Software Defined Network Firewall: A Survey

Niharika Pathania[1], Sudesh Kumar[2]

[1]Student, Department of Computer Science Engineering, Shri Mata Vaishno Devi University Katra, J&K
[2]Professor, Department of Computer Science Engineering, Shri Mata Vaishno Devi University Katra, J&K

Abstract -*Software-Defined Networking (SDN) is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable. Architecture of SDN disconnects the network control and forwarding functions so that the network control can be directly programmable and the elemental framework is withdrawn for applications and network services. The primary notion of SDN is to segregate the control layer and concentrate it to one single point of network, that is every last network device only needs to attend data layer and transfer data packets from one node to another which rely on forwarding decisions which are taken by SDN controllers. The protocol which is named as OpenFlow is the base of SDN which allows the disassociation of control and data plane, which causes extensive difficulties such as unauthorized entry, Distributed Denial of Services(DDoS), inconsistency in the OpenFlow policies in OpenFlow legalized switches and clashes in firewall strategies and controlling traffic. The above mentioned problems can be overcome by SDN directed firewall.*

***Key Words*: Software Defined Network (SDN), OpenFlow Protocol, POX, Firewalls.**

1. INTRODUCTION

Firewall is a framework primarily designed to secure network from unwanted access to and from a private network [1].Firewalls can be performed through hardware, software, or an aggregation of both. It basically furnishes an obstacle between a reliable intrinsic network and unreliable extrinsic network like Internet. Typically, firewall is classified as stateful and stateless based on packet filtering. Previously used stateless firewall has a drawback in filtering the packets as the firewall doesn't keep the tracks of traffic pattern as well as the data flow whereas the stateful firewall keeps the track of the traffic flow from beginning node to end and they also have the knowledge of connecting paths which are applied through various IP Security functions namely tunnels and encryption as showed in Fig.1 and Fig.2

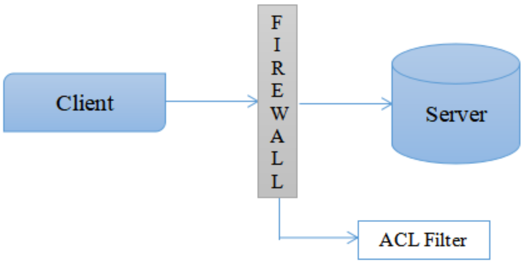

Figure- 1: Block Diagram of Stateless Firewall

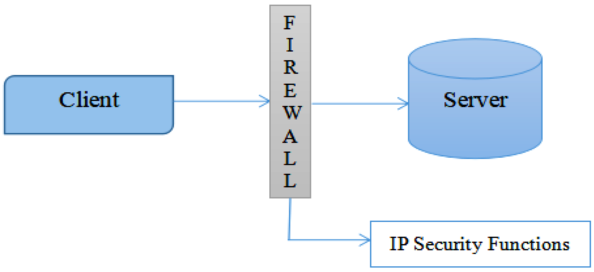

Figure- 2: Block Diagram of Stateful Firewall

1.1 Software Defined Network

By using software application, SDN's network architecture point of view becomes primarily manageable and discreet. Inspite of the fundamental network topology, the manager take care of the network continuously and thoroughly. SDN permits the programming of network conduct in a centrally controlled manner across software applications with the help of open APIs. Without hindering switches in the network, traffic can be filtered from centralized console and this all is in hands of the administrator of the SDN. Despite of the selected connections between server and devices, the centralized SDN controller handles the switches to transfer network services to wherever. Each network devices builds traffic rulings relying completely on the completed routing tables in tradition network design and SDN process is different from traditional one.

Firewalls are also becoming slower and more vulnerable to lags and firewall explosions as no if attacks are increasing nowadays. With Software Defined Networking (SDN) emerging rapidly, it has become difficult to implement network components including firewalls in traditional networks. Because of which upgrade is compulsory. Firewalls configured from one point to all nodes or router is very beneficial and also effortless, as one can administer, control traffic, shape and apply rules from one single point.

1.2 Traditional Versus Software Defined Network

SDN is a network based on software which is the main difference between itself and the traditional one. To build a network and perform better, the traditional network counts on switches and routers. Whereas in SDN, control plane handles the allocation of resources at a virtual layer. In the midst of collaborating with hardware to design new devices, the user is associating with software.

