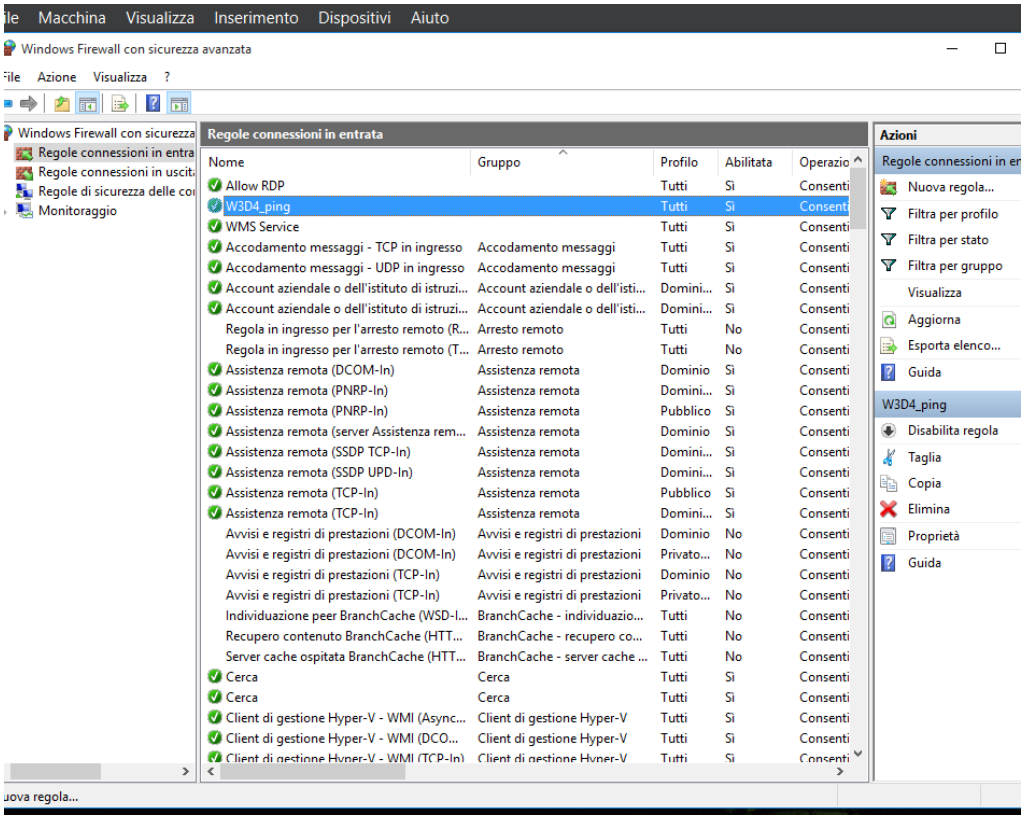
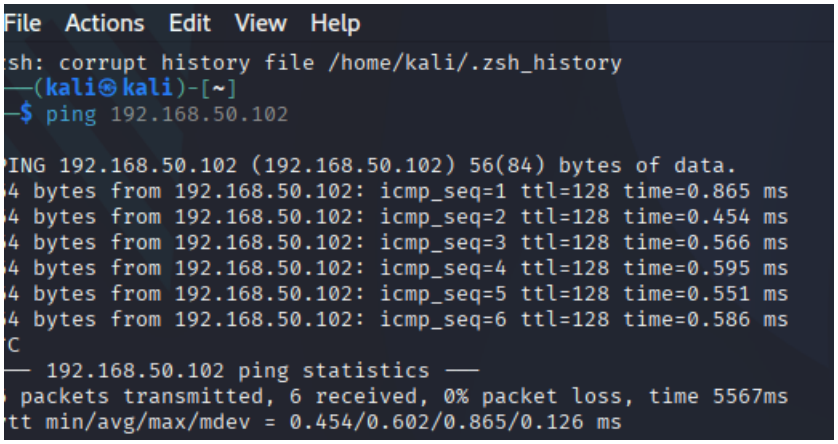
CONFIGURARE IL FIREWALL PER IL PING

Dovevo far funzionare il ping dalla mia macchina Kali Linux (IP 192.168.50.100) verso la macchina Windows (IP 192.168.50.102). Ho aperto il terminale su Kali e ho provato a lanciare ping 192.168.50.102, ma non arrivava nulla: tutti i pacchetti persi! Sapevo che il firewall di Windows blocca i ping di default, quindi sono passato alla macchina Windows per sistemare la cosa.
Sul Pannello di controllo di Windows, ho cercato “Windows Firewall” e ho aperto le impostazioni. Sono andato su “ Inbound Rules” (le regole per il traffico in entrata) e ho cliccato su “Nuova regola”. Ho scelto “Personalizzata” per avere pieno controllo. Non mi interessava un programma specifico, quindi ho selezionato “Tutti i programmi”. Poi, ho impostato il protocollo su ICMP, che è quello del ping. Ho lasciato “Qualsiasi indirizzo IP” per sorgente e destinazione, così non limitavo nulla. Ho scelto “Consenti la connessione” e ho applicato la regola a tutti i profili (dominio, privato, pubblico). Per trovarla facilmente, l’ho chiamata “W3D4_ping” con una breve descrizione. Ho salvato tutto e sono tornato su Kali.

