Evaluating Performance of Intrusion Detection Systems under Different Configurations in SDN

Dennis Yeom
dennis.yeom@temple.edu
Temple University
Philadelphia, PA, USA

Nadia Niknami
Temple University
Philadelphia, PA, USA

Jie Wu
Temple University
Philadelphia, PA, USA

ABSTRACT

Software Defined Networks (SDN) have been proposed as a possible development for next generation networking technology. In a SDN, Virtual Network Functions (VNFs) are used to replace the functions of traditional middleboxes. All of these VNFs can be controlled from a centralized controller, which comes with its own security concerns. However, there are many benefits that come from having a centralized controller as traffic can be easily controlled from a single point. This makes it interesting to further study security when it concerns SDN. This work looks to test intrusion detection system (IDS) performance under different configurations in a SDN in order to make security in SDN more robust. In this work, we take two IDSs, Snort and Suricata, and test their performance under different configurations and traffic loads. We test four different configurations: single, chain, parallel, and cross. Our findings seem to suggest that the cross configuration has the best performance of these IDS configurations.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**.

KEYWORDS

Intrusion detection systems (IDS), Snort, Suricata, detection rate, packet loss.

ACM Reference Format:
Dennis Yeom, Nadia Niknami, and Jie Wu. 2023. Evaluating Performance of Intrusion Detection Systems under Different Configurations in SDN. In *The Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '23), October 23–26, 2023, Washington, DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3565287.3617614

1 INTRODUCTION

As technology continues to advance, an increasing number of companies are turning to Software Defined Networking (SDN) to enhance their network capabilities. In addition to firewalls and routers, middleboxes can be replaced with Virtual Network Functions (VNFs) that are controlled through a centralized program, creating a more streamlined SDN infrastructure [16]. Virtualizing these resources offers greater flexibility in meeting real-time demands while also addressing the cost considerations associated with traditional middleboxes. Commercially available middleboxes often incur substantial expenses and rely on proprietary software, which can be overcome by adopting open-source alternatives. By implementing open-source VNFs like Snort in an SDN framework, operational and maintenance costs can be significantly reduced, eliminating the need for physical visits to individual middleboxes. Instead, the SDN's centralized controller provides a single point of control for managing the entire network efficiently [4]. Furthermore, SDN allows for quick and easy implementation of changes and modifications to the network.

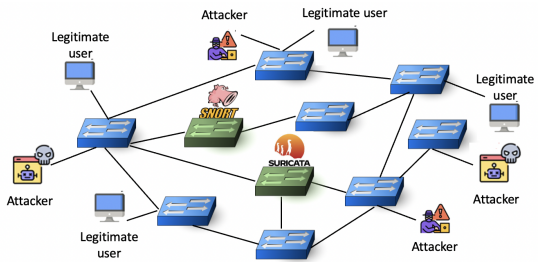


Figure 1: IDS in a network. IDS deployed on green switches.

Like traditional networks, software-defined networks are also susceptible to security breaches, and it is essential to implement appropriate measures to ensure network security[11][12]. One such measure is the use of an Intrusion Detection System (IDS), which plays a vital role in safeguarding the network. The primary function of an IDS is to monitor all incoming network traffic and analyze individual packets. Upon detecting any malicious traffic, the IDS promptly alerts the user and logs the incident. This enables the user to take necessary actions to address potential security threats effectively. Essentially, the IDS acts as a vigilant alarm system for the network, alerting users to potential attacks as traffic passes through. Without the presence of an IDS, identifying the source of attacks and troubleshooting network issues in real-time would be a daunting task. Hence, the accuracy and continuous operation of the IDS are of utmost importance.

As shown in Fig. 1, IDSs are installed as Host-based Intrusion Detection Systems, allowing them to analyze all traffic passing through the green switches. Whenever suspicious activity is detected, an alert is promptly sent to notify the user, enabling an appropriate response. Deploying an IDS in a software-defined network is crucial for maintaining a secure network environment and promptly detecting and mitigating potential threats. By leveraging

MobiHoc '23, October 23–26, 2023, Washington, DC, USA

ACM ISBN 978-1-4503-9926-5/23/10…$15.00
https://doi.org/10.1145/3565287.3617614

the benefits of open-source IDS solutions, organizations can bolster their network security while also optimizing operational costs. When considering how to configure IDSs in a network, there are many options we can take. One group was able to achieve higher performance for their IDS when using parallel IDSs[13].

