IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 8 Issue 6, June 2021

ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72

www.ijiset.com

Analysis of Software defined Networking (SDN) based Firewall

Siddhesh Dashrath Deshmukh[1], Prof. Nitin Nagori[2], Prof. Amey Gawde[3]

[1]Department of Electronics and telecommunication, K.J. Somaiya college Of Engineering,
Mumbai, Maharashtra, India

[2]Department of Electronics and telecommunication, K.J. Somaiya college Of Engineering,
Mumbai, Maharashtra, India

[3]Department of Electronics and telecommunication, K.J. Somaiya college Of Engineering,
Mumbai, Maharashtra, India

Abstract

Software-Defined Networking (SDN) is a developing technology that will power the next generation of networks. The flexibility to introduce their networks is provided to network managers. However, it also carries with its new security concerns. The SDN offers network administrators with a concise description of the whole network topology. It decouples a network's control and forwarding systems, allowing physical and logical networks to be managed separately. This method makes it easier to programmatically and efficiently reallocate network traffic flows to meet growing demands. SDN allows networks to be entirely managed by software applications, allowing existing network infrastructures to be pushed to their limits. this study examines a firewall application that operates on an OpenFlow-based SDN controller to demonstrate that most firewall functions may be implemented in software.

Keywords: *Software Defined Networking (SDN), Firewall, Controller, OpenFlow.*

1. Introduction

Software Defined Networking (SDN) is a sort of new technology which can enable companies consumers operate on-demand apps by altering the network unlocking essential intelligence and delivering innovative feature and information to require run on-demand result as there was a lot of demand for SDN implementation in datacenters there is a program that prevents harmful components from entering as is known as firewalls are used in methods regulate errors and phases difference within communicators as it also serves traffic filter harmful are blocking the packets a upon some set can be created such IP addresses mac some domain names protocols ports or even specific words if a firewall isn’t installed thousands of devices could be to malicious firewalls should be implemented at every link path which is all together to public area in large companies use to control web traffic there is a program prevents harmful components from entering as is known as firewalls are used in methods regulate errors and phases difference within communicators as it also serves harmful are blocking the packets a set upon rules a firewall is a system that is designed to illegal activities taking place at both combinations as job about filtering in we can manage traffic inflow data a result the routes all that restrict traffic employs policies networks are also known as software-defined the same managing detection is being used to protect from threats be enabled by SDN loss other network issues dos in few new perform there are software-based servers automatic can control a variety of ways.

