SDN-Based Stateless Firewall

Student Name: Tiriac Ioana-Raluca

Email: itiriac@asu.edu

Submission Date: 7.11.2021

Class Name and Term: CSE548 Fall B 2021

I. Project Overview

In this project a software defined environment based on mininet and containernet is created. This SDN is OpenFlow-based, and a flow-based SDN firewall implementation is needed. To achieve this flow-based firewall filtering policies are set in place to regulate traffic and L2 and L3 SDN controllers are used. POX OpenFlow controllers also need to be used. Techniques to test network traffic are set in place.

II. Network Setup

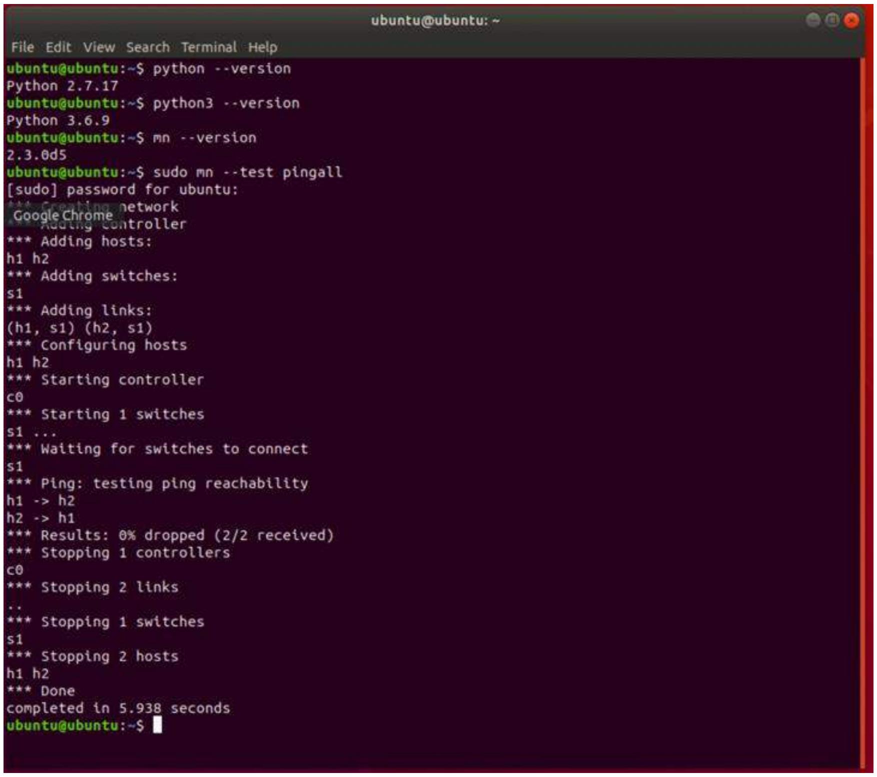
A new VM has been created using the given vdi image and following lab CS-SYS-00101 again this VM has been setup with NAT network (10.0.2.0/24).

III. Software

For the lab the following software has been used:

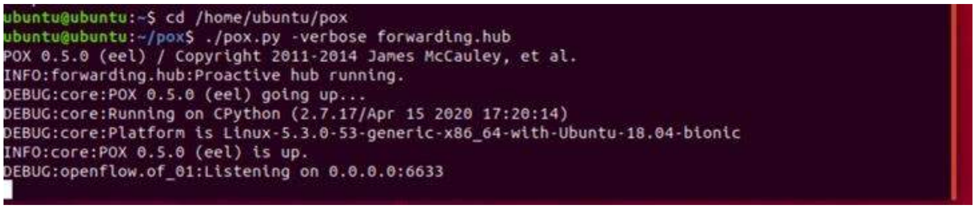
- Open vSwitch https://www.openvswitch.org/
- Mininet
- POX Controller
- ContainerNet
- packages installed: mininet, POX, OVS , Python & Python 3

“Mininet is **a software emulator for prototyping a large network on a single machine**. Mininet can be used to quickly create a realistic virtual network running actual kernel, switch and software application code on a personal computer.”
Containernet is a fork of the famous Mininet network emulator and allows to use Docker containers as hosts in emulated network topologies.



```
ubuntu@ubuntu:~$ python --version
Python 2.7.17
ubuntu@ubuntu:~$ python3 --version
Python 3.6.9
ubuntu@ubuntu:~$ mn --version
2.3.0d5
ubuntu@ubuntu:~$ sudo mn --test pingall
[sudo] password for ubuntu:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
s1
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 1 controllers
c0
*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 5.938 seconds
ubuntu@ubuntu:~$
```

Check mininet installation



```
ubuntu@ubuntu:~$ cd /home/ubuntu/pox
ubuntu@ubuntu:~/pox$ ./pox.py -verbose forwarding.hub
POX 0.5.0 (eel) / Copyright 2011-2014 James McCauley, et al.
INFO:forwarding.hub:Proactive hub running.
DEBUG:core:POX 0.5.0 (eel) going up...
DEBUG:core:Running on CPython (2.7.17/Apr 15 2020 17:20:14)
DEBUG:core:Platform is Linux-5.3.0-53-generic-x86_64-with-Ubuntu-18.04-bionic
INFO:core:POX 0.5.0 (eel) is up.
DEBUG:openflow.of_01:Listening on 0.0.0.0:6633
```

Check POX installation

IV. Project Description

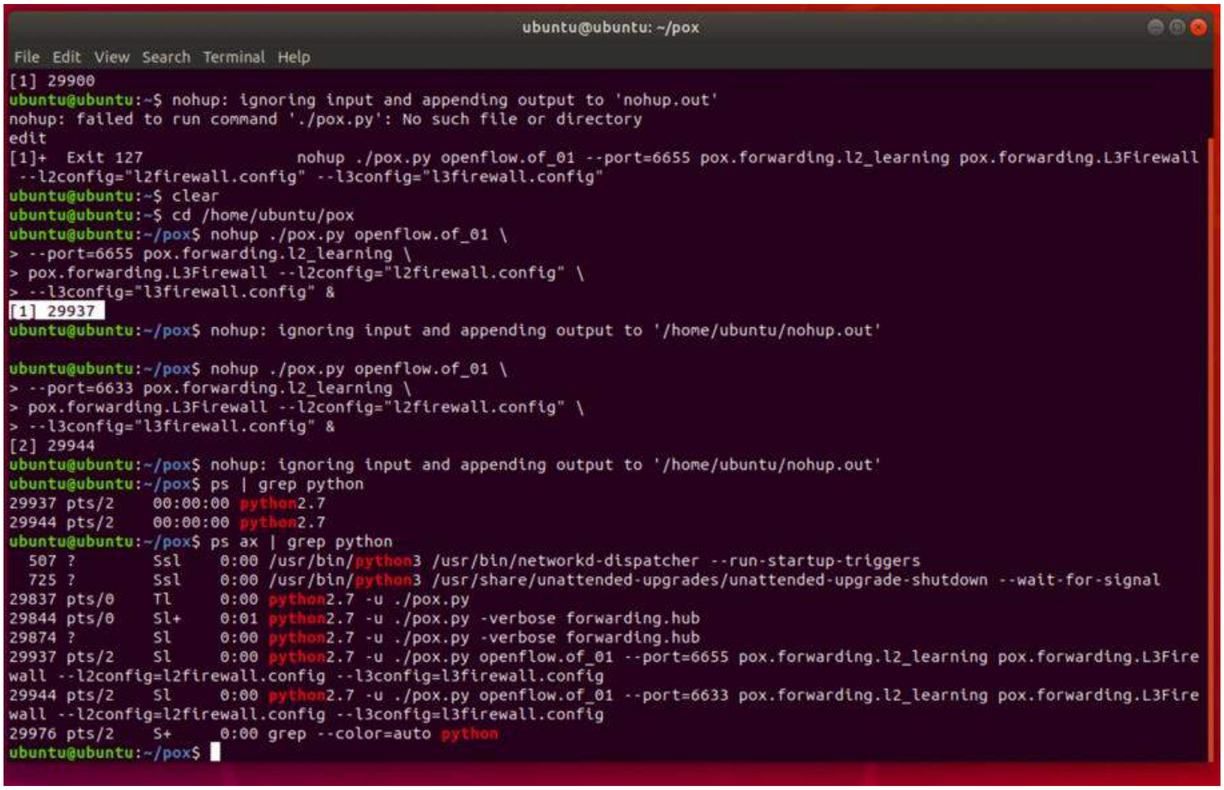
Lab assessments

1.) *(5 points) Create a mininet based topology with 4 container hosts and one controller switches and run it.*
   a. *Add link from controller1 to switch 1.*
   b. *Add link from controller2 to switch 1.*
   c. *Add link from switch 1 to container 1.*
   d. *Add link from switch 1 to container 2.*
   e. *Add link from switch 1 to container 3.*
   f. *Add link from switch 1 to container 4.*

First, we'll need to create two controllers with POX for port 6655 and 6633.

