

## REPORT SIMULAIZONE

Ho impostato l'IP 192.168.32.100 su Kali Linux tramite il comando `sudo nano /etc/network/interfaces`, salvato le modifiche e riavviato la macchina.

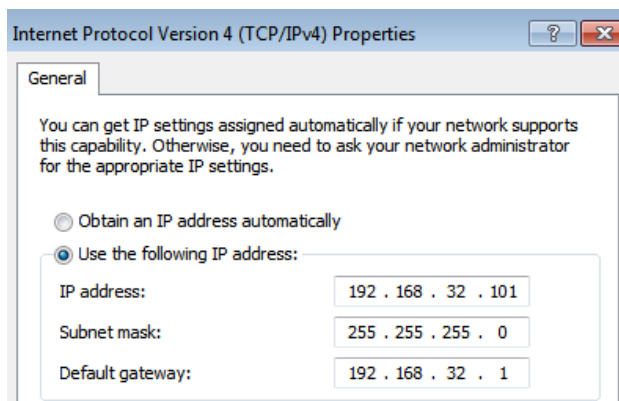
```
source /etc/network/interfaces.d/*

# 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

loop default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.100/24 brd 192.168.32.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Per impostare l'IP 192.168.32.101 su Windows 7 sono andata a cambiare le proprietà della Local Area Connection (percorso: *Control Panel\Network and Internet\Network Connections*) agendo nelle proprietà dell'*Internet Protocol Version 4 (TCP/IPv4)*.



```
C:\Users\Monia>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::a4df:b307:1ec1:a6e7%11
    IPv4 Address. . . . . : 192.168.32.101
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.32.1
```

Da specificare che entrambe le macchine lavorano in assenza di connessione di rete, impostata su *rete interna*, e a Windows 7 sono disabilitati i firewall.

Provando su entrambi i sistemi si riescono a pingare senza problemi.

```
(kali㉿kali)-[~]  
$ ping 192.168.32.101  
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.  
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=1.03 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.382 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.552 ms
```

```
C:\Users\Monia>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
```

Poi ho avviato InetSim su Kali e ho inserito i dati per preparare la simulazione, con il comando `sudo mousepad /etc/inetsim/intesim.conf` andando poi a configurare i dati di *Service\_bind\_address*, *DNS\_default\_IP* e *DNS\_static*, salvato e riavviato.

```
(kali㉿kali)-[~]  
$ sudo mousepad /etc/inetsim/intesim.conf
```

```
69 service_bind_address 192.168.32.100  
70
```

```
207 dns_default_ip 192.168.32.100  
208
```

```
242 dns_static epicode.internal 192.168.32.100  
243
```

Tornando nelle impostazioni di Windows 7, già precedentemente modificate, ho cambiato il *Preferred DNS server* in 192.168.32.100, e riavviato la macchina.

☒ Use the following DNS server addresses:

Preferred DNS server:	192 . 168 . 32 . 100
Alternate DNS server:	. . .

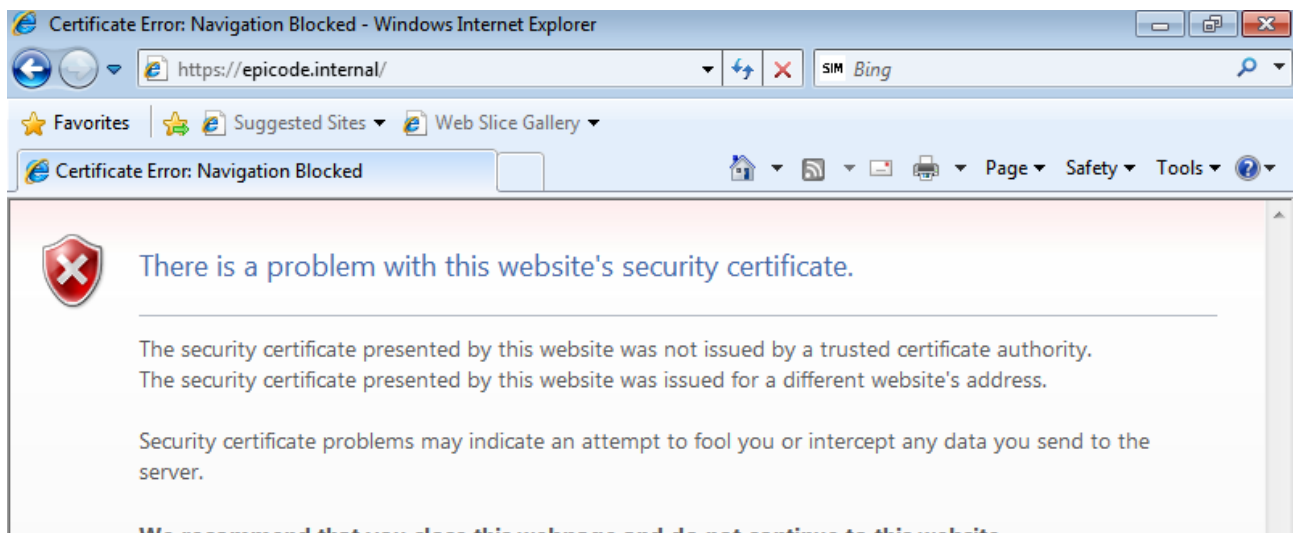
☒ Validate settings upon exit

Advanced...

Su Kali ho avviato la simulazione con il comando `sudo 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 2069) ==
Session ID:      2069
```

