W4D4

PROVA DI FINE MODULO

Traccia e requisiti

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora.
Lo studente verrà valutato sulla base della risoluzione al problema seguente.

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:

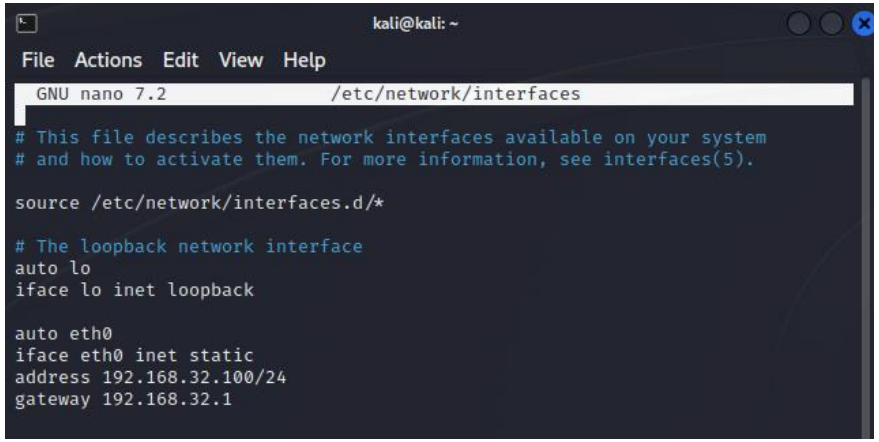
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 (Kali).

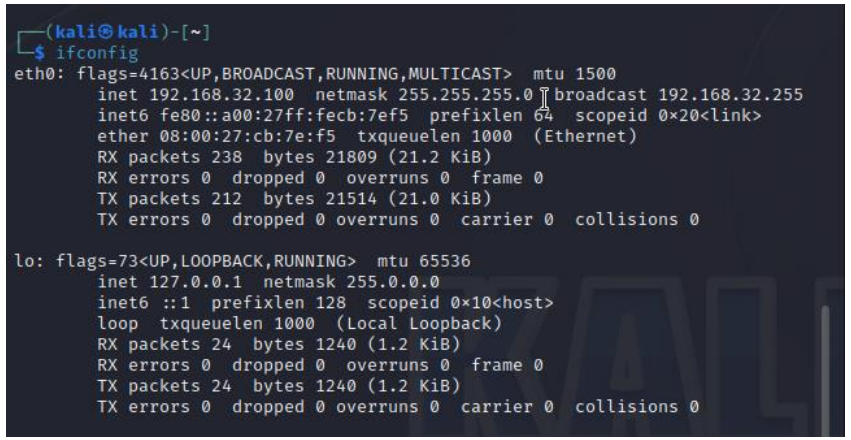
Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

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.

Setting del laboratorio (Kali)

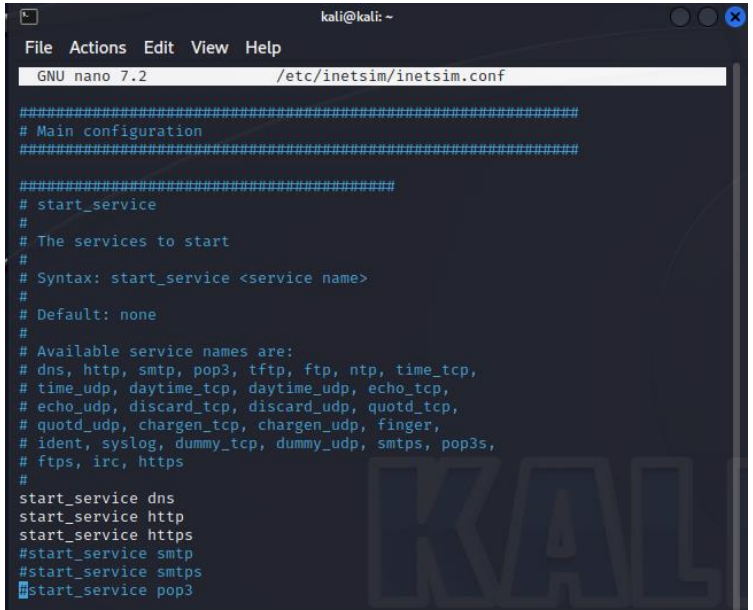
Innanzitutto prepariamo il laboratorio in modo da essere in linea con i requisiti richiesti, per cui nel file di configurazione «interfaces» di Kali impostiamo l'indirizzo IP di eth0 e il gateway (come da requisiti indicati) e verifichiamo la configurazione con il comando «ifconfig».
Per Kali intenderemo la macchina virtuale dove è installato l'OS Kali Linux e per Win7 la macchina virtuale dove è installato l'OS Windows 7.

