Consegna Benchmark – week 4 day 4 – Daniele Rufo

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:

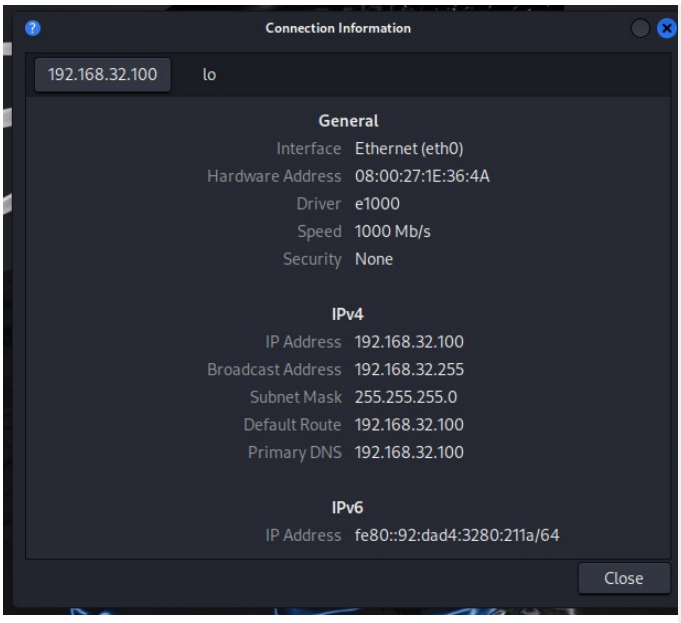
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richiede tramite web browser una risorsa all'hostname **epicode.internal** che risponde all'indirizzo 192.168.32.100 (Kali).

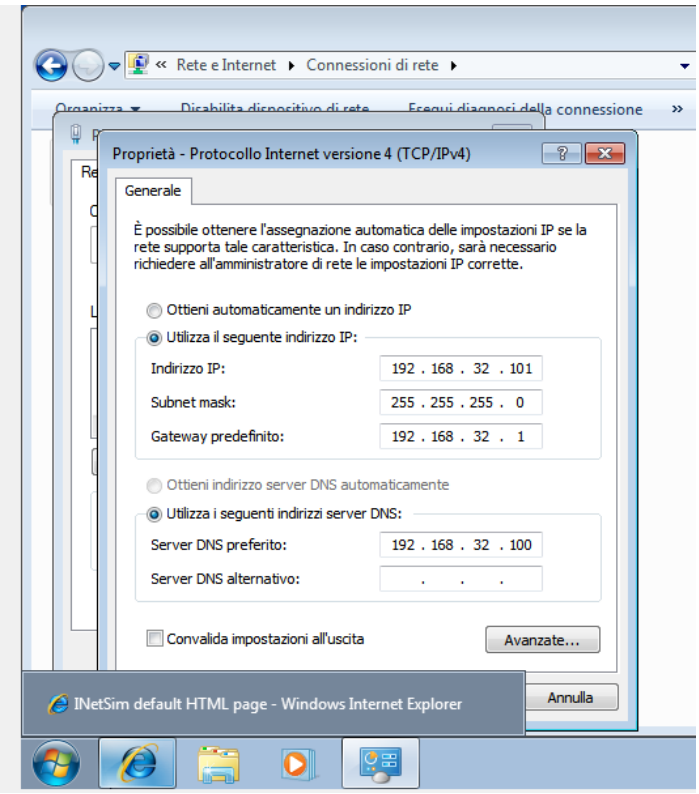
Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

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.

