Towards Analysis of the Performance of IDSs in Software-Defined Networks

Nadia Niknami[1], Emily Inkrott[2], and Jie Wu[1]
[1] Department of Computer and Information Science, Temple University
[2]Department of Computer Science, Gonzaga University

2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 978-1-6654-7180-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MASS56207.2022.00124

***Abstract*—As a promising technique for the design of 5G wireless networks, software-defined networks (SDNs) have been proposed. However, SDNs are vulnerable to most of the attacks that traditional networks are vulnerable to. Various techniques have been developed and designed to help in the detection as well as the prevention of various attacks. An intrusion detection system (IDS) is one of the common techniques used to detect malicious activity in a network. Intrusion detection systems have strengths and weaknesses when it comes to detecting intrusions. It becomes a challenging task for IDS to process any mixture of traffic that results in packet drop and delay. In this study, we scrutinized two open-source IDS, including Snort IDS and Zeek IDS, to assess the IDS performance in terms of various parameters such as detection rate, dropping rate, and latency. The method of detection was one of the main differences between Snort and Zeek. Zeek IDS uses an anomaly-based detection method as opposed to Snort IDS, which uses a signature-based detection method. Differences between them had an impact on the way network traffic was handled. Such a thought analysis is expected to be of great value in selection and further enhancement of IDS in SDN.**

***Index Terms*—Denial-of-service, detection rate, intrusion detection system, network traffic, software-defined network.**

I. Introduction

In network technology, software-defined networks (SDN) is a new paradigm using a central controller [1] [2]. SDN will enable the expansion of IoT devices, increase network resource sharing efficiency, and improve IoT service-level agreements. Additionally, the 5G wireless network infrastructure will be based on SDN, which enables communication between cloud-based applications and services, as well as between users' mobile devices. With resource virtualization, the network can be dynamically managed according to real-time requirements. There are, however, the same threats in SDN as there are in traditional networks. For SDN, security administration plays a crucial role in network management. Intrusion detection systems (IDSs) are primarily designed to ensure the availability, confidentiality, and integrity of critical information systems in a network. There are many types of IDS available, both commercial and open source. In view of the fact that most commercial intrusion detection systems cost thousands of dollars and entail significant resource requirements, their use is not feasible for small networks. It is, therefore, mostly

This work was supported by the NSF grant CNS-1757533 as part of the Research Experiences for Undergraduates (REU) program. Research was facilitated by Temple University during the Pervasive Computing for Smart Health, Safety, and Well-being REU.

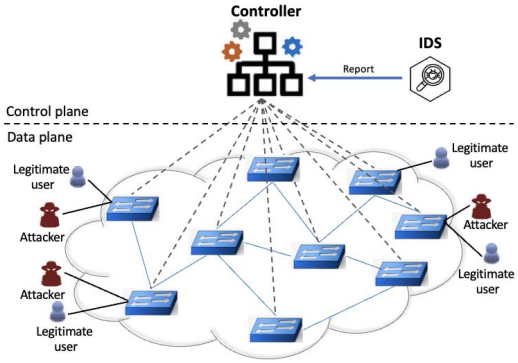


Fig. 1: Anomaly detection in SDN.

open source IDS that are being used. It is possible to identify unauthorized usage and mishandling by attackers in computer network systems by using a *intrusion detection system* (IDS) [3]. SDN uses Openflow [4] which is a new network technology, and it refers to an open standard in which the control plane and data plane of network equipment are separated. Therefore, Openflow provides a protocol for programming flow tables in different switches and routers. In traditional networks, the core is stateless, and each packet is serviced individually. However, in an SDN, the concept of a stateful core with flows and centralized control can be exploited to improve security features. SDN will be very effective in monitoring traffic due to the direct control it would have over all networks, and if an attack occurs, the attack can be discovered immediately due to the implementation of IDS. The implementation of IDS in SDN will be easier than in other networks because IDS will be able to monitor all devices on the network [5]. Many open-source IDSs exist, including Snort and Bro, which are considered to be among the most reliable IDS technologies. The dilemma, however, is determining which is most effective in detecting intrusions. Analyzing different types of IDS for the purpose of presenting an independent evaluation of their effectiveness in detecting various threats is the proper procedure.

