Comparative Analysis of the Performance of Network Intrusion Detection Systems: Snort, Suricata and Bro Intrusion Detection Systems in Perspective

Godwin Kudjo Bada
E.P. College of Education,
Amedzofe, Ghana
Department of Mathematics and ICT

Williams Kwame Nabare
Valley View University, Ghana
Department of Information Technology

Daniel Kwame Kwansah Quansah
University of Cape Coast, Ghana
Department of Medical Education and IT

ABSTRACT

There exists a number of intrusion detection systems particularly those that are open-source. These intrusion detection systems have their strengths and weaknesses when it comes to intrusion detection. This work compared the performance of open-source intrusion detection systems namely Snort, Suricata and Bro. The comparative analysis of these intrusion detection systems was carried out to present an independent view of their performance regarding intrusion detection. It took into consideration their effectiveness in detecting Denial of Service, probe, scan, User-to-Local and User-to-Root attacks and also detection accuracy in terms of false positive, false negative and true positive alarms. All three IDS were installed on virtual machines with the same specification with a network switch linking them to a target server in a virtual environment using maximum Ethernet speed of 5Gigabits per second (Gbps). False positive, false negative and true positive alarm rates of Snort, Suricata and Bro IDSs have also been determined in this work through the injection of normal and malicious attacks such as DoS, probe, scan and user-to-root. Transmission Control Protocol, User Datagram Protocol and Internet Control Message Protocol were the normal traffic used.

Keywords

intrusion, packets, false-alarms, vulnerabilities, malicious, denial-of-service

1. INTRODUCTION

In the networks of most organisations, intrusion often occurs. Intrusion into computer networks can emanate from hackers and crackers; these people can sometimes use computer viruses, spam, and denial of service to inhibit the smooth operation of networks. According to [11], all these acts of intrusion can either come from inside or outside the organization's network. The use of Intrusion Detection Systems (IDSs) is one of the complex techniques to ensure security of information systems and networks. Many open-source IDSs exist including Snort, Suricata and Bro. The dilemma however, is how to determine the one that is the most effective in detecting intrusions. It is proper to conduct analysis of these three IDSs to present an independent view of their effectiveness in detecting various threats.

2. PROBLEM STATEMENT

There are various open-source IDSs available which are used for detecting threats in computer networks and information systems. The challenge faced by users of these open-source IDSs is the existence of an independent view as regards how these IDSs perform.

3. RESEARCH QUESTIONS

a. How do open-source IDSs such as Snort, Suricata and Bro respond to specific network attacks?

b. What are the levels of false and negative alarms generated by open-source IDSs such Snort, Suricata and Bro for normal and malicious network traffic?

4. LITERATURE REVIEW

4.1 Snort as Signature-based IDS

Snort engine by its design allows for a single rule to be applied to different network protocols. The Snort software carries out analysis of protocol and content matching. It is also commonly used to actively block or passively detect a variety of attacks and effectively conducts probes such as buffer overflows, stealth port scans, web application attacks, and operating system fingerprinting attempts, and a couple of other features. The Snort software is mostly used for intrusion prevention purposes, thus, by handling attacks as they happen [5].

Some notable features of the Snort software include working on any operating system, capability of examining protocol, examining the condition of packets, and reassembling packets. Snort is single-threaded; meaning it can only use one CPU core at a time. In addition, Snort provides graphical user interface showing various components for analyzing results. Snort rules can be specified easily by normal users, but powerful enough to detect a wide variety of hostile or those that are merely suspicious network traffic. There are three basic actions that Snort can trigger in case a packet matches a specified rule pattern and these include log, alert, or pass.

Snort IDS requires an outer data packet-sniffing library. Stacks in network operating systems are commonly in charge of re-gathering packets and giving application visibility to the bundle's payload. A packet-sniffer will expect access to crude packets not altered to distinguish and have the capacity to recognize normal behaviors. Snort is most regularly deployed utilizing a Libpcap as its packet-sniffing library. Libpcap widely works across various operating systems making Snort IDS very flexible.