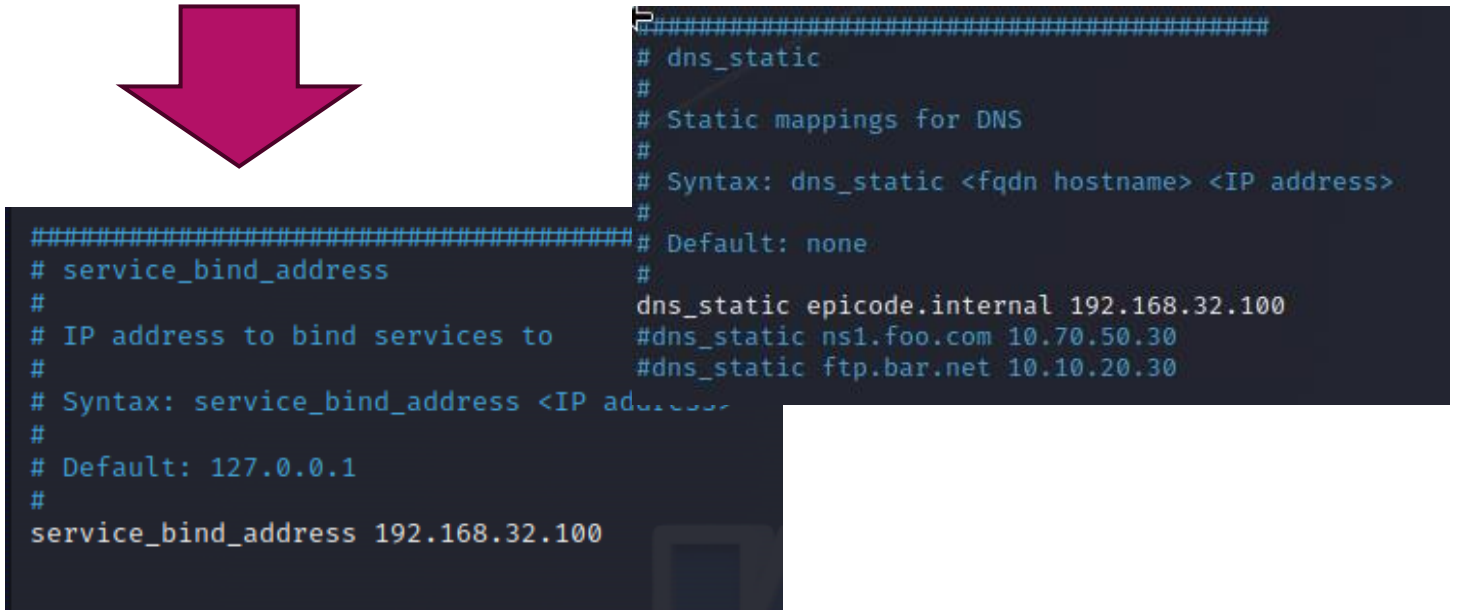




Setting del laboratorio (Kali)

