Implementation of ICMP flood detection and mitigation system based on software-defined network and sFlow-RT

Rikie Kartadie[1], Adi Kusjani[2], Rangga Warsito[3], Yudhi Kusnanto[3], Lucia Nugraheni Harnaningrum[4]
[1]Department of Computer Engineering, Faculty of Information Technology, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia
[2]Computer Engineering Vocational Program, Faculty of Information Technology, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia
[3]Department of Informatics, Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
[4]Department of informatics, Faculty of Information Technology, Universitas Teknologi Digital Indonesia, Yogyakarta, Indonesia

Article Info

Article history:

Received Apr 23, 2024
Revised Mar 24, 2025
Accepted May 10, 2025

Keywords:

Internet control message protocol flood
Network security
OpenFlow
Software-defined networks
sFlow-RT

ABSTRACT

This study evaluates internet control message protocol (ICMP) flood detection and mitigation in software-defined networks (SDN) using an SDN architecture with sFlow-RT for real-time traffic monitoring. OpenFlow switches and sFlow agents detect malicious patterns, following the prepare, plan, design, implement, operate, optimize (PPDIOO) methodology. Unlike prior approaches, this system leverages SDN programmability and sFlow-RT's real-time analytics to reduce ICMP packets from 311,130.2 to 99 and latency by 80%, outperforming traditional methods in speed and responsiveness. It ensures network availability, with practical benefits for large-scale networks like internet service providers (ISPs). However, sFlow sampling rates may affect accuracy in high-speed networks, and a single OpenDaylight (ODL) controller limits generalizability. Future work should test alternative controllers and extend to other DDoS types like user datagram protocol (UDP) floods in diverse topologies.

Corresponding Author:

Rikie Kartadie
Department of Computer Engineering, Faculty of Information Technology
Universitas Teknologi Digital Indonesia
St. Raya Janti Karang Jambe No. 143, Yogyakarta 55198, Indonesia
Email: rikie@utdi.ac.id

1. INTRODUCTION

The rapid development of the internet increases the need for network security, especially against distributed denial-of-service (DDoS) attacks such as internet control message protocol (ICMP) floods that disrupt service availability. Computer networks continue to evolve rapidly, with the internet being a major driver of this growth. As of 2021, there were 5.38 billion internet users globally, reflecting a 67.9% growth rate. In Indonesia, this number was projected to reach 212 million by 2022, marking a 76.3% increase [1]. Effective network management is essential to ensure high availability and security. One critical aspect of risk assessment is evaluating the impact of potential counterattacks, particularly in mitigating DDoS attacks.

DDoS attacks exploit architectural weaknesses in network security, particularly the separation of data and control planes, as noted by Sangodoyin *et al.* [2]. These attacks disrupt legitimate access to network resources, causing significant delays. This study examines DoS attack impacts on software-defined networks (SDNs) using Mininet, OpenDaylight (ODL), and network measurement tools like iPerf and ping. It specifically targets ICMP flood attacks on OpenFlow-connected user datagram protocol (UDP) and transmission control protocol (TCP) servers, where simulations show a 26,300% increase in jitter and a

37.5% drop in network throughput [2], [3]. We further extend this study by testing ICMP flood mitigation on a complex campus network topology.

The proposed ICMP flood detection and mitigation system integrates SDN and sFlow-RT to counteract DDoS attacks effectively [3], [4]. The exponential growth in internet traffic has made network security a pressing issue. ICMP flood attacks, in particular, present a significant threat by overwhelming network resources and causing severe disruptions [5]. Traditional security mechanisms often struggle with SDN's control data plane separation, resulting in inefficient detection and mitigation [6].

This research aims to develop a real-time, scalable ICMP flood detection and mitigation system using SDN and sFlow-RT. By leveraging SDN's centralized control and sFlow-RT's traffic monitoring capabilities, our approach enhances detection speed, minimizes attack impact, and improves network performance [7], [8]. Our implementation focuses on deploying OpenFlow switches within an SDN framework to dynamically identify and mitigate ICMP flood attacks.

2. RELATED WORK

Existing research has explored various strategies for mitigating DDoS attacks, particularly ICMP flood assaults. Liu *et al.* [9] proposed an intelligent framework for real-time mitigation of DDoS attacks, including TCP synchronize (SYN), UDP, and ICMP flood. This framework learns attack traffic patterns and effectively limits malicious flows. However, our study focuses solely on ICMP flood and employs sFlow as a detection trigger instead of the proposed framework.

