International Journal of Information Security
https://doi.org/10.1007/s10207-023-00794-9

REGULAR CONTRIBUTION

A perspective–retrospective analysis of diversity in signature-based open-source network intrusion detection systems

H. Asad[1] · **S. Adhikari**[2] · **Ilir Gashi**[1]

Abstract
The signature-based network intrusion detection systems (IDSs) entail relying on a pre-established signatures and IP addresses that are frequently updated to keep up with the rapidly evolving threat landscape. To effectively evaluate the efficacy of these updates, a comprehensive, long-term assessment of the IDSs' performance is required. This article presents a perspective–retrospective analysis of the Snort and Suricata IDSs using rules that were collected over a 4-year period. The study examines how these IDSs perform when monitoring malicious traffic using rules from the past, as well as how they behave when monitoring the same traffic using updated rules in the future. To accomplish this, a set of Snort Subscribed and Suricata Emerging Threats rules were collected from 2017 to 2020, and a labeled PCAP data from 2017 to 2018 was analyzed using past and future rules relative to the PCAP date. In addition to exploring the evolution of Snort and Suricata IDSs, the study also analyses the functional diversity that exists between these IDSs. By examining the evolutionary behavior of signature-based IDSs and their diverse configurations, the research provides valuable insights into how their performance can be impacted. These insights can aid security architects in combining and layering IDSs in a defence-in-depth deployment.

Keywords Security · Diversity of security tools · Evolution of diversity · Intrusion detection systems

1 Introduction

Network intrusion detection systems (IDSs) are some of the most widely used security defence tools. Some of these IDSs are available open-source, and the most widely used open-source IDSs are Snort and Suricata. Both of these tools are signature-based and rely on rules having already known signatures to identify malicious activity. The rules identify malicious activity based on content, protocols, ports etc., as well as on the origin of the activity/traffic—in this latter case, the suspicious IP addresses are "blacklisted" and traffic originating from these IPs are alerted. Depending on the configuration of the IDS, the traffic can be alerted but allowed, or alerted and dropped—the latter happens when the IDS is running in Intrusion Prevention System (IPS) mode.

Open-source signature-based IDSs rely on a constantly updated database of signatures and Blacklisted IPs (BIPs). Signatures and BIPs are added, modified or deleted. However, from the time of the new vulnerabilities being exploited (Zero-day attacks), or a new BIP is identified, to their inclusion in the rule set, there will always be some delay. The delay between the identification of new vulnerabilities or blacklisted IP addresses and their inclusion in the rule set can have a significant impact on an enterprise's security, especially when critical assets and processes require up-to-date defences. To assess the effectiveness of IDS rule updates, empirical studies often analyze labeled PCAP data from the past using current rules and BIP addresses. While this type of static analysis can measure the effectiveness of IDSs against malicious traffic of a particular time (past), it cannot measure the 'improvement/degradation' in IDS performance resulting from rule and BIP evolution. To address this, in this paper we analyze the traffic using both past and future rules (using the PCAP collection date as a reference). This enriches the analysis, as using older rules may reveal an IDS's inability to detect zero-day attacks, while analyzing the same traffic with newer rules allows estimating the number of attacks that may be missed by an IDS at any given time. Besides,

✉ H. Asad
hafizul.asad@city.ac.uk

[1] Department of Computer Science, School of Science and Technology, City, University of London, London, UK

[2] School of Engineering, Computing and Mathematics, University of Plymouth, Plymouth, UK

Published online: 20 December 2023

since the BIPs change over time, this should also be reflected in the rules as they evolve. The frequency of changes in rules and BIPs is an important consideration for security architects and can only be quantified through an analysis involving past and future rules. Additionally, recent works have shown the usefulness of using diverse IDSs in a defence-in-depth strategy, but dynamic measurement of security gains through diversity, considering the frequency at which IDSs evolve, would provide more insight. The study presented in [1] investigates the evolution of rule sets and blacklisted IP addresses in Snort and Suricata IDSs over a 5-month period. Building on this work, [2] extends the analysis to include both configurational and functional diversities between the two IDSs. While configurational diversity was found to be significant, the functional diversity analysis in [2] has some limitations. Firstly, the use of unlabeled PCAP data in the performance analysis undermines its usefulness. Secondly, the functional diversity assessment relies on identifiers based on source/destination IPs/ports and timestamps, which may not accurately reflect differences between Snort and Suricata, as they may use different source ports or detect traffic at different times. This is the reason, in [2], there is very little overlap between the alerts of Snort and that generated by Suricata, which is not a true depiction of the functional diversity between the Snort and Suricata IDSs. Furthermore, the functional diversity analysis conducted in [2] is based on a limited dataset that spans only a month, which may not provide sufficient information to make generalizations about the functional diversity analysis results.

This article presents an extension to the research conducted in [2] by analyzing the evolutionary behavior of Snort and Suricata IDSs using "labeled PCAP" data. This study improves upon the previous work by utilizing labeled PCAP data in conducting individual and cross-platform diversity analysis of Snort and Suricata IDSs. In addition, the study uses a larger rule set spanning 4 years, from 2017 to 2020, compared to [2], where the functional diversity analysis was based on a 2-week rule set. The analysis involves two experiments using two different PCAP datasets from the Canadian Institute for Cybersecurity (CIC), CIC-IDS-2017 and CSE-CIC-IDS2018 [3], collected on July 6, 2017, and between Feb 22-23, 2018, respectively. For the perspective–retrospective analysis, the PCAP datasets and rules were selected so that there are rules from both past and future with respect to the PCAP date. The PCAP datasets are analyzed using approximately 70 sets of rules to determine if the Snort and Suricata rules have been modified to improve their performance. The evolution of the IDSs is analyzed based on True Positive Rate (TPR) and True Negative Rate (TNR) using the ground truths. Furthermore, the study investigates the types of rules triggered by the PCAP data to determine the source of evolution. Similarly, recognizing the significance of diverse IDSs in a defense-in-depth topology, the study explores the diversity (and its evolution) of these IDSs by comparing their deployment in two different configurations:

- 1-Out-Of-2 (1OO2) system, where a particular flow is labeled as malicious if either Snort OR Suricata generates an alert.
- 2-Out-Of-2 (2OO2) system, where a flow is labeled as malicious if and only if both Snort AND Suricata generates an alert.

This study goes beyond comparing the sensitivity and specificity of Snort and Suricata and their evolution over time; it also investigates the root causes of this evolution, such as modifications to the rules. Furthermore, it examines the differences in their alerting behavior, including timing and the use of identifiers such as source and destination IP addresses and ports. By doing so, this analysis provides a more comprehensive understanding of the diversity that exists between Snort and Suricata. The main contribution(s) of this paper is to answer the following research questions:

- How does Snort and Suricata's detection performance change over time, and what causes these changes?
- What kind of functional diversity exists between Snort and Suricata?
- What are the effects of this diversity on the performance of 1OO2 and 2OO2 configurations of diverse signature-based IDSs, and how does this evolve over time?

The rest of the paper is organized as follows, Sect. 2 describes the dataset, tools and the experimental architecture used. Section 3 shows the individual performance and their evolution for Snort and Suricata. Section 4 discusses the diversity between the two IDSs and their evolution, followed by some discussions and limitations of the results in Sect. 5. Section 6 presents related work and finally Sect. 7 presents conclusions and further work.

2 Signature-based IDS

An IDS is a system that can potentially differentiate between malicious and benign traffic. It can be deployed on an individual host as HIDS or at a choke point in a network monitoring the network traffic as NIDS. Without loss of generality, we will refer to a NIDS as IDS in the rest of this paper. A signature-based IDS uses a database of traffic signatures, such as IP address, port numbers, protocol and payload patterns, and generates alerts if it encounters the same signatures. On the other hand, an anomaly-based IDS works by looking for anomalies in the network traffic using predictive models that are trained using normal and malicious traffic [4]. An IDS can be deployed as a passive sensor—where it can

analyze the traffic in a promiscuous mode, and as an active sensor in the In-Line mode—where it stops/allows traffic and is called an IPS.

For the detailed inside architecture of an IDS, see [2]. Signature-based IDSs use rules that usually have various fields—actions, protocols, source/destination IP, source/dest ination ports, message to be stored/displayed, regular expressions for payload etc. Rules are written targeting traffic at different Open System Interconnection (OSI) layers—network (IP), Transport (ports), and the application layer payloads. The two most famous open-source signature-based IDSs are Snort and Suricata. The Snort IDS comes with three different default rule configurations available from the Snort web pages (Community rules, Registered rules, and Subscribed rules). The difference between these rules is explained on the Snort website [5]. In summary, the website states the following for these different rules: the Subscribed (paid) rules are the ones that are available to users in real time as they are released; the Registered rules are available to registered users 30 days after the Subscribed users; the Community rules are a small subset of the subscribed/registered rulesets and are freely available to all users. The Suricata IDS uses the Emerging Threats (ET) ruleset [6]. There are rules intended for the BIPAs and while the Suricata IDS have these BIPAs embedded in the rule file, Snort has rules pointing to a directory having files with BIPAs [7]. These rules and BIPAs can be automatically updated using tools such as Pulledpork [8], and Suricata update [9]. Both Snort and Suricata offer various ways of customized logs that can be saved or sent to various logs-plugins [10, 11]. These logs have dozens of fields showing information about the IP address, ports, protocols, time stamps, session details, rules information, payload signatures, CVE information etc.

