International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 10, October 2022)

A System for Collaborative Intrusion Detection & Prevention

Deepak Shrivastava[1], Dr. Pankaj Richhariya[2]
[1,2]*Bhopal Institute Of Technology and Science, Bhopal (M.P.), India*

***Abstract*-- The Internet of Things (IoT), big data, and data centres, along with existing networks, have created new problems for networks. These challenges include the need for smarter and more creative methods of dynamically controlling traffic and allocating scarce network resources. These problems are intended to solve the software defined network (SDN), which, through network vitalization, decouples the control plane from the data plane. In this article, the OpenFlow protocol is used to implement and analyse the SDN architecture. Additionally, it has assessed some of its benefits over conventional network architectures, safety issues, and potential solutions for future research and similar projects in developing nations like India.**

***Keywords*-- SDN; OpenFlow; Mobile Networks; Network Security; IoT; Big Data**

I. INTRODUCTION

Background

In order to increase the programmability and flexibility of the control and management of a network, Software Defined Networking (SDN) decouples the control plane from the data plane. Legacy networks are viewed as being rigid and complex, hard to scale and operate, and too expensive, yet SDN offers a more creative and dynamic network design that converts conventional network architecture into rich service-delivery platforms [1]. SDN adds a layer of software to the network that functions as a network operating system and communicates with each network router. Its inherent security and streamlined networking are significant results of its design and development. Its emergence provides a stable environment for developing dynamic, cost-effective, adaptable, and flexible future networks that are ideal for high bandwidth utilisation and the dynamic character of current applications [2].

A Software Defined Network model made consisting of the application plane, the controller plane, and the data plane was suggested by the Architecture and Framework working group [3]. By separating the control plane (the controller), which determines where packets are routed, from the data plane (the physical network), which delivers traffic to its destination, SDN developers hope to achieve scalability and agility in network administration [3]. In order to achieve its infrastructure objectives, SDN increasingly leverages elastic cloud topologies and dynamic resource allocation [4].

Flow-based forwarding decisions are made in SDN as opposed to destination-based ones in traditional networks. The majority of businesses have now chosen to embrace other protocols from Openflow, which was the most widely used SDN protocol. Open network environment by Cisco and network virtualization platform by Nicira are two protocols that are now in use [3]. The purpose of this paper is to outline some of the difficulties and areas for future study related to SDN adoption in India. Data centres, wide-area backbone networks, enterprise networks, internet exchange points, and home networks are a few examples of SDN's uses.

Figure 1 below consists of the 3 distinct layers: application layer; control layer; and infrastructure or physical layer.

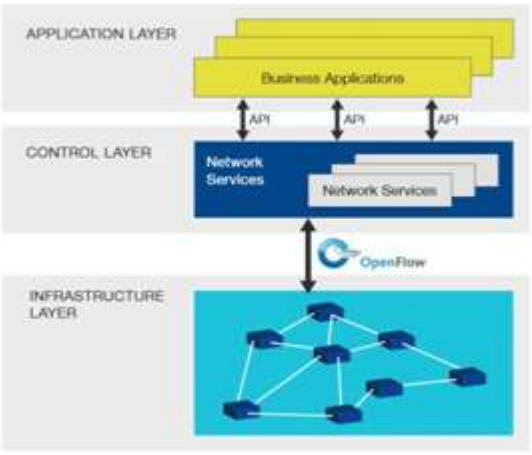

Figure 1. SDN Architecture [3]

Benefits of SDN

The ability of the network to adapt to changing networks is increased by the separation of the control and data planes. Because of efficient operations, centralised management, and full utilisation of current hardware, one of the main advantages for operators and service providers is a decrease in operating costs. The networking infrastructure is scalable and more dynamic since it can be managed and programmed. Along with the improved user experience brought on by SDN's ability to react to changing user needs, additional advantages include the higher network security and reliability highlighted in this paper.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 11, Issue 10, October 2022)

By segmenting the traffic and assisting in the organisation of the data, SDN is also anticipated to manage the influx of traffic from internet of things (IoT) devices. Additionally, SDN is anticipated to make it possible for networks to keep up with the rate of change on a network without having to continuously invest in new hardware or infrastructure.

