Hindawi
Security and Communication Networks
Volume 2021, Article ID 9971705, 7 pages
https://doi.org/10.1155/2021/9971705

Research Article

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

Nitheesh Murugan Kaliyamurthy,[1] Swapnesh Taterh,[1] Suresh Shanmugasundaram,[2] Ankit Saxena,[3] Omar Cheikhrouhou,[4] and Hadda Ben Elhadj[5,6]

[1]*Amity Institute of Information Technology, Amity University, Jaipur, India*
[2]*Faculty of Engineering and Applied Sciences, Botho University, Gaborone, Botswana*
[3]*Department of CSE, Invertis University, Bareilly, India*
[4]*College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[5]*Higher Institute of Computer Science of Mahdia, Hiboun, Tunisia*
[6]*Laboratory of Signals, Systems, Artificial Intelligence and Networks, Sfax, Tunisia*

Correspondence should be addressed to Hadda Ben Elhadj; hadda.ibnelhadj@esti.rnu.tn

Received 24 March 2021; Revised 14 April 2021; Accepted 4 May 2021; Published 12 May 2021

Academic Editor: Manjit Kaur

Software-defined networking is an evolving network architecture beheading the traditional network architecture focusing its disadvantages in a limited perspective. A couple of decades before, programming and networking were viewed as different domains which today with the lights of SDN bridging themselves together. This is to overcome the existing challenges faced by the networking domain and an attempt to propose cost-efficient effective and feasible solutions. Changes to the existing network architecture are inevitable considering the volume of connected devices and the data being held together. SDN introduces a decoupled architecture and brings customization within the network making it easy to configure, manage, and troubleshoot. This paper focuses on the evolving network architecture, the software-defined networking. Unlike a generic view on the evolving network, which makes work as a review, this work addresses various perspectives of the architecture leaving it an intermediate work in between the review of the literature and implementation, contributing towards factors like the design, programmability, security, security behaviors, and security lapses. This paper also analyses various weak points of the architecture and evolves the attack vectors in each plane leaving a conclusion to further progress towards identifying the impacts of the attacks and proposing mitigation strategies.

1. Introduction

With the increased requirements, the connected devices over a period of time suffocates in executing its operations as intended. Being many reasons stated for this condition, listing a few proven causes as such the volume of data, the exponential increase in connected devices, and the need for high-speed processing of data. In addition to all these operational factors, security is being one of the highlighted reasons throughout this scenario as it voids any mitigation proposals in the recent past [1].

This paper addresses the existing architecture of the connected devices and the recent developments held over the past decade to mitigate the suffocation. Focusing on the recent developments, there are various exciting proposals, in which this paper addresses one of the proposals which is software-defined networking (SDN). In the domain of networks, the SDN approach is considered as another trending endeavor to address the existing challenges faced in the traditional connected devices [2].

As it is one of the evolving architectures, various flaws were identified in the due course and were made open to the research forums to come up with mitigation methods. These flaws again focus on various operations within the network models. To be more precise in achieving both quantitative and qualitative progress in the work, this work funnels down

after its discussion on traditional network architecture and SDN architecture into the security aspects. As there is already adequate proven research on the security aspects of traditional network architecture, this paper limits itself to focusing on the security aspects of SDN architecture. Further funneling down, within the security aspects of SDN architecture, this work concentrates on identifying and addressing the security problems on the grounds of the application layer in the architecture [3].

Thus, this work concludes by bringing in a spotlight on one of the key security issues in the trending SDN architecture over the past decade leaving scope for further research on mitigating the issue. This key security issue is unique to the SDN architecture as it is a possibility because of the architecture's decoupled approach in dealing with the systems and its feasibility to support programmability features. The flow of this paper is structured as stated in Figure 1. It starts from introducing the evolving network architecture and its efficiency towards network programmability followed by its various security issues and factors within the SDN architecture and diving towards the vital part of this work and analyzing the security issues, its types, and impact. This paper in its final part concludes by leaving progressive pathway for other researchers to move forward and propose various solutions to mitigate the addressed security issue effectively.

