Student Name: Roey Ben Hayun  
Email: [rroeybh@gmail.com](mailto:rroeybh@gmail.com), rbenhayu@asu.edu  
Submission Date: 1.11.2022  
Class Name and Term: CSE548 Summer 2021

OpenFlow Based Stateless Firewall

# Project Overview

The main objective of this project was to emulate DoS attacks on different components of the SDN infrastructure. In this project I have setup an SDN based firewall networking environment based on Mininet, Containernet, Pox controller and Open Flow Switch (OVS).

**Links for recorded video**

**YouTube**

[ASU CSE548 project 2](https://youtu.be/Ck1kdXYbKFE)

**Google Drive**

<https://drive.google.com/drive/folders/1Yq8uRjyVU-uPy2nie_0nSaMz9oTVHeLs?usp=sharing>

# Network Setup

Not required for this lab

# Software

For this project I have used the following tools

* ifconfig – to check the network configuration
* ping – check host connectivity
* nmap – to perform UDP and TCP port scan
* hping3 – to send TCP/UDP packets
* wget - non interactive network downloader

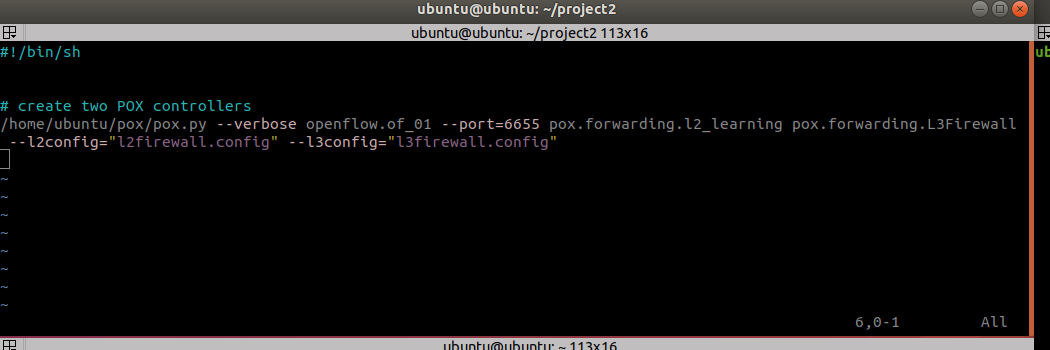
# Project Description

Creating the client and Server/Gateway VM is explain in the supporting docs.

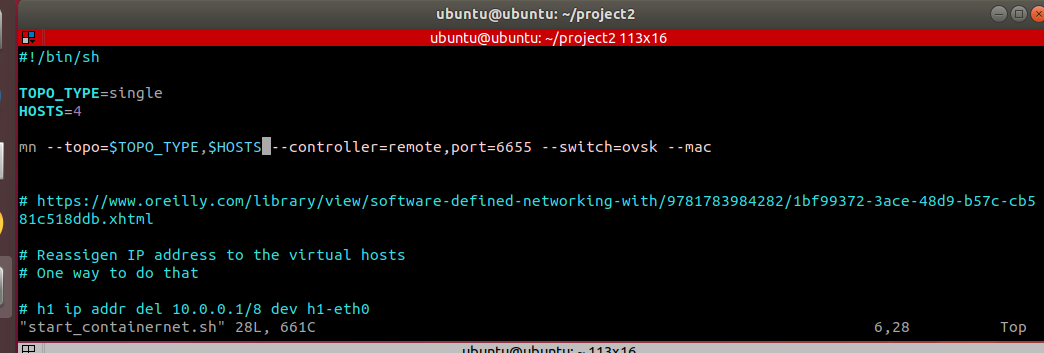
**Create Mininet topology with 4 container hosts, one controller and switch**

For ease of usage, I have created to helper scripts: one for creating pox (start\_pox.sh) and the second for setup containernet (set\_containernet.sh).

