International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 4, Jul - Aug 2016

RESEARCH ARTICLE OPEN ACCESS

Snort Using Parallel Architecture for Intrusion Detection in Busy Network

Aiswarya Mohan K [1], Jyothi B [2]
Student [1], Professor [2]
Department of Computer Science d Engineering, GISAT
Mahathma Gandhi University, Kottayam

ABSTRACT
In our day to day life internet has its own importance, the technology has a rapid economic growth as well as the attacks in network also booming day by day, in such a situation network security has its own importance. There, like firewalls and antivirus but these all fails against contemporary malicious activities. Now a day's one of the reliable technology is IDS(intrusion detection system) ,here the IDS monitors the network, the activities going on the system for the malignant activities, scheme policy infringement.
Snort is one of the network intrusion detection system used to run on Linux platforms. In snort, detection of potential network intrusion is using a rule based intrusion detection mechanism. In case of a busy network or a high traffic network, it is difficult for detect all the incoming packets. Whenever it came to snort, it will drop all packets when the traffic is high. In such cases, there may be chances of loss of incoming packets which are not malicious. To overcome such a situation, we could handle the incoming packets by introducing parallelism.
Keywords: — network security, IDS, parallel architecture, network traffic

I. INTRODUCTION

Security is becoming a valuable element in present-time network foundation for covenanting the security of composite information system. The main assignment of a network intrusion system is to scrutinize network traffic with the goal of discernment and the corroboration of illegitimate activities. There are certain other methods to fortify the seclusion, like analyzing each packets.

To track and regroup distinct connections we need a NIDS. The throughput of recorded traffic and the number of connections can influence the size of memory and computational capacity that are needed by each NIDS.

Due to the fact that various computer systems are unable to prevent attacks such DoS and DDoS because of the threats are re severe and unrepairable. The main function of these attacks is to send more high-speed traffic to a network address and thereby make high complexity, which blocks or deliberate the performance of users' systems by utilizing the vulnerabilities bugs, errors and misconfigurations generated from internal and external networks. Therefore it is important to implement ID systems (IDSs) in computer networks which have a capability to handle high traffic and high-speed networks. IDSs may be either software applications or hardware to hear for and detect malicious activities to and from a individual or network systems. Therefore, an IDS's mechanism is important to implement at the network gateway.

In this paper, we prefer an novel parallel NIDS architecture that accounts high performance by combining snorts in parallel. Parallel architectures are the most important thing for assuring better performance of NIDS even to face upcoming high volume networks. We manifest that the proposed NIDS can effectively scale and deal with increasing traffic volumes traffic distribution and load balancing technique, that vigorously freights incoming traffic to the available receivers. Proposed idea allows the NIDS to scrutinize high speed links with no neither packet loss nor negative collision on the precision of the traffic analysis, that remain genuine.

Here we are discussing about the architecture and idea by which it works.

II. RELATED WORKS

The major weakness of an NIDS is the difficulty faced by it during handing high speed networks. The solution for this can be clarified by hardware based components ASIC (Application specific integrated circuits)[3],[4].but they are of high cost.

Vasiliadis, Polychronakis and Ioannidis [5] has proposed a new parallel architecture named multi-parallel IDS architecture (MIDEA) for controlling and handling high-performance processing and stateful analysis of network congestion. They proposed the parallel architecture for network traffic processing and analysis in three levels, which are: multiple GPUs ,multi-queue NICs and multiple CPUs, They manifest that processing speeds increased up to 5.2Gbit/**S** with zero loss in packets.

The first implementation of parallel architecture for NIDS [6]
had some drawbacks in sending algorithm which only classify the incoming packets only on the basis of the IP address. One of the major limits in almost all configuration was that, the incoming packets are only classified on the basis of ip address only. If we contemplate the Network Address Translation (NAT) mechanisms which hide whole the networks under a single IP address [7].Even the traffic going between two IP addresses may be huge for the capacity of an individual NIDS sensor, which can result in bottlenecks and limits to the architecture scalability.

