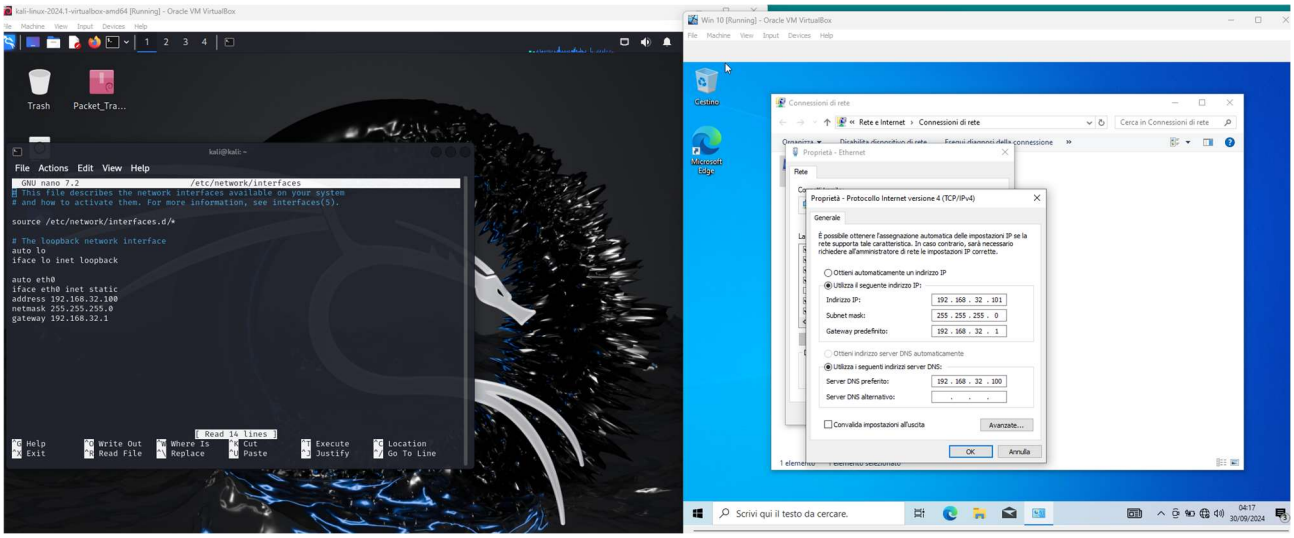
In questa esercitazione, metteremo in comunicazione efficacemente due macchine virtuali, Kali Linux e Windows 10 Home, utilizzando l'architettura client server. Utilizzando il tool software inetsim, saremo in grado di simulare di richiedere una risorsa web che risponde all'indirizzo Ip di Kali, dopodichè, utilizzeremo Wireshark per intercettare il traffico, evidenziare gli indirizzi Mac delle due macchine virtuali e spiegare le eventuali differenze se riscontrate, motivandole.

1) Configurazione degli Ip ed avvio dell'indirizzo "epicode.internal" dal browser Edge di Windows 10.

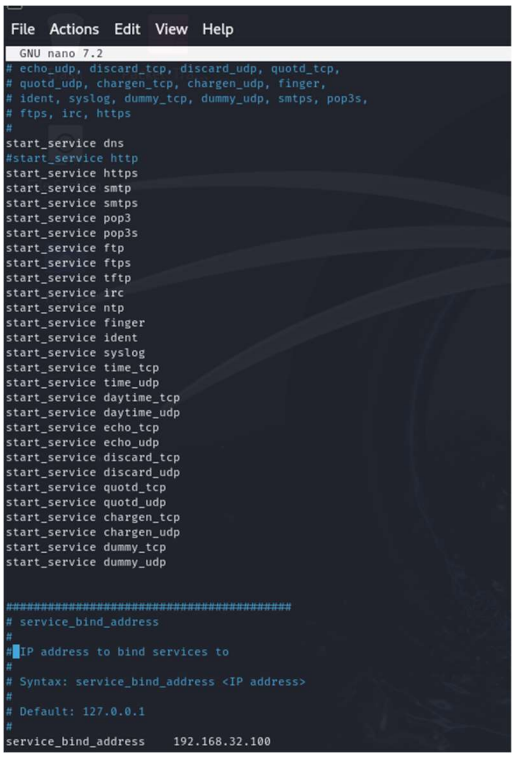
Primo step, impostare gli indirizzi IP statici alle macchine

