JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A SURVEY ON SDN (SOFTWARE- DEFINED NETWORK) FIREWALL

Niharika Pathania, Jammu, UT of Jammu and Kashmir, India

Abstract: SDN is an emerging architecture that is dynamic, manageable, cost- effective and adaptable to this new generation of networks. The primary notion of SDN is to segregate the control layer and concentrate it on one single point of the network, that is every network device only needs to attend the data layer and transfer data packets from one node to another which rely on forwarding decisions taken by SDN controllers. In order to accommodate increasing demands, this technique facilitates programmability and effectively reallocates network traffic flows. The OpenFlow protocol, which enables the separation of the control and data planes, is the cornerstone of SDN. This leads to several serious issues, including Distributed Denial of Services (DDoS), unauthorized access, inconsistent OpenFlow policies in OpenFlow-approved switches, conflicts with firewall tactics, and traffic control issues. SDN directed firewalls can solve the issues listed above.

IndexTerms **– SDN, OpenFlow, DDos, Firewall**

I. INTRODUCTION

Firewall is a framework primarily designed to secure a network from unwanted access to and from a private network [1]. Hardware, software, or a combination of the two can be used to operate firewalls. It provides a barrier between an untrustworthy extrinsic network such as the Internet and a trustworthy intrinsic network. Based on packet filtering, the firewall is typically categorized as stateful or stateless. The stateful firewall, as illustrated in Fig. 2, keeps track of the traffic flow from beginning to end and has knowledge of connecting paths that are applied through various IP Security functions, such as tunnels and encryption. In contrast, the previously used stateless firewall as shown in Fig 1, has a drawback in filtering the packets because the firewall does not keep the track of traffic patterns as well as the data flow.

