

INTRODUZIONE ED OBIETTIVO ESERCIZIO

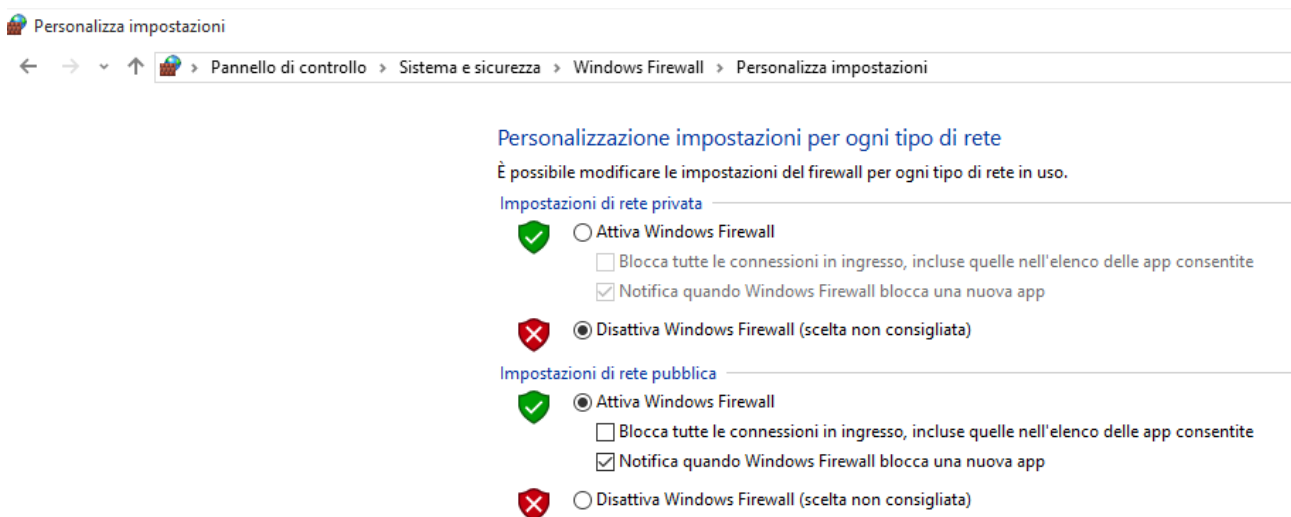
L'attività di laboratorio è stata condotta in ambiente virtuale con una macchina Windows in ruolo di client e una macchina Kali Linux destinata sia alla simulazione di servizi sia all'analisi del traffico.

Lo sviluppo del laboratorio ha previsto la preparazione, la definizione delle impostazioni minime di sicurezza su Windows e l'impiego degli strumenti di Kali per generare e osservare flussi di rete significativi. In pratica è stata abilitata la comunicazione ICMP mediante una regola dedicata del firewall e successivamente sono stati avviati con INetSim servizi "fittizi" per riprodurre scenari HTTP/HTTPS in modo controllato.

La comunicazione tra i due host è stata infine tracciata con Wireshark in modo tale da documentare con precisione il comportamento dei protocolli e l'evoluzione dei pacchetti lungo il percorso.

L'obiettivo è offrire una dimostrazione chiara della corretta messa in rete dei sistemi, della gestione consapevole del filtro firewall e della capacità di acquisire e interpretare il traffico generato da ping e richieste web.

PASSAGGIO 1



La prima fase del laboratorio è consistita nella verifica e nella regolazione delle impostazioni del firewall di Windows, con l'obiettivo di consentire la comunicazione ICMP proveniente dalla macchina Kali Linux.