A critical question is raised: What is the optimal configuration for IDSs within a network? It is imperative to identify a configuration that enables IDSs to efficiently and effectively identify malicious network traffic. To address this question, several factors must be considered, including the extent to which traffic is analyzed, dropped, and identified as malicious. The goal of this paper is to determine the most effective IDS configurations and ascertain which setups exhibit the highest reliability and performance, building upon previous research conducted in a similar laboratory setting [10]. By identifying the most suitable IDS configurations, we aim to enhance SDN security.

2 RELATED WORK

Singh *et al* [14] highlight the numerous advantages of SDN. The dynamic nature of SDNs allows users to modify their networks based on real-time requirements, making it an appealing option for companies seeking enhanced flexibility. The benefits cited include cost reduction, improved scalability, and enhanced efficiency. Cox *et al* [4] further discuss how SDNs offer centralized control, enabling users to dynamically and flexibly manage all network devices through a centralized controller. This approach has been adopted by tech giants like Google and Microsoft. However, it is crucial to recognize that SDNs also face specific security threats different from traditional networks. Li *et al* [8] focus on the significant threat posed by DoS attacks on SDNs and propose methods to detect and defend against such attacks. The vulnerability of the centralized controller as a prime target for attackers is acknowledged, given its ability to control all network traffic. To address this, Li's group implemented DoSGuard as a defense mechanism against DoS attacks, which involves blocking the host responsible for originating malicious packets. In a related research endeavor, a member of our research group tested various SDN types by implementing IDSs within an SDN and investigating the effectiveness of these IDSs in distinguishing between normal and malicious packets [10].

IDSs play a critical role in identifying individual malicious packets, enabling subsequent actions to block malicious traffic while permitting normal traffic to flow unhindered. The performance of IDSs was assessed under different traffic loads and various types of attacks. Ahmad *et al* [1] conducted work comparing the performance of signature-based and anomaly-based IDSs. Both types showed strengths and weaknesses, with previous claims suggesting that signature-based IDSs might be unable to detect zero-day attacks. However, experiments by Holm and Hannes [6] suggest that signature-based IDSs are indeed capable of detecting zero-day attacks.

In summary, SDNs offer a range of benefits, but they also face specific security challenges. Research efforts have been dedicated to developing effective defense mechanisms against threats like DoS attacks and assessing the performance of different types of IDSs in SDN environments. These endeavors contribute to making SDNs more secure and reliable in the face of evolving cyber threats.

3 BACKGROUND

3.1 Software-Defined Network (SDN)

Modern networks have become an integral part of our daily lives, facilitating efficient information transfer and accessibility. Among the latest advancements in networking technology, SDN has emerged as a promising innovation [14]. In a conventional network, physical switches autonomously determine the optimal routing of traffic based on flow tables and routing algorithms. However, in SDN, these switches are centrally controlled by what is known as the centralized controller. The centralized controller governs the flow of traffic and can dynamically alter how it traverses through the switches, offering unprecedented flexibility in traffic manipulation [17]. This level of control also enables the rerouting of traffic to bypass compromised hardware, ensuring continued functionality in the event of a hardware breach. Furthermore, the ability to interact with all switches through the centralized controller allows remote configuration without needing on-site physical interaction with hardware [10].

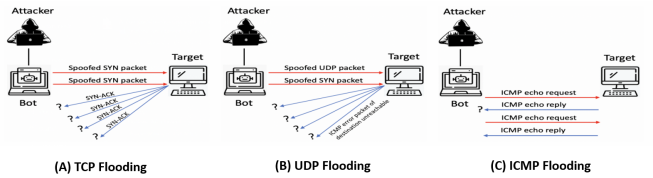
However, SDNs introduce unique security concerns when compared to traditional networks. As the centralized controller governs all traffic within an SDN, unauthorized access to this controller would grant attackers complete control over network traffic [10]. Consequently, it is imperative to implement robust security measures to safeguard the centralized controller from potential breaches. It has been observed that centralized controllers are particularly vulnerable to DoS attacks, and IDSs may struggle to perform effectively under high traffic loads [5].

