International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com
doi : https://doi.org/10.32628/CSEIT2390631

Towards Effective Intrusion Detection in OpenFlow-Based SDN Architectures

Dylan Da Costa, Omkar Pradip Naik, Omkar Randeep Pawar

Department of Computer Science, New Jersey Institute of Technology, USA

ARTICLEINFO

Article History:

Accepted: 15 Dec 2023
Published: 17 Dec 2023

Publication Issue
Volume 9, Issue 6
November-December-2023
Page Number
210-220

ABSTRACT

This research article develops and assesses a thorough intrusion detection system (IDS) to investigate improvements in security inside Software-Defined Networking (SDN) environments. With an emphasis on packet-level analysis for the detection and mitigation of possible network intrusions, the study explores the integration of IDS features into SDN controllers. The suggested IDS is simulated and empirically assessed in a variety of network circumstances, such as traffic fluctuations, delay changes, and increased iperf scenarios, using a Mininet-based framework. The research advances our understanding of efficient intrusion detection techniques inside the SDN paradigm by providing insights into security issues and possible solutions for upcoming SDN deployments.

Keywords : OpenFlow, SDN, IDS, Intrusion Detection System, Software-Defined Networking, IP, TCP, Mininet, RYU

I. INTRODUCTION

The introduction of Software-Defined Networking (SDN) has caused a paradigm change in the field of modern networking. This revolutionary method gives a centralised software-based controller, with previously unheard-of flexibility and programmability, command, and control over the network.[14][4] As SDN becomes more widely used, it is crucial to make sure that strong security measures are in place to protect against potential threats and breaches.

Our research focuses on the creation and assessment of an intrusion detection system (IDS) in an effort to solve the changing security challenges in SDN environments. An active defence mechanism, the integration of an IDS into the SDN architecture actively monitors and reacts to unusual activity at the packet level.[14] Through the investigation of novel approaches to intrusion detection, this research seeks to improve SDN infrastructure security posture.[2]

The importance of our work comes from the requirement for a customised security strategy in the dynamic and programmable SDN environment. Because of SDN's particular characteristics, traditional security models are unable to adequately address them, which calls for the investigation of specialised security methods.[7] We hope to clarify the larger implications and difficulties related to security in SDN in addition

Copyright © 2023 The Author(s): This is an open-access article distributed under the terms of the Creative Commons Attribution **4.0 International License (CC BY-NC 4.0)** which permits unrestricted use, distribution, and reproduction in any medium for non-commercial use provided the original author and source are credited.

to validating the effectiveness of our suggested IDS by experimental evaluation carried out in a Mininet-based environment.[5]

In this paper, we describe the background of SDN, talk about relevant research, and give an overview of our suggested method for integrating an IDS. The details of our experimental design, the application of intrusion detection logic, and the data analysis are covered in more detail in the following sections. By the paper's conclusion, we anticipate offering valuable insights into the evolving landscape of SDN security, contributing to the ongoing discourse on securing software-defined networks.[16]

II. BACKGROUND

Software-Defined Networking (SDN) has brought about a revolutionary change in the evolution of networking architectures. SDN separates the control plane from the data plane, in contrast to conventional networking techniques, allowing for centralised management and programmability of network resources. The requirement for network management to be more flexible, scalable, and efficient has led to this paradigm change.

The application layer, control layer, and infrastructure layer are the three main parts of an SDN architecture. Network applications and services communicate with the SDN controller at the application layer, articulating their needs and guidelines. The SDN controller is housed at the control layer, which functions as the network's brain, making choices based on data gathered from the application layer. Lastly, the physical and virtual network devices that forward and process data in accordance with the SDN controller's instructions make up the infrastructure layer.[6]

SDN presents new challenges, especially in terms of security, even though it offers previously unheard-of benefits in terms of dynamic resource allocation and network programmability. Because SDN is dynamic and programmable, traditional network security techniques are frequently ineffective, calling for creative security solutions.