SDN and Mobile Networks

Mobile networks, including wireless access points, mobile backhaul networks, and core networks, can benefit from OpenFlow-based SDN in a number of ways [5], [6]. By embracing cutting-edge methods of monitoring and regulating the network, SDN will allow carrier networks to benefit from its architecture [5], [6]. Additionally, it would improve flexibility by enabling a more rapid introduction of value-added services and new service bundles in India. The SDN principle will be incorporated into the framework and network slicing concept of the upcoming 5G network. According to a recent position paper on SDN-based mobile networks by Malik et al. [5], it can simplify mobile networks and reduce management expenses. Additionally, SDN in mobile networks is anticipated to give future carriers the greatest flexibility, openness, and programmability possible without requiring modifications to user-equipment. SDN might also provide cell carriers more control over their hardware and infrastructure and make network administration simpler.

II. Literature Review: Related Works

Switches and routers configured for data packet and routing capacity are used in computer networks to simplify communication between the network and its host. The devices are manually configured by converting high-level network regulations into device-specific low-level commands using command-line or graphical user interfaces (GUI) [7]. This is vulnerable to network dangers and attacks include Denial of Service (DoS) attacks, attacks utilising compromised controllers (faulty or hijacked controllers), spoofing attacks, malicious interjection attacks, traffic anomaly attacks, and forwarding control link attacks.

Colville and Spafford [8] contend that the absence of integrated network control makes managing networks challenging and that the prone configuration process results in network vulnerabilities, malfunctions, and security breaches. According to Feldmann et al. [9], rigidity has effectively put a stop to network innovation. However, the SDN paradigm directly addresses this problem by separating the control element's or control plane's control plane from the forwarding devices' or data plane's capabilities for packet forwarding [6].

Decoupling, often known as separation in technical jargon, is still a key element of SDN. Decoupling produces innovative network design, in which the network switches serve as basic forwarding devices and the control logic is merged into a logically centralised controller [10].

The integrity and security of SDNs have not been independently validated, according to Akhunzada et al. [11], because management functionality is centrally located on a single virtual server, making it easier to compromise the entire network through a single point of failure. However, SDN presents a unique opportunity for swiftly discovering and managing network security concerns in home and commercial networks, claim Medhi et al. [12]. Four well-known track anomaly detection techniques can be implemented in an SDN framework employing Open flow compatible switches and NOX (an open source development environment for C++-based SDN control applications), according to research by Medhi et al. [12].

controller. Additionally, they claimed that these algorithms are far more accurate than Internet Service Providers (ISP) at identifying fraudulent activities on residential networks [12].

The main security issue for SDN is self-security. According to Kreutz et al. [6], security and dependability should be incorporated into the SDN design from the very beginning. They assert that SDN is susceptible to a variety of threats, such as attacks on switches, controllers, and control plane communications that target network entities using fictitious traffic flow [13]. Potential attacks that exploit weaknesses in network switches and the Secure Socket Layer (SSL) and Transport Layer Security (TLS) protocol implementations may target the interface between the controller and high-level applications [14]. These are the gaps that this study will attempt to close on the security issues raised by the advancement of SDN and the use of it by service providers in India.

Since monotony-regimes are one of the only workable remedies to the vulnerabilities found in the present SDN [16], Kreutz et al. [15] proposed stringent authentication procedures and trust models that might withstand common identity-based attacks. In order to reduce implementation vulnerabilities that are regularly observed, it is necessary to diversify the controllers, tools, and protocols used. Shin et al[17] .'s proposal of FRESCO, a security-specific application development framework for OpenFlow networks, aims to safeguard the SDN architecture. FRESCO accelerates the transfer of application programming interface (API) scripts to make it simpler to construct threat-detection logic and security monitoring as programming libraries [17]. However, Akhunzada et al. [11] claim that FRESCO does not improve the security of the application and infrastructure layers of SDN.