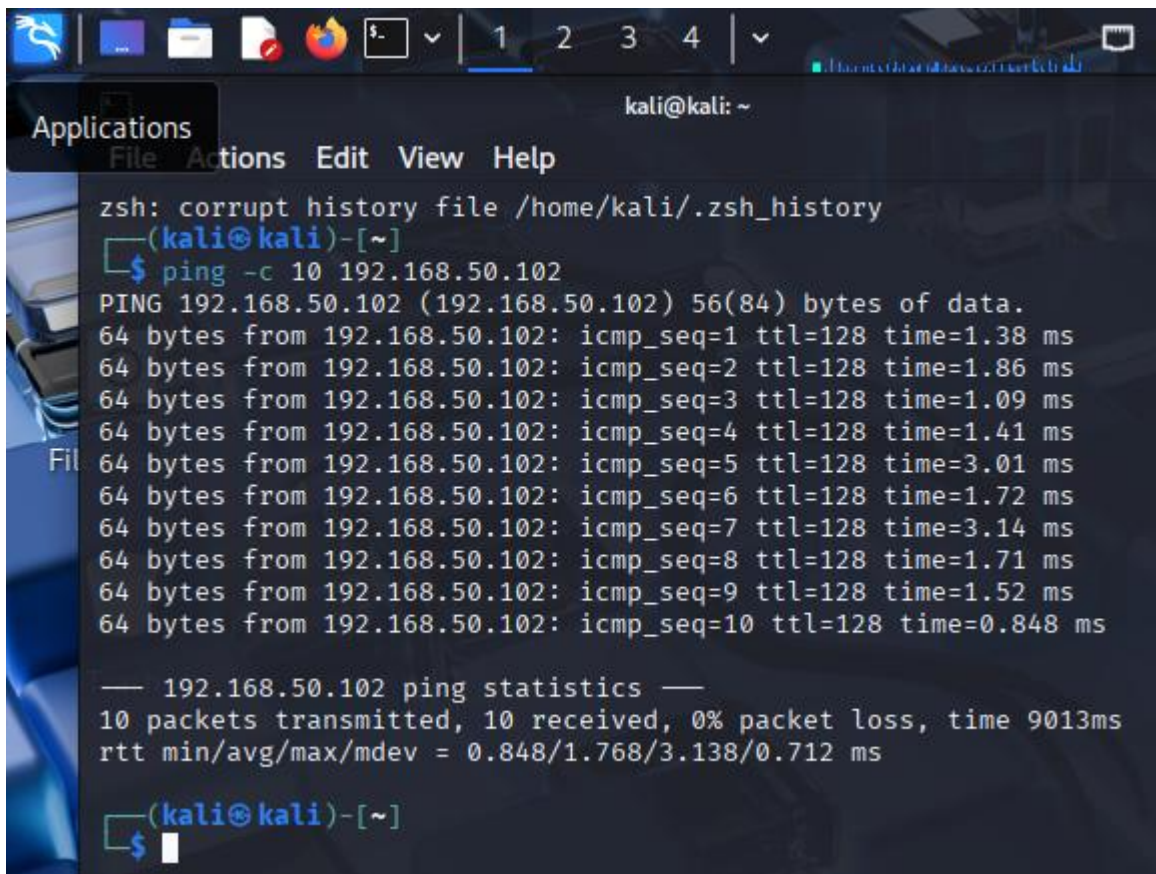
È stata aperta la console di gestione del firewall, analizzati i profili di rete attivi assicurandosi che la protezione fosse attiva sia per la rete privata che per quella pubblica.

Successivamente è stata creata una regola in entrata dedicata al protocollo ICMPv 4, necessaria per permettere il ping da Kali verso Windows.

Questa operazione ha rappresentato un passaggio fondamentale per garantire la corretta visibilità reciproca tra le due macchine.

La configurazione ha permesso di predisporre l'ambiente per le prove successive di connettività e di cattura del traffico.