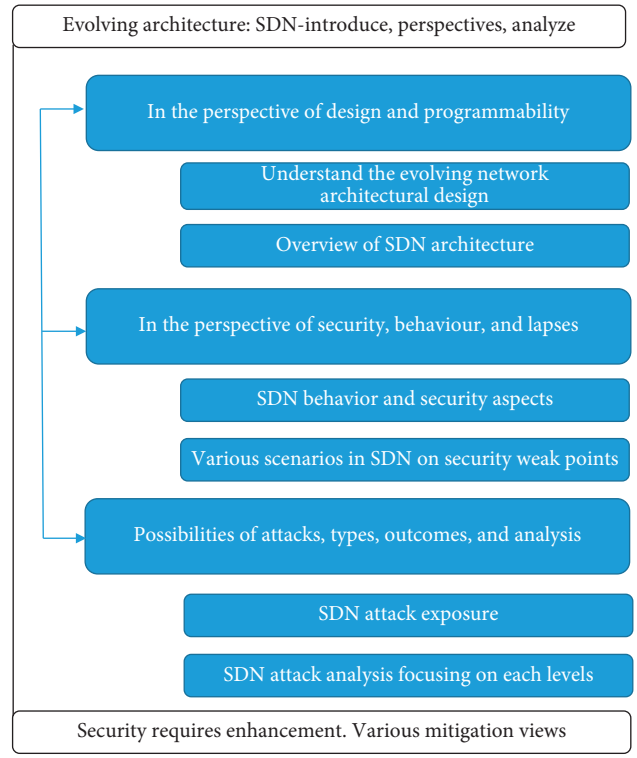

Figure 1: Diagrammatic representation of the work.

2. Evolving Architecture: In the Perspective of Design and Programmability

There are two deviant points that have to be made clearer while discussing and understanding the evolving network architecture. This section will walk through and help to gain crystal clear insights on the two aspects of evolving network architectural design. While thinking out of the box from the existing traditional network architectural (Banjar et al. [4]) design, the main focus could be to overcome the problems which are currently experienced such as exponential growth of the connected devices, the volume of data it generates, and the capacity of the devices to manage the overwhelming data which successively could be categorized within quality of service, load balancing, resource management, and Ssecurity. There are various international standards, proprietary protocols, and algorithms implemented in the existing network architecture to overcome the above-stated issues [5]. They, at one point, execute or function as anticipated overcoming the problems. However, in another dimension or perspective, they further make the network architecture more complex.

The networking domain, a couple of decades ago, was in a similar situation but for a different problem. The depletion of IPv4 addresses leads to the design of IPv6 protocol. Even though the problem was well mitigated by proposing the most secured and scalable IPv6 addressing scheme, it stills raises challenges in completely adopting IPv6 and aborting IPv4 [6]. It took over the next decade after the solution being proposed on a problem to effectively implement in the real-time operations not completely but at least to a wide level. This experience is kept in consideration while new scopes were defined to the current problems or issues being faced in the network architecture.

To overcome the complexity of the existing network architecture, a new approach is proposed, which is the network programmability. This is a feature that allows or supports programmability within a network helping to overcome resource management issues, security, and so on by programming and virtualizing [7] the network. So far before this concept, programmability and virtualization are supported within the network architecture at a limited scale. If they are already available, the difference is it was already available but not customizable. This statement helps to clarify a common myth that programming and virtualization are new concepts. Network programmability comes with a combination of various entities like the architectures, the protocols, support towards multiple programming languages and scripts, web coding tools, and the application programming interfaces. These entities help in a different order at different levels within the network to establish communication within the network devices in a comparatively simple way to the existing traditional network architecture [8].

The protocols like OpenFlow are used in network programming where it places itself as an intermediate between the programs and the forwarding devices to establish communication. Similar to the protocols, network programming also supports multiple programming languages and scripts like C, C++, Java, and Python. Along with these supportable resources, the network programming is also enriched with the web interface called REST application programming interface and with a collection of library

resources, the JAVA API (ARC). These entities add a strong core to network programming leveraging it to subdue the existing legacy traditional network architecture.

On addressing the capacity of overpowered network programming ability which exhibits a completely different face of a network domain, the other deviant is also equally placed amongst the interest of an infinite research community, which is software-defined networking. The history of software-defined networking leaves traces and tracks of various attempts which were made in over 30 years since now in overcoming the issues and complexity of traditional network architecture design. Various projects like GeoPlex [9] an initiative of ATandT and Supranet Transaction Server [10] from Ericson in the early 2000's are evidences of the traces and tracks.

