

Esercitazione di fine modulo

Scopo dell'esercizio

Utilizzando i seguenti requisiti:

Kali Linux ☐ IP **192.168.32.100**

Windows 7 ☐ IP **192.168.32.101**

HTTPS server: **attivo**

Servizio DNS per risoluzione nomi di dominio: **attivo**

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) 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.

Premesse:

Per procedere con l'esercitazione sarà necessario usare inetsim , un applicativo per sistemi linux based il cui scopo è quello di simulare vari servizi tra cui anche Https-Http-Dns.

Fase 1: Setup degli ip

Seguendo i dati forniti iniziamo a impostare gli ip su Kali linux - windows 7 e successivamente su inetsim.

Kali linux:

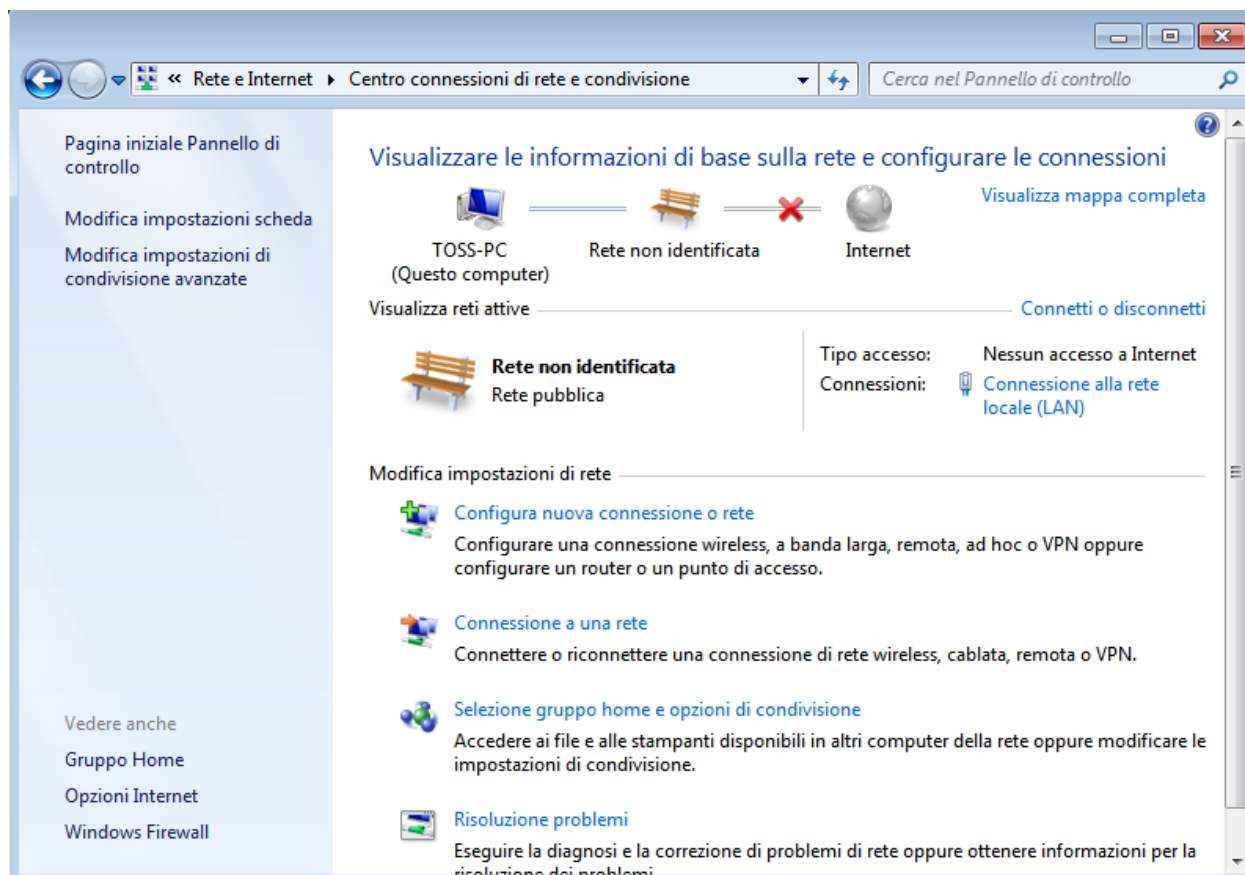
Usando l'editor di testo nano , bisogna modificare il file all'indirizzo **/etc/network/interfaces**. Successivamente è necessario cambiare l'ip come da consegna e di conseguenza anche il gateway.