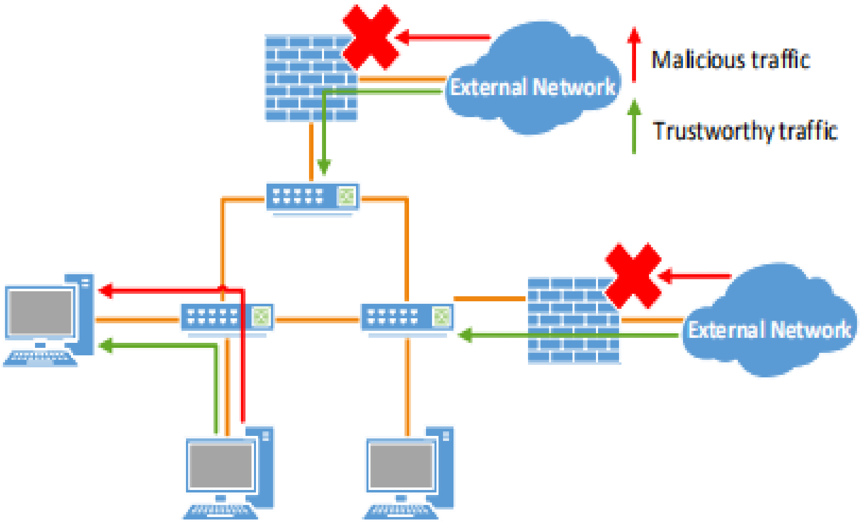

Fig-3: Traditional firewalls are unable to filter malicious flows of traffic between hosts as they are typically deployed at network boundaries. [2]

As shown in Fig3, firewalls are basically installed at the margin of the network by the network security infrastructure. And on this assumption, it concludes that only from outdoors of the network, unwanted traffic is appeared. Bring Your Own Device (BYOD) network which permits users to connect their own devices, means that this assumption is false [2]. TO perform actions of the flows of traffic such as allowing or not allowing traffic to pass or not respectively is the fundamental feature of SDN. At the points of assortment, traditional firewalls perform such actions in which one of them is to filter traffic. For packet filtering in a network, OpenFlow [3] has been proved up to the mark, as by all the arrangement of multiple switches this technology is not recognized and also not considered as how it can be filter traffic in a network [2].

Following are some differences between SDN and Traditional firewall:

- Firewall rules are centrally build and also implemented at the controller.

- Firstly, the packet goes through the controller and then after that it is filtered by SDN firewall.
- Traditional firewalls can't see the traffic inside a network and also can't filter it.
- Policy checker and packet filtering are both function-able on SDN based firewall.

The flow of consecutive packets directly harmonizes with the flow rules which are build in the controller.

1.3 Advantages of Software Defined Network over Traditional Network

There are several benefits of SDN in enterprise network as well as private network; some of them are as follows:

a) Network-Wide Intrusion Detection: The overall network view let the SDN controller to operate a network-wide Intrusion Detection System (IDS) that examines the traffic index to check for malicious traffic from all the network switches. It differs from a traditional network, in which IDS is installed on fixed part of the network and due to its limited visibility it provides fewer amounts of detection capabilities [4].

b) Detection of Malicious Switch Behavior: The universal network view not merely assist increased efficient detection of interference raised from infectious traffic, but also benefits in identifying network switch's destructive behavior[4].

c) Directly Programmable: SDN is programmed directly as the control functions are separated from forwarding functions which allows the network to be configured programmatically by open source automation tools or by proprietary [5].

d) Centralized Management: Network Intelligence is analytically determined to experience a universal aspect of the network, which appears to applications and policy engines as logical, single switches in SDN controller software [4].

e) Network Functions Virtualization (NFV): SDN and NFV are both self-sufficient, closely related, have common beneficial technologies and are interrelated. NFV controls transfers of network functions from devoted appliances to generic servers whereas SDN functions involves separation of data and control; integration of control and programmability of network. SDN software is operated by the framework given by NFV through which NFV supports SDN [5].

2. LITERATURE REVIEW

Analysis of sdn: On OpenFlow prototype the current firewall is based upon in which certain rules are defined in flow table that controls the traffic flow and switch ensures that flow of traffic must be according to the table rules. But all explained security region are not able to cover by the existing firewall. In future, there must be improvement in the existing stateful firewall.

Implementation of virtual firewall function in Software Defined Networking: In this paper, the main factor which has helped in fixing a network's performance is redundancy. Performance and security always clashes and every time one is compromised for other. To establish a secure and high performance network, firewall functions needs to be implement in software defined network controller. Using redundancy approach, the performance is improved even in the absence of firewall.

