Literature Review about Performance Enhancement on Snort Intrusion Detection System

Anuja Shah[1]
[1]M.Tech. Student
[1]Department of Information Technology
[1]U. V. Patel College of Engineering Ganpat University, Ahmedabad, Gujarat, India

Abstract— As the use of internet is growing rapidly the possibility of attack is also increasing, thus the more secure systems is required. The major challenges for organizations are how to protect their valuable information and internal resources from unauthorized accessed. Intrusion detection consists of various procedures for detection of illegal activities among the system that can identify the intruders. Any attempt to compromise the security of the resource is termed as an intrusion. Intrusion Detection Systems (IDS) detects destructions, generate an alert and activates an alarm. Recently Snort is a very useful tool for Network based Intrusion detection.

Key words: Intrusion Detection System, Snort, Signature based NIDS, Parallel IDS

I. INTRODUCTION

Intrusion Detection System (IDS) is the software for detecting and monitoring the data packet traffic on the network. When it founds the abnormal data of packet traffic which is the attacking pattern, the system will generate alert. Snort is a popular Intrusion Detection System which use for protecting the system's risk from attacker. It is open source lightweight software. . It searches and matches the network traffic's data packet with the rules for checking abnormal data packet traffic. The Snort-IDS utilize the rules matching with the data packet traffic network.

A. Snort

Snort can be configured as a packet sniffer mode that reads the packets from the network, packet logger mode logs packets to the storage device and NIDS enables the Snort to analyse the network traffic against set of defined rules in order to detect intrusion threats. [6]

B. Architecture of Snort

Snort is basically the combination of multiple components.

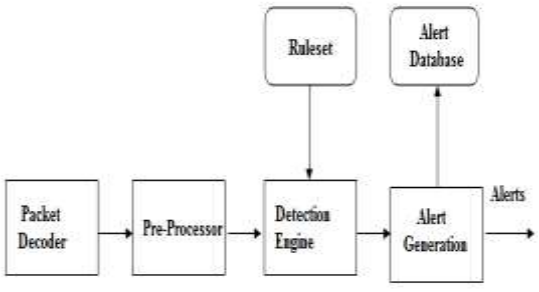

Fig. 1: Snort Architecture

1) Packet Decoder:
The packet decoder collects packet from different network interfaces and send to be pre-processor or detection engine.

2) Pre-Processors:
A pre-processor captures the raw packet and check them against certain plug-ins like behavior of the packets. Pre-processor detects anomalies in packet headers and then generate alerts.

3) Detection Engine:
The detection engine is the signature-based IDS in Snort. It takes the data that comes from the pre-processor and that data is checked through a set of rules. If the rules match the data in the packet, they are sent to the alert.

4) Logging and Alerting System:
Generation of alerts and logging of packets and messages are done in this system. According to what a detection engine find in a packet, packet is used to log activity or generate alert.

5) Output Modules:
Output module saves the output generated by the logging and alerting system of Snort. It is logging in alerts file or some other file. It is sending messages to syslog facility. It can modify configuration on routers and firewalls.

The basic structure of the Snort-IDS rules which are divided into two logical parts: the rule header and the rule option. It contains the criteria definition for matching between a rule and the data packet traffic network. In addition, the action field of the rule header also able to define the type of action such as pass, log alert etc. The rule options follow the rule header and they are within a pair of parentheses.

Fig. 2: Snort Rule

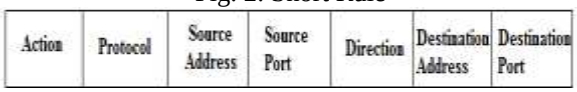

Fig. 3: Rule Header Format

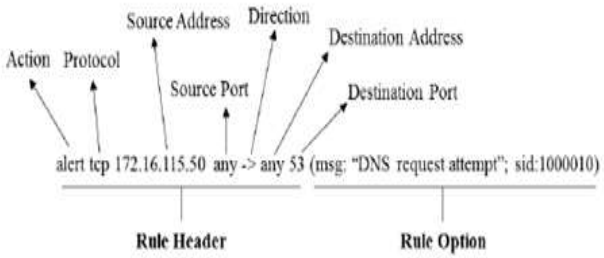

Fig. 4: Rule Example

C. WinPcap

Snort doesn't have own packet capture tool, which needs to catch WinPcap by the outside tool bag. WinPcap is an open source library for packet capture and network analysis for the Windows System. It provides facilities to capture raw packets.