Another parallel architecture is [7] with a load balancer with custom hardware .here the flow of packets are analyzed with a sensor. It uses hashing function for the load balancer. Depending upon the resulting hash values the destination are selected. The dispatching rules here are dynamically accepted by the balancer, and the already established connections were redirected to different sensor and make it difficult and impossible to make a stateful analysis.

Our approach is that to make a simple way to handle the high traffic by load balancing the incoming packets and their by give them to the NIDS arranged in parallel.

III. PROPOSED SYSTEM

This paper discuss about how we can handle the incoming packets from a high speed network to a snort without losing any of the packets.

Snort is one of the free cost system available today, which is an open source network intrusion detection system based on rule based IDS (intrusion detection system). It stores data's and information in form of text files. All the files are combined together and kept in a main file name "snort.conf".

Snort sniffs the incoming data from the packet stream and send them to the packet decoder and organize the packets and send them to the pre-processor. In pre-processor the received data are filters, organizes and modifies the UDP or TCP packets, port numbers as fast as possible for transferring it to the detection engine. As we already said that snort capture and drops the malicious packets. But it has one drawback that it cannot handle high

The main part of snort is the detection engine where detection of error or malicious packets are done. Depending upon the length of the packet the utilized differs for each inputs. In detection engine it compare the incoming packets with the rules in snort, if any packet doesn't match with the rules then the packet will be dropped

Logging and alert displays the contents detected in the received packets, it give and alert related to the detected error or malicious packet.

Output module controls the follow-up of the logging and alert system.

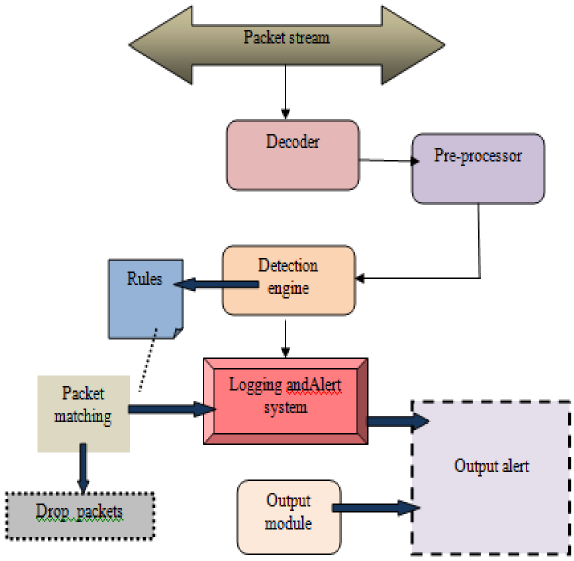

Fig1.Snort Architecture

Traffic in such cases it drops all the incoming packets. without checking or analyzing its contents. In such cases there may be chances for loss of important packets. So as to manage such situations is so important. To handle such situation we are introducing our parallel architecture for handling all the incoming packets by balancing the load and their by passing them to NIDS (snort).

IV. MODULE DESCRIPTION

A. Source queue

The source queue is the one who provide packets from the network to the disperser. The source queue captures the incoming packets and forward them to the disperser for the purpose of checking whether there is any kind of malicious contents are there within the packet.

B. Disperser

The disperser is the main part of our proposed system who manages the incoming packets to minimize the complexities. The disperser analyse the incoming packets and share them to the segmenter's which are directly connected to the disperser. Disperser shares the incoming packets equally to the segmenter. Here the disperser distributes the incoming packets to the snort in a round robin fashion.