Samta and Sood examined DDoS attacks in SDN environments, comparing traditional networks with SDN-based networks. Their findings indicate that SDNs perform better under attack conditions [10]. Rather than making direct comparisons, our study replicates a campus network using SDN-based tools to analyze mitigation effectiveness. Several studies have investigated DDoS attacks, including SYN flood, UDP flood, and ICMP flood [11]-[13]. These attacks are typically generated using hping3 and analyzed via ping and iPerf. Machine learning approaches, such as decision trees and logistic regression, have been used to classify attack traffic [14]. Unlike dataset-driven approaches, our research employs SDN-based mitigation techniques.

Other works have focused on identifying critical DDoS attack features in SDN environments [15]. Studies using Mininet-generated datasets highlight key traffic attributes for attack detection, including TCP, UDP, and ICMP traffic [16]. Feature selection methods indicate that packet count is the most significant metric for detecting DDoS threats in SDNs [17]-[19]. Our study focuses specifically on ICMP flood mitigation using modified SDN switches. Sainz *et al.* [20] explored SDN-based security in industrial networks through small-scale experiments involving packet payload modifications and ICMP flooding. Their findings suggest potential scalability issues, necessitating further research. Gao *et al.* [21] proposed an efficient and low-cost defense (ELD) mechanism for TCP SYN, UDP, and ICMP flood mitigation. ELD reduces the computational burden on controllers and differentiates attack traffic from legitimate flows [22]. Our study, however, relies on sFlow and fictitious routers instead of ELD.

Xiao *et al.* [23] introduced a back-propagation neural network (BPNN) approach for DDoS detection in SDNs, showing improved accuracy and reduced detection time compared to K-means algorithms. Our research applies a different strategy by designing an SDN-based architectural solution implemented on OpenFlow switches. Unlike prior studies that broadly address DDoS mitigation, our research specifically targets ICMP flood attacks, utilizing real-time traffic monitoring with sFlow-RT. The integration of sFlow-RT with OpenFlow-based SDN controllers allows for dynamic mitigation while minimizing resource overhead [5], [9], [10].

3. PROBLEM STATEMENT

This study addresses ICMP flood attacks in SDN environments using sFlow for traffic monitoring and OpenFlow switches for mitigation. Following the prepare, plan, design, implement, operate, optimize (PPDIOO) life cycle [24], the system leverages sFlow for continuous traffic analysis via agents and collectors [25], while software-based OpenFlow switches (e.g., modified TP-Link devices) support dynamic control [26]. In our test topology, modeled after a campus network in Yogyakarta, Indonesia, with a fake router as a gateway [27], ICMP traffic surges reached 423.67 MB, degrading availability. This research develops an SDN-based ICMP flood detection and mitigation system using sFlow-RT, enhancing real-time threat response while maintaining network performance.

4. RESEARCH METHOD

To verify the improved results of implementing ICMP Flood detection with SDN and sFlow-RT, we conducted experiments on our test topology implemented on our prepared devices. In this section, we first set up a software-based OpenFlow switch. Then, we simulate the actual topology on the test topology. Finally, we analyze the experimental results.

4.1. OpenFlow switch prototyping

To implement the SDN architecture, we modified a TP-Link TL-WR1043ND switch to support the OpenFlow protocol via firmware updates or manual flashing, as detailed in prior work [26]. An sFlow agent was integrated into the switch for continuous traffic monitoring, with data sent to an sFlow collector for real-time analysis (Figure 1) [28]. Controlled by ODL, this setup combines sFlow and OpenFlow to enable effective traffic monitoring and control, forming the basis of our ICMP flood detection and mitigation system. The system dynamically analyzes incoming packets, updating flow tables to mitigate attacks when traffic exceeds predefined thresholds.

