IT Technology
Assignment 54
SRX security Address book, Zones and Policies

University College

Author
Aleksandra Voronina - alvo28899@edu.ucl.dk
Aubrey Jones – asjo28903@edu.ucl.dk
Dainty Olsen - dlbo28887@edu.ucl.dk
Gladys Waithera - gwma28853@edu.ucl.dk
Henrik Hansen - hhha28796@edu.ucl.dk
Lukasz Zwak - luzw28872@edu.ucl.dk
Thobias Selmann - tjse28878@edu.ucl.dk

Date: 06/03-2021

Tabel of contents

1 Introduction

This assignment is building on the network develop in assignment 11. The naming of subnets is different but the basic configurations of the two routers vSRX_1 and vSRX_2 are identical to those done in assignment 11. The assignment also builds on assignment 53, where a Raspberry Pi as a Web server was introduced. Here this Web Server on PC2 is also part of the assignment.

2 Audience

This report is designed for the fellow students of the IT Technology course. This document will show how Group B3 set up the networks to fulfill the requirements of Assignment 54. Everything following will show how the group went about setting up security zones within the network.

3 Inventory

- VMWare Workstation
- At least two virtual computers (preferably Xubuntu, and one RaspberryPi Buster)
- A laptop or computer
- Junos-vsrx-12.1X47-D15.4-domestic.ovf
- Junos-vsrx-12.1X47-D15.4-domestic.mf
- Junos-vsrx-12.1X47-D15.4-domestic-disk1.vmdk
- Raspios-buster-i386.iso

Installed on the Raspios-buster

- NGINX

4 Learning objectives

Learning objectives. After having worked with this assignment:

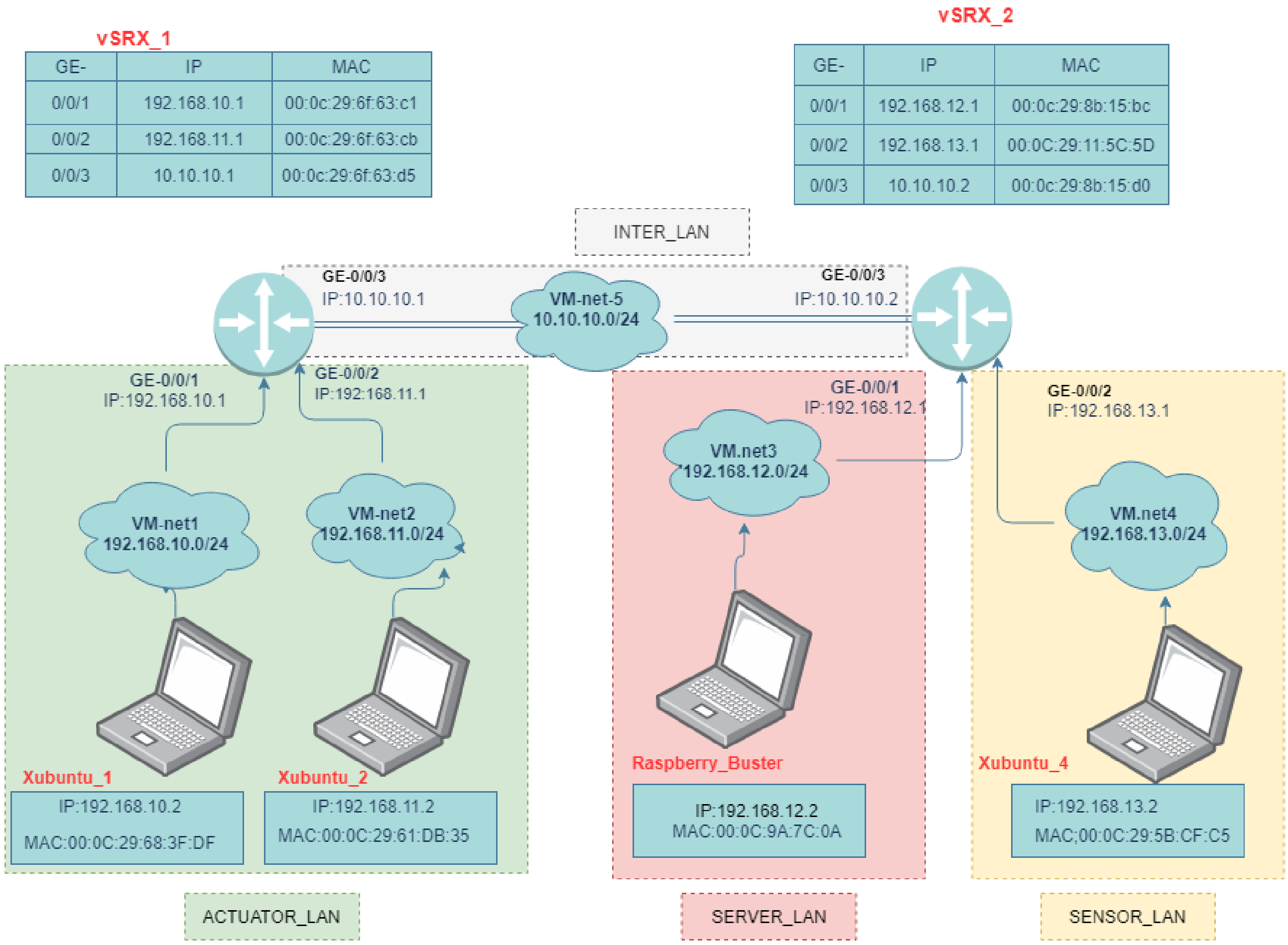
The student can at a basic level explain:

- SRX zones.
- SRX policies.
- SRX address book.
- SRX applications.

The students will also have a basic level of understanding in:

- Specify security requirements for a given simple network.
- List simple security specifications.
- Configure Junos SRX address book, zones, policies and applications.
- Elaborate a test plan and test simple security configurations accordingly.

5 Network Diagram.



vSRX_1

GE-	IP	MAC
0/0/1	192.168.10.1	00:0c:29:6f:63:c1
0/0/2	192.168.11.1	00:0c:29:6f:63:cb
0/0/3	10.10.10.1	00:0c:29:6f:63:d5

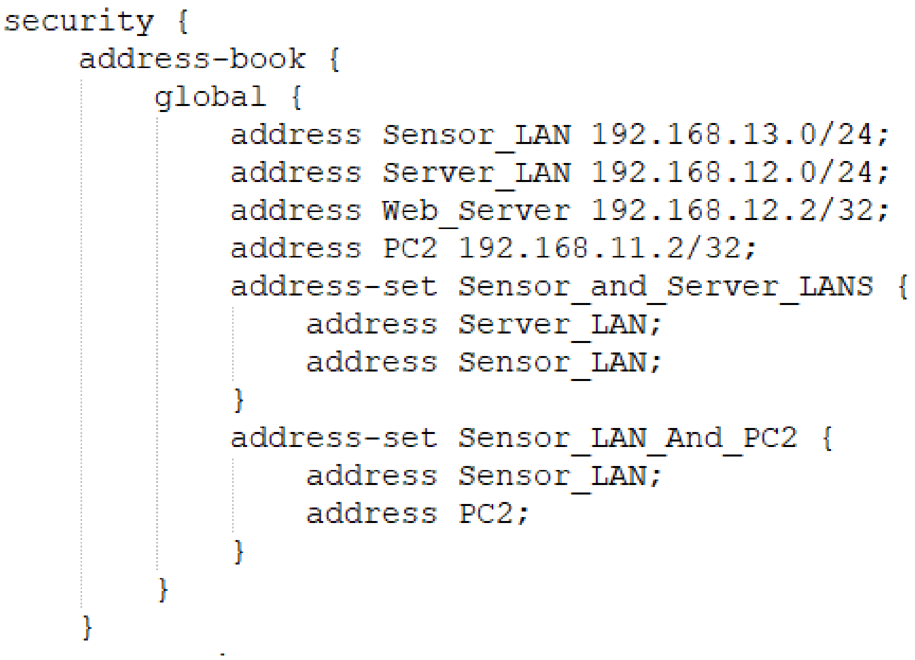
vSRX_2

GE-	IP	MAC
0/0/1	192.168.12.1	00:0c:29:8b:15:bc
0/0/2	192.168.13.1	00:0C:29:11:5C:5D
0/0/3	10.10.10.2	00:0c:29:8b:15:d0

I have in this task chosen to use the same network setup as in task 53. The only difference is that this time I have built some security zones into vSRX1_2 to direct traffic on and from my RaspberryPi.

