Misuse Detection System based- Snortrules-JESS Using Multiagents

Asmaa shaker ashoor , prof. Sharad Gore

Abstract : In this work, we propose a novel Network Intrusion Detection Systems (NIDSs) architecture utilizing the misuse detection approach. This Network Intrusion Detection System architecture utilizes misuse detection agent. The proposed misuse detection agent adopts the novel framework by using Java Expert System Shell (JESS) and Snort rules along with the integration of multi agents. This approach achieves efficient misuse detection by detecting various types of network attacks and improves system performance. This approach introduces the framework for the network database security by implementation of a real time monitoring system using multi-agents. The NIDS uses the set of rules which defines the misuse behavior of user. This rule generation system is used based on JESS and Snort rules in order to use the rules for well known attacks and then taking the further decisions depended on multi-agents before intrusion occurs.

Keywords: NIDS, Misuses detection, Intrusion, JESS, Snort rules, Multi-agent

1. Introduction

With the incredible growth in the internet users day by day, the network intrusion continues to be a threat that intimidates almost all organizations. And to avoid the dangers resulting from these threats, a different approach should be followed by an expert that is to think like an intruder to know how they attack systems, thus security experts can build an effective network security solution. Another important aspect is to know what will protect, because the accurate knowledge of the protected environment provide the preemptive security measures, and enable detecting threats in order for it to be network security successful in preventing information loss, should follow three basic principles: *integrity, confidentiality, availability.* Intrusion in a particular network is not a simple operation; it passes by different techniques and layers, but generally all attacks follow a general life-cycle to bypass the many layers standing between the attacker and its victim. Intrusion detection (ID) is a part of the solution for protecting today's networks. IDSs are used to improve system security by detecting attacks and intrusions, an IDS system is a defense system, which detects hostile activities or exploits in a network. IDS systems are having the following drawbacks (Delay of time, a single point of failure, Limited scalability, Hard to communicate mutually between different IDSs). We solve this drawback; by implementing multi-agent-based IDS. In this architecture, a set of agents monitors specific aspects of a machine requiring security.

[1] *Asmaa shaker ashoor, Department of Computer Science, Pune University – Pune- India, asmaa_ zaid218@yahoo.com*

[1] *prof. Sharad Gore, Department of Statistic, Pune University – Pune- India, sdgore@stats.unipune.ernet.in*

2. Aims and Objectives

Our main aim behind this research is to present the multi agent IDS framework for real time dataset based on JESS

and Snort rules, in order to improve alarms rates and performance. Use of multi agents can be deployed to implement security solutions to achieve advantageous operation. Some objectives for the proposed system are: To develop Network intrusion detection system using multi agent architecture, to protect secure information of an organization from outside and inside intruders, to evaluate the performance of proposed research work in comparison with existing IDs as per literature survey. To overcome the problem of network intrusion detection in case of real time approach.

3. Methodology of proposed system

- **JADE**

Java Agent DEvelopment framework is an open-source Java-based environment to build and host agent-based systems. It follows FIPA standards and contains libraries that provide different levels of communication and control functions that can be used to define the behavior of agents. JADE supports an ACL (Agent Communication Language) message exchange layer as well as a basic ontology that can be extended for specific applications. JADE lacks support for intelligent agent functions but, given that it is Java based software, it can interact with other systems that do provide specialized functionality. JADE models can be distributed across several machines running different operating systems as support for agent location and mobility is also available. A GUI allows users to visualize agents, message exchanges, and the status of other system components. This simplifies debugging and allows for a more intuitive understanding of the agent platform.