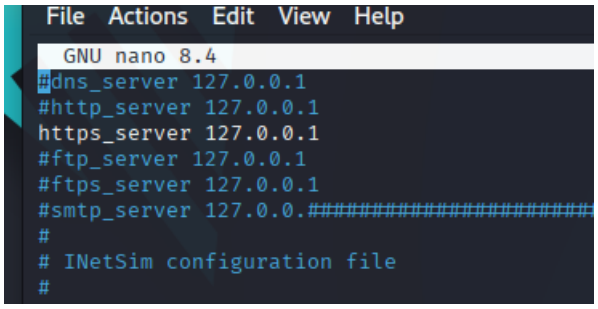


Ho rilanciato il comando ping 192.168.50.102 e tutti i pacchetti sono arrivati. Il ping funzionava.

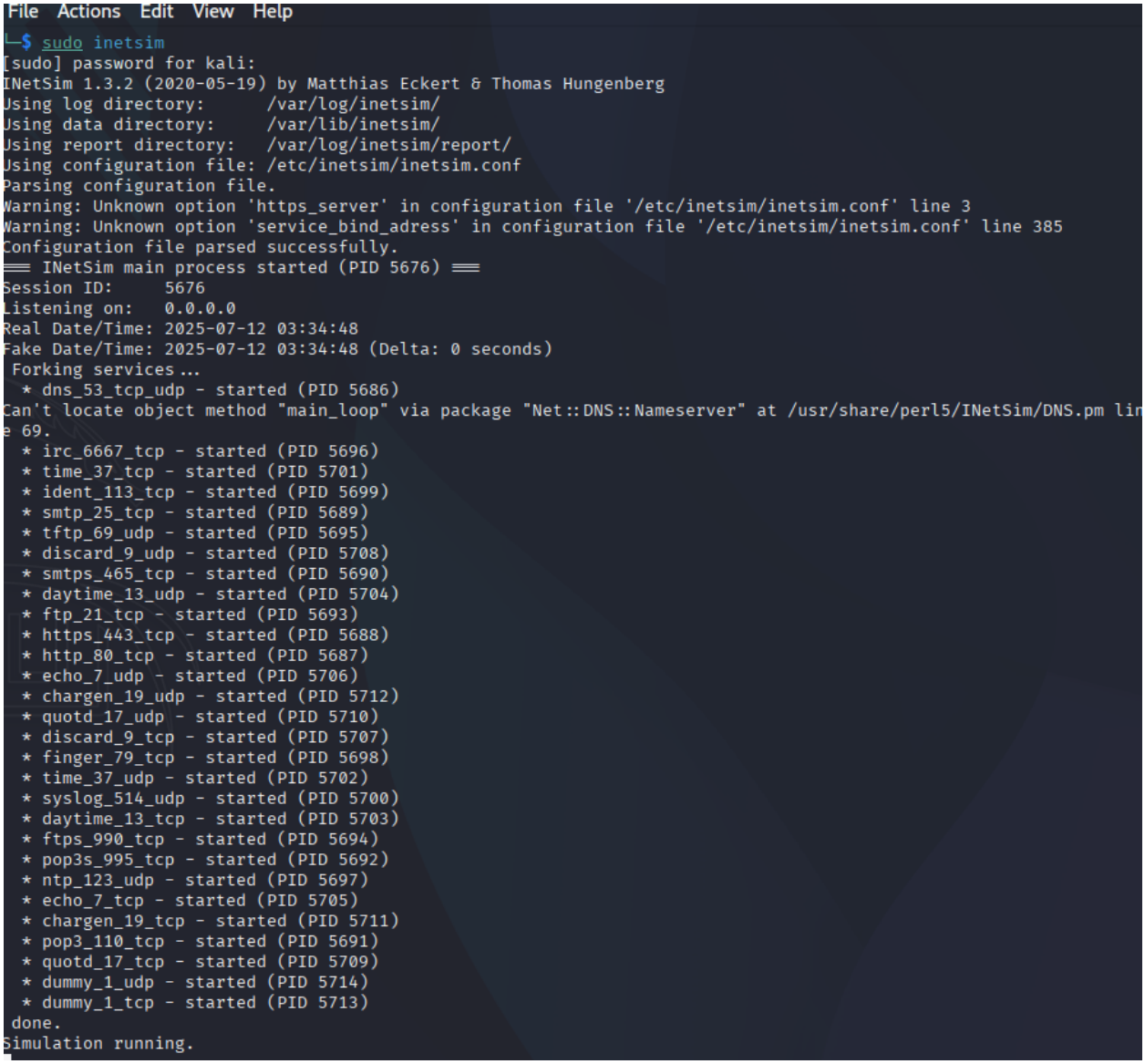


CONFIGURARE INTESIM PER HTTPS E USARE WIRESHARK

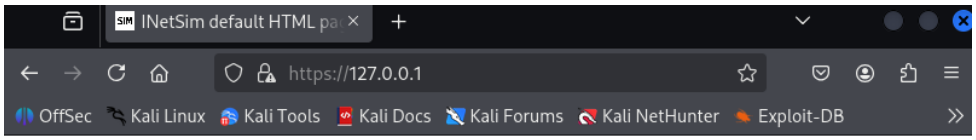
Ho aperto un terminale su Kali e ho digitato sudo nano /etc/inetsim/inetsim.conf per modificare il file di configurazione di InetSim. Quando ho aperto il file, c'era una lista infinita di servizi attivi: DNS, HTTP, FTP, ecc… Ma a me serviva solo HTTPS.