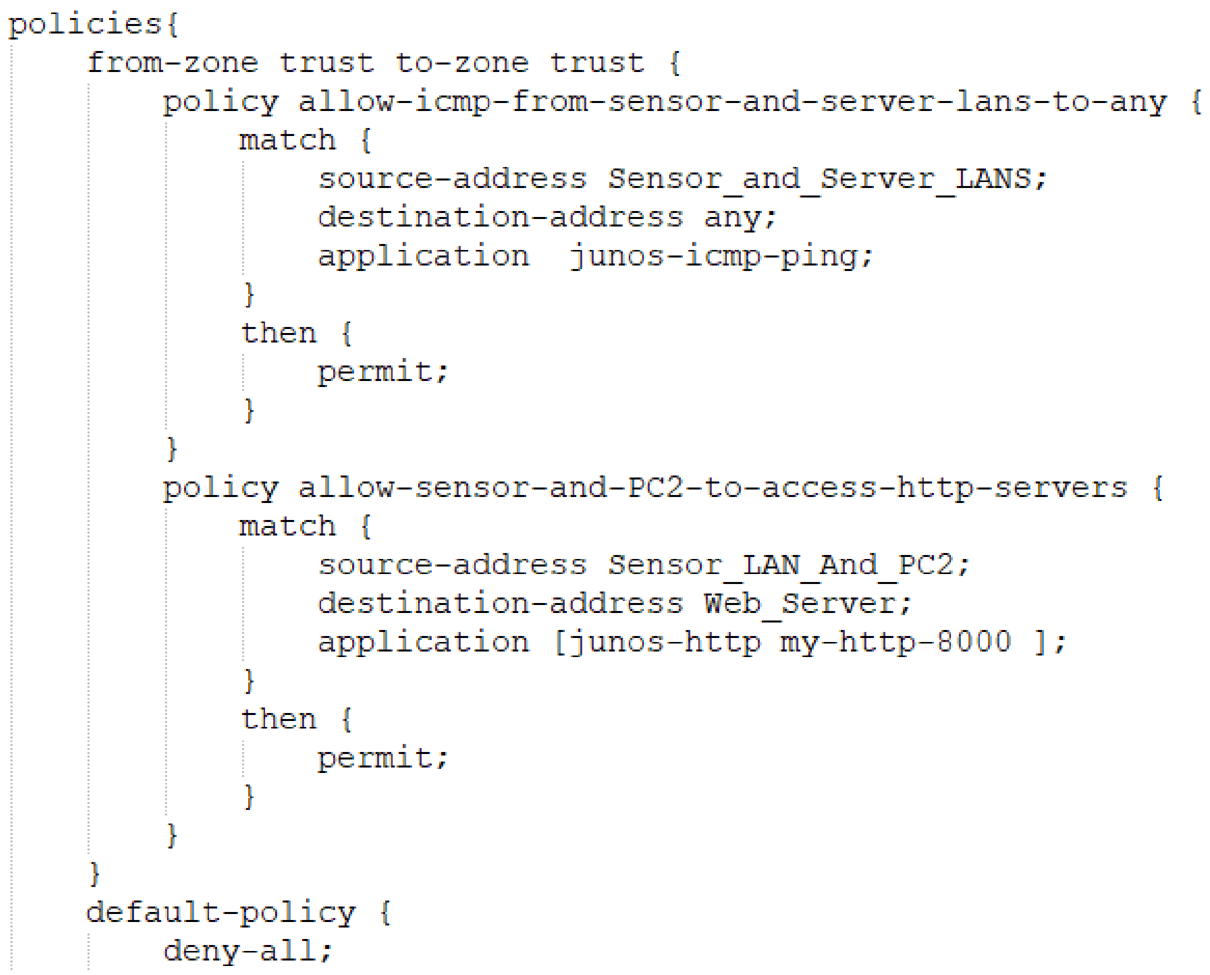
6 List of security specifications

▪ PC 1: Only communicate devices until 10.10.10.1.
▪ PC 2: Only communicate devices until 10.10.10.1 and fetch webpage from PC2
▪ RaspberryPi: Can Communicate all and fetch webpage from PC2.
▪ PC4: Can Communicate all and fetch webpage from PC2.



```
security {
    address-book {
        global {
            address Sensor_LAN 192.168.13.0/24;
            address Server_LAN 192.168.12.0/24;
            address Web_Server 192.168.12.2/32;
            address PC2 192.168.11.2/32;
            address-set Sensor_and_Server_LANS {
                address Server_LAN;
                address Sensor_LAN;
            }
            address-set Sensor_LAN_And_PC2 {
                address Sensor_LAN;
                address PC2;
            }
        }
    }
```

There is make an address book containing all the computers present. This can be seen on their IP addresses, subsequently they have been divided into two subcategories Sensor_and_Server_LANS and Sensor_LAN_And_PC2



```
policies{
    from-zone trust to-zone trust {
        policy allow-icmp-from-sensor-and-server-lans-to-any {
            match {
                source-address Sensor_and_Server_LANS;
                destination-address any;
                application  junos-icmp-ping;
            }
            then {
                permit;
            }
        }
        policy allow-sensor-and-PC2-to-access-http-servers {
            match {
                source-address Sensor_LAN_And_PC2;
                destination-address Web_Server;
                application [junos-http my-http-8000 ];
            }
            then {
                permit;
            }
        }
    }
    default-policy {
        deny-all;
```

Here you can see that in the two subcategories are now in two different safety zones

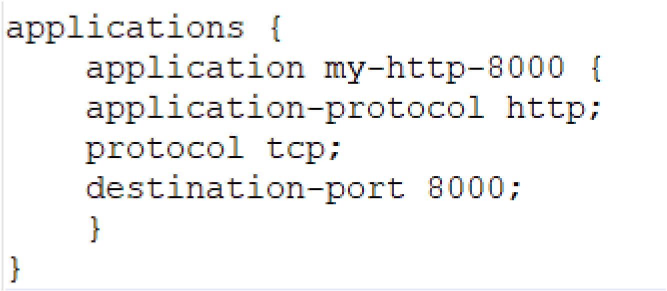
6.1 First Zone

In the first zone it can be observed that if you come from the subcategory Sensor_and_Server_LANS, and just want to go somewhere, and want to use the applications junos-icmp-ping then you are allowed.

