A Proposed Multi-Layer Firewall to Improve the Security of Software Defined Networks

https://doi.org/10.3991/ijim.v17i02.36387

Mohammed AbdulRidha Hussein
Institute of Applied Arts, Middle Technical University, Baghdad, Iraq
mohammed_it@mtu.edu.iq

Abstract—One of the most important motivations behind software-defined networking (SDN) is the desire to move from the approach of traditional networks to a more flexible and intelligent software development technology. This paper focuses on the importance of the SDN-based platform POX controller security firewall modules and their effectiveness on networks, including the central administration of the SDN-based platform controller for protecting the network from possible attacks. The work is performed using the Mininet emulator with the Iperf bandwidth measurement tool. Result reveals that the proposed multi-layer firewall does not compromise the flexibility of the network in any way.

Keywords—Software defined networks (SDN), SDN-based platform controller (POX), Network Control Program (NCP), Open Networking Foundation (ONF), SDN-based platform controller OpenDayLight (ODL), Quality of service (QoS), a protection architecture for enterprise networks (SANE)

1 Introduction

Researchers are now looking at virtual technologies like the Metaverse, the Internet of Things, and Software Defined Networks (SDN) that enable user access while being less expensive for the owners of these technologies than traditional networks.

However, the most significant challenges in these technologies still revolve around providing security and protection for data and users [1].

You have a property or a business in the virtual world (metaverse) and accounts for cryptocurrencies, which are financial matters.

All of these highly essential matters call for high security [2].

In terms of the Internet of Things, it will be disastrous for the original owner if the attacker has access to the house data and can operate the house instead of them [3].

Imagine creating an intelligent home and controlling equipment through the network [4].

As a result, creating an excellent firewall and protecting against penetration continue to be the primary challenges [1].

SDN is the subject of research, which we shall go into further depth about.

As a result of the latest networking achievements, a novel networking paradigm known as software defined networking (SDN) has emerged.

However, the concept of programmable networks is quite old, and it has been used in industry and education under various names over the years.

The last modification was the transition from NCP to TCP/IPAs a result of the significant advancements in information technology, particularly in the area of virtual technology, there has been no change in the network infrastructure, which made a virtual simulation of all layers of network design where the network infrastructure layer remained incapable of this technology [5].

The SDN's primary concept is to eliminate the intermediary devices, such as the firewalls which separate the data plane or what is known as the forwarding plane from the control plane, so that the role of network devices such as Switches/Routers are limited only to data forwarding.

Furthermore, the control layer and application layer are the new layers in which management, control, and services are run.

The separation of the two layers necessitates the existence of a protocol that regulates the communication between them [6][7][8].

The ONF organization has approved the SDN, so it is necessary to agree on a protocol that deals with the control and infrastructure layer, namely OpenFlow, to define a packet path based on predefined rules by the network engineer [9].

The OpenFlow protocol outlines the appropriate function (Action), such as forwarding or dropping the packet [10].

As the SDN is a modern model of networks, the most prominent difficulties lie in the transition to their application due to the presence of devices that currently work on the traditional system [11].

By comparing with the Traditional Networks, it is possible to differentiate the SDN as it could be configured easily while the network is working in addition to the advantage of control de-centralization considered as one of the ingredients for the success of this type of network to reduce the cost of operating and managing the network [12][13].

The ONF organization defined an architecture for SDN technology in the form of a three-layer module:

1.1 Application layer

Applications and services offered by the network to users are contained in this layer. The most prominent examples are routing policies and QoS. This layer communicates with the control layer using the application programming interfaces (APIs) [14]. On the other hand, Engineers use interfaces to help the network perform its services and applications by programming them.

1.2 Control layer

This layer represents the control point for centralizing network devices, such as giving orders to networked devices (routers or switches) to achieve control and management in all devices Infrastructure. The component that performs the function of Control and management is called the controller [15]. Many controllers, some of which are open source, such as OpenDayLight (ODL), are based on a programming language (Java). In contrast, others are specific to certain companies, such as Cisco or VMWare.

