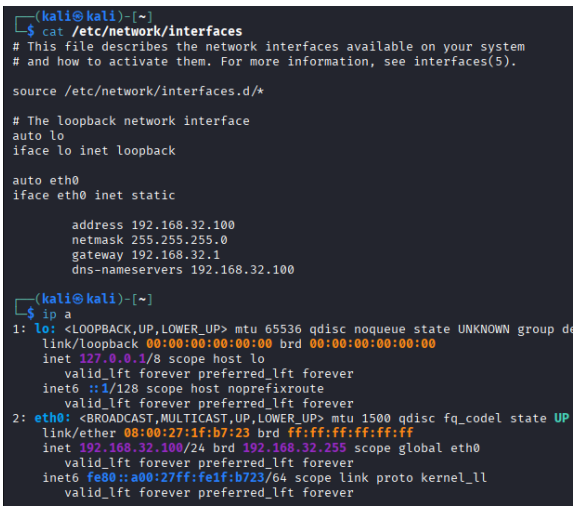
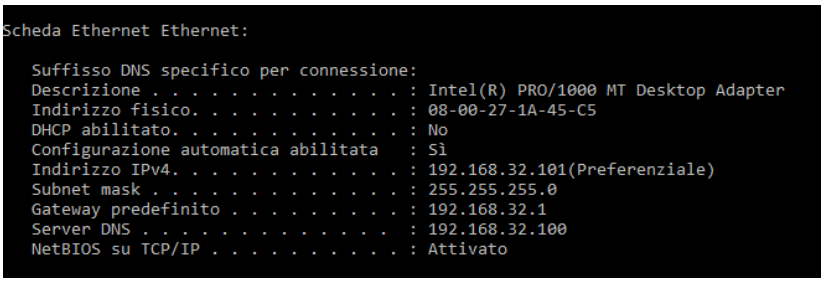
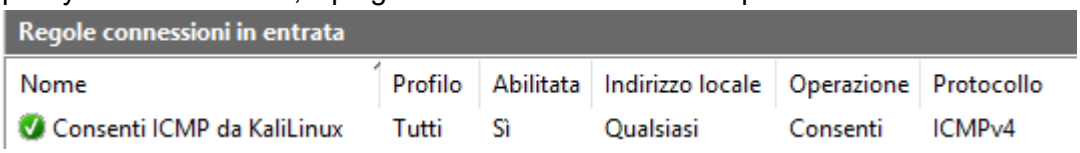
Progetto finale Modulo 1 (W4D4) - Alessandro Criscuoli

In questa esercitazione, eseguita su laboratorio virtuale VirtualBox, su Internal Network ho impostato gli IP delle macchine Kali Linux e Windows nei seguenti modi:





Come si può notare dalle immagini ho settato il DNS server con IP della KaliLinux su entrambe le vm.
Per verificare che le vm comunicassero correttamente ho fatto un test di ping reciproco, inizialmente ciò avveniva con esito positivo solo per il ping da Windows a Kali ma non da Kali a Windows, questo dato dalla protezione del Windows Firewall, una volta inserita la policy con action Allow, il ping da Kali a Windows è stato possibile da testare con esito positivo.



Una volta configurata la rete locale l'obiettivo era attivare i servizi DNS, HTTP e HTTPS in modo da renderli disponibili all'interno della rete, per poter simulare una architettura client-server.
In questo report si descrivono i passaggi per configurare come server la vm Kali e come client la vm Windows, come obiettivo finale il raggiungimento della risorsa tramite hostname **epicode.internal**

Ho optato per l'utilizzo di **Inetsim** per la simulazione dei servizi, iniziando dal DNS questo non poteva essere simulato per dei problemi di stabilità della versione di Net::DNS, un modulo in utilizzo da Inetsim, ho dovuto così eseguire un upgrade del modulo alla versione 1.37, recuperato dalla seguente repository: https://cpan.metacpan.org/authors/id/N/NL/NLNETLABS/Net-DNS-1.37.tar.gz

comandi utilizzati (da terminale)
wget *link*
tar -xzf Net-DNS-1.37.tar.gz #comando di estrazione, xzf sono le opzioni di estrazione
cd *path cartella estratta NetDNS*
perl Makefile.PL #controlla le dipendenze e genera il "Makefile" (file per l'installazione del modulo)
make #compilazione del modulo tramite il Makefile
make test #test del modulo
sudo make install #installa il modulo con permessi admin