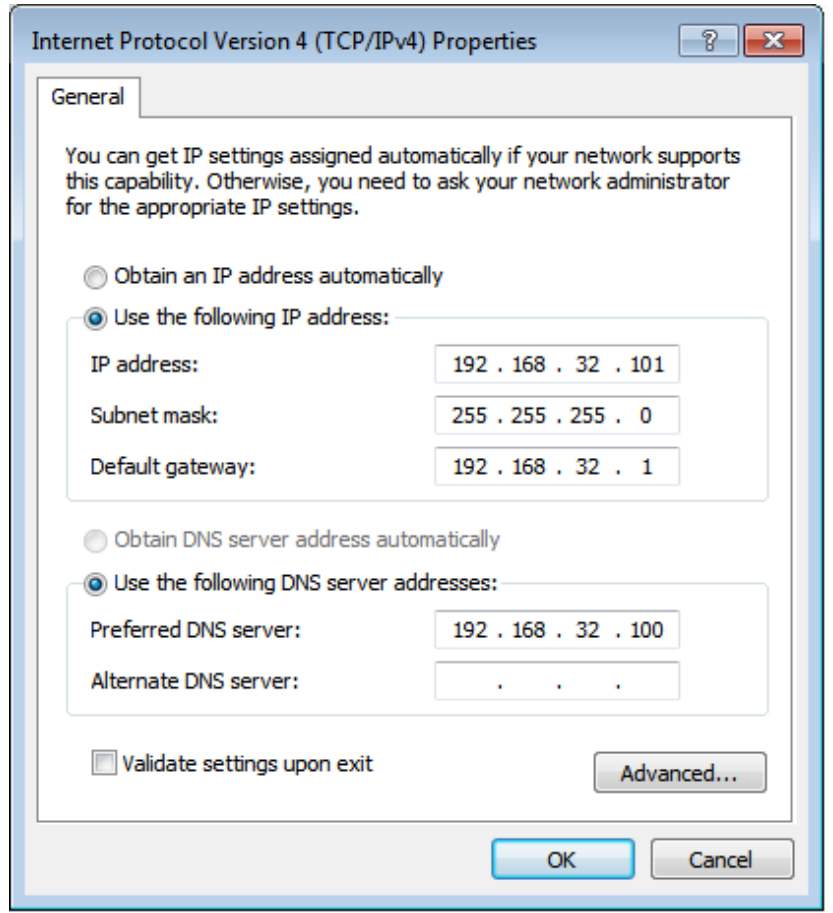
Successivamente modifichiamo il file di configurazione di InetSim su Kali che ci permetterà di attivare i servizi HTTP, HTTPS e DNS su tale server simulato.
Inoltre imposteremo il bind address, cioè collegheremo i servizi del server simulato all'indirizzo IP di Kali facendo si che quest'ultimo agisca da Server.
I protocolli HTTP e HTTPS permetteranno al server di restituire delle risorse, il DNS di risolvere un hostname e collegarlo ad un indirizzo IP, infatti sul file di configurazione associo all'hostname «epicode.internal» l'indirizzo IP di Kali.





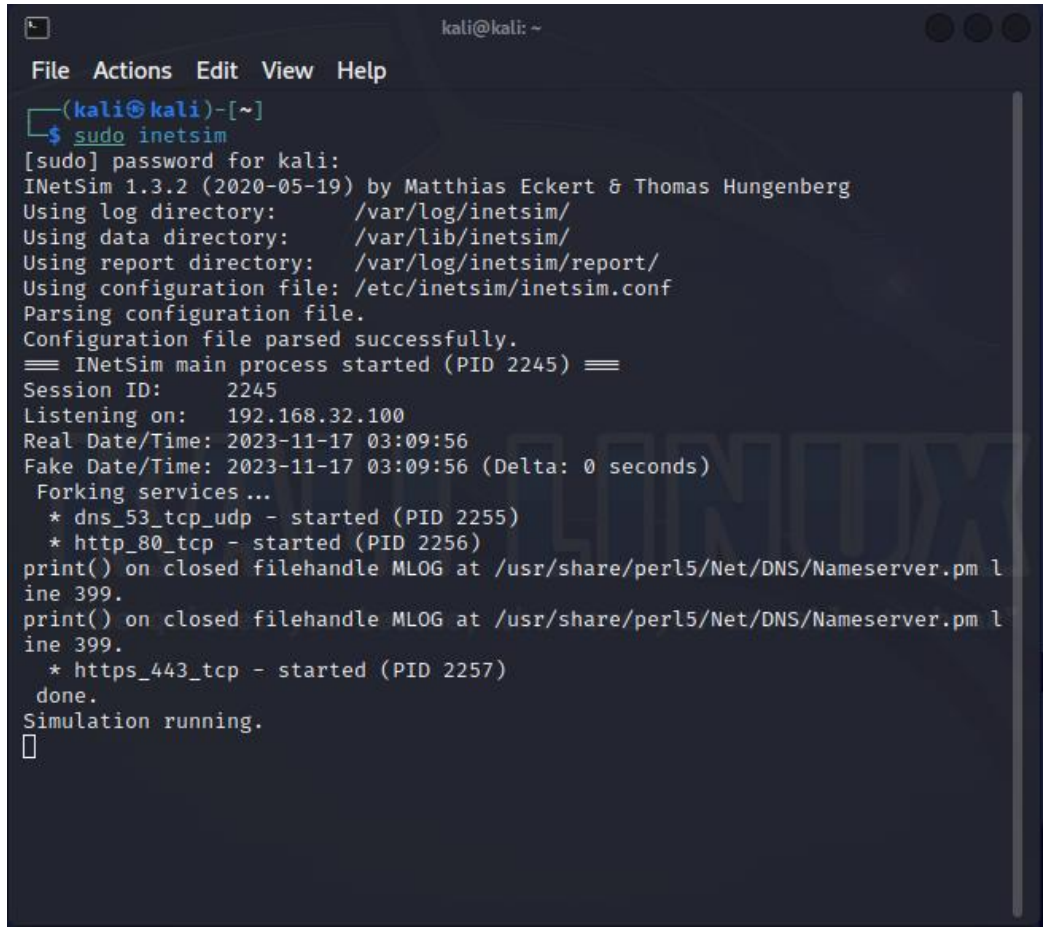
Setting del laboratorio (Windows 7)

