A Comparison of Two Intrusion Detection Systems

[1]R.China Appala Naidu, [2]P.S.Avadhani
Dept. of CS & SE, Andhra University, Visakhapatnam, Andhra Pradesh, India

Abstract
In this paper the performance of the intrusion detection system SNORT, SURICATA are analyzed and tested for Packet loss. It is observed that there is a significant increase in the packet drop when the traffic speed is increased simultaneously. Similarly when the packet size is increased then the drop in packets also decreases.

Keywords
SNORT, SURICATA, Intrusion Detection System, Packet Loss

I. Introduction

An Intrusion Detection System or IDS can be defined as software, hardware or combination of both used to detect intruder activity. It may have different capabilities depending upon how complex and sophisticated the components are. Many Companies offer these IDS appliances which are a combination of hardware and software. An IDS may use signatures, anomaly-based techniques or both. Intrusion Detection Systems are like a burglar alarm for the computer network... they detect unauthorized access attempts. They act the first line of defense for the computer systems. An intrusion detection system inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system Intruders have signatures, like computer viruses, that can be detected using software..

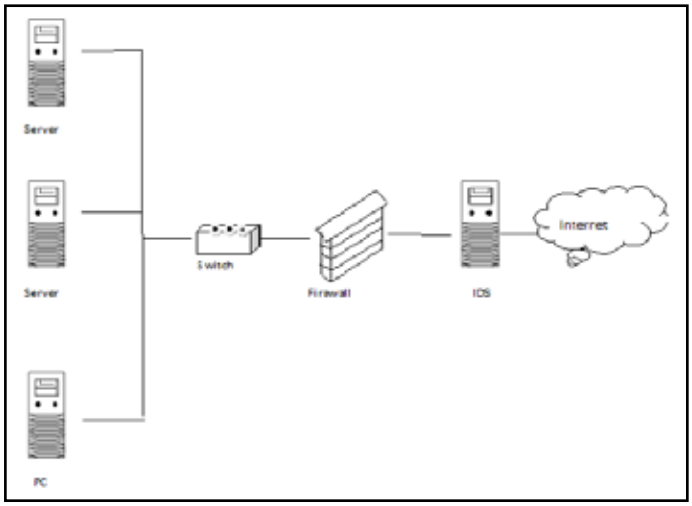

Fig. 1: An Intrusion Detection System With Web Interface

Data packets that contain any known intrusion-related signatures or anomalies related to Internet protocols are found. Based upon a set of signatures and rules, the detection system is able to find and log suspicious activity and generate alerts. Anomaly-based intrusion detection usually depends on packet anomalies present in protocol header parts. In some cases these methods produce better results compared to signature-based IDS. Generally data from network is captured by IDS and the rules are applied to that data and anomalies are detected in it.

II. Research Methodology

Suricata is an open source-based intrusion detection system (IDS). It was developed by the Open Information Security Foundation (OISF). A beta version was released in December 2009, with the first standard release following in July 2010.
Suricata is a high performance Network IDS, IPS and Network Security Monitoring engine. Open Source and owned by a community run non-profit foundation, the Open Information Security Foundation (OISF). Suricata is developed by the OISF and its supporting vendors [1-3].

A. Highly Scalable

Suricata is multi threaded. This means one instance can be run and it will balance the load of processing across every processor on a sensor Suricata is configured to use. This allows commodity hardware to achieve 10 gigabit speeds on real life traffic without sacrificing rule set coverage.

B. Protocol Identification

The most common protocols are automatically recognized by Suricata as the stream starts, thus allowing rule writers to write a rule to the protocol, not to the port expected. This makes Suricata a Malware Command and Control Channel hunter like no other. Off port HTTP CnC channels, which normally slide right by most IDS systems, are child's play for Suricata! Furthermore, thanks to dedicated keywords you can match on protocol fields which range from http URI to a SSL certificate identifier[2].

C. File Identification, MD5 Checksums, and File Extraction

Suricata can identify thousands of file types while crossing a network. Not only can you identify it, but should you decide you want to look at it further you can tag it for extraction and the file will be written to disk with a meta data file describing the capture situation and flow. The file's MD5 checksum is calculated on the fly, so if you have a list of md5 hashes you want to keep in your network, or wants to keep out, Suricata can find it[3].
Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS) created by Martin Roesch in 1998.NIDS is the type of Intrusion Detection System (IDS) that is used for scanning data flowing on the network[4][5][10]. Snort is, by far, the gold standard among open source NIDS systems, with over 100,000 users and 3 million downloads to date. Snort signatures are kept up-to-date by its dedicated users and the Snort website has ample documentation including tutorials. It is not, however, easy to use and requires an experienced security IT professional to configure it properly. The fact that it's free makes it the darling of small and medium-sized businesses that cannot afford the fancy GUIs and wizards of commercial network security products
There are also host-based intrusion detection systems, which are installed on a particular host and detect attacks targeted to that host only. Although all intrusion detection methods are still new, Snort is ranked among the top quality systems available today.
Snort is logically divided into multiple components. These components work together to detect particular attacks and to generate output in a required format from the detection system. A Snort-based IDS consists of the following major components: Packet Decoder, Preprocessors, Detection Engine, Logging and Alerting System and Output Modules
Any data packet coming from the Internet enters the packet decoder.

