Results of Firewall Implementation

I will discuss the problem and results of the solution implemented with the RYU controller to manage the network traffic. In this set up there are four networks attached to a singular switch constituting the patient, doctor, records and public access networks.

In this scenario the doctor network may exchange packets with the patient network and patient records however the patient and records networks may not communicate, and no one may communicate with the public access network.

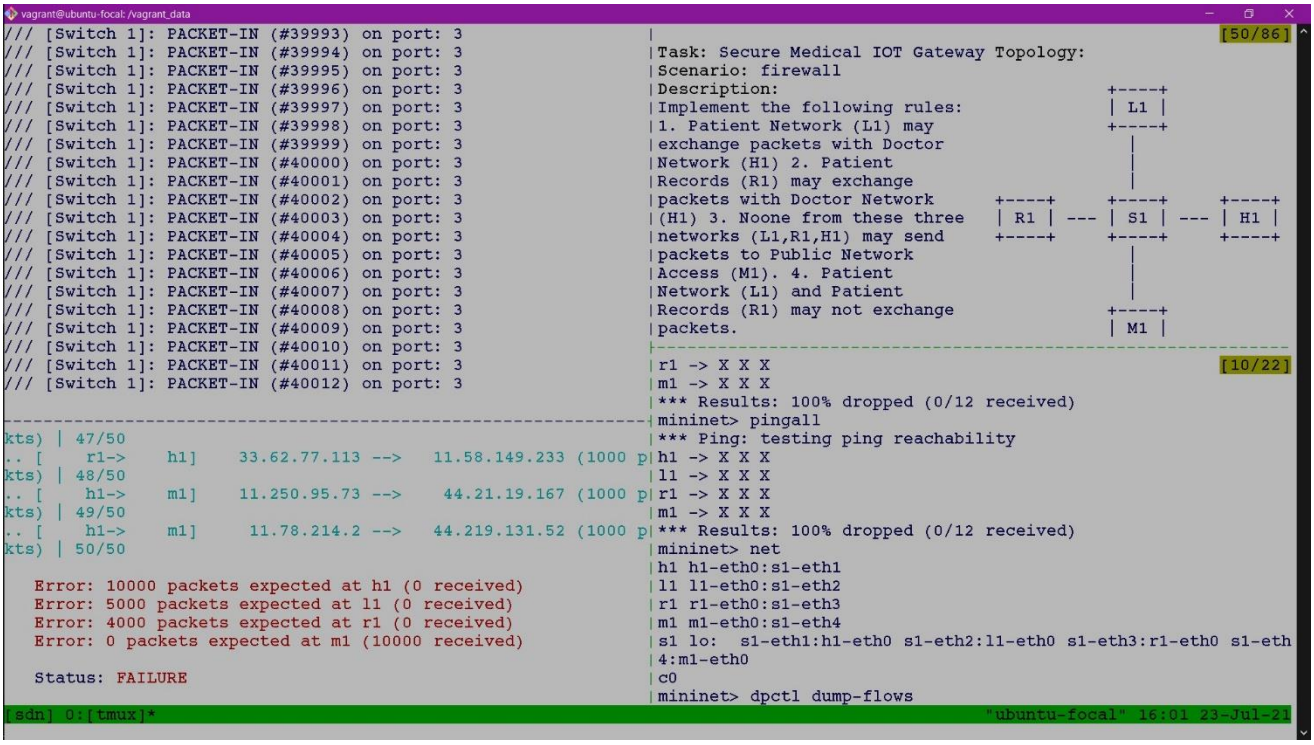

Figure 1. A demonstration of the base application with all packets sent to the public access network.

The test involves sending traffic and counting the packets received at its destination as shown the base application fails to pass the test as all traffic is sent to m1. The controller has no other flows established to handle incoming traffic as such traffic is not effectively routed.

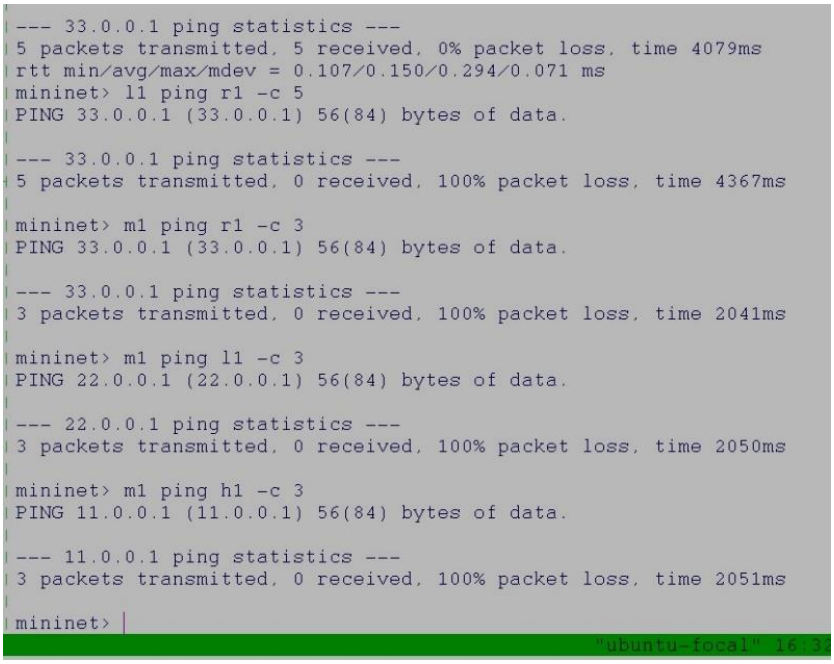

Figure 2. The RYU controller logic unable to effectively direct ICMP traffic.