Building L2-L4 firewall: To implement hardware firewall is costly in traditional firewall. All the region of the network is not reckoned as the traditional firewalls could not be placed everywhere. Due to this, virus or attacks can affect our network. To block such type of attacks or unwanted traffic, they have applied L2-L4 firewall. As their firewall is software is software firewall which is low-priced than traditional one. Its advantage is it can be placed anywhere in the network.

Network wide virtual firewall using SDN/OpenFlow: For security reasons, OpenFlow issues needs for implementation of switches which are compatible with OpenFlow. As researches done before they were offered versatility granted by the technology in which they can use switch individually or as one but they didn't benefit from it. Keeping this in view, to protect the hosts in the network from both inside and outside threats, a new mechanism is presented in which it combines two approaches, also filtering process is done at different switches simultaneously.

Programmable firewall using Software Defined Network: Firewall application based on OpenFlow are designed and developed. Without using assigned hardware, these implementations prove that many of the firewall programming can be built using software. For experiments, an open source namely POX controller which is based upon Python is used. For virtualization solution VMPlayer is used and for building network topologies Mininet emulator is installed to perform experiments. Both implementation details and experimentation results of firewall applications are presented in this paper.

3. SOFTWARE DEFINED NETWORK FIREWALL

The application layer, the control layer and the infrastructure layer are the three layers which constitutes SDN architecture. The typical network applications or functions organizations use such as load balancing, firewalls or intrusion detection system, this all constitutes in the application layer. In contrast, specialized appliances is used in traditional firewall like load balancer or firewall, and here the appliance gets replaced by a SDN in which a controller is being by an application to handle data plane behavior. The three layers are distributed by SDN architecture, but keep connected by the APIs which are northbound and southbound.

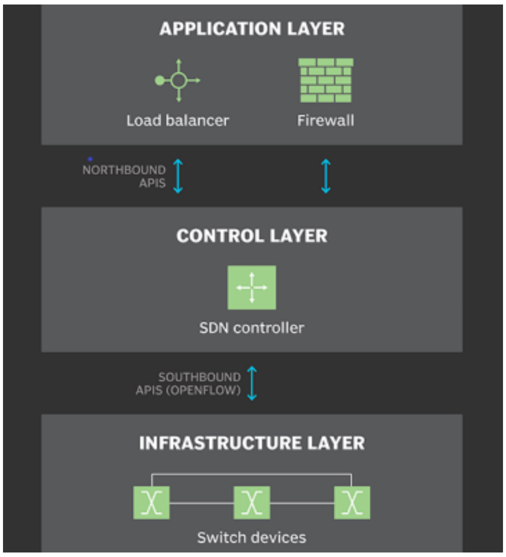

Fig-4: SDN Architecture

The mind of the SDN is said to be the control layer which portrays the centralized SDN controller. Course of traffic and policies are all managed from a server. Physical switches in the network comprises of the infrastructure layer. Using northbound and southbound APIs, these layers communicate with each other. For instance, switches and controller converse using southbound interface and whereas controller and applications talks across with northbound interface, like OpenFlow, though other protocols exist. As compared to OpenFlow as a globally southbound interface, there is hardly any approved standard for controller's northbound API. For productive standard in future OpenDaylight controller's northbound API will be used for its vast vendor support.

3.1 How SDN works

Technologies like functional separation, virtualization and also automation through programming are accepted by SDN. Separation of network control plane and data plane was the main focus of SDN technology. In the meantime, the travelling of data packets via network was resolved by the control plane and to transfer the packets from end to end was responsibility of data plane. At network switch a packet emerge and after that in switch's unshared firmware rules are built so that it knows the instruction where to direct the packet, this happens in traditional SDN's. From centralized controller, packet handling standards are transferred to switch. The data plane device is also known as switch asks for instructions from the controller and then controller passes information about managing the traffic. As the packets are moving towards their destination following the unvarying path, the switch labels it and all packets are in control in literally the same way.

An operation mode is consumed by SDN which is sometimes also called as dynamic or adaptive in here the packet which does not have a specific route is given a path by switch to a controller. Adaptive routing differs from this type of process, as in it by means of algorithms and routers build on the network topology are given route requests. Through virtual overlay, the virtualization aspect of SDN runs, it is rationally different network above the physical network. To withdraw the fundamental network and split the network traffic, users can carry out start-to-end overlays. For operators with multiple user cloud environment, services of cloud and service suppliers the separation of fundamental network and network traffic is beneficiary, as they supply diverse virtual network with respective policies for each user.

