Nitin Nagavel

CPRE 431

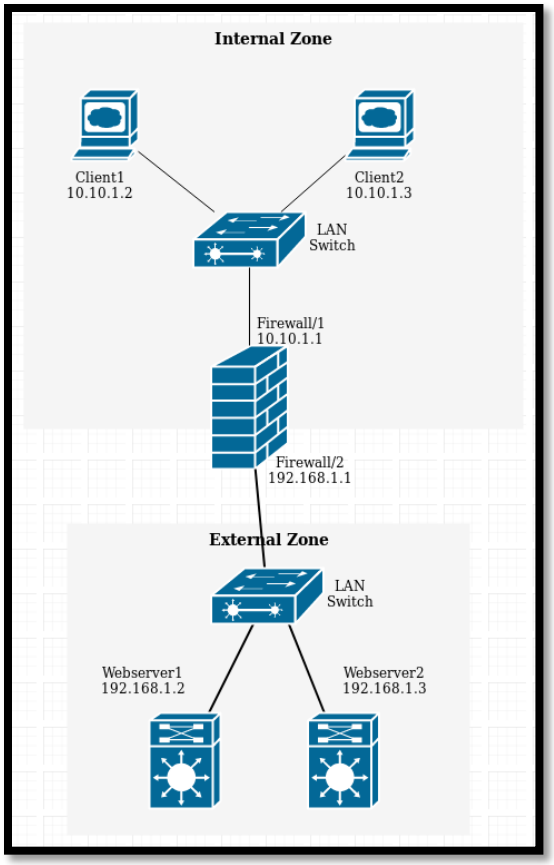
M06 Lab HW

Assignments will be submitted in PDF format via Canvas.

Please submit your homework online through Canvas. Late homework will not be accepted.
Important: Your submission must be in .pdf format ONLY!
Please ensure that you support all your answers with the correct screenshots showing your solutions.

Notes:

- The goal of this lab is to use iptables to create simple firewall rules.
- Use the “Introduction to iptables” page on canvas to help you in the needed commands for this homework.
- This homework consists of 5 tasks. **The answer for each task requires you to record a command/result and write an explanation and support this with screenshots.**
- Use the template "M06 - Answers.txt" file (attached with the homework) as a guideline of how the answer should look like. Your answer submission must be in .pdf format ONLY!
- The tasks assume you have created the below virtual network topology on GENI platform. You can use the rspec file "M06 - rspec.xml" (attached with the homework) to create the network topology on GENI platform (if the reservation failed because there is no enough pcs, please use a different InstaGENI site).
- This virtual network consists of two zones, internal (Clients) and external (Servers). The internal zone consists of two clients (Client1 and 2) connected through a LAN switch to a Firewall that connects them to the external zone. The Firewall has two interfaces “Firewall/1” connected to the internal network, and “Firewall/2” connected to the external zone. The external zone has many servers connected through a LAN switch to the Firewall. However, we will only focus on two servers (Webserver1 and 2). The below figure has the network diagram and the needed details for the zones, clients, servers, and firewall:



- **In all your answers, try to write rules as general as possible.** For example, although we are only focusing on two servers in the external zone, try to write rules such that the policy is achieved even if there were more than two servers.
- **Always remember to flush all iptables rules between tasks** (for example: after completing task 1, make sure there are no rules before you start task 2).

<u>Task 1: (Aim: understand the difference between INPUT, FORWARD, and OUTPUT chains):</u>

1. Add **Rule1** to the firewall filter table that blocks ping (icmp) packets being forwarded between the two subnets.

```
ome4nit@router:~$ sudo iptables -A FORWARD -p icmp -j DROP
ome4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A FORWARD -p icmp -j DROP
ome4nit@router:~$
```

a. Client1 to Webserver1

```
home4nit@client1:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2052ms
```

b. Webserver2 to Client2

```
home4nit@webserver2:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3057ms

home4nit@webserver2:~$
```

c. Client1 to Firewall

```
home4nit@client1:~$ ping -c 3 router
PING Router-link-1 (10.10.2.1) 56(84) bytes of data.
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=1 ttl=64 time=1.23
ms
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=2 ttl=64 time=0.767
 ms
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=3 ttl=64 time=0.854
 ms

--- Router-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.767/0.953/1.239/0.206 ms
home4nit@client1:~$
```

