Software-Defined Networking: A New Approach to Fifth Generation Networks – Security Issues and Challenges Ahead

[1]Behrooz Daneshmand
Faculty of infocommunication technologies
ITMO University
Saint Petersburg, Russia
daneshmandbehrooz@gmail.com

[2]Tu Anh Le
Faculty of infocommunication technologies
ITMO University
Saint Petersburg, Russia
ale@itmo.ru

***Abstract*— In order to satisfy the demands of 5G mobile networks, Software Defined Networking (SDN) has been developed. Security concerns, including man-in-the-middle (MITM) assaults, denial of service (DoS) assaults, and other cases are made possible by the separation of the data planes and control planes. Security concerns to each SDN layer are examined in this article, including the application layer, southbound/northbound interfaces, controller, and data layers. The SDN platform's constituents, from a security perspective, possess some few weaknesses that could be exploited by attackers to carry out harmful operations, influencing the network and its operators. In summary, this work identifies architectural flaws and builds attack vectors at each layer, leading to future development in recognizing the repercussions of attacks and suggesting preventive solutions**.

***Keywords*— Software-Defined Networking, SDN, Security, Threats,5G/IMT-2020**

I. INTRODUCTION

In efforts including 5G or the Internet of Things (IoE), the software-defined network paradigm (SDN), which is dramatically revolutionizing telecommunications networks, is universally acknowledged as an influential technology. In fact, the sheer information volume, the exponential rise in the number of linked devices, and the requirement for lightning-fast data processing are all compelling arguments. SDNs have been around for more than two decades, but they are continually advancing and there are more and more requirements in the tech industry demanding dynamic, more adaptable, and more secured SDNs. [3,4]

At a compound annual growth rate (CAGR) of 28.2%, the global market for software-defined networking is expected to reach $72,630 million by 2027, up from the current estimated 2019 value of $9,995 million. The SDN market is being propelled forward by an increase in the frequency of connected endpoint devices as well as widespread use of server virtualization systems. The engagement of cloud service providers (CSPs) in SDN systems to automate network infrastructure, a considerable decrease in CAPEX and OPEX, increased consumption of cloud services, data center convergence, and server virtualization are the primary development factors for the SDN industry. They raise the standard for field-based service efficiency by increasing the requirement for enterprise flexibility. However, an SDN controller is considered an ideal assault surface, allowing hackers to gain complete access over the network in any situation. [4]

SDN's high-level design was recommended by the Open Networking Foundation [5]. An infrastructure layer, a control layer and an application layer are all constructed on top of each other in this model's three-layer architecture. The physical and virtual switches, routers, and wireless connection sites that make up the data plane constitute the majority of the infrastructure layer. Using open interfaces, the control layer, also referred as the control plane, preserves the connection between the application and the infrastructure layers.

Automatic network is administered by controller, which is the most crucial component of the network, as it is accountable for gathering and maintaining all network status data. The controller can connect with other layers through three communicating links: the southbound link, which communicates with the infrastructure layer, the northbound link, which communicates with the application layer, and the east/westbound link, which communicates with the controller units. Moreover, the application layer is created with the primary goal of meeting the demands of the end user. Network surveillance, firewalls, load balancing, intrusion identification and protection technologies, in-depth surveillance, and accessibility controls are examples of end-user business applications utilizing network services [6]. Flexible, scalability, redundancy and efficiency are just a few of the advantages of using SDN. As a result of its extensive acceptability, OpenFlow [3] is the contemporary SDN standard and has a significant accomplishment. [7]. The SDN design, on the other side, has some drawbacks [8][9]. Using these apps, intruders will be prevented from gaining access to sensitive portions of the network. Because of the inability to ensure the privacy of SDNs, their advancement will be met with considerable opposition and may even become utterly obsolete in the pathway of replacing the existing network design. This essay focuses on the security features of SDN architecture since there is ample investigation on the security issues of conventional network design. This article's main goal is to describe SDN design, security weaknesses, and attack mitigation strategies [7]. Furthermore, this research focuses on detecting and resolving security vulnerabilities on the basis of

the application layer, control layer, and infrastructure layer in the design by funneling down, inside the security elements of SDN design [5]. From the fundamental technology to the security issues that arise at each layer of the SDN architecture, the common security challenges of SDN are ultimately examined in detail. The following is how the remainder of the paper is arranged:

In the second part, we provide a quick overview of the SDN design to help with the topic of SDN security. The controller-based method to threat modeling is discussed in the third part. SDN security concerns and mitigation strategies are the topic of forth part. SDN and security defense are the subject of the fifth section of this paper, which examines nine different sorts of threats. The SDN threat is reviewed and analyzed at each stage in our sixth part. Eventually, in the seventh part, we come to a conclusion about the topic.

