

CSCE 665: Homework 3

SDN Application Development

1. Overview

The main target of this assignment is to expose how to write SDN (Software Defined Networking) applications to achieve expected functions and design your own applications to control over networks. You will be introduced to OpenFlow protocol, SDN controllers (e.g. Floodlight) and FRESKO. After finishing this assignment, you are expected to be able to develop and implement your own SDN controller apps.

2. Background

This assignment assumes that you already have the basic knowledge of SDN/OpenFlow and roughly how OpenFlow network works. In this assignment we choose Floodlight [1] SDN controller and FRESKO[2] platform. Floodlight is a commercial grade controller and widely used in enterprise and cloud networks. FRESKO is an SDN-based network security service composition framework.

3. Environment Setup

Here is a VM image from SDN Hub. This VM has pre-installed most the required tools (e.g., Floodlight Controller).

<http://sdnhub.org/tutorials/sdn-tutorial-vm/>

I suggest you download the 32-bit OVA file.

Download FRESKO in the virtual machine use command:
git clone https://github.com/xuraylei/fresco_floodlight.git

After our basic lab introduction in the class, you should know and be able to finish the following things by yourself before starting the tasks (you can also refer to [3]):

- (1). Use VirtualBox to import the VM image.
- (2). Use Eclipse to open the Floodlight/FRESKO project. (you may use Floodlight to finish first and FRESKO in separate eclipse project)
- (3). Compile and run the controller.
- (4). Use Mininet to simulate the network environment.
- (5). Use Wireshark to monitor OF Messages.

4. Task One: Your First App!

Now you are going to create your first OpenFlow application (also called module in Floodlight) in the Floodlight controller! Our first application is called *mactracker*. It will watch for new MAC

addresses that have not been seen before, and log the MAC and switch they were seen on. This task is to help you familiar with how to create a new application, how to add an event listener and how to handle the Packet_In Message. We already provide the source code for you. Here is a tutorial about how to finish this task step by step:

<https://floodlight.atlassian.net/wiki/display/floodlightcontroller/How+to+Write+a+Module>

Just follow the instructions. One important thing you should note is that at the end of the tutorial, it uses a command to simulate the network environment, e.g.,

“`sudo mn --topo linear,2 --controller=remote,ip=127.0.0.1,port=6653`”. For more Mininet usage, please refer to [4].

Requirement: In your report, you need to provide screenshots of important results to show that it is successful. You also need to provide your observation and thinking.

5. Task Two: SimpleSwitch

From Task One you should already know how to analyze the MAC address of incoming packets and keep track of where the host with each MAC address is located. In this task you need to write an application in the Floodlight controller which acts as an L3 switch. After running your application, any host in the network can *ping* other hosts.

Requirements: You should submit your source code in a java file with the name of `MySimpleSwitch.java`. In your report, you need to provide screenshots of important results, describe your learning algorithm, observation and thinking.

Hint: You can directly modify the *mactracker* program from Task One, modify the Packet-In handle function and implement your logic. When you receive a Packet-In message, you can learn the MAC address and IP address of this packet. Then you are able to forward packets whose destination addresses are already learned from previous packets. Our switch is actually a learning switch. In order to get *ping* reachability, you only need to consider and deal with two types of packets (ARP and ICMP). Your application should work in any kind of topology.

6. Task Three: Run FRESCO App

Now you are going to run SDN/OpenFlow application in FRESCO. The application is called *blacklist*. It will drop network traffic from a specific host in the network.

In this task, you can put exiting *blacklist.fre* into the enable folder of FRESCO, do not forget to configure the prohibited IP address in the second module of *blacklist.fre*.

For how to build and run FRESCO, you can read [link](http://success.cse.tamu.edu/fresco/document/howtoinstall.php) (<http://success.cse.tamu.edu/fresco/document/howtoinstall.php>).

One important thing you should note is to simulate the network environment, such as:

`sudo mn --topo linear,2 --controller=remote,ip=127.0.0.1,port=6653`

Then, your application are expected to block incoming packets from a specific host, e.g., 10.0.0.1.

Requirements: In the report, you should showcase the result. For example, without *blacklist.fre* enabled, the attacker can *ping* other hosts, but with *blacklist.fre*, the attacker can not *ping* other hosts any more.

7. Task Four: Write FRESCO App

From Task Three you should already know how to run an SDN application in FRESCO . In this task you need to write an application which is used to block port scanning attack. After running your application, any scanning attempts in the network are prohibited.

In this task, you need to know how to write an SDN application in FRESCO by using a script language (more tutorials are in [link](#) , <http://success.cse.tamu.edu/fresco/document/howtowriteapp.php>).

Also, you can refer to existing module list ([link](#), <http://success.cse.tamu.edu/fresco/store/module.php>) to pick up modules you need for the purpose.

Requirements: In your report, you need to provide screenshots of important results to show that it is successful. You also need to provide your observation and thinking.

Hint: You can use the same mininet topology from Task Three and use one host as the attacker and one host as victim. On the victim, you can run a webserver. Then, You can use *nmap* program to generate port scanning traffic. If your application is correct, you can see no scanning result from nmap.

8 Task Five: The Real Task of this homework – Create your own creative SDN app!

After the above warmups, you now need to create a creative, new, small SDN app (not in existing apps) to implement some new, cool network function (not necessarily tied to security functions) in FRESCO. You should submit “MyNewApp.fre” for your new fresco application and necessary FRESCO modules. In addition, you should attach screenshots for testing your application. We use the following factors to grade your new app development: novelty (the most important factor!), depth and effort (should not be too simple!), correctness (it should be complete and work as desired), and report and code quality.

8. Reference:

[1] <http://www.projectfloodlight.org/floodlight/>

[2] <http://success.cse.tamu.edu/lab/fresco>

[3] http://archive.openflow.org/wk/index.php/OpenFlow_Tutorial

[4] <http://mininet.org/walkthrough/>