

W3D4 Policy & Packet Capture

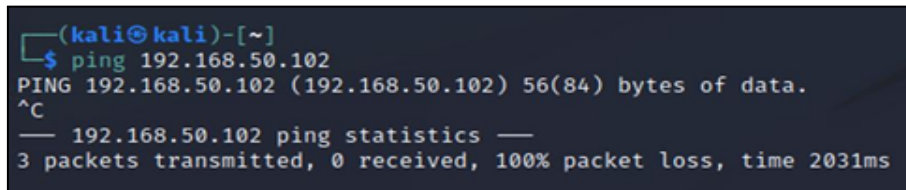
Obiettivo dell'esercitazione:

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

Facoltativo:

- Simulare altri servizi con InetSim;
- Procedere con lo sniffing delle comunicazioni;
- Analizzare il contenuto dei pacchetti.

Durante la configurazione del laboratorio, siamo riusciti a mettere in comunicazione le macchine Linux, ma non siamo riusciti ad effettuare il ping sulla macchina Windows. Ciò si verifica perché, per impostazione predefinita, la macchina virtuale blocca le richieste ping in entrata. Per consentire il ping, è necessario modificare le regole del firewall. Prima di procedere, verifichiamo che tutte le configurazioni di rete siano corrette, quindi eseguiamo un comando ping dal terminale di Kali Linux verso l'indirizzo IP 192.168.50.102, assegnato staticamente alla macchina Windows. Come previsto, le statistiche mostrano la perdita di tutti i pacchetti trasmessi.



```
(kali@kali)-[~]  
$ ping 192.168.50.102  
PING 192.168.50.102 (192.168.50.102) 56(84) bytes of data.  
^C  
— 192.168.50.102 ping statistics —  
3 packets transmitted, 0 received, 100% packet loss, time 2031ms
```

Attivazione Windows Firewall:

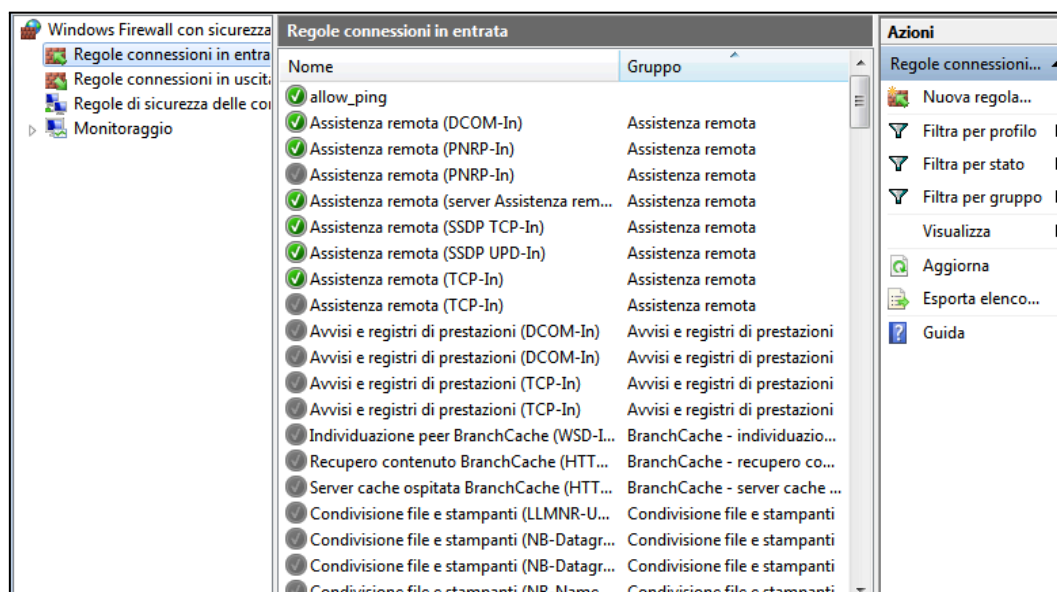
1. Avviare il servizio Windows Firewall
 - Aprire il menu *Start* e cercare "*Servizi*"
 - Nella finestra dei servizi, trovare "*Windows Firewall*"
 - Fare doppio clic sul servizio e impostare il tipo di avvio su *Manuale*.
 - Cliccare su *Applica*, poi su *Avvia* per attivare il servizio.
2. Abilitare il Firewall dalle impostazioni
 - Aprire il *Pannello di controllo* e navigare in:
Pannello di controllo → Tutti gli elementi del Pannello di controllo → Windows Firewall → Personalizza impostazioni
 - Selezionare "*Attiva Windows Firewall*" per le reti private e pubbliche

Dopo questi passaggi, Windows Firewall sarà attivo e funzionante.

