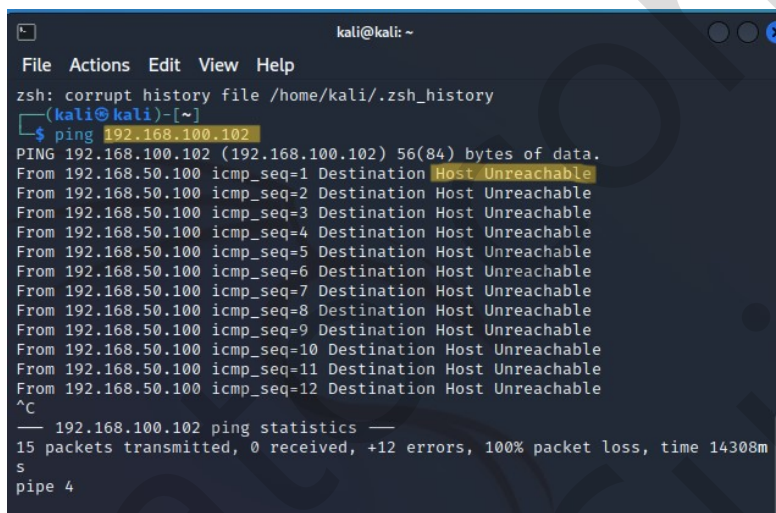


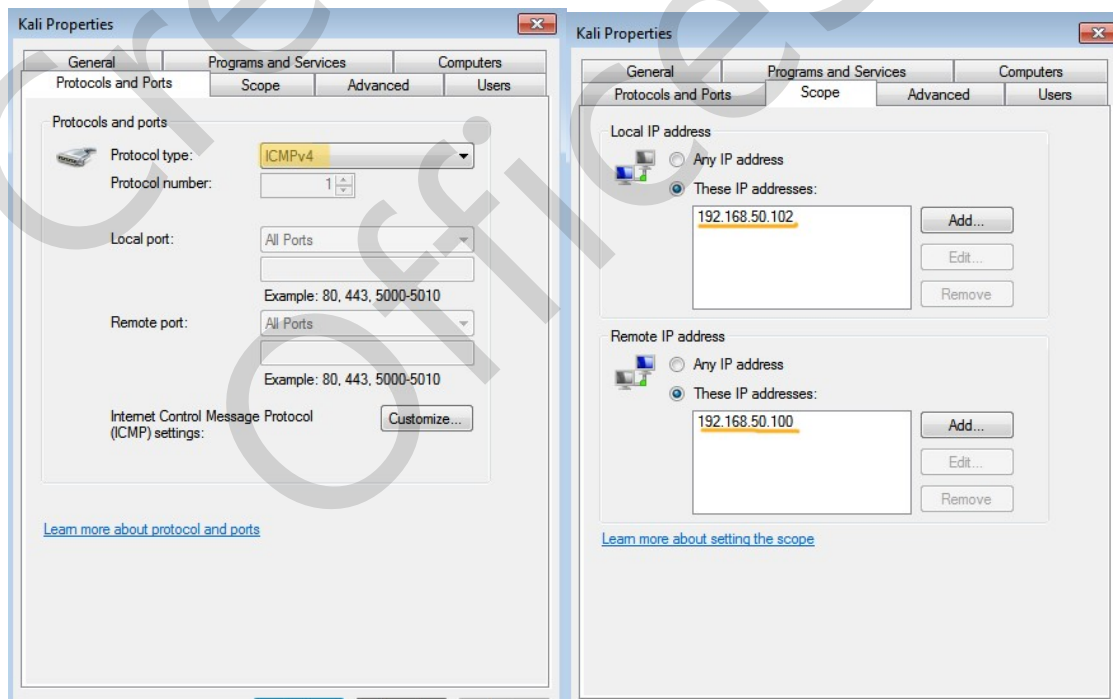
## Configurare policy su firewall di windows & Intercettare pacchetti tramite Wireshark in Kai linux

### Step 1



```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ ping 192.168.100.102  
PING 192.168.100.102 (192.168.100.102) 56(84) bytes of data.  
From 192.168.50.100 icmp_seq=1 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=2 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=3 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=4 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=5 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=6 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=7 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=8 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=9 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=10 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=11 Destination Host Unreachable  
From 192.168.50.100 icmp_seq=12 Destination Host Unreachable  
^C  
— 192.168.100.102 ping statistics —  
15 packets transmitted, 0 received, +12 errors, 100% packet loss, time 14308m  
s  
pipe 4
```

Come primo step vediamo come non il firewall attivo di windows , Kai linux non riesce a pingare.



Vado quindi a creare una nuova regola nel Firewall di windows assegnando il protocollo ICMPv4 e in “scope” come Local address l’Ip di windows e come Remote address l’Ip di Linux.

