Student Name: Roey Ben Hayun  
Email: [rroeybh@gmail.com](mailto:rroeybh@gmail.com), rbenhayu@asu.edu  
Submission Date: 19.11.2022  
Class Name and Term: CSE548 Summer 2021

SDN Based DoS Attacks and Mitigation

# 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**

[project3 cse548](https://youtu.be/QolGeCDGpgc)

**Google Drive**

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

# Network Setup

# Software

For this project I have used the following tools

* ifconfig – to check the network configuration
* ping – check host connectivity
* hping3 – to send initiate the DoS attack
* wget - non interactive network downloader

# Project Description

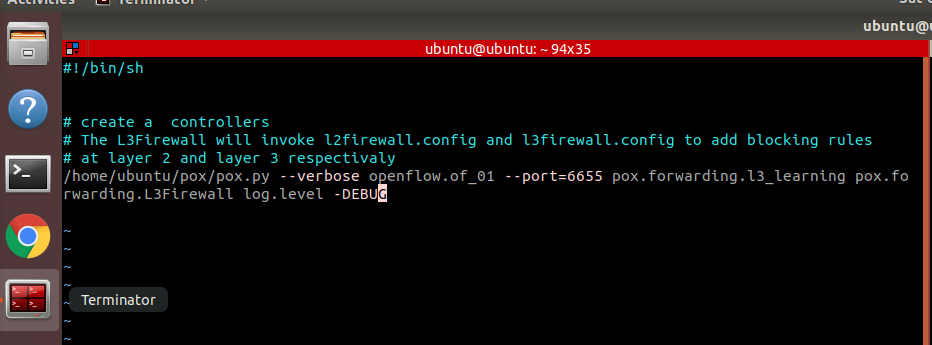
Creating the client and Server/Gateway VM is explain in the supporting docs.

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

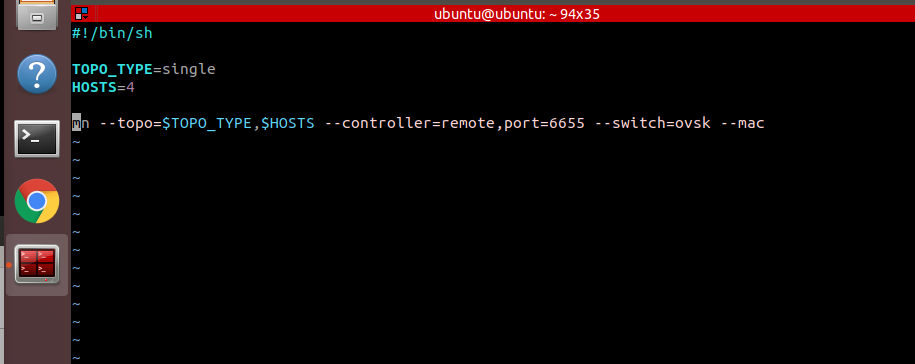
For ease of usage, I have created to helper scripts: one for creating pox (start\_pox.sh) and the second for setup containernet.

(set\_containernet.sh).

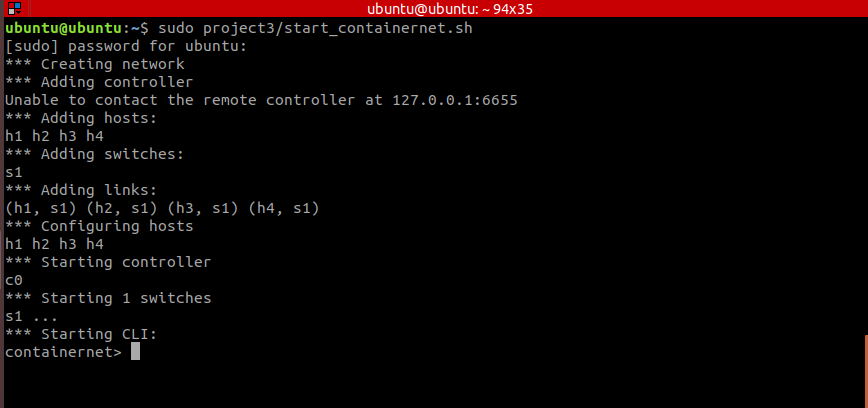
start\_pox.sh:



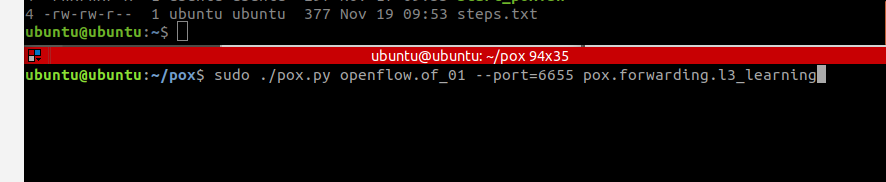
start\_containernet.sh

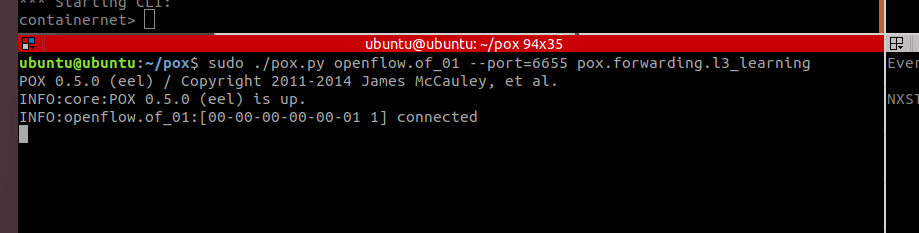


Let’s start Containernet. From the terminal run start\_containernet.sh



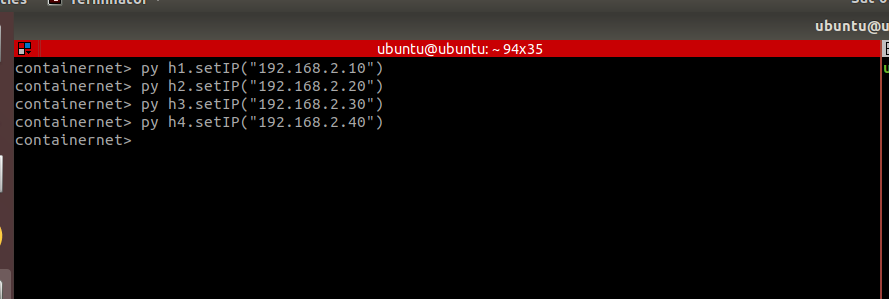
Now, let’s start POX . You should run start\_pox.sh from the POX directory as sudo





**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:



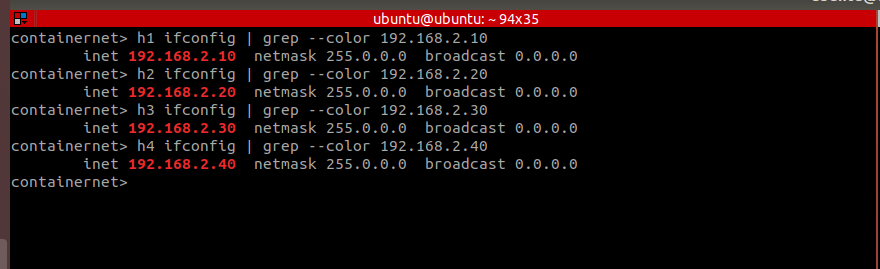
Use the same command for all other hosts:

py h2.setIP(192.168.2.20)

py h3.setIP(192.168.2.30)

py h4.setIP(192.168.2.40)

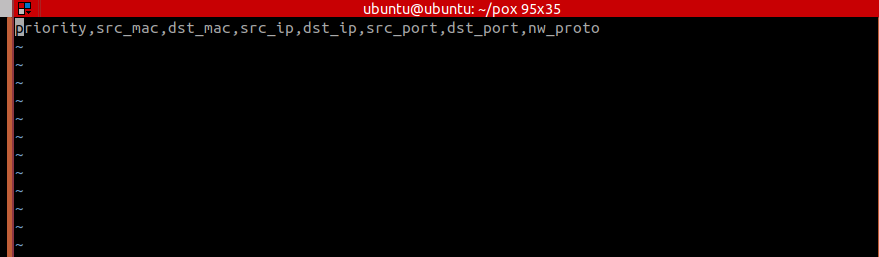
Make sure IP address has been changed. Run ifconfig and grep for the expected IP address