II. LITERATURE REVIEW

There are main two methodologies. Signature based IDS and Anomaly based IDS .The main advantage of signature - based systems are that they usually produce very few false

positives. But it cannot detect unknown attacks and sometimes it cannot detect the variations of known attacks. Anomaly can detect unknown attacks. But it leads to relative high false positive rate. By default snort will not provide any anomaly detection and is purely a misuse based system. Extra plug-in is required for anomaly detection system.

When snort is in its active detection mode it will utilize 100% CPU and will slow down the performance of the system. During a DoS attack snort throughput increases drastically and drop large number of packet. Therefore possibilities of detecting possible attack patterns are more but it fails to analyse those dropped packets.

Paper [6, 7] describes about Snort IDS Using Pattern Matching Technique**.** Snort use predefined algorithms for pattern matching namely Aho-Corasick algorithm. Snort rule tree structure (automata) will be constructed for both the header and the options portion. In Snort, the packet will be compared against all the rules irrespective of its protocol. This will increase the number of searches and it result in deeper tree structure, it requiring more processing power and memory usage.

In case of improved Snort the finite automata construction for the header portion is completely avoided and the automaton is constructed for the options portion alone. This is further divided by constructing the automata only for the content portion of the predefined rule set. Initial pre-processing is done by grouping the rules based on the protocols and storing them separately in four different files for TCP, UDP, ICMP and IP. It extracts the protocol field from the packet‘s header and then packets are redirected to the file that contains the rules related the protocol to which the packet is associated.

Paper [2] describes about Intrusion Detection System Based on Snort Rules for Network Probe Attack Detection. The MIT-DARPA 1999 dataset is tested and evaluated by MIT Lincoln Laboratory. The dataset consists of normal and abnormal connection which was recorded in many file formats. MIT-DARPA 1999 dataset has total 656 probe attacks. The total number of the Snort-IDS detected network probe attack is 1945 times[2]. Snort IDSs can generate thousands of alarms in a day that flood network administrators. Many of these alarms are usually considered to be so called false alarms. As a result, network administrators run the risk of missing good alarms lost in the noise generated by the false alarms.

Paper [5] describes about IDS Alerts Classification using Knowledge-based Evaluation.

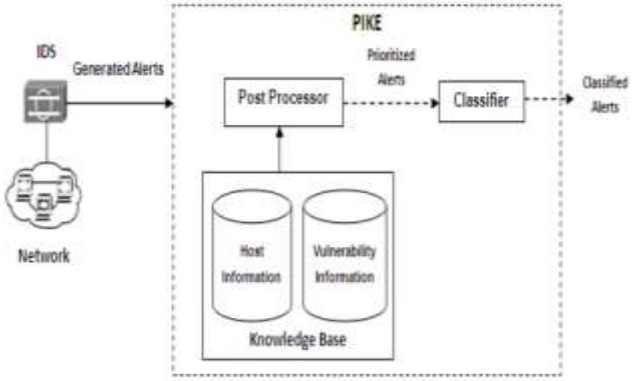

Fig. 5: PIKE Architecture

Post-processor for IDS alerts using Knowledge-based Evaluation (PIKE) makes use of a knowledge base that contains information about hosts present on the network and known vulnerabilities. It is built by using remote information gathering tools and vulnerability information sources. The alerts generated by the IDS are processed along with entries of knowledge base to generate a relevance score. Relevance is the measure of importance of an alert. The alerts are classified by a Binary Classifier on the basis of threshold value. The threshold value depends on Host Configuration and Vulnerability Information which can differ on company to company and as per requirement. We can classify packets as they are harmful or not using threshold value.

Paper [9] describes Hybrid Based Methodology .The hybrid based methodology works by combining signature and anomaly of methodologies. Snort is modified by adding an anomaly based engine to its signature based engine to create a better detection and the new hybrid systems is tested against the regular Snort using same test data. First it used anomaly based model to filter the data and then it used signature based model to detect intrusion attempts. So, which packets are not detected by anomaly based technique they are detected by signature based technique, vice versa.