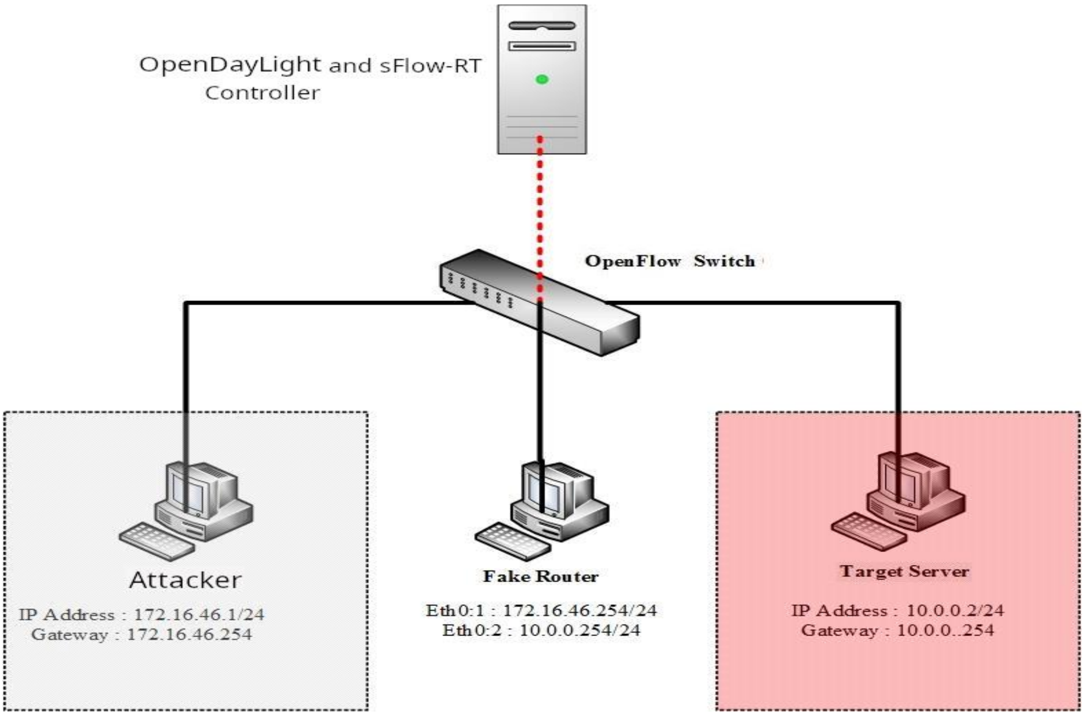


Figure 1. Test network topology

4.2. OpenFlow switch prototyping

To test and evaluate the system’s capacity to identify an intrusion or attack conducted via a computer network on the target computer and determine how a software-defined network can automatically integrate with sFlow-RT, three scenarios are used, among others, in the system test scenario currently in place;

1) First test scenario, this test is conducted to see the system’s ability to detect and mitigate attacks on the attacker’s computer will send ICMP packets using the ping utility on the Linux Ubuntu system against the target. The addition of the -f parameter to the ping utility is used to send packets in a flood;
2) The second test scenario is conducted to evaluate the performance of the network as data packets are transmitted to the server with an escalating number of users, specifically from 1 user, 10 users, 50 users, to 100 users. Then the computer acting as a client will ping 50 packets to the server to see the latency when the data packet is sent and blocking the data packet. For the latency value, the average will be taken from 5 trials for each number of users. In this scenario, we use the TCP protocol in sending packets to the server because no application can benchmark using ICMP. The tool used is siege. On the server side, the cross-platform, Apache, MySQL, PHP and Perl (XAMPP) application is installed as a web server;

3) Third scenario, a test scenario was conducted to test the security of the system that has been designed when data packets are sent and blocking the sending of data packets. The parameter taken is the number of ICMP packets passing through the network. The computing device of the assailant shall transmit data packets at intervals corresponding to 30 seconds, 60 seconds, 90 seconds, and 120 seconds. Subsequently, the quantity of ICMP packets traversing the network will be monitored at the moment of data packet transmission and upon the successful obstruction of the data packet by the system.

5. SYSTEM TEST

5.1. Detection and mitigation system testing

This test uses 2 hosts that act as attackers and victims. The attacker’s computer will send ICMP data packets to the victim’s computer as shown in Figure 2, and then the system that has been designed will detect the attacker by capturing and accumulating traffic from each sFlow agent which is then sent to the collector for analysis. When the sFlow collector detects some traffic as an attack, the OpenFlow controller will modify the rules in the OpenFlow table as an attack mitigation step by blocking the attack traffic. It can be seen in Figure 2 the detection and mitigation process on the network when too many packets are sent or an attack occurs.