PASSAGGIO 2



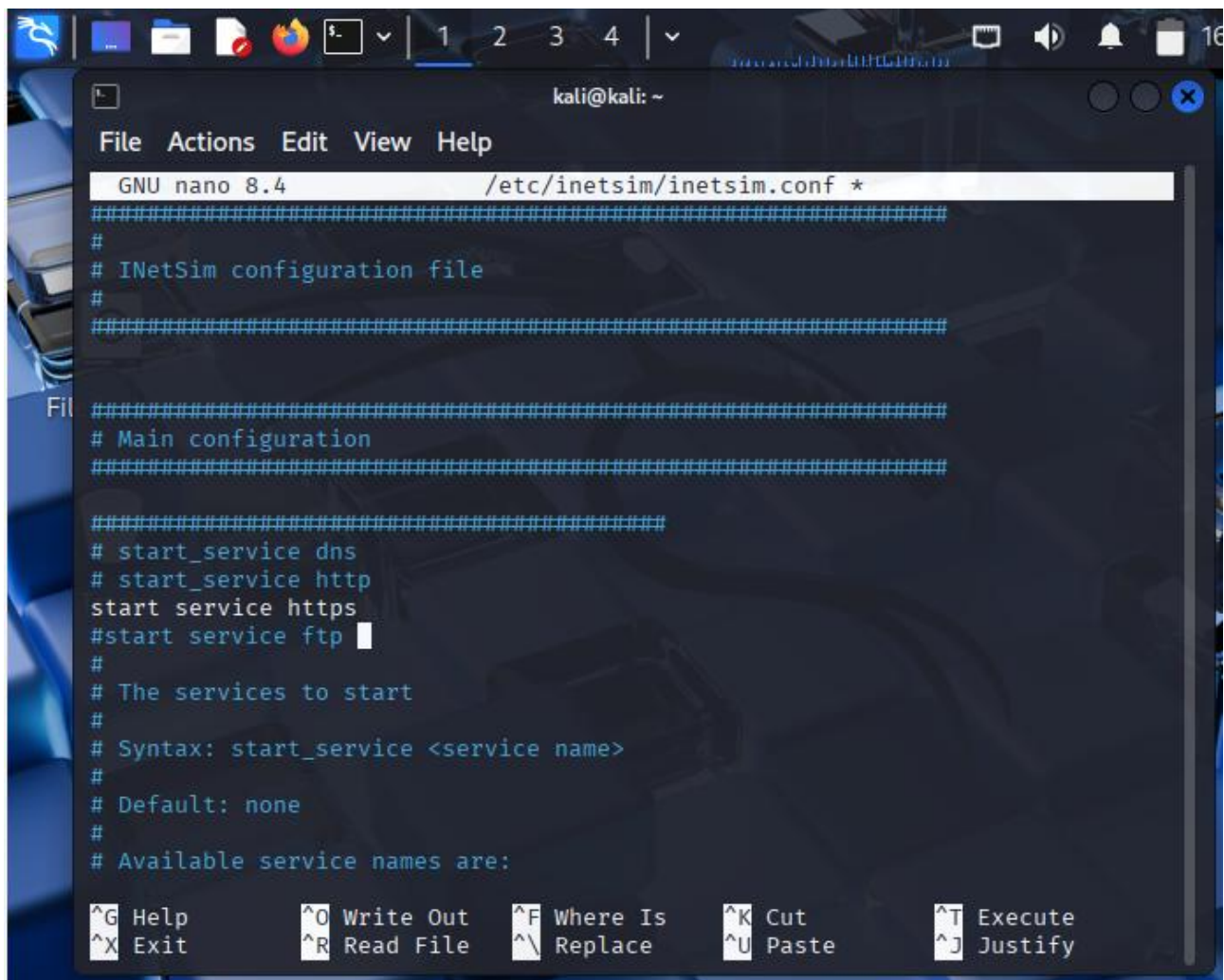
```
kali@kali: ~  
Applications  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ ping -c 10 192.168.50.102  
PING 192.168.50.102 (192.168.50.102) 56(84) bytes of data.  
64 bytes from 192.168.50.102: icmp_seq=1 ttl=128 time=1.38 ms  
64 bytes from 192.168.50.102: icmp_seq=2 ttl=128 time=1.86 ms  
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=1.09 ms  
64 bytes from 192.168.50.102: icmp_seq=4 ttl=128 time=1.41 ms  
64 bytes from 192.168.50.102: icmp_seq=5 ttl=128 time=3.01 ms  
64 bytes from 192.168.50.102: icmp_seq=6 ttl=128 time=1.72 ms  
64 bytes from 192.168.50.102: icmp_seq=7 ttl=128 time=3.14 ms  
64 bytes from 192.168.50.102: icmp_seq=8 ttl=128 time=1.71 ms  
64 bytes from 192.168.50.102: icmp_seq=9 ttl=128 time=1.52 ms  
64 bytes from 192.168.50.102: icmp_seq=10 ttl=128 time=0.848 ms  
  
— 192.168.50.102 ping statistics —  
10 packets transmitted, 10 received, 0% packet loss, time 9013ms  
rtt min/avg/max/mdev = 0.848/1.768/3.138/0.712 ms  
  
(kali@kali)-[~]  
$
```

Completata la configurazione del firewall, si è proceduto a testare la comunicazione tra le due macchine.