II. SDN ARCHITECTURE IN TERMS OF CRITICAL COMPONENTS

There is a decentralized model for the control plane implemented by conventional networks in this context. Guidelines, including ARP, STP, OSPF, EIGRP, BGP, and others function individually on each network machine. No centralized device oversees the whole network or summarizes the most significant distinction between traditional network and SDNs, although these network components are connected. SDNs, on the other hand, are often software-based, whereas conventional networks are hardware-based. SDN is more versatile because it is software-based and allows users to effectively monitor and control resources wirelessly via a control panel.

Furthermore, with the ability to decouple software from hardware, it is characterized as a new model that is fast emerging as a viable alternative to networks that are incapable to resolve the problems of conventional networks. In SDN, the hardware of a centralized software program is under the administration of a management/control. There is a complete separation between this software package and hardware. Open-source frameworks and layered structure are essential components of SDN's central infrastructure. In computer networks, software is more effective, more versatile in programming, and fosters innovation since it can be generated quickly by a variety of sellers [10]. It is possible to think of software-defined networking as an extension of network programming that tries to make networking more effective by integrating it with more productive software programs. Due to the software-partitioned isolated network method, the word "effective" is purposely employed here [3].

The following characteristics are found in SDN(i)The network control is separated from the data plane (i.e., switches and routers); (ii) A program open interface like OpenFlow, for instance, can be used to manage the plane directly; (iii) Network infrastructure characteristics and efficiency are managed by a network controller (like an SDN controller). The dynamic character of network administration and high bandwidth can both benefit from SDN [5]. It also allows for software-based network configuration adjustments, avoiding the requirement for corresponding hardware alterations. In addition, relative to conventional hardware-based network topologies, it makes it easier to implement and operate innovative programs and services.

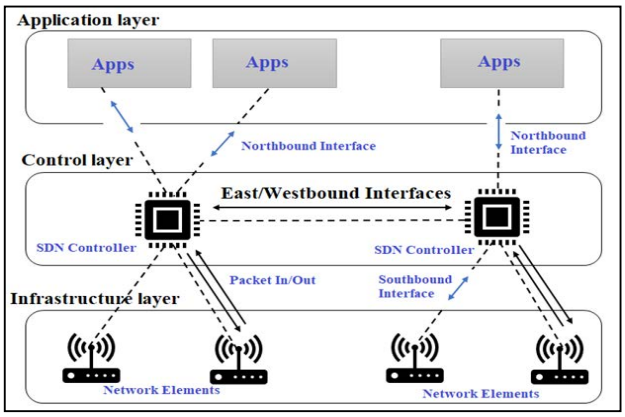

Fig. 1. SDN Architecture [8]

The northbound API, on the other side, refers to connection between the application and the controller. Nevertheless, the Northbound APIs are still lagging behind, making customized APIs more seller favorable as compared to the Southbound APIs.

The independent design is obviously depicted in Fig.1. In a design that enables eagle vision and control over the network, the forwarding machines are segregated from the controlling components. When working with physical machines that are independent from the controller [11], the stated software network design is useful. Physical machines on the network, including switches, are the only forwarding technologies that greatly minimize the intricacy of resource use and network control competence in this respect.

III. SDN IN TERMS OF SECURITY, BEHAVIOR AND WEAKNESS

SDN architecture has various benefits over conventional network design because it is in the place of building network architecture. The separated architecture, as explained in the preceding portion, provides a further level of security. A distinct architecture places the controller in a dominant position where it can have an eagle's eye view of the network and manage data transmission. There are various elements involved in establishing a network, particularly examining input packets and balancing the load on transmitting units, that must be taken into consideration. As an added benefit, relative to more conventional network architectures, the SDN architecture's centralized control point makes it faster and more responsive to network security flaws. In addition, irrespective of the size of the network, the processing of a large volume of data necessitates a strong concentration on security. Generally, in terms of its flexibility, redundancy, accessibility, scalability, and resource consumption, a network's characteristics are quantified. All of these conditions, nevertheless, will be disqualified provided the network is exposed to threats [14. A greater understanding of architectural performance from the standpoint of data organization in

networks is required to properly comprehend infrastructure in considerations of security [15].