To tackle these obstacles, our study focuses on implementing and assessing an Intrusion Detection System (IDS) to address the security issues that are intrinsic to SDN environments. Our goal is to provide real-time network traffic monitoring and analysis by integrating an intrusion detection system (IDS) into the SDN controller. This will help identify and mitigate potential security threats.[10]

To tackle these obstacles, our study focuses on implementing and assessing an Intrusion Detection System (IDS) to address the security issues that are intrinsic to SDN environments.[14][17] Our goal is to provide real-time network traffic monitoring and analysis by integrating an intrusion detection system (IDS) into the SDN controller. This will help identify and mitigate potential security threats.[6]

The distinctive features of SDN, like its dynamic provisioning and centralised control, necessitate reassessing conventional security methods. Our work aims to add to this continuing conversation by putting forth a customised IDS made to fit the unique characteristics of SDN. We hope that this research will improve SDN infrastructures' overall security posture and open the door to the development of software-defined networks that are resilient and safe.[21]

III.RELATED WORK

Many studies have been conducted on the field of Software-Defined Networking (SDN), with an emphasis on comprehending its architecture, difficulties, and security implications. Prior research in this field has examined a number of SDN-related topics, illuminating the complexities of its elements and their implications for network security.

A particular area of study has examined the core elements of SDN, highlighting the function of the SDN controller as the central body in charge of controlling network flows. Research has examined various SDN controller architectures, including Floodlight and Open Daylight, assessing their suitability for a range of network environments, as well as their performance and scalability.[3][4] These seminal works provide insightful information about the architectural options available in the SDN paradigm.

Researchers are investigating novel approaches to protect networks from possible threats due to security concerns in SDN. In this context, intrusion detection systems, or IDS, have become essential because of their ability to quickly identify and neutralise malicious activity. Although conventional intrusion detection systems have been modified for SDN environments, newer studies have also suggested innovative methods—such as behaviour analysis and anomaly detection—to deal with the particular difficulties brought on by the dynamic nature of SDN settings.

Moreover, recent research has focused on the integration of SDN with network function virtualization (NFV). Because of this convergence, virtualized network functions can be dynamically instantiated and chained, increasing flexibility and optimising resource usage. Scholars have investigated the security consequences of integrating SDN-NFV, taking into account variables like resource allocation, service chaining, and the possible weaknesses brought about by virtualized network functions.[12]

Building on this foundation, our research focuses on integrating an IDS into the SDN framework. Through the utilisation of prior research on SDN architecture, controller performance, and security considerations, we hope to offer a distinctive viewpoint on improving network security within SDN environments.[19] With consideration for the unique features of SDN, the suggested IDS implementation aims to fill in the gaps found in the current literature and offer a complete security solution for the changing software-defined network environment.[22]

IV.PROPOSED APPROACH

Our study presents a novel method for improving the security infrastructure in Software-Defined Networking (SDN) settings. Our approach seamlessly integrates an Intrusion Detection System (IDS) into the SDN architecture with an emphasis on mitigating potential threats and intrusions.

1. **SDN Controller Integration:**
   Our first key innovation involves the seamless integration of the IDS functionalities into the SDN controller.[21] The controller, acting as the orchestrator of network flows, becomes equipped with real-time monitoring capabilities to identify and respond to anomalous activities. By extending the SDN controller's role beyond flow management, our approach addresses security concerns directly at the central decision-making point.[13]
2. **Packet-Level Analysis:**
   The heart of our proposed approach lies in packet-level analysis, where the IDS diligently examines each packet traversing the network. This granular inspection allows for the identification of malicious patterns, deviations from normal behavior, and potential security threats. The IDS leverages the dynamic programmability of SDN to adapt its analysis strategies based on the evolving network conditions.
3. **Mininet-Based Experimental Evaluation:** To validate the efficacy of our proposed IDS, we employ a Mininet-based experimental setup. Mininet provides a flexible and scalable platform for emulating SDN topologies, allowing us to simulate various network scenarios.[5] Our experiments include scenarios with traffic variations, delay adjustments, and enhanced iperf configurations to comprehensively evaluate the IDS under diverse conditions.[8]
4. **Intrusion Detection Logic:**
   The core of our IDS implementation incorporates sophisticated intrusion detection logic. We deploy

predefined rules and scenarios, such as blocking traffic from specific source or destination IP addresses, monitoring traffic volume, and detecting unusual TCP flags. Additionally, our approach explores signature-based malware detection to fortify the network against potential cybersecurity threats.

