

Report W3D4 – Peppoli

L'esercizio di oggi mira a consolidare le conoscenze acquisite.

Vedremo due esercizi:

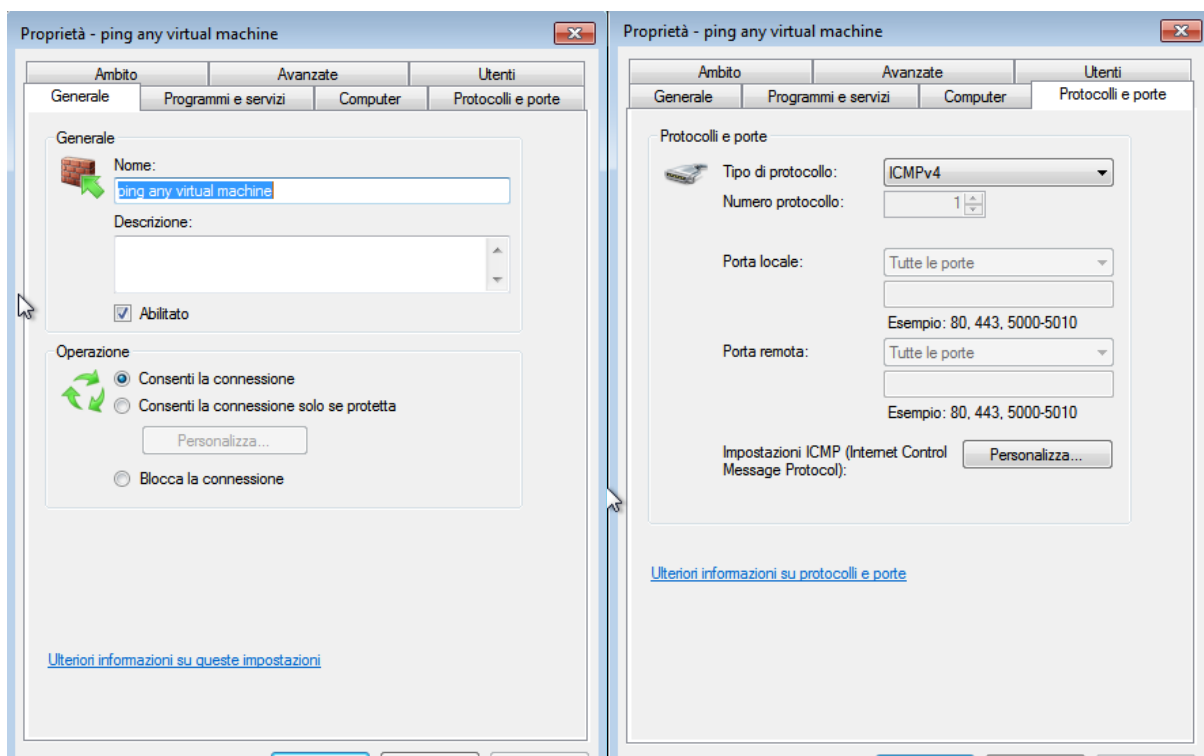
- I) la configurazione di una policy sul firewall windows;
- II) una packet capture con Wireshark.

Vedremo anche come simulare alcuni servizi di rete con un tool pre-installato su Kali Linux (InetSim)

Esercizio:

- Configurare policy per permettere il ping da macchine Linux a Macchina Windows 7 nel nostro laboratorio (Windows firewall);
- Utilizzo dell'utility InetSim per l'emulazione di servizi Internet;
- Cattura di pacchetti con Wireshark.

Configurare policy per permettere il ping da macchine Linux a Macchina Windows 7 nel nostro laboratorio (Windows firewall)

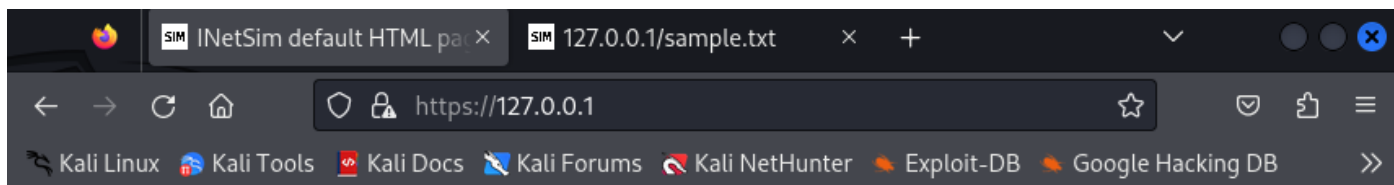


Nel corso dell'esercitazione odierna, abbiamo affrontato la configurazione di una **policy** sul **firewall** di **Windows**. L'obiettivo principale era consentire il ping da macchine Linux a una macchina Windows 7 nel nostro laboratorio. La configurazione è stata effettuata utilizzando le funzionalità del firewall di Windows, garantendo la comunicazione richiesta senza compromettere la sicurezza della rete.