Snort [6] and *Zeek* [7] are two of the most popular network-based IDS tools used for traffic analysis. Snort is a widely used signature-based intrusion detection system that supports both IDSs and *intrusion prevention system* (IPS) modes. In order to detect malicious packets, the detection engine applies rules to the traffic. Depending on whether the rules match, it is capable of monitoring network traffic, comparing packets against

2155-6814/22/$31.00 ©2022 IEEE
DOI 10.1109/MASS56207.2022.00124

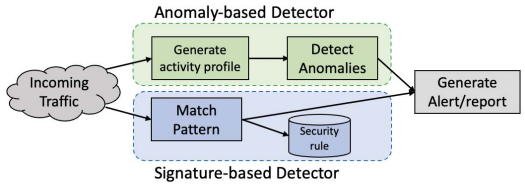


Fig. 2: Signature-based detection vs anomaly-based detection.

signatures, logging attacks, and presenting attack statistics on the console. By default, Snort only supports misuse detection and does not support anomaly-based detection. In IDS mode, it only generates alerts based on detection, whereas in IPS mode, it blocks malicious packets. In the output block, a text file is generated for the user to view later. Zeek is another open-source NIDS tool that supports only IDS mode. A Zeek framework deploys agents known as workers on network devices, and these workers communicate their logs with the manager. An event engine in the Zeek manager converts every packet that arrives from the network into an event, which is then passed to the next component, which is the policy script interpreter. As a result of the function of the policy script interpreter, the Zeek rules are applied to the events generated by the event engine. This may result in alerts in the event of malicious activity being detected. With the Zeek architecture, performance improvements are easily achieved by allocating more hardware resources to the workers and the manager. As a zero-day attack detection system, Zeek is particularly useful in environments where anomaly-based detection is required - a feature is not available in Snort. Zeek is quite efficient, easy to deploy, and flexible, but it has a limited set of default rules or signatures, which limits its widespread use. Conversely, Snort has a much broader rule base and a community that actively contributes to its development. Thus, it is generally accepted for deployment. The contributions of this study are:

- We conduct a method to evaluate effectiveness of an anomaly detection-based IDS and a signature-based IDS under different attacks in a SDN.
- We study how do open-source IDSs, such as Snort and Zeek, under different detection methods respond to specific network attacks?
- We provide a comprehensive and detailed comparison of the two types of IDSs in terms of detection rate, dropping rate, and delay with different types of attack traffic and different amounts of incoming traffic.
- We investigate the levels of false positive and false negative alarms generated by open-source IDSs such as Snort and Zeek for normal and malicious network traffic.
- We analyze the effect of integration of signature-based IDS and anomaly detection-based IDS on detecting attacks.

II. Related work

It is widely believed that SDN will be the next leading networking platform, and studies pertaining to SDN have been actively conducted as a result [8]. Due to SDNs' vulnerabilities, research has been done addressing how best to protect the controller when it is targeted. Li *et al.* in [9] implement a system for monitoring and mitigating attacks within the control plane. This approach proved to be effective, with low overhead and no additional hardware requirements. However, the method used to defend against attacks resulted in blocking the suspicious hosts, rather than the traffic itself, which could result in the blocking of non-malicious traffic. Others have looked into the viability of IDS in protecting SDNs. Ahmad *et al.* in [10] perform a survey of different varieties of IDS, such as whether they are network-based or host-based, signature-based or anomaly-based, the various methods of anomaly detection used by IDS, and the strengths and weaknesses of different approaches. Likewise, Hraisat *et al.* in [11] survey the different types of IDS, and additionally detailed the various datasets used for training and testing anomaly-based detection systems.

Parameter	Snort	Zeek
Installation/deployment	Easy	Typical
Intrusion prevention capability	Yes	No
Network traffic	IPv4/IPv6	IPv4
Intrusion detection method	Signature-based	Anomaly detection
Support high speed network	Medium	High

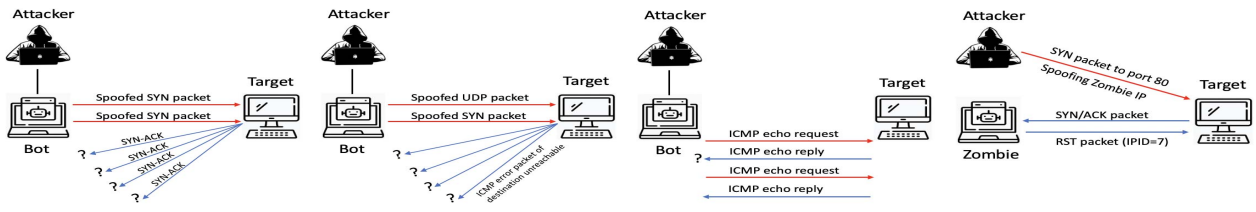
TABLE I: Comparison table of Snort and Zeek IDSs

