

Windows Firewall

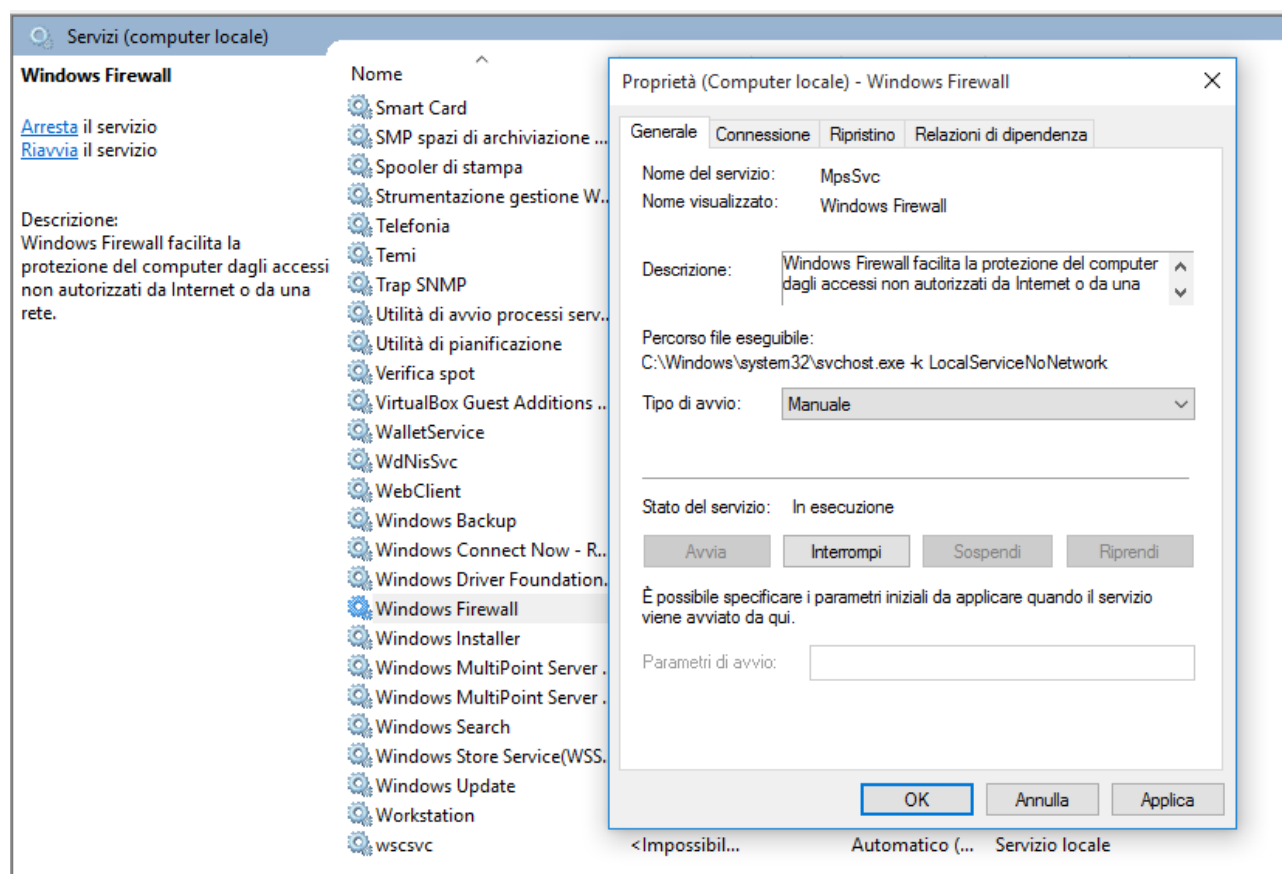
Per prima cosa andremo a configurare il Firewall di Windows, permettendo alla macchina di Kali di riuscire a comunicare con Windows. Con il Firewall impostato in automatico possiamo vedere come Kali non sia in grado di pingare la macchina.

```