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

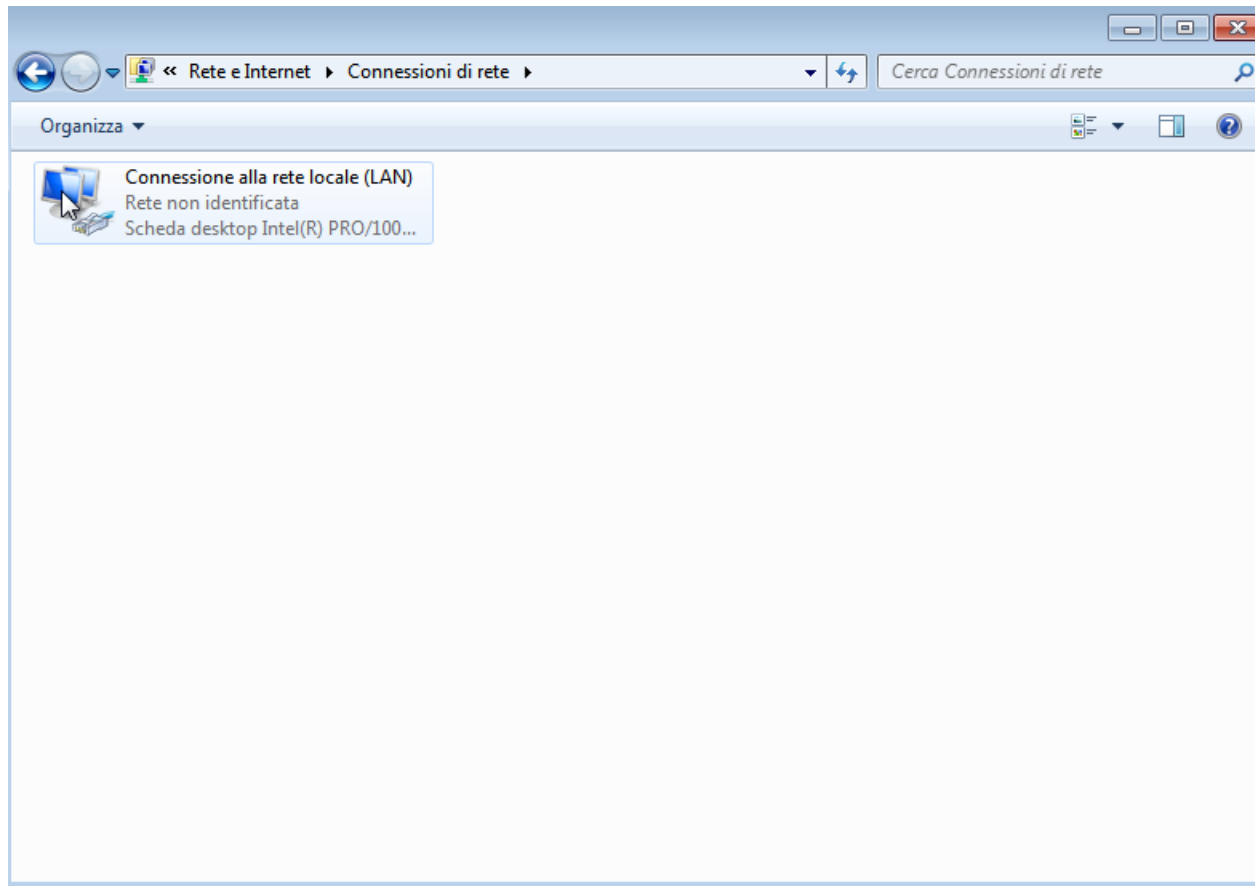
Dopo aver salvato la configurazione bisognerà riavviare l'interfaccia di rete utilizzando il comando **sudo systemctl restart networking**

Windows 7 :

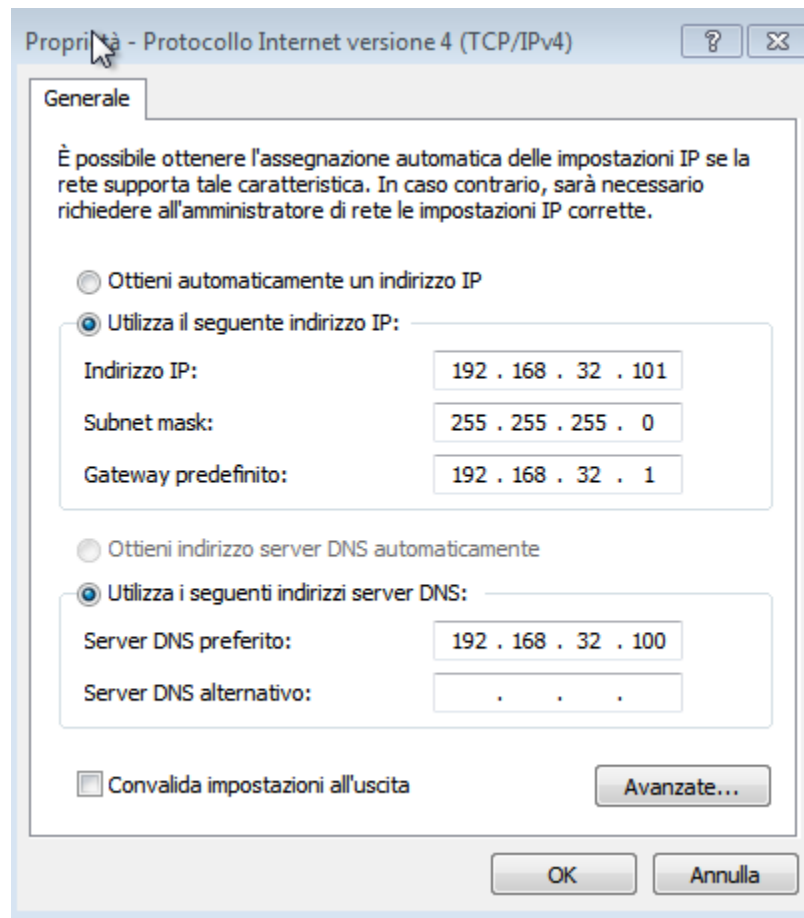
Da start digitare **centro di connessioni di rete e condivisione**



Modifica impostazioni scheda



Infine basterà cliccare sulla scheda di rete e modificare le proprietà del protocollo IPV4



nella sezione relativa al DNS va inserito l'ip della macchina che hosterà il server (kali linux), così facendo riusciremo a collegarci al sito sia via ip che via dominio.