Fernande *et al.* in [12] provide more specific details on approaches to anomaly detection, and describe the different varieties of network anomalies. Research has also been conducted into the different kinds of distributed DoS attacks that pose a significant threat to SDNs. For example, which IDS might be most effective in defending against DoS attacks. Swami *et al.* in [13] provide an overview of how different distributed DoS attacks are performed and the dangers and drawbacks of each one. Manso *et al.* in [14] aim to defend against these kinds of attacks by monitoring the data plane using IDS and notifying the SDN controller when an attack occurs. Tests were conducted against three different scenarios and demonstrated that there was no packet loss, however many alerts could cause congestion in the SDN controller. Hendrawan *et al.* in [15] compare two intrusion detection tools under some QoS measurements such as throughput, delay, packet loss, CPU usage, and memory usage. However, they did not consider different kinds of malicious traffic. Also, they did their experiment on Mininet. Nevertheless, in our study, we work to examine the performance of different kinds of IDSs against various DoS attacks and analyze whether they can be effectively combined to provide stronger protection.

III. Background

A. Software-Defined Networks

Our society is permeated by networks, which are integral to modern technology. To address the diverse array of needs in our expanding technological landscape, new network technologies have been developed. One such development is the software-defined network (SDN) [16]. In a traditional network, each switch independently determines the best way to route traffic, either by referencing a table of known flows,



(a) SYN flooding (b) UDP flooding (c) ICMP flooding (d) Port scanning

Fig. 3: Different kinds of attacks

or by using routing algorithms to expand the flow table for a given packet. In SDN, there is a physical separation of the network control plane from the forwarding plane, where a control plane controls several devices. The separation of the data plane, which consists of the switches that forward traffic, and the control plane, which creates new flow rules when switches do not know how to handle packets, has several advantages. The *centralized controller* that interfaces with all the switches and provides new flow rules to switches based on an omniscient view of the network. First, in addition to routing traffic with context, the SDN controller can avoid compromised switches and route potentially malicious traffic through additional security layers. Furthermore, the controller can be interfaced with software, so any updates or adjustments to the network can be performed without interacting with hardware. However, there are also security risks that come with the benefits of an SDN. Because the controller has access to all the switches, it makes a very compelling target for attackers. A compromised controller can allow attackers to manipulate the entire network at once. Additionally, the controller is especially vulnerable to *denial of service* (DoS) attacks. Since the controller needs to process Packet-In messages from all the switches, determine a course of action, and respond with appropriate flow rules, high volumes of malformed traffic can easily overload the controller and render the network unusable for legitimate users. This is true for *distributed denial of service* (DDoS) attacks, which uses methods such as botnets to flood the network from multiple sources which makes the attack more difficult to detect and mitigate.

B. Types of IDS

DoS attacks, and many others, can be detected by *intrusion detection systems* (IDS) installed throughout the network. IDSs monitor traffic on the network or on a specific host and generate alerts when potentially malicious traffic is detected.

- *Host-based intrusion detection systems* (HIDS) are installed on a particular host and they monitor network traffic as it is sent to and from the host.
- *Network-based intrusion detection systems* (NIDS) are installed on switches in the network to monitor traffic as it is routed through the switch. When an NIDS generates an alert, the controller can determine how to handle the malicious traffic and respond with appropriate flow rules.

Intrusion detection systems employ a variety of intrusion detection methods. Fig. 2 illustrates the difference between two methods of intrusion detection. These methods can be described as follows:

- *Signature-based detection*: This relies on prior knowledge of different attacks in order to spot malicious traffic. Usually, this comes from a database of known attacks that are kept up to date on the latest threats. These IDSs are consistent in recognizing known attacks but have the disadvantage of not being able to recognize zero-day attacks that are not yet in their database [17].
- *Anomaly-based detection*: This makes use of machine learning and statistical approaches to classify traffic as "normal" or "anomalous". These IDSs have the advantage of being able to detect zero-day attacks, but can generate more false positives when handling legitimate traffic that deviates from normal network activity [18].