The commands used were:

```
nohup ./pox.py openflow.of_01 --port=6655 pox.forwarding.l2_learning --
l3config="l3firewall.config" &

nohup ./pox.py openflow.of_01 --port=6633 pox.forwarding.l2_learning --
l3config="l3firewall.config" &
```



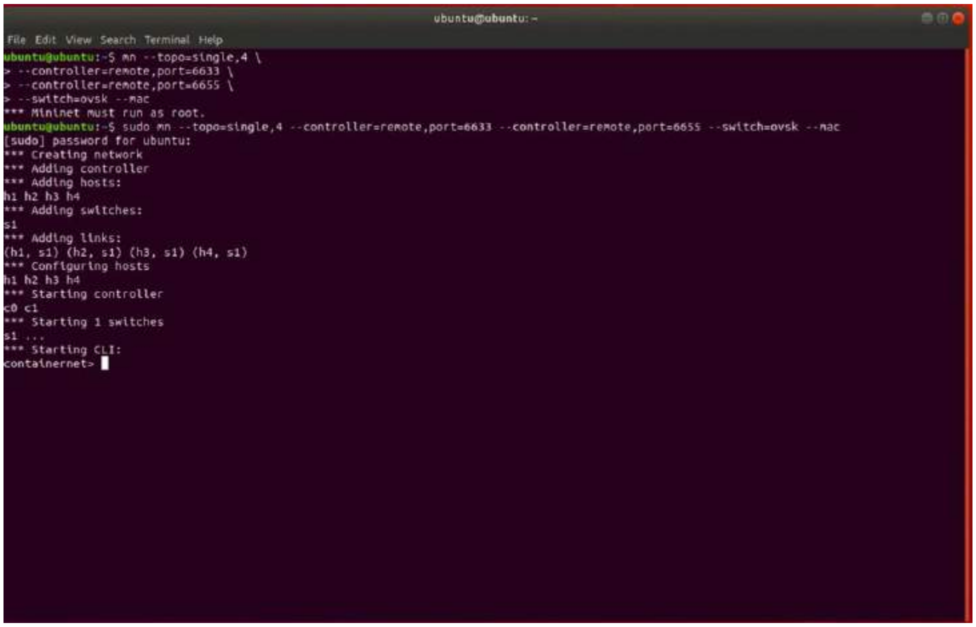
ubuntu@ubuntu: ~/pox

```
[1] 29900
ubuntu@ubuntu:~$ nohup: ignoring input and appending output to 'nohup.out'
nohup: failed to run command './pox.py': No such file or directory
edit
[1]+  Exit 127                nohup ./pox.py openflow.of_01 --port=6655 pox.forwarding.l2_learning pox.forwarding.L3Firewall
 --l2config="l2firewall.config" --l3config="l3firewall.config"
ubuntu@ubuntu:~$ clear
ubuntu@ubuntu:~$ cd /home/ubuntu/pox
ubuntu@ubuntu:~/pox$ nohup ./pox.py openflow.of_01 \
> --port=6655 pox.forwarding.l2_learning \
> pox.forwarding.L3Firewall --l2config="l2firewall.config" \
> --l3config="l3firewall.config" &
[1] 29937
ubuntu@ubuntu:~/pox$ nohup: ignoring input and appending output to '/home/ubuntu/nohup.out'

ubuntu@ubuntu:~/pox$ nohup ./pox.py openflow.of_01 \
> --port=6633 pox.forwarding.l2_learning \
> pox.forwarding.L3Firewall --l2config="l2firewall.config" \
> --l3config="l3firewall.config" &
[2] 29944
ubuntu@ubuntu:~/pox$ nohup: ignoring input and appending output to '/home/ubuntu/nohup.out'
ubuntu@ubuntu:~/pox$ ps | grep python
29937 pts/2    00:00:00 python2.7
29944 pts/2    00:00:00 python2.7
ubuntu@ubuntu:~/pox$ ps ax | grep python
  507 ?        Ssl    0:00 /usr/bin/python3 /usr/bin/networkd-dispatcher --run-startup-triggers
  725 ?        Ssl    0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
29837 pts/0    Tl     0:00 python2.7 -u ./pox.py
29844 pts/0    Sl+    0:01 python2.7 -u ./pox.py -verbose forwarding.hub
29874 ?        Sl     0:00 python2.7 -u ./pox.py -verbose forwarding.hub
29937 pts/2    Sl     0:00 python2.7 -u ./pox.py openflow.of_01 --port=6655 pox.forwarding.l2_learning pox.forwarding.L3Fire
wall --l2config=l2firewall.config --l3config=l3firewall.config
29944 pts/2    Sl     0:00 python2.7 -u ./pox.py openflow.of_01 --port=6633 pox.forwarding.l2_learning pox.forwarding.L3Fire
wall --l2config=l2firewall.config --l3config=l3firewall.config
29976 pts/2    S+     0:00 grep --color=auto python
ubuntu@ubuntu:~/pox$ 
```

We can see 2 separate processes have spawned up for the two pox commands, their IDs are 29937 and 29944 which we can check easily with *ps and filter with grep*, which means two pox controllers are running as processes at the same time at port 6655 and 6633.

Next, we'll run mininet using containernet in another terminal or window. We'll create the required mininet environment consisting of 4 containernet hosts, one OVS switch and two controllers running on port 6633 & 6655 that will bind to the 2 previously created POX controllers. As described in the lab the –mac option will assign small, unique & fixed set of mac address based on host id. The command used is :

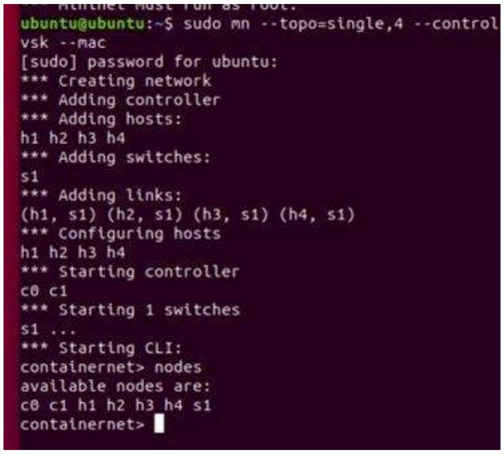
```
mn --topo=single,4 --controller=remote,port=6633 --controller=remote,port=6655
--switch=ovsk --mac
```



ubuntu@ubuntu: ~

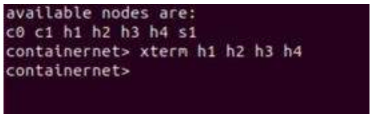
```
ubuntu@ubuntu:~$ mn --topo=single,4 \
> --controller=remote,port=6633 \
> --controller=remote,port=6655 \
> --switch=ovsk --mac
*** Mininet must run as root.
ubuntu@ubuntu:~$ sudo mn --topo=single,4 --controller=remote,port=6633 --controller=remote,port=6655 --switch=ovsk --mac
[sudo] password for ubuntu:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0 c1
*** Starting 1 switches
s1 ...
*** Starting CLI:
containernet> 
```

By using nodes command available nodes (or created topology) shows up :