A questo punto, dal menu Start, cerchiamo "Windows Firewall" per aprire le impostazioni avanzate. Selezioniamo la sezione "Regole in entrata" (Inbound Rules), che definisce le policy per la gestione del traffico in arrivo sulla macchina. A differenza delle "Regole in uscita" (Outbound Rules), che controllano il traffico generato dalla macchina, queste regole determinano quali connessioni in ingresso vengono consentite o bloccate.

Ricordiamo che il firewall utilizza un approccio **top-down** per l'elaborazione del traffico, analizzando le regole nell'ordine in cui sono elencate. In pratica, scorre il policy set partendo dalla prima regola e si interrompe non appena trova una regola che corrisponde al flusso in esame. L'azione associata alla regola determinerà se il traffico verrà consentito o bloccato. Per consentire il **ping** dalla macchina Kali Linux alla macchina Windows, aggiungiamo una regola firewall in cima al policy set che permetta questo tipo di comunicazione.

1. Aggiungere una nuova regola:
 - Nel pannello a destra, fare clic su *"Nuova regola" (New Rule)*
2. Selezionare il tipo di regola:
 - Scegliere l'opzione *"Personalizzata" (Custom)* e procedere
3. Definire l'ambito della regola:
 - Poiché l'obiettivo è accettare il ping da una sorgente esterna, il tipo di programma che lo invia non è rilevante. Selezionare *"Tutti i programmi" (All programs)* e andare avanti
4. Scegliere il protocollo:
 - Selezionare ICMPv4
5. Impostare gli indirizzi IP:
 - Per il momento, lasciare *"Qualsiasi indirizzo IP" (Any)* sia per l'indirizzo locale che per quello remoto
6. Definire l'azione della regola
 - Selezionare *"Consenti la connessione" (Allow the connection)* e abilitare tutti i profili di applicazione: Dominio, Privato, Pubblico
7. Assegnare un nome alla regola
 - Inserire il nome **"allow_ping"**, aggiungere una breve descrizione e completare la configurazione cliccando su *Fine*



Torniamo sulla macchina Kali Linux per eseguire un secondo test. Questa volta, tutte le richieste ping sono state correttamente ricevute dalla destinazione.

```
(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=0.973 ms
64 bytes from 192.168.50.102: icmp_seq=2 ttl=128 time=1.05 ms
64 bytes from 192.168.50.102: icmp_seq=3 ttl=128 time=0.465 ms
64 bytes from 192.168.50.102: icmp_seq=4 ttl=128 time=0.523 ms
^C
--- 192.168.50.102 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3011ms
rtt min/avg/max/mdev = 0.465/0.753/1.053/0.261 ms
```

Nella seconda parte dell'esercitazione utilizzeremo **Wireshark** per catturare e analizzare i pacchetti di rete. Per supportare questa attività, sfrutteremo **InetSim**, un tool pre-installato su Kali Linux, che consente di simulare servizi di rete come HTTP/HTTPS e FTP, permettendoci di emulare una connessione e studiarne il comportamento.

InetSim, essendo un simulatore di servizi internet già integrato in **Kali Linux**, può essere avviato semplicemente eseguendo il comando "inetsim" nel terminale. Tuttavia, eseguendolo senza configurazione, InetSim avvia numerosi servizi che non sono necessari per la nostra analisi. Pertanto, vedremo come configurarlo per attivare solo i servizi essenziali.