Utilizzo dell'utility InetSim per l'emulazione di servizi Internet

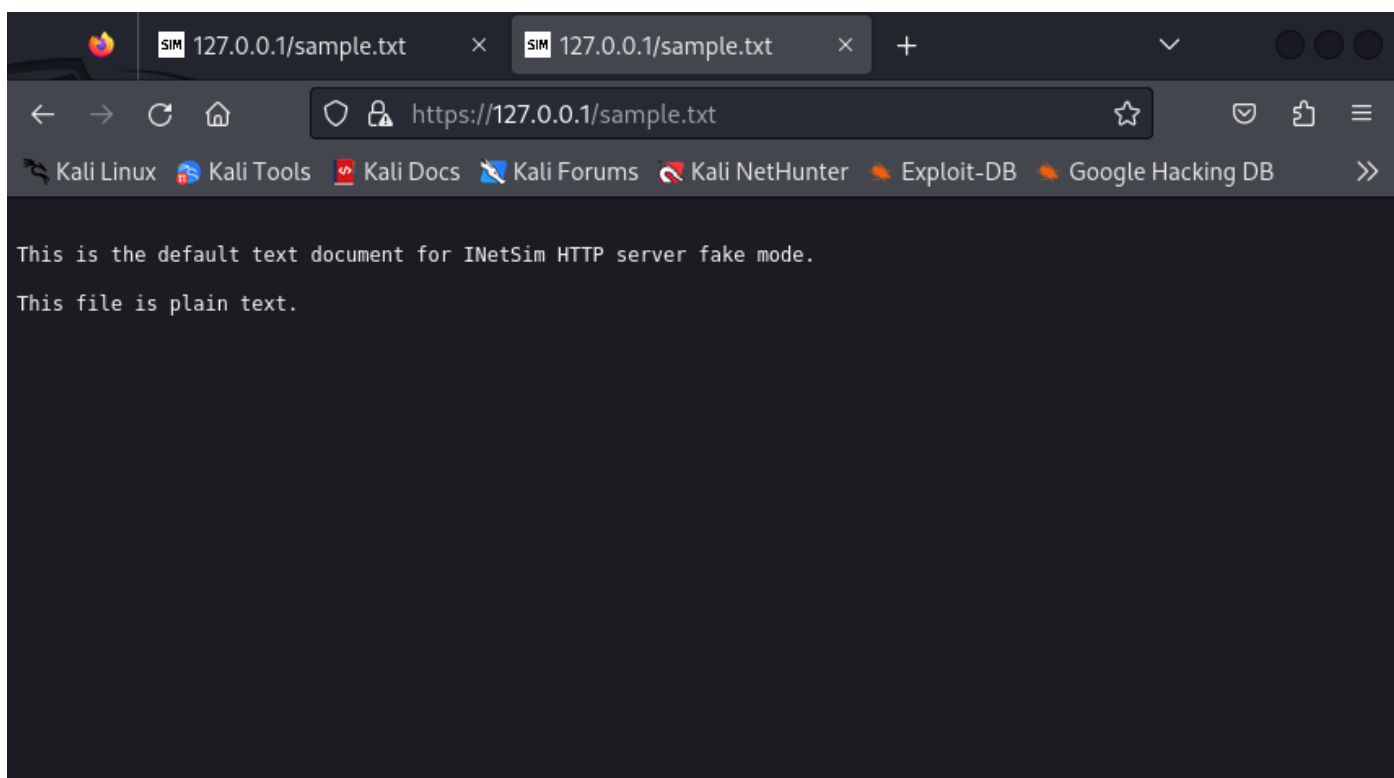
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#####  
#  
# INetSim configuration file  
#  
#####  
  
