

Melanie Wong

Final Project

CMPE 150/L

6/7/19

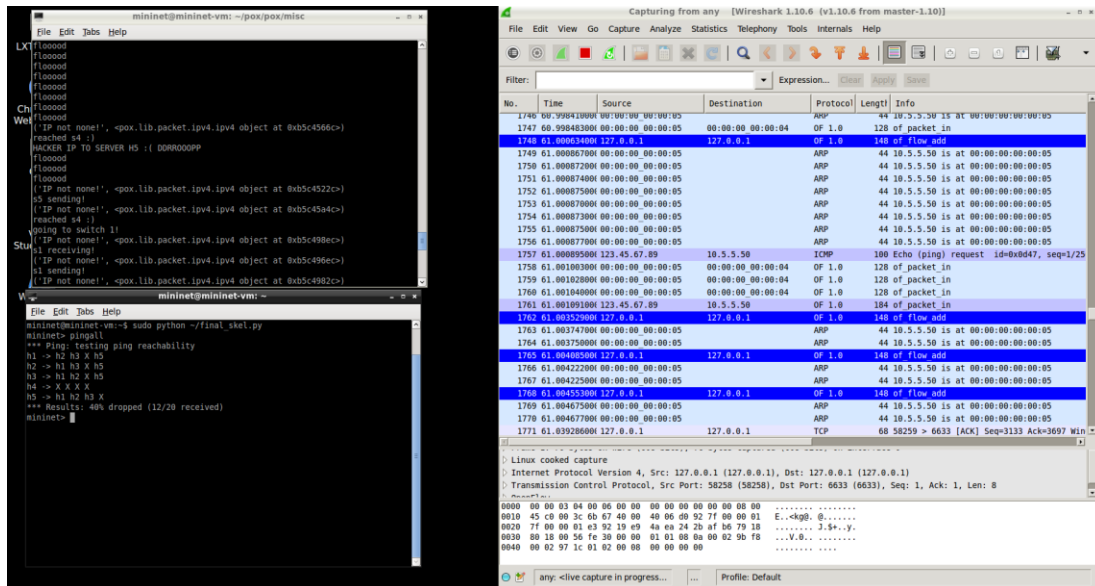
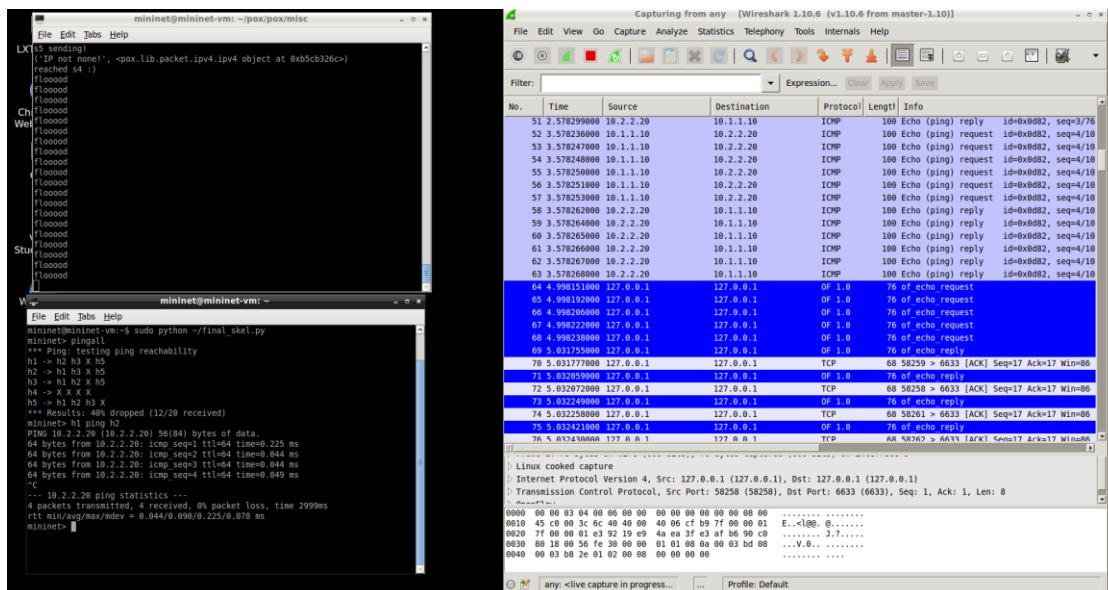