6.2 Second Zone

In the second zone it can be observed that if you come from the subcategory Sensor_LAN_And_PC2, and want to enter a Webserver, and will use the applications junos-http or my-http-8000 then you are allowed.

6.3 Specification of the application



```
applications {
    application my-http-8000 {
    application-protocol http;
    protocol tcp;
    destination-port 8000;
    }
}
```

This application gives access to connect to an http 8000 destination port. But unlike the applications **junos-icmp-ping** we do not have permission to ping the device.

7 Test plan and proof of connection.

7.1 Test plan and results.

	Can Ping pc 1	Can Ping pc 2	Raspberry	Can Ping pc 4	Can Fetch the web page
PC 1	■	Yes	no	no	no
PC 2	Yes	■	no	no	Yes
RaspberryPi	Yes	Yes	■	Yes	Yes
PC 4	Yes	Yes	Yes	■	Yes

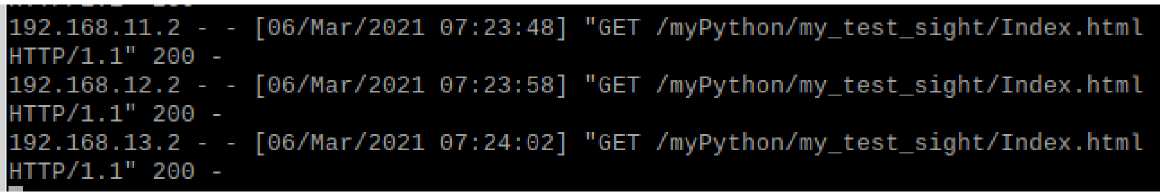
This color field means that this communication should not possible
This color field means that this communication should be possible

Yes, means that it succeeds
No, means it failed

7.2 Proof of connection.

In this chapter the different conditions of the computers will be tested.

7.2.1 What RaspberryPi shows us

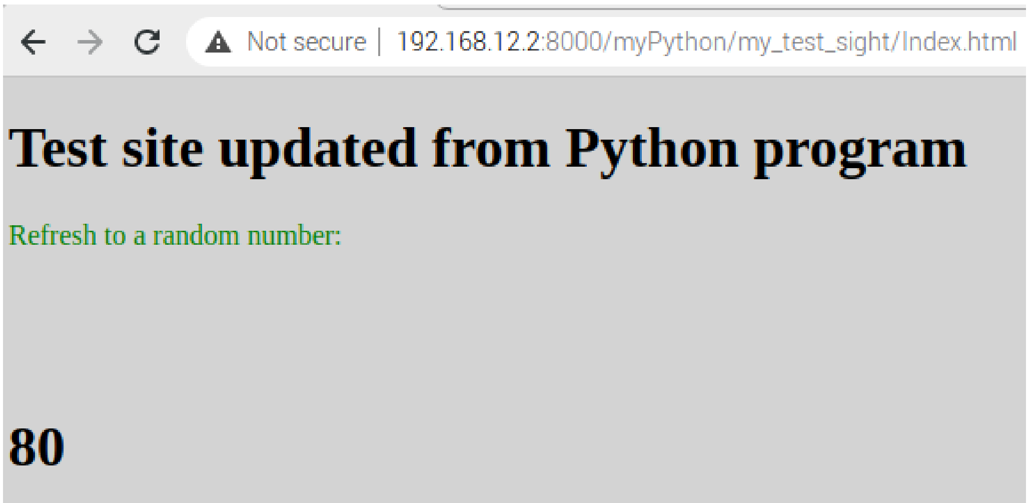


```
192.168.11.2 - - [06/Mar/2021 07:23:48] "GET /myPython/my_test_sight/Index.html
HTTP/1.1" 200 -
192.168.12.2 - - [06/Mar/2021 07:23:58] "GET /myPython/my_test_sight/Index.html
HTTP/1.1" 200 -
192.168.13.2 - - [06/Mar/2021 07:24:02] "GET /myPython/my_test_sight/Index.html
HTTP/1.1" 200 -
```

In this first figure of the terminal that is running the http server, it can be observed that 3 IP-addresses have connect to the webpage and that is:

1. 192.168.11.2 which is pc2
2. 192.168.12.2 which is the RaspberryPi
3. 192.168.13.2 which is pc4

And that is exactly what is in the table in chapter **7.2 Test plan and results**.



