Fault Torelant Distributed Intrusion Detection System Using Advanced Honeypots

Akshay Kulkarni, Sujal Raul, Vidhya Suresh

Department of Computer Engineering
St. Francis Institute of Technology
Mumbai, Maharashtra, India

***Abstract* – An Intrusion Detection System (IDS) acts as a network security tool and provides various approaches for detecting unauthorized activity, and have given us an insight into some of the problems which are yet to be solved. This paper proposes a Distributed Intrusion Detection System (DIDS) for private LAN's by using Honeypots and Fault Tolerance mechanism. The Architecture is customized by combining several Host IDS with different functionalities. Each IDS will perform a specific task and thus various types of attacks on the private LAN's can be avoided using this system. Central IDS will act as master and will be used to control the other IDS's. A Honeypot will help us to identify and learn about different types of attack because of the delusion the Honeypots create. In addition, a fault tolerance mechanism is used to provide better reliability in case of failure.**

Keywords: Intrusion Detection System, Distributed Intrusion Detection System, Honeypots, LAN's, Fault Tolerance.

I. INTRODUCTION

In today's era, Internet is used on such a scale which no one expected. The main component that forms the basic infrastructure of internet is the network. With the increasing use of internet in our day to day life, protecting these networks from intruders is essential in order to provide reliability and availability to users. Different approaches like Firewall's, Demilitarized Zone (DMZ) have been used but have not been that effective. IDS have thus emerged a solution to provide better security as compared to other methods [13]. IDS runs constantly in a system background, monitors computer systems and network traffic, and analyzes traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization [9]. System administrators rely on such tools to monitor and secure their network and systems. They detect inappropriate use or activity of a network or computer system by monitoring events and sending alerts when certain events, such as scanning network to determine connected computer systems occur.

Honeypots were introduced to monitor unused IP spaces to learn about attackers. The advantage of Honeypots over other monitoring solutions is to collect only suspicious activity. However, current Honeypots are expensive to deploy and complex to administrate especially in the context of large organization networks. Honeypots attracts the attacker and thus gives an invitation for attack. [5]

Fault tolerance is a critical point for long running parallel distributed applications executing in Massive Cluster of Workstations (MCOW). The long running applications demand a fault tolerance scheme that should be independent of cluster scalability. Fault tolerance becomes inevitable for certain systems as unpredictable failure can result in complete restart of system application. This makes the whole system inactive/unproductive for considerable amount of time. In order to prevent such loss, providing fault tolerance is necessary. [14]

II. BACKGROUND ON IDS AND HONEYPOTS

IDS and Honeypots can be classified into different types based upon their functionalities. The preceding part gives classification of IDS and Honeypots.

A. Types of Intrusion Detection System:

IDS can be categorized in to different types depending on many factors like type of information source, analysis strategy, time aspects, architecture and response. The source of information for IDS can be audit trails (e.g. system logs), network packets, application logs, wireless network traffic or sensor alerts produced by other intrusion-detection systems [3, 4]. An IDS 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.

The several ways to classify IDS are:

1) *Signature based vs. Anomaly based:* In Signature based, IDS has a database of signatures of attack. Each time the packet traverses through the network IDS checks it with database if not found then the packet is forwarded else blocked. If an attacker makes slight change to its signature the attacked might not get detected. In Anomaly based defines a baseline or normal, state of the network's traffic load, breakdown, protocol, and packet size. The anomaly based continuously monitors the system and checks for anomalies by comparing segments with baseline.
2) *Network-based vs. host-based systems:* In a network-based system, or NIDS, the individual packets flowing through a network are analyzed. The NIDS can detect malicious packets that are designed to be overlooked by a firewall's simplistic filtering rules. In a host-based system, the IDS examines at the activity on each individual computer or host.
3) *Passive system vs. reactive system:* In a passive system, the IDS detects a potential security breach, logs the information and signals an alert. In a reactive system, the IDS responds to the suspicious activity by logging off a user or by reprogramming the firewall to block network traffic from the suspected malicious source.

B. Types of Honeypots:

Honeypots are used to watch what attacker do, to lure an attacker to a place in which we may be able to learn enough about the attacker to identify and stop him and to provide lucrative but diversionary playground. It is used to monitor the system and the monitoring is done in such a way that the attacker doesn't know that he is being monitored.

