2021-10-22

Dynamical Analysis of Diversity in Rule-Based Open Source Network Intrusion Detection Systems

Asad, Hafiz ul

10.1007/s10664-021-10046-w
Empirical Software Engineering: an international journal
Springer Verlag

Noname manuscript No.
(will be inserted by the editor)

Dynamical Analysis of Diversity in Rule-Based Open Source Network Intrusion Detection Systems

Hafizul Asad* · Ilir Gashi

Received: date / Accepted: date

Abstract Diverse layers of defence play an important role in the design of defence-in-depth architectures. The use of Intrusion Detection Systems (IDSs) are ubiquitous in this design. But the selection of the "right" IDSs in various configurations is an important decision that the security architects need to make. Additionally, the ability of these IDSs to adapt to the evolving threat-landscape also needs to be investigated. To help with these decisions, we need rigorous quantitative analysis. In this paper, we present a diversity analysis of open-source IDSs, Snort and Suricata, to help security architects tune/deploy these IDSs. We analyse two types of diversities in these IDSs; configurational diversity and functional diversity. In the configurational diversity analysis, we investigate the diversity in the sets of rules and the Blacklisted IP Addresses (BIPAs) these IDSs use in their configurations. The functional diversity analysis investigates the differences in alerting behaviours of these IDSs when they analyse real network traffic, and how these differences evolve. The configurational diversity experiment utilises snapshots of the rules and BIPAs collected over a period of 5 months, from May to October 2017. The snapshots have been collected for three different off-the-shelf default configurations of the Snort IDS and the Emerging Threats (ET) configuration of the Suricata IDS. The functional diversity investigates the alerting behaviour of these two IDSs for a sample of the real network traffic collected in the same time window. Analysing the differences in these systems allows us to get insights into where

Hafizul Asad
School of Engineering, Computing & Mathematics,
University of Plymouth
Tel.: +44-1752586238

Ilir Gashi
Department of Computer Science,
School of Mathematics, Computer Science and Engineering,
City, University of London
Tel.: +44-2070400273
* Corresponding author: Hafizul Asad , E-mail: hafizul.asad@plymouth.ac.uk

the diversity in the behaviour of these systems comes from, how does it evolve and whether this has any effect on the alerting behaviour of these IDSs. This analysis gives insight to security architects on how they can combine and layer these systems in a defence-in-depth deployment.

Keywords Security · Diversity of Security Tools · Evolution of Diversity · Intrusion Detection Systems

1 Introduction

An important paradigm in security design is defence-in-depth: "layering" defences to reduce the probability of successful attacks. Guidance documents now advocate defence-in-depth as an obvious need, but their qualitative guidance ignores the decision problems[1]. Crucially, these problems concern the effectiveness of diversity: defences should be diverse in their weaknesses. Any attack that happens to defeat one defence should with a high probability be stopped or detected by another one. Ultimately, diversity and defence in depth are two facets of the same defensive design approach. The important questions are not about defence in depth being "a good idea", but about whether a set of specific defences would improve security more than another set; and about—if possible—quantifying the security gains.

Network Intrusion Detection Systems (NIDSs) are some of the most widely used security defence tools. Some of these NIDSs are available open-source, and the most widely used open-source NIDSs are Snort [2], Suricata [3] and Zeek [4] (Previously known as Bro). While Snort and Suricata are signature-based and rely on rules to identify malicious activity, Zeek uses customised scripts to detect anomalies/violations-of-policies in the traffic. An open-source Host-based IDS (HIDS), Wazhu [5], is both signature and anomaly based. In this paper, we focus on the rule-based NIDSs, namely Snort and Suricata, since they are the most widely used NIDSs and follow similar architecture, making the diversity analysis more suitable. 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.

While from the software engineering point of view there are many differences between Snort and Suricata that could potentially affect their alerting behaviours, it is the differences and variation in the signature rules and Blacklisted IP Addresses (BIPAs) that would be expected to be mainly responsible for the diversity in their detection capabilities. While other performance metrics may also be used to compare these IDSs, such as packet-processing speed, drop-rates etc, their contribution towards diversity would be relatively small compared to the contribution from rules and BIPAs. Rules and BIPAs are added, modified or deleted regularly. In a previous work [6], we analysed the

evolution of the rulesets and BIPAs of Snort and Suricata IDSs over 5-months from May to October 2017. Analysing the differences, and how these evolve, allows us to get insights into where the diversity in the behaviour of these systems comes from. The new work presented here is an extension of that paper with new results about the evolution of diversity in the alerting behaviour. While we reuse the results of our previous work for the configurational diversity analysis, the functional diversity is something completely novel in this paper. Besides, additional sections and text have been added to explain the experiment in more detail.

In this paper, we present an empirical study to investigate the configurational and functional diversities in the Snort and Suricata IDSs. The configurational diversity deals with identifying the differences, if there are any, between the Snort and Suricata rules and BIPAs. Note that we perform this comparison by considering three Snort rulesets, namely "community", "registered" and "subscribed"; we also use "ET ruleset" for Suricata. There are no seperate rulesets for BIPAs per NIDS: the diversity analysis is performed on two sets of BIPAs associated respectively with Snort and Suricata. The functional diversity deals with the alerting behaviour of these IDSs against real-world traffic. Specifically, we investigate whether these two IDSs are different in their alerting behaviour when subjected to the same network traffic. Essentially, these two types of analysis are complementary. If rulesets and BIPAs are different (e.g. having different regular expressions as signatures, analyse different types of payloads in the traffic etc.), then that should be reflected in the alerting behaviour of these IDSs as well. The functional diversity analysis is thus a testing of the configurational diversity using real-world network data. This analysis is not only static but dynamic in time: we do not only compare sets of data of a particular time-window, e.g., 24 hours, but of a moving time window over several months. The dynamic analysis allows us to see the way rules are modified, added or discarded, as well as the changes in the set of BIPAs. Similarly, analysing the network traffic, of a particular time window, by the rules/BIPAs of the corresponding time window as well as by those collected in the past and future time windows, we gain insights on the evolution of the alerting behaviour of these IDSs. The configurational diversity analysis makes use of the rules and BIPAs data over a 5-month period, and collected between May and October 2017. The functional diversity analysis presents alerts data of these two IDSs by sniffing out a representative sample network traffic from City, University of London DMZ network. These analyses allow us to get answers to questions such as, for the same traffic, does a later configuration of Snort/Suricata generate more alerts compared to an earlier Snort/Suricata configuration? Do we see deterministic behaviour in the alerting behaviour (i.e., does the same rule on the same traffic always raise an alert?) How long does it take for traffic that was not alerted by an earlier configuration of Snort/Suricata to be alerted by a later configuration? Do we observe any alerts that have been alerted by both Snort/Suricata? etc. In this paper, we provide answers to some of these questions, which will help security architects and other researchers to gain more insight on how these products

evolve, what diversity exists between them, and what effect does this evolution of IDSs have on alerting behaviour when analysing real network traffic. To the best of our knowledge, a similar study has not been reported elsewhere.

The rest of the paper is organised as follows: Section 2 gives a background of the NIDSs; this is followed by a description of our data collection Infrastructure in section 3. In sections 4 we discuss the configurational diversity analysis between Snort and Suricata IDSs. Section 5 discusses the functional diversity analysis. Section 6 presents a discussion and limitations of the results. Section 7 presents our proposed IDSs deployment strategies, followed by the related work in section 8. Section 9 concludes the paper.

2 Signature based IDS Background

An IDS is a system that can potentially differentiate between malicious and benign network 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. 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 [7]. An IDS can be deployed as a passive sensor—where it can analyse 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 Intrusion Prevention System (IPS). An IDS software system consists of several sub-systems as shown in Figure 1 for Snort IDS (other signature-based IDSs may have other sub-systems/plugins, but essentially follow the same structure). The sensor part of the IDS consists of several libraries and software systems. The network traffic is captured using Packet-Capture (Pcap) libraries, such as LibPcap for Linux and WinPcap for Windows. The packet decoder converts the traffic into the relevant data structures and strips out the TCP, UDP and ICMP protocols. The headers of different packets are checked for any inconsistencies. The correct packets are further processed by the pre-processor block so that they are in the format to be used by the detection engine. Pre-processors are also used to detect any malicious ports. The detection engine is the main software sub-systems detecting malicious traffic using rules. The rules for both Snort and Suricata follow the same structure and have various fields—actions, protocols, source/destination IP, source/destination 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 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 [8]. 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

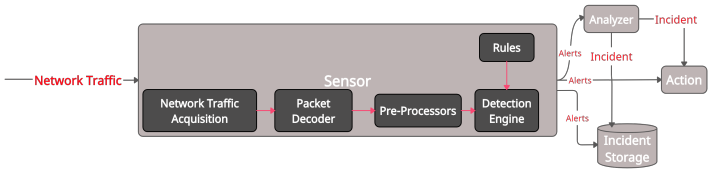


Fig. 1: Snort IDS Architecture

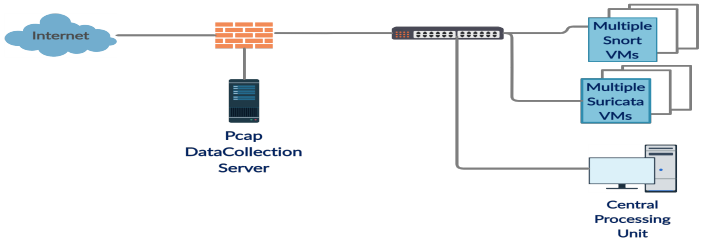


Fig. 2: Data Collection Infrastructure

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 [9]. 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 [10]. These rules and BIPAs can be automatically updated using tools such as Pulledpork [11], and Suricata update [12]. Alerts generated by the sensor are sent for storage or to an analyzer. The analyzer can also access the storage for further analysis of the alerts so that actions are taken accordingly. Both Snort and Suricata offer various ways of customized logs that can be saved or sent to various logs-plugins [13][14]. The information in the logs/outputs can be from TCP packets or only the malicious alerts, depending on the mode of an IDS. 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 DATA COLLECTION INFRASTRUCTURE

