

- 1) **Impostiamo indirizzi IP delle MV di Kali** (IP 192.168.32.100) e **W7** (IP 192.168.32.101)

Come modificare IP e gateway Windows7: Pannello di controllo/Rete & Internet/Centro connessioni di rete e condivisione/modifica impostazioni di scheda/proprietà di stato connessione alla rete locale LAN/proprietà-protocollo internet versione 1 (TCP/IPv4)

☒ Utilizza il seguente indirizzo IP:

Indirizzo IP:

Gateway predefinito:

☐ Utilizza i seguenti indirizzi server DNS:

Server DNS preferito:

Come modificare IP e gateway su kali: Terminale/ sudo nano /etc/network/interfaces/ modificare ip/salvare con CTRL+o, invio e CTRL+z/ RIAVVIARE

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

VERIFICA PING:

```
— 192.168.32.101 ping statistics —
70 packets transmitted, 70 received, 0% packet loss, time 70580ms
rtt min/avg/max/mdev = 0.019/0.083/0.340/0.051 ms
```

- 2) **Configurare il simulatore di rete Inetsim** con il comando nano /etc/inetsim/inetsim.conf impostando protocollo DNS on/https on/service_bind_address(IP di kali)/DNS default IP (IP kali)/dns_static con dominio

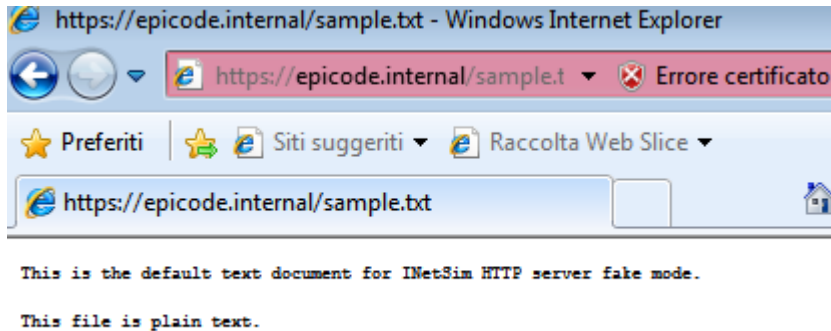
```
start_service dns
#start_service http
start_service https
```

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

```
#####
# 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 192.168.32.100
```