Inetsim:

da terminale utilizziamo il comando **cd /etc/inetsim** per posizionarsi all'interno della cartella di Inetsim, successivamente utilizziamo il comando **ls** per visionare quale file .conf configurare. Utilizzando ancora una volta editor di testo nano digitiamo il comando **sudo nano inetsim.conf** per “accedere” al file conf.

Come prima modifica sarà necessario commentare usando il # tutte le funzioni non necessarie , in questo caso utilizzeremo http-https-dns

```
#
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
#start_service irc
#start_service ntp
#start_service finger
#start_service ident
#start_service syslog
#start_service time_tcp
#start_service time_udp
#start_service daytime_tcp
#start_service daytime_udp
#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
```

Nella sezione bind address inseriamo l'ip al quale vogliamo assegnare i servizi che abbiamo attivato. In questo caso possiamo inserire l'ip della macchina kali

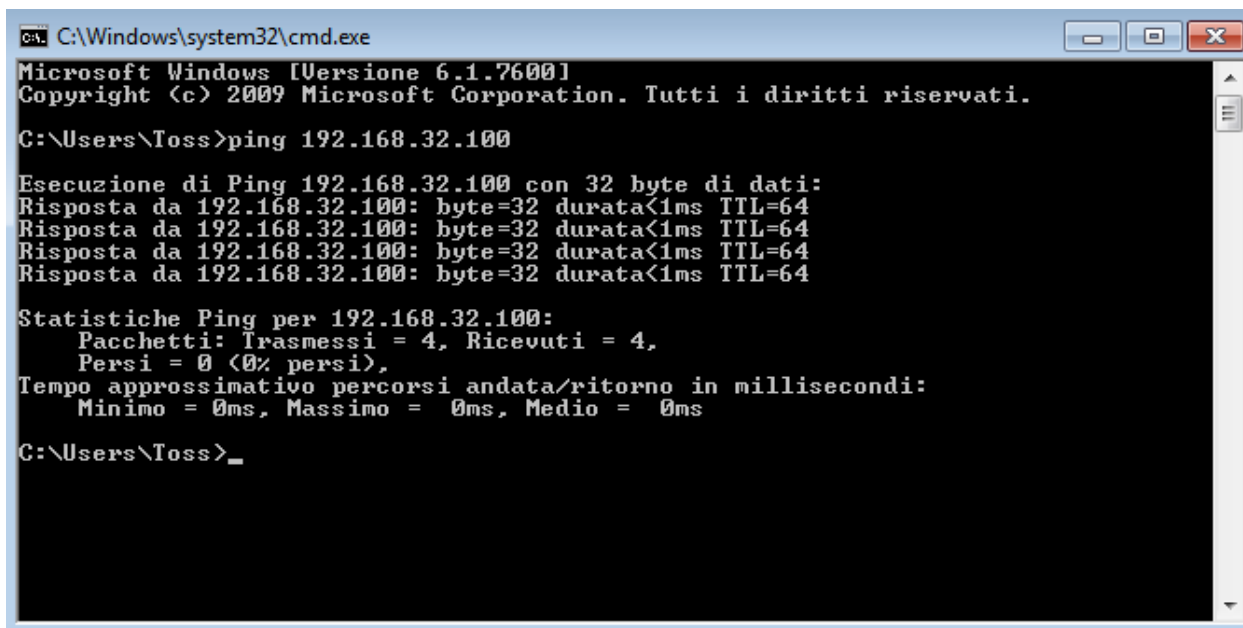
```
#####  
#service_bind_address  
#  
# IP address to bind services to  
#  
# Syntax: service_bind_address <IP address>  
#  
#Default: 127.0.0.1  
#  
service_bind_address 192.168.32.100
```

Assegniamo un DNS statico con ip uguale a quello della macchina kali e come host name **epicode.internal** così facendo quando inseriremo il dominio sul browser della macchina windows il DNS resolver risponderà al browser con il rispettivo ip.