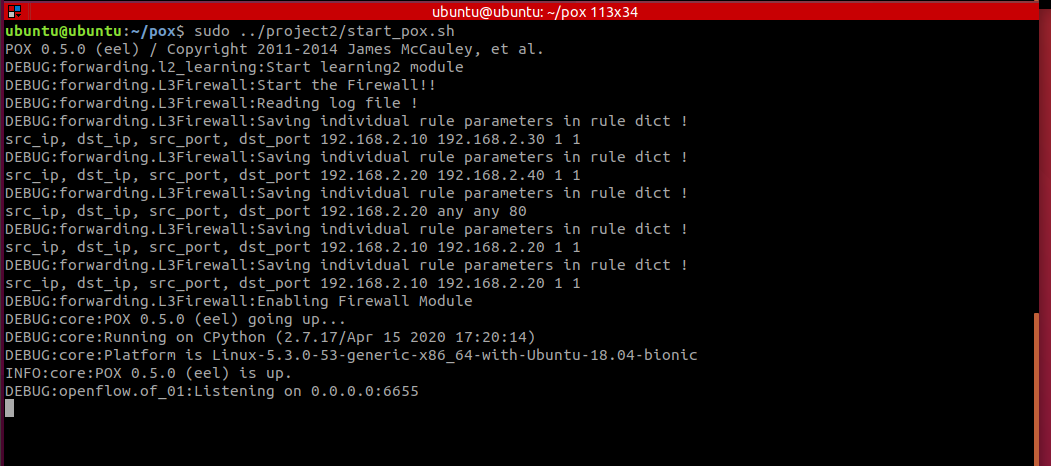
start\_pox.sh:



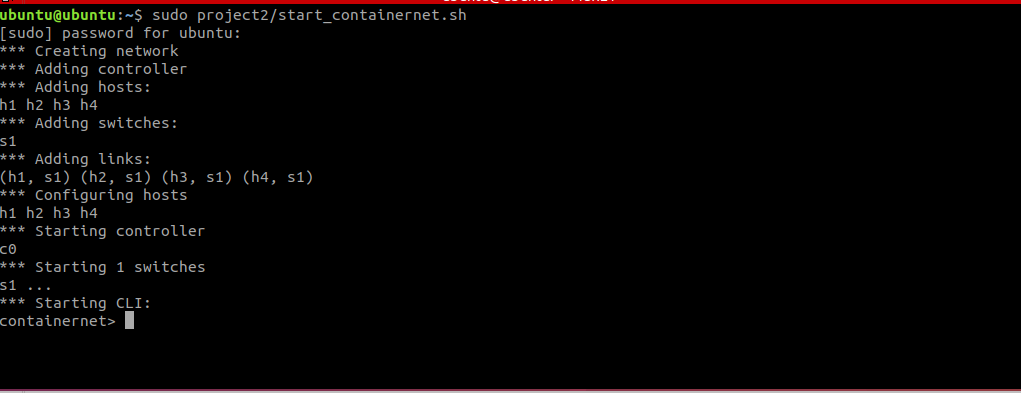
start\_containernet.sh



Let’s start POX first. You should run start\_pox.sh from the POX directory as sudo

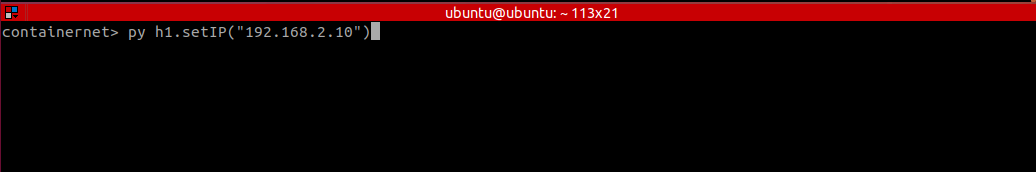


Next step is start containernet. From another terminal run start\_containernet.sh



**Make the interfaces up and assign ip address to interface of each host**

One way to set the IP address is to use python command from the containernet terminal. For example, to set the IP address of host1 you should execute the following command:



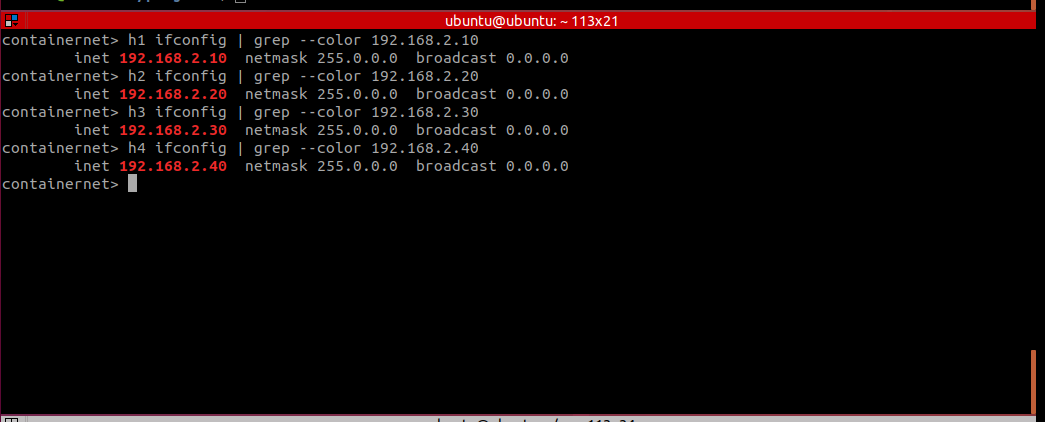
Use the same command for all other hosts:

py h2.setIP(192.168.2.20)

py h3.setIP(192.168.2.30)

py h4.setIP(192.168.2.40)

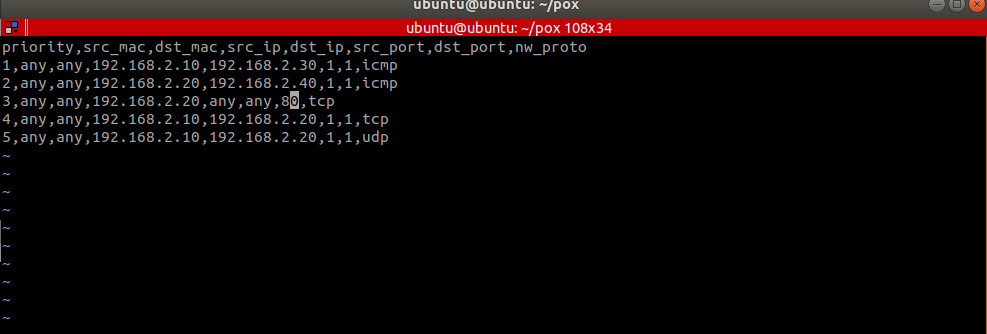
Make sure IP address has been changed. Run ifconfig and grep for the expected IP address



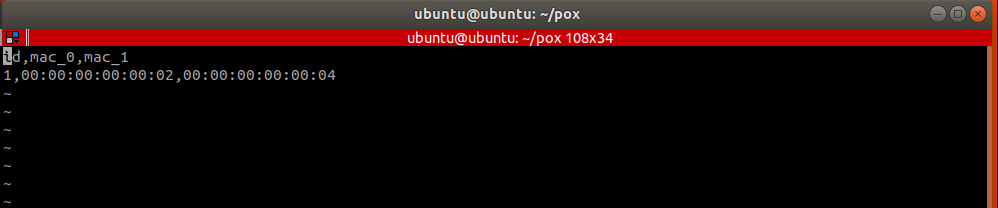
**SDN l2 and l3 firewall rules**

The add firewall rules you should update l2firewall.config and l3firewall.config under pox directory