- **Java Expert System Shell (JESS)**

It is a rule engine and scripting environment written entirely in Sun's Java language by Ernest Friedman-Hill at Sandia National Laboratories in Livermore, CA. Its powerful scripting language gives you access to all of Java's APIs. It was originally inspired by the CLIPS expert system shell,

but now, it has grown into a distinct Java-influenced and rule-based environment. It provides a tool to develop systems with intelligent reasoning abilities. It has a fast and efficient algorithm which is called Rete algorithm. Rete algorithm can build a network of pattern-matching nodes to solve problems with rules. First, it will do the pattern matches, then use a set of memories to store the information about the results of the matches, and then give out the available matches. During all the available rule engines, JESS is very small, light, and is also one of the fastest engines. JESS is written in Java language which is easily to integrate with other Java based techniques such as JADE etc. Using JESS, the user can design rules according to using knowledge, then build Java software to reason the rules.

- **Snort Rules**

Snort is one of the most popular signature-based IDS in use at present. It is an open-source packet sniffer/logger and network IDS. It analyses the packets that arrive to the network interface, trying to match their characteristics with those contained in the rules stored in its rule base. If a specific packet matches the premises of any rule, this rule is executed and a specific action is generated to give notice of this fact. Snort has different set of rules for each set of specified signatures, wherein the snort signatures are based on specified types of attacks. The rules in Snort have the following structure:
1 . Rule header: Contains the basic information about the rule, including:
Rule action: The action that will be taken when rule conditions are met. The main actions are: alert (generate an alert), log (log the packet) and pass (ignore the packet).
Protocol: The protocol used by the packet being analysed. Currently, Snort understands the following protocols: IP, TCP, ICMP and UDP.
Source information: IP address and port of the source computer from where the packet originated. It is also possible to use the key word 'any' to apply the rule on all packets irrespective of the IP address or port number.
Destination information: IP address and port of the destination computer in the packet. The keyword 'any' can be used again with the same meaning as before.
2. Rule options: Contains alert messages and information on the parts of the packet that should be inspected to determine if the rule action should be taken.

4. Architecture of Misuse Detection Agent

This system is able to provide the accurate protection to network against network attacks, Before building a network security system, security experts must define the defense strategy, system objectives, and the used techniques to develop the system. These specifications enable developing a system that is able to achieve its objectives with a high degree of performance and which enables identifying and blocking attacks at earlier stages before propagating over the protected network. The general architectural NIDS-misuse detection system framework, consists of multi agents:

Misuse detection agent: The main objective of this agent to detect attacks, according to the detection strategy and the agent raises an alert to the other agents , if there is a similarity between the filtered packet and signatures, and then removes these anomalous packets from further analysis, this agent Consists of three main agents(detection agent and central agent and analysis agent).

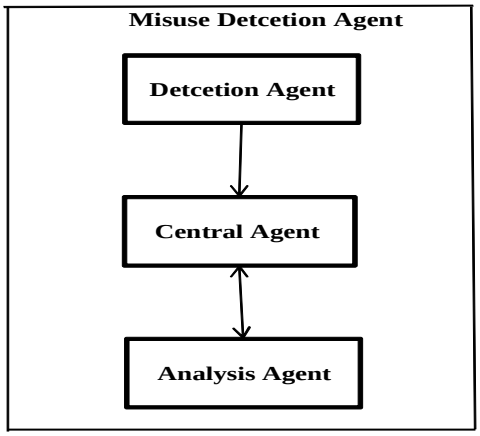

Figure.1. Illustrates Misuses Detection Agent Architecture

1. *Detection agent:* This agent captures packets from the network using a packet sniffer, which scans the packet and creates the knowledge base of the information it contains. This agent consists of:

Packet sniffer: A packet sniffer is a program that allows eavesdropping on traffic traveling between networked computers. The packet sniffer will capture data that is addressed to other machines.

Knowledge base: This base is the input to the expert system components where they are converted into facts using the constructs provided in JESS. These facts are then matched against to the rule-based intelligent Jess inference engine if any rules matches against the facts then the rule if fired and the send to the central agent, and the rules it is update regularly. The purpose of the use pattern matching between data and the stored rules to provide a high-level performance to a given other agents, and promotes weak coupling between the other agents and the system. When executed, the rules result sent to the central agent.