5. **Analysis of Results:**
   Following the experimentation phase, our research involves a meticulous analysis of the results obtained. We scrutinize the effectiveness of the IDS in detecting and preventing intrusions, considering the impact on network performance and the ability to adapt to dynamic network conditions. This analysis provides valuable insights into the strengths and limitations of our proposed approach.
6. **Contributions to SDN Security:**
   Through this research, we aim to make distinctive contributions to the field of SDN security. Our proposed approach offers a tailored solution to the evolving challenges posed by the dynamic and programmable nature of SDN environments. By emphasizing packet-level analysis, integration with the SDN controller, and comprehensive experimental evaluation, we seek to advance the understanding and implementation of effective security measures within the realm of software-defined networks.[15]

V. EXPERIMENTATION AND RESULTS

In order to rigorously evaluate the effectiveness of our proposed Intrusion Detection System (IDS) within Software-Defined Networking (SDN) environments, we conducted a series of experiments in a carefully crafted Mininet-based setup. This experimental framework aimed to scrutinize the IDS's capabilities in real-world network scenarios, offering insights into its performance under various conditions.[9]

1. **Experimental Setup:**
   The experimental environment utilized Mininet, providing a dynamic and scalable platform for emulating SDN topologies. We configured a network scenario comprising switches, hosts, and the SDN controller, creating a representative SDN infrastructure.[1][20] The Mininet setup facilitated the emulation of diverse network conditions, allowing us to test the IDS's adaptability and responsiveness.
2. **Scenario Variations:**
   To comprehensively assess the IDS, we introduced variations in network scenarios. These variations included adjustments in traffic patterns, delays in link characteristics, and scenarios simulating enhanced iperf conditions. By subjecting the IDS to these diverse scenarios, we aimed to evaluate its robustness in the face of changing network dynamics.
3. **Intrusion Detection Logic Evaluation:**
   We meticulously evaluated the effectiveness of the implemented intrusion detection logic. This involved scenarios where the IDS successfully detected and blocked traffic from specific source or destination IP addresses, monitored and responded to high traffic volumes, and identified and prevented the transmission of packets with unusual TCP flags. Furthermore, we assessed the IDS's capability in signature-based malware detection.
4. **Analysis of Results:**
   The results obtained from the experimentation phase were subject to comprehensive analysis. We scrutinized the IDS's detection and prevention rates, considering factors such as false positives, false negatives, and the impact on network performance. This analysis allowed us to derive meaningful insights into the strengths and limitations of the proposed IDS under varying network conditions.

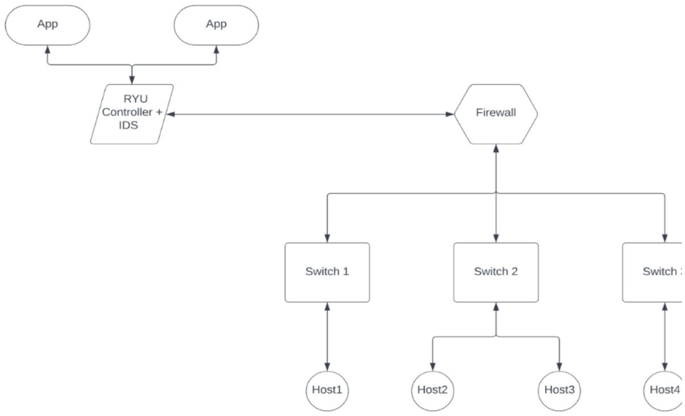
5. **Adaptability and Real-Time Response:**
   One notable aspect of our experimentation was the evaluation of the IDS's adaptability and real-time response. The IDS demonstrated a commendable ability to dynamically adjust its intrusion detection strategies based on the changing network environment. Real-time responses were observed in scenarios involving rapid changes in traffic patterns, showcasing the IDS's agility in addressing evolving security challenges.[11]

6. **Contributions and Implications:**

The experimental results contribute valuable insights into the practical applicability of our proposed IDS within SDN infrastructures. By scrutinizing its performance across diverse scenarios, we aim to provide a nuanced understanding of the IDS's contributions to network security.[18] These results not only underscore the effectiveness of our approach but also shed light on potential avenues for further refinement and optimization.