A questo punto faccio un controllo per vedere se le macchine pingano tra loro con successo.

```
(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=0.779 ms  
64 bytes from 192.168.50.102: icmp_seq=2 ttl=128 time=0.464 ms  
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=0.421 ms  
64 bytes from 192.168.50.102: icmp_seq=4 ttl=128 time=0.399 ms  
^C  
— 192.168.50.102 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3055ms  
rtt min/avg/max/mdev = 0.399/0.515/0.779/0.153 ms
```

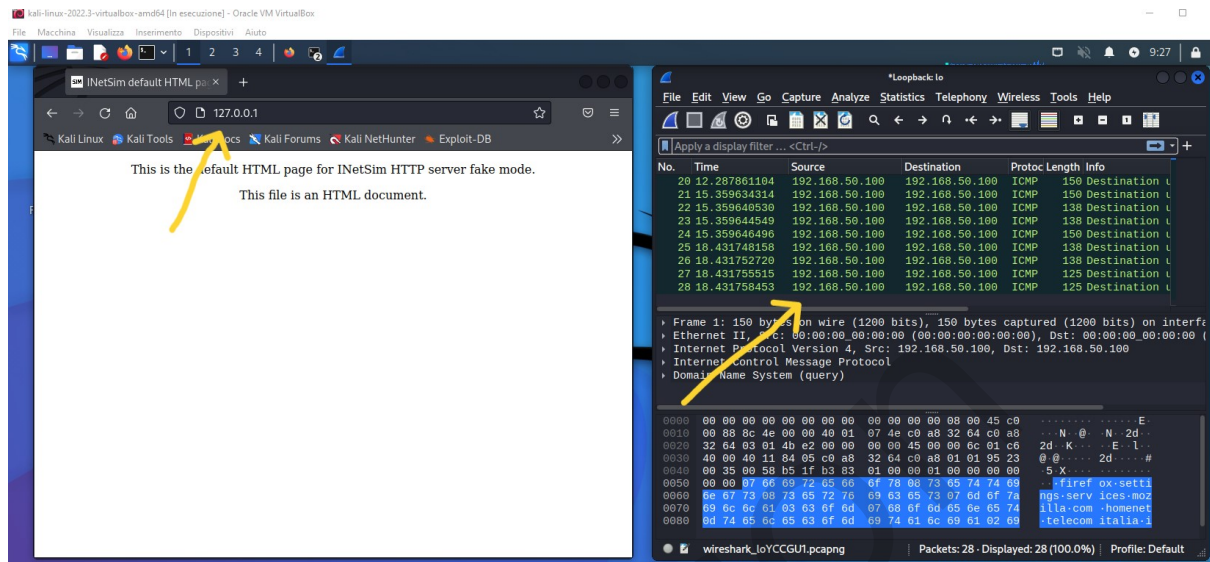
```
(kali㉿kali)-[~]  
$
```

```
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
  
C:\Users\Claudio>ping 192.168.50.100  
  
Pinging 192.168.50.100 with 32 bytes of data:  
Reply from 192.168.50.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.50.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.50.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.50.100: bytes=32 time<1ms TTL=64  
  
Ping statistics for 192.168.50.100:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms  
  
C:\Users\Claudio>
```

## Step 2

```
kali@kali: ~  
File Actions Edit View Help  
$ sudo inetsim  
[sudo] password for kali:  
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg  
Main logfile '/var/log/inetsim/main.log' does not exist. Trying to create it.  
..  
Main logfile '/var/log/inetsim/main.log' successfully created.  
Sub logfile '/var/log/inetsim/service.log' does not exist. Trying to create i  
t ...  
Sub logfile '/var/log/inetsim/service.log' successfully created.  
Debug logfile '/var/log/inetsim/debug.log' does not exist. Trying to create i  
t ...  
Debug logfile '/var/log/inetsim/debug.log' successfully created.  
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 14545) ==  
Session ID: 14545  
Listening on: 127.0.0.1  
Real Date/Time: 2022-10-27 09:14:15  
Fake Date/Time: 2022-10-27 09:14:15 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 14551)  
* irc_6667_tcp - started (PID 14561)  
* time_37_tcp - started (PID 14566)  
* ntp_123_udp - started (PID 14562)  
* echo_7_udp - started (PID 14571)  
* ident_113_tcp - started (PID 14564)  
* dummy_1_udp - started (PID 14579)  
* discard_9_tcp - started (PID 14572)  
* daytime_13_tcp - started (PID 14568)  
* chargen_19_tcp - started (PID 14576)
```

Come step 2 su kali linux tramite il prompt vado ad avviare Inetsim e faccio partire la simulazione.



Ora apro Firefox e vado a digitare l'ip creato da Inetsim che mi farà visualizzare una pagina HTML.  
A questo punto apro Wireshark e lo faccio partire , possiamo vedere nella parte destra della foto come Wireshark stia intercettando i pacchetti in transito e me li visualizzi nella taabella.