To increase the security of SDN, CloudWatcher, a framework for monitoring clouds, and FleXam, a sampling extension for OpenFlow, are suggested as alternatives by Shing and Gu [16] and Shirali-Shahrez and Ganjali [18]. Kreutz et al. [13] recommend L-IDS, a learning intrusion detection system, as a crucial security-boosting strategy. It is intended to keep mobile devices safe in a certain location. Furthermore, Wang et al. [19] present a systematic approach for identifying and resolving conflicts in an SDN firewall by investigating the firewall authorization space and flow space using "header space analysis" to assess the efficacy and efficiency of this approach in addressing security concerns.

Using connection migration, a feature of the data panel that reduces contacts between the data and control panel, is recommended by Shin et al. [17] as a defence against DoS assaults on the southbound interface. This is comparable to the approach suggested by Ying-Dare et al. [20] for reducing the controller's traffic overhead and implementing NFV through an improved SDN architecture. Their analysis shows that the expanded controller in the extended SDN architecture only manages 0.12% of the input traffic, as opposed to the conventional design's controller, which manages 77.23% [20]. Ali et al. [7] state that OpenWatch, an adaptive method of flow counting to detect anomalies in SDN, is a credible solution for security analyses and is expected to improve the overall security of Network protocols like OpenFlow as cyber-threats continue to evolve and become more sophisticated. Therefore, moving away from the reactive strategy used by Akhunzada et al. [11] is necessary. It is obvious that SDN is susceptible to a wide variety of vulnerabilities. In order for SDN to be widely embraced and used in India, this study will address security concerns to SDN configuration.

III. Discussions And Future Research

As SDN is being adopted, there is demand for secured SDN solutions and a more adaptable secured framework. Several issues relating to SDN are currently actively being researched, however, there is also need for security vulnerability assessment because this is an important process that must be conducted to fully secure a system before its deployment. The Control plane in SDN handles configuration management of devices, responsible for routing decisions and monitoring the network. The controller is considered as a single point of failure [15], and it is a major security target. In this regard, there is need to investigate new security architectures for the controller to support more innovative security services andintelligent network defense systems.

FRESCO by Shin et.al. [17], is an extension of the research work done by Kreutz et.al. [15], which that makes it easy to make and deploy security services in SDN, however, they believe none of those works adopts or enforces the security of SDN itself [15]. Furthermore, there is need for research on creating more secured and resilient SDN controllers and approaches to addressing the security issues. This therefore generates the normative question of how innovative SDN-based security applications can potentially replace existing security applications? There is also the question of how new vulnerabilities in SDN controllers can be exploited through threat vectors and possible solutions and improvements to address these problems? In addition, there is the need to question the handling of malicious applications being developed and deployed on SDN controllers?

IV. Conclusions

The notion of a software-defined network framework, which leverages network virtualization to isolate the control plane from the data plane, has been covered in this paper. SDN can easily manage networks and make them able to adapt and deal with unpredictable traffic patterns that can put a lot of strain on the scarce network resources. Security is a big worry in any network, though, thus it is crucial to analyse the security issues with SDN from the standpoint of attacks on SDN controllers and develop novel mitigation strategies and security models. To do this, further research, data collection, testing, expert consultations, and feasibility studies are needed. Future research will result in more creative ways to manage threats, mitigate assaults on SDN controllers, and improve overall network management and security. This will enable safer networks and encourage the use of SDN, which is thought to be more economical and helpful for developing nations like India.

REFERENCES

[1] Liu, S., and Li, B. 2015. On Scaling Software-Defined Network in Wide-Area Networks. Tsinghua Since and Technology. 20(3). 221-232.

[2] Open Network Foundation 2015. Principles and Practices for Securing Software-Defined Networks. ONF TR-511.

[3] Anthony, L. 2015. Security Risks in SDN and Other New Software Issues. RSA Conference. Frost and Sullivan.

[4] Malik, M.S., Montanari, M., Huh, J.H., Bobba, R.B., and Campbell, R.H. 2013. Towards SDN enabled network control delegation in clouds. 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN).

[5] Open Network Foundation (ONF). 2013. SDN Architecture Overview.