d. Firewall to Webserver1

```
home4nit@router:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=1 ttl=64 time=1.4
7 ms
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=2 ttl=64 time=0.8
30 ms
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=3 ttl=64 time=0.7
75 ms

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.775/1.026/1.474/0.318 ms
home4nit@router:~$
```

e. Client1 to Client2

```
home4nit@client1:~$ ping -c 3 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=2 ttl=64 time=0.463 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=3 ttl=64 time=0.542 ms

--- client2-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2033ms
rtt min/avg/max/mdev = 0.463/0.733/1.196/0.329 ms
home4nit@client1:~$
```

2. Delete Rule1, and then add **Rule2** that blocks ping packets coming into the firewall.

```
home4nit@router:~$ sudo iptables -A INPUT -p icmp -j DROP
home4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -j DROP
-A FORWARD -p icmp -j DROP
home4nit@router:~$
```

a. Client1 to Webserver1

```
home4nit@client1:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2031ms

home4nit@client1:~$
```

b. Webserver2 to Client2

```
home4nit@webserver2:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3079ms

home4nit@webserver2:~$
```

c. Client1 to Firewall

```
home4nit@client1:~$ ping -c 4 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3075ms

home4nit@client1:~$
```

d. Firewall to Webserver1

```
home4nit@router:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2035ms

home4nit@router:~$
```

e. Client1 to Client2

```
home4nit@client1:~$ ping -c 3 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=1 ttl=64 time=0.848 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=2 ttl=64 time=0.470 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=3 ttl=64 time=0.466 ms

--- client2-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.466/0.594/0.848/0.181 ms
home4nit@client1:~$
```

3. Delete Rule2, and then add **Rule3** that blocks ping packets coming out of the firewall.

```
home4nit@router:~$ sudo ipdates -S
sudo: ipdates: command not found
home4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -j DROP
-A FORWARD -p icmp -j DROP
-A OUTPUT -p icmp -j DROP
home4nit@router:~$
```

a. Client1 to Webserver1

```
home4nit@client1:~$ ping -c 4 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3079ms

home4nit@client1:~$
```

b. Webserver2 to Client2

```
home4nit@webserver2:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3066ms

home4nit@webserver2:~$
```

c. Client1 to Firewall

```
home4nit@client1:~$ ping -c 3 router
PING Router-link-1 (10.10.2.1) 56(84) bytes of data.

--- Router-link-1 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2038ms

home4nit@client1:~$
```

d. Firewall to Webserver1

```
home4nit@router:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2034ms

home4nit@router:~$
```

e. Client1 to Client2

```
home4nit@client1:~$ ping -c 3 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=1 ttl=64 time=0.447 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=2 ttl=64 time=0.411 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=3 ttl=64 time=0.495 ms

--- client2-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2038ms
rtt min/avg/max/mdev = 0.411/0.451/0.495/0.034 ms
home4nit@client1:~$
```

To test each of the above rules, try to ping between the following pairs of nodes, and observe whether it is successful or not:

- Client1 to Webserver1
- Webserver2 to Client2
- Client1 to Firewall
- Firewall to Webserver1
- Client1 to Client2

4. Record your rules' results (allowed or blocked) and an explanation of the difference between the three chains in your answer (take screenshots of webserver/client/firewall terminals to show your syntax and results of pings).

   Forward: forward ping request from the firewall gets denied

   Input: ping request gets denied to the router/firewall

   Output: ping request gets denied from the firewall

Task 2: (Aim: filter based on ports and IPs):

This task will be testing SSH connections between client and server. SSH is enabled by default on the webservers. GENI platform forces SSH through key-pair authentication only. We will need to enable SSH to use password authentication as well.

1. On Webserver1 and Webserver2, to enable password authentication for SSH, we need to change the SSH configuration as follows:

   ```
   > sudo nano /etc/ssh/sshd_config
   ```

   - Near the end of the file, search for the line having: "`PasswordAuthentication no`"
   - Change it to "`PasswordAuthentication yes`", then save the changes and exit.

