A Comparative Analysis of Snort 3 and Suricata

Akram Abd Eldjalil Boukebous *, Mohamed Islem Fettache *, Gueltoum Bendiab*, Stavros Shiaeles†
* Department of Electronics, University of Frères Mentouri, Constantine 25000, Algeria
boukebbousakramabdeldjalil@gmail.com, islemmohamed423@gmail.com, kelthoum.bendiab@umc.edu.dz,
†Centre for Cybercrime and Economic Crime, University of Portsmouth, PO1 2UP, Portsmouth, UK
sshiaeles@ieee.org

***Abstract*—The threat of intrusion has become a reality in modern network infrastructures, especially with the increased usage of IoT devices, cloud computing and wireless telecommunications. In this context, Network intrusion detection systems (NIDS) are becoming strategic security solutions, offering thorough defence against potential threats to the integrity, confidentiality, and availability of the data on a network. Many NIDS systems have been proposed in the literature, but Snort and Suricata are the most known in the open-source market. This paper compares the performance of the two NIDS, especially with the release of Snort 3 which is considered as the next generation of the Snort NIDS by integrating new ideas such as multithreading, expanded bindings and better cross-platform support. The quantitative study is done in a virtualised network environment in order to measure the performance of each NIDS in terms of accuracy, memory and processor usage, packet processing rate and the number of packet losses of each NIDSs. From this study, we have concluded that Snort 3 has better performance than Snort 2 and both Snort 3 and Suricata perform well but are not perfect and have some limitations that should be tackled.**

***Index Terms*—NIDS, Multithreaded, Network Security, Snort, Suricata, Signature-based detection.**

I. Introduction

Due to the growing complexity of cyber attacks, cybersecurity is becoming a rising concern for businesses of all kinds [1], in particular with the increased use of IoT devices, which are highly vulnerable to cyberattacks [2]. As highlighted by RiskIQ research [3], these malicious activities cost organisations more than $2.9 million every minute, and major businesses lose $25 per minute as a result of sensitive data violations. Another study by RiskBased Security research [4] revealed that in the first three quarters of 2020, cyberattacks led to the exposure of 36 billion records. Organizations must therefore protect their networks, systems, and users against a large number of serious cybersecurity threats. [2].

Network Intrusion Detection Systems (NIDS) are perfect tools to provide in-depth security against these threats [2]. These tools inspect the network traffic to identify and report malicious network activity that could pose a possible threat to the information confidentiality, integrity, or availability [5] (see Fig. 1). Over the last decade, numerous open-source NIDSs have been proposed in the market such as Snort, Suricata, Zeek, Suricata, OpenWIPS-ng, Prelud Hybrid IDS, Sguil and Security Onion. Most of these tools are based on a signature-based method to detect potential threats. This approach utilises a set of rules (or signatures of known intrusions) to find and report network traffic that matches against them. Each rule defines the network activity that is considered malicious and specifies the action that should be taken in case of suspect network traffic, such as generating an alert or dropping the packet [6]. Although this approach has a high detection accuracy and a low rate of false alarms, it cannot identify unknown or zero-day threats [7].