Software-defined networking could be reckoned as a further enhancement of the network programming, which is intended to facilitate the network with software programs more efficiently. The statement "efficiently" is deliberately stated here because of the approach of software-defined networking's decoupled architecture. Software-defined networking architecture decouples the root of traditional network architecture [11], the control, and the data plane.

Even though various attempts were made in the past, the term software-defined networking was coined in the late 2000s and supported by open network foundation [12] since 2011. ONF is an international consortium led by a group of over 200 companies as members to formulate standards and make the new approach more viable. Similar to ONF is the Open Day Light which also focuses on ensuring various common industry standards [13]. Talking about the standards, OpenFlow is a standard proposed by ONF which helps in establishing communication between the data and the control plane. Open Stack is another software platform that focuses on cloud computing to facilitate infrastructure services as and when required [14, 15].

Figure 2 clearly states the decoupled architecture. The forwarding devices are separated from the controlling devices in the architecture facilitating an eagle-eye view and control over the network. The software-defined networking architecture works with the physical devices disassociated from the controller [16]. The physical devices, like switches in the network, are only forwarding devices that would predominantly reduce the complexity of resource utilization and controllability of the network.

The controller, as its name states, would control the network by sending instructions to all the forwarding devices based on the updated topology view and the commands executed by the applications which are customized as per the network sitting over the controller. The main purpose of the controller would be to initially configure the network, manage the network, and monitor and troubleshoot the network if required [17]. The API's help in communicating between the controller and the data plane, in this case, is OpenFlow and marked as Southbound API.

These intended operations of the controller like configuring, managing, and monitoring would be automated using customized programming features by the applications sitting above the controller in the application plane. The

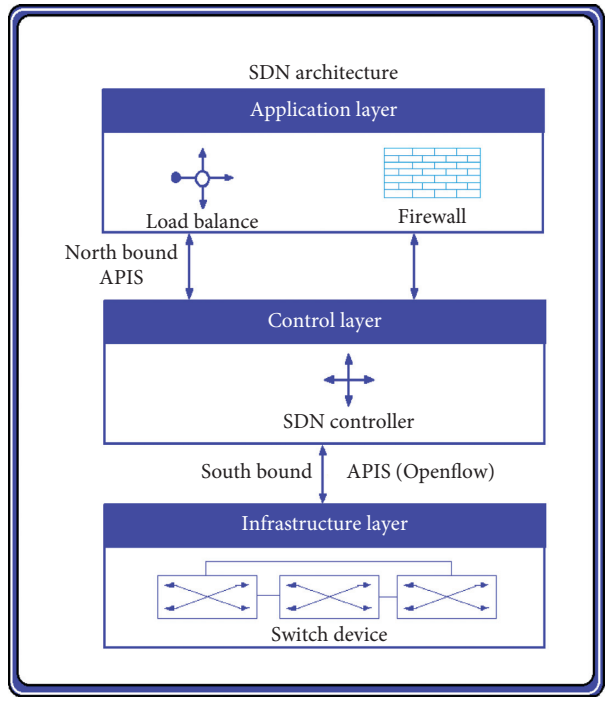

Figure 2: SDN architecture.

communication between the application and the controller is marked as northbound API. However, compared with the standards established within the Southbound API's, the Northbound APIs still lag behind and are more vendor-specific APIs.

By stating the architecture of software-defined networking and the capabilities of network programming marking with the existing traditional networking architecture, this paper has clarified the two deviant points categorizing the evolving network architecture. In the upcoming sections, the security aspects are discussed with adequate merits [18].

3. Evolving Architecture: In the Perspective of Security, Behavior, and Lapses

Being in the position of evolving network architecture, the SDN architecture brings in various advantages compared with the traditional networking architecture. As discussed in the previous section, the decoupled design itself is an added advantage in the aspect of security [19]. Because of its decoupled design, the controller places itself in a dominant position having an eagles-eye view over the network and able to control the flow of data. Controlling the data flow includes various factors of operating a network including inspecting the packets entering the network and balancing the load within the forwarding devices.