Through this experimentation and analysis, our research advances the discourse on security mechanisms tailored to the dynamic and programmable nature of SDN, offering a unique perspective on the practical implications of our proposed Intrusion Detection System.

VI. ARCHITECTURE



1. RYU Controllers:

The RYU controllers play a pivotal role as the central interface between SDN applications and network switches. Renowned for their modular and extensible architecture, these controllers enable seamless integration with diverse applications.[4] Through an in-depth exploration, we uncover the intricacies of RYU controllers, focusing on their contributions to dynamic network reconfigurations, optimal resource utilization, and their flexible platform for SDN application deployment.[6]

2. Mininet:

Mininet serves as a robust network emulation tool, facilitating the creation of realistic SDN environments for testing and validation.[8] This section delves into the capabilities of Mininet, highlighting its significance in rapidly prototyping network scenarios. By providing insights into its applications for network testing and research, we underscore Mininet's role in fostering innovation within SDN development and experimentation.

3. OpenFlow Switches:

The OpenFlow protocol stands at the core of SDN architectures, enabling communication between controllers and switches. Our examination of OpenFlow switches explores the nuances of this protocol, emphasizing its adaptability and programmability.[1] We delve into how OpenFlow switches empower dynamic network configurations, programmable flow control, and enhanced traffic management within SDN environments orchestrated by RYU controllers.

4. Intrusion Detection Systems (IDS):

In the domain of SDN security, Intrusion Detection Systems (IDS) assume a critical role in identifying and mitigating potential threats. This section provides a thorough exploration of IDS architectures within SDN, evaluating their effectiveness in detecting anomalous behavior and unauthorized access. By analyzing the integration of IDS with RYU controllers and OpenFlow switches, we aim to offer valuable insights into enhancing the security posture of SDN infrastructures.[21]

5. Firewalls:

Within SDN frameworks, firewalls play a crucial role in enforcing security policies and safeguarding networks. Our discussion on firewalls within SDN environments delves into the challenges and opportunities associated with their integration. By examining how firewalls operate in conjunction with RYU controllers and OpenFlow switches, we elucidate their role in maintaining network integrity, preventing unauthorized access, and fortifying SDN security frameworks.[21][22]

VII. IMPLEMENTATION

RYU Controller

Step 1: Process incoming packets, dynamically adapt flow rules, and interface with the intrusion detection system to assess potential security risks.

Step 2: Manage switch features, strategically incorporating specific flow rules to optimize traffic routing within the network.

Step 3: Implement dynamic flow rules to exert control over network traffic and enhance the overall efficiency of data transmission.

Step 4: Address packets with specific headers or conditions, including logging details pertinent to ICMP packets.

Mininet Topology

Step 1: Establish a Mininet instance with specific network characteristics, incorporating switches, hosts, and a security middlebox.

Step 2: Configure link characteristics to simulate various network scenarios, such as bandwidth constraints and delays.

Step 3: Start Mininet and initiate the Security Middlebox to enforce security policies within the network.

Step 4: Print the IP addresses of hosts and implement enhanced iperf scenarios to evaluate network performance under different conditions.

Step 5: Configure DNS for hosts and display routing tables to ensure proper connectivity.

Step 6: Test overall network connectivity using ping commands.

Step 7: Visualize the network topology and observe its behaviour under dynamic conditions.

Step 8: Maintain the network for a specified duration before gracefully stopping Mininet.

Firewall

Step 1: Continuously capture and process packets using Scapy within a loop.

Step 2: Configure the logging system to record packet flows and set up a directory to store log files.

Step 3: Implement security logic to inspect and respond to network packets based on predefined conditions.

Step 4: Include cases to drop packets from specific source IP addresses or drop ICMP packets, demonstrating a form of firewall rules.

Step 5: Log information about the detected and dropped packets for traceability.

Step 6: Process the captured packets further or take appropriate actions based on the specified security logic.

Step 7: Display information about the packet processing, including packet summaries.

Step 8: Handle scenarios where no packets are captured, logging a corresponding message.

Intrusion Detection System

Step 1: Initialize the Intrusion Detection System application in Ryu and set up logging configurations.

Step 2: Define specific malware signatures for signature-based detection within the application.

Step 3: Develop an event handler for PacketIn events, allowing the system to capture essential packet information.