The basic firewall for SDN was initiated by POX [6]. Improving commands of layer 2 in POX and creating easy friendly user interface was the main purpose of installing this firewall. To uncover header fields of traffic, the execution of this firewall is mandatory [7]. There are two ways in the already present firewall architecture, the first way is to build MAC table and manage rules to filter the traffic and this is applied on POX (SDN controller) and the other is OpenFlow switches which acts a firewall [8].

A. POX

SDN applications are developed by POX which is Python Programmed open source controller. For communication, between controller and switches, POX controller is required for orderly manner to execute OpenFlow protocol. To execute different applications such as switch, load balancer, hub and firewall. Using OpenFlow protocols, POX controller communicates with switches [6]. In the firewall system, POX operates two applications; first one is to make a MAC table in which it copies address to ports, and the second one is to implement a firewall which refines the traffic [7].

B. OpenFlow

OpenFlow which is an open standard protocol operates automatic operation on OpenFlow activated switches to issue user interface to guide functioning for different controllers and to handle traffic [9]. It records the flow of table which includes rules for the firewall on how to filter traffic as being dependent on the controller [7].

4. FIREWALL

Whether, the data packets need to be allowed or should be dropped from the computer network, this is decided by a firewall which has certain set of rules. Its sole motive is to lessen the threat that malicious packets moving over the open internet will obviously have an effect on the security of private network and to clean the traffic. [10].

4.1 Types of Firewalls:

To keep a network secure, a firewall can be both software and hardware based. Its sole purpose is to prohibit unwanted entry to and from the network [11]. It mainly builds a security wall or an overpass between two networks in which one is not secure and can't be trusted which is Internet and another one is secured as well as trusted[12].

i. Network Layer Firewalls: On the base of ports, source address, destination address, it makes their decision in individual packets. Traditional network layer firewall is implemented in a simple router which can't make decisions of what contents are present in the packet and also its source and destination address. You either need an assigned IP address block or a private internet address block so that the traffic can pass through the network layer firewall, this is the main difference. Also network layer firewalls possess to be fast and almost transplant to their users [12] [10].

ii. Application Layer Firewalls: These are hosts which run proxy servers whose function is to restrict traffic to pass between the networks, and examine the traffic which passes through them and perform elaborate logging. As proxy applications are software based firewall, so it is possible to perform actions such as access control and logging from one place. These firewalls can also be used as translators of network address as the entry and exit of the traffic are different after passing through an application that covers beginning of the launching connection[12][10].

5. SOFTWARE DEFINED NETWORK PRACTICAL APPLICATIONS

a) Load Balancing:

Among many resources of the same types, load is assigned which is basically the responsibility of a load balancer. In physical equipment or in software, load balancing is installed. Its features are minimizing resource utilization, boosting measurability, escalated production, restricting overload of any solo asset [13].

b) Energy efficient network:

By applying on different units of the SDN architecture or in the SDN itself energy optimization can be imposed. It can be possible by hardware based alterations or by algorithmic rule. Also, the hardware based outcomes are only accomplished on the forwarding switches [14].

c) Cross-Layer Design:

To amp up arrangement of units at different layers in a layered architecture, cross-layer approach is highly recommended. For instance, in OSI reference model information is exchanged among each other by authorizing entities at various layers. Cross-layer technique can be effortlessly developed on the platform offered by SDN in which application can freely access network status information [15].

d) Adaptive Routing:

Routing and packet switching are the main functions of a network. On distributed conceptualization, routing and switching designs are founded for strength. Despite this slow convergence, multiplex implementations are the disadvantages of distributed designs [16]. Enclosed loop control, allowing applications to sharply manage a network, providing applications with updated worldwide network status information, these all are offered by SDN. For improved routing designs, several prepositions have been made to apply on the SDN platform [15]

6. COMPARISON TABLE

S.NO.	AUTHOR NAME	YEAR OF PUBLICATION	PROTOCOLS/TOOLS USED	DESCRIPTION
1	Amandeep Kour, Vikramjeet Singh	June 2017	Mininet simulator, OpenFlow protocol, POX controller	With the help of POX controller, traffic is obstructed on the base of MAC addresses. Instead of web traffic, ICMP traffic is used. For easy configuration of firewall module, GUI is designed.
2	Salaheddine Zerkane,	May 2016	OpenFlow protocol, Stateful Firewall, Orchestrator	To lessen some Denial of Service (DoS) attacks like SYN flooding, it uses a receptive

