Military Technical College
Kobry El-Kobba
Cairo, Egypt

12-th International Conference
on
Aerospace Sciences &
Aviation Technology

AN ARCHITECTURE FOR DISTRIBUTED INTRUSION DETECTION SYSTEM USING AUTONOMOUS AGENTS

Ismail A. Ghaffar *, Mohamed AboRizka **, Khaled A. Fatah ***

ABSTRACT

Intrusion Detection Systems (IDSs) are security tools that attempt to detect malicious activities, which are targeted against a network and its resources. As internetworking among computer systems via the internet becomes more widely and keeps rapid increase, widespread attacks involving those networks occurs more frequently which present a new challenge to IDSs. Many approaches have been used in implementing a reliable IDS like neural networks, statistical analysis… etc, but they have some limitations. In this paper we propose a multi-agent based intrusion detection system, which will satisfy not all but most of the requirements of reliable and secure IDS. The main goals of this system, which distinguish it from other solutions, are its distributed architecture, scalability, efficiency and the use of Agent concept.

KEYWORDS

Intrusion Detection, Autonomous Agent, Distributed Agents.

* Professor, Dpt. of Computers, Military Technical Collage, Cairo, Egypt.
**PhD. Dpt. of E-Commerce, Arab Academy for Science and Technology, Cairo, Egypt.
*** *PostGraduate Student, Dpt. of Computers, Military Technical Collage, Cairo, Egypt.*

1. INTRODUCTION

In the past few years, intrusions and other attacks have become more and more widespread and sophisticated. A number of mechanisms and technologies have been applied in defending network resources. Firewalls, encryption, authentication, vulnerability checking, and other measures can all offer-improved security. But computer systems are still susceptible to attacks from hackers who take advantages of system flaws and any misconfigurations. In addition, computer systems with no connection to public networks remain vulnerable to disgruntled employees or other insiders who misuse their privilege .So, it is essential to establish a second line of defense for a network environment in the form of an (IDS) [1]. Firewalls and intrusion detection system are two complementary techniques to increase the security of a network while a firewall acts as a first barrier to repel hackers [2].

The main goal of any IDS is to detect all intrusions and only intrusions in an efficient way; the effectiveness of IDS is measured by the rate of false positive and false negative warnings over all events [1].
For effective intrusion detection it is necessary that IDSs find an acceptable balance between false positive and false negative rates. Computer network security has become a critical issue and it's important to develop mechanisms to defend against the intrusion.
The structure of the paper is as follows: section 2 is a survey on intrusion detection, section 3 introduction to Multi-agent systems, section 4 IDS related work, section 5 the proposed Model, section 6 System implementation, section 7 conclusion and future work

2. IDS OVERVIEW

Recently intrusion detection systems received more attentions due to the rapid growth of the internetworking between the computer systems, the unauthorized activates threat the computer network not only from external attackers but also may be from internal sources. Intrusion detection is one of the important techniques used to protect computer networks from such activities.

2.1. Definition and classification

An intrusion can be defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [3]. Intrusion detection is defined as “the problem of identifying individuals who are using a computer system without authorization (i.e., ‘crackers’) and those who have legitimate access to the system but are abusing their privileges (i.e., the ‘insider’ threat’).
An IDS attempts to detect an intruder breaking into your system or a legitimate user misusing system resource and should preferably perform its task in real time. The IDS will run constantly on your system, working away in the background, and only notifying you when it detects something it considers suspicious or illegal or taking some immediate action to prevent damage [4].
Based on the characteristics of intrusions that an ID system can detect, IDSs can be categorized into three models:

- **Anomaly detection model.**

The IDS detects intrusions by attempting to characterize normal operation and to detect any deviation from normal.

- **Misuse detection model**.

The IDS detects intrusions by looking for activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities.

- **Hybrid detection model.**

The hybrid detection model uses anomaly and misuse detection at the same time. By combining these approaches to a hybrid approach, the best of both approaches can be attained.

According to the audit data source of IDS, IDSs can be classified into two categories:

- **Host-based.**

Host-based IDSs directly monitor the computer on which they run, often through tight integration with the operating system. Audit data from a single host is used to detect intrusions.

- **Network-based.**

Network-based IDSs monitor network traffic between hosts. Network traffic data, along with audit data from one or more hosts, is used to detect intrusions.

2.2 Limitation of existing IDS

However, many of the existing network-based and host-based IDSs perform data collection and analysis centrally using a monolithic architecture, that is, the data is collected by a single host, either from audit trails or by monitoring packets in a network, and analyzed by a single module using different techniques. Other IDSs perform distributed data collection (and some preprocessing) by using modules distributed in the hosts that are being monitored, but the collected data is still shipped to a central location where it is analyzed by a monolithic engine [3].