CPU usage is one important factor that is used to determine the performance of IDS/IPS. Packet dropping occurs due to the fact that packets outnumber the capacity and resources of the processor. Also, the number of available rules for an IDS increases requirements for processing tasks [1].

4.2 Suricata as Signature-based IDS

Suricata as signature-based IDS is a free and open source, fast and robust network threat detection engine. It is capable of real-time intrusion detection, network security monitoring and offline packet capture (pcap) processing. Suricata inspects the network traffic using powerful and extensive rules and signature language, and can detect complex threats.

Because Suricata is multi-threaded, it can run many threads by taking advantage of all the CPU cores available on a machine. Suricata does not only log packets but can also capture and log Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates, HTTP requests, and DNS requests. Multi-thread as an advance feature of the Suricata detection engine is necessary as network bandwidth increases (Nielsen, 2010). Suricata in its true capability when installed should be able to handle traffic ranging from 100 to 200 Mbps even before getting to the limit of one processor and could drop packets to compensate. According to Open Information Security Foundation (OISF), Suricata from the beginning was designed to make use of multi-processors. Suricata intrusion-detection system contains multiple threads within the same detection engine which enables it to split processing tasks and it is able to harness processing activities among these threads all for the same detection engine [9].

[7] postulated that the processing power of CPU would double every eighteen months in relation to single core architectures. It is clear that multi-threaded processing can take advantage of that prediction.

4.3 Bro as a Hybrid IDS

Bro as IDS has analysis engine which converts traffic captured into a series of events. The events being described could be a user logon to FTP, a connection to a website or any other activity of users online**. As** open-source IDS, it supports Linux, FreeBSD, and MacOS platforms. It is network-based intrusion detection system that monitors network traffic for suspicious activity. Bro IDS detects intrusions by initially parsing network traffic to extract its application-level structure and eventually executes event- oriented analyzers that compare the network activity with patterns that are considered as threats. Upon detecting any activity of interest, Bro's detection engine can be instructed to either generate a log entry, alert the operator in real-time, execute an operating system command which can lead to termination of a connection or block a malicious host activity. Providing detailed log files can be particularly useful for forensics.

Bro IDS is capable of targeting high-speed in Gigabits per second (Gbps) and high-volume of traffic for intrusion detection. This IDS uses the idea of packet-filtering hence; it can achieve efficiency while running on commercially available PC hardware, meaning that it can serve as a cost-effective means of monitoring a site's Internet connection. Bro IDS contains several analyzers comprising protocol decoders for a variety of network protocols and a signature matching engine as well that communicates through events within a network. It has its own scripting language which enables its users to define event handlers in their environment-specific policy [2].

This IDS relies on script interpreter instead of separate processing engines, processors, and decoders. It makes use of data packets gathered from network using standard packet acquisition libraries, for instance libpcap. In addition to making handling of throughputs that are high by its engine, it does offer clustering choices for considerably high-throughput environments enabling it to deal with high-volume data packets. Bro IDS also has capabilities of dealing with multi-threading tasks though it does not form an integral part of it [6].

According to [10], Bro at a later date brought file extraction and correlating characteristics just like those used in Suricata. With this technique, there is file hash extraction support and correlation which allows for automated file extraction and alarms using custom file hashes or through publicly available hash data sources. There is the need to focus on improving per-core processing efficiency which will consequently lead to enhancement of efficiency in clustered or distributed networks to match the regular ascendancy in network throughput [3].

Using Bro IDS with its scripting options provide considerable ease of use in handling threats, logging, and even after-detection tasks. The choice of Snort and Suricata with its in-line technique gives alternatives to control traffic by blocking regarding those that correlate with signature rules. Not only does Bro log and block signatures that do not match provides but it also provides alternatives for sending email messages or automatically truncate network connections. By virtue of Bro's scripting policy, it provides specific alternatives to cater for rate-limit flows that will correlate configured policies and also the operating system and application response [4].