L3firewall.config



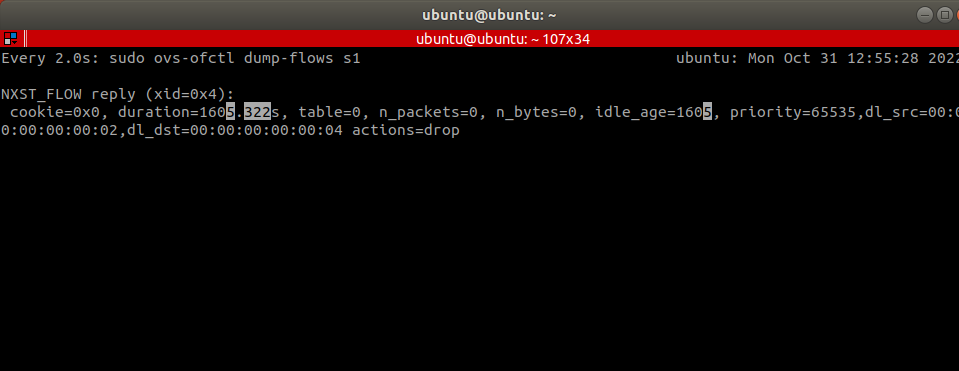
L2firewall.config



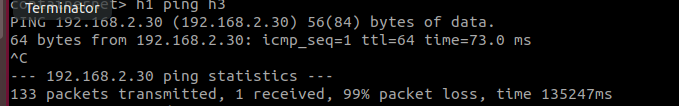
**Block ICMP traffic from h1 to h3**

From containernet ping from h1 to h3. In another window dump the flows status

dump-flows before running ping

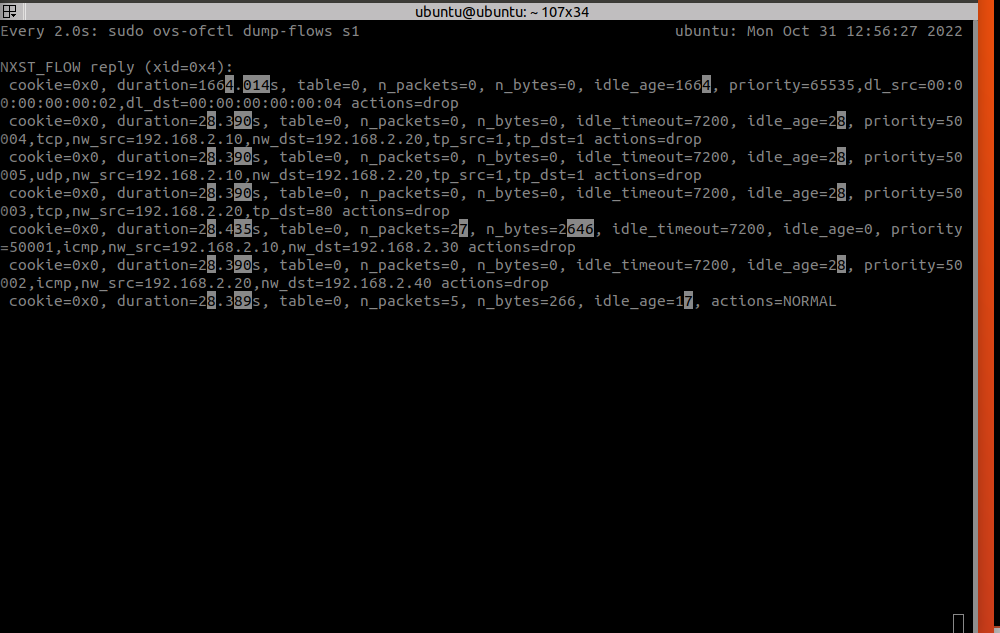


We can see here that only the first packet has accepted however all next packets are dropped. In the next try, all packet are dropped



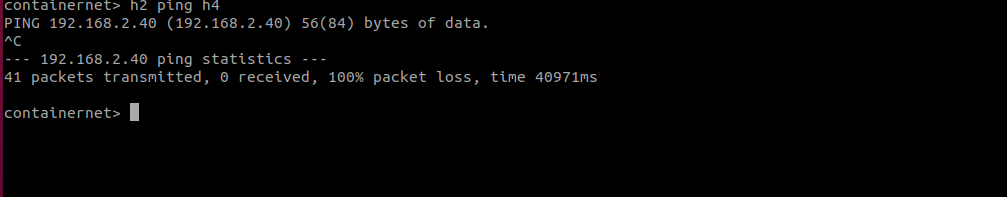
dump-flows after pinging h3 from h1. We can see here packets are dropped for the relevant rule entry

