

Router IP Tables

Indice

- Router IP Tables
 - [Indice](#)
 - [Introduzione](#)
 - [Informazioni sul progetto](#)
 - [Scopo](#)
 - [Analisi](#)
 - [Analisi e specifica dei requisiti](#)
 - [Implementazione](#)
 - [Configurazione macchine virtuali](#)
 - [Router](#)
 - [PC rete interna](#)
 - [Webserver](#)
 - [Impostazione proxy](#)
 - [Impostazione indirizzo ip](#)
 - [Installazione e configurazione Apache](#)
 - [Installazione](#)
 - [Configurazione porte in ascolto](#)
 - [Installazione e configurazione IP Tables](#)
 - [Installazione](#)
 - [Mostrare tutte le regole attive](#)
 - [Default configuration](#)
 - [SSH al router](#)
 - [IP forwarding](#)
 - [Accesso a internet](#)
 - [Port forwarding](#)
 - [Ping dall'interno al router](#)
 - [Regole iptables persistenti](#)
 - [Comandi eseguiti](#)
 - [Test](#)
 - [Protocollo di test](#)
 - [Risultati test](#)
 - [Conclusioni](#)
 - [Considerazioni personali](#)
 - [Sitografia](#)

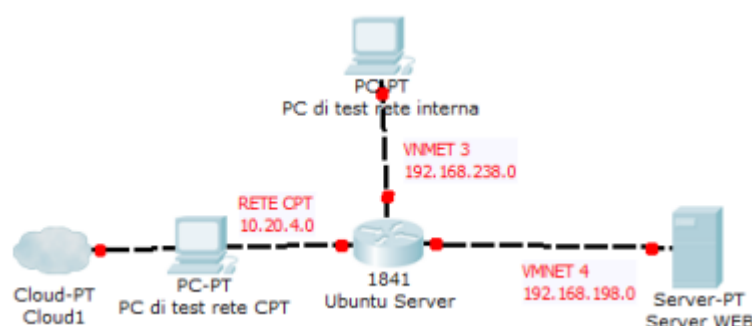
Introduzione

Informazioni sul progetto

- **Titolo:** Questionario patenti
- **Allievi coinvolti nel progetto:**
 - Samuel Banfi, samuelbanfi@samtrevano.ch
 - Dennis Donofrio, dennis.donofrio@samtrevano.ch
- **Classe:** I4AC, Scuola Arti e Mestieri Trevano, sezione Informatica
- **Committente:** Massimo Sartori
- **Data d'inizio:** 24.10.2022
- **Data di fine:** 21.11.2022

Scopo

Lo scopo del progetto **IP Tables** è quello di utilizzare un server Debian che fa da router e gestisce tutte le richieste in entrata e in uscita. Il Server Debian serve anche per dividere tutto in 2 reti. Una interna e una DMZ. Una con indirizzo di rete **192.168.238.0/24** e una con indirizzo di rete **192.168.198.0/24**. Quest'ultima viene utilizzata come DMZ dove mettere il server WEB.



Analisi

Analisi e specifica dei requisiti

ID	REQ-001
Nome	Configurazione VM NET 3
Priorità	1
Versione	1.0

Note Bisogna avere una rete privata con indirizzo di rete **192.168.238.0/24**.
Tutti i computer nella rete possono accedere ad internet.
Non bisogna accedere dall'esterno.

ID	REQ-002
Nome	Configurazione Server Apache
Priorità	1
Versione	1.0

Note Bisognerà avere un server WEB con installato Apache.
Il servizio di Apache dovrà ascoltare sulla porta **443** (HTTPS) e **8080** (interno).
Bisognerà poter accedere alla pagina.

ID	REQ-003
Nome	Configurazione VM NET 4
Priorità	1
Versione	1.0

Note Bisogna avere una rete DMZ con indirizzo di rete **192.168.198.0/24**.
Bisogna avere un server WEB al suo interno (REQ-002).
Bisogna aprire le connessioni in uscita.
Bisogna disabilitare le connessioni in entrata, escluso Apache.
Il server deve essere raggiungibile dall'esterno solo tramite l'IP del router.

ID	REQ-004
Nome	Ping disabilitato
Priorità	1
Versione	1.0
Note	Il ping verso le reti interne deve essere impossibile.

ID	REQ-005
Nome	Accesso tramite SSH
Priorità	1
Versione	1.0
Note	Bisogna rendere possibile l'accesso al router tramite SSH per la configurazione.

ID	REQ-006
Nome	Configurazione IP forwarding
Priorità	1
Versione	1.0
Note	Bisogna abilitare ip_forwarding

ID	REQ-007
Nome	Impostare regole di default per IP Tables
Priorità	1
Versione	1.0
Note	Bisogna impostare le regole di default in modo da avere INPUT e FORWARD bloccati. Invece OUTPUT deve essere aperto.