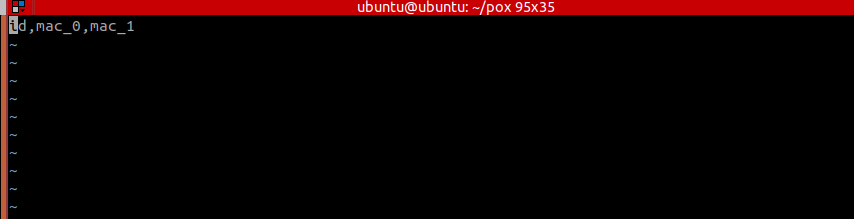
**SDN l2 and l3 firewall rules**

For this project, the DoS attack mitigation was done by editing L3Firewall.py. Both L2 and

L3 config files should be empty (without any rules)



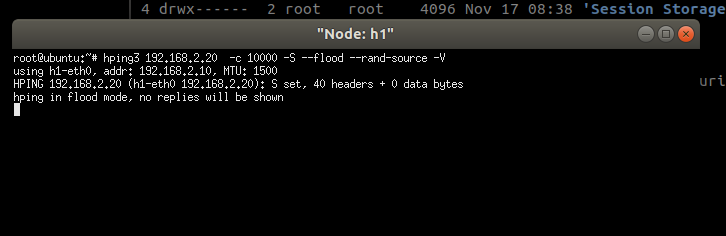
L2firewall.config



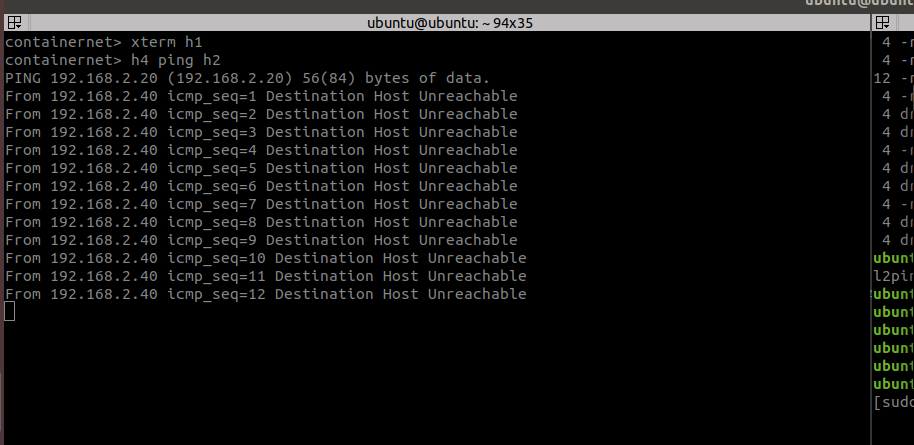
**Perform a DoS flooding attack**

Run POX with the l3\_learning (following the same process as was explained in the lab)

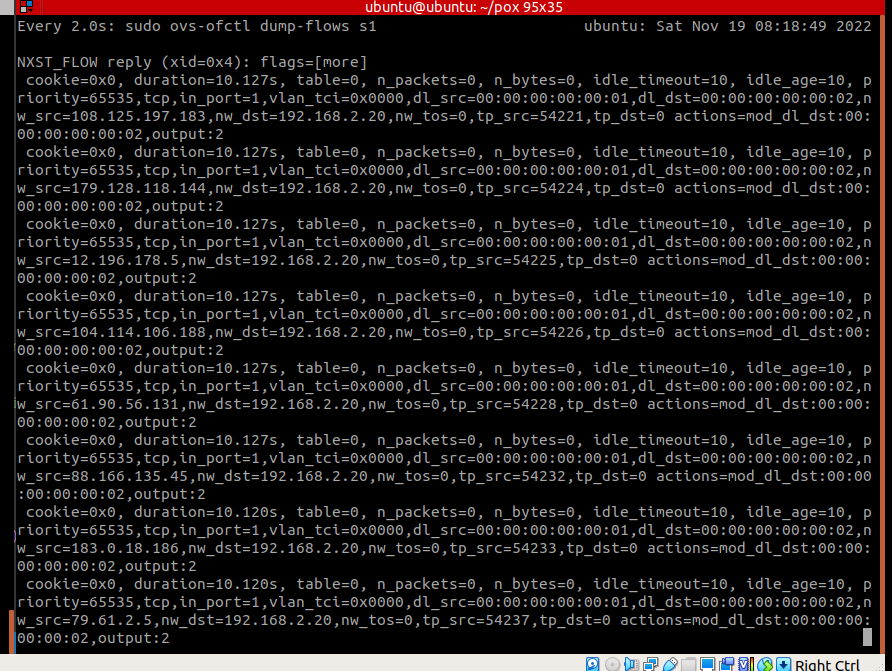
From Containernet open h1 terminal using xterm h1 and initiate the flood attack to h2