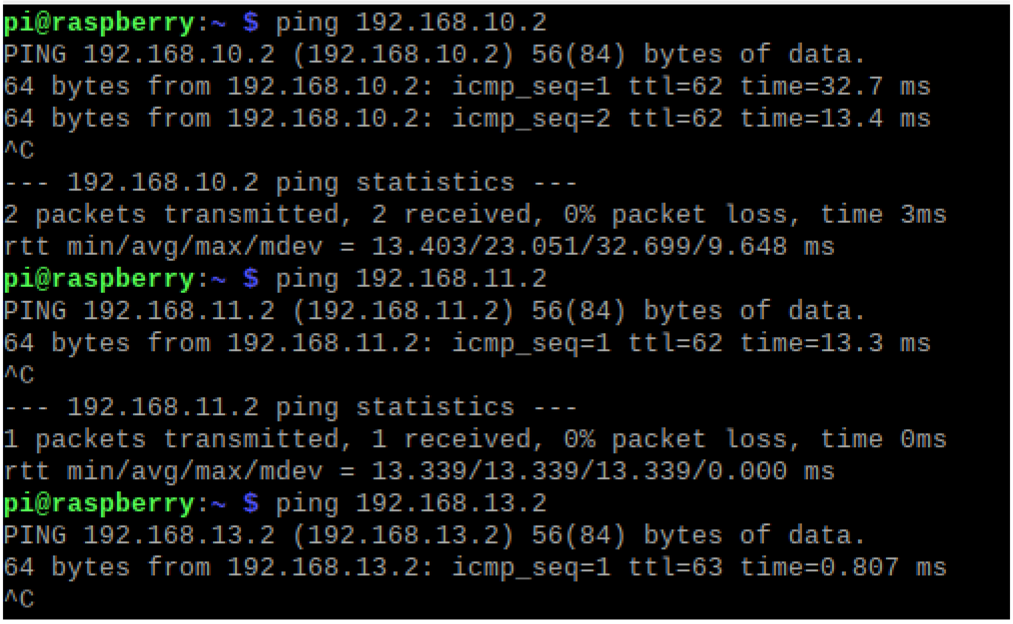
Not secure | 192.168.12.2:8000/myPython/my_test_sight/Index.html

Test site updated from Python program

Refresh to a random number:

80

And the Webpage look like this on the RaspberryPi

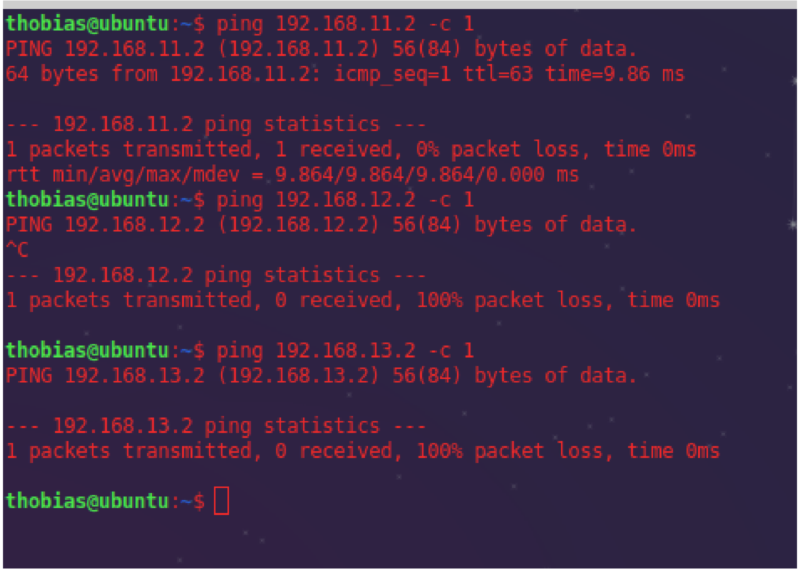


```
pi@raspberry:~ $ ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=62 time=32.7 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=62 time=13.4 ms
^C
--- 192.168.10.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 3ms
rtt min/avg/max/mdev = 13.403/23.051/32.699/9.648 ms
pi@raspberry:~ $ ping 192.168.11.2
PING 192.168.11.2 (192.168.11.2) 56(84) bytes of data.
64 bytes from 192.168.11.2: icmp_seq=1 ttl=62 time=13.3 ms
^C
--- 192.168.11.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.339/13.339/13.339/0.000 ms
pi@raspberry:~ $ ping 192.168.13.2
PING 192.168.13.2 (192.168.13.2) 56(84) bytes of data.
64 bytes from 192.168.13.2: icmp_seq=1 ttl=63 time=0.807 ms
^C
```

And here the RaspberryPi has:

- A successfully Ping to computer 1 (192.168.10.2)
- A successfully Ping to computer 2 (192.168.11.2)
- A successfully Ping to computer 4 (192.168.13.2)

