Volume 6, Issue 6, June 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Development of Top Layer Web Based Filtering Firewall using Software Defined Networking

Heena, Japinder Singh
Department of Computer Science and Engineering, SBSSTC, Ferozepur,
Punjab, India

Abstract—Software defined networking is a new paradigm in the field of networking which is bringing a drastic change in this technology. The main logic behind SDN is separation of control plane and data plane which makes network management simpler. With the separation of control and data plane, now it is much easier to introduce new ideas through software programs as these are easier to manipulate and change them in proprietary network devices. This paper focuses on developing a web based filtering firewall which will work on top layer of TCP/IP model with the help of openflow protocol. This implementation shows that firewall functionalities can be built on software without the need of dedicated hardware. We have chosen VMware workstation and POX controller written in python for the implementation purpose. For creating network topology, we have used mininet emulation tool. This study covers implementation details as well as experimental results.

Keywords— SDN, Firewall, Openflow, Mininet, Pox

I. INTRODUCTION

In traditional networks, Most of the network devices are dedicated and purpose built devices such as switches, firewalls, routers, load balancers and proxies. All these devices have a specific purpose and during manufacturing that specific purpose or logic is coded into these devices which are very difficult to modify in future. Also, these devices are very time consuming and costly. Software of all these devices is vendor specific.

To cope up with these issues, Software defined networking was introduced. SDN is a new and exciting technology in the field of networking which makes networks more flexible, cost efficient and dynamic by decoupling data plane and control plane. SDN uses configurable and customizable software that is independent of hardware [2]. Fig 1 shows the architecture of SDN.

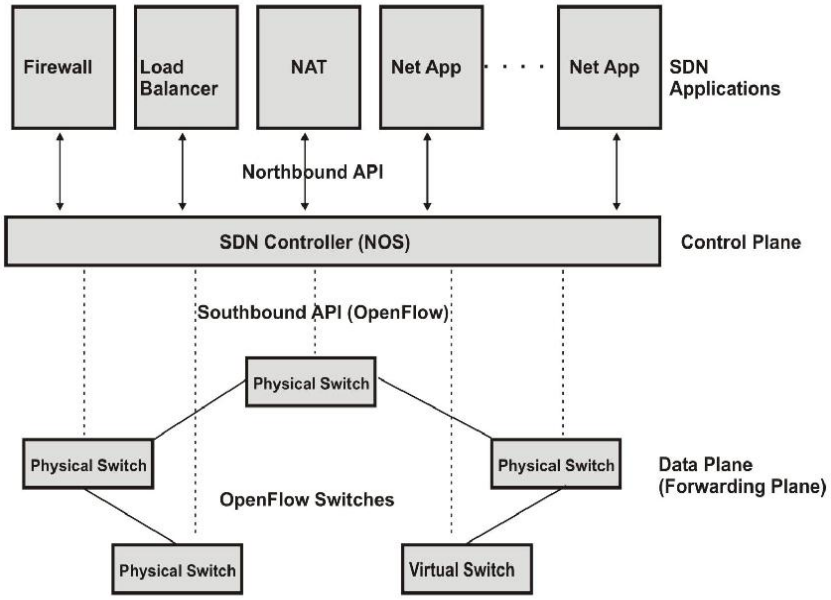

Fig. 1 Architecture of SDN

SDN can be defined as network architecture with 3 pillars [3]:

1) In software defined networking, data plane and control plane are separated. Network devices will become simple packet forwarding devices and will be separated from control functionality.
2) control plane contains SDN controller which is the brain of network devices and all the control logic will be stored in controller

3) Network is programmable and can be modified through Software applications that run on the top of controller which interacts with the data plane devices. This is main characteristic of SDN.

The interface used for communication between control plane and data plane is Southbound API also known as openflow protocol which is a part of southbound interface, it defines the way controller and packet forwarding devices interact with each other and interface which helps in communication of control plane and application plane(where all applications reside) is Northbound API which represents a northbound interface i.e. interface for developing applications, network configuration and management [3].