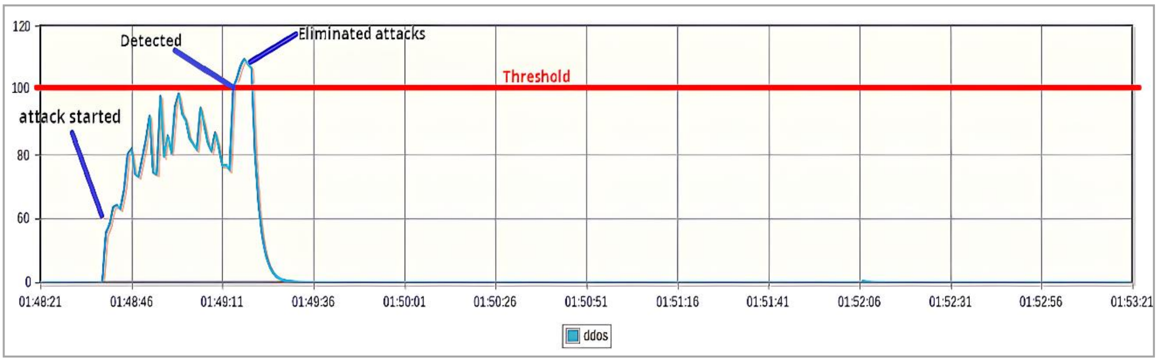
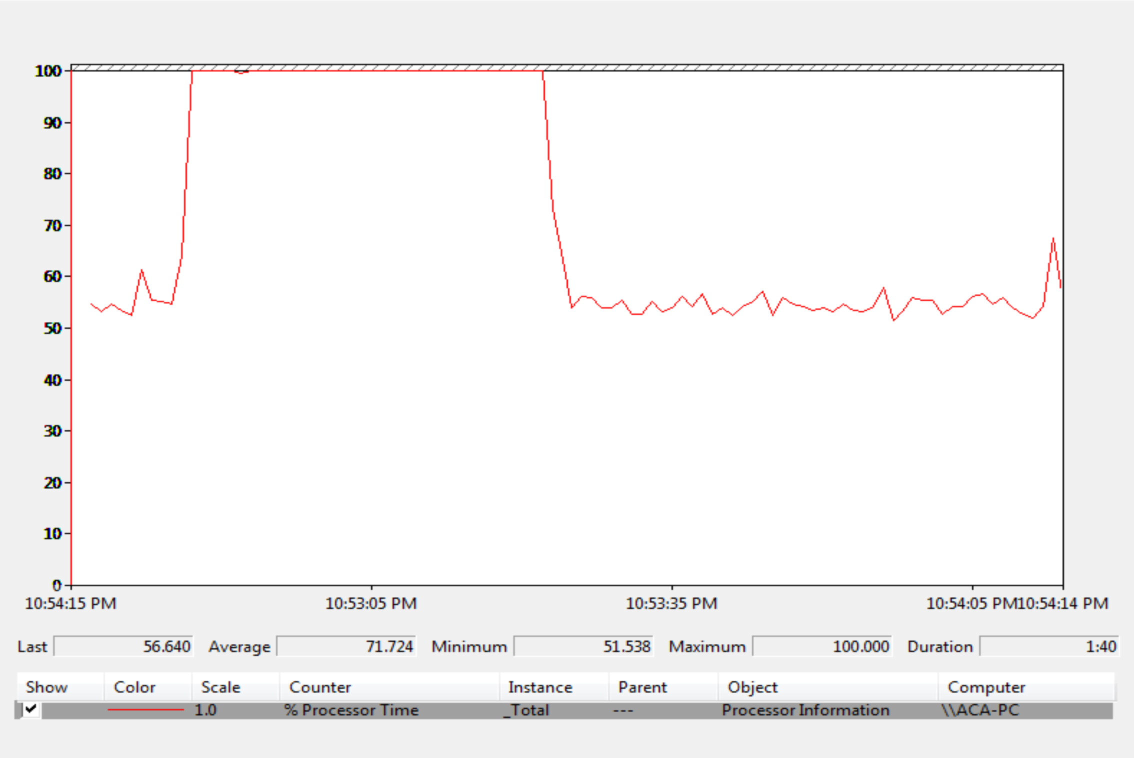


Figure 2. Detection and mitigation process

The 100 bytes/second threshold used in this experiment is shown in Figure 2. The sFlow-RT algorithm will visit the representational state transfer application programming interface (REST API), identify the packet as an attack packet, insert it into the metric, and record it in the attack log if the sampled packet exceeds the predefined threshold. Subsequently, the switch agent utilizing the sFlow-RT method will instruct the controller to block the attack packet’s originating internet protocol (IP) address. After getting the command, the controller will update its flow table and remove or block the IP address that is the source of the assault.