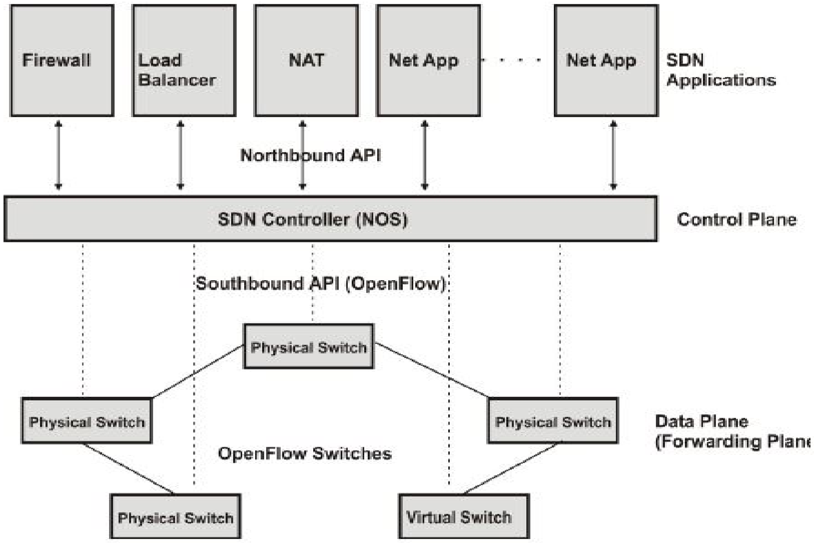

Figure 1 Basic SDN Architecture

The control plane may be accessed on software-based via the connected device using SDN this access allows it managers to control traffic more precisely from a centralized interface UI this site allows you to handle your network functioning and network setup more effectively in particular the ability to manage various networking configurations from a single UI quickly for network segmentation SDN has become an excellent substitute for the old networking system as it makes it possible to offer it manager resources and bandwidth without further physical infrastructure costs when necessary. The concepts of programmable decoupling of network data planes SDN is causing a radical shift in networking it ensures easier way pf installation of new services or upgrades. application is primarily intended to meet the needs of users end-user business apps leverage network makeup category can use the control plane to switching at layer network visualization dynamic control security mobility and migration cloud computing load the features firewalls main goal is to process packets based on their encoded headers then accept or reject them based on the laws we specify to function as an l2- learned switch sort it must be used in conjunction with the switching module responsible for implementing the OpenFlow switch pox is an open-source controller for SDN application development.

2. Background

Losing data or unrestricted access to sensitive data can prove very costly for small as well as large companies. Protecting data or restricting access requires use of proprietary, expensive and inflexible traditional network devices such as firewalls, IDS or IPS. But SDN is changing all this as presented in [1] SDN introduces network programmability in networks. SDN is based on the concept of decoupling of packet or frame forwarding functionality at the lower layer from the control logic that decides how to forward the traffic intelligently and efficiently. In traditional networks, each device must be configured individually. SDN allows centralized controller to dynamically manage all the devices.

In [2], authors proposed the architectural design of FORTRESS, a stateful firewall for SDN networks designed to run entirely within the data plane. There are two architectural models supported by FORTRESS: Stand-Alone and Cooperative. FORTRESS makes it possible to leverage the characteristics of a stateful firewall in the Stand-Alone Architecture, thereby entirely removing packets and the computational load from the controller. Performance was measured in terms of packets transferred between the control plane and data plane, comparing FORTRESS with Flow Tracker, the state-of-the-art stateful firewall for SDN.

Kaur [3] used Mininet emulator as testbed software and used POX Controller based on Python as our SDN Controller for developing distributed firewall which is able to handle ICMP, TCP and UDP traffic. SDN will reduce network management cost and enhance programmability, advancing the network to be easily configured and managed as presented in [4]. One of the examples where SDN is most useful is Data Center. Data Centre’s carry tens of thousands of virtual machines that migrate over the underlying topology. Instead of having to configure each switch in accordance to the new virtual network, they can employ programmable switches that the central database can control. OpenFlow protocol, a controller and a switch can communicate with each other as well as the controller can manage the switch over the secure channel.

Easy and flexible configuration and robust control on application placed SDN to a popular network technology in today's era. In [5], author has given the analysis of firewall technologies based on OpenFlow prototype, where firewall controller controls the traffic flow based on rule set define in flow table and switch that makes enforce on controller regulating traffic flow according to their flow table rules. But the firewall cannot able to cover all explained security region. Separation of control and data plane gives rise to new security challenges that will grow with development of SDN. Authors discussed about SDN security limitation to prevent the unauthorized control of switches, and a centralized server that suffer from a single point of failure or compromise.

In [6], author implemented firewall using two different approaches. single firewall or multi/parallel firewall. A single firewall approach contains just a single firewall controller on the network; the firewall is connected to the internal network and public internet. Rules and policies are implemented on the firewall to filter the incoming and outgoing traffic and secure the network from unauthorized access. Whereas multi/parallel firewall have two or more firewall controllers on the network. It divides the work based on the number of incoming and outgoing packets thus the rules and policies are implemented on both firewalls. redundancy has shown to be an important factor in improving a network's performance. There is always a clash between security and performance; one is compromised for the other.