ID	REQ-008
Nome	Salvataggio regole IP Tables
Priorità	1
Versione	1.0
Note	Bisogna salvare le regole di iptables .

ID	REQ-009
Nome	Caricamento automatico regole IP Tables ad ogni riavvio
Priorità	1
Versione	1.0
Note	Bisogna fare in modo che le regole di iptables vengano caricate automaticamente ad ogni riavvio.

Spiegazione elementi tabella dei requisiti:

ID: identificativo univoco del requisito

Nome: breve descrizione del requisito

Priorità: indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente.

Versione: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l'ultima versione, mentre le vecchie dovranno essere inserite nei diari.

Note: eventuali osservazioni importanti o riferimenti ad altri requisiti.

Implementazione

Configurazione macchine virtuali

Router

Il router è configurato nel seguente modo:

- CPU: 2 core
- RAM: 2 GB
- Rete:
 - NAT network DHCP
 - Internal network lan (192.168.238.1/24)
 - Internal network dmz (192.168.198.1/24)
 - Host-Only DHCP
- Sistema operativo: Debian

PC rete interna

Il PC della rete interna è configurato nel seguente modo:

- CPU: 2 core
- RAM: 2 GB
- Rete:
 - Internal network lan (192.168.238.10/24, 192.168.238.1)
 - Host-Only DHCP
- Sistema operativo: Debian

Webserver

Il webserver è configurato nel seguente modo:

- CPU: 2 core
- RAM: 2 GB
- Rete:
 - Internal network dmz (192.168.198.10/24, 192.168.198.1)
 - Host-Only DHCP
- Sistema operativo: Debian

Impostazione proxy

Durante l'installazione di tutte le macchine virtuali bisogna configurare il proxy impostando l'indirizzo **10.0.2.2:5865**. Il proxy serve solamente all'inizio su tutte le macchine per fare la configurazione iniziale e installare tutti gli aggiornamenti necessari. Per utilizzare successivamente il proxy bisogna modificare il file **/etc/environment** e aggiungere le seguenti variabili d'ambiente:

```
http_proxy="http://10.0.2.2:5865"  
https_proxy="http://10.0.2.2:5865"  
HTTP_PROXY="http://10.0.2.2:5865"  
HTTPS_PROXY="http://10.0.2.2:5865"  
no_proxy=localhost,127.0.0.1
```

L'ultima regola **no_proxy=localhost,127.0.0.1** serve per evitare l'uso del proxy in locale. L'indirizzo ip **10.0.2.2:5865** è quello di **px-py**.

Per essere sicuri del funzionamento del proxy si può usare il comando **curl** per farsi ritornare la pagina html perché questo comando deve passare attraverso il proxy. Senza la configurazione di iptables il seguente comando funziona solo per il router.

```
curl google.com
```

Impostazione indirizzo ip

Visto che stiamo lavorando su **Debian** per modificare l'indirizzo ip delle macchine bisogna modificare il file **/etc/network/interfaces** aggiungendo le varie schede di rete e impostando gli indirizzi ip (**address**), le subnet mask (**netmask**) e i **gateway**. Bisogna inserire quest'ultimo campo solo se quella determinata scheda di rete serve per uscire su internet.

```
auto <interface>  
iface <interface> inet static  
    address <ip_address>  
    netmask <ip_subnet>  
    gateway <ip_gateway>  
    dns-nameservers 8.8.8.8
```

Installazione e configurazione Apache

Installazione

Per installare **Apache 2.4** bisogna eseguire l'**update** per aggiornare tutte le librerie. In seguito bisogna eseguire il comando **sudo apt install apache2 -y** per installare Apache. Attenzione, per installare i pacchetti bisogna eseguire le operazioni come **sudo**, ovvero come **superuser**.

```
sudo apt update
sudo apt install apache2 -y
```

Configurazione porte in ascolto

Per configurare le porte in ascolto da Apache sul server bisogna modificare il file **/etc/apache2/ports.conf** e aggiungere un **Listen** per la porta **8080**. Non serve aggiungerlo per la porta **443** perché è già presente di default come per la porta 80.

```
sudo nano /etc/apache2/ports.conf
```

```
Listen 8080
```


Installazione e configurazione IP Tables

Installazione

Di default **iptables** non è presente su Debian. Quindi va installato tramite **apt**. Una volta installato tutte le regole di default vengono impostate su **accept**.

```
sudo apt install iptables -y
```

Mostrare tutte le regole attive

Per mostrare tutte le regole attive con IP Tables bisogna eseguire il comando **sudo iptables** aggiungendo il parametro **-L** per listare tutte le regole. Usando **iptables** vengono mostrate tutte le regole per l'**IPv4**. Se si volessero vedere le regole per **IPv6** bisogna usare il comando **sudo ip6tables -S**

```
sudo iptables -L
```

Default configuration

Per configurare il router con le impostazioni sicure di default bisogna bloccare tutti i pacchetti in entrata.

```
iptables --policy INPUT DROP  
iptables --policy FORWARD DROP
```