5.2. Latency testing

By sending flooding packets to the server under server conditions and blocking flooding packets with an increasing number of users from 1 user, 10 users, 50 users, and 100 users - this test aims to determine how long it takes to transmit packets to the server. After that, a machine posing as a client will ping 50 packets to the server to measure the latency during packet transmission and packet blocking. Each number of users will have an average of five attempts for the delay metric. Testing server latency during packet transmission and packet blocking involves the following steps: (1) to test the server, you can use the siege application with a target IP of 10.0.0.2; (2) To see the performance of the server when sending data packets until blocking data packets can be seen in Figure 3. It’s using the performance monitoring application. It can be seen that when sending data packets to 50 users, the process on the server central processing unit (CPU) increases to 100 per cent. But after the detection and mitigation system is activated, it can restore the process on the server CPU to normal again.



Last 56.640 Average 71.724 Minimum 51.538 Maximum 100.000 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	―	1.0	% Processor Time	_Total	---	Processor Information	\\ACA-PC

Figure 3. Server performance

5.3. ICMP packet count testing

The number of packets sent during an ICMP flood attack before and after the implementation of a software-defined network-based detection and mitigation system. The testing process is carried out before and after the implementation of the detection and mitigation system when an ICMP flood attack occurs, with an attack time of 30 seconds, 60 seconds, 90 seconds, and 120 seconds from five trials. By doing this experiment, the security system can function properly. This experiment uses the ping -f [target ip] -w [attack time] command. By using this command, the attacker will flood host 10.0.0.2 with ICMP packets for 30 seconds.

6. EXPERIMENTAL RESULTS

This section will discuss the results of testing the system that has been built. The following is an analysis of the results of the tests carried out on the sFlow-RT-based ICMP flood detection and mitigation system and software-defined network.

6.1. Latency testing results

Latency tests using the ping utility (50 packets, 5 trials per user group) show significant improvement post-mitigation (Table 1). For 30 users, latency dropped from 42.822 ms to 8.807 ms (79.43% decrease), with similar reductions for 60, 90, and 120 users (Figure 4). This confirms the system’s ability to reduce latency under attack conditions.

Table 1. Number of packet transmissions, data packet blocks, and percentage decrease

Num. of users (users)	Packet’s delivery (milliseconds)	Data packet’s block (milliseconds)	Packet’s of decrease (%)
30	42.822	8.807	79.43
60	87.265	17.557	79.88
90	134.678	28.724	78.67
120	176.783	36.409	79.40

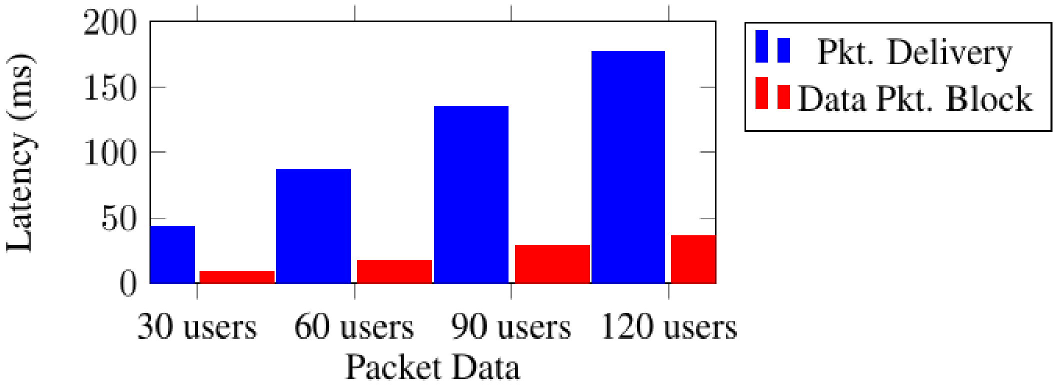


Figure 4. Average latency graph

6.2. ICMP packet count testing result

ICMP packet tests with flood parameters (Table 2) show a reduction from 311,130.2 to 99 packets over 120 seconds (99.97% decrease), validating system efficacy. Variations in blocked packets result from sampling (1 of 5 packets every 10 seconds), enabling sFlow to detect and OpenFlow to mitigate attacks by dropping malicious traffic.

