

# Implementation of SDN Stateful Firewall at Data Plane Applying in Open vSwitch

Harsh Kasavalekar<sup>1</sup>, Mayur Kachare<sup>2</sup>, Prof. K. S. Suresh Babu<sup>3</sup>

<sup>1,2</sup>BE Student, Department of Information Technology, MES Pillai College of Engineering, Navi Mumbai, India

<sup>4</sup>Assistant Professor, Department of Computer Engineering, MES Pillai College of Engineering, Navi Mumbai, India

\*\*\*

**Abstract** - A Software Define Network (SDN) has been deployed in the current network system. Together with Network Virtualization (NVF), it makes the network more flexible. The firewall can be implemented in SDN. However, with the limitation of earlier version of OpenFlow protocol used in SDN, the stateful firewall could not be implemented with the SDN standard. The development of OpenFlow enables some features that can be used for implementing the stateful firewall. In our project, we will implement the stateful firewall in the SDN switch on the data plan. The Open vSwitch is used. We also evaluate the performance of the SDN stateful firewall with more complex network topology and more security attack cases such as SYN flood attack, HTTP flood attack, UDP flood attack, Ping flood attack.

**Key Words:** Software Defined Networking(SDN), Data plane, Stateful firewall, Open vSwitch, mininet.

## 1. INTRODUCTION

A Software Defined Network (SDN) become used in the network. It allows the routing and policy definitions into a network system. It aims to centralized a network. SDN uses different protocols to deliver packages to other networking systems. In SDN, the controllers are in control plane and network devices are in the data plan. The OpenFlow protocol managed for communication between the control plane and the data plane. Insecurity concern, a stateful firewall is used. The stateful firewall examines the connections and allows only the flows that are connections that have been initiated by the clients of the internal network. A stateful firewall does not permit traffic from outside networks. In Software Defined Network, we implement Syn flood, UDP flood, HTTP flood also Ping flood attacks in the networks. Their Open vSwitch is used. There are many packets that continue sent to the targeted server to stop the strength of that device to process. The stateful firewall implement in protecting the targeted server, resulting in authentic traffic.

## 2. LITERATURE SURVEY

1. Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch[1].

In this paper, Software Defined Network applies in the network system. The firewall can be implemented in SDN. The OpenFlow protocol are permits the features that can be used for enforcing the stateful firewall. In this work, we implement a stateful firewall in the data plane. These examine the performance of the SDN stateful firewall. The results show that our SDN stateful firewall can work well with reducing overhead increased in SDN switches.

2. Extending Open vSwitch to facilitate creation of stateful SDN applications[2].

In this paper(Extending Open vSwitch to facilitate the creation of stateful SDN application) this paper details a continuation to Open vSwitch that admits for ensuring applications to run with reduced controller interaction. This can reduce the overall network capacity also can create utilization faster. It is verifying the need for reduced controller interaction in Open vSwitch.

3. An SDN-Based Lightweight Countermeasure for TCP-SYN flooding attack[3].

In this paper(An SDN-Based Lightweight Countermeasure for TCP-SYN flooding attack) is lightweight extensions to controls the Syn flood attacks. It is implemented in a control plane in the SDN and monitors the network connections. A SLICOTS is able to detect and prevent unauthorized activity. It also blocks the attacker. The results show the SLICOTS is an effective solution over a Syn flood attack.

4. FLOWTRACKER: A SDN stateful firewall solution with adaptive connection trackers and minimized controller processing[4].

In this paper a SDN firewall connection are established and the processing of the SDN controller that is minimised. In this firewall challenges are evaluated. The SDN stateful firewall is used to set up the connecion so that the network that is created that can communicated between the control plane and the data plane. In this the firewall is preventing the packet tracing in the system. The stateful firewall flows the packets to the different hosts through the ovswitch.

5. Supporting virtualized network functions with stateful data plane abstraction[5].

In this paper it supports the virtualized network that functions with stateful data plane abstraction. The network virtualization is based on the software defined network that enhance the programming features of the virtualized network. However, these function are needs to be concrete stateful information or the data that effectively process the network. In these, the openflow protocol is used to provide simple action and a shortage of virtualized network function that transfer or process the data plane in the SDN. The SDN controller is used to maintain the scalability and performance problems. A pre-processing unit and the processor is designed for SDN switches to manage and control the stateful information through the instructions that are provided. In this it shows the result of the network virtualization that process the stateful information and the capabilities are improved.