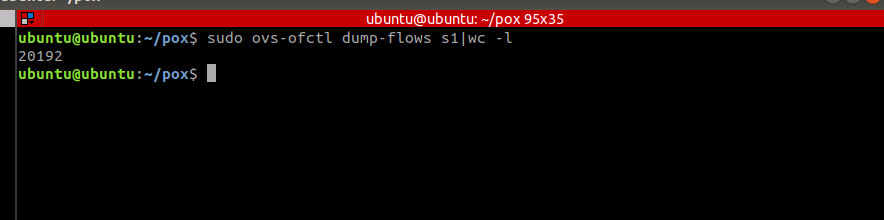
Try ping from h4 to h2 (host is unreachable)



Let’s examine the dump flows (obviously we can see that traffic is not blocked)



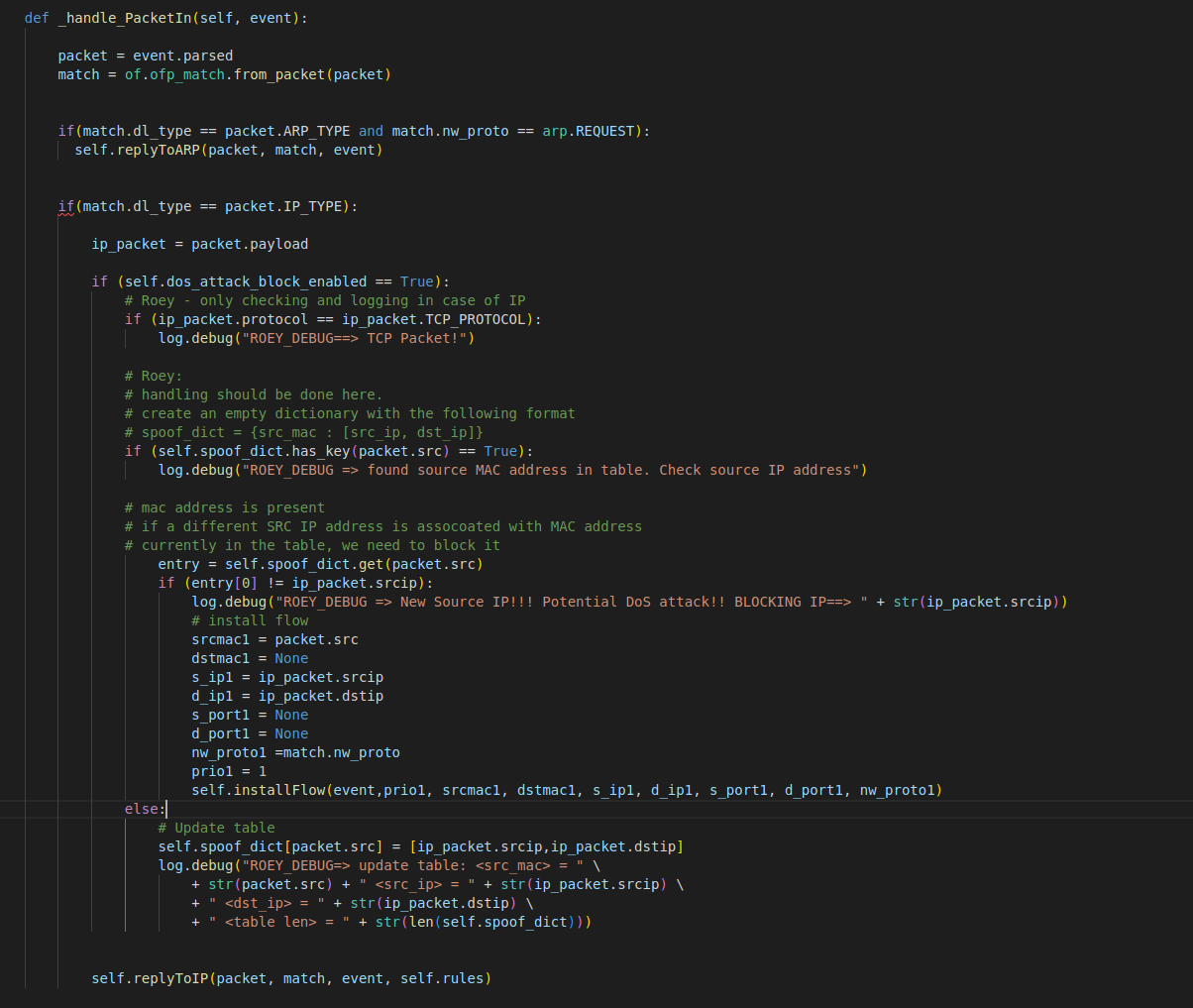
Number or rules



**Mitigate DoS attack**

The simplest way to mitigate the DoS attack was to modify the L3Firewall.py script

Source code snippets which handles DoS attack

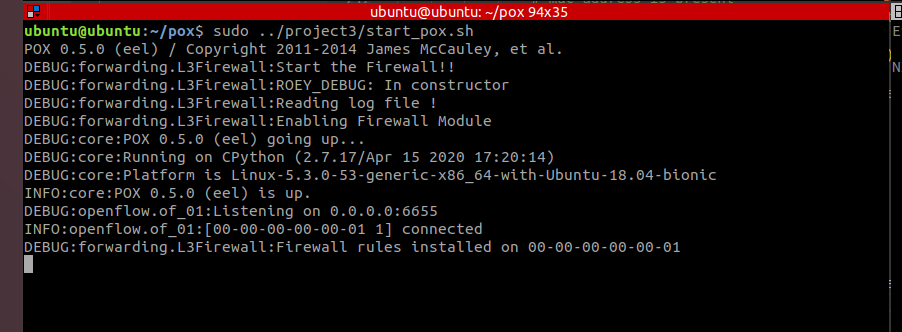


I defined a dictionary to store the mapping between the source MAC address, destination and source IP address