(kali㉿kali)-[~]
$ ping -c4 192.168.50.102
PING 192.168.50.102 (192.168.50.102) 56(84) bytes of data.

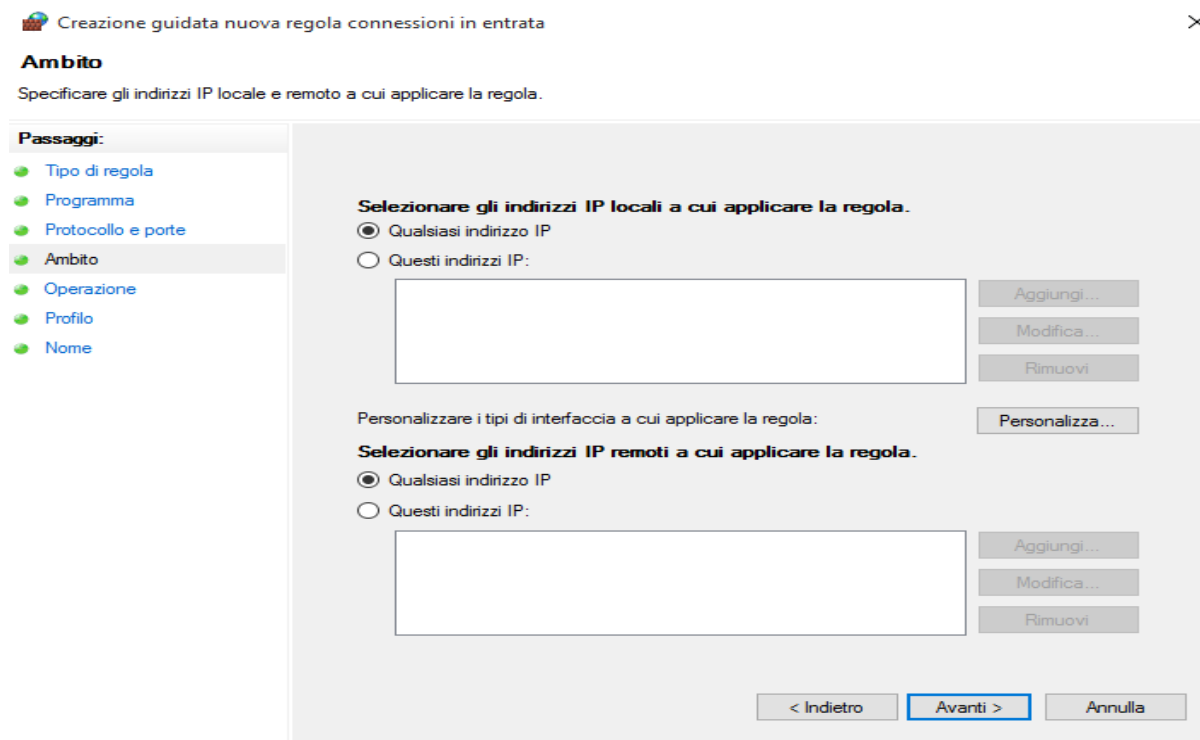
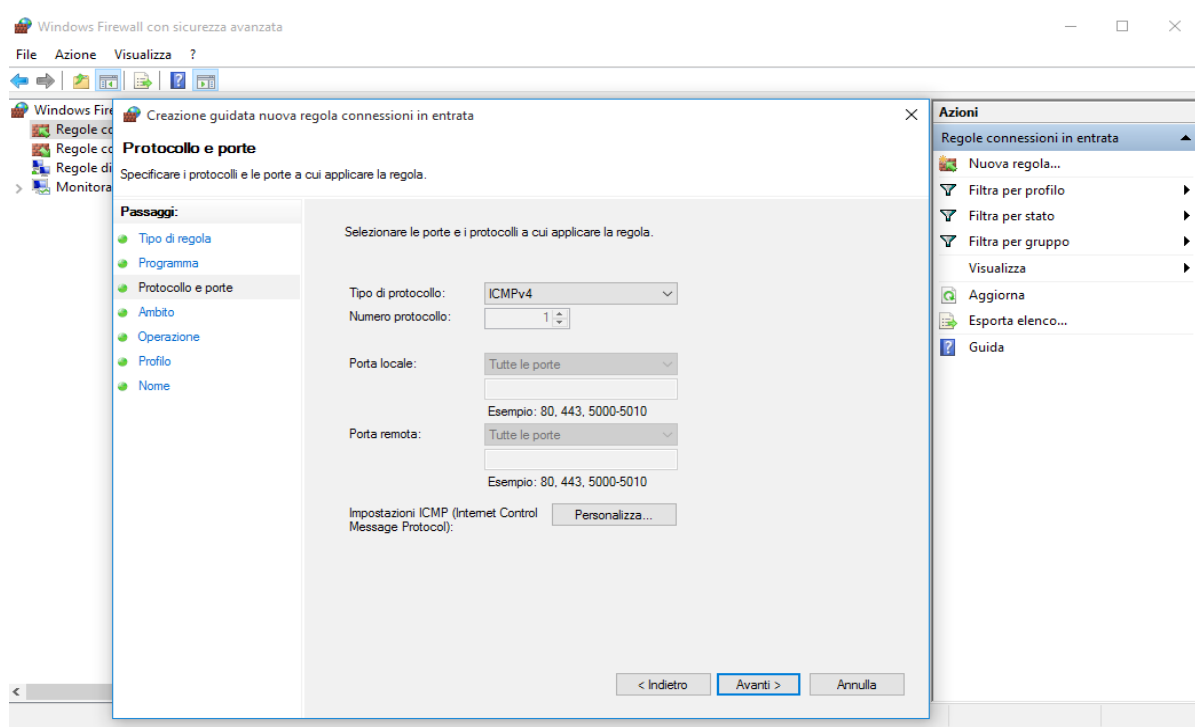
— 192.168.50.102 ping statistics —
4 packets transmitted, 0 received, 100% packet loss, time 3059ms

(kali㉿kali)-[~]
$
```