3 Description of the tools, data, and the experimental infrastructure

3.1 Description of the data used

For this experiment, we use two datasets from the CIC repository, CIC-IDS2017 and CSE-CIC-IDS2018. These are both labeled datasets generated by the CIC themselves, aimed toward the purpose of evaluating IDSs. They employ novel methods like B-Profile system to generate benign background network traffic based on abstract human behavior resulting in realistic Datasets [3]. CIC-IDS2017, which we further refer to as dataset 1 for the duration of this paper, contains network capture data starting from July 3 to July 7, 2017 for a total of 5 days, within an infrastructure of 25 machines. This PCAP data contains attacks that include Brute Force-FTP, Brute Force-SSH, DoS, Botnet, DDoS and Web Attacks. These were some of the popular attacks at the time, as argued in [12]. However, for our experiments, we use only the data collected on July 6 with Brute force attacks, Web Attacks and Infiltration Attacks. This is mainly due to our limited processing and memory resources. The resulting PCAP file is of size 8.5 GB. Similarly, CSE-CIC-IDS2018, which we further refer to as dataset 2, was generated with the collaboration between CIC and Communications Security Establishment (CSE), Canada's national cryptologic agency. This dataset, having different web-based attacks, is collected over the span of 9 days, but we only use PCAP data of 22 and 23 February, 2018. This infrastructure was hosted in Amazon Web Services which allowed it to have a larger selection and diversity of machines—around 450 machines in the victim infrastructure and 50 more machines in the attacker infrastructure, thus resulting in a PCAP file of size 34.2 GB [13]. Table 1 lists the percentage distribution of packets in these two PCAP datasets, while Table 2 shows the distribution of benign and malicious flows as discerned from the label files of the datasets.

In this paper, we use Snort's Subscribed and Suricata's Emerging Threats (ET) rules spanning 4 years, 2017 to 2020—there are 20 unique rule files per year per IDS, except for 2019 and 2020, where we were able to collect only 19 and 9 rule files for Snort and Suricata, respectively. The exact dates when the rules were collected is shown in Table 3. We also use Snort's BIPs for the same dates as the rules, whereas Suricata has BIPs within the rule files. We tried to use labeled PCAP data collected at a time such that the rules dates are uniformly distributed before and after the capture date. However, the acute lack of labeled PCAP data made this difficult.

3.2 Experimental architecture

Our experimental setup is given in Fig. 1. In this experiment, we used three virtual machines (VMs) located in a ESXi server—two VMs for Snort and Suricata, while we used the third VM for the data processing and analysis. The two VMs used for analyzing the PCAP files with Snort and Suricata have similar specifications—16 GB memory, 1 TB storage, and 4 × 2.34 GHZ AMD EPYC CPUs. The VM used for processing and analysis has 35 GB memory, 1 TB of storage, and 8 × 2.34-GHZ AMD EPYC CPUs.

While Snort was configured to log the alerts in unified2 binary format, which was then converted to json format for further processing, Suricata was configured to log its alerts in json format. We used the latest version of Snort and Suricata available at the time of the experiment, which is Snort version 2.9.7.0, and Suricata version 6.0.8.

Table 1 Statistics of the data packets, Dataset 1

Protocol	Percentage distribution Dataset 1 (%)	Percentage distribution Dataset 2 (%)
ICMP	0.015	0.025
UDP	7.504	4.198
TCP	91.608	94.348

Table 2 Number of malicious and benign flows

	Benign	Malicious
Dataset 1	456,752	2216
Dataset 2	2,096,222	2053

Table 3 Rule files collection dates

Snort				Suricata			
2017	2018	2019	2020	2017	2018	2019	2020
19-05	23-03	02-05	10-01	01-05	08-02	05-06	10-01
23-05	27-03	07-05	14-01	03-05	09-02	07-06	14-01
25-05	29-03	09-05	16-01	04-05	12-02	11-06	16-01
30-05	03-04	14-05	21-01	05-05	14-02	12-06	21-01
01-06	05-04	16-05	22-01	06-05	15-02	13-06	22-01
06-06	10-04	20-05	28-01	19-05	16-02	14-06	28-01
08-06	12-04	23-05	30-01	20-05	19-02	17-06	30-01
13-06	17-04	24-05	04-02	22-05	20-02	18-06	04-02
15-06	19-04	28-05	06-02	23-05	21-02	19-06	06-02
20-06	24-04	30-05		24-05	22-02	20-06	07-02
22-06	26-04	04-06		25-05	23-02	22-06	08-02
27-06	01-05	06-06		30-05	26-02	24-06	11-02
29-06	13-06	11-06		31-05	27-02	25-06	12-02
03-07	14-06	13-06		01-06	01-03	27-06	13-02
06-07	19-06	18-06		02-06	02-03	21-06	14-02
11-07	21-06	20-06		05-06	05-03	01-07	15-02
13-07	26-06	21-06		06-06	06-03	02-07	18-02
18-07	28-06	25-06		07-06	08-03	03-07	19-02
20-07	03-07	27-06		08-06	10-03	04-07	20-02
25-07	05-07			09-06	12-03		

4 Dynamic analysis of Snort and Suricata

A IDS classifies any network traffic it encounters into two classes, malicious or benign, making it a binary classifier. Given that we use labeled data, this classification can be divided into four classes:

- True Positive (TP): the IDS correctly labels malicious traffic
- False Positive (FP): the IDS incorrectly labels benign traffic as malicious,
- True Negative (TN): the IDS correctly labels benign traffic,

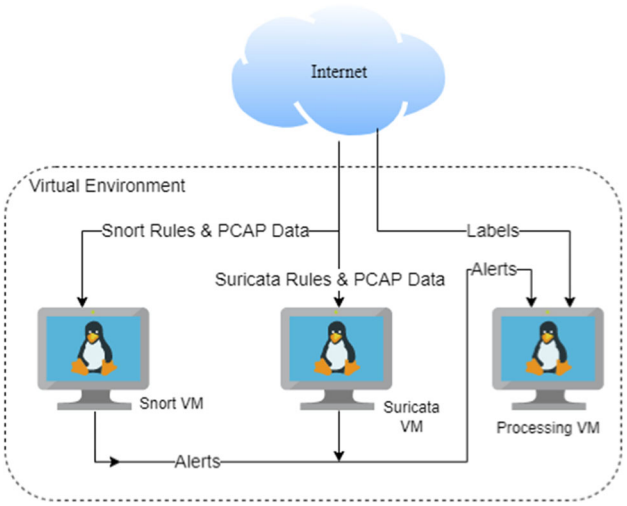

Fig. 1 Experimental setup used

- False Negative (FN): the IDS incorrectly labels malicious traffic as benign.

Many measures and performance indicators can be derived from these four counts, among them accuracy and $F1$ scores are typically used to measure the performance of binary classifiers, however we must also consider that our dataset is imbalanced: the number of malicious traffic is very low compared to the number of benign traffic as seen in Table 2, and therefore, accuracy will not be a proper measure of performance. Hence, we have to look at other well known and conventional metrics to gauge the performance of a binary classifier. These are TPR, also termed as sensitivity, and the TNR which is termed as specificity as well. Sensitivity (TPR) is calculated as,

$$\text{TPR} = \frac{\text{TP}}{(\text{TP} + \text{FN})}.$$

Similarly, Specificity (TNR) is calculated as

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}.$$

4.1 Snort

In this section, we present results of the Snort PCAP data analysis. The processing time taken by Snort to analyze the

