Progetto fine Modulo 1

Rebecca Caldarella

Requisiti e servizi:

- 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

Traccia:

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.

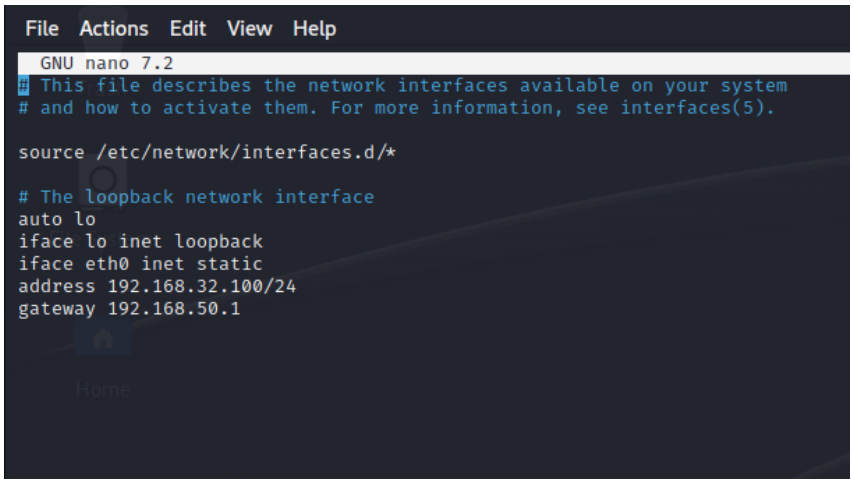
Spiegare, motivandole, le principali differenze se presenti.

Svolgimento

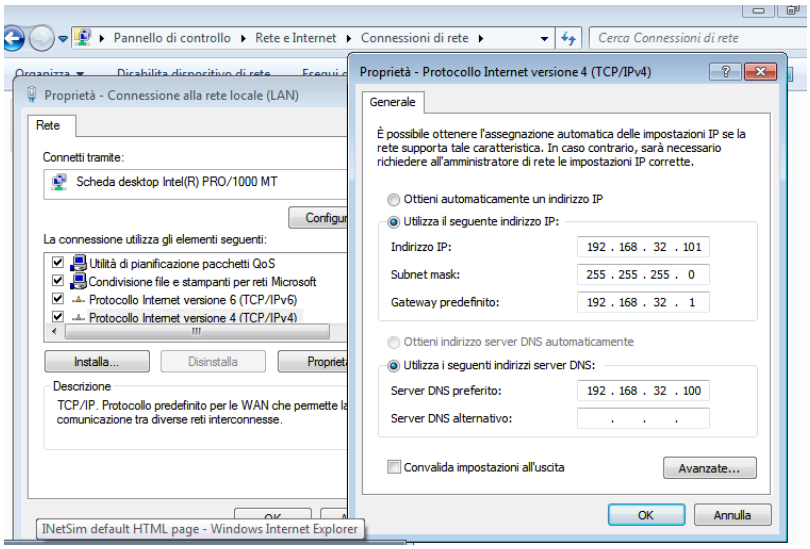
I. Configurazione IP Kali Linux e Windows 7

L'esercitazione prevede l'utilizzo delle macchine virtuali Kali Linux e Windows 7. Per fare in modo che le macchine comunichino tra loro il primo passo è configurare gli indirizzi IP dei due ambienti:

Kali Linux con IP 192.168.32.100

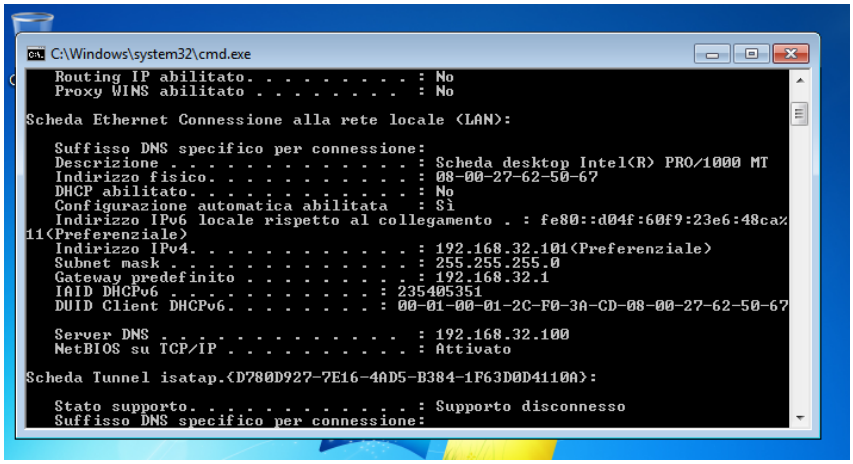


E Windows 7 con IP 192.168.32.101



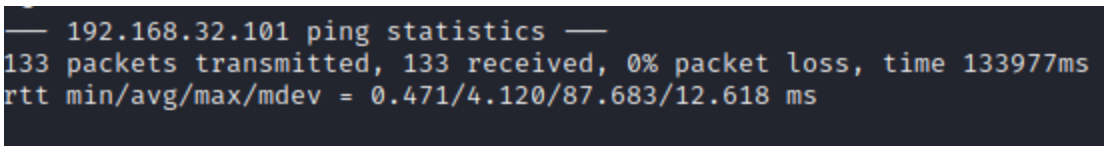
Per quanto riguarda il Windows 7 è stato necessario anche configurare il servizio DNS, in quanto sarà questa la macchina che andrà a richiedere al browser l'hostname epicode.internal. Da traccia, infatti, l'IP del server DNS deve corrispondere a quello della macchina Kali Linux, pertanto è stato inserito l'IP 192.168.32.100 nella voce "server DNS preferito".

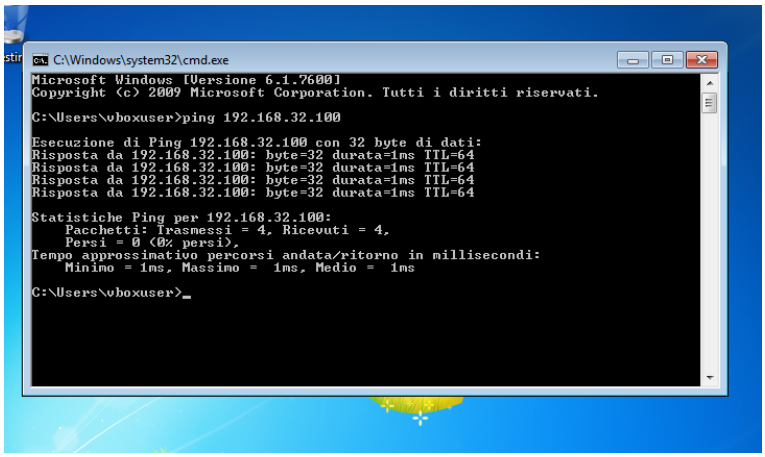
Per verificare che la configurazione sia avvenuta con successo basta aprire il prompt dei comandi e lanciare il comando "ipconfig all":



da qui infatti è possibile vedere che l'indirizzo IPv4 e il Server DNS sono quelli impostati manualmente tramite Pannello di Controllo.

Una volta configurate entrambi gli ambienti virtuali, possiamo assicurarci che comunichino correttamente tramite il comando "Ping":



