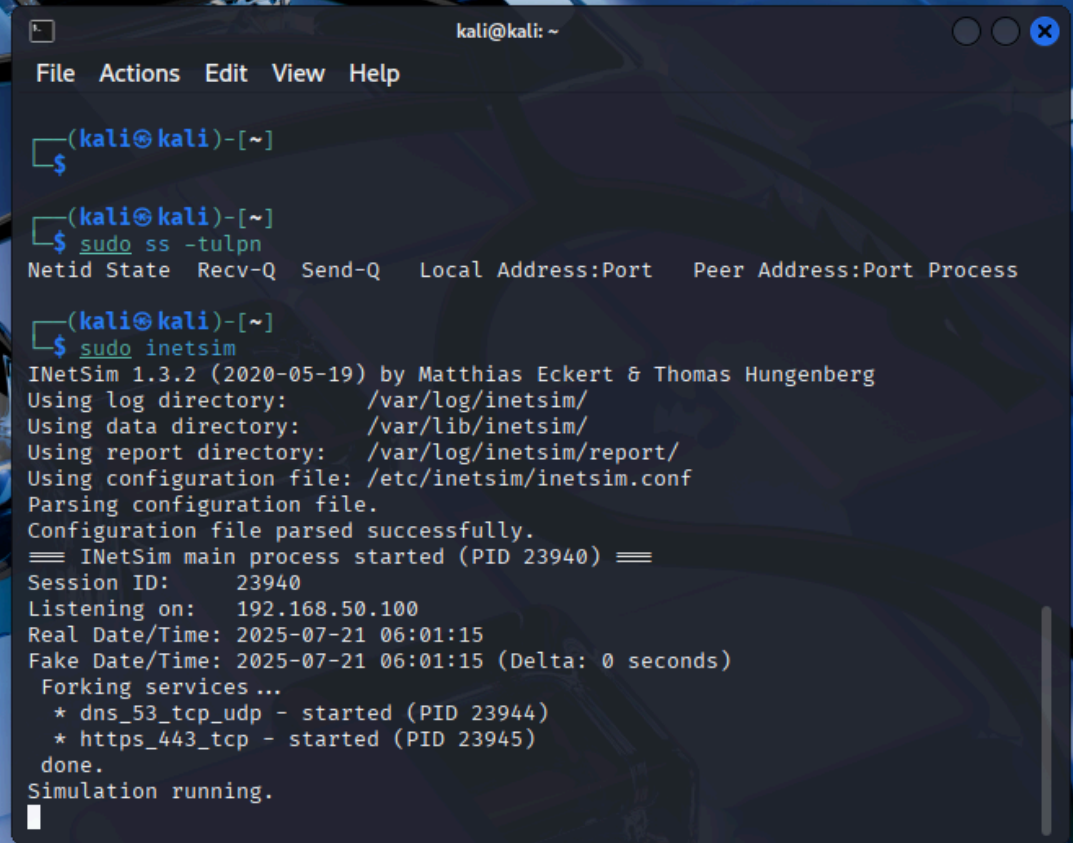
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf  
# 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 192.168.50.100  
  
#####  
# dns_version  
#  
# DNS version  
#  
# Syntax: dns_version <version>  
#  
# Default: "INetSim DNS Server"  
#  
#dns_version "9.2.4"  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

