

## **Week 1 - Esercizio 3**

### Configurazione Firewall Windows 7

#### **Consegna**

L'esercizio di oggi mira a consolidare le conoscenze acquisite nella lezione del mattino. Vedremo due esercizi: I) la configurazione di una policy sul firewall windows; II) una packet capture con Wireshark. Vedremo anche come simulare alcuni servizi di rete con un tool pre-installato su Kali Linux (InetSim)

**Esercizio:**

- ☐ Configurare policy per il ping da macchine Linux a Macchina Windows nel nostro laboratorio
- ☐ Utilizzo dell'utility InetSim per l'emulazione di servizi Internet
- ☐ Cattura di pacchetti con Wireshark

#### **1. Configurazione delle Policy del Firewall**

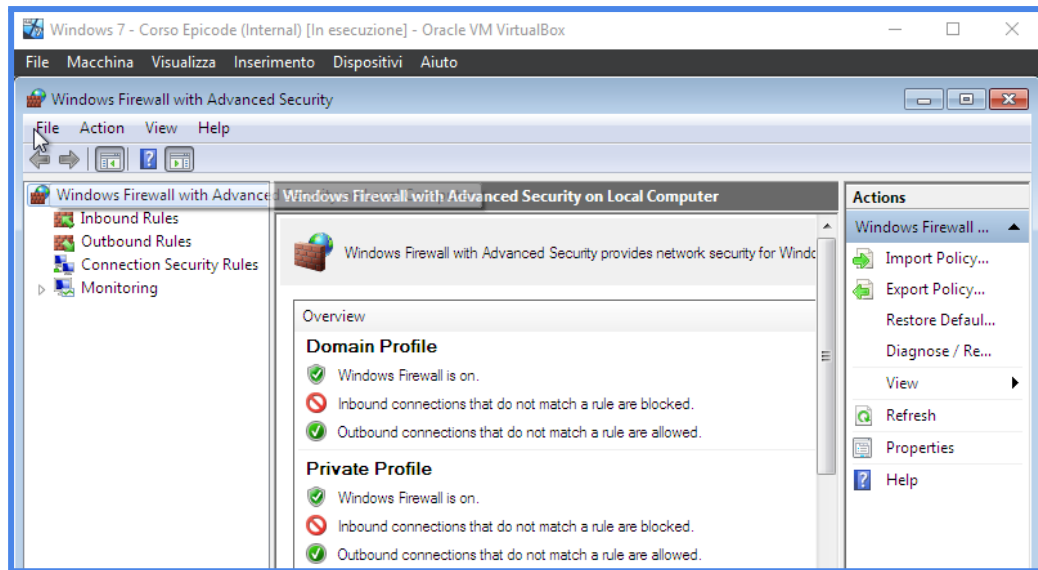
Abbiamo a nostra disposizione 2 macchine virtuali (Kali e Win 7) già configurate (con IP rispettivamente 192.168.50.100 e 192.168.50.102) e configurazione di rete settata in Internal

Se andiamo a far comunicare le due macchine virtuali tramite il comando Ping noteremo che esse non comunicano. O meglio, la macchina con Windows 7 installato non riceve i pacchetti che stiamo spedendo dalla macchina con Kali.