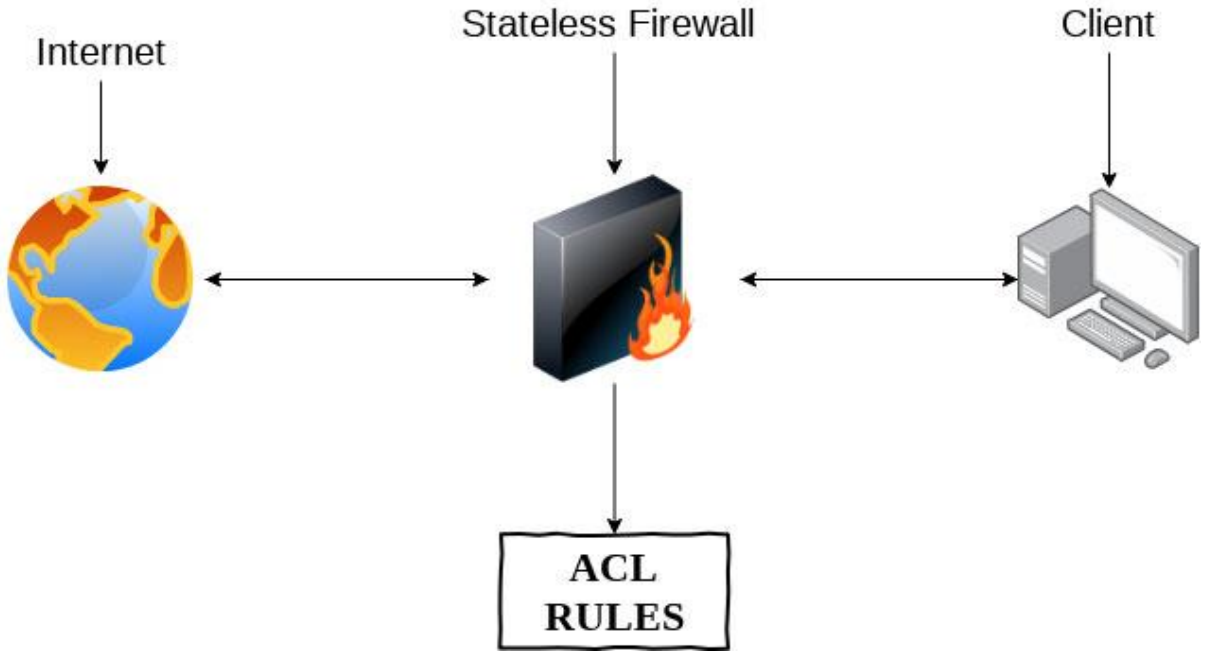

Fig 1. Diagram of Stateless Firewall

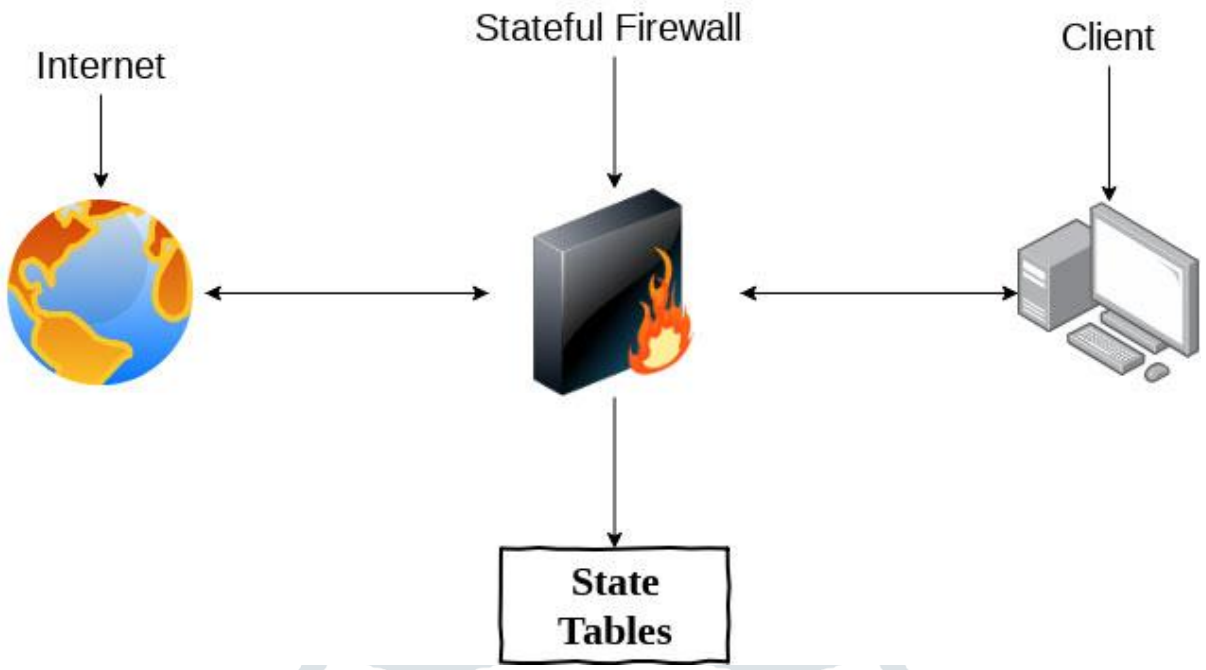

Fig 2. Diagram of Stateful Firewall

Stateful Firewall:

In the Open Systems Interconnection (OSI) model, Layers 3 and 4 are where this firewall is located. A stateful firewall, as its name implies, continuously monitors the status of network connections. A stateful firewall adds a kind of traffic to its state table after it has been authorized. Transmission Control Protocol (TCP) streams and User Datagram Protocol (UDP) datagrams that are allowed to flow through the firewall are documented in the status tables in accordance with the security policy that has been put in place. The connection gets deleted from the status table if there is not any traffic for a predetermined amount of time (depending on implementation).

Pros:

i. Stateful firewalls are quite good at spotting fraudulent communications and unwanted access.
ii. These firewalls can remember important details about network connections, thanks to their robust memory.
iii. A system such as this possesses intelligence. According to previous and current results, they determine how to filter in the future. If it detects a cyberattack in the future, it will automatically cease it and will not need updating.
iv. These firewalls only require a few ports to be open for proper communication.

Cons:

i. Stateful firewalls may be targeted because of distributed denial-of-service attacks (DDoS).
ii. Updates to the most recent software versions are required for these firewalls; if not, vulnerabilities might allow hackers to take over the firewall.
iii. Even something as basic as browsing a webpage might trick them into permitting a dangerous connection to the network.
iv. These firewalls may be more susceptible to man-in-the-middle (MITM) attacks, in which an outsider listens in on a discussion between both the parties with the intention of eavesdropping or manipulating the data.

