International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 8, October - 2012

Investigation And Analysis Of High Performace Multi Agent Anomaly Detection System In Backbone Networks

Dr. Prof.P.K.Deshmukh Dr. Prof. A.B.Bagwan, Mrs. D.S.Naigaonkar,

I. ABSTRACT

Many researchers proposed IDS to collect the network data and use the information from known types of attacks in order to detect the network hacking or attacks or anomalous activities. Using intrusion detection methods, we can collect and use information from known types of attacks and find out if someone is trying to attack your network or particular hosts. Recently mobile agent based new IDS technique proposed for distributed networks, but according to literature survey it is clear that most MA-based IDS that are available are not quite effective because their time to detection is high and detect limited intrusions. Thus in this paper we are proposing the Java agents based IDS system for efficient and effective intrusion detection over distributed network environment. The experimental result claims that proposed approach is more effective and efficient as compared to IDS systems previously proposed.

Index Terms— IDS, SNORT, JAVA Agents, Multi Agents, Mobile nodes, Bro database.

II. INTRODUCTION

Due to the remarkable growth in network based services as well as sensitive data communication over networks, security of networks is becoming more important. Now day's security is always major concern over the all kinds of networks used for enterprise environments. There are many attempts applied successfully by hackers and intruders in order to break the networks of high profile company and web services. To prevent such threats, different kinds of methods have been proposed over Internet such as encryption, firewalls, virtual private networks etc. In addition to these one more new technique is started appearing from last few years called as Intrusion Detection System (IDS) [1, 2].

Hence the intrusion is nothing but the set of actions which deals with compromising the integrity, availability of resources or confidentiality of resources. Whenever the security vulnerabilities exploited by the attacker or attackers, this means that intrusion takes place and hence the CIA guarantees of system is violated. Hence the mechanism of intrusion detection is required for providing the additional security approach for securing the networks. This is way of detecting the illegitimate end users, vulnerabilities and attacks which could affect the basic computer systems functionality. However, IDS systems are having the following drawbacks:

(i) Delay of time.
(ii) A single point of failure.
(iii) Limited scalability.
(iv) Hard to communicate mutually between different IDSs

In order to solve the aforementioned shortcomings, mobile agent technology is currently applied to IDS. Mobile agent is a particular type of software agents which has the capability of moving from one host to another. It is an autonomous program situated within an environment, which senses the environment and acts upon it using its knowledge base to achieve its goals. Mobile agent is of the features of reducing network overload, overcoming network latency, synchronous and autonomous execution, robustness and fault-tolerance, system scalability and operating in heterogeneous environments. To this end, MA technology is very suitable to solve intrusion detection in a distributed environment (Chan & Wei, 2002), hence the advent of Mobile Agentbased IDS (MA-IDS) [3].

MA-IDSs are also faced with some shortcomings such as:

a. *High time to detection:* MA solutions may not be fast enough to meet the needs of IDS One of the major challenging problems facing MA-IDS is improving the speed with which they can identify malicious activities.

b. *Performance:* though MA technology has improved greatly on detection performance, but effective detection of autonomous attacks is still very low. Also, agents are often written in scripting or interpreted languages, which are easily ported between different platforms. Their mode of execution is still very low compared to native codes (Kruegel and Toth, 2002) [4].

c. *Security:* Another major problem is protecting the protector (MA-IDS) from attacks.

IJERT

Hence, in this paper discussing about use of the different agents for different purpose for distributed systems rather than using single mobile agent in order to improve the effectiveness and efficiency of intrusion detection system. Here we are using SNORT and Bro databases for experiment purpose.

III. RESEARCH OBJECTIVES

As we described above, our main aim behind this research is to present the agent based IDS systems with improved effectiveness and speed. Apart from this following are objectives of study:

a) To develop intrusion detection system using java agents.

b) To protect secure information of an organization from outside and inside intruders

c) To provide a secure environment for work.

d) To decrease the bottleneck in the main server by distributing the sensors in the particular hosts.

e) The log of the intrusion detection system can be used for forensic purpose to find out the culprit.

f) To compare the performance of proposed system and existing systems using SNORT and Bro.

