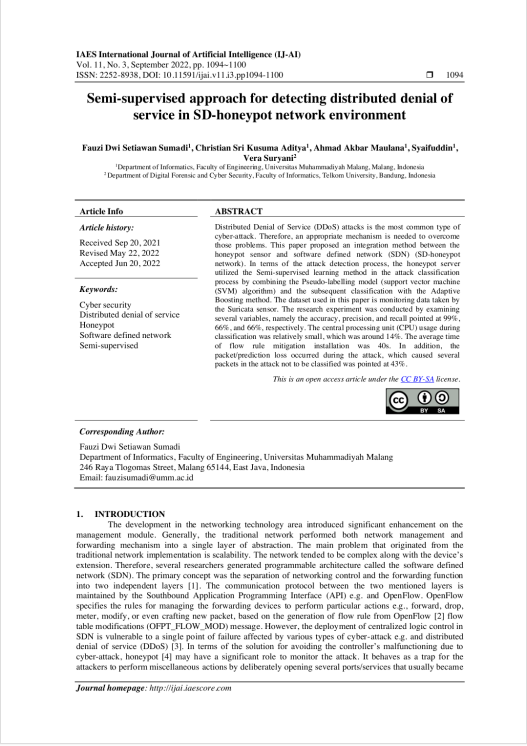
Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Assignment title:Paper_1
Submission title:	ty_Distributed_denial_of_service_Honeypot_Software_defined...
File name:	ty_Distributed_denial_of_service_Honeypot_Software_defined...
File size:	364.73K
Page count:	7
Word count:	4,176
Character count:	23,253
Submission date:	24-Jul-2023 01:01PM (UTC+0700)
Submission ID:	2135923196



IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 3, September 2022, pp. 1094~1100
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i3.pp1094-1100

Semi-supervised approach for detecting distributed denial of service in SD-honeypot network environment

Fauzi Dwi Setiawan Sumadi[1], Christian Sri Kusuma Aditya[1], Ahmad Akbar Maulana[1], Syaifuddin[1], Vera Suryani[2]
[1]Department of Informatics, Faculty of Engineering, Universitas Muhammadiyah Malang, Malang, Indonesia
[2]Department of Digital Forensic and Cyber Security, Faculty of Informatics, Telkom University, Bandung, Indonesia

Article Info

Article history:

Received Sep 20, 2021
Revised May 22, 2022
Accepted Jun 20, 2022

Keywords:

Cyber security
Distributed denial of service
Honeypot
Software defined network
Semi-supervised

ABSTRACT

Distributed Denial of Service (DDoS) attacks is the most common type of cyber-attack. Therefore, an appropriate mechanism is needed to overcome those problems. This paper proposed an integration method between the honeypot sensor and software defined network (SDN) (SD-honeypot network). In terms of the attack detection process, the honeypot server utilized the Semi-supervised learning method in the attack classification process by combining the Pseudo-labelling model (support vector machine (SVM) algorithm) and the subsequent classification with the Adaptive Boosting method. The dataset used in this paper is monitoring data taken by the Suricata sensor. The research experiment was conducted by examining several variables, namely the accuracy, precision, and recall pointed at 99%, 66%, and 66%, respectively. The central processing unit (CPU) usage during classification was relatively small, which was around 14%. The average time of flow rule mitigation installation was 40s. In addition, the packet/prediction loss occurred during the attack, which caused several packets in the attack not to be classified was pointed at 43%.

Corresponding Author:

Fauzi Dwi Setiawan Sumadi
Department of Informatics, Faculty of Engineering, Universitas Muhammadiyah Malang
246 Raya Tlogomas Street, Malang 65144, East Java, Indonesia
Email: fauzisumadi@umm.ac.id