Stateless Firewall:

It does not retain connection status data and is also referred to as an access control list (ACL). To determine the origin and destination port numbers, stateless ACLs can be applied to the network, physical, and occasionally the transport layers. Before dumping or rejecting a packet that has been routed via a firewall and filtered, the device checks to see whether it meets any of the firewall's pre-set ACL rules.

Pros:

i. Because they don't consider as many variables as stateful firewalls, stateless firewalls are sometimes perceived as being less strict. This is why they move quickly.
ii. Because it does not go too deep, it performs rather well in high traffic.
iii. Their cost is often lower than that of stateful firewalls.

Cons:

i. Because it cannot assess each packet or piece of network data, a stateless firewall cannot identify the kind of traffic. It is not as secure as stateful firewalls as a result.
ii. Depending on the circumstances, these firewalls may need to be carefully configured by a person who is familiar with the many kinds of network traffic and attacks. To achieve this, additional time and effort is may be required.

NOTE: Stateful and stateless firewalls are not mutually exclusive.

Traditional Network

Within proprietary hardware, the control plane and data plane are connected in conventional networks, as seen in Fig 3. Dedicated appliances, which might be one or more switches, routers, or application delivery controllers, are the primary devices through which network functionality is executed [2]. Use of Application Specific Integrated Circuits, or ASICs, is commonplace in this device, as the majority of its functionality is only realized in specialized hardware [3].

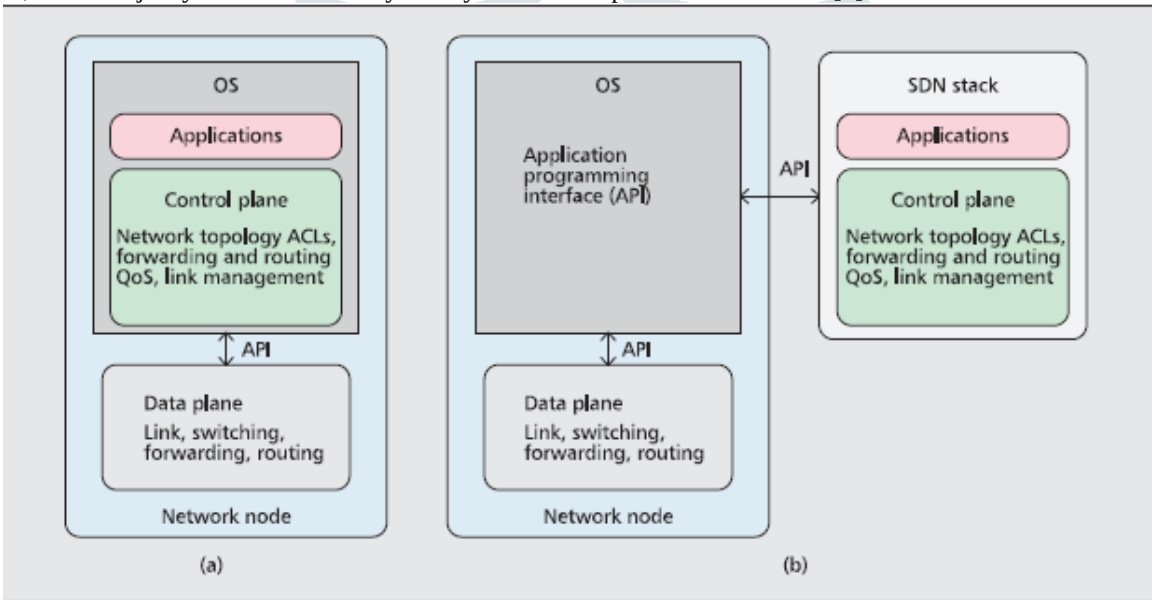

Fig 3. (a) Traditional network. (b) Software- defined network

Limitations of Traditional Networks:

i. **Setting up the network required a lot of effort and was inconsistent:** An IT administrator must take multiple steps in a typical network to add or remove a single device. Manual configuration of switches, routers, firewalls, and other network equipment is the initial stage. Next, he must use device-level administration tools to update several configuration settings, including as ACLs, VLANs, and Quality of Service. For an administrator, deploying a consistent set of policies becomes much more difficult using this setup method [2].

ii. **Various vendors:** Since conventional networks comprise a multitude of physical devices, this also implies a variety of suppliers, necessitating a high degree of proficiency and in-depth familiarity with every device on the network [3].

iii. **Distributed control plane**: Since the data plane and control plane in network devices are coupled, the intelligence of the network is in numerous locations in conventional networks instead of only in the control plane. For a network administrator, managing the network becomes quite challenging because of the somewhat complicated setup [2].