Several ways to classify Honeypots are given in [11]:

1) *Low Interaction Honeypots:* Low Interaction Honeypots allow only limited interaction for an attacker or malware. All services offered by a these Honeypots are emulated. Thus they themselves are not vulnerable and will not become infected by an exploit attempt. They are usually easy to deploy and are used by corporations.
2) *High Interaction Honeypots:* These make use of actual vulnerable services or software. They are difficult to deploy because of their increased complexity. It provides a better insight of an attack or how a particular malware executes in real-time. It does not involve any emulated service. They are normally used by governments, research or military.

III. ARCHIECTURE OF PROPOSED SYSTEM

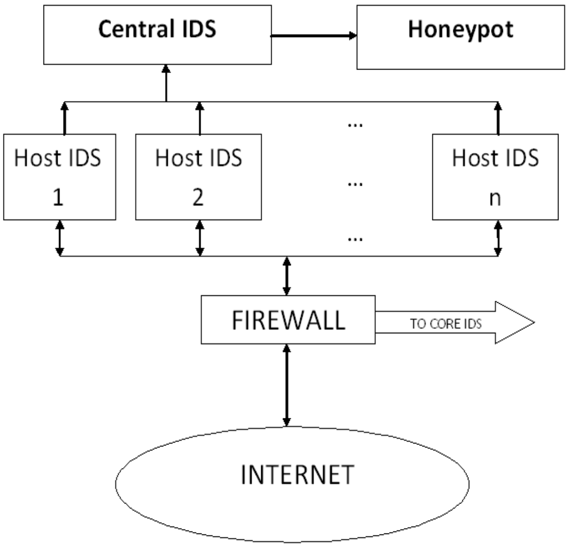

Fig. 1 Architecture of Proposed System

The Figure 1 above shows the general architecture of our proposed system. It consists of mainly three modules Host Intrusion Detection System (HIDS), Central Intrusion Detection System (CIDS), and Honeypots. The data packets forwarded to the LAN will be received by the intended host computer on which the HIDS resides. The HIDS analyzes the packets and acts according to the result of analysis. If it detects an unusual or malicious packet it forwards the packet to CIDS for further processing else it allows the host PC to have access to the packet. The CIDS determines the type of attack and stores the log of the packet in its database. It then forwards the packet to the Honeypot to deflect the attacker and thus protects the actual data stored in host PC from the intruder.

IV. HOST IDS ARCHITECTURE

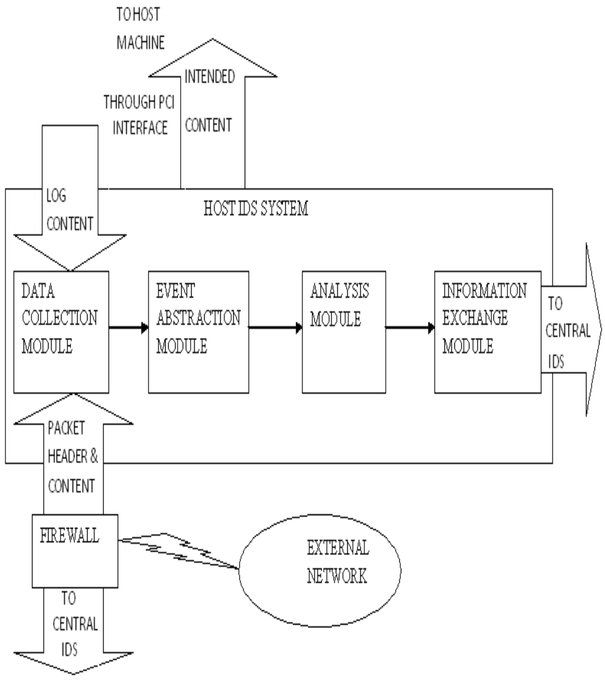

Fig. 2 Host IDS Architecture

The main function of HIDS is to perform a complete analysis of the packet. HIDS is a piece of code that is running in parallel with the host PC. Every host PC has the HIDS code installed in it. Host IDS limits the impact on host PC and offers limited opening to tampering and disruption of the actual IDS components. This module looks at the individual packets coming from the network and performs header search, content search and also looks for any signs of port scanning. This module also collects the attack information (audit trails or any application level event collector) from the application layer through PCI interface and further analyzes it. Figure 2 shows the block diagram of HIDS. HIDS mainly consists of four modules as follows.