In the event that a new flow or packet comes, several search techniques are invoked from the initial search table and result in either a matching in the flow charts or an error, depending on the requirements given by the controller. Even in the case of a single input packet, if packets are unsure of what to do using it, the baseline data for transmitting packets to the controller in the instance of a distinctive entrance is "Send to controller." Routers send event-based notifications to the controller when a link or port changes. Increasing the rule counter and, therefore, the impact of controlled activities occurs after the standards are in synchronization with the stream. This can result in a packet being sent to a specified port once some of its header fields have been modified, or (i) the packet being deleted and (ii) being reported to the controller [5]. In terms of data flow efficiency inside forwarding planes, the SDN design has a benefit compared to the conventional network design.

However, considering security must be applied manually, the SDN can enhance the administrator's workload via centralized control plane management, although this can enable superb network control. It's also easier to program networks with centralization, allowing them to be more automated and flexible. SDNs are characterized by their capacity to be easily reprogrammed by network. Challenges are inevitably uncovered when the networked system is introduced and its core functions are assigned to configurable software.

To better comprehend the shortcomings in SDN designs, the security scenario categorized in table 2 provides an assessment of all organizational levels. The in-depth investigation of the weak points inside each plane [16] [17] is disregarded in this section because the purpose of this work is to examine the security elements of the SDN design and to serve as a starting point for addressing the security vulnerabilities.

TABLE I. SDN SECURITY SCENARIO

Possible security vulnerabilities	**The reason for classifying weaknesses**
Application Layer	The vulnerability of the network-specific applications could have disastrous consequences.
Control Layer	As previously stated, if the controller's central authority is breached, it is possible to exert a significant impact on the network's stream.
Forwarding Layer	No matter how scalable, flexible, redundant, or efficient a network is, when the flow table in the forwarder's devices (data plane) is hacked, input and output data flows in the network will be misled and may even cause significant harm.

IV. LEVEL SURVEY OF NINE KIND OF ATTACKS BASED ON-SDN AND PROTECTION WAYS

The development of networks has resulted in the emergence of new types of threats, known and unknown hazards, and zero-day exploitation. As of present, there is no background of real-world SDN threats, making it difficult to identify and create security around current weaknesses. There is also the option to employ a categorization of possible threats to serve as a reference point and establish a foundation for security. The SDN design is depicted in Figure 2 as well as the vectors of potential threats (in red) [18].

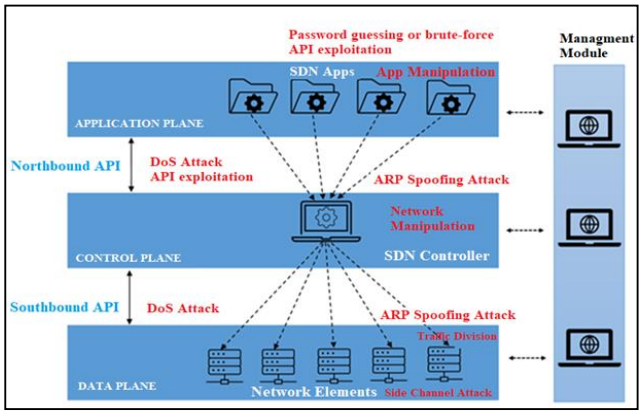

Fig. 2. Types of Attacks Based on SDN Architecture

1. Network Manipulation: A fundamental assault that takes place on the control plane. The SDN controller is damaged by an attacker, resulting in incorrectly arranged network data and various threats on the network in general.

How to protect: The SDN controller should have a redundant entity and communication routes should be encrypted strongly to protect against this threat.

2. Traffic diversion: The data plane is the target of this threat, which targets network components. In order to rearrange traffic and listen in on conversations, the assault exploits a network element.

How to protect: The use of powerful encryption to protect network components and their channels of communication.

3. Side channel attack: This threat has the potential to affect network components on the data plane. A hacker can determine if a flow criterion exists based on the time it requires for a new network link to be established, for example.

How to protect: A powerful encryption algorithm can be used to protect network components

4. App manipulation: The application plane is the target of this threat. It is possible to induce dysfunction, disruption of services, or data eavesdropping through the usage of an application weakness. An SDN application could be breached by an attacker with high privileges, allowing them to conduct illicit actions.

How to protect: Update the servers using the newest patches at all times.