The accuracy of post processor and pre-processor is not up to mark. Snort IDS performance goes degradable for more traffic and drop the packets without examine them, because of that sometimes it can miss good alerts and generate false alarms. The most important weakness of NIDS for whole network traffic is a time consuming job. The network speeds rises day by day, so need of efficient intrusion detection techniques that reduce the processing time for more traffic emerges. To solve this problem different researchers give different techniques and IDS models using parallel computing.

The parallel computing is configured with two type of array.

- Function parallel System
- Data parallel System

1) Function Parallel System:

In a function-parallel system the policy rules are distributed across the array of processors, so each processor has a smaller localpolicy. The data (packet payload) is then duplicated across the array of processors and every processor searches the data for a smaller number of signatures (defined by the local policy). Distributing the rules across each processor reduces processing delay.

2) Data Parallel System:

In a data parallel configuration, each processor in the array has the same policy (same signatures). The data is then sent to one processor; such that each processor has 1/*n* of the original load (load balancing is the objective).

Paper [14] describes a Parallel Technique for Improving the Performance of Signature-Based NIDS**.** It is so hard to define all the 65536 ports in ip table or packet forwarding program because of time consuming, they chose a smaller range of ports for the implementation.

Sensor Number	**Range of Destination ports**	**Dedicated Rules**
Sensor 1 (snort 1)	25, 80, 110, 143, 8080 (SMTP,HTTP,POP3,IMAP4)	Rules for all the packets with destination ports according to sensor 1

Sensor 2 (snort 2)	21,22,23,53,3306 (Ftp, SSH, Telnet, DNS server, MYSQL database server	Rules for all the packets with destination ports according to sensor 2

Table 1: Ports and Rules Distribution

1) Step 1: Packets are captured by Winpcap.
2) Step 2: These packets should be split and loaded across the dedicated NIDS sensors. Packets were loaded.
3) Step 3: By the use of packet forwarding program, the packets should distribute between sensors based on their port numbers.

Each sensor takes appropriate rule to detect signature. If the attack recognized, alert will be generated and sent to the server. Run time for packet processing for the first snort with 34406 packets was 0.937000 seconds, and for the second one with 25455 packets was 0.531000 seconds. The whole system can achieve a higher throughput. This results show the proposed architecture reduces the processing time of the traffic, improves the performance of signature based network intrusion detection system than centralized architecture.

Paper [15] describes Parallel Component Agent Architecture to Improve the Efficiency of Signature Based NIDS. Agents are software computing entities that perform intrusion detection tasks autonomously and need to be able to affect environment using some type of predefined mechanisms. In proposed architecture we divide the main database in small databases. Intrusion detection module takes packet as input and extracts the signature and compares it with available signatures in the small databases.

Agent Number	Destination Ports	Rules
Agent 1	53,80,110,143	1 to 200 rules
Agent 2	23, 3306, 22	200 to n rules

Table 2: Describe The Port And Rules Set

Firstly agent 1 database compares the packets signature with their small database consist of the snort rules given range of protocol and at the same time the copy of same packet is processed by the agent 2 database in similar way. We can increase or decrease the number of agents depends on the network traffic.

Paper [16] describes divided data parallel system consists of an array of *n* processors, each implementing the same policy. The packet payload is divided across the array of processors. Each processor inspects a different portion or fragment of the same packet. A divided data parallel system where a packet is divided into fragments then forwarded to an array of processors. The match-bit allows one processor to quickly indicate to other processors that a match has been found for a given packet. It allows the processors operate independently. Once the notification has been received, the remaining processors can start inspecting another packet. Initially match –bit set to false, a match-bit for a packet is set to true if a processor finds a pattern match with an associated fragment. If the match-bit associated with a packet is true, then the processor can ignore any fragments associated with that packet. This also helps the processors to operate more asynchronously since they can quickly ignore certain fragments.

III. Related Work

From Literature Survey, the accuracy of post processor and pre-processor is not up to mark. Snort IDS performance goes degradable for more traffic and drop the packets without examine them, because of that sometimes it can miss good alerts and generate false alarms. Here we have proposed various following approaches to enhance the performance Snort IDS in terms of accuracy in alerts.