3.1 The Architecture

A representative block diagram of our data collection and the experimental setup is depicted in Figure 2. Except for the DMZ network and the firewalls, the rest of the network is virtual. We have used multiple virtual machines (VMs) to collect and process the data. The Packet Capture (pcap) data of

the City, University of London have been collected in the server at the DMZ network, whereas the Snort and Suricata rules and BIPAs have been collected in our localised virtual environment. The virtual environment is based on VMware VSphere data center using the HPE ProLiant BL460c Gen9 blade servers. This data collection setup has 10 data hosts each having 150 TB storage capacity, 200 GB RAM of 2400MHZ effective speed, 32×2.3GHz Intel Xeon E5-2650V4 CPUs, and network speed of 10Gb. There are 2 hosts for Suricata, 7 are for Snort, and 1 host serves as our centralized data processing machine based on Windows operating system. Suricata, being capable of multi-threading, could analyse multiple files in parallel with the help of only two hosts. On the other hand, we ran 7 Snort instances on 7 separate hosts to catch up with the speed of processing the same number of files, as Snort did not support multithreading at the time. The live traffic was saved in pcap format in the DMZ network. Snort and Suricata then analysed the saved pcap data. At the start of the experiment, we installed the latest versions of these IDSs on the Ubuntu operating system: Snort 2.9.9.0 and Suricata 3.2.1. Note that these were the versions of these IDSs during the time of the experiment, and since then, there have been updates to both Snort and Suricata with Snort 3.0 now being able to support multi-threading. However, at the time of this experiment, Snort 3.0 was still in its beta state, and we wanted to use a more stable version of 2.9.9.0.

3.2 The Experimental Data

There are two types of data that we use for this experiment—the *configurational* diversity utilises the rules and BIPAs—the *functional* diversity uses the pcap data along with rules and BIPAs. Using automated bash scripts, we saved snapshots of both rules and BIPAs for 5 months: from 20th May 2017 to 31st October 2017. We used the pulledpork tool to retrieve the rules from the corresponding web pages of Snort and Suricata [11]. The strategy was to save the two data sets of rules and BIPAs, for both Snort and Suricata, at a sufficiently high frequency to enable us to analyse the evolution of these tools. The snapshots were only taken when there was an update since the last saved snapshot. Therefore, for Snort, we have a total of 15,812 blacklisted files (28 less that if there had been an update every 15 minutes of the 165 days of the experiment). Contrary to Snort, where rules using the BIPAs have to use a path to a file having these IP addresses, Suricata uses BIPAs within its rule file. Similar to Snort, we used pulledpork and took snapshots of these rules files every 15 minutes. However, the rate of update of these Suricata rule/blacklisted files is rather on the daily basis. To do the BIPAs comparisons, we extracted these from the rule files for Suricata IDS. For the completeness of the experiment, we saved snapshots of all three rule types for Snort for the entire duration of the experiment. We used only the freely available ET ruleset for Suricata.

With the help of the University's IT team, we saved copies of the network traffic in the pcap format for retrospective analysis of attacks and incidents.

We saved the pcap data for the entire duration of the experiment. However, mainly due to logistical reasons of handling a large number of alerts data, we restricted the functional diversity analysis based on a sample of that data. To put things in perspective, for a week pcap data, we needed to use 7 sets of rules and BIPAs. This resulted in 49 data sets of alerts logs in the order of tens of GBs. Similarly, there needed to be 49 instances of Snort/Suricata to perform these experiments (7 pcap sets X 7 rules/BIPAs). The post-processing of the alerts and the management of a large number of sets of data was another reason that we restricted the functional diversity analysis to two weeks of pcap data. Essentially, we analysed the pcap traffic collected for 14 days (1st, 3rd, 8th, 9th, 10th, 15th, 17th, 23rd, 24th, 29th, 31st of August; 6th, 9th and 12th of September 2017) using the rules and BIPAs of the corresponding dates. The selection of these dates was not random. It was based on our observation that, for these dates, there had been updates in both the Snort subscribed and Suricata ET rules (hence enabling a fair comparison of these two IDSs). We believe that this is a reasonably large representative sample of the pcap data that covers around 1/3 of the time duration of our experiment. Besides, the selection of dates, for which there had been updates, is to make sure that we use the most up-to-date rules and BIPAs for the pcap data of those dates. This was also because, though we were analysing the traffic retrospectively, we made sure that it was as close to the real scenario as possible. The 14 days of pcap data were analysed in two separate experiments; each experiment analysed 7 days of pcap data with 7 days of rules/blacklisted IPs. For each experiment, we sent the pcap data, of each day, to a Snort and Suricata configuration as they were on each of these dates. So, we have 7 (days of pcap data) x 7 (configurations of IDSs) x 2 (Snort and Suricata) = 98 result sets for each experiment. In total, we have 98x2 = 196 results for both experiments. A point worth noting is that the pcap data of 10th and 15th of August was not saved for the entire 24 hours duration, that is why, in the results presented in section III, we observe a smaller number of alerts for these dates. Also, for the functional diversity experiment, we used the subscribed rules for Snort. This is after we observed in the configurational diversity experiment that the subscribed ruleset is a superset of the other two Snort rulesets (and get most frequently updated). We used, however, the Suricata ET ruleset in the functional diversity experiment as well.

4 CONFIGURATIONAL DIVERSITY ANALYSIS

In this section, we present the empirical study we carried out, analysing the configurational diversity of the Snort and Suricata IDSs. The study has two parts: first we show results of our finding about the differences and similarities in the BIPAs sets; we later present diversity analysis of the rulesets.

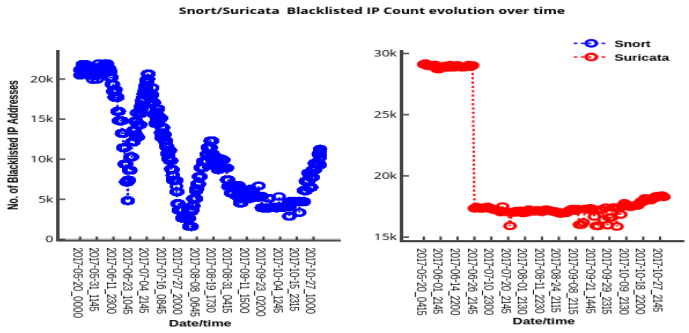


Fig. 3: Count of Blacklisted IPs in Snort and Suricata in our collection period.

4.1 DIVERSITY IN THE BIPAs OF SNORT AND SURICATA

4.1.1 Analysis of Individual IDSs

This section presents the analysis of the BIPAs for individual IDSs. The data we use for this analysis was collected, from May 20 to October 31, 2017, at a sampling rate of every 15 minutes. We observed, however, that the rate of change of the blacklisted IP files was, in some cases, less frequent than every 15 minutes for Snort, and was even further less for Suricata IDS (which tended to be every 24 hours). Figure 3 depicts the time-series data of the BIPAs for Snort and Suricata, in the left and right plots respectively. The y-axis shows the total count of the blacklisted IPs and the x-axis shows the data collection points. Comparing the two plots in Figure 3, we can clearly see the difference in the dynamics of the counts of the two sets. The left plot of Snort shows more fluctuations than those in the right plot of Suricata. It is worth noting, that around 21 June 2017, a large number of IP addresses were removed from the BIPAs set, for both Snort and Suricata. However, afterwards, the trends for the two sets remained very different, throughout the rest of the experiment. While the Snort BIPAs count still showed considerable fluctuations, the count of the Suricata BIPAs remained relatively smooth. Besides, we observe that there were two types of BIPAs in both the sets—those remained blacklisted for the entire duration of the experiment (or change their states only once, e.g., they are removed from the black-lists) are termed as “continuous”, and those that changed state twice or more (e.g., blacklisted, removed, blacklisted etc.) are called “discrete”. The general statistics of these BIPAs are given in Table 1. Here, the second column shows counts of the total number of files containing BIPAs for the whole experiment period; the third column shows the total number of distinct IP addresses; the fourth and fifth columns show the counts of the “continuous” and “discrete” IP addresses respectively. We have also considered the amount of time IP addresses remained black-listed during the experiment, since this may be an important feature in the quantification of

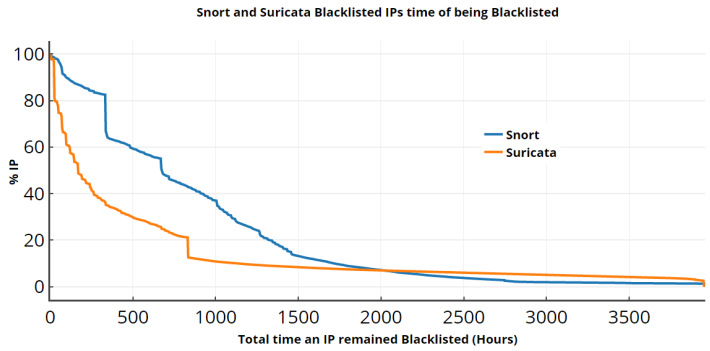


Fig. 4: Total time (Hours) an IP remained blacklisted

Table 1: General Statistics of BIPAs, BIP:=Blacklist IP, IPA:=IP Addresses

Source BIP	#Files	#IPA	#IP("continuous")	#IP("discrete")
Snort	15,812	46,701	5,383	41,318
Suricata	129	135,791	28,883	106,908

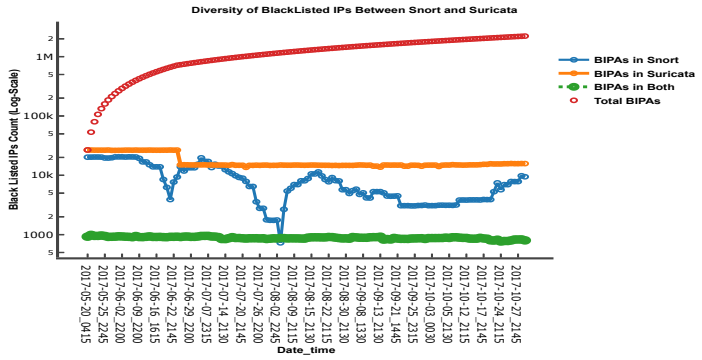


Fig. 5: Diversity in BIPAs as collected from Snort and Suricata sources.

diversity between the Snort and Suricata IDSs. Figure 4 depicts this analysis as the total time an IP remained blacklisted (x-axis) against the proportion of IP addresses (y-axis). We observe that on average, IP addresses stayed blacklisted longer in Snort than in Suricata.