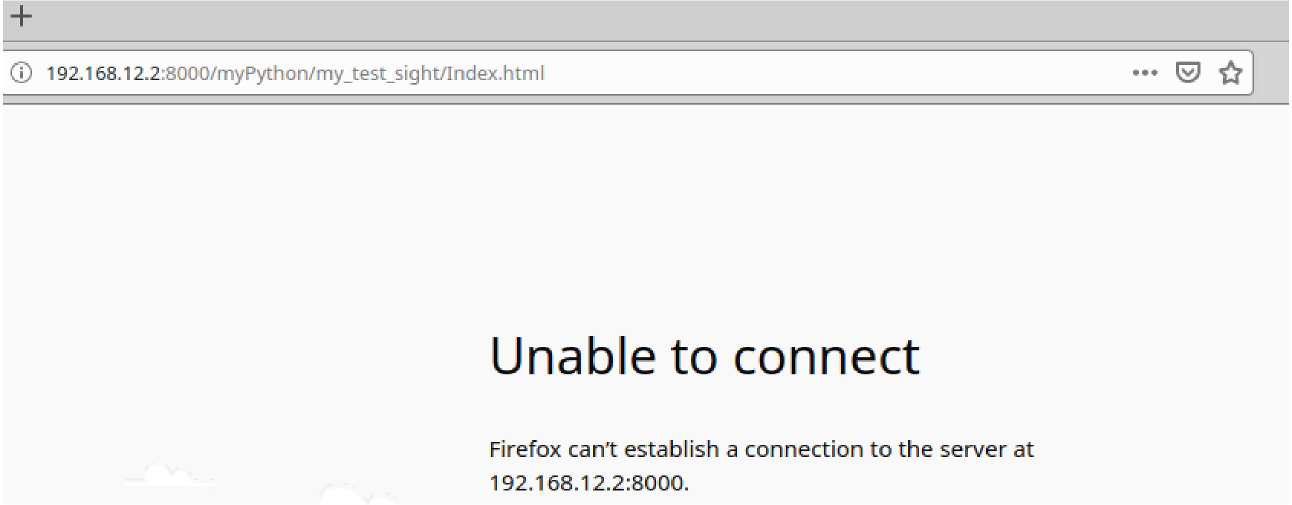
7.2.2 What PC1 shows us



```
thobias@ubuntu:~$ ping 192.168.11.2 -c 1
PING 192.168.11.2 (192.168.11.2) 56(84) bytes of data.
64 bytes from 192.168.11.2: icmp_seq=1 ttl=63 time=9.86 ms

--- 192.168.11.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 9.864/9.864/9.864/0.000 ms
thobias@ubuntu:~$ ping 192.168.12.2 -c 1
PING 192.168.12.2 (192.168.12.2) 56(84) bytes of data.
^C
--- 192.168.12.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

thobias@ubuntu:~$ ping 192.168.13.2 -c 1
PING 192.168.13.2 (192.168.13.2) 56(84) bytes of data.

--- 192.168.13.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

thobias@ubuntu:~$ 
```

And here the Computer 1 has:

- successfully Pinged to computer 2 (192.168.11.2)
- failed Pinging computer 3 (192.168.12.2)
- failed Pinging computer 4 (192.168.13.2)

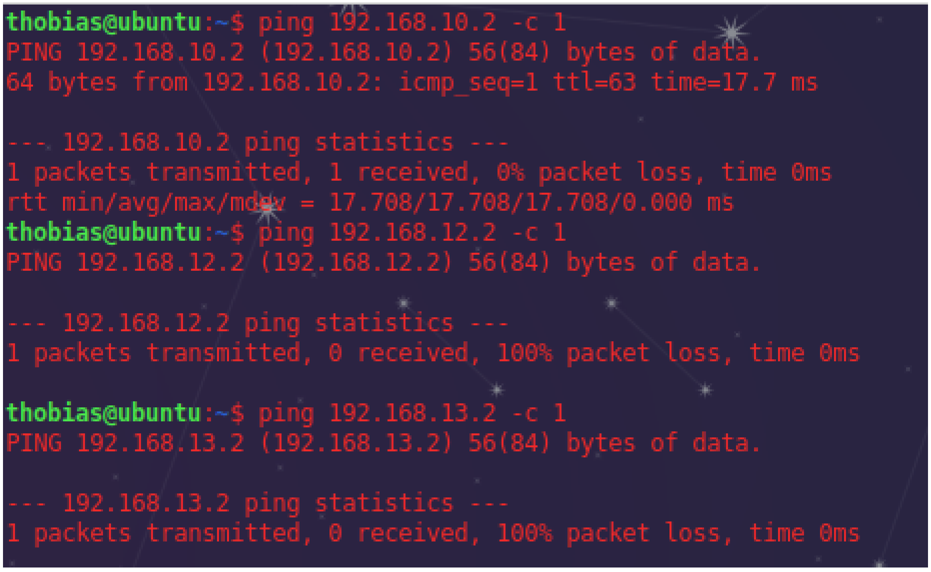


Unable to connect

Firefox can't establish a connection to the server at 192.168.12.2:8000.