1.3 Data layer

This layer is made up of networking devices that carry out data forwarding,

However, whether physical or virtual, it should be distinguished that devices must support the OpenFlow protocol for any communication between the controller and the device. When a user requests to connect to an entity, the network device communicates with the controller to determine the customized path and apply restrictions. Then the connection process takes place [16].

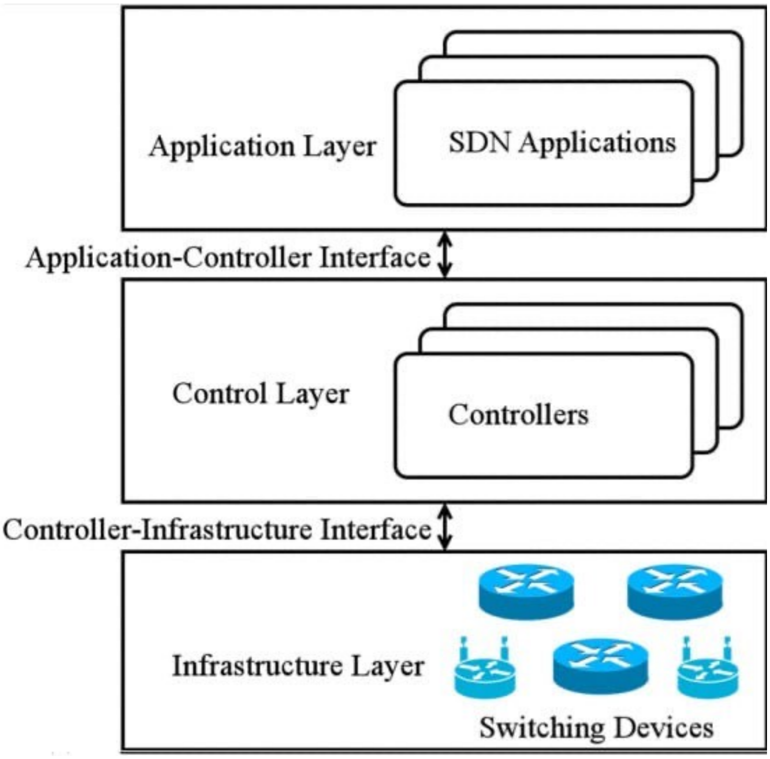

Fig. 1. SDN-based platform architecture [17]

As shown in the Table 1 There are many differences between SDN architecture and traditional networks, Figure 2 and Figure 3, respectively [18].

Table 1. Compare SDN and a traditional network [18] [19]

Characteristics	SDN	Traditional network
Switching and router device responsibility	forwarding data in the forwarding table	forwarding data in the forwarding table and controlling
Time of execution	faster, and controller utilization	Slower
Repair cost	Because there are fewer parts, it is less expensive.	high cost due to complex infrastructure
Global network view	the controller's centric view	Complexity
Security	More security strategies	Less secure strategies

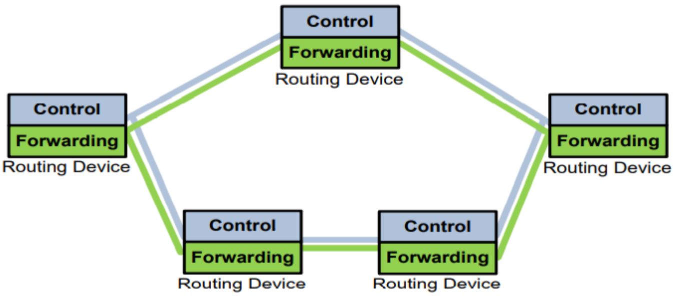

Fig. 2. Routing device in Traditional network. [20]