4.1.2 Diversity Analysis of the BIPAs

This section presents a comparative analysis of the sets of BIPAs used by Snort and Suricata during the experiment. To this end, we use the sets of BIPAs that were collected at the same date/time points (to the nearest second). In total, out of 15,812 Snort files, and 129 Suricata files, 128 files had a common date/time overlap. We use these overlapping files for this analysis. Figure 5

Table 2: Statistics of the data points observed in Snort and Suricata overlapping periods,01:= Observed in Snort Only, 10:= Observed in Suricata only, 11:= Observed in both

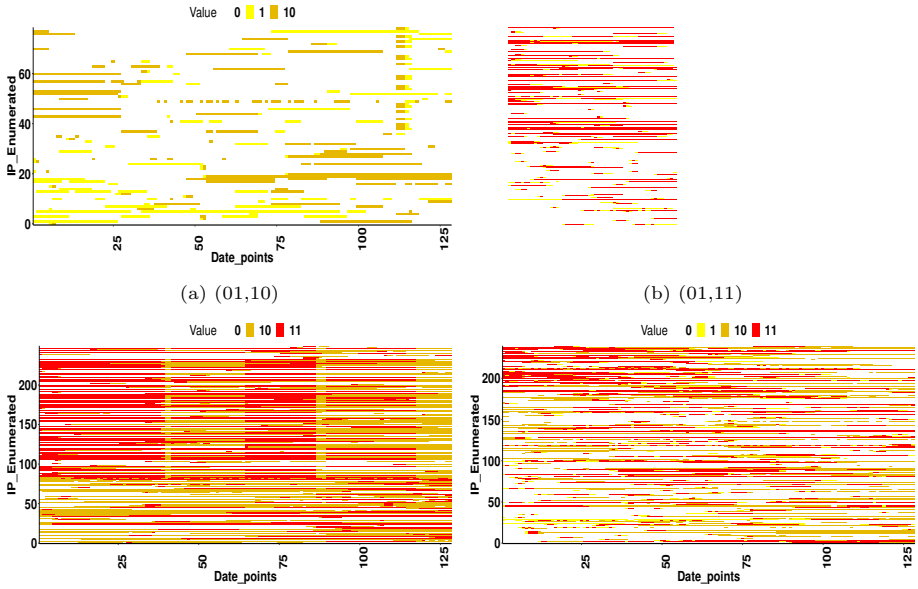
No. of BIPAs in 128 files of Snort		46,187
No. of BIPAs in 128 files of Suricata		135,308
No. of BIPAs in either Snort or Suricata		177,504
No. of BIPAs in both Snort and Suricata		3,991
No. of (IP/date pairs) observed in Snort and Suricata overlapping periods	(01)	1,129,180
	(10)	2,219,330
	(11)	113,152

Table 3: Break-down of different BIPAs States,01:= Observed in Snort Only, 10:= Observed in Suricata only, 11:= Observed in both

Single states	# IPs	Multiple states	# IPs	Observed first in:	# IPs
(01)	42,196	(01,10)	79	(01)	35
				(10)	44
(10)	131,317	(01,11)	2,834	(01)	1,257
				(11)	1,577
(11)	588	(10,11)	250	(01)	84
				(11)	166
		(01,10,11)	240	(01)	102
				(10)	82
				(11)	56

depicts this analysis as the overlapping date/time slots (in the x-axis) vs the counts of different categories of BIPAs (y-axis). We have three main categories of interest: BIPAs which were blacklisted in Snort only, BIPAs which were blacklisted in Suricata only, and BIPAs which were blacklisted in both Snort and Suricata. We have also shown the total BIPAs in both sets. Likewise, we observe that the overlap between the two BIPAs sets is relatively small and the total number of IPs that appear in blacklists of both Snort and Suricata is relatively constant for the duration of our experiments.

Table 2 shows the breakdown of the overlapping BIPAs. We have a total of 177,504 distinct BIPAs observed in either Snort or Suricata in the 128 overlapping files. Of these, 3,991 have been observed in both Snort and Suricata. We can think of each data-point in our data set consisting of an IP/date pair, and for each of these data points the value is either "observed in Snort-only" (labelled as 01), "observed in Suricata only" (labelled 10), or "observed in both Snort and Suricata at the same time" (labelled 11). The statistics for these data points are given in the last three rows of Table 2. It is worth noting, that the BIPAs which appeared either in Snort or Suricata can be of several types—those having state 01,10,11 or a combination of these states. The breakdown of these single and hybrid states is given in Table 3, giving a more detailed split of the 177,504 BIPAs observed in Snort and Suricata. The first two columns show the counts of those BIPAs appeared in the "single states" of either (01), (10), or (11). The third and fourth columns show the counts of BIPAs appeared in multiple states. For instance, the third row shows that there are 79 BIPAs, though appeared in both Snort and Suricata files, but at a different time. Similarly, some BIPAs appeared either in Snort or Suricata and later in both sets at the same time. These are labelled as (01,11), (10,11) and (01,10,11). We show a further breakdown of the BIPAs that appeared in



(a) (01,10) (b) (01,11)

(c) (10,11) (d) (01,10,11)

Fig. 6: Distribution of BIPAs on which appeared in multiple states: 01:=Black (Snort only),10:=Green (Suricata only),11:=Light Red (Both Snort&Suricata),0:=White (No Data)

Table 4: General Statistics of Different rulesets, R:=Rules, #F:=No. of Files, #R:=No. of Rules, #RNVC:=No. of rules with no version change, #RVC:=No. of rules with version change

Rules	#F	#R	#RNVC	#RVC
SnortReg	52	10,675	2,259	8,416
SnortSub	51	10,736	2,399	8,337
SnortCom	166	903	472	431
SuricataET	106	19,584	523	19,061

multiple states—for example, for the 79 BIPAs appeared in multiple states of (01,10), 35 appeared first in the Snort and 44 in the Suricata sets respectively. To visualise the dynamic behaviour of those BIPAs appeared in multiple states (i.e., those of columns three and four from Table 3), Figure 6 shows color maps of their time-varying observations in various sets. The x-axis shows the number of date/time points and the y-axis the enumeration of these BIPAs. These color maps shed light on the diversity of the Snort and Suricata BIPAs. Subplot 6a shows that several BIPAs appeared earlier in Suricata (Green points) than in the Snort BIPAs set (Black Points) and vice versa. The rest of the three subplots, 6b, 6c, and 6d, depict an interesting behaviour of many BIPAs—From time to time, it appears many IP addresses were removed from either Snort or Suricata before being reinstated again (we can see blocks of red (Snort and Suricata) becoming green (Suricata only), and then red again).

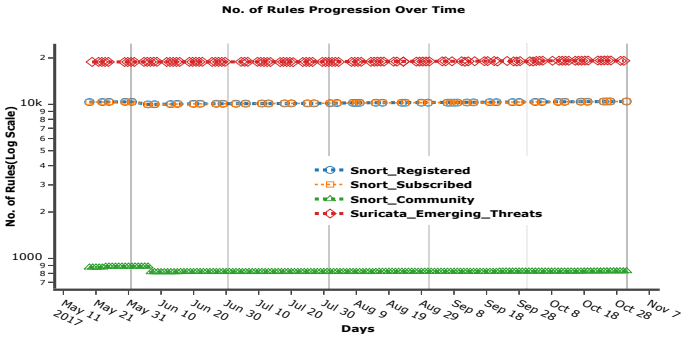


Fig. 7: Snort and Suricata rule counts over the duration of the experiment.

4.2 DIVERSITY IN RULES USED BY SNORT AND SURICATA

4.2.1 Overall Analysis

The signature-based rulesets is another source of configurational diversity in Snort and Suricata. We present a quantitative analysis of these rules in this section. This analysis uses rules data collected from 20 May to 31 October 2017. The types of rules we consider are Community, Registered, and Subscribed for the Snort IDS, and ET for the Suricata IDS. Similar to BIPAs, these rules were collected at a sampling rate of 15 minutes. However, the rate at which the rules were updated was much lower compared with BIPAs—mainly every 24 hours, but sometimes with lags of 5 days with no updates. Snort Community rules are an exception, where we noticed an update of 4 rules multiple times a day. We present the analysis by comparing the rulesets across all versions once every 24 hours. Table 4 shows the counts of different rulesets we use in the analysis. The two important features worth noting in this table are the differences in the number of rules and the rules with change in their VNs. We observe that the number of rules for Suricata is almost double that for Snort Registered and Snort Subscribed (which are very similar), and that the count of Snort Community is much smaller. There are rules with change in their VNs while their SID (Signature ID) remains the same—columns four and five of Table 4 give these counts. More than 80% of the Snort Registered and Subscribed rulesets, and 97% of Suricata ET ruleset reported version changes during the experiment. Figure 7 shows the dynamics of different rulesets. We notice that the total number of rules in each set remains relatively constant for the duration of the experiment.