1. Aprire il file di configurazione di InetSim:
 - Avviare un terminale e digitare il comando: `sudo nano /etc/inetsim/inetsim.conf`
2. Osservare i servizi predefiniti:
 - Nella configurazione predefinita, InetSim emula diversi servizi, tra cui DNS, HTTP, HTTPS, FTP, FTPS, IRC, e molti altri
3. Disattivare i servizi non necessari:
 - Per attivare solo HTTPS, è necessario disabilitare tutti gli altri servizi
4. Commentare le righe non necessarie:
 - Aggiungere il carattere "#" all'inizio di ogni riga corrispondente a un servizio che non deve essere avviato
 - Le righe commentate non verranno eseguite, impedendo l'attivazione dei relativi servizi
5. Salvare e chiudere il file
 - Premere CTRL + X per uscire dall'editor
 - Confermare con Y e premere Invio per salvare le modifiche

Per rendere la simulazione più realistica, InetSim fornisce fake file, ossia file vuoti con estensioni definite, che possono essere richiesti e trattati come risorse reali.

```
# 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
```

http_fakefile	txt	sample.txt	text/plain
http_fakefile	htm	sample.html	text/html
http_fakefile	html	sample.html	text/html
http_fakefile	php	sample.html	text/html
http_fakefile	gif	sample.gif	image/gif
http_fakefile	jpg	sample.jpg	image/jpeg
http_fakefile	jpeg	sample.jpg	image/jpeg
http_fakefile	png	sample.png	image/png
http_fakefile	bmp	sample.bmp	image/x-ms-bmp
http_fakefile	ico	favicon.ico	image/x-icon
http_fakefile	exe	sample_gui.exe	x-msdos-program
http_fakefile	com	sample_gui.exe	x-msdos-program

Fake file

Attivazione https

6. Avviare InetSim con la nuova configurazione

- Dopo aver modificato il file, avviare InetSim con il comando: `sudo inetsim`
- A questo punto, solo il servizio HTTPS sarà attivo

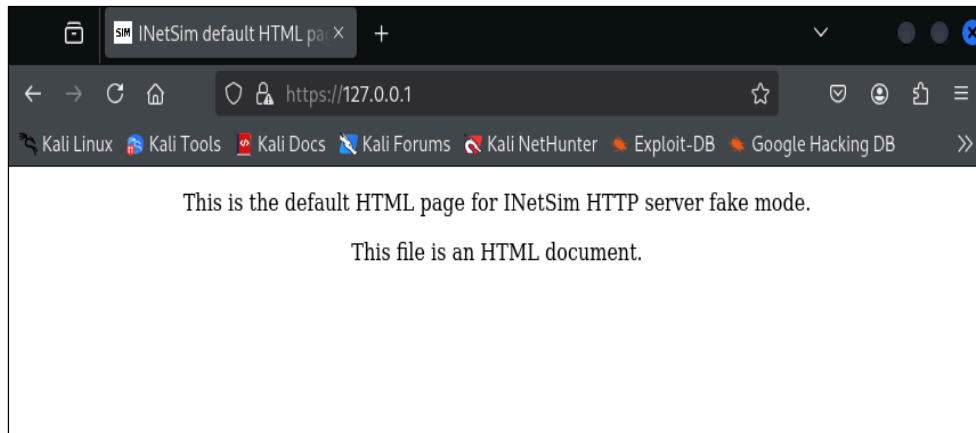
```
(kali㉿kali)-[~]
$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 13328) ==
Session ID:      13328
Listening on:    127.0.0.1
Real Date/Time:  2025-03-17 11:46:09
Fake Date/Time:  2025-03-17 11:46:09 (Delta: 0 seconds)
Forking services...
* https_443_tcp - started (PID 13330)
done.
Simulation running.
```

Il servizio HTTPS è in ascolto sulla porta 443 del localhost

Eseguire un controllo della configurazione tentando una connessione alla porta 443 del **localhost** tramite un web browser su Kali Linux.

1. Aprire il browser e digitare nella barra degli indirizzi: <https://127.0.0.1>
2. Accettare l'avviso di sicurezza visualizzato
3. Se la configurazione è corretta, verrà mostrata la pagina iniziale fittizia di InetSim

La visualizzazione di questa pagina conferma che il servizio è attivo, disponibile e correttamente raggiungibile sul **localhost** della macchina.