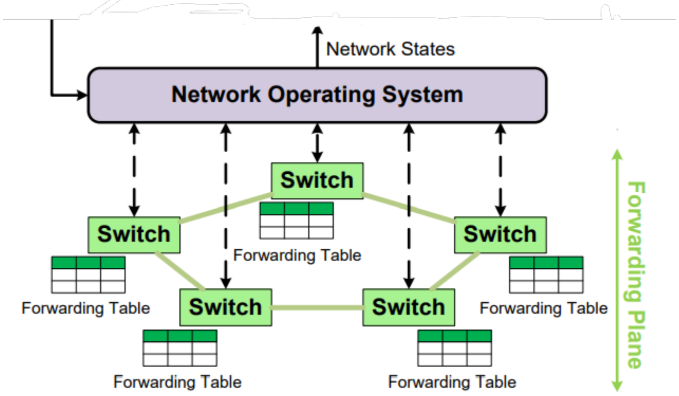

Fig. 3. Switching devices in SDN

2 Related works

The centralized controller management and programmability attributes impose several network security challenges [21].

A good example is the denial of service DoS. DoS or DDoS attacks are launched by generating several new flows that flood up the control plane's, OpenFlow switches', and SDN controller's bandwidth., which results in network failure for legitimate hosts [22].

The other example is the open-programmable API between the applications and controllers, controllers, network devices and controllers [23].

The "man-in-the-middle" monitors can also collect or consume data without being noticed, leading to a black hole attack as the SDN controllers and switches are not physically connected throughout the data transfer [24].

Security experts must safeguard data, communication transactions, and networking assets all over the network to build a safe network from malicious attacks or unintended harm.

As a result, In order to maintain network security, the changes brought about by SDN to the network architecture must be evaluated. In an early version of SDN, the SANE protection architecture of enterprise networks introduced a security issue between forwarding and control platforms.

The SANE emphasizes a logically centralized controller responsible for policy enforcement and host authentication [25].

The SANE's work extended by dealing with a different approach that required less modification to the original network [22], which was done by managing the network via two components: the ethane switches that forwarded the packets based on flow table rules and a centralized controller, which is responsible for applying the global policy. As a result of modifying elements or the relationships between elements in the SDN architecture, additional vulnerabilities and security problems have been addressed.

The complete analysis of the OpenFlow protocol, which focused mainly on the execution of Dos attacks and information leaks, was represented by the STRIDE threat methodology [26].

The overall analysis of SDN security concluded that new threats are presented due to the centralized controller's nature, and the network's programmability, which demanded new responses, was introduced [27].

Installation of Mininet and the POX controller are necessary for developing an SDN-based firewall, where both are open source and free [28]. It emphasizes the creation of a simple network topology using POX in Python [29].

As the SDN has evolved to replace the traditional design of the current network, the need to develop modules related to network security has become urgent. However, there needed to be more focus on employing proactive logic designed by the administrator through multiple layers to prevent several DOS and fingerprinting attacks.

3 SDN security strengths and shortcomings

The operation and discipline of network security are extraordinarily sophisticated and technological.

Several networking technologies, including SDN, use virtualization techniques, and security is one of the main problems [30].

One of these technologies is blockchain, which is one of the ways to give SDN security, guess the load for the control plane by monitoring network traffic, and then distribute the load balance [31].

3.1 SDN security strengths

From a security standpoint, we find three key characteristics that set SDN networks apart from traditional ones. Next, we go over each of those characteristics, describing why they are absent from traditional networks and how each might be used to increase network security [20].

- Global Network View

SDN's most significant security advantage over the traditional network is that the controller in the SDN paradigm has a global network perspective. Centralization and the fact that every network component is gathering and reporting traffic statistics are credited for this network perspective [20].

- Autonomous Repair Mechanisms

The activated rule describes how the switch should react When a particular condition is fulfilled.

These responses offer automated resistance against attackers [20].

- Increased Control Capabilities

The SDN controller can improve access control by specifying the types of packets that should be sent into the network according to the payload type, the source address, or any other header field value.