Snort Rules: Snort is an intrusion detection system which is signature-based [1]. We are implementing the snorts rules. In the misuse detection agent, the comparison is made between updated snorts rules with the incoming facts (received packets at packet sniffer) by using Java Expert System Shell [5]. The rules database is loaded into JESS (Rule Engine) along with incoming packets.

Jess Rule Engine: JESS is a rule-based programming environment, a rule engine and scripting environment. It has a fast and efficient algorithm called Rete algorithm.

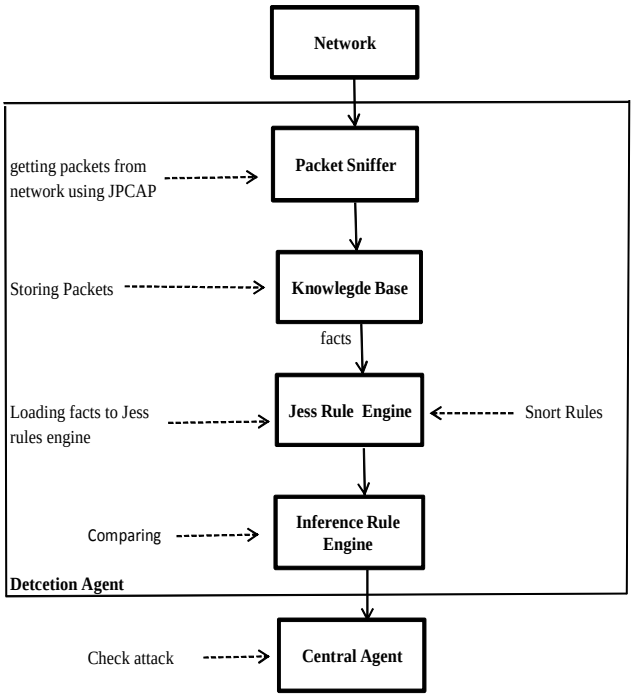

Figure. 2. Illustrates Detection Agent Architecture

2. *Central agent:* this agent is intermediate between detection agent and other agents, it is act as data source

3. *Analysis agent:* this agent includes:

Conversion Agent: It acts as communication bridge between central agent and multi agent, and providing them information required.

Response Agent: Provides notifications to UI agent and it can stopping the connection of the attacker, and send the informing to alarm agent to give out corresponding alarms.

Alarm agent: This agent sends notification to user/ System Administrator if any attack found.

Registration agent: It stores the results of the intrusion detection system into databases. This agent logs all the intrusion information includes (source IP and port, destination IP and port, type of attack, packet type, date and time) to be sent user interface agent.

User interface agent: this agent interact between the users with the system (e.g., system administrators), the aim of integrating them flexibility into the multi-agent architecture.