2

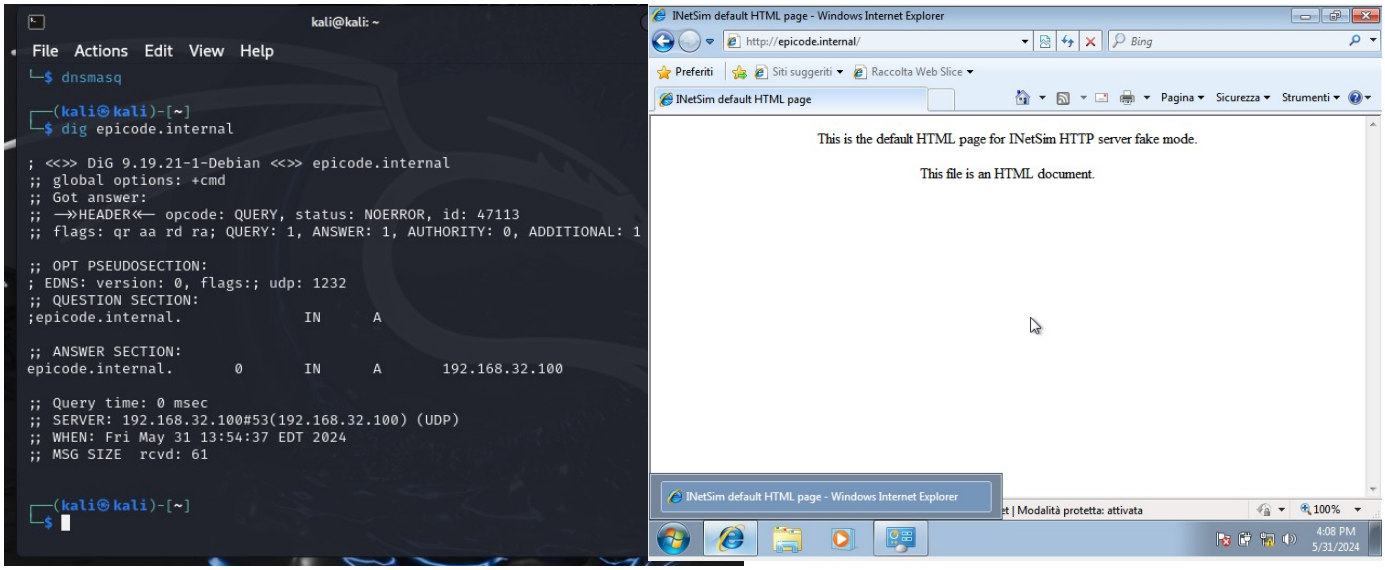
Configurazione ambiente richiesto:



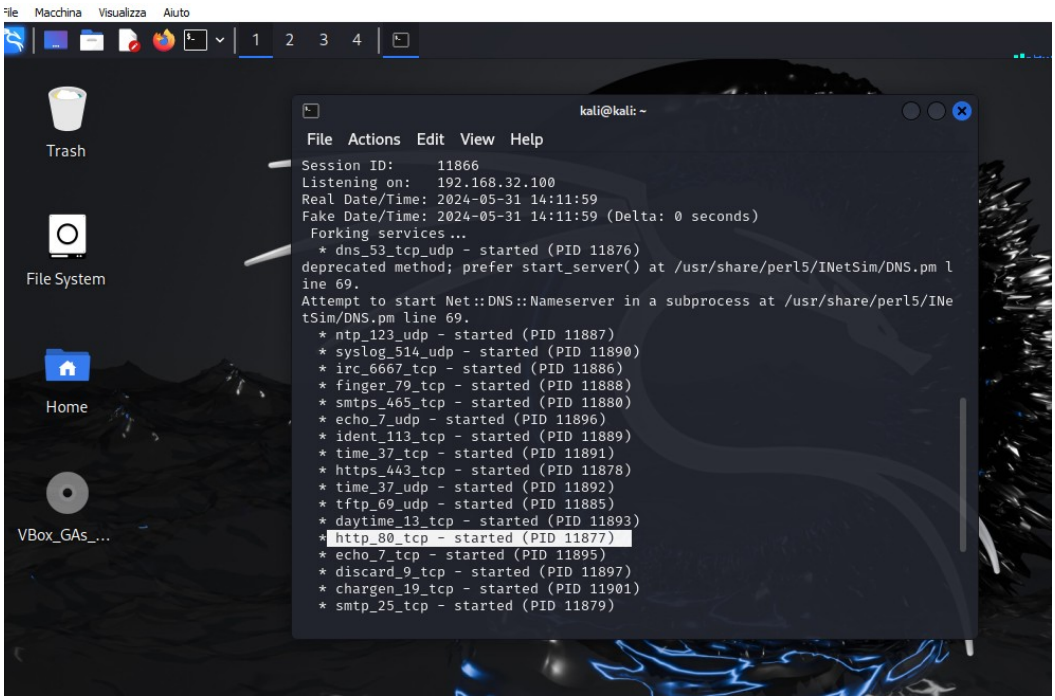


Gli IP sono corretti e le macchine sono connesse alla stessa rete interna. Inoltre si vede già che su Windows ho impostato l'IP del server kali come DNS.

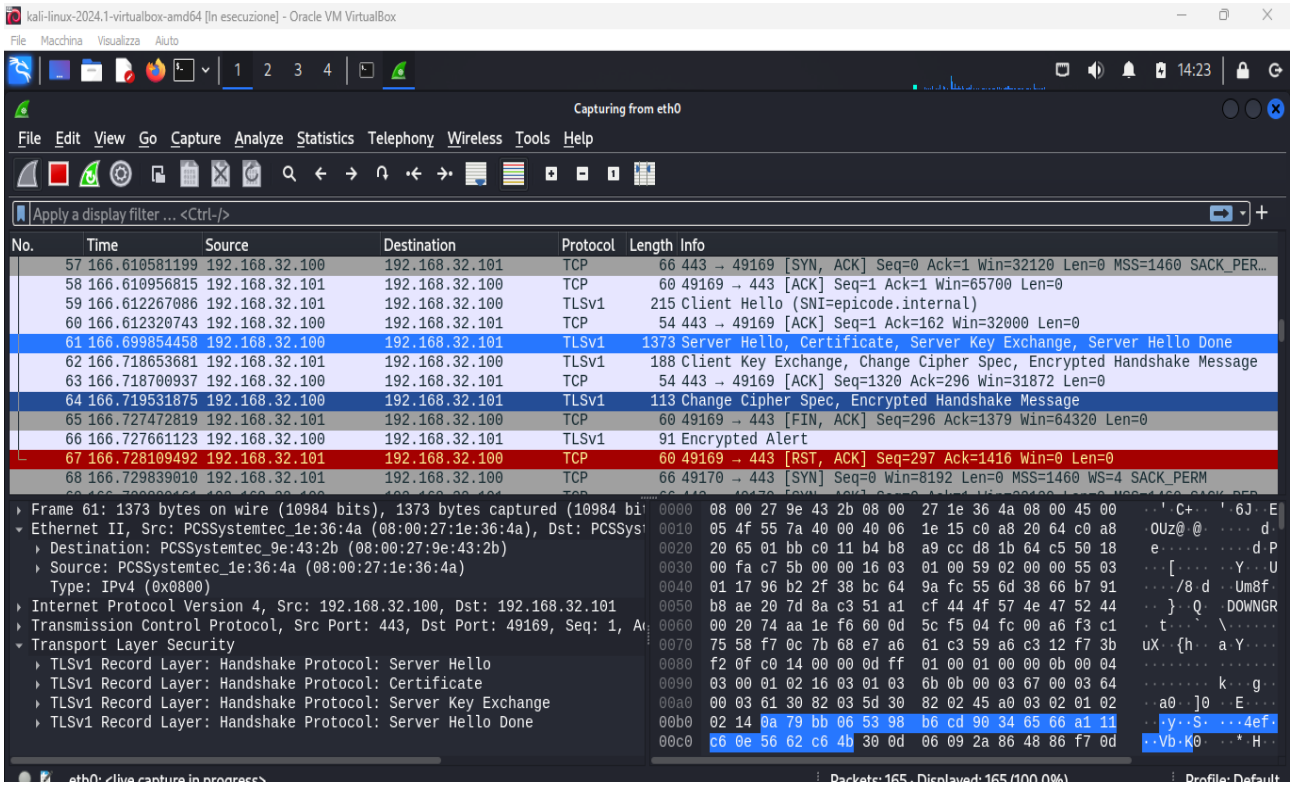
Per simulare l’architettura richiesta ho utilizzato DNSMASQ.
Nel fare ciò ho creato una cartella che ho editato con nano per configurare la scheda di rete e collegato epicode.internal all’IP 192.168.32.100.
In questo modo l’hostname indicato verrà risolto con l’IP della macchina kali.
Prove effettuate dal terminale di kali con comando “dig <hostname>” e dal web browser di windows 7.



