

ESERCIZIO W3D4

Policy & Packet Capture

Questa esercitazione è suddivisa in due esercizi:

- La configurazione di una policy sul firewall di Windows
- Effettuare una “packet capture” su Kali con il tool Wireshark

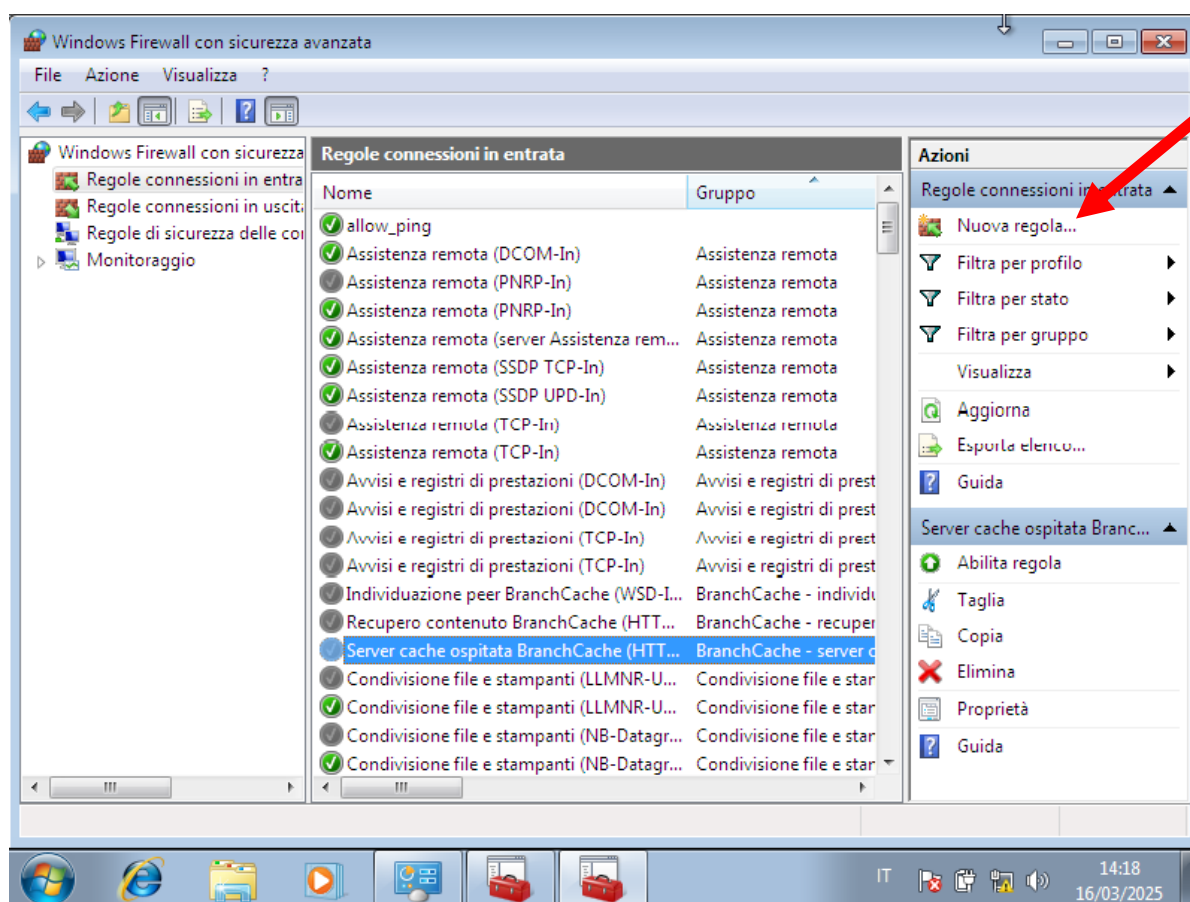
COME CREARE UNA POLICY NEL FIREWALL DI WINDOWS

In uno degli esercizi precedenti, quando abbiamo configurato le connessioni tra le macchine virtuali e le abbiamo testate tramite il comando di *ping*, abbiamo notato come quest'ultimo, effettuato dalle altre macchine verso il sistema Windows, dava esito negativo. Questo è dato dal fatto che le policy del firewall di Windows impediscono di ricevere e rispondere ai *ping* di altri dispositivi.