Openflow Switch consists of three main components.1) openflow compliant switch 2) Secure channel and 3) a Controller. To forward packets, flow table containing flow entries is used by openflow switch. Each flow entry consists of three parts that are match field, counter and actions [4]. Each incoming packet is compared to a particular flow entry and then a specific action is performed on that packet which can be 1) forward the packet to particular port 2) Drop the packet and 3) if no matching occurs then encapsulate and forward the packet to controller. Depending on the combination of rules installed, an openflow switch can behave as a router, firewall or load balancer [5]. Fig 2 represents openflow switch.

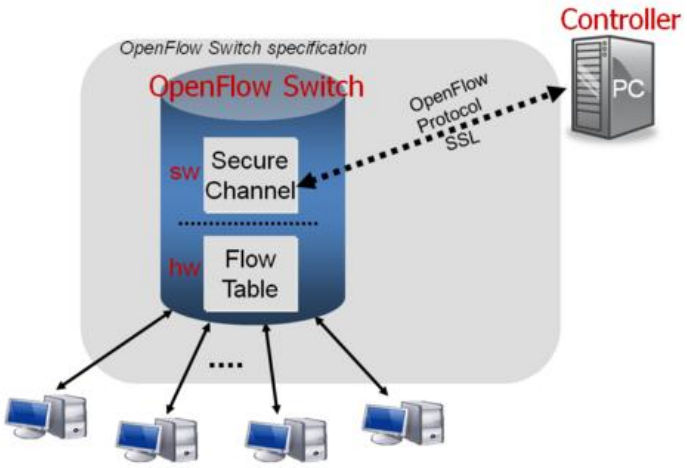

Fig. 2 Openflow Switch

Two types of approaches are used by openflow controller in implementing firewall i.e. Reactive and Proactive [6]. In reactive Approach, packets are handled as they come in. This mode reacts according to traffic then the openflow controller is consulted and rules are created in the flow table based on the instruction that's why it takes additional setup time. On the other hand, in Proactive approach, rules are pre-installed in the flow table of openflow switch so this approach has zero additional flow setup time [7]. This is shown in fig 3.

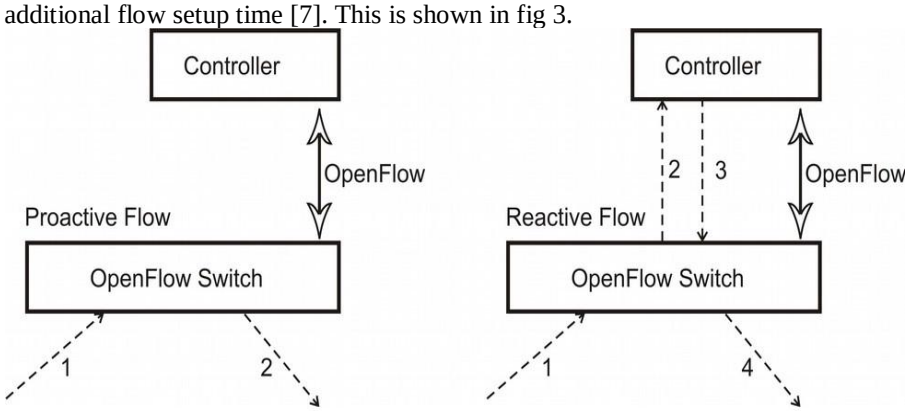

Fig. 3 Proactive and Reactive Flows

Firewall may work at different layers in OSI or TCP/IP network models. The lowest of which is network layer in OSI model, which is packet filter firewall. In this type of firewall, filtering can be done on basis of source and destination MAC address, Source and destination IP address or Source and destination port numbers [8]. This is concept of packet filter firewalls which are first generation firewalls. Then second generation firewalls are stateful firewalls which keeps track of state of each incoming packet and these firewalls retain packet until enough information is available to find its state. This is known as stateful packet inspection. These firewalls operate up to transport layer [9]. Third kinds of firewalls are third generation firewalls that are application layer firewalls. Application layer firewall works on the top layer of TCP/IP model and it may intercept all packets travelling to and from an application. This study focuses on implementation of top layer web based filtering firewall which will block certain applications based on signatures and rules installed in firewall table and application table.