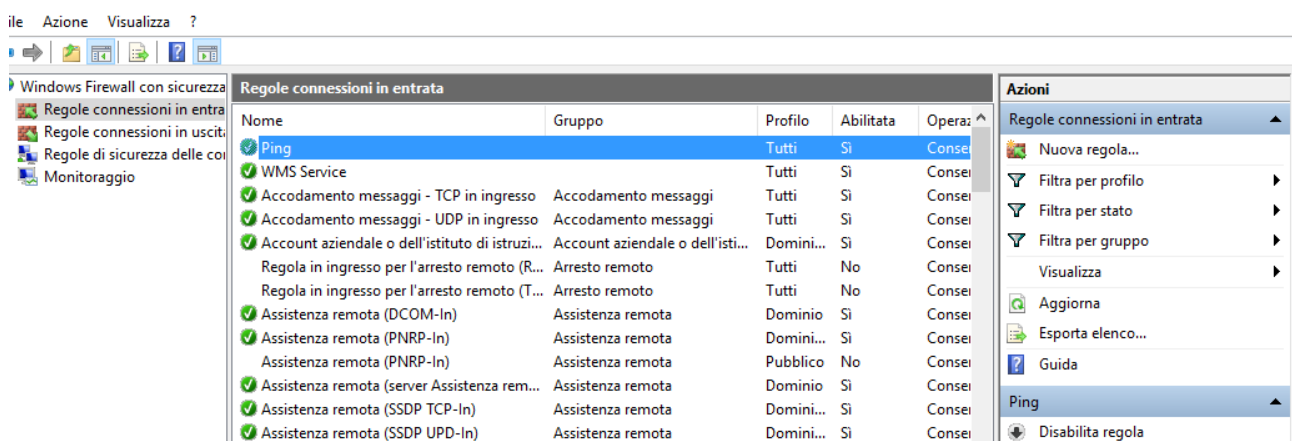
Quindi, per prima cosa attiviamo il firewall e impostiamolo su “Manuale”, tramite la schermata di servizi.



Ora possiamo creare delle nuove “Regole”, andando sulla schermata del firewall tramite pannello di controllo, inseriamo una nuova regola che permette al protocollo ICMP di fare comunicare determinati indirizzi IP, in questo caso li lasciamo su tutti gli indirizzi IP.