Dal prompt dei comandi di Kali Linux è stato eseguito il comando ping verso l'indirizzo IP della macchina Windows, con l'obiettivo di verificare che le richieste ICMP fossero correttamente inoltrate e ricevute.

L'esito del test ha mostrato la ricezione di tutte le risposte senza perdita di pacchetti, confermando che la regola impostata sul firewall funziona come previsto e la connettività di rete tra i due host è pienamente operativa.

PASSAGGIO 3



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf *  
#####  
#  
# INetSim configuration file  
#  
#####  
# Main configuration  
#####  
#####  
# start_service dns  
# start_service http  
start_service https  
#start_service ftp  
#  
# The services to start  
#  
# Syntax: start_service <service name>  
#  
# Default: none  
#  
# Available service names are:  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Dopo aver verificato la connettività, l'attenzione si è spostata sulla configurazione di INetSim, uno strumento che consente di simulare vari servizi di rete in modo controllato.

Attraverso l'editor di testo nano è stato aperto il file di configurazione principale del programma `'/etc/inetsim/inetsim.conf'` per abilitare i servizi necessari al laboratorio. In questa fase sono stati rimossi i commenti dalle righe relative ai servizi HTTPS e FTP, in modo da consentirne l'avvio automatico all'esecuzione di INetSim. Questa configurazione permette di creare un ambiente in cui la macchina Kali si comporta come un server che risponde alle richieste web provenienti dal client Windows.