And PC1 is not allowed to connect to the Webpage

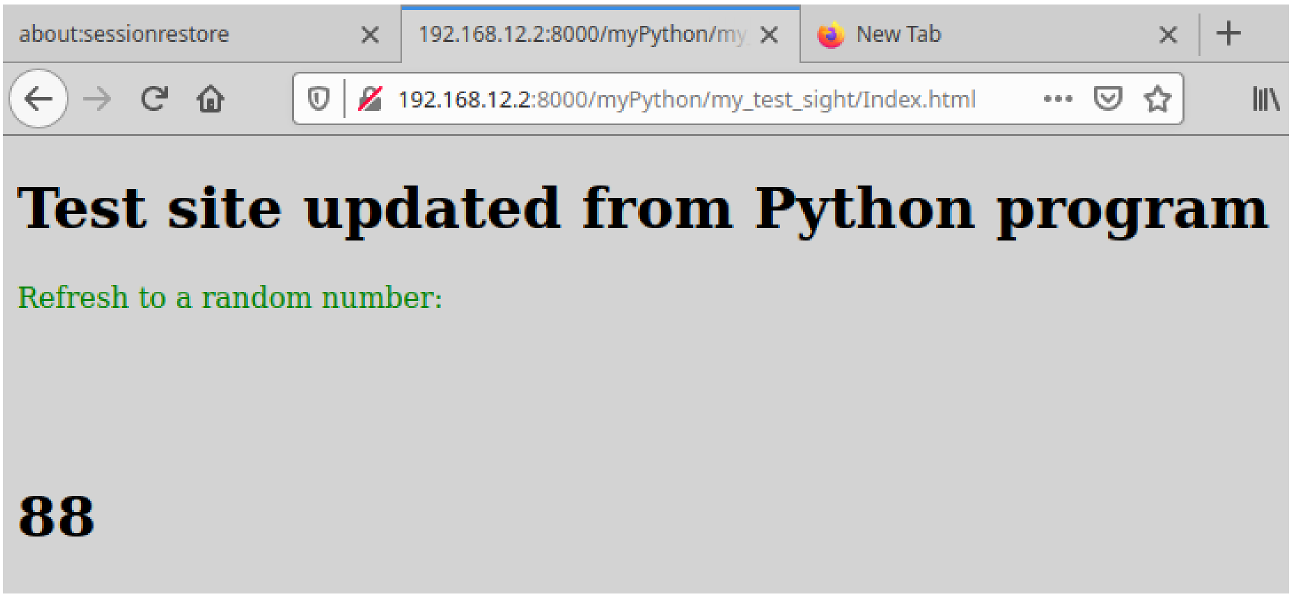
7.2.3 What PC2 shows us



```
thobias@ubuntu:~$ ping 192.168.10.2 -c 1
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=63 time=17.7 ms

--- 192.168.10.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 17.708/17.708/17.708/0.000 ms
thobias@ubuntu:~$ ping 192.168.12.2 -c 1
PING 192.168.12.2 (192.168.12.2) 56(84) bytes of data.

--- 192.168.12.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms

thobias@ubuntu:~$ ping 192.168.13.2 -c 1
PING 192.168.13.2 (192.168.13.2) 56(84) bytes of data.

--- 192.168.13.2 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
```

And here the Computer 2 has:

- successfully Pinged to computer 1 (192.168.10.2)
- failed Pinging computer 3 (192.168.12.2)
- failed Pinging computer 4 (192.168.13.2)


about:sessionrestore
192.168.12.2:8000/myPython/my
New Tab
192.168.12.2:8000/myPython/my_test_sight/Index.html