(which is the 5th entry in the screenshoot)

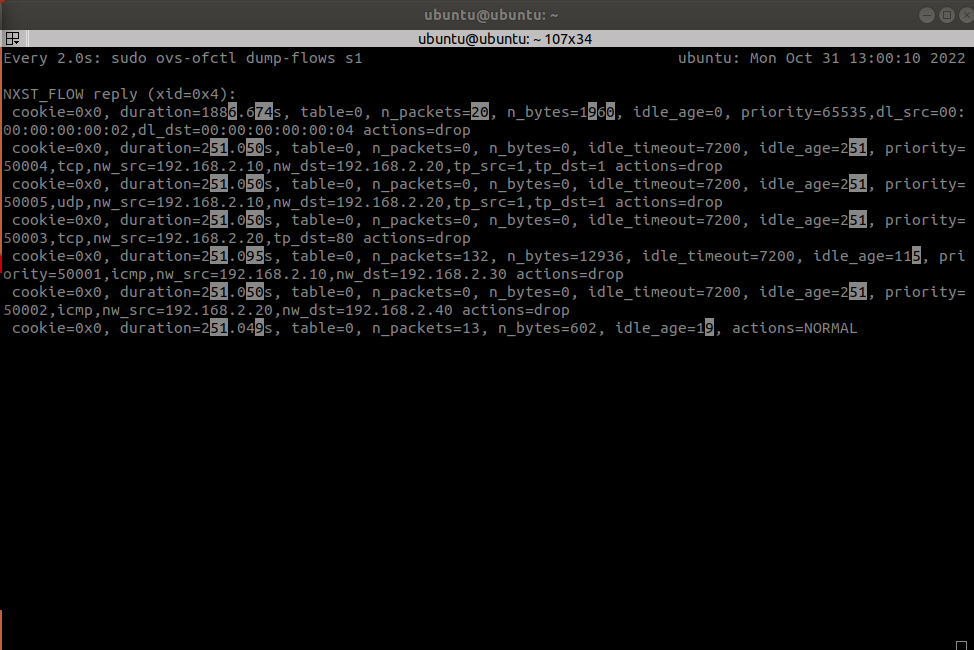


**Block ICMP traffic from h2 to h4**

Follow the same process: run ping from h2 to h4 in containernet (containernet> h2 ping h4)



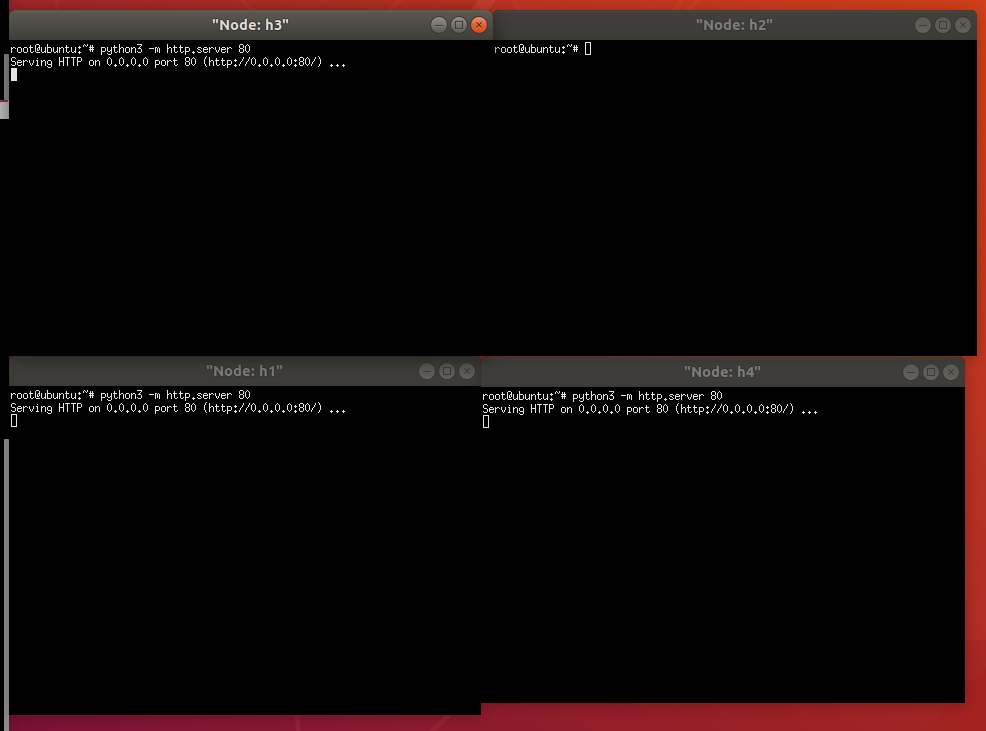
We can see l2 rule is blocking the traffic there are also some packets which were also blocked by the l3 rule. However, next tries are only blocked by the l2 rule