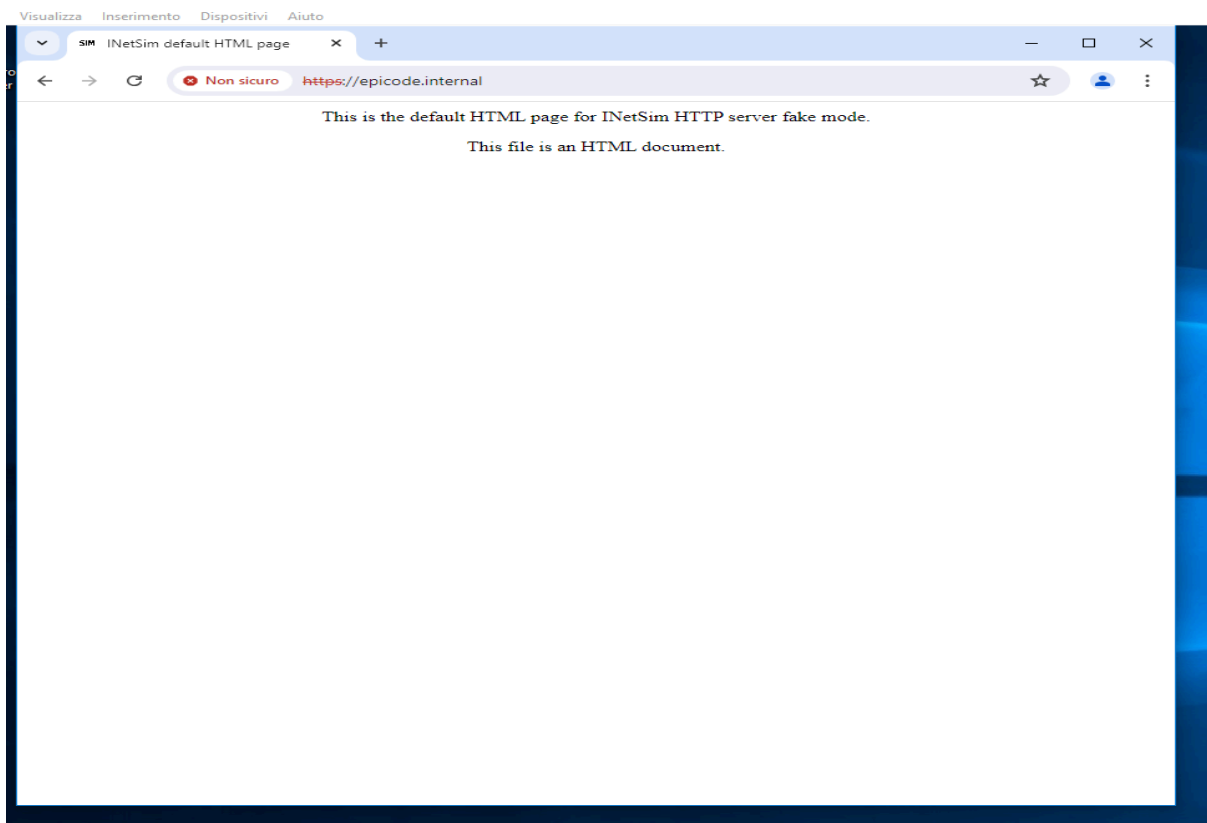
per prima cosa ho controllato la configurazione degli ip delle VM, come da esercizi e indicazioni precedenti ho configurato la macchina kali con ip 192.168.50.100 e la macchina windows 10 con ip 192.168.50.102 e poi ho controllato tramite le impostazioni di rete su windows 10 che il server DNS preferito era la macchina kali, dopo sono andata su kali e ho aperto la configurazione di inetsim con il comando `sudo nano /etc/inetsim/inetsim.conf` ho attivato i servizi HTTPS e DNS eliminando l'# davanti, poi ho configurato il service bind address mettendo l'ip della macchina kali la porta 53 il DNS default e infine il DNS static creando il nome epicode.internal.