5. Denial of Service "DoS": This is one of the most popular assaults, and it has the potential to disrupt any aspect of the SDN infrastructure. SDN services could be reduced or completely disrupted if an attacker employs DoS against them

How to protect: The controller plane should employ rate-limiting and packet dropping methods.

6.ARP Spoofing Attack: ARP cache-poisoning is another name for a Man-in-the-Middle threat. Infiltrating a network, sniffing traffic, modifying it, and even stopping it are all possible through the employment of ARP spoofing by hacker. The network topology data and topology aware SDN programs are both corrupted by this type of assault. Other protocols, including LLDP or IGMP, can also be used for poisoning.

How to protect: It is suggested that robust authentication techniques be used.

7. API exploitation: Data breaches in a software component's APIs could allow an attacker to gain access illegally to sensitive data. The northbound interface can potentially be exploited, resulting in the loss of network communications.

How to protect: Keep servers updated with latest patches.

8. Traffic sniffing: A hacker's favorite approach for capturing and analyzing network communications is called a sniffing threat. A hacker can also eavesdrop on information from system components or connections and embezzle valuable data by using sniffing techniques. In a place where there is a lot of traffic, sniffing may occur.

How to protect: A powerful encryption approach is used. (SSL Certificates).

9. Password guessing or brute force: A non-SDN element can be the target of this threat. Unauthorized users could get entry to the SDN by using brute force or password estimation.

How to protect: Alter default passwords from vendors, utilize powerful passwords, and upgrade them on a regular basis.

V. ANALYSIS OF SDN ATTACK BY FOCUSING ON EACH PLANE AND INTERFACE

The SDN notwithstanding its many advantages, is not a perfectly secured strategy. It must be safeguarded from numerous dangers and cyberattacks, just similar to any other architecture or construction. There are several levels of attack classification that can be used to categorize these threats. This does not imply that the SDN design is ineffective because it is continually fixing weaknesses. [6]

While discussing network vulnerabilities and other issues, the attention is mainly on the security components of the network, rather than on the prospective network operations and functionalities. SDN design outperforms conventional network architecture in terms of overcoming current challenges. Nevertheless, in order to create and secure the network, it is necessary to raise security weaknesses [19].

A. Application plane threats

Applications can govern a portion or the entire network using SDN controllers. In addition to adding novel features to the network, programs also have read and write access to the controller. As a result, it is critical that all programs are permitted and authorized. Malware, on the other hand, has the ability to breach network security by using well-known methods or by violating the standards of confidentiality and consistency.

Today, the controller developer or a third party can offer a broad range of services like firewalls, routing strategies, protocols, and so on [20].

TABLE II. SUMMARY OF SECURITY THREATS BASED ON APPLICATION LAYER

Attack Surface	Security Threat	Description of the threat
AP[1]	Lack of Unauthorized/ Unauthenticated applications	Insufficiently convincing systems for program authentication and authorization pose a concern, and the presence of numerous third-party apps heightens this threat. Without sufficient security measures, these applications can breach network resources. As a result, the identification and authorization of third-party apps running on the SDN with a centralized (logical) controller represent a significant security problem.
	Fraudulent flows rules insertion.	Applications that are malicious or hacked can establish the incorrect parameters, and it can be challenging to identify if an application has been hacked.
	Lack of access control & accountability.	Third-party applications and nested applications that use network resources pose a significant challenge in terms of implementing access control and accountability. If a hacker masquerades in an application, he/she can get and change all of the network resources [29].
	Illegal Function Calling	Malicious applications can command the controller to separate other processes, resulting in the termination or destruction of the Event Liner program because of a vulnerability in the Northbound API [30].
	Trust between applications and controller	Whenever a third-party program links to the controller via an unprotected NBI, it obtains complete authority to modify, control, or manipulate the network. Numerous threats can be launched when these vital resources are used by malicious programs, including STRIDE attacks. Establishing the integrity and dependability of third-party applications, on the other hand, is a difficult undertaking [30].
	Malicious Flow Rule Injection	When a malicious OF program uses rootkit methods to implement destructive rules and routing policies to OF switches, it does so without involving the database and trying to draw attention from the universal network operator.