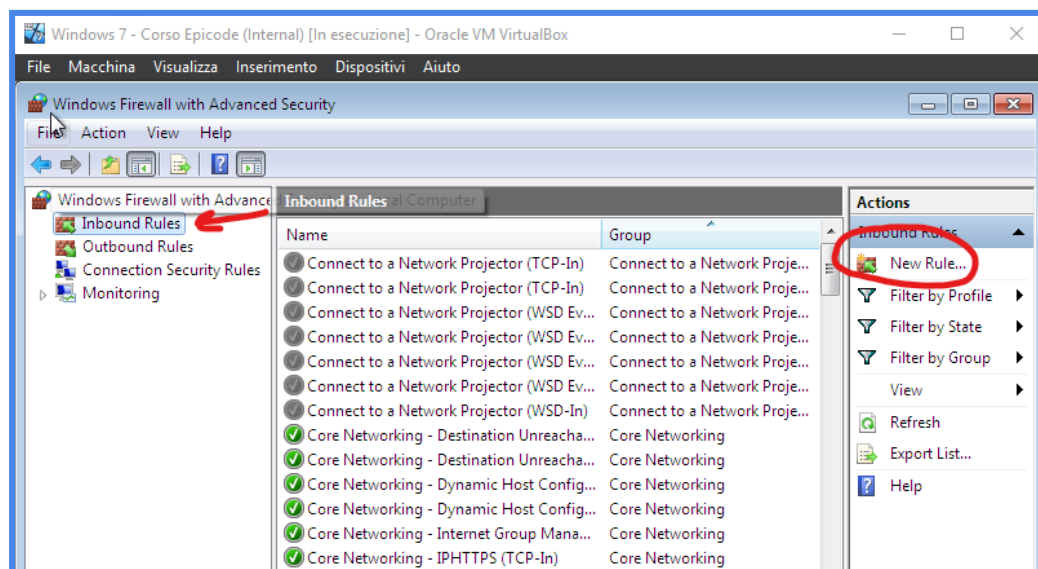
Questo accade a causa del Firewall di Windows 7

#### **Procedimento**

- 1.** Dal momento che il Firewall di Windows sta bloccando i pacchetti inviati dalla macchina Kali, andiamo a controllare le policy del Firewall Windows

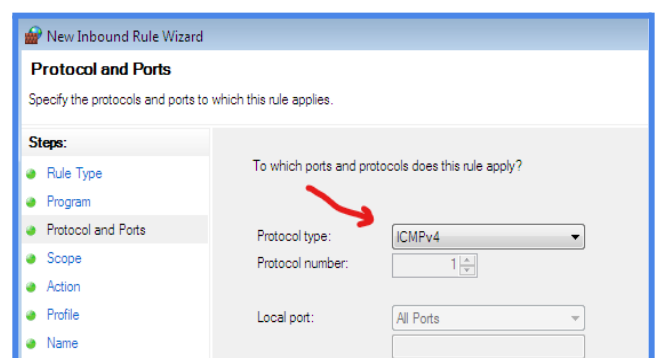


2. Faccio un rapido controllo alle regole attualmente attive in ricezione ("Inbound Rules")



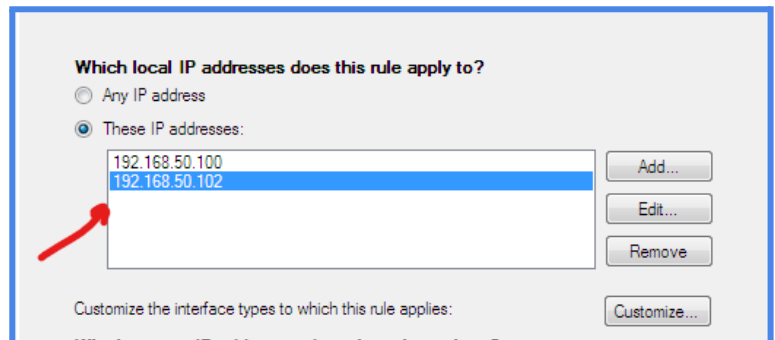
3. Vado poi a creare una nuova regola personalizzata che mirerà a far accettare i pacchetti inviati da Kali

E nella sezione "Protocol and Ports" seleziono **ICMPv4**, ovvero il protocollo dedicato allo scambio di pacchetti di



controllo (esattamente il tipo che stiamo cercando di inviare in questo caso).