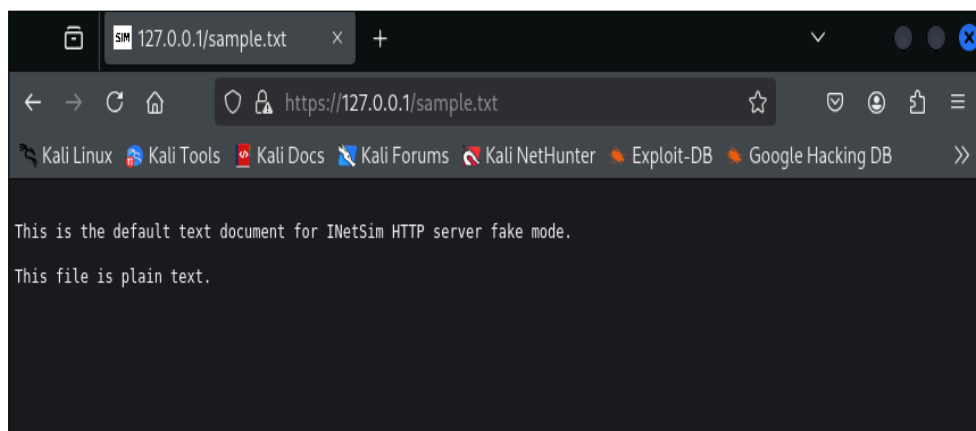


Default HTML page for InetSim HTTP

Per effettuare un ulteriore test, proviamo a richiedere uno dei fake file forniti da InetSim:

1. Aprire il web browser e digitare nella barra degli indirizzi: <https://127.0.0.1/sample.txt>
2. Se la configurazione è corretta, il browser restituirà un file **.txt** fittizio generato da InetSim

Questo conferma che il servizio è attivo e in grado di rispondere alle richieste.



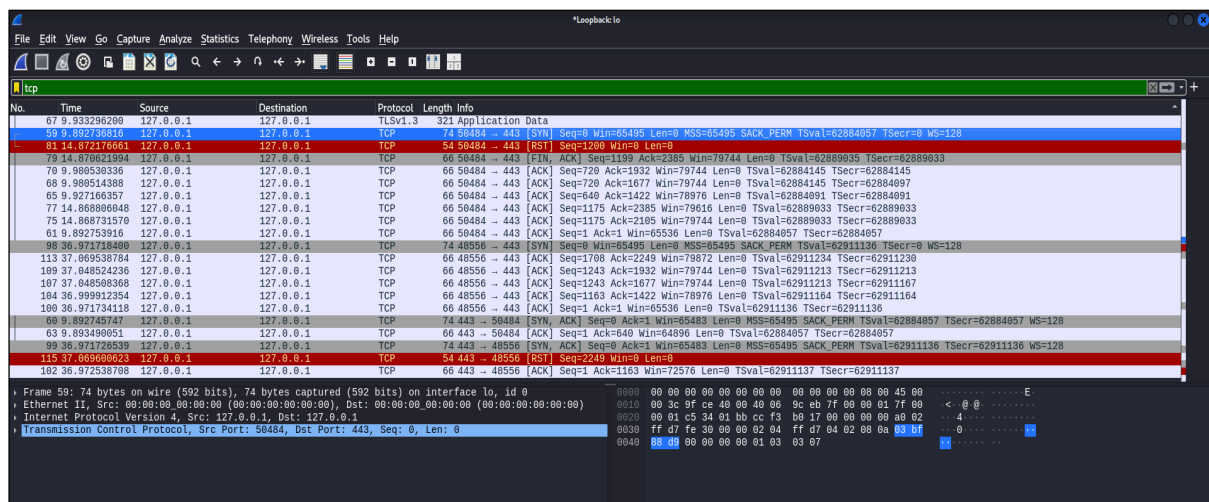
file .txt fittizio

Avviamo Wireshark e impostiamo l'ascolto sull'interfaccia di loopback per monitorare il traffico locale. Successivamente, ci connettiamo al localhost utilizzando un web browser.

Analizzando il traffico catturato, possiamo osservare l'instaurazione della connessione TCP tra client e server. In particolare, viene eseguita la procedura di 3-way handshake, composta dai seguenti passaggi:

1. Il client invia un pacchetto SYN per avviare la comunicazione
2. Il server risponde con SYN + ACK per confermare la richiesta
3. Il client conclude il processo inviando un ACK

Questa sequenza conferma l'avvenuta instaurazione della sessione TCP tra le due parti.