ATTIVAZIONE DEL SERVIZIO E SVOLGIMENTO ESERCIZIO: SECONDA PARTE

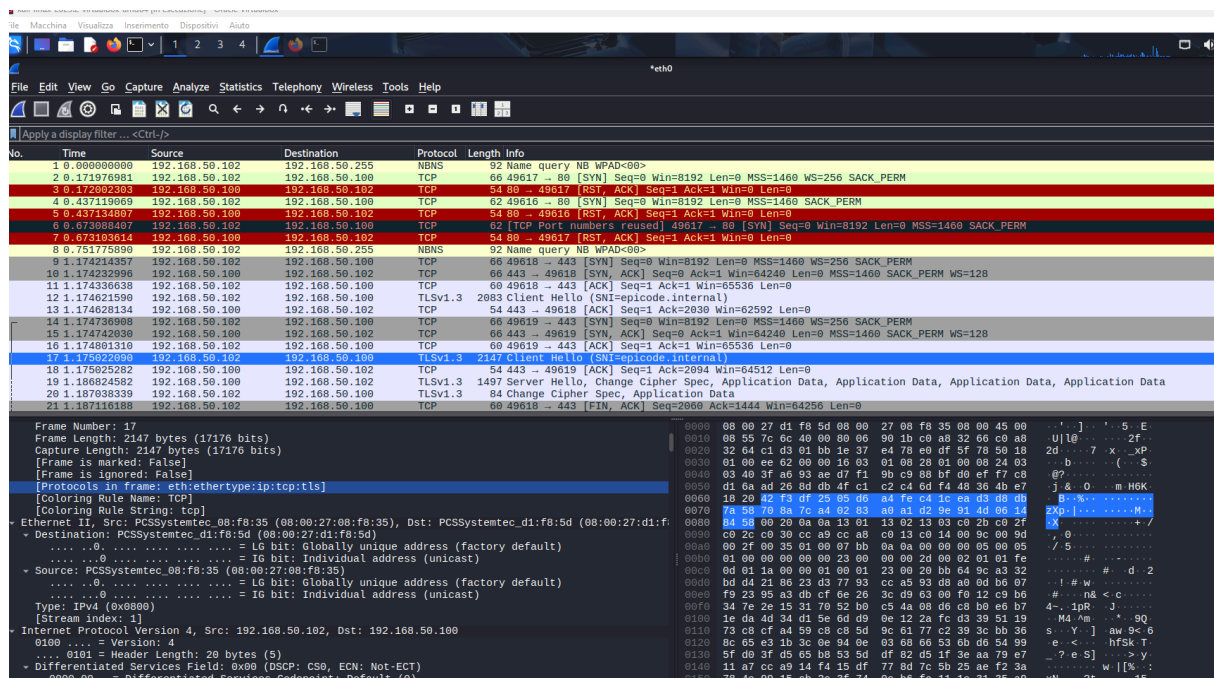


```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$  
  
(kali@kali)-[~]  
$ sudo ss -tulpn  
Netid State Recv-Q Send-Q Local Address:Port Peer Address:Port Process  
  
(kali@kali)-[~]  
$ sudo inetsim  
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg  
Using log 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 23940) ===  
Session ID: 23940  
Listening on: 192.168.50.100  
Real Date/Time: 2025-07-21 06:01:15  
Fake Date/Time: 2025-07-21 06:01:15 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 23944)  
* https_443_tcp - started (PID 23945)  
done.  
Simulation running.  
█
```

dopo la configurazione ho attivato il servizio di inetsim



dalla macchina windows ho cercato epicode.internal e il servizio mi ha risposto.



```
[Coloring Rule Name: tcp]
[Coloring Rule String: tcp]
Ethernet II, Src: PCSSystemtec_08:f8:35 (08:00:27:08:f8:35), Dst: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d)
  Destination: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8:5d)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Source: PCSSystemtec_08:f8:35 (08:00:27:08:f8:35)
    ....0. .... = LG bit: Globally unique address (factory default)
    ....0. .... = IG bit: Individual address (unicast)
  Type: IPv4 (0x0800)
  [Stream index: 1]
Internet Protocol Version 4, Src: 192.168.50.102, Dst: 192.168.50.100
```