3. AGENT AND MULTI-AGENT TECHNOLOGIES

Agent technology is a very active field of distributed artificial intelligence (DAI) research in the recent years. Over the last decade the popularity of agent-based systems has increased rapidly because agents bring intelligence, reasoning and autonomy to software systems [6]. Agents are being used in an increasingly wide variety of applications. An agent is supposed to act spontaneously (with initiative); executing pre-emptive and independent actions that eventually benefit the user through accomplishment of the assigned goals. In general an agent has the following characteristics [3] [12] [13]:

- Autonomous (able to exercise control over their own actions)
- Proactive (be goal oriented and able to accomplish goals without prompting from the user, recognize and react to changes in their environment).
- Persistent (keep running a process continually until the desired result has been achieved).
- Communicative (be able to communicate with other agents).
- Mobile (able to roam networks freely).

To solve a particular problem, it often happens that a complete system consisting of several different agents has to be designed to cope with a complex problem involving

either distributed data, knowledge, or control. A multi-agent system (MAS) can therefore be defined as a collection of computational entities, possibly heterogeneous, that possesses their own problem-solving capabilities and are able to interact in order to reach an overall goal [3][12].

4. IDS RELATED WORKS.

Due to the increasing in complexity of computer networks as well as the growing sophistication of computer attacks, intrusion detection has recently gained more attention from academic, military, and commercial sectors. For example as academic intrusion detection systems **EMERALD** [16] and **NetStat** [5] gather and relate data from different sources. They have static, hierarchical architecture where sensors located at different hosts collect data and send it to a central entity where events are related.

DIDS Distributed intrusion detection system centralized director obtained information from the monitors to detect intrusions [17].

A proposed architecture for a distributed intrusion detection system based on multiple independent entities called autonomous agent for intrusion detection (**AAFID**) framework [3].

Sparta (security policy adaptation reinforced through agents) is a system architecture, which is capable of monitoring a network to detect network intrusions and security policy violations; Sparta is a distributed design in dynamic environments [8].

GrIDS is a graph based intrusion detection system for large networks which constructs activity graphs from network traffic data to detect large scale automated attacks in real time [15].

In the proposed system the data collection processes and preprocessing processes are carried out through independent entities (Agents), cover the scalability and distributed architecture of the system.

5. PROPOSED ARCHITECTURE

The proposed design of Distributed IDS uses the agents' concept to provide a solution for some of the IDS limitations and problems.

5.1. System design

The framework used for modeling, analysis and construction of agent-based IDS is rooted in the Belief Desire Intention (BDI) formalism and extends the Unified Modeling Language (UML) to model MAS. We introduce several modeling constructs to illustrate the framework of the proposed model [6] [7]**.**

Figure1 illustrates the proposed model in a collaborative diagram [9], which is presented in a hierarchical structure in 3 levels. Level 1 contains the Interface Manager Agent, Action Agent, and Communication Agent. Level 2 contain the Anomaly Manager Agent, Misuse Manager Agent, and Performance Agent. Level 3 contains the Capture Agents, Type Agents, Performance Counter Agent, and Log Files Evaluation Agent. Each level is responsible for the creation and management of the lower levels; this model exists in all nodes within the desired network.

This hierarchical structure with multiple levels of agents makes the system scalable [14] [15].
Figure 2 illustrates the Top-level view of the proposed model in Client/Server network, on each node in the network the proposed system operates independently and cooperatively with other nodes in the network.