Traditional Networking	**Software Defined Networking**
▪ Static and non-adaptable networks.	▪ They are agile and flexible.
▪ Logical distribution characterizes these networks.	▪ They are located in one place.

▪ They operate according to protocols.	▪ Various APIs (Application programming interfaces) are utilized, including Southbound and Northbound APIs.
▪ There is utilization of FPGAs and custom ASICs.	▪ It makes use of merchant silicon.
▪ One device handle both high-level routing and packet forwarding.	▪ The data path and control path are divided in this OpenFlow switch.
▪ Command line is used to configure the switches utilized in it.	▪ The SDN controller provides an interface for programming switches by utilizing OpenFlow.

Table 1 Traditional and Software-Defined Networking Comparative Analysis

Software Defined Network

Software Defined Networking or SDN, is a concept that allows networks to be intelligently and centrally controlled, or "programmed," using software applications. Regardless of the basic network structure, this aids operators in efficiently and thoroughly managing the whole network.

Software programs with open APIs can be used to centrally manage network behavior using SDN. By opening previously closed network platforms and establishing a standard SDN control layer, operators can safely monitor any network and its devices, regardless of how complex the underlying network technology is.

SDN makes network virtualization more intriguing since it allows each tenant's control logic to run on a controller rather than physical switches. The abstraction of higher-level functions is a method of computer networking that gives network managers the ability to control network services.

SDN networks make network operations more programmable, flexible, and straightforward. Without any physical wire modifications, traffic can be redirected, modified, or personalized. Through unified control and traffic management, an SDN promises more automated, adaptable, and mountable network security.

Benefits of SDN over Conventional Network

Private and corporate networks can benefit from several SDN features, some of which are listed below:

i. **Network wide intrusion detection:** Because the full network view allows the SDN controller to search the traffic index for malicious traffic coming from every network switch, the Intrusion Detection System (IDS) may be run network-wide. It differs from a traditional network, in which IDS is installed on a fixed part of the network, and due to its limited visibility, it provides a fewer amount of detection capabilities [5].

ii. **Detection of malicious switch behavior:** The universal network views not merely assist increased efficient detection of interference raised from infectious traffic, but also benefit in identifying network switch's destructive behavior [5].

iii. **Directly programmable:** Network configuration can be done programmatically using open-source automation tools or commercial software since SDN allows control functions to be coded apart from forwarding functions [6].

iv. **Centralized management:** In SDN controller software, intelligence is analytically found to experience a universal component of the network, which manifests as logical, single switches to applications and policy engines [5].

v. **Network Functions Virtualization (NFV**): Both SDN and NFV are independent, tightly connected, share advantageous technology, and are connected. While SDN tasks include separating data and control, or integrating control and network programmability, NFV manages the transfers of network functions from dedicated appliances to generic servers. The framework provided by NFV, which enables NFV to support SDN, manages SDN software [6].

SDN Architecture

The main goal of SDN design is to enable network managers to utilize a controller based on software to oversee and govern the whole network. This goal is achieved by simplifying networking services through the separation of the data plane and control plane [4].