1. INTRODUCTION

The development in the networking technology area introduced significant enhancement on the management module. Generally, the traditional network performed both network management and forwarding mechanism into a single layer of abstraction. The main problem that originated from the traditional network implementation is scalability. The network tended to be complex along with the device's extension. Therefore, several researchers generated programmable architecture called the software defined network (SDN). The primary concept was the separation of networking control and the forwarding function into two independent layers [1]. The communication protocol between the two mentioned layers is maintained by the Southbound Application Programming Interface (API) e.g. and OpenFlow. OpenFlow specifies the rules for managing the forwarding devices to perform particular actions e.g., forward, drop, meter, modify, or even crafting new packet, based on the generation of flow rule from OpenFlow [2] flow table modifications (OFPT_FLOW_MOD) message. However, the deployment of centralized logic control in SDN is vulnerable to a single point of failure affected by various types of cyber-attack e.g. and distributed denial of service (DDoS) [3]. In terms of the solution for avoiding the controller's malfunctioning due to cyber-attack, honeypot [4] may have a significant role to monitor the attack. It behaves as a trap for the attackers to perform miscellaneous actions by deliberately opening several ports/services that usually became

ty_Distributed_denial_of_service_Honeypot_Software_defined_n.pdf

by

Submission date: 24-Jul-2023 01:01PM (UTC+0700)
Submission ID: 2135923196
File name: ty_Distributed_denial_of_service_Honeypot_Software_defined_n.pdf (364.73K)
Word count: 4176
Character count: 23253

IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 11, No. 3, September 2022, pp. 1094~1100
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i3.pp1094-1100

Semi-supervised approach for detecting distributed denial of service in SD-honeypot network environment

Fauzi Dwi Setiawan Sumadi[1], Christian Sri Kusuma Aditya[1], Ahmad Akbar Maulana[1], Syaifuddin[1], Vera Suryani[2]
[1]Department of Informatics, Faculty of Engineering, Universitas Muhammadiyah Malang, Malang, Indonesia
[2]Department of Digital Forensic and Cyber Security, Faculty of Informatics, Telkom University, Bandung, Indonesia

Article Info

Article history:

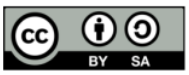
Received Sep 20, 2021
Revised May 22, 2022
Accepted Jun 20, 2022

Keywords:

Cyber security
Distributed denial of service
Honeypot
Software defined network
Semi-supervised

ABSTRACT

Distributed Denial of Service (DDoS) attacks is the most common type of cyber-attack. Therefore, an appropriate mechanism is needed to overcome those problems. This paper proposed an integration method between the honeypot sensor and software defined network (SDN) (SD-honeypot network). In terms of the attack detection process, the honeypot server utilized the Semi-supervised learning method in the attack classification process by combining the Pseudo-labelling model (support vector machine (SVM) algorithm) and the subsequent classification with the Adaptive Boosting method. The dataset used in this paper is monitoring data taken by the Suricata sensor. The research experiment was conducted by examining several variables, namely the accuracy, precision, and recall pointed at 99%, 66%, and 66%, respectively. The central processing unit (CPU) usage during classification was relatively small, which was around 14%. The average time of flow rule mitigation installation was 40s. In addition, the packet/prediction loss occurred during the attack, which caused several packets in the attack not to be classified was pointed at 43%.



Corresponding Author:

Fauzi Dwi Setiawan Sumadi
Department of Informatics, Faculty of Engineering, Universitas Muhammadiyah Malang
246 Raya Tlogomas Street, Malang 65144, East Java, Indonesia
Email: fauzisumadi@umm.ac.id

1. INTRODUCTION

The development in the networking technology area introduced significant enhancement on the management module. Generally, the traditional network performed both network management and forwarding mechanism into a single layer of abstraction. The main problem that originated from the traditional network implementation is scalability. The network tended to be complex along with the device's extension. Therefore, several researchers generated programmable architecture called the software defined network (SDN). The primary concept was the separation of networking control and the forwarding function into two independent layers [1]. The communication protocol between the two mentioned layers is maintained by the Southbound Application Programming Interface (API) e.g. and OpenFlow. OpenFlow specifies the rules for managing the forwarding devices to perform particular actions e.g., forward, drop, meter, modify, or even crafting new packet, based on the generation of flow rule from OpenFlow [2] flow table modifications (OFPT_FLOW_MOD) message. However, the deployment of centralized logic control in SDN is vulnerable to a single point of failure affected by various types of cyber-attack e.g. and distributed denial of service (DDoS) [3]. In terms of the solution for avoiding the controller's malfunctioning due to cyber-attack, honeypot [4] may have a significant role to monitor the attack. It behaves as a trap for the attackers to perform miscellaneous actions by deliberately opening several ports/services that usually became