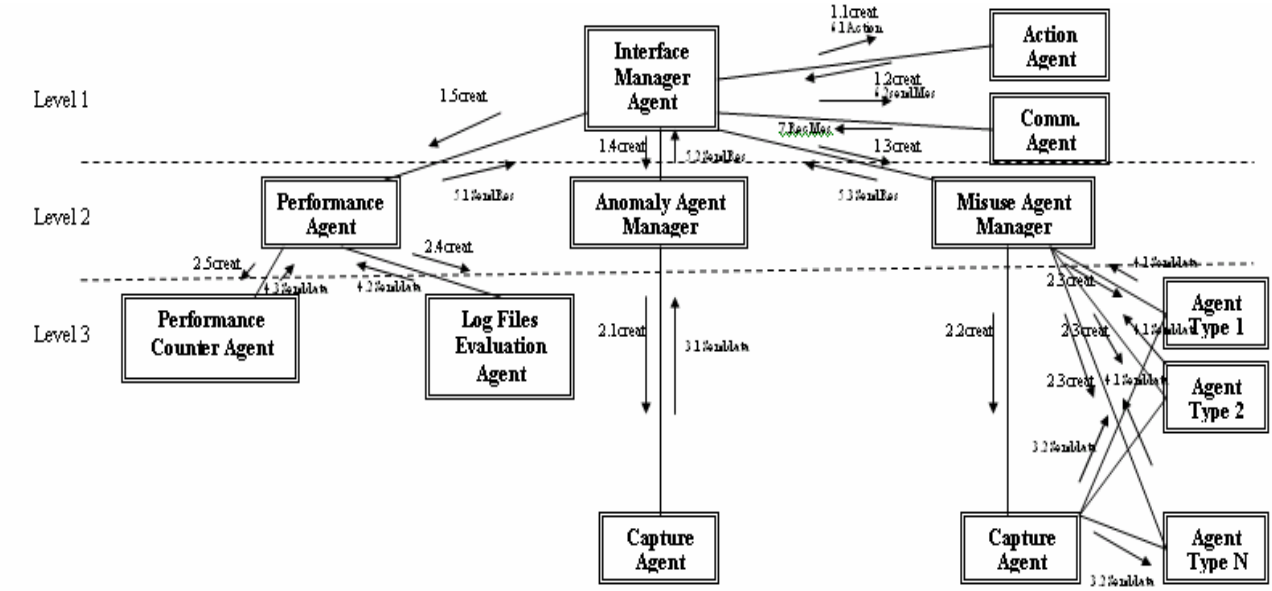

Fig. 1 Collaborative diagram

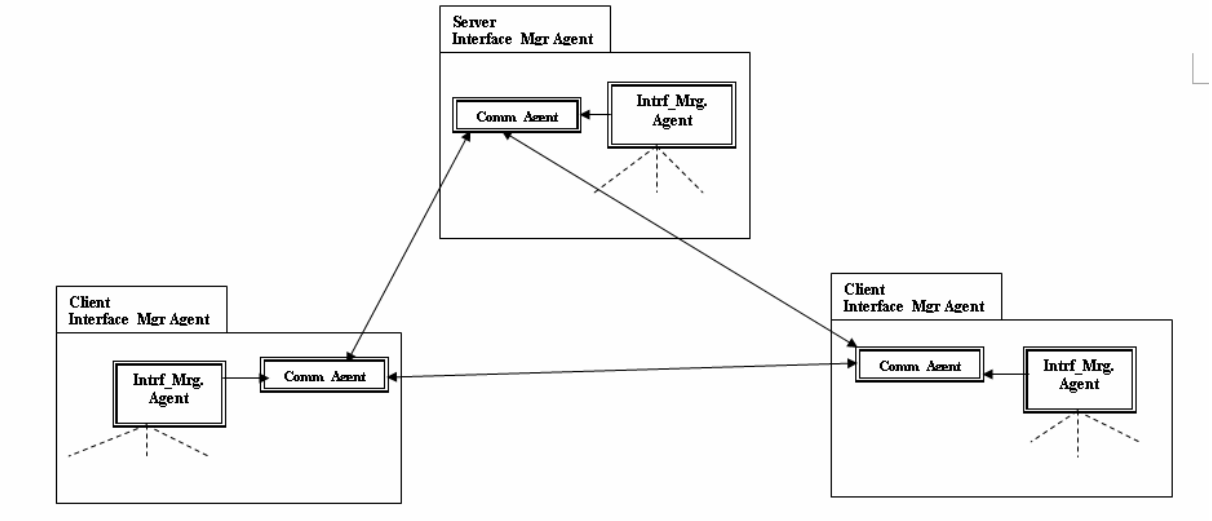

Fig. 2 Top-level view for the system in Client/server Network Model

Figure 3 illustrates the Use Case Goal Diagram (UCGD) [6] [7] which combine the existing Use Case Diagram (UCD) and the Agent Goal Diagram (AGD) and show the relationships between use cases and goals. That is, which use cases would affect which goal and vice versa. This information provides a high level guidance to Agent

Sequence Diagram (ASD) construction. It can also be used to check the consistency between UCGD and ASD.