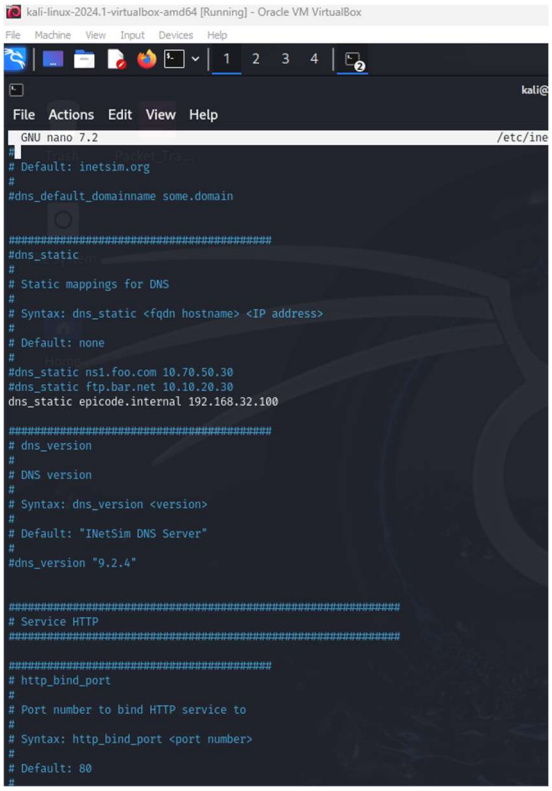


Secondo Step, configurazione inetsim.

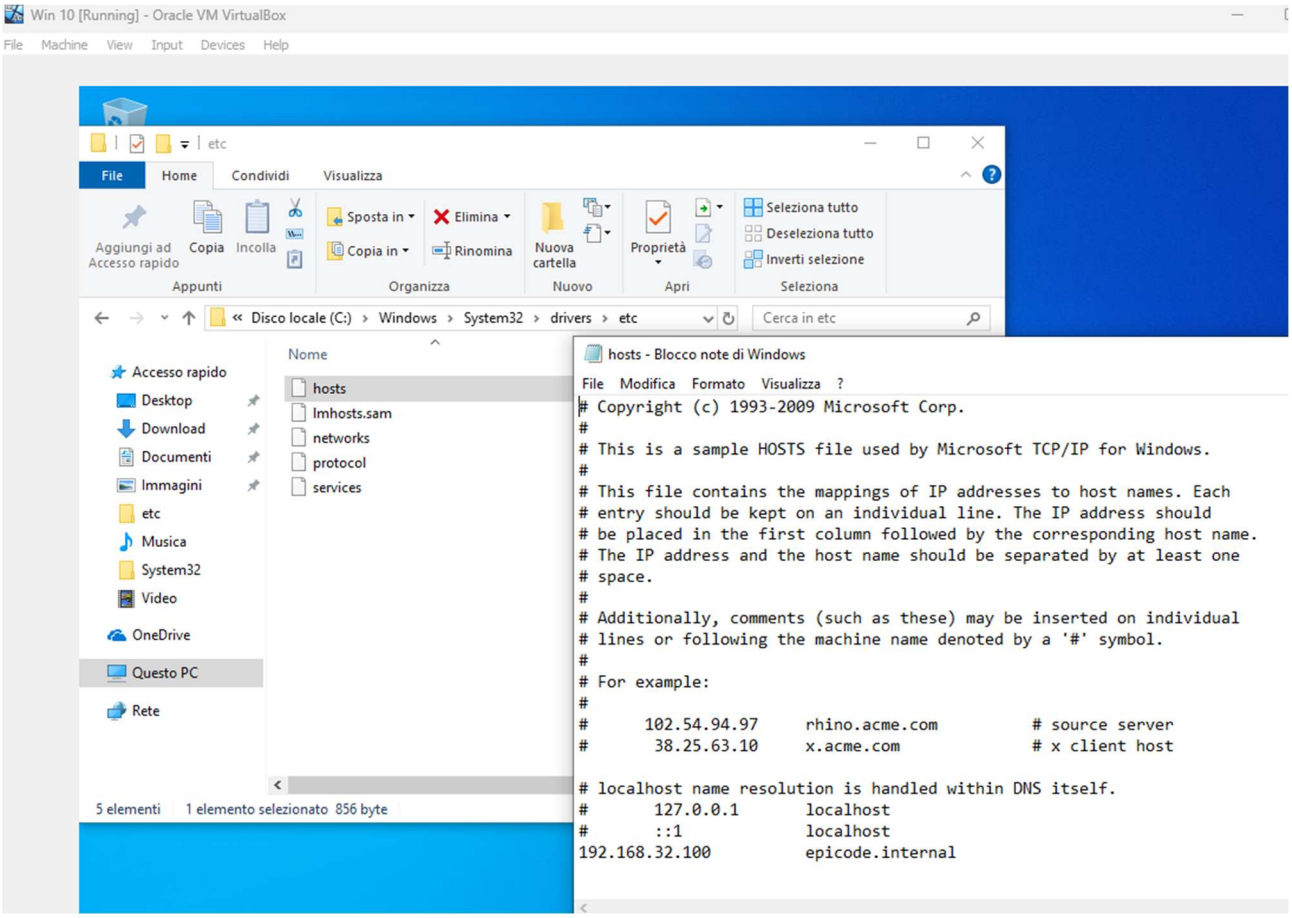
Ho configurato, procedendo a commentare il servizio http per il momento.