For instance, Only TCP packets arriving from a particular host may be allowed to be routed via the network according to the rules set up by the controller.

Helps to stop malicious traffic from coming into the SDN network or coming from any of its switches [20].

3.2 SDN security shortcomings

Different threat vectors have already been identified in SDN, along with several problems and weak points in SDNs built on the OpenFlow protocol. Some of these threat vectors are typical of current networks, whilst others, such as attacks on logically centralized controllers and control plane communications, are connected to SDN.

There are at least seven known threat vectors for SDN architecture. The first uses fake and forged data plane traffic flows to attack controllers or forwarding equipment. The second enables an attacker to take advantage of forwarding device vulnerabilities and cause network havoc. The riskiest vectors are three, four, and five because they potentially jeopardize network performance by attacking the control plane. If the attack is successful, controllers and applications will readily give an attacker control of the network. Attacks and vulnerabilities in the administrative department are the sixth threat vector. The final vector shows a lack of reliable forensics and repair resources that can prevent backing up the network in a secure and functional form [21].

4 Contribution

Through the reforms and growth in network size, it has become challenging to carry out maintenance work alongside security.

With the help of the SDN, Layer 4–7 firewalls, load balancers, and IPS/IDS systems might be replaced with low-cost, high-performance switches and a logically centralized controller.

In this project, proactive rules associated with the physical address-based module on POX runtimes developed a multi-layer firewall that includes Layers 2, 3, and 4.

The firewall application is provided with a list of MAC address pairs, i.e., an access control list (ACL).

The application inserts static flow rule entries in the OVS OpenFlow switch database to block all communication between each MAC pair when a connection is made between the controller and the switch.

The proposed procedure managed successfully to block several applications like specific links, host-to-host connectivity and destination process without causing any lack in network flexibility.

5 Experimental configuration

The experimental work is performed by utilizing several tools. Initially, the OpenFlow protocol ver.1.0 is selected as the communication protocol between the POX controller (selected as the default controller due to its free, open-source utilities that enables the removal and the addition of the reusable components) and the OpenFlow OVS switch. Port 6633 is the default communication port between the controller and the OVS OpenFlow switch.

The Mininet emulator creates the virtual SDN-based platform network topology of type linear.

For n number of hosts, a linear architecture necessitates n number of switches. Each host will be connected to its switch. In this scenario, 8 hosts holding similar conditions connect to 8 OpenFlow OVS switches.

The network topology, and the complete configuration steps, are shown in Figure 4 and Table 2 respectively.