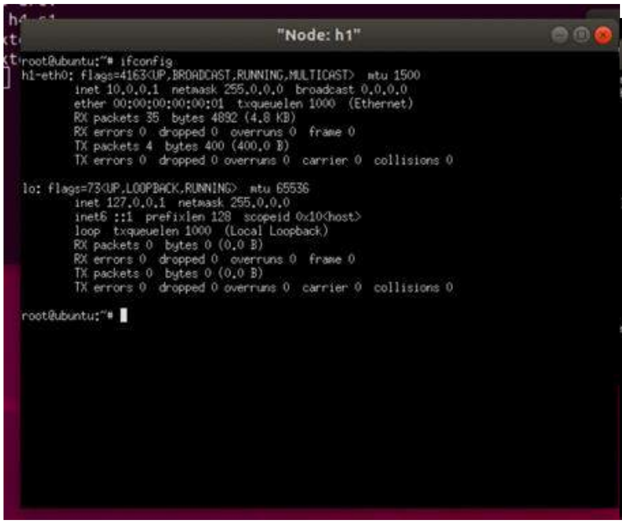


```
ubuntu@ubuntu:~$ sudo mn --topo=single,4 --control
vsk --mac
[sudo] password for ubuntu:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0 c1
*** Starting 1 switches
s1 ...
*** Starting CLI:
containernet> nodes
available nodes are:
c0 c1 h1 h2 h3 h4 s1
containernet>
```

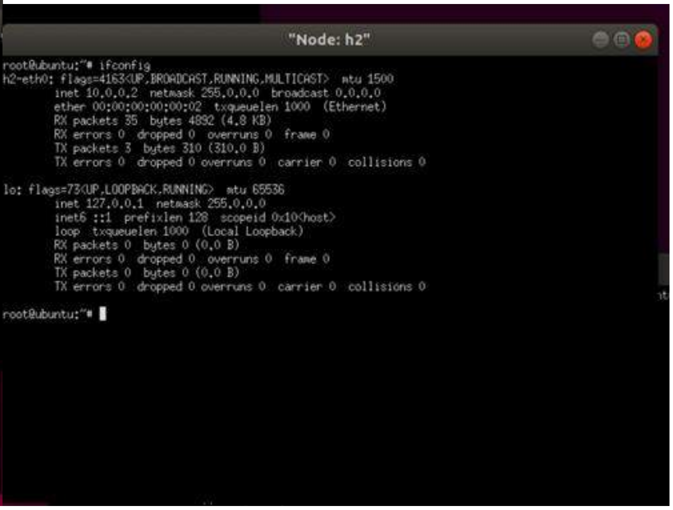
And by using xterm h1 h2 h3 h4 – for all 4 containers, we get access to the CLI of each container on which ifconfig has been run to see network interface & IP :



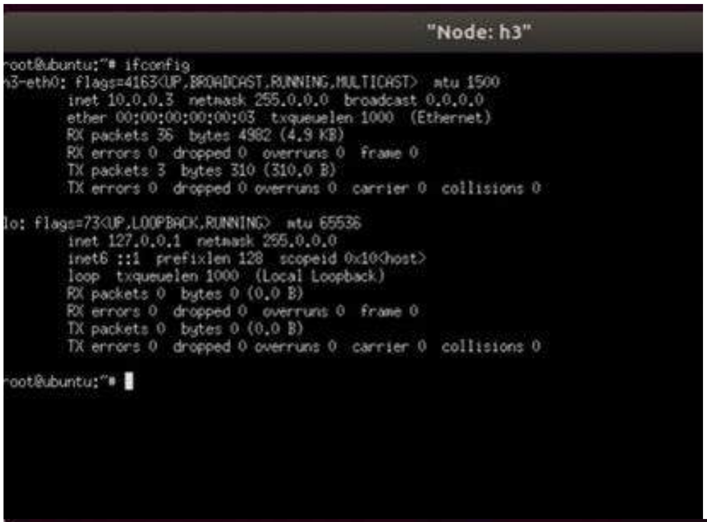
```
available nodes are:
c0 c1 h1 h2 h3 h4 s1
containernet> xterm h1 h2 h3 h4
containernet>
```



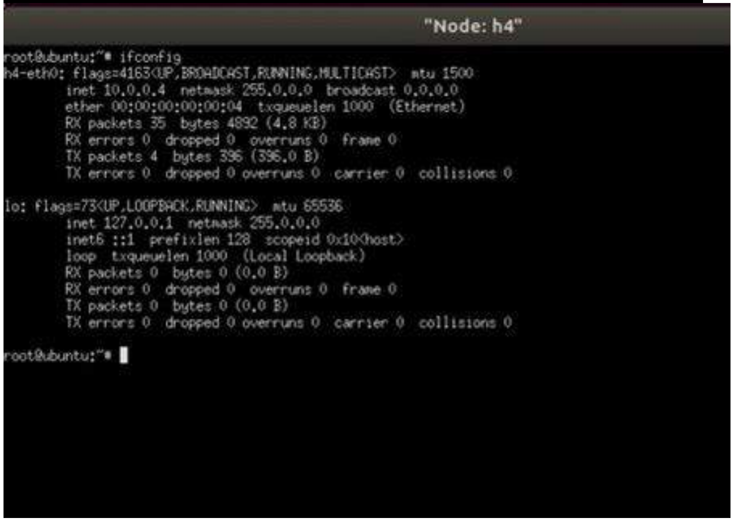
```
root@ubuntu:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.0.0.0  broadcast 0.0.0.0
        ether 00:00:00:00:00:01  txqueuelen 1000  (Ethernet)
        RX packets 35  bytes 4892 (4.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~#
```



```
root@ubuntu:~# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.0.0.0  broadcast 0.0.0.0
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 35  bytes 4892 (4.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 310 (310.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~#
```



```
root@ubuntu:~# ifconfig
h3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.3  netmask 255.0.0.0  broadcast 0.0.0.0
        ether 00:00:00:00:00:03  txqueuelen 1000  (Ethernet)
        RX packets 36  bytes 4982 (4.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 310 (310.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```



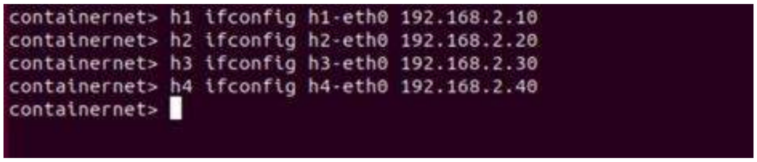
```
root@ubuntu:~# ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.4  netmask 255.0.0.0  broadcast 0.0.0.0
        ether 00:00:00:00:00:04  txqueuelen 1000  (Ethernet)
        RX packets 35  bytes 4892 (4.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```

We can see their IP addresses are 10.0.0.1, 10.0.0.2, 10.0.0.3 and 10.0.0.4 respectively.

2.) ***Make the interfaces up and assign IP addresses to interfaces of container hosts.***

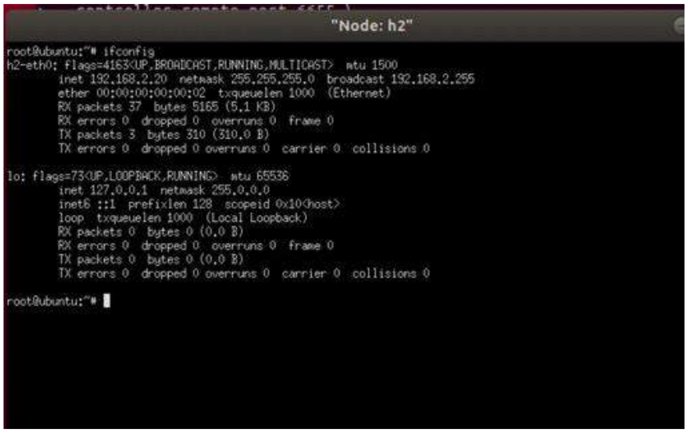
- ***Assign IP address 192.168.2.10 to container host #1.***
- ***Assign IP address 192.168.2.20 to container host #2.***
- ***Assign IP address 192.168.2.30 to container host #3.***
- ***Assign IP address 192.168.2.40 to container host #4.***

The IP addresses are configured from the containernet command line using the following commands :