3.2 Intrusion Detection Systems (IDS)

IDS can be implemented either as dedicated devices or software solutions. IDSs are strategically placed within a network to capture and analyze all incoming traffic. There are two main types of IDSs: signature-based and anomaly-based [9]. A signature-based IDS examines incoming traffic by comparing it to a library of known attack signatures. If a match is found, indicating a pattern similar to known attacks, an alert is triggered. On the other hand, an anomaly-based IDS establishes a baseline of normal traffic and flags any traffic that deviates from this baseline as a potential attack, prompting an alert for further action. IDSs are prone to generating false positives and negatives, which can be attributed to the rule sets used by the IDSs [5]. Therefore, optimizing the IDS configuration to enhance its performance and accuracy is crucial. Performance evaluation of an IDS can be measured by considering metrics such as false positives, false negatives, dropped packets, and correctly identified packets. Proper configuration of the IDS is vital to achieve optimal performance. The rules must be carefully tuned to efficiently detect malicious traffic while minimizing false positives and negatives. In our experiment, we utilized Snort and Suricata as the two IDSs under investigation.

3.3 Types of Attacks

One of the main attacks performed is a form of Denial of Service attack which involves sending many packets at a fast rate that overwhelms the network and makes services unavailable[3]. Another type of attack is a port scan. A port scan is an attempt to gain



(A) TCP Flooding (B) UDP Flooding (C) ICMP Flooding

Figure 2: Three types of attacks. Adapted from paper [10].

information about the status of a target port. The attacker hopes to gain information such as the state of the port, the operating system of the target, and the packet filters in place. There are also options for changing flags that are used while executing the commands to generate packets [15].

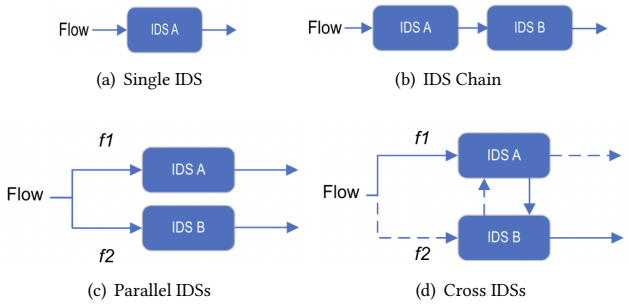
Fig. 2 gives us a visual representation of what TCP, UDP, and ICMP attacks would look like. We utilized three different flags available for the packets: SYN Flag, UDP Flag, and ICMP Flag. Using these flags, we can combine them with the different attack commands to generate different attacks, such as:

(1) SYN Flooding: SYN flooding involves sending as many SYN packets to a target. When a SYN packet is sent, a response in the form of a SYN-ACK packet from the target is expected. After the response SYN-ACK packet is sent, a response from the original host is expected in the form of an ACK packet. However, if the original host spoofed their IP, that final ACK packet will never be received, leaving that port occupied[2].
(2) UDP Flooding: UDP flooding involves sending as many UDP packets to a target. Once the target receives the UDP packets, it will send back responses in the form of ICMP packets. If the original IP is spoofed, the responses get sent off into nowhere. With high enough rates of traffic, it is possible to disrupt the target network and possibly even cause the entire system to crash[7].
(3) ICMP Flooding: ICMP flooding involves sending as many ICMP ping requests as possible to a target. Each ping elicits an echo response from the target, and the goal is to occupy all bandwidth of the target. This ends up denying legitimate users bandwidth needed to access services. These attacks also end up consuming resources of the host as well[10][2].

Attacks are based on the concept of consuming as many of the target's resources as possible. Normally the user will send a SYN packet to the target, which will prompt a SYN-ACK response from the target. Lastly, the user responds with his own ACK, and the connection is set. If the user were to spoof their address, the target would send their SYN-ACK off to nowhere. This means that step 3 never occurs and the port will remain open, waiting for a response that will never come. If we were to flood the target with multiple SYN packets by initiating TCP with a spoofed address, resources would be rapidly consumed, and services would be unavailable for normal users. This idea of spoofing your IP and initiating TCP, UDP, or ICMP in order to occupy resources is the running theme of these attacks.