Fase 2: Avvio di inetsim e verifica configurazioni

Salviamo la configurazione effettuata premendo CTRL+ X e successivamente Y + ENTER adesso lanciamo inetsim usando il comando **sudo inetsim**

Ancor prima di provare a collegarci al sito, per verificare che le due macchine comunichino correttamente, ci basterà lanciare un ping da windows a kali e vice versa.



```

C:\Windows\system32\cmd.exe
Microsoft Windows [Versione 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

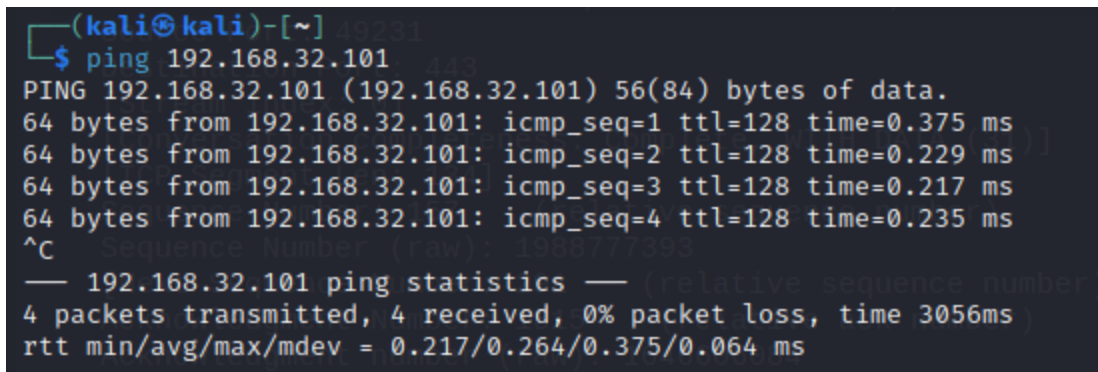
C:\Users\Toss>ping 192.168.32.100

Esecuzione di Ping 192.168.32.100 con 32 byte di dati:
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.32.100:
    Pacchetti: Trasmessi = 4, Ricevuti = 4,
    Persi = 0 (0% persi),
    Tempo approssimativo percorsi andata/ritorno in millisecondi:
        Minimo = 0ms, Massimo = 0ms, Medio = 0ms

C:\Users\Toss>_