Table 2. Number of ICMP packages

Time (second)	Package delivery (packet's)	Package block (packet's)	Percentage decrease (%)
30	77038.6	106.4	99.86
60	153997.8	93.2	99.94
90	232667.6	100	99.96
120	311130.2	99	99.97

6.3. Comparison and implications

Compared to Liu *et al.* [9] and Samta & Sood [10], our system excels in ICMP-specific mitigation using sFlow-RT, validating hypotheses of reduced latency and packet counts. Its scalability suits large-scale networks like internet service providers (ISPs), with potential adaptation for UDP or DNS floods via threshold adjustments.

7. CONCLUSION

This study shows SDN and sFlow-RT mitigating ICMP flood attacks, cutting packets from 311,130.2 to 99 and latency from 42.822 ms to 8.807 ms (30 users), ensuring network integrity. Unlike broader DDoS defenses, it excels in real-time ICMP detection, aiding scalable networks like ISPs. Adaptable to Smurf or UDP floods, it fits IoT with threshold tweaks. Lacking false positive/negative metrics limits reliability. Future work should refine sampling, test controllers like ONOS, and assess IoT or diverse topologies for scalability and accuracy, advancing SDN-based DDoS protection.

FUNDING INFORMATION

The authors declare that this research did not receive any funding from any agency, whether public, private, or commercial.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rikie Kartadie	✓	✓		✓		✓		✓	✓	✓			✓	
Adi Kusjani		✓		✓	✓			✓	✓		✓			
Rangga Warsito		✓	✓			✓	✓	✓	✓					
Yudhi Kusnanto	✓		✓			✓		✓		✓	✓		✓	
Lucia Nugraheni Harnaningrum	✓				✓	✓				✓		✓		

C :	**C**onceptualization	I :	**I**nvestigation	Vi :	**Vi**sualization
M :	**M**ethodology	R :	**R**esources	Su :	**Su**pervision
So :	**So**ftware	D :	**D**ata Curation	P :	**P**roject administration
Va :	**Va**lidation	O :	Writing - **O**riginal Draft	Fu :	**Fu**nding acquisition
Fo :	**Fo**rmal analysis	E :	Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no financial, personal, or professional interests that could influence the results of this study.

DATA AVAILABILITY

The authors declare that no new data were generated or analyzed in this study. All information used is from published literature and is referenced in the article.

REFERENCES

[1] International Telecommunication Union, "Measuring Digital Development: Facts and Figures 2024," ITU, 2024. [Online]. https://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx

[2] A. Sangodoyin, T. Sigwele, P. Pillai, Y. F. Hu, I. Awan, and J. Disso, “DoS attack impact assessment on software defined networks,” in *Wireless and satellite systems*, vol. 231, pp. 11–22, 2018, doi: 10.1007/978-3-319-76571-6_2.

[3] Y. Lu and M. Wang, “An easy defense mechanism against botnet-based DDoS flooding attack originated in SDN environment using sFlow,” in *Proceedings of the 11th international conference on future internet technologies*, New York, NY, USA: Association for Computing Machinery, 2016, pp. 14–20. doi: 10.1145/2935663.2935674.

[4] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, “Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments,” *Computer Networks*, vol. 62, pp. 122–136, 2014, doi: 10.1016/j.bjp.2013.10.014.

[5] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita, “Detecting distributed denial of service attacks: Methods, tools and future directions,” *The Computer Journal*, vol. 57, no. 4, pp. 537–556, Mar. 2013, doi: 10.1093/comjnl/bxt031.

[6] F. Rebecchi, J. Boite, P.-A. Nardin, M. Bouet, and V. Conan, “DDoS protection with stateful software-defined networking,” *International Journal of Network Management*, vol. 29, no. 1, p. e2042, 2019, doi: 10.1002/nem.2042.

[7] T. Wang, H. Chen, and C. Qi, “MinDoS: a priority-based SDN safe-guard architecture for DoS attacks,” *IEICE Transactions on Information and Systems*, vol. E101.D, no. 10, pp. 2458–2464, 2018, doi: 10.1587/transinf.2017EDP7419.