4 PROPOSED METHODOLOGY

A single Intrusion Detection System proves inadequate in ensuring the effective security of a network, particularly when confronted



(a) Single IDS (b) IDS Chain

(c) Parallel IDSs (d) Cross IDSs

Figure 3: Different IDS configuration.

with high packet speeds and a variety of attack types. The limitations become evident as the single IDS struggles to accurately identify and analyze all incoming packets, raising concerns about implementing the IDS without addressing these challenges. Among the prevalent network attacks, the DoS attack stands out as a significant threat, inundating the target with an overwhelming volume of packets. Consequently, our network may fail to detect simultaneous attacks executed in conjunction with a DoS attack. This vulnerability arises due to the inherent limitation of the IDS, which can only process a finite number of packets at any given time. As the network speeds escalate, the IDS encounters bottlenecks, leading to system overload and reduced efficiency. As a result of these limitations, the IDS experiences a surge in drop rate, indicating the rate at which packets are discarded, and a decline in detection rate, representing the successful identification of malicious packets. Both drop rate and detection rate serve as vital performance indicators. In pursuit of an optimal IDS configuration, our primary objective is to achieve the lowest possible drop rate and the highest possible detection rate. To address these challenges, it is crucial to explore alternative IDS configurations that can mitigate the drop rate and enhance the detection rate. By implementing innovative solutions, we aim to fortify our network's defenses against emerging threats.

Fig. 3 illustrates the chain configuration of IDSs, which has been observed to face challenges. To address the drop in detection rate, we have implemented a chain configuration of IDSs, as depicted in Fig. 3(b). In this configuration, traffic follows a sequential path through the IDSs. Initially, the traffic is analyzed by the first IDS before passing through to the second IDS for further analysis. After both IDSs have scrutinized the traffic, appropriate actions, such as issuing alerts or clearing the packets, are taken. The chain configuration provides two opportunities for identifying malicious packets, resulting in an expected increase in the detection rate. However, we anticipate that the drop rate may be comparable to or higher than in the single IDS configuration, as resource limitations could still lead to delays in analyzing all incoming traffic. Therefore, while the chain configuration excels in detection rate, it may exhibit weaknesses in drop rate performance.

To counteract the potential rise in dropping rate, we have implemented a parallel configuration of IDSs, as shown in Fig. 3(c). This setup divides the initial flow of traffic into two sub-flows, referred to as $f1$ and $f2$. Each sub-flow is directed to a corresponding IDS: $f1$ is analyzed by IDS A, while $f2$ is processed by IDS B. Both sub-flows are individually analyzed within their respective IDSs before

Table 1: Comparison of different configurations under different speeds of traffic. The packet size is 1024B.

Speed (pps)	Attack Traffic	Detection Rate(%)					Dropping Rate(%)				
		Single	Chain(AB)	Chain(BA)	Parallel	Cross	Single	Chain(AB)	Chain(BA)	Parallel	Cross
50k	UDP	98.1	100	100	98.57	100	0.5	0.5	0.35	0	0
	UDP+TCP	97.84	100	100	98.14	100	1.13	1.13	0.98	0	0
	UDP+TCP+ICMP	96.67	98.12	100	97.17	100	1.6	1.6	1.13	0	0
150k	UDP	97.4	100	100	97.87	100	3.10	3.11	2.05	0	0
	UDP+TCP	96.20	98.71	98.71	96.80	98.71	5.60	5.61	5.0	0.12	0.12
	UDP+TCP+ICMP	93.23	95.6	95.6	94.03	95.6	9.87	9.87	9.10	0.45	0.45
300k	UDP	93.82	96.60	96.60	94.12	96.60	8.4	8.4	7.14	0.48	0.48
	UDP+TCP	90.55	94.78	94.78	91.40	94.78	12.12	12.12	11.72	1.13	1.13
	UDP+TCP+ICMP	87.32	90.41	90.41	87.82	90.41	15.3	15.3	14.12	2.56	2.56

actions are taken, such as issuing alerts or clearing packets. By adopting the parallel configuration, we benefit from reduced traffic load on each IDS. This reduction in workload enables the IDSs to handle their tasks more efficiently, leading to a decrease in the dropping rate. Nonetheless, we anticipate that the detection rate in the parallel configuration may be comparable to or slightly higher than that of the single IDS, as the parallel configuration lacks the redundancy of the chain configuration. Nevertheless, the parallel configuration offers the advantage of easing the traffic burden on each IDS, thereby enhancing overall performance.