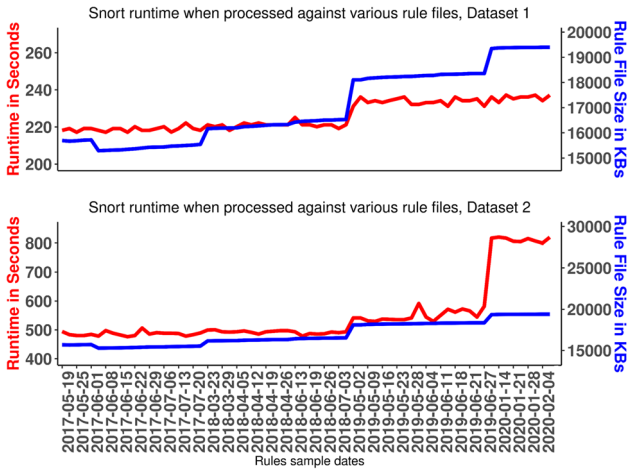

Fig. 2 Snort runtime with respect to filesize

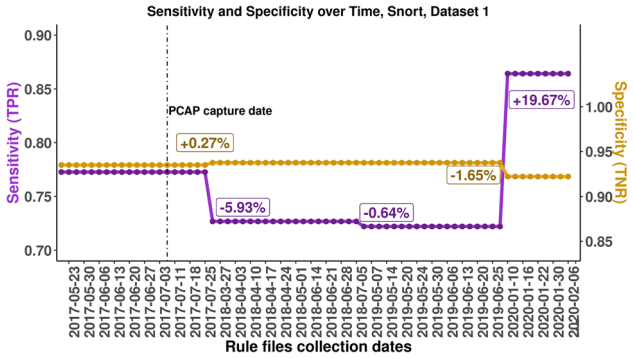

Fig. 3 Sensitivity and specificity of Snort, dataset 1

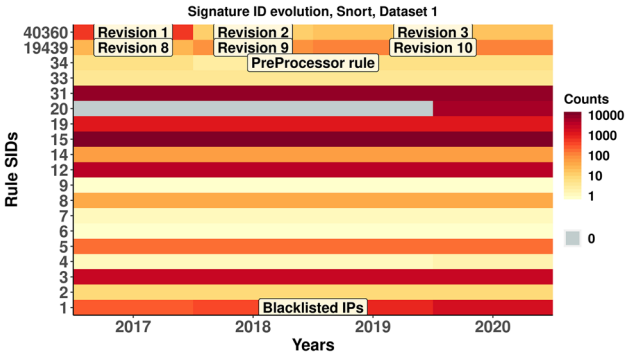

Fig. 4 Rule SID evolution of Snort, dataset 1

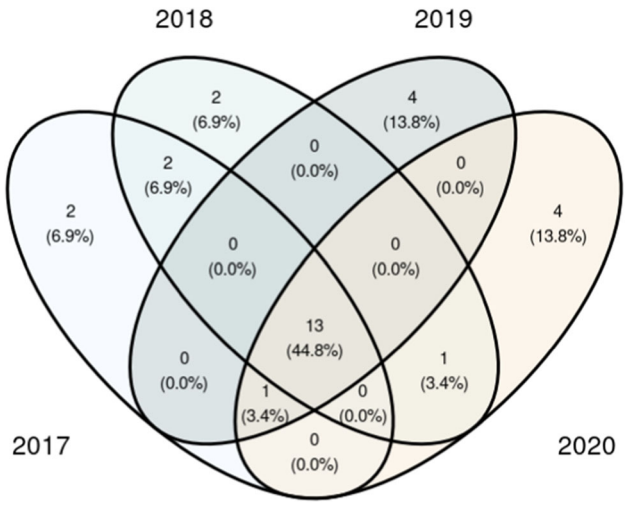

Fig. 5 Venn-diagram showing overlap between (SIDs, count) pairs of Snort, Dataset 1

two sets of PCAP datasets, using different rules, is shown in Fig. 2. Here, the red line represents the Snort processing time in seconds and the blue line represents the file size of the rules in Kilobytes (KBs). Figure 2 shows a steady increase in the time required for Snort to process the PCAP data as it uses rules from 2017 to 2020. This is mainly due to the increase in file size of rules from 2017 to 2020. While prior to 2019, the linear relationship between the processing time and file sizes is not that obvious, this is much clearer going from 2018 to 2019 and from 2019 to 2020. Since the 2020 rules are of the highest size, the increase in the processing time from 2019 to 2020 is more pre-dominant for the larger PCAP dataset 2.

4.1.1 Analysis of dataset 1

The efficacy of the Snort IDS to detect malicious/benign traffic of dataset 1 is given in Fig. 3. It shows that while the Snort sensitivity effectively remains constant during a particular year, it drops in steps going from 2017 to 2018 and from 2018 to 2019, whereas it improves substantially in 2020. On the other hand, the specificity trend in Fig. 3 is the reverse of that of sensitivity in that it increases from 2017 to 2019 and then drops off in 2020. Figure 3 also shows the percentage changes in these two parameters while going from 1 year to the next. We observe that while there is a decrease of more than 5% in the Snort sensitivity from 2017 to 2018, this decrease is less than 1% from 2018 to 2019. The sensitivity though increased by more than 19% from 2019 to 2020. The change in Snort's specificity is, however, very minimal year-on-year, where it increased by a modest 0.2% from 2017 to 2019 before decreased by 1.6% in 2020.

To investigate the underlying reasons for the changes in Snort's sensitivity and specificity, we analyze the Signature IDs (SIDs) of the rules that were triggered during the experiment. The heatmap in Fig. 4 shows the counts of these SIDs year-on-year. We observe that with an exception to SID 20, which shows only in 2020, other SIDs have been triggered for all rule sets (2017–2020). Rule SID 20 is a pre-processor rule, which was introduced in 2020 and contributed to larger number of alerts. However, though the count of most of the SIDs is constant, it is different for some SIDs year-on-year. The distribution of these (SIDs, count) pairs in various per-

Table 4 Revised Snort Rules from 2017 to 2020

Rule SID	Rule category
1	Black listed IP
12	Pre-processor rule—TCP small segments exceeds the configured threshold
20	Pre-processor rule—TCP 3-way handshake not seen for this TCP session
40360	Detected traffic exploiting vulnerabilities over the network—denial of service attempt
19439	Detected traffic associated with SQL injection—possible SQL injection attempt
42016	SCADA protocol activity—SCADA moxa discovery packet information disclosure attempt
31136	Command and control (CNC) rule violation—MALWARE-CNC Win.Trojan.Zero.Access inbound connection
23496	FILE-IDENTITY CUR file download request

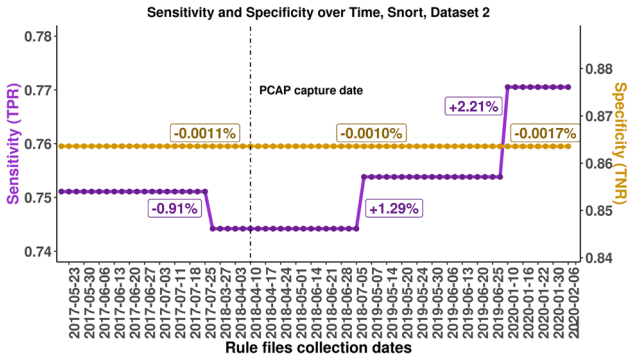

Fig. 6 Sensitivity and specificity of Snort, dataset 2

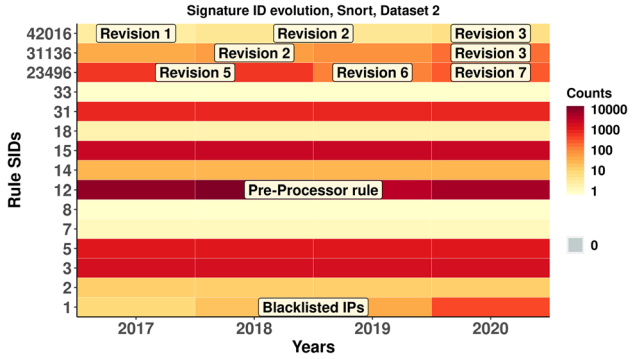

Fig. 7 Rule SID evolution of Snort, Dataset 2

mutations is shown as a Venn diagram in Fig. 5. Here, we see that 13 (44.8%) SIDs have been triggered the same number of times by all rule sets, while there are 5 SIDs, (1, 20, 34, 19439, 40360), that have a variable frequency of occurrence for different rule sets. This is the reason we see 2, 2, 4 and 4 unique (SID, count) pairs in 2017, 2018, 2019 and 2020, respectively. Note that these pairs are unique owing to their counts of occurrence and not because of the SID. Investigating these 5 SIDs, Fig. 4 shows that SID 34 is a pre-processor rule, while SID 1 is used to generate alerts for blacklisted IPs. Since the set of blacklist IPs do change year-on-year, we therefore see that these two SIDs have been triggered at a variable frequency by different rule sets. Similarly, Fig. 4 also shows that SID 40360 and 19439 are the rules that were revised multiple times from 2017 to 2020, and thus were triggered differently by various rule sets. Additionally, the rule with SID 20 was triggered in thousands and contributed to large changes in both sensitivity and specificity in 2020. We believe that these 5 rules have contributed to the changes in the sensitivity/specificity of Snort while analyzing dataset 1. The details of various SIDs is given in Table 4.

4.1.2 Analysis of dataset 2

The efficacy of the Snort IDS to detect malicious/benign traffic of dataset 2 is depicted in Fig. 6. Here, we observe the same trends as that in Fig. 3, except that the sensitivity goes up going from 2018 to 2019. The specificity though follows the same downward staircase trend year-on-year. Additionally, the percentage changes are lower than those in Fig. 3. The count of rule SIDs that were triggered by dataset 2 is shown in Fig. 7. Here again, the heatmap shows that, except 5, most of the SIDs have been triggered the same number of times by all rule sets. Figure 8 shows the distribution of (SID, count) pairs in various permutations as a Venn-diagram. We note that 38.5% of the (SID, count) pairs are common between all the years, while there are around 61% of SIDs which were triggered at varying frequencies by different rule sets. As shown in Fig. 7, we see that this change in the frequency was caused by 5 rules, with SIDs 1,12, 23496, 31136 and 42016. Here, SID 1 is a BIP rule and understandably was triggered variable number of time due to the change in the set of BIPs from 2017 to 2020. Rules, with SIDs 23496, 31136 and 42016, have been updated from 2017 to 2020 as has been given in Fig. 7 with their revision number changed. Similarly, SID 12 is a pre-processor rule which is often revised depending on the Snort engine pre-processor sub-system. We again believe that changes in these 5 rules is the main contributor of the variations we have observed in Snort's sensitivity/specificity while analyzing dataset 2.