4.4 Types of Alarms Generated by Intrusion Detection Systems

There are four types of alarms that are produced by IDSs which includes *true-positive, true-negative, false-positive,* and *false-negative.* To start with, the *true positive* is when the detection engine of the intrusion detection system generates an alert based on the correct identification of a potential threat. The *true negative* is when a detection engine does not generate an alert for normal traffic though there could be a threat and this occurs during a benign network traffic flow. In the case of *false positive*, a detection engine generates an alert for an event that is not malicious and thus, giving false alarm. The fourth condition and perhaps the most dangerous of all is the *false negative*; a situation where a detection engine does not alert or give any alarm at all on malicious traffic, consequently allowing it to enter the network without notice.

It therefore means that it is possible for IDSs to identify a normal activity as a malicious one, which is termed a *false positive (FP)*, or even malicious traffic as normal, resulting in a *false negative (FN)*. Many security problems can be caused due to false positives and false negatives; for instance, false negatives are able to generate unauthorized or abnormal activities on the Internet or in computer systems/information systems. Also, when false positives are many, they may easily conceal real attacks and thus overwhelm the IDSs which are responsible for the security of computer systems. It should be noted that when real attacks occur, *true positives* (real alerts) are deeply buried within false positives making it easy for the security operator to miss them [12].

5. METHODOLOGY

The research method used for this work is the *experimental research method*. In this research, experiments were conducted to test and compare the performance and accuracy of three open-source intrusion detection systems namely; Suricata, Snort and Bro in a virtual network environment with a maximum Ethernet speed of 5Gbps. The experiments were carried out to evaluate the performance of these IDSs by comparing how effective the IDSs were in detecting attacks and the rates of false and true alarms generated. A determination was made regarding the rate of false positive, false negative and true positive alarms by the IDS engines under consideration. The accuracy of the IDSs was measured by capturing and analyzing network traffic available in all the IDSs under consideration in controlled tests and comparing the alarms generated. The request and responses from the virtual machine were captured using Metasploit (Kali Linux) as a tool and analyses performed using the detection engines of the three IDSs.

Both the malicious and normal or legitimate traffic generated were combined and used as input for the three IDSs namely Snort, Suricata, and Bro. The network for the experiment was setup using Oracle Virtual Box. In all, five virtual machines (VMs) were used for the purpose of this experiment. The specific experiment at a time determines whether normal or malicious network packets be produced at different network speeds with the network traffic generator software. All five Virtual Machines were connected via a virtual switch capable of carrying 5Gbps of internet data through Ethernet links. The network for the experiment consisted of Virtual Machines that are of high-performance running the Snort, Suricata, and Bro IDSs. The Snort version 2.9.11.1, Suricata version 4.0 and Bro version 2.5 were used for the experiments in this work. Metasploit (Kali Linux) version 2.0 was also used for generating malicious traffic.