Indico poi a quali indirizzi IP si applica questa regola e inserisco gli IP delle 2 macchine



4. Creo la regola che abbiamo appena configurato e verifico che lo scambio di pacchetti tra le 2 macchine avviene correttamente tramite il comando **ping**.  
E notiamo che lo scambio avviene effettivamente senza alcuna perdita di pacchetti.

```
kali@kali: ~  
File 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=0.383 ms  
64 bytes from 192.168.50.102: icmp_seq=2 ttl=128 time=0.448 ms  
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=0.420 ms  
^Z  
zsh: suspended ping 192.168.50.102
```

## 2. Creazione di un Server con Inetsim e cattura dei pacchetti con Wireshark

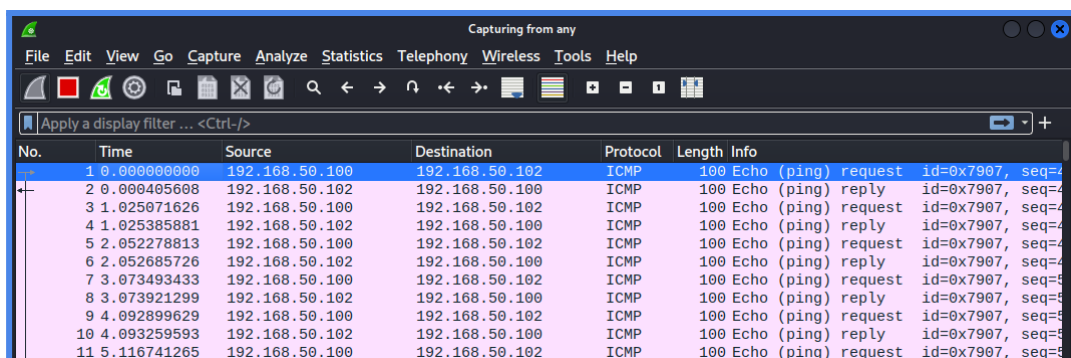
1. Vado ad eseguire il comando **inetsim** (tramite amministratore) per simulare un Server HTTP sulla mia macchina Kali

```
(kali@kali)-[~]  
$ sudo inetsim  
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 it ...  
Sub logfile '/var/log/inetsim/service.log' successfully created.  
Debug logfile '/var/log/inetsim/debug.log' does not exist. Trying to create it ...  
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 8968) ==  
Session ID: 8968  
Listening on: 127.0.0.1  
Real Date/Time: 2022-10-28 06:45:39  
Fake Date/Time: 2022-10-28 06:45:39 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 8974)  
* finger_79_tcp - started (PID 8986)
```

2. Con il comando **ip a** verifichiamo che l'ip della macchina sia quello della simulazione di Server

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:22:46:4f brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.100/24 brd 192.168.50.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe22:464f/64 scope link
        valid_lft forever preferred_lft forever
```

3. Successivamente metto nuovamente in comunicazione la macchina Kali e la macchina Windows attraverso il comando **ping** e vado ad utilizzare **Wireshark** da Kali per fare "Sniffing" ed analizzare lo scambio di dati tra le due macchine
4. Avvio la cattura dei pacchetti e procedo quindi ad analizzarli attraverso Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=4
2	0.000405608	192.168.50.102	192.168.50.100	ICMP	100	Echo (ping) reply id=0x7907, seq=4
3	1.025071626	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=4
4	1.025385881	192.168.50.102	192.168.50.100	ICMP	100	Echo (ping) reply id=0x7907, seq=4
5	2.052278813	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=4
6	2.052685726	192.168.50.102	192.168.50.100	ICMP	100	Echo (ping) reply id=0x7907, seq=4
7	3.073493433	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=5
8	3.073921299	192.168.50.102	192.168.50.100	ICMP	100	Echo (ping) reply id=0x7907, seq=5
9	4.092899629	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=5
10	4.093259593	192.168.50.102	192.168.50.100	ICMP	100	Echo (ping) reply id=0x7907, seq=5
11	5.116741265	192.168.50.100	192.168.50.102	ICMP	100	Echo (ping) request id=0x7907, seq=5