Image #1 above uses pingall to involve transfers/pings between all host pairs. Because pings involve ICMP packets, the ones from h4 “hacker” are dropped to any other host and pings fail, as shown in wireshark where there is a request but no reply as well as the print statements coded in the controller. The image #2 below shows successful ping between trusted hosts h1 and h2, which involve requests and replies.



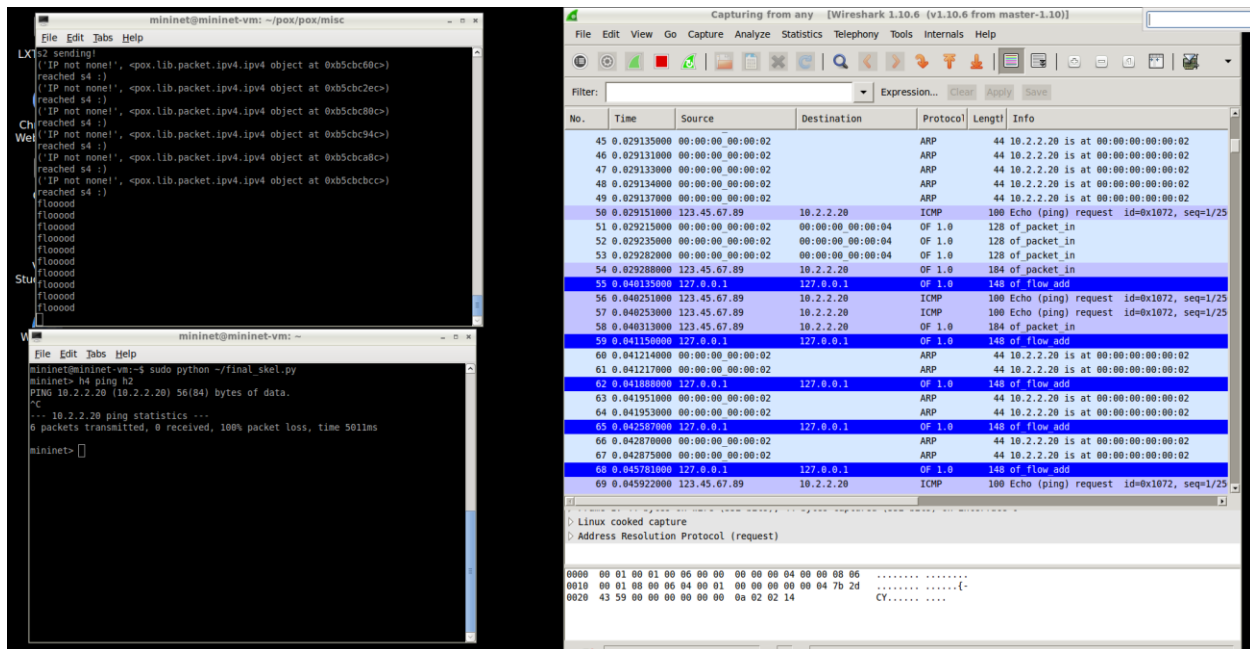


Image #3 above shows ping involving the untrusted host h4, in which there are ICMP echo requests but no replies because the packets are identified and dropped at switch 4.