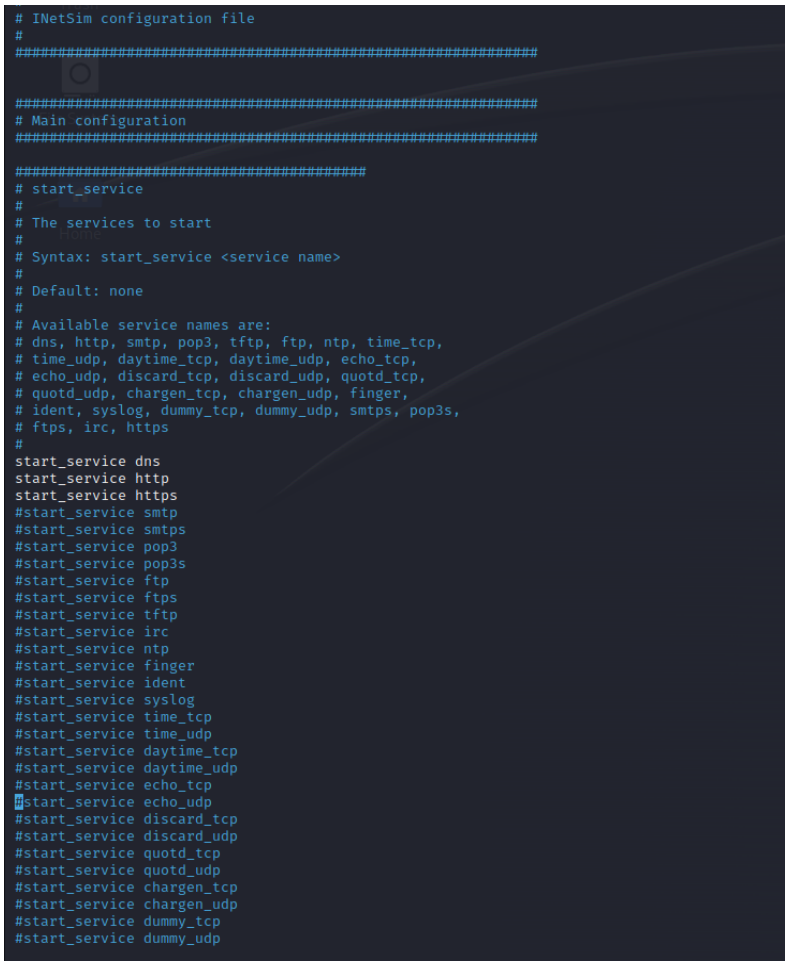


II. Configurazione INetSim

INetSim è un tool oper source che ci permette di simulare i servizi internet standard, per cui seguendo la traccia lo utilizzeremo per configurare i servizi DNS, HTTP e HTTPS.

Per poter effettuare le modifiche che ci interessano bisogna lanciare dal terminale di Kali Linux il comando “sudo nano /etc/inetsim/” e recarci all’interno di “inetsim.conf”. Il comando “sudo” ci fornisce i permessi di super amministratore e il comando “nano” ci permette di aprire il file in modo da poterlo editare.

Da qui possiamo mantenere attivi i servizi che ci interessano, come da figura:



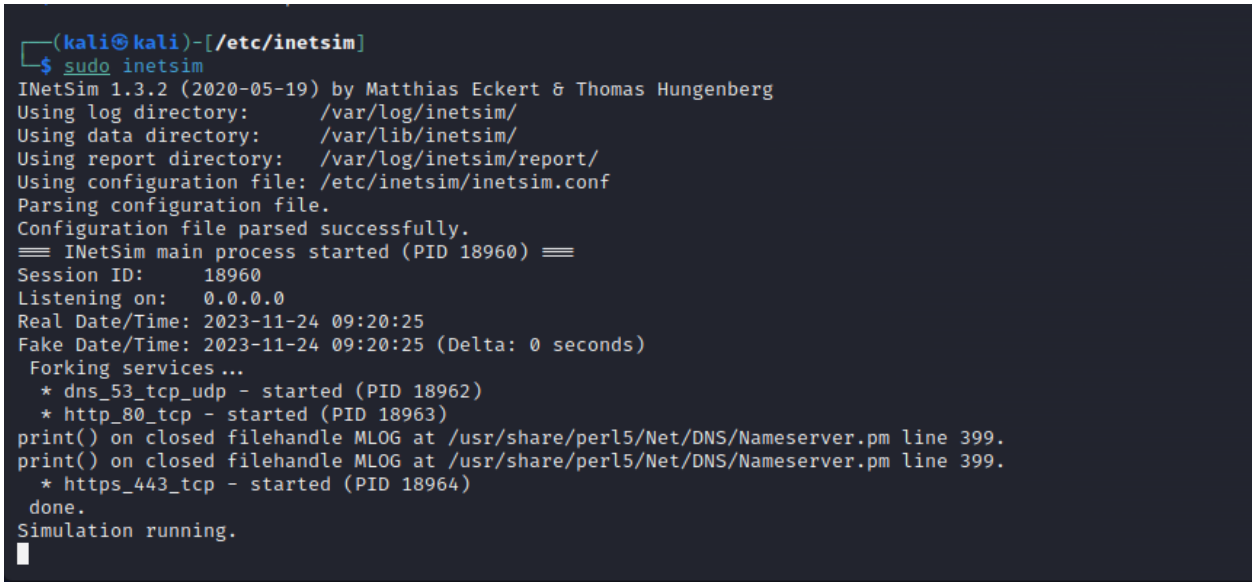
Una volta fatta questa operazione possiamo impostare il server bind address 0.0.0.0 (valido quindi per qualunque interfaccia)

```
##############################################
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 0.0.0.0
```

per poi configurare il DNS di default con l'IP di Kali Linux e il DNS statico che associ all'hostname epicode.internal l'IP 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_default_hostname
#
# Default hostname to return with DNS replies
#
# Syntax: dns_default_hostname <hostname>
#
# Default: www
#
#dns_default_hostname somehost