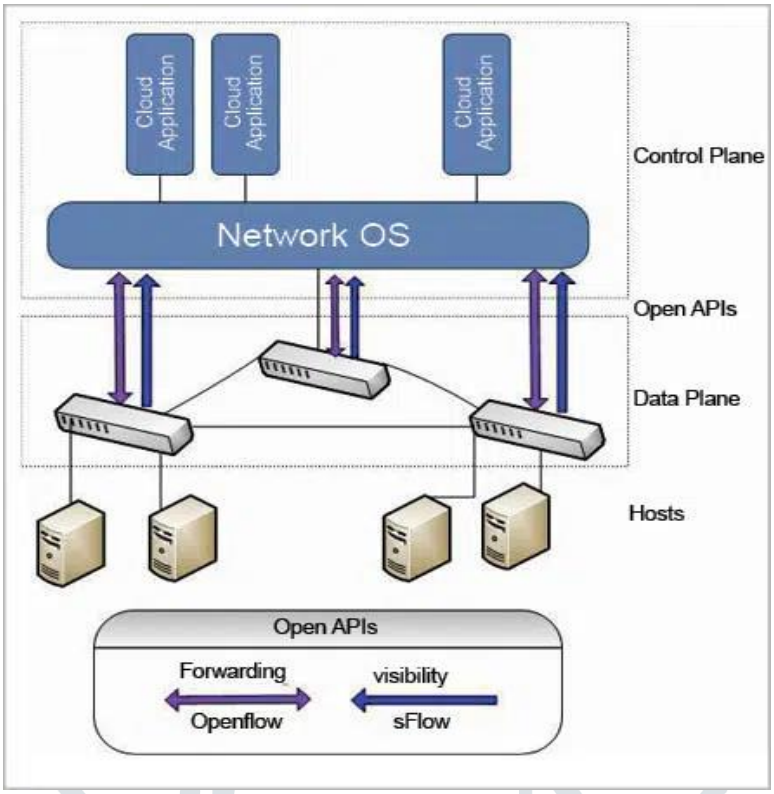

Fig 4. SDN Architecture

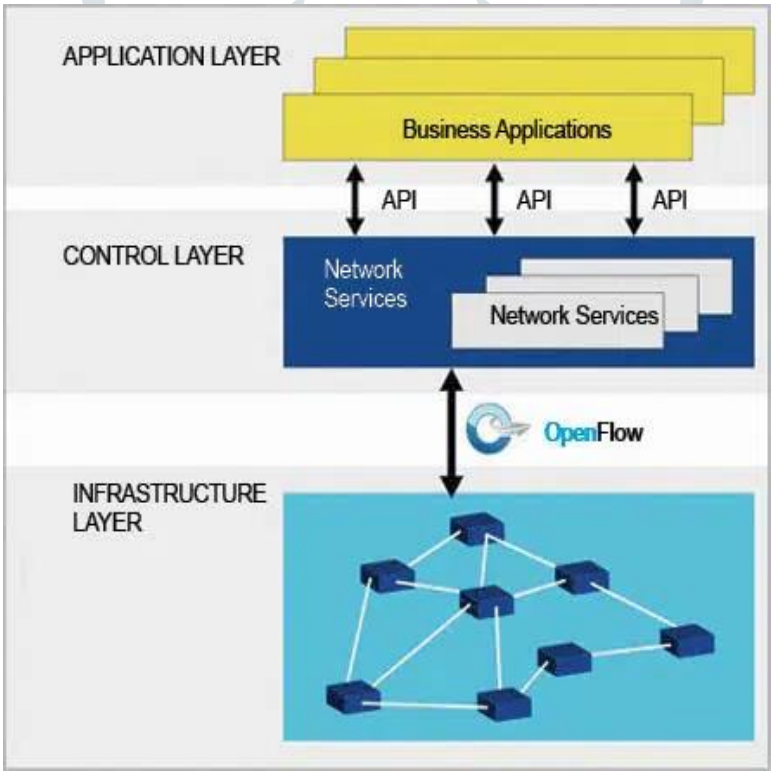

Fig 5. SDN layer architecture

Traditional Firewall VS Software Defined Firewall

i. A traditional firewall cannot view internal traffic and is unable to restrict it.
ii. A firewall based on SDN serves as a policy checker in addition to a packet filter.
iii. The SDN firewall filters the first packet after it passes through the controller.
iv. At the controller, the firewall policy is centrally developed and managed.
v. The subsequent packets in the flow instantly match the flow rules defined in the controller.

Literature Review:

A. A Review Paper on Software Defined Networking

The author intended to convey that the dynamic traffic management in networks enabled by SDN technology gives users access to additional bandwidth. Eliminating the need for specialist equipment is also a cost-effective approach. A simplified perspective of the network is shown. SDN is seen to be the greatest option for addressing the changing networking requirements. Research is continually being done to increase the networking efficiency of SDN because it is still a relatively young technology. It is hoped that academics working in this area would find this introduction to SDN, its architecture, and the controllers discussed here useful.