PASSAGGIO 4

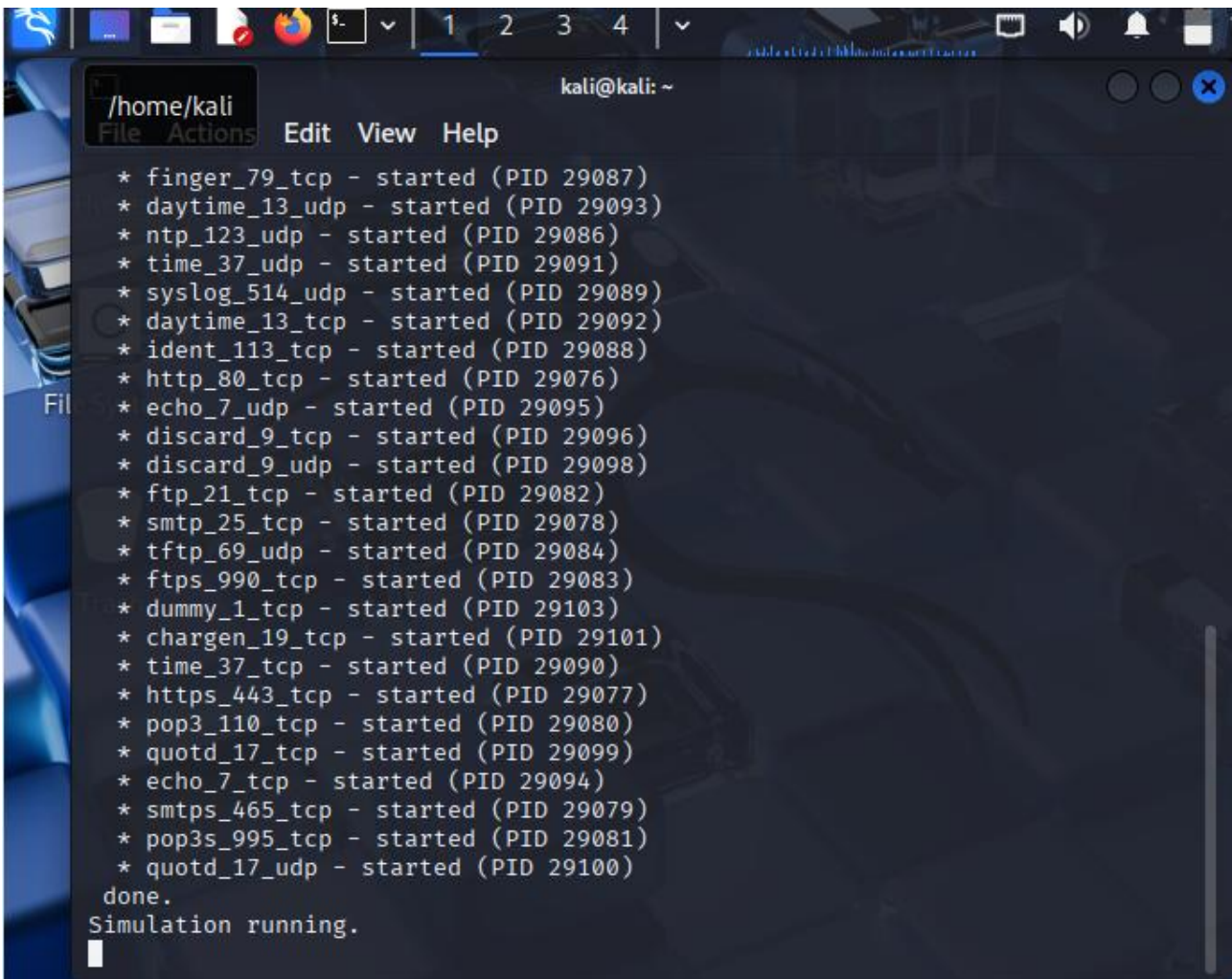
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.4 /etc/inetsim/inetsim.conf  
#  
# Syntax: http_static_fakefile <path> <filename> <mime-type>  
#  
# Default: none  
#  
#http_static_fakefile /path/ sample_gui.exe x-msdos-program  
#http_static_fakefile /path/to/file.exe sample_gui.exe x-msdos-program  
#####  
# Service HTTPS  
#####  
#####  
# https_bind_port  
#  
# Port number to bind HTTPS service to  
#  
# Syntax: https_bind_port <port number>  
#  
# Default: 443  
https_bind_port 443
```

Dopo aver abilitato i servizi HTTPS e FTP, è stato necessario completare la configurazione di INetSim abilitando la porta predefinita utilizzata per il traffico sicuro.

Nel file /etc/inetsim/inetsim.conf è stata quindi rimossa la linea di commento davanti alla voce https bind port 443, in modo da permettere al servizio HTTPS di mettersi in ascolto sulla porta 443.

Questa modifica garantisce che, una volta avviato INetSim, la macchina Kali sia in grado di simulare correttamente un server HTTPS, rispondendo alle richieste provenienti dal client Windows.

PASSAGGIO 5



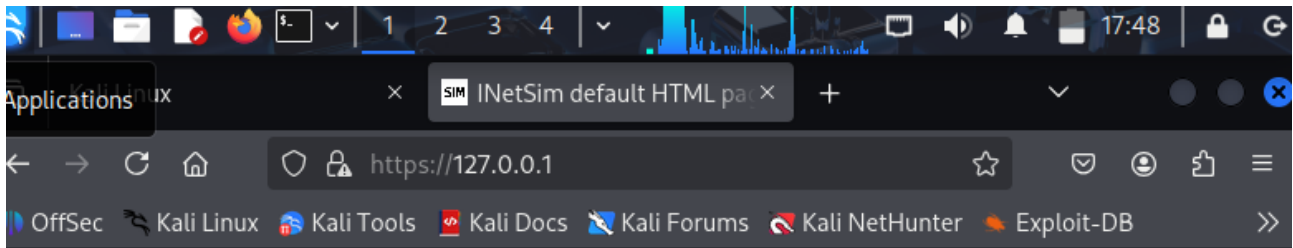
```
kali@kali: ~  
File Actions Edit View Help  
* finger_79_tcp - started (PID 29087)  
* daytime_13_udp - started (PID 29093)  
* ntp_123_udp - started (PID 29086)  
* time_37_udp - started (PID 29091)  
* syslog_514_udp - started (PID 29089)  
* daytime_13_tcp - started (PID 29092)  
* ident_113_tcp - started (PID 29088)  
* http_80_tcp - started (PID 29076)  