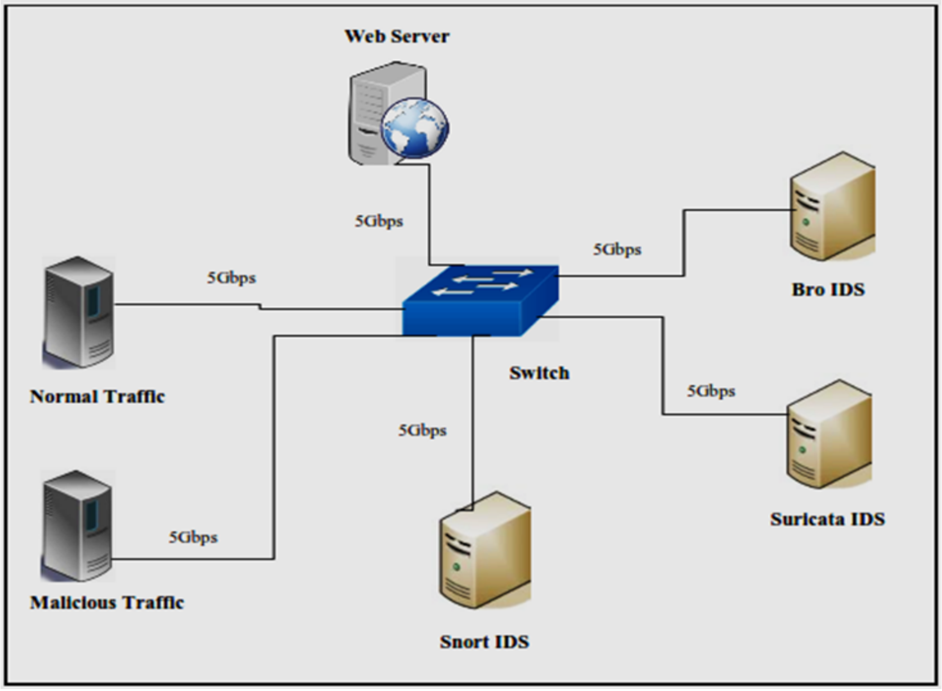

Fig 1: Virtual Network Setup for Experiments

6. RESULTS

6.1 Effectiveness of Snort, Suricata and Bro in Detecting Network Attacks

Table 1 shows the percentage of average true positive alarms indicated by the three IDSs within a period of 18hours when the various attacks were injected. The average result for Snort for the five attacks were DoS (97.4%), probe (95.0%), U2L (99.1%), scan (99.0%) and U2R (98.9%).

For Suricata IDS, the results were DoS (94.5%), probe (96.3%), U2L (98.7%), scan (99.3%) and U2R (98.2%).

The results of average true positive alarms triggered by Bro were DoS (94.8%), probe (93.00%), U2L (97.9%), scan (98.5%) and U2R (97.9%).

Table 1: Average true positive rate of the IDSs

Attack category	Snort IDS TPR (%)	Suricata IDS TPR (%)	Bro IDS TPR (%)
DoS	97.4	94.5	94.8
Probe	95.0	96.3	93.0
U2L	99.1	98.7	97.9
Scan	99.0	99.3	98.5
U2R	98.9	98.2	97.9
Average	**97.88**	**97.40**	**96.40**

The results in Table 1 are graphically represented in Fig 2 below;

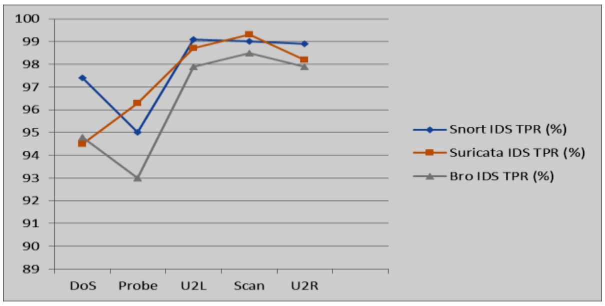

Fig 2: Average TPR of the IDSs

6.2 Average False Alarm Rates for Malicious Traffic Detection by the IDSs

For the period of 18hours made up of 6hours for each block, Snort IDS triggered an average false positive alarm of 3.7% for DoS attacks as compared to Suricata's 9.3% and Bro's 6.3% and this is an indication that Snort IDS is capable of detecting DoS attacks than Suricata and Bro.

On the average, Snort triggered 9.0% for probe attacks whereas Suricata and Bro triggered 10.53% and 9.9% false positives respectively.

The false positive rate for Snort in terms of U2L attack was 23.0%, Suricata triggered 25.0% and Bro recorded 24.3%.

For scan attack, the rate of false positives indicated by Snort was 7.0% that of Suricata was 11.3%, as Bro recorded 14.7% false positives.

Finally, it was realized that for U2R attack, Snort IDS triggered 12.0% false positive alarms, Suricata triggered 16.7% as Bro recorded 12.3% false positives.

For all the attacks, Snort IDS triggered an average of 54.7% false positives, Suricata on average triggered 72.83% false positives and Bro triggered 67.5%. Snort in this case triggered the fewest false positive alarms among all three IDSs. The average false positive alarm rates are as shown in Table 2 below:

Table 2: False positive rates of the IDSs

Attack category	Snort FPR (%)	Suricata FPR (%)	Bro FPR (%)
DoS	3.7	9.3	6.3
Probe	9	10.53	9.9
U2L	23	25	24.3
Scan	7	11.3	14.7
U2R	12	16.7	12.3
Average	**54.7**	**72.83**	**67.5**

The false positive rates for malicious traffic are graphically represented in Fig 3 below:

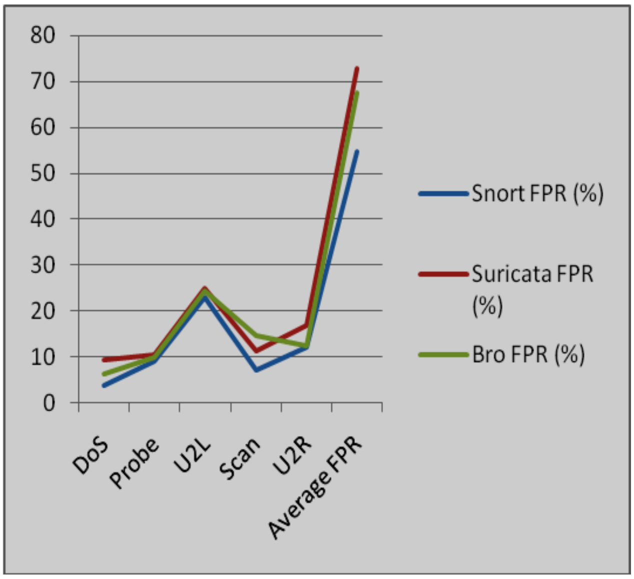

Fig 3: FPR for malicious traffic

The results of false negatives are shown in Table 3. In terms of false negatives, Snort had 1.7%, Suricata 2.3% and Bro had 1.3% for DoS attacks which put Bro IDS above Snort and Suricata IDSs.

Snort triggered no false negative alarm for probe attack, Suricata triggered 2.4% and Bro had 1.27%. Here too, Snort can be said to be above the other IDSs in intrusion detection.

For U2L attack, Snort had 1.0% false negative rate whereas Suricata and Bro triggered no false negative alarm.

In the case of false negatives, Snort had 4.0% on the average; Suricata had 3.3% whilst Bro had 4.6% for scan attack.

The percentage of false negatives indicated by Snort for U2R attack was 0.4%, Suricata 3.3% and Bro had 1.8% false negatives.

It can be observed that Snort triggered average false negative alarms of 7.1%; Suricata triggered 11.3% false negative alarms, whereas Bro triggered 8.97% false negative alarms. Since the fewest false negative alarms produced determine the effectiveness of any IDS, it can be concluded that Snort IDS is superior to Suricata and Bro having all IDSs used their default configuration rules.

Table 3: False negative rates for malicious traffic

Attack category	Snort FNR (%)	Suricata FNR (%)	Bro FNR (%)
DoS	1.7	2.3	1.3
Probe	0	2.4	1.27
U2L	1	0	0
Scan	4	3.3	4.6

U2R	0.4	3.3	1.8
Average FNR	**7.1**	**11.3**	**8.97**

The results in table 3 are graphically shown in the figure 4 below;

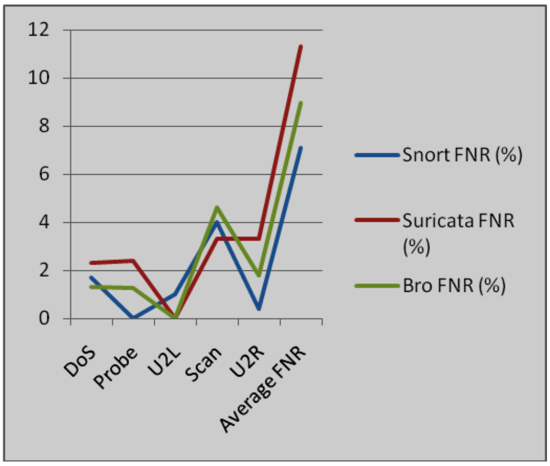

Fig 4: FNR for malicious traffic

6.3 Normal Traffic Classification by the IDSs

The results of passing normal traffic such as TCP, UDP and ICMP through the three IDSs are analyzed in this discussion. Snort recorded false positive rate of 9% for TCP packets but no false negative and true positive rate for the same packet type.

In processing the UDP packets, Snort triggered 12% false positive rate alarms, 1% false negative and no alarm for true positive.

