# Assignment N0.3

**Aim :** Create a SDN environment on mininet and configure a switch to provide a firewall functionality using pox controller.

**Objectives :**

1. Learn installation of Mininet
2. configured a switch to provide firewall functionality using POX controller.

**Theory :**

**SDN environment**

An SDN (Software Defined Networking) environment is a network architecture that separates the network control plane from the data plane, allowing for centralized management and programmability. In an SDN environment, the network control plane is managed by a software controller, which communicates with the network devices using the OpenFlow protocol.

To create an SDN environment, several components are required, including:

1. SDN controller: The SDN controller is the central management component that communicates with the network devices using the OpenFlow protocol.

2. Network devices: The network devices, such as switches and routers, are responsible for forwarding data packets based on the instructions from the SDN controller.

3. OpenFlow protocol: The OpenFlow protocol is used to communicate between the SDN controller and the network devices.

4. Network applications: Network applications can be developed to run on top of the SDN environment, providing additional network services and functions.

5. Network topology: The network topology is the physical or logical arrangement of the network devices and links.

SDN environment is a network architecture that separates the control plane from the data plane, allowing for centralized management and programmability. SDN environments offer greater flexibility, scalability, and automation in network management, and enable new network services and applications.

**firewall functionality**

Firewall functionality is a security feature that is used to protect computer networks from unauthorized access and malicious traffic. In the context of SDN, a firewall can be implemented using an SDN controller, which controls the network switches and enforces security policies.

The firewall functionality implemented by an SDN controller typically involves setting up flow rules on the network switches to block or allow traffic based on certain criteria, such as the source and destination IP addresses, the transport protocol used, and the ports being used. For example, an SDN controller can set up flow rules to block traffic from certain IP addresses, or to block traffic on certain ports that are known to be used by malware.

**pox controller**

POX is an open-source SDN controller that is written in Python and is designed to support the OpenFlow protocol. POX is designed to be lightweight and flexible, and it provides a number of useful features for building and managing SDN networks.

Some of the key features of POX include:

1. Modular design: POX is designed to be modular and extensible, which makes it easy to add new features and functionality.

2. Python-based: POX is written in Python, which makes it easy to read, understand, and modify the code. This also means that it is easy to integrate with other Python-based tools and libraries.

3. OpenFlow support: POX supports the OpenFlow protocol, which is used to communicate between the controller and the switches in an SDN network.

4. Debugging tools: POX includes a number of built-in debugging tools that make it easy to test and troubleshoot SDN applications.

5. Flexible API: POX provides a flexible API that can be used to build custom SDN applications and services.

**Installation Process :**

To create an SDN environment on Mininet and configure a switch to provide a firewall functionality using POX controller, you can follow the steps below:

1. Install Mininet and POX controller:
   - Follow the installation guide at http://mininet.org/download/ to install Mininet
   - Follow the installation guide at https://github.com/noxrepo/pox/wiki/Installation to install POX controller
2. Create a simple topology:
   - Open a terminal and run **sudo mn --topo single,3 --mac --switch ovsk --controller remote**
   - This will create a topology with one switch and three hosts, and set the controller as remote.
3. Configure the switch to provide firewall functionality:
   - Open a new terminal and navigate to the POX controller directory
   - Create a new Python file called **firewall.py**
   - Copy and paste the following code into the **firewall.py** file:

**python**

```
from pox.core import core
import pox.openflow.libopenflow_01 as of

log = core.getLogger()

class Firewall(object):

    def __init__(self):
        core.openflow.addListeners(self)
        log.info("Firewall initialized.")

    def _handle_ConnectionUp(self, event):
        msg = of.ofp_flow_mod()
        msg.priority = 65535
        msg.match.dl_type = 0x0800
        msg.match.nw_proto = 6
        msg.match.tp_dst = 80
        msg.actions.append(of.ofp_action_output(port=of.OFPP_CONTROLLER)
)
        event.connection.send(msg)
```

```
def launch():
  core.registerNew(Firewall)
```

4. Run the POX controller:
   - In the terminal where you created the **firewall.py** file, run **./pox.py log.level --DEBUG openflow.discovery openflow.spanning_tree --no-flood --hold-down firewall**
   - This will run the POX controller with the **firewall.py** module.
5. Test the firewall functionality:
   - In the Mininet terminal, run **h1 ping h2**
   - The ping should be successful
   - In the Mininet terminal, run **h1 wget h2**
   - The wget should fail because port 80 is blocked by the firewall.

That's it! You have created an SDN environment on Mininet and configured a switch to provide firewall functionality using POX controller.

**Conclusion :**

We learn how to install Mininet and how to created an SDN environment on Mininet and configured a switch to provide firewall functionality using POX controller.