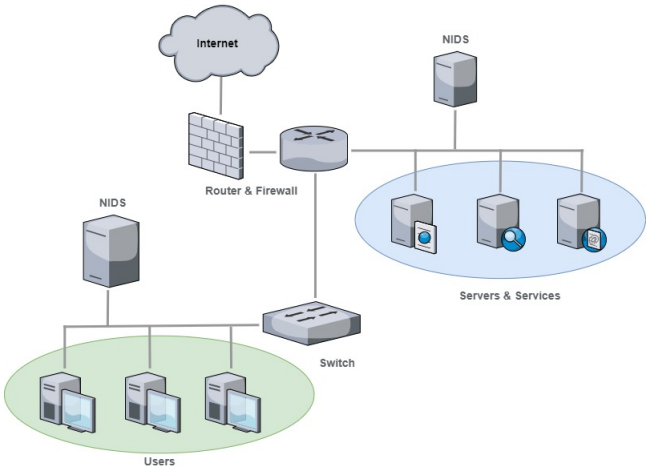

Fig. 1. Network protected by NIDS

The two NIDSs that have been used the most frequently over the past few decades are Snort and Suricata. Martin Roesch from Sourcefire 4 created Snort for the first time in 1998 and it is now maintained by Cisco after Sourcefire was purchased by Cisco in 2013. Suricata, a more recent NIDS than Snort, was developed by the Open Information Security Foundation (OISF) in 2010 in an effort to satisfy the needs of contemporary infrastructures [8]. However, the majority of comparison studies show that Suricata, which uses a multi-threaded processing architecture, outperforms Snort, which only uses a single-threaded architecture and hence does not efficiently utilise multiple CPUs (core) at a time [1], [2], [8]–[11]. This makes Suricata more suitable than Snort for the increasing networking needs of organisations as it can manage greater network traffic in comparison with Snort. Additionally, using several Snort instances to achieve what Suricata can with its multi-threaded design raises the cost to run and maintain a Snort environment [10]. In this context, the president of OISF confirmed that a multi-threaded NIDS is crucial for high performance and scalability [10]. This is the main reason why the Snort development team started rethinking the concepts

and architecture of Snort and investing in rewriting the core engine from scratch (from 2005). This work led to the release of Snort Engine 3.0, which was made available in January 2021. Because this new version was created in C++, its code base is more modular and simpler to maintain. Furthermore, it enables multiple packet processing to release additional memory for increased packet processing power [12].

The performance of the new multi-threaded Snort version is evaluated in this study in terms of a number of characteristics, including resource consumption, packet processing, packet drop rate, and throughput, and it is compared to Suricata and the old Snort version (Snort V2). This would be beneficial to both security experts and researchers by presenting novel opportunities for research into how to make this new NIDS engine function better. The rest of the paper is organized as follows. Section II gives an overview of some prior work that is similar to our work. Section 3 includes a high-level comparison of Snort III, Snort 2 and Suricata. The proposed performance evaluation methodology is explained in Section IV, followed by the findings of our assessment. Section V concludes the paper while discussing future avenues of research.

II. Related work

One of the most crucial security tools used by organisations to safeguard their networks and data from assaults is intrusion detection systems (IDSs). However, performance problems can significantly affect the functionality of those tools. For instance, overworked or malfunctioning hardware may lead to dropping packets and permit malicious ones to enter a network [15]. These issues have made performance testing of these systems a very active area of research in recent years [1], [8], [10], [11], [15]–[17]. In this context, several open-source NIDS solutions, in particular, Snort (i.e., the single-threaded variant) and Suricata, have been tested in many prior works based on their performance and alert behaviour.

In this regard, the effectiveness of the single-threaded Snort NIDS has been thoroughly studied in both simulation and real networks [13], [18]. In [19], Suricata and Snort (single-threaded version) were both tested on several platforms including Linux 2.6, ESXi virtual server and FreeBSD. This study found that Suricata performed well on Linux 2.6 and outperformed FreeBSD, Virtual Linux, and Snort on FreeBSD, especially when managing high speeds. In [13], a series of tests were performed on the Snort NIDS based on a combination of different parameters including the number of packets, packet sizes, and traffic rate. These experiments revealed that Snort's packet drop rate increased and its number of packets inspected decreased when exposed to heavy and fast traffic. The authors obtained similar results with large-sized packets. Study in [20] achieved the same conclusions as in [13]. Performance comparisons between the two NIDSs Snort and Suricata were conducted in a number of different research. for instance, in [9], authors evaluated the scalability and reliability of Suricata and the single-threaded variant of Snort by using high traffic. The evaluation parameters used are; CPU and memory usage, packet drop, and packed analysis rate. The study's findings showed that Suricata is scalable to handle increased network traffic and outperformed Snort in all tests. The authors came to the conclusion that Suricata appears as the best option for contemporary large-scale, high-performance networks. In a similar study [15], authors investigated how well the two open-source NIDSs performed in identifying different types of malicious traffic in heavy traffic (10 Gbps network speed). This study showed that Snort has lesser CPU and RAM usage compared to Suricata for different traffic speeds. The study also indicated that Snort is more accurate than Suricata, but it has higher false-positive alert rates, which may undermine its credibility because the administrator must spend time and resources investigating the false-positive alerts. Snort also dropped more packets at a higher speed and has a lower packet processing rate.

