Final Project Lab Report

First, let's start with <u>TCP/ipv4.</u> As you can see, h1, h2, h3, and h5 can communicate just fine due to the rules installed in the flow table.

```
mininet@mininet-vm:~/lab/CE150/lab4$ sudo python final_skel.py
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['19.9 Gbits/sec', '19.9 Gbits/sec']
mininet> iperf h5 h3
*** Iperf: testing TCP bandwidth between h5 and h3
*** Results: ['30.9 Gbits/sec', '30.9 Gbits/sec']
mininet> iperf h4 h3
*** Iperf: testing TCP bandwidth between h4 and h3
```

However, what happens when h4, an outsider, tries to connect? The connection completely freezes up, and you have to tell it to give up. Why? I designed the switches 'learn' that h4 is not a part of trusted hosted by ofpmod, as seen by dpctl dump-flows here: (Notice the actions=drop)

```
mininet> iperf h3 h4

*** Iperf: testing TCP bandwidth between h3 and h4

C
Interrupt
mininet> dpctl dump-flows

*** s1

*** S2

*** S2

*** S3

*** S4

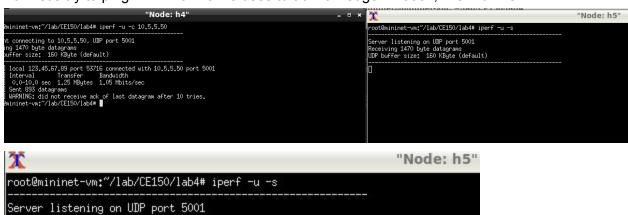
*** S4

*** S5

*** S
```

Similar behavior for udp, another ipv4 protocol. h3 -> h5 udp connection pinging by xterm is sucessful below.

Now let's try to ping h4 -> h5. H5 Refuses to acknowledge h4 at all, like with TCP!



Now the attacker tries to use pingall. What happens?

```
*** Ping: testing ping reachability
h1 -> h2 h3 h4 h5
h2 -> h1 h3 h4 h5
h3 -> h1 h2 h4 h5
h4 -> X X X
h5 -> h1 h2 h3 X
```

As you can see, h4 can still attempt to send out <u>ARP packets</u>. The moment that h4 even attempts to try <u>ICMP packets</u> through ping, it is blocked out from the network. Check out all the actions= 'drop' rules installed.

```
*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1

*** $1
```

Overall, I learned a lot about how packet handling works in openflow. You absolutely have to broadcast rules to the appropriate switches when forwarding ports. If I were to expand upon this assignment, I would try to see how to handle a DDoS attack or Trojan attack, as ARP leaves a hole open in our network.