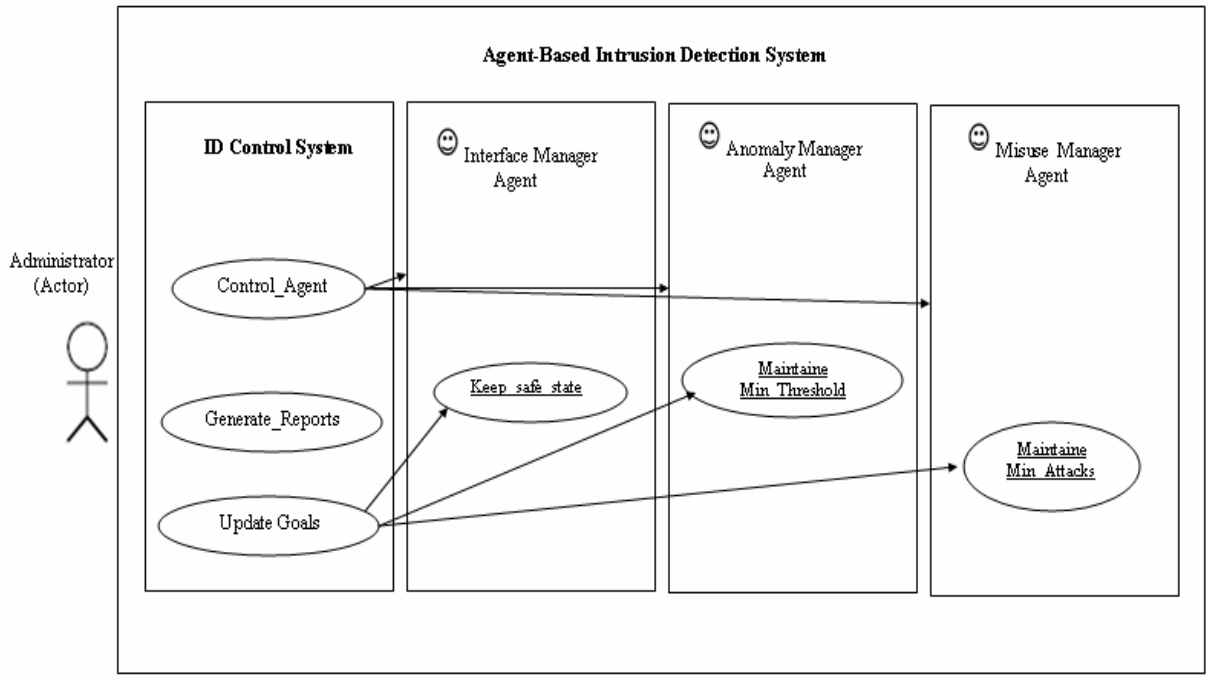

Fig. 3 Use Case Goal Diagram for the proposed model

Figure 4 illustrates the Agent Sequence Diagram (ASD) [6][7][9] used to model interactions within the system, which presents the interactions and the sequence of events between all the agents within the system, a detailed sequence diagrams for all the manager agents presented in Figures 5 and 6*.*