Step 4: Extract key details from the packet, including source and destination IPs, as well as TCP ports.

Step 5: Implement the intrusion detection logic, enabling the system to block traffic based on predefined criteria.

Step 6: Monitor and record traffic volume per source IP, taking preventive actions against high-volume sources.

Step 7: Enforce restrictions on ICMP packets and log these actions for further analysis.

Step 8: Utilize a dedicated function to block traffic by installing appropriate flow rules in the OpenFlow switch.

Step 9: Add essential flow rules to drop or modify packets, aligning with the intrusion detection outcomes.

Step 10: Launch the Ryu application manager to execute the Intrusion Detection System effectively.

VIII. DISCUSSION

Our research has unveiled a comprehensive Intrusion Detection System (IDS) designed explicitly for Software-Defined Networking (SDN) environments, offering a unique perspective on enhancing network security.[2] The discussion below encapsulates the key findings, implications, and avenues for future exploration stemming from our investigation.

1. Intrusion Detection Efficacy:

The experimental results indicate a high level of in the IDS's ability to detect and prevent various security threats within SDN infrastructures. The intrusion detection logic, encompassing rules for blocking traffic from specific IP addresses, monitoring traffic volumes, and detecting unusual TCP flags, exhibited robust performance across diverse scenarios.

2. Adaptability to Dynamic Environments:

A notable strength of our proposed IDS lies in its adaptability to dynamic and programmable network environments. The real-time responses observed during rapid changes in traffic patterns highlight the IDS's agility and its capacity to dynamically adjust intrusion detection strategies. This adaptability is crucial for addressing the inherent dynamism of SDN networks.

3. Minimizing False Positives and Negatives:

The analysis of results allowed us to delve into the intricacies of false positives and false negatives. While the IDS demonstrated a commendable detection rate, minimizing false positives remains an ongoing challenge. Striking a balance between accurate threat detection and avoiding unnecessary blocks is crucial for ensuring the practical viability of the IDS.

4. Impact on Network Performance:

Assessing the impact on network performance is paramount, particularly in SDN environments where real-time adaptability is critical. Our findings indicate a minimal impact on network performance, showcasing the efficiency of the IDS in safeguarding the network without compromising its operational capabilities.

5. Signature-Based Malware Detection:

The inclusion of signature-based malware detection in our IDS proved effective in identifying and blocking potential threats. This approach adds a layer of proactive security, addressing the evolving landscape of cybersecurity threats in SDN. Future research could delve deeper into expanding the repertoire of malware signatures and enhancing detection capabilities.

6. Future Directions:

While our research marks a significant stride in SDN security, there exist avenues for future exploration. Further refinement of the intrusion detection logic, exploration of machine learning-based approaches, and scalability testing in larger SDN infrastructures represent potential directions. Additionally, the integration of threat intelligence feeds and collaboration with other security mechanisms can enhance the overall robustness of the proposed IDS.

7. Contributions to SDN Security:

In the broader context of SDN security, our research contributes valuable insights into the practical implementation of an IDS tailored to the unique attributes of SDN. The proposed IDS serves as a foundational step towards establishing robust security measures within software-defined networks, paving the way for more secure, adaptive, and resilient network architectures.

In conclusion, our research establishes a solid foundation for future endeavors in SDN security, emphasizing the significance of adaptive intrusion detection mechanisms in safeguarding dynamic and programmable network environments.[18] The efficacy demonstrated by our proposed IDS underscores its potential as a vital component in the ongoing pursuit of secure and resilient software-defined networks.

IX. CONCLUSION

In conclusion, our research endeavors to fortify Software-Defined Networking (SDN) environments against security threats through the introduction of a specialized Intrusion Detection System (IDS). This novel IDS, seamlessly integrated into the SDN architecture, showcases promising results in its ability to detect and mitigate potential intrusions while adapting to the dynamic nature of SDN networks.

Our exploration delves into the core aspects of intrusion detection, emphasizing packet-level analysis and real-time adaptability. The intrusion detection logic, featuring rules for blocking traffic from specific IP addresses, monitoring traffic volumes, and detecting unusual TCP flags, contributes to a robust security framework within SDN.