```

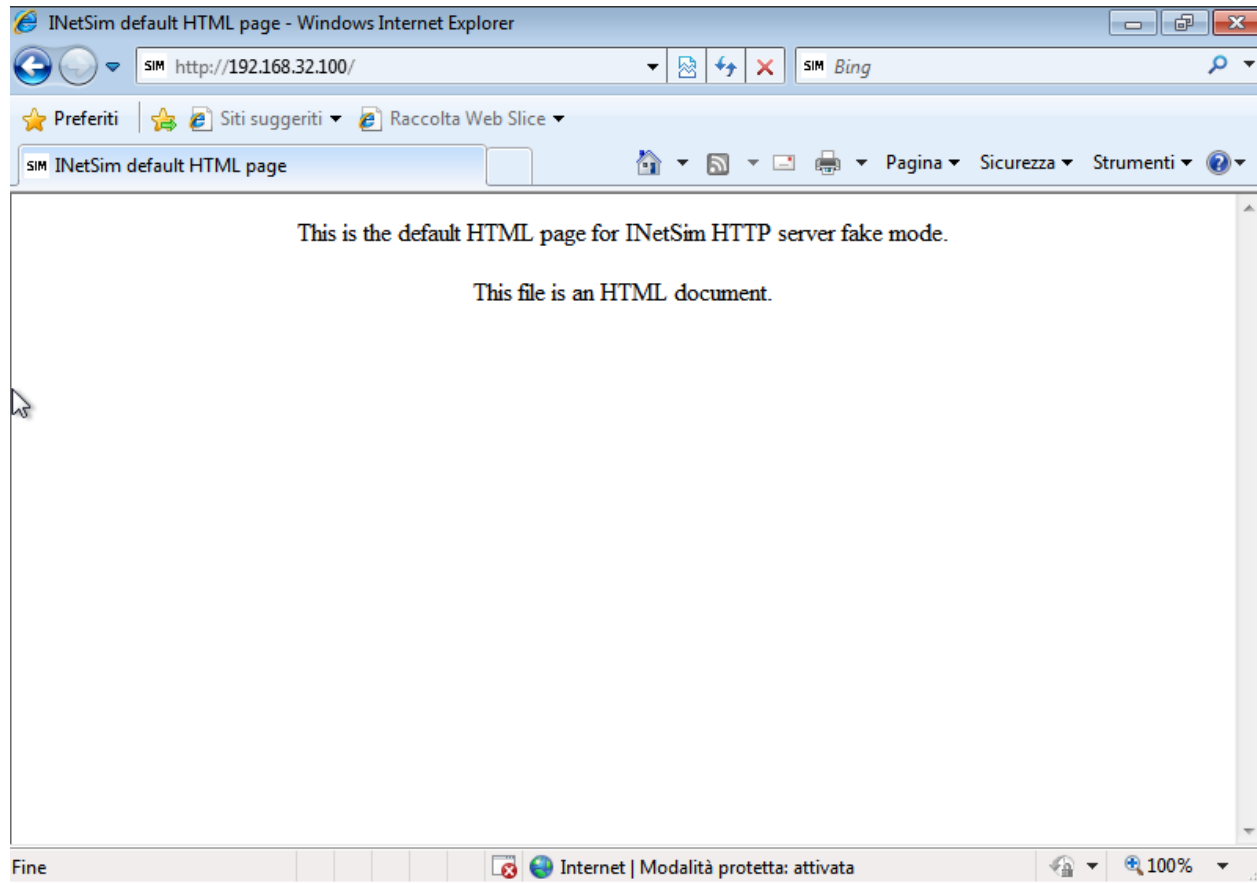


```

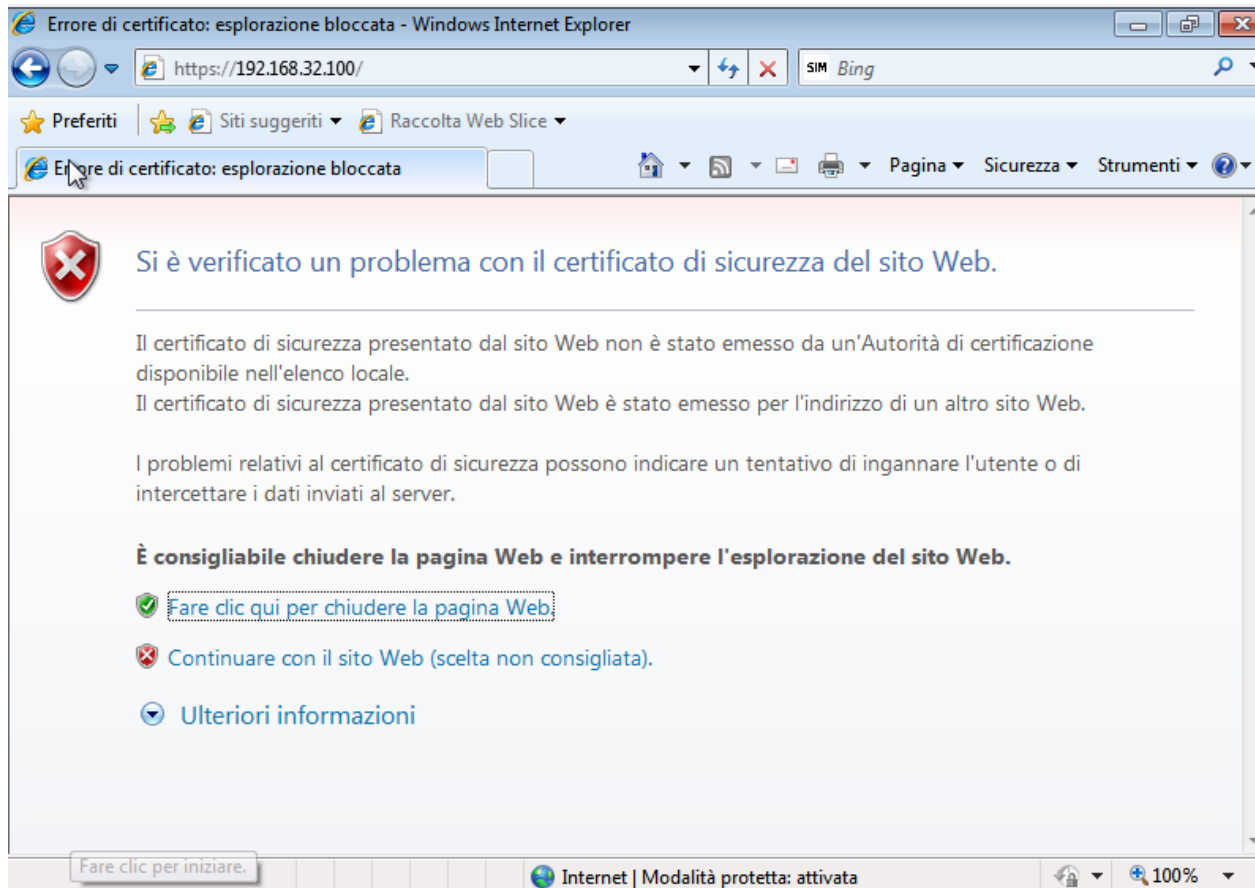
(kali㉿kali)-[~]
└─$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data:
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=0.375 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.229 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.217 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.235 ms
^C
Sequence number (raw): 196677393
— 192.168.32.101 ping statistics — (relative sequence number)
4 packets transmitted, 4 received, 0% packet loss, time 3056ms
rtt min/avg/max/mdev = 0.217/0.264/0.375/0.064 ms