II. EXPERIMENTAL SETUP

For experimental setup, mininet emulation tool has been used. With the help of this tool, large networks on virtual machine or simple single computer with limited resources can be developed [10].This tool also helps you to create different topologies with number of virtual hosts, switches, controllers and links. VMWARE WORKSTATION has been used for virtualization. We have created topology in mininet as shown in figure 4.our topology consists of one controller labelled as c0, one switch labelled as s1 and 3 hosts labelled as h1,h2,h3 and virtual links have been used to connect them. Mininet topology is shown in figure 4.

In our topology, with the help of POX controller, openflow switch s1 is made to behave like a firewall .Pox controller has been connected to openflow switch for this purpose .We have worked on two terminals. On first, firewall application has been run which denies or allows the traffic based on rules installed. On second, we have run mininet topology. SSH client has been used for making remote connections to virtual hosts for ease of experimentation.

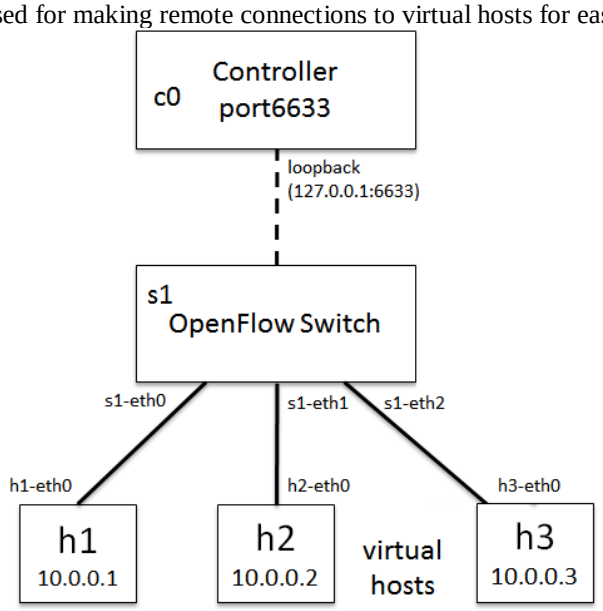

Fig. 4 Mininet Topology

Table I Application Table

S. No.	Application	Host	Action
1	YouTube	10.0.0.2	Deny
2	Facebook	10.0.0.0/8	Deny

III. FIREWALL DESIGN

A firewall is a device that keeps track of incoming and outgoing network traffic and then according to predefined security rules, it allows or rejects a specific type of information [11].It may be a service, a device or an application. Our firewall blocks various applications based on signatures .When packet is received at the switch, the openflow switch matches the information in the incoming packet with the entries in flow table. If there is a match, then corresponding action is applied to the packet. If there is no match, the complete packet is sent to the controller for further processing. Then it is checked that whether packet is IP packet or not because this firewall filters only IP packets. Then the next step is to check whether packet is from inside network or outside network. After that, two tables are maintained for filtering the traffic which is firewall table and application table. First of all, packet is matched against firewall table. If there is a match, then particular action will be applied to the packet. If there is no match, then packet is checked against application table which will contain the list of applications which are to be blocked.