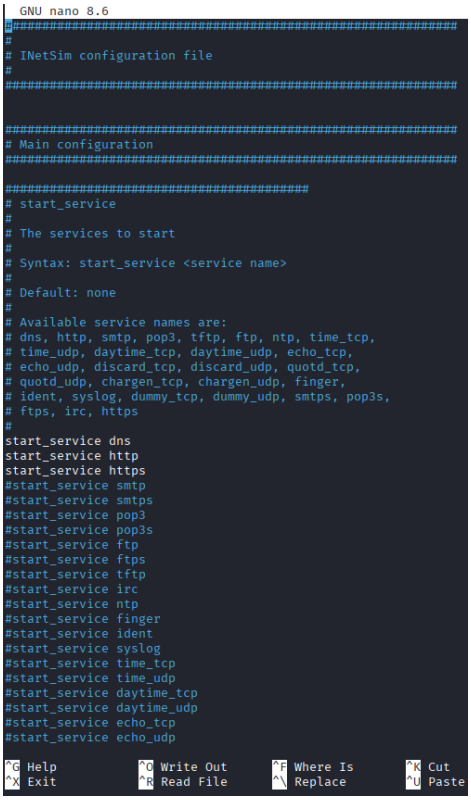
Nota: Makefile è un file di testo che contiene istruzioni per automatizzare la compilazione e l'installazione di software.

Fatti i passaggi per il FIX dei problemi di DNS service ho continuato con la configurazione di Inetsim.

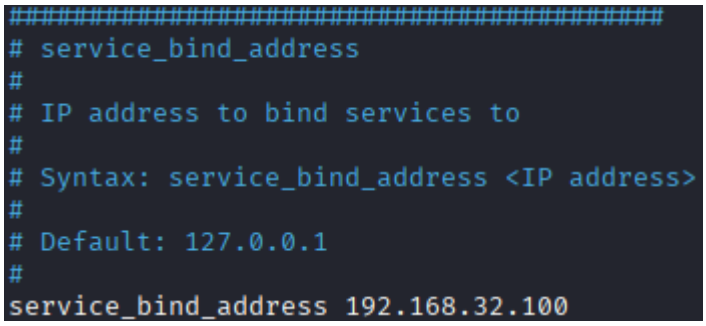
Configurazione Inetsim

entrando in modifica nel file di configurazione di Inetsim tramite comando:

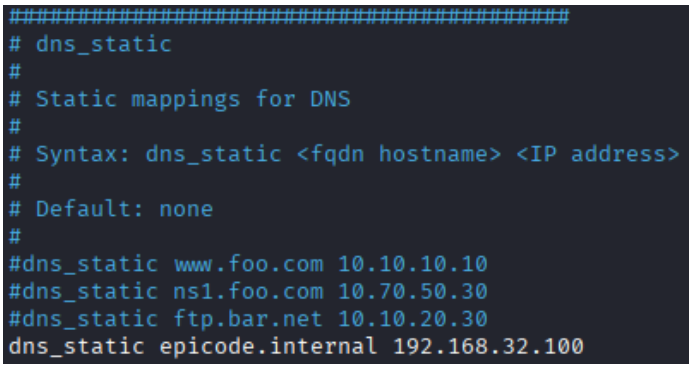
sudo nano /etc/inetsim/inetsim.conf



Qui ho aggiunto il carattere # per commentare lo start_service dei servizi non essenziali per la simulazione, tutti eccetto dns, http e https. Successivamente ho specificato l'IP address della macchina su cui verranno esposti i servizi simulati, ovvero la KaliLinux [192.168.32.100].