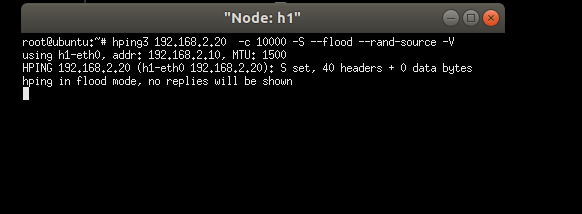
When a new packet is received, I check if there is an existing mapping in the dictionary. If true I check if it is same source IP address. If not, there is a high chance of DoS attack. In this case I am installing a new flow to **block** this traffic.

I have followed the same DoS attack procedure to validate the solution

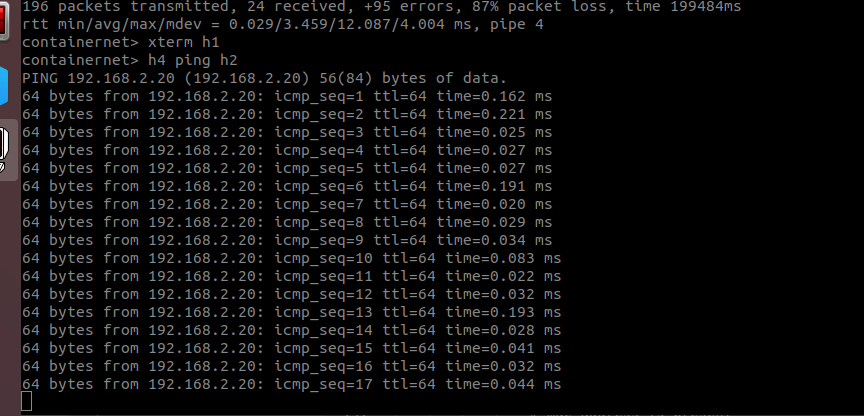
Start Pox with the L3Firewall application



