CAN201 In-Class Test Report

DoS attacks against the Data Plane of an SDN Network

Student Name: Ruiyang.Wu

Student ID: 2257475

Q1: What is the initial flow entry (before attacking) installed in the switch of this lab?

The initial flow entry is shown below:

```
cookie=0x0, duration=30.395s, table=0, n_packets=16, n_bytes=1272,
priority=0 actions=CONTROLLER:65535
```

Q2: Explain what this initial flow entry does.

The initial flow entry has the lowest priority (priority=0), so that it can match any packets which do not match other higher priority rules. In this case, as it is the only rule exist initially in the flow table, it will send any packets to the OpenFlow controller so that the controller will install new flow entries to the switch or decide how to handle them based on the packets received.

Q3: Show the flow 'match' rule (in the lab11.py) used to cope with the above flooding traffic.

In this lab, we use the following hping3 command flooding the flow table:

```
h1 hping3 h2 -c 10000 --udp --flood --rand-source
```

According to the parameter "--udp", the programme will only flood udp packets, so it will only match the match rule of udp protocol:

```
#  If UDP Protocol
elif protocol == in_proto.IPPROTO_UDP:
    u = pkt.get_protocol(udp.udp)
    match = parser.OFPMatch(eth_type=ether_types.ETH_TYPE_IP,
in_port=in_port, ipv4_src=srcip, ipv4_dst=dstip, ip_proto=protocol,
udp_src=u.src_port, udp_dst=u.dst_port,)
```

Q4: Explain how the flooding command exhausts the switch's flow table?

According to the parameter "—rand-source", the programme will create udp packets which source is random. It means that the switch has to install a new flow entry for every packet. However, the number of packets is much higher than the size of the flow table, once the flow table is full (exhaust), the switch cannot accept new flows, leading to packet drops.

Q5: Revise the flow 'match' rule in the original lab11.py file to solve the vulnerability.

Due to the fact that although the sending source is different, the receiving address is always the same; by changing the match rule, we make the flow table only create entries with different target sources, rather than creating entries with different sources. This avoids the problem of flow table exhaustion.

Modified part:

```
match = parser.OFPMatch(eth_type=ether_types.ETH_TYPE_IP,
in_port=in_port, ipv4_dst=dstip, ip_proto=protocol,
udp_dst=u.dst_port)
```

This is the END of the Report