Di seguito andremo a vedere come creare una nuova policy nel firewall, in modo da permettere il *ping* da parte degli altri sistemi verso Windows.

Accendiamo quindi la nostra macchina virtuale con Windows, una volta avviata nella barra di ricerca cerchiamo *Firewall*, aperta la finestra andiamo nelle *Impostazioni avanzate*

Clicchiamo ora nelle *Regole di connessione in entrata* (dove si trova la lista di policy per il traffico di dati in entrata)

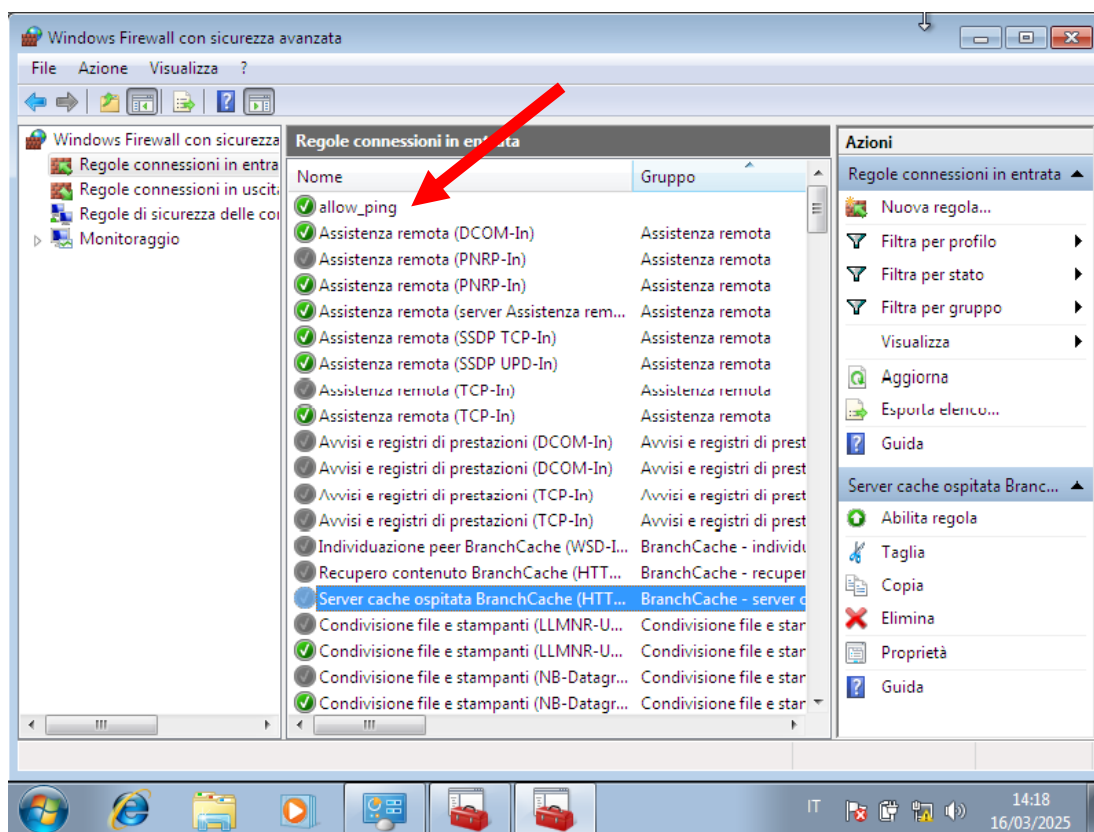


Ora clicchiamo su *Nuova regola* per aggiungere una nuova policy al firewall.

Dalla finestra che apparirà andremo a configurare passo per passo la policy per permettere il *ping* in entrata, quindi:

- Selezioniamo come tipologia di regola Personalizzata
Questa selezione ci permette di selezionare i tipi di protocolli che vogliamo permettere in entrata
- Selezionare Tutti i programmi
Essendo il nostro scopo ricevere un ping da un'altra sorgente, non è un'impostazione che ci interessa
- Selezioniamo dal menu a tendina il protocollo ICMP
Applichiamo la regola a questo protocollo, che è quello utilizzato nel ping
- Nella prossima schermata abbiamo la possibilità di scegliere a quali indirizzi IP applicare la policy, noi selezioneremo Qualsiasi indirizzo IP
- Ora selezioniamo cosa vogliamo che faccia il firewall con le connessioni che specificate nella regola, quindi selezioniamo Consenti la connessione
- Nei profili a cui applicare la regola selezioniamoli tutti: Dominio, Privato e Pubblico
- Infine scegliamo il nome per la nostra regola, come ad esempio allow_ping e eventualmente aggiungiamo una breve descrizione