B. Northbound interface threats

In responsibility of the communication between the application plane and the control plane, the NBIs are application-programming interfaces (APIs) (such as, RESTful

[1] Application Plane

APIs or Java language APIs) [29]. The standardization issue is now the largest security issue for the northbound interface [21][22]. There are no standard guidelines for permission and authentication techniques because SDN programs are always changing and evolving. The northbound interface between the control layer and the application layer is more vulnerable than the southbound interface between the control layer and the data layer. As a result, hackers can take advantage of the northbound interface's accessibility and programming software to run a threat.

TABLE III. SUMMARY OF SECURITY THREATS BASED ON NORTHBOUND INTERFACE

SDN Layer/ Attack Surface	**Security Threat**
Northbound interface (NBI)	Fraudulent rule insertion
	Code injection
	Flow rule manipulation
	Data leakage

C. Control plane threats

The programming languages and interfaces for more than 30 controllers are now available to the general public. Many are free to use, while others impose limitations. There are centralized and decentralized controller architectures [23].

Accordingly, it will be impossible for the network to actualize any untested methods. Since all controllers must be unreliable, ensuring the layout of various controllers is critical. Mitigation of DDoS attacks is also essential [24]. On the control plane, there's a problem with authorization and verification. In other words, the controller should only be accessible to approved users and apps. In addition, because SDN is centralized, the damage created by an erroneous organization can be as widespread as the entire network. As a result, it is imperative to limit any chance of the network being subjected to the incorrect rules.

TABLE IV. SUMMARY OF SECURITY THREATS BASED ON CONTROL LAYER

Attack Surface	**Security Threat**	**Description of the threat**
CP[2]	DoS, DDoS attacks	The control plane's visible character, concentrated intelligence, and restricted resources all contribute to its vulnerability to DoS assaults.
	Unauthorized controller access	There is currently no convincing technique for implementing access control on applications in existence.
	Scalability and Availability	Scalability and accessibility issues will almost certainly arise if intelligence is concentrated in a single organization.

D. Southbound interface threats

A number of protocols, including Open-Flow, OVSDB, OpFlex, NETCONF, and Strengths, connect the data plane and control plane on the southbound interface [4]. Communications breaches in the OpenFlow protocol are the responsible for the vulnerability of the southbound interface security.

The SSL/TLS protocol employed by OpenFlow to encrypt information is not safe. TLS is specified to be optional in OpenFlow 1.3.0, implying that the channel can be used without any security precautions. Following that, the southbound interface is vulnerable to hearing (eavesdropping), controller impersonation (controller forging), information leakage, and other security issues [25].

TABLE V. SUMMARY OF SECURITY THREATS BASED ON SOUTHBOUND INTERFACE

Attack Surface	**Security Threat**
Southbound interface	MITM attack
	Malicious scanning
	Packet-in messages
	Flow-mod message manipulation

E. Data plane threats

Switches and other transmission components are at the center of the data plane. All decisions taken to respond to requirements are carried out by network elements.

The data plane characteristics of SDN networks are similar to those of traditional networks. To put it another way, only authorized clients should be able to connect data plane node management and verify their own permission. Malicious conduct, on the other hand, has the potential to generate a variety of problems. For instance, previously associated rules about connection can be deleted or modified.

Another difficulty is ensuring that only approved devices are linked to the network. Different security procedures may be implemented if a damaged device is connected to the system. Data plane nodes are also vulnerable to DoS threats [26].

TABLE VI. SUMMARY OF SECURITY THREATS BASED ON DATA LAYER

Attack Surface	**Security Threat**	**Description of the threat**
DP[3]	Fraudulent flow rules	Since a data plan is essentially disposal, it's more vulnerable to flow regulations of malicious data.
	Flooding attacks	A small number of flow instructions may be stored in the OpenFlow switches' flow table at any given time.
	Controller hijacking or compromise	The data plane's security is entirely reliant on the controller's security because the data plane is entirely reliant on the control plane.
	TCP-Level attacks	TLS is vulnerable to threats at the TCP level.
	Man-in-the-middle attack (MITM)	It is because of the alternative usage of TLS and the intricacy of configuring TLS.

[2] Control Plane

[3] Data plane

VI. DESCRIPTION OF *STRIDE* THREAT MODELING METHODOLOGY BASED ON SDN TECHNOLOGY

The opponent's goal is to cause network disruption, which is referred to as an attack. Attacks can range from qualified inside developers to external attackers, from approved clients to attackers masquerading as genuine clients. The following threat classifications [27] can be used to categorize and identify potential attacks to various resources.