the main target e.g., secure shell (SSH), server message block (SMB), internet control message protocol (ICMP). Several types of honeypot sensors are specialized to attract specific types of attacks [4] e.g., Dionea can log and capture malware activity that uses several types of protocols such as hypertext transfer protocol (HTTP), file transfer protocol (FTP), voice over internet protocol (VOIP), and the other protocols [5], Cowrie focuses to monitor and store malicious activity regarding the brute-force attacks performed under the Telnet or SSH [6], Suricata is directed to create an intrusion detection system (IDS) for complex circumstances [7]. The integration of SDN and honeypot provides a comprehensive attack representation that aims to overwhelm the architecture. SD-honeypot network may become a single system for developing both IDS and intrusion prevention system (IPS).

Previously, several researchers have already investigated the capability of DDoS for overloading traditional networks. The former concept for detecting DDoS was categorized into two approaches namely the statistic [8], [9] and artificial intelligence (AI) [10]–[14]. Several statistics approaches have been deployed e.g., the Entropy [8] which calculated the data randomness and specified the DDoS threshold by its value, Bloom-filter [9] which focused the detection phase by comparing the hash value of the incoming packet to assure the packet was not considered as SYN flood attack. The statistical approaches are predominantly constant at measuring the pattern, if the attackers alter the flooding scheme, these methods may not identify the attacks. Therefore, several papers also introduced AI techniques for detecting DDoS. Maslan *et al*. [10] in implemented the feature selection combined with several classification algorithms to detect DDoS using their dataset. The researchers selected 4 from the whole 25 features extracted using CICFlowMeter-V3 and concluded that the most effective algorithm is Random Forest. Similarly, Fadlil *et al*. [11] used their dataset by capturing the attack on simulation using low orbit ion cannon (LOIC). The results stated the Naïve Bayes algorithm could predict the outcomes precisely even though there was no apparent result for the classification metric. Several papers also conducted the classification based on available datasets [12]–[14] (NSL-KDD, UNB ISCX 12, and UNSW-NB15). Idhammad *et al*. [12] proposed the Semi-supervised learning for classifying the DDoS attack gained 98.23% for accuracy. Mohammed *et al*. [13] and Muhammad *et al*. [14] utilized deep learning (DL) for detecting DDoS and achieved an accuracy at 97.82% and 99.60% respectively. The programmability feature in SDN may provide manageable structure for implementing AI to detect DDoS. Several papers provided analysis by maintaining the dataset based on the OpenFlow extraction process or existing dataset. Sumadi *et al*. [15] compared several machine learning (ML) algorithms using datasets generated from the port statistic message combined with the default features for packet extraction information. The results stated that SVM could create the best outcomes in terms of accuracy (100%). Dey and Rahman [16] used network security laboratory KDD (NLS-KDD) dataset as the primary dataset for detecting DDoS using both ML and DL gained 88% in accuracy for the result of the gated recurrent unit long short-term memory (GRU-LSTM) model.

The other possible technique for resolving DDoS/Cyber-attack is integrating the honeypot in the SDN environment [17]–[21]. Wang and Wu [17] proposed a customized topology by combining existed SDN architecture, high-level and low-level honeypot's topology. The attacks were redirected to the honeypots topology based on their level. The rest of the mentioned papers [18]–[21] presented the deployment of honeypot for migrating the cyber-attacks in software defined internet of things (SD-IoT) network. Similarly, the authors were directed their research for only monitoring the attack and did not perform further analysis.

The former papers concluded that honeypot was appropriate as a tool for attracting, capturing, and monitoring cyber-attacks. There was still no paper that aimed to perform in-depth processes for analyzing the data captured from the honeypot sensor in SDN. Therefore, this paper is focused to investigate the possibility of Semi-supervised learning to detect the ICMP flood attack in the SD-honeypot network environment. The main contribution of this paper is constructing an IDS and IPS system which proposes the SD-honeypot for resolving DDoS attacks, applying the semi-supervised method for classifying the captured data from the Suricata sensor, and mitigating the attack using representational state transfer application programming interface (REST-API). The effectiveness of the proposed method is measured using standard classification metrics, resource usage, and the time value for installing the mitigation rule.