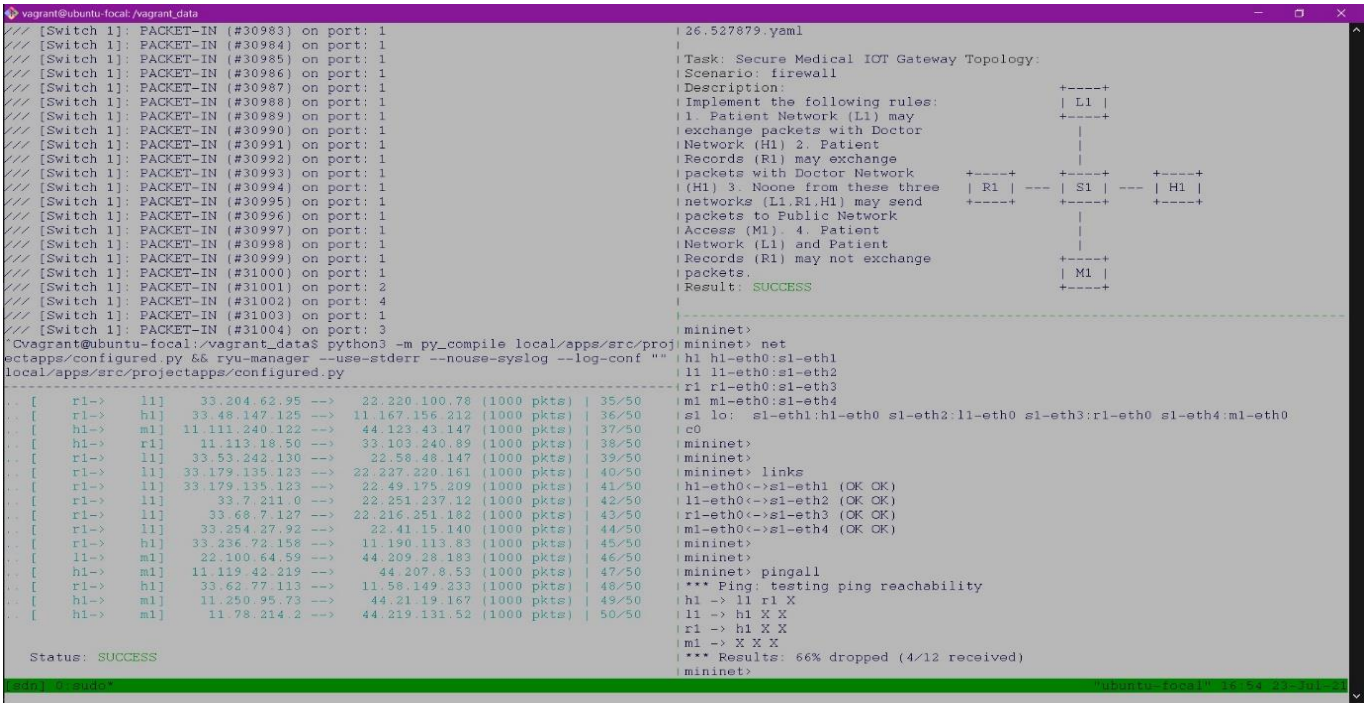

Figure 3. The solution implemented and the results when the traffic scenario is instantiated.

When the firewall.py application is run in the RYU controller It successfully passes the scenario that the unaltered application did not.

This was achieved by creating flow roles based upon incoming packet ipv4_src and ipv4_dst.

Below in figure 4. The pingall command demonstrates a reachability that matches the plan document and the flow roles which have been instigated.

