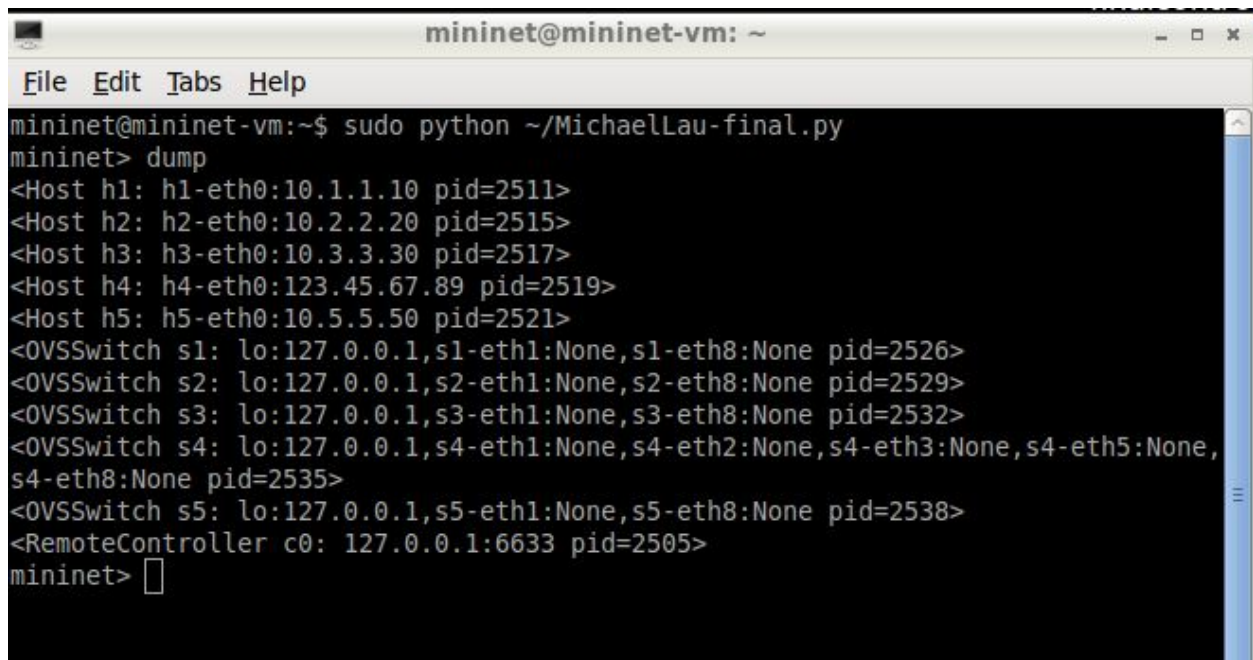
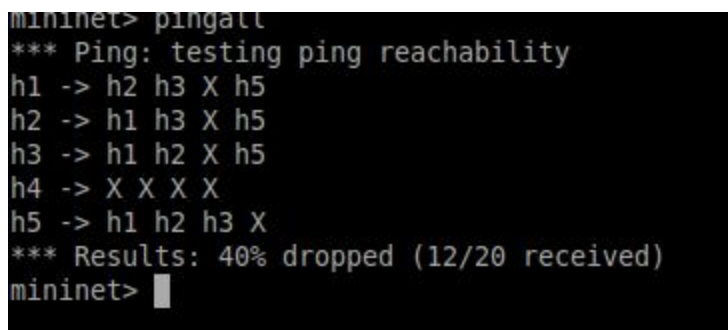