```

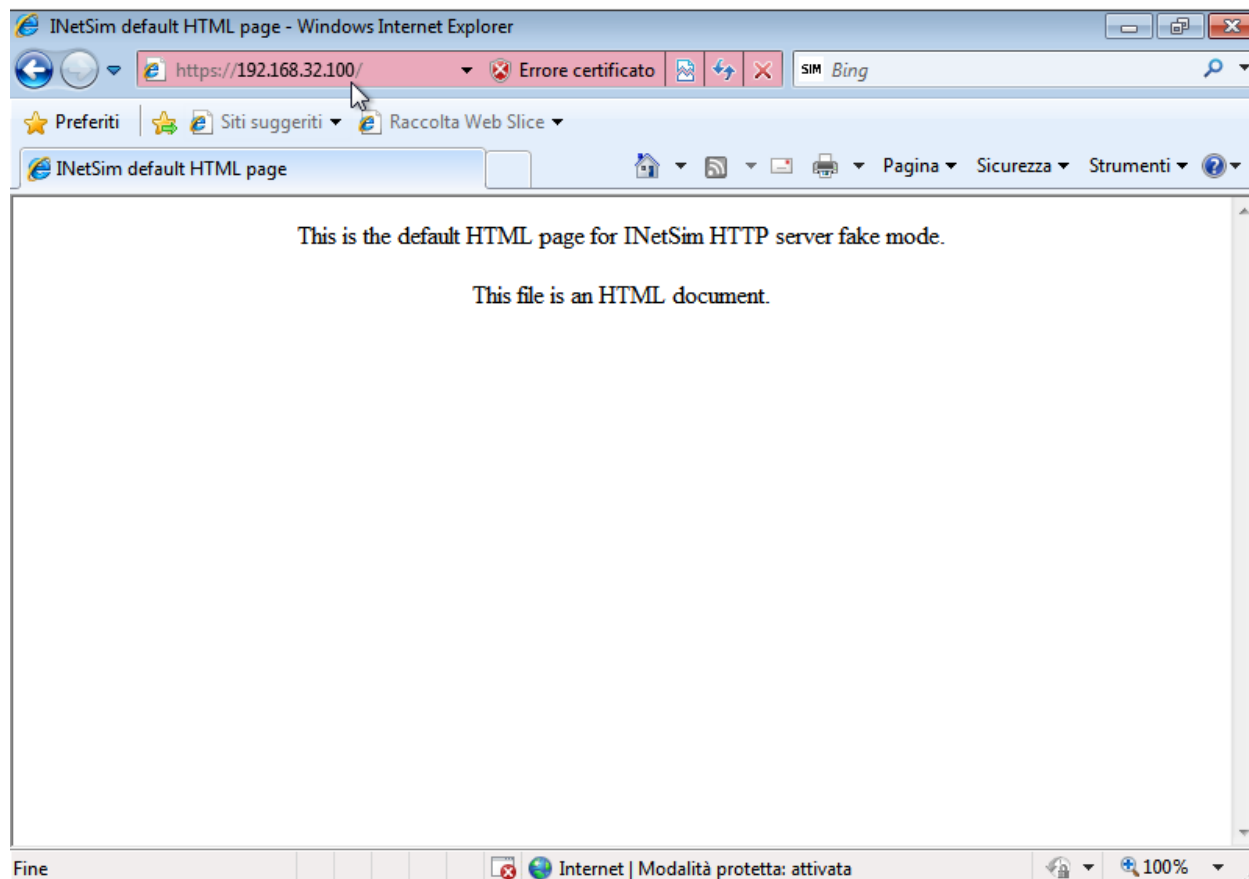
Confermato che le macchine comunicano correttamente, possiamo collegarci al server digitando l'ip sul browser preceduto da **http**



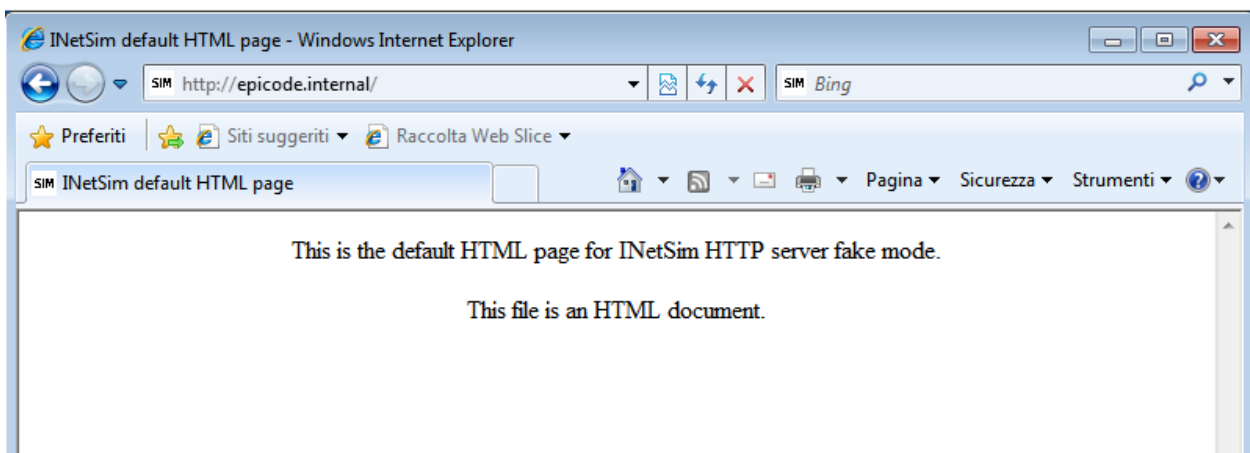
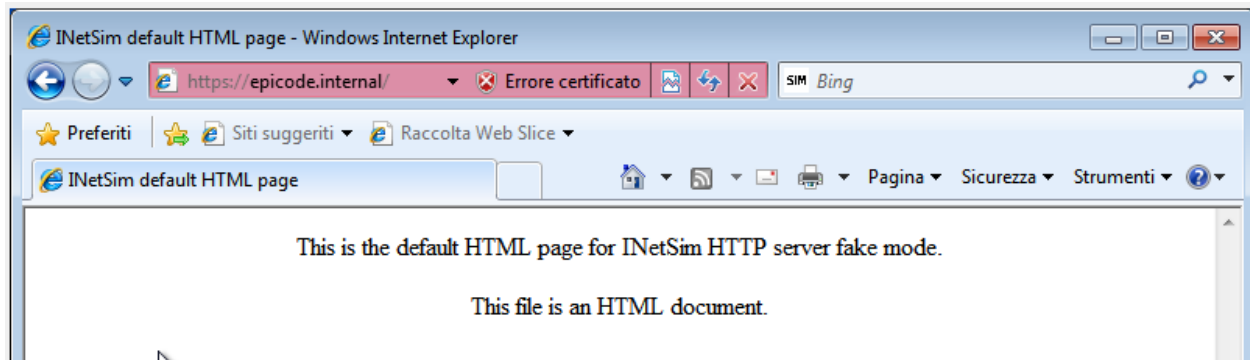
e successivamente da **https**