[8] J. Wang and L. Wang, “SDN-defend: a lightweight online attack detection and mitigation system for DDoS attacks in SDN,” *Sensors*, vol. 22, no. 21, Art. no. 8287, 2022, doi: 10.3390/s22218287.

[9] Y. Liu, M. Dong, K. Ota, J. Li and J. Wu, “Deep Reinforcement Learning based Smart Mitigation of DDoS Flooding in Software-Defined Networks,” *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, Spain, 2018, pp. 1-6, doi: 10.1109/CAMAD.2018.8514971.

[10] R. Samta and M. Sood, “Analysis and mitigation of DDoS flooding attacks in software defined networks,” in *International Conference on Innovative Computing and Communications. Advances in Intelligent Systems and Computing*, Singapore: Springer Singapore, vol. 1059, 2020, pp. 337–355, doi: 10.1007/978-981-15-0324-5_30.

[11] L. Kavisankar, C. Chellappan, S. Venkatesan, and P. Sivasankar, “Enhanced efficient SYN spoofing detection and mitigation scheme for DDoS attacks,” *International Journal of Internet Technology and Secured Transactions*, vol. 8, no. 4, pp. 583–600, 2018, doi: 10.1504/IJITST.2018.095936.

[12] V. K. Yadav, M. C. Trivedi, and B. M. Mehtre, “DDA: An approach to handle DDoS (ping flood) attack,” in *Proceedings of international conference on ICT for sustainable developmen*t, S. C. Satapathy, A. Joshi, N. Modi, and N. Pathak, Eds., Singapore: Springer Singapore, vol. 408, 2016, pp. 11–23, doi: 10.1007/978-981-10-0129-1_2.

[13] S. S. Kolahi, A. A. Alghalbi, A. F. Alotaibi, S. S. Ahmed and D. Lad, “Performance comparison of defense mechanisms against TCP SYN flood DDoS attack*,” 2014 6th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, St. Petersburg, Russia, 2014, pp. 143-147, doi: 10.1109/ICUMT.2014.7002093.

[14] T. Arvind and K. Radhika, “An SDN-based DDoS traffic generation, collection and classification using machine learning techniques,” in *Advanced engineering optimization through intelligent techniques*, R. Venkata Rao and J. Taler, Eds., Singapore: Springer Nature Singapore, 2023, pp. 421–428, doi: 10.1007/978-981-19-9285-8_39.

[15] P. Zhai, Y. Song, X. Zhu, L. Cao, J. Zhang and C. Yang, “Distributed Denial of Service Defense in Software Defined Network Using OpenFlow,” *2020 IEEE/CIC International Conference on Communications in China (ICCC)*, Chongqing, China, 2020, pp. 1274-1279, doi: 10.1109/ICCC49849.2020.9238872.

[16] H. -C. Wei, Y. -H. Tung and C. -M. Yu, “Counteracting UDP flooding attacks in SDN,” *2016 IEEE NetSoft Conference and Workshops (NetSoft), Seoul, Korea (South)*, 2016, pp. 367-371, doi: 10.1109/NETSOFT.2016.7502468.