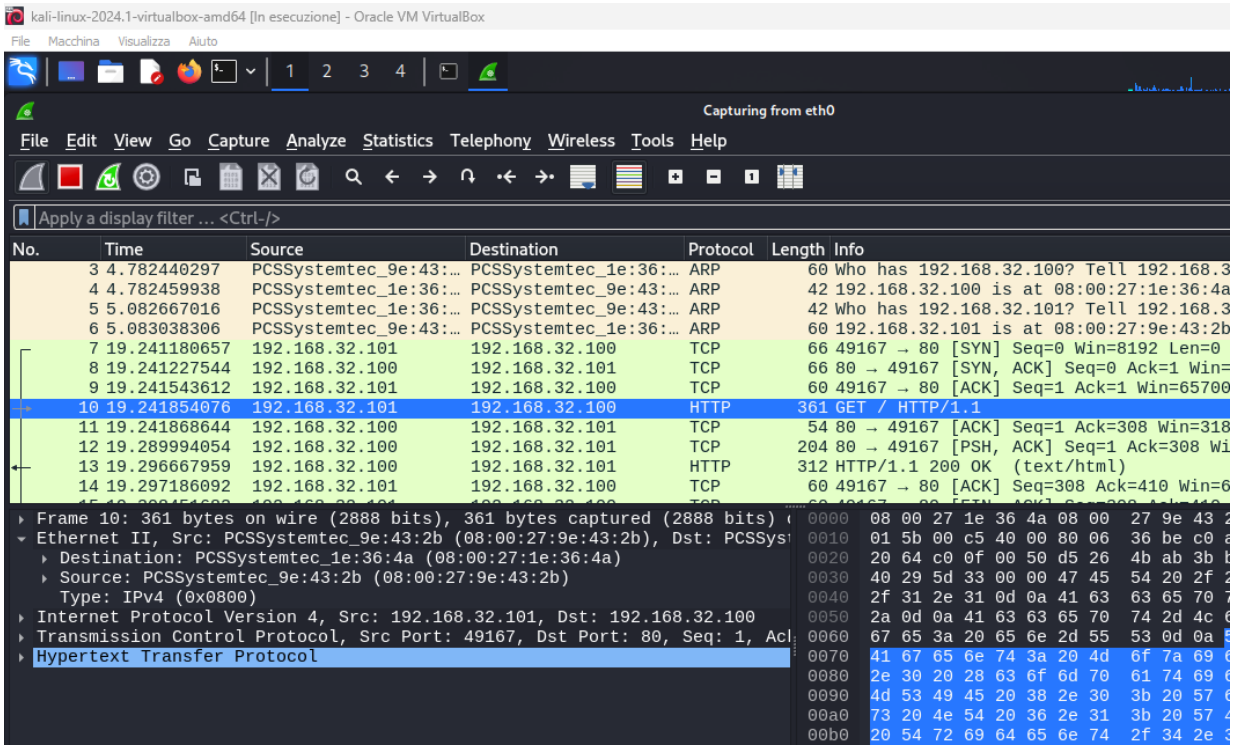
Adesso utilizzando inetsim da kali simuliamo i servizi internet.
Con privilegi di amministratore e utilizzando un editor si può modificare la configurazione “sudo nano /etc/inetsim/inetsim.conf” così ho usato “service_bind_address 192.168.32.100” per collegare i servizi di inetsim alla macchina kali.
Nella figura sotto si vede come inetsim abbia simulato protocolli http e https.
Da notare anche l’errore segnalato riguardante il dns, va in conflitto con la distro di kali forse. In realtà non ha dato problemi nello svolgimento dell’esercizio però per sicurezza in seguito ho editato il file asteriscando il servizio DNS di inetsim così da trasformarlo in semplice nota e quindi disattivarlo.