4.2.2 Snort Rules Diversity Analysis

Next, we compare different Snort rulesets. To make this comparison tractable, we use the SID that, along with the Version Number (VN), can be considered

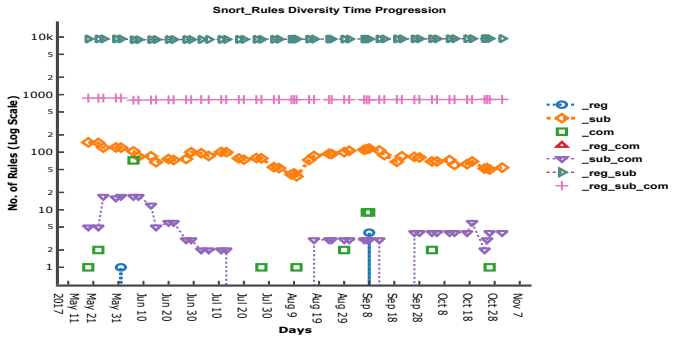


Fig. 8: Time Progression of Diversity in Snort Rules.

as a unique identifier for each rule. We use these identifiers (SID+Version No.) consistently across different rulesets (i.e., the same SID and same VN in Registered and Subscribed means that the rule is also the same). Figure 8 depicts dynamics of the rule counts for various Snort rule types. The y-axis shows, in a log scale, the counts of rules in different categories for each day of the experiment (x-axis). Here, "_reg" is the count of rules which are only in the Snort Registered set, "_reg_com" shows only those rules that are in the Registered and Community rulesets, etc. We notice that the majority of the rules are those that exist in both Registered and Subscribed rulesets (brown dots), followed by those that are common among all three rulesets (pink dots), and those that exist in the Subscribed ruleset only (orange dots).

We use binary states representation to label different Snort rulesets—01:Snort Registered, 10:Snort Subscribed, 11:Snort Community, 100:Snort Registered and Subscribed, 101: Snort Registered and Community, 110: Snort Subscribed and Community, and 111: All three Snort rulesets. The counts of SIDs and data points in different permutations of these sets are given in Table 5. A particular rule can be part of a single or multiple states, similar to what we discussed for BIPAs. Table 6 shows the counts of various SIDs (rules) being part of either single state, 2-states, 3-states, or 4-states. It is worth noting, however, that we have a maximum number of 27 permutations of different sets, but we only show those cases that have the non-zero count of SIDs (most of the other combinations have zero counts). Once a rule is established to be part of multiple states, then it is important to determine which state that rule was first observed in. That is why, Table 6 also shows, in columns 5, 6, 9, and 10, the breakdown of rule counts in states they were first observed in. For rules that were observed in a single state, the majority are in the '100' (The union of Registered and Subscribed rules). The other stand-out feature in Table 6 is that of the rules observed in multiple states. These rules had always been first observed in a state where there is the Subscribed ruleset. This is quite consistent with what we stated earlier about different Snort rulesets. To capture the time evolution of Snort registered and community rules, with

Table 5: Statistics of the data points observed in the Snort rulesets overlapping periods,01:= Snort Reg. Only, 10:= Snort Sub. only, 11:= Snort Com. only, 100:= Snort Reg. and Sub. only, 101:= Snort Reg. and Com. only, 110:= Snort Sub. and Com. only, 111: All three only

#SIDs Snort Reg		12,161
#SIDs Snort Sub		12,257
#SIDs Snort Com		959
#distinct SIDs in any		12,267
#Data points (SID/date pairs)	01	4
	10	4,255
	11	100
	100	469,390
	101	0
	110	210
	111	41,913

Table 6: Statistics of SIDs in different Snort rulesets, S:=States, OFI:=Observed-First-In, 01:= Snort Reg. Only, 10:= Snort Sub. only, 11:= Snort Com. only, 100:= Snort Reg. and Sub. only, 101:= Snort Reg. and Com. only, 110:= Snort Sub. and Com. only, 111: All three only

1-S	#SIDs	2-S	#SIDs	OFI	#SIDs	3-S	#SIDs	OFI	#SIDs
(01)	0	(01,100)	4	01	0	(10,110,111)	17	10	17
				100	4			110	0
								111	0
(10)	91	(10,100)	480	10	480	(11,100,111)	1	11	0
				100	0			100	1
								111	0
(11)	10	(10,110)	3	10	3	(11,110,111)	2	11	0
				110	0			110	2
								111	0
(100)	10,733	(11,111)	76	11	0	(100,110,111)	2	100	0
				111	76			110	2
								111	0
(101)	0	(100,111)	24	100	17	4-S	#SIDs	OFI:	#SIDs
				111	7				
(110)	2	(110,111)	7	110	7	(10,11,110,111)	1	10	1
				111	0			11	0
(111)	814							110	0
								111	

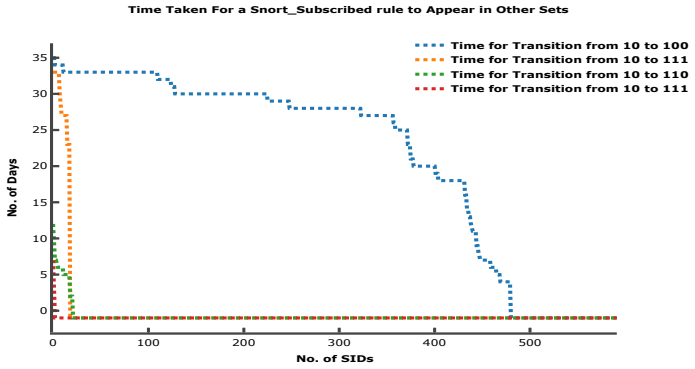


Fig. 9: The time lag for Subscribed rules to appear in the other Snort Rulesets.

respect to the Subscribed rules, we depict, in Figure 9, the time it takes for the Snort Subscribed rules to appear in other rulesets(i.e., the SIDs in the sets: (10,100), (10,110), (10,110,111) and (10,11,110,11) from Table 6). The figure confirms what is stated in the Snort website for these Subscribed rules: most of these become available to Registered users on average 30 days after they are available in the Subscribed ruleset.

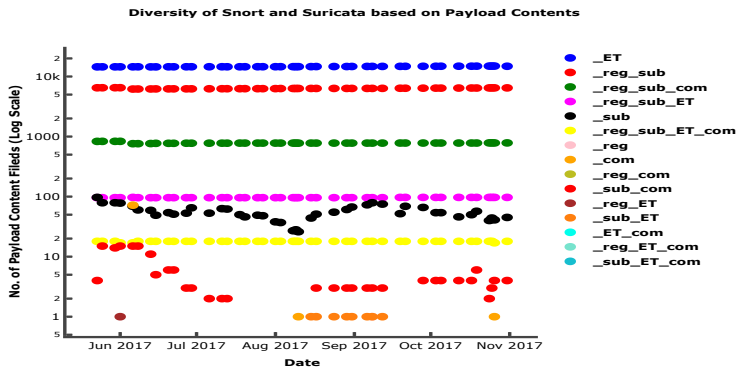


Fig. 10: Evolution of Configurational Diversity of the Snort and Suricata rulesets.

4.2.3 Configurational Diversity Analysis of Snort and Suricata Rules

This section quantifies the similarities and differences between Suricata ET and various types of Snort rules. To this end, we need to have fields in the rulesets which are comparable. However, contrary to the Snort rules analysis, where we use SIDs, Suricata ET rules do not share common SIDs with that of Snort rules. We, instead, use the "content" field in the rules, which are defined as regular expressions and contain the important "signature" information of the malicious payload of a packet. It is the "content" field that IDSs utilise to detect malicious payloads. However, the "content" field is limited to those rules that are responsible for known signatures in the payload of a TCP/IP traffic. Rules that check either BIPAs or ports, do not have the content fields. The analysis in this section considers rules that have the "content" filed (73.4% of the rules of Snort Registered and Subscribed have this field, 77.8% of Suricata ET and 97.7% of Snort Community rules have the "content" field).

Figure 10 shows the configurational diversity of Snort and Suricata rulesets based on the content field. Here, the x-axis shows the days and the y-axis the number of SIDs with content fields, in log scale. This figure depicts the evolution of the counts of the "content" fields not only for the individual rulesets but that of the intersection of various rulesets as well. The shorthand notation is the same as previous (e.g., "_ET" represents the SIDs with content fields observed only in the Suricata ET ruleset etc.) The largest overlap, among the intersection of rulesets, is that in the intersection set of Suricata ET, Snort Registered and Snort Subscribed rulesets (the magenta dotted line that hovers around the 100 marks in the y-axis). Table 7 gives the counts of SIDs with a content field in different rulesets. It also gives the number of data points observed in different sets. Note that, Table 7 gives additional rulesets and has used binary state representations, 1000...1111, to label them. Table 8 gives a further analysis of SIDs (with content field) that appeared in either single or multiple states. These two tables confirm that there is relatively little overlap

Table 7: Data points in Snort and Suricata rules with the contents field, S:= State, D-P:= Data Points, 01:= Snort Reg. Only, 10:= Snort Sub. only, 11:= Snort Com. only, 100:= Snort Reg. and Sub. only, 101:= Snort Reg. and Com. only, 110:= Snort Sub. and Com. only, 111:= All three only, 1000:= ET only, 1001:= ET and Reg only, 1010:= ET and Sub. only, 1011:= ET and Com. only, 1100:= ET and Reg. and Sub. only, 1101:= ET and Reg. and Com. only, 1110:= ET and Sub. and Com. only, 1111:= All four only

#SIDs Snort Reg with content field				7,840
#SIDs Snort Sub with content field				7,901
#SIDs Snort Com. with content field				883
#SIDs in Sur.ET with the contents field				15,239
#Distinct SIDs with content field in any of above				23,014
#(SID-content,date) pairs in Snort and Suricata.	S	#D-P	S	#D-P
	01	1	1000	644,159
	10	2,443	1001	0
	11	74	1010	8
	100	278,911	1011	0
	101	0	1100	4,236
	110	177	1101	0
	111	34,409	1110	0
			1111	748

Table 8: Statistics of SIDs with content field in Snort and Suricata rulesets, S:=States, OFI:= Observed-First-In, For other labels, see caption of Table 7

1-S	#SIDs	2-S	#SIDs	OFI	#SIDs	3-S	#SIDs	OFI	#SIDs
10	57	(01,100)	1	01	0	(10,110,111)	18	10	18
				100	1			110	0
100	6,548	(10,100)	315	10	314	(11,110,111)	2	111	0
				100	1			11	0
110	2	(10,110)	2	10	2	(1000,1010,1100)	1	110	2
				110	0			111	0
111	760	(11,111)	72	11	0			1000	1
				111	72			1010	0
1000	15,113	(100,111)	3	100	3			1100	0
				111	0				
1100	96	(110,111)	7	110	7				
1111	17			111	0				

between Suricata ET and Snort rules, as evident from the counts of SIDs in the intersection sets, 1001...1111.

