A Parallelism Technique to Improve Signature Based Intrusion Detection System

Prof. Menka Patel
Computer Engineering Department
U V Patel College of Engineering
Kherva-Mehsana, INDIA
menka.patel@ganpatuniversity.ac.in

Prof. Hitesh Rajput
Computer Engineering Department
U V Patel College of Engineering
Kherva-Mehsana, INDIA
hitesh.rajput@ganpatuniversity.ac.in

Prof. Himansu Patel
Information Technology Department
U V Patel College of Engineering
Kherva-Mehsana,INDIA
hhp01@ganpatuniversity.ac.in

***Abstract*— nowadays, it is vital for organization to protect their valuable information and internal resources from malicious access. Firewall is one of solution to prevent from unauthorized access, but it cannot monitor network traffic. To monitor and detect threats network monitoring tool like Intrusion Detection System (IDS) is required. Different IDS uses several techniques for Intrusion Detection. Signature based detection techniques are widely used in networks for fast response to detect threats. Because of the high-speed a large volume of data should be analysed and processed with high-speed infrastructure. It is time consuming process because signature based IDS scan all the network traffic and detect malicious packets. Snort is the best tool for signature based intrusion detection system can monitor the network traffic and generate alert for malicious packet. A parallel technique is a best alternative to reduce processing time and improve the performance of network intrusion detection system. In this paper, we have proposed data parallelism technique for signature based intrusion detection system using Snort in which detection rate is increased, the time to analyse packets and dropped packets are decreased. Our system is horizontally scalable that means we can increase or decrease hosts as per requirement.**

***Keywords*— Intrusion Detection System; Snort; Data Parallelism; Signature-based**

I. INTRODUCTION

Information security has become an important part of most systems and software in the last 20 years. To protect information from various malicious activities today, most of the organizations also pay attention to the security of their products based on performance or design. The most common way attackers harm computer systems is by using malicious software known as Malware. Using this type of software one can design activities like gain access to system without the knowledge of authorized user, spy on or destroyed a system [1]. According to statistics from Symantec [2] the number of new malware variants in 2015 was 431 million, 36% more than in 2014, and the ransom ware numbers increased by more than 35% during that same time. The statistics report from Kaspersky's Security Bulletin 2016 [3] shows they detected financial malware in more than 2.5 million devices, which is 46% higher than in 2015. In light of above statistics it is required to have IDS which monitors the network traffic and detect the malicious activities. In recent times the information

security research has been focusing much attention on the IDS.

To detect the pattern and signatures of these malicious attacks normally IDS parameters divided into signature based or misuse detection and anomaly based system or behaviour based methodology. An IDS that relies on predefined knowledge about attacks to detect anomalous traffic is known as Signature based IDS. For that set of rules are already defines to identify intruder. Anomaly based IDS works by comparing observed activity against a baseline profile. The baseline profile is learned normal behaviour of the monitored system. It is developed during the learning period where IDS learns the environment and develops a normal profile of the monitored system. This environment can be networks, users and systems. Anomaly-based intrusion detection triggers an alarm on the IDS when some type of unusual behaviour occurs in network. This would include any event, state, content, or behaviour that is considered to be abnormal by a pre-defined standard [4].

In Signature-based IDS, every signature requires an entry in the database and complete database might contain hundreds or even thousands of entries. Each packet is to be compared with all the entries in the database. This database contains known attack signatures. Any signature observed in the monitored environment that matches the signatures on file is flagged as a violation of the security policy or as an attack. Events that do not match with any of the attack models are considered as a part of legitimate activities. The process is very efficient if signature database is up-to-date.

But it is time consuming and slow down the throughput of the signature of network IDS [5]. Fig. 1 shows the working mechanism of signature based IDS [6].

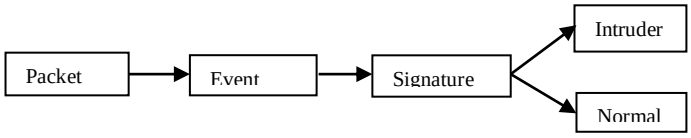

Fig. 1. Signature based IDS working mechanism

The remainder of this paper is structured as follows. In section 2 we review Snort tool for signature based intrusion detection systems. Section 3 presents an overview of related works in this area. Section 4 provides proposed module and workflow. In section 5 we discuss an implementation of the proposed module and results analysis. Section 6 presents conclusion.