Cattura del traffico di rete in Wireshark

Parte Facoltativa

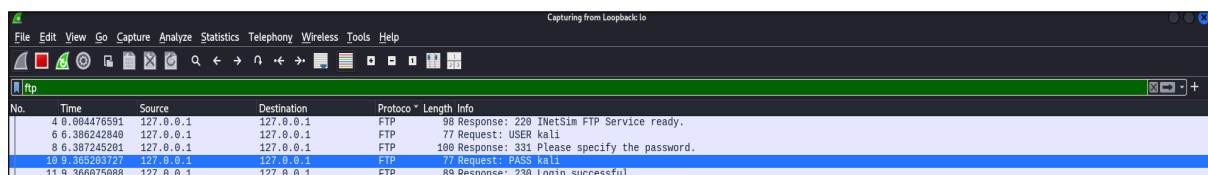
1. Simulare altri servizi con InetSim – Configurare InetSim per emulare ulteriori servizi di rete e analizzarne il comportamento
2. Sniffing delle comunicazioni – Utilizzare Wireshark per intercettare e monitorare il traffico generato dai servizi simulati
3. Analisi dei pacchetti – Esaminare il contenuto dei pacchetti catturati per comprendere il flusso delle comunicazioni e i dati trasmessi tra client e server

Per la parte facoltativa dell'esercizio ho deciso di configurare InetSim per emulare il servizio FTP.

```
# 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
```

Configurazione servizio FTP

Cattura del traffico FTP in Wireshark – Analisi dei pacchetti di comunicazione tra il client e il server FTP emulato con InetSim. La foto mostra i principali comandi e risposte FTP, come la richiesta di login (USER, PASS).

A screenshot of the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The main display area shows a list of captured packets. The filter bar at the top of the packet list is set to 'ftp'. The packet list contains five entries, all of which are FTP packets between source IP 127.0.0.1 and destination IP 127.0.0.1. The first packet (No. 4) is an FTP response (code 220) from InetSim. The second (No. 6) is an FTP request (code 331) for the user 'kall'. The third (No. 8) is an FTP response (code 331) asking for a password. The fourth (No. 10) is an FTP request (code 331) for the password 'kall'. The fifth (No. 11) is an FTP response (code 230) indicating a successful login. The packet details pane on the right shows the structure of the selected packet (No. 10), including the FTP command field.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.004476591	127.0.0.1	127.0.0.1	FTP	98	Response: 220 InetSim FTP Service ready.
6	0.386242840	127.0.0.1	127.0.0.1	FTP	77	Request: USER kall
8	0.387245291	127.0.0.1	127.0.0.1	FTP	100	Response: 331 Please specify the password.
10	0.385203727	127.0.0.1	127.0.0.1	FTP	77	Request: PASS kall
11	0.366075088	127.0.0.1	127.0.0.1	FTP	89	Response: 230 Login successful.

Cattura del traffico FTP con Wireshark

In questa schermata di Wireshark, viene visualizzato un pacchetto FTP contenente il comando USER (nome utente) seguito dalla PASS (password). Questo pacchetto rappresenta la comunicazione tra il client e il server FTP, in cui l'utente invia i propri dati di autenticazione. L'argomento (la password) è visibile in chiaro, dato che il trasferimento non è cifrato. Questo evidenzia la vulnerabilità di FTP nella trasmissione delle credenziali senza protezione.

```
Sequence Number: 12      (relative sequence number)
Sequence Number (raw): 1255868059
[Next Sequence Number: 23      (relative sequence number)]
Acknowledgment Number: 67      (relative ack number)
Acknowledgment number (raw): 3823786320
1000 .... = Header Length: 32 bytes (8)
▶ Flags: 0x018 (PSH, ACK)
Window: 32742
[Calculated window size: 130968]
[Window size scaling factor: 4]
Checksum: 0xfe33 [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
▶ Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
▶ [Timestamps]
▶ [SEQ/ACK analysis]
TCP payload (11 bytes)
▼ File Transfer Protocol (FTP)
  ▼ PASS kali\r\n
    Request command: PASS
    Request arg: kali
  [Current working directory: ]
```

Analisi di un pacchetto FTP contenente la richiesta di login