**Block HTTP trafic from host 2 ip (192.168.2.20)**

In order to check this rule, you need to start http server on hosts h1,h3 and h4. Easy way to do that is to use python httpserver command

First, I opened 4 terminals, one for each host. Next, I set the http server for hosts 1,3 and 4 separately

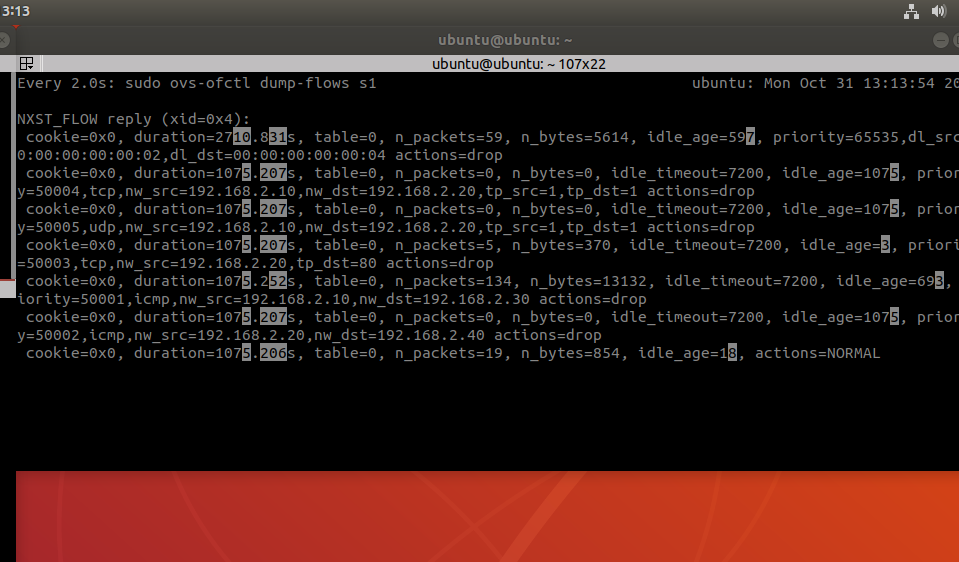


Now, call wget from h2 terminal to each of the hosts

wget 192.168.2.10:80

wget 192.168.2.30:80

wget 192.168.2.40:80



Packets are dropped for the fourth entry which is the rule in l3firewall.config