Configuro anche la macchina virtuale di Windows 7 reimpostando l'indirizzo IP come da requisiti e indicando l'indirizzo del DNS (prima configurato su Kali) che mi permetterà di risolvere il nome del dominio utilizzando appunto il servizio DNS attivato su Kali tramite InetSim.
Win7 agirà da Client.



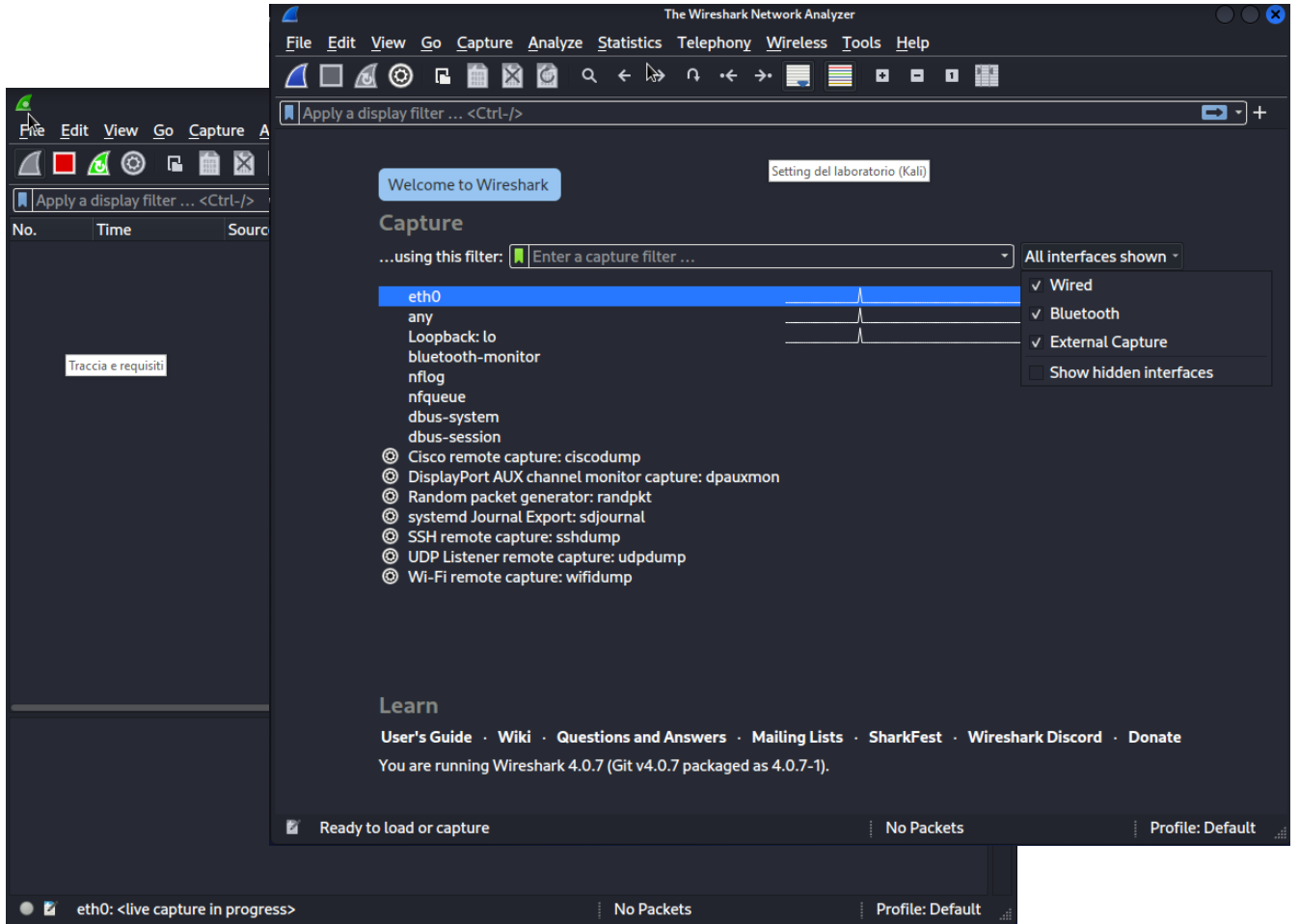
Simulazione richiesta Http e Https

Avendo attivato e collegato i servizi Server di InetSim all'indirizzo IP di Kali possiamo