1) Improvement in Algorithm which is used for Pattern Matching.
   – Snort uses Aho-Corasick Algorithm for Pattern Matching. We can improve this algorithm by modifying the algorithm which yields better performance.
   – We can add more efficient algorithm for Pre-processor and Post-processor processing.
2) Increasing Snort Performance by parallism.

In parallel processing a file is distributed to multiple nodes (slaves) which can be accessed concurrently. Master (main server) which capture the packets from real time network and distribute those data in slave computers to do job.

IV. Conclusion

In Snort IDS system possibility of packet dropping, false alarms, CPU usage, and time required to generate the alerts is more in the case of heavy traffic. The proposed approaches may reduce the rate of packet dropping, false alarms, resource and time consumption and enhance the performance of Snort IDS.

References

[1] G. Kurundkar , N. Naik , Dr. S. Khamitkar , "Network Intrusion Detection using SNORT", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 2,Mar-Apr 2012, pp 1288-1296.
[2] N. Khamphakdee, N. Benjamas, S. Saiyod, "Improving Intrusion Detection System Based on Snort Rules for Network Probe Attack Detection", 2nd International Conference on Information and Communication Technology (ICoICT), IEEE, May 2014, pp 69-74.
[3] F. Massicotte, Y. Labiche, "On the Verification and Validation of Signature-Based, Network Intrusion Detection Systems",23rd International Symposium on Software Reliability Engineering, 27-30 Nov. 2012 ,pp61 - 70
[4] A. Jarrah, A. Arafat, "Network Intrusion Detection System Using Attack Behaviour Classification" 5th International Conference on Information and Communication Systems (ICICS) 1-3 April 2014 IEEE , pp1 - 6
[5] D. Gupta, P. Joshi, A. Bhattacharjee, R. Mundada. IDS Alerts Classification using Knowledge-based Evaluation". 2012 IEEE
[6] C. Kacha1, A. Shevade, Dr. S. Raghuwanshi. "Improved Snort Intrusion Detection System Using Modified Pattern Matching Technique". International Journal of Emerging Technology and Advanced Engineering ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 7, July 2013
[7] C. Huang, J. Xiong, Z. Peng . "Applied Research on Snort Intrusion Detection Model in The Campus Network", Symposium on Robotics and Applications (ISRA) 15-18 March 2012 IEEE, pp1 - 6

[8] F. Alserhani, M. Akhlaq, I. U. Awan, A. J. Cullen, J. Mellor, P. Mirchandani , "Snort Performance Evaluation". Informatics Research Institute, University of Bradford, Bradford, BD7 1DP, United Kingdom.
[9] D. Mudzingwa , R. Agrawal , "A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS)" 2012 IEEE
[10] A. Jadhav, P. Jadhav, P. Kulkarni , "A Novel Approach for the Design of Network Intrusion Detection System(NIDS)", 2013 International Conference on Sensor Network Security Technology and Privacy Communication System (SNS & PCS) 18-19 May 2013 IEEE , pp22 - 27
[11] S. Shah, P. Singh , "Signature-Based Network Intrusion Detection System Using SNORT And WINPCAP", International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 10, December- 2012 ISSN: 2278-0181
[12] P.Agarwal, S.Satapathy, "Implementation of Signature-based Detection System using Snort in Windows",International Journal of Innovations & Advancement in Computer Science IJIACS ISSN, Vol 3, Issue 3 May 2014 , pp 2347 – 8616
[13] G. Ahmed, H. Mehdi, M.N.A. Khan, "Characterizing Strengths of Snort-based IDPS", Research Journal of Recent Sciences, ISSN 2277-2502 Vol. 3, no 4, Apr. 2014 ,pp88-94
[14] F. Shiri, B. Shanmugam, N. Idris, "A Parallel Technique for Improving the Performance of Signature-Based NetworkIntrusion Detection System'', Communication Software and Networks (ICCSN), 27-29 May 2011, pp692 – 696
[15] A. Umar, C. Li, Z. Ahmad "Parallel Component Agent Architecture to Improve theEfficiency of Signature Based NIDS'' Journal of Advances in Computer Networks, Vol. 2, No. 4, December 2014 ,pp 269-273
[16] C.Kopek, E. Fulp, P. Wheeler "Distributed data parallel techniques for content-matching intrusiondetection systems**"** Military Communications Conference, 29-31 Oct. 2007 IEEE, pp 1 - 7