########################################
# dns_default_domainname
#
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
#dns_default_domainname some.domain


########################################
# 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

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

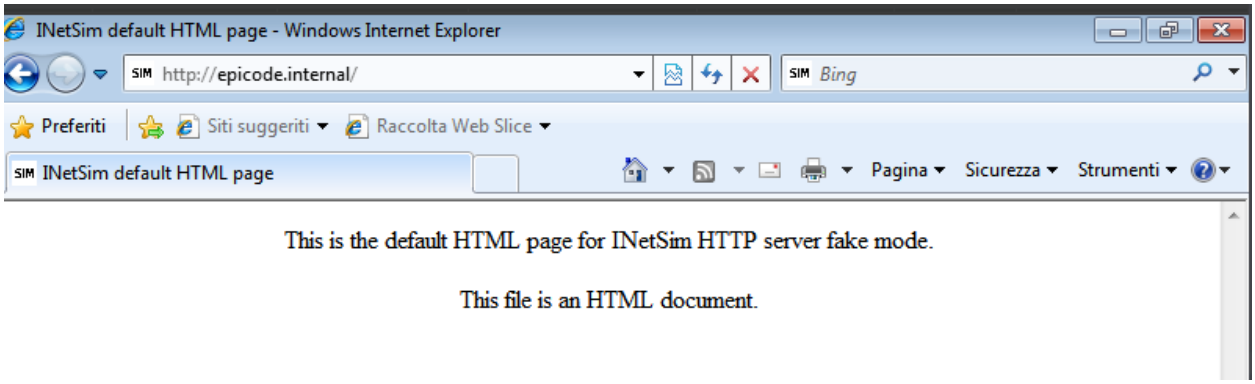
Dopodiché possiamo avviare una simulazione per assicurarci che la configurazione sia andata a buon fine:

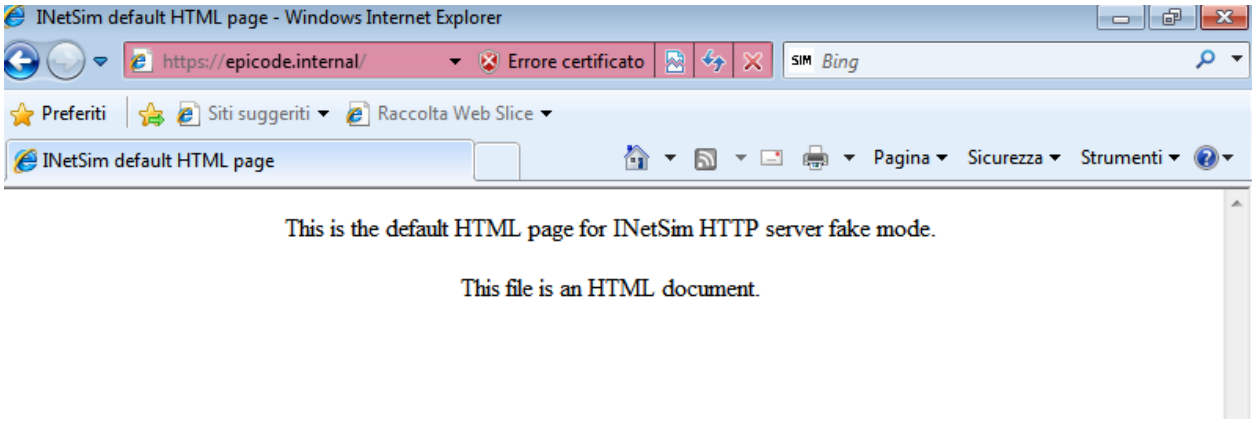


Da questa immagine possiamo vedere che i servizi dns, http e https si avviano correttamente.