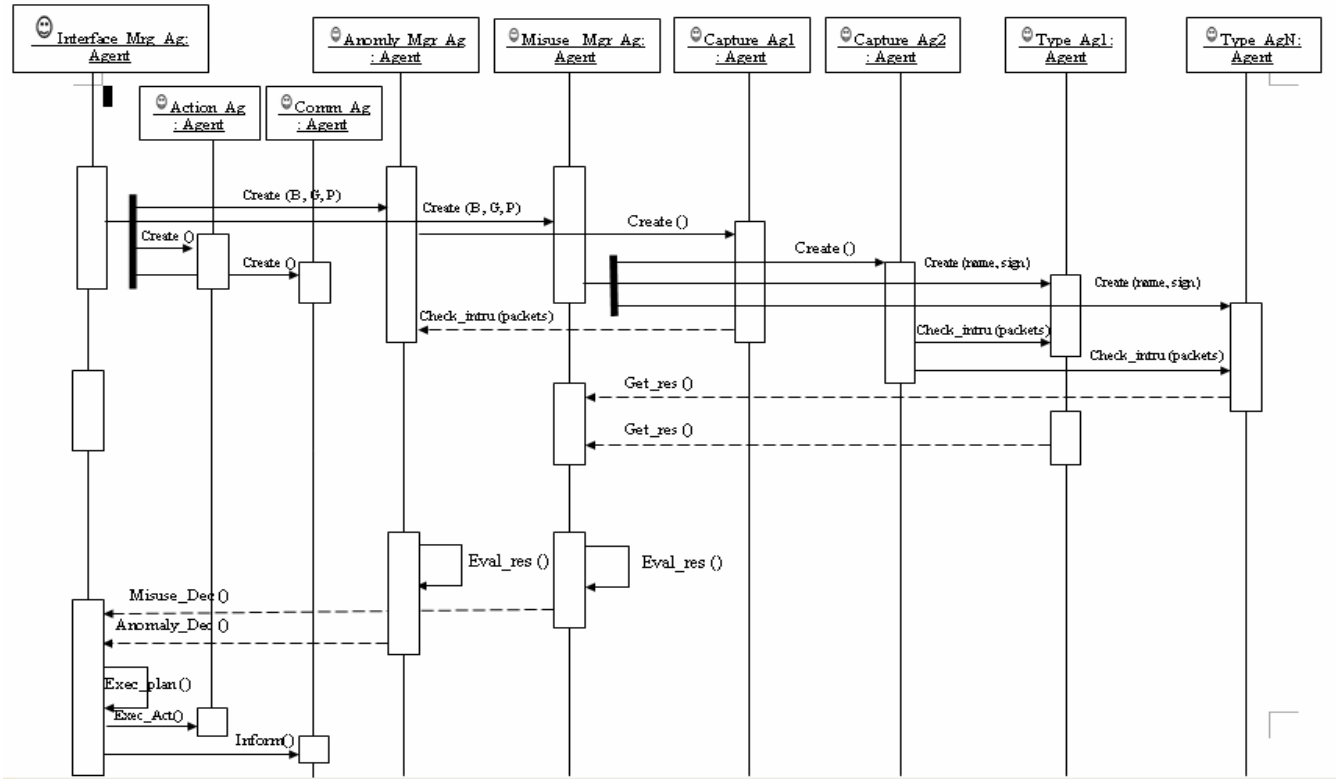

Fig. 4 Agent Sequence Diagram for the Proposed System

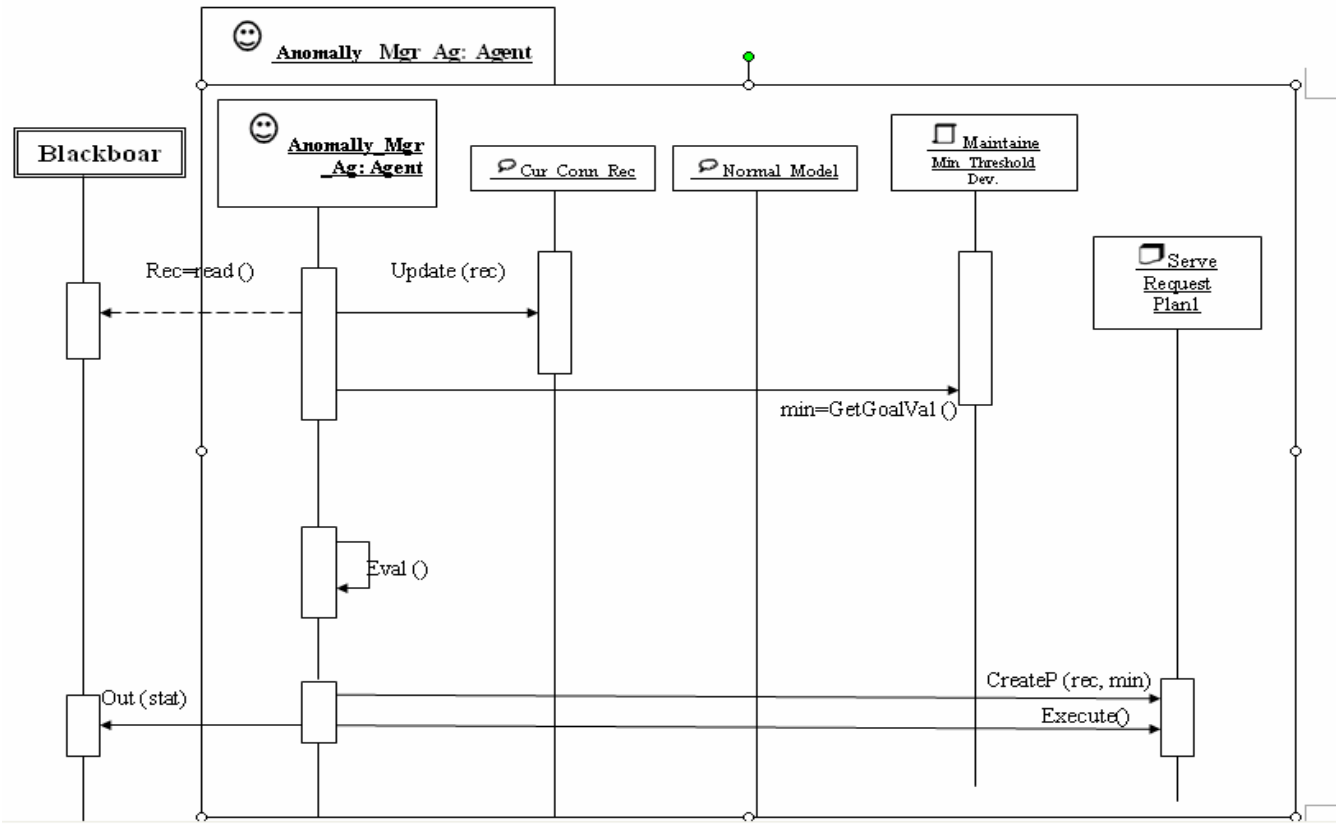

Fig. 5 Agent Sequence Diagram for the Anomaly Manager

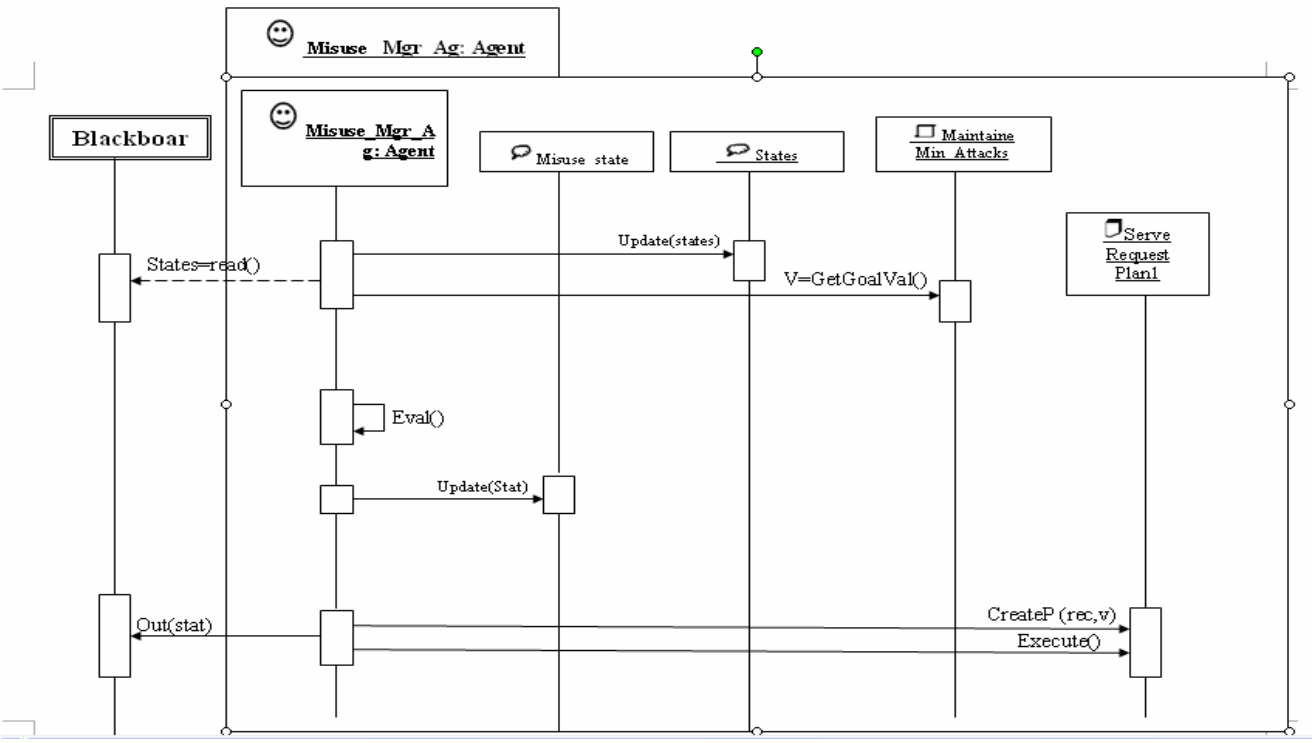

Fig. 6 Agent Sequence Diagram for the Misuse Manager

Figures 7, 8 and 9 illustrate An Agent Domain Model (ADM), which represents the domain knowledge that is internal to each agent including the definitions of the Beliefs, Goals, and Plans and their intrinsic relationships.