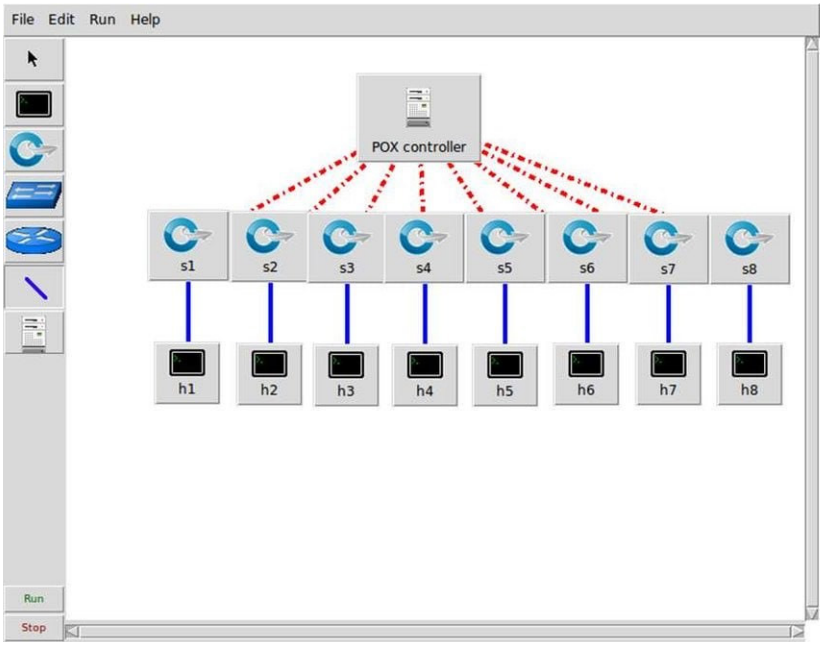

Fig. 4. SDN-based platform network topology

Table 2. SDN-based platform network topology configuration

Function	Configuration
Construct the SDN network topology	Mininet> Sudo mn –topo single, 8 –switch OVSK –controller=remote Where : -The (Sudo) assigns the administrator role privileges to the user -The (mn) initializes the Mininet emulator -The (--topo) constructs the network topology consisting of 8 hosts -The (--switch) invokes the OVS OpenFlow switch into the network topology -The (--controller) invokes the SDN-based platform, remote POX controller, into the network topology
Initialize the POX controller	Mininet> ~/pox/pox.py forwarding.l2_learning Where: -The (Forwarding.l2_learning) is the module invoked by the SDN-based platform POX controller to act as a layer-2 switch

6 Proposed multi-layer firewall design

Generally, a firewall could be utilized to protect the network from the Internet. The proposed firewall modifies filtering rules to apply application-level access control policies across multiple network segments. It is assumed as the entity that blocks and filters the incoming traffic based on some rules.

For an SDN-based platform firewall, an SDN-based platform POX controller filters traffic that passes between hosts according to some rules, thus allowing it to pass.

A firewall prototype is started using the Python platform.

When POX connects with the OVS OpenFlow switch, the component activates the connection, and specific types of packets are permitted to pass along by adding flow entries with low priorities.

A rule in the firewall configuration file blocks all packets defined by that rule. When the firewall receives a packet, the configuration rules are investigated. If there is no match, then a symmetric flow entry is pushed based on the flow specifications into the OVS OpenFlow switch, causing the OVS OpenFlow switch to add an entry for the packet. Otherwise, the packet is dropped.

The Proposed Multi-Layer firewall module is represented by the POXController_Firewall.py file invoked by the SDN-based platform POX controller. The Proposed Multi-Layer firewall consists of the rules listed in the Table 3.

Table 3. Proposed Multi-Layer firewall rules

Firewall-Type	Rule	Output
Layer-2 (Link connection blocked)	-AddRule(dataPath_id, Host_MAC Address, NULL, NULL, NULL)	-The ARP L2-packets will be dropped by the SDN-Based platform POX Controller.
Layer-3 (Host-to-Host Connectivity blocked)	-AddRule (dataPath_id, NULL, Source_Host IP Address, Destination_Host IP address NULL) -Block all traffic from Host-h1 to Host-h5 -AddRule(dataPath_id, NULL, 10.0.0.1, 10.0.0.5, NULL)	-The SDN-based platform POX Controller will drop the IP Layer packets.
Layer-4 Destination Process blocked	-AddRule (dataPath_id, NULL, NULL, Server IP Address, Server Port No.)	-The SDN-based platform POX Controller drops All TCP traffic to Destination h3 and port 80.

Algorithm–POX Controller Multi-Layer Firewall

```
  Input:
  No. of OVS OpenFlow Switches
  No. of hosts
  Firewall_LayerNo.Rule
  Output:
  Apply the Multi-layer Firewall module to Filter the Packet_in traffic
  Process:
  Initialize Mininet libraries
  Initialize the L2_learning switch for the remote POX controller
  Initialize the POX controller & Test the Connection Between Hosts
  Firewall_Table[1,…3]= ‶ Firewall_Layer2Rule, Firewall_Layer3Rule, Firewall_Layer4Rule
  While Packet_in is True
  For Firewall_LayerNo.Rules 1 to 3 ➔
  If the Packet_in message matches the Firewall_Ta-
ble[Firewall_LayerNo.Rule]
  Then drop the packet from the OVS OpenFlow switch
   Else a symmetric flow entry with is pushed into the
OVS OpenFlow switch
      End for
  End While
  Stop the POX controller and clear the topology
  End
```

7 Performance evaluation

Through this paper, two scenarios are conducted:

The first scenario involves the run of the SDN-based platform network topology without a firewall and the second scenario involves the run of the SDN-based platform with the proposed firewall.