avviare InetSim da terminale in modo da permettere al Server di «ascoltare» le richieste da parte del Client.



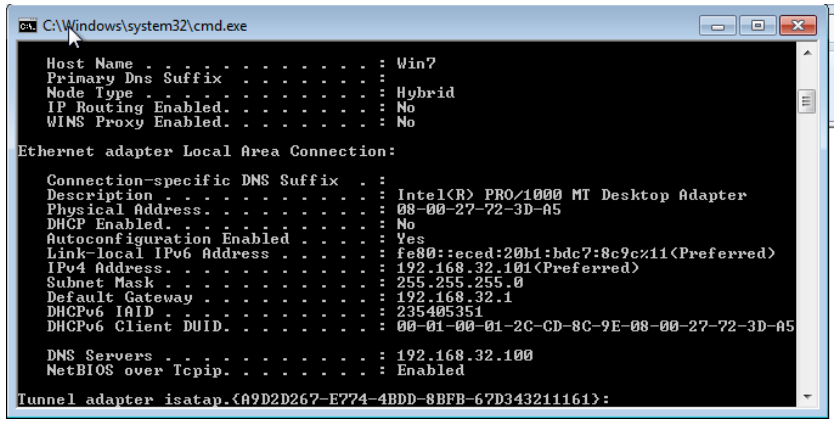
Simulazione richiesta Http e Https

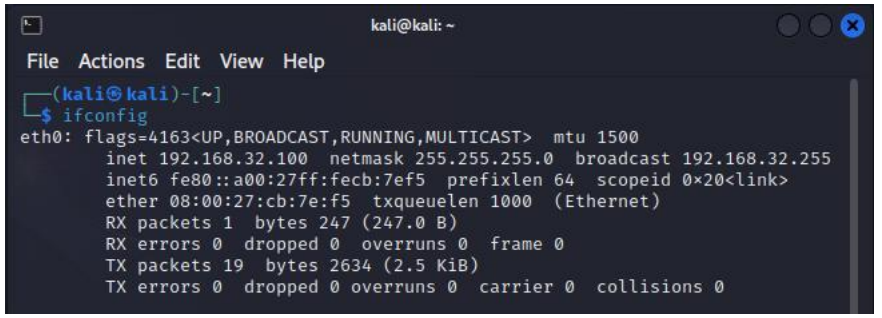
I pacchetti partiranno da Win7 e arriveranno a Kali(e viceversa), ciò vuol dire che passeranno per la eth0 di Kali per cui se su Wireshark settiamo eth0 vedremo il traffico che passerà per tale interfaccia sia in entrata che in uscita.