	David Espes, Philippe Le Parc , Frederic Cuppens			performance to minimize the amount of OpenFlow rules in the dataplane devices. To eliminate valueless traffic not interrelated to their transition's conditions, the firewall processes the Finite State Machine of network protocols.
3	M. Saad Waheed, A. Baig	May 2017	Floodlight, Open vSwitches, Linux, Mininet	Two topologies are used, the initial topology runs a network with a single controller configured with firewall rules and policies and the Second topology contains multiple/parallel controllers running in conjunction with each other. Three tests were run on them: response time, UDP traffic. UDP with and without firewall.
4	Sukhveer Kaur, Japinder Singh	August 2014	POX, OpenFlow, Mininet	Mininet can create complex network topology for testing purposes, without configuring physical networks. Various applications can be created using POX but in this they had created HUB application in which single switch and 5 hosts topology is used.

7. CONCLUSION

Though Firewall has its own limitations, it's a healthy and safe procedure for the betterment of networking in various countries round the globe. Not only our security is increased, but its various problems are solved due to it thus resulting in less preaching and disloyalty. Irrespective of the fact that with new technology new threats are released but dealing with that threats and breaking it down to the very core thus ensuring integrity is the main reason to build a firewall and secure our network.

REFERENCES

1) Richa Sharma, Chandresh Parekh ,"Firewalls: A Study and Its Classification," International Journal of Advanced Research in Computer Science, Volume 8, No. 4, May – June 2017.

2) Jarrod N. Bakker, Ian Welch, Winston K.G. Seah, "Network-wide Virtual Firewall using SDN/OpenFlow", IEEE Conference on Network Function Virtualization and Software Defined Networks, November 2016.

3) N.McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008

4) Mehiar Dabbagh, Bechir Hamdaoui, Mohsen Guizani† and Ammar Rayes, “Software-Defined Networking Security: Pros and Cons”, IEEE Communications Magazine (Volume 53 , Issue: 6 , June 2015).

5) Study Paper on Software Defined Networking (SDN) as a tool for energy efficiency approaches in Information and communication technology (ICT) networks.

6) Sukhveer Kaur, Japinder Singh and Navtej Singh Ghumman, “Network Programmability Using POX Controller”, 2014 ICCCS pp. 134-138.

7) Mr. Dhaval Satasiya, Raviya Rupal D., “Analysis of Software Defined Network Firewall (SDF)”, IEEE 2016 WiSPNET Conference.

8) Karamjeet Kaur, Krishan Kumar, Japinder Singh, Navtej Singh Ghumman “Programmable Firewall Using Software Defined Networking” 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) (978- 9- 3805-4416-8/15) IEEE pp. 2125-2129.

9) Kazuya Suzuki, Nobuyaki Tomizawa, Yutaka Yakuwa, Terutaka Uchida, Yuta Higuchi, Toshio Tonouchi, Hideyuki Shimonishi, “A Survey on OpenFlow Technology”, 2014 IEICE pp. 375-386.

10) Gunjan Katwal, Manu Sood, “A Comparative Study of Traditional Network Firewalls and SDN Firewalls” International Journal of Latest Trends in Engineering and Technology (IJLTET).

11) http://sankar-information-security.blogspot.com/2012/09/types-of-firewalls-and-their-functions.html

12) https://searchsecurity.techtarget.com/definition/firewall.

13) Ali Akbar Neghabi, Nima Jafari Navimipour, Mehdi Hosseinzadeh*, Ali Rezaee, “Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature”, Article in IEEE Access March 2018, DOI: 10.1109/ACCESS.2018.2805842.

14) Beakel Gizachew Assefa and Oznur Ozkasap, “State-of-the-art Energy Effificiency Approaches in Software Defifined Networking”, ICN 2015: The Fourteenth International Conference on Networks.

15) Wenfeng Xia, Yonggang Wen, Senior Member, IEEE, Chuan Heng Foh, Senior Member, IEEE, Dusit Niyato, Member, IEEE, and Haiyong Xie, Member, IEEE, “A Survey on Software-Defifined Networking”, IEEE COMMUNICATION SURVEYS & TUTORIALS, VOL. 17, NO. 1, FIRST QUARTER 2015.

16) J. Fu, P. Sjödin, and G. Karlsson, “Intra-domain routing convergence with centralized control,” Comput. Network, vol. 53, no. 18, pp. 2985–2996, Dec. 2009.