On its way towards the output modules, it is either dropped, logged or an alert is generated [6].

D. Packet Decoder

Packets from different types of network interfaces are taken by packet decoder and the packets to be pre processed or to be sent to the detection engine are prepared. The interfaces may be Ethernet, SLIP, PPP and so on.

E. Preprocessors

Preprocessors are components or plug-ins that can be used with Snort to arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder[7]. Some preprocessors also perform detection by finding anomalies in packet headers and generating alerts. Preprocessors are very important for any IDS to prepare data packets to be analyzed against rules in the detection engine. Hackers use different techniques to fool an IDS in different ways[8]. Preprocessors are also used for packet defragmentation. When a large data chunk is transferred to a host, the packet is usually fragmented

F. The Detection Engine

The responsibility of a detection engine which is the meet important part of Snort is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The rules are read into internal data structures or chains where they are matched against all packets. If a packet matches any rule, appropriate action is taken; otherwise the packet is dropped. Appropriate actions may be logging the packet or generating alerts [12].

G. Logging and Alerting System

Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files, tcpdump-style files or some other form[11]. All of the log files are stored under /var/log/ snort folder by default. Many command line options discussed in the next chapter can modify the type and detail of information that is logged by the logging and alerting system.

H. Output Modules

Different operations depending on how you want to save output generated by the logging and alerting system of Snort can be done by output module or plug ins. Basically these modules control the type of output generated by the logging and alerting system[9].

III. System Architecture

A. Architecture of the System

The System architecture contains four computers. This network contains high performance PCs running both open source and commercial tools to generate traffic at high speeds and monitor the network performance. A high performance pc was connected to generate more traffic as needed.

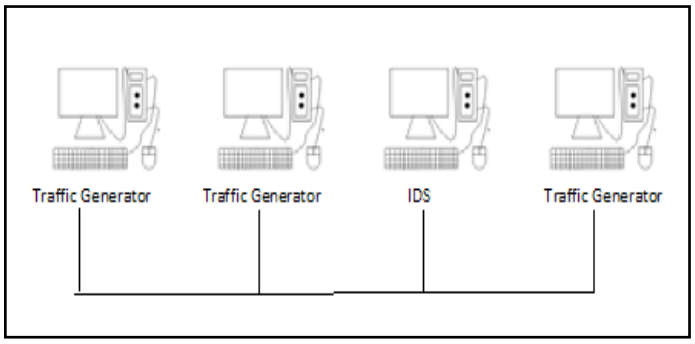

Fig. 2: Basic Network Design

IV. Experimental Results

The network was designed to test the performance of Snort, Suricata on Linux operating system. To get accurate results we tested with packet sizes(512,1024) for both TCP and UDP. The test was performed for the speed range from 250Mbps,500Mbps, 750Mbps, 1Gbps,1.5Gbps,2Gbps.

A. TCP

In this section the Snort, Suricata performed on TCP protocol was addressed. When the packet size is 512, Snort was performed very well at there were no packet drop recorded on 250Mbps ,500Mbps and 750Mbps, in other hand suricata was not dropped packets at 250Mbps. Table 1 shows that when the speed reached 1Gbps Snort started to drop packets. At the speed of 1Gbps Snort dropped 3% packets , at the speed of 1.5Gbps Snort dropped 35% packets, at the speed of 2Gbps Snort dropped 42%packets.

Table 1: Packet Size = 512

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	No Packet Loss
1Gbps	3%
1.5Gbps	35%
2Gbps	42%

Table 2 shows that when the speed reached 500Mbps Suricata started to drop packets. At the speed of 500Mbps Suricata dropped 2% packets , at the speed of 750Mbps Suricata dropped 5% packets, at the speed of 1Gbps Suricata dropped 28% packets, at the speed of 1.5Gbps Suricata dropped 39% packets, at the speed of 2Gbps Suricata dropped 43% packets.