After the connection between the POX controller and the OVS, the OpenFlow switch is established. Each pair of hosts is usually verified through the entire network by typing a Pingall in the Mininet console.

However, it should be remembered that there is a need to wait for the timeout, which takes about 10 seconds to get through a pair of hosts.

The Pingall sends a series of Internet Control Message Protocol (ICMP) messages. The ICMP offers two query messages that cooperate. The ECHO REQUEST message is a probe sent by a user to a destination system. The destination system replies with an ICMP ECHO RESPONSE message, which shows the amount of time needed for each packet to complete its round trip and offers a summary that includes the number of packets sent and received and the minimum, maximum and average response times.

The ECHO_REQUEST will be sent after ping exits start from 10 up to 100. the results clearly show that the proposed multi-layer firewall does not influence the filtering process of the network traffic.

Moreover, the average times in both scenarios are not significantly different, which implies that network latency seems to be not adversely affected by multi-layer firewalls, as shown in the Figure 5.

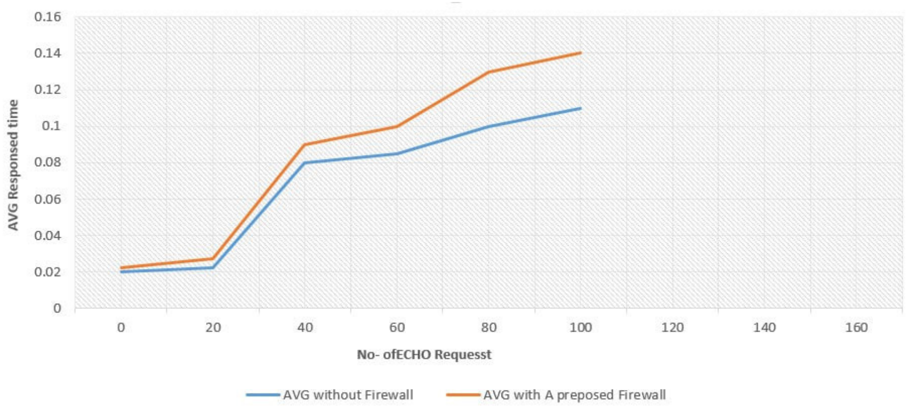

Fig. 5. Average time with and without the proposed multi-layer firewall

8 Conclusion

The suggested SDN Firewall divides the flat network into virtual LAN segments following security standards. The aim is to overcome the limitations of the existing firewall systems in the SDN-Based platform environment, which will provide adequate protection and manual work reduction. However, based on the above results, the proposed approach will help the users reduce their workload and conflicts.

9 References

[1] Shiravi, H., Shiravi, A. and Ghorbani, A.A., 2011. A survey of visualization systems for network security. *IEEE Transactions on visualization and computer graphics*, *18*(8), pp.1313-1329. https://doi.org/10.1109/TVCG.2011.144

[2] Abdulsattar Jaber , T. . (2022). Security Risks of the Metaverse World. *International Journal of Interactive Mobile Technologies (iJIM)*, *16*(13), pp. 4–14. https://doi.org/10.3991/ijim.v16i13.33187

[3] Al Reshan, M. S. (2021). IoT-based Application of Information Security Triad. *International Journal of Interactive Mobile Technologies (iJIM)*, *15*(24), pp. 61–76. https://doi.org/10.3991/ijim.v15i24.27333

[4] Jaber, T.A. and Hussein, M.A., 2019, May. Study on known models of NB-IoT Applications in Iraqi environments. In *IOP Conference Series: Materials Science and Engineering* (Vol. 518, No. 5, p. 052013). IOP Publishing. https://doi.org/10.1088/1757-899X/518/5/052013

[5] Edelman, J., Lowe, S. S., & Oswalt, M. (2018). Network Programmability and Automation: Skills for the Next-Generation Network Engineer. " O'Reilly Media, Inc.".