#####  
# Main configuration  
#####  
  
#####  
# start_service  
#  
# The services to start  
#  
# Syntax: start_service <service name>  
#  
# Default: none  
#  
# Available service names are:  
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,  
# time_udp, daytime_tcp, daytime_udp, echo_tcp,  
# echo_udp, discard_tcp, discard_udp, quotd_tcp,  
# quotd_udp, chargen_tcp, chargen_udp, finger,  
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,  
# 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  
[ Read 1998 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

```
(kali@kali)-[~]  
$ sudo inetsim  
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 14820) ==  
Session ID: 14820  
Listening on: 127.0.0.1  
Real Date/Time: 2023-12-16 12:35:13  
Fake Date/Time: 2023-12-16 12:35:13 (Delta: 0 seconds)  
Forking services ...  
* https_443_tcp - started (PID 14833)  
done.  
Simulation running.  
█
```



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

This file is an HTML document.



Abbiamo anche esplorato l'utilizzo del tool preinstallato su Kali Linux chiamato **InetSim**.

InetSim è uno strumento utilizzato per simulare servizi di rete come **HTTP** e **HTTPS**. È progettato per creare un ambiente di test in cui è possibile simulare il comportamento di questi servizi senza connettersi effettivamente a Internet. Questo è particolarmente utile per scopi di test, addestramento, o per analizzare il comportamento di programmi o sistemi in un ambiente controllato.

InetSim offre una vasta gamma di servizi simulati, tra cui:

HTTP e HTTPS: Per simulare il comportamento dei server web e le richieste di navigazione.

DNS: Per simulare risposte DNS e testare la risoluzione dei nomi.

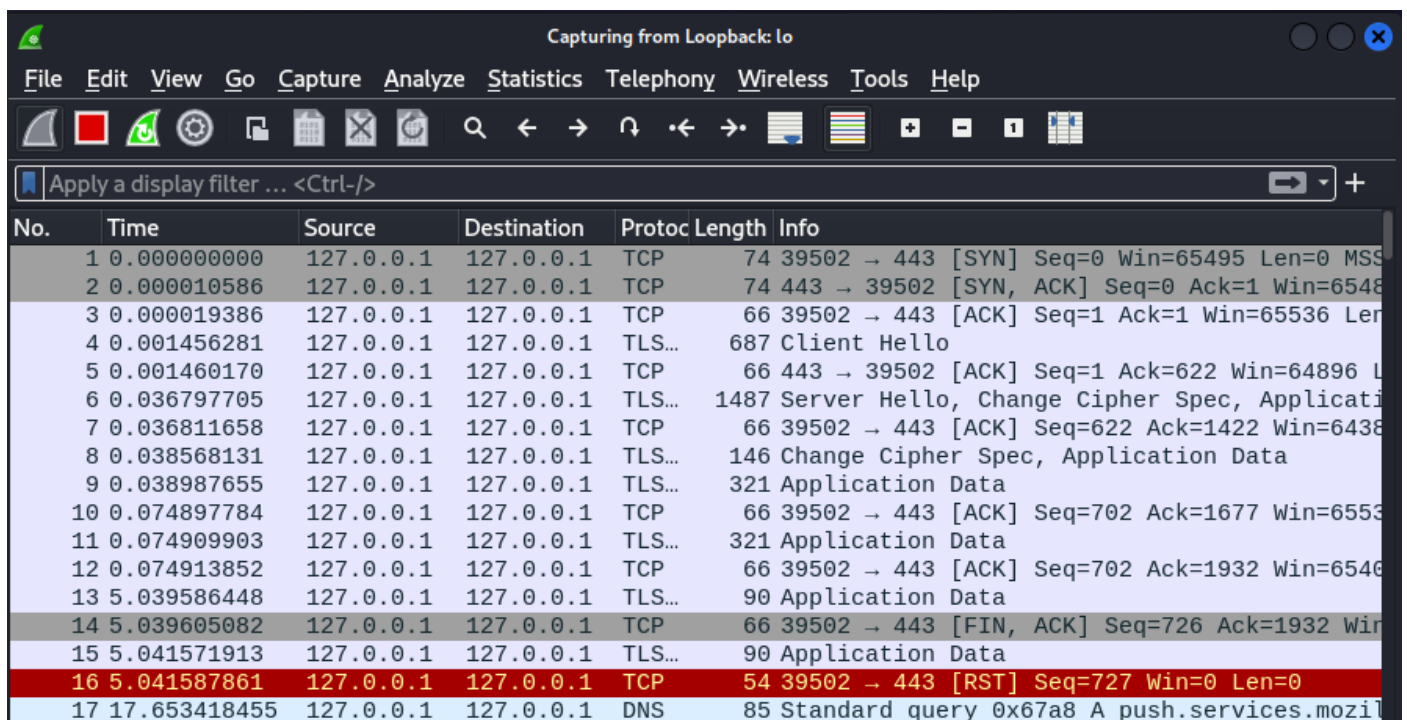
FTP: Per simulare un server FTP e consentire il download e l'upload di file.