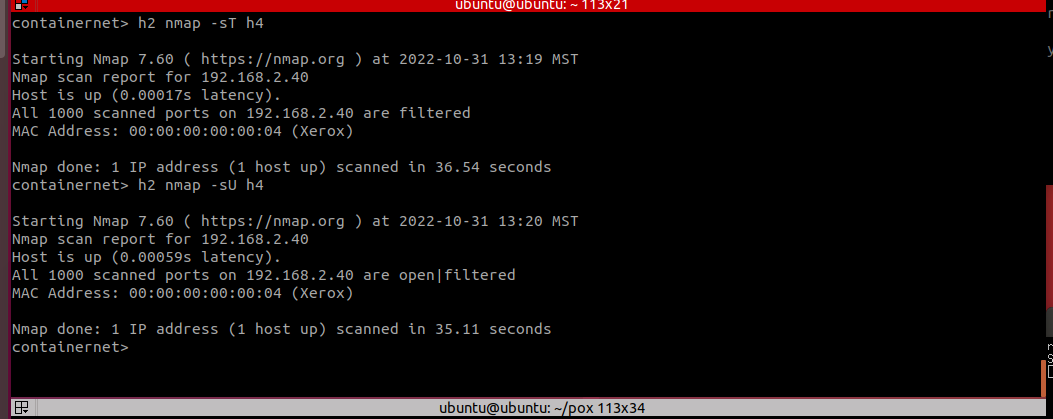
**Block traffic from MAC address 00:00:00:00:00:02 to 00:00:00:00:00:04**

This is l2firewall rule.

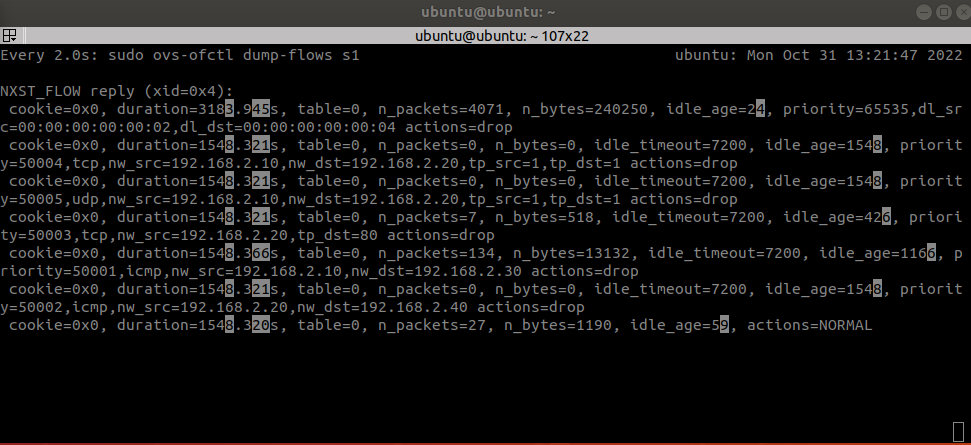
To check this rule I run nmap (fast scan to scan the 1000 most popular ports) for UDP and TCP

h2 nmap –sT h4

h2 nmap –sU h4

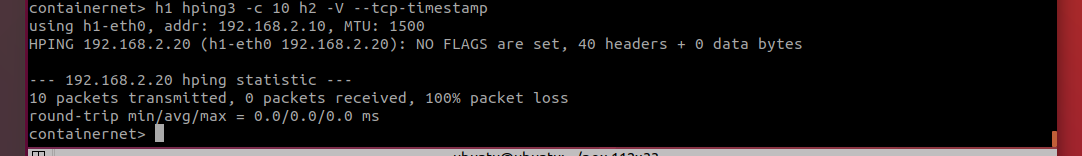


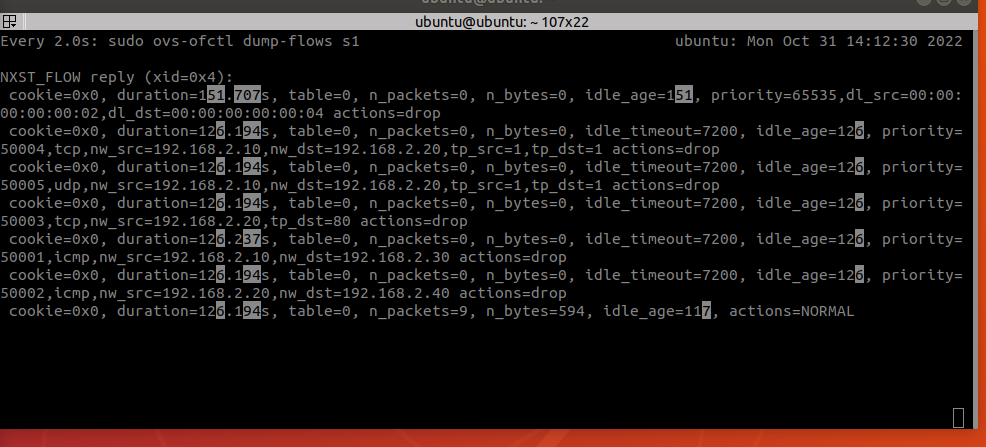
We can clearly see the big packet increase in the first entry due to nmap scan



