

CORSO CYBER SECURITY W3D4

Studente: Orazio Morgillo

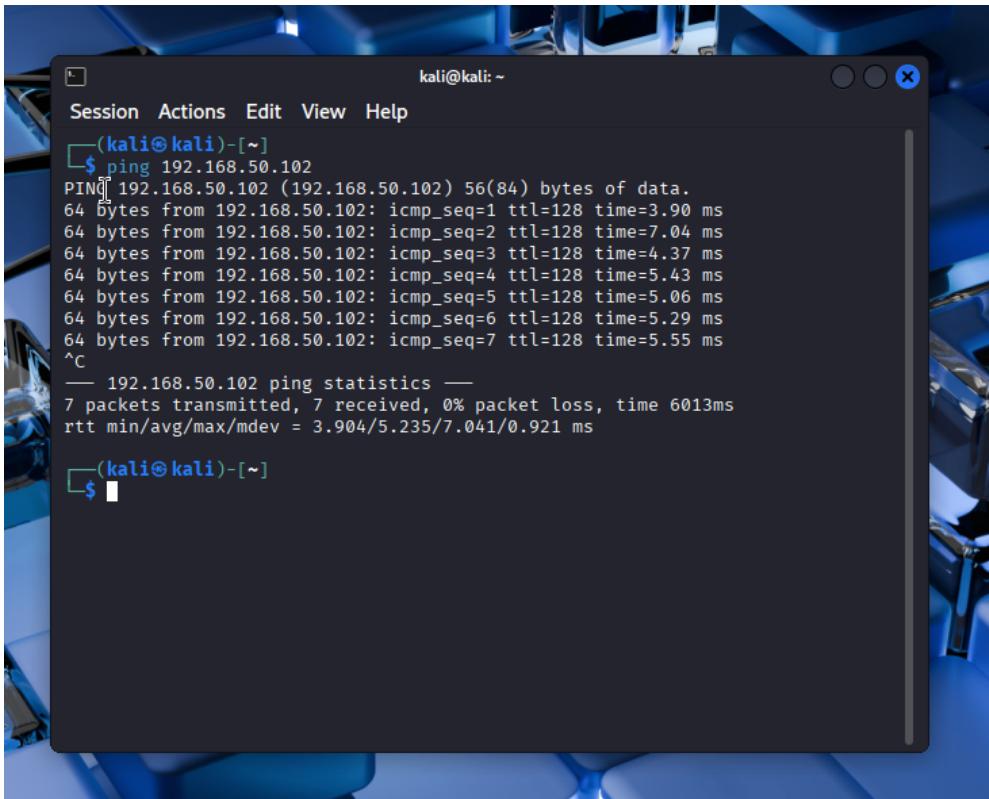
PRIMA PARTE

L'obiettivo della prima parte dell'esercizio è attivare il firewall sulla macchina virtuale windows, poiché di base esso risulta disattivato, e a pingarlo con la macchina virtuale Kali Linux.

The screenshot shows the 'Personalizzazione impostazioni per ogni tipo di rete' (Customization settings for every type of network) window. It includes sections for 'Impostazioni di rete privata' (Private network settings) and 'Impostazioni di rete pubblica' (Public network settings). Both sections have radio buttons for 'Attiva Windows Firewall' (Enable Windows Firewall) and 'Disattiva Windows Firewall (scelta non consigliata)' (Disable Windows Firewall (not recommended)). Under each setting, there are checkboxes for 'Blocca tutte le connessioni in ingresso, incluse quelle nell'elenco delle app consentite' (Blocks all incoming connections, including those in the allowed apps list) and 'Notifica quando Windows Firewall blocca una nuova app' (Notify when Windows Firewall blocks a new app).

Dopodiché, creiamo su windows firewall una nuova regola personalizzata con protocollo ICMP (utilizzato dal ping) aperto a tutti gli indirizzi IP e testiamo da Kali che il ping sia stato correttamente instaurato tra le due macchine virtuali.

The screenshot shows the 'Windows Firewall con sicurezza avanzata' (Windows Firewall with Advanced Security) interface. On the left, a navigation pane lists 'Regole connessioni in entrata', 'Regole connessioni in uscita', 'Regole di sicurezza delle conn...', and 'Monitoraggio'. The main pane displays a table titled 'Regole connessioni in entrata' (Inbound connection rules) with columns: Nome (Name), Gruppo (Group), Profilo (Profile), and Abilitata (Enabled). A right-hand sidebar titled 'Azioni' (Actions) provides options for managing rules, including 'Nuova regola...' (New rule...), 'Filtra per profilo' (Filter by profile), 'Filtra per stato' (Filter by status), 'Filtra per gruppo' (Filter by group), 'Visualizza' (View), 'Aggiorna' (Update), 'Esporta elenco...' (Export list...), 'Guida' (Help), 'Disabilita regola' (Disable rule), 'Taglia' (Cut), 'Copia' (Copy), 'Elimina' (Delete), and 'Proprietà' (Properties).