SSH al router

Per potersi collegare al router in ssh bisogna accettare le connessioni ssh all'interfaccia esterna del router.

```
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
```

IP forwarding

L'IP forwarding ha lo scopo di eseguire il forwarding dei pacchetti in uscita verso le altre schede di rete. Per abilitarlo in modo permanente bisogna modificare il file `/etc/sysctl.conf`.

```
sudo nano /etc/sysctl.conf
```

In seguito bisogna decommentare la seguente riga al file per abilitare l'IP forwarding:

```
net.ipv4.ip_forward = 1
```

Accesso a internet

Per poter accedere ad internet dalla lan bisogna impostare le regole di nat. Questo serve per cambiare l'ip e la porta di destinazione.

```
iptables -t nat -A POSTROUTING -s <rete interna>/<maschera> -o  
<interfaccia di uscita> -j MASQUERADE
```

Inoltre bisogna accettare tutte le connessioni dalla rete interna verso l'interfaccia in internet.

```
iptables -A FORWARD -s <rete interna>/<maschera> -i <interfaccia di  
uscita> -j ACCEPT
```

Infine bisogna accettare tutte le connessioni in entrata che hanno lo stato **RELATED** o **ESTABLISHED**.

```
iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

Port forwarding

Per fare il port forwarding dalla porta 443 esterna alla porta 8080 interna bisogna prima cambiare la porta di destinazione.

```
iptables -t nat -A PREROUTING -i <interfaccia in entrata> -p tcp --dport 443 -j DNAT --to-destination <ip interno>:8080
```

Inoltre bisogna accettare tutte le connessioni sulla porta 8080.

```
iptables -A FORWARD -i <interfaccia in entrata> -p tcp --dport 8080 -j ACCEPT
```

Ping dall'interno al router

Per poter effettuare un ping dalla rete interna all'interfaccia del router bisogna accettare le connessioni **ICMP**, ma solo dalla rete interna.

```
iptables -A INPUT -i <interfaccia rete interna> -p icmp -j ACCEPT  
iptables -A OUTPUT -o <interfaccia rete interna> -p icmp -j ACCEPT
```

Regole iptables persistenti

Per rendere le regole di iptables persistenti bisogna installare un pacchetto aggiuntivo.

```
sudo apt-get install iptables-persistent
```

Comandi eseguiti

```
sudo nano /etc/sysctl.conf
net.ipv4.ip_forward = 1
```

```
iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport ssh -j ACCEPT

iptables -t nat -A POSTROUTING -s 192.168.198.0/24 -o enp0s3 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.238.0/24 -o enp0s3 -j MASQUERADE

iptables -A FORWARD -s 192.168.198.0/24 -i enp0s9 -j ACCEPT
iptables -A FORWARD -s 192.168.238.0/24 -i enp0s8 -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 443 -j DNAT --to-
destination 192.168.198.10:8080
iptables -A FORWARD -i enp0s3 -p tcp --dport 8080 -j ACCEPT

iptables -A INPUT -i enp0s9 -p icmp -j ACCEPT
iptables -A OUTPUT -o enp0s9 -p icmp -j ACCEPT
iptables -A INPUT -i enp0s8 -p icmp -j ACCEPT
iptables -A OUTPUT -o enp0s8 -p icmp -j ACCEPT
```