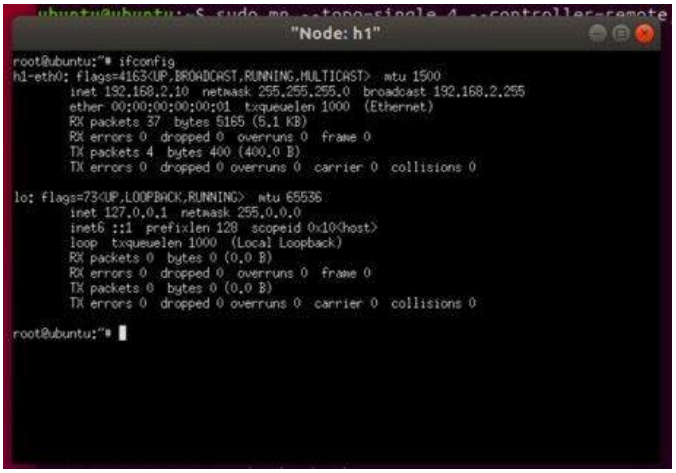
```
containernet> h1 ifconfig h1-eth0 192.168.2.10
containernet> h2 ifconfig h2-eth0 192.168.2.20
containernet> h3 ifconfig h3-eth0 192.168.2.30
containernet> h4 ifconfig h4-eth0 192.168.2.40
containernet> 
```

We check our containers IP’s again by using xterm & ifconfig for each container CLI :



```
root@ubuntu:~# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.20  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 37  bytes 5165 (5.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 310 (310.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```

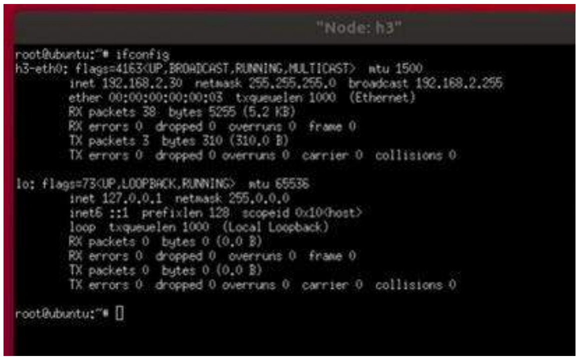
h2: 192.168.2.20



```
ubuntu@ubuntu:~$ sudo mn --topo=single,4 --controller=remote
```

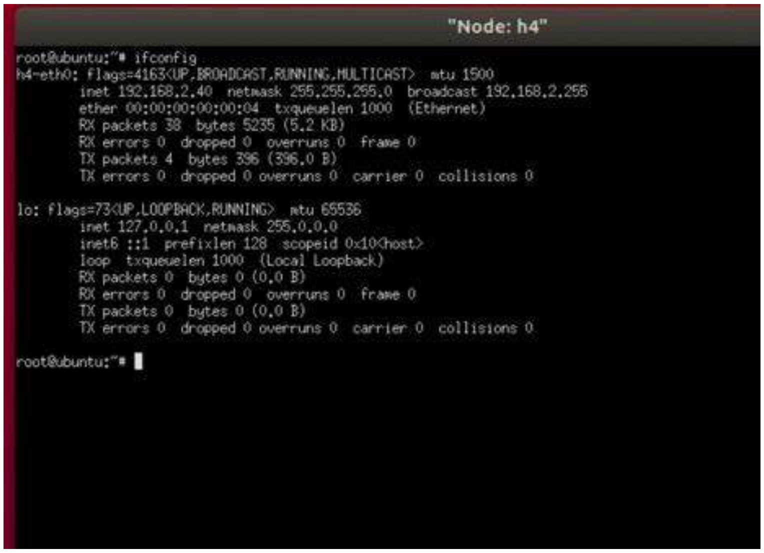
```
root@ubuntu:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:01  txqueuelen 1000  (Ethernet)
        RX packets 37  bytes 5165 (5.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 400 (400.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```

h1: 192.168.2.10



```
root@ubuntu:~# ifconfig
h3-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.30  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:03  txqueuelen 1000  (Ethernet)
        RX packets 38  bytes 5255 (5.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3  bytes 310 (310.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```

h3: 192.168.2.30



```
root@ubuntu:~# ifconfig
h4-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.40  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:04  txqueuelen 1000  (Ethernet)
        RX packets 38  bytes 5235 (5.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# 
```

h4: 192.168.2.40

3.) ***(15 points) Add new rule to l3config file for blocking ICMP traffic from source IP 192.168.2.10 and destination IP 192.168.2.30.***

4.) ***(15 points) Add new rule to l3config file for blocking ICMP traffic from source IP 192.168.2.20 and desti- nation IP 192.168.2.40.***

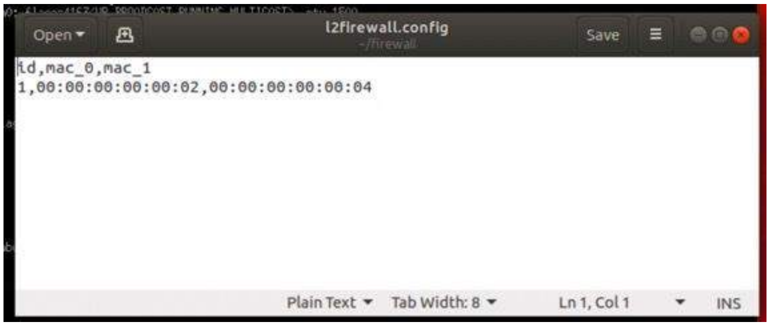
5.) ***(15 points) Add new rule to l3config file for blocking HTTP traffic from source IP 192.168.2.20.***

6.) ***(15 points) Add new rule to l2config file for blocking traffic from MAC address 00:00:00:00:00:02 to desti- nation MAC address 00:00:00:00:00:04.***

7.) ***(15 points) Add new rule to l3config file for blocking tcp traffic from 192.168.2.10 to 192.168.2.20.***

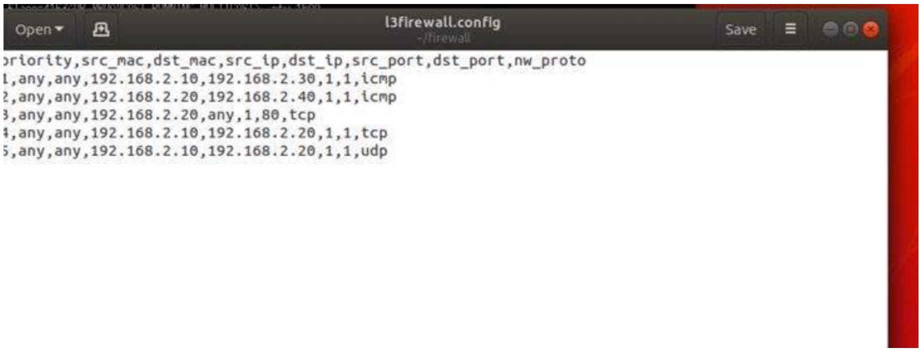
8.) ***(15 points) Add new rule to l3config file for blocking udp traffic from 192.168.2.10 to 192.168.2.20.***

The following is a screenshot of *l2firewall.config* containing rules for layer 2 at MAC address level, blocked traffic between source MAC address to destination MAC address:



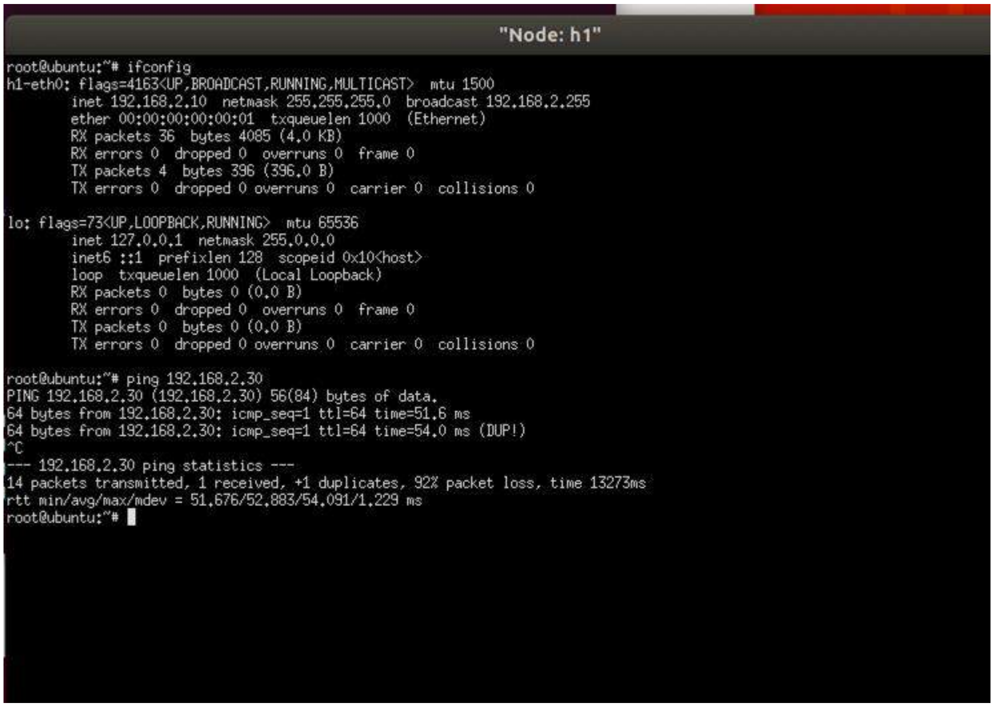
```
id,mac_0,mac_1
1,00:00:00:00:00:02,00:00:00:00:00:04
```

The following is a screenshot of *l3firewall.config* containing flow rules for layer3 describing blocked traffic:



```
priority,src_mac,dst_mac,src_ip,dst_ip,src_port,dst_port,nw_proto
1,any,any,192.168.2.10,192.168.2.30,1,1,icmp
2,any,any,192.168.2.20,192.168.2.40,1,1,icmp
3,any,any,192.168.2.20,any,1,80,tcp
4,any,any,192.168.2.10,192.168.2.20,1,1,tcp
5,any,any,192.168.2.10,192.168.2.20,1,1,udp
```

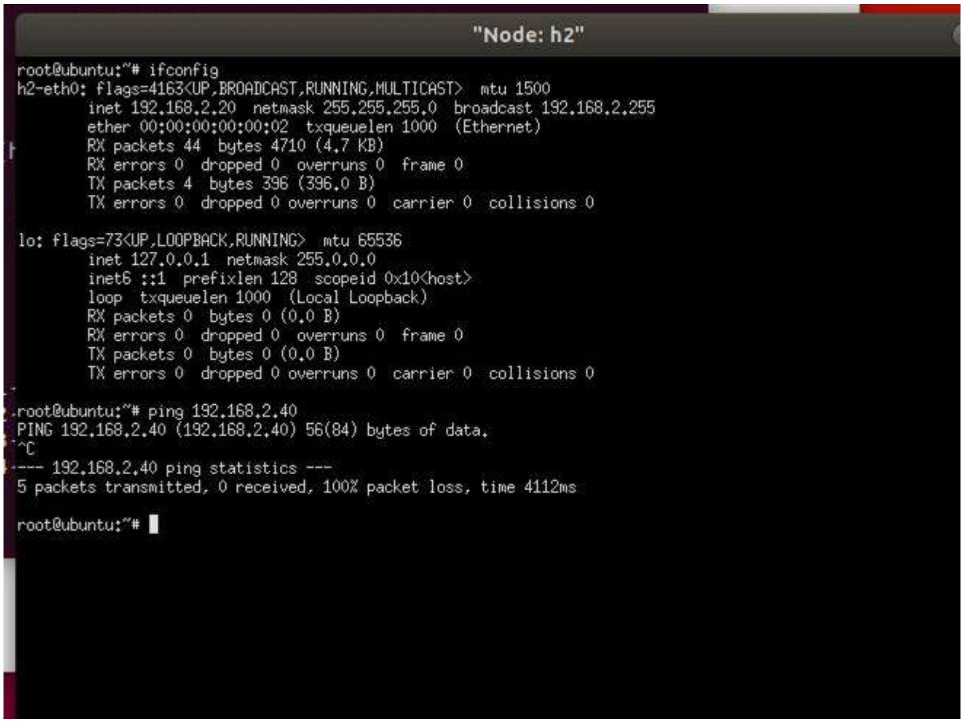
Then pox and mininet need to be restarted.
Let’s now test the firewall rules one by one. First, we’ll ping host3 from host1 to see if ICMP traffic has been blocked.



```
root@ubuntu:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:01  txqueuelen 1000  (Ethernet)
        RX packets 36  bytes 4085 (4.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# ping 192.168.2.30
PING 192.168.2.30 (192.168.2.30) 56(84) bytes of data.
64 bytes from 192.168.2.30: icmp_seq=1 ttl=64 time=51.6 ms
64 bytes from 192.168.2.30: icmp_seq=1 ttl=64 time=54.0 ms (DUP!)
^C
--- 192.168.2.30 ping statistics ---
14 packets transmitted, 1 received, +1 duplicates, 92% packet loss, time 13273ms
rtt min/avg/max/mdev = 51.676/52.883/54.091/1.229 ms
root@ubuntu:~# 
```

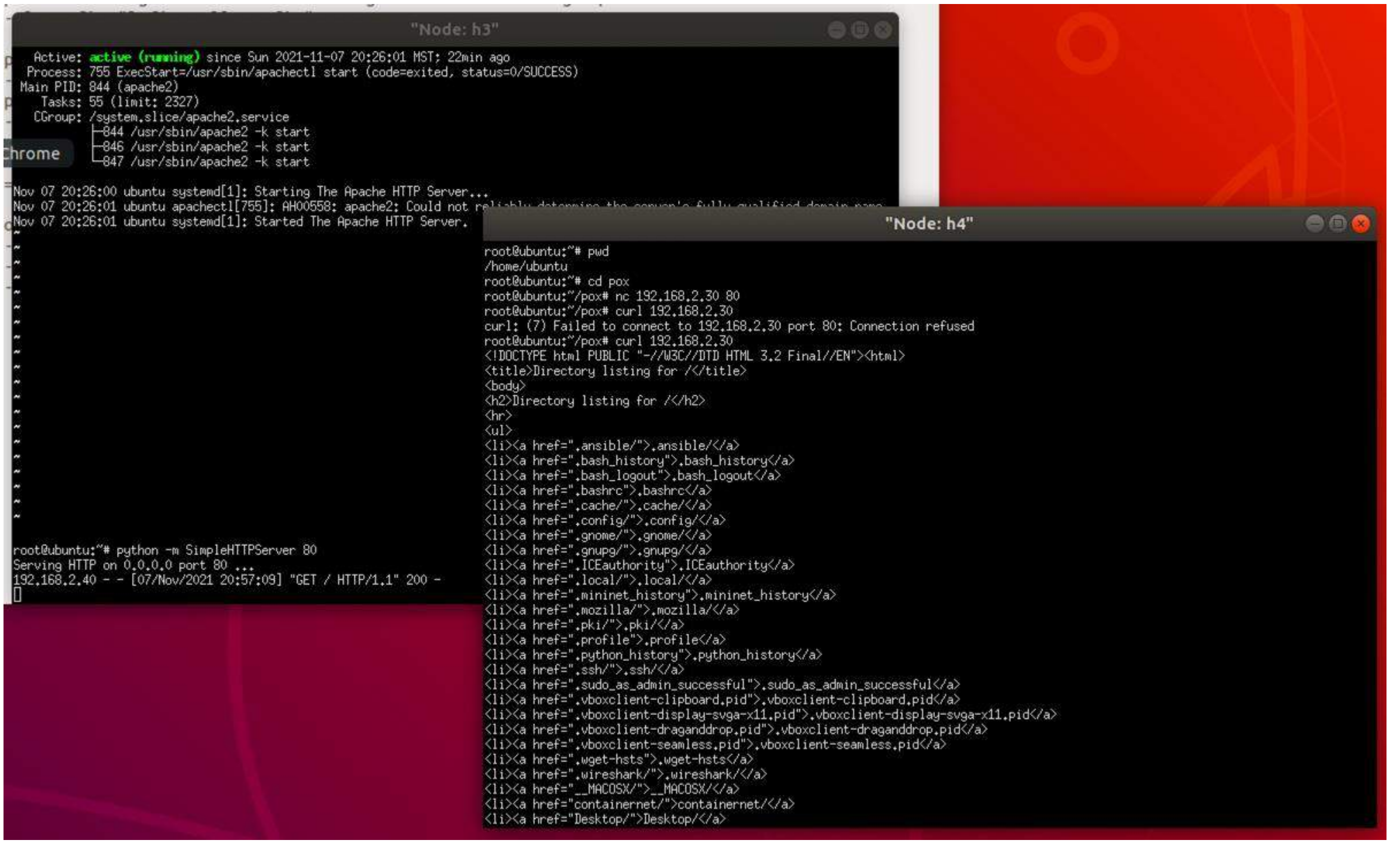
We can see that starting with the second packet sent the traffic is blocked and there is 92% packet loss.

Next, pinging host4 from host2 we notice all traffic is blocked and there is 100% packet loss.