Simulazione richiesta Http e Https

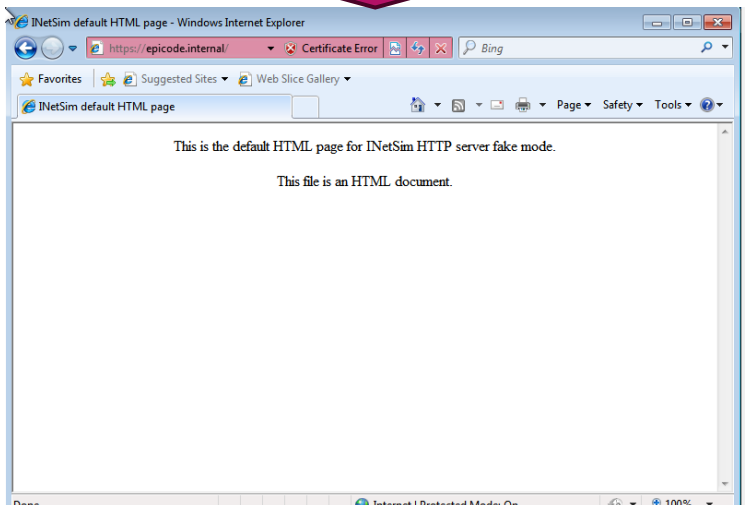
Il primo obiettivo è quello di evidenziare i MAC address di sorgente e destinazione sulle rispettive VM per poi verificarne la corrispondenza nei pacchetti intercettati con Wireshark.
Su Kali per vedere il Mac digiteremo «ifconfig» mentre su Win7 digiteremo «ipconfig».
Il MAC address di Kali è 08:00:27:cb:7e:f5 , quello di Win7 è 08:00:27:72:3d:ab .





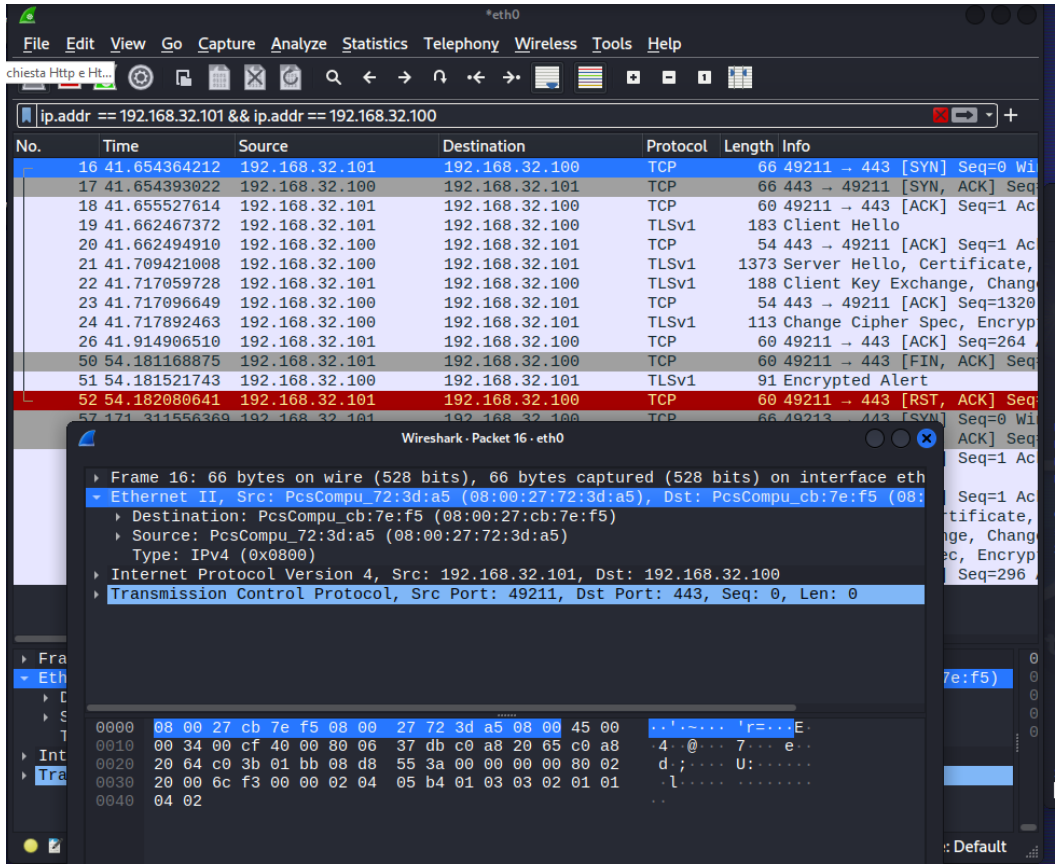
Simulazione richiesta Http e Https

Apro il browser «Internet Explorer» e nella barra degli indirizzi prima digito l'hostname «epicode.internal» usando il protocollo HTTPs e verifico se funziona la risoluzione dell'hostname «epicode.internal» grazie al servizio DNS attivato sul Server. Nel frattempo verifico il traffico generato su Wireshark isolando i risultati relativi agli Ip delle due VM.
La risorsa presente sul server viene visualizzata correttamente, questo vuol dire la risoluzione dell'hostname è avvenuta correttamente è che grazie al servizio HTTPs è stata fornita la risorsa.