2. Restart the SSH service:

   ```
   > sudo service ssh restart
   ```

3. On Webserver1, Webserver2 and the Firewall, we need to create a test user to be used in verifying the SSH connections:

   ```
   > sudo adduser test
   ```

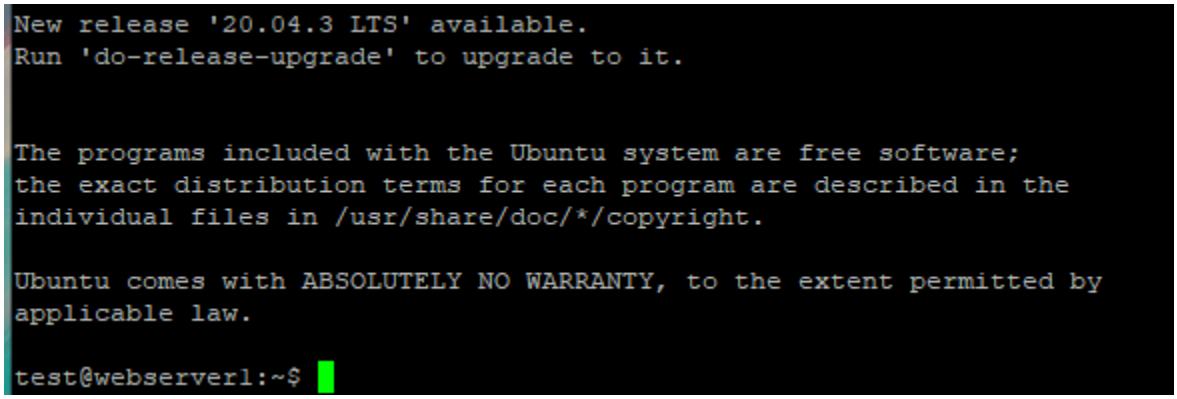
   - Add the user password and information needed to create the user.
     Webserver1Info: 1234
     Webserver2Info: 5678
     routerUserInfo: 90
     client1Info: 9876

4. Verify that the SSH connection can be established as follows:

   - On Client1:

   ```
   > ssh test@webserver1
   ```

   - Ensure that you get a prompt for password, then enter the password for user "test" and ensure the SSH connection is successfully established.



5. Now we need to configure the Firewall to control the SSH connections. Add **Rule4** to the firewall filter table to prevent Client1 from SSHing to any outside nodes, but it can still SSH to the firewall itself.

```
home4nit@client1:~$ ssh test@webserver1
ssh: connect to host webserver1 port 22: Connection timed out
home4nit@client1:~$
```

6. Test the correctness of **Rule4**.

```
home4nit@router:~$ sudo iptables -A FORWARD -i eth2 -p tcp --dport 22 -j D
ROP
home4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -j DROP
-A FORWARD -p icmp -j DROP
-A FORWARD -i eth2 -p tcp -m tcp --dport 22 -j DROP
-A OUTPUT -p icmp -j DROP
home4nit@router:~$
```

```
test@router's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

test@router:~$
```

Task 3: (Aim: filter based on ports, understand HTTP request/response format):

The first part of this task is similar to how we setup the environment of HW04. Please refer to HW04, if you need help with the commands.

1. Install and enable the Apache webserver on both Webserver1 and Webserver2.
2. The Lynx web browser should be installed by default on both Client1 and Client2. To verify run the following command on the clients:

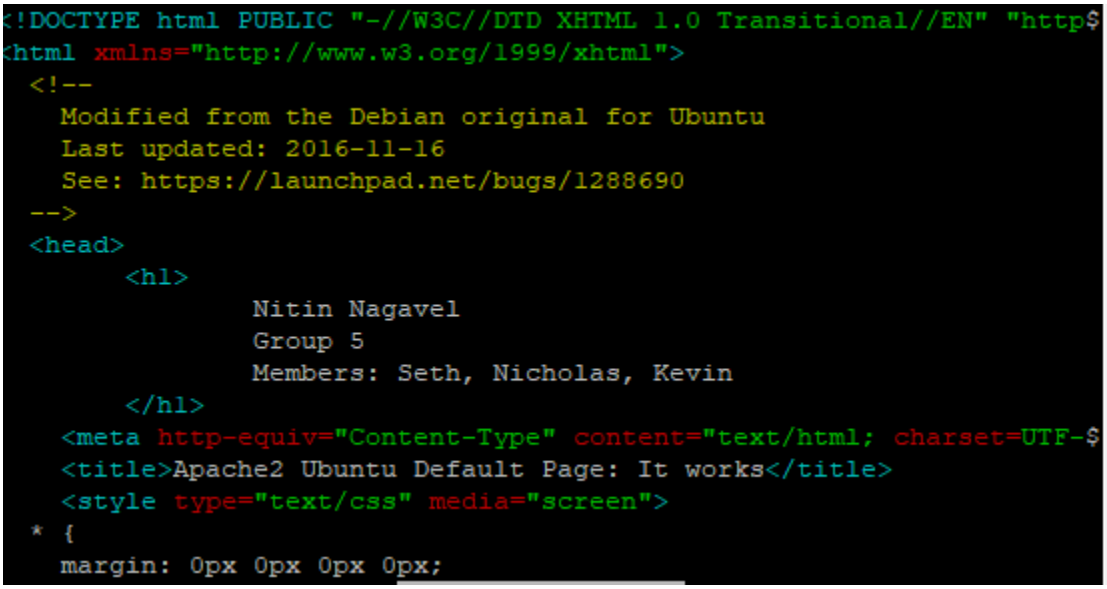
   ```
   > lynx -version
   ```

3. Verify that the web servers are running by connecting to them from the Lynx browsers on the clients, and you should see the Apache2 Ubuntu Default Page.

   ```
   > lynx http://webserver1
   ```

4. Use a text editor to change the index.html file to include your group number and members' names in the heading of the default webpage (use <h1> tags):

   ```
   > sudo nano /var/www/html/index.html
   ```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http$
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2016-11-16
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
        <h1>
                Nitin Nagavel
                Group 5
                Members: Seth, Nicholas, Kevin
        </h1>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-$
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
                          [ Read 380 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell
```

5. Verify that your changes are applied by connecting to the server again using the Lynx browser from the clients. Check the response and make sure it is what you expected.

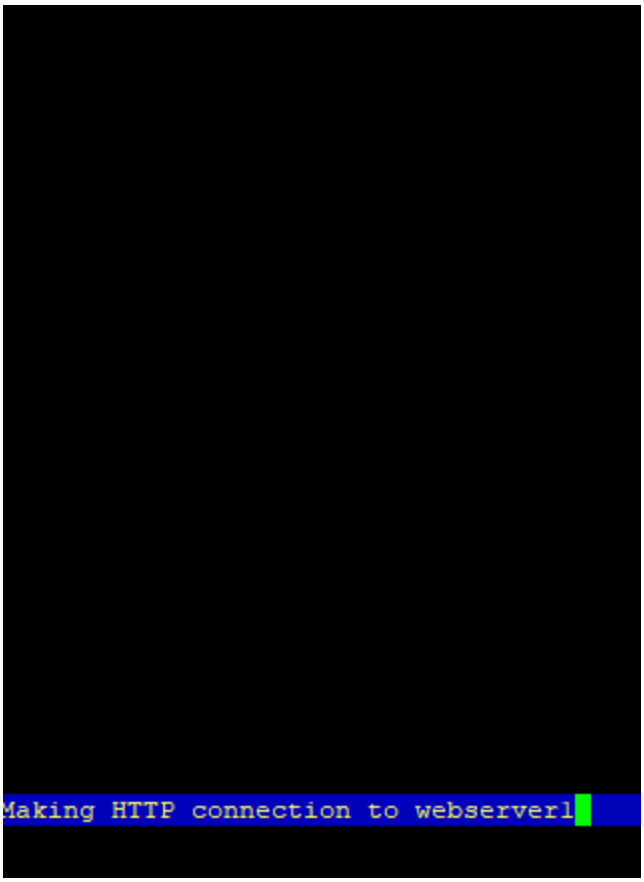
```
home4nit@client1:~$ lynx http://webserver1
                                Apache2 Ubuntu Default Page: It works (pl of
             Nitin Nagavel Group 5 Members: Seth, Nicholas, Kevin

   Ubuntu Logo Apache2 Ubuntu Default Page
   It works!