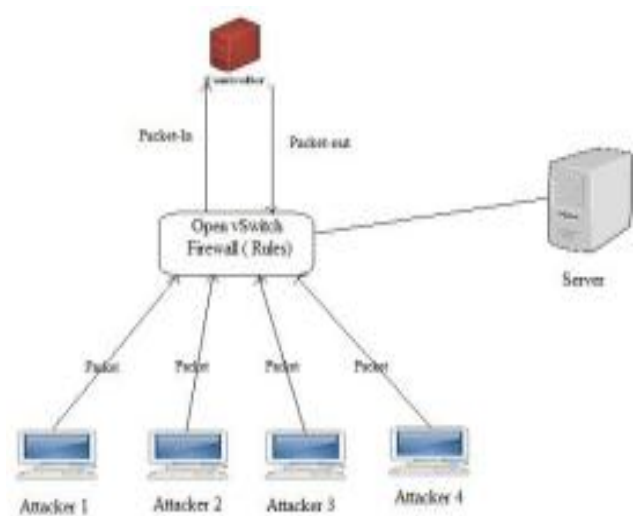
6. SDPA: Enhancing stateful forwarding for Software Define Networking[6].

In this paper it enhances the stateful forwarding for the SDN network in the system. In this the architecture of the stateful data plane is described for the Software Defined Networking data plane. In SDPA the switches are designed by the processor to control the state information to set the instructions and the state table. It implements the OpenFlow protocol to make the Communication between the controller and the processor. To implement it, both hardware and software of Software Data Plane Architecture SDPA switches are develop a network function for the defence attack reflection in the stateful firewall in an SDPA switch. It improved the SDPA architecture in which applications are needed for forwarding the stateful SDN based network.

7. An OpenFlow based prototype of SDN oriented Software Hardware Firewall[7].

In this paper(An OpenFlow based prototype of SDN oriented Software Hardware Firewall) implements an Open Flow-based prototype with a stateful hardware firewall. In this paper, the firewall defines the security rules in the flow tables. The controller is used to control decisions and only allows legitimate traffic flows. This channel is used to sends information to the controller because it is able to make decisions. Pressurizing this communication overhead is important to the applicability of the prototype because a high communication overhead could monitor the performance evaluation of the operation.

3. SYSTEM DESIGN



The proposed system architecture consists of the Open vSwitch Firewall which is connected to controller, server and the attacks used in the system. The controller is used for the packet IN-OUT transmission which is connected to the Open vSwitch Firewall. The attacks used in the Open vSwitch Firewall are sending signals to the Server. When the Open vSwitch Firewall receives the signals if it is check signal is legitimate then it will send to the server otherwise block it. Firewall adds the rules using the Open vSwitch and OpenFlow. After the controller again sends the signals to the other system in the networks. Mininet SDN simulator is used to create virtual hosts and virtual switches. Wireshark is used for the different attacks packets analyzer in real-time. In this system implementing the more than one flood attack in the networking using the floodlight controller. We implement the SYN flood, UDP flood, HTTP flood, and ping flood attacks.

4. IMPLEMENTATION DETAILS

4.1 OpenFlow Protocol

OpenFlow protocol operates into a south-board of the Software Defined Networking architecture. Openflow having the components at least one flow table and one group table. Openflow uses basically packet lookup and forwarding. With the help of the controller, openflow can add, update and delete flow entries in flow tables and they can do either reactively or proactively.

4.2 Open vSwitch

An OpenFlow switch is an OpenFlow-enabled data switch that communicates over the OpenFlow channel to an external controller. It is an capable virtual switch that is typically used with hypervisor to interconnect virtual machine within a host and virtual machine between

different hosts across network. In open vSwitches creates the virtual hosts and performs a stateful firewall.

### 4.3 Stateful Firewall

A stateful firewall uses the state table to identify legitimate connections and keep track of them. If the packets are matching the rule that allows and remaining packets will be rejected. It has a security feature it is known as a Dynamic Packet Filter. This firewall checks the network traffic and knowledge of connections. It holds a record of the network connections. The stateful firewall controls incoming and outgoing packets.

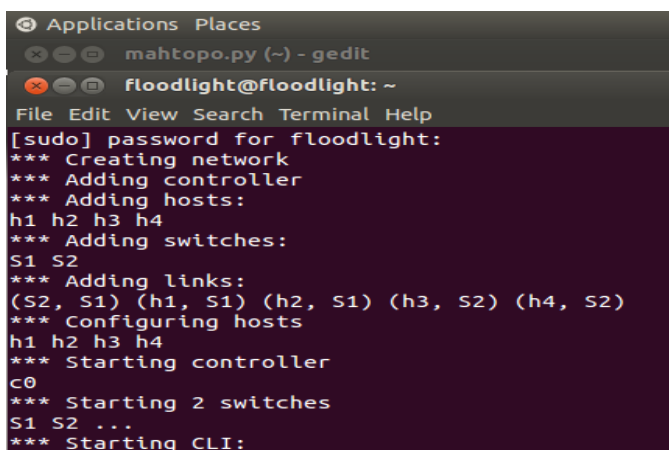
### 4.4 Mininet

It is an network emulator software that allows you to launch a virtual network with switches, hosts and a SDN controller all with a single command. In mininet its check the testing ping reachability of the host.