This centralized control point of SDN architecture allows to effectively respond to security flaws within the network comparing to the traditional network architecture. Focusing on security becomes a very important point irrespective of the size of the network, the volume of data being handled within the network, and so on. In a generic perspective, a

network's capabilities would be measured based on its resilience, redundancy, availability, scalability, resource utilization, and so on. However, all these metrics will be void if the network is vulnerable to attacks [20]. This brings the importance to analyze the behavior and lapses with respect to security measures within the SDN architecture.

This evolving architecture has centralized control, which could be an advantage in responding to any vulnerabilities. In the same way, the network as a whole is controlled in a central architecture, and an attack on the centralized controllers would in no time bring out the network. In this case, the advantage of SDN architecture on all aspects compared with the traditional networking architecture itself becomes a weak point in the perspective of security [21]. To understand the infrastructure more precisely in the aspect of security, the behavior of the architecture in the view of handling data within the networks needs in-depth view [22].

In SDN architecture, the packet when ingresses towards the interface matches with the forwarding device's flow table. When a successful match is found in the flow table, based on the information related to the path of the destination, the packets will flow through the network. When there is no match in the flow table with the ingress packets, they will be tagged with a "packet-in" message and will be forwarded to the control plane. The controller, based on its information received through the customized applications and the protocols, will forward the packets to the network by updating the forwarding devices' flow tables in the data plane. This behavior of SDN architecture is the advantage that overcomes the traditional network architecture by its efficient data flow within the forwarding planes.

This brings in the valid point deliberately showing multiple points within the SDN architecture where the decision of data movement depends on [23]. These are classified as different scenarios so that the lapse of the architecture in the aspect of security or, in other words, the weak points and vulnerabilities could be identified. To further address the security vulnerabilities within the architecture and to pave the way for future scope of this work entropy-based algorithms [24], machine-learning approaches and genetic algorithms [25] could be considered within the architecture.

SDN security scenario classified in Table 1 helps to understand the lapses within the SDN architectures all hierarchical levels in a briefer overview. In the study by Casado et al. [26], looking in-depth within each plane could pave a path to discuss and analyze various weak points which could be distinguished as subweak points within the different planes. Not only the weak points but also the attacks could be analyzed and determined if the lapses are identified. As this paper purposes to address the SDN architecture's security aspects and paves a pathway to address the security issues, this part skips the in-depth analysis of the weak points within each plane [27, 28].

Based on the above analysis relating to the architecture of the evolving SDN architecture, its design, programmability, behavior, and lapses, it is clearly visible that each hierarchical planes are vulnerable and are exposed to attacks which in-turn potentially reduces the overall efficiency of the architecture [29]. In the next section, the above-briefed scenario is taken, and the possibilities of attacks and its types are discussed concluding the work opening to future research contributions in mitigating the attacks and enhancing the evolving network architecture.

4. Evolving Architecture: Possibilities of Attacks, Types, Outcomes, and Analysis

The views from various existing literatures depict the current situation of the evolving network architecture. After considering the facts from the various research literatures, continuous assessments are done thoroughly in this work reckoning the architecture of software-defined networking architecture, and weak points were identified in all the hierarchical levels of the architecture. Continuously addressing the weak points does not constitute that SDN architecture is inefficient. [30]. The weak points and other discussions are majorly focused on the security aspects of the network and not the potential network operations and functions. Comparing to the traditional network architecture, the evolving SDN architecture overcomes the existing problems more efficiently. The need here is to enhance the security flaws to make the network more productive and secured [31].

In the earlier section, the identified weak points are further analyzed to look out for the possibilities of attacks, their type, and the impact that they could cause on the efficient functioning of the network. As stated during the beginning of this work, this part of the work remains the vital part discussing precisely the attacks, their types, and the impact they could cause to the entire network paving a path to further analyze each and every type of attack stated and work further to propose mitigation strategies to each type of attacks [32, 33].

In today's scenario, where attacks are peeking in a sky rate, the pandemic situations around the globe [34] also provide more flexible endurance for the attackers to succeed. A couple of years ago, today's situations like working from home and accessing cloud storage were not considered as an aspect within the infrastructure. [35]. This obviously forces the organizations to increase their budgets in infrastructures and its security. This proportionally increases the chances of the existing network architectures to move towards evolving network architecture irrespective of its size, being an enterprise, data centers, SoHo networks, and so on.