With a similar intention, authors in [14] compared Snort, Suricata, and Zeek based on a number of parameters including CPU and RAM utilisation packet drop rate and the speed of packet processing. More specifically, this paper analysed the multi-threaded variant of Snort, but the solution was in the beta phase, which is a non-stable version. Results from this study indicate that Suricata performs better than Snort and Bro IDs in high-speed networks. In another recent work [11], both versions of Snort (single-threaded and multi-threaded), Suricata, and Bro-IDS(Zeek) have been tested in a virtualised environment based on various parameters including detection engine algorithms, packet capturing, packet size, and ruleset size. According to the study's findings, Suricata performs better than Snort and Zeek (Bro-IDS) in all aspects. In another similar work [16], the performance of the multi-threaded variant of Snort has been tested and compared to Suricata in terms of resource utilisation and generated alerts. This study found that both NIDS use resources similarly, however when looking at malicious traffic, Suricata with their default rulesets identified more attacks than Snort 3.

In conclusion, the review of prior work on this topic, as shown in Table I, illustrated that Suricata performs better than Snort, especially its single-threaded variant (snort v2), under high-traffic conditions, which makes it more suitable for modern high-performance and large-scale networks. As network speeds and complexity continue to increase, Snort 3 was designed to address performance issues of the single-threaded variant of Snort and handle larger volumes of traffic in gigabit networks. The performance of this newly released NIDS (2021) has not been deeply investigated. We found only two recent studies [11], [16]. Thereby, in this work, we conduct a performance assessment of both variants of Snort (single-threaded and multi-threaded) and Suricata based on some performance metrics.

III. NIDS solutions Snort and Suricata

Snort and Suricata are free open-source Network Intrusion Detection Systems (NIDS) and Network Intrusion Protection Systems (NIPS). As mentioned before, Snort was developed by Sourcefire in 1998 but is currently maintained and developed by Cisco, while Suricata is a recent NIDS compared to

TABLE I
RECENT RESEARCH ON OPEN-SOURCE NIDS SYSTEMS COMPARISON

Authors	O-S NIDS solutions	Evaluation metrics	Results
Saber et al. [13]	SNv2	PDR, PAR, Packets captured	Snort is not efficient in high-speed networks with heavy traffic
Qinwen et al. [14]	SN++Beta, Suricata, Zeek	CPU and RAM usage, PDR, PAR	Suricata works better than Snort and Zeek
Shah et al [15]	SNv2, Suricata	Accuracy, FPR, CPU and RAM usage, PAR, PDR	Snort is more accurate and use less resources, but it has higher FPR and less TAR
Gupta et al [9]	SNv2, Suricata	CPU and RAM usage, PDR, PAR	Suricata is more scalable and uses less resources
Waleed et al [11]	SNv2, SNv3, Suricata, Zeek	PC, DE algorithms, packet size, ruleset size	Suricata outperforms other NIDSs in all aspects
Hoover et al [16]	SNv3, Suricata	CPU and RAM usage, Number of generated alerts	Very similar in terms of CPU and RAM usage, but Suricata perform better in term of generated alerts

- (SNv2) Single-threaded variant of Snort, - (SNv3): multi-threaded variant of Snort, - (FPR) False Positive Rate,
- (PDR) Packet Drop Rate, - (PAR): Packet Analysis Rate, - (PC): Packet Capturing, - (DE): Detection Engine

Snort. It was developed by the American Foundation Open Information Security Forum (OISF) in 2010. With the introduction of a multi-threaded design, the goal of this NIDS is to accelerate network traffic analysis and get beyond the computational constraints of the single-threaded architecture. Operating systems (OS) like GNU/Linux, Microsoft Windows, Mac OS, FreeBSD, and OpenBSD are all supported by the Snort and Suricata NIDSs. In addition, both NIDS systems have a graphical interface or GUI (Graphical User Interface) that simplifies the use and configuration of the NIDS for the network administrator. In particular, the SELKS version of Suricata [21], which integrates the well-known three components Elasticsearch, Logstash and Kibana, presents an excellent solution to analyse, store and visualize the events generated by Suricata in its log file. Further, Snort and Suricata also support both versions of the Internet protocol (IPv4 and IPv6) that most TCP/IP networks use.