Ora che abbiamo creato correttamente questa nuova regola proviamo ad eseguire di nuovo la prova da Kali e vedere come la macchina ora riesce a comunicare.



```
(kali@kali)-[~]
$ ping -c4 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=1.12 ms
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=4.05 ms
64 bytes from 192.168.50.102: icmp_seq=4 ttl=128 time=1.05 ms

— 192.168.50.102 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3027ms
rtt min/avg/max/mdev = 1.052/1.897/4.049/1.247 ms

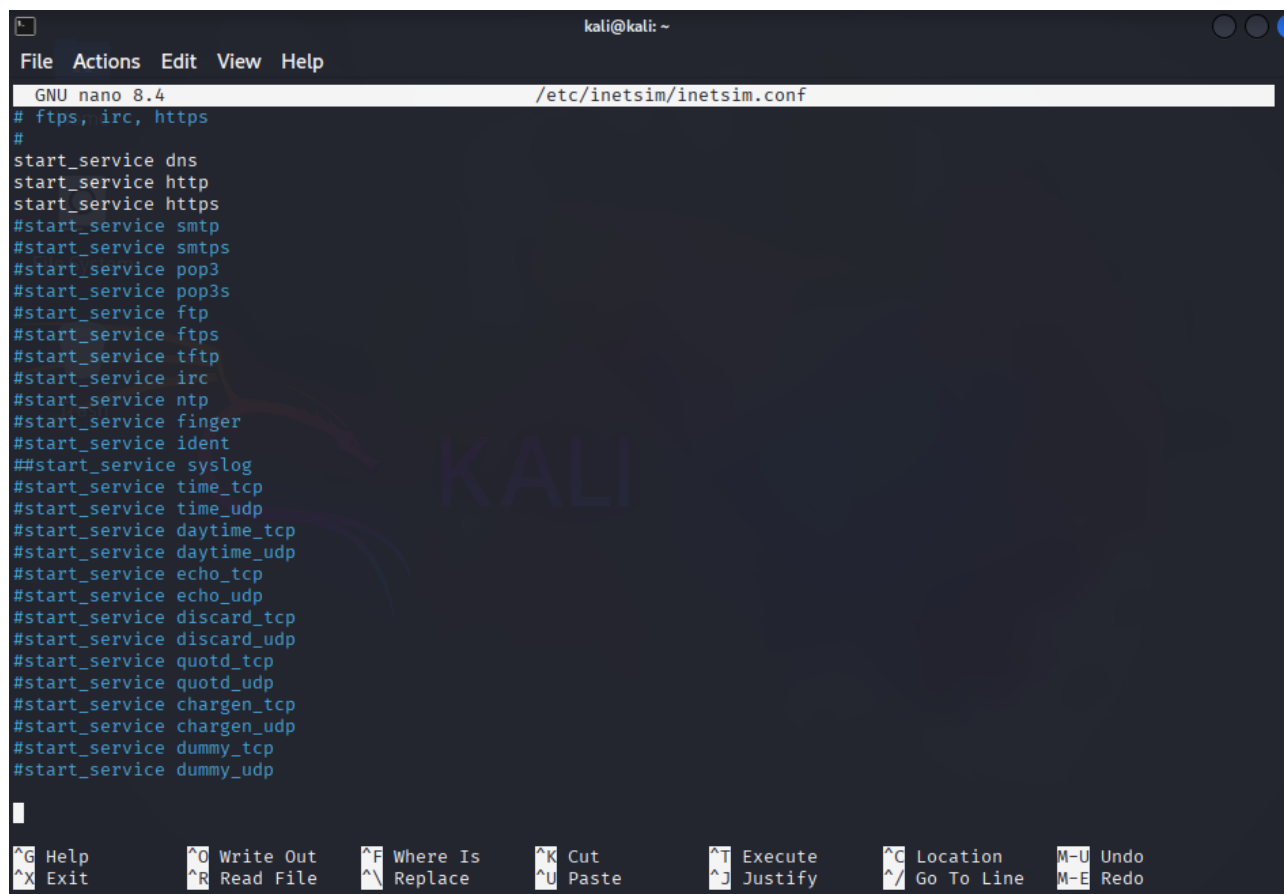
(kali@kali)-[~]
$
```

Wireshark

Ora utilizzeremo Wireshark per catturare dei pacchetti e analizzarne il contenuto.