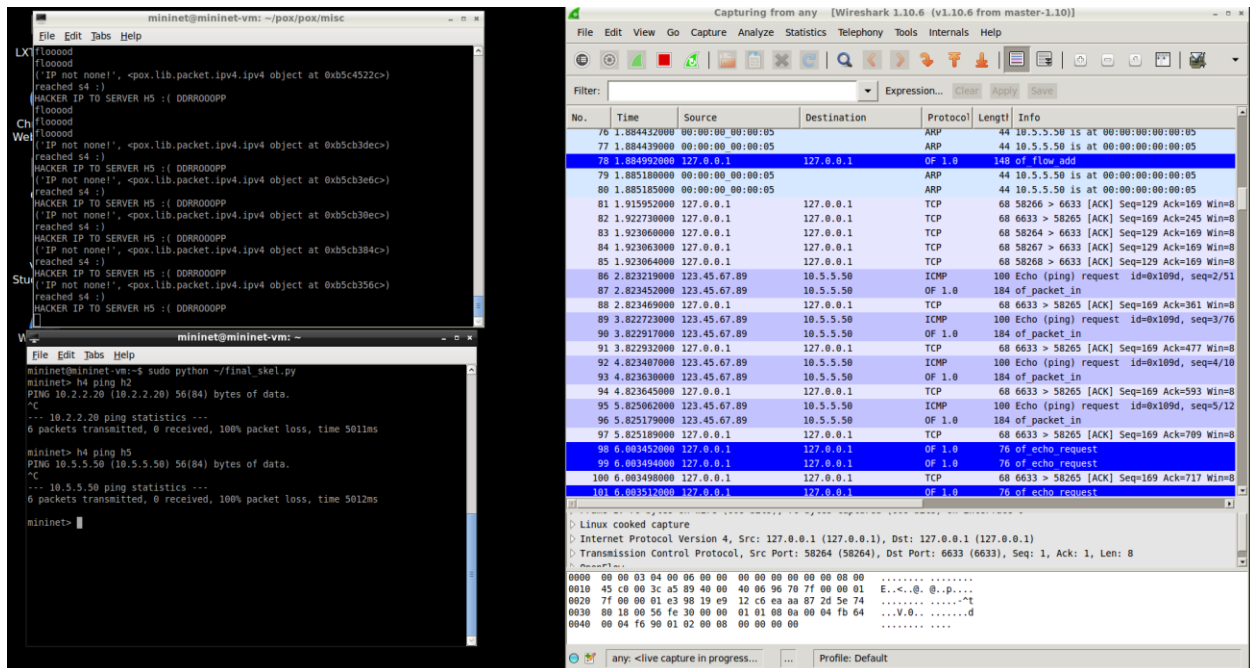
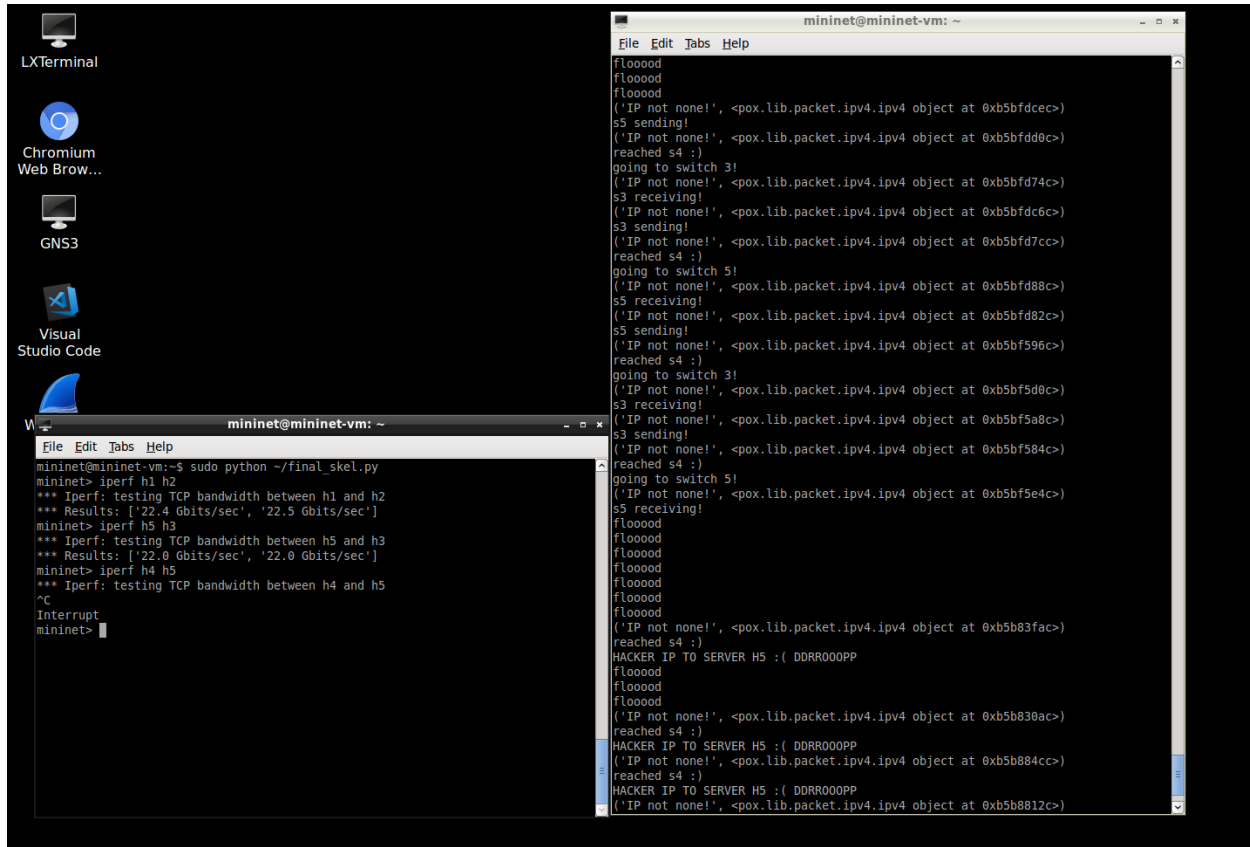