Snort works by default in NIDS mode which uses a signature-based detection method. For that, it provides a set of predefined rules that are freely available on its website (https://www.snort.org/). It can also operate in NIPS mode (Snort Inline version) and block certain types of cyberattacks. Actually, the Snort Inline project is a modified version of Snort NIDS that is capable of blocking network intrusions/attacks. This version is available on the website of Snort (http://snort-inline.sourceforge.net). Similarly, Suricata can operate in either NIDS or NIPS mode, but Suricata came from the beginning as a NIDS/NIPS. Like Snort, Suricata is a rules-based NIDS that is compatible with the open rulesets of Snort and Emerging Threats (ET). Additionally, Suricata incorporates the Lua scripting language (https://www.lua.org/) which provides greater flexibility to create rules capable of identifying advanced and complex threats that would be difficult or impossible to identify with a legacy Snort rule. Another similarity between Snort and Suricata is that both systems are capable of performing real-time or offline network traffic analysis. Snort can perform offline analysis of multiple files at the same time, while Suricata can only analyse one file every single time. Unlike Snort, Suricata provides Deep Packet Inspection (DPI) and automatic detection of protocols which makes it more efficient for cyber-threat detection. DPI is an advanced technique for examining and managing network traffic that allows Suricata to locate, identify, classify, or block packets containing specific data or code (i.e., malware). In addition, the automatic detection of protocols allows Suricata to detect communications that can be made on non-standard ports. Actually, Suricata performs automatic detection of the following application layer protocols: DCE/RPC, DNS3, DNS, HTTP, IMAP, FTP, MODBUS, SMB, SMB2, SHH and TLS (SSLv2, SSLv3, TLSv1, TLSv1.1 and TLSv1.2) [22]. However, Snort can only scan three TCP/IP protocols for suspicious behaviour: TCP, UDP, and ICMP. Another important difference between Snort and Suricata concerns the volume of source code for each NIDS. According to the study carried out by the founding company of Suricata (OISF) [23], Suricata managed in two years to reach the same volume as the code of Snort 2, which is old compared to Suricata. Not only that but also, it is becoming the bulkiest today.

TABLE II
COMPARISON BETWEEN SURICATA, SNORT 2 AND SNORT3

Criteria	Suricata	Snort 2	Snort 3
Provider	OISF	Cisco System	Cisco System
O-S license	GNU GPL	GNU GPL	GNU GPL
Portability	Win/Unix/Mac	Win/Unix/Mac	Win/Unix/Mac
GUI configuration	Yes	Yes	Yes
Network traffic	IPv4/IPv6	IPv4/IPv6	IPv4/IPv6
Operating mode	NIDS, NIPS	NIDS, NIPS	NIDS, NIPS
M-T processing	Yes	No (S-T)	Yes
DPI	Yes	No	Yes
ADP	Yes	No	No
Offline Analysis	Yes (for SN)	Yes (for MF)	Yes (for MF)
Installation and D	Medium	Easy	Easy
SC Volume	Evolves rapidly	Evolves slowly	New code

- (O-S) Open source, (SN) Single File, (MN) Multiple File
- (D) Deployment, (ADP) Automatic Detection of Protocols,
- (M-T) Multi-Threaded, (S-T) Single-Threaded
- (DPI) Deep Packet Inspection, (SC) Source Code,

Snort 3, officially launched on January 1, 2021, is the next generation of the Snort NIDS, specially designed to make Snort more powerful. This required a complete source code rewrite in C++ (Snort++) [12] to create a new NIDS that brings many new and improved features. Indeed, this new version is the result of more than seven years of development within the framework of a project called "SnortSP" (Snort Security Platform) started by Roesch in 2005. Multithreaded

architecture, new rules syntax, new rules to detect zero-day attacks and automatic detection of services are some of the main improvements brought by Snort 3 [12]. Table II presents a high-level comparison between Snort 3, Snort 2 and Suricata.

IV. Experimental Evaluation Methodology

In this section, we will present an experimental study of the NIDSs Snort 2, Snort 3 and Suricata in order to identify the performance of each system in terms of resource consumption (CPU, RAM, etc.), packet drop and loss of alerts.