Ho impostato il bind address a 192.168.32.100, per stabilire la comunicazione, poi, ho proceduto con l'impostazione del dns_static epicode.internal 192.168.32.100, in modo da comunicare il dns statico con relativo indirizzo "epicode.internal" ad entrambe le macchine.



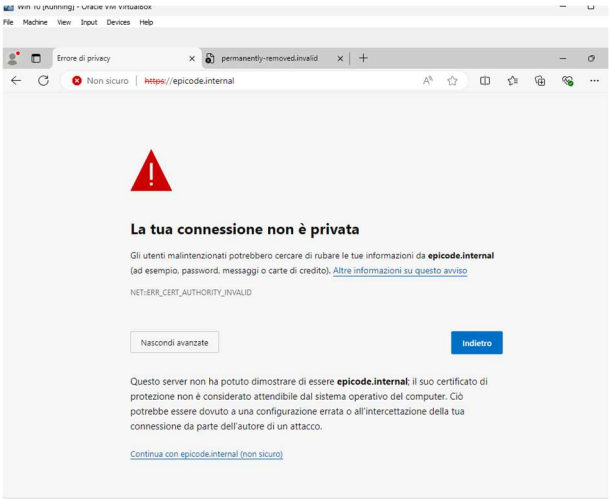


In Windows 10, dopo aver impostato la policy nel firewall in uscita, ho impostato nelle impostazioni del DNS, il nome ”epicode.internal”, raggiungibile all’indirizzo IP di Kali (modello client-server):