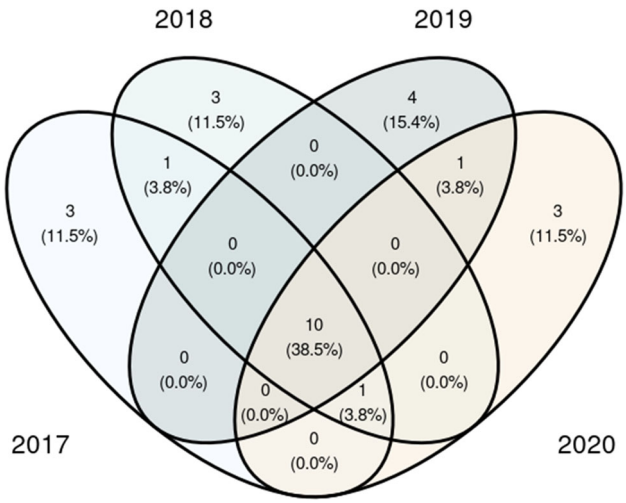

Fig. 8 Venn-diagram showing overlap between (SIDs,count) pairs of Snort, Dataset 2

Overall it can be inferred, that from 2017 to 2020, Snort's efficacy in terms of TPR/TNR degraded from 2017 to 2018, before it got improved from 2019 to 2020.

4.2 Suricata

This section describes the performance evolution of Suricata when it processes the two datasets. Figure 9 depicts the Suricata's processing time of the two datasets using rules from 2017 to 2020. We observe that Suricata does not display the same relationship between rule file sizes and processing runtime as we observed with Snort in Sect. 3.1. Notwithstanding some blips, the PCAP processing time does not seem to be affected as much, while the rule file size increases. There is though a sharp drop in the processing time of dataset 2 in 2020 before it increased to the same level as earlier. Due to the Suricata's multi-threading capability, it processed the same PCAP data 6 min faster than that by Snort.

4.2.1 Analysis of dataset 1

Next, we look at the evolution of Suricata's sensitivity and specificity when it processes dataset 1 as visualized in Fig. 10. The sensitivity trend is consistently increasing over the years, with a maximum increase of 0.69% observed when going from 2019 to 2020. In the previous years, going from 2017 to 2018 and 2018 to 2019, the increase in sensitivity is 0.50% and 0.23%, respectively. On the contrary, we observe a small, but consistent drop in its specificity year-on-year.

To investigate the changes in Suricata's sensitivity/specificity for dataset 1, we visualize the SIDs of the rules and the number of times they were triggered each year in Fig. 11. We observe that the distribution of SIDs and their corresponding counts are much sparser as compared to that of Snort. There

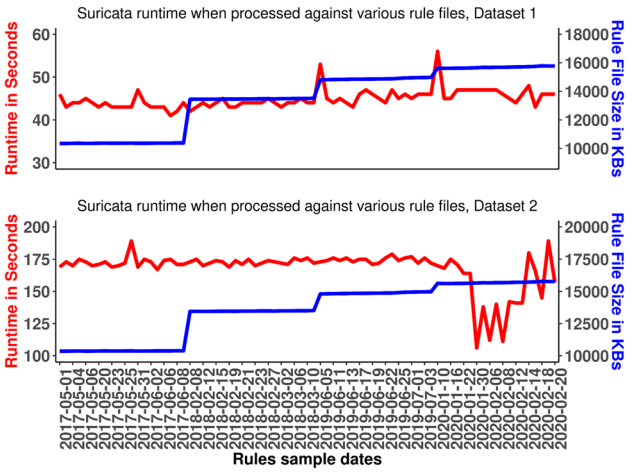

Fig. 9 Suricata runtime with respect to filesize

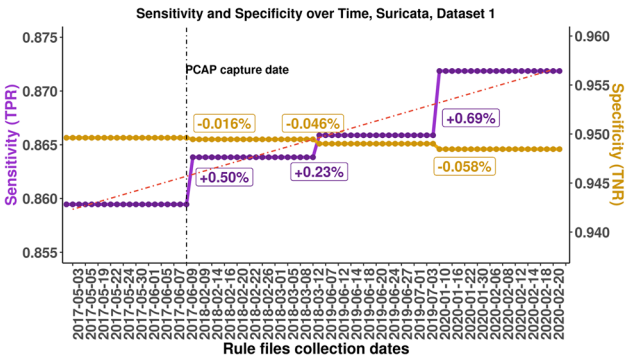

Fig. 10 Sensitivity and specificity of Suricata, dataset 1

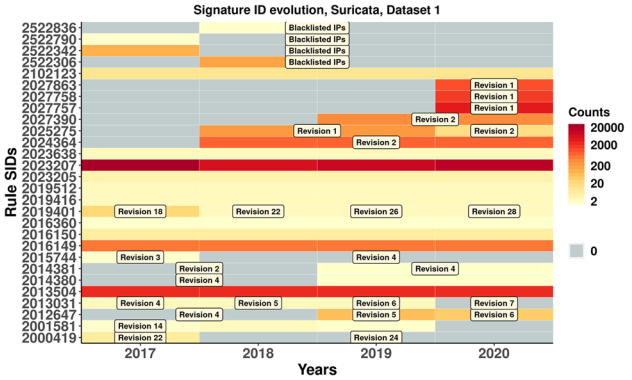

Fig. 11 Rule SID evolution, Suricata, dataset 1

are fewer SIDs that were triggered the same number of times from 2017 to 2020, as shown by the horizontal strips with no change in color in Fig. 11. Note that the gray-spaces show the absence of SIDs in the alert files in a particular year. There are two possible explanations for these gray SIDs—a rule was not there in the previous year, or it has been revised over the years. This is shown as gray with and/or without revision numbers, for no-rule and/or revised-rule, respectively. For example, rule SID 2025275 was introduced in late 2017, thus

Table 5 Revised Suricata rules from 2017 to 2020, Dataset 1

Rule SID	Rule category
2522836	Black listed IP
2522790	Black listed IP
2522342	Black listed IP
2522306	Black listed IP
2027863	Suspicious DNS query
2027758	Suspicious DNS query
2027757	Suspicious DNS query
2027390	Microsoft device metadata retrieval attempt
2025275	Microsoft device metadata retrieval attempt
2024364	Possible NMAP scan
2019401	Vulnerable Java version detected
2015744	Possible malware debugging scan
2014381	Invalid outbound HTTP HEAD
2014380	Invalid outbound HTTP POST
2013031	Python library—suspicious user agent
2012647	Off-site file backup in use
2001581	Unusual port 135 traffic—potential scan or infection
2000419	Policy violation—EXE or DLL download

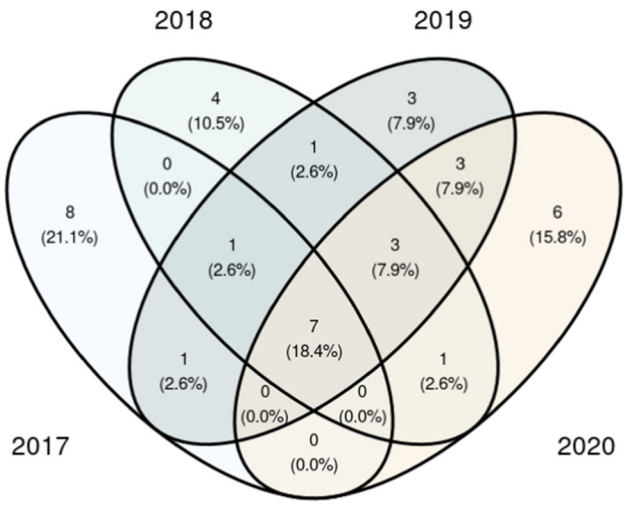

Fig. 12 Venn-diagram showing overlap between (SIDs, count) pairs of Suricata, dataset 1

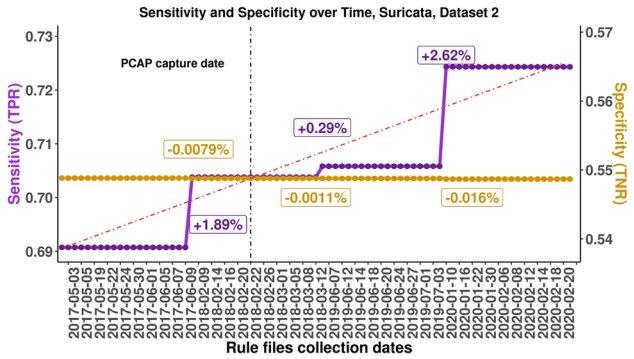

Fig. 13 Sensitivity and specificity of Suricata, dataset 2

doesn't appear in our 2017 analysis; this rule is revised again in 2019, therefore the number of alerts changes in 2020 while it remains constant during 2018–2019. However, a change in revision number doesn't always translate to variation in counts, as seen with rule SID 2019401; This particular rule goes through multiple revisions even within a year, however, the difference in counts remains minimal (2017–2018), and no difference at all (2018–2020). We also observe another interesting occurrence where rule SID 2023207 has no discernible difference going from 2017 to 2020, but the number of times this is triggered varies. Other prominent example is that of SID 2019401, where it's revision changes from 2017 to 2020 thus results into different counts. Additionally, there are rules triggered by BIPs, (2522836, 2522790, 2522342, 2522306), and they have variable counts year-on-year. This is mainly due to the changing list of blacklist IPs from 2017 to 2020. The details of various types of rules that have been triggered is shown in Table 5. The summary of various permutations of (SID, Count) pairs is shown in Fig. 12. It shows only 7 instances where the rule SID are present in all the years and their counts remain constant. Correlating this with the heatmap of Fig. 11, these rules are present in the mid-section of the heatmap. Figure 11 also shows that there are 8, 4, 3, and 6 unique (SID, Count) pairs during each year of the rule sets. Lastly, we observe the concentration of red cells in 2020, as compared to the previous years, showing the increase in the sensitivity.