A. Experimental set-up

The environment in which we performed the configuration and deployment of the three NIDSs Snort 2, Snort 3 and Suricata (6.0.10 Release) is based on a physical machine equipped with an Intel (R) Core (TM) i7 9750 H processor, CPU @ 2.60 GHz, 2.59 GHz, 16 GB RAM and 1 TB hard drive. The comparative study was carried out in a simulation environment which is based on the virtualization software VMware 15. As illustrated in Fig. 2, the simulation environment is composed of a certain number of VMs, the NIDSs (Snort2, Snort3, Suricata) and the external network, which is usually the source of attacks against the internal network. Finally, we made all the necessary configurations to perform the tests. Each VM has 4 cores for CPU, 2 GB of RAM and 20 GB of HD. The operating systems used are Ubuntu 20.04 for the 3 NIDs (Snort 2, Snort 3 and Suricata) VMs, FreeBSD for the pfsence VM, and Ubuntu 16.04 and Android-x86 for the other VMs. Fig. 2 summarizes the configuration adopted for the test bed. In the experiments, we have used 16 pcap files of malicious traffic (i.e., traffic corresponding to attacks), which have been collected from public sources of malicious network traffic, including the renowned malware analysis repository "Malware Traffic Analysis [24] and the "NETRESEC" Malicious Traffic Repository [25]. The pcap files range in size from 10 MB to 5 GB and contain various harmful traffic types that were produced by various forms of attacks, including malware, botnets, DDoS, ransomware, backdoors, etc.

B. Results and discussion

As mentioned before, the aim of these experiments is to study the performance of the three systems Snort 2 (version 2.9.18), Snort 3 (version 3.1.8.0) and Suricata (version 6.0.10) based on the CPU and memory usage, packet drop and alerts loss. The experiments were conducted for different types of attacks with different network traffic throughputs. Malicious traffic is sent to the internal network from the external network with the TCPreplay command. In each experiment, the NIDS is used in test mode. In this mode, the NIDS applies the ruleset on the captured packets. If a packet matches a rule, then the packet is logged, and an alert is triggered and displayed on the screen. After each test, we restore the snapshot of the initial state of the NIDS VM to avoid any abnormal behaviour after the execution of the attack. We have used the Pulledpork tool to download 18737 Snort rules in the directory (/etc/Snort/rules) of Snort 2 and Snort 3 (/usr/local/etc/rules/).

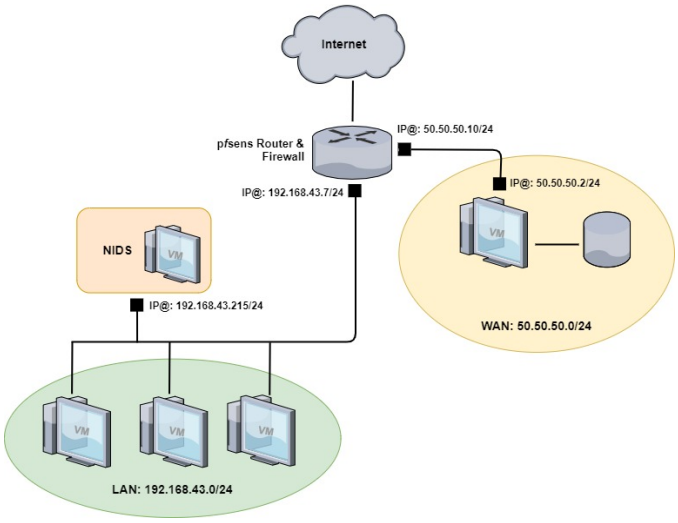

Fig. 2. High-level architecture of the test-bed

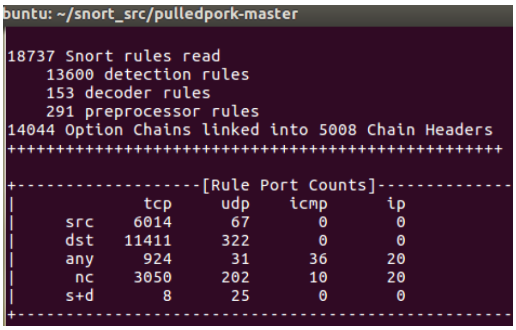

Fig. 3. Number of the rules in the ruleset

1) CPU and memory usage: In this first group of experiments, we evaluate the performance of the three NIDSs in terms of CPU and memory utilisation, i.e. the rate of processing and memory resources that are used by the NIDS process during the analysis of received packets and generation of alerts. Maximum processor load and memory usage is 100%. The most efficient system is the one that consumes fewer resources. To determine the CPU and memory consumption rates, we recorded the CPU consumption rate, taking the highest value that appears on the TOP interface during the duration of the test. Each test is repeated several times in order to obtain reliable results. The final results of this group of experiments are presented in Fig. 4 and Fig. 5.