ho fatto partire la cattura con wireshark come segnalato nelle ultimi 2 screenshot ho cliccato sul pacchetto client Hello (SNI- epicode.internal) ho espanso il pacchetto per visualizzare i dettagli come da esercizio per individuare il Mac sorgente e il Mac destinatario come si può vedere dall'immagine l'indirizzo Mac destinatario è (08:00:27:d1:f8:5d) e l'indirizzo Mac sorgente è (08:00:27:08:f8:35)

```
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>ipconfig /all

Configurazione IP di Windows

Nome host . . . . . : DESKTOP-9K104BT
Suffisso DNS primario . . . . . :
Tipo nodo . . . . . : Ibrido
Routing IP abilitato. . . . . : No
Proxy WINS abilitato . . . . . : No

Scheda Ethernet Ethernet:

Suffisso DNS specifico per connessione:
Descrizione . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Indirizzo fisico. . . . . : 08-00-27-08-F8-35
DHCP abilitato. . . . . : No
Configurazione automatica abilitata : Si
Indirizzo IPv4. . . . . : 192.168.50.102(Preferenziale)
Subnet mask . . . . . : 255.255.255.0
Gateway predefinito . . . . . : 192.168.50.1
Server DNS . . . . . : 192.168.50.100
NetBIOS su TCP/IP . . . . . : Attivato

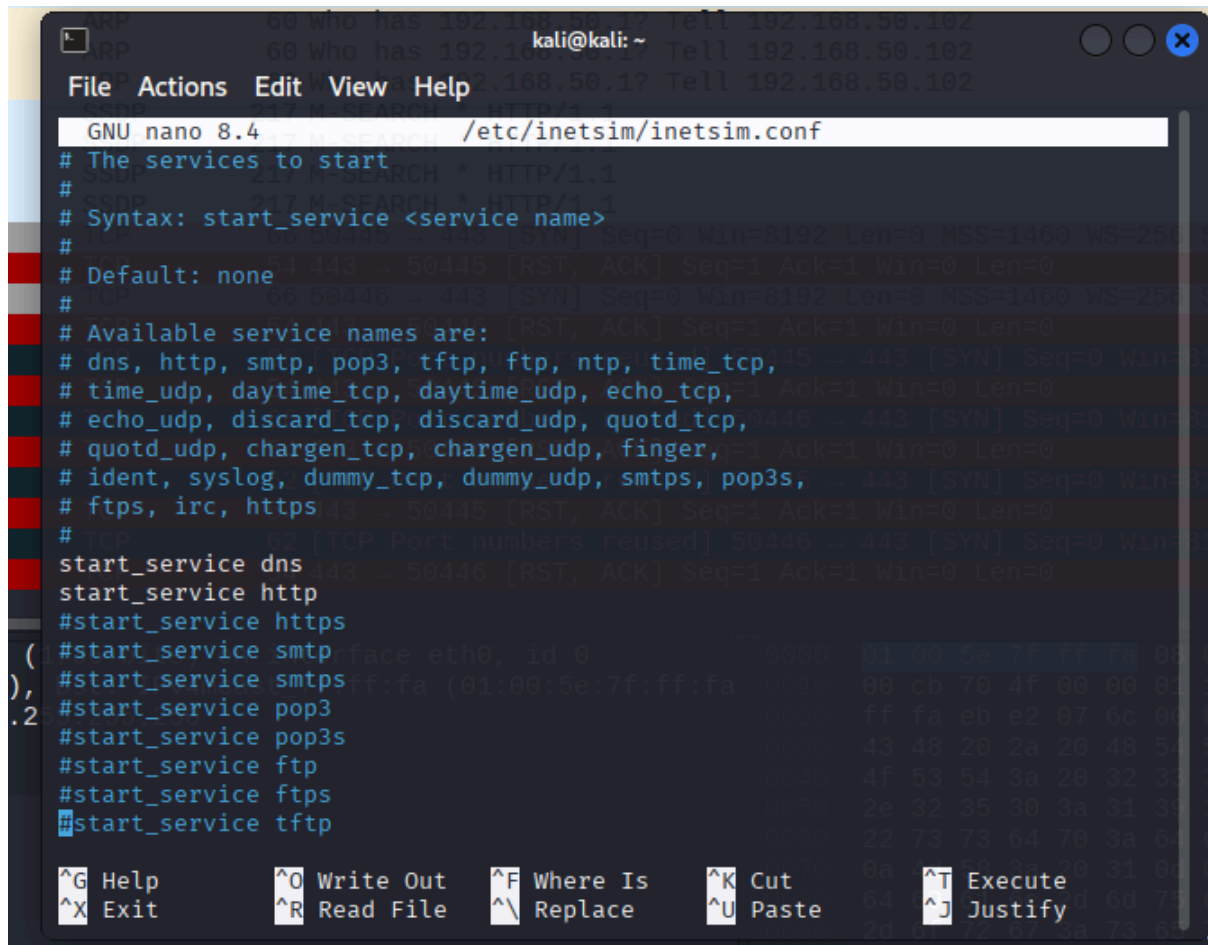
Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

Stato supporto. . . . . : Supporto disconnesso
Suffisso DNS specifico per connessione:
```