2. RESEARCH METHOD

The experiment was implemented using a real-hardware environment depicted in Figure 1. There was one controller (C1) using Ryu [22], three SDN-enabled routers (R1, R2, and R3) using Mikrotik [23] which supported OpenFlow version 1.1, and four hosts (H1-H4) using Ubuntu OS. H1 and H3 were pointed as normal hosts for communicating using normal ICMP packets. The attacker resided in H2 where the flooding type was an ICMP flood attack. The transmitted packets consisted of randomly generated medium access control (MAC) and internet protocol (IP) addresses using Scapy [24]. The attack's flow was at a rate of 100; 200; 500; 1,000; 2,000 packets per second which were transferred using Tcpreplay [25]. H4 was installed by the modern honey network (MHN) server integrated with the Suricata sensor for detecting ICMP flood.

The detailed information of system workflow is described in Figure 2. The received packet was inspected by Mikrotik based on the available Flow Rules. If there was no flow filtering the incoming packet the switch generated Packet-In Message (OFPT_PACKET_IN) encapsulating the packet. However, if the packets were intended to attack the vulnerabilities of the Suricata sensor, the switch automatically sent the packet to H4. Subsequently, the H4 stored the packet's information on the MongoDB database. Based on the proposed scenario, the application installed in H4 collected the data from MongoDB within a range of 10, 20, 30, 40, 50, 60, 70, 80, 90, 100 seconds. The extracted packets were pointed as the data test for the Pseudo-labelling and Semi-supervised approaches. If there were no packets categorized as DDoS packets, the application notified as normal circumstances. In contrast, the application transmitted the Flow Modification Message (OFPT_FLOW_MOD) encapsulated in JavaScript Object Notation (JSON) format for commanding the controller to generate mitigation flow to all of the available switches through REST-API. The flow mitigation had consisted of a flow match structure for filtering the attack based on the protocol's type and flow action for dropping the packet (no available action needed to be specified based on the OpenFlow protocol).

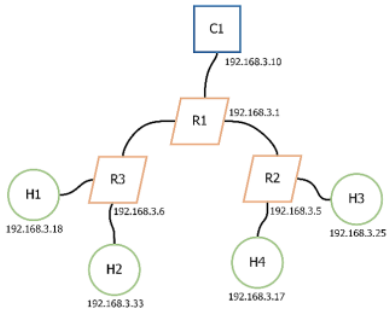


Figure 1. Simulation's topology

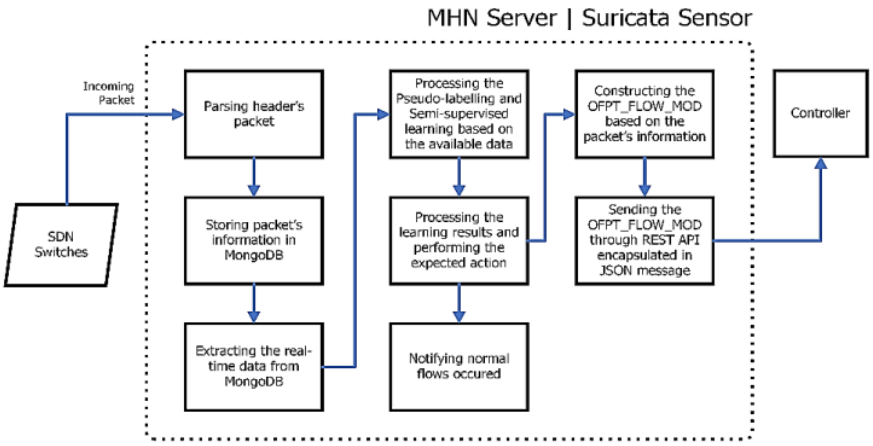


Figure 2. MHN server's block diagram

The classification process included the Pseudo-labelling and Supervised Learning method described in Figure 3. The Pseudo-labelling was performed by the support vector machine (SVM) algorithm using a Linear Kernel. The dataset used during the experiment contained 27,000 labeled data trains from 70,000 data in total. The installed application extracted the live data test from the MongoDB database which was utilized by the MHN to gather the stored attack data from the honeypot sensors (Suricata). The extraction process was experimented within several ranges of times (10-90 s). The application divided the extracted data into two components of an unlabeled dataset. The first fraction of data was being classified using the labeled data

train. Then the data were integrated which produced the combination of labeled train data and the fraction of classified data.