The results of these experiments show that with increasing traffic throughput, CPU utilisation also increases linearly for all three systems (see Fig. 3). We also observed that Snort 2, which is a single-threaded application, has high CPU usage with values between 4% and 68.3%. Similarly, Snort 3, which is a multi-threaded application, has higher CPU usage with values between 10% and 90.4%, knowing that the tests are performed on a 4-core system. However, Suricata, which is also a multi-threaded application, achieved lower CPU consumption than Snort 2 and Snort 3 in the many tests (9.30%

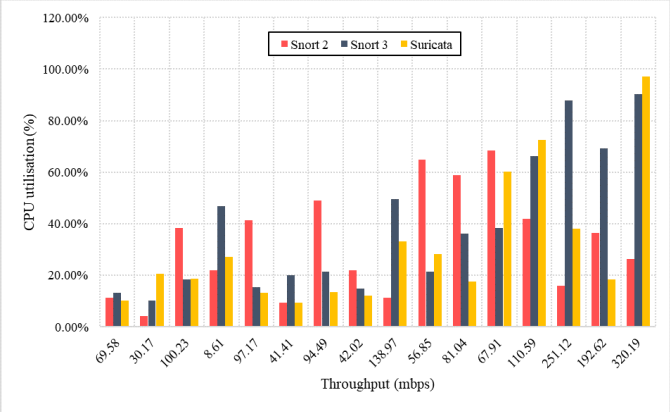

Fig. 4. CPU utilisation (%) by Snort 2, Snort 3 and Suricata

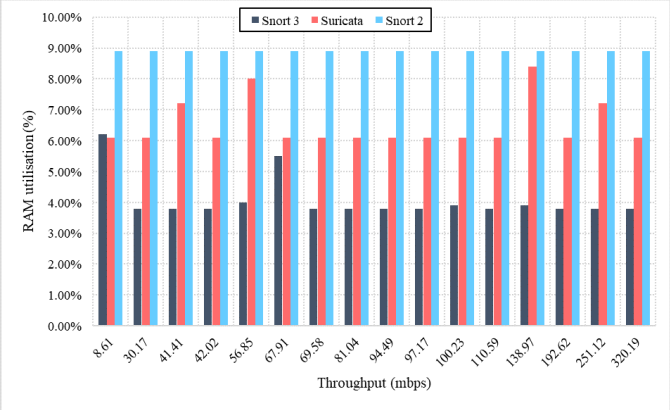

Fig. 5. RAM utilisation (%) by Snort 2, Snort 3 and Suricata

- 97.12%), but the values obtained were close to those achieved by snort 3 (see Fig. 4).

As shown in Fig. 5, the rates of memory utilisation were constant for the three systems in most tests, with simple variations, especially when the number of alerts generated by the NIDS increased. We also observed that Snort 2 has the highest memory consumption rate with an estimated rate of 8.90%. Snort 3, which is an upgrade of the Snort system, achieved lower memory usage rates (3.80% - 6.20%), with a very significant decrease (5%) compared to Snort 2. These results are better than those of Suricata which recorded values ranging from 6.10% to 8.40% as the maximum value.

2) Packets drop and Alerts loss: In this group of experiments, we evaluate the performance of the three systems in terms of packet loss (or Packet Drop), i.e., the percentage of packets lost and not analysed by the NIDS. This problem usually appears when the speed and volume of traffic increase. In this case, the NIDS fails to perform an inspection of all packets, which increases the False Negative (FN) (i.e., alert loss) rate and allow potential malicious packets to enter the internal network without being detected. Therefore, the NIDS must be efficient enough to process network traffic when the speed and volume of traffic increase and hence minimise the packet loss rate. The results of this group of experiments are presented in Fig. 6 and Fig. 7.