IV. EXISTING METHODS AND TOOLS OF IDS

The recent research on detecting and eliminating the rough access point based master and slave agents is given below along with its limitation.

1. HIDS

Using OS auditing mechanisms: e.g. BSM in Solaris logs all direct and indirect events generated by a user; trace monitors system calls made by a program. Monitoring user activities: analyzing shell commands. Monitoring executions of system programs e.g. send mail's system calls.

Advantages

- Can detect attacks that cannot be seen by NIDS.
- Can operate in an environment in which network traffic is encrypted.
- Unaffected by switched networks.
- Can help detect Trojan horse or other attacks that involve software integrity breaches.

Disadvantages

Since at least the information sources reside on the host targeted by attacks, the IDS may be attacked and disabled as port of the attack are not well suited by detecting network scans or other such surveillance that targets an entire network. Since they use the computing resources of the hosts they are monitoring, therefore inflicting a performance cost on the monitored systems [6, 7].

2 NIDS

Using packet sniffing. Looking at IP header as well as data parts.

Disadvantages of Network-Based IDSs:

NIDS may have difficult processing all packets in a large or busy network and therefore, may fail to recognize an attack launched during periods of high traffic. Modern switch-based networks make NIDS more difficult: Switches subdivide networks into many small segments and provide dedicated links between hosts serviced by the same switch. Most switches do not provide universal monitoring ports NIDS cannot analyze encrypted information. Most NIDS cannot tell whether or not an attack was successful.

Where IDS should be placed in Network Topology

Depending upon your network topology, you may want to position intrusion detection systems at one or more places. It also depends upon what type of intrusion activities you want to detect: internal, external or both. For example, if you want to detect only external intrusion activities, and you have only one router connecting to the Internet, the best place for an intrusion detection system may be just inside the router or a firewall. If you have multiple paths to the Internet, you may want to place one IDS box at every entry point. However if you want to detect internal threats as well, you may want to place a box in every network segment. In many cases you don't need to have intrusion detection activity in all network segments and you may want to limit it only to sensitive network areas [6]. Note that more intrusion detection systems mean more work and more maintenance costs. Your decision really depends upon your security policy, which defines what you really want to protect from hackers. Following figure 1 showing the same:

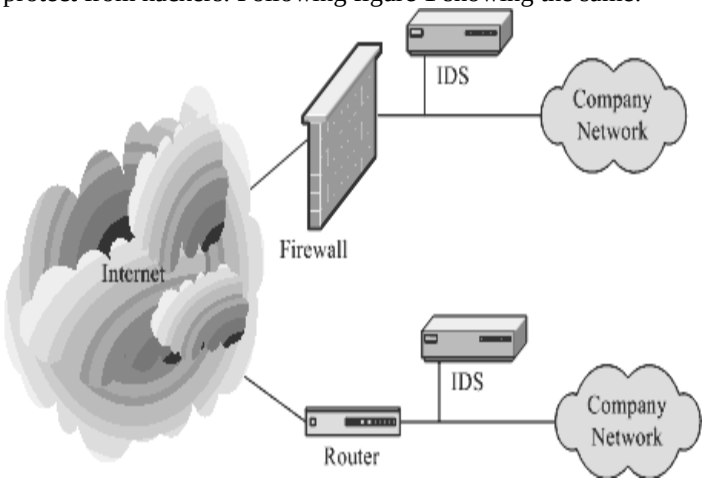


Figure 1: Network Topology with IDS

Tools used by intrusion detection systems

One of the most widely deployed intrusion detection systems is SNORT, a *misuse* detection system that has been used in the APHIDS system described later. An IDS function first generates a log to keep record of an attack once it detects an intrusion. After that, an IRS (Intrusion Response System) takes over. This is helpful in cases when AN ID generates more alerts than a human can handle.

Many vulnerability assessment tools are also available in the market that can be used to assess different types of security

IJERT

holes present in your network. A comprehensive security system consists of multiple tools, including Snort, bro, Emerald, STAT, Enterasys Dragon, McAfee Intru- Shield [8].

V. PROPOSED IDS ARCHITECTURE

Following figure 2 shows the architecture of proposed IDS system using monitor agent, response agent and sensor agent.