A causa dell'assenza di un certificato di sicurezza verrà promptato questo avviso , basterà cliccare su **continuare con il sito web**

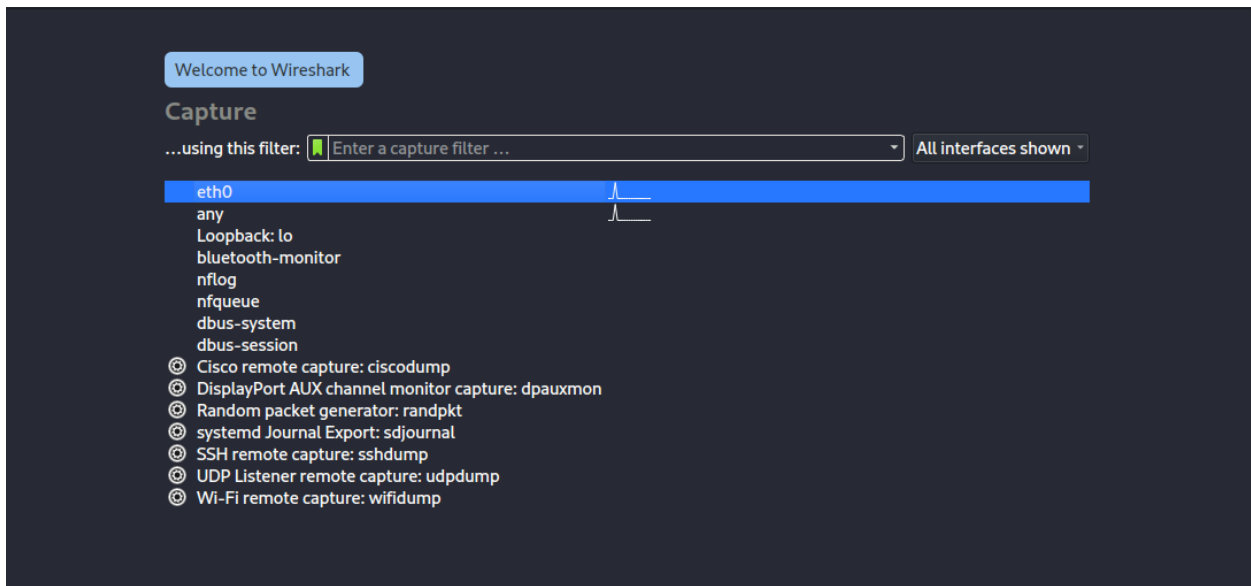


infine possiamo verificare che avvenga la risoluzione del dns digitando il nome host.



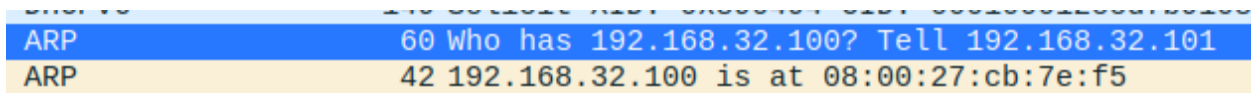
Fase 3: Analisi del traffico tramite wireshark

Iniziamo aprendo wireshark e selezionando l'interfaccia di rete corretta ossia quella precedentemente configurata eth0

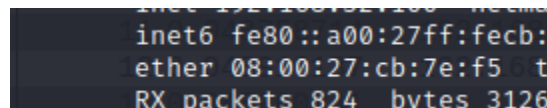


Analizziamo i pacchetti **http**:

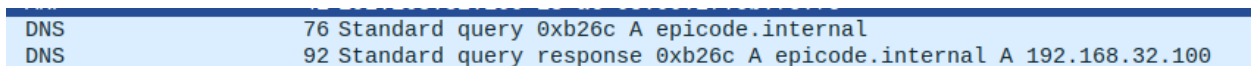
Avviamo la cattura dei pacchetti su wireshark e proviamo dunque ad accedere al sito dal browser di windows digitando **http://epicode.internal**.