IV. RESULTS AND DISCUSSIONS

This paper demonstrates that we can replace expensive hardware devices such as firewalls with SDN controller and simple openflow switches. In this implementation; we have managed to block various applications like Facebook and YouTube with the help of two different scenarios. In our first scenario, we have successfully blocked the access of Facebook application from the complete network i.e. from 10.0.0.0/8 and in second, YouTube application has been blocked from a single host h2 whose IP address is 10.0.0.2. These hosts can access any other website on the internet. This firewall works on the base of firewall table and application table where rules have been installed as shown in table 1.We have used NAT for translating addresses of these hosts for ease of implementation. With the help of command log. Level --DEBUG, logging level was set. We have combined the firewall module with switch module in this code and created a "MASTER" class which manages the order in which module functions will be called. This experiment has been done on three hosts by taking two applications into consideration. Figure 5, 6, 7 and 8 show the experimental results for this implementation.

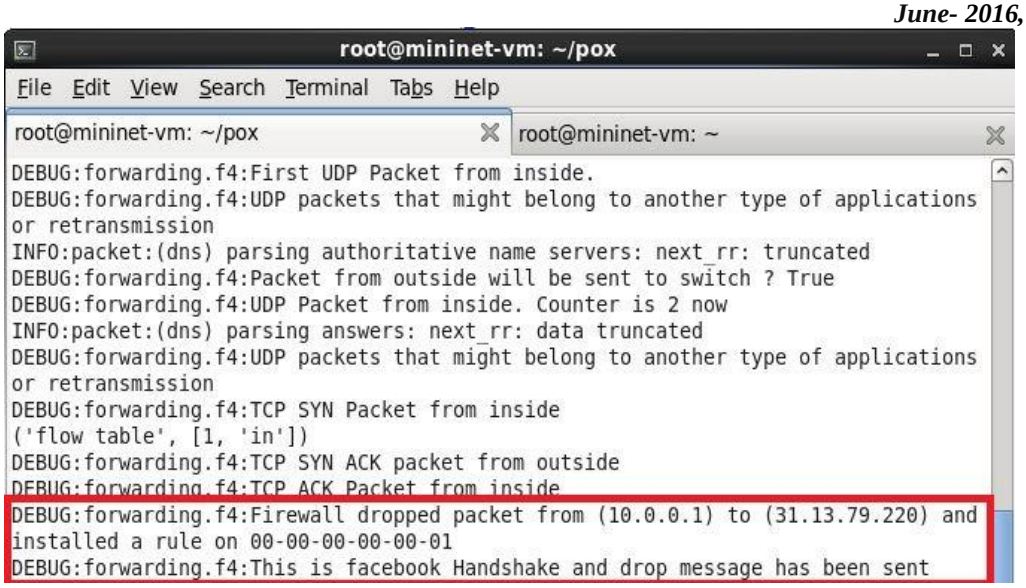

Fig. 5 Facebook blocked from host 1

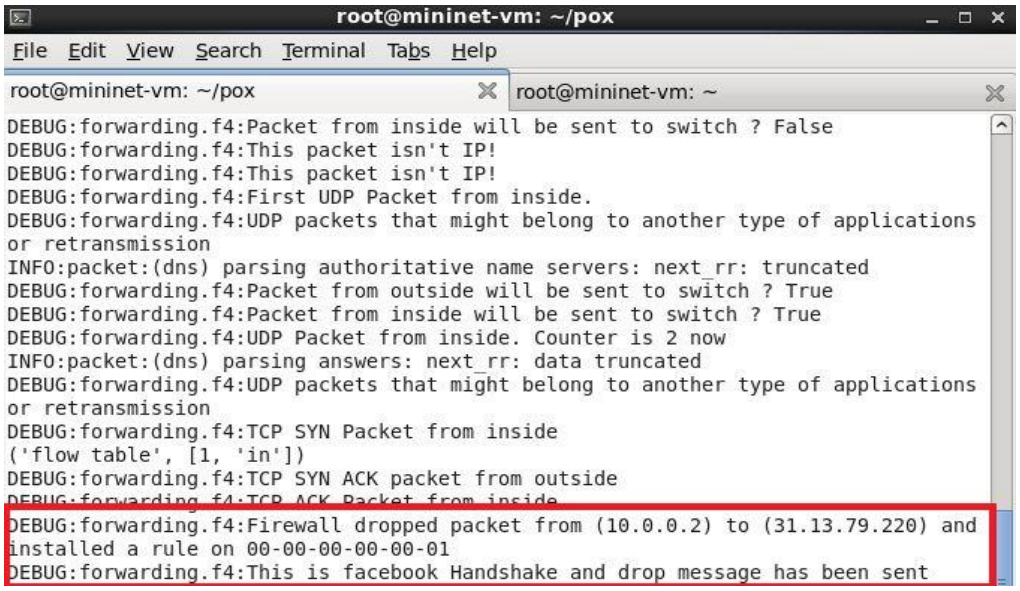

Fig. 6 Facebook blocked from host 2

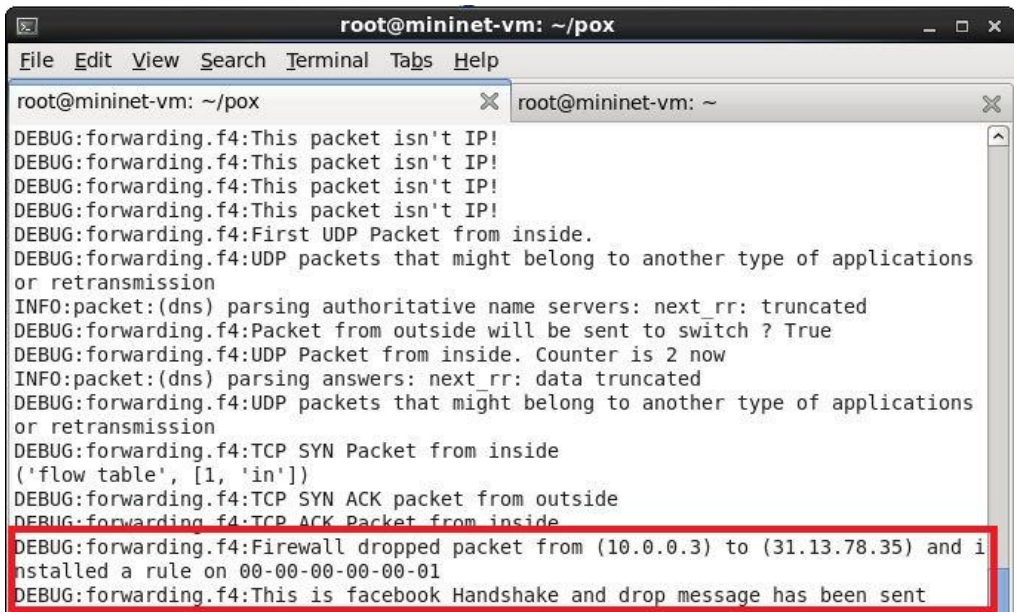

Fig.7 Facebook blocked from host 3

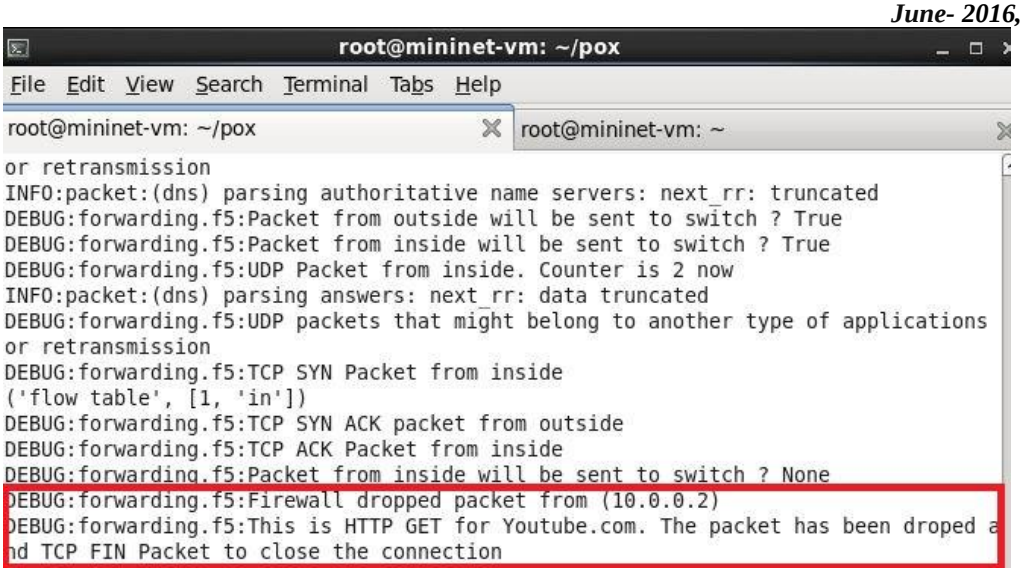