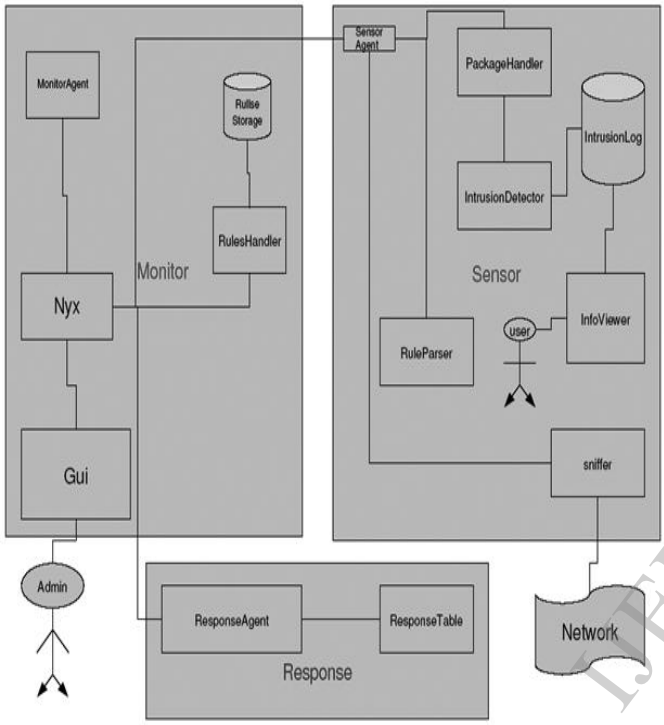


Figure 2: IDS based on MA: System Architecture

We want to grant snort and bro at the same time the ability to be social ID. To understand this in clear let us suppose you have installed Snort or Bro at your site in India, and let other instance installed in Iraq. Initially, both will have their own rules and each will start monitoring and stopping intrusions, after a while each one will get an experience in dealing with some sort of signatures or network events that the other might not.

What would be the case if there is a way that every Snort instance be able to exchange experience with other instances.

The agents that compose the proposed system are as follows:

1. Detection agents: These agents determine whether an access is an intrusion, and if it so then notify the other agents so that they can take action accordingly.
2. Response agent: If any intrusion is detected, these agents handle them by stopping the connection of the attacker and they inform the system security manager.
3. Evidence agent: These agents keep or record the user information so that they can use it as an evidence to track an intruder.
4. Prevention agent: The major task of these agents is to prevent intrusion in the system
5. Sensor agent: Works like packet sniffer
6. Interface agent: Integrates all the above mentioned agents so that they can co-ordinate with each other.
7. Communication agent: It works like a connection between the snort and the system.

In this architecture here main process is to create SNORT integration environment ontology where Java Intelligent agents or other third party can communicate and update snort rule file autonomously and intellectually. Java agents will work autonomously in the network environment and will monitor network status for certain suspicious events (spoofing, flooding… etc) and eventually add rule to SNORT rule file.

VI. PRACTICAL APPROACH

We are using the Java platform to implement proposed approach for real time dataset. We are here considering the real time bank dataset for training and testing purpose. We are basically using following tools for implementation:

- Bro Systems
- SNORT
- Aglets Software Development Kit (ASDK)
- API Java Aglet (J-AAPI)

IDE used:

- Netbeans
- Myeclipse

Following use case diagram snapshot showing the implementation approach which we are using for this proposed IDS system. There are mainly two parts of IDS. First one is the core system where the entire process takes place while second one is the web based GUI for the easier use for administrator.

This proposed multi agents approaches are gets managed according to the incoming traffic in the network. This multi agents effectively increases the detection rate.