The combined data was pointed as the training set for the Supervised Learning model using Adaptive Boosting algorithm with the number of estimators at ten. The classification process using the Supervised model was performed on the second fraction of the live dataset. The results of the classification were evaluated using standard variables including the accuracy, precision, recall, F1-score, the packet loss during the classification, the central processing unit (CPU) usage of application during the whole process, and the time for the mitigation flow to be installed on the SDN switch/mikrotik.

The sample of the data train and live dataset during the experiment is illustrated in Table 1. It has seven features and one label consisting of two categories, DDoS, and normal packet. The features were the default data provided by the MHN server.

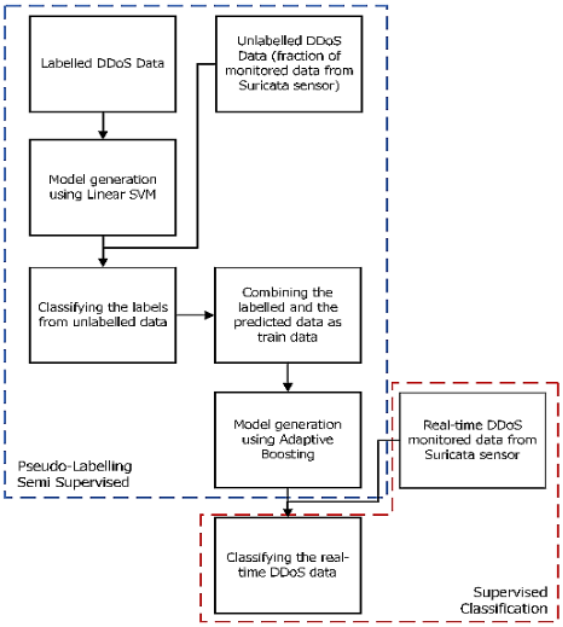


Figure 3. Semi-supervised learning implementation

Table 1. The sample of the dataset used during the experiment

Protocol	Hpfeed_id	Timestamp	Source_ip	Destination_ip	Identifier	Honeypot	Type
ICMP	ObjectId(5e4fad27f8 1c700cab511e8d)	2020-02-21T10:12:55.066Z	154.25.125.196	192.168.3.25	: :	suricata	DDOS
ICMP	ObjectId(5e280d913 186f205962bef16)	2020-01-22T08:53:37.924Z	192.168.3.33	192.168.3.25		suricata	NORMAL
ICMP	ObjectId(5e4cf03d3 186f205953a64c0)	2020-02-19T08:22:21.679Z	192.168.3.17	192.168.3.25		suricata	NORMAL
ICMP	ObjectId(5e53f1871 d41c80851461452)	2020-02-24T15:53:43.407Z	228.156.186.17 7	192.168.3.25		suricata	DDOS

3. RESULTS AND DISCUSSION

The research results were extracted from several scenarios by comparing the fluctuated rate of the database's extraction interval and the packet's sending rate. The MongoDB data acquisition process was delayed after several time intervals within the range of 10, 30, 50, 70, and 90 s. The attacker also maintained the sending rate using Tcpreplay at ranges of 100; 500; 1,000; and 2,000 packets/s. Based on the results provided by Table 2, the average accuracy was pointed at 99% and 66% for the precision, recall, and F1-score. Although the accuracy produced a high value, the precision still pointed at low indicating that the generated model could predict the result and was almost precise. Moreover, the growth of the packet's

sending rate not significantly impacted the accuracy and the other variables which indicated that the Pseudo-labelling model was consistent to perform the classification process. The low value of precision might be originated from the prediction loss during the attack or normal flow.

Table 3 describes the percentage of prediction loss during the real-time attack scenario for all options. The average prediction loss was 43.5%. This event might be occurred because the Suricata sensor was overwhelmed by the attack and normal flow; therefore, most of the normal packets were dropped and caused the precision value to drop significantly.