Service DNS

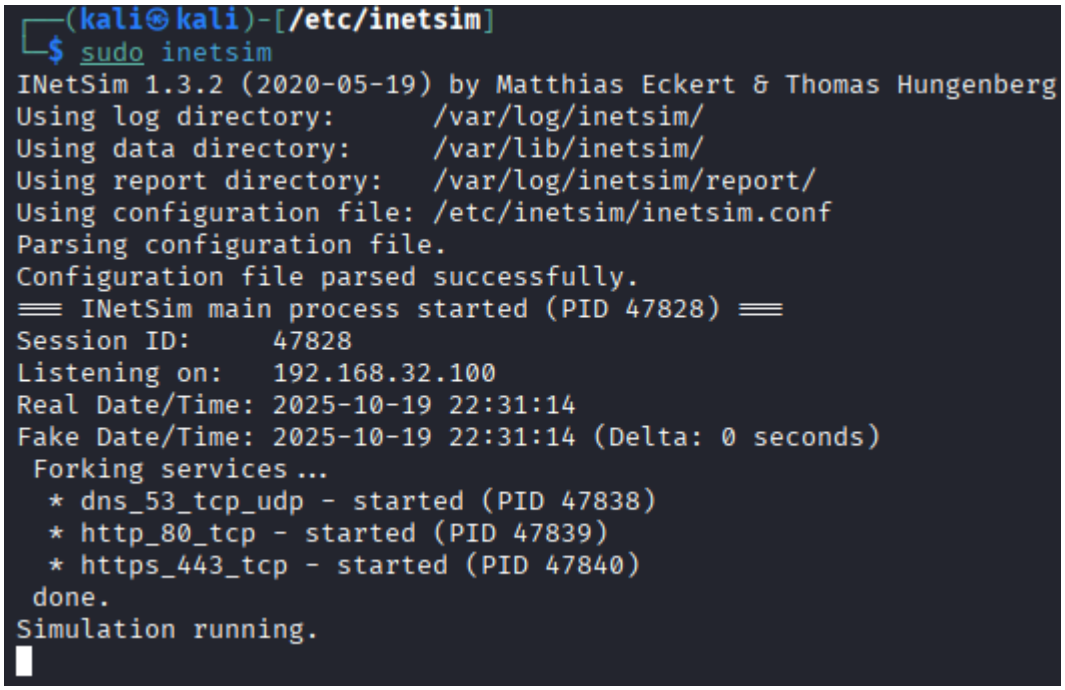


Per aggiungere il record DNS "epicode.internal" ho aggiunto sotto dns_static una riga in più, non commentata, dove specifico hostname seguito dall'IP address della vm KaliLinux.

I servizi HTTP/HTTPS non necessitano di nessuna configurazione, sono già impostati di default sulle rispettive porte 80 e 443, mentre il loro contenuto non è importante ai fini della simulazione.

Start dei servizi

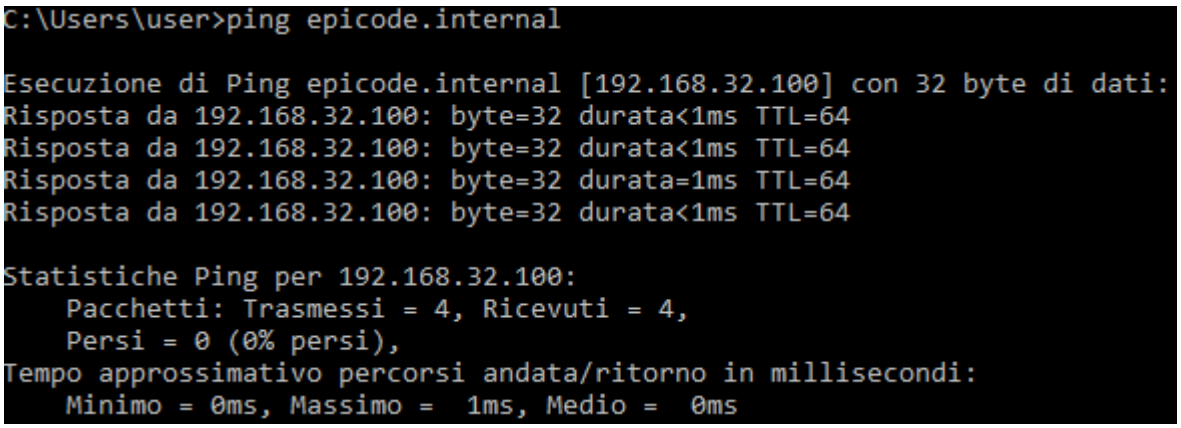
Per effettuare lo start dei servizi ho digitato da riga di comando “sudo inetsim” in modo da eseguire l’applicativo con permessi elevati.



Avendo effettuato tutto correttamente si possono vedere i PID (Process ID) del processo principale Inetsim e dei servizi a cui abbiamo lasciato lo start_service in file di configurazione, assieme allo stato “simulation running”.