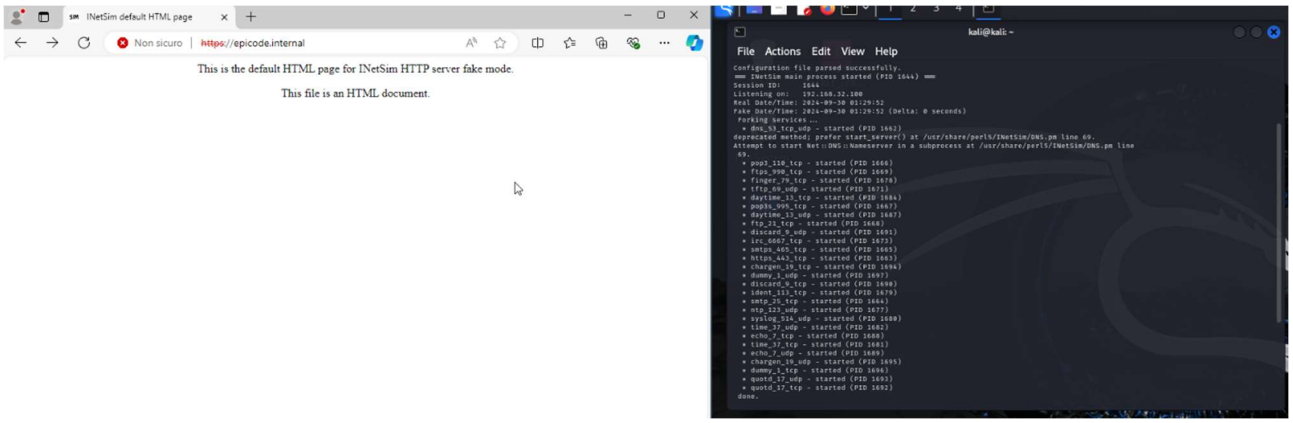


Adesso come fase successiva, l'avvio di inetsim con servizio https:

Il sistema operativo Windows 10, risponde immediatamente con un'errore di certificato, non riconosce infatti il certificato di protezione offerto dal sito; ciò ci fa subito pensare al fatto che HTTPS è un protocollo sviluppato per offrire sicurezza nella comunicazione e ciò è un bene; infatti nel caso in cui un malintenzionato sia in ascolto sulla stessa porta per fare un esempio, noi saremo in grado di agire tempestivamente.

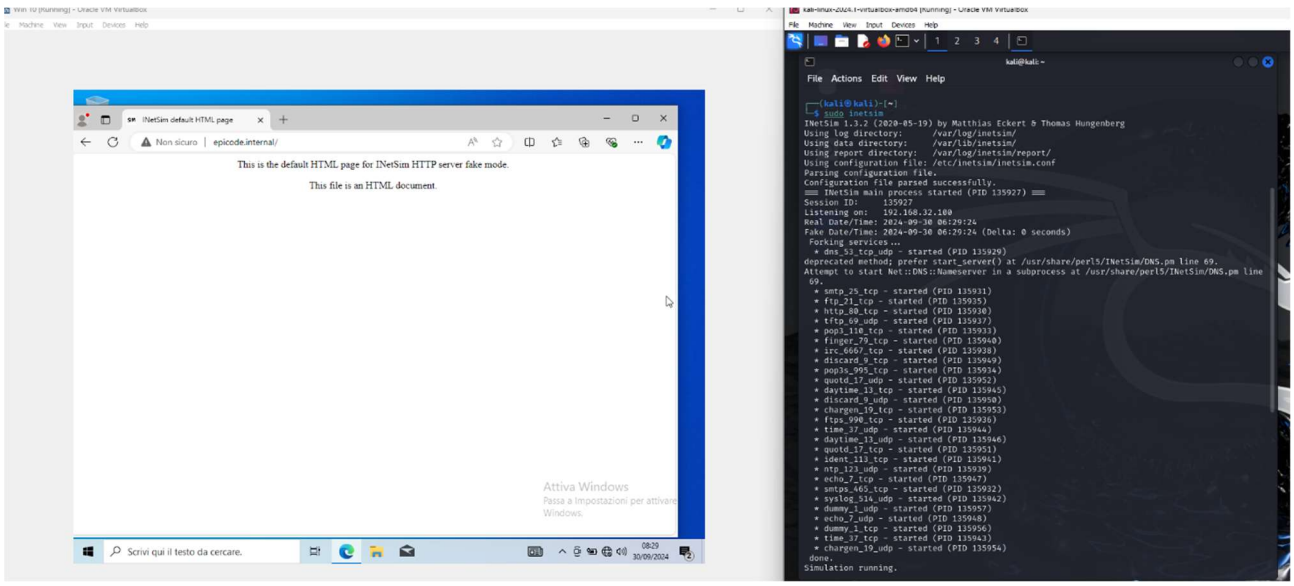


1)



(L'esecuzione della risorsa "https://epicode.internal/" da Windows 10; il servizio inetsim in esecuzione su kali linux.)