5 FUNCTIONAL DIVERSITY ANALYSIS OF SNORT AND SURICATA

In this section, we analyse how the configurational diversity manifests itself in the alerting behaviour of Snort and Suricata IDSs. We analyse the alerting behaviour of each IDS by investigating the City, University of London network traffic, saved as pcap files. The analysis is not only static, but dynamic in time as well. It is worth noting, we use only the Subscribed ruleset for Snort in this analysis, as this being the superset of all other Snort rules.

5.1 Description of the Data used in Functional Diversity Analysis

Essentially, we analyse whether a ruleset, saved at a later date, generates more/fewer alerts for the traffic captured at an earlier date and vice versa. We evaluate this evolution, in the behaviour of IDSs, individually, as well as for the cross-platform comparison between Snort and Suricata. To this end, we analyse 14 days of pcap data collected in August and September 2017. There is no specific reason for selecting these 14 days, except that both the Snort

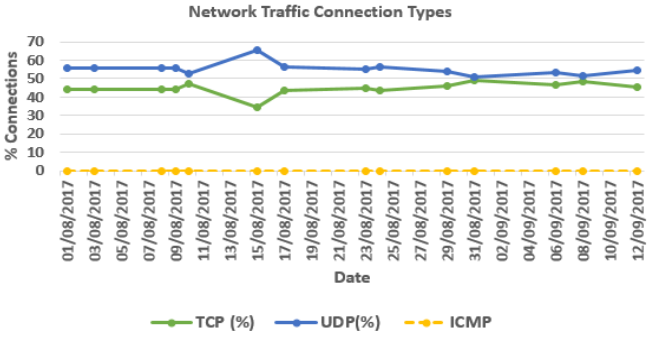


Fig. 11: Pcap Traffic Connections type breakdown.

and Suricata rulesets showed updates on these days. We divide the functional diversity analysis experiment into two sub-experiments; each analyzing 7 days of pcap data using the corresponding 7 days of rules. We do this for each IDS separately, which means we have 7×7 of alerts data per IDS, and hence a total of 7×7×2 data sets. Figure 11 depicts the proportions of various types of connections that we have in the pcap data. This shows that the majority of the traffic, on all days, is of UDP and TCP types (ICMP traffic to the DMZ network is blocked by the University's firewall). We use the subscribed ruleset, being the superset of other Snort rules, and the ET ruleset for Snort and Suricata IDs respectively. This also makes the analysis more scalable. As described previously, we selected 14 days of rulesets and BIPAs for Snort and Suricata on the same dates as those shown in Figure 11.

To analyse the effects of configurational changes (rules and BIPAs) on the alerting behaviour of IDSs, we need to know the day-to-day changes that took place in the Snort and Suricata rules/BIPAs during the 14 days of our experiment. The description of the evolution of the rules/BIPAs between adjacent days is shown in Table 9 and Table 10, for Snort and Suricata respectively. These tables show the comparison between two sets of rules/BIPAs next to each other in the order of days on which they were collected. For example, we compare the ruleset (and BIPAs) of 1 August with that of 3 August, and that of 3 August with 8 of August, and so on. The comparison of rules is given in columns 2-5 and that of BIPAs is in columns 6-8 of Table 9 and Table 10. The columns named 'i-1 & not i' and 'i & not i-1' show the number of rules that were found in the set, saved at an earlier day, and not in the current day, and vice versa respectively. Similarly, the columns named 'BIPAs in i-1 & not i' and 'BIPAs in i & not i-1' show the number of BIPAs found at an earlier day and not in the current day and vice versa respectively. We have also shown that how many rules have been changed between the adjacent days due only to VN change in the rules. This is shown by the column named 'changes of VN b/w i & i+1'. The third and fourth columns of both Tables 9, 10 show that there have been changes, however small, from day to day in both Snort and Suricata rulesets. Similarly, column 5 of both these tables shows that

Table 9: Summary of the Snort Subscribed Rules Evolution, i:=Current-Date,i-1=Previous-Date, VN:= Version Number

Dates	#Rules	Rules in i & not in i-1	Rules in i-1 & not in i	changes of VN b/w i & i+1	#BIPAs	BIPAs in i & not in i-1	BIPAs in i-1 & not in i
01/8	10,228		0	32	2,603		1,088
03/8	10,258	30	0	8	1,614	99	177
08/8	10,266	8	0	0	6,296	4,859	34
09/8	10,268	2	0	3	6,905	643	44
10/8	10,277	9	2	8	7,837	976	148
15/8	10,314	39	0	4	10,592	2,903	81
17/8	10,330	16			11,459	948	
23/8	10,340		0	14	9,783		1,099
24/8	10,345	5	3	17	10,053	829	968
29/8	10,354	12	6	17	7,425	3,596	70
31/8	10,360	12	2	16	6,577	918	1,940
06/9	10,368	10	436	448	5,644	2,873	1,386
08/9	10,374	442	4	18	5,958	1,072	126
12/9	10,394	24			5,049	1,035	

Table 10: Summary of the Suricata ET Rules Evolution, i:=Current-Date,i-1=Previous-Date, VN:= Version Number

Dates	#Rules	Rules in i & not in i-1	Rules in i-1 & not in i	changes of VN b/w i & i+1	#BIPAs	BIPAs in i & not-in i-1	BIPAs in i-1 & not in i
01/8	18,842		2	1,200	16,137		1,184
03/8	18,860	20	17	1,240	16,339	1,377	1,884
08/8	18,860	17	2	1,198	16,194	1,758	701
09/8	18,872	14	8	1,203	16,281	786	840
10/8	18,867	3	0	1,237	16,203	761	2,234
15/8	18,900	33	7	1,216	16,334	2,355	1,303
17/8	18,945	52			16,254	1,213	
23/8	18,940		13	1,195	16,185		1,364
24/8	18,939	12	7	1,228	16,122	1,309	3,462
29/8	18,945	13	3	1,095	16,068	3,426	712
31/8	18,973	31	4	1,229	16,051	695	2,919
06/9	19,025	56	25	1,219	16,348	3,202	1,761
08/9	19,031	31	137	1,125	16,198	1,598	2,303
12/9	18,912	18			15,744	1,822	

there have been changes in VNs of many rules from one day to the next. This change, however, is relatively large for Suricata rules (more than a thousand) as compared to that of Snort rules (less than 20 for most days). There have also been changes of BIPAs sets from day to day for both Snort and Suricata, as given in columns 6-8.

5.2 Evolution in the Alerting behaviour of Snort and Suricata

After having had the 14 days of pcap data analysed by both the Snort and Suricata IDSs, we investigate the evolution in the alerting behaviour of the individual IDS. To this end, we perform relative analysis by comparing alerts generated, for the pcap data on a particular date, by all the rulesets on 7 days of an experiment(s). We do this by finding the difference of alerts by the rules on all days from that on the current day—where the pcap and rules have the same date. Besides, we divide the alerts into two categories—those generated by BIPAs and other types of rules. Figure 12 and Figure 13 depict the evolution of alerting behaviours for Snort and Suricata respectively. Here, the x and y axes enumerate the dates of the rulesets and the pcap data, respectively. The cells show the normalized change of the alerts with respect to the current date—cells on the diagonal show the current dates. Cells to the left and right

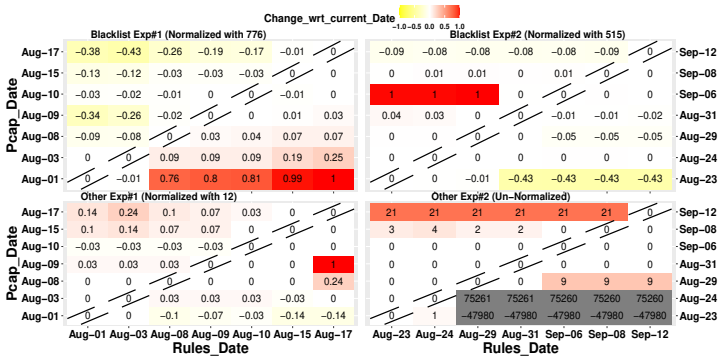


Fig. 12: Evolution of the Snort Alerting Behaviour

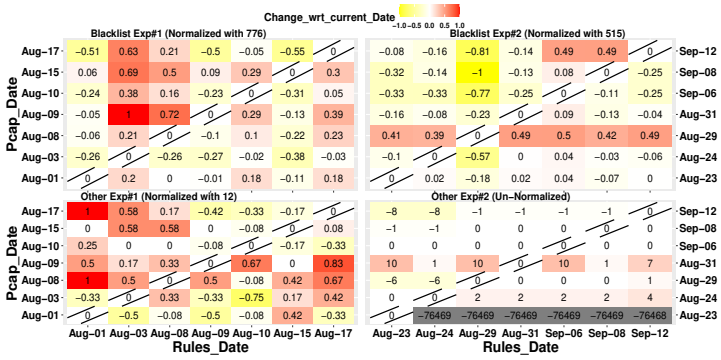


Fig. 13: Evolution of the Suricata Alerting Behaviour

of the diagonals show results of the analysis of a pcap data by the past and future rules, respectively. The results have been normalized by the maximum of the absolute values in the grid. For example, the top left Figure 12 has been normalized with the value of 145. This gives us uniform ranges of values between -1 and 1 (negative values mean that the number of alerts is smaller in a given resultset range than the reference for that row, which is given by the diagonal cell). Results of experiment-2 for the "other" rules have been left un-normalised due to outliers in the number of alerts in the 23rd and 24th of August (we observe this in the bottom right plot of both Figure 13 and 14). We give the absolute values for these graphs to make them easier to follow.