In our experiments, we examine the responses of Snort, a signature-based IDS, and Zeek , an anomaly-based IDS. Snort uses rule-sets that can be downloaded from their website to monitor for specific traffic. Different rule-sets can be enabled and disabled to provide protection against specific threats, but enabling too many rule-sets can consume resources and slow down the IDS. For our experiment, we used only the default Snort rule-set. Zeek, on the other hand, has no hard-coded rule-sets, and evaluates network traffic in real-time to flag malformed or malicious flows. Table I shows some differences between these two IDSs.

C. Different Types of Attacks

In our experiment, we staged several attacks against a server being monitored by an IDS. Some of these attacks fall under the category of denial of service attacks, which aim to overwhelm the network and make it unusable for legitimate users. The remaining attacks are all variations of *port scanning*, which aims to map the status of ports on a network. A port scanning operation attempts to determine which applications are running on which ports, which ports are being filtered, and in some instances, how the firewall is configured. The following are some types of cyberattacks:

1) SYN Flooding Attack: SYN flooding is an attack that exploits the TCP handshake protocol to consume the network's resources and exhaust its connections for legitimate users [19]. The attacker initiates the handshake protocol by sending a SYN packet to the target with a spoofed IP address. The target responds with a SYN-ACK packet, as the handshake protocol requires, but does not get the responding ACK packet since the original IP address was spoofed. The target waits for a

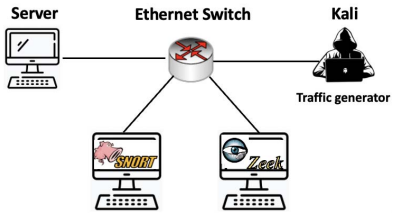
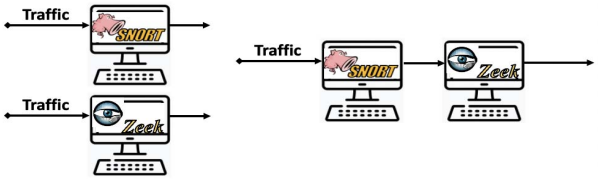


Fig. 4: Testbed setup for IDS mode.



(a) One IDS (b) Sequence IDSs

Fig. 5: Different deployment of IDS.

response with an open port, which consumes resources. Since this is a flooding attack, the attacker floods the target with SYN packets, causing this exchange to occur repeatedly until the target's resources are exhausted. Fig. 3 part (a) illustrates this type of attack.

2) UDP Flooding Attack: A large number of User Datagram Protocol (UDP) packets are sent to a targeted server to cripple its function. It is intended to overwhelm the target to the point at which it is unable to respond to legitimate requests. The target then checks for an application listening to the port that received the datagram, and upon finding none, sends an ICMP error packet in response to the spoofed IP address [20]. Fig. 3 part (b) illustrates this type of attack.

3) ICMP Flooding Attack: ICMP flooding is when the attacker floods the target with ICMP echo requests, or pings. Each ping requires an echo response, consuming both incoming and outgoing bandwidth for the target. Generally, this is executed with the help of a botnet, so the attacker does not suffer from the same consumption of bandwidth that the target is forced to deal with [21]. Fig. 3 part (c) illustrates this type of attack.

4) DoS Attack: A denial-of-service attack occurs when legitimate users cannot access information systems, devices, or other network resources as a result of the actions of a malicious cyber threat actor [22]. There are many ways to carry out DoS attacks. Network servers are commonly attacked by flooding them with traffic. The attacker overloads the target server with traffic by sending several requests. Authenticating the requestor is difficult due to the illegitimate service request and fabricated return address. Continual junk requests overwhelm the server, resulting in a DoS condition.

5) Port Scanning Attack: Port scanning is done by sending requests to the target's different ports and using the responses to map the ports' statuses [23]. Each port is either open, closed, or filtered. That information can be used by attackers to find open and unused ports that they can use for purposes of infiltration. The following are typical scanning methods:

- SYN scans, also known as TCP half open scans, sends SYN packets to a port, prompting a SYN-ACK response. This utilizes the TCP handshake protocol in a way that resembles SYN flooding, since the third part of the three-way handshake, the ACK packet, is never sent to the target. SYN-ACK responses indicate open ports. RST responses indicate closed ports. ICMP responses or nonresponses indicate filtered ports.
- TCP Connect scans are similar to SYN scans, but they complete the three-way handshake by sending the final ACK response. This is more likely to be detected by an IDS, since the connection is completed, but does not require higher privileges than the average user in the way that SYN scans do. This makes them both easier to detect and easier to execute.
- Ping scanning is the simplest kind of port scanning, done by sending ICMP echo requests to each port and listening for a response. The response usually lacks details. The only information that can be acquired using ping scans is whether a computer is on the other end of the pinged port.
- TCP ACK scans are used to map firewall rule-sets, informing attackers which ports are filtered and whether a firewall is stateful. The attack sends ACK packets, and receives RST packets from both open and closed ports that are unfiltered. From there, the attacker can learn which ports are being filtered by examining which ports did not give a response.