IJERT

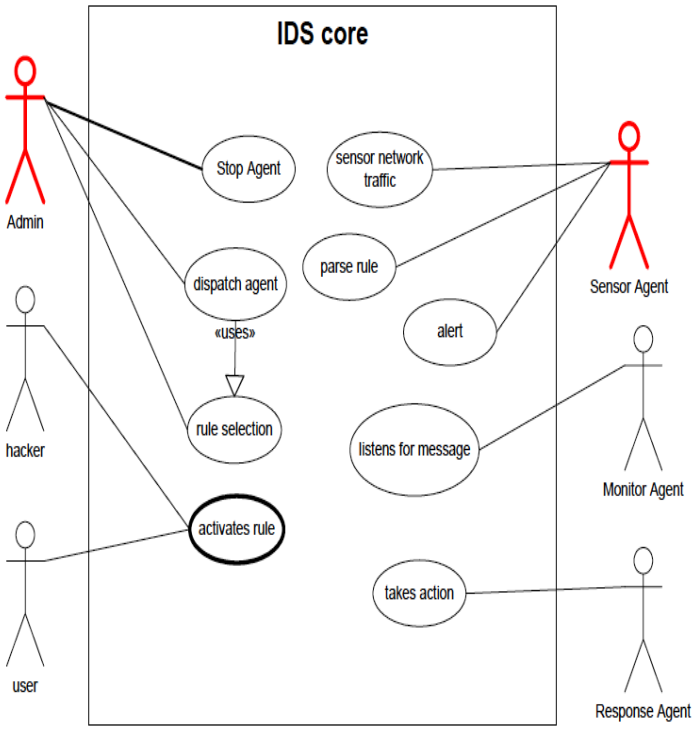


Figure 3: Use case 1: IDS System using Java Agents

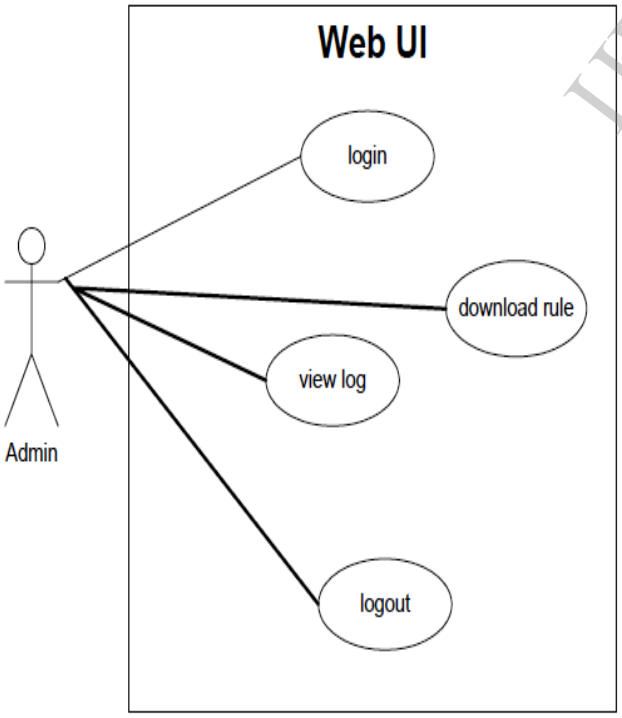


Figure 4: Use Case 2: Web Based GUI

Results and Screenshots

Here we start capturing packets which are downloaded in our system. Following figure 5 shows hot to capture network packets using capture agent and sensor agent for capturing and sensing data respectively when any packet comes from network. It Sense that packet and pass to that packet to next agent for further process.

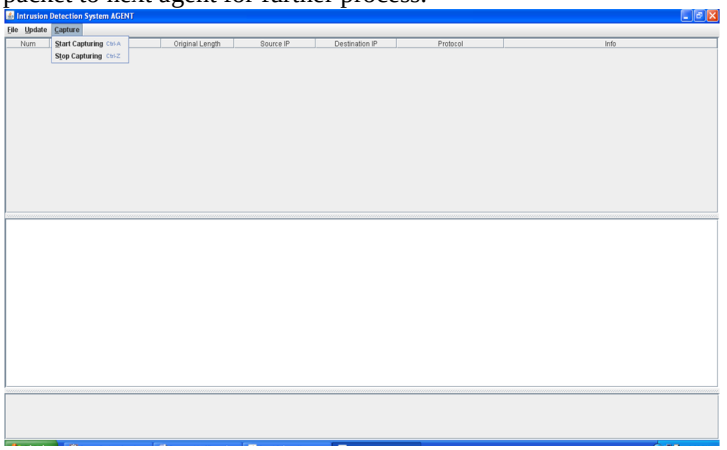


Figure 5: Packet Capturing and Sensing Process

Before Capturing Packet select the snort rule which apply on that incoming packets. When applying rule Detection Agents are use for determine whether an access constitutes an intrusion, and if there is any intrusion detected then they give notice to other agents. Figure 6 showing the selection of snort rule for detection process.