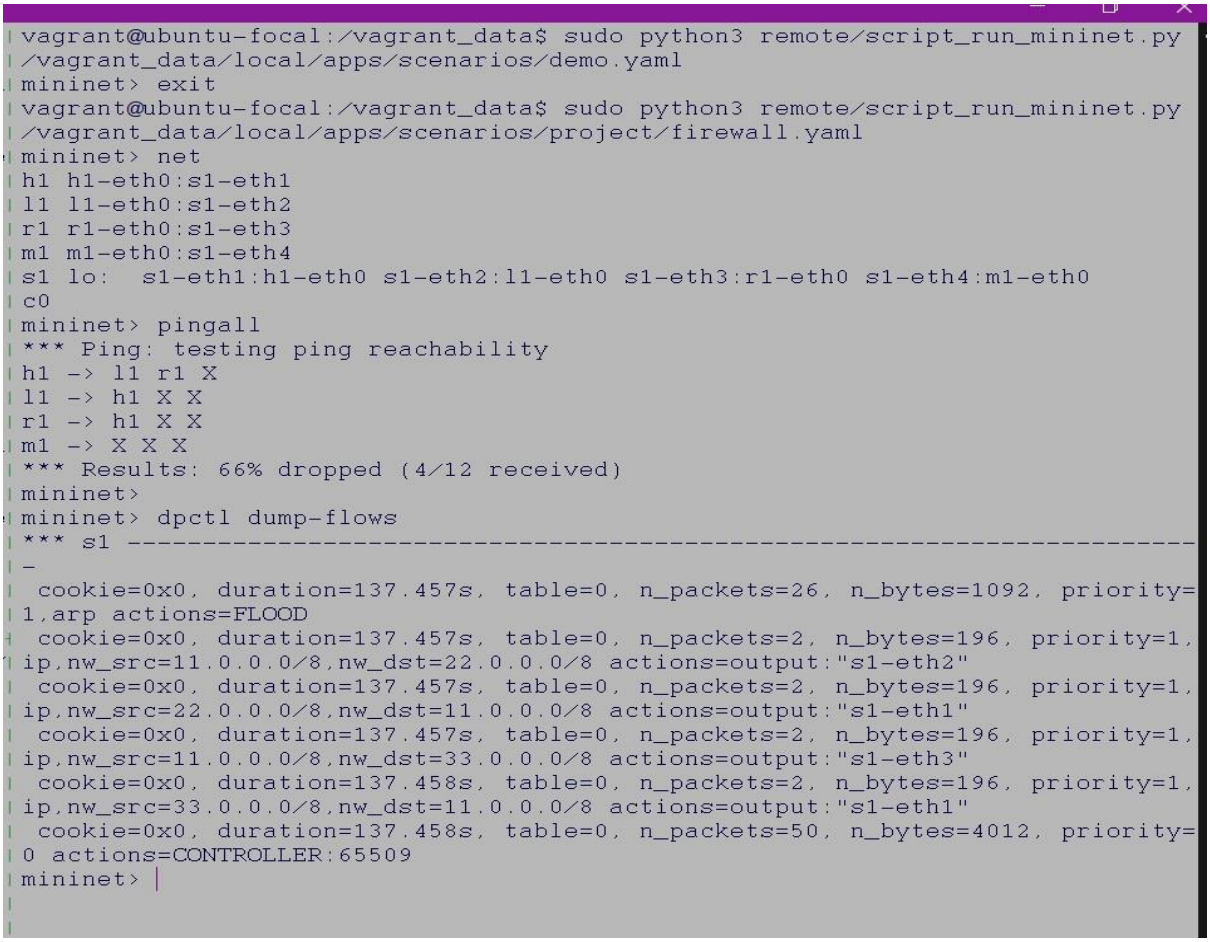

Figure 4. Demonstration of switch connectivity and the flows implemented.

```
--- 22.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4100ms
rtt min/avg/max/mdev = 0.064/0.148/0.287/0.077 ms
mininet> h1 ping l1 -c 5
PING 22.0.0.1 (22.0.0.1) 56(84) bytes of data.
64 bytes from 22.0.0.1: icmp_seq=1 ttl=64 time=0.209 ms
64 bytes from 22.0.0.1: icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from 22.0.0.1: icmp_seq=3 ttl=64 time=0.058 ms
64 bytes from 22.0.0.1: icmp_seq=4 ttl=64 time=0.105 ms
64 bytes from 22.0.0.1: icmp_seq=5 ttl=64 time=0.103 ms

--- 22.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4085ms
rtt min/avg/max/mdev = 0.036/0.102/0.209/0.059 ms
mininet> h1 ping r1 -c 5
PING 33.0.0.1 (33.0.0.1) 56(84) bytes of data.
64 bytes from 33.0.0.1: icmp_seq=1 ttl=64 time=0.294 ms
64 bytes from 33.0.0.1: icmp_seq=2 ttl=64 time=0.118 ms
64 bytes from 33.0.0.1: icmp_seq=3 ttl=64 time=0.116 ms
64 bytes from 33.0.0.1: icmp_seq=4 ttl=64 time=0.118 ms
64 bytes from 33.0.0.1: icmp_seq=5 ttl=64 time=0.107 ms

--- 33.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.107/0.150/0.294/0.071 ms
```

Figure 5. Depicts reachability between the doctor–patient and doctor–records networks

ICMP test determines reachability by sending echo request packets and await the echoed response from the target host. Various statistics can be derived from this such as packet loss percentage, time from message sent to response receival including the fastest, slowest, packet average and standard deviation from the mean.