[6] Bruno Astuto A., et al., "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," IEEE Communication Surveys and tutorials, vol. 16, no. 3, pp. 1617-1634, 2014. https://doi.org/10.1109/SURV.2014.012214.00180

[7] Wenfeng Xia, Yonggang Wen, Chuan Heng Foh, Dusit Niyato, Haiyong Xie, "A Survey on Software- Defined Networking," IEEE Communication Surveys and Tutorials, vol. 17, no. 1, pp. 27-51, 2015. https://doi.org/10.1109/COMST.2014.2330903

[8] Buranova, M., & Muthanna, A. (2021, October). Performance evaluation of software defined networking based on openflow protocol. In 2021 13th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT) (pp. 143-148). IEEE. https://doi.org/10.1109/ICUMT54235.2021.9631571

[9] Salazar-Chacón, G. D., & Marrone, L. (2020, November). OpenSDN Southbound Traffic Characterization: Proof-of-Concept Virtualized SDN-Infrastructure. In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) (pp. 0282-0287). IEEE. https://doi.org/10.1109/IEMCON51383.2020.9284938

[10] Hong, E. T. B., & Wey, C. Y. (2017, January). An optimized flow management mechanism in OpenFlow network. In 2017 International Conference on Information Networking (ICOIN) (pp. 143-147). IEEE. https://doi.org/10.1109/ICOIN.2017.7899493

[11] Lali, M. I., Mustafa, R. U., Ahsan, F., Nawaz, M. S., & Aslam, W. (2017). Performance evaluation of software defined networking vs. traditional networks. The Nucleus, 54(1), 16-22.

[12] Prajapati, A., Sakadasariya, A., & Patel, J. (2018, January). Software defined network: Future of networking. In 2018 2nd International Conference on Inventive Systems and Control (ICISC) (pp. 1351-1354). IEEE. https://doi.org/10.1109/ICISC.2018.8399028

[13] Singh, U., Vankhede, V., Maheshwari, S., Kumar, D., & Solanki, N. (2019, August). Review of Software Defined Networking: Applications, Challenges and Advantages. In International Conference on Inventive Computation Technologies (pp. 815-826). Springer, Cham. https://doi.org/10.1007/978-3-030-33846-6_89

[14] Sarhan, S. A. E., Sobh, M. A., & Bahaa-Eldin, A. M. (2018, December). Data Inspection in SDN Network. In 2018 13th International Conference on Computer Engineering and Systems (ICCES) (pp. 436-441). IEEE.

[15] Paliwal, M., Shrimankar, D., & Tembhurne, O. (2018). Controllers in SDN: A review report. IEEE Access, 6, 36256-36270. https://doi.org/10.1109/ACCESS.2018.2846236

[16] Li, T., Chen, J., & Fu, H. (2019, April). Application scenarios based on SDN: an overview. In Journal of Physics: Conference Series (Vol. 1187, No. 5, p. 052067). IOP Publishing. https://doi.org/10.1088/1742-6596/1187/5/052067

[17] Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. IEEE Communications Surveys & Tutorials, 17(1), 27-51. https://doi.org/10.1109/COMST.2014.2330903

[18] Deb, R. and Roy, S., 2022. A comprehensive survey of vulnerability and information security in SDN. Computer Networks, p.108802. https://doi.org/10.1016/j.comnet.2022.108802

[19] Singh, A.K. and Srivastava, S., 2018. A survey and classification of controller placement problem in SDN. International Journal of Network Management, 28(3), p.e2018. https://doi.org/10.1002/nem.2018