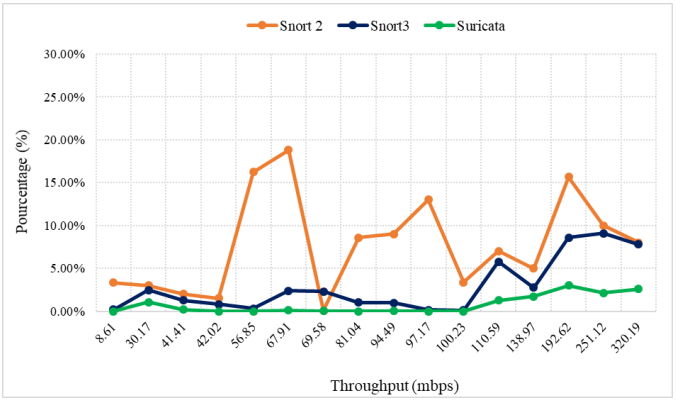

Fig. 6. Results of packet drop

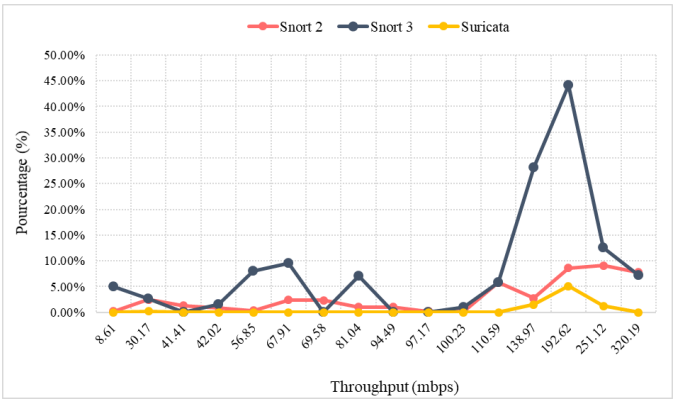

Fig. 7. Results of alert loss

As shown in Fig. 6, Snort 2 achieved different results for packet loss from one sample to another, ranging from zero 0.0% in one sample (of DDoS attack), to reach 59.66% as the maximum value recorded in these experiments. In this context, Snort3 performed better than Snort 2, we noted changes between 0.1% and 3.60%. We recorded the highest value in sample 6 (9.08%). In these experiments, Suricata recorded low rates of packet drop (0%-3%). we got the same conclusions with regard to alert loss.

From these experiments, we observed that Snort 3 performed better in terms of memory consumption. Snort 3 used an average of 4.08% memory, which is better than Suricata's memory usage (average 6.50%) and lower than Snort 2's (average 8.90%). In fact, Snort 3 has improved the memory usage of the Snort system with a reduction of approximately 5% compared to previous versions. Similarly for the CPU, Suricata recorded the best results compared to Snort 2 and Snort 3 with an average utilisation rate of 21.28%. On this particular point, Snort 3 recorded better results than Snort 2 (33%). We have also observed that Suricata is more reliable than Snort 2 and Snort 3 in terms of packet loss and false negatives (loss of alerts). The latter recorded low packet loss

rates in most tests (0.0% -3.0%) for average network speeds (50 Mbps – 300 Mbps). We have also concluded that Snort 3 has improved the performance of the Snort NIDS compared to previous versions, however, it is very new and is not yet stable, and it will definitely be more improved in the future.

We have also noticed that the type of attack has an effect on the behaviour of the NIDS, especially, DDoS attacks that generate more alerts than all other attacks. For example, The NIDSs have generated approximately 20758 alerts for 82521 analysed packets for the DDoS scenario of attack (TESTDDOSDATA.pcap), which increased the memory consumption for the three NIDSs and the packet drop rate, especially for the Snort 2 NIDS. However, in the ransomware scenario of attack (CryptoMix_28112016.pcap), the NIDSs have generated approximately 1728 alerts for 945454 analysed packets, with less memory usage and packet drop rate compared to the previous scenario of attack.

V. Conclusion

This study compared the performance of the new releases of the Snort NIDS, Snort 3, with the previous one (Snort 2) and the Suricata NIDS in terms of resource consumption, packet drops and alerts loss. From this comparative study, we identified that Suricata is the most performant open-source NIDS in its field, however with the arrival of Snort 3, we can say today that Suricata already has a powerful competitor who could be able to reserve a very important place in the future. Snort 3 has significantly improved the memory usage of the Snort system compared to previous versions and the speed of packet processing. further, it is worth mentioning that this is a very new version and is not yet stable, and it will definitely be more improved in the future. Finally, we would like to highlight that due to the lack of material resources, our experiments were conducted in a virtual simulation environment. Therefore, in our future work, we plan to use a real network environment to conduct the experiments, with larger pcap files and higher network speeds. We, also intend to use more NIDSs in the comparison like Zeek (formerly Bro-NIDS) and OpenWIPS-ng, with different rulesets.

Acknowledgment

This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 957406 and 101021936. The work reflects the authors' view and the Agency is not responsible for any use that could be made from the information it contains.