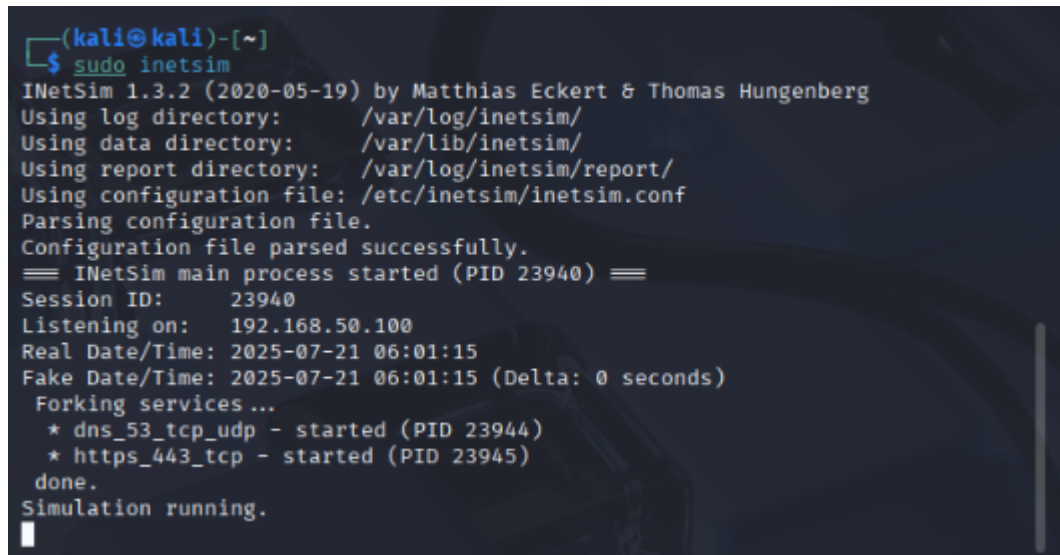
```
kali@kali: ~
File Actions Edit View Help
inet6 fe80::a00:27ff:fed1:f85d prefixlen 64 scopeid 0x20<link>
ether 08:00:27:d1:f8:5d txqueuelen 1000 (Ethernet)
RX packets 7095 bytes 1210480 (1.1 MiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 3220 bytes 842301 (822.5 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

ho controllato se gli indirizzi mac corrispondevano da windows con il prompt dei comandi e corrispondeva e da kali sempre dal prompt dei comandi con il comando ifconfig e corrispondevano.

CONFIGURAZIONE DEL SERVIZIO HTTP E SVOLGIMENTO: TERZA PARTE



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf  
# 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  
.2 #start_service pop3  
#start_service pop3s  
#start_service ftp  
#start_service ftps  
#start_service tftp  
^G Help ^O Write Out ^F Where Is ^K Cut  
^X Exit ^R Read File ^N Replace ^U Paste ^T Execute  
^J Justify
```



```
(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 23940) ==  
Session ID: 23940  
Listening on: 192.168.50.100  
Real Date/Time: 2025-07-21 06:01:15  
Fake Date/Time: 2025-07-21 06:01:15 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 23944)  
* https_443_tcp - started (PID 23945)  
done.  
Simulation running.
```

per la terza ed ultima parte dell'esercizio sono andata di nuovo sulla configurazione di inetsim e ho attivato il HTTP dopo ho fatto partire inetsim.