tra i primi pacchetti catturati ci sarà il la richiesta broadcast del protocollo arp per l'associazione dell'indirizzo MAC all'ip. Digitando il comando **ifconfig** è osservando il MAC della scheda di rete eth0 possiamo confermare che è corretto



successivamente avviene la risoluzione del DNS



che restituisce al browser l'ip del server

16	15.495879959	192.168.32.101	192.168.32.100	TCP	66	49210 → 80	[SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
17	15.495907253	192.168.32.100	192.168.32.101	TCP	66	80 → 49210	[SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
18	15.496016312	192.168.32.101	192.168.32.100	TCP	60	49210 → 80	[ACK] Seq=1 Ack=1 Win=65700 Len=0
19	15.496096081	192.168.32.101	192.168.32.100	HTTP	361	GET / HTTP/1.1	
20	15.496104681	192.168.32.100	192.168.32.101	TCP	54	80 → 49210	[ACK] Seq=1 Ack=308 Win=64128 Len=0
21	15.508386932	192.168.32.100	192.168.32.101	TCP	204	80 → 49210	[PSH, ACK] Seq=1 Ack=308 Win=64128 Len=150 [TCP segment of a reassembled PDU]
22	15.510165919	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK	(text/html)
23	15.510288460	192.168.32.101	192.168.32.100	TCP	60	49210 → 80	[ACK] Seq=308 Ack=410 Win=65292 Len=0
24	15.510288600	192.168.32.101	192.168.32.100	TCP	60	49210 → 80	[FIN, ACK] Seq=308 Ack=410 Win=65292 Len=0
25	15.510310231	192.168.32.100	192.168.32.101	TCP	54	80 → 49210	[ACK] Seq=410 Ack=309 Win=64128 Len=0

Infine qui è possibile vedere rispettivamente:

La 3 way handshake tra client e server

la richiesta **GET** da parte di windows con successiva risposta da parte del server **codice 200**.

Notiamo che non prevedendo HTTP una connessione criptata è possibile vedere in chiaro il contenuto della pagina cliccando sul protocollo HTTP contenente la risposta da parte del server

[Request URI: http://epicode.internal/]		00b0	3e 49 4e 65 74 53 69 6d	20 64 65 66 61 75 6c
File Data: 258 bytes		00c0	20 48 54 4d 4c 20 70 61	67 65 3c 2f 74 69 74
Line-based text data: text/html (10 lines)		00d0	65 3e 0a 20 20 3c 2f 68	65 61 64 3e 0a 20 20
<html>\n		00e0	62 6f 64 79 3e 0a 20 20	20 20 3c 70 3e 3c 2f
<head>\n		00f0	3e 0a 20 20 20 20 3c 70	20 61 6c 69 67 6e 3d
<title>INetSim default HTML page</title>\n		0100	63 65 6e 74 65 72 22 3e	54 68 69 73 20 69 73
</head>\n		0110	74 68 65 20 64 65 66 61	75 6c 74 20 48 54 4d
<body>\n		0120	20 70 61 67 65 20 66 6f	72 20 49 4e 65 74 53
<p>\n		0130	6d 20 48 54 54 50 20 73	65 72 76 65 72 20 66
<p align="center">This is the default HTML page for INetSim HTTP server fake mode.</p>\n		0140	6b 65 20 6d 6f 64 65 2e	3c 2f 70 3e 0a 20 20

Frame (312 bytes) Reassembled TCP (408 bytes)

Analizziamo adesso i pacchetti **https**:

La prima cosa che è possibile notare è che essendo il traffico criptato dopo la 3 way handshake segue la TLS handshake. Durante questa fase il client inizia la handshake con un **client hello** nel quale sono contenuti la versione di TLS supportata e la suite di crittografia supportata. Il server risponde con il **server hello**, dove sono contenute il certificato ssl del server, la suite di cifratura scelta dal server

TCP	66	49231 → 443	[SYN]	Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
TCP	66	443 → 49231	[SYN, ACK]	Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
TCP	60	49231 → 443	[ACK]	Seq=1 Ack=1 Win=65700 Len=0
TLSv1	210	Client Hello		
TCP	54	443 → 49231	[ACK]	Seq=1 Ack=157 Win=64128 Len=0
TLSv1	1368	Server Hello, Certificate, Server Key Exchange, Server Hello Done		
TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message		
TLSv1	113	Change Cipher Spec, Encrypted Handshake Message		
TLSv1	395	Application Data		
TLSv1	235	Application Data		
TLSv1	384	Application Data, Encrypted Alert		
TCP	60	49231 → 443	[ACK]	Seq=632 Ack=1886 Win=65700 Len=0
TCP	60	49231 → 443	[FIN, ACK]	Seq=632 Ack=1886 Win=65700 Len=0
TCP	54	443 → 49231	[ACK]	Seq=1886 Ack=633 Win=64128 Len=0

essendo le successive informazioni criptate, non è possibile proseguire con l'analisi come nel caso del protocollo http, rimarranno comunque visibili l'assegnazione del mac all'ip e gli ip sorgente e destinazione.

Conclusioni:

Dunque le differenze tra http e https è l'assenza della TLS handshake, l'impossibilità di vedere in chiaro la pagina HTML e dunque l'assenza della richiesta GET effettuata dal browser per accedere alla risorsa