```
sudo nano /etc/environment
http_proxy="http://10.0.2.2:5865"
https_proxy="http://10.0.2.2:5865"
HTTP_PROXY="http://10.0.2.2:5865"
HTTPS_PROXY="http://10.0.2.2:5865"
no_proxy=localhost,127.0.0.1
```

Test

Protocollo di test

Test Case	TC-001
Nome	Test assegnazione IP
Riferimento	REQ-001, REQ-003
Descrizione	Controllare che il webserver abbia un indirizzo nella rete 192.168.198.0. Controllare che il PC di test abbia un indirizzo nella rete 192.168.238.0. Il router invece deve avere tre schede di rete, una con la rete 10.0.2.0, una 192.168.198.0 e una 192.168.238.0
Prerequisiti	-
Procedura	1. Aprire le macchine virtuali. 2. Eseguire il comando <code>ip a</code> . 3. Controllare l'indirizzo IP con la dicitura <code>inet</code> .
Risultati attesi	Il webserver ha un indirizzo nella rete 192.168.198.0. Il PC di test ha un indirizzo nella rete 192.168.238.0. Il router ha tre schede di rete, una con l'indirizzo di rete 10.0.2.0, una 192.168.198.0 e una 192.168.238.0

Test Case	TC-002
Nome	Connessione PC test ad internet
Riferimento	REQ-001
Descrizione	Bisogna verificare che il PC di test nella rete 192.168.238.0 possa accedere ad internet.
Prerequisiti	-
Procedura	1. Avviare il PC di test. 2. Eseguire il comando <code>curl google.com</code> .
Risultati attesi	Viene mostrato il contenuto della pagina HTML di Google.

Test Case	TC-003
Nome	Funzionamento Apache
Riferimento	REQ-002
Descrizione	Bisogna verificare che il servizio di Apache è in ascolto sulla porta 443 (dall'esterno) e sulla 8080 (dall'interno).
Prerequisiti	-
Procedura	<ol style="list-style-type: none">1. Avviare il PC di test (rete interna).2. Eseguire il comando <code>curl 192.168.198.10:8080</code>.3. Usare il PC host.4. Cercare su un browser <code>localhost:443</code>.
Risultati attesi	Viene mostrato il contenuto della pagina HTML del webserver interno.

Test Case	TC-004
Nome	Verifica connessioni in entrata disabilitate
Riferimento	REQ-003
Descrizione	Bisogna verificare che le connessioni in entrata sono disabilitate, escluso Apache.
Prerequisiti	-
Procedura	<ol style="list-style-type: none">1. Usare il PC host.2. Cercare su un browser <code>192.168.198.10:8080</code>.3. Usare il PC host.4. Cercare su un browser <code>localhost:443</code>.
Risultati attesi	<p>Nel primo caso la richiesta fallisce.</p> <p>Nel secondo caso invece viene mostrata la pagina HTML del webserver locale.</p>

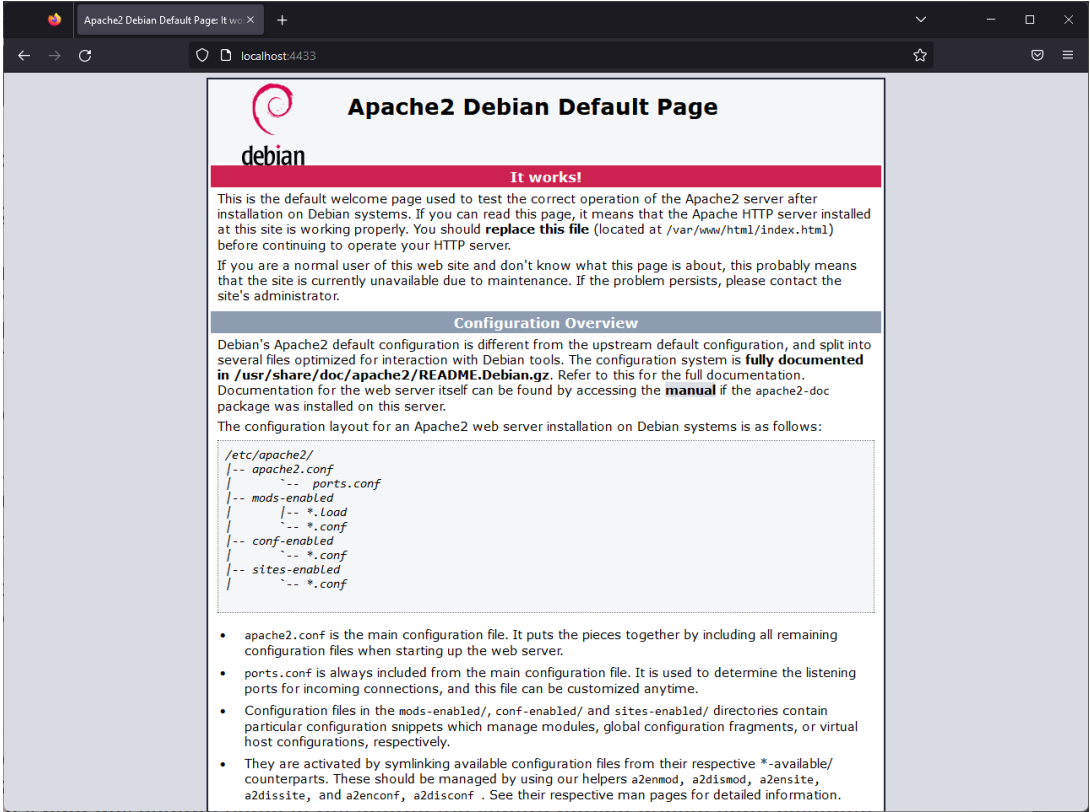
Test Case	TC-005
Nome	Verifica ping impossibile verso le reti interne
Riferimento	REQ-004
