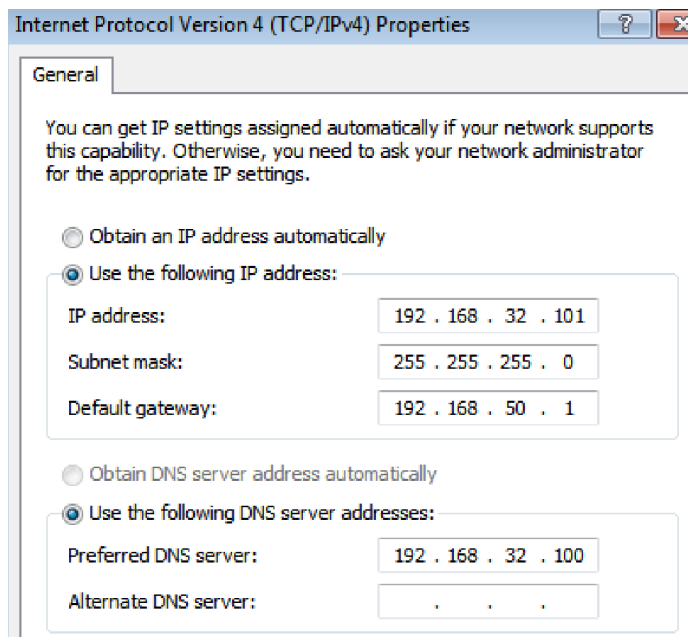


5. Simulazione di rete complessa

Per risolvere tale esercizio iniziamo dal setup degli indirizzi IP che ci vengono forniti.

In Windows 7 è necessario modificare le impostazioni del protocollo IPv4 accedendo dal pannello di controllo alle impostazioni della scheda di rete come si vede in figura.



Così facendo Windows 7 avrà IP 192.168.32.101. Si noti inoltre che ho già impostato il server DNS con l'IP di Kali (passaggio che sarà approfondito più avanti). Aggiungiamo le eventuali regole al firewall di Windows così da permettere la comunicazione.

Ora andiamo a modificare l'indirizzo IP di Kali, modificando come segue il file nel percorso: /etc/network/interfaces.

```
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 192.168.50.1
```

Fatto ciò si può eseguire un test di ping da una qualsiasi delle 2 macchine virtuali per verificare il corretto funzionamento del collegamento.

Il prossimo passo prevede il setup dell'applicazione Inetsim per emulare i servizi di rete.

Procediamo nel modificare il file al seguente percorso:

/etc/inetsim/inetsim.conf

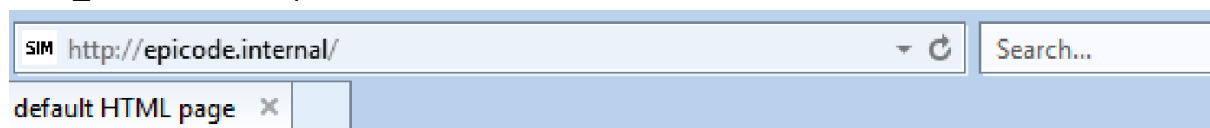
```
#####
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.32.100

#####
# 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 epicode.internal 192.168.32.100
```

Nella prima immagine è stato assegnato l'IP al server DNS mentre nella seconda è stata aggiunta la regola tale per cui ad ogni richiesta del tipo "epicode.internal" il client verrà reindirizzato all'IP desiderato, in questo caso quello della nostra macchina.

Per verificare la corretta impostazione di Inetsim lanciamo tale applicazione dal terminale di Kali.

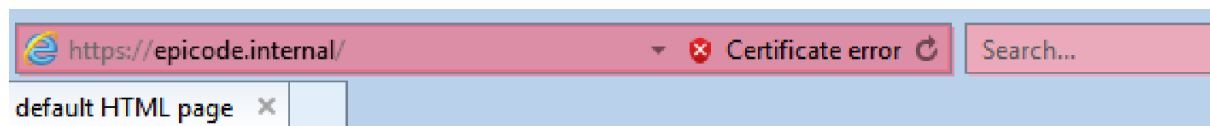
Passiamo ora al browser di Windows 7 (internet explorer purtroppo) dove andremo a digitare l'host_name ovvero "epicode.internal" così ottenendo:



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

This file is an HTML document.

ripetiamo l'operazione inotrando una richiesta del tipo HTTPS ottendo:



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

This file is an HTML document.

Notiamo subito che il browser cerca di proteggerci da un eventuale minaccia dato che il server fittizio di Inetsim è sprovvisto dei necessari certificati SSL necessari per la crittografia TLS.

Infine torniamo su Kali aprendo Wireshark, necessario per sniffare i pacchetti della conversazione sul canale eth0.

E ripetiamo sia la richiesta HTTP che quella HTTPS dal browser di windows 7 così da catturare i pacchetti.

No.	Time	Source	Destination	Protocol	Length	Info
37	5.315862482	192.168.32.100	192.168.32.101	TCP	54	80 → 49275 [FIN, ACK] Seq=409 Ack=253 Win=64128 Len=0
38	5.316131019	192.168.32.101	192.168.32.100	TCP	60	49275 → 80 [ACK] Seq=253 Ack=410 Win=65280 Len=0
39	5.874982252	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
40	7.121781263	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
41	7.274914352	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
42	7.875554419	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
43	8.875149462	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
44	10.265606066	192.168.32.101	192.168.32.100	TCP	66	49276 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
45	10.265672360	192.168.32.100	192.168.32.101	TCP	66	80 → 49276 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
46	10.266374262	192.168.32.101	192.168.32.100	TCP	60	49276 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
47	10.267144229	192.168.32.101	192.168.32.100	HTTP	249	GET /msdownload/update/v3/static/trustedr/en/authrootstl.cab?69f01908c3eac8d8 HTTP/1.1
48	10.267182381	192.168.32.100	192.168.32.101	TCP	54	80 → 49276 [ACK] Seq=1 Ack=196 Win=64128 Len=0
49	10.305437112	192.168.32.100	192.168.32.101	TCP	284	80 → 49276 [PSH, ACK] Seq=1 Ack=196 Win=64128 Len=150 [TCP segment of a reassembled PDU]
50	10.309037213	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
51	10.399379744	192.168.32.101	192.168.32.100	TCP	60	49276 → 80 [ACK] Seq=196 Ack=410 Win=65280 Len=0
52	10.399568273	192.168.32.101	192.168.32.100	TCP	60	49276 → 80 [FIN, ACK] Seq=196 Ack=410 Win=65280 Len=0
53	10.399587740	192.168.32.100	192.168.32.101	TCP	54	80 → 49276 [ACK] Seq=410 Ack=197 Win=64128 Len=0
54	10.324899221	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
55	10.416480582	192.168.32.101	192.168.32.100	TCP	66	49277 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1

Frame 50: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_22:46:4f (08:00:27:22:46:4f), Dst: PcsCompu_51:72:7a (08:00:27:51:72:7a)

Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101

Transmission Control Protocol, Src Port: 80, Dst Port: 49276, Seq: 151, Ack: 196, Len: 258

[2 Reassembled TCP Segments (488 bytes): #49(150), #50(238)]

Hypertext Transfer Protocol

Line-based text data: text/html (10 lines)

```
<html>\n
<head>\n
<title>InetSim default HTML page</title>\n
</head>\n
<body>\n
<p></p>\n
<p align="center">This is the default HTML page for InetSim HTTP server fake mode.</p>\n
```

In questa conversazione di tipo HTTP possiamo notare evidenziati gli indirizzi IP e MAC dei destinatari e riceventi, oltre alla porta 80 caratteristica di questo tipo di comunicazione, 200 ovvero il codice di avvenuta comunicazione e in fondo il codice HTML della pagina.

No.	Time	Source	Destination	Protocol	Length	Info
7	1.081678551	192.168.32.100	192.168.32.101	TCP	66	443 → 49234 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1
8	1.081921779	192.168.32.101	192.168.32.100	TCP	60	49234 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
9	1.082451852	192.168.32.101	192.168.32.100	TLSv1.2	271	Client Hello
10	1.082488768	192.168.32.100	192.168.32.101	TCP	54	443 → 49234 [ACK] Seq=1 Ack=218 Win=64128 Len=0
11	1.106422987	192.168.32.100	192.168.32.101	TLSv1.2	1821	Server Hello, Certificate, Server Key Exchange, Server Hello Done
12	1.106840770	192.168.32.101	192.168.32.100	TCP	60	49234 → 443 [ACK] Seq=218 Ack=1768 Win=65700 Len=0
13	1.128764700	192.168.32.101	192.168.32.100	TLSv1.2	372	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
14	1.128798201	192.168.32.100	192.168.32.101	TCP	54	443 → 49234 [ACK] Seq=1768 Ack=536 Win=64128 Len=0
15	1.134534001	192.168.32.100	192.168.32.101	TLSv1.2	105	Change Cipher Spec, Encrypted Handshake Message
16	1.134850737	192.168.32.101	192.168.32.100	TCP	60	49234 → 443 [ACK] Seq=536 Ack=1819 Win=65648 Len=0
17	1.154230771	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
18	1.874427269	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
19	2.874636773	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
20	4.294135854	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
21	4.874968550	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
22	5.874969895	PcsCompu_51:72:7a	Broadcast	ARP	60	Who has 192.168.50.1? Tell 192.168.32.101
23	6.273770204	PcsCompu_22:46:4f	PcsCompu_51:72:7a	ARP	42	Who has 192.168.32.101? Tell 192.168.32.100
24	6.274291991	PcsCompu_51:72:7a	PcsCompu_22:46:4f	ARP	60	192.168.32.101 is at 08:00:27:51:72:7a
25	7.425041417	192.168.32.101	192.168.32.100	DNS	83	Standard query 0xa5fc A ctldl.windowsupdate.com

Frame 15: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_22:46:4f (08:00:27:22:46:4f), Dst: PcsCompu_51:72:7a (08:00:27:51:72:7a)

Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101

Transmission Control Protocol, Src Port: 443, Dst Port: 49234, Seq: 1768, Ack: 536, Len: 51

Transport Layer Security

- TLSv1.2 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
- TLSv1.2 Record Layer: Handshake Protocol: Encrypted Handshake Message

Content Type: Handshake (22)

Version: TLS 1.2 (0x0303)

Length: 40

Handshake Protocol: Encrypted Handshake Message

Per ultimo andiamo ad analizzare la conversazione HTTPS che come da immagine inizia comunicando dalla porta 443 utilizzando il metodo TLS v1.2 e notando la presenza del transport layer security. Tuttavia essendo il server sprovvisto di certificato SSL la connessione non sarà sicura nè criptata.

è possibile aggiungere il certificato a Inetsim modificando il file di configurazione in tal modo:

```
#####
# https_ssl_certfile
#
# Name of the SSL certificate file.
#
# The file must be placed in <data-dir>/certs/
#
# Syntax: https_ssl_certfile <filename>
#
# Default: default_cert.pem
#
https_ssl_certfile https_cert.pem
```

così da ottenere una conversazione criptata.