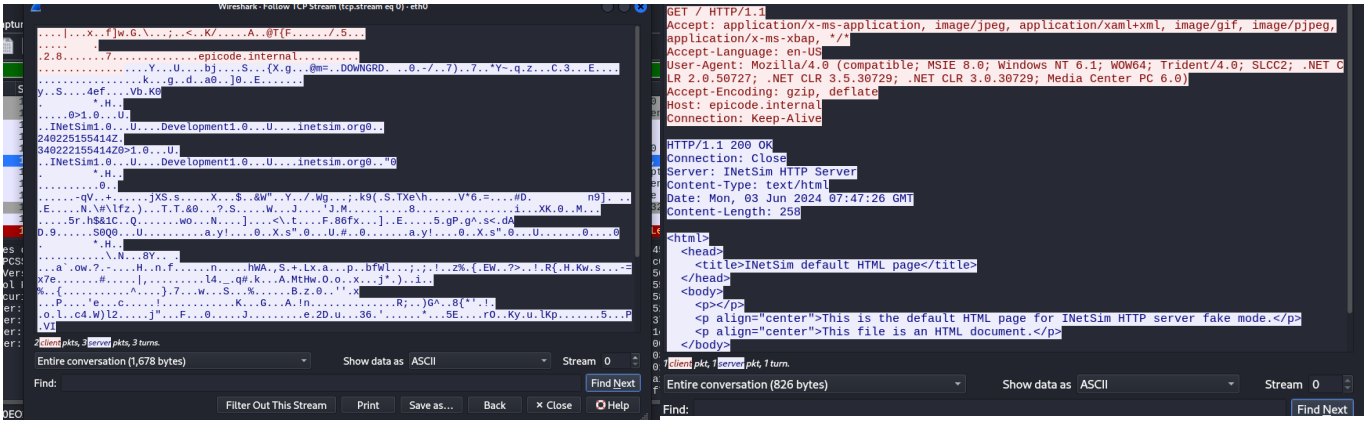
Utilizzando wireshark abbiamo catturato i pacchetti. Lo screen sotto è relativo al protocollo https. Abbiamo info sugli indirizzi MAC sorgente e destinazione (sono i MAC della macchina kali e windows 7) nelle informazioni riguardanti il livello 2 (iso/osi) del pacchetto (nello screenshot sono visibili nella parte in basso a sinistra sotto “Ethernet 2”.
Importante è sottolineare come usando il protocollo https la comunicazione sia cifrata grazie all’utilizzo di certificati TLS e rende impossibile avere accesso a tutte le info dei pacchetti catturati.



La situazione cambia se usiamo invece un protocollo http che è meno sicuro.



In questo caso abbiamo accesso a molte più informazioni che erano invece criptate dal protocollo https. I prossimi due screenshot sono relativi al flusso di due singoli pacchetti catturati (http e https).



Evidentissimo come quello di sinistra faccia riferimento all'https poiché è cifrato quindi i dati non sono leggibili. Nello screenshot di destra invece il flusso fa riferimento al pacchetto con traffico http. Quest'ultimo non è sicuro e abbiamo la possibilità di vedere sia la REQUEST completa del client (windows 7 nel nostro caso) con l'http verb (GET), user-agent (explorer), hostname (epicode.internal), l'accept (che indica la tipologia di documento che il client si aspetta di ricevere e la lingua) che la RESPONSE da parte del server con la riga di stato (versione protocollo 1.1 e il codice di stato 200 che indica che la risorsa è stata trovata), data, più info sul server (inetsim) e lunghezza del contenuto.
Specialmente le informazioni sul server nella response e sul user-agent nella request sono importanti per capire che tipo di comunicazione sta avvenendo e con quali strumenti.
Ciò può risultare un gran vantaggio per riconoscere e sfruttare le debolezze delle infrastrutture da attaccare. Questa è quindi la differenza più importante che riscontro tra i due protocolli.

Grazie.
Daniele Rufo