Descrizione	Bisogna verificare che il <code>ping</code> verso le reti interne è impossibile.
Prerequisiti	-
Procedura	<ol style="list-style-type: none">1. Usare il PC host.2. Eseguire il comando dal terminale <code>ping 192.168.198.10</code>.
Risultati attesi	I pacchetti inviati con il ping vanno in timeout.

Test Case	TC-006
Nome	Verifica accesso al router tramite SSH
Riferimento	REQ-005
Descrizione	Bisogna verificare l'accesso al router tramite SSH per la configurazione.
Prerequisiti	-
Procedura	1. Usare il PC host. 2. Eseguire il comando dal terminale ssh administrator@10.0.2.6:2222 .
Risultati attesi	Il collegamento SSH avviene correttamente e come utente attuale si vede administrator .
Test Case	TC-007
Nome	Verificare regole di default per IP Tables
Riferimento	REQ-007
Descrizione	Bisogna verificare che le regole di default siano impostate in modo da avere INPUT e FORWARD bloccati. Invece OUTPUT deve essere aperto.
Prerequisiti	-
Procedura	1. Usare il router. 2. Eseguire il comando dal terminale sudo iptables -S .
Risultati attesi	Il comando mostra INPUT e FORWARD bloccati. Invece OUTPUT risulta aperto.
Test Case	TC-008
Nome	Verificare caricamento automatico regole
Riferimento	REQ-009
Descrizione	Bisogna verificare che le regole di IP Tables vengano caricate automaticamente ad ogni riavvio.
Prerequisiti	REQ-008
Procedura	1. Usare il router. 2. Caricare le regole di IP Tables. 3. Riavviare il router. 4. Controllare le configurazioni con il comando iptables -S
Risultati attesi	Vengono mostrate tutte le regole aggiunte in precedenza.

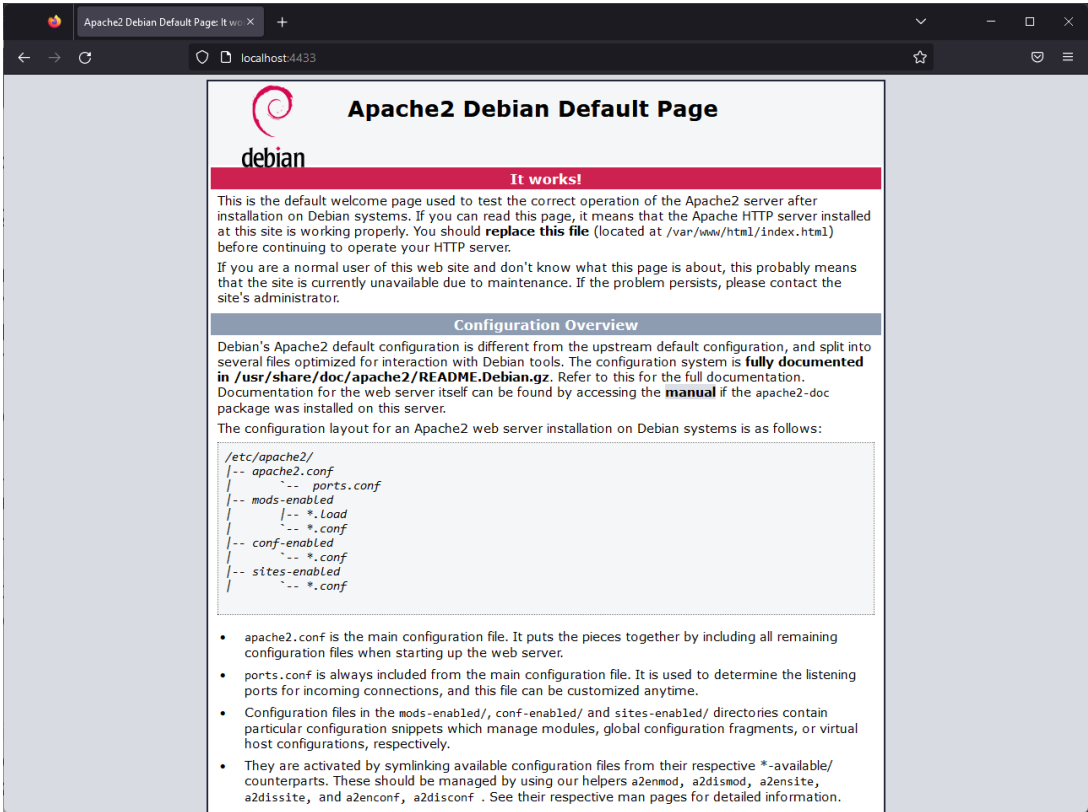
Risultati test

Test Case	TC-001
	<pre>root@router:/home/administrator# 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:fb:17:87 brd ff:ff:ff:ff:ff:ff inet 10.0.2.6/24 brd 10.0.2.255 scope global dynamic enp0s3 valid_lft 85955sec preferred_lft 85955sec inet6 fe80::a00:27ff:febf:1787/64 scope link valid_lft forever preferred_lft forever 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:ca:2f:d3 brd ff:ff:ff:ff:ff:ff inet 192.168.238.1/24 brd 192.168.238.255 scope global enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:feca:2fd3/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:98:04:6e brd ff:ff:ff:ff:ff:ff inet 192.168.198.1/24 brd 192.168.198.255 scope global enp0s9 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe98:46e/64 scope link valid_lft forever preferred_lft forever 5: enp0s10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:7e:b0:a3 brd ff:ff:ff:ff:ff:ff inet 192.168.56.100/24 brd 192.168.56.255 scope global enp0s10 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe7e:b0a3/64 scope link valid_lft forever preferred_lft forever</pre>