The top left/right heat maps of Figure 12 and 13 demonstrate that there are changes in the number of alerts, for the same pcap data, and by the BIPAs rulesets collected on different days. For example, in the top left of Figure 12, we see an increase (from yellow/white to red on the opposite sides of the diagonal) in the number of alerts by most of the blacklist rulesets and for every pcap data set. These changes in the alerting behaviour due to BIPAs are consistent with the changes in the Subscribed and ET BIPAs, as shown

in Table 9 and 10. Similarly, the evolution of the alerting behaviour due to all other rules can be seen from the bottom left/right plots of Figure 12 and Figure 13, for Snort and Suricata respectively. These maps underline the fact that there is an evolution of alerting behaviour, in both Snort and Suricata. However, the behavioural changes are more random and cannot be generalized. For instance, the bottom-right plot of Figure 12 shows a big increase in the difference of alerts from August 24 to 29, for the pcap data collected on 24 August. However, there is a substantial reduction in the difference of alerts, between the same dates, but for the pcap data of 23 August. This is supported by the differences in the rules between 24 and 29 of August, as shown in the third and fourth column of Table 9. However, the changes with respect to the current date are rather small on other days, as shown in the bottom left/right heat maps of Figure 12. For Suricata, we observe a similar decrease in the alerts by rules of 24 August and for the pcap data of 23 August shown in the bottom right heat map of Figure 13. This may be explained by a large number of version changes of rules between August 23 and 24 (fifth column of Table 10).

5.3 Diversity in Time between Snort and Suricata

To further investigate the evolution of alerting behaviour of the IDSs, we look more closely at the count of connections alerted, for each day of pcap data, and by each version of an IDS ruleset. We use the convention of 0 (no alert) or 1 (alert) for each connection. Since in each experiment, every connection was inspected by 7 different versions of Snort and Suricata rules, we use a concatenation of the labels for the 7 days to make the comparison easier. For instance, in the first experiment, the label 0000001 means that a connection was alerted only by a ruleset of 17 August (i.e., the 7th and last one in our data set of the first experiment). Similarly, 1000000 means a connection alerted only by the 1 August ruleset (i.e., the 1st ruleset version in our data set), etc. The same principle applies to the second experiment as well. Most of the connections in our experiment have never been alerted (i.e., they are 0000000), followed by connections that were alerted by all the rulesets (i.e., 1111111).

Of particular interest to us are the connections where we observe an order (non-alerts followed by alerts, e.g., 0000111; or alerts followed by non alerts 1100000). Those connections, where we have a non-alert (by an earlier ruleset) followed by an alert (by a latter ruleset), may be considered suspicious that were missed by an earlier ruleset. Those connections, where we have an alert (by an earlier ruleset) followed by a non-alert (by a latter ruleset), may be considered to be false positives and have been eradicated by the more recent rulesets. We have an exhaustive list of all the patterns we observed for Snort and Suricata, but due to space constraints, we cannot show those tables here. Instead, we show, in Figure 14, the frequency of more frequent alert combinations as Pareto plots. Note that, these alerts combinations are the ones caused only by the BIPAs rules, other rules, and in some cases a combination of these

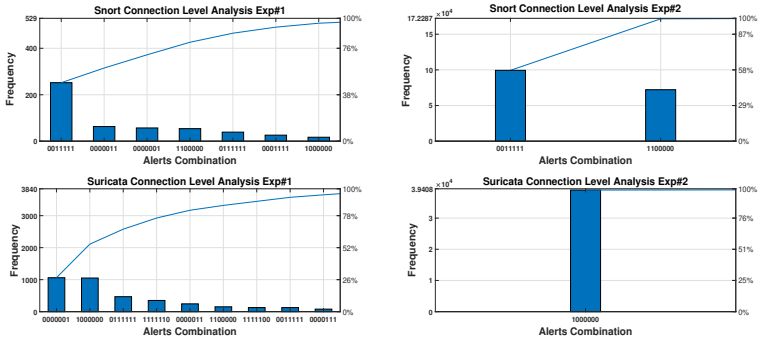


Fig. 14: Diversity in Time, Snort: Top row, Suricata: Lower row.

two. Figure 14 shows that the standout alert combination for Snort has been '0011111', in both experiments. Upon investigating, we notice that these alerts were predominantly due to BIPAs. This fact is also substantiated by the column 'BIPAs in i & not i-1' in Table 9; we see a big increase in the number of BIPAs and rules added after 3 and 24 of August. Similarly, for Suricata, the prominent rule combinations which appeared more frequent are '0000001', in the first experiment, and '1000000' in both the first and second experiments. Contrary to Snort, these were the 'other type' of rules that contributed to these combinations being dominant. This is also supported by the columns, 'i & not i-1' and 'i-1 & not i' in Table 10.

For connections, where we observe non-continuous patterns (e.g., 1010101), we investigate them further for clues about the non-existence of a particular rule on the days of no-alerts. We observe that there are several cases of Suricata alerts where the rule exists but was not triggered by a connection. There is no such observation for Snort, however. This is why, we contacted the developers of Suricata and Snort, and we got the advice to run Suricata in a mode using the flag '–runmode=single'. This mode works fine for a small pcap file and resulted in no non-determinism in the alerting behaviour. However, even with this mode, in our full-scale experiments, we observe several instances of alerts that were alerted in a non-continuous (non-deterministic) manner by the rulesets. However small these instances are, practitioners should be aware of it and should use the flag '–runmode=single' while running Suricata.

5.4 Functional Diversity Analysis between Snort and Suricata

In this section, we investigate the cross-platform functional diversity between Snort and Suricata. This analysis is the testing of the configurational diversity we discussed in section 4 of this paper. To this end, we compare the pcap connections that have been alerted by Snort and Suricata. Table 11 shows different statistics of this comparison for the 14 days of two experiments. This table substantiates the earlier observation of large configurational diversity

Table 11: Summary of alerted Connections Diversity in Snort and Suricata; Su: Suricata, Sn: Snort

Date	Sn&¬Su	Su&¬Sn	Sn&Su	Same-in Sn&Su	Different-in Sn&Su	Start-in Su	Start-in Sn	Start-in both
Aug-01	1,012,895	777,204	53	42	11	3	0	8
Aug-03	900,073	829,656	53	43	10	2	0	8
Aug-08	956,331	859,790	53	26	7	2	0	5
Aug-09	971,229	837,406	38	25	13	6	1	6
Aug-10	451,611	461,047	18	12	6	2	0	4
Aug-15	667,868	544,822	41	38	3	1	0	2
Aug-17	1,043,366	958,518	77	65	12	4	0	8
23-Aug	869,585	761,936	27	21	6	0	0	6
24-Aug	869,737	878,191	8	5	3	3	0	0
29-Aug	878,227	666,433	35	35	0	0	0	0
31-Aug	1,081,173	892,611	248	246	2	0	0	2
06-Sep	952,582	1,315,178	45	37	8	2	0	6
08-Sep	811,995	1,247,364	44	35	9	0	2	7
12-Sep	931,712	887,899	44	34	10	0	1	9

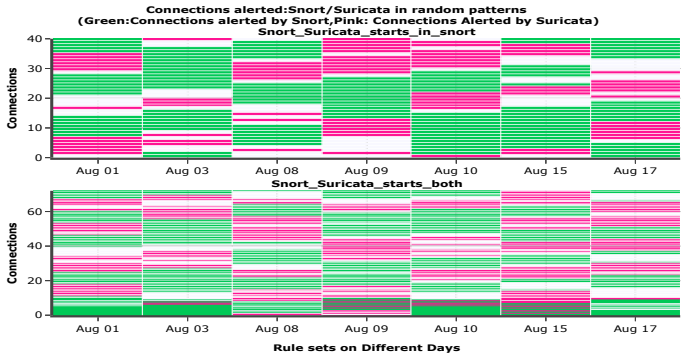
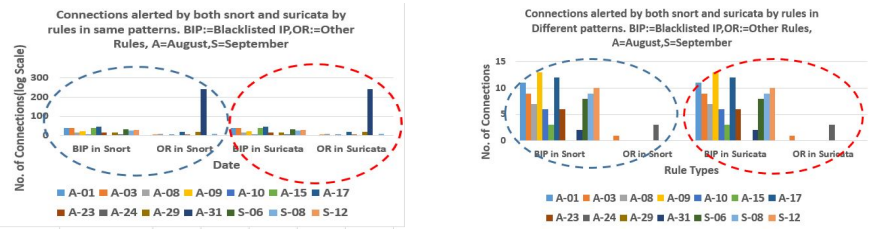


Fig. 15: Distribution of connection alerted by both in different patterns. Green: Snort; Pink: Suricata Patterns, experiment 1.

between the two IDSs resulting in large functional diversity. We notice, from columns 2 and 3 of Table 11, that the two IDSs are functionally very diverse and that there is minimal overlap of connections that have been alerted by both. Note that, the column "Sn&¬Su" denotes the number of connections alerted by Snort but not Suricata, and the "Su&¬Sn" column shows the number of connections alerted by Suricata but not Snort. There are only a handful of connections that have been alerted by both Snort and Suricata, as given in the "Sn&Su" column. There are two possible alerting behaviours for the connections jointly alerted by the two IDSs; they may have been alerted by the same rule patterns in both (e.g., 1100111; this is given by the column "Same-in Sn&Su"), or differently (e.g., 1111001 Snort, and 0011101 in Suricata; this is given by the column "Different-in Sn&Su"). For the latter case, the last three columns show which IDS alerted these connections ahead of the other or at the same time. Figure 15 shows the visualisation of the connection alerted in varying patterns of the Snort and Suricata rules in experiment 1 (It is similar for experiment 2). Here, we show two cases; connections that were alerted by Snort (green) ahead of Suricata (Pink), and those that were alerted by Snort and Suricata at the same time. Suricata has only one connection alerted ahead of Snort. In Figure 15, a connection is represented by two lines, green and pink,



(a) Alerts in same rules patterns

(b) Alerts in different rules patterns

Fig. 16: Analysis of Connections Alerted by Both Snort and Suricata

one for Snort and one for Suricata respectively. This figure also confirms, that even for the jointly alerted connections, the patterns of alerts are quite diverse, for the Snort and Suricata IDSs. Besides, we analyse the types of rules responsible for alerting the common connections. Figure 16 depicts the breakdown of these connections into BIPAs and other types of rules. It shows two plots, one each for the connections alerted by the same and different patterns of rules. We observe from Figure 16, that there are similarities, however small, in the alerting behaviours of Snort and Suricata. This is evident from the exactly same break-down of connections per rule type, in both the Snort (dotted blue circle in both plots) and Suricata (dotted red circle in both plots) IDSs.