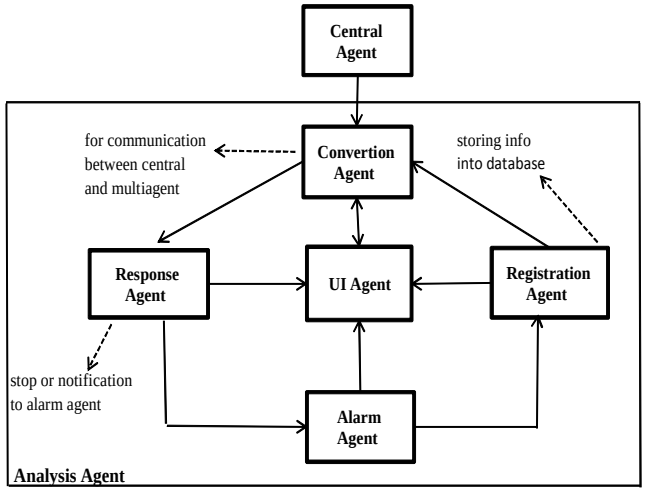

Figure.3. Illustrates Analysis Agent Architecture

5.Evolutions of Results

The Proposed Multi-Agent System (MAS), utilizes multiple interacting intelligent agents. We are using Java Agent Development framework (JADE)[7] for creating agents. The improved Network Intrusion detection system we are adopting approach of misuse detection system , wherein the misuse detection agent leverages JESS Rule engine and Snort Rules [1] for detection of known attacks. The primary objective of proposed research work is development of an improved Network Intrusion Detection System which achieves misuse detection without compromising on system performance and alarms rates and detect attacks. Also improved Network Intrusion Detection System is to achieve less complex structure and faster system response time by implementing Multi-agents.

Protocol	Capture Packet	Misuse Detection	Total perccentage
TCP	20	5	20
UDP	100	20	20
ICMP	10	3	30
IP	10	6	60

Table.1. Protocols types for misuse detection

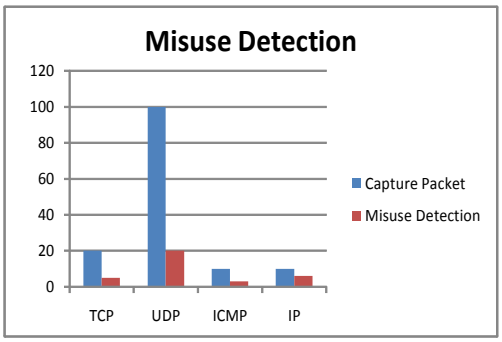

Figure4. Ilustrates misuse detection and capture packet

5. Conclusion

Thus as per the discussion in above points, here detailed description of new NIDS framework using misuse detection system, is presented which based on multi agent approach as well as JESS and Snort rules, in order to form efficient NIDS system which will effectively and efficiently deals with misuse from intruder. So; using multi-agent technology in developing Intrusion detection Systems provides many features that improve the performance of these systems. Collaborative multi-agent between them and information sharing may thus improve the overall rate of detecting intrusions. For the misuse detection system used JESS and Snort rule generated is implemented in order to detect known attacks and improve performance and alarms rates.

Bibliography

[1] "Snort Rule " http://www.snort.org/snort-rules/

[2] Allam Appa Rao, P.Srinivas, B. Chakravarthy, K.Marx, and P. Kiran "A Java Based Network Intrusion Detection System (IDS)", Andhra university college of engineering , India, proceeding of the 2006 IJME-INTERTECH Conference.

[3] Guy Helmer, Johnny S.K. Wong , Vasant Honavar, Les Miller, Yanxin Wang " Lightweight agents for intrusion detection", The Journal of Systems and Software 67 (2003) , http://www.cs.iastate.edu/~honavar/Papers/jss-lightweight.pdf

[4] M. Benattou, and K. Tamine " Intelligent Agents for Distributed Intrusion Detection System ", World Academy of Science, Engineering and Technology 6 2005 http://www.waset.org/journals/waset/v6/v6-45.pdf

[5] JESS Rules, http://www.jessrules.com/

[6]RATE Algorithm,
http://herzberg.ca.sandia.gov/docs/52/rete.html
http://www.perada.eu/documents/articles-perspectives/multi-agent-systems.pdf

[7] Fabio Bellifemine1, Agostino Poggi, and Giovanni Rimassa " Developing Multi-agent Systems with JADE",2004.
http://www.abdn.ac.uk/~csc232/teaching/CS4027/abdn.only/jade_book.pdf

[8] Aijaz Ahmed, signature-based network intrusion detection system using JESS(SNIDJ),master thesis,2004.

[9] E. Mosqueira-Rey, A. Alonso-Betanzos, B. Guijarro-Berdiñas, D. Alonso-Ríos and J. Lago-Piñeiro A Snort-based agent for a JADE mulit-agent intrusion detection system, Int. J. Intelligent Information and Database Systems, Vol. 3, No. 1, 2009.[12]