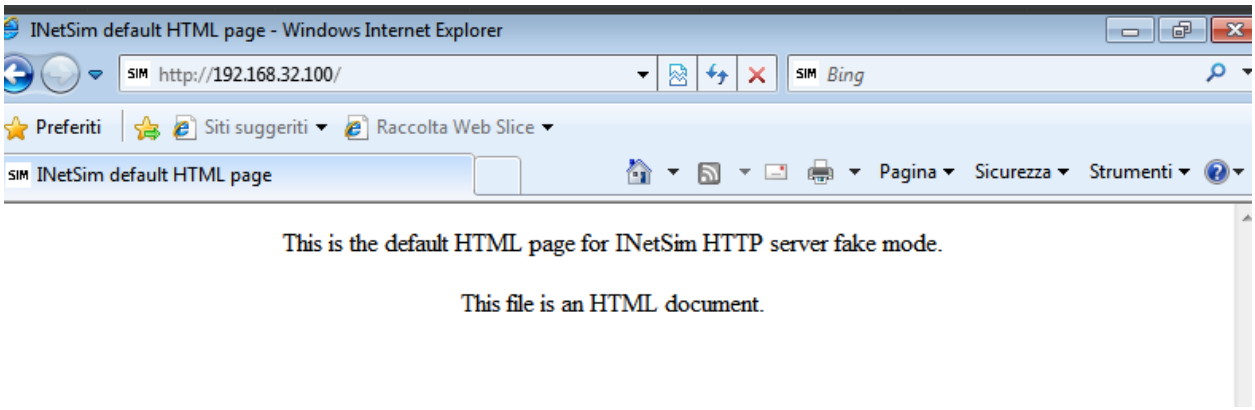
III. Richiesta HTTP e HTTPS

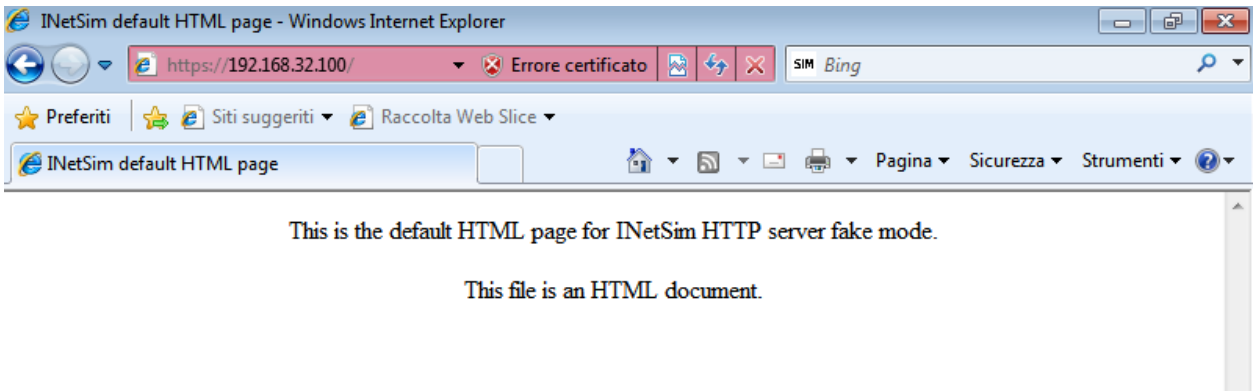
Adesso che tutto è stato configurato, è necessario verificare la buona riuscita della configurazione recandosi su Windows 7 e lanciando le richieste HTTP e HTTPS tramite hostname epicode.internal:





Verificare poi che le richieste vengano aperte correttamente anche con IP 192.168.32.100:



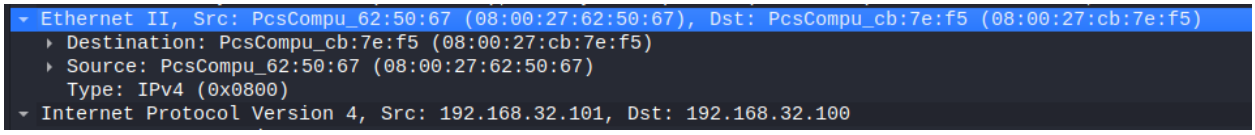


IV. Wireshark

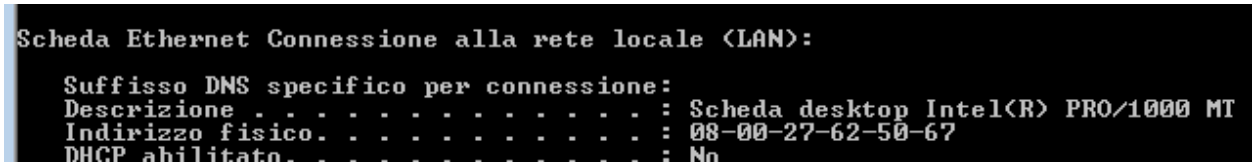
Wireshark è un network sniffer e packet analyzer che ci permette di analizzare qualsiasi pacchetto, flusso di traffico o connessione che passa dalla scheda di rete del computer.