How can we attain the benefits associated with the "double filter" and reduced traffic load, as observed in the chain and parallel IDS configurations? We propose the cross configuration, illustrated in Fig. 3(d). In this configuration, the initial traffic flow is partitioned into two sub-flows, with each sub-flow undergoing analysis by its respective IDS, similar to the parallel configuration. This design is expected to yield the advantage of a reduced traffic rate, akin to the benefits observed in the parallel IDS configuration. Consequently, we anticipate achieving a low drop rate that is comparable to that of the parallel IDS configuration. Following the analysis by their respective initial IDS, the sub-flows are then routed to the opposite IDS for further examination. This endeavor to implement the double filter emulates the behavior observed in the chain IDS configuration. We foresee that this approach will offer the advantage of a higher detection rate, akin to what is experienced in the chain configuration. By amalgamating the key features contributing to the advantages of both the chain and parallel configurations, we have devised the cross configuration.

In summary, our investigation of the cross configuration leads us to anticipate favorable results. We expect the cross configuration to exhibit a drop rate that is on par with the performance of the parallel IDS configuration and a detection rate that is comparable to that of the chain IDS configuration. These expectations stem from the unique combination of features inherent in the cross configuration, which draw inspiration from the strengths of both the chain and parallel setups. To verify and validate these assumptions, we have undertaken a rigorous testing process under a diverse range of network conditions. Our comprehensive testing strategy involves varying parameters such as traffic speed, attack types, and packet sizes. By subjecting the cross configuration to various scenarios, we aim to capture a holistic view of its performance capabilities.

The evaluation part includes comparative analyses with the single IDS, chain IDS, and parallel IDS configurations. This comparative approach provides us with valuable insights into the strengths and weaknesses of each configuration and facilitates an informed decision-making process. Ultimately, our goal is to identify the optimal IDS configuration that strikes a balance between drop rate and detection rate, fortifying our network's resilience against sophisticated threats. Armed with empirical evidence from our extensive testing, we will be better equipped to make data-driven choices in configuring our IDS infrastructure, bolstering our network's security posture.

5 EVALUATION

The malicious traffic was generated using Kali Linux Metasploit framework. Four different total configurations are tested under different network conditions. Network conditions will vary in the rates of packets sent as well as the size of the packet bodies. The scenarios are as follows: *Scenario 1:* Different speed traffic, *Scenario 2:* Different sizes of packets, *Scenario 3:* Different proportions of traffic. In order to test the effectiveness of the IDSs there are some measurements that are of particular interest to us. Drop rate tells us how many packets do not make it to the target to be analyzed by the IDS. If the drop rate is high, performance is considered poor as traffic is not able to make it to the intended destination. Drop rate can be calculated with the following equation:$(P_{Receive} - P_{Analyze})/P_{Receive}$, where $P_{Receive}$ denotes the number of received packets at IDS and $P_{Analyze}$ denotes the number of packets that are analyzed by IDS. The number of packets that IDS does not process and drops are $(P_{Receive} - P_{Analyze})$.

The detection rate tells us how well an IDS is able to correctly identify malicious traffic and alert and log the event. This can be calculated by: P_{Detect}/P_{Attack}, where P_{Detect} denotes the number of detected attack packets at IDS and P_{Attack} denotes the number of attack packets that are received by IDS. Using these metrics, we are able to measure the performance of IDSs by seeing how often they correctly identify packets and how efficiently the IDSs handle different traffic loads. Using these measurements, we are able to determine the efficiency of each IDS by seeing how often the IDSs correctly identify traffic as either malicious or normal. This will allow us to see how each IDS performs under different traffic loads and attacks and determine which configuration of IDSs may be the most promising.

Table 2: Detection and dropping rate of IDS configurations with changing packet sizes. Packets send at a rate of 50k pps.