Image #4 above shows a similar situation with ping between untrusted host h4 and server h5, where the ICMP echo requests are also identified and dropped which is also shown in the coded

print statements in the controller. Some of these print statements occur and alternate after the ‘IP not none’, which means IP packets are dropped and don’t reach server h5 as well.



```
mininet@mininet-vm: ~  
File Edit Tabs Help  
mininet@mininet-vm:~$ sudo python ~/final_skel.py  
mininet> iperf h1 h2  
*** Iperf: testing TCP bandwidth between h1 and h2  
*** Results: ['22.4 Gbits/sec', '22.5 Gbits/sec']  
mininet> iperf h5 h3  
*** Iperf: testing TCP bandwidth between h5 and h3  
*** Results: ['22.0 Gbits/sec', '22.0 Gbits/sec']  
mininet> iperf h4 h5  
*** Iperf: testing TCP bandwidth between h4 and h5  
^C  
Interrupt  
mininet>
```

```
mininet@mininet-vm: ~  
File Edit Tabs Help  
floodood  
floodood  
floodood  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfdcec>)  
s5 sending!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd80c>)  
reached s4 :)  
going to switch 3!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd74c>)  
s3 receiving!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd6c>)  
s3 sending!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd7cc>)  
reached s4 :)  
going to switch 5!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd88c>)  
s5 receiving!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bfd82c>)  
s5 sending!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bf596c>)  
reached s4 :)  
going to switch 3!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bf5d0c>)  
s3 receiving!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bf5a8c>)  
s3 sending!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bf584c>)  
reached s4 :)  
going to switch 5!  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5bf5e4c>)  
s5 receiving!  
floodood  
floodood  
floodood  
floodood  
floodood  
floodood  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5b83fac>)  
reached s4 :)  
HACKER IP TO SERVER H5 :( DDRR000PP  
floodood  
floodood  
floodood  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5b830ac>)  
reached s4 :)  
HACKER IP TO SERVER H5 :( DDRR000PP  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5b884cc>)  
reached s4 :)  
HACKER IP TO SERVER H5 :( DDRR000PP  
(('IP not none!', <pox.lib.packet.ipv4.ipv4 object at 0xb5b8812c>)
```

Image #5 above shows successful transmissions between trusted hosts h1 h2 h3 and h5 with the first 2 iperf commands, as well as the print statements that are generated when packets are sent from one host and received by another. The last iperf command doesn’t automatically resolve and requires interruption because the IP packets never reach server h5 from the untrusted host h4 after they are dropped by the controller.