Keeping this current situation in mind, this work progresses in categorizing various types of attacks focused on this evolving network architecture. The categorization is done based on the reviews of the existing literature that addressing the types of attacks [36]. Based on the reviews, all the possible attacks in the evolving architecture are matched with the above-stated identified weak points in the architecture, and the following types or branches are arranged. They are arranged into six categories such as (1) Access Problems, (2) Data Outflow, (3) Denial of Service Attacks and Distributed Denial of Service attacks, (4) Data Alterations, (5) Misconfigurations, and (6) Malicious Applications which are the overall categories [37].

Table 1: SDN security scenario.

Possible security weak points	Reason to classify weak points
Flow table—data plane	The flow table in the forwarding devices, if compromised, will mislead the ingress and egress data flow in the network and could cause vital damage irrespective of how scalable, resilient, redundant, and efficient a network is.
Controller—control plane	The controller, as addressed earlier, being the central authority could cause a high impact over the flow of the network if compromised (in this case, we are discussing more focused on a single controller scenario; however, an SDN architecture could support distributed controllers within a network).
Applications—application plane	The applications which are customized for the network could lead to a devastating result if compromised.

Table 2: SDN attack analysis focusing on each level.

SDN architecture	Attacks vectors on each level of the SDN architecture
Data plane	- The data flow within the network could be forged and redirected - Manipulating session maintenance between the devices
Control plane	- SDN services could be denied to the network causing a denial of service/distributed denial of service - Compromised network topology information - The network could be manipulated because of its centralized and distributed controller attributes
Application plane	- Legitimate applications could be compromised and manipulated - Misconfigurations within the legitimate applications
Combination of all planes	- Majority of the attacks could be initiated using compromised trusted networks causing distributed denial of service - Sniffing the packets to gain network information
Interfaces	- Exploiting the application programming interface

These above categorized attacks are specific and could potentially make software-defined networking architecture vulnerable; however, the vulnerabilities are not limited to the above-stated attacks alone. Further adding values and contributing towards the work, few more possible attacks are listed here including compromising admin credentials, network manipulation, and man-in-the-middle attacks which might lead to activities like capturing the packets and analyzing the packets for enhanced attacks, session-related attacks, compromised applications, and the APIs. An optimized design [38] is vital to mitigate these categorized and noncategorized attacks.

If and in case the evolving network architecture fails to take appropriate security mitigation methods focusing on the above-stated attacks [39], the networks are very viable and easy to get exposed to these attack vectors. Based on the categorized and discussed attack types, to further analyze the outcome of these attacks, they are further placed over the architectures' weak points identified in the earlier section.

The SDN attack analysis described in Table 2 gives a detailed view on the overall analysis of the types of attacks, their impact, and the outcome within the SDN architecture. With a clear view of the weak points placed at different positions within an SDN architecture, it would now be a comparatively convenient approach to further classify and move out to proceed with various methods of mitigation. Various algorithms [40] at different levels for different purposes could be considered in enhancing the security within the architecture. Considering the intensity of these diversified attack methods and their scopes, more automated and advanced technologies like artificial neural network [41] approach could also be considered in effectively mitigating the weak points. This paper with above classifications made would help diverse technology researchers to showcase their skillset [42] in mitigation approach.

5. Conclusion

This work concludes after the analysis of various types of attacks, classifying them based on the architectural levels of SDN gives a broader view to understand and move forward in mitigating the attacks, thus making a unique representation. This work also underlines the attacks and their impact on the evolving network architecture that the SDN architecture is exposed to various attacks and those attacks are similar to the legacy networking architecture. This again places the research at the starting point of the problem where the evolving SDN architecture is also vulnerable to the attack vectors to which the traditional network architecture is exposed too. As a fact of analysis and thorough literature studies, it is an unfortunate yes until the first point making the statement "exposed to similar threats" true. However, that does not mean that the whole research towards the evolving network architecture is forced to come back to a point where it started because, even though both these architectures are exposed to similar kinds of attack vectors, and the evolving architecture, SDN always has an upper hand advantage in mitigating these attacks. The decoupled architecture of SDN is its advantage adding along with the programmability and interoperability features. The disadvantage of this evolving network architecture will exist if it fails to take appropriate security mitigation methods focusing on the above-stated and discussed attacks. To conclude the work, furthermore research works should focus on

identifying the operation of attacks in each attack vector focusing on the various planes and proposing an effective mitigation solution.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Omar Cheikhrouhou thanks Taif University for their support under the Taif University Researchers Supporting Project (TURSP-2020/55), Taif University, Taif, Saudi Arabia.