[6] Ali, S.T., Sivaraman, V., Radford, A., and Jha, S. 2013. A Survey of Securing Networks Using Software Defined Networking. IEEE Transactions on Reliability. 3(64).

[7] Colville, J., and Spafford, G. 2010. Configuration Management for Virtual and Cloud Infrastructures. Gartner Inc.

[8] Feldmann, A., Kind, M., Maennel, O., Schaffrath, G., and Werle, C. 2013. Network Virtualization - An Enabler for Overcoming Ossification. Future Internet Technology. European Community in Information Technology (ERCIM) News.

[9] Open Network Foundation. 2013. OpenFlow-Enabled Mobile and Wireless Networks. ONF Solution Brief.

[10] Akhunzada, A., Ahmed, E., Gani, A., Khan, M.K., Imran, I. and Guizani, S. 2015. Security and Privacy in Emerging Networks: Securing Software DefinedNetworks: Taxonomy, Requirements, and Open Issues. IEEE Communications Magazine. 34-44.

[11] Mehdi, S.A., Khalid, J., and Khayam, S.A. 2011. Revisiting traffic anomaly detection using software defined networking. In Proceedings of 14th Int. Symposium on Recent Advances in Intrusion Detection (RAID). 6961. 161–180.

[12] Kreutz, D., Ramos, F.M.V., and Verissimo, P. 2013. Software-Defined Networking: A Comprehensive Survey. In Proceedings of the IEEE,103(1). 55-60.

[13] Dabbagh, M., Hamdaoui, B., Guizani, M., and Rayes, A. 2015. Software-Defined Networking Security: Pros and Cons. IEEE Communications Magazine, Communications Standards Supplement.

[14] Kreutz, D., Ramos, F.M.V., and Verissimo, P. 2013. Towards secure and dependable software defined networks, In Proceedings of the second ACM SIGCOMM Workshop on Hot topics in software defined networking. ACM, 55–60.

[15] Shin, S., and Gu, G. 2013. CloudWatcher: network security monitoring using OpenFlow in dynamic cloud networks. Springer. 92–103.

[16] Shin, S., Yegneswaran, V., Porras, P.A., and Gu, G. 2013. AVANT-GUARD: Scalable and Vigilant SwitchFlow Management in Software-Defined Networks, In Proceedings of 2013 ACM SIGSAC Conference on Computer and Communications Security. 413–424.

[17] Shirali-Shahrez, S., and Ganjali, Y. 2013. FleXam: Flexible Sampling Extension for Monitoring and Security Applications in OpenFlow. In Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. HotSDN'13. 167-168.

[18] Wang, Y., Wen, X., Chen, Y., Hu, C., and Shi, C. 2013. Towards a Secure Controller Platform for Openflow Applications, Proc. 2nd ACM SIGCOMM Workshop on Hot topics in Software Defined Networking, 171–72.

[19] Ying-Dar, L., Po-Ching, L., Chin-Hung, Y., Yao-Chun, W., and Yuan-Cheng, L. 2015. An Extended SDN Architecture for Network Function Virtualization with a Case Study on Intrusion Prevention, IEEE Network.

[20] Li, Y. 2014. Computer Networks 72. 74–98.

[21] Metzler, J. 2012, Understanding Software-Defined Networks, Information Week Reports. 1–25.

[22] Scott-Hayward, S., Natarajan, S., and Seker, S. 2016. A Survey of Security in Software Defined Networks. IEEE Communication Surveys and Tutorials. 18(1).

[23] Shin, S., Porras, P., Yegneswaran, V., Fong, M., Gu, G., and Tyson, M. 2013. FRESCO: Modular Composable Security Services for Software-Defined Networks. IOSC Network and Distributed System Security Symposium (NDSS).

[24] Anon. 2016. Software-Defined Networking (SDN) Definition. [online] Available at: http://www.opennetworking.org. [Accessed 5 Mar.2007].

[25] Son, S., Shin, S., Yegneswaran, V., Porras, P.A., and Gu, G. 2013. Model Checking Invariant SecurityProperties in OpenFlow. In Proceedings of IEEE ICC. 74–79.