Le impostazioni di HTTPS erano già ok: il servizio ascoltava su localhost (127.0.0.1) e sulla porta 443, quella standard.



Ho salvato il file e ho avviato InetSim con sudo inetsim. L'output mi ha confermato che HTTPS era attivo sulla porta 443.



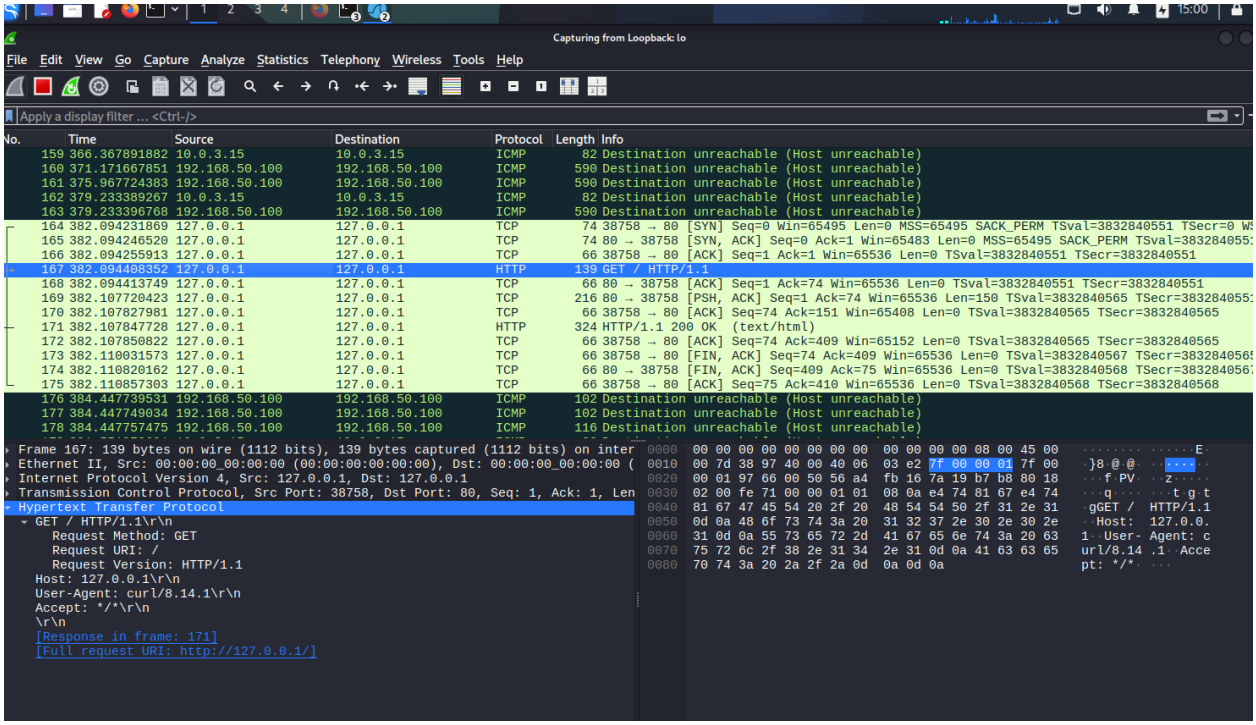
Per sicurezza, ho aperto un browser su Kali e ho digitato https://127.0.0.1



This is the default HTML page for INetSim HTTP server fake mode.