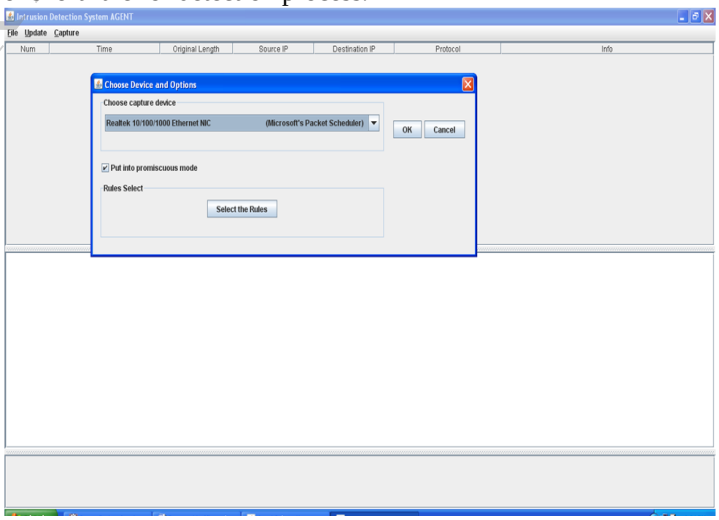


Figure 6 Snort rule section window

Here in figure 7, select particular rules from rule selection box and after selecting rule click on start button, which resulted into the packet capturing, sensing, detecting processes started using respected agents.

IJERT

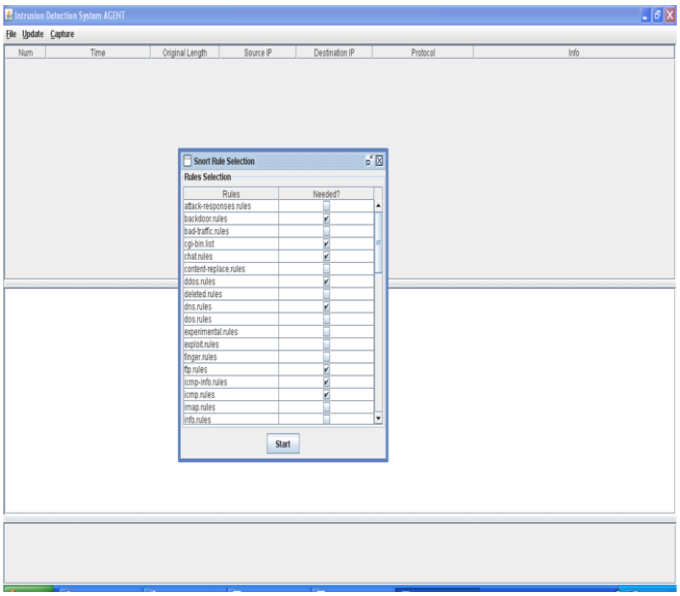


Figure 7 Rules selection

Figure 8 and 9 are showing continues packet capturing process. One by one capture packet shown in table format. Here an Evidence agent collects data that can be used as evidence to incriminate the intruders. Conversation agents used for communication between Sensor Agents and Evidence Agents.