The application installed in the MHN server also performed a mitigation scheme by commanding the controller to send OFPT_FLOW_MOD. The mechanism could be implemented by deploying REST API -HTTP POST request provided by Ryu. The time needed to install the mitigation flow was extracted, as shown in Table 4 for measuring the effectiveness of the mitigation approach. The average time for installing the mitigation flow increased along with the growth of packet sending rate. The time growth was affected by the duration for performing the classification since the number of datasets also expanded.

In terms of resource usage during the classification process, Table 5 shows the MHN server's CPU utilization for performing attack detection, classification, and mitigation processes. In average, the CPU usage pointed at 14.5%, indicating that the mentioned processes did not significantly exhaust the MHN server despite the fact that the server was flooded by the DDoS attack.

Table 2. Classification results

Packet's sending rate	Database extraction interval	Accuracy	Precision	Recall	F1-score
100 Packets/s	10s	99%	99%	99%	99%
	30s	99%	54.12%	54.12%	54.12%
	50s	99%	58.17%	58.17%	58.17%
	70s	99%	55.41%	55.41%	55.41%
	90s	99%	59.29%	59.29%	59.29%
500 Packets/s	10s	99%	99%	99%	99%
	30s	99%	57.93%	57.93%	57.93%
	50s	99%	56.16%	56.16%	56.16%
	70s	99%	60.78%	60.78%	60.78%
	90s	99%	55.19%	55.19%	55.19%
1,000 Packets/s	10s	99%	99%	99%	99%
	30s	99%	57.29%	57.29%	57.29%
	50s	99%	58.06%	58.06%	58.06%
	70s	99%	61.74%	61.74%	61.74%
	90s	99%	58.66%	58.66%	58.66%
2,000 Packets/s	10s	99%	99%	99%	99%
	30s	99%	59.19%	59.19%	59.19%
	50s	99%	56.99%	56.99%	56.99%
	70s	99%	56.34%	56.34%	56.34%
	90s	99%	58.72%	58.72%	58.72%

Table 3. Packet/prediction loss during the experiment

Packet's sending rate	Number of packet being sent (normal and DDoS)	Number of packet's receive	Packet/prediction loss
100 Packets/s	30,000	12,458	41.53%
500 Packets/s		11,722	49.07%
1,000 Packets/s		12,444	41.48%
2,000 Packets/s		12,577	41.92%

Table 4. The duration for installing the flow mitigation

Packet's sending rate	Timestamp for flow installation (datetime to epoch ms)	Timestamp of the attack (datetime to epoch ms)	Time taken for install the mitigation flow
100 Packets/s	1585896414491	1585896400083	14408ms ~ 14s
500 Packets/s	1585897516757	1585897476750	40007ms ~ 40s
1,000 Packets/s	1585897981133	1585897951125	30008ms ~ 30s
2,000 Packets/s	1585898513413	1585898433410	80003ms ~ 80s

Table 5. MHN's CPU usage in average

Packet's sending rate	CPU usage in percentage
100 Packets/s	14.46%
500 Packets/s	15.24%
1,000 Packets/s	14.04%
2,000 Packets/s	14.51%

4. CONCLUSION

The integration of SD-honeypot network might become one of availability problem occurred on computer network. The implementation of SD-honeypot integration produced positive impacts proven by the classification metrics stated in the previous section. The precision, recall, and F1-Score were not pointed at a high value because there was a fraction of the data test that was not being classified because of the packet loss. The time needed to install the mitigation rule increased with the growth of the database's extraction interval. This might be occurred since the size of the captured data also expanded. In order to increase the classification metrics as a future project's reference, the utilization of Extract, transform, and load (ETL) technique can be deployed for capturing all of the attacks over several similar honeypot sensors directly without involving MHN.

ACKNOWLEDGEMENTS

The authors would like to express sincere gratitude to the Informatics laboratory at The University of Muhammadiyah Malang for providing resources during the experiment.