For ICMP packets, Snort had 5% false positives. With regard to false negative alarms, Snort had none thus, 0% but triggered 1% true positive.

The results for Snort IDS in the classification of normal traffic are shown in Table 4;

Table 4: Normal traffic classification by Snort IDS

Normal Traffic	Snort IDS		
	FPR (%)	FNR (%)	TPR (%)
TCP	9	0	0
UDP	12	1	0
ICMP	5	0	1

The results in table 4 are shown in the graph in Figure 5 below;

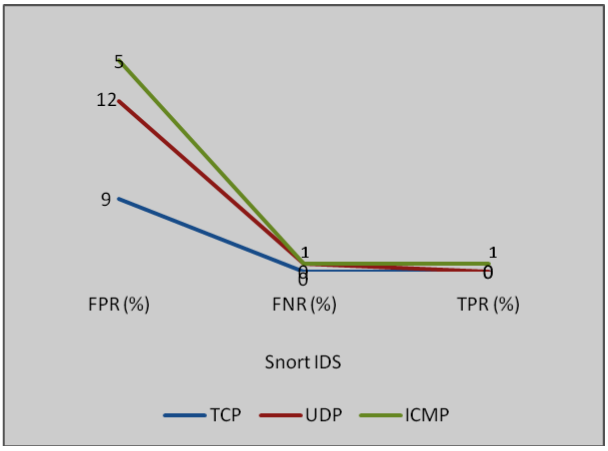

Fig 5: Normal traffic classification by Snort IDS

The results for Suricata in relation to normal traffic classification are as indicated in Table 5 below;

As regards TCP packets, Suricata IDS had false positive rate of 19%. For false negatives, it recorded 4% but had no true positive alarm.

For UDP packets, it triggered 27% false positives and 8% false negatives but had no true positives.

The record for ICMP showed that Suricata triggered 34% false positives and 23% false negatives whereas the true positive was 2%.

Table 5: Normal traffic classification by Suricata IDS

Normal Traffic	Suricata IDS		
	FPR (%)	FNR (%)	TPR (%)
TCP	19	4	0
UDP	27	8	0
ICMP	34	23	2

Graphically, the results in table 5 are represented in figure 6 below;

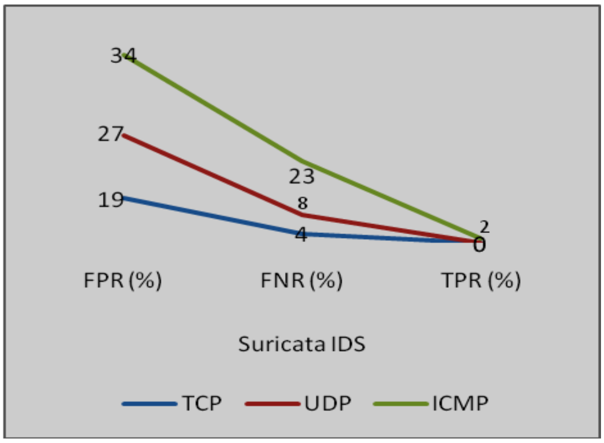

Fig 6: Normal traffic classification by Suricata IDS

The normal traffic classification results for Bro IDS are shown in the Table 6 below;

Table 6: Normal traffic classification by Bro IDS

Normal Traffic	Bro IDS		
	FPR (%)	FNR (%)	TPR (%)
TCP	11	1	2
UDP	15	0	3
ICMP	4	2	0

It can be seen that for TCP packets, Bro IDS recorded 11% false positive, 1% false negative and 2% true positives.

In relation to UDP packets, Bro IDS again triggered 15% false positive alarms but had none for false negative. It however recorded 3% true positives for the UDP packet.

For the ICMP packets, Bro IDS triggered 4% false positive alarms and 2% false negatives but had no alarm indicated as true positive.