   This is the default welcome page used to test the correct operation of
   the Apache2 server after installation on Ubuntu systems. It is based on
   the equivalent page on Debian, from which the Ubuntu Apache packaging
   is derived. If you can read this page, it means that the Apache HTTP
   server installed at this site is working properly. You should replace
   this file (located at /var/www/html/index.html) before continuing to
   operate your HTTP server.
```

```
home4nit@client2:~$ lynx http://webserver2
                     Apache2 Ubuntu Default Page: It works (pl of
        Nitin Nagavel Group 5 Members: Seth, Nicholas, Kevin

   Ubuntu Logo Apache2 Ubuntu Default Page
   It works!

   This is the default welcome page used to test the correct
   operation of the Apache2 server after installation on
   Ubuntu systems. It is based on the equivalent page on
   Debian, from which the Ubuntu Apache packaging is derived.
   If you can read this page, it means that the Apache HTTP
```

6. Add **Rule5** to the Firewall filter to prevent internal zone clients from accessing the webpage on Webserver1.

```
home4nit@router:~$ sudo iptables -A FORWARD -i eth2 -p tcp --dport 80 -j DROP
home4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A FORWARD -i eth2 -p tcp -m tcp --dport 80 -j DROP
home4nit@router:~$
```

7. Test the correctness of **Rule5**.



<u>Task 4: (Aim: change default policy):</u>

1. Up to this point, we were using the default policy of ACCEPT on the Firewall. For **task 5** we will use the default policy of DROP.
2. Change the default policy of the firewall "forwarding" table to DROP.

```
home4nit@router:~$ sudo iptables -P FORWARD DROP
home4nit@router:~$ sudo iptables -S
-P INPUT ACCEPT
-P FORWARD DROP
-P OUTPUT ACCEPT
home4nit@router:~$
```

3. Verify that all connections tested through tasks 1 to 3 are blocked without adding their rules.
    a. Client1 to Webserver1

```
home4nit@client1:~$ ping -c 3 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.

--- webserver1-link-0 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2028ms

home4nit@client1:~$
```

b. Webserver2 to Client2

```
home4nit@webserver2:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3072ms

home4nit@webserver2:~$
```

c. Client1 to Firewall

```
home4nit@client1:~$ ping -c 3 router
PING Router-link-1 (10.10.2.1) 56(84) bytes of data.
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=1 ttl=64 time=1.26 ms
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=2 ttl=64 time=0.858 ms
64 bytes from Router-link-1 (10.10.2.1): icmp_seq=3 ttl=64 time=0.920 ms

--- Router-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.858/1.014/1.264/0.178 ms
home4nit@client1:~$
```

d. Firewall to Webserver1

```
home4nit@router:~$ ping -c 4 webserver1
PING webserver1-link-0 (10.10.1.3) 56(84) bytes of data.
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=1 ttl=64 time=0.850 ms
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=2 ttl=64 time=0.660 ms
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=3 ttl=64 time=0.763 ms
64 bytes from webserver1-link-0 (10.10.1.3): icmp_seq=4 ttl=64 time=0.943 ms

--- webserver1-link-0 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.660/0.804/0.943/0.104 ms
home4nit@router:~$
```

e. Client1 to Client2

```
home4nit@client1:~$ ping -c 3 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=1 ttl=64 time=1.15 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=2 ttl=64 time=0.589 ms
64 bytes from client2-link-1 (10.10.2.3): icmp_seq=3 ttl=64 time=0.499 ms

--- client2-link-1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.499/0.746/1.150/0.288 ms
home4nit@client1:~$
```

Task2:

```
home4nit@client1:~$ ssh test@router
test@router's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Oct 31 16:27:59 2021 from 10.10.2.2
test@router:~$
```

```
home4nit@client1:~$ ssh test@webserver1
```

Task 3:

Task 5: (Aim: use Stateful Packet Inspection):

1. Enable Stateful Packet Inspection on the firewall with:

   ```
   > sudo iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
   ```

2. Add **Rule6** so that inside hosts can access outside websites

```
home4nit@router:~$ sudo iptables -A FORWARD -i eth2 -p tcp --dport 80 -j ACCEPT
```

3. Add **Rule7** so that outside hosts can SSH into Client1. To enable SSH to Client1, please refer to commands used in task 2.