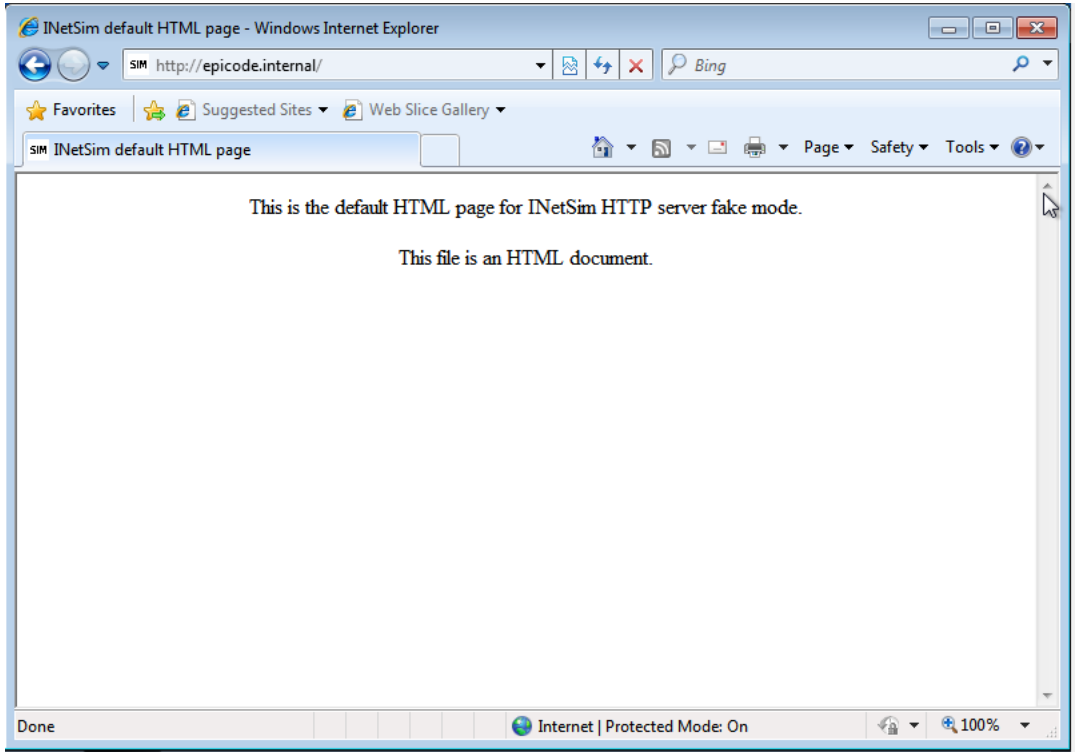
Simulazione richiesta Http e Https

Digito il filtro «ip.addr == 192.168.32.101 && ip.addr == 192.168.32.100» su Wireshark per isolare la visione del traffico tra le sole due VM(Kali e Win7) interessate. Tramite browser su Win7 mi connetto a «epicode.internal» osservo la generazione di pacchetti per stabilire la connessione tra le due VM secondo il processo di «three-way-handshake».
Nella prima delle tre fasi di quest'ultima osserviamo la richiesta SYN da parte di Win7 (evidenziata nella figura accanto): abbiamo detto che è Win7 a fare la richiesta per cui il source MAC dovrebbe corrispondere a quello della scheda di rete di tale macchina, infatti è 08:00:27:72:3d:ab . Il destination MAC address deve corrispondere a quello di Kali e infatti è 08:00:27:cb:7e:f5.