References

[1] W. Park and S. Ahn, "Performance comparison and detection analysis in snort and suricata environment." *Wireless Personal Communications*, vol. 94, no. 2, 2017.

[2] J. C. S. Sicato, S. K. Singh, S. Rathore, and J. H. Park, "A comprehensive analyses of intrusion detection system for iot environment," *Journal of Information Processing Systems*, vol. 16, no. 4, pp. 975–990, 2020.

[3] "The evil internet minute 2019," RISQIQ. [Online]. Available: https://www.riskiq.com/resources/infographic/evil-internet-minute-2019/

[4] "Cybersecurity bulletin - issue 8 - february 2021," GDS Globalcom Data Services. [Online]. Available: https://www.gds.com.lb/Cybersecurity%20bulletin%20-%20ed%208-%20February%202021.pdf

[5] M. Chauhan and M. Agarwal, "Study of various intrusion detection systems: A survey," *Smart and Sustainable Intelligent Systems*, pp. 355–372, 2021.

[6] J. S. White, T. Fitzsimmons, and J. N. Matthews, "Quantitative analysis of intrusion detection systems: Snort and suricata," in *Cyber sensing 2013*, vol. 8757. International Society for Optics and Photonics, 2013, p. 875704.

[7] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, "A survey of neural networks usage for intrusion detection systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 497–514, 2021.

[8] F. Alsakran, G. Bendiab, S. Shiaeles, and N. Kolokotronis, "Intrusion detection systems for smart home iot devices: experimental comparison study," in *International Symposium on Security in Computing and Communication*. Springer, 2019, pp. 87–98.

[9] A. Gupta and L. S. Sharma, "Performance evaluation of snort and suricata intrusion detection systems on ubuntu server," in *Proceedings of ICRIC 2019*. Springer, 2020, pp. 811–821.

[10] E. Albin and N. C. Rowe, "A realistic experimental comparison of the suricata and snort intrusion-detection systems," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, 2012, pp. 122–127.

[11] A. Waleed, A. F. Jamali, and A. Masood, "Which open-source ids? snort, suricata or zeek," *Computer Networks*, vol. 213, p. 109116, 2022.

[12] "Why snort 3?" Snort. [Online]. Available: https://www.snort.org/snort3

[13] M. Saber, M. G. Belkasmi, S. Chadli, M. Emharraf, and I. El Farissi, "Implementation and performance evaluation of intrusion detection systems under high-speed networks," in *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, 2017, pp. 1–6.

[14] Q. Hu, M. R. Asghar, and N. Brownlee, "Evaluating network intrusion detection systems for high-speed networks," in *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2017, pp. 1–6.

[15] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to snort system," *Future Generation Computer Systems*, vol. 80, pp. 157–170, 2018.

[16] C. Hoover, "Comparative study of snort 3 and suricata intrusion detection systems," *Computer Science and Computer Engineering Undergraduate Honors Theses*, 2022.

[17] S. B. Chalmers, "Comparison of different security tools to detect risks in networks," *International Journal Of Computer Sciences and Mathematics Engineering*, vol. 1, no. 1, pp. 13–19, 2022.

[18] H. M. Elshafie, T. M. Mahmoud, and A. A. Ali, "Improving the performance of the snort intrusion detection using clonal selection," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*. IEEE, 2019, pp. 104–110.

[19] A. Alhomoud, R. Munir, J. P. Disso, I. Awan, and A. Al-Dhelaan, "Performance evaluation study of intrusion detection systems," *Procedia Computer Science*, vol. 5, pp. 173–180, 2011.

[20] I. Karim, Q.-T. Vien, T. A. Le, and G. Mapp, "A comparative experimental design and performance analysis of snort-based intrusion detection system in practical computer networks," *Computers*, vol. 6, no. 1, p. 6, 2017.

[21] "Selks — turn-key suricata-based ids/nsm and threat hunting system," Stamus Networks. [Online]. Available: https://www.stamus-networks.com/selks

[22] "Suricata observe, protect, adapt," Suricata. [Online]. Available: https://suricata.io/

[23] "Présentation de l'ids suricata." [Online]. Available: https://docplayer.fr/14045261-Presentation-de-l-i-d-p-s-suricata.html

[24] "A source for packet capture (pcap) files and malware samples." [Online]. Available: https://www.malware-traffic-analysis.net/

[25] "Network forensics and network security monitoring." [Online]. Available: https://www.netresec.com/