Every system component, including the SDN controller and its relationships with other SDN and external elements, is subject to the STRIDE. Furthermore, each component's input/output and data flows are described in detail. It is important to remember that the emphasis is on data streams and connections. In particular, the focus is on the ability of SDN components to communicate intelligently with one another and with other parties.

Many sorts of assaults are explored for each of the components following the STRIDE technique [12]. The phrase itself is an abbreviation generated from the acronyms of the six major threat types, which are outlined in the table below:

TABLE VII. MICROSOFT STRIDE ATTACK TYPES AND SECURITY PROPERTIES BASED ON SDN

Attack Type	Security Property	Types of attacks
Spoofing	Unauthorized user access	ARP spoofing. LLDP spoofing. IGMP packet. Malicious device connection. Malicious network. application injection. Faked controller connection.
Tampering	Data's integrity	Flow rule manipulation. Core services manipulation. Internal storage tempering.
Repudiation	Non-repudiation	Execution chain interrupting. Application conflict. Flow rule conflict.
Information Disclosure	Private data's confidentiality.	Vulnerability exploitation. Malicious scanning. Man-in-the-middle. Data leakage.
Denial of Service (DoS)	Availability	Packet-in flooding DoS. Code injection (Controller-level DoS). Command injection (System-level DoS). Resource exhaustion. Malicious flow attack.
Elevation of Privileges	Authorization	Priority-bypassing attack. Flow rule injection. Flow rule circuit. Buggy application affection. Zero-day attack.[29]

VII. CONCLUSION

Especially for wide-area networks (SD-WAN), next-generation network systems (5G/IMT-2020), and dynamic IoT devices, the software-defined networking strategy offers self-evident emphasis points. As of now, there are a significant number of SDN devices available on the marketplace, and SDN are widely used in developing network application situations, including cloud computing, data centers, enterprise networks, and 5G/ITM-2020, with its potential spectrum of application usage expanding as well. The SDN design also provides a virtualized network, transforming the current network into a platform that can be easily customized and programmed. SDN will become the emerging standard for networks because the development of next-generation networks becomes progressively reliant on software. SDN security vulnerabilities are growing as the number of SDN applications grows. SDN security developments are the subject of this investigation. SDN security challenges, some of which are similar to those of traditional networks and others specific to SDN, are growing in tandem with the progressive development of SDN services. The current state of SDN security development is the subject of this paper. A broader perspective on comprehending and mitigating threats has been provided by studying the many types of assaults across the various layers of the SDN network, resulting in an intriguing and broad viewpoint. Programmability and interoperability are further benefits of the independent SDN design. There will be negative repercussions when this developing network architecture fails to implement proper security mitigation mechanisms that are focused on the threats listed above. SDN architectural layers can be protected from a variety of assaults and risks in the coming years by conducting tests and comparing different approaches.

REFERENCES

[1] White Paper 5G Network Technology Architecture, IGP Group, Chennai, India, 2015

[2] Z. Lv and N. Kumar, "Software defined solutions for sensors in 6G/IoE," Comput. Commun., vol. 153, pp. 42-47, Mar. 2020.

[3] 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", Security and Communication Networks, vol. 2021.

[4] M. B. Jiménez, D. Fernández, J. E. Rivadeneira, L. Bellido and A. Cárdenas, "A Survey of the Main Security Issues and Solutions for the SDN Architecture," in IEEE Access, vol. 9, pp. 122016-122038, 2021.

[5] Nam Tuan Le, Mohammad Arif Hossain, Amirul Islam, Do-yun Kim, Young-June Choi, Yeong Min Jang, "Survey of Promising Technologies for 5G Networks", Mobile Information Systems, vol. 2016, Article ID 2676589, 25 pages, 2016.

[6] D. Melkov and S. Paulikas, "Security Benefits and Drawbacks of Software-Defined Networking," 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 2021, pp. 1-4.

[7] Karthik.S, Saravanan.M, Prabaharan.S, " Security threats and countermeasures in software defined networking ", International Journal of Emerging Technologies and Innovative Research, Vol.6, Issue 3, page no.386-391, March-2019.

[8] ON Foundation. (2014). SDN Architecture. [Online]. Available:

https://www.opennetworking.org/images/stories/downloads/sdnresources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

[9] A. Danping, M. Pourzandi, S. Scott-Hayward, H. Song, M. Winandy, and Z. Dacheng. (Jul. 2016). Threat Analysis for the SDN Architecture. [Online]. Available: https://www.opennetworking.org

[10] Haji, S. H., Zeebaree, S. R. M., Saeed, R. H., Ameen, S. Y., Shukur, H. M., Omar, N., Sadeeq, M. A. M., Ageed, Z. S., Ibrahim, I. M., & Yasin, H. M. (2021). Comparison of Software Defined Networking with Traditional Networking. Asian Journal of Research in Computer Science, 9(2), 1-18.