Fig.8 YouTube blocked from host 2

V. CONCLUSION

SDN is becoming very popular day by day. Even though there are outstanding firewalls present in the market but those firewalls are expensive and cannot be programmed so it becomes difficult for network administrators to modify or extend the functionalities of firewall [12]. On the other hand, SDN is cost efficient and provides programmability by separating control plane from data plane. We have successfully implemented an openflow based top layer web based filtering firewall which is capable of detecting and blocking websites such as Facebook and YouTube. The limitation of this firewall is that it is not able to detect encrypted traffic. Therefore, firewall policies can be easily bypassed by a user by running a VPN client. Future directions can be to add more security and support for multiple openflow switches. Also, future direction can be towards designing and implementing Intrusion detection system.

ACKNOWLEDGMENT

The authors would like to thank Sukhveer Kaur, Karamjeet Kaur and Mr. Vipin Gupta for their valuable guidance for this work

REFERENCES

[1] K. Bakshi, "*Considerations for software defined networking (SDN): Approaches and use cases*," in IEEE, 2013.
[2] F. M. V. R. Diego Kreutz , "*Software-defined networking: A comprehensive survey*," in Proceedings of the IEEE, Vol. 103, January 2015
[3] K. S. Kazuya Suzuki, "A survey on openflow technologies," IEICE TRANS.COMMUNICATION, VOL.E97-B, pp. 375–386, 2014