Table 2: Packet Size = 512

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	2%
750 Mbps	5%
1Gbps	28%
1.5Gbps	39%
2Gbps	43%

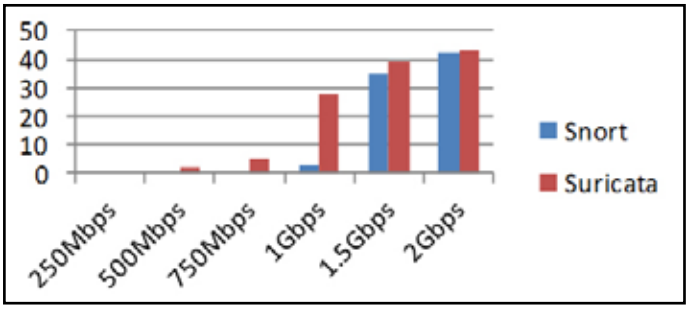

Fig. 3: Comparison of Packet Loss Between Snort and Suricata Where Packet Sizes 512

When the packet size is 1024, Snort was performed very well as there were no packet drop recorded on 250Mbps , 500Mbps and 750Mbps. Table III shows that when the speed reached 1Gbps Snort started to drop packets. At the speed of 1Gbps Snort dropped 1% packets , at the speed of 1.5Gbps Snort dropped 2% packets, at the speed of 2Gbps Snort dropped 9%packets.

Table 3: Packet Size = 1024

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	No Packet Loss
1Gbps	1%
1.5Gbps	2%
2Gbps	9%

When the packet size is 1024, Suricata was performed very well there were no packet drop recorded on 250Mbps , 500Mbps and 750Mbps. Table IV shows that when the speed reached 1Gbps Suricata started to drop packets. At the speed of 1Gbps Snort dropped 16% packets , at the speed of 1.5Gbps Snort dropped 35% packets, at the speed of 2Gbps Snort dropped 39%packets.

Table 4: Packet Size = 1024

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	No Packet Loss
1Gbps	16%
1.5Gbps	35%
2Gbps	39%

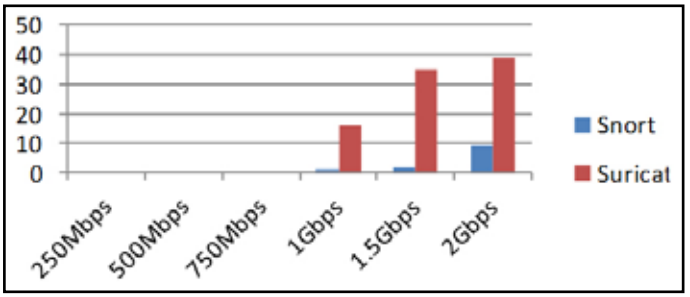

Fig. 4: Comparison of Packet Loss Between Snort and Suricata Where Packet Sizes 1024

Fig. 3 and fig. 4 show that When the packet size was increased, percentage of packet loss decreased and when the speed was increased the packet loss increased.

B. UDP

According to Table 5, When the packet size is 512, Snort was performed very well there were no packet drop recorded on 250Mbps and 500Mbps. when the speed reached 750Mbps Snort started to drop packets. At the speed of 750Mbps Snort dropped 30% packets, at the speed of 1Gbps Snort dropped 42% packets , at the speed of 1.5Gbps Snort dropped 43% packets, at the speed of 2Gbps Snort dropped 43%packets.

Table 5: Packet Size = 512

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	30%
1Gbps	42%
1.5Gbps	43%
2Gbps	43%

According to Table 6 When the packet size is 512, Suricata was dropped packets from the starting. At the speed of 250Mbps Suricata dropped 1% packets, at the speed of 500Mbps Suricata dropped 3% packets, at the speed of 750Mbps Suricata dropped 21% packets, at the speed of 1Gbps Suricata dropped 38% packets , at the speed of 1.5Gbps Suricata dropped 41% packets, at the speed of 2Gbps Suricata dropped 43%packets.

Table 6: Packet Size = 512

Speed	Packet Loss Information
250 Mbps	1%
500 Mbps	3%
750 Mbps	21%
1Gbps	38%
1.5Gbps	41%
2Gbps	43%

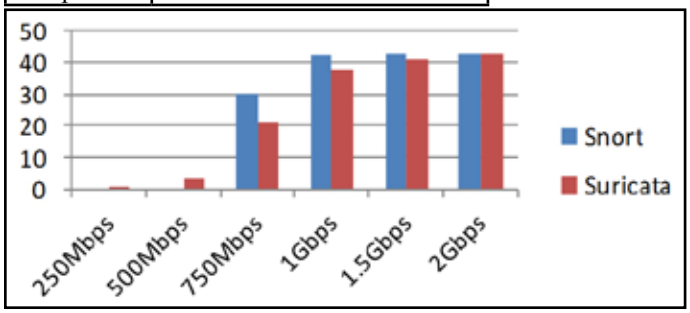

Fig. 5: Comparison of Packet Loss Between Snort and Suricata Where Packet Sizes 512

When the packet size is 1024, Snort was performed very well at there were no packet drop recorded on 250Mbps , 500Mbps and 750Mbps. Table 7 shows that when the speed reached 1Gbps Snort started to drop packets. At the speed of 1Gbps Snort dropped 30% packets , at the speed of 1.5Gbps Snort dropped 35% packets, at the speed of 2Gbps Snort dropped 40% packets. Fig 5 shows that When the packet size was increased percentage of packet loss was decreased and when the speed was increased the packet loss was increased.