Table 6 Rules revised and added over the years

Rule SIDs	2017	2018	2019	2020	Rule type
2000419	22	23	24	24	Policy violation—EXE or DLL download
2001972	19	20	20	20	Potential scan or infection
2010935	2	2	3	3	Suspicious inbound to MSSQL port 1433
2010936	2	2	3	3	Suspicious inbound to MSSQL port 1521
2010937	2	2	3	3	Suspicious inbound to MySQL port 3306
2010939	2	2	3	3	Suspicious inbound to PostgreSQL port 5432
2012063	1	2	3	3	SMB protocol potential exploit (CVE-2009-3103)
2019401	18	22	26	28	Vulnerable Java version detected
2025275	x	1	1	2	Microsoft device metadata retrieval attempt
2025649	x	x	1	1	Possible ETERNALBLUE probe (MSF Style)
2025650	x	x	2	3	ETERNALBLUE system response
2025822	x	x	2	3	Edge devices scan detected
2025992	x	x	1	3	Possible ETERNALBLUE probe
2027390	x	x	2	2	Blacklisted IPs
2027412	x	x	1	1	Blacklisted IPs
2027413	x	x	1	2	Blacklisted IPs
2027757	x	x	x	1	Blacklisted IPs
2027758	x	x	x	1	Blacklisted IPs
2027759	x	x	x	1	Blacklisted IPs

4.2.2 Analysis of dataset 2

We present the evolution of Suricata's sensitivity and specificity for dataset 2 in Fig. 13. Here, we see consistent increase in sensitivity and slight decrease in specificity year-on-year, something similar to what we observed in the previous section. However, some minor differences are visible, such as the increase in specificity is more noticeable here with about 2.42% increase in 2020. The specificity decrease is also lower, with the largest change being a 0.016% drop in 2020. The Suricata's average of both sensitivity and specificity for dataset 2 is also lower than that for dataset 1—it is approx. 0.86 for dataset 1 while it is around 0.71 for dataset 2. Figure 14 depicts the evolution of Suricata rule SIDs that were triggered from 2017 to 2020. Due to the large number of Suricata's SIDs triggered by dataset 2 and space constraint, here we have enumerated the SIDs instead of showing their actual values on the y-axis. For the same reason, we are also not able to display revision changes etc. This heatmap is rather much sparse as compared to the Suricata's analysis of dataset 1. We notice that there are fewer SIDs which have been alerted the same number of times by all rule sets, as has been depicted in lines with no color change. The prominent outlier in this heatmap is the group of SIDs, shown in the upper half of Fig. 14, that have been alerted in 2018, but have been observed very sparsely in all other years. These SIDs have been alerted by BIPs, and since these change over years, thus the difference in their alerting frequency. Table 6 gives a summary of other SIDs that were either revised or added

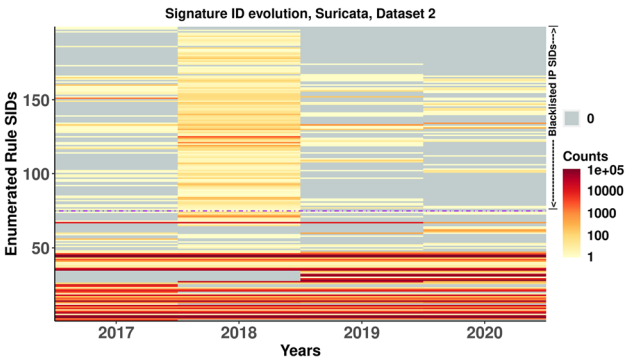

Fig. 14 Rule SID evolution, Suricata, Dataset 2

to a rule set of a particular year. Here, we show the revision number of the rules in a particular year, while 'x' denotes that the rule was not present in that year. There are 7 SIDs that were added in 2019 that resulted in quite high number of alerts in 2019 and 2020. This is also visible in the lower part of Fig. 14, where there is a gray strip before 2019. Figure 15 depicts different permutations of the overlap between the (SID, count) pairs from 2017 to 2020. This figure clearly shows that 29% of the (SID, count) pairs were unique to 2018, which are predominantly the SIDs triggered by blacklist IPs. Other noteworthy statistics is that 23% of the (SID, count) pairs are common among all years, and while there are only 2% of (SID, count) pairs that are unique to 2020, it did result into large number of alerts and have improved the sensitivity significantly.

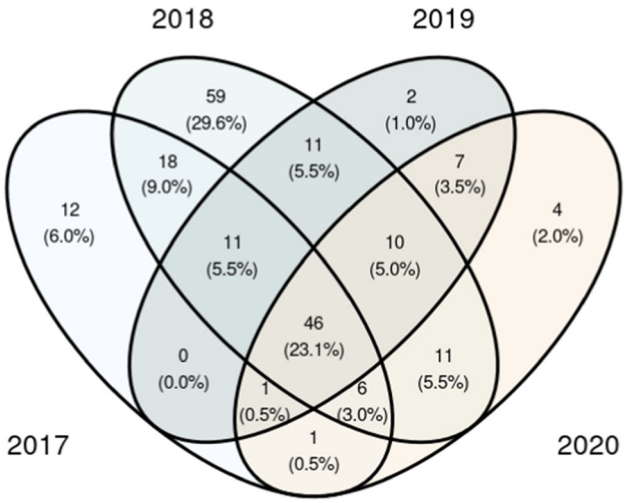

Fig. 15 Venn-diagram showing overlap between (SIDs,count) pairs of Suricata, dataset 2

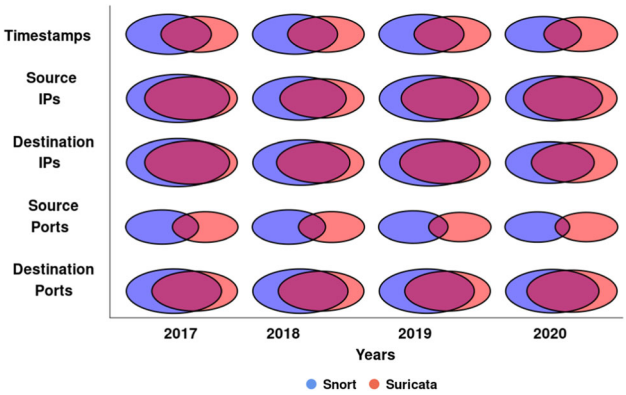

Fig. 16 Euler diagram of the fields used as flow ID

Fig. 17 Alerting pattern of Snort and Suricata

Overall, it can be inferred from the results of these two experiments that Suricata's sensitivity has improved from 2017 to 2020, while its specificity remained almost unchanged. Also, Suricata's sensitivity/specificity is higher for dataset 1 when compared to dataset 2, but that is expected given the differences in volume of traffic and size of infrastructure between the two datasets.

5 Diversity in Snort and Suricata

In this section, we report the diversity analysis of the Snort and Suricata IDSs. In this analysis, we compare the results when these IDSs are used in 1OO2 and 2OO2 configurations.

5.1 Differences in alerting behavior

During the diversity analysis, we found some interesting insights while looking for common alerts between Snort and Suricata—when using a 5 tuple ID (Timestamp, Source IP, Destination IP, Source Port, Destination Port), to compare Snort's and Suricata's alerts, we did not see much overlap. Looking deeper, we found considerably fewer overlaps in the timestamp and source port fields of the alerts as compared to other fields. This is visualized in a series of Venn diagrams in Fig. 16 for dataset 1. The timestamp-based alerting behavior for Snort and Suricata is also depicted in Fig. 17 for dataset 1. Here, the lower plot is the distribution of malicious (red dots) and benign (green dots) flows, while the upper two plots are the distribution of Snort and Suricata TP (in red) and FP (in green), respectively, over a time window when dataset 1 was generated. The top two plots have four sub-plots, one each using 4 years of rules. It is quite evident from Fig. 17 that while the Snort TPs are quite spread, the Suricata TPs are rather concentrated at different time points. For example, for the window of attacks between 14:00 and 16:00, we note that Suricata generates alerts between 14:00 and 15:00, whereas Snort's alerts are more spread out from 14:00 to 16:00. This may be explained by the Snort tendency to group malicious flows, and generate a single alert if they are close to each other [14]. We believe that while the fewer timestamp overlaps is quite random, the disparity in source port between Snort and Suricata can be explained in the way source ports are assigned in TCP and UDP connections. Whenever a new connection is established, source/destination IP addresses are known while the destination port is tied to the service being contacted. It is only the source port which is assigned randomly from a list of ephemeral ports. Essentially, this means that Snort and Suricata do not always generate alerts on the same exact flow, but may produce alerts on flows immediately later or earlier than each other. For the reasons mentioned above, for the rest of the diversity analysis, we only use Source IP, Destination IP, and Destination Port as ID.