IV. Effectiveness of Different kinds of IDSs

This study was conducted using the experimental research method. We have conducted several experiments in this study in order to test and compare the performance and accuracy of two open-source intrusion detection systems, namely Snort and Zeek. In the experiments, the effectiveness of these IDSs at detecting attacks was compared in order to evaluate their efficacy. In order to assess the accuracy of the IDSs, we captured and analyzed network traffic available in all the IDSs under consideration and compared the alarms that were generated. The testbed is depicted in Fig. 4. It consisted of a traffic generator, target host, Snort IDS, and Zeek IDS. For the purpose of generating malicious traffic, Kali Linux version 2.0 was also used. The legitimate traffic was generated by using Ostinato traffic generator. The Ostinato [24] as a network traffic generator can be used in normal mode and burst mode to generate legitimate traffic.

By using iperf [25], we analyzed the requests and responses from the virtual machine using the detection engines of both IDSs. It is important to note that both malicious and legitimate traffic were combined and used as inputs to the two IDSs, namely Snort and Zeek. We investigated the effectiveness of collaborative intrusion detection systems as well. Fig. 5 displays different combinations of IDS. Fig. 5 (a) illustrates a scenario in which each flow passes through one IDS. Fig. 5 (b) shows the scenario in which flows go through the IDS sequence, including both Snort and Zeek. It is possible

TABLE II: Reaction of Snort and Zeek against different types of attacks

IDS	Attacks									
	ICMP		SYN		UDP		DoS		Port Scan	
	Detection	Flag	Detection	Flag	Detection	Flag	Detection	Flag	Detection	Flag
Snort	Yes	Not Bad	Yes	Bad	Yes	Bad	Yes	Bad	Yes	Not Bad
Zeek	No	Not flagged	Yes	Weird	Yes	Weird	Yes	Not flagged	Yes	Not flagged

for an IDS to identify normal traffic as malicious traffic, which is termed as a False Positive (FP), or even malicious traffic as normal, resulting in a False Negative (FN). When FPs are many, they may easily conceal real attacks. The metrics that we consider in the evaluatation of different IDSs are as follows:

- Detection rate The detection rate is calculated as the ratio between the number of correctly detected attacks and the total number of attacks. We can formulate detection rate as $TP/(TP+FN)$. The number of attacks that are detected by an IDS is known as True Positive (TP) while the number of attacks that are missed by an IDS is known as False Negative (FN). As a result, the total number of attacks in the system can be calculated. By measuring the detection rate, it is possible to determine how accurate the system is.
- Dropping rate Packet loss occurs when packets of data fail to reach their destination. This occurs due to the fact that packets outnumber the capacity and resource of the processor. The higher the packet loss, the worse the network is constructed. There are several factors that may contribute to packet loss, including crash packets, high traffic volumes, and the loss of signal when using a non-wired network.
- Delay Delay of data packet is the amount of time it takes from the time it begins to be delivered until it reaches its destination. The higher the delay, the worse the network is designed. There are several factors that affect the delay, including the distance between nodes, the network topology, the density of network traffic, and the size of data packets.

As part of the evaluation, both IDSs have been subjected to heavy and mixed traffic attacks. Snort alerts include any anomalous network traffic and suspicious connections reportings. Snort writes log entries and each entry contains the date and time of the event, the packet header, a description of the type of breach that was detected, and a severity rating. Zeek IDS's logging system is split between multiple different files. We checked at three specific files to get these preliminary results.

- conn.log: this file logs all connections. When the attack appeared in this file, we can count it as being detected.
- weird.log: this file logs "weird" traffic - anything that is potentially malicious or malformed goes here. When an attack is classified as "weird" below, it is because it appeared on weird.log but was not converted into a notice.
- notice.log: this file logs notices made by the IDS which are equivalent to sending an alert.