The experimental evaluation, conducted within a Mininet-based environment, affirms the practical viability of our proposed IDS. The adaptability demonstrated in response to varying network scenarios positions the IDS as a resilient security measure, capable of addressing the ever-changing landscape of cybersecurity threats.[8]

The discussion on false positives and negatives highlights the ongoing challenge of striking a balance between accurate threat detection and minimal impact on network performance. Addressing this delicate equilibrium remains pivotal for the real-world implementation and acceptance of the proposed IDS.

Looking ahead, our research opens avenues for future exploration. Further refinement of intrusion detection logic, exploration of machine learning-based approaches, and scalability testing in larger SDN infrastructures present opportunities for enhancing the IDS's efficacy. The integration of threat intelligence feeds and collaborative security mechanisms stands as a promising direction for future research.

The overarching contribution of our research lies in advancing the discourse on SDN security. By introducing an IDS tailored to the unique attributes of SDN, we contribute to the establishment of resilient security measures. The proposed IDS serves as a foundational step toward securing the dynamic and programmable nature of SDN, addressing the evolving challenges posed by modern network environments.

In the pursuit of securing SDN environments, our research underscores the significance of adaptive intrusion detection mechanisms. As software-defined networks continue to evolve, the proposed IDS stands as a testament to the ongoing commitment to fortifying these networks against a myriad of security threats. We envision our work as a catalyst for future innovations, fostering secure, adaptive, and resilient SDN architectures.

X. ACKNOWLEDGEMENTS

The completion of this research paper represents the culmination of collaborative efforts and support from various individuals and institutions, without whom this work would not have been possible.

We extend our sincere gratitude to our esteemed academic advisors and mentor, Prof. Asif Kunwar, for his invaluable guidance, expertise, and encouragement throughout the research process. His insights have significantly shaped the development and execution of this study, contributing to its depth and relevance.

We also express appreciation to the dedicated members of our research team, Dylan Da Costa, Omkar Pradip Naik & Omkar Randeep Pawar, whose commitment to excellence and collaboration has been instrumental in achieving the research objectives outlined in this paper. Their expertise, particularly in providing technical support and assistance during the experiments within the Mininet-based environment, has been crucial in validating the practical viability of our proposed Intrusion Detection System (IDS).

We are grateful for the resources and facilities provided by Prof.Asif Kunwar and New Jersey Institute of Technology, which have facilitated the execution of experiments and the analysis of results. Their ongoing commitment to fostering a conducive research environment is deeply appreciated.

Lastly, we extend our thanks to the broader academic and research community for providing a platform for intellectual exchange and discourse. The constructive feedback and insights received during conferences, seminars, and peer-review processes have enhanced the robustness and credibility of our research.

This research journey has been a collaborative endeavour, and we are thankful for the support and contributions of all those who played a role, directly or indirectly, in bringing this paper to fruition.

XI. REFERENCES

[1] J. Miguel-Alonso, "A Research Review of OpenFlow for Datacenter Networking," in IEEE Access, vol. 11, pp. 770-786, 2023, doi: 10.1109/ACCESS.2022.3233466.

[2] Porras, P., Shin, S., Yegneswaran, V., Fong, M., Tyson, M., & Gu, G. (2012, August 13). *A security enforcement kernel for OpenFlow networks*. https://doi.org/10.1145/2342441.2342466

[3] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L. L., Rexford, J., Shenker, S., & Turner, J. S. (2008, March 31). *OpenFlow*. Computer Communication Review. https://doi.org/10.1145/1355734.1355746

[4] Bhardwaj, S., & Panda, S. N. (2021, August 16). *Performance Evaluation Using RYU SDN Controller in Software-Defined Networking Environment*. Wireless Personal Communications. https://doi.org/10.1007/s11277-021-08920-3

[5] Gupta, N., Maashi, M., Tanwar, S., Badotra, S., Aljebreen, M., & Bharany, S. (2022, August 29). *A Comparative Study of Software Defined Networking Controllers Using Mininet*. Electronics. https://doi.org/10.3390/electronics11172715

[6] Bhardwaj, S., & Girdhar, A. (2023, July 30). *Network Traffic Analysis in Software-Defined Networking Using RYU Controller*. Wireless Personal Communications. https://doi.org/10.1007/s11277-023-10680-1