II. SNORT

Snort is an open source, lightweight, popular IDS, which is using for protecting the system's risk from an attacker. Snort can be installed on computer architecture and operating system platform. Snort-IDS also generate alerts in the real-time. It searches and matches the network traffic with the rules for checking abnormal data packet traffic [7]. Snort can be configured as a packet sniffer, packet logger and NIDS [8].

A. Architecture of Snort

Snort is basically the combination of multiple components, works together to find a particular attack and then take the corresponding action that is required for that particular attack. [9, 10] Fig. 2 shows architecture of snort.

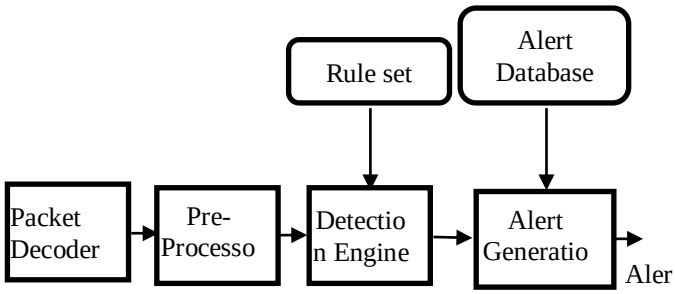

Fig 2 Snort Architecture [9]

The details of each component of snort are as follow:

1. Packet Decoder: It collects packet from different network interfaces and send to pre-processor or to the detection engine.
2. Pre-processors: It captures the raw packet and checks them against certain plug-ins. This plug-ins check for a certain type of behavior of the packets. Pre-processor detects anomalies in the packet headers and generate alerts. Pre-processors are very important for any IDS to prepare data packets to be analyzed against rules in the detection engine.
3. Detection Engine: The detection engine is the signature-based IDS. This takes the data that comes from the pre-processor and that data is checked through a set of rules. If the rules are match with the data, it will generate an 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 finds in a packet, it is used to log activity or generate alerts.
5. Output Modules: Output module saves the output generated by the logging and alerting system of Snort. Depending on the configuration, functions of output modules are as follow:
   - Simply logging in alerting file or some other file.
   - Sending messages to SYSLOG facility.
   - Modifying configuration of routers and firewalls.
   - Sending Server Message Block (SMB) messages to Microsoft Windows-based machines (pop-up)

The basic structure of the Snort-IDS rules 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.

B. WinPcap

Snort does not have its own packet capturing tool; therefore WinPcap is used. WinPcap is an open source library for packet capture and network analysis for the Windows System. It provides facilities to capture raw packets. Filter the packets according to user-specified rules before dispatching them to the application. Transmit raw packets to the network. Gather information on the network traffic.

III. RELATED WORK

As we have discussed in previous section signature based IDS is accurate as compared to anomaly based IDS but now a days we have huge amount of network traffic and therefore it is required to increase the speed for data monitoring. Ref [8] focussed on analysis of the performance of Snort under heavy traffic conditions. Snort has been evaluated on different operating systems platforms and hardware resources by subjecting it to different categories of input traffic. Attacks were also injected to determine the detection quality of the system under different conditions. Author has done evaluation based on detection rate, packet loss and CPU usage.

Performance of Snort has been evaluated in relation to OS by generating attacks from similar OS platform and observing the packet loss. Snort has shown quite good performance up to 400 Mbps of network traffic by detecting 100% attacks; however its performance declined above 500 Mbps. At 1.0 Gbps traffic Snort was able to capture only 30% of the generated attacks. By analysing network traffic of 500 Mbps, the system dropped more than 50 %