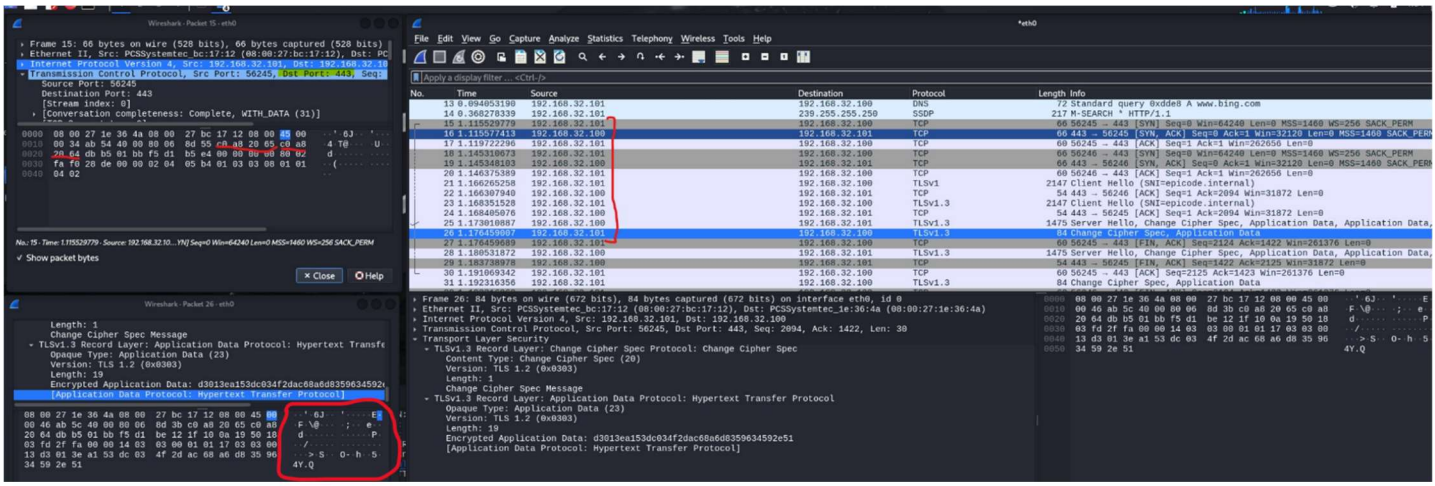
2)



(L'esecuzione della risorsa "http://epicode.internal/" da Windows 10; il servizio inetsim in esecuzione su kali linux.)

2) L’esercizio richiede di individuare i MAC di sorgente e destinazione ed il contenuto della richiesta HTTPS utilizzando Wireshark.

Avviando Wireshark, con risorsa HTTPS da browser Edge di Windows, elenco con il pennarello rosso, la trasmissione dei pacchetti, effettuata, per evidenziare la risorsa avviata.



Nello screenshot in questione, analizzo la trasmissione dall’avvio della richiesta da parte del browser Edge dal pacchetto 15 al pacchetto 26 (che ho segnato con la penna rossa) dove viene eseguito il protocollo di avvio HTTPS.

In alto a sinistra, evidenziata in giallo (443) la porta utilizzata, che ci rimanda come studiato nella teoria, alla porta HTTPS. Nel contenuto del pacchetto 15 evidenziato in alto a sinistra vediamo i Mac address:
1) c0 a8 20 65- Windows 10; 2) c0 a8 20 64 Kali Linux