REFERENCES

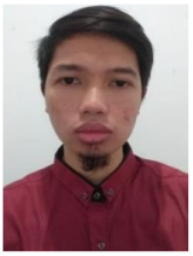
[1] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013, doi: 10.1109/MCOM.2013.6461195.
[2] "OpenFlow switch specification." Open Networking Foundation.
[3] M. A. Naagas, E. L. Mique Jr, T. D. Palaoag, and J. S. Dela Cruz, "Defense-through-deception network security model: securing university campus network from DOS/DDOS attack," *Bull. Electr. Eng. Informatics*, vol. 7, no. 4, pp. 593–600, Dec. 2018, doi: 10.11591/eei.v7i4.1349.
[4] C. Kelly, N. Pitropakis, A. Mylonas, S. McKeown, and W. J. Buchanan, "A comparative analysis of honeypots on different cloud platforms," *Sensors*, vol. 21, no. 7, Apr. 2021, doi: 10.3390/s21072433.
[5] V. Sethia and A. Jeyasekar, "Malware capturing and analysis using dionaea honeypot," in *2019 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2019, pp. 1–4., doi: 10.1109/CCST.2019.8888409.
[6] W. Cabral, C. Valli, L. Sikos, and S. Wakeling, "Review and analysis of cowrie artefacts and their potential to be used deceptively," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2019, pp. 166–171., doi: 10.1109/CSCI49370.2019.00035.
[7] K. Nam and K. Kim, "A study on SDN security enhancement using open source IDS/IPS Suricata," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2018, pp. 1124–1126., doi: 10.1109/ICTC.2018.8539455.
[8] X. Qin, T. Xu, and C. Wang, "DDoS attack detection using flow entropy and clustering technique," in *2015 11th International Conference on Computational Intelligence and Security (CIS)*, Dec. 2015, pp. 412–415., doi: 10.1109/CIS.2015.105.
[9] T. M. Thang, C. Q. Nguyen, and K. Van Nguyen, "Synflood spoofed source DDoS attack defense based on packet ID anomaly detection with bloom filter," in *2018 5th Asian Conference on Defense Technology (ACDT)*, Oct. 2018, pp. 75–80., doi: 10.1109/ACDT.2018.8593121.
[10] A. Maslan, K. M. Bin Mohamad, and F. Binti Mohd Foozy, "Feature selection for DDoS detection using classification machine learning techniques," *IAES Int. J. Artif. Intell.*, vol. 9, no. 1, pp. 137–145, Mar. 2020, doi: 10.11591/ijai.v9.i1.pp137-145.
[11] A. Fadlil, I. Riadi, and S. Aji, "Review of detection DDoS attack detection using naive bayes classifier for network forensics," *Bull. Electr. Eng. Informatics*, vol. 6, no. 2, pp. 140–148, Jun. 2017, doi: 10.11591/eei.v6i2.605.
[12] M. Idhammad, K. Afdel, and M. Belouch, "Semi-supervised machine learning approach for DDoS detection," *Appl. Intell.*, vol. 48, no. 10, pp. 3193–3208, Oct. 2018, doi: 10.1007/s10489-018-1141-2.
[13] A. J. Mohammed, M. H. Arif, and A. A. Ali, "A multilayer perceptron artificial neural network approach for improving the accuracy of intrusion detection systems," *IAES Int. J. Artif. Intell.*, vol. 9, no. 4, pp. 609–615, Dec. 2020, doi: 10.11591/ijai.v9.i4.pp609-615.
[14] A. W. Muhammad, C. F. M. Foozy, and K. M. bin Mohammed, "Multischeme feedforward artificial neural network architecture for DDoS attack detection," *Bull. Electr. Eng. Informatics*, vol. 10, no. 1, pp. 458–465, Feb. 2021, doi: 10.11591/eei.v10i1.2383.
[15] F. D. S. Sumadi and C. S. K. Aditya, "Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network," in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020*, Dec. 2020, pp. 152–157., doi: 10.1109/ISRITI51436.2020.9315510.
[16] S. K. Dey and M. M. Rahman, "Effects of machine learning approach in flow-based anomaly detection on software-defined networking," *Symmetry (Basel).*, vol. 12, no. 1, Dec. 2019, doi: 10.3390/sym12010007.
[17] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Mar. 2019, pp. 1602–1606., doi: 10.1109/ITNEC.2019.8729425.
[18] M. Du and K. Wang, "An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things," *IEEE Trans. Ind. Informatics*, vol. 16, no. 1, pp. 648–657, Jan. 2020, doi: 10.1109/TII.2019.2917912.
[19] H. Lin, "SDN-based in-network honeypot: preemptively disrupt and mislead attacks in IoT networks," May 2019
[20] X. Luo, Q. Yan, M. Wang, and W. Huang, "Using MTD and SDN-based honeypots to defend DDoS attacks in IoT," in *2019 Computing, Communications and IoT Applications (ComComAp)*, Oct. 2019, pp. 392–395., doi: 10.1109/ComComAp46287.2019.9018775.
[21] W. Tian, M. Du, X. Ji, G. Liu, Y. Dai, and Z. Han, "Honeypot detection strategy against advanced persistent threats in industrial internet of things: a prospect theoretic game," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17372–17381, Dec. 2021, doi: 10.1109/JIOT.2021.3080527.
[22] S. Asadollahi, B. Goswami, and M. Sameer, "Ryu controller's scalability experiment on software defined networks," in *2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, Feb. 2018, pp. 1–5., doi: 10.1109/ICCTAC.2018.8370397.
[23] J. M. Ceron, C. Scholten, A. Pras, and J. Santanna, "MikroTik devices landscape, realistic honeypots, and automated attack classification," in *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, Apr. 2020, pp. 1–9., doi: 10.1109/NOMS47738.2020.9110336.