packets. This value has increased up to 75% for 1.0 Gbps of input traffic. Snort CPU usage touched 98% for input traffic of 1 Gbps and 80% for input traffic of 500 Mbps. This high CPU usage scenario has caused the major performance bottleneck. To increase the scanning and detection speed many researchers have proposed various techniques related to parallel processing of signature based IDS. Ref [1] proposed a module is intended to decrease database size inside hosts and network nodes as well as use parallel processing technique to maximize detection speed. Decreased database size will result in better performance. The difference between the proposed module and other models mentioned in the previous section is that the previous models did not solve the problem of the new discovered signatures and how to deal with them. In addition, using parallel processing with two small databases will improve IDS performance even more. This proposal will not skip any malware event, even if the malware is old. Another model introduced for improving IDS performance is parallel technique [5]. In proposed method, a switch or router can be used to split the incoming traffic between two sensors according to their switching or router table. Each sensor is dedicated parts of the whole Snort rules. When the signature of a known attack is recognized by the detection engine based on the dedicated rules in the Snort, the alerts messages will be sent to the log file and also in database. Ref [6] developed a framework by combining the two approaches, multithreading and parallelizing IDS. In this researcher has mainly focused on how to reduce the time needed to compare the signatures and update the small databases in agents. Author has used a duplicator module, UDP packet duplicator which is used to send same packet to every agent. Instead of the UDP packet duplicator it is possible to use a system with Linux. The Linux kernel version 2.6.35 introduces a new configuration option CONFIG_NETFILTER_XT_TARGET_TEE:
This option adds a "TEE" target with which a packet can be cloned and this clone can be rerouted to another next hop. By using this method, agents can detect intrusions more quickly by comparing each network packet with the small agent's databases. Then agent follow the complete process, compare the signature in the frequent database, in positive case packet will be intruder and in negative case the packet is considered to be a normal packet. Ref[11] proposed a new model called Dynamic Multi-Layer Signature based IDS using Mobile Agents, which can detect imminent threats with very high success rate by dynamically and automatically creating and using small and efficient multiple databases, and at the same time, provide mechanism to update these small signature databases at regular intervals using Mobile Agents. Ref [12] proposed a framework for multi sensor intrusion detection called Fuzzy Agent-Based Intrusion Detection System. A unique feature of this model is that the agent uses data from multiple sensors and the fuzzy logic to process log files. Use of this feature reduces the overhead in a distributed intrusion detection system. Author has developed an agent communication architecture that provides a prototype implementation. In this author has also discussed the issues of combining intelligent agent technology with the intrusion detection domain. Ref [13] author has proposed multi threading technique to improve the performance of signature based IDS. The multithreading concept is used to handle the network traffic. In proposed technique author has

implemented preprocessing part, where incoming packets are first distinguish according to protocols. Ref [14] author has developed a framework which may be used to classify various approaches to parallelizing intrusion detection systems. Parallelization of IDS can occur at three general levels: node (entire system), component (specific task), and sub-component (function within a specific task). Researcher has proposed node-level data parallel approach. In this author's goal is not to construct a fully-functioning, deployable node level parallel IDS, but rather to attempt to provide a upper-bound on the performance of the node-level data parallel approach. Ref [15] this 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. In divided data parallel system 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.

IV. PROPOSED APPROACH

Snort can be used as single system and as parallel system. As per the related work, there some pros and cons in snort. Snort IDS performance goes degradable for more traffic and drop the packets without examine, because of that sometimes it can miss good alerts and generate false alarms. When snort is in its active detection mode it will utilize 100% CPU and will slow down the performance of the system. 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 cost of function parallel system will affect the speed of the system. If fragmentation is done on the packet payload, it is difficult to analyse alert file and its time consuming job. If any processor is failed to analyse any fragment due to any reason, then it can miss good alerts and generate false alerts alarms. If main database is divide in small database, and sending duplicate packets to every sensor then every packet is loaded on all processor. It is no need to waste time to send same packets to every processor and wait for alert file of all processor. So, we can say that it takes more time, more space. Rules distribution based on the range of destination ports is difficult, when you deploy the system in different organization.

We have proposed parallel architecture in which network packets are received by a main server distributed to multiple nodes which can be accessed concurrently using load balancer module to do the job. The packets are distributed one by one equally on each node. Each node has individual Snort IDS system with same set of rule to detect intrusion.

Each node individually traces the captured packets and analyse them. Then alert files are generated on each node.