```
home4nit@router:~$ sudo iptables -A INPUT -i eth2 -p tcp --dport 22 -j ACCEPT
```

4. No other access should be allowed.

Hint: In iptables rules, you can use "-i" and "-o" to specify the input or output interfaces.

5. To view the SPI table, you need to install the “conntrack” package on the firewall:

```
> sudo apt-get install conntrack
```

6. Then you can view the SPI table with:

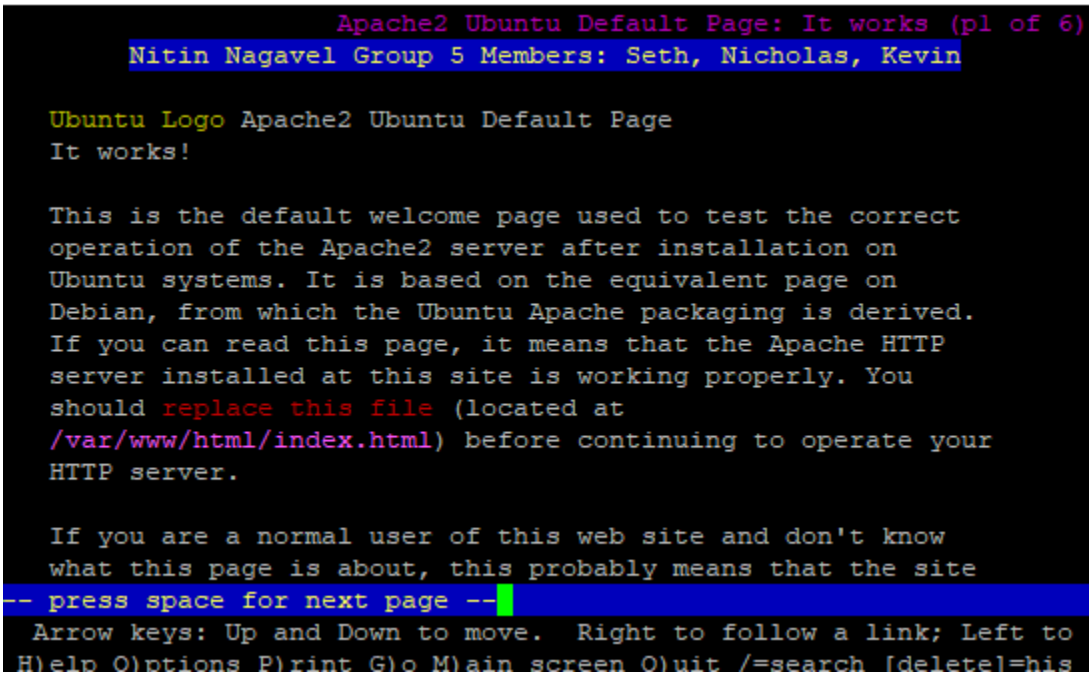
```
> sudo conntrack -L
```

7. To view the SPI entries created on the fly while testing the firewall:

```
> sudo conntrack –E
```

Hint: GENI platform uses the network 172.17.0.0/16 for internal maintenance. You can filter out these packets/connections through your analysis by using –s <Scr_IP> or –d <Dst_IP> with conntrack commands.

8. Test the correctness of **Rule6** by connecting to the webpage on Webserver1 from Client2.



9. Test the correctness of **Rule7** by SSHing to Client1 from Webserver1.

```
home4nit@webserver1:~$ ssh test@client1
test@client1's password:
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-159-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch
New release '20.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

test@client1:~$
```

10. Test any other connections and ensure they are all blocked.

```
home4nit@webserver2:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3055ms

home4nit@webserver2:~$
```

```
home4nit@webserver2:~$ ping -c 4 client1
PING client1-link-1 (10.10.2.2) 56(84) bytes of data.

--- client1-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3055ms

home4nit@webserver2:~$
```

```
home4nit@webserver1:~$ ping -c 4 client1
PING client1-link-1 (10.10.2.2) 56(84) bytes of data.

--- client1-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3052ms

home4nit@webserver1:~$
```

```
home4nit@webserver1:~$ ping -c 4 client2
PING client2-link-1 (10.10.2.3) 56(84) bytes of data.

--- client2-link-1 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3067ms

home4nit@webserver1:~$
```