[24] R. R. S, R. R, M. Moharir, and S. G, "SCAPY-a powerful interactive packet manipulation program," in *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, Dec. 2018, pp. 1–5., doi: 10.1109/ICNEWS.2018.8903954.
[25] Y. Li, R. Miao, M. Alizadeh, and M. Yu, "DetER: deterministic TCP replay for performance diagnosis," in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2019, pp. 437–451.

BIOGRAPHIES OF AUTHORS

Fauzi Dwi Setiawan Sumadi achieved his master degree program in computer science at the University of Queensland, Australia which focused to analyze the vulnerability in software defined network. Nowadays, he becomes one of the main lecturers in Informatics Department at the University of Muhammadiyah Malang and maintains his research in the implementation of artificial intelligence in computer network, distributed computing, Cyber security, IoT, and the SDN. He can be contacted at email: fauzisumadi@umm.ac.id.

Christian Sri Kusuma Aditya graduated with Master of Computer from Sepuluh Nopember Technological Institute (ITS), Surabaya. Currently, he is a lecturer in the Informatics Department University of Muhammadiyah Malang (UMM). His areas of interest are Data Science, Machine Learning, and Text Processing. He can be contacted at email: christianskaditya@umm.ac.id.



Ahmad Akbar Maulana graduated with bachelor's degree of Informatics from the University of Muhammadiyah Malang (UMM), Malang. Currently he is a Fullstack Java Consultant in PT Xsis Mitra Utama, Jakarta. His area of interest is software development, computer network, and DevOps. He can be contacted at email: alanmy.maulana@gmail.com.

Syaifuddin graduated with Master of Computer from Sepuluh Nopember Technological Institute (ITS), Surabaya. Currently, he is a senior lecturer in the Informatics Department University of Muhammadiyah Malang (UMM) and active as a cyber community builder at the local campus and regional levels. His areas of interest are cyber security, network forensics, malware analysis and security analysis. He can be contacted at email: syaifuddin_skom@umm.ac.id.

Vera Suryani works as a Lecturer at School of Computer and Informatics, Telkom University since 2003. She achieved her Ph.D. from the Department of Electrical and Information Engineering Technology, Gadjah Mada University, Indonesia in 2019. Her research interests include computer networks, cybersecurity, distributed systems, and Internet of Things security. She can be contacted at email: verasuryani@telkomuniversity.ac.id.

ty_Distributed_denial_of_service_Honeypot_Software_define...

ORIGINALITY REPORT

13%	12%	8%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

4%

★ Tiara Intana Sari, Zalfa Natania Ardilla, Nur Hayatin, Ruhaila Maskat. "Abusive comment identification on Indonesian social media data using hybrid deep learning", IAES International Journal of Artificial Intelligence (IJ-AI), 2022

Publication

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		