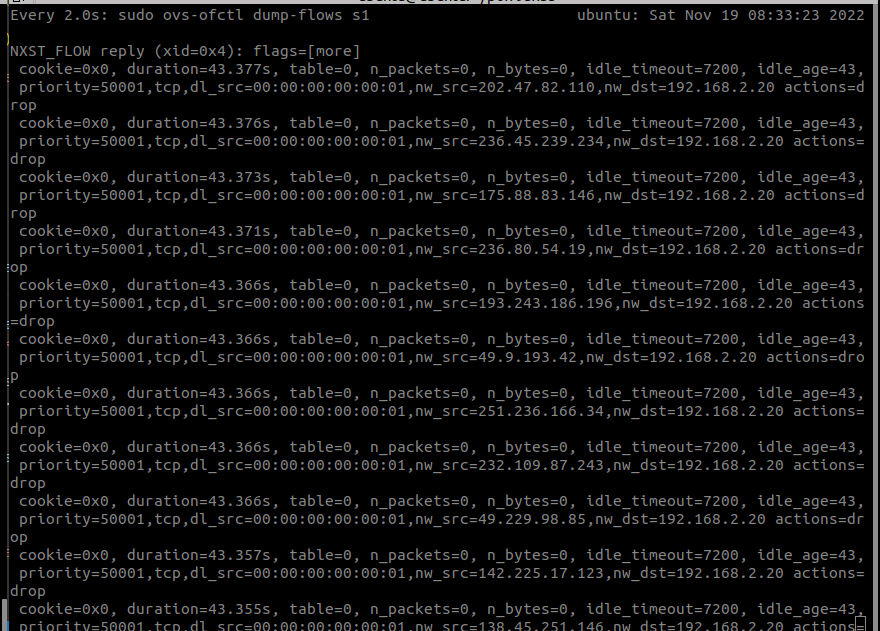
Run DoS attack from h1



Ping from h4 to h2 – ok. No h2 is reacable

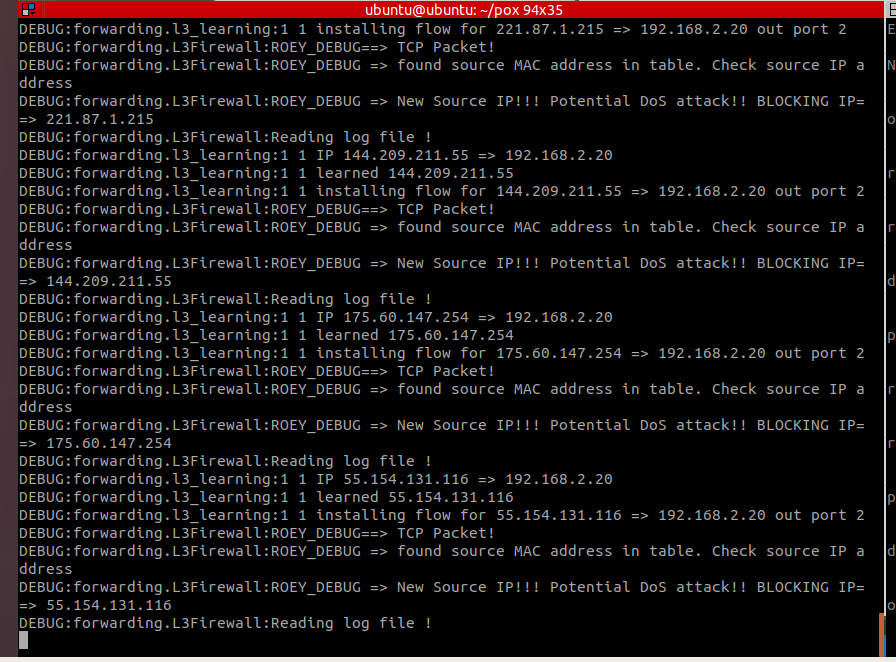


Check flow rules – ok. We are dropping the suspicions packets



A partial output logged from L3Firewall.py. We can see in the logs the detection of the Dos

Attack (“Potential DoS Attack” log)



We can conclude the mitigation work as expected

# Conclusion

We learned how DoS attacks can be mitigated by write the port security function in L3Firewall.py directly

# 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