6 Discussion and Limitations

The results are intriguing, and they show that there is a large amount of diversity in the rules and BIPAs of Snort and Suricata. This configurational diversity does manifest itself in the alerting behaviours of these IDSs. Whether this diversity is helpful or harmful for a given deployment depends on the context. The rules and blacklists alert for potentially harmful behaviour that has been observed somewhere in the world by users of these products. In a different deployment, the alerts from some of these rules may not cause harm. For example, a service or port for which a rule alerts may not exist in that environment. Hence, even if the alerts are for malicious traffic, it is likely that this attack will not cause any harm in the systems of that deployment. The data set we used in Section 5, real pcap traffic that the University's IT team gave us access to, is unfortunately not labelled, so we cannot do a conventional analysis of sensitivity and specificity of these IDSs and their diverse combinations. Secondly, though we were given pcap data of almost 5 months duration, we used only 14 days of this due to the large amount of alert data that we could not handle in our infrastructure. While we did observe the evolution of rules in terms of alerts for a pcap data, we cannot generalize the results for the complete set of data that we used in the experiment. We observe that though there are overlaps in the Snort and Suricata rulesets, there is huge diversity in their alerting behaviour. It shows that in their default

configurations, these two IDSs are tuned for a different set of malicious attacks. Also, we observed a non-deterministic behaviour in the Suricata IDS when it was used to analyse large traffic. For small traffic, though, the non-determinism can be avoided using a particular mode.

We shared the findings with the University's IT team, who found the results interesting. Currently, they use a smaller subset of Suricata ruleset for analysis. Interestingly, they mentioned that even if the alerts are for services that they do not run (hence would be harmless in their environment) they would like to know about them as it provides insight on security exposure for services that users may request in the future, and because they can use the alerts to check if they are precursors for attacks on other services that they value.

How can individual user organizations decide whether diversity is a suitable option for them, with their specific requirements and usage profiles? The cost is reasonably easy to assess: costs of the software products, the required middleware (if any), added complexity of management, hardware costs, run-time costs and possibly more complex diagnosis and more laborious alert sifting. The gains in improved security (from protection to attacks and exploits) are difficult to predict except empirically. This uncertainty will be compounded, for many user organizations, by the lack of trustworthy estimates of their baseline security. We note that, for some users, the evidence we have presented would already indicate that diversity to be a reasonable and relatively cheap precautionary choice, even without predictions of its effects. These are users who have serious concerns about security (e.g., high costs for interruptions of service or undetected exploits), and sufficient extra personnel to deal with a larger number of alerts.

7 IDSs deployment Strategies based on our Analysis

Based on the analysis of this paper, including the configurational and functional diversity analysis, we propose various IDS deployment strategies for the security architects. These are shown in Figure 17. It is worth noting, however, the security architects may want to consider other performance metrics, such as packet-processing speed, drop-rates etc, in their design strategies as well. The following are our recommendations:

- If there is a constraint of using single IDS, either Snort and Suricata, it is recommended to combine the rulesets and the BIPAs of both and use them in a single IDS. The rulesets and BIPAs can be used interchangeably by either of these IDSs. However, to avoid a high false alarm rate, both the rulesets should be properly tuned by an IT administrator based on the organization's security policy.
- The two IDSs can be deployed in parallel with the help of an adjudicating scheme. While this strategy may be more efficient in reducing the false alarm rate, this may, however, increase the overhead delays in the network.
- The two IDSs can be deployed in series, with an adjudicating system at the end of the series link. We cannot prescribe which IDS to be deployed

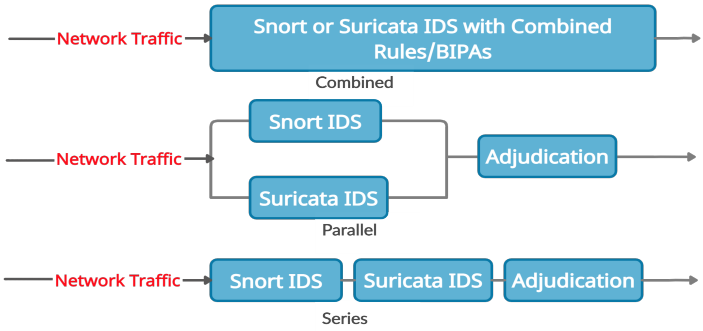


Fig. 17: IDS Deployment Strategies

first in this strategy. This is because our analysis is limited in terms of the 'actual' attack analysis and that which IDS perform better in terms of various detection metrics.

Further work would be needed to then analyse the IDS pipeline workloads to minimize overhead delays of traffic. We would expect that parallel architectures can be optimised to reduce overheads, depending on the adjudication scheme chosen. (i.e. whether we always need to wait for all IDSs to respond). In principle, any of these schemes can also be deployed either fully "on-premise" or via a "hybrid" approach (with some on-premise and some on cloud). The overheads and benefits would be difficult to predict except empirically.

8 Related Work

The security community is well aware of **diversity** as potentially valuable [15], [16]. Discussion papers argue the general desirability of diversity among network elements, like communication media, network protocols, operating systems etc. Research projects studied distributed systems using diverse off-the-shelf products for intrusion tolerance (e.g., the U.S. projects Cactus [17], HACQIT [18] and the EU MAFTIA project [19], but only sparse research exists on how to choose diverse defenses (some examples in [20], [21] [16]). The benefits of design diversity for fault-tolerant systems are discussed in [22].

A very extensive survey on the **evaluation of intrusion detection systems** is presented in [23]. This survey discusses many research works in the field. The main features analyzed in the survey are the workloads used to test the IDSs, the metrics utilised for the evaluation of the collected experimental data, and the used measurement methodology. The survey demonstrates that IDS evaluation is a key research topic and can help with guidelines on how to improve IDS technologies. A similar, yet more comprehensive, survey about IDSs is given in [24]. The paper details the current state-of-the-art in the design of IDSs for a diverse set of domains. The authors also discuss the metrics,

evaluation criteria and the data sets that have been used in recent research works on IDSs. In [25], the authors show the benefits of using a diverse set of IDSs in an empirical study. The authors have shown the efficacy of diversity by deploying the IDSs in different configurations such that to minimize false negatives/positives. To reduce the false-positive rate, in [26], the authors propose an off-the-shelf diversity architecture such that it masks false-positives.

Performance evaluation of Snort and Suricata has been studied in [27]. The authors have used different performance benchmarks, such as speed, drop-rates and detection accuracy to compare Snort and Suricata. Our work is different from this paper in several aspects. While we consider rulesets and BIPAs worth five months to quantify the configurational diversity, they just considered one set of the default rules for their experiments. Besides, while they used an open-source tool to generate synthetic data for the evaluation of detection accuracies, we use real-world traffic to check the differences in the alerting behaviours. More importantly, our analysis focuses on the diversity analysis as compared to the work in [27], which empahized more on the performance comparison of the two IDSs. In [28], Salah et al. analysed the effects of various operating systems on the performance of Snort IDS. In [29], Thongkanchorn et al. evaluated the detection accuracies of Snort, Suricata and Zeek IDSs while considering other performance metrics in parallel. The Snort and Suricata IDSs have been compared for their speeds, memory requirements and accuracy in [30]. The authors have demonstrated that Suricata can handle large volumes of traffic with similar accuracy. In [31], the authors showed that the speed and packet-loss performance of Suricata exceeded that of Snort with a reduced accuracy, however. In [32], the authors have used detection accuracy as the metric to compare Snort and Suricata in a cloud network. They have proposed the use of fuzzy logic in conjunction with these two IDS for improved performance. Pihelgas, in [33], compared the Snort, Suricata and Bro IDSs using various performance metrics of CPU usage, drop out packets and memory utilisation. It has been shown that Suricata performed better than the other two IDS for CPU usage and drop out packet, while it was Bro that outperformed others for memory usage. Ho et al. in [34] provided statistical analysis of the real-world traffic analysed by an IDS/IPS. They have shown that it is the IDS/IPS causing most of the false-positive and false-negatives in a real-world scenario. In [30], the authors have compared Snort and Suricata using real-world traffic. They have shown Suricata to be more CPU and Memory intensive, while it performed better when it came to the packet drop rate. A similar experimental evaluation of signature and anomaly based IDSs have been performed in [35]. The authors in [35] have compared Snort, Ourmon and Samhain for their characteristics of network degradation, CPU/Memory Usage and the number of alerts each of these IDSs generate. Snort has shown to be better from the CPU load and amount of alerts generation, while the other two IDSs used less memory and degraded the memory bandwidth slightly less than Snort. There has been research on how to automatically feed-in a NIDS with signatures and thus to avoid manual work in this regard. To this end authors in [36] have shown a hybrid anomaly and signature based IDSs using

the former to feed the new signatures to the latter. A comprehensive list of tools currently being used for attack detection and **signature generation** is given in [37]. **Machine learning-based** techniques have recently become quite popular for anomaly based NIDS. A review of different supervised and un-supervised learning-based intrusion detection algorithms is given in [38]. Similarly, to increase the detection capabilities of NIDSs, reinforcement learning has been gaining popularity in the research community, e.g., [39][40].