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

3) Prova di risoluzione IP



4) Comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

PC1

```
▶ Frame 7: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0, id 0
▶ Ethernet II, Src: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: PcsCompu_10:6d:b4 (08:00:27:10:6d:b4)
▶ Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
▶ Transmission Control Protocol, Src Port: 443, Dst Port: 50101, Seq: 1, Ack: 157, Len: 0
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000029907	PcsCompu_cb:7e:f5	PcsCompu_10:6d:b4	ARP	42	192.168.32.100 is at 08:00:27:cb:7e:f5
3	0.000295100	192.168.32.101	192.168.32.100	TCP	60	50101 → 443 [SYN] Seq=0 Win=0 Len=0 MSS=1460 SACK_PERM WS=1
4	0.000330425	192.168.32.100	192.168.32.101	TCP	60	443 → 50101 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.000616277	192.168.32.101	192.168.32.100	TCP	60	50101 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.001001000	192.168.32.100	192.168.32.101	TLSv1	210	Client Hello
7	0.001110009	192.168.32.100	192.168.32.101	TCP	54	443 → 50101 [ACK] Seq=1 Ack=157 Win=64128 Len=0
8	0.041004147	192.168.32.100	192.168.32.101	TLSv1	1368	Server Hello, Certificate, Server Key Exchange, Server Hello Done
9	0.045919800	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.045941002	192.168.32.100	192.168.32.101	TCP	54	443 → 50101 [ACK] Seq=1515 Ack=291 Win=64128 Len=0
11	0.046400357	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
12	0.052100726	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
13	0.248954730	192.168.32.101	192.168.32.100	TCP	60	50101 → 443 [ACK] Seq=291 Ack=1374 Win=64324 Len=0
14	0.762486415	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
15	1.735833985	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
16	3.194189914	192.168.32.101	192.168.32.100	DNS	81	Standard query 0x0764 A wpa.epicode.internal
17	3.227335002	192.168.32.100	192.168.32.101	DNS	97	Standard query response 0x0764 A wpa.epicode.internal A 192.168.32.100
18	3.228450650	192.168.32.101	192.168.32.100	TCP	60	50102 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
19	3.228478957	192.168.32.100	192.168.32.101	TCP	54	80 → 50102 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
20	3.274743159	192.168.32.101	192.168.32.100	TCP	60	[TCP Retransmission] 50102 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
21	3.788080690	192.168.32.100	192.168.32.101	TCP	54	80 → 50102 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
22	4.242457260	192.168.32.101	192.168.32.100	TCP	62	[TCP Retransmission] 50102 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM
23	4.242492055	192.168.32.100	192.168.32.101	TCP	54	80 → 50102 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
24	4.249577750	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
25	4.761297367	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
26	5.735964463	PcsCompu_10:6d:b4	Broadcast	ARP	60	Who has 192.168.32.17? Tell 192.168.32.101
27	7.307823044	192.168.32.101	192.168.32.100	TCP	60	50103 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
28	7.307917704	192.168.32.100	192.168.32.101	TCP	54	80 → 50103 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
29	7.900959759	192.168.32.101	192.168.32.100	TCP	60	[TCP Retransmission] 50103 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
30	7.900962373	192.168.32.100	192.168.32.101	TCP	54	80 → 50103 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
31	8.400601600	192.168.32.101	192.168.32.100	TCP	62	[TCP Retransmission] 50103 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM
32	8.400745970	192.168.32.100	192.168.32.101	TCP	54	80 → 50103 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
33	8.410733784	192.168.32.101	192.168.32.100	DNS	90	Standard query 0x0b05 A www.download.windowsupdate.com
34	8.408843391	192.168.32.100	192.168.32.101	DNS	160	Standard query response 0x0b05 A www.download.windowsupdate.com A 192.168.32.100
35	8.403654444	192.168.32.101	192.168.32.100	TCP	60	50104 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
36	8.403672430	192.168.32.100	192.168.32.101	TCP	54	80 → 50104 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
37	8.412847100	192.168.32.101	192.168.32.100	TCP	60	[TCP Retransmission] 50104 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
38	8.973911188	192.168.32.100	192.168.32.101	TCP	54	80 → 50104 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
39	9.481518578	192.168.32.101	192.168.32.100	TCP	62	[TCP Retransmission] 50104 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM

5) comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTP

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	60	50108 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000035922	192.168.32.100	192.168.32.101	TCP	60	80 → 50108 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000319446	192.168.32.101	192.168.32.100	TCP	60	50108 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.000547175	192.168.32.101	192.168.32.100	HTTP	350	GET /sample.txt HTTP/1.1
5	0.000561348	192.168.32.100	192.168.32.101	TCP	54	80 → 50108 [ACK] Seq=1 Ack=297 Win=64128 Len=0
6	0.017917031	192.168.32.100	192.168.32.101	TCP	204	80 → 50108 [PSH, ACK] Seq=1 Ack=297 Win=64128 Len=150 [TCP segment of a reassembled PDU]
7	0.019542880	192.168.32.100	192.168.32.101	HTTP	151	HTTP/1.1 200 OK (text/plain)
8	0.019779060	192.168.32.101	192.168.32.100	TCP	60	50108 → 80 [ACK] Seq=297 Ack=249 Win=65452 Len=0
9	0.019779262	192.168.32.101	192.168.32.100	TCP	60	50108 → 80 [FIN, ACK] Seq=297 Ack=249 Win=65452 Len=0
10	0.019811480	192.168.32.100	192.168.32.101	TCP	54	80 → 50108 [ACK] Seq=249 Ack=298 Win=64128 Len=0
11	5.252500888	PcsCompu_cb:7e:f5	PcsCompu_10:6d:b4	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
12	5.252669785	PcsCompu_10:6d:b4	PcsCompu_cb:7e:f5	ARP	60	192.168.32.101 is at 08:00:27:10:6d:b4

6) Differenze tra Http e Https

```
[Header checksum status: Unverified]
Source Address: 192.168.32.100
Destination Address: 192.168.32.101
▶ Transmission Control Protocol, Src Port: 80, Dst Port: 50108, S
▶ [2 Reassembled TCP Segments (247 bytes): #6(150), #7(97)]
▼ Hypertext Transfer Protocol
  ▶ HTTP/1.1 200 OK\r\n
    Date: Fri, 22 Dec 2023 18:58:16 GMT\r\n
    Server: INetSim HTTP Server\r\n
    Connection: Close\r\n
  ▶ Content-Length: 97\r\n
    Content-Type: text/plain\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 0.018995705 seconds]
    [Request in frame: 4]
  [Request URI: http://epicode.internal/sample.txt]
  File Data: 97 bytes
  ▶ Line-based text data: text/plain (5 lines)
```

Dai pacchetti catturati tramite il Tool Wireshark possiamo notare che i pacchetti Https sono Criptati e non permette la visualizzazione di ciò che l'utente sta eseguendo. possiamo vedere solo gli indirizzi Mac e gli IP. Mentre nell' Http notiamo come le stringhe dei pacchetti siano decisamente inferiori e che tutti i protocolli non sono criptati ma in chiaro permettendo di vedere la navigazione al dominio epicode.internal