**Block TCP traffic from h1 to h2**

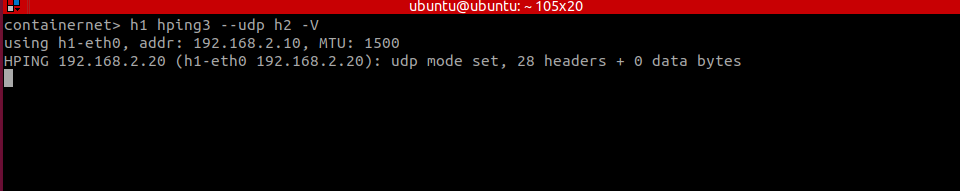
Use the following command to send UDP traffic from h1 to h2



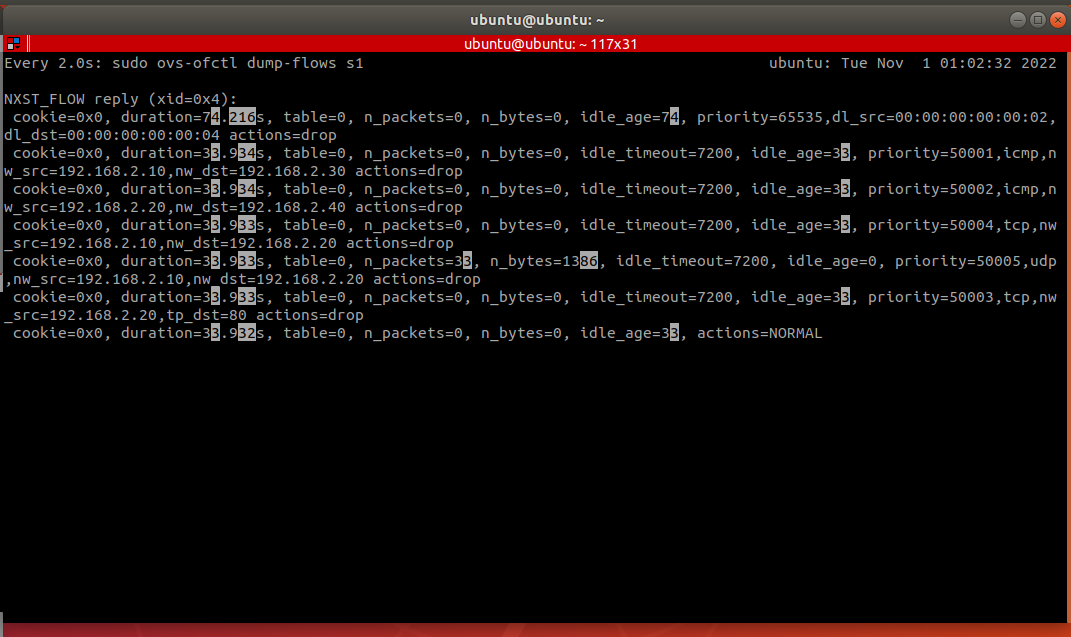


**Block UDP Traffic from h1 to h2**

Use the following command to send UDP traffic from h1 to h2



We can see the 5th rule here is blocking this traffic



# Conclusion

# Appendix B: Attached files

All project files are under project2 in the following git repo

<https://github.com/roeybenhayun/cse548-advanced_computer_network_security>

# References