Verifica da client

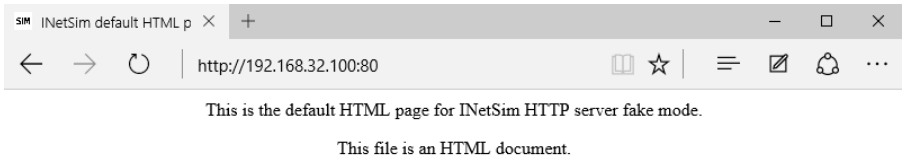
Da vm Windows sono andato ad effetuare le verifiche dei servizi:



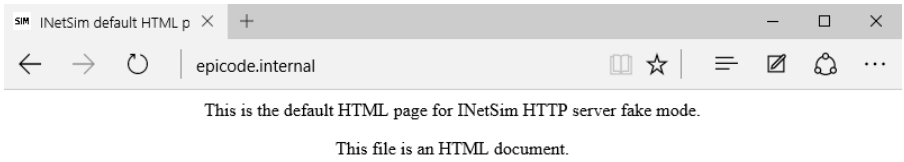
Pingando l’hostname *epicode.internal* verrà tradotto l’IP 192.168.32.100 [Kali]

Questo dimostra che la simulazione del server DNS sta funzionando sulla internal network.

Visita da Web Browser (http/https)



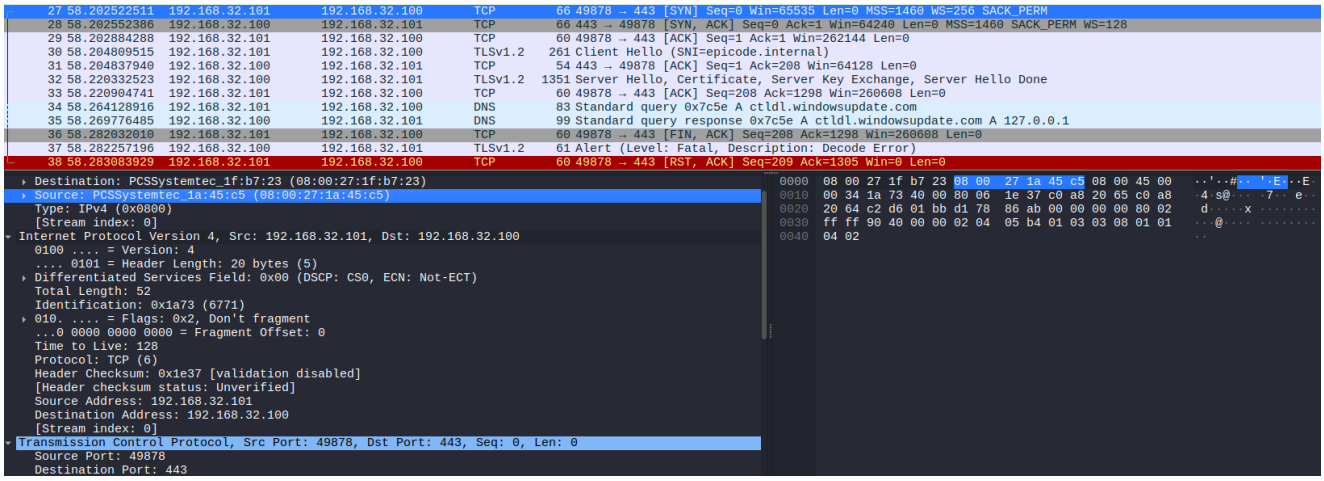
Immettendo l'indirizzo IP della KaliLinux sono stato indirizzato alla pagina http esposta sulla porta 80.



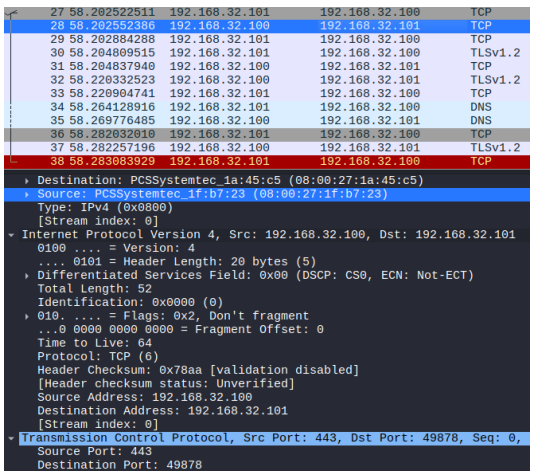
Immettendo l'hostname epicode.internal sono stato indirizzato alla pagina https esposta sulla porta 443.