Funzionamento	<pre>root@intSrv:/home/administrator# 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:a7:eb:54 brd ff:ff:ff:ff:ff:ff inet 192.168.198.10/24 brd 192.168.198.255 scope global enp0s3 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fea7:eb54/64 scope link valid_lft forever preferred_lft forever 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:86:92:80 brd ff:ff:ff:ff:ff:ff inet 192.168.60.100/24 brd 192.168.60.255 scope global enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe86:9280/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000 link/ether 08:00:27:0e:6b:93 brd ff:ff:ff:ff:ff:ff</pre>
	<pre>root@intCli:/home/administrator# 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:56:50:46 brd ff:ff:ff:ff:ff:ff inet 192.168.238.10/24 brd 192.168.238.255 scope global enp0s3 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fe56:5046/64 scope link valid_lft forever preferred_lft forever 3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000 link/ether 08:00:27:f7:e9:1b brd ff:ff:ff:ff:ff:ff inet 192.168.50.100/24 brd 192.168.50.255 scope global enp0s8 valid_lft forever preferred_lft forever inet6 fe80::a00:27ff:fef7:e91b/64 scope link valid_lft forever preferred_lft forever 4: enp0s9: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc pfifo_fast state DOWN group default qlen 1000 link/ether 08:00:27:4b:89:c7 brd ff:ff:ff:ff:ff:ff</pre>
Commento	Negli screenshot si vedono le 3 macchine con i rispettivi indirizzi ip.
Data	26.11.2022

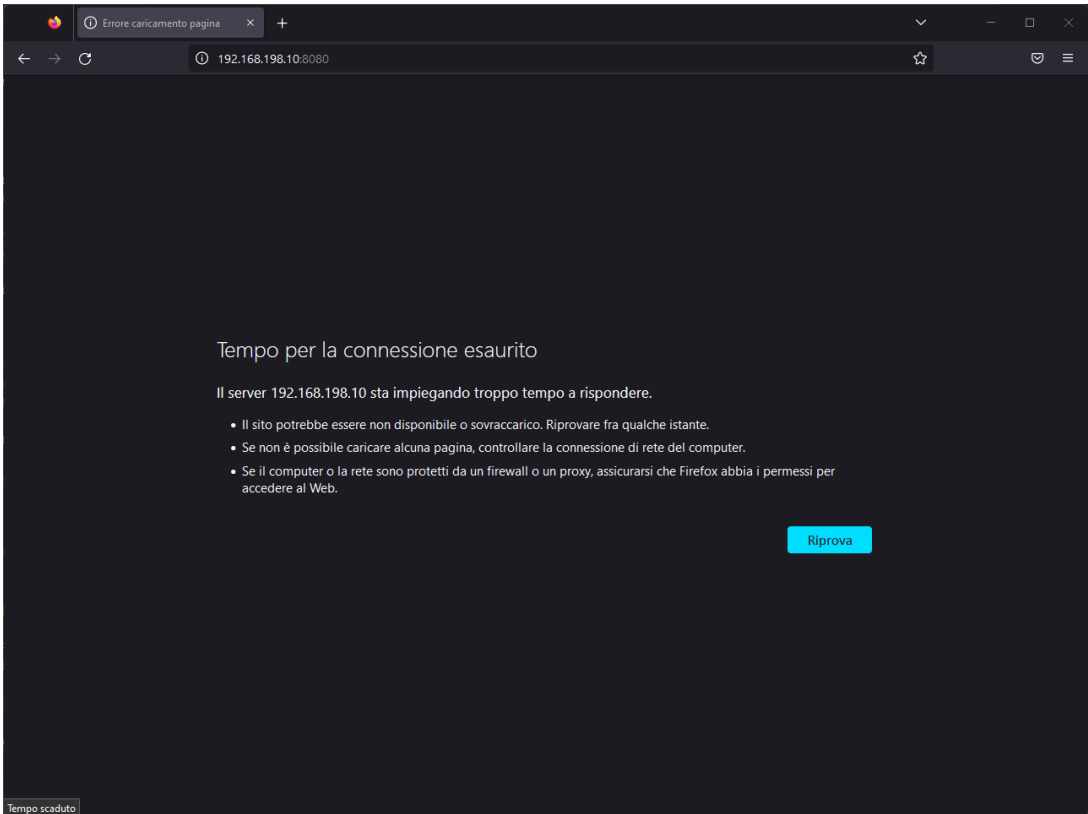
Test Case	TC-002
Funzionamento	<pre>root@intCli:/home/administrator# curl google.com <HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8"> <TITLE>301 Moved</TITLE></HEAD><BODY> <H1>301 Moved</H1> The document has moved here. </BODY></HTML></pre>
Commento	Nello screenshot si vede la macchina client che esegue il comando curl verso google.com.
Data	26.11.2022

Test Case	TC-003
Funzionamento	<pre>root@intCli:/home/administrator# curl 192.168.198.10:8080 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head> <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" /> <title>Apache2 Debian Default Page: It works</title> <style type="text/css" media="screen"> * { margin: 0px 0px 0px 0px; padding: 0px 0px 0px 0px; } </pre> 