9 Conclusion

In this paper, we have presented an analysis of the configurational and functional diversities between the Snort and Suricata IDSs. Some data used in this paper is given in a git repository[1]. In the configurational diversity analysis, we have investigated the evolution of the BIPAs and rulesets that the Snort and Suricata IDSs use against the possible known attacks. Besides, we have presented the cross-platform diversity analysis between the corresponding BIPAs and rules configurations of these two IDSs. Data worth more than 5-months of duration has been used for this purpose. We have considered three different off-the-shelf default configurations of the Snort IDS and the ET configuration of the Suricata IDS. In the functional diversity analysis, we have investigated the manifestation of the configurational diversity in the alerting behaviours of the Snort and Suricata IDSs. We have used real network traffic collected at City, University of London in this analysis. We have undertaken this study intending to provide insight to security architects on how they can combine and layer these systems in a defence-in-depth deployment. The main conclusions from our analysis are:

- There is a significant amount of diversity in the BIPAs of Snort and Suricata, and this is maintained throughout our observation period. The amount of overlap between these BIPAs is relatively small. Depending on the adjudication mechanism that a system architect wishes to deploy, having access to a larger pool of BIPAs may be beneficial to increase protection against a larger pool of malicious sources. However, if a user observes a large number of false positives from these blacklists at a given time, then diversity can be a help to keep the false positive rate low (for example by only raising alarms if an IP appears in multiple blacklists) until the vendors "clean up" the blacklists;
- We observe the evolution of rule diversity for both Snort and Suricata butto generalize these results, we need to analyse pcap data of a longer duration.
- We observe a significant amount of diversity in the rules of Snort and Suricata. When analyzing the rules based on the "content" field, only 1% of the rules of Snort and Suricata return a match. This indicates that these systems would alert on potentially very diverse traffic. This is indeed

[1] https://github.com/Hasad/D3S_Data

confirmed from our experiment that we ran with real traffic from City, University of London. There was very little overlap in the alerting behaviour of these products.

We have underscored that these results are only prima facie evidence for the usefulness of diversity. What is important is to assess these products in real deployment on their capability to improve the security of a given system. The results presented here will, we hope, provide the security architects with evidence on the diversity that exists in the design of these products and whether this diversity remains as these products evolve.

As further work, we plan to investigate the diversity with IDSs and other defence-in-depth tools in real deployments, with labelled datasets, to assess the benefits as well as potential harm that diversity may bring due to the interplay between the risks from false negatives and false positives. Currently, we are investigating the adjudication mechanisms that can help balance the risks associated with these failures. Also, we plan to increase the size of the pcap data while analysing the evolution of rule diversity. Finally, since anomaly based IDSs and hybrid IDSs (i.e. anomaly+signature IDSs) are regular and current topics in cybersecurity, we also plan to investigate the diversity of these existing and emerging IDSs.

Acknowledgements 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.

References

[1] Brett van Niekerk and Pierre Jacobs. "ISACA JOURNAL". In: (2015).
[2] Snort. 2021. URL: `https://www.snort.org` (visited on 04/18/2021).
[3] Suricata. 2021. URL: `https://suricata-ids.org` (visited on 04/18/2021).
[4] Zeek. 2021. URL: `https://docs.zeek.org/en/lts/abou.html` (visited on 04/18/2021).
[5] Wazuh. 2021. URL: `https://wazuh.com/` (visited on 04/18/2021).
[6] Hafizul Asad and Ilir Gashi. "Diversity in Open Source Intrusion Detection Systems". In: *International Conference on Computer Safety, Reliability, and Security*. Springer. 2018, pp. 267–281.
[7] Al-Sakib Khan Pathan. *The state of the art in intrusion prevention and detection*. CRC press, 2014.
[8] Snort Rules. 2021. URL: `https://snort.org/documents/registered-vs-subscriber` (visited on 04/18/2021).
[9] Emerging Threat Rules. 2021. URL: `https://rules.emergingthreats.net/open/suricata/` (visited on 04/18/2021).
[10] Snort Blacklists. 2021. URL: `https://talosintelligence.com/documents/ip-blacklist` (visited on 04/18/2021).
[11] JJ Cummings and Michael Shirk. *Pulledpork*. `https://github.com/shirkdog/pulledpork`.

[12] Suricata Update Tool. 2021. URL: https://suricata-update.readthedocs.io/en/latest/ (visited on 04/18/2021).
[13] Snort logs. 2021. URL: http://manual-snort-org.s3-website-us-east-1.amazonaws.com/node21.html (visited on 04/18/2021).
[14] Suricata logs. 2021. URL: https://suricata.readthedocs.io/en/suricata-6.0.2/output/eve/eve-json-output.html (visited on 04/18/2021).
[15] Bev Littlewood and Lorenzo Strigini. "Redundancy and diversity in security". In: *European Symposium on Research in Computer Security*. Springer. 2004, pp. 423–438.
[16] Miguel Garcia et al. "Analysis of operating system diversity for intrusion tolerance". In: *Software: Practice and Experience* 44.6 (2014), pp. 735–770.
[17] Matti A Hiltunen et al. "Survivability through customization and adaptability: The cactus approach". In: *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*. Vol. 1. IEEE. 2000, pp. 294–307.
[18] James Reynolds et al. "The design and implementation of an intrusion tolerant system". In: *Proceedings International Conference on Dependable Systems and Networks*. IEEE. 2002, pp. 285–290.
[19] MAFTIA Research Project. 2003. URL: http://research.cs.ncl.ac.uk/cabernet/www.laas.research.ec.org/maftia/ (visited on 04/20/2021).
[20] William H Sanders et al. "Probabilistic validation of intrusion tolerance". In: *Supplemental Volume Int'l Conf. Dependable Systems and Networks (DSN-2002)*. 2002, pp. 78–79.
[21] Vishu Gupta et al. "Dependability and performance evaluation of intrusion-tolerant server architectures". In: *Latin-American Symposium on Dependable Computing*. Springer. 2003, pp. 81–101.
[22] Algirdas Avizienis and John PJ Kelly. "Fault tolerance by design diversity: Concepts and experiments". In: *Computer* 8 (1984), pp. 67–80.
[23] Aleksandar Milenkoski et al. "Evaluating computer intrusion detection systems: A survey of common practices". In: *ACM Computing Surveys (CSUR)* 48.1 (2015), p. 12.
[24] Lionel N Tidjon, Marc Frappier, and Amel Mammar. "Intrusion detection systems: A cross-domain overview". In: *IEEE Communications Surveys & Tutorials* 21.4 (2019), pp. 3639–3681.
[25] Areej Algaith et al. "Diversity with intrusion detection systems: An empirical study". In: *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*. IEEE. 2017, pp. 1–5.
[26] Frédéric Majorczyk, Éric Totel, and Ludovic Mé. "Experiments on cots diversity as an intrusion detection and tolerance mechanism". In: *Proceedings of the First Workshop on Recent Advances on Intrusion-Tolerant Systems (WRAITS 2007)*. 2007.
[27] Qinwen Hu, Se-Young Yu, and Muhammad Rizwan Asghar. "Analysing performance issues of open-source intrusion detection systems in high-

speed networks". In: *Journal of Information Security and Applications* 51 (2020), p. 102426.

[28] Khaled Salah and A Kahtani. "Performance evaluation comparison of Snort NIDS under Linux and Windows Server". In: *Journal of Network and Computer Applications* 33.1 (2010), pp. 6–15.

[29] Kittikhun Thongkanchorn, Sudsanguan Ngamsuriyaroj, and Vasaka Visoottiviseth. "Evaluation studies of three intrusion detection systems under various attacks and rule sets". In: *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. IEEE. 2013, pp. 1–4.

[30] Eugene Albin and Neil 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.

[31] Syed Ali Raza Shah and Biju Issac. "Performance comparison of intrusion detection systems and application of machine learning to Snort system". In: *Future Generation Computer Systems* 80 (2018), pp. 157–170.

[32] Saeed M Alqahtani and Robert John. "A comparative study of different fuzzy classifiers for cloud intrusion detection systems' alerts". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2016, pp. 1–9.

[33] Mauno Pihelgas. "A comparative analysis of open-source intrusion detection systems". In: *Tallinn: Tallinn University of Technology & University of Tartu* (2012).

[34] Cheng-Yuan Ho et al. "Statistical analysis of false positives and false negatives from real traffic with intrusion detection/prevention systems". In: *IEEE Communications Magazine* 50.3 (2012), pp. 146–154.

[35] Xinli Wang et al. "Administrative evaluation of intrusion detection system". In: *Proceedings of the 2nd annual conference on Research in information technology*. 2013, pp. 47–52.

[36] Pedro Garcia-Teodoro et al. "Automatic generation of HTTP intrusion signatures by selective identification of anomalies". In: *Computers & Security* 55 (2015), pp. 159–174.

[37] Sanmeet Kaur and Maninder Singh. "Automatic attack signature generation systems: A review". In: *IEEE Security & Privacy* 11.6 (2013), pp. 54–61.

[38] Zeeshan Ahmad et al. "Network intrusion detection system: A systematic study of machine learning and deep learning approaches". In: *Transactions on Emerging Telecommunications Technologies* 32.1 (2021), e4150.

[39] Mohammad Alauthman et al. "An efficient reinforcement learning-based Botnet detection approach". In: *Journal of Network and Computer Applications* 150 (2020), p. 102479.

[40] Manuel Lopez-Martin, Belen Carro, and Antonio Sanchez-Esguevillas. "Application of deep reinforcement learning to intrusion detection for supervised problems". In: *Expert Systems with Applications* 141 (2020), p. 112963.