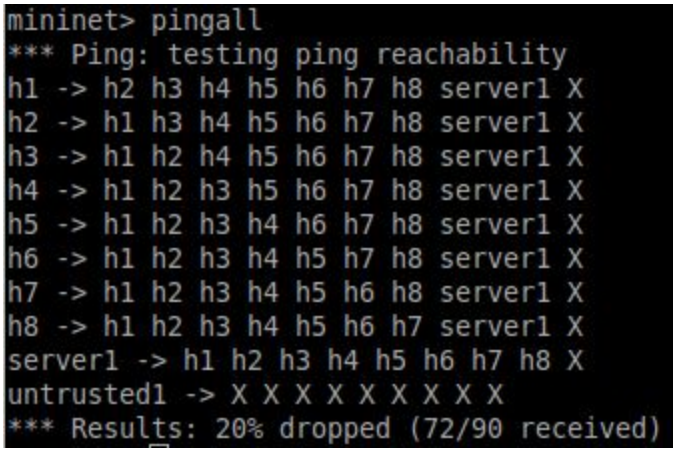
Kyle Zhang
kmzhang
kmzhang@ucsc.edu
1669388
Final Project

Few comments:

1. I changed the IP address of the untrusted host to a different one that I saw on piazza.
2. Not sure why the topology would have errors sometimes, but restarting the VM seemed to fix it.

Outputs:

1. As you can see from the output of the pingall command, all pings from the company hosts and server to the untrusted host as well as pings from the untrusted host to the hosts and server were caught and not allowed by the filter. I accomplished this by checking the source and destination IPs in the controller, and taking action based on what they were. Based on the switch id, source and destination IPs would be checked and the action output port number would be set based on these conditions.



2. As you can see from the iperf outputs, both iperf between two company hosts and iperf between one company host and the untrusted host are able to be completed, but the iperf between the untrusted host and the server is unable to be completed. This shows that all ICMP traffic between the untrusted host and company hosts is blocked, but other traffic like TCP traffic can go through. However, all traffic is blocked between the untrusted host and server since both the pings and iperf commands were unable to be completed between the untrusted host and server. I implemented the logic between the company hosts and the untrusted host in the controller by only allowing action to be taken if the type was not ICMP or the source IP was not the untrusted host. I implemented the logic

between the untrusted host and server by adding a statement checking that the source IP and destination IPs were not the untrusted host and server respectively.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['37.1 Gbits/sec', '37.2 Gbits/sec']
mininet> iperf h1 untrusted1
*** Iperf: testing TCP bandwidth between h1 and untrusted1
*** Results: ['34.1 Gbits/sec', '34.2 Gbits/sec']
mininet> iperf untrusted1 server1
*** Iperf: testing TCP bandwidth between untrusted1 and server1
```

3. As you can see from the dump-flows output, “actions” shows a number and not just FLOOD. This value specifies the destination port numbers of the IP traffic. To get this, instead of setting the port to of.OFPP.FLOOD, I set the port to the specific port numbers that I had set inside my topology. This dump-flows is only based on the iperf from host 1 to host 2 because for some reason after interrupting the iperf command between the untrusted host and server, dump-flows would not show any output.

```
mininet> dpctl dump-flows
*** s1 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
 cookie=0x0, duration=20.086s, table=0, n_packets=660421, n_bytes=28009872778, i
dle_timeout=30, hard_timeout=30, idle_age=15, tcp,vlan_tci=0x0000,dl_src=00:00:0
0:00:00:01,dl_dst=00:00:00:00:00:02,nw_src=10.0.1.10,nw_dst=10.0.1.20,nw_tos=0,t
p_src=54713,tp_dst=5001 actions=output:2
 cookie=0x0, duration=20.098s, table=0, n_packets=4, n_bytes=272, idle_timeout=3
0, hard_timeout=30, idle_age=20, tcp,vlan_tci=0x0000,dl_src=00:00:00:00:00:01,dl
_dst=00:00:00:00:00:02,nw_src=10.0.1.10,nw_dst=10.0.1.20,nw_tos=16,tp_src=54712,
tp_dst=5001 actions=output:2
 cookie=0x0, duration=20.096s, table=0, n_packets=3, n_bytes=206, idle_timeout=3
0, hard_timeout=30, idle_age=20, tcp,vlan_tci=0x0000,dl_src=00:00:00:00:00:02,dl
_dst=00:00:00:00:00:01,nw_src=10.0.1.20,nw_dst=10.0.1.10,nw_tos=0,tp_src=5001,tp
_dst=54712 actions=output:1
 cookie=0x0, duration=20.083s, table=0, n_packets=534403, n_bytes=35279722, idle
_timeout=30, hard_timeout=30, idle_age=15, tcp,vlan_tci=0x0000,dl_src=00:00:00:0
0:00:02,dl_dst=00:00:00:00:00:01,nw_src=10.0.1.20,nw_dst=10.0.1.10,nw_tos=0,tp_s
rc=5001,tp_dst=54713 actions=output:1
*** s2 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s3 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s4 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s5 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
*** s6 ------------------------------------------------------------------------
NXST_FLOW reply (xid=0x4):
mininet>
```