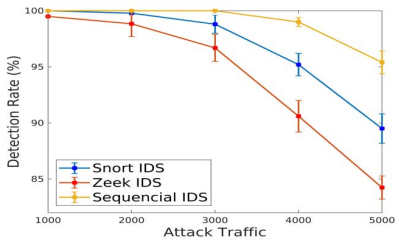


Fig. 6: Effectiveness of sequence IDS.

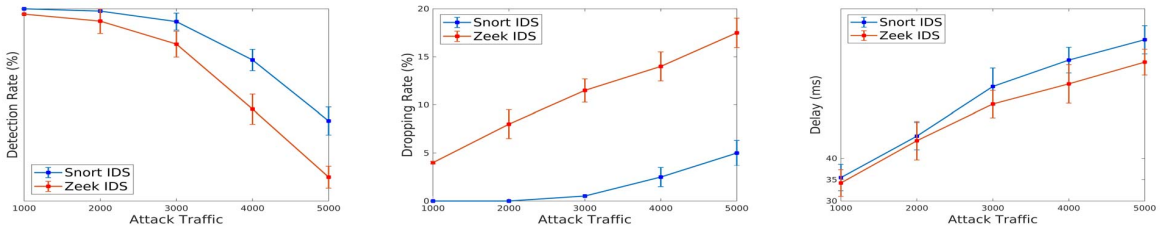
A. Results

Table II lists the result of attack detection based on the alarms generated by two IDSs in response to various types of attacks. The results indicate that both IDSs can detect an attack on the network when there is a port scanning attack. However, they do not consider this kind of traffic to be malicious. As a result, Snort and Zeek produce False Negative (FN) results for such malicious traffic. Both IDSs are capable of detecting UDP flooding attacks successfully. As for the ICMP flooding attack, Snort detects the traffic, but does not classify it as dangerous. On the other hand, Zeek has no ability to identify this traffic. In terms of detecting DoS attacks, Snort provides a better detection rate than Zeek, as it is able to identify malicious traffic. It should be noted that these are the results for a small volume of attack traffic.

Fig. 6 shows the effectiveness of providing two IDSs, includes Snort and Zeek, in a sequence. It can be concluded that sequence IDS can improve the attack detection rate even when there are large volumes of attack traffic. The second IDS can assist the first one in detecting attack traffic that was not detected by the first IDS, thus increasing the detection rate.

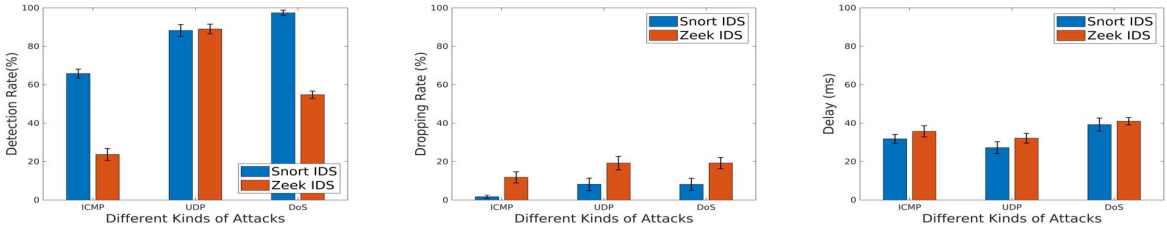
Fig. 7 presents the effectiveness of Snort IDS and Zeek IDS under different amounts of attack traffic. The results concerning detection rate shows that Snort performs better than Zeek under large volumes attack traffic. The average result for packet loss measurement shows that Snort is better than Zeek in the case of monitoring all the passing packets. In the case of delay, they are close, but under large traffic, Snort performs better than Zeek.

Fig. 8 illustrates how Snort IDS and Zeek IDS perform with respect to detecting different types of attacks. For UDP flooding attacks, Snort has similar detection rates to Zeek, however Zeek has a greater dropping rate. As a result of the average results, Snort is able to detect DDoS attacks more accurately than Zeek and with a smaller dropping rate than Zeek. Their performance is similar in terms of delay. Based on the average result, Zeek is less effective at detecting ICMP flooding attacks than Snort. There can be no doubt that when it comes to effectiveness, Snort is more effective than Zeek in



(a) Detection rate (b) Dropping rate (c) Delay

Fig. 7: Detection rate, dropping rate, and delay under different amount of incoming attack traffic.



(a) Detection rate (b) Dropping rate (c) Delay

Fig. 8: Detection rate, dropping rate, and delay for different kinds of attacks under mixture traffic.

terms of the detection rate and the types of attacks it detects.