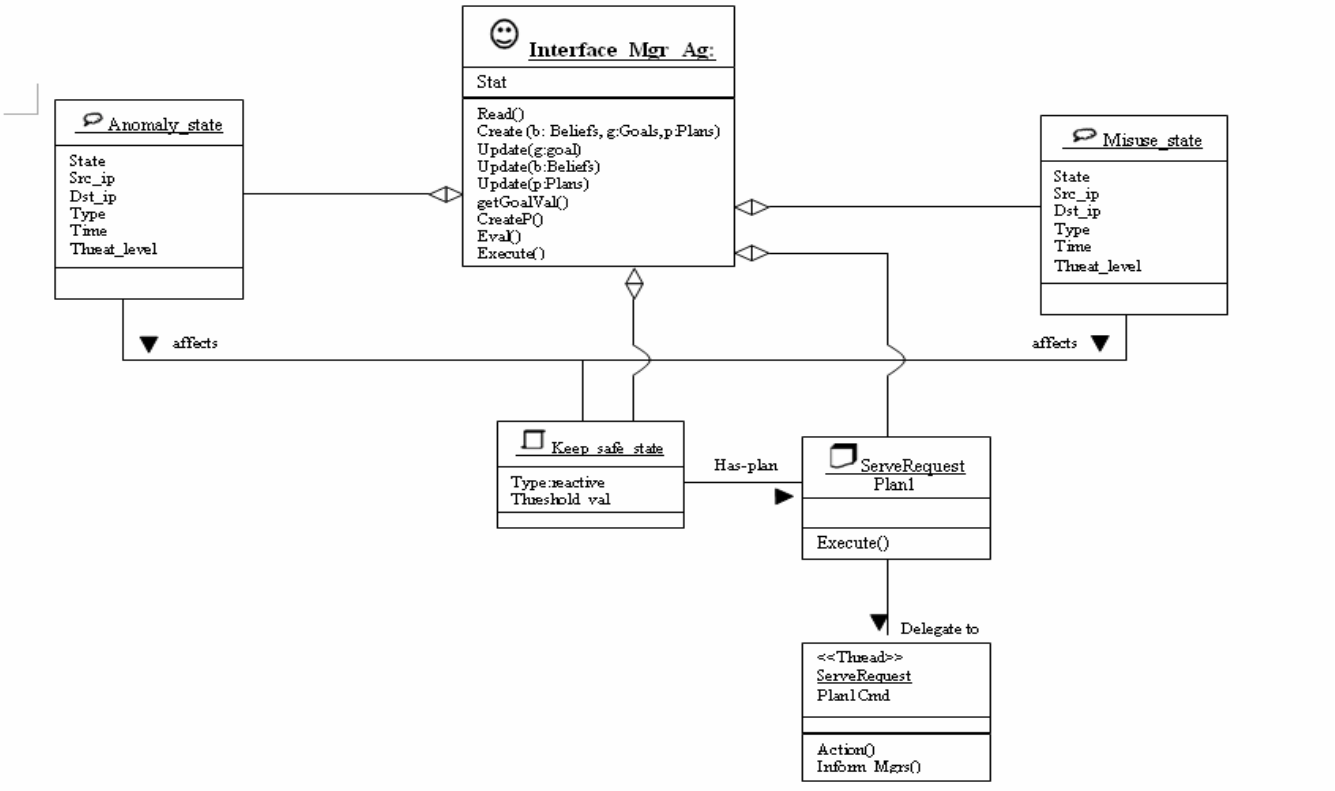

Fig. 7 an Agent Domain Model for the Interface Agent

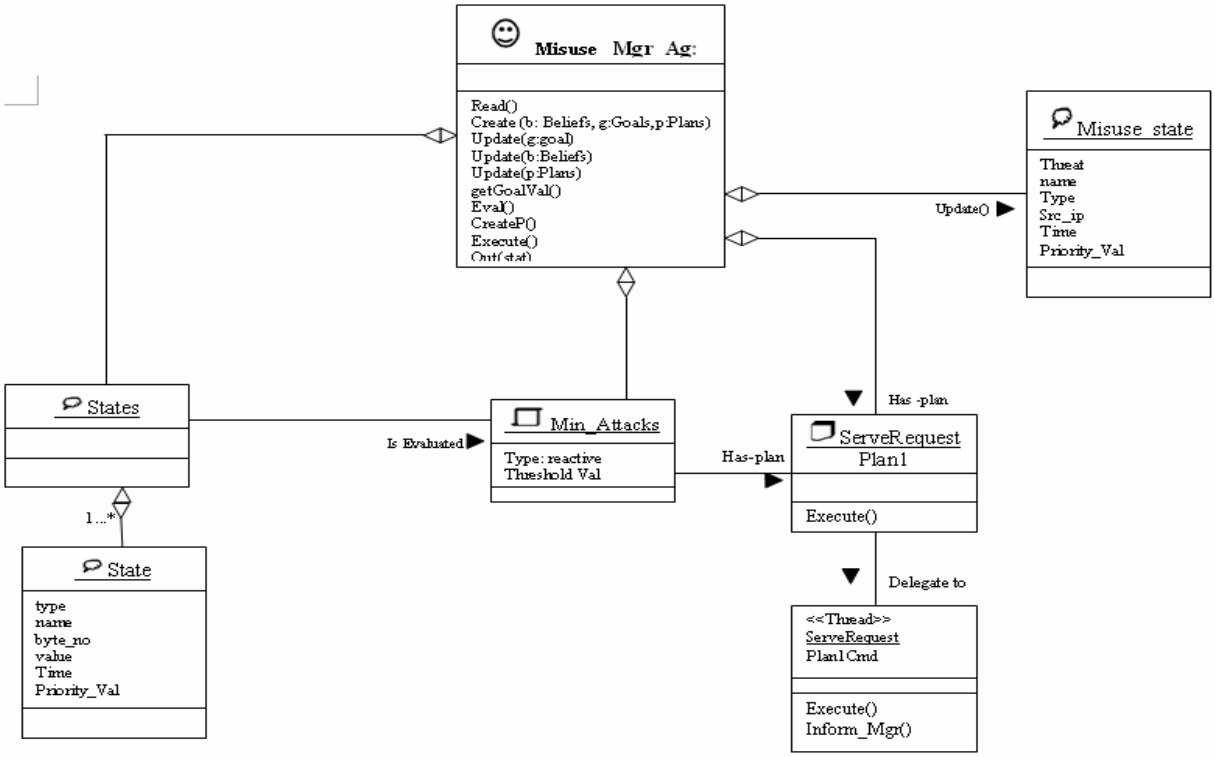

Fig. 8 an Agent Domain Model for the Misuse Agent

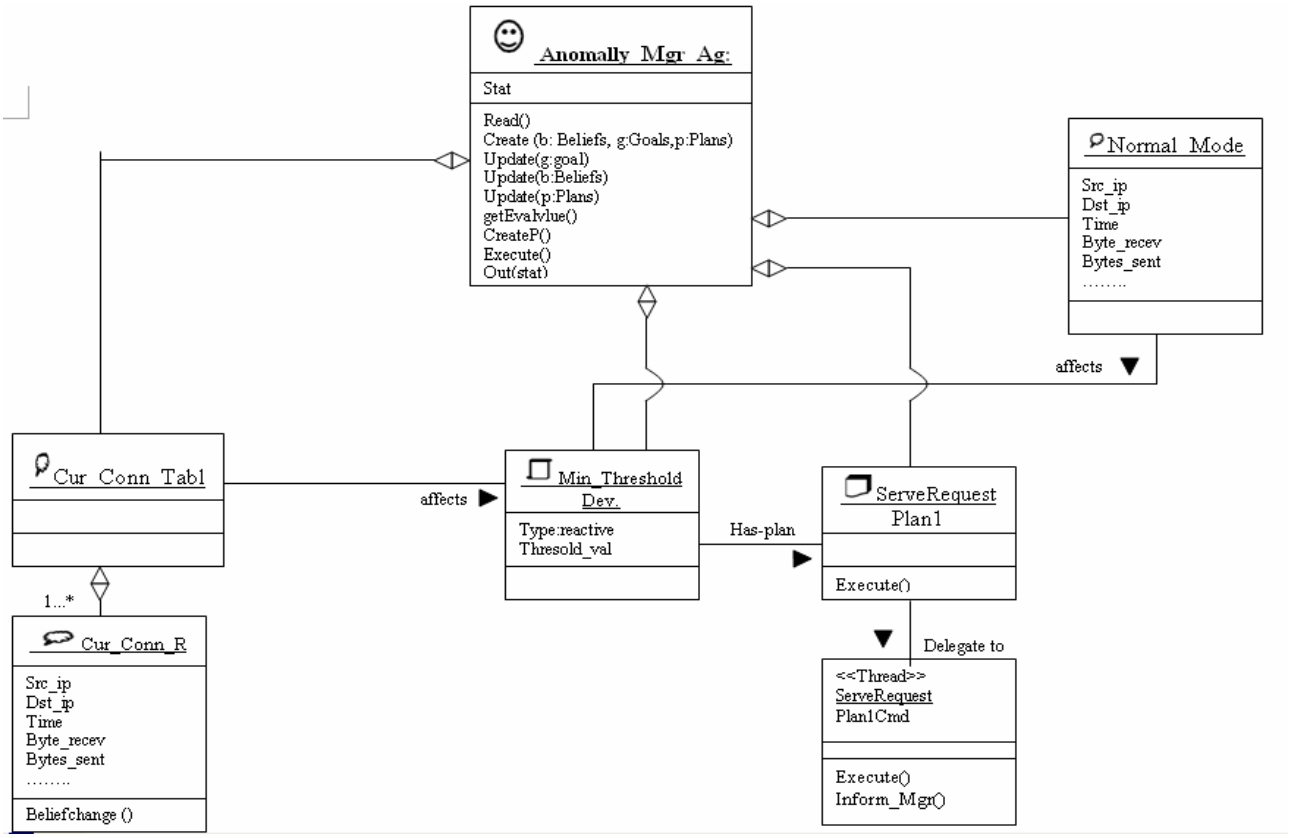

Fig. 9 an Agent Domain Model for the Anomaly Agent

5.2. Agent responsibilities and system operation

5.2.1. Interface Manager Agent.

Is the main agent in this model through which the administrator can manage the IDS in the desired node and in the whole network. Once the system lunched, the interface agent begin to create and initialize the second level agents and the action agent, communication agent ,also it is responsible for collecting the status of other managers in order to generate the management reports for the administrator and take the appropriate actions on the hosted system.

5.2.2. Action Agent.

It is a first level agent, which contains a registration of different responses that the system can perform; also it executes the Interface Manager Agent Decisions on the corresponding systems and according to the registered actions.

5.2.3. Communication Agent.

It is a first level agent that contains a registration for all the active communication agents in the network ,and it responsible for handle the communication between Interface Manager Agent in different sites and the Interface Manager Agent (in the server) through ACL Messages.

5.2.4. Misuse Manager Agent.

This second level agent, once initialized, creates the lower level agents (capture agent, different Agent Types) and initializes them to start their execution. It collects

the results from the Agent Types, making reasoning on these results and informs its decision to the Interface Manager Agent.

5.2.5. Anomaly Manager Agent.
This second level agent, once initialized, creates the lower level agent (capture agent) and initializes it to start its execution. It also receives the processed packets from the capture agent, performs its preprocessing using datamining algorithm to detect any type of intrusions referencing a normal model which is constructed during a training phase, and then it informs its decision to the Interface Manager Agent.

5.2.6. Performance Manager Agent.
This second level agent, once initialized, creates the lower level agents (performance counter, log files evaluation) and initializes them to start their execution. It also collects the results from them; making reasoning on those results and informs its decision to the Interface Manager Agent (out of scope).