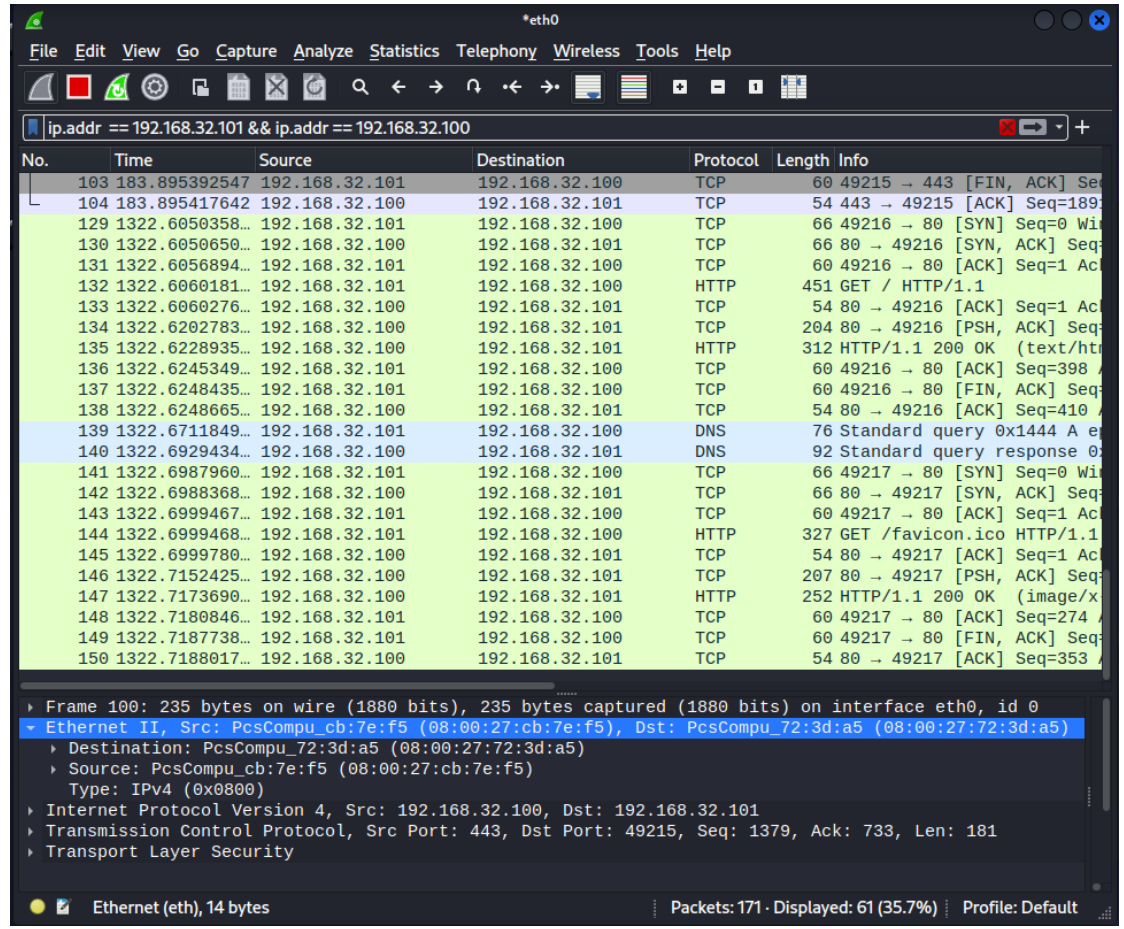
Simulazione richiesta Http e Https

Ritorno su Win7 e questa volta la richiesta verrà effettuata tramite il protocollo HTTP.
La risoluzione dell'hostname avviene con successo.
Vediamo cosa succede su Wireshark.



Simulazione richiesta Http e Https

Anche qui c'è il processo di three-way-handshake.
Nulla varia per quanto riguarda la corrispondenza dei MAC address.
Qui possiamo visualizzare anche il protocollo DNS in azione, il Server leggerà l'hostname e grazie a questo servizio fornirà il corrispondente indirizzo IP.
La differenza più importante però è che qui a differenza dell'HTTPs abbiamo la possibilità di vedere il contenuto della richiesta e della risorsa trasmessa in chiaro.



Simulazione richiesta Http e Https: differenze tra i due protocolli e conclusioni

Nell'immagine di fianco possiamo infatti osservare che cliccando sul pacchetto possiamo visionare il contenuto della pagina restituita.
Nel protocollo HTTPs infatti interviene nel processo di handshake il protocollo TLS che prevede lo scambio della chiave tra Client e Server per cifrare i contenuti, nella richiesta HTTP ciò non avviene.
Altra differenza è che l'HTTP usa la porta 80 mentre l'HTTPs usa la 443.
In sostanza col protocollo HTTPs su Wireshark vedremo solo connessione stabilita tramite il protocollo TCP e TLS mentre in quella HTTP anche il protocollo DNS e soprattutto il contenuto della richiesta.