```
root@ubuntu:~# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.20  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 44  bytes 4710 (4.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# ping 192.168.2.40
PING 192.168.2.40 (192.168.2.40) 56(84) bytes of data.
^C
--- 192.168.2.40 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4112ms

root@ubuntu:~# 
```

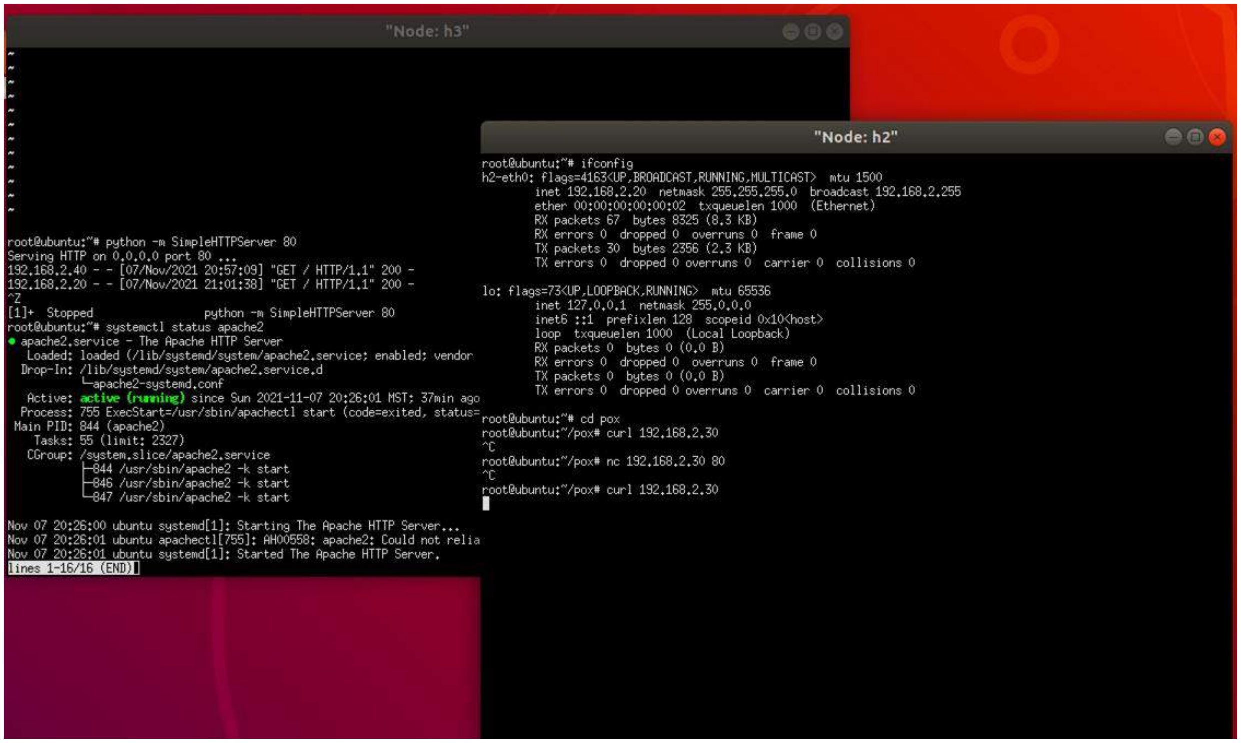
Started a web server on host3 using python SimpleHttpServer on port 80 and retrieved content from the web server using host4 that had no problem connecting to it :



```
  Active: active (running) since Sun 2021-11-07 20:26:01 MST; 22min ago
 Process: 755 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
Main PID: 844 (apache2)
   Tasks: 55 (limit: 2327)
  CGroup: /system.slice/apache2.service
          ├─844 /usr/sbin/apache2 -k start
          ├─846 /usr/sbin/apache2 -k start
          └─847 /usr/sbin/apache2 -k start

Nov 07 20:26:00 ubuntu systemd[1]: Starting The Apache HTTP Server...
Nov 07 20:26:01 ubuntu apachectl[755]: AH00558: apache2: Could not re
Nov 07 20:26:01 ubuntu systemd[1]: Started The Apache HTTP Server.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~

root@ubuntu:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.2.40 - - [07/Nov/2021 20:57:09] "GET / HTTP/1.1" 200 -
```

```
root@ubuntu:~# pwd
/home/ubuntu
root@ubuntu:~# cd pox
root@ubuntu:~/pox# nc 192.168.2.30 80
root@ubuntu:~/pox# curl 192.168.2.30
curl: (7) Failed to connect to 192.168.2.30 port 80: Connection refused
root@ubuntu:~/pox# curl 192.168.2.30
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"><html>
<title>Directory listing for /</title>
<body>
<h2>Directory listing for /</h2>
<hr>
<ul>
<li><a href=".ansible/">.ansible/</a>
<li><a href=".bash_history">.bash_history</a>
<li><a href=".bash_logout">.bash_logout</a>
<li><a href=".bashrc">.bashrc</a>
<li><a href=".cache/">.cache/</a>
<li><a href=".config/">.config/</a>
<li><a href=".gnome/">.gnome/</a>
<li><a href=".gnupg/">.gnupg/</a>
<li><a href=".ICEauthority">.ICEauthority</a>
<li><a href=".local/">.local/</a>
<li><a href=".mininet_history">.mininet_history</a>
<li><a href=".mozilla/">.mozilla/</a>
<li><a href=".pki/">.pki/</a>
<li><a href=".profile">.profile</a>
<li><a href=".python_history">.python_history</a>
<li><a href=".ssh/">.ssh/</a>
<li><a href=".sudo_as_admin_successful">.sudo_as_admin_successful</a>
<li><a href=".vboxclient-clipboard.pid">.vboxclient-clipboard.pid</a>
<li><a href=".vboxclient-display-svga-x11.pid">.vboxclient-display-svga-x11.pid</a>
<li><a href=".vboxclient-draganddrop.pid">.vboxclient-draganddrop.pid</a>
<li><a href=".vboxclient-seamless.pid">.vboxclient-seamless.pid</a>
<li><a href=".wget-hsts">.wget-hsts</a>
<li><a href=".wireshark/">.wireshark/</a>
<li><a href="__MACOSX/">__MACOSX/</a>
<li><a href="containernet/">containernet/</a>
<li><a href="Desktop/">Desktop/</a>
```

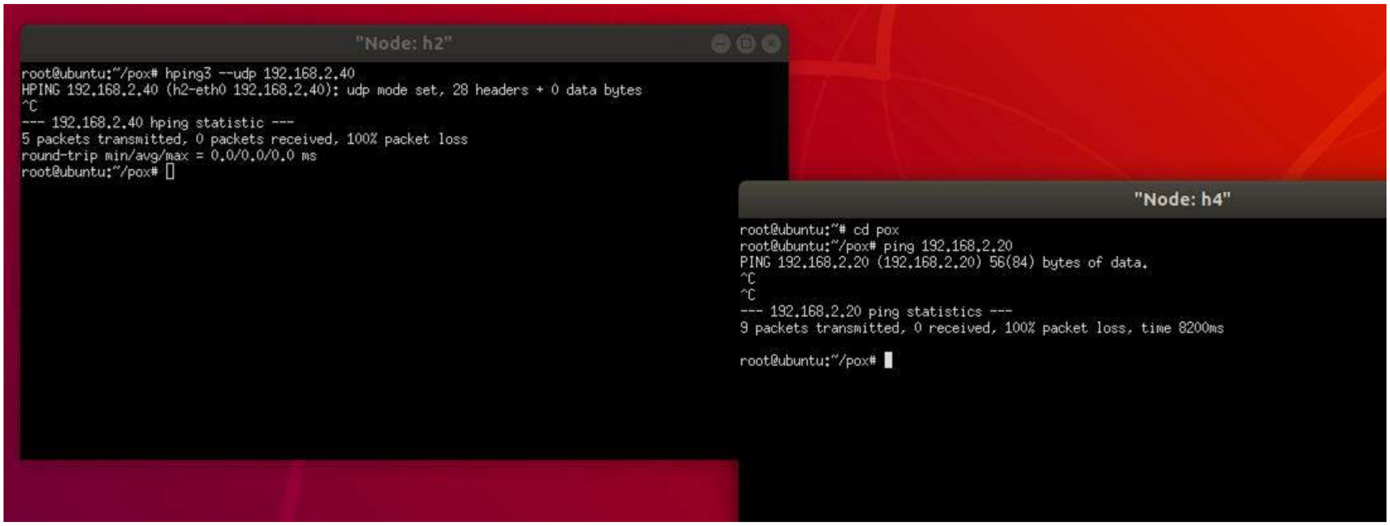
However Host2 is not able to receive any HTTP traffic and thus trying to reach the web server and access content from host 2 is not possible:



```
~
~
~
~
~
~
~
~
~
~
~
~

root@ubuntu:~# python -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
192.168.2.40 - - [07/Nov/2021 20:57:09] "GET / HTTP/1.1" 200 -
192.168.2.20 - - [07/Nov/2021 21:01:38] "GET / HTTP/1.1" 200 -
^Z
[1]+  Stopped                 python -m SimpleHTTPServer 80
root@ubuntu:~# systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
   Active: active (running) since Sun 2021-11-07 20:26:01 MST; 37min ago
  Process: 755 ExecStart=/usr/sbin/apachectl start (code=exited, status=
 Main PID: 844 (apache2)
    Tasks: 55 (limit: 2327)
   CGroup: /system.slice/apache2.service
           ├─844 /usr/sbin/apache2 -k start
           ├─846 /usr/sbin/apache2 -k start
           └─847 /usr/sbin/apache2 -k start

Nov 07 20:26:00 ubuntu systemd[1]: Starting The Apache HTTP Server...
Nov 07 20:26:01 ubuntu apachectl[755]: AH00558: apache2: Could not relia
Nov 07 20:26:01 ubuntu systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

```
root@ubuntu:~# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.20  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 67  bytes 8325 (8.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 30  bytes 2356 (2.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# cd pox
root@ubuntu:~/pox# curl 192.168.2.30
^C
root@ubuntu:~/pox# nc 192.168.2.30 80
^C
root@ubuntu:~/pox# curl 192.168.2.30
```

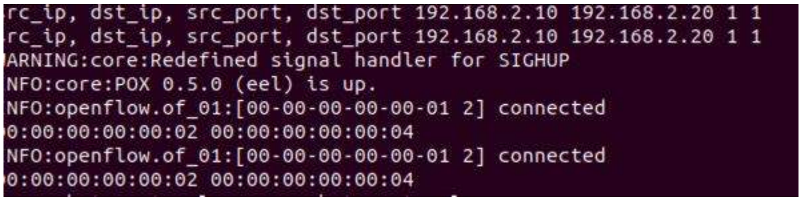
To demonstrate no traffic is allowed between Mac address of host2 and host4, host2 uses udp to ping (hping3) host4 but no traffic is allowed as visible from screenshot.



```
root@ubuntu:~/pox# hping3 --udp 192.168.2.40
HPING 192.168.2.40 (h2-eth0 192.168.2.40): udp mode set, 28 headers + 0 data bytes
^C
--- 192.168.2.40 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@ubuntu:~/pox# 
```

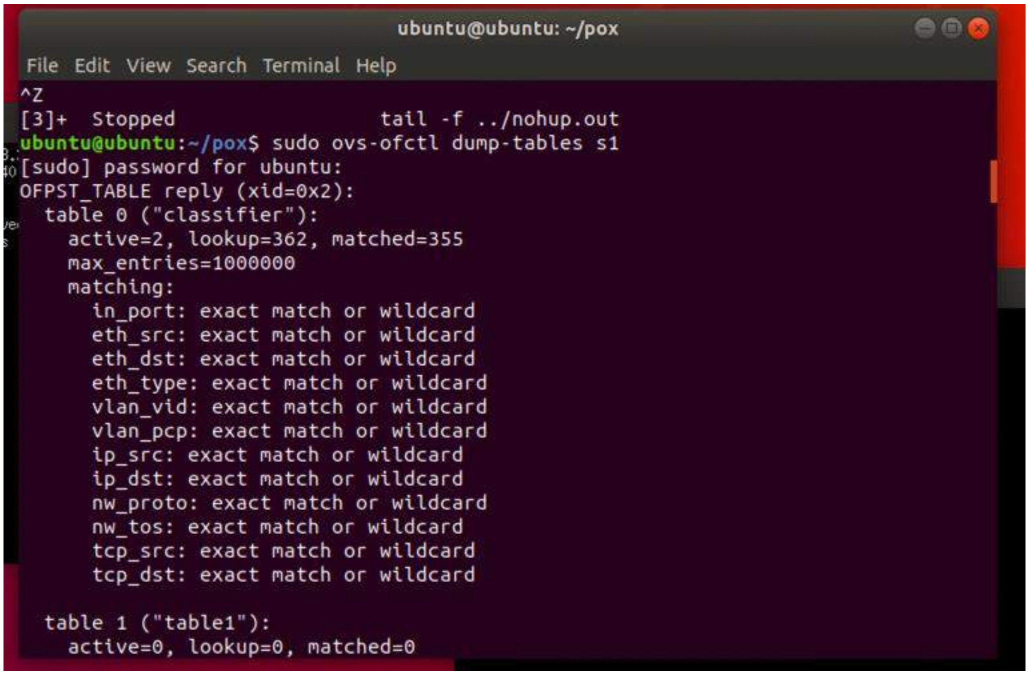
```
root@ubuntu:~# cd pox
root@ubuntu:~/pox# ping 192.168.2.20
PING 192.168.2.20 (192.168.2.20) 56(84) bytes of data.
^C
^C
--- 192.168.2.20 ping statistics ---
9 packets transmitted, 0 received, 100% packet loss, time 8200ms

root@ubuntu:~/pox# 
```

The enforced blocked traffic between the 2 MAC addresses can be found inside the nohup.out file of the pox controller:



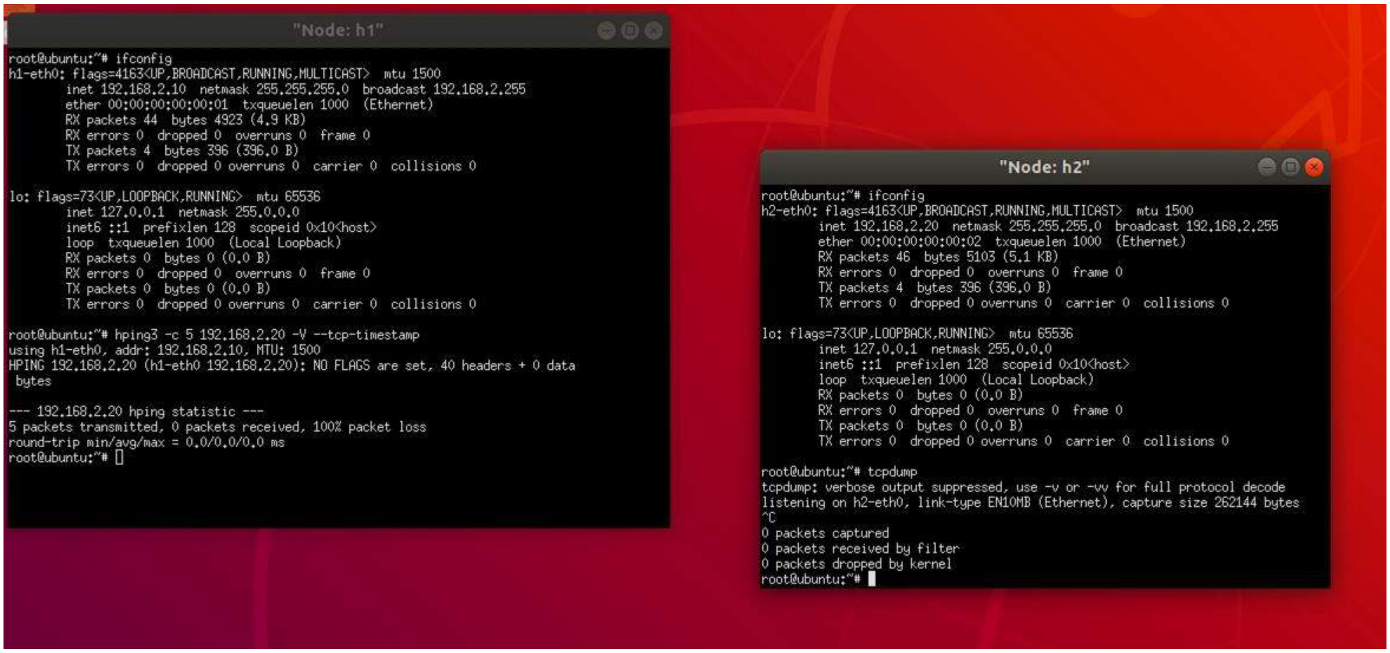
```
rc_ip, dst_ip, src_port, dst_port 192.168.2.10 192.168.2.20 1 1
rc_ip, dst_ip, src_port, dst_port 192.168.2.10 192.168.2.20 1 1
ARNING:core:Redefined signal handler for SIGHUP
NFO:core:POX 0.5.0 (eel) is up.
NFO:openflow.of_01:[00-00-00-00-00-01 2] connected
0:00:00:00:00:02 00:00:00:00:00:04
NFO:openflow.of_01:[00-00-00-00-00-01 2] connected
0:00:00:00:00:02 00:00:00:00:00:04
```