Michael Lau
mlau10@ucsc.edu
1493231
3/13/18
Final Project Lab



```
mininet@mininet-vm: ~  
File Edit Tabs Help  
mininet@mininet-vm:~$ sudo python ~/MichaelLau-final.py  
mininet> dump  
<Host h1: h1-eth0:10.1.1.10 pid=2511>  
<Host h2: h2-eth0:10.2.2.20 pid=2515>  
<Host h3: h3-eth0:10.3.3.30 pid=2517>  
<Host h4: h4-eth0:123.45.67.89 pid=2519>  
<Host h5: h5-eth0:10.5.5.50 pid=2521>  
<OVSSwitch s1: lo:127.0.0.1,s1-eth1:None,s1-eth8:None pid=2526>  
<OVSSwitch s2: lo:127.0.0.1,s2-eth1:None,s2-eth8:None pid=2529>  
<OVSSwitch s3: lo:127.0.0.1,s3-eth1:None,s3-eth8:None pid=2532>  
<OVSSwitch s4: lo:127.0.0.1,s4-eth1:None,s4-eth2:None,s4-eth3:None,s4-eth5:None,  
s4-eth8:None pid=2535>  
<OVSSwitch s5: lo:127.0.0.1,s5-eth1:None,s5-eth8:None pid=2538>  
<RemoteController c0: 127.0.0.1:6633 pid=2505>  
mininet> 
```

This is a screenshot of the dump command which shows all the hosts and switches are set accordingly.



```
mininet> pingall  
*** Ping: testing ping reachability  
h1 -> h2 h3 X h5  
h2 -> h1 h3 X h5  
h3 -> h1 h2 X h5  
h4 -> X X X X  
h5 -> h1 h2 h3 X  
*** Results: 40% dropped (12/20 received)  
mininet> 
```

This is the pingall command which utilizes ICMP and shows that h4, the intruder, can't send ICMP to anybody in the network. However, this doesn't show if the other hosts can send h4 ICMP.

99	0.858485000	10.2.2.20	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bdc, seq=1/256, ttl=64 (re
100	0.869996000	10.5.5.50	10.3.3.30	ICMP	98 Echo (ping) request	id=0x0bdd, seq=1/256, ttl=64 (req
101	0.870015000	10.3.3.30	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bdd, seq=1/256, ttl=64 (re
102	0.856973000	10.5.5.50	10.2.2.20	ICMP	98 Echo (ping) request	id=0x0bdc, seq=1/256, ttl=64 (req
103	0.856990000	10.2.2.20	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bdc, seq=1/256, ttl=64 (re
104	0.882464000	10.5.5.50	123.45.67.89	ICMP	98 Echo (ping) request	id=0x0bde, seq=1/256, ttl=64 (req
105	0.882481000	123.45.67.89	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bde, seq=1/256, ttl=64 (re
106	0.868177000	10.5.5.50	10.3.3.30	ICMP	98 Echo (ping) request	id=0x0bdd, seq=1/256, ttl=64 (req
107	0.872463000	10.3.3.30	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bdd, seq=1/256, ttl=64 (re
108	0.868171000	10.5.5.50	10.3.3.30	ICMP	98 Echo (ping) request	id=0x0bde, seq=1/256, ttl=64 (req
109	0.872469000	10.3.3.30	10.5.5.50	ICMP	98 Echo (ping) reply	id=0x0bdd, seq=1/256, ttl=64 (re

Therefore, I have to use Wireshark to show that the other hosts, like h5, the server, is sending ICMP to the other hosts including h4 and shows that h4 attempts to reply that it received the ICMP packet but because ICMP is blocked from h4, the reply is never received by h5 and therefore in pingall shows it isn't reachable. This is the case whenever any host tries to ping h4.

```
mininet> iperf h1 h2
*** Iperf: testing TCP bandwidth between h1 and h2
*** Results: ['54.8 Gbits/sec', '54.8 Gbits/sec']
mininet> iperf h1 h4
*** Iperf: testing TCP bandwidth between h1 and h4
*** Results: ['42.3 Gbits/sec', '42.3 Gbits/sec']
mininet> iperf h4 h1
*** Iperf: testing TCP bandwidth between h4 and h1
*** Results: ['32.7 Gbits/sec', '32.7 Gbits/sec']
mininet> iperf h4 h5
*** Iperf: testing TCP bandwidth between h4 and h5
^C
Interrupt
mininet> iperf h5 h1
*** Iperf: testing TCP bandwidth between h5 and h1
*** Results: ['44.1 Gbits/sec', '44.1 Gbits/sec']
mininet> █
```