V. EXPERIMENTAL RESULTS

For experiment we have installed WinPcap for capturing packets. To enable IDS system it is required to change snort.conf file. It is a main file for snort to make the changes for expanding all rules and path for a detection process. We have configured HOME_NET Address and DNS Address as IP address. To perform data parallelism we have used Apache Httpd 2.4.12 module for load balancing. We have made the HTTPClient Java program uses apache-httpclient library for preparing the HTTP packet, it sends an HTTP packet to the desired location by creating an HTTP socket. It internally uses Java Naming Directory Interface (JNDI) library and Java Network API (JNA) library for creating an Http Socket via http protocol and establishes the Http connection with the server.

FLOW OF CREATING HTTP REQUEST:

It requires site and request number as input parameters.

1. The parameters are passed via System property along with running the jar file
2. The site parameter takes the input for the URL to which the http request is to be sent.
3. The request number parameter takes the input for the number of packets to be sent.
   Example: java -jar -Dsite=http://localhost:80/test -Dreqno=50 httpclient.jar

FLOW OF LOAD BALANCING MODULE:

1. Firstly requests are coming on Httpd server.
2. We set load balancing module on Httpd server. All requests are sending by Http server to load balancing module.
3. Then all requests are going to configure proxy balancer from load balancing module.
4. After configuration of Proxy balancer, it passes requests to proxy method.
5. Then it forward packets to host server as we decided in proxy balancer as ratio of 1:1:1.
   Example:
   <Proxy balancer://mycluster>
   BalancerMember http://192.168.0.3:8000
   BalancerMember http://192.168.0.4:8005
   BalancerMember http://192.168.0.5:8008
   </Proxy>
   ProxyPass /test balancer://mycluster

EXPERIMENTAL RESULTS AND ANALYSIS

Table 1 shows the malicious packets detection rate, packets dropped rate and time in second for different numbers of nodes with different number of packets. The results show that load balancing gives better detection result. This technique reduces packet dropped rate. Using data parallel architecture we can increase packet detection rate.

Table 1. Experimental Results

No. of Nodes	Packets	Detection Rate (%)	Dropped Rate (%)	Time (second)

1	1000	99.7	0.3	70
2		99.85	0	53
3		99.90	0	45
1	2000	99.60	0.4	120
2		99.79	0	110
3		99.89	0	71
1	3000	99.39	0.61	190
2		99.86	0	139
3		99.87	0	138
1	4000	99.33	0.67	193
2		99.78	0	139
3		99.86	0	115
1	5000	99.25	0.742	200
2		99.80	0	175
3		99.86	0	160

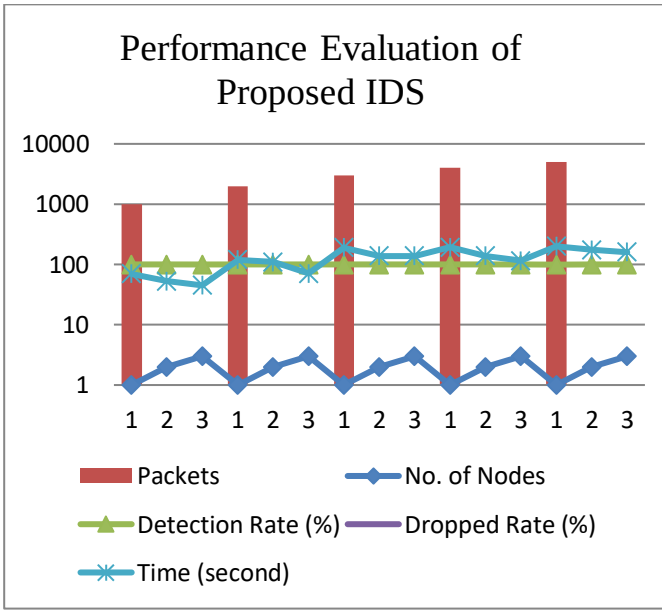

Fig. 3. Performance Evaluation of Proposed IDS

Fig. 3 represents the analysis of Snort processing using single processor against snort processing done on distributed environment using either two or three processor. The graph represents the readings manipulated by four comparison parameters. No of nodes Vs Time taken for processing Vs Detection Rate (%) Vs Dropped Rate (%).