V. Conclusion

The experiments in this study were carried out to evaluate the performance of different IDSs by comparing how effective the IDSs were in detecting attacks and in dropping rate. Based on the results, it can be concluded that in the case of having single IDS, Snort IDS can be said to be above Zeek IDS in the case of detection rate, dropping rate, but not for delay. Under default configuration, neither Snort IDS nore Zeek IDS detect the port scanning attack. As future work, it is recommended to improve the rule-sets of IDS to increase the accuracy of port scanning attack detection. It is also recommended to evaluate sequence IDS under different mixture of traffic as well as different kinds of traffic.

References

[1] T. Wang and H. Chen, "Sguard: A lightweight sdn safe-guard architecture for dos attacks," *China Communications*, vol. 14, no. 6, pp. 113–125, 2017.

[2] J. H. Cox, J. Chung, S. Donovan, J. Ivey, R. J. Clark, G. Riley, and H. L. Owen, "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25 487–25 526, 2017.

[3] D. S. Vijayarani and M. M. Sylviaa, "Intrusion detection system–a study," *Privacy and Trust Management (IJSPTM) Vol*, vol. 4, 2015.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[5] J. Ibrahim and S. Gajin, "Sdn-based intrusion detection system," *Infoteh Jahorina*, vol. 16, pp. 621–624, 2017.

[6] [Online]. Available: https://www.snort.org/

[7] [Online]. Available: http://bro.org/

[8] K. Shipulin, "We need to talk about ids signatures," *Network Security*, vol. 2018, no. 3, pp. 8–13, 2018.

[9] J. Li, T. Tu, Y. Li, S. Qin, Y. Shi, and Q. Wen, "Dosguard: Mitigating denial-of-service attacks in software-defined networks," *Sensors*, vol. 22, no. 3, p. 1061, 2022.

[10] A. A. Ahmad, S. Boukari, A. M. Bello, and M. A. Muhammad, "A survey of intrusion detection techniques on software defined networking (sdn)."

[11] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[12] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019.

[13] R. Swami, M. Dave, and V. Ranga, "Ddos attacks and defense mechanisms using machine learning techniques for sdn," in *Research Anthology on Combating Denial-of-Service Attacks*, 2021, pp. 248–264.

[14] P. Manso, J. Moura, and C. Serrão, "Sdn-based intrusion detection system for early detection and mitigation of ddos attacks," *Information*, vol. 10, no. 3, p. 106, 2019.

[15] H. Hendrawan, P. Sukarno, and M. A. Nugroho, "Quality of service (qos) comparison analysis of snort ids and bro ids application in software define network (sdn) architecture," in *Proc. of 7th Intl. Con. on Information and Communication Technology (ICoICT)*, 2019, pp. 1–7.

[16] S. Singh and R. K. Jha, "A survey on software defined networking: Architecture for next generation network," *Journal of Network and Systems Management*, vol. 25, no. 2, pp. 321–374, 2017.

[17] Y. Otoum and A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things," 2021.

[18] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly- and signature-based ids for network security using hybrid inference systems," *Mathematical Problems in Engineering*, 2021.

[19] M. Bogdanoski, T. Suminoski, and A. Risteski, "Analysis of the syn flood dos attack," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 5, no. 8, pp. 1–11, 2013.

[20] S. S. Kolahi, K. Treseangrat, and B. Sarrafpour, "Analysis of udp ddos flood cyber attack and defense mechanisms on web server with linux ubuntu 13," in *Proc. of Intl. Conf. on Communications, Signal Processing, and their Applications (ICCSPA'15)*, 2015, pp. 1–5.

[21] M. Bogdanoski and A. Risteski, "Wireless network behavior under icmp ping flooddos attack and mitigation techniques," *Intl. Journal of Communication Networks and Information Security (IJCNIS)*, vol. 3, no. 1, 2011.

[22] Z. Chao-yang, "Dos attack analysis and study of new measures to prevent," in *Intl. Conf. on Intelligence Science and Information Engineering*, 2011, pp. 426–429.

[23] M. u. Nisa and K. Kifayat, "Detection of slow port scanning attacks,"

in *Proc. of Intl. Conf. on Cyber Warfare and Security (ICCWS)*, 2020, pp. 1–7.
[24] [Online]. Available: https://ostinato.org/
[25] [Online]. Available: https://iperf.fr/iperf-doc.php