B. Analysis of Software defined Networking (SDN) based Firewall

Author is saying that an essential network component is a firewall, which monitors and controls all network traffic and makes it possible to identify and stop illicit activities. Costs associated with hardware and device updates may also extend to the installation of physical firewalls. When a firewall is physically removed, replacing it becomes a major issue, and every system connected to a firewall is configured to fix issues. By separating the hardware and software control of the firewall, SDN revolutionizes not only the programmability of firewalls but also their adaptability and controllability. The OpenFlow prototype—which employs a fire wall to regulate traffic flow based on a rule-set flow table and a switch to implement a traffic control system in accordance with the rules defined in the flow table—is responsible for today's commonly used firewall technologies, which enable robust monitoring and simple yet flexible configuration. In contrast, firewalls can do the same tasks as administrators, but administrator programming is not as versatile. A controller can be used to transform inexpensive silicon devices from a vendor into a firewall, load-balancer hub, switch, router, or middlebox.

C. Programmable firewall using Software Defined Network:

Firewall applications based on OpenFlow are designed and developed. Without using assigned hardware, these implementations prove that many of the firewall programming can be built using this 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.

D. Software- defined Networking: Challenges and Research opportunities for Future Internet

We have reviewed many current and cutting-edge SDN initiatives in this article. Based on survey results, we classified main difficulties and prospects along many dimensions, including architectural models, programmability, convergence, wireless and mobility, cloud platforms, and security. We showed that while the networking community is actively working on SDN research, the majority of studies still focus on topics such as call graphs, networking models, scalability of solutions, distributed vs. centralized control plane, control plane/data plane, etc [7].

E. Software-Defined Networking: An Evolving Network Architecture- Programmability and Security Perspective

The research highlights the assaults and how they affect the dynamic network architecture, pointing out that the SDN architecture is vulnerable to a range of attacks that are similar to those that target older networking design. This once more puts the research at the outset of the issue, namely that the emerging SDN architecture is susceptible to the same attack vectors that the conventional network design is. According to careful literature research and analysis, the answer is regrettably yes up until the first point, at which time the sentence "exposed to similar threats" is verified. Nevertheless, this does not imply that the entire field of study on evolving network architectures must return to its beginnings since, despite the fact that both of these architectures are vulnerable to comparable attack vectors, SDN consistently has the upper hand in thwarting these attacks. SDN's benefit adding to its programmability and interoperability capabilities is its decoupled design. If the above-mentioned and suggested attacks are not the focus of adequate security mitigation techniques, this growing network architecture will have a drawback. In order to wrap up, additional study needs be done to determine how assaults function inside each attack vector that targets different planes and to provide practical mitigation measures. [8].

SDN Firewall

By preventing unwanted network traffic produced by viruses and worms, a firewall serves as a barrier to safeguard computers connected to a network [4]. To develop an enterprise security policy that controls network traffic, hardware, software, or a mix of the two may be utilized [9]. To manage both incoming and outgoing network traffic, each firewall makes use of a database containing policy rules. [10]. Throughout the TCP/IP protocol stack, firewalls can operate as filters at various levels. As a result, firewalls fall into three categories: application-level firewalls, stateful firewalls, and static packet filters [11].

A stateless firewall, sometimes referred to as a packet filtering-based firewall, operates by accepting or rejecting packets according to their port numbers, source or destination addresses, or both. The firewall does not preserve the status of a packet to be utilized for processing subsequent packets in the same flow; instead, each packet is kept independently [12]. Stateful firewalls, on the other hand, keep track of network connection status while filtering packets. They frequently save data in a database called the state table that contains details about every traffic flow that goes through them. Entries in this state database correspond to the connection session that is presently in use [13]. These firewalls are widely used due to their low cost, ease of usage and upkeep, and high throughput. Nevertheless, since these stateless and stateful firewall filter packets based on source and destination addresses and may also look at UDP/TCP port numbers and flags, they do not require a great deal of knowledge about the traffic they are examining. They lack the intelligence to see the significance of preventing network breaches and assaults [11].

The basic firewall for SDN was initiated by POX [14]. Improving commands of layer 2 in POX and creating an 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 [15]. The current firewall design consists of two methods: the first uses OpenFlow switches that function as firewalls and builds a MAC table and rules to filter traffic. This method is used on POX (the SDN controller) [16].

I. POX

SDN applications are developed by POX, which is Python-Programmed open-source controller. It was created by NOX Controller and carried over. For communication, between controller and switches, the POX controller is required in an orderly manner to execute the OpenFlow protocol. To execute different applications such as a switch, load balancer, hub, and firewall. Using OpenFlow protocols, the POX controller communicates with switches [14]. Two programs run by POX in the firewall system are creating a MAC table, which transfers addresses to ports, and implementing a firewall, which filters traffic [15].