* echo_7_udp - started (PID 29095)  
* discard_9_tcp - started (PID 29096)  
* discard_9_udp - started (PID 29098)  
* ftp_21_tcp - started (PID 29082)  
* smtp_25_tcp - started (PID 29078)  
* tftp_69_udp - started (PID 29084)  
* ftps_990_tcp - started (PID 29083)  
* dummy_1_tcp - started (PID 29103)  
* chargen_19_tcp - started (PID 29101)  
* time_37_tcp - started (PID 29090)  
* https_443_tcp - started (PID 29077)  
* pop3_110_tcp - started (PID 29080)  
* quotd_17_tcp - started (PID 29099)  
* echo_7_tcp - started (PID 29094)  
* smtps_465_tcp - started (PID 29079)  
* pop3s_995_tcp - started (PID 29081)  
* quotd_17_udp - started (PID 29100)  
done.  
Simulation running.  
█
```

Dopo aver completato la configurazione, INetSim è stato avviato dal terminale di Kali Linux. L'esecuzione ha restituito una serie di messaggi che confermano l'attivazione dei diversi servizi simulati.

La presenza della dicitura finale "Simulation running" indica che tutti i servizi configurati sono stati correttamente avviati e che la macchina Kali è ora in grado di comportarsi come un vero e proprio server di rete.

In questa fase l'ambiente è quindi pronto per ricevere richieste da parte del client Windows e per generare traffico utile all'analisi successiva in Wireshark.

PASSAGGIO 6



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

This file is an HTML document.

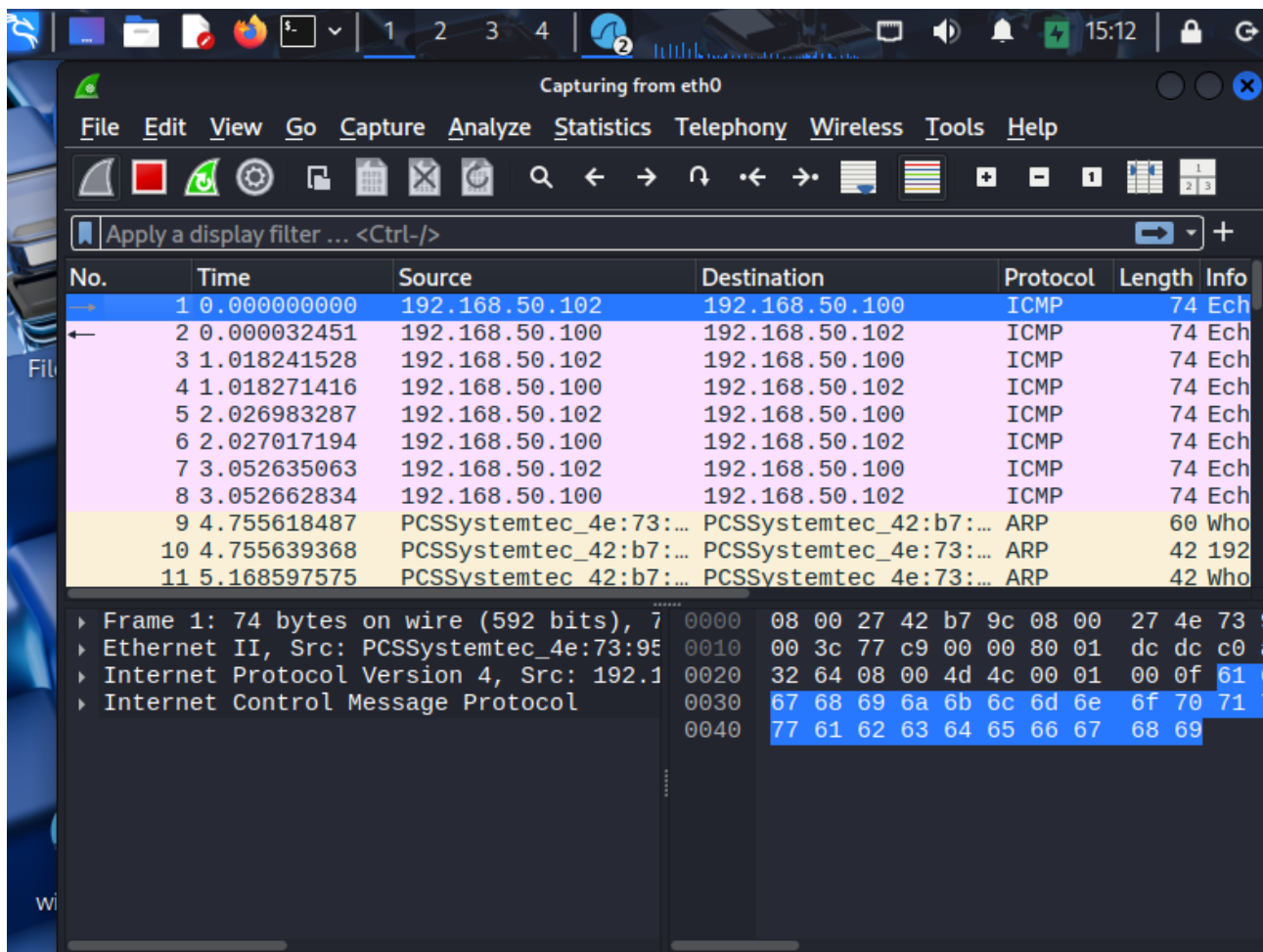
Dopo l'avvio di INetSim, è stata effettuata una prima verifica locale del corretto funzionamento del servizio HTTP simulato.