These results are also shown in figure 7 below;

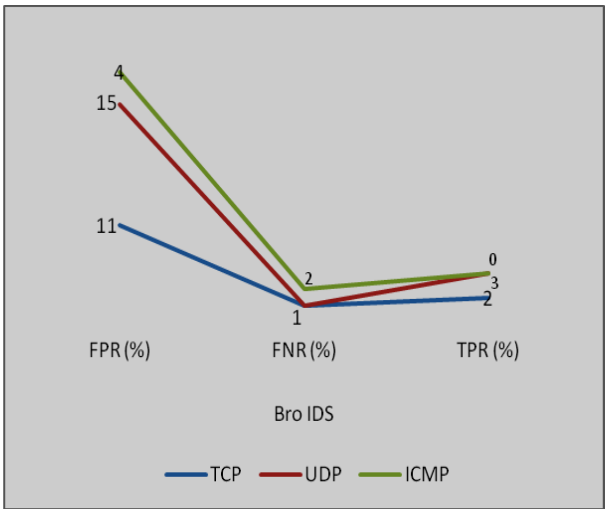

Fig 7: Normal traffic classification by Bro IDS

7. CONCLUSION

For the period of 18 hours experiment blocked into 6hours each, the average true positive rate for Snort for all five malicious attacks was 98.50%, Suricata was 97.40% whereas Bro had average true positive rate of 96.40%. It therefore implies that Snort IDS was ahead of Suricata and Bro in terms of intrusion detection accuracy in this work though the margin was not large.

Under normal circumstance, the three IDSs were not supposed to trigger any alarms because the traffic they were dealing with was not malicious, thus, TCP, UDP and ICMP packets. Therefore, the alarms triggered by these IDSs were misplaced due to possible errors in their rule-sets or misconfiguration.

It should be noted that a high false positive rate alarm will result to less effective IDS and a high false negative rate alarm makes the intrusion detection system vulnerable to intrusions. So, in order to maximize IDSs performances, false positive and false negative rate alarms must be minimized while maximizing accuracy.

8. RECOMMENDATIONS

This work should be extended to real network environment in order to observe the true behavior of these intrusion detection systems. It is also recommended that the network speed for a similar work be increased to 10Gbps so that the IDSs under consideration would have higher speed to deal with various attacks and this can help determine how fast data packets get to the engines of the intrusion detection systems for analysis.

9. REFERENCES

[1] Amira, S.A., Salama, M., Hassanien, A.E., Hanafi, S.E. & Tolba, M.F. (2013). "Multi-layer hybrid machine learning techniques for anomalies detection and classification approach". In. *13th International Conference on Hybrid Intelligent Systems (HIS)* (pp. 215-220). IEEE.

[2] Bro-ids (2008). *Bro-ids Technical Report.* California: International Computer Science Institute.

[3] Bro-ids (2014). *http://www.bro.org.* Retrieved 8 22, 2017, from bro.org: http://www.bro.org

[4] Gerber, J. (2010, 08 26). *http://blog.securitymonks.com.* Retrieved 12 14, 2017, from securitymonks.com: http://www.securitymonks.com

[5] Johnson, K. (2008). Basic Analysis and Security Engine, Tech. Report.

[6] Mehra, P. (2012). A brief study and comparison of Snort and Bro Open Source Network Intrusion Detection System. *International Journal of Advanced Research in Computer and Communication Engineering, 1* (6), 384-389.

[7] Moore, G. (1965). Cramming more components onto integrated circuits. *IEEE, 11* (3), 114-117.

[8] Nielsen, J. (2010, 08 18). *http://www.nngroup.com/articles/law-of-bandwidth.* Retrieved 08 18, 2011, from http://www.nngroup.com

[9] Open Information Security Foundation (OISF). (2011, 04 05). *www.openinfosecfoundation.org.* Retrieved 12 10, 2017, from OISF Foundation web site: http://www.openinfosecfoundation

[10] Paxson, V. (1999). "Bro: a system for detecting network intruders in real-time,". *Journal of Computer Networks, 31*, 2435-2463.

[11] Ross, S. (2007). IS Security Matters. *Information Systems Control Journal, 6*, 14-19.

[12] Sourour, M., Adel, B. & Tarek, A. . (2009). Environmental awareness intrusion detection and prevention system toward reducing false positives and false negatives. *In Proceedings of IEEE Symposium on Computational Intelligence in Cyber Security (CICS '09)* (pp. 107 –114). Oslo: IEEE.