C. Segmenter

The Segmenter captures the files and transfer it to the snort. Snort check and analyse the incoming packets with the help of rule set. If there found any kind of mismatch while comparing the incoming packets with the rule set, snort will reject the packet. By parallel architecture even for a high traffic snort

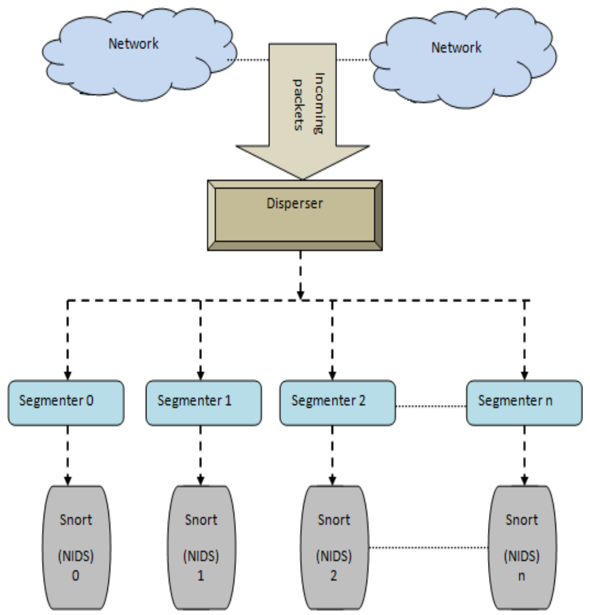

Fig 2.Snort Parallel Architecture

can handle the packets. Snort is directly connected to the segmenter and only a part of the incoming packets are given to each snort or the NIDS. To make the process like computations cost low, we tried to reduce the complexity as low as possible and simple, and also the incoming packets are equally distributed among the snorts or the NIDS so that we could easily balance the load.

V. CONCLUSION

There are different kinds of attacks, to categorize them and to handle them is very difficult especially in high traffics. In such cases we use NIDS for detecting the malicious activities. But it has also has some drawbacks, difficulty to handle the high speed and high traffic networks. So as to avoid such situation we introduce a concept in this paper. In this paper we proposed a snort with parallel architecture for managing the heavy traffic, which manages the heavy load by load balancing. It helps the snort to manage all the incoming packets, to analyse the packets, and to check whether there is any kind of rule violations by comparing it with the rule set. We didn't kept any kind of limit for adding the parallel architecture components. We can implement it with low cost. Future work will be based on how the receiving packets after the analysis of NIDS can be rearranged.

ACKNOWLEDGMENT

Its my pleasure to express my gratitude to the the people who have supported and helped me so much throughout this period for completing my work. Firstly I express my sincere gratitude to our respected principal and our Department head of GISAT for providing necessary facilities and support. I would like to thank my tutor, Mrs. Jyothi B. for her valuable guidance. You definitely provided me with the tools that I needed to choose the right direction and successfully complete my thesis work. I would also like to thank my parents for all their support. Finally, to my friends. Who supported me by helping me to solve my problems and supported my findings.

REFERENCES

[1] Michele Colajanni Micro Marachetti"A parallel architecture for stateful intrusion detection in high traffic networks"

[2] Waleed Bul'ajoul, Anne James, Mandeep Pannu." Network intrusion detection systems in high-speed traffic in computer networks" 2013 IEEE 10th International Conference on e-Business Engineering.

[3] "Top layer networks." [Online]. Available: http://tcpreplay.sourceforge.net

[4] "Juniper networks." [Online]. vailable: http://www.juniper.net

[5] Multiparallelintrusion detection architecture," Proc. 18th ACM Conf.Computer and Communications Security. ACM, 2011, pp. 297-308.

[6] Y. Jianying, Z. Jiantao, W. Pei, and T. Wang, "An application of networkaddress translation on gateway," in *Proceedings of the 2003 InternationalConference on Neural Networks and Signal Processing*, 2003, pp. 229–238.

[7] L. Schaelicke, K. Wheeler, and C. Freeland, “Spanids: a scalablenetwork intrusion detection loadbalancer,” in *CF ’05: Proceedings ofthe 2nd conference on Computing frontiers*. New York, NY, USA:ACM Press, 2005, pp. 315–322

.