SCHOOL OF SCIENCE AND ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

New Paltz
STATE UNIVERSITY OF NEW YORK

EGC493: Software Defined Networks
Instructor: Nisar Ahmad
Final Project Report

Distributed Denial of Service Attack Simulation with Floodlight Controller Integration

Team Member	Major
George Dagis	CE
	CE
	CE
	EE
	CE

December 18, 2018

Abstract

This project is designed to launch a Distributed Denial of Service (DDoS) attack on a Mininet virtual network using a Floodlight topology, detect the attack using Floodlight, and graph the resulting traffic flow via sFlow. Mininet is used on a virtual machine in order to quickly create a virtual network, with Floodlight being used to quickly generate the topology for the network. A DDoS attack simulation is then generated by using terminal commands which will flood the network with undesired and useless traffic. The attack will be identified and prevented using the existing features of Floodlight, while sFlow keeps a running graphical log of the traffic experienced by the network.

Table of Contents

Page

Introduction

A Software Defined Network (SDN) is a method of cloud computing that allows for improved efficiency in network configuration, performance, and monitoring. One popular SDN association is with OpenFlow, which is a communication protocol allowing the user to access to the appropriate plane so that they can analyze the path of network paths across network switches. The SDN framework provides a level of flexibility that does not exist in more traditional static architectures.[1]

Due to the controller's central view of the network, the SDN framework offers an ability to enhance network security applications. Therefore, to better understand the mannerisms and features of such an architecture, a Distributed Denial of Service (DDoS) attack will be simulated. A DDoS attack occurs when the bandwidth or resources of a network are flooded in order to prevent normal usage of the network's resources by its intended users. To simulate such an attack, a terminal command will continuously ping the network, thus flooding it with undesired and useless traffic.

Mininet, which is a virtual network emulator, can generate a network of virtual hosts. This tool will be used to simulate the network which will be flooded by the DDoS attack. The goal of this project will be to develop a methodology to prevent the attack from flooding the network. This will be achieved by implementing the Floodlight controller, which has the capabilities of detecting and monitoring intrusions. This will identify the DDoS attack, isolate it on the data plane, and prevent the flooding so as to preserve the intended functionality of the created network.

Theory

In order to generate a virtual network, simulate an attack, and demonstrate an ability to prevent it, several applications must be analyzed and understood. These include: Mininet, Floodlight, and sFlow.

Mininet

Mininet is a network emulator running off of a virtual machine in order to creates a virtual network of hosts, controllers, links, and switches.[2] Mininet is an incredibly versatile tool that is used widely for research and development, but also for developing prototypes and debugging. It allows simulation of system behavior and performance so that experimentation can yield meaningful results.

Floodlight

The Floodlight OpenFlow controller is an open source controller that can both detect intrusions, but can also prevent them. It can analyze traffic in real time, logging each package on the Internet Protocol (IP) network. Floodlight can be used with both physical and virtual OpenFlow-compatible switches. Working in a variety of environments, the versatility of this controller makes it extremely beneficial, as businesses can easily implement it into whatever they already have at their disposal. [3]

sFlow

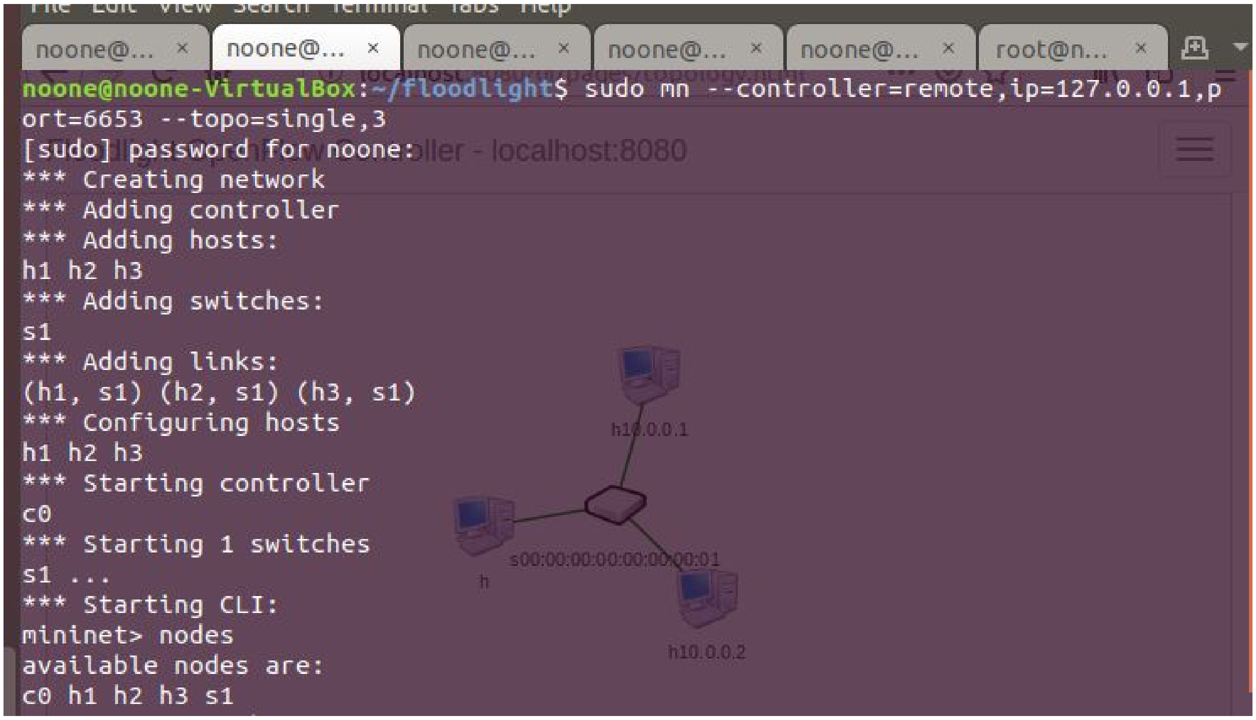
sFlow is an industry standard used for analysis of high speed switched networks. By utilizing sFlow, a graphical analysis of the incoming packets can be conducted and interpreted. Through this, the effects of intrusion prevention can be seen more clearly.

Procedure

In order to begin this project, both VirtualBox and Mininet must be downloaded. After installing each application, Mininet needs to be configured via VirtualBox and a host-only network adapter must be created. Having completed configuration, the user is able to login by using “mininet” as both the username and password.

After logging in, all required packages need to be installed. This is accomplished by the *mininet/util/install.sh -a* command. Floodlight and sFlow also need to be installed via the *sudo apt-get* command.

Next, the DDoS attack simulation must be generated. A basic network is created via Floodlight to allow for creation of two interfaces for three different hosts (though, only two will communicate). The creation can be seen in Figure 1, whereas the final topology can be seen in Figure 2.



```
noone@noone-VirtualBox:~/floodlight$ sudo mn --controller=remote,ip=127.0.0.1,p
ort=6653 --topo=single,3
[sudo] password for noone:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> nodes
available nodes are:
c0 h1 h2 h3 s1
```

Figure 1. The command lines for generating the network

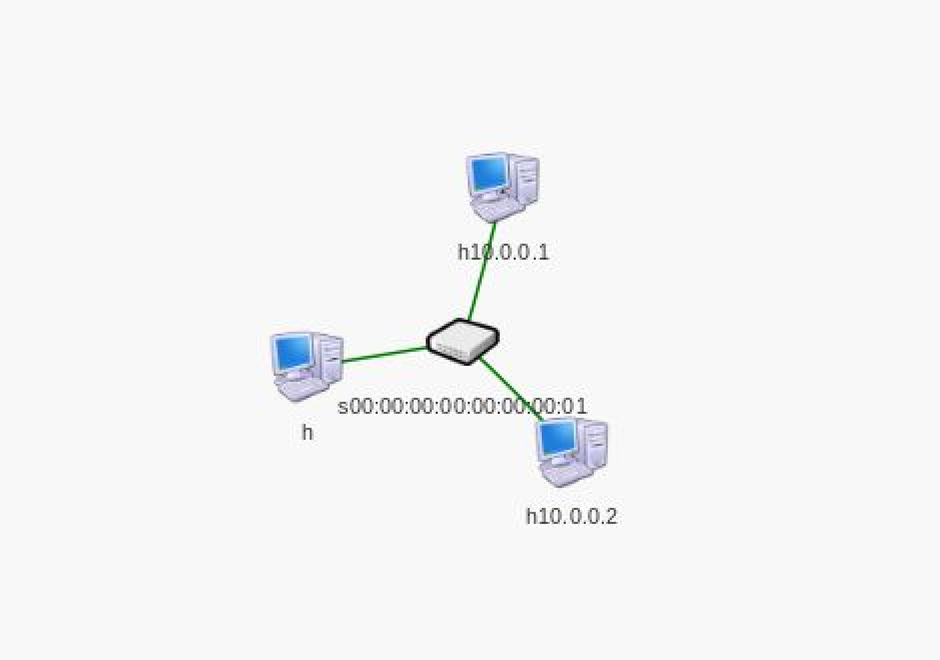
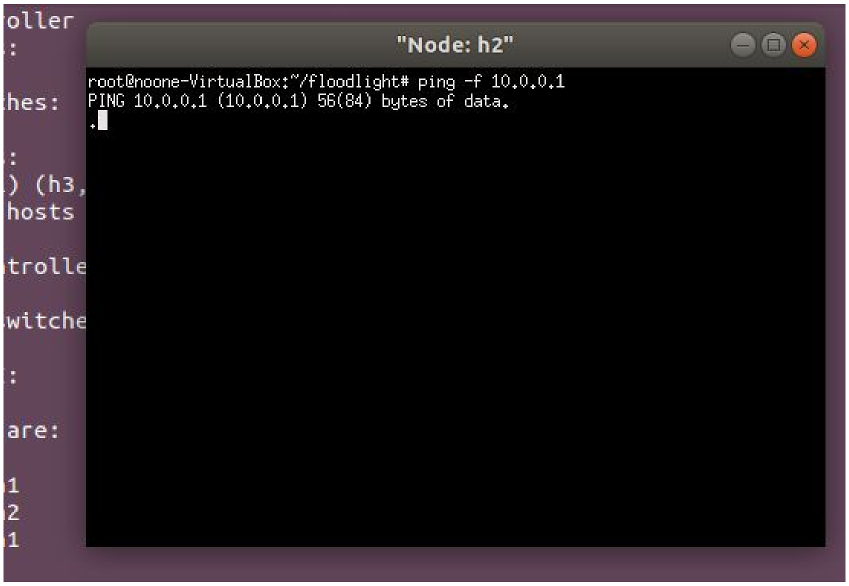


Figure 2. The resulting topology

Now, one of the hosts will attack the other using the *ping -f* command followed by the target IP address. By doing this, the second host will be flooded attacks sending irrelevant packets. This command can be seen in Figure 3.



```
root@noone-VirtualBox:~/floodlight# ping -f 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
.
```

Figure 3. The command to flood

To identify and prevent such an attack, Floodlight's firewall feature is needed to detect threats occurring via real-time traffic analysis and packet logging. Finally, after executing the activating the firewall, the system will be able to identify any threats as well as determine their origin, with sFlow monitoring the traffic flow.

Results and Discussion

The initial design of this project was intended to use a Mininet developed network and to prevent a DDoS attack using Snort integration. Unfortunately, after repeated attempts to achieve this, the only step that could not be achieved was full Snort functionality. Due to an inability to resolve this issue, and to make the most of the available time allotted, the project shifted focus to using a different OpenFlow controller to assist in intrusion detection, in this case, Floodlight.

After having followed the required steps for installing Floodlight and making use of its many features, a successful DDoS attack was launched, and could be analyzed using sFlow, as seen in Figure 4 on the following page.

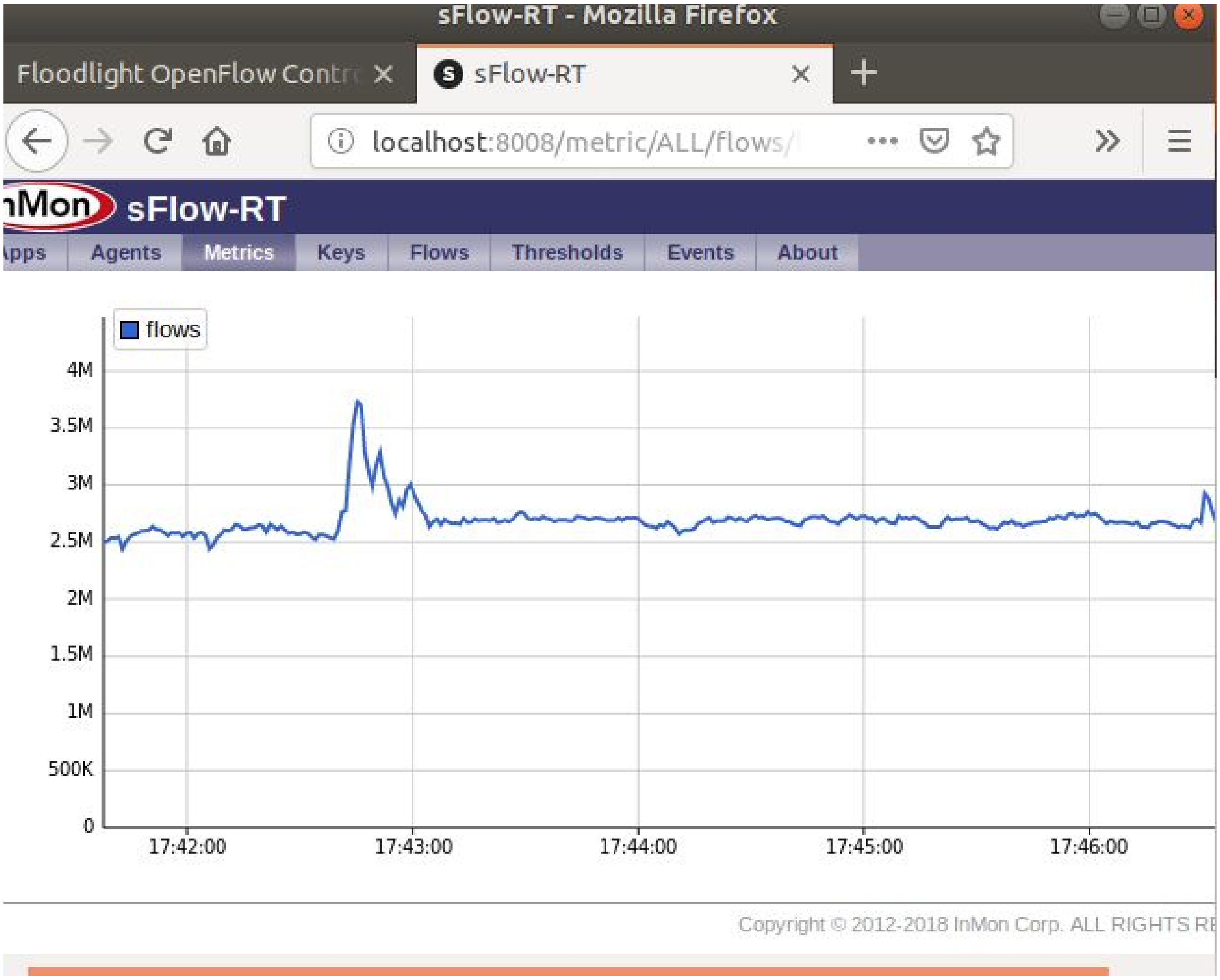
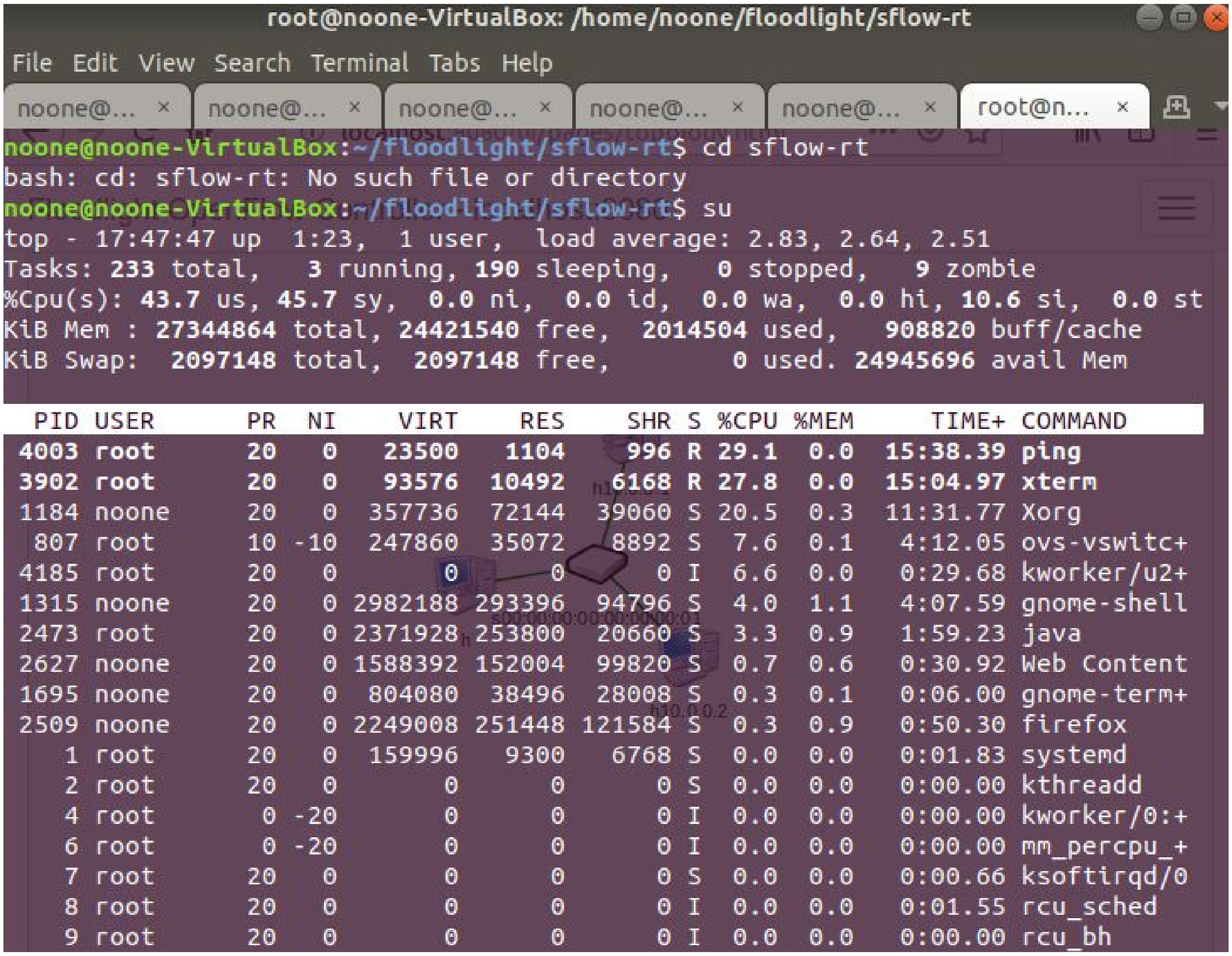


Figure 4. The sFlow traffic graph

Floodlight also offers the ability to do more in depth analysis of all traffic flow, and is an extremely useful tool for better understanding the nature of intrusion detection. Such data can be seen in Figure 5 on the following page.



root@noone-VirtualBox: /home/noone/floodlight/sflow-rt
File Edit View Search Terminal Tabs Help
noone@... ×
noone@... ×
noone@... ×
noone@... ×
noone@... ×
root@n... ×

```
noone@noone-VirtualBox:~/floodlight/sflow-rt$ cd sflow-rt
bash: cd: sflow-rt: No such file or directory
noone@noone-VirtualBox:~/floodlight/sflow-rt$ su
top - 17:47:47 up  1:23,  1 user,  load average: 2.83, 2.64, 2.51
Tasks: 233 total,   3 running, 190 sleeping,   0 stopped,   9 zombie
%Cpu(s): 43.7 us, 45.7 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi, 10.6 si,  0.0 st
KiB Mem : 27344864 total, 24421540 free,  2014504 used,   908820 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used. 24945696 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 4003 root      20   0   23500   1104    996 R 29.1  0.0  15:38.39 ping
 3902 root      20   0   93576  10492   6168 R 27.8  0.0  15:04.97 xterm
 1184 noone     20   0  357736  72144  39060 S 20.5  0.3  11:31.77 Xorg
  807 root      10 -10  247860  35072   8892 S  7.6  0.1   4:12.05 ovs-vswitc+
 4185 root      20   0       0      0      0 I  6.6  0.0   0:29.68 kworker/u2+
 1315 noone     20   0 2982188 293396  94796 S  4.0  1.1   4:07.59 gnome-shell
 2473 root      20   0 2371928 253800  20660 S  3.3  0.9   1:59.23 java
 2627 noone     20   0 1588392 152004  99820 S  0.7  0.6   0:30.92 Web Content
 1695 noone     20   0  804080  38496  28008 S  0.3  0.1   0:06.00 gnome-term+
 2509 noone     20   0 2249008 251448 121584 S  0.3  0.9   0:50.30 firefox
    1 root      20   0  159996   9300   6768 S  0.0  0.0   0:01.83 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:+
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_+
    7 root      20   0       0      0      0 S  0.0  0.0   0:00.66 ksoftirqd/0
    8 root      20   0       0      0      0 I  0.0  0.0   0:01.55 rcu_sched
    9 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_bh
```

Figure 5. In depth analysis of traffic

Conclusion

By completing this project, a more in depth and meaningful understanding of Software Defined Networks, Distributed Denial of Service attacks, and detection/prevention methodologies was achieved. The Mininet virtual network and Floodlight controller provided many useful tools when generating the network, launching the DDoS attack, and preventing said attack with the Floodlight firewall. Therefore, this project can be considered a success, as it resulted in a stronger understanding of the SDN framework.

References

[1] Benzekki Kamal et al.Software‐defined networking (SDN): a survey., Security and Communication Networks 9, no. 18 (2016): 5803-5833.
Benzekki Kamal et al.Devolving IEEE 802.1 X authentication capability to data plane in software‐defined networking (SDN) architecture., Security and Communication Networks 9.17 (2016): 4369-4377.

[2] Team, M. (2018). Mininet: An Instant Virtual Network on your Laptop (or other PC) - Mininet. [online] Mininet.org. Available at: http://mininet.org/ [Accessed 19 Dec. 2018].

[3] Project Floodlight. (2018). Floodlight OpenFlow Controller -. [online] Available at: http://www.projectfloodlight.org/floodlight/ [Accessed 19 Dec. 2018].