References

[1] O. Ben Fredj, A. Mihoub, M. Krichen, O. Cheikhrouhou, and A. Derhab, "CyberSecurity attack prediction: a deep learning approach," in *Proceedings of the 13th International Conference on Security of Information and Networks*, pp. 1–6, Turkey, November 2020.

[2] A. Derhab, M. Guerroumi, M. Belaoued, and O. Cheikhrouhou, "BMC-SDN: blockchain-based multi-controller architecture for secure software-defined networks," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9984666, , 2021.

[3] M. Kaur, D. Singh, and V. Kumar, "Color image encryption using minimax differential evolution-based 7D hyper-chaotic map," *Applied Physics B*, vol. 126, no. 9, pp. 1–19, 2020.

[4] A. Banjar, P. Pupatwibul, and R. Braun, "Comparison of TCP/IP routing versus openflow table and implementation of intelligent computational model to provide autonomous behavior," *Computational Intelligence and Efficiency in Engineering Systems, Part II*, Springer International Publishing, vol. 595, pp. 121–142, , NY, USA, 2015.

[5] S. M. AlShehri, "Software defined networking: research issues, challenges and opportunities," *Indian Journal of Science and Technology*, vol. 10, no. 29, pp. 1–9, 2017.

[6] R. Perlman, D. Eastlake, D. G. Dutt, S. Gai, and A. Ghanwani, "Routing bridges (rbridges): base protocol specification," *Technical Reports*, 2011.

[7] K. Giotis, G. Androulidakis, and V. Maglaris, "Leveraging SDN for efficient anomaly detection and mitigation on legacy networks," in *Proceedings of the 2014 Third European Workshop on Software Defined Networks*, p. 6, Budapest, Hungary, September 2014.

[8] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: past, present, and future of pro- grammable networks,," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 3, pp. 1617–1634, 2014.

[9] P. Dutta, "Internet object caching," in *Proceedings of the 7th IEEE Intelligent Network Workshop*, Bordeaux, France, May 1998.

[10] "Network Security," Network Security: http://www.networxsecurity.org/members-area/glossary/s/sdn.html.

[11] A. Doria, J. H. Salim, R. Hass et al., *Forwarding and Control Element Separation (ForCES) Protocol Specification*, Internet Engineering Task Force, Fremont, CA, USA, 2010, http://www.ietf.org/r/fc/rfc5810.txt.

[12] Open Networking Foundation, *Open Networking Foundation*, Open Networking Foundation, Menlo Park, CA, USA, 2011, https://www.opennetworking.org/about/onf-overview.

[13] "Linux foundation collaborative project," 2013, http://www.opendaylight.org.

[14] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with OpenFlow/NOX architecture," in *Proceedings of the 19th annual IEEE International Conference on Network Protocols, ICNP 2011*, pp. 7–12, Vancouver, BC, Canada, October 2011.

[15] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *CCS '13: Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, pp. 413–424, Berlin, Germany, November 2013.

[16] 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.

[17] M. Casado, T. Garfinkel, A. Akella et al., "SANE: a protection architecture for enterprise networks," in *Proceedings of the 15th USENIX Security Symposium*, vol. 15, Vancouver, B. C., Canada, July 2006.

[18] B. Nordquist, *An-Introduction-to-Software-Defined-Networking*, Storagecraft, Draper, UT, USA, 2019, https://blog.storagecraft.com/an-introduction-to-software-defined-networking/.

[19] 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.

[20] P. Göransson and C. Black, *Software Defined Network, A comprehensive approach*, Morgan Kaufmann Publishers, Burlington, MA, USA, 1 edition, 2014.