I used iperf command to show that everyone can communicate except h4 can't send traffic to h5. I was unable to set a timeout for the iperf between h4 and h5 therefore it idles for a long time and I have to interrupt it manually.

```

mininet> h4 ping h5 -c 5
PING 10.5.5.50 (10.5.5.50) 56(84) bytes of data.

--- 10.5.5.50 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 3999ms

mininet> █

```

To ensure and prove that h4 can't send traffic to h5 only, I pinged h4 to h5 for 5 packets and verified 100% drop.

33	5.810705000	123.45.67.89	10.5.5.50	TCP	74 [TCP Retransmission] 57580 > complex-link [SYN] Seq=6
34	6.807188000	123.45.67.89	10.5.5.50	OF 1.0	158 [TCP Retransmission] of packet in
35	6.807205000	127.0.0.1	127.0.0.1	TCP	66 6633 > 40008 [ACK] Seq=17 Ack=201 Win=88 Len=0 TSval=3
36	6.807028000	123.45.67.89	10.5.5.50	TCP	74 [TCP Retransmission] 57580 > complex-link [SYN] Seq=6
37	8.811269000	123.45.67.89	10.5.5.50	OF 1.0	158 [TCP Retransmission] of packet in
38	8.811281000	127.0.0.1	127.0.0.1	TCP	66 6633 > 40008 [ACK] Seq=17 Ack=293 Win=88 Len=0 TSval=3
39	8.811166000	123.45.67.89	10.5.5.50	TCP	74 [TCP Retransmission] 57580 > complex-link [SYN] Seq=6

In addition, I used wireshark to verify during iperf h4 h5 that traffic isn't going through. This is indeed the case as iperf repeatedly tries to retransmit until I manually interrupt iperf.

72	2.163514000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 123.45.67.89? Tell 10.1.1.10
73	2.167132000	00:00:00_00:00:04	00:00:00_00:00:01	ARP	42 123.45.67.89 is at 00:00:00:00:00:04
74	2.171503000	10.1.1.10	123.45.67.89	ICMP	98 Echo (ping) request id=0x08ae, seq=1/256, ttl=64
75	2.132423000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 10.2.2.20? Tell 10.1.1.10
76	2.138029000	00:00:00_00:00:02	00:00:00_00:00:01	ARP	42 10.2.2.20 is at 00:00:00:00:00:02
77	2.150521000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 10.3.3.30? Tell 10.1.1.10
78	2.156533000	00:00:00_00:00:03	00:00:00_00:00:01	ARP	42 10.3.3.30 is at 00:00:00:00:00:03
79	2.165035000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 123.45.67.89? Tell 10.1.1.10
80	2.167130000	00:00:00_00:00:04	00:00:00_00:00:01	ARP	42 123.45.67.89 is at 00:00:00:00:00:04
81	2.132426000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 10.2.2.20? Tell 10.1.1.10
82	2.138033000	00:00:00_00:00:02	00:00:00_00:00:01	ARP	42 10.2.2.20 is at 00:00:00:00:00:02
83	2.150524000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 10.3.3.30? Tell 10.1.1.10
84	2.156549000	00:00:00_00:00:03	00:00:00_00:00:01	ARP	42 10.3.3.30 is at 00:00:00:00:00:03
85	2.165038000	00:00:00_00:00:01	Broadcast	ARP	42 Who has 123.45.67.89? Tell 10.1.1.10

This screenshots of wireshark when I pingall shows other traffic that isn't IP is flooded. This is the case because ARP is not IP.