Packet (byte)	Attack Traffic	Detection Rate(%)					Dropping Rate(%)				
		Single	Chain(AB)	Chain(BA)	Parallel	Cross	Single	Chain(AB)	Chain(BA)	Parallel	Cross
512	UDP	98.78	100	100	99.18	100	0.1	0.1	0.06	0	0
	UDP+TCP	98.35	100	100	98.73	100	0.21	0.21	0.17	0	0
	UDP+TCP+ICMP	97.76	100	100	98.16	100	0.38	0.38	0.3	0	0
1024	UDP	98.10	100	100	98.75	100	0.5	0.5	0.34	0	0
	UDP+TCP	97.87	100	100	98.12	100	1.13	1.13	0.95	0	0
	UDP+TCP+ICMP	96.67	98.12	98.12	96.90	98.12	1.6	1.6	1.1	0	0
2048	UDP	96.89	98.5	98.5	97.25	98.5	1.5	1.5	0.93	0.61	0.61
	UDP+TCP	96.34	97.71	97.71	96.70	97.71	2.14	2.14	1.85	1.42	1.42
	UDP+TCP+ICMP	95.23	96.34	96.34	95.67	96.34	4.34	4.34	3.95	3.10	3.10

5.1 Single IDS

5.1.1 Impact of Traffic Speed and Packet Size. Packets were transmitted at rates of 50k packets per second, 150k packets per second, and 300k packets per second, and the corresponding outcomes were recorded in Table 1. Focusing specifically on UDP attacks, the detection rate exhibited an initial value of 98.1%, which subsequently declined to 93.82%. Simultaneously, the dropping rate experienced an increase from 0.5% to 8.4% with escalating traffic rates. This consistent pattern of declining detection rate and escalating dropping rate remains evident throughout the entire table for a single IDS configuration. All packets were transmitted at a constant rate of 50k packets per second. Table 2 presents the results obtained from different runs when packet sizes were set to 512 bytes, 1024 bytes, and 2048 bytes, with only UDP attacks present. The single IDS initially achieved a detection rate of 98.78%, which subsequently decreased to 96.89%. Correspondingly, the drop rate started at 0.1% and increased to 1.5% as the packet sizes increased. This consistent trend persisted across most data points, except for cases where the rates reached 100%. As the packet size increased, we observed a slight decline in detection rates and a minor rise in dropping rates within the single IDS configuration.

5.1.2 Impact of Attack Proportion. We extended the scope of the attacks by introducing TCP and ICMP flooding attacks in addition to UDP flooding. Upon analyzing the scenarios where attack traffic and traffic rate were kept constant, a consistent trend emerged, showing a decrease in the detection rate as the number of attacks increased. Furthermore, a similar pattern was observed wherein the detection rate decreased with the inclusion of more attacks.

5.2 Chain Configuration

5.2.1 Impact of Traffic Speed and Packet Size. When varying the traffic speed, the chain configuration demonstrates superior performance in terms of detection rate. This can be attributed to the repeated analysis of packets, effectively enhancing the likelihood of alerting to a malicious packet. However, similar to other configurations, the detection rate tends to decrease with increasing traffic speed. Furthermore, the dropping rate in the chain configuration remains comparable to that of the single IDS as traffic speed is adjusted. Additionally, we conducted tests by employing the chain in both directions: IDS A followed by IDS B and vice versa. These tests yielded minor differences, which we attribute to the multithreading capability of Suricata, unlike Snort, which lacks multithreading support. Detection rate was higher compared to a single IDS, and a drop in performance occurs under much larger packets compared to a single IDS. Drop rate was again comparable to a single IDS configuration.

5.2.2 Impact of Attack Proportion. Depending on packet size and traffic rate, we can see that there are some conditions where detection rate was maintained even under changing attack traffic. Drop rate for chain configuration was comparable to drop rate in a single IDS configuration except for very minor differences when reversing the order of the chain.