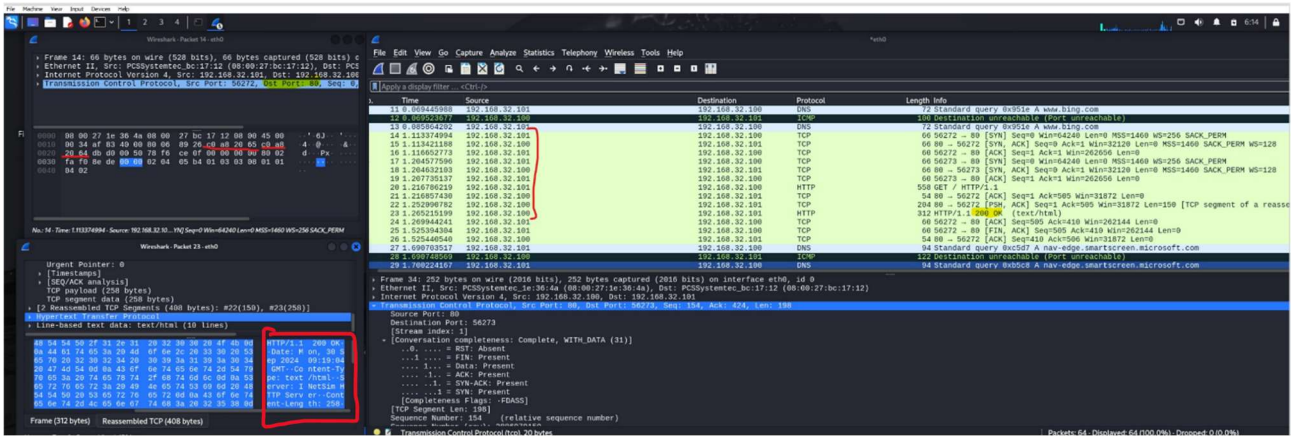
Il MAC è possibile vederlo nella descrizione centrale in basso, nella stringa **Ethernet 2**, che non ho evidenziato in questa cattura ma che ci mostra nel pacchetto 26 una descrizione più completa dei Mac Address in esadecimale, sia della sorgente che del destinatario.

In basso a sinistra, nella descrizione dettagliata del pacchetto 26, è evidenziato in blu l’avvio del pacchetto http.

La spiegazione del riquadro in rosso la farò più avanti quando si elencheranno le differenze.

Ora vedremo la cattura con il protocollo http.

2.1) L’esercizio richiede di individuare i MAC di sorgente e destinazione ed il contenuto della richiesta http utilizzando Wireshark.



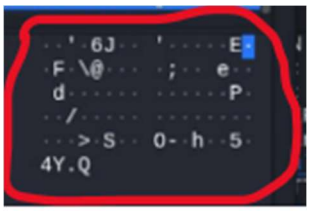
Come la precedente cattura, elenco con il pennarello dall’avvio della richiesta da parte del browser Edge di Windows, che avviene al pacchetto 13, per arrivare al pacchetto 23 dove viene eseguito il protocollo di avvio HTTP, evidenziati in rosso con la penna ed evidenziato in giallo dal codice di stato “200 ok” che implica che la richiesta ha avuto buon fine e che viene indicato anche nella descrizione dettagliata ed evidenziata in blu, dello stesso pacchetto in basso a sinistra.

In alto a sinistra, la descrizione dettagliata del pacchetto 14, ci mostra la porta utilizzata (80), che ci rimanda come studiato in teoria alla porta standard per l’applicativo http. Segue come per HTTPS prima, gli indirizzi Mac Address di Source (Windows 10) e destination (Kali Linux).

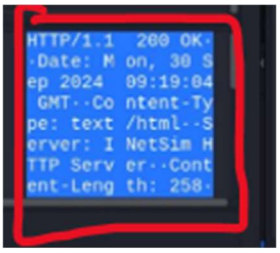
2) **Adesso l'esercizio ci richiede di evidenziare le differenze e motivarle spiegandole.**

Differenze riscontrate

1) **Chiarezza o visibilità del contenuto**;
   Nel pacchetto 26 prima e 23 qui sopra, guardando in basso a sinistra (..e riportati qui sotto per comodità), vediamo delineati da un riquadro rosso, il contenuto dei pacchetti:



Nel caso della richiesta HTTPS, il contenuto sembra crittografato.



Nel caso della richiesta http, il contenuto sembra abbastanza leggibile.

2) **Il numero di pacchetti**; HTTPS impiega più pacchetti rispetto ad http;

3) **Il tipo di protocolli usati;** Nella striga dei vari pacchetti vediamo che quando si avvia il protocollo http, esso è **visibilmente** indicato (Https utilizza TLS v. 1.3);

4) **Il protocollo di sicurezza TLS**; tale protocollo è tipicamente usato in connessioni cifrate e ci permette di utilizzare una maschera o security handshake al canale in chiaro impostato dal TCP (three-way handshake).

Grazie,

Federico Presti