

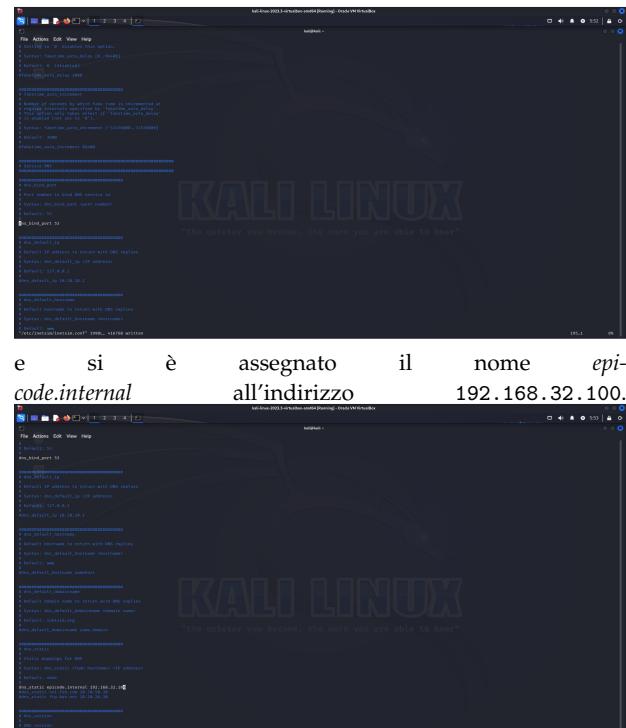
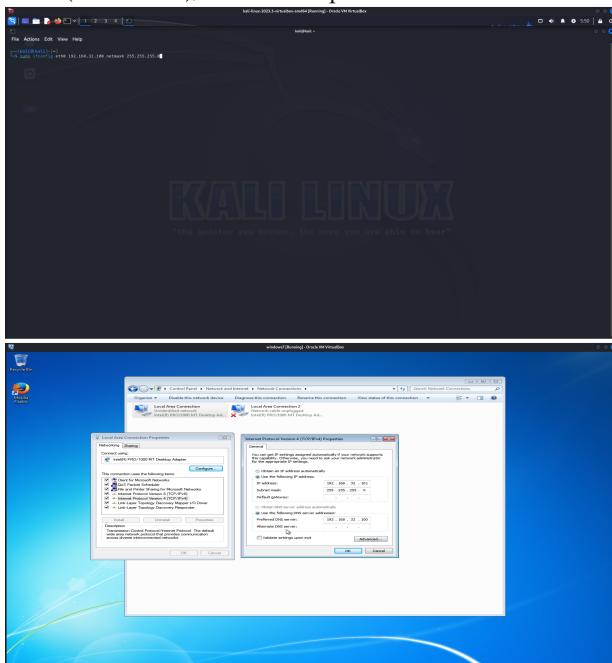
Creazione di una rete complessa

Alessandro Morabito

Il progetto prevedeva la simulazione di una rete composta da un client e un server. Il client montava il sistema operativo Windows 7 64-bit. Come server è stato utilizzato *INetSim* su Kali Linux per simulare un sistema DNS e un server HTTP/HTTPS. L'obiettivo era intercettare con *Wireshark* i pacchetti scambiati tra i due dispositivi e confrontarli successivamente.

1. Assegnazione indirizzi IP

Per disporre i due dispositivi sulla stessa rete interna è stato assegnato l'indirizzo IP 192.168.32.100/24 al server (Kali) utilizzando il comando *ifconfig*, e 192.168.32.101/24 al client (Windows 7), attraverso il pannello di controllo.

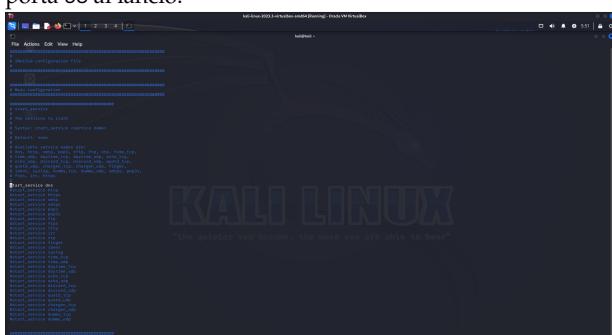


2. Configurazione Server

Il file di configurazione di *INetSim* è */etc/inetsim/inetsim.conf*.

2.1 Configurazione DNS

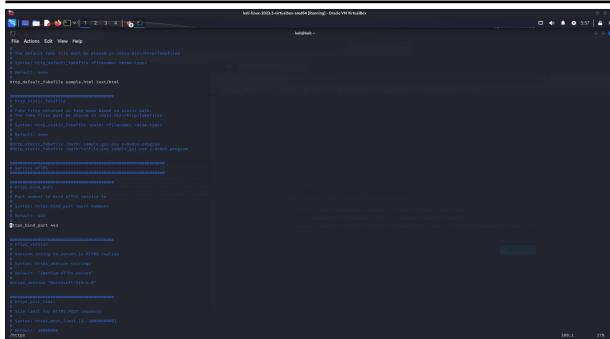
Dal momento che si voleva simulare un sistema DNS, si è fatto in modo tale che *INetSim* avviasse tale servizio sulla porta 53 al lancio.