[20] Dabbagh, M., Hamdaoui, B., Guizani, M. and Rayes, A., 2015. Software-defined networking security: pros and cons. IEEE Communications Magazine, 53(6), pp.73-79. https://doi.org/10.1109/MCOM.2015.7120048

[21] Aziz, N.A., Mantoro, T. and Khairudin, M.A., 2018, September. Software defined networking (SDN) and its security issues. In *2018 International Conference on Computing, Engineering, and Design (ICCED)* (pp. 40-45). IEEE. https://doi.org/10.1109/ICCED.2018.00018

[22] Shamsan, A. H., & Faridi, A. R. (2021, August). Security Issues and Challenges in SDN. In International Conference on Advances in Cyber Security (pp. 515-535). Springer, Singapore. https://doi.org/10.1007/978-981-16-8059-5_32

[23] Kandoi, R., & Antikainen, M. (2015, May). Denial-of-service attacks in OpenFlow SDN networks. In 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM) (pp. 1322-1326). IEEE. https://doi.org/10.1109/INM.2015.7140489

[24] Dayal, N., Maity, P., Srivastava, S., & Khondoker, R. (2016). Research trends in security and DDoS in SDN. Security and Communication Networks, 9(18), 6386-6411. https://doi.org/10.1002/sec.1759

[25] Brooks, M., & Yang, B. (2015, September). A Man-in-the-Middle attack against OpenDayLight SDN controller. In Proceedings of the 4th Annual ACM Conference on Research in Information Technology (pp. 45-49). https://doi.org/10.1145/2808062.2808073

[26] Bholebawa, I. Z., & Dalal, U. D. (2018). Performance analysis of SDN/OpenFlow controllers: POX versus floodlight. Wireless Personal Communications, 98(2), 1679-1699. https://doi.org/10.1007/s11277-017-4939-z

[27] Scott-Hayward, S., O'Callaghan, G., & Sezer, S. (2013, November). SDN security: A survey. In 2013 IEEE SDN For Future Networks and Services (SDN4FNS) (pp. 1-7). IEEE. https://doi.org/10.1109/SDN4FNS.2013.6702553

[28] Jiménez, M. B., Fernández, D., Rivadeneira, J. E., Bellido, L., & Cárdenas, A. (2021). A Survey of the Main

[29] Kaur, K., Kumar, K., Singh, J., & Ghumman, N. S. (2015, March). Programmable firewall using software defined networking. In *2015 2nd International Conference on Computing for Sustainable Global Development(INDIACom)* (pp.2125-2129).IEEE.

[30] Gautam, Y., Gautam, B. P., & Sato, K. (2020, December). Experimental security analysis of SDN network by using packet sniffing and spoofing technique on POX and Ryu controller. In *2020 International Conference on Networking and Network Applications (NaNA)* (pp.394-399). IEEE. https://doi.org/10.1109/NaNA51271.2020.00073

[31] Shiravi, H., Shiravi, A. and Ghorbani, A.A., 2011. A survey of visualization systems for network security. *IEEE Transactions on visualization and computer graphics*, *18*(8), pp.1313-1329. https://doi.org/10.1109/TVCG.2011.144

[32] Mohammed, M. A. ., & Abdul Wahab, H. B. . (2022). Proposed New Blockchain Consensus Algorithm. *International Journal of Interactive Mobile Technologies (iJIM)*, *16*(20), pp. 162–176. https://doi.org/10.3991/ijim.v16i20.35549

10 Author

Mohammed AbdulRidha Hussein is an Iraqi computer scientist Holding a Master's Degree from GGSIPU, Delhi, India. Faculty of IT centre and Quality Assurance department. He often works as a government Institute Lecturer at (email: mohammed_it@mtu.edu.iq).

Article submitted 2022-10-25. Resubmitted 2022-11-29. Final acceptance 2022-12-11. Final version published as submitted by the author.