Sniff pacchetti con Wireshark (HTTPS)

A questo punto per analizzare i pacchetti e vedere le request e response ho iniziato lo "sniffaggio" dell'interfaccia eth0 su Kali:



Le fasi catturate sono del momento in cui ho richiesto dal browser client di visitare l'indirizzo https://epicode.internal:443, si possono verificare i MAC address di source e destination, le porte di entrambi e le informazioni di ogni pacchetto. Nella prima request c'è:
Source MAC address [08-00-27-1A-45-C5] -> Windows / Source port -> 49878
Destination MAC address [08:00:27:1f:b7:23] -> Kali / Destination port -> 443

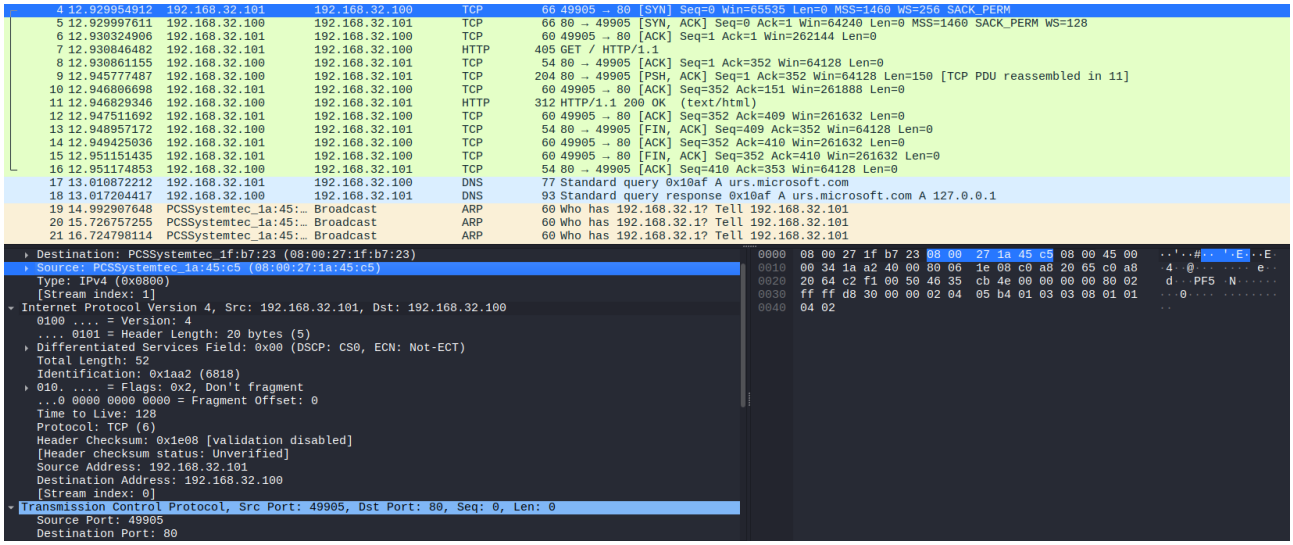


Subito dopo abbiamo la prima response con Source e Destination invertiti.

Nota: In questi screenshot si può notare un Error riguardante il protocollo TLS, dovuto al fatto che la pagina HTTPS non riesce a restituire nessun certificato SSL/TLS.

Sniff pacchetti con Wireshark (HTTP)

Ho ripetuto il procedimento mentre visitavo la pagina in http, questo è ciò che ne emerge:



Le fasi catturate sono del momento in cui ho richiesto dal browser client di visitare l'indirizzo http://192.168.32.100:80, oltre a notare gli stessi MAC address in Source e Destination si può notare un cambio delle porte, ora la porta in uso dal server è la 80, la default del servizio http. Oltre a questo mentre prima (in https) non si riusciva a vedere i contenuti delle request/response in chiaro ora questo è possibile.

Conclusione

Tramite l'utilizzo di Inetsim sono riuscito a simulare una architettura client-server in cui ho esposto ed utilizzato i servizi DNS, HTTP e HTTPS. Con Wireshark invece ho potuto analizzare lo scambio di pacchetti che avviene tra client e server durante la simulazione dei servizi.

Ho potuto constatare personalmente che la difficoltà maggiore della simulazione consisteva nel trovare una valida alternativa per la simulazione del DNS dato che quella proposta da Inetsim ha problemi di stabilità, dopo varie ore di troubleshooting ho trovato la soluzione aggiornando il Modulo Perl utilizzato da Inetsim per il DNS.