[17] M. T. Kurniawan, S. Yazid, and Y. G. Sucahyo, "Comparison of feature selection methods for DDoS attacks on software defined networks using filter-based, wrapper-based and embedded-based," *JOIV: International Journal on Informatics Visualization*, vol. 6, no. 4, pp. 809–814, 2022, doi: 10.30630/joiv.6.4.1476.
[18] T. V. Phan, T. Van Toan, D. Van Tuyen, T. T. Huong and N. H. Thanh, "OpenFlowSIA: An optimized protection scheme for software-defined networks from flooding attacks," *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, Ha-Long, Vietnam, 2016, pp. 13-18, doi: 10.1109/CCE.2016.7562606.
[19] K. -y. Chen, A. R. Junuthula, I. K. Siddhrau, Yang Xu and H. J. Chao, "SDNShield: Towards more comprehensive defense against DDoS attacks on SDN control plane," *2016 IEEE Conference on Communications and Network Security (CNS)*, Philadelphia, PA, 2016, pp. 28-36, doi: 10.1109/CNS.2016.7860467.
[20] M. Sainz, I. Garitano, M. Iturbe, and U. Zurutuza, "Deep packet inspection for intelligent intrusion detection in software-defined industrial networks: A proof of concept," *Logic Journal of the IGPL*, vol. 28, no. 4, pp. 461–472, Dec. 2019, doi: 10.1093/jigpal/jzz060.
[21] S. Gao, Z. Peng, B. Xiao, A. Hu, Y. Song and K. Ren, "Detection and Mitigation of DoS Attacks in Software Defined Networks," in *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1419-1433, June 2020, doi: 10.1109/TNET.2020.2983976.
[22] Y.-C. Wang and Y.-C. Wang, "Efficient and low-cost defense against distributed denial-of-service attacks in SDN-based networks," *International Journal of Communication Systems*, vol. 33, no. 14, p. e4461, 2020, doi: 10.1002/dac.4461.
[23] H. Xiao, T. Xiang, and S. Tang, "Research on detection and defense methods for software-defined network architecture after hybrid attack by distributed denial of service," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 19, no. 6, doi: 10.1002/tee.24026.
[24] R. Rizky and Z. Hakim, "Analysis and Design of Voip Server (Voice Internet Protocol) using Asterisk in Statistics and Statistical Informatics Communication of Banten Province using Ppdioo Method," *Journal of Physics: Conference Series*, vol. 1179, no. 1, p. 012160, Jul. 2019, doi: 10.1088/1742-6596/1179/1/012160.
[25] sFlow, "Traffic monitoring using sFlow," sFlow.org, 2003. [Online]. Available: https://sflow.org/sFlowOverview.pdf
[26] R. Kartadie, F. Rozi, and E. Utami, "Openflow switch software-based performance test on its implementation on campus network," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 13, pp. 4136–4146, 2018.
[27] A. Hakiri, A. Gokhale, P. Berthou, D. C. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Computer Networks*, vol. 75, pp. 453–471, 2014, doi: 10.1016/j.comnet.2014.10.015.
[28] T. V. Phan, N. K. Bao and M. Park, "A Novel Hybrid Flow-Based Handler with DDoS Attacks in Software-Defined Networking," *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, Toulouse, France, 2016, pp. 350-357, doi: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0069.

BIOGRAPHIES OF AUTHORS

Rikie Kartadie he received his Master of Computer Science from AmikomYogyakarta University (2014). Now a lecturer at the Department of Computer Engineering, Universitas Technologi Digital Indonesia. In addition, he also has an assessment of the national professional certification agency, computer network competence. In 2018 he was assigned by the Ministry of Education, Culture, Research and Technology as a lecturer workload assessor. He has published more than 35 journal papers, 2 written books, and 6 scopus papers. Research interests are computer networks, SDN, and IoT, teaching in the field of Networking. Have received several grants from the Ministry of Research, Technology and Higher Education.He can be contacted at email: rikie@utdi.ac.id.

Adi Kusjani he received his Master of Computer Science from Universiti Master of Electrical Engineering from the Faculty of Electrical Engineering, Gadjah Mada University Yogyakarta (2014) is now a lecturer at the Computer Engineering Vocational Program, Universitas Technologi Digital Indonesia, Research interests in Computer Networks, IoT, He has received several grant programs from the Ministry of Education and Culture of the Republic of Indonesia, written several books and published 10 scientific articles both in journals and seminars. He can be contacted at email: adikusjani@utdi.ac.id.

Rangga Warsito he completed his Bachelor of Computer Science from Amikom University Yogyakarta and is now a practitioner in the field of computer networks, currently working as a Network Engineer for the largest network company in Indonesia. He can be contacted at email: ranggaawarsito@gmail.com.

Yudhi Kusnanto he received his Master of Electrical Engineering from Indonesia University (2013) is now a lecturer at the Computer Engineering Vocational Program, Universitas Teknologi Digital Indonesia, Research interests in Computer Networks, System Security and Network Security. He has received several grant programs from the Ministry of Education and Culture of the Republic of Indonesia, written several articles both in journals and seminars. He can be contacted at email: yudhi@utdi.ac.id.

Lucia Nugraheni Harnaningrum She received her Doctor of Computer Science from Gadjah Mada University, Yogyakarta (2022). Now a lecturer at the Department of Informatics, Universitas Teknologi Digital Indonesia. Research interests are computer networks, Embedded System, mobile application and IoT, teaching in the field of Embedded System. Have received several grants from the Ministry of Research, Technology and Higher Education, written and translate several books and published scientific articles both in International journals and seminars. She can be contacted at email: ningrum@utdi.ac.id.