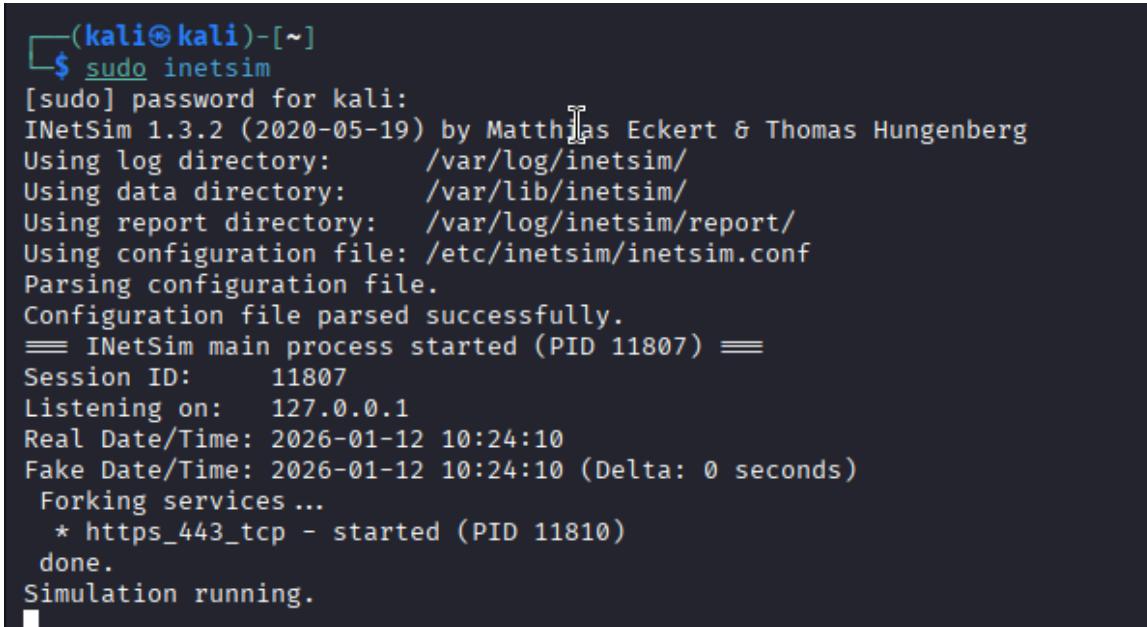


```
kali@kali: ~
Session Actions Edit View Help
└─(kali㉿kali)-[~]
$ ping 192.168.50.102
PING[ 192.168.50.102 (192.168.50.102) 56(84) bytes of data.
64 bytes from 192.168.50.102: icmp_seq=1 ttl=128 time=3.90 ms
64 bytes from 192.168.50.102: icmp_seq=2 ttl=128 time=7.04 ms
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=4.37 ms
64 bytes from 192.168.50.102: icmp_seq=4 ttl=128 time=5.43 ms
64 bytes from 192.168.50.102: icmp_seq=5 ttl=128 time=5.06 ms
64 bytes from 192.168.50.102: icmp_seq=6 ttl=128 time=5.29 ms
64 bytes from 192.168.50.102: icmp_seq=7 ttl=128 time=5.55 ms
^C
--- 192.168.50.102 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6013ms
rtt min/avg/max/mdev = 3.904/5.235/7.041/0.921 ms
└─(kali㉿kali)-[~]
$
```

Come si vede dallo screenshot sopra, i pacchetti sono stati trasmessi quindi vuol dire che il ping tra le macchine virtuali è andato a buon fine.

La seconda parte dell'esercizio, invece, consiste nell'utilizzare il software Wireshark preinstallato in Kali per la cattura dei pacchetti e l'analisi del contenuto.

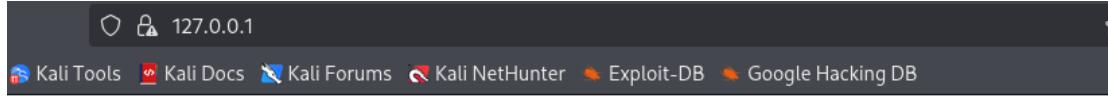
Utilizziamo prima inetsim (un altro software preinstallato in Kali, un simulatore di servizi internet) e configuriamolo scegliendo di avviare solo il servizio https.



```
└─(kali㉿kali)-[~]
$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 11807) ==
Session ID:    11807
Listening on:  127.0.0.1
Real Date/Time: 2026-01-12 10:24:10
Fake Date/Time: 2026-01-12 10:24:10 (Delta: 0 seconds)
Forking services ...
 * https_443_tcp - started (PID 11810)
done.
Simulation running.
```

Lanciando dal terminale il servizio inetsim con il comando sudo inetsim, notiamo che il servizio HTTPS è in ascolto sulla porta 443. Successivamente testiamo su internet se effettivamente tutto è andato a buon fine collegandoci ad un indirizzo <https://127.0.0.1>. Riceviamo come schermata di output la pagina fittizia di inetsim.

Vuol dire che il servizio è attivo e raggiungibile dalla macchina.



```
kali@kali: ~
Session Actions Edit View Help
└─(kali㉿kali)-[~]
$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 1587) ==
Session ID:      1587
Listening on:    127.0.0.1
Real Date/Time: 2026-01-12 12:01:28
Fake Date/Time: 2026-01-12 12:01:28 (Delta: 0 seconds)
Forking services...
* https_443_tcp - started (PID 1597)
done.
Simulation running.
```

Adesso andiamo su whireshark in loopback e avviamo il software.

Per semplicità possiamo aggiungere il filtro TCP che ricerca solo questo protocollo e possiamo notare, tramite la parentesi a sinistra, whireshark ha “preso un frame” (un insieme di pacchetti).

La source e la destination coincidono quindi stiamo operando nello stesso ambiente.

In info leggiamo che dalla porta 51064 alla porta 443 abbiamo mandato una richiesta di ping e abbiamo ricevuto una risposta dalla porta 443 alla porta 51064.