5.2.7. Anomaly Capture Agent.
This third level agent, once created and initialized, starts capturing the network traffic and performs preprocessing operations on the captured traffic to place it in a suitable format for the upper levels Agents.

5.2.8. Misuse Capture Agent.
This third level agent, once created and initialized, starts capturing the network traffic and performs preprocessing operations on the captured traffic to place it in a suitable format for different types of Agents.

5.2.9. Intrusion Types Agent.
This third level agent, once created and initialized, receives the preprocessed network traffic and checks for the desired signature on it and then informs its decision to the Misuse Manager.

6. SYSTEM IMPLEMENTATION.

The prototype Agent-Based IDS has been implemented using jade3.4 [10], it is a Java-based agent development framework, a combination of two products (A FIPA-Compliant Agent Platform, a package to develop Java agents), includes the following agents

- AMS (Agent Management System)
- DF (Directory Facilitator)
- Sniffer
- RMA

All the system Agents work concurrently all the time. Once the Interface Manager Agent takes a decision against an intrusion type occurring in the system, it orders the Action agent to take the appropriate action against that intrusion and it also communicates with other Manager Agents in other sites through the Communication Agent.
The administrator is allowed to add a new Agent Type with its signature to the system, and it has a full control over all the agents in the system (start, stop, resume,

kill…). This capability allows the administrator to reconfigure the IDS (or part of it) without having to restart it or disturbing any of the running agents.
The system operates in 2 modes:(1) training mode: it can be either offline training through Datasets or online training through a secure network and a simulator software for different types of intrusions; and (2) the detection mode: the system runs continuously to detect intrusions in a live network.
Any Agent that goes down during its operation does not affect the continuity of the system; once the upper level Manager is informed, it immediately generates another Agent with all its capabilities.

The communication between the Agents within the same node is carried out through Blackboard (shared memory as shown in the figures), while the communication between Interface Manager Agents in different sites is carried out through message passing technique