Dal browser della macchina Kali è stato inserito l'indirizzo <http://127.0.0.1>, che corrisponde al loopback della stessa macchina.

Il sistema ha restituito la pagina predefinita di INetSim, recitando il messaggio "This is the default HTML page for INetSim HTTP server fake mode. This file is an HTML document."

La visualizzazione di questa pagina conferma che il server HTTP fittizio è attivo e risponde correttamente alle richieste, dimostrando che la configurazione precedente è stata applicata con successo e che il traffico HTTP potrà essere analizzato nella fase di cattura successiva.

PASSAGGIO 7

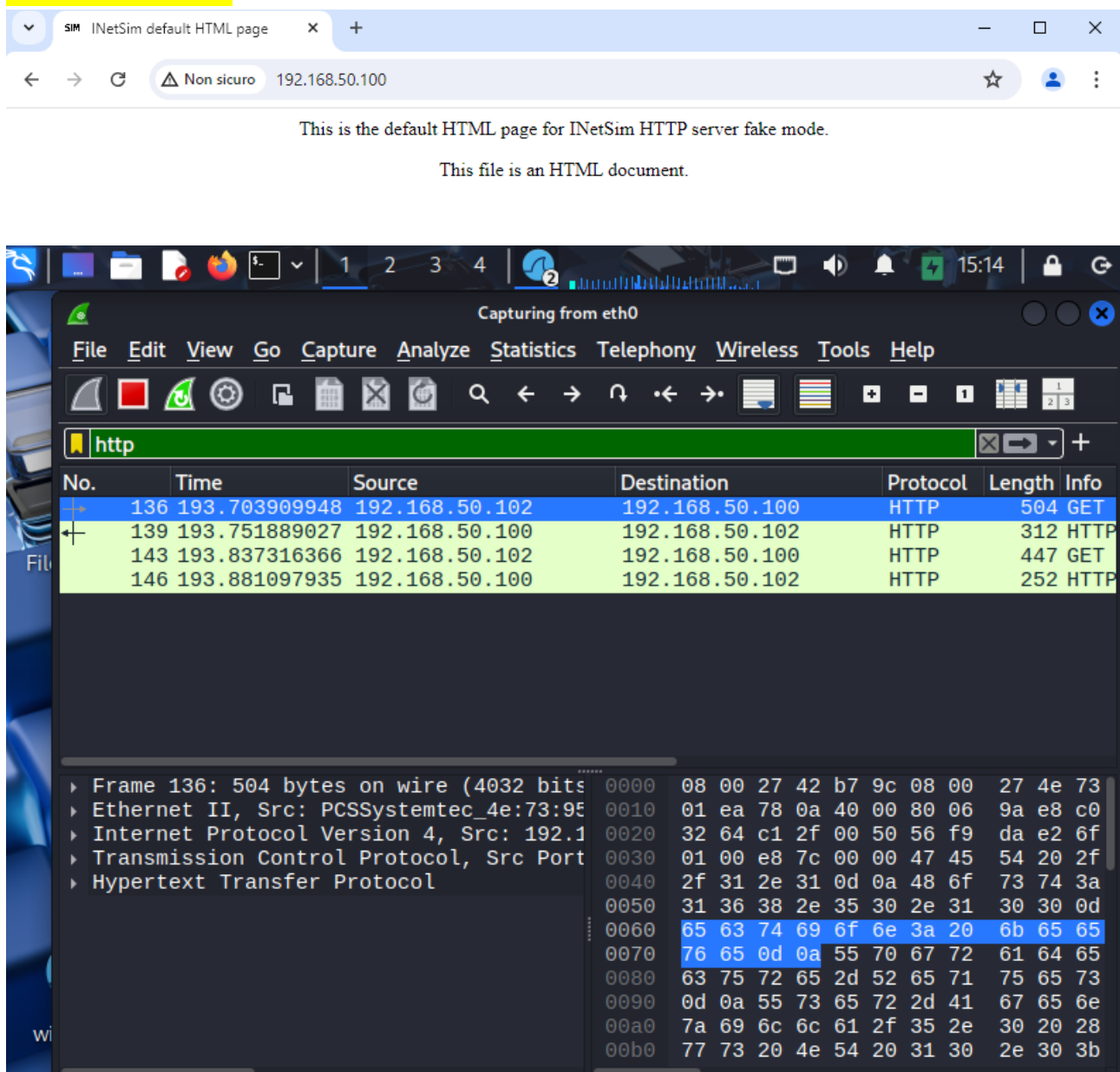


Per documentare la connettività è stata avviata una cattura sull'interfaccia di rete della macchina Kali configurata.

Durante l'esecuzione del comando ping dal client Windows sono emersi chiaramente i pacchetti ICMP Echo Request provenienti dall'host remoto e le corrispondenti Echo Reply inviate dalla macchina Kali, oltre a pacchetti ARP riconducibili alla risoluzione degli indirizzi.

L'analisi dei singoli frame ha permesso di verificare gli indirizzi IP sorgente e destinazione, la sequenza dei messaggi ICMP e l'assenza di packet loss, confermando così la correttezza della comunicazione a livello di rete e la validità della regola firewall precedentemente applicata.

PASSAGGIO 8



Per concludere l'esercitazione è stato eseguito un test di connessione HTTP dal browser della macchina Windows verso l'indirizzo della macchina Kali (<http://192.168.50.100>).