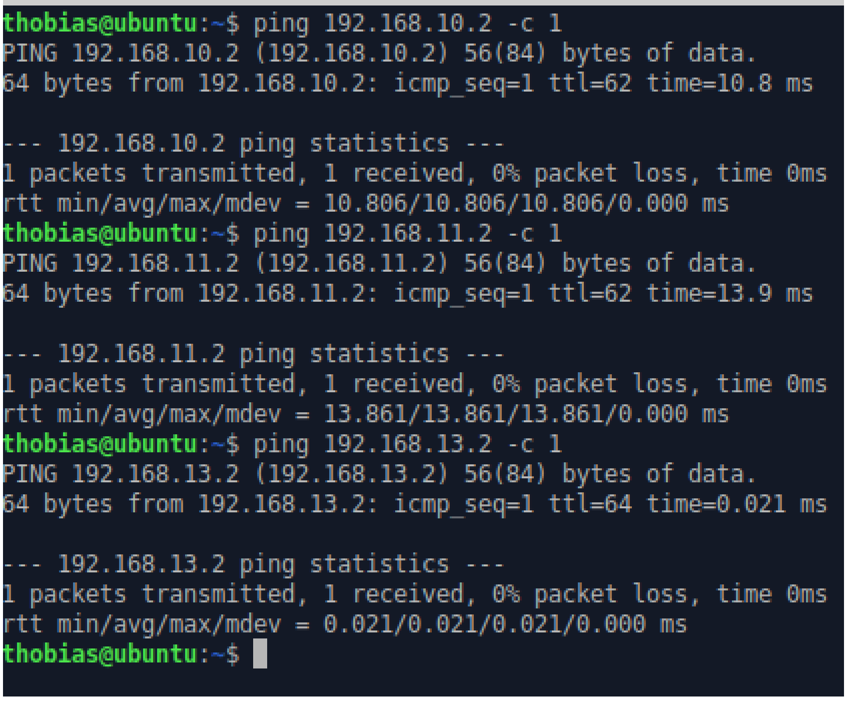
Test site updated from Python program

Refresh to a random number:

88

But here computer 2 has successfully connected to the webpage

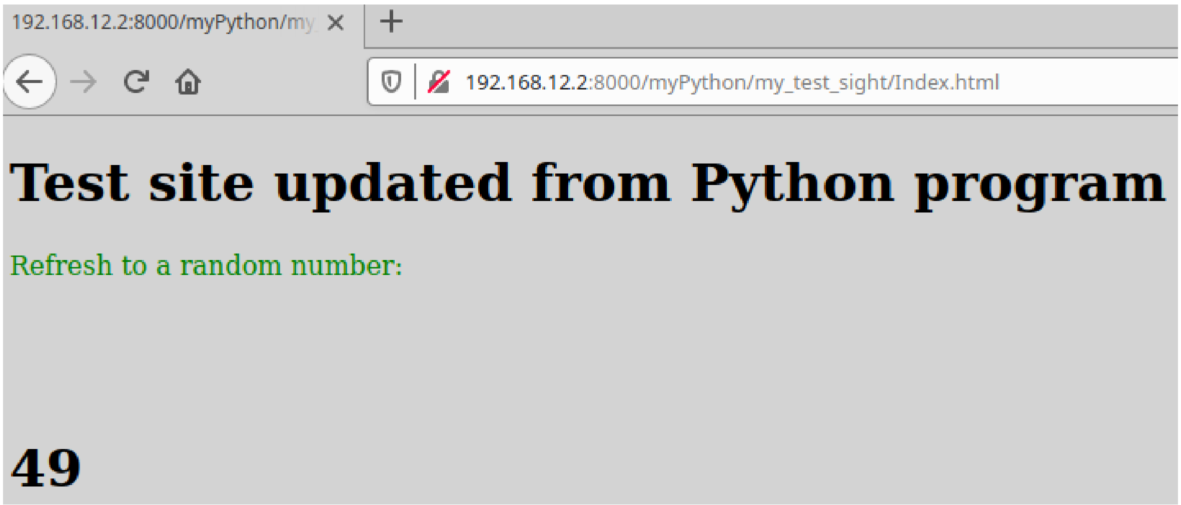
7.2.4 What PC4 shows us



```
thobias@ubuntu:~$ ping 192.168.10.2 -c 1
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=62 time=10.8 ms

--- 192.168.10.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 10.806/10.806/10.806/0.000 ms
thobias@ubuntu:~$ ping 192.168.11.2 -c 1
PING 192.168.11.2 (192.168.11.2) 56(84) bytes of data.
64 bytes from 192.168.11.2: icmp_seq=1 ttl=62 time=13.9 ms

--- 192.168.11.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 13.861/13.861/13.861/0.000 ms
thobias@ubuntu:~$ ping 192.168.13.2 -c 1
PING 192.168.13.2 (192.168.13.2) 56(84) bytes of data.
64 bytes from 192.168.13.2: icmp_seq=1 ttl=64 time=0.021 ms

--- 192.168.13.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.021/0.021/0.021/0.000 ms
thobias@ubuntu:~$ 
```

And here the Computer 4 has:

- successfully Pinged to computer 1 (192.168.10.2)
- successfully Pinged to computer 2 (192.168.11.2)
- successfully Pinged to RaspberryPi (192.168.12.2)


192.168.12.2:8000/myPython/my
192.168.12.2:8000/myPython/my_test_sight/Index.html

Test site updated from Python program

Refresh to a random number:

49

Computer 4 has successfully connected to the webpage.

8 Conclusion

This has been one of the more difficult assignments. This is based on the fact that there was a lack of documented guides, but also because that it has been a lot more self-learning experience. We came together as a team to troubleshoot and problem solve and we managed to grasp how to configure our networks with zones.