Commento	<p>Nel primo screenshot si vede che dalla macchina client si riesce a contattare il webserver nella dmz.</p> <p>Nella seconda immagine si vede che da un browser della macchina host, il nostro internet, si riesce a vedere la pagina del webserver tramite l'ip del router.</p>
Data	26.11.2022

Test Case TC-004



Funzionamento

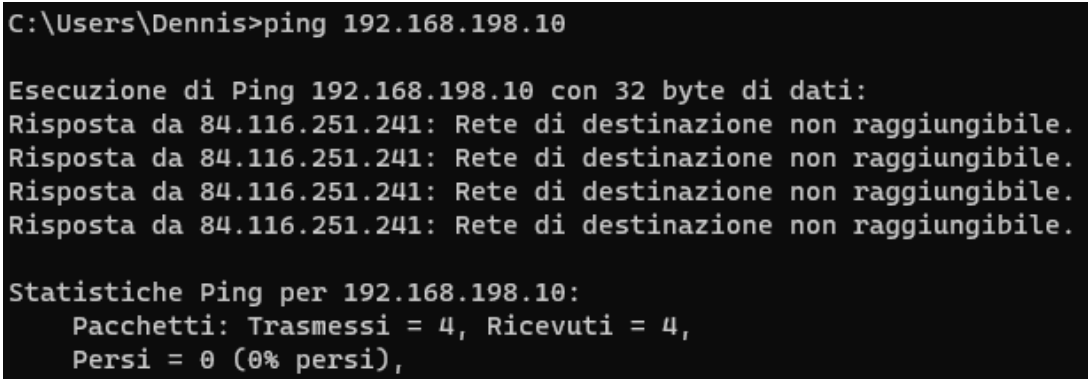
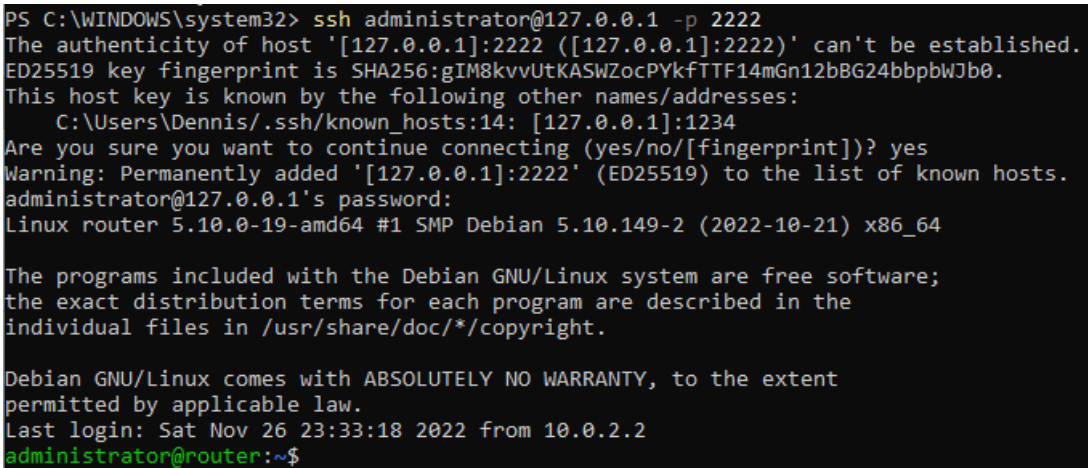
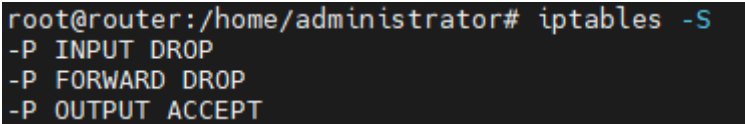


Commento

Nel primo screenshot si vede la connessione che non può essere stabilita. La causa è che non è possibile arrivare al webserver direttamente con il suo ip.
Nel secondo screenshot si vede la connessione stabilita perchè viene chiamato l'ip del router.

Data

26.11.2022

Test Case	TC-005
Funzionamento	 <pre>C:\Users\Dennis>ping 192.168.198.10 Esecuzione di Ping 192.168.198.10 con 32 byte di dati: Risposta da 84.116.251.241: Rete di destinazione non raggiungibile. Risposta da 84.116.251.241: Rete di destinazione non raggiungibile. Risposta da 84.116.251.241: Rete di destinazione non raggiungibile. Risposta da 84.116.251.241: Rete di destinazione non raggiungibile. Statistiche Ping per 192.168.198.10: Pacchetti: Trasmessi = 4, Ricevuti = 4, Persi = 0 (0% persi),</pre>
Commento	Nello screenshot si vede il tentativo del ping ma non riesce perchè non conosce quella rete.
Data	26.11.2022
Test Case	TC-006
Funzionamento	 <pre>PS C:\WINDOWS\system32> ssh administrator@127.0.0.1 -p 2222 The authenticity of host '[127.0.0.1]:2222 ([127.0.0.1]:2222)' can't be established. ED25519 key fingerprint is SHA256:gIM8kvvUtKASWZocPYkFTTF14mGn12bBG24bbpbWJb0. This host key is known by the following other names/addresses: C:\Users\Dennis/.ssh/known_hosts:14: [127.0.0.1]:1234 Are you sure you want to continue connecting (yes/no/[fingerprint])? yes Warning: Permanently added '[127.0.0.1]:2222' (ED25519) to the list of known hosts. administrator@127.0.0.1's password: Linux router 5.10.0-19-amd64 #1 SMP Debian 5.10.149-2 (2022-10-21) x86_64 The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*/copyright. Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law. Last login: Sat Nov 26 23:33:18 2022 from 10.0.2.2 administrator@router:~\$</pre>
Commento	Nello screenshot si vede l'accesso effettuato tramite ssh alla porta 2222 dalla macchina host.
Data	26.11.2022
Test Case	TC-007
Funzionamento	 <pre>root@router:/home/administrator# iptables -S -P INPUT DROP -P FORWARD DROP -P OUTPUT ACCEPT</pre>
Commento	Nello screenshot si vede la configurazione delle catene INPUT, FORWARD e OUTPUT. Le prime due sono DROP e l'ultima è ACCEPT .
Data	26.11.2022

Test Case

TC-008