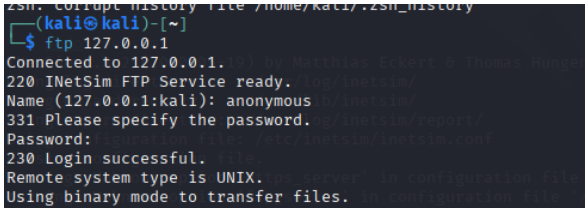
This file is an HTML document.

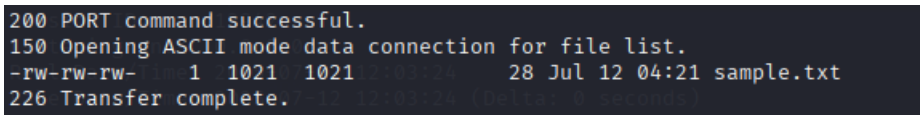
Poi, ho aperto Wireshark e ho messo in ascolto l'interfaccia di loopback (localhost). Ho rifatto la connessione a https://127.0.0.1dal browser e ho catturato il traffico. In Wireshark, ho visto e analizzato i dettagli dei pacchetti.



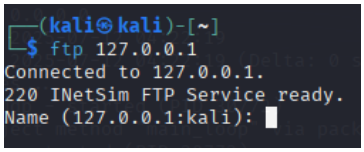
ESERCIZIO FACOLTATIVO
SIMULARE ALTRI SERVIZI CON INTESIM E ANALLIZARE IL TRAFFICO

Ho provato a simulare un altro servizio con InetSim, catturare il traffico e analizzarlo.
Ho scelto di configurare il servizio FTP, perché mi sembrava un buon modo per vedere qualcosa di diverso da HTTPS. Sono tornato al file di configurazione di InetSim con sudo nano /etc/inetsim/inetsim.conf. Ho cercato la sezione dedicata a FTP e ho rimosso il "#" dalla riga che abilita il servizio FTP, lasciando commentati gli altri servizi (tranne HTTPS, che avevo già configurato).
Le impostazioni di default per FTP erano: indirizzo localhost (127.0.0.1) e porta 21, che è quella standard. Non ho cambiato nulla, ho salvato il file e ho riavviato InetSim con sudo inetsim. L'output mi ha confermato che FTP era attivo sulla porta 21.
Per testarlo, ho aperto un terminale su Kali e ho provato a connettermi con ftp 127.0.0.1. Ho usato le credenziali di default di InetSim (utente: "anonymous", password: password). La connessione è riuscita, e ho visto la directory fittizia di InetSim.

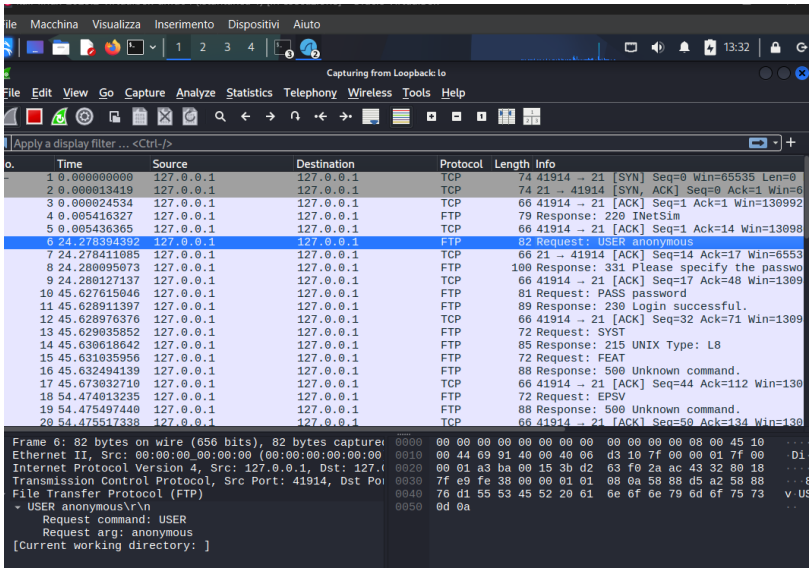




Poi, ho riavviato Wireshark, sempre sull'interfaccia di loopback, e ho rifatto la connessione FTP



Ho catturato il traffico e ho notato i pacchetti TCP per la connessione iniziale, ma anche i comandi FTP come " USER" e "PASS" per l'autenticazione. Nel pannello di Wireshark, ho analizzato il contenuto: potevo vedere chiaramente i comandi FTP inviati e le risposte del server simulato.



È stato utile per capire come funziona il protocollo FTP e come i pacchetti si muovono tra client e server