When Snort processing is done by distributed environment, throughput achieved in distributed environment yields better results against the traditional snort processing. It is observed that if snort processing is done by distributed environment, it yields almost zero percent dropped rate and increase detection rate along with decrease time for processing.

VI. CONCLUSION

The experiment results clearly revealed that parallelism approach gives better detection result. This technique reduces the amount of packets per processor, time of traffic, maintaining state. It can achieve a higher throughput, improves the performance of signature based network intrusion detection system than centralized architecture, lesser CPU and resource utilization for snort processing because packets are distributed among processors. Using parallel architecture, packets are not dropped and detection rate is increased. Proposed architecture gives fast process and it has high availability. We have used load balancing module so, if any machine takes more time to analyse packet, or busy with any other schedule, then it send packets to another machine which is free and if any machine is failing, load balancer takes another route to send packets automatically.

The proposed solution is horizontally scalable means one can increase or decrease hosts as per requirement, also independent of hosting service means it can work in any operating system. As a future work, take all alert files from all host machines and gather it at one host machine to analyse that how many requests have same source id and snort port number for finding DOS attack. After collecting alert files from each host machine use function parallel system to get more detection rate and it can reduce false alarms.

REFERENCES

[1] A. Almutairi and N. Abdelmajeed, "Innovative signature based intrusion detection system: Parallel processing and minimized database Sign In or Purchase", in the Frontiers and Advances in Data Science (FADS), 2017 International Conference on, Xi'an, China, 2018. ISBN: 978-1-5386-3149-2

[2] Symantec Internet Security Threat Report, Vol. 21, Apr 2016.

[3] M. Garnaeva, F. Sinitsyn, Y. Namestnikov, D. Makrushin and A. Liskin, "Kaspersky Security Bulletin Overall Statistics", 2016.

[4] F. Gong, "Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection", White Paper from McAfee Network Security Technologies Group 2003.

[5] F. Shiri, B. Shanmugam and N. Idris, "A parallel technique for improving the performance of signature-based network intrusion detection system", in 3rd International Conference on Communication Software and Networks, Xi'an, China, 2011, pp. 692 - 696.

[6] H. Umar, C. Li and Z. Ahmad, "Parallel Component Agent Architecture to Improve the Efficiency of Signature Based NIDS", Journal of Advances in Computer Networks, vol. 2, no. 4, pp. 269-273, 2014.

[7] N. Khamphakdee, N. Benjamas and S. Saiyod, "Improving Intrusion Detection System based on Snort rules for network probe attack detection", in Information and Communication Technology (ICoICT), 2014 2nd International Conference on, Bandung, Indonesia, 2014.

[8] F. Alserhani, Monis 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] A. Jadhav, A. Jadhav, P. Jadhav and P. Kulkarni, "A Novel Approach for the Design of Network Intrusion Detection System(NIDS)", in Sensor Network Security Technology and Privacy Communication System (SNS & PCS), 2013 International Conference on, Nangang, China, 2013.

[10] 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

[11] M. Uddin, K. Khowaja and A. Abdul Rehman, "Dynamic Multi-Layer Signature Based Intrusion Detection System Using Mobile Agents", International Journal of Network Security & Its Applications, vol. 2, no. 4, pp. 129-141, 2010.

[12] R. Wasniowski, "Multisensor Agent Based Intrusion Detection", World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical,

Automation, Control and Information Engineering, vol.1, no. 5, pp 1465- 1468, 2007.

[13] D.Gaikwad, P. Pabshettiwar, P. Musale, P. Paranjape, A. S. Pawar, “A Proposal for Implementation of Signature Based Intrusion Detection System Using Multithreading Technique”, International Journal Of Computational Engineering Research, vol. 2, Issue. 7, pp. 59- 65, 2012

[14] P. Wheeler and E. Fulp, "A taxonomy of parallel techniques for intrusion detection", in ACM-SE 45 Proceedings of the 45th annual southeast regional conference, 2007, pp. 278-282.

[15] C. Kopek, E. Fulp and P. Wheeler, "Distributed Data Parallel Techniques for Content-Matching Intrusion Detection Systems", in Military Communications Conference, 2007. MILCOM 2007. IEEE, Orlando, FL, USA, USA, 2007.