1) *Data Collection Module:* This module collects the packet received from the firewall and also retrieves the data from the log stored in the host PC. The log contains the events of activities from the host computer's application layer.
2) *Event Abstraction Module:* This module is essentially used to extract data for further processing without making any changes to the actual form of the data. The packet collected by the data collection module is a datagram and the log is a record stored in the host PC. The event abstraction module extracts required data from both and provides this data to the analysis module.
3) *Analysis Module:* The Analysis module processes events according to some defined detection strategy. At least two detection methodologies are currently in discussion: misuse and anomaly detection. It seems to us that these methodologies are complementary. It is our goal to have a hybrid (misuse and anomaly) intrusion detection strategy. This module uses the detection strategy to differentiate between a malicious or suspicious

packet and a non-malicious packet. It then sends the conclusion of the detection process to the information exchange module.

4) *Information Exchange Module:* This module acts as an interface between the HIDS and the CIDS. It transfers the malicious or suspicious packets to the CIDS and the non-malicious packets to the host PC.

V. CENTRAL IDS ARCHITECTURE

1) Central Information Exchange Module: This unit collects the information from each individual HIDS. To connect each HIDS to the CIDS, most effective connection is Ethernet. Host can send and receive the information as TCP packets (for reliable transmission) or as UDP packets (for wireless transmission).

2) Packet Reassembly Module: It reassembles the packets from HIDS agents which require more sophisticated statefull analysis. Analyzing individual packet is not effective in some attacks such as port scanning. Reassembly and analysis prevents port scanning type of attacks.

3) Pattern & Rule Database: It maintains the records of certain suspicious attack patterns. If needed information is collected from other hosts and conclusions are made for the attacks. This module also contains rules which assist data mining and pattern matching algorithms. Some details can also be stored in external memory through external memory interface.

4) Central Analysis & Response Module: This module is the heart of CIDS and makes all the important decisions. This module collects the information from Central information exchange module, Packet reassembly module and Pattern & rule database, analyzes it and takes the respective decisions. The decisions are then communicated to each individual HIDS.

5) Rescue Element: Main concept behind rescue element is Provision of Fault tolerance. The proposed DIDS system is fault tolerant. The rescue element in CIDS provides fault tolerance to the system. In case any of the HIDS crashes, the firewall will forward the packet destined to that host PC to the CIDS. CIDS will pretend to be a host (crashed system) to firewall. A copy of the HIDS implementation (software code) resides in the rescue element of the CIDS. This part of the CIDS performs the task of the HIDS in the host PC until the HIDS in the host PC is restored. Assist manager monitors the traffic flow at the host machines. If any particular host is congested then the Assist Manager diverts some of the traffic from the host to the rescue element for packet analysis thus providing for congestion control to some extent. The Assist Manager provides the interface between the host PC and the CIDS. In case the packet forwarded by the firewall is concluded to be non-malicious, the Assist Manager forwards the packet to the intended host PC.