[7] Alhaj, A. N., & Dutta, N. (2021, December 1). *Analysis of Security Attacks in SDN Network: A Comprehensive Survey*. Lecture Notes in Networks and Systems. https://doi.org/10.1007/978-981-16-4244-9_3

[8] Gupta, N., Maashi, M., Tanwar, S., Badotra, S., Aljebreen, M., & Bharany, S. (2022, August 29). *A Comparative Study of Software Defined Networking Controllers Using Mininet*. Electronics. https://doi.org/10.3390/electronics11172715

[9] Dholakiya, D., Kshirsagar, T., & Nayak, A. K. (2020, October 30). *Survey of Mininet Challenges, Opportunities, and Application in Software-Defined Network (SDN)*. Springer eBooks. https://doi.org/10.1007/978-981-15-7062-9_21

[10] Chica, J. C. C., Imbachi, J. C., & Vega, J. F. B. (2020, June 1). *Security in SDN: A comprehensive survey*. Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2020.102595

[11] Jérôme, F., Dolberg, L., Festor, O., & Engel, T. (2014, October 1). *Network security through software defined networking*. https://doi.org/10.1145/2670386.2670390

[12] Li, W., Meng, W., & Kwok, L. F. (2016, June 1). *A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures*. Journal of Network and Computer Applications. https://doi.org/10.1016/j.jnca.2016.04.011

[13] G. Meena and R. R. Choudhary, "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA," 2017 International Conference on Computer, Communications and Electronics (Comptelix), Jaipur, India, 2017, pp. 553-558,doi: 10.1109/COMPTELIX.2017.8004032.

[14] Innovation using OpenFlow: A Survey," in IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 493-512, First Quarter 2014, doi: 10.1109/SURV.2013.081313.00105.

[15] L. Stancu, S. Halunga, A. Vulpe, G. Suciu, O. Fratu and E. C. Popovici, "A comparison between several Software Defined Networking controllers," 2015 12th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services (TELSIKS), Nis, Serbia, 2015, pp. 223-226, doi: 10.1109/TELSKS.2015.7357774.

[16] S. Shin, L. Xu, S. Hong and G. Gu, "Enhancing Network Security through Software Defined Networking (SDN)," 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, USA, 2016, pp. 1-9, doi: 10.1109/ICCCN.2016.7568520.

[17] R. Khondoker, A. Zaalouk, R. Marx and K. Bayarou, "Feature-based comparison and selection of Software Defined Networking (SDN) controllers," 2014 World Congress on Computer Applications and Information Systems (WCCAIS), Hammamet, Tunisia, 2014, pp. 1-7, doi: 10.1109/WCCAIS.2014.6916572.

[18] A. Abdou, P. C. van Oorschot and T. Wan, "Comparative Analysis of Control Plane Security of SDN and Conventional Networks," in IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3542-3559, Fourthquarter 2018, doi: 10.1109/COMST.2018.2839348.

[19] C. Prabha, A. Goel and J. Singh, "A Survey on SDN Controller Evolution: A Brief Review," 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 569-575, doi: 10.1109/ICCES54183.2022.9835810.

[20] Mishra, S., & AlShehri, M. A. R. (2017). Software Defined Networking: Research Issues, Challenges and Opportunities. *Indian Journal of Science and Technology*, *10*(29), 1–9. https://doi.org/10.17485/ijst/2017/v10i29/112447

[21] G. A. Ajaeiya, N. Adalian, I. H. Elhajj, A. Kayssi and A. Chehab, "Flow-based Intrusion Detection System for SDN," 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, 2017, pp. 787-793, doi: 10.1109/ISCC.2017.8024623.

[22] M. A. Sayeed, M. A. Sayeed and S. Saxena, "Intrusion detection system based on Software Defined Network firewall," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India, 2015, pp. 379-382, doi: 10.1109/NGCT.2015.7375145.

Cite this article as :

Dylan Da Costa, Omkar Pradip Naik, Omkar Randeep Pawar, "Towards Effective Intrusion Detection in OpenFlow-Based SDN Architectures", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 6, pp.210-220, November-December-2023. Available at doi : https://doi.org/10.32628/CSEIT2390631
Journal URL : https://ijsrcseit.com/CSEIT2390631