Su kali avvio il programma Wireshark e subito dopo su Windows7 inserisco l'indirizzo <https://epicode.internal> che segnala un problema di sicurezza in quanto non presente il certificato.



Quindi su Wireshark, un packet sniffer cioè un programma che analizza “catturando” i pacchetti che passano sulla rete, si può visualizzare l'indirizzo MAC selezionando il frame n.01 dove con “Who has 192.168.32.100? Tell 192.168.32.101” si chiede *Chi è il server e di dare risposta al Client*, nella sezione sottostante ci viene quindi indicato il MAC in quanto il protocollo ARP è quello incaricato di assegnarlo.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_af:3d:77	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000020749	PcsCompu_c7:e1:36	PcsCompu_af:3d:77	ARP	42	192.168.32.100 is at 08:00:27:c7:e1:36
3	0.000389949	192.168.32.101	192.168.32.100	TCP	66	49197 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=
4	0.000425283	192.168.32.100	192.168.32.101	TCP	66	443 → 49197 [SYN, ACK] Seq=0 Ack=1 Win=642
5	0.000600971	192.168.32.101	192.168.32.100	TCP	60	49197 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len
6	0.001127252	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
7	0.001154888	192.168.32.100	192.168.32.101	TCP	54	443 → 49197 [ACK] Seq=1 Ack=162 Win=64128
8	0.006190679	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exch
9	0.075127950	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, E
10	0.075168025	192.168.32.100	192.168.32.101	TCP	54	443 → 49197 [ACK] Seq=1320 Ack=296 Win=641
11	0.075924755	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Me
12	0.087715824	PcsCompu_af:3d:77	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
13	0.278862983	192.168.32.101	192.168.32.100	TCP	60	49197 → 443 [ACK] Seq=296 Ack=1379 Win=643
14	1.028250827	PcsCompu_af:3d:77	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101

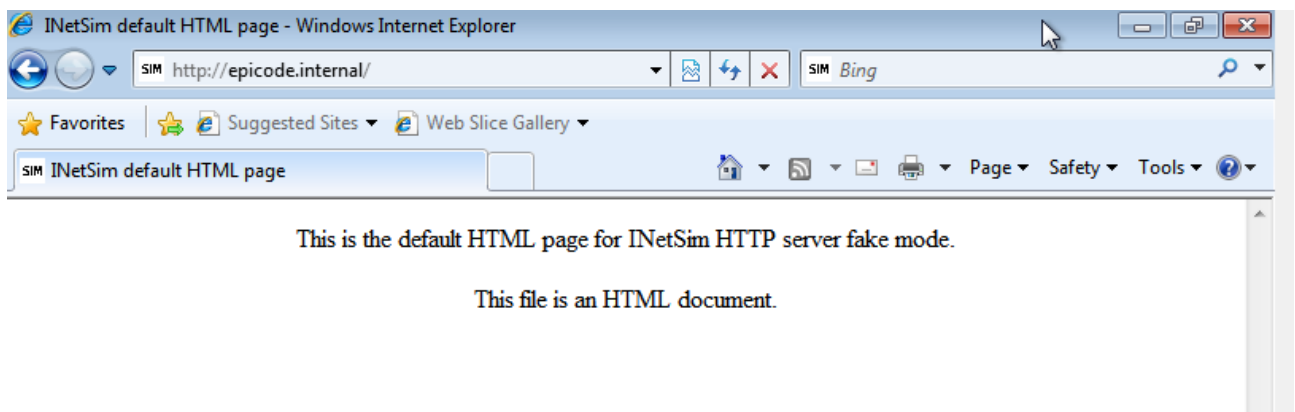
  

▶ Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface e	0000	ff ff ff ff ff ff 08 00	27 af
▶ Ethernet II, Src: PcsCompu_af:3d:77 (08:00:27:af:3d:77), Dst: Broadcast (ff:ff:ff	0010	08 00 06 04 00 01 08 00	27 af
▶ Destination: Broadcast (ff:ff:ff:ff:ff:ff)	0020	00 00 00 00 00 00 c0 a8	20 64
▶ Source: PcsCompu_af:3d:77 (08:00:27:af:3d:77)	0030	00 00 00 00 00 00 00 00	00 00
Type: ARP (0x0806)			
Padding: 00			
▶ Address Resolution Protocol (request)			