## 5. PROJECT INPUT AND OUTPUT SCREENSHOTS

### 5.1 Network Topology

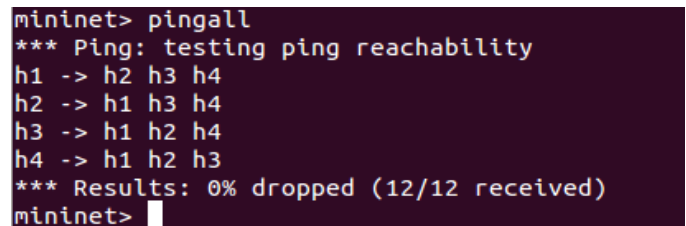
In our project we created a network topology by typing a command such as "sudo mn" and in this command, we gave a file path which is a python file and that we have created and saved on a desktop window so that the floodlight can easily fetch the file. The network topology consists of 4 hosts, 2 switches and 1 controller.



```
Applications Places
mahtopo.py (~) - gedit
floodlight@floodlight: ~
File Edit View Search Terminal Help
[sudo] password for floodlight:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
S1 S2
*** Adding links:
(S2, S1) (h1, S1) (h2, S1) (h3, S2) (h4, S2)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 2 switches
S1 S2 ...
*** Starting CLI:
```

### 5.2 Ping Command

Ping command is used to send packets of information or data to a specific IP address on a network. The PCs send several packets of data to the device and wait for a response the ping command shows how long each packets takes their time to reach. By ping command, we can ping our loopback address.



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)
mininet>
```

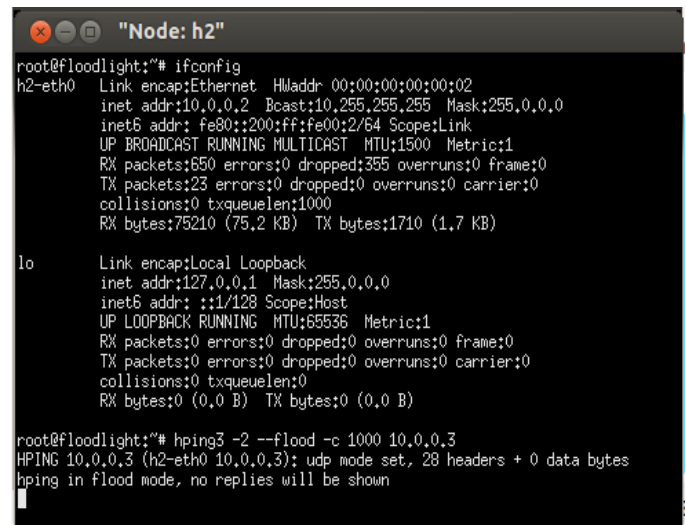
Pingall command is used to ping all hosts and test its ping reachability.

### 5.3 UDP Attack and Wireshark

We are sending the packet to our target to see if we get a response. Since port 80 uses TCP as its transport layer protocol, it should be interesting to see what kind of response we get when we send it a UDP packet.

#### hping3 -2 -flood -c 1000 10.0.0.3

By using -2 in this command, we specify to use UDP as our transport layer protocol. We can see in the output that we got ICMP Port Unreachable response due to that port not being open for UDP traffic.



```
"Node: h2"
root@floodlight:~# ifconfig
h2-eth0  Link encap:Ethernet  HWaddr 00:00:00:00:00:02
         inet addr:10.0.0.2  Bcast:10.255.255.255  Mask:255.0.0.0
         inet6 addr: fe80::200:ff:fe00:2/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:650 errors:0 dropped:355 overruns:0 frame:0
         TX packets:23 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:75210 (75.2 KB)  TX bytes:1710 (1.7 KB)