[21] Open Networking Foundation, *Principles and Practices for Securing Software-Defined Networks Version No. 1.0 ONF Document Type: TR (Technical Recommendation)*, Open Networking Foundation, Menlo Park, CA, USA, 2015, https://opennetworking.org/.

[22] M. 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.

[23] A. Shaghaghi, M. A. Kaafar, R. Buyya, and S. Jha, "Software-defined network (SDN) data plane security: issues, solutions, and future directions," in *Handbook of Computer Networks and Cyber Security*, B. Gupta, G. Perez, D. Agrawal, and D. Gupta, Eds., Springer, Cham, Switzerland, 2020.

[24] T. A. Sangeetha and G. M. Amalanathan, "Outlier detection in neutrosophic sets by using rough entropy based weighted density method," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 2, pp. 121–127, 2020.

[25] B. R. Murlidhar, R. K. Sinha, E. T. Mohamad, R. Sonkar, and M. Khorami, "The effects of particle swarm optimisation and genetic algorithm on ANN results in predicting pile bearing capacity," *International Journal of Hydromechatronics*, vol. 3, no. 1, p. 69, 2020.

[26] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: taking control of the enterprise," *ACM*

SIGCOMM Computer Communication Review, vol. 37, no. 4, pp. 1–12, 2007.

[27] 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.

[28] 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.

[29] S. Shin, L. Xu, S. Hong, and G. Gu, Enhancing network security through software defined networking (SDN)," in *Proceedings of the 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, Waikoloa, HI, USA, August 2016.

[30] G. Kannan, K. C. Meng, A literature review on Software-Defined Networking (SDN) research topics, challenges and solutions," in *Proceedings of the Fifth International Conference on Advanced Computing (ICoAC)*, pp. 293–299, Chennai, India, December 2013.

[31] 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.

[32] O. B. Fredj, O. Cheikhrouhou, M. Krichen, H. Hamam, and A. Derhab, "An OWASP top ten driven survey on web application protection methods," in *Risks and Security of Internet and Systems*, J. Garcia-Alfaro, J. Leneutre, N. Cuppens, and R. Yaich, Eds., Springer International Publishing, Cham, Switzerland, pp. 235–252, 2021.

[33] I. Jemal, O. Cheikhrouhou, H. Hamam, and A. Mahfoudhi, "Sql injection attack detection and prevention techniques using machine learning," *International Journal of Applied Engineering Research*, vol. 15, pp. 569–580, 2020.

[34] Hiscox, *Hiscox Cyber Readiness Report*, Hiscox, NY, USA, 2020.

[35] N. M. Kaliyamurthy, S. Taterh, and S. Suresh, "Vulnerability of SDN network architecture and proposed countermeasures on enhancing security," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 4, pp. 2277–3878, 2019.

[36] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proceedings of the 2010 IEEE 35th Conference on Local Computer Network Conference*, pp. 408–415, Denver, CO, USA, October 2010.

[37] K. M. Modieginyane, B. B. Letswamotse, R. Malekiana, and A. M. Abu-Mahfouz, "Software defined wireless sensor networks application opportuni- ties for efficient network management: a survey," *Computers and Electrical Engineering*, vol. 66, pp. 274–287, 2018.

[38] C. Kandilli and B. Mertoglu, "Optimisation design and operation parameters of a photovoltaic thermal system integrated with natural zeolite," *International Journal of Hydromechatronics*, vol. 3, no. 2, p. 128, 2020.

[39] X. Huang, X. Du and B. Song, An effective DDoS defense scheme for SDN," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC); 2017*, Paris, France, May 2017.

[40] C. Zhu, W. Yan, X. Cai, S. Liu, T. H. Li, and G. Li, "Neural saliency algorithm guide bi-directional visual perception style transfer," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 1–8, 2020.

[41] M. Safa, M. Ahmadi, J. Mehrmashadi et al., "Selection of the most influential parameters on vectorial crystal growth of highly oriented vertically aligned carbon nanotubes by adaptive," *International Journal of Hydromechatronics (IJHM)*, vol. 3, no. 3, pp. 238–251, 2020.

[42] Z. Ali and T. Mahmood, "Complex neutrosophic generalised dice similarity measures and their application to decision making," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 2, pp. 78–87, 2020.