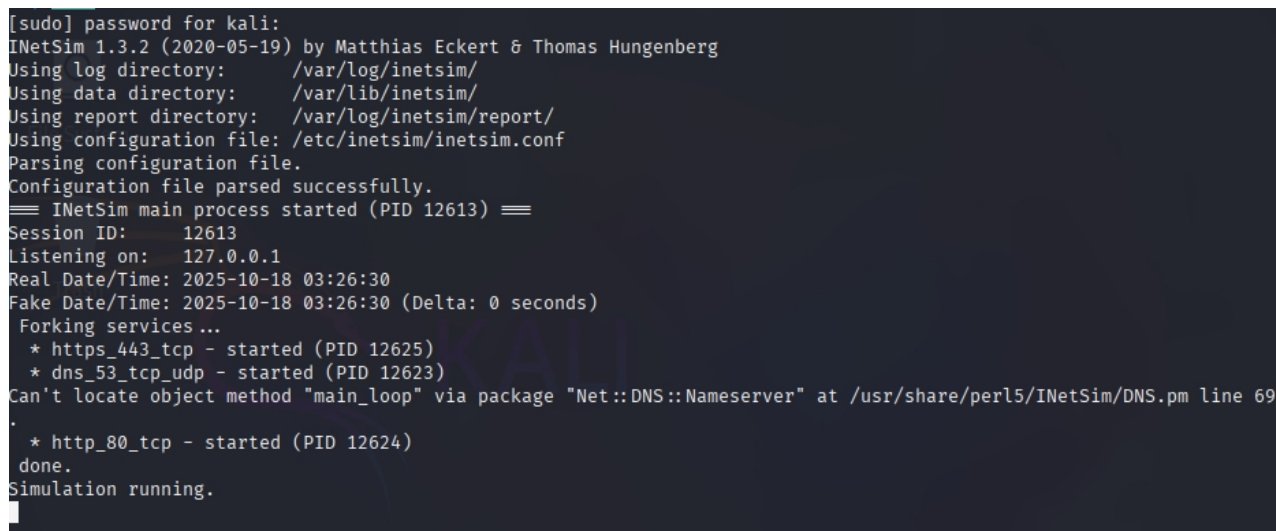
Per aiutarci utilizzeremo InetSim, che ci permette di simulare alcuni servizi, come Dns, Http, Https ecc. Per eseguirlo apriamo un terminale e lo eseguiamo come: “`sudo nano /etc/inetsim/inetsim.conf`”

Una volta aperta la schermata ci mostrerà una lista di servizi che è in grado di simulare. Possiamo andare ad aggiungere il carattere # accanto a tutti i servizi che non vogliamo attivare al suo avvio. Poi salviamo con Ctrl+O → Invio, e per chiudere Ctrl+X.



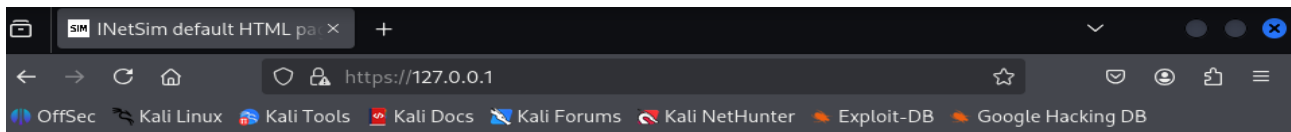
```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.4 /etc/inetsim/inetsim.conf
# ftps, irc, https
#
start_service dns
start_service http
start_service https
#start_service smtp
#start_service smtps
#start_service pop3
#start_service pop3s
#start_service ftp
#start_service ftps
#start_service tftp
#start_service irc
#start_service ntp
#start_service finger
#start_service ident
##start_service syslog
#start_service time_tcp
#start_service time_udp
#start_service daytime_tcp
#start_service daytime_udp
#start_service echo_tcp
#start_service echo_udp
#start_service discard_tcp
#start_service discard_udp
#start_service quotd_tcp
#start_service quotd_udp
#start_service chargen_tcp
#start_service chargen_udp
#start_service dummy_tcp
#start_service dummy_udp
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

Ora eseguendo Inetsim tramite il comando `Sudo Inetsim` eseguiremo la simulazione. Per default utilizzerà il Localhost 172.0.0.1.



```
[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 12613) ===
Session ID: 12613
Listening on: 127.0.0.1
Real Date/Time: 2025-10-18 03:26:30
Fake Date/Time: 2025-10-18 03:26:30 (Delta: 0 seconds)
Forking services ...
* https_443_tcp - started (PID 12625)
* dns_53_tcp_udp - started (PID 12623)
Can't locate object method "main_loop" via package "Net::DNS::Nameserver" at /usr/share/perl5/INetSim/DNS.pm line 69
* http_80_tcp - started (PID 12624)
done.
Simulation running.
```

Una volta avviato la simulazione possiamo andare a verificarne il funzionamento cercando nel browser l'indirizzo <https://127.0.0.1/>, verificando che il servizio sia attivo, e raggiungibile dalla macchina.



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

This file is an HTML document.

Infine, grazie a Wireshark possiamo eseguire uno sniffing dei pacchetti in entrata e in uscita sulla nostra macchina. Avviando Wireshark e chiedendo di mostrare tutti i pacchetti in Loopback riusciremo a vedere tutto il traffico, tra cui la nostra comunicazione con il servizio Https. Saremo in grado di vedere sempre il mittente e il destinatario, ma non il contenuto in quanto il servizio Https critta i nostri dati.