Cliccando su Fine ora avremo creato la nostra policy, che possiamo vedere nella lista della finestra delle impostazioni del firewall



È importante accertarsi che la nuova regola sia in cima alla lista, in quanto il firewall legge ed applica le policy con il sistema *top-down* attivandole quindi in ordine dalla prima all'ultima della lista.

Ora che la policy è stata configurata non ci resta che testare che funzioni, quindi accediamo su un'altra macchina virtuale del nostro laboratorio e inviamo un comando di *ping* al sistema Windows.

Se tutto è stato configurato correttamente il *ping* avrà esito positivo, quindi la nuova policy funziona a dovere.

PACKET CAPTURE CON WIRESHARK

In questa seconda parte dell'esercizio andremo a simulare, sul sistema Kali, una trasmissione di dati tramite il tool *InetSim* che cattureremo ed analizzeremo tramite il tool *Wireshark*

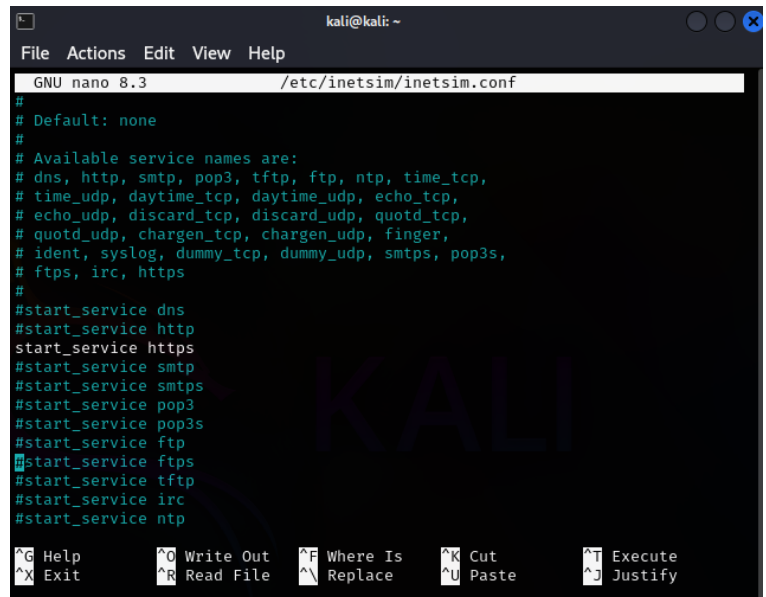
InetSim è un simulatore di servizi internet, preinstallato sul sistema Kali, per il nostro esercizio andremo a configurarlo in modo che possa avviare solo i servizi che vogliamo analizzare, nel nostro caso HTTPS

Per avviare la configurazione del tool digitare nel prompt dei comandi

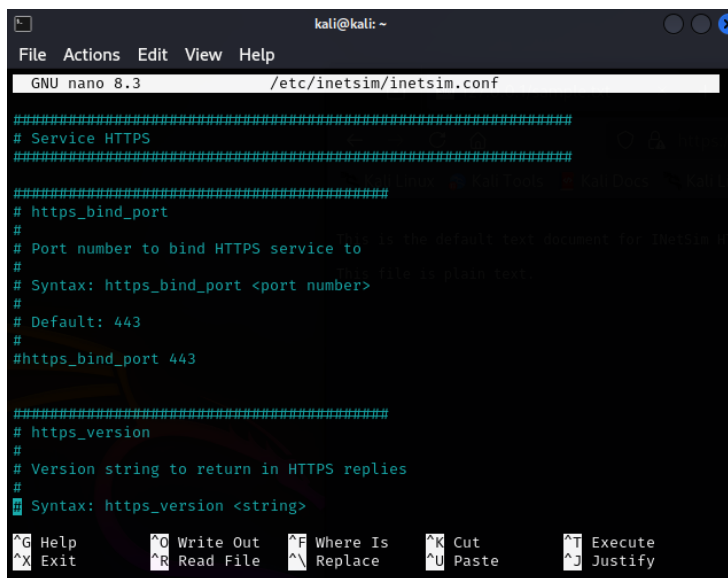
```
sudo nano /etc/inetsim/inetsim.conf
```