Apply a display filter... <Ctrl-F>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.50.102	192.168.50.100	TCP	66	50774 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
2	0.000019370	192.168.50.100	192.168.50.102	TCP	66	80 → 50774 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000064220	192.168.50.102	192.168.50.100	TCP	66	50775 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
4	0.000067280	192.168.50.100	192.168.50.102	TCP	66	80 → 50775 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.000105270	192.168.50.102	192.168.50.100	TCP	60	50774 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
6	0.000221600	192.168.50.102	192.168.50.100	TCP	60	50775 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
7	0.000400150	192.168.50.102	192.168.50.100	HTTP	532	GET / HTTP/1.1
8	0.000407180	192.168.50.100	192.168.50.102	TCP	54	80 → 50774 [ACK] Seq=1 Ack=479 Win=64128 Len=0
9	0.000575015	192.168.50.102	192.168.50.255	NBNS	92	Name query NB WPAD<00>
10	0.000678825	192.168.50.102	224.0.0.252	LLMNR	64	Standard query 0x1df5 A wpad
11	0.011289330	192.168.50.100	192.168.50.102	TCP	204	80 → 50774 [PSH, ACK] Seq=1 Ack=479 Win=64128 Len=150 [TCP PDU reassembled in 12]
12	0.012607819	192.168.50.100	192.168.50.102	HTTP	312	HTTP/1.1 200 OK (text/html)
13	0.012723389	192.168.50.102	192.168.50.100	TCP	60	50774 → 80 [ACK] Seq=479 Ack=410 Win=65280 Len=0
14	0.012795779	192.168.50.102	192.168.50.100	TCP	60	50774 → 80 [FIN, ACK] Seq=479 Ack=410 Win=65280 Len=0
15	0.012802169	192.168.50.100	192.168.50.102	TCP	54	80 → 50774 [ACK] Seq=410 Ack=400 Win=64128 Len=0
16	0.018082029	192.168.50.102	192.168.50.100	HTTP	451	GET /favicon.ico HTTP/1.1
17	0.018095079	192.168.50.102	192.168.50.102	TCP	54	80 → 50775 [ACK] Seq=1 Ack=398 Win=64128 Len=0
18	0.025733814	192.168.50.100	192.168.50.102	TCP	207	80 → 50775 [PSH, ACK] Seq=1 Ack=398 Win=64128 Len=153 [TCP PDU reassembled in 19]
19	0.026953374	192.168.50.100	192.168.50.102	HTTP	252	HTTP/1.1 200 OK (image/x-icon)
20	0.027039729	192.168.50.102	192.168.50.100	TCP	60	50775 → 80 [ACK] Seq=398 Ack=353 Win=65280 Len=0
21	0.027123908	192.168.50.102	192.168.50.100	TCP	60	50775 → 80 [FIN, ACK] Seq=398 Ack=353 Win=65280 Len=0

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0															
Ethernet II, Src: PCSSystemtec_08:f0:35 (08:00:27:08:f0:35), Dst: PCSSystemtec_d1:f8:5d (08:00:27:d1:f8)															
Internet Protocol Version 4, Src: 192.168.50.102, Dst: 192.168.50.100															
Transmission Control Protocol, Src Port: 50774, Dst Port: 80, Seq: 0, Len: 0															
<pre> 0000 08 00 27 d1 f8 5d 08 00 27 08 f0 35 08 00 45 00 ...]...5 E 0010 08 34 11 6a 40 00 00 06 03 3f c0 a8 32 66 c0 a8 4 j0...? 2f.. 0020 32 64 c6 56 00 50 a1 68 3d 48 00 00 00 00 02 2d V P h =H 0030 20 00 c3 9d 00 00 02 04 05 b4 01 03 03 08 01 01 0040 04 02 </pre>															

SPIEGAZIONE:

nel primo caso wireshark cattura pacchetti HTTPS che vengono identificati come TLS nei pacchetti HTTPS i dati sono crittografati nel secondo caso avendo attivato dalla configurazione di inetsim non più l'HTTPS ma l'HTTP i pacchetti visibili in cattura sono quelli HTTP questo perchè le configurazioni di inetsim sono state cambiate.

Non ho aggiunto altro perchè l'unica differenza che ho notato è stata questa perchè gli altri pacchetti come ad esempio i TCP risultavano in entrambe le configurazioni.