II. OpenFlow

Within the Southbound SDN system, OpenFlow [17] is a commonly utilized protocol. OpenFlow is a standard communication protocol defined by the Open Networks Foundation (ONF) that describes how switches with OpenFlow enabled interact with the OpenFlow controller. To put it broadly, OpenFlow configures various messages that allow the remote controller to add, remove, or change flow table entries in these switches to regulate their behaviour [11].

III. Firewall

Whether the data packets need to be allowed or should be dropped from the computer network, is decided by a firewall that has a certain set of rules. Its sole motive is to lessen the threat that malicious packets moving over the open internet will influence the security of private networks and to clean the traffic [18].

i. **Network Layer Firewalls:** It decides in each packet based on the ports, source address, and destination address. A basic router is used to build the classic network layer firewall; this router is unable to determine the contents of a packet or its source and destination addresses. There are two primary types of internet address blocks that are required for traffic to flow past a network layer firewall: assigned IP address blocks and private internet address blocks. Network layer firewalls are also quick and nearly invisible to their users [19][18].

ii. **Application Layer Firewalls:** These servers act as proxy servers on these hosts, limiting the amount of traffic that may flow through them, inspecting that traffic, and taking detailed logs. As proxy applications are software-based firewalls, 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 the beginning of the launching connection [19][18].

Conclusion

Despite its drawbacks, firewalling is a safe and healthy practice that has improved networking in many nations worldwide. Not only does this improve our security, but it also solves several other issues, which reduces proselytizing and disloyalty. Notwithstanding the fact, that new technologies bring with them new risks, the primary motivation for constructing a firewall and safeguarding our network is to address these threats head-on, and eliminate them completely to ensure integrity. SDN is seen to be the greatest option for addressing the changing networking requirements. Since SDN is still a relatively new technology, research is ongoing to improve its networking efficiency.

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. Wickboldt, Juliano Araujo, Wanderson Paim de Jesus, Pedro Heleno Isolani, Cristiano Bonato Both, Juergen Rochol, and Lisandro Zambenedetti Granville “Software-Defined Networking: Management Requirements and Challenges”, IEEE Communications Magazine, January 2015.

3. Sumit Badotra, Japinder Singh, “A Review Paper on Software Defined Networking,” International Journal of Advanced Research in Computer Science, Volume 8, No. 2, March – April 2017.

4. https://www.opensourceforu.com/2016/07/implementing-a-software-defined-network-sdn-based-firewall/

5. 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).

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

7. Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas Schmidt, "Software- defined Networking: Challenges and Research opportunities for Future Internet."

8. Nitheesh Murugan Kaliyamurthy, Swapnesh Taterh, Suresh Shanmugasundaram, Ankit Saxena, Omar Cheikhrouhou, Hadda Ben Elhadj, "Software-Defined Networking: An Evolving Network Architecture- Programmability and Security Perspective."

9. Sharma, R. K., Kalita, H. K., Issac, B.: Different firewall techniques: a survey. In: 5th IEEE ICCCNT, Hefei, Anhui, China, pp. 1–6, (2014).

10. Cheng, Y., Wang, W., Wang, J., Wang, H.: FPC: a new approach to firewall policies compression. Tsinghua Sci. Technol. **24**(1), 65–76 (2018). https://doi.org/10.26599/TST.2018.9010003

11. Fahad N. Nife, Zbigniew Kotulski, "Application- Aware Mechanism for Software Defined Networks",(2020) https://link.springer.com/article/10.1007/s10922-020-09518-z

12. Duan, Q., Al-shaer, E.: Traffic-aware dynamic firewall policy management: techniques and applications. IEEE Communication Mag. **51**(7), 73–79 (2013). https://doi.org/10.1109/MCOM.2013.6553681

13. Trabelsi, Z.: Teaching stateless and stateful firewall packet filtering: a hands-on approach. In Proc. 16th Colloquium for Information Systems Security Education, Florida, USA, pp. 95- 102 (2012).

14. Sukhveer Kaur, Japinder Singh, and Navtej Singh Ghumman, "Network Programmability Using POX Controller", 2014 ICCCS pp. 134-138.

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

16. 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.

17. The Open Networking Foundation, OpenFlow Switch Specification (2014).

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

19. https://searchsecurity.techtarget.com/definition/firewall.