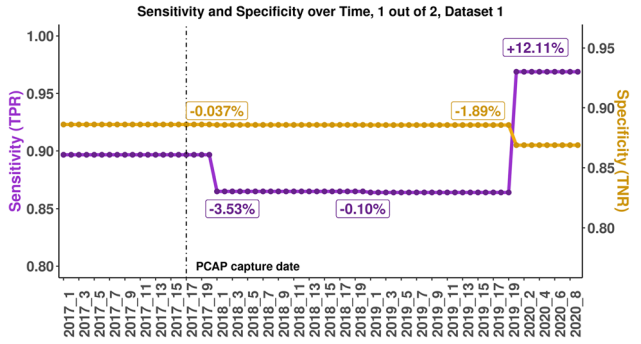

Fig. 18 Evolution of sensitivity and specificity of 1OO2 system, dataset 1

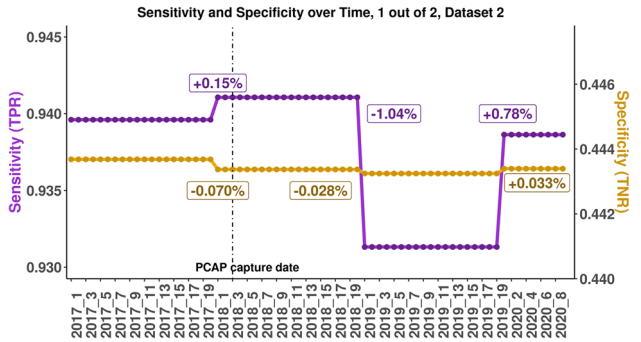

Fig. 19 Evolution of sensitivity and specificity of 1OO2 system, dataset 2

5.2 Diversity analysis of 1OO2 system

5.2.1 Analysis of dataset 1

The evolution of sensitivity and specificity of 1OO2 system using dataset 1 is shown in Fig. 18. Noticeable here is, on average, the sensitivity of 1OO2 system is much higher than those of both Snort and Suricata for dataset 1. This remains true throughout, while it goes down by 3.5% and 0.1% from 2017 to 2018 and 2018 to 2019, and goes up by more than 12% from 2019 to 2020. While the absolute value of sensitivity for the 1OO2 system is higher, the evolutionary trend is close to that of Snort. However, the specificity of the 1OO2 system remains lower than that of both Snort and Suricata, and it changes very insignificantly year-on-year. We attribute the evolution of both sensitivity and specificity of the 1OO2 systems to the changes in the rules of Snort or Suricata, as has been discussed in Sect. 3. Note that, while it is the Suricata's higher sensitivity, shown in Fig. 10, having more weightage in pushing the 1OO2 system's overall sensitivity upward, the Snort's lower specificity, given in Fig. 3, seems to be playing an opposite role in pushing the overall sensitivity of the 1OO2 system down.

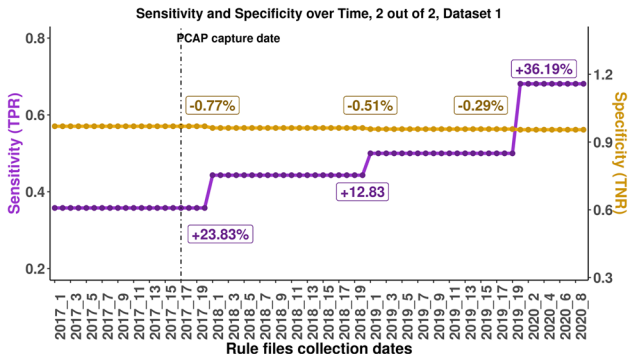

Fig. 20 Evolution of sensitivity and specificity of 2OO2 system, dataset 1

5.2.2 Analysis of dataset 2

Similarly, Fig. 19 depicts the evolutionary trends of sensitivity and specificity for the 1OO2 system using dataset 2. Here too, while the evolution of sensitivity has seen ups and downs by not more than 1% year-on-year, on average, it remains much higher than those of Snort and Suricata for dataset 2. However, the specificity remains much lower than those of Snort and Suricata for the same dataset 2. This is due to the fact that 1OO2 systems are inherently more sensitive rather than specific, as we combine the alerts of both IDSs thus pushing both TPs and FPs upwards. For dataset 2, both Snort's and Suricata's individual sensitivities, shown in Figs. 6 and 13, seem to contribute almost equally to the higher sensitivity of the 1OO2 system, it is the Suricata's lower specificity having more impact on the lower specificity of this system.

Looking at the results of 1OO2 systems for both datasets, it can be inferred that it performs much better against malicious traffic, and it has much higher sensitivity as compared to the individual IDSs. We have also observed the impact of the evolution of the individual IDSs on the sensitivity and specificity of this system.

5.3 Diversity analysis of 2OO2 system

5.3.1 Analysis of dataset 1

We show the evolution of sensitivity and specificity of the 2002 system for dataset 1 in Fig. 20. Noteworthy here is, that specificity of the 2OO2 system remains almost constant, with variations not more than 0.7% at a particular point in time. The average value of specificity is also higher than those of individual IDSs for the same dataset 1, as shown in Figs. 3 and 10. Conversely, the sensitivity of the 2OO2 system, though follows an upward trend year-on-year with jumps as high as 36% (2019 to 2020), it remains lower than the sensitivities of the individual IDSs. While the 2OO2 system's sensitivity and that of Suricata in Fig. 10 has a close resemblance, Snort does

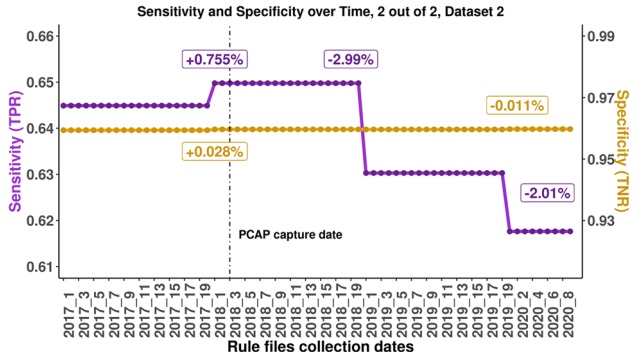

Fig. 21 Evolution of sensitivity and specificity of 2OO2 system, dataset 2

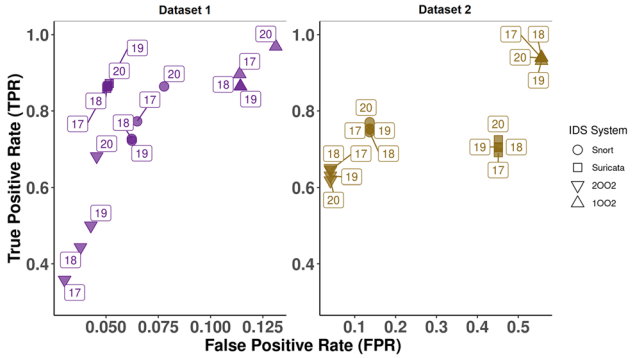

Fig. 22 ROC points of different detection systems

have a considerable impact on its large increase of more than 36% going from 2019 to 2020 (Fig. 3). Similarly, the higher specificities of both Snort and Suricata for dataset 1 seem to have played similar roles in maintaining the higher sensitivity of the 2OO2 system for this dataset.

5.3.2 Analysis of dataset 2

The evolution of 2OO2 system's sensitivity and specificity, using dataset 2, is shown in Fig. 21. Here too, we observe that the sensitivity not only has a lower average value than those of the individual IDSs, as shown in Figs. 6 and 13, it also demonstrates an overall decreasing trend—although it increases by 0.7% from 2017 to 2018, it, however, decreases by about 5% from 2018 to 2020. In addition, the specificity of 2OO2 system remains higher than that of the individual IDSs for the same dataset 2, and it changes very insignificantly from 2017 to 2020.

Summarizing the analysis of the 2OO2 system using the two datasets, we can argue that this system is more biased toward the benign traffic and thus has higher specificity than the individual IDSs. In addition, while the functional diversity that exists between Snort and Suricata has an impact on the overall performance of the 2OO2 system, this goes through the similar evolution as that of the individual IDSs.

5.4 Overall analysis

An ideal IDS system is the one having high TPR and low False Positive Rate (FPR). On a typical Receiver Operating Characteristic (ROC) plot, this will correspond to points in the top left corner. In Fig. 22, we show average values of TPR and FPR for each year, each dataset and for all IDS systems we have studied so far. We notice that, in both the datasets, the 1OO2 system has the highest TPR, but also have the highest FPR. Since dataset 2 is a larger dataset, we see a large increase in the FPR of Suricata pushing it and the 1OO2 system out of the ideal zone. The 2OO2 system in both datasets have the least FPR and TPR. This figure also shows that the Suricata TPR is better than that of Snort, while it is the Snort that has a lower FPR. In addition, we also notice that the impact of evolution of individual IDSs is more on the 2OO2 system as compared to the 1OO2 system. This can be seen from the spread of points for these systems and for each dataset. Lastly, however very unpredictable, but the data type and size impacted Suricata more than Snort, as can be seen by the spread of their TPR/FPR points for the two datasets.