Table 7: Packet size=1024.

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	No Packet Loss
1Gbps	30%
1.5Gbps	35%
2Gbps	40%

When the packet size is 1024, Suricata was performed very well at there were no packet drop recorded on 250Mbps , 500Mbps and 750Mbps. Table 8 shows that when the speed reached 1Gbps Suricata started to drop packets. At the speed of 1Gbps Suricata dropped 8% packets , at the speed of 1.5Gbps Suricata dropped 23% packets, at the speed of 2Gbps Suricata dropped 38% packets.

Table 8: Packet Size = 1024

Speed	Packet Loss Information
250 Mbps	No Packet Loss
500 Mbps	No Packet Loss
750 Mbps	No Packet Loss
1Gbps	8%
1.5Gbps	23%
2Gbps	38%

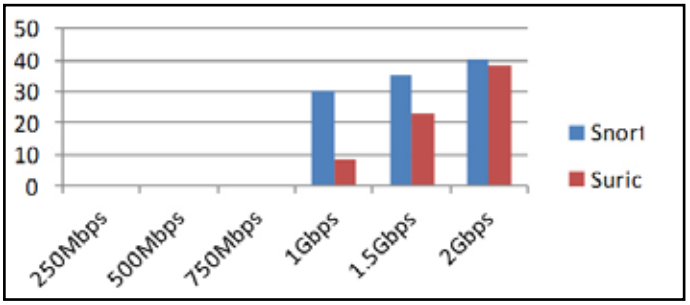

Fig. 6: Comparison of Packet Loss Between Packet Sizes 512 and 1024

V. Conclusion

Identifying packet loss is essential in the networks. In this study Snort and Suricata were used to identify the packet loss in the network. When the packet size was increased percentage of packet loss decreased and when the speed was increased the packet loss increased. Implementing an intrusion detection system and finding more packet loss and all alerts in networks will be indentified in further study.

References

[1] Suricata (2012),"Suricata", [Online] Available: http://www.suricata-ids.org

[2] Surecata(software) (2012),"Suricata (Software)". [Online] Available: http:// www.en. wikipedia.org/wiki/Suricata_ (software)

[3] Surecata Vs Snort (2012),"Suricata-vs-snort", [Online] Available: http:// www.aldeid.com /wiki/Suricata-vs-snort

[4] Rafeeq Ur Rehman,"Intrusion Detection Systems with Snort", Pearson Education, 2003.

[5] Intrusion Detection System (2012),"Intrusion Detection Systems", [Online] Available: http://www.springerlink.com/content/978-0-387-77266-0.

[6] David J.Day, Benjamin M.Burns. (2011),"A Performance Analysis of Snort and Suricata Network Intrusion Detection and Prevention Engines" The Fifth International Conference on Digital Society, 2011, pp.1-4.

[7] Qing-xiu Wu.,"The Network Protocol Analysis Technique in Snort", International Conference on Solid State Devices and Materials Science, 2012, pp.1-4.

[8] Adeeb Alhomoud, Rashid Munir, Jules Pagna Disso, Irfan Awan, Al-Dhelaan, 2011,"Performance Evalution Study of Intrusion Detection Systems", The 2nd International Conference on Ambiems, Networks and Technologies, 2011. pp.1-4.

[9] Snort (2011)."Snort", [Online] Available: http://www.snort.org.

[10] Snort Software (2012),"Snort(software)", [Online] Available: en.wikipedia.org/wiki/Snort_ (software)

[11] Richard Bejtlich,"The Tao of Network Security Monitoringt", Addison-Wesley, 2004.

[12] Packet Loss (2011),"Packet loss", [Online]Available: http://www.nessoft.com/kb/42

Prof.P.S.Avadhani holds a Ph.D. Degree in Computer Science from IIT, Kanpur and completed his M.Tech(CS) from IIT, Kanpur, India. Presently working as Professor in the Department of Computer Science & System Engineering in Vishakhapatnam, Andhra Pradesh, India. His research interests include Data Security, Cryptography, Algorithms, Computer Graphics, Database Management systems and Fuzzy Systems.

R.China Appala Naidu is a Ph.D. student and received Master of Technology in Computer Science and Engineering from University of Mysore, Mysore. He is Research Scholar in the Department of Computer Science and System Engineering, Andhra University in Visakhapatnam, Andhra Pradesh, India. His main research interests include Information Security, Network Security and Fuzzy Systems.