L'output del comando mostrerà una lista di tutti i servizi possibili da emulare, noi andremo a lasciare attivo solo HTTPS, quindi per disattivare gli altri, aggiungiamo un simbolo del cancelletto [#] davanti alla loro dicitura, così da renderli "commenti" ed evitare la loro attivazione all'avvio del tool di simulazione.

In immagine notiamo come il servizio HTTPS sia rimasto attivo



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf
#
# 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
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^_ Justify
```



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf
#####
# Service HTTPS
#####
# https_bind_port
#
# Port number to bind HTTPS service to
# Syntax: https_bind_port <port number>
# Default: 443
#
#https_bind_port 443
#####
# https_version
#
# Version string to return in HTTPS replies
# Syntax: https_version <string>
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^_ Justify
```

Scorrendo in basso e arrivando alla voce *Service HTTPS*, abbiamo la possibilità di cambiare le configurazioni del servizio, come ad esempio l'IP o la porta di servizio ed analizzare i file di sample che il tool mette a disposizione per ricreare uno scenario più verosimile di un sito internet.

Lasciamo le impostazioni come trovate, che di default utilizza:

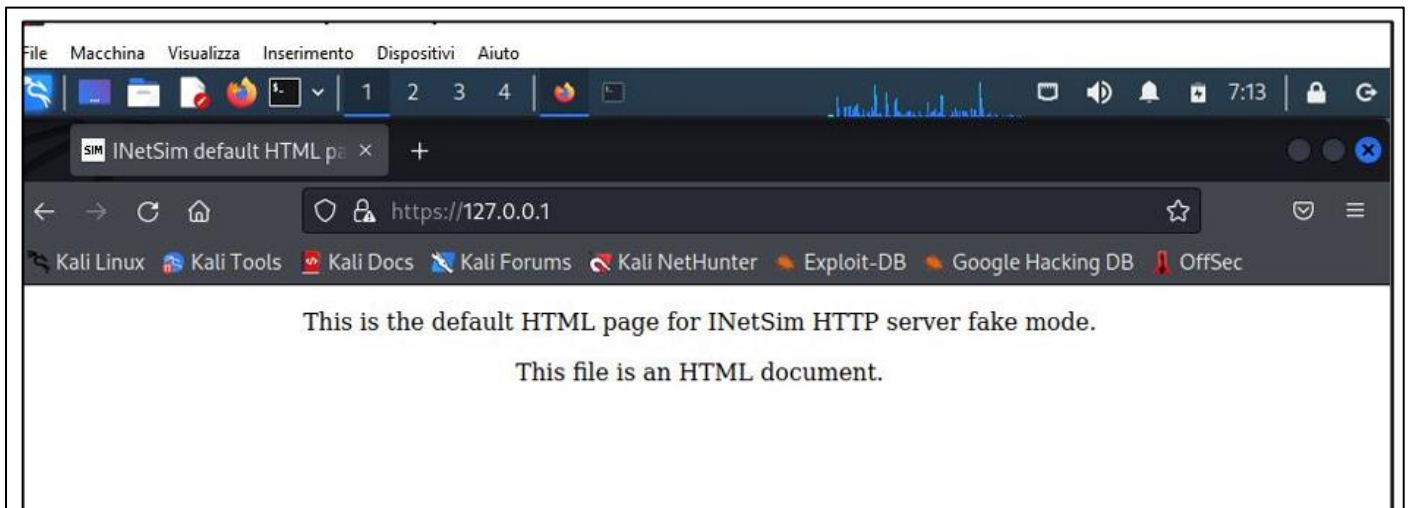
IP: 127.0.0.1

Porta: 443

Salviamo il file delle impostazioni ed avviamo il tool dal terminale utilizzando il comando:

```
sudo inetsim
```

Adesso la simulazione del servizio HTTPS dovrebbe essere avviata, per accertarsi basta aprire il browser e collegarsi all'IP del servizio 127.0.0.1