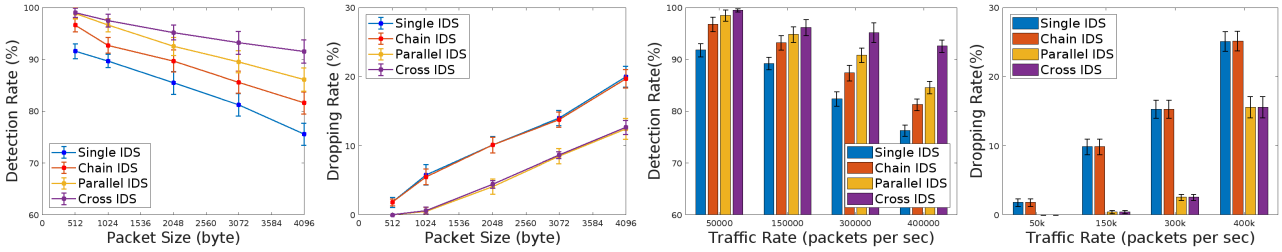
5.3 Parallel Configuration

5.3.1 Impact of Traffic Speed and Packet Size. The results in Table 1 show that the parallel configuration consistently has a comparable detection rate to a single IDS. However, the parallel configuration demonstrated that it posses a lower dropping rate when compared to both the single and chain IDS configurations. At the lower traffic rate, the parallel configuration actually experiences 0% packet loss meaning that all traffic was analyzed. This was not possible with the chain and single IDS configurations as the speed of traffic was too high for the IDSs to deal with. Parallel configuration performed ever so slightly better than the single IDS configuration when it came to detection rate, and slightly worse when compared to chain configuration. However, when it came to drop rate, parallel performed the best. The packets started dropping for parallel configuration only when the packet size hit 2048 bytes.

5.3.2 Impact of Attack Proportion. Increasing the attack proportions caused a decrease in detection rate and an increase in drop rate. However, whenever the dropping rate would increase, it would not increase by a large amount.

5.4 Cross Configuration

5.4.1 Impact of Traffic Speed and Packet Size. Traffic speed affected the detection rate of the cross configuration similarly to how the chain configurations detection rate falls as traffic rate is increased. The dropping rate rose similarly to the dropping rate of parallel configuration when traffic rate was increased. When traffic speed is changed, benefits of the chain configuration and parallel configuration seem to be retained in the cross configuration. The cross



(a) Detection rate (speed: 30K pps). (b) Dropping rate (speed: 30K pps). (c) Detection rate (packet:4096B). (d) Dropping rate (packet:4096B).

Figure 4: Evaluation of different configurations under varying speeds and packet sizes.

configuration had similar high detection rates as the chain configuration and low dropping rates as the parallel configuration. As packet size increased, we saw the same trend of lowered detection rate and higher drop rate. We see the benefits of the chain configuration and parallel configuration present in the cross configuration even when adjusting packet size.

5.4.2 Impact of Attack Proportion. When changing the proportion of attack, the cross configuration had comparable detection rates to the chain configuration and comparable drop rates to the parallel configurations. Even when changing the attack we are able to retain the benefits of the cross and parallel configuration.

5.5 Impact of Configuration

Fig. 4 shows a graphical representation of dropping rate and detection rate under varying traffic rates and packet sizes. The chain configuration performed a higher detection rate compared to the single IDS. The parallel configuration has a lower drop rate compared to single and chain. We attributed this to the fact that we are lightening traffic loads for each IDS, making it easier for the IDS to analyze all traffic it sees. The cross configuration has a comparable detection rate to the chain configuration. With the largest packet size and speeds, the cross configuration had the highest detection rate while having dropping rates comparable to the parallel configuration. This suggests that we were able to successfully make a configuration with the benefits of other configurations.

6 CONCLUSION

From the data collected, the cross configuration seems to be the best-performing configuration in terms of detection rate and dropping rate. The cross configuration takes the benefit of the high detection rate of chain configuration and the low drop rate of the parallel configuration. The benefit of the high detection rate comes from each flow of traffic being analyzed by two IDSs, increasing the opportunity malicious packets are detected. The benefit of the low drop rate comes from the overall rate of traffic to each IDS being reduced from splitting the initial traffic flow evenly. Due to time constraints, this experiment did not take into consideration the rate of false positives and false negatives. We also did not take into consideration the delay and time it takes for all of the packets to be analyzed. These are also extremely important metrics when it comes to measuring performance overall. In the future, we plan to address the issues with delay, false positives, and false negatives.

7 ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation Grant CNS 2150152.