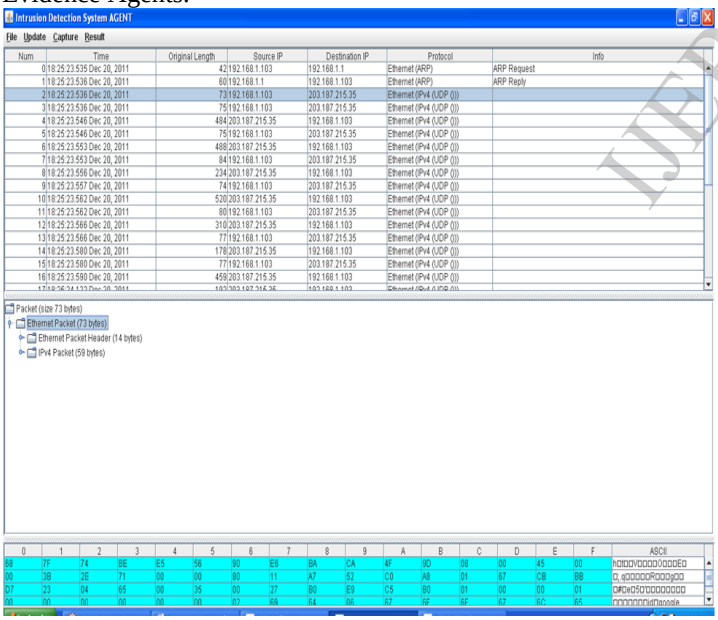


Figure 8: Packet Capturing and Evidence Agent

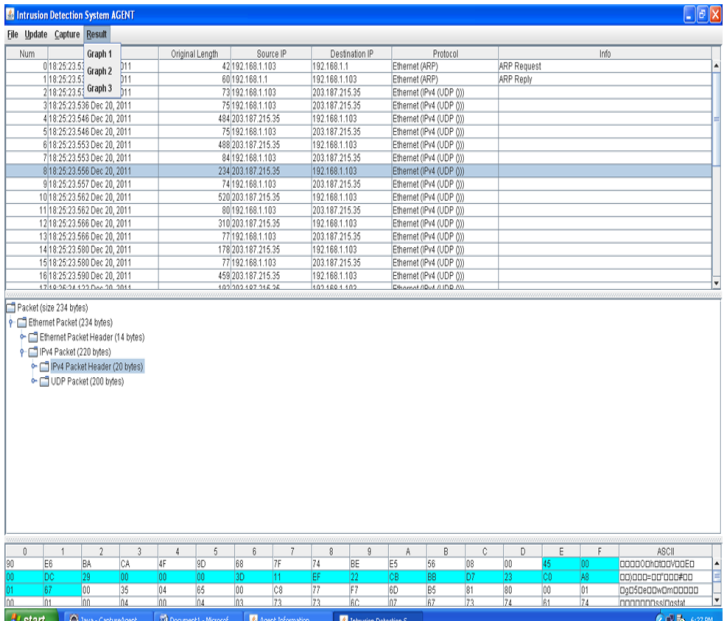
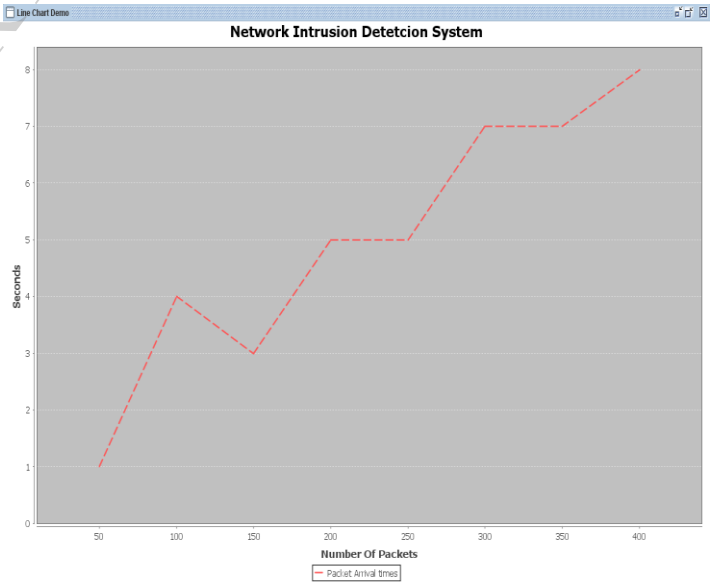