[4] A. K. Adrian Lara and B. Ramamurthy, *"Network innovation using openflow: A survey," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 16, pp. 493–512, 2014.*
[5] M. M. Bruno Astuto A. Nunes, "A survey of software-defined networking: Past, present, and future of programmable networks," IEEE COMMUNICATIONS SURVEYS & TUTORIALS, 2014.
[6] R. G. Alexander Gelberger, Niv Yemini, "*Performance analysis of software-defined networking (SDN)*," in IEEE 21st International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2013
[7] M. P. Fernandez, "*Comparing openflow controller paradigms scalability: Reactive and proactive*," in IEEE 27th International Conference on Advanced Information Networking and Applications, 2013
[8] B. L. Michelle Suh, Sae Hyong Park*, "Building firewall over the software-defined network controller," in ICACT, 2014.*
[9] W. H. Hongxin Hu, "*Flowguard: Building robust firewalls for software defined networks*," in HotSDN14, 2014.
[10] N. M. Bob Lantz, Brandon Heller, "A network in a laptop: Rapid prototyping for software-defined networks," Hotnets 10, 2010.
[11] J. S. Karamjeet Kaur, "*Programmable firewall using software defined networking*," in 2nd International Conference on Computing for Sustainable Global Development, 2015.
[12] A. R. Tariq Javid, Tehseen Riaz, "*A layer2 firewall for software defined network*," in Conference on Information Assurance and Cyber Security (CIACS),IEEE, 2014.