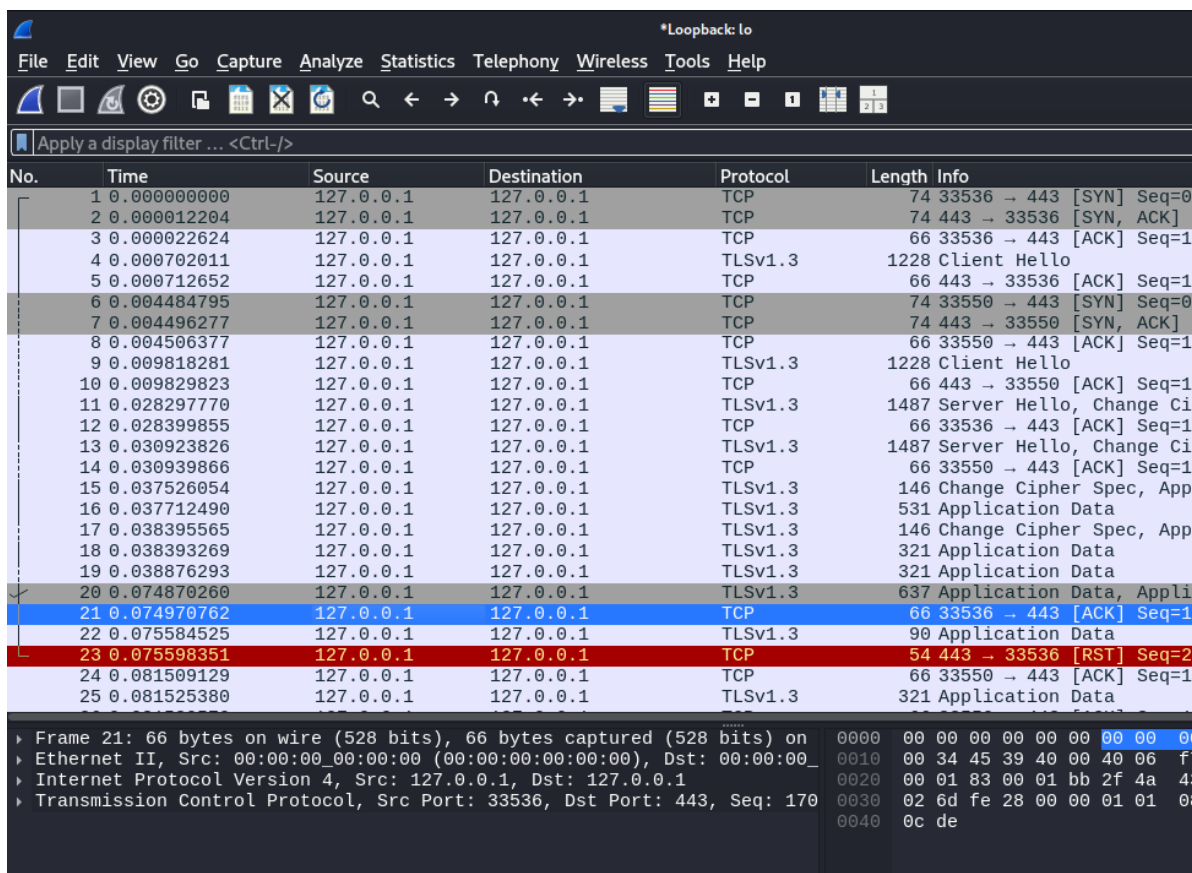


Ci verrà quindi aperta la pagina creata dai file di sample citati sopra, ciò vuol dire che il servizio è attivo e raggiungibile.

Avviamo adesso *Wireshark*, in ascolto sull'interfaccia *loopback*, connettiamoci dal browser e richiediamo ad esempio il file di sample *.txt* digitando nella barra di ricerca

<https://127.0.0.1/sample.txt>

Ritornando su *Wireshark* vedremo la lista di tutti in pacchetti inviati durante la comunicazione.



Wireshark ci permette di selezionare ogni pacchetto di dati ed analizzarlo, nel nostro esempio possiamo vedere come i dati viaggiano utilizzando i protocolli TCP e TLS (tipici del HTTPS), e come la connessione TCP viene instaurata con la sequenza *3 way handshake*.