Filtrando, invece, tramite il numero della porta 443, cioè la porta dedicata all'https, vediamo sia la procedura di Three way handshake nel quale il client invia un pacchetto dati SYN, gli viene risposto con una ricevuta SYN/ACK dal server e il client risponde di nuovo con un pacchetto ACK, da qui si crea la connessione, che il protocollo TLSv1 che indica la crittografia prevista per l'https.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000277493	192.168.32.101	192.168.32.100	TCP	66	49188 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000312932	192.168.32.100	192.168.32.101	TCP	66	443 → 49188 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM
5	0.000525305	192.168.32.101	192.168.32.100	TCP	60	49188 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.000839368	192.168.32.101	192.168.32.100	TLSv1	190	Client Hello
7	0.000856362	192.168.32.100	192.168.32.101	TCP	54	443 → 49188 [ACK] Seq=1 Ack=137 Win=64128 Len=0
8	0.058929519	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
9	0.068592978	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10	0.068643288	192.168.32.100	192.168.32.101	TCP	54	443 → 49188 [ACK] Seq=1320 Ack=271 Win=64128 Len=0
11	0.069866446	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
13	0.274458011	192.168.32.100	192.168.32.101	TCP	113	[TCP Retransmission] 443 → 49188 [PSH, ACK] Seq=1320 Ack=271 Win=64128
14	0.274900753	192.168.32.101	192.168.32.100	TCP	66	49188 → 443 [ACK] Seq=271 Ack=1379 Win=64320 Len=0 SLE=1320 SRE=1379
111	34.358590467	192.168.32.101	192.168.32.100	TCP	60	49188 → 443 [FIN, ACK] Seq=271 Ack=1379 Win=64320 Len=0
112	34.358770175	192.168.32.100	192.168.32.101	TLSv1	91	Encrypted Alert
113	34.360412170	192.168.32.101	192.168.32.100	TCP	60	49188 → 443 [RST, ACK] Seq=272 Ack=1416 Win=0 Len=0

Ripetendo il procedimento ma inserendo sul browser di Windows 7 l'indirizzo <http://epicode.internal> accertando il funzionamento del collegamento.



In questo caso nella cattura di Wideshark invece del protocollo TLSv1 vediamo il protocollo HTTP che non prevede la crittografia.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_af:3d:77	Broadcast	ARP	60	Who has 192.168.32.100? Tell 192.168.32.101
2	0.000028123	PcsCompu_c7:e1:36	PcsCompu_af:3d:77	ARP	42	192.168.32.100 is at 08:00:27:c7:e1:36
3	0.000228580	192.168.32.101	192.168.32.100	TCP	66	49253 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000255570	192.168.32.100	192.168.32.101	TCP	60	80 → 49253 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM
5	0.000377650	192.168.32.101	192.168.32.100	TCP	60	49253 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.000638470	192.168.32.101	192.168.32.100	HTTP	361	GET / HTTP/1.1
7	0.000693523	192.168.32.100	192.168.32.101	TCP	54	80 → 49253 [ACK] Seq=1 Ack=308 Win=64128 Len=0
8	0.024379682	192.168.32.101	192.168.32.101	TCP	284	80 → 49253 [PSH, ACK] Seq=1 Ack=308 Win=64128 Len=150 [TCP segment o..
9	0.027583282	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
10	0.027899117	192.168.32.101	192.168.32.100	TCP	60	49253 → 80 [ACK] Seq=308 Ack=410 Win=65292 Len=0
11	0.028065800	192.168.32.101	192.168.32.100	TCP	60	49253 → 80 [FIN, ACK] Seq=308 Ack=410 Win=65292 Len=0
12	0.028087150	192.168.32.100	192.168.32.101	TCP	54	80 → 49253 [ACK] Seq=410 Ack=309 Win=64128 Len=0
13	0.049960214	192.168.32.101	192.168.32.100	TCP	66	49254 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
14	0.050018854	192.168.32.100	192.168.32.101	TCP	66	443 → 49254 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PER
15	0.050444310	192.168.32.101	192.168.32.100	TCP	60	49254 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
16	0.051544509	192.168.32.101	192.168.32.100	TCP	66	49255 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM

Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0  
 Ethernet II, Src: PcsCompu\_af:3d:77 (08:00:27:af:3d:77), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
 Source: PcsCompu\_af:3d:77 (08:00:27:af:3d:77)  
 Type: ARP (0x0806)  
 Padding: 00000000000000000000000000000000  
 Address Resolution Protocol (request)