6 Discussion and limitations

The results of this work are interesting as it shows that functional diversity exists between Snort and Suricata, and that this diversity also evolves over time. While this has also been shown previously elsewhere using unlabeled data, we have empirically demonstrated this using labeled PCAP data. Given that we selected a PCAP data with attacks that were famous at the time, our results show that it is not always a case of linear improvement in the performance of Snort and Suricata IDSs. While Suricata showed almost a linear improvement/degradation in its sensitivity/specificity, it is, however, much nonlinear for Snort. The Snort ability to identify TPs first degrades before it improves over the years. However, we observe that Snort performed much better in labeling the TNs and that its specificity evolution showed minimal variations over the years. Based on the two experiments we undertook, Suricata performed better than Snort for the malicious traffic, whereas it is Snort that outperformed Suricata in labeling the benign traffic. Linking the evolutionary behavior of IDSs to the changes of rules that were triggered, we found out that these IDSs evolved as a result of changes to relatively small number of rules and BIPs. In Snort, while both other types of rules and changes in BIPs have contributed to the evolution of its behavior, it is more due to other rules initially, but it is due to blacklist IPs from 2019 to 2020 that we observed the biggest change in its performance. Similarly, Suricata has seen changes to both its BIPs

rules and other types of rules, but on the whole, the contribution of other types of rules in improving its performance is relatively more than the blacklist IP rules. Noteworthy here is, that both Snort and Suricata had seen a major overhaul of rules in 2020, which resulted into large number of both TPs and FPs.

Using the functional diversity between Snort and Suricata, we have also analyzed the efficacy of 1OO2 and 2OO2 diverse system. We observe that, overall, 1OO2 system performs much better to label malicious traffic, while it is the 2OO2 system that performs superior to any other configuration in order to detect benign traffic. The evolution of Snort and Suricata, however, impact the evolution of 1OO2 and 2OO2 systems very differently. While the evolution of 1OO2 system does follow certain trends of both Snort and Suricata, especially from 2019 to 2020, the evolution of 2OO2 system show some degree of uncertainty. In fact, we observe that while the 2OO2 system show more variations as a result of evolution of the individual IDSs, the 1OO2 system remains relatively stable.

We believe that these results must be treated with caution, as there may not always be a linear relationship between the types of attacks/exploits and that of changes in the IDSs, and that it may not be a straightforward conclusion that one type of IDS has improved over the other. Since our analysis use Snort and Suricata rules in their default configurations, these may have a significant impact on their TPR/TNR. In addition, the results of diversity analysis are also based on the default rules configurations of Snort and Suricata, which are subjected to changes given a different context. A TP alert in one context may be something very harmless in another. For example, looking at the signatures of the rules alerted in our experiment, we found a number of alerts for RDP protocols across all years. While most organizations are very cautious about the RDP traffic and therefore use rules to monitor RDP traffic, but for the creators of the dataset used in the experiment, all RDP traffic is benign.

Given our limited processing and storage resources, we have only used two datasets of modest sizes with a certain set of attacks. This may have impacted our findings and with larger PCAP files, covering many days and having more diverse balanced data, the results could certainly be improved. Besides, while our choice of selecting only 20 (or so) rules was pretty random, the files themselves were not selected randomly, and will have impacted the results. However, we were limited in our choice given that for some years, we only had the rules that were used in this experiment, and we did not have the option to choose them randomly. Having said that, these results may still be very significant in at least highlighting that diverse IDSs may be an option in improving the security of an Enterprise network.

These results are also important in pointing out that while the developers of these IDSs continuously strive and improve their performance, the speed of this evolution may not be something one would expect. As we have witnessed, that given the same types of attacks, the IDSs did take some time before they showed significant improvement in their performance. Besides, the diversity analysis leads us to some important observations and suggestions for their deployment in a real-world scenario.

Deploying the two IDSs in parallel and using an adjudication mechanism, we could use the best of both systems to improve the overall performance of an integrated IDS. Depending on whether an organization is currently experiencing an environment with many attacks, or a relatively quiet period where most of the traffic is benign, the IT administrators could, for example, tune the diversity configuration from 1OO2 for the former (to catch as many attacks as possible, even with an increase in false positives) or 2OO2 for the latter (to reduce the false positive rate). The exact tuning would inevitably depend on the relative cost of these two types of failures. There, of course, will be overheads from processing and storage resources and that of the additional cost of multiple IDSs. More empirical studies will be needed to optimize the time taken to process the alerts in a 1OO2 or 2OO2 topology, and that the task could be distributed using a hybrid approach of "on-premise"- and "cloud"-based IDSs.

7 Related work

The work presented in [15] provides an extensive look at the **tools and metrics used to evaluate IDSs** in existing literature. The majority of works included in this survey use the same metrics used in our analysis, namely: Sensitivity, Specificity and ROC plots. They also note that the choice of these metrics are dependent upon the analysis being conducted and also the underlying hardware where the test is being performed. Another extensive survey work in [16] concludes that much depends upon the tuning of the rules according to the infrastructures requirements. They note that IDSs in their default configuration will generate too many false positives. They also discuss an obstacle that restricts the adoption of diverse systems—vendors are more likely to incorporate proprietary alerts and event formats that will massively hinder interoperability. This will pose problems, especially in processing overhead if the adjudicating systems will need to perform additional operations in order to meaningfully correlate alerts between different IDSs. In [17], the authors present the state of the art for automatic signature generation methods in a bid to alleviate the drawback of signature-based system's inability to detect new attacks. They mention various implementations attempting to integrate signature-based and anomaly-based IDSs. Their goal is to implement techniques that can detect 0-day attacks and generate signatures for it in real time. To achieve this goal, [18] proposes a promising

system that can generate signatures closely matching those created by human specialists. These systems allow for high autonomy in IDSs, which is desirable in order to keep up with ever-increasing traffic volume and speed.

Snort and Suricata have both been **compared** previously in works like [19–21]. All three studies highlight the IDS's reliance on CPU and memory, as well as their effectiveness in terms of dropped packets and detection accuracy. They also report the limitations of Snort and Suricata in a high speed network environment, where network speed of over 40 GB/s pose problems due to dropped packets. Our setup circumvents this problem by running the PCAP file in read mode instead of replaying it over the network. Furthermore, our study focuses on benefits of using the IDSs together rather than a comparison against each other. Other studies investigating accuracy and performance when presented with a larger network traffic report similar results [22, 23]—i.e., Suricata with it's multi-threaded capabilities performs better in terms of processing speed and packet drop rates, however with reduced accuracy in traffic labeling. These findings are also reflected in our experiments, where Suricata has lower runtimes when compared to Snort, but the sensitivity and specificity values are not ideal when processing the larger dataset. Salah presents a comparison of the performance of these IDSs across different operating systems, [24]; It was emphasized that the performance of IDSs is also influenced by the underlying operating system, therefore to mitigate this dependency, our study exclusively focuses on the Linux OS.

The **benefits of diversity** between IDSs is empirically studied in [25], and they also provide numerical evidence demonstrating the advantages of using functionally similar IDSs. Our work differs from this as we evaluate the dynamic evolution of this diversity over four years. Our study can also be considered as a direct continuation and improvement upon [2] where Snort and Suricata rules were found to be diverse. Given this diversity, we endeavored to see what effects this diversity in the rules, called configurational diversity, would have on the detection performance of the two IDSs.

Comparing two survey papers, [26] from 2018 and [27] from 2023 provides an interesting insight: there is a shift of research focus from traditional signature- and heuristic-based techniques to **machine learning techniques**. This is unsurprising given the applicability of Machine Learning techniques in a plethora of fields due to their ability to process vast amount of data to extract meaningful insights, identify patterns and make predictions based on the data. Furthermore, [28] discusses the unique needs of the security domain, including the presence of an active adversary, non-stationarity of data, asymmetrical costs of misclassification, and the challenge of zero-day attacks, which are better addressed with data driven techniques such as machine learning. Works like [29–31] are heavily focused on using machine learning techniques to improve upon the current intrusion detection capabilities. Applying fuzzy logic-based techniques to prioritize alerts has been discussed in [32], however the metrics they use as input for their fuzzy logic includes concepts such as the relationship between the alert under evaluation and previous alerts, and the social activities between the attacker and victim. In contrast, our work presented here is different: Our study focuses more on demonstrating the evolution of the IDSs, the diversity between them, and the effects of the evolution in the diversity. We establish that the alerts generated are different between the IDSs, but how this difference can be used requires further work, we provide some suggestions based on our observations, but alert prioritization is not necessarily the primary goal of this paper.

Another limitation of our study is our dependence on the two publicly available datasets from CIC. There has been a recent criticism of the use of these datasets for optimizing IDSs, especially for approaches that use ML [33]. We think this critique of the dataset, while valid, is not a critical factor on our analysis for the following reasons: we are not attempting to optimize an ML algorithm by training it on this data; we are not evaluating an up-to-date IDS using this older data (which would be unfair). The aim of our study is to study the evolution of IDSs and their ability to correctly classify network traffic with rulesets as they were for the time windows that we studied (2017–2020).