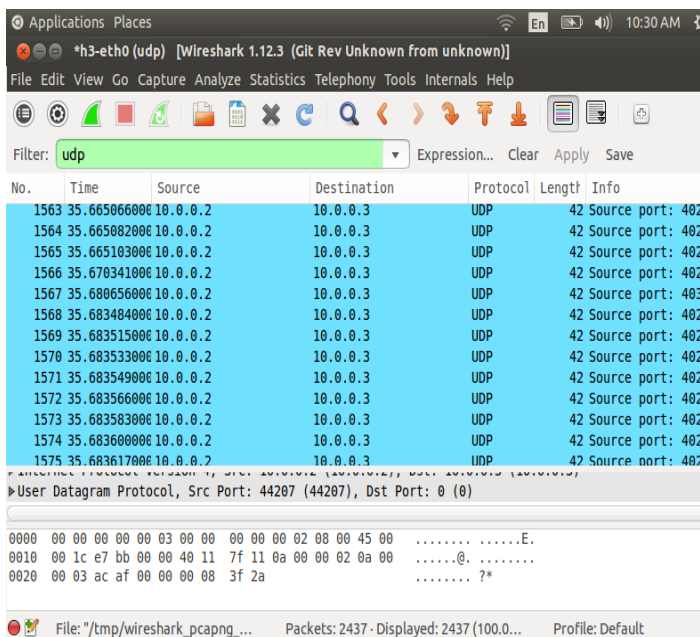
lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:65536  Metric:1
         RX packets:0 errors:0 dropped:0 overruns:0 frame:0
         TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@floodlight:~# hping3 -2 --flood -c 1000 10.0.0.3
HPING 10.0.0.3 (h2-eth0 10.0.0.3): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

In the above command, the hping3 is used as the name of the application binary. The -2 in this command tells hping3 to use ICMP, which by default sends an Echo Reply. The -c 1000 states that we only want to send one packet and host3 is our target. The -flood in this command is used for sending packets as fast as possible, without taking care to show incoming replies and flood mode.

### 5.4 Wireshark

In Wireshark, we will see the datagram packets that flow. In Wireshark, the data packets are filtered in which the UDP we select and enable the switch and the loopback address to display the data.



The image shows a Wireshark network traffic capture window. The filter is set to 'udp'. The packet list table shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
1563	35.665066000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1564	35.665082000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1565	35.665103000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1566	35.670341000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1567	35.680656000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1568	35.683484000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1569	35.683515000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1570	35.683533000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1571	35.683549000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1572	35.683566000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1573	35.683583000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1574	35.683600000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402
1575	35.683617000	10.0.0.2	10.0.0.3	UDP	42	Source port: 402

The packet details pane shows: User Datagram Protocol, Src Port: 44207 (44207), Dst Port: 0 (0). The packet bytes pane shows hex and ASCII data.

## 6. CONCLUSION

In these paper, we propose Implementation of the SDN stateful firewall in the data plane using Open vSwitch. In our implementations, we uses OpenFlow protocols determines the path of network packets across a networks of switches and open vSwitch which add the rules in the SDN switch. In a project, the SDN stateful firewall is implemented and aslo implement a network virtual topology using SDN controller. In floodlight we will see the data that is transferred from one host to another host via a SDN controller. Simultaneously we are preventing the attacks with the help of stateful firewall. The packes are send simultaneously from source to another destination systems. The flow rules works correctly in the stateful firewall.

## ACKNOWLEDGEMENT

We would like to thank our mentor, Prof. Suresh Babu for guidance and unwavering support throughout the project and the semester. We would like to thank our HOD, Dr. Satish Kumar L. Varma for their encouragement and motivation to learn and implement projects of sorts. Lastly, we would like to thank our principal, Dr. Sandeep Joshi for providing us opportunities to explore our domain and for motivating us to do better.

## REFERENCES

- [1] Ali Zeineddine, Wassim El-Hajj "Stateful Distributed Firewall as a Service in SDN.", 2018.
- [2] Jake Collings, Jun Liu "An OpenFlow based prototype of SDN oriented Software Hardware Firewall.", 2014.

- [3] Jun Bi, Shuyong Zhu, Chen Sun "Supporting virtualized network functions with stateful data plane abstraction.", 2016.
- [4] Laura Galluccio, Sebastiano Milardo, Sergio Palazzo "SDN-WISE: Design, prototyping and experimentation of a stateful SDN solution for Wireless Sensor networks", 2015.
- [5] Pakapol Krongbaramee(Chiang Mai University ), Yuthapong Somchit(Chiang Mai University). "Implementation of SDN Stateful Firewall on Data Plane using Open vSwitch", 2018.
- [6] Reza Mohammadi, Reza Javidan, Mauro Conti "An SDN-Based Lightweight Countermeasure for TCP-SYN flooding attack.", 2017.
- [7] Salaheddine Zerkane, David Espes, philippe Le Parc "Software Defined Networking Reactive Stateful Firewall.", 2016.
- [8] Shuyong Zhu, Jun Bi, Chen Sun, Chenghui Wu, Hongxin Hu "SDPA: Enhancing stateful forwarding for Software Define Networking.", 2015.
- [9] Timothy Adam Hoff "Extending Open vSwitch to facilitate creation of stateful SDN applications.", 2018.