CSCS 4720
Securing Self-driving Networks
Lab Assignment 2
Due Thursday October 6

1. (100 pts) For this assignment, you need to use Mininet in Kali Linux on Virtualbox. Use the following to install Mininet on Kali Linux.

```
sudo apt-get update
sudo apt-get -y install mininet
sudo apt-get -y install openvswitch-testcontroller
service openvswitch-switch start
```

To run mininet type the following

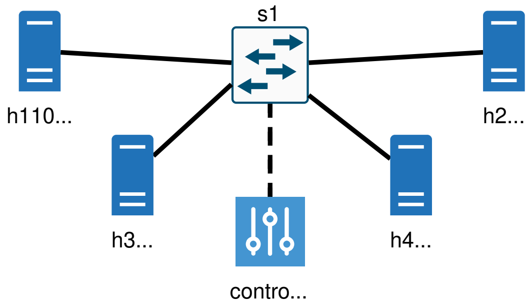
```
sudo mn
```

This will run mininet with OVS controller.

For Assignment 2 we will use POX controller. To install Pox controller do the following

```
git clone http://github.com/noxrepo/pox
cd pox
git checkout dart
```

Use the mininet topology topo2.py, to setup your network which assumes a remote controller listening on the default IP address and port number 127.0.0.1:6633. You do not need to modify the topology file. The topology that this file will setup is as follows. Note that h1 and h4 are on the same subnet and a different one from h2 and h3.



Use the provided skeleton POX controller file controller.py. This file will be where you will make your modifications to create the firewall. Launch the controller with the skeleton file controller.py

To start POX controller copy *controller.py* to *pox/pox/misc* directory and type the following in pox directory

```
./pox.py misc.controller
```

Once the controller is running, you can run mininet with

```
sudo python3 topo2.py.
```

Switch s1 will implement the following rules.

src ip	*dst ip*	*protocol*	*action*
any ipv4	any ipv4	icmp	accept
any	any	arp	accept
any	ipv4	any ipv4	drop

Basically, your Firewall should allow all ARP and ICMP traffic to pass. However, any other type of traffic should be dropped. It as acceptable to flood the allowable traffic out all ports. Be careful! Flow tables match the rule with highest priority first, where priority is established based on the order rules are placed in the table. When you create a rule in the POX controller, you need to also have POX "install" the rule in the switch. This makes it so the switch "remembers" what to do for a few seconds. Do not handle each packet individually inside of the controller! Hint: To do this, look up *ofp_flow_mod*. The OpenFlow tutorial (specifically " Sending OpenFlow messages with POX") and the POX Wiki are both useful resources for understanding how to use POX.

Submit the following for this assignment.

- A screenshot of the pingall command. Note that h1 and h4 should be able to ping each other (h2 and h3 as well), but not across subnets. Also, the iperf h1 h4 command should fail (as you're blocking IP traffic). This is realized as the command hanging.
- A screenshot of the output of the dpctl dump-flows command. This should contain all of the rules you've inserted into your switch(es).
- Your controller.py file.

Following links can be useful

- *OpenFlow Tutorial:* `https://github.com/mininet/openflow-tutorial/wiki/Create-a-Learning-Switch#Controller_Choice_POX_Python`
- *POX Wiki:* `https://noxrepo.github.io/pox-doc/html/`

Follow the steps below to share a folder between host OS and Kali Linux VM.

(a) Power off Kali Linux VM

(b) Click settings for Kali Linux VM

(c) Click Shared Folders on the left

(d) Click + to add a shared folder

(e) Select your folder and check auto mount

(f) Start Kali Linux VM

(g) Shared folder will be under /media with prefix sf