REFERENCES

[1] Ahmad Ajiya Ahmad, Souley Boukari, Abdullahi Musa Bello, and Mustapha Aliyu Muhammad. 2021. A survey of intrusion detection techniques on software-defined networking. *Intl. Journal of Innovative Science and Research Technology* (2021).

[2] Mitko Bogdanoski, Tomislav Suminoski, and Aleksandar Risteski. 2013. Analysis of the SYN flood DoS attack. *Intl. Journal of Computer Network and Information Security (IJCNIS)* 5, 8 (2013), 1–11.

[3] Zhang Chao-Yang. 2011. DOS attack analysis and study of new measures to prevent. In *Proc. of the IEEE Intl. Conf. on Intelligence Science and Information Engineering*. 426–429.

[4] Jacob H. Cox, Joaquin Chung, Sean Donovan, Jared Ivey, Russell J. Clark, George Riley, and Henry L. Owen. 2017. Advancing Software-Defined Networks: A Survey. *IEEE Access* 5 (2017), 25487–25526.

[5] Alka Gupta and Lalit Sen Sharma. 2020. Performance Analysis and Comparison of Snort on Various Platforms. *Intl. Journal of Computer Information Systems & Industrial Management Applications* 12 (2020).

[6] Hannes Holm. 2014. Signature Based Intrusion Detection for Zero-Day Attacks: (Not) A Closed Chapter?. In *2014 47th Hawaii International Conference on System Sciences*. 4895–4904. https://doi.org/10.1109/HICSS.2014.600

[7] Samad S. Kolahi, Kiattikul Treseangrat, and Bahman Sarrafpour. 2015. Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13. In *Proc. of the Intl. Conf. on Communications, Signal Processing, and their Applications (ICCSPA)*. 1–5. https://doi.org/10.1109/ICCSPA.2015.7081286

[8] Jishuai Li, Tengfei Tu, Yongsheng Li, Sujuan Qin, Yijie Shi, and Qiaoyan Wen. 2022. DoSGuard: Mitigating denial-of-service attacks in software-defined networks. *Sensors* 22, 3 (2022), 1061.

[9] Avadhani Naidu. 2013. An Effective Evolution of Packet Loss with Snort. *IJCST* 4 (2013), 43–46.

[10] Nadia Niknami, Emily Inkrott, and Jie Wu. 2022. Towards Analysis of the Performance of IDSs in Software-Defined Networks. In *Proc. of the IEEE 19th Intl. Conf. on Mobile Ad Hoc and Smart Systems (MASS)*. 787–793.

[11] Nadia Niknami and Jie Wu. 2022. Enhancing Load Balancing by Intrusion Detection System Chain on SDN Data Plane. In *Proc. of the IEEE Conference on Communications and Network Security (CNS)*. 264–272.

[12] Nadia Niknami and Jie Wu. 2022. Entropy-KL-ML: Enhancing the Entropy-KL-Based Anomaly Detection on Software-Defined Networks. *IEEE Transactions on Network Science and Engineering* 9, 6 (2022), 4458–4467.

[13] Longwen Shuai and Suo Li. 2021. Performance optimization of Snort based on DPDK and Hyperscan. *Procedia Computer Science* 183 (2021), 837–843.

[14] Jha Singh. 2017. A Survey on Software Defined Networking: Architecture for Next Generation Network. *Journal of Network and Systems Management* 25 (2017).

[15] Mehr u Nisa and Kashif Kifayat. 2020. Detection of slow port scanning attacks. In *Proc. of the IEEE Intl. Conf. on Cyber Warfare and Security (ICCWS)*. 1–7.

[16] Tao Wang and Hongchang Chen. 2017. SGuard: A lightweight SDN safe-guard architecture for DoS attacks. *China Communications* 14, 6 (2017), 113–125. https://doi.org/10.1109/CC.2017.7961368

[17] Jiao Zhang, Zenan Wang, Ningning Ma, Tao Huang, and Yunjie Liu. 2018. Enabling Efficient Service Function Chaining by Integrating NFV and SDN: Architecture, Challenges and Opportunities. *IEEE Network* 32, 6 (2018), 152–159. https://doi.org/10.1109/MNET.2018.1700467