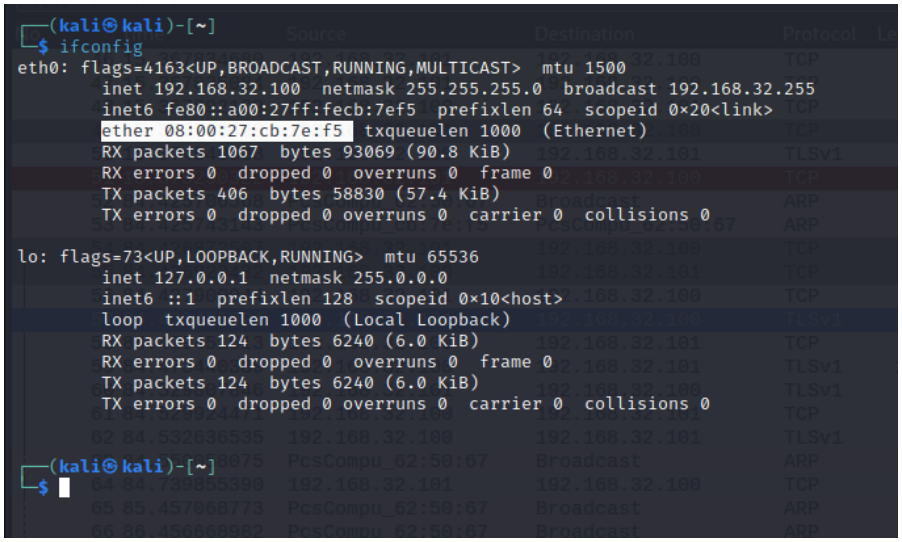
Apriamo Wireshark da Kali Linux, scegliamo la NIC eth0 e nel frattempo accediamo a "http://epicode.internal" da Windows 7. Vedremo che la scheda inizierà a popolarsi di pacchetti.

Tra i vari dati che ci mette a disposizione Wireshark, è possibile visualizzare sia per le richieste HTTP che per le HTTPS gli indirizzi IP di Windows 7 e Kali Linux così come i MAC address sorgente e destinazione:

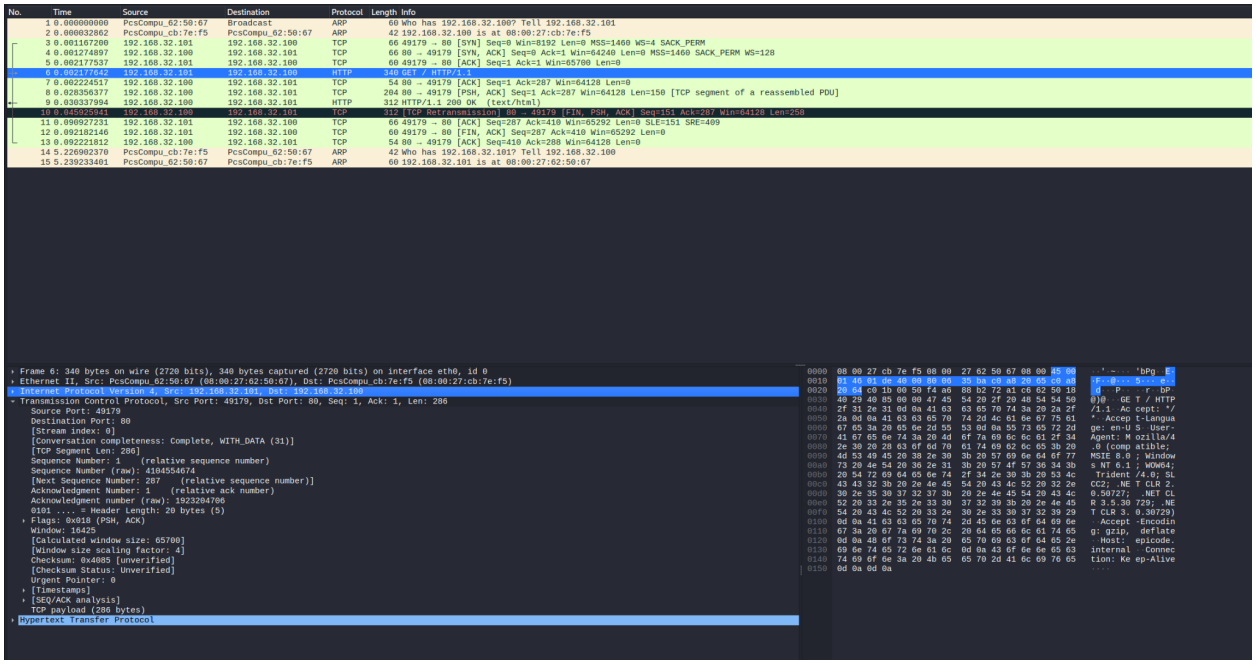


Per avere conferma che anche i MAC address corrispondano a quelli delle macchine virtuali è possibile aprire il terminale da Windows 7 e fare una verifica tramite il comando "ipconfig /all" e poi fare lo stesso da Kali Linux tramite comando "ifconfig":





Richiesta HTTP:

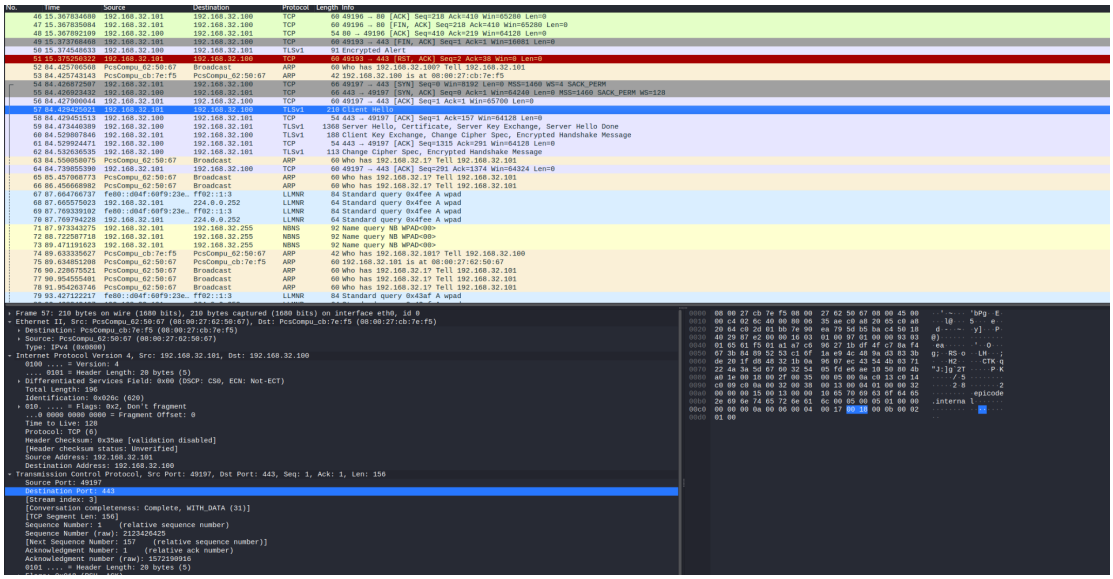


Da questa schermata è possibile vedere i vari protocolli. Trattandosi di una richiesta HTTP (traffico non protetto), possiamo notare che non è presente alcun protocollo SSL (Secure Sockets Layer) o TLS (Transport Layer Security) che garantiscano la trasmissione sicura dei dati.

Da qui possiamo notare la richiesta effettuata (GET) inviata dal Windows7 (src 192.168.32.101) e la risposta (200 OK) che il server (src 1902.168.32.100) manda al Windows.

Possiamo notare che la “destination port” che è la 80, porta utilizzata per il traffico HTTP.

Richiesta HTTPS:



Trattandosi qui di una richiesta HTTPS, prima del trasferimento dei dati viene stabilita una connessione sicura e crittografata, com'è possibile notare già dal riquadro nero a destra dell'immagine.

Vediamo inoltre che la destination port è la 443, porta standard per il traffico HTTPS.

A differenza della richiesta HTTP, vediamo che qui è presente il protocollo TLSv1, si può notare infatti che è stata avviata una comunicazione mirata alla protezione dei dati. Inoltre, essendo stata stabilita una connessione sicura, non è visibile qui la richiesta GET né la risposta con codice di stato.