**Software Defined Networks**

**CS 6301.501**

**Layer 2 Firewall**

Table of Contents

[Objectives 3](#_Toc39177633)

[Introduction 3](#_Toc39177634)

[Tools Used 5](#_Toc39177635)

[Topology 6](#_Toc39177636)

[Implementation 7](#_Toc39177637)

[Output 8](#_Toc39177638)

[Conclusion 8](#_Toc39177639)

# **Objectives**

* Implement a Layer-2 SDN Firewall.
* Use the Python based POX OpenFlow controller instead of the default Mininet Controller.
* Understand what an SDN based Firewall does and implement it from scratch using POX.

# **Introduction**

Firewall and SDN Firewall:

Broadly speaking, a computer firewall is a software program that prevents unauthorized access to or from a private network. [Firewalls](https://www.comodo.com/home/internet-security/firewall.php?af=7639) are tools that can be used to enhance the security of computers connected to a network, such as LAN or the Internet. They are an integral part of a comprehensive security framework for your network.

A firewall absolutely isolates your computer from the Internet using a "wall of code" that inspects each individual "packet" of data as it arrives at either side of the firewall — inbound to or outbound from your computer — to determine whether it should be allowed to pass or be blocked.

A Firewall can be understood to be something that obstructs traffic coming its way and filters it according to some *rules*. A general Firewall can be employed to protect a network from the internet. For an SDN based Firewall, we are going to use the OpenFlow controller to filter traffic between hosts according to some rules and accordingly let it pass through or not. The way we will do this is by using the POX controller to establish our required *policies* or *rules* and filter traffic between hosts using the switches.

A picture containing table

Description automatically generated

About Layer-2 Security Threats:

Layer 2 switched environments, typically found in enterprise customer wiring closets, can be easy targets for network security attacks.

One of the most common security threats in the Layer 2 domain, and one of those least likely to be detected, is the threat targeted at disabling the network or compromising network users with the purpose of gleaning sensitive information such as passwords. These attacks exploit normal protocol processing such as a switch's ability to learn MAC addresses, end-station MAC address resolution via Address Resolution Protocol (ARP-RFC 826), or Dynamic Host Control Protocol (DHCP) server IP address assignments.

Because any user can gain access to any Ethernet port and be a potential hacker, open campus networks cannot guarantee network security. Because the OSI model was built to allow different communications layers to work without knowledge of each other, Layer 2 security is critical. If this layer -- which provides hackers access to the information power hackers seek -- is being hacked, security is compromised without communication between the other layers being affected and without any users being aware their application-layer information had been compromised.

# **Tools Used**

* Mininet - to create a virtual network topology of choice
* POX Controller – A platform written in Python, which communicates with OpenFlow 1.0 and includes support for Open vSwitch.
* Python – The firewall rules are applied through Python

Mininet:

Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.

Mininet supports research, development, learning, prototyping, testing, debugging, and any other tasks that could benefit from having a complete experimental network on a laptop or other PC.

Mininet allows you to create different topologies consisting of OpenFlow switches and hosts (end devices). Topologies can vary from a single switch to more complicated topologies consisting of multiple switches and multiple links. You can even create your own topologies.

The virtualized switches and hosts act like real devices and you can send traffic from between hosts and then view the flows on the switches. In addition, applications such as iperf can be used to measure performance on the network.

POX Controller:

POX is a Python based open source OpenFlow/Software Defined Networking (SDN) Controller. POX is used for faster development and prototyping of new network applications. POX controller comes pre installed with the mininet virtual machine. Using POX controller you can turn dumb openflow devices into hub, switch, load balancer, firewall devices. The POX controller allows easy way to run OpenFlow/SDN experiments. POX can be passed different parameters according to real or experimental topologies, thus allowing you to run experiments on real hardware, testbeds or in mininet emulator. In this paper, first section will contain introduction about POX, OpenFlow and SDN, then discussion about relationship between POX and Mininet. Final Sections will be regarding creating and verifying behavior of network applications in POX.

# **Topology**

A close up of a logo

Description automatically generated

# **Implementation**

* Topology Set up in Mininet: We will be adding MAC addresses while initializing the hosts so that we can easily define the Layer-2 rules for our Firewall.
* For our Firewall, we will have the 4 following rules:

1) h1 and h2 are mutually blocked.

2) h2 and h4 are mutually blocked.

3) h2 and h7 are mutually blocked.

4) h3 and h8 are mutually blocked.

* We create our SDNFirewall class in which the controller is going to be accessing and checking flows and modifying flowtables accordingly.
* \_handle\_ConnectionUp will fire up each time a host tries to reach another through the switches.
* POX requires a launch function where we pass our SDNFirewall class to it.

A screen shot of a computer

Description automatically generated

# **Output**

A screenshot of a cell phone

Description automatically generated

# **Conclusion**

* We implemented a layer-2 firewall using the POX controller.
* Using rules specifying Layer-2 characteristics we were able to control the switches using POX to either permit traffic between hosts or restrict it.