Figure 9: Packet Capturing and Evidence Agent

Graphs

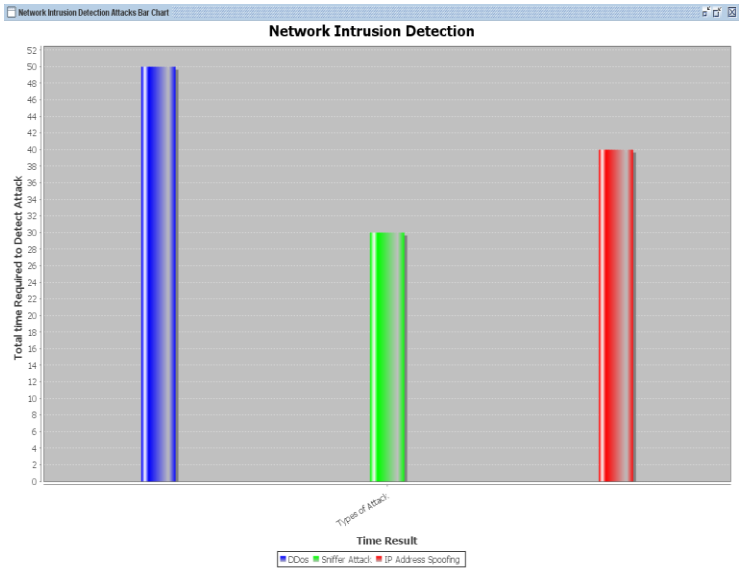
Here Display result in Graph format: First graph (graph 1) shows the capturing Packets from network. X-axis represent packets in byte and Y-axis represent time in Seconds.



Graph 1: captured packets per second

This Graph (graph2) show the information about Attack and total time required for detecting that attack:

IJERT



Graph 2: Comparative Analysis

VII. CONCLUSION

Thus as per the above discussion, here detailed description of new IDS framework is presented which based on multi agent approach as well as SNORT approaches in order to form efficient NIDS system which will effectively and efficiently deals with misuse from intruder. Here real time scenario considered and the system learning and updating the previously observed user behavior for detecting and preventing intrusions. The statistical IDS system presented for the same. For this approach expected results from this approach showing the viability for the intrusions detection and this results are compared with previously executed IDS approaches in order to outperform those methods. For the misuse detection special rule generated is implemented in order improve the accuracy and speed. Here for effectiveness we used the SNORT rules and those are updated automatically.

VIII. REFERENCE

[1] "D-SCIDS: Distributed soft computing intrusion detection system", Ajith Abrahama,, Ravi Jainb, Johnson Thomasc, Sang Yong Han, 28 June 2005.

[2] Almgren, M., Barse, E. L. & Jonsson, E. (2003). Consolidation and evaluation of IDS taxonomies. *Nordic Workshop on Secure IT Systems (NordSec 2003)*, pgs 57 -70, Norway, Oct. 2003.

[3] Anderson, J. P. (1980). *Computer security threat monitoring and surveillance*. Technical report, James P. Anderson Co., Box 42, Fort Washington, February 1980.

[4] Bace, R., & Mell, P. (2002). *Intrusion detection system*. Technical report, NIST Special Publication on Intrusion Detection, 2002.

[5] Chan, P. C., & Wei, V. K. (2002). Preemptive distributed intrusion detection using mobile agents. *Proceedings of 11th IEEE International Workshops on Enabling Technologies*. June, 2002, pp. 103 – 108.

[6] Deeter, K., Singh, K., Willson, S., Filipozzi, L., & Vuong, S. (2004). *APHIDS: A mobile agent- based programmable hybrid intrusion detection*.

[7] Paxson, V.: Bro: A system for detecting network intruders in real-time. Computer Networks 31 (1999) 2435–2463

[8] Vigna, G., Kemmerer, R.A.: Netstat: A network-based intrusion detection system. Journal of Computer Security 7 (1999).

[9] Chandra, Satish and Peter J. McCann, ''Packet Types,'' *Second Workshop on Compiler Support for Systems Software (WCSSS)*, May, 1999.

[10] Paxson, V., ''Bro: A System for Detecting Network Intruders in Real-Time,'' *USENIX Security*, 1998.

[11] Roesch, Martin, ''Snort – Lightweight Intrusion Detection for Networks,'' *13th Systems Administration Conference*, USENIX, 1999.

IJERT