I also tried using dump flow tables with the command: *sudo ovs-ofctl dump-tables s1* but I could not get much information out of it related to the MAC addresses, here is a screenshot:



```
^Z
[3]+  Stopped                 tail -f ../nohup.out
ubuntu@ubuntu:~/pox$ sudo ovs-ofctl dump-tables s1
[sudo] password for ubuntu:
OFPST_TABLE reply (xid=0x2):
  table 0 ("classifier"):
    active=2, lookup=362, matched=355
    max_entries=1000000
    matching:
      in_port: exact match or wildcard
      eth_src: exact match or wildcard
      eth_dst: exact match or wildcard
      eth_type: exact match or wildcard
      vlan_vid: exact match or wildcard
      vlan_pcp: exact match or wildcard
      ip_src: exact match or wildcard
      ip_dst: exact match or wildcard
      nw_proto: exact match or wildcard
      nw_tos: exact match or wildcard
      tcp_src: exact match or wildcard
      tcp_dst: exact match or wildcard

  table 1 ("table1"):
    active=0, lookup=0, matched=0
```

To demonstrate no tcp packet reaches host2 from host1 the following screenshot has been taken:

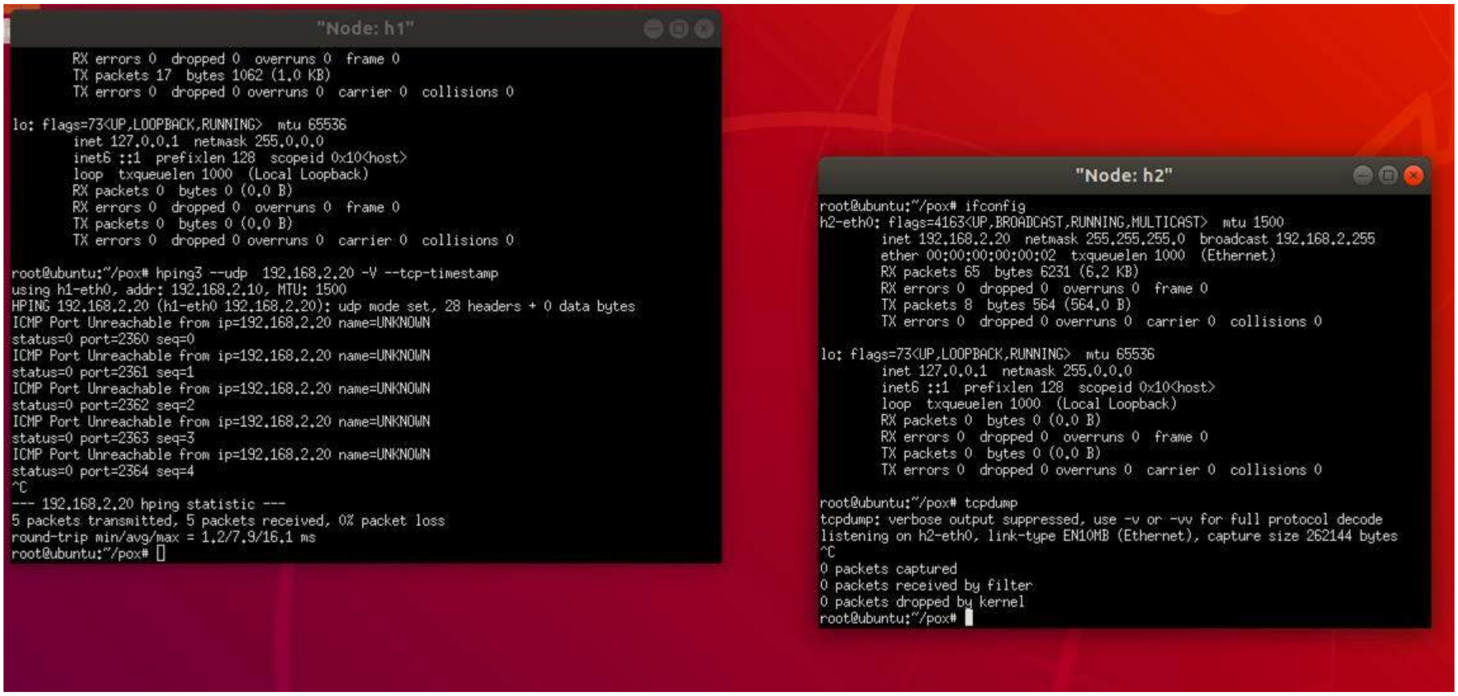


```
root@ubuntu:~# ifconfig
h1-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.10  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:01  txqueuelen 1000  (Ethernet)
        RX packets 44  bytes 4923 (4.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# hping3 -c 5 192.168.2.20 -V --tcp-timestamp
using h1-eth0, addr: 192.168.2.10, MTU: 1500
HPING 192.168.2.20 (h1-eth0 192.168.2.20): NO FLAGS are set, 40 headers + 0 data
 bytes

--- 192.168.2.20 hping statistic ---
5 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
root@ubuntu:~# 
```

```
root@ubuntu:~# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.20  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 46  bytes 5103 (5.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 396 (396.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@ubuntu:~# 
```

On host1 hping3 has been used to send host2 5 packets and none of them were sent or received, as the tcpdump run on host2 shows.

Similar to the above check-up UDP traffic between host1 and host2 has also been blocked, as the following screenshot depicts:



```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1062 (1.0 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~/pox# hping3 --udp  192.168.2.20 -V --tcp-timestamp
using h1-eth0, addr: 192.168.2.10, MTU: 1500
HPING 192.168.2.20 (h1-eth0 192.168.2.20): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.2.20 name=UNKNOWN
status=0 port=2360 seq=0
ICMP Port Unreachable from ip=192.168.2.20 name=UNKNOWN
status=0 port=2361 seq=1
ICMP Port Unreachable from ip=192.168.2.20 name=UNKNOWN
status=0 port=2362 seq=2
ICMP Port Unreachable from ip=192.168.2.20 name=UNKNOWN
status=0 port=2363 seq=3
ICMP Port Unreachable from ip=192.168.2.20 name=UNKNOWN
status=0 port=2364 seq=4
^C
--- 192.168.2.20 hping statistic ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.2/7.9/16.1 ms
root@ubuntu:~/pox# 
```

```
root@ubuntu:~/pox# ifconfig
h2-eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.20  netmask 255.255.255.0  broadcast 192.168.2.255
        ether 00:00:00:00:00:02  txqueuelen 1000  (Ethernet)
        RX packets 65  bytes 6231 (6.2 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 8  bytes 564 (564.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@ubuntu:~/pox# tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on h2-eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
root@ubuntu:~/pox# 
```

V. APPENDIX B: ATTACHED FILES

- Project Report 2 PDF
- l2firewall.config – screenshot
- l3firewall.config - screenshot

VI. REFERENCES

1. CONTAINERNET HTTPS://CONTAINERNET.GITHUB.IO/
2. MININET & POX TUTORIAL https://www.comp.nus.edu.sg/~tbma/teaching/cs4226y16_past/tutorial-Mininet-POX.pdf