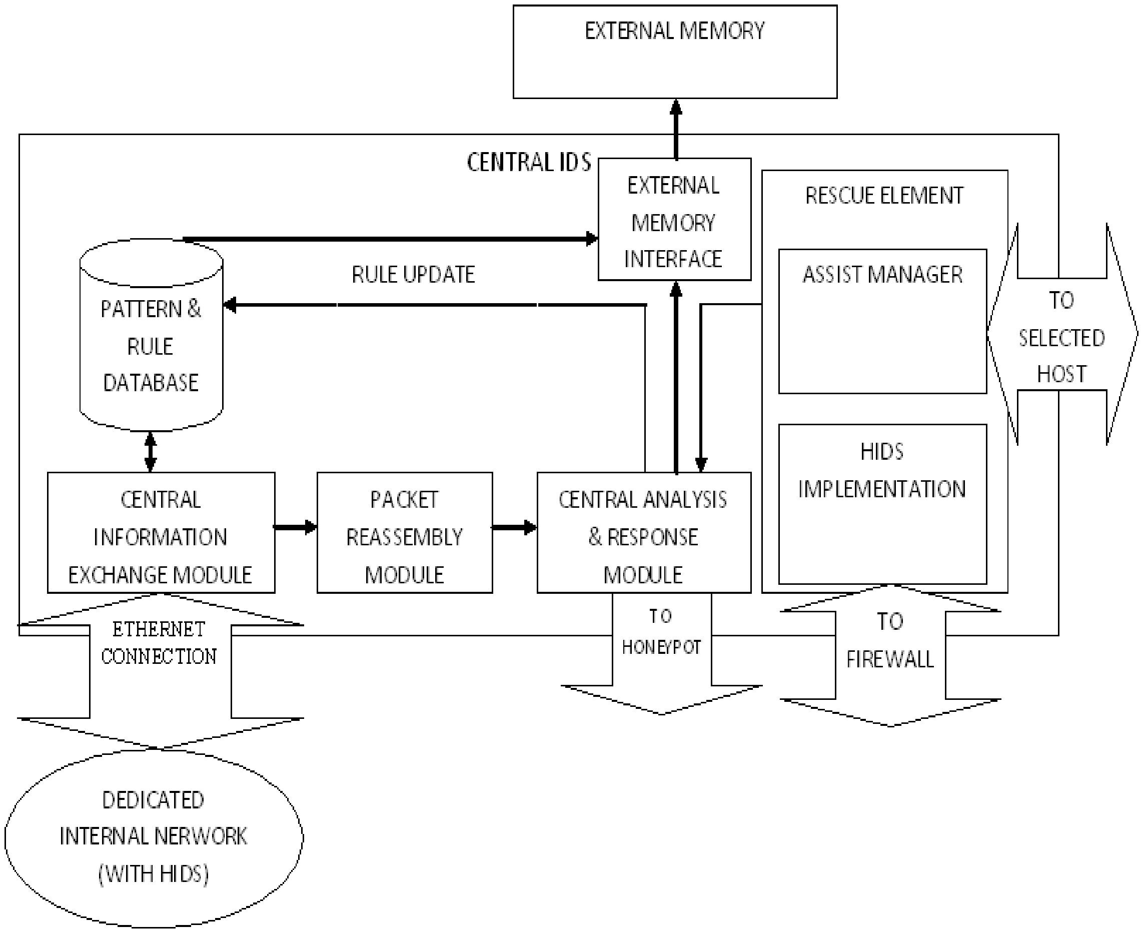

Fig. 3 Architecture of Central IDS

VI. HONEYPOTS

Variety of misconceptions about honeypots, everyone has their own definition. This confusion has caused lack of understanding, and adoption. Any security resource whose value lies in being probed, attacked, or compromised. A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.

In order to understand how Honeypots compromise themselves it is necessary to understand how network attacks are spread. For this purpose, an attack process is defined as a sequence of communication between an attacker and victim with malicious purpose. Hence, we divide the attack process in three phases [11]:

1) *Phase 1:* In the this phase, we need to find the victim. This can be done by attempting a communication with a specific service on host PC. One of the technique used to find the victim is to scan all the network addresses within a specific subnet. The next phase is reached only if the victim replies and the service is open to attacker.

2) *Phase 2:* Once the phase 1 is completed, the victims service is exploited by launching an attack payload. It is difficult to separate phase 1 and phase 2 because the most of the times the attack consists of single network packet and due to this the communication attempt and attack payload overlap. The attackers moves towards the next phase only if the service is sucessfully compromised based on the attack launched.

3) *Phase 3:* The attacker will now use the corrupted victims machine and may gain aceess to specific resource or a malware that is spreading from one machine to another. In such case, the malware can be installed on this corrupt machine and can be used to attack other machines. After this the phase 1 starts again.

Honeypot can perform three functions- detect, defect and learn attacks. In our proposed system honeypot must deflect attack and learn about attack. As shown in figure 4, Honeypot system receives information from CIDS. Information about unkown attacks is forwarded to leaning Honeypot. If attack is known information is given to deflecting or attacking Honeypot. [1, 7]