2.2 Configurazione server HTTPS

Si è fatto quindi in modo che venisse avviato anche il servizio HTTPS sulla porta di default (443).



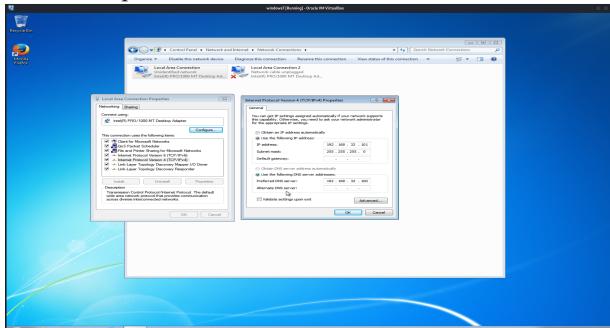


Infine, per rendere accessibile il tutto anche ad altri dispositivi all'interno della rete, si è assegnato l'indirizzo IP del server ai servizi di *INetSim*.

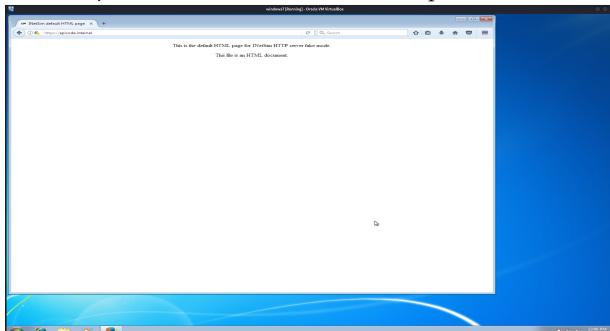


3. Configurazione Client

È stato quindi associato il server DNS al client.



Ci si è, in questo modo, riusciti a collegare al server *epicode.internal* utilizzando il protocollo HTTPS.

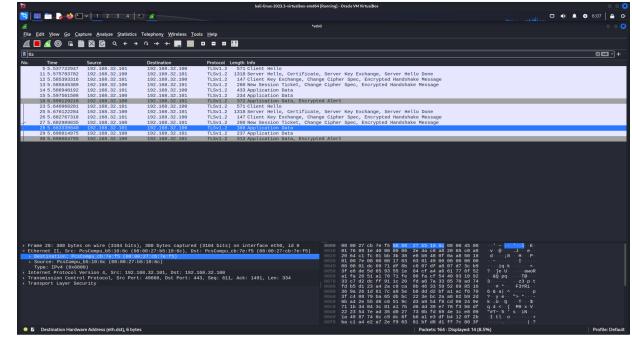


4. Configurazione Wireshark

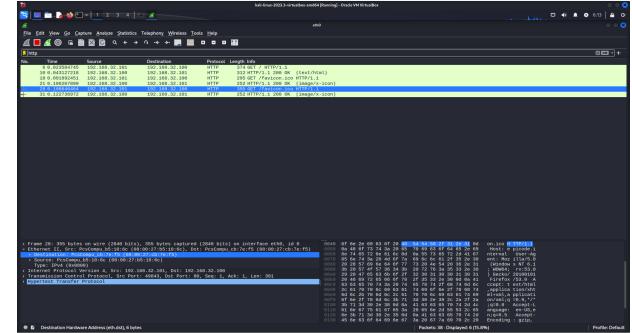
Si voleva intercettare il traffico tra i due dispositivi, in particolare il traffico HTTPS, per poi confrontarlo in un secondo momento con il traffico HTTP. Si è quindi avviato *Wireshark* sul server, e si sono poi filtrati i risultati in maniera tale che venissero visualizzati solamente i pacchetti che sfruttavano TLS.

Dopo aver avviato *INetSim* sul server, il client ha potuto raggiungere il server tramite il nome

epicode.internal, come impostato in precedenza.



Sono stati ripetuti i procedimenti di 2.2 per avviare un server HTTP sulla porta di default (80), ed è stato intercettato nuovamente il traffico, questa volta selezionando solamente i pacchetti che utilizzavano il protocollo HTTP.



5. Conclusioni

I pacchetti intercettati nel primo scenario utilizzavano il protocollo HTTPS, crittato. Infatti, si nota che il corpo dei pacchetti non era in chiaro: convertendo carattere per carattere il suo contenuto, infatti, i caratteri ASCII non ci risultano comprensibili. Al contrario, nello scenario in cui il client si connetteva al server utilizzando il protocollo HTTP, la sola intercettazione del pacchetto ci ha permesso di risalire al contenuto in chiaro.