In [7], author implemented an SDN based firewall with the help of mininet emulator and used Wireshark and jpref tools to provide testing and performance analysis. In order to evaluate the performance of firewall author did Ping utility to measure RRT (Response Time) and Jperf utility to measure bandwidth. The comparative results are shown in the form of graphs and table with and without using firewall. The limitation of firewall implemented is it does not keep record of the state of connection which makes it stateless. This limitation is due to using OpenFlow version 1.0 which does not keep track of the state of the packets as well as only few header fields are supported.

3. Related Works

Existing architectural firewall design in two ways first is a SDN controller used for building a mac table and second is an OpenFlow switch that acts as a firewall. Existing firewall approaches are explained in terms of their components and implementation.

3.1 Open Flow

OpenFlow is an opensource standard that uses an automatic operation on switches to manage traffic flow and provide an interface to instructing operations for separate controllers [9]. It also maintains a flow table that defines enforcement rules for traditional firewalls to restrict or accept packet flows, which is dependent on the controller.

3.2 Controller

Controller is required to implement the OpenFlow protocol for communication between controllers and switches in an effective manner. It may also be used to operate a variety of applications such as a switch, hub, and a firewall.

3.3 Mininet

Mininet is a simulation platform that can operate a variety of virtual hosts, controllers, switches, and connections. Container-based virtualization is used to allow a single device to serve as a whole network [10]. It's a simple, scalable, and low-cost network tool for developing and testing OpenFlow based applications. Mininet can create a complicated network topology without establishing the physical network for testing reasons. Custom topologies are supported. It has a simple and versatile Python API for creating and testing networks. Mininet has Controller classes built in to support various network controllers.

By using the 'mn' command, you can select a controller.

Table 1. Mininet Commands

Command	**Description**
mn	Run mininet
--topo single,5	Create 1 switch with 5 hosts
--xterm h1 h2	Open node h1 and h1 in different terminal window
--mac	Make mac address same as node number on host
--arp	Install static ARP entries
--switch ovsk	Use Open vSwitch
--controller remote	Use remote controller
--ip	Remote controller IP address

3.4 Implemented Firewall

Under traditional security model, internal users are considered trusted. Internal traffic is not inspected so as such not filtered by firewall. Insiders can very easily perform attacks. This problem can be easily solved with OpenFlow by placing firewall at any one place or at multiple places in a network as presented in [8]. An OpenFlow switch consists of one or more tables of packet handling rules. Each rule is matched against traffic and actions are performed on the traffic that matches a rule. Actions can be drop, forward, or flood. Depending on rules installed OpenFlow switch acts as a router, firewall, switch or load balancer.

Losing data or unrestricted access to sensitive data can prove very costly for small as well as large companies. Protecting data or restricting access requires use of proprietary, expensive and inflexible traditional network devices such as firewalls, IDS or IPS. But SDN is changing all this as presented in [1] SDN introduces network programmability in networks. SDN is based on the concept of decoupling of packet or frame forwarding functionality at the lower layer from the control logic that decides how to forward the traffic intelligently and efficiently.

In traditional networks, each device has to be configured individually. SDN allows centralized controller to dynamically manage all the devices. Kaur [3] used Mininet emulator as testbed software and also used POX Controller based on Python as our SDN Controller for developing distributed firewall which is able to handle ICMP, TCP and UDP traffic. SDN will reduce network management cost and enhance programmability, advancing the network to be easily configured and managed as presented in [4]. One of the examples where SDN is most useful is Data Center. Data Centers carry tens of thousands of virtual machines that migrate over the underlying topology. Instead of having to configure each switch in accordance to the new virtual network, they can employ programmable switches that the central database can control. OpenFlow protocol, a controller and a switch can communicate with each other as well as the controller can manage the switch over the secure channel.

In short focusing on implementing easily comprehensible and effective SDN firewalls and providing decent UI so that the largest users may maintain rules outside of the controller and enhance firewall restrictions for these implementations by blocking traffic with these implementations.

4. Issues and Challenges

Major component of future network technology is the separation of control and data plans as well as centralized monitoring and logical network infrastructure deployment however other security problems are being seen in the course of SDN deployment. SDN has some security restrictions in order to avoid unauthorized switch checks and a centralized server would suffer from a loss event or compromise source which means that SDN-based security services must be provided.