```

root@router:/home/administrator# iptables --policy INPUT DROP
iptables --policy FORWARD DROP
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport ssh -j ACCEPT

iptables -t nat -A POSTROUTING -s 192.168.198.0/24 -o enp0s3 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 192.168.238.0/24 -o enp0s3 -j MASQUERADE

iptables -A FORWARD -s 192.168.198.0/24 -i enp0s9 -j ACCEPT
iptables -A FORWARD -s 192.168.238.0/24 -i enp0s8 -j ACCEPT
iptables -A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

iptables -t nat -A PREROUTING -i enp0s3 -p tcp --dport 443 -j DNAT --to-destination 192.168.198.10:8080
iptables -A FORWARD -i enp0s3 -p tcp --dport 8080 -j ACCEPT

iptables -A INPUT -i enp0s9 -p icmp -j ACCEPT
iptables -A OUTPUT -o enp0s9 -p icmp -j ACCEPT
iptables -A INPUT -i enp0s8 -p icmp -j ACCEPT
iptables -A OUTPUT -o enp0s8 -p icmp -j ACCEPT

```

Funzionamento

```

root@router:/home/administrator# uptime
23:47:51 up 0 min, 1 user, load average: 1.44, 0.36, 0.12
root@router:/home/administrator# iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i enp0s9 -p icmp -j ACCEPT
-A INPUT -i enp0s8 -p icmp -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -s 192.168.198.0/24 -i enp0s9 -j ACCEPT
-A FORWARD -s 192.168.238.0/24 -i enp0s8 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A FORWARD -i enp0s3 -p tcp -m tcp --dport 8080 -j ACCEPT
-A OUTPUT -o enp0s9 -p icmp -j ACCEPT
-A OUTPUT -o enp0s8 -p icmp -j ACCEPT

```

Commento

Nel primo screenshot si vede l'inserimento delle regole in iptables.
 Nel secondo screenshot si vedono le regole di iptables anche dopo il riavvio.

Data

26.11.2022

Conclusioni

Considerazioni personali

- Samuel Banfi: A me personalmente è piaciuto molto questo progetto. Mi ha aiutato a capire meglio le funzionalità di Linux, ma soprattutto come rendere sicura una rete da possibili intrusioni esterne. Sono però dell'idea che se avessimo avuto un po' di tempo in più saremmo riusciti a migliorare ancora di più la sicurezza. Credo che questo progetto mi tornerà utile in futuro in una azienda.
- Dennis Donofrio: Questo progetto è stato bello ed utile. Mi è piaciuto il fatto di lavorare con un programma molto semplice ma allo stesso tempo molto utile. Inoltre iptables lo si può utilizzare su qualsiasi macchina linux e non serve per forza usarlo come router. Una cosa che abbiamo notato è che non è installato di default su tutte le macchine linux. Su debian bisogna installarlo manualmente, come la maggior parte dei programmi di base, come il comando `sudo`. Mi sono trovato molto bene a lavorare in gruppo perchè siamo riusciti a dividerci i compiti. Questo ha reso il tutto più semplice.

Sitografia

- moodle.edu.ti.ch, Data ultima visita: 26.11.2022
- stackoverflow.com, Data ultima visita: 25.11.2022