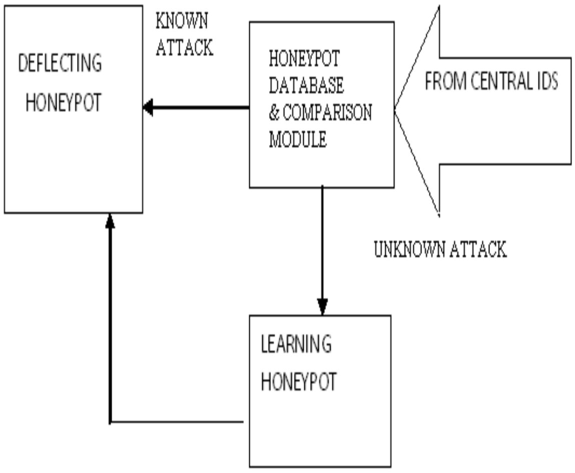

Fig. 4 Honeypot Architecture

VII. COMBINING HONEYPOTS AND IDS

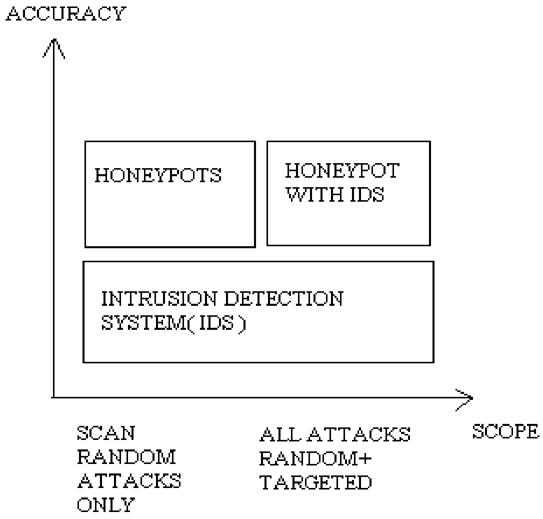

Fig. 5 Combining Honeypots and IDS

The above figure 5 shows a simple classification of Honeypots and anomaly detection systems, based on attack detection accuracy and scope of detected attacks. Targeted attacks may use lists of known (potentially) vulnerable servers, while the scan-based attacks will target any system that is believed to run a vulnerable service. Anomaly Detection systems can detect both type of attacks, but with lower accuracy than a specially instrumented system (Honeypot). However, Honeypots are blind to targeted attacks, and may not see scanning attack until after it has succeeded against the real server. Hence, we find a need to combine Honeypots and IDS.

VIII. CONCLUSION

We have proposed architecture of a Distributed Intrusion Detection System (DIDS) using Honeypot and providing fault tolerance and congestion control. In most of the existing DIDS's the role of the host IDS component is mainly passive i.e. they collect the events and forward it to the Central IDS for processing. In our proposed architecture the main processing load at each host is taken care by the Host IDS component which analysis the packet and forwards only the suspicious or malicious packets to the CIDS. Additionally, the rescue element in the CIDS provides for fault tolerance in the system. The system would function efficiently even if any of the host IDS crashes. Also, in case of congestion at any of the host PC, the Assist Manager in the rescue element of CIDS redirects the traffic to the CIDS to resolve the problem of congestion at the host PC.

IX. REFERENCES

[1] Christopher Hecker, Kara L. Nance, and Brian Hay, "Dynamic Honeypot Construction," Univ. of Maryland, Adelphi, MD, Proceedings of the 10th Colloquium for Information Systems Security Education, 5-8, June 2006.

[2] R. Puttini, J.-M Percher, L. Me, R. de Sousa,"A fully distributed IDS for MANET", *IEEE Computers and Communications,* vol. 1, pp. 331-338, July 2004.

[3] Ashok Kumar Tummala, Parimal Patel, "Distributed IDS using Reconfigurable Hardware", *IEEE Parallel and Distributed Processing Symposium*, pp. 1-6, March 2007.

[4] Charles P. Pfleeger and Shari Lawrence Pfleeger, *Security in Computing*, 4th ed., Pearson Education, Oct. 2006.

[5] (2007) The Honeypot Website. [Online]. Available: http://www.honeypots.net/

[6] J. Sen, I. Sengupta, P.R. Chowdhury, “An architecture of a distributed intrusion detection system using cooperating agents”, *IEEE Computing & Informatics*, pp. 1-6, June 2006.

[7] I. Kuwatly, M. Sraj, Z. Al Masri, H. Artail, “A dynamic honeypot design for intrusion detection”, in *Proc. IEEE/ACS*, 2004, paper 10.1109, pp. 95-104.

[8] H. Sallay, K.A. AlShalfan, O.B. Fred, “A scalable distributed IDS Architecture for High Speed Networks”, *International Journal of Computer Science and Network Security*, vol. 9 No. 8, pp. 9-16, August 2009.

[9] John Carroll, *Computer Security*, 3rd ed., Butterworth-Heinemann, 1997.

[10] LU Zhi-Jun, ZHENG Jing, HUANG Hao, “A Distributed Real-Time Intrusion Detection System for High-Speed Network”, *in Proc. ICCRD*, 2011, pp. 667-673.

[11] Lance Spitzner, *Honeypots: tracking hackers*, 2nd ed., Addison-Wesley Professional, 2003.

[12] I. Mokube, M. Adams,”Honeypots: concepts, approaches and challenges”, in *Proc 45th ACM-SE*, 2007, paper 10.1145, pp. 321-326.

[13] William Stallings, *Cryptography and Network Security: Principles and Practice*, 2nd ed., Prentice-Hall, 2000.

[14] J. Sen and I. Sengupta, “Autonomous agent-based distributed fault-tolerant intrusion detection system”, in *Proc. of the ICDCIT*, 2005, pp. 125-131.

[15] J. Fraga, F. Siqueira, F. Favarim,”An adaptive fault-tolerant component model”, in *Proc. Object Oriented Real-Time Dependable Systems*, 2003, pp. 179-186.