SMTP e POP3: Per simulare server di posta elettronica.

IRC: Per simulare server e canali IRC.

Questo strumento può essere configurato per emulare il comportamento di servizi specifici, generando risposte predefinite o configurabili. Ad esempio, nel caso di HTTPS, InetSim può simulare certificati **SSL/TLS**, risposte del server e interazioni tra client e server. Questo è utile per testare come un'applicazione o un sistema si comporterebbe con diverse risposte da un server HTTPS, senza dover effettivamente connettersi a un server esterno.

Cattura di pacchetti con Wireshark



No.	Time	Source	Destination	Protoc	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	TCP	74	39502 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS
2	0.000010586	127.0.0.1	127.0.0.1	TCP	74	443 → 39502 [SYN, ACK] Seq=0 Ack=1 Win=6548
3	0.000019386	127.0.0.1	127.0.0.1	TCP	66	39502 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len
4	0.001456281	127.0.0.1	127.0.0.1	TLS...	687	Client Hello
5	0.001460170	127.0.0.1	127.0.0.1	TCP	66	443 → 39502 [ACK] Seq=1 Ack=622 Win=64896 L
6	0.036797705	127.0.0.1	127.0.0.1	TLS...	1487	Server Hello, Change Cipher Spec, Applicati
7	0.036811658	127.0.0.1	127.0.0.1	TCP	66	39502 → 443 [ACK] Seq=622 Ack=1422 Win=6438
8	0.038568131	127.0.0.1	127.0.0.1	TLS...	146	Change Cipher Spec, Application Data
9	0.038987655	127.0.0.1	127.0.0.1	TLS...	321	Application Data
10	0.074897784	127.0.0.1	127.0.0.1	TCP	66	39502 → 443 [ACK] Seq=702 Ack=1677 Win=6553
11	0.074909903	127.0.0.1	127.0.0.1	TLS...	321	Application Data
12	0.074913852	127.0.0.1	127.0.0.1	TCP	66	39502 → 443 [ACK] Seq=702 Ack=1932 Win=6546
13	5.039586448	127.0.0.1	127.0.0.1	TLS...	90	Application Data
14	5.039605082	127.0.0.1	127.0.0.1	TCP	66	39502 → 443 [FIN, ACK] Seq=726 Ack=1932 Win
15	5.041571913	127.0.0.1	127.0.0.1	TLS...	90	Application Data
16	5.041587861	127.0.0.1	127.0.0.1	TCP	54	39502 → 443 [RST] Seq=727 Win=0 Len=0
17	17.653418455	127.0.0.1	127.0.0.1	DNS	85	Standard query 0x67a8 A push.services.mozil

No.	Time	Source	Destination	Protoc	Length	Info
1	0.000000000	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=11/2810
2	0.000569990	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=11/2810
3	1.009287471	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=12/3072
4	1.009770505	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=12/3072
5	2.039101892	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=13/3328
6	2.039661484	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=13/3328
7	3.061149918	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=14/3584
8	3.061657234	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=14/3584
9	4.085920061	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=15/3840
10	4.086515686	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=15/3840
11	5.110429605	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=16/4096
12	5.111215638	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=16/4096
13	6.119658883	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=17/4352
14	6.120065080	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=17/4352
15	7.122070170	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=18/4608
16	7.122618095	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=18/4608
17	8.154407849	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=19/4864
18	8.154797843	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=19/4864
19	9.169180964	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0xd8ff, seq=20/5120
20	9.169681143	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0xd8ff, seq=20/5120

Durante l'esercitazione, abbiamo utilizzato **Wireshark** per effettuare una **packet capture**. La cattura di pacchetti ci ha fornito un'analisi dettagliata del traffico di rete, consentendoci di identificare eventuali anomalie o comportamenti sospetti. Wireshark si conferma un potente strumento di analisi del traffico di rete, contribuendo alla sicurezza complessiva del sistema.