Network and operator services application plan for certain thread is vulnerable to lack of access control, authorization, authentication and false flow rules generation. control plan which provides centralised control and traffic flow management has a certain thread like dos, ddos attacks, man-in-the middle attacks, flooding attacks and TCP attacks when multiple threads make concurrent updates on firewall policy or flow policy in case of conflicting updates lack concurrency handling leads to low priority rules with a different action on same match being handled before the higher priority rule which leads to disregard for concurrent updates.

Challenges faced by firewall are, i) OpenFlow protocol actions by match in the flow rule can be chained i.e. if a flow packet matches a flow rule in a button the same packet can do several appropriate actions upon detecting a violation in the flow rule deleting all violations of and non-violation of a flow rule the measures specified under a rule can only violate the policy in part just as the whole flow rule might be violated in part.

ii) Different flow rules for identical header fields can create interdependence problems which may lead to interconnection between illegal hosts.

5. Conclusions

Firewall is an important network feature, which watches and regulates all network traffic and enables illegal activity to be detected and prevented. The installation of physical firewalls might also be costly for hardware and device upgrades. Firewall replacement is an important problem when the firewall is physically removed, and every firewall-related system configures to resolve problems. SDN is a revolution not only in making the firewalls programmability by separating the firewall hardware and the software control, but also for making it adaptable and controllable. Applications placed in the popular firewall technology today have easy and flexible configuration and robust control based on an OpenFlow prototype, which a fire wall controls traffic flow based on a flow table set by rule and switch that implements the traffic control system according to the flow table rules. In comparison, programming for administrators isn't flexible, but firewalls can accomplish the same things. For converting affordable vendor-specific silicon devices to firewall, load-balancer hub, switch, router or middlebox, controller can be used.

References

[1] Karamjeet Kaur, Sukhveer Kaur, Vipin Gupta, “Software Defined networking-based routing firewall”, IEEE International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT) 11-13 March 2016.

[2] Maurantonio Caprolu, Simone Raponi, and Roberto Di Pietro, “FORTRESS: An Efficient and Distributed Firewall for Stateful Data Plane SDN”, Hindawi Security and Communication Networks, Volume 2019, Article ID 6874592, 25 November 2018.

[3] Sukhveer Kaur, Karamjeet Kaur, Vipin Gupta, “Implementing Openflow Based Firewall”, IEEE International Conference on Information Technology (InCITe) - The Next Generation IT Summit on the Theme - Internet of Things: Connect your Worlds 6-7 Oct. 2016.

[4] Michelle Suh, Sae Hyong Park, Byungjoon Lee, Sunhee Yang, “Building firewall over the software-defined network controller”, 16th IEEE International Conference on Advanced Communication Technology 16-19 Feb. 2014.

[5] Dhaval Satasiya, Raviya Rupal, “Analysis of Software Defined Network Firewall (SDF)”, IEEE WiSPNET 2016 conference March 2016.

[6] Saad Waheed M., Mufarrej Al M., Sobhieh M., Barrak Al A., Baig A., Mazyad Al A., “Implementation of Virtual Firewall Function in SDN (Software Defined Networks)”, 9th IEEE-GCC Conference and Exhibition (GCCCE) May 2017.

[7] Ryhan Uddin, Md Fahad Monir, “Performance analysis of SDN based firewalls: POX vs ODL”, International Conferences on Advances in Electrical Engineering (ICAEE) 26-28 September 2019, Dhaka, Bangladesh.

[8] Karamjeet Kaur, Krishan Kumar, Japinder Singh, Navtej Singh Ghumman, “Programmable firewall using Software Defined Networking” 2nd IEEE International Conference on Computing for Sustainable Global Development (INDIACom) 11-13 March 2015.

[9] https://en.wikipedia.org/wiki/OpenFlow.

[10] https://en.wikipedia.org/wiki/Mininet.