[11] S. Shin and G. Gu, "CloudWatcher: network security monitoring using OpenFlow in dynamic cloud networks (or: how to provide security monitoring as a service in clouds?), " in Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, TX, USA, October 2012.

[12] Threat Analysis for the SDN Architecture, https://opennetworking.org/wp-content/uploads/2014/10/Threat_Analysis_for_the_SDN_Architecture.pdf

[13] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking, " IEEE Communications Surveys & Tutorials, vol. 16, no. 4, pp. 1955–1980, 2014.

[14] P. G¨oransson and C. Black, Software Defined Network, A comprehensive approach, Morgan Kaufmann Publishers, Burlington, MA, USA, 1 edition, 2014.

[15] Boujelben, O. Cheikhrouhou, M. Abid, and H. Youssef, "Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks, " in Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications, pp. 442–448, Athens, Greece, June 2009.

[16] P. Porras, S. Shen, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks, " in HotSDN'12: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, pp. 121–126, Helsinki, Finland, August 2012.

[17] M. Kaur, D. Singh, and R. Singh Uppal, "Parallel strength pareto evolutionary algorithm-II based image encryption, "IET Image Processing, vol. 14, no. 6, pp. 1015–1026, 2020.

[18] Aayush Pradhan, Rejo Mathew,Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN), Procedia Computer Science, Volume 171, 2020,Pages 2581-2589.

[19] A. S. Alshra'a and J. Seitz, "External device to protect the software-defined network performance in case of a malicious attack, " in Proceedings of the 3rd International Conference on Future Networks and Distributed Systems, pp. 1–6, Orsay, France, July 2019.

[20] Y. Liu, B. Zhao, P. Zhao, P. Fan and H. Liu, "A survey: Typical security issues of software-defined networking," in China Communications, vol. 16, no. 7, pp. 13-31, July 2019.

[21] F. Klaedtke, G.O. Karame, R. Bifulco, et al., "Access control for SDN controllers", Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined NETWORKING, 2014, pp. 1325-1335

[22] C.R. Vasconcelos, R.C.M. Gomes, A.F.B.F. Costa, et al., "Enabling high-level network programming: A northbound API for Software-Defined Networks", Proc. 31st International Conference on Information Networking (ICOIN), 2017, pp. 662-667.

[23] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in software-defined networking (SDN)," Comput. Netw., vol. 112, pp. 279-293, Jan. 2017.

[24] I. Ahmad et al., "Security in Software Defined Networks: A Survey" in IEEE Communications Surveys & Tutorials, Vol.17, No.4, Fourth Quarter. 2015.

[25] K. Benton, L.J. Camp, C. Small, "OpenFlow vulnerability assessment", Proc. ACM SIGCOMM Workshop on Hot Topics in Software Defined NETWORKING, 2013, pp. 151-152.

[26] A. Abdou, P. C. van Oorschot, T. Wan, "Comparative Analysis of Control Plane Security of SDN and Conventional Networks" in IEEE Communications Surveys & Tutorials, Vol.20 No.4, Fourth Quarter. 2018.

[27] Andi Bidaj ,2016, "Security Testing SDN Controllers", Master thesis, Aalto University, Finland.

[28] R. K. Arbettu, R. Khondoker, K. Bayarou and F. Weber, "Security analysis of OpenDaylight, ONOS, Rosemary and Ryu SDN controllers," *2016 17th International Telecommunications Network Strategy and Planning Symposium (Networks)*", 2016, pp.37-44.

[29] Bilal Rauf, Haider Abbas, Muhammad Usman, Tanveer A. Zia, Waseem Iqbal, Yawar Abbas, and Hammad Afzal. 2021. Application Threats to Exploit Northbound Interface Vulnerabilities in Software Defined Networks. ACM Comput. Surv. 54, 6, Article 121 (July 2022), 36 pages.

[30] Seungsoo Lee, Changhoon Yoon, and Seungwon Shin. 2016. "The smaller, the shrewder: A simple malicious application can kill an entire SDN environment". In ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization. ACM, 23–28.