8 Conclusions

In this paper, we have presented a perspective and retrospective analysis of the Snort and Suricata IDSs, using labeled PCAP data and rules from the past and future dates with respect to the time when the PCAP data was collected. We have used two sets of PCAP data from the CIC, and have analyzed this using rules from 2017, 2018, 2019 and 2020. Likewise, we have also analyzed the efficacy of diverse 1OO2 and 2OO2 systems and have investigated the impact of the evolution of the individual IDSs on these systems. We have shown that while the sensitivity of the individual IDSs do improve over the years, this comes with a cost of some degradation of their specificities. We have observed that Suricata shows more linear progress in terms of it's improvement/degradation in sensitivity/specificity, while it is very nonlinear for Snort. Investigating this evolution further, we have identified the changes in rules that had the impact on the performance of these IDSs. Following are the main conclusions that we draw from this work:

– Suricata's sensitivity increased when processing both datasets as the rules were updated, however Snort displays a different trend, sensitivity drops for a couple of years then jumps significantly in 2020.

- The evolution of detection performance was more linear for Suricata and nonlinear for Snort given the two labeled datasets.
- The rate of evolution was low from 2017 to 2019, but it was much higher from 2019 to 2020. This is true for both the IDSs.
- The change in BIPs and pre-processor rules contributed more to the evolution of Snort as compared to other types of rules. It is the other types of rules in Suricata that contributed more to its evolution.
- Suricata performed better with the smaller dataset, whereas Snort did better with the larger dataset. However, Suricata was more sensitive to the data size/type as compared to Snort.
- As expected, the 1OO2 system performs better for malicious traffic, and that the evolution of both Snort and Suricata do impact its performance. It is, however, the higher Suricata's sensitivity that seem to be contributing more to the higher sensitivity of the 1OO2 system.
- The 2OO2 system performs better for benign traffic, and that the evolution of Snort and Suricata do affect its performance, albeit very unpredictably.
- While the diverse system's detection performance drops in certain years, the performance always remains higher than the individual systems.
- Evolution of individual IDSs impacts the 2OO2 system more than the 1OO2 system.

We have highlighted that the open-source signature-based IDSs do evolve, using new signatures and BIPs. However, the rate of evolution may be very unpredictable for some IDSs, and it may be much slower than what one would expect. We have also underscored that great functional diversity exists between Snort and Suricata, and that it would be useful to consider using a combination of these IDSs in order to improve the security of a given system. We hope that our results will instil more confidence into the decision-making process about the deployment of multiple IDSs. This may especially be the case, where security architects have to protect important infrastructure and where even small improvements are very critical.

As further work, we plan to investigate the diversity with IDSs and other defence-in-depth tools in real deployments, with labeled datasets, to assess the benefits as well as potential harm that diversity may bring due to the interplay between the risks from FN and FP. We plan to use large balanced PCAP data, spreading over several days, and rules evenly and randomly spread over years. Likewise, we plan to extend our work to anomaly-based IDSs as well. Currently, we are investigating the efficacy of diverse IDSs in IoT networks.

Acknowledgements Part of this work was supported by the UK EPSRC Project D3S Award Number EP/M019462/1 and in part by the EU H2020 framework DiSIEM Project Award Number 700692.

Data availability The datasets analyzed and generated during the current study are available in the GitHub repository: github.com/Shashwat Adh/Evolution-and-Diversity-of-IDS

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Asad, H., Gashi, I.: Diversity in open source intrusion detection systems. In: International Conference on Computer Safety, Reliability, and Security, pp. 267–281. Springer (2018)
2. Asad, H., Gashi, I.: Dynamical analysis of diversity in rule-based open source network intrusion detection systems. Empir. Softw. Eng. **27**(1), 1–30 (2022)
3. Canadian Institute for Cybersecurity. CIC - University of New Brunswick. https://www.unb.ca/cic/about/hub.html (2022). Accessed 03 Jan 2022
4. Pathan, A.-S.K.: The State of the Art in Intrusion Prevention and Detection. CRC Press, Boca Raton (2014)
5. Snort Rules: https://snort.org/documents/registered-vs-subscriber (2021). Visited on 18 Apr 2021
6. Emerging Threat Rules. https://rules.emergingthreats.net/open/suricata/ (2021). Visited on 18 Apr 2021
7. Snort Blacklists. https://talosintelligence.com/documents/ip-blacklist (2021). visited on 18 Apr 2021
8. Cummings, J.J., Shirk, M.: Pulledpork. https://github.com/shirkdog/pulledpork
9. Suricata Update Tool: https://suricataupdate.readthedocs.io/en/latest/ (2021). Visited on 18 Apr 2021
10. Snort logs: http://manual-snortorg.s3-website-us-east-1.amazonaws.com/node21.html (2021). Visited on 18 Apr 2021
11. Suricata logs: https://suricata.readthedocs.io/en/suricata-6.0.2/output/eve/eve-json-output.html (2021). Visited on 18 Apr 2021
12. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, vol. 1, pp. 108–116 (2018)
13. Realistic Cyber Defense Dataset (CSE-CIC-IDS2018). https://registry.opendata.aws/cse-cicids2018 (2018). Accessed 01 May 2022
14. Granberg, N.: Evaluating the effectiveness of free rule sets for Snort. MA thesis, Linköping University-Department of Computer

and Information Science. http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-183361 (2022)
15. Milenkoski, A., et al.: Evaluating computer intrusion detection systems: a survey of common practices. ACM Comput. Surv. (CSUR) **48**(1), 12 (2015)
16. Tidjon, L.N., Frappier, M., Mammar, A.: Intrusion detection systems: a cross-domain overview. IEEE Commun. Surv. Tutor. **21**(4), 3639–3681 (2019)
17. Kaur, S., Singh, M.: Automatic attack signature generation systems: a review. IEEE Secur. Priv. **11**(6), 54–61 (2013)
18. Garcia-Teodoro, Pedro, et al.: Automatic generation of HTTP intrusion signatures by selective identification of anomalies. Comput. Secur. **55**, 159–174 (2015)
19. Alhomoud, Adeeb, et al.: Performance evaluation study of intrusion detection systems. Procedia CS **5**, 173–180 (2011). https://doi.org/10.1016/j.procs.2011.07.024
20. Hu, Q., Yu, S.-Y., Asghar, M.R.: Analysing performance issues of opensource intrusion detection systems in high-speed networks. J. Inf. Secur. Appl. **51**, 102426 (2020)
21. Yang, J., et al.: A high-performance round- robin regular expression matching architecture based on FPGA. In: 2018 IEEE Symposium on Computers and Communications (ISCC), pp. 1–7 (2018). https://doi.org/10.1109/ISCC.2018.8538459. ISSN: 1530-1346
22. Shah, S.A.R., Issac, B.: Performance comparison of intrusion detection systems and application of machine learning to Snort system. Future Gener. Comput. Syst. **80**, 157–170 (2018)
23. Alqahtani, S.M., John, R.: A comparative study of different fuzzy classifiers for cloud intrusion detection systems' alerts. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–9. IEEE (2016)
24. Salah, K., Kahtani, A.: Performance evaluation comparison of Snort NIDS under Linux and Windows Server. J. Netw. Comput. Appl. **33**(1), 6–15 (2010). https://doi.org/10.1016/j.jnca.2009.07.005. (**ISSN: 1084-8045**)
25. Algaith, A.: Diversity with intrusion detection systems: an empirical study. In: IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–5. IEEE (2017)
26. Jose, S., et al.: A survey on anomaly based host intrusion detection system. J. Phys. Conf. Ser. **1000**(1), 012049 (2018). https://doi.org/10.1088/1742-6596/1000/1/012049. (**ISSN: 1742-6596**)
27. Dina, A.S., Manivannan, D.: Intrusion detection based on machine learning techniques in computer networks. Internet Things **16**, 100462 (2021). https://doi.org/10.1016/j.iot.2021.100462. (**ISSN: 2542-6605**)
28. Verma, R.: Security analytics: adapting data science for security challenges. In: Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics. IWSPA'18, pp. 40–41. Association for Computing Machinery, New York, NY, USA. ISBN: 9781450356343. https://doi.org/10.1145/3180445.3180456 (2018)
29. Ahmad, Z., et al.: Network intrusion detection system: a systematic study of machine learning and deep learning approaches. Trans. Emerg. Telecommun. Technol. **32**(1), e4150 (2021)
30. Alauthman, M., et al.: An efficient reinforcement learning-based Botnet detection approach. J. Netw. Comput. Appl. **150**, 102479 (2020)
31. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A.: Application of deep reinforcement learning to intrusion detection for supervised problems. Expert Syst. Appl. **141**, 112963 (2020)
32. Alsubhi, K., Al-Shaer, E., Boutaba, R.: Alert prioritization in intrusion detection systems. In: NOMS 2008—2008 IEEE Network Operations and Management Symposium, pp. 33–40. https://doi.org/10.1109/NOMS.2008.4575114 (2008)
33. Catillo, M., Pecchia, A., Villano, U.: Machine learning on public intrusion datasets: academic hype or concrete advances in NIDS? In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks—Supplemental Volume (DSN-S), pp. 132–136. IEEE Computer Society, Los Alamitos, CA, USA. https://doi.org/10.1109/DSN-S58398.2023.00038 (2023)