Il server INetSim ha risposto correttamente restituendo la pagina predefinita, segno che il servizio è attivo e funzionante.

Parallelamente, sul sistema Kali è stata avviata la cattura del traffico con Wireshark, filtrando i pacchetti con il protocollo http.

Dall'analisi è stato possibile osservare lo scambio tra le due macchine: la richiesta GET proveniente dal client Windows e la risposta generata dal server INetSim.

Questo ha permesso di visualizzare i campi chiave del protocollo confermando il corretto funzionamento della comunicazione a livello applicativo.

L'insieme delle due evidenze mostra in modo chiaro il completamento dell'intero flusso: la richiesta web simulata, la risposta del server e la cattura dettagliata del traffico che ne documenta la validità.

CONCLUSIONE

L'attività di laboratorio ha confermato la piena operatività dell'ambiente di test e la correttezza delle configurazioni applicate.

La regola sul firewall di Windows ha consentito il traffico ICMP in ingresso, rendendo possibile la validazione della connettività di base.

L'impiego di INetSim ha permesso di simulare in modo controllato servizi HTTP/HTTPS, mentre la successiva acquisizione con Wireshark ha documentato in maniera puntuale il comportamento dei protocolli e la sequenza degli scambi tra gli host.

Le evidenze raccolte dimostrano la coerenza tra obiettivi e risultati: la rete è stata predisposta correttamente, le dinamiche di filtraggio sono state applicate in modo mirato e il traffico generato è stato acquisito e interpretato con precisione.