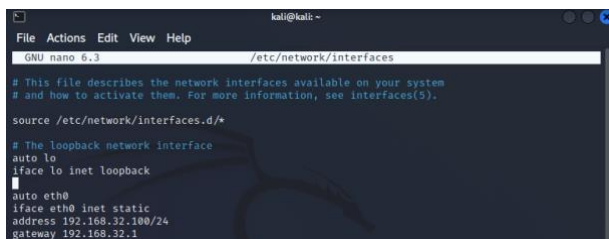


Il progetto consiste nel:

1. Simulare virtualmente un'architettura client server dove il client richiede tramite browser una risorsa all'hostname "epicode.internal" che risponde all'indirizzo IP associato alla macchina Kali
2. Intercettare la comunicazione con Wireshark, evidenziando MAC address di sorgente e di destinazione e il contenuto della richiesta HTTPS
3. Fare stessa procedura del punto 2 cambiando il protocollo da HTTPS a HTTP

1.

Per iniziare si configurano gli indirizzi IP delle due macchine virtuali



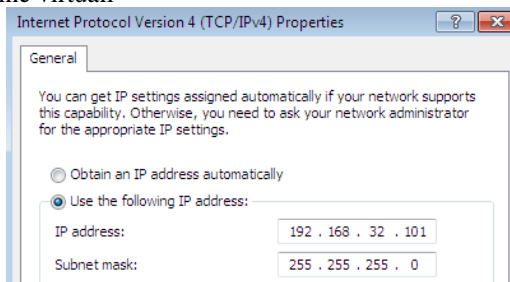
```
GNU nano 6.3 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

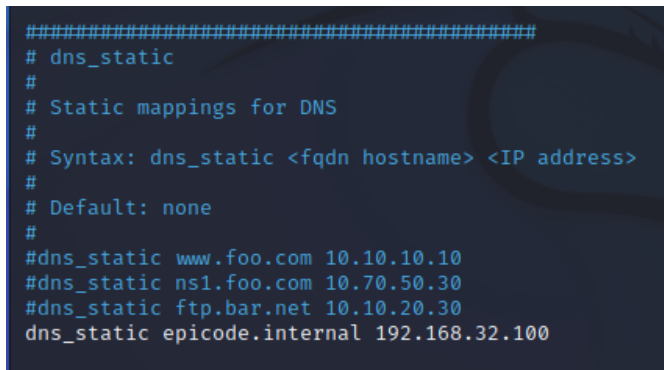
# The ethernet interface
auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1
```

IP address di Kali



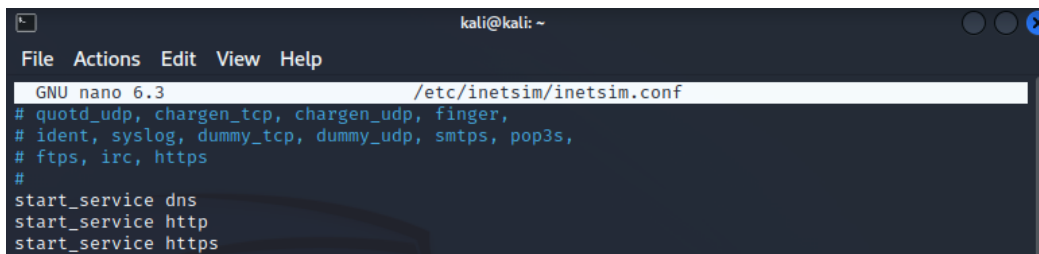
IP address di Windows

Una volta assegnati gli IP si passa a configurare il DNS da inetSim (con il comando "sudo nano /etc/inetsim/inetsim.conf") associandogli il nome di "epicode.internal" all'IP di Kali



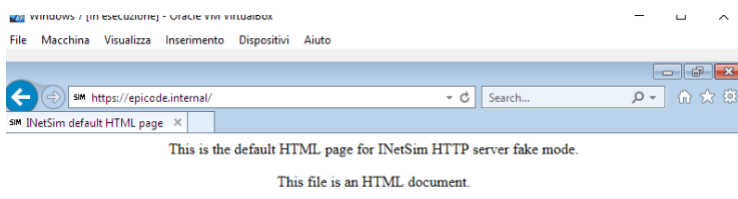
```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static epicode.internal 192.168.32.100
```

Assicurandosi che i servizi DNS-HTTP-HTTPS che serviranno per il progetto siano tutti attivati



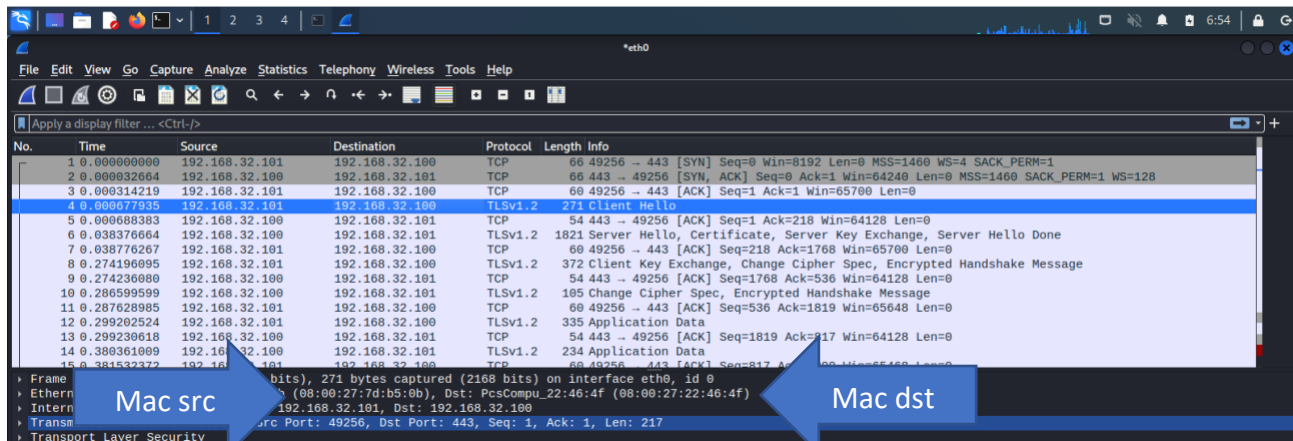
```
File Actions Edit View Help
GNU nano 6.3 /etc/inetsim/inetsim.conf
# quot_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
start_service http
start_service https
```

Apriamo internet explorer da windows, cerchiamo il nostro DNS e si aprirà la pagina HTML di default di inetsim

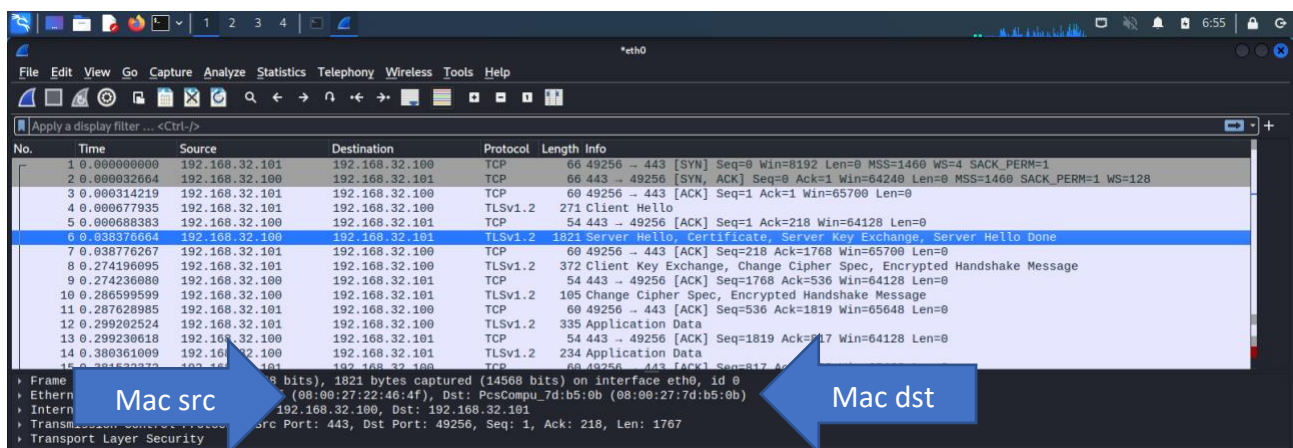


2.

Una volta associato il DNS lo cerchiamo su internet con il protocollo HTTPS e facciamo partire l'applicazione "Wireshark" di kali per intercettare la comunicazione tra client e server



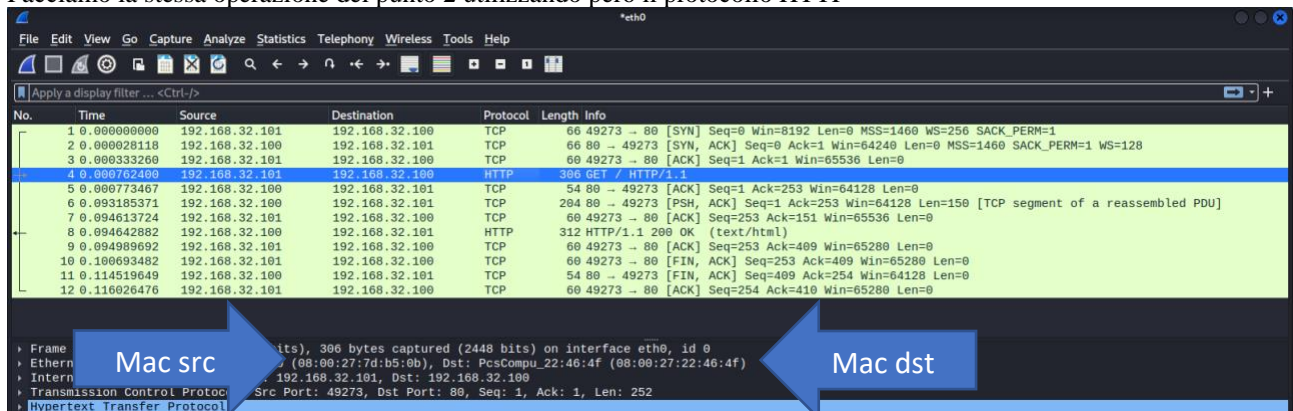
Lato Client



Lato Server

3.

Facciamo la stessa operazione del punto 2 utilizzando però il protocollo HTTP



Lato Client

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49273 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000028118	192.168.32.100	192.168.32.101	TCP	66	80 → 49273 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000333260	192.168.32.101	192.168.32.100	TCP	60	49273 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
4	0.000762400	192.168.32.101	192.168.32.100	HTTP	306	GET / HTTP/1.1
5	0.000773467	192.168.32.100	192.168.32.101	TCP	54	80 → 49273 [ACK] Seq=1 Ack=253 Win=64128 Len=0
6	0.093185371	192.168.32.100	192.168.32.101	TCP	204	80 → 49273 [PSH, ACK] Seq=1 Ack=253 Win=64128 Len=150 [TCP segment of a reassembled PDU]
7	0.094613724	192.168.32.101	192.168.32.100	TCP	60	49273 → 80 [ACK] Seq=253 Ack=151 Win=65536 Len=0
8	0.094647802	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
9	0.094989692	192.168.32.101	192.168.32.100	TCP	60	49273 → 80 [ACK] Seq=253 Ack=409 Win=65280 Len=0
10	0.100693482	192.168.32.101	192.168.32.100	TCP	60	49273 → 80 [FIN, ACK] Seq=253 Ack=409 Win=65280 Len=0
11	0.114519649	192.168.32.100	192.168.32.101	TCP	54	80 → 49273 [FIN, ACK] Seq=409 Ack=254 Win=64128 Len=0
12	0.116026476	192.168.32.101	192.168.32.100	TCP	60	49273 → 80 [ACK] Seq=254 Ack=410 Win=65280 Len=0

Frame 8 (312 bytes captured on interface eth0, id 0)
Ethernet II, Src: PcsCompu.7d:b5:0b (08:00:27:7d:b5:0b), Dst: 192.168.32.101 (08:00:27:22:46:4f)
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
Transmission Control Protocol, Src Port: 80, Dst Port: 49273, Seq: 151, Ack: 253, Len: 258
2 Reassembled TCP Segments (258 bytes): #6(150), #8(258)
Hypertext Transfer Protocol
Line-based text data: text/html (10 lines)

Lato Server

In conclusione, possiamo notare che le principali differenze sono:

1. Indirizzo MAC diverso per client e server
2. Come si può vedere con il protocollo HTTPS le informazioni sono criptate dal protocollo di cifratura TLS che non lascia far leggere i messaggi tra client e server rendendoli più sicuri. Al contrario del protocollo HTTP che non avendo questo protocollo di cifratura, lascia in chiaro i messaggi che si scambiano client e server lasciando libera lettura a chiunque.