

LAB 08

Implementing a Firewall with ACLs in OpenFlow using Mininet

CO515: Advances in Computer Networks: Selected topics

Objective

Implement a firewall using Access Control Lists (ACLs) in OpenFlow using Mininet.

Tools Required:

- Mininet: A network emulator that creates a virtual network on a single machine.
- Open vSwitch: A software switch implementation with support for OpenFlow.
- Ryu: An OpenFlow controller.

Create the Mininet Topology

Create a simple network topology in Mininet with the following components:

- Two hosts (h1 and h2)
- One switch (s1) connecting the hosts.

Launch Mininet with a basic topology

```
sudo mn --topo single,2 --mac --switch ovsk --controller remote
```

Start Ryu controller: Start the Ryu controller with the `simple_switch_13` application, which will act as the OpenFlow controller.

```
$ryu-manager ryu.app.simple_switch_13
```

Implement ACLs with OpenFlow: Configure ACLs to allow or block traffic based on source IP address. Below is a part of the configuration file.

- This Ryu application listens for switch features and packet-in events.
- The **switch_features_handler** method sets a default rule to send unknown packets to the controller.
- The **packet_in_handler** method inspects incoming packets and applies ACL rules.

```
# ACLFirewall.py

from ryu.base import app_manager
from ryu.controller import ofp_event
from ryu.controller.handler import CONFIG_DISPATCHER, MAIN_DISPATCHER
from ryu.controller.handler import set_ev_cls
from ryu.ofproto import ofproto_v1_3
from ryu.lib.packet import packet
from ryu.lib.packet import ethernet, ipv4

class ACLFirewall(app_manager.RyuApp):
    OFP_VERSIONS = [ofproto_v1_3.OFP_VERSION]

    def __init__(self, *args, **kwargs):
        super(ACLFirewall, self).__init__(*args, **kwargs)
        self.mac_to_port = {}

    @set_ev_cls(ofp_event.EventOFPSwitchFeatures, CONFIG_DISPATCHER)
    def switch_features_handler(self, ev):
        datapath = ev.msg.datapath
        ofproto = datapath.ofproto
        parser = datapath.ofproto_parser

        # Table-miss flow entry
        match = parser.OFPMatch()
        actions = [parser.OFPActionOutput(ofproto.OFPP_CONTROLLER, ofproto.OFPCML_NO_BUFFER)]
        self.add_flow(datapath, 0, match, actions)

    def add_flow(self, datapath, priority, match, actions):
        ofproto = datapath.ofproto
        parser = datapath.ofproto_parser

        inst = [parser.OFPInstructionActions(ofproto.OFPIT_APPLY_ACTIONS, actions)]
        mod = parser.OFPFlowMod(datapath=datapath, priority=priority,
                                match=match, instructions=inst)
        datapath.send_msg(mod)

    @set_ev_cls(ofp_event.EventOFPPacketIn, MAIN_DISPATCHER)
    def packet_in_handler(self, ev):
```

Task

1. Test packet switching again by using ping or other connectivity checks.

mininet> h1 ping h2

2. Verify that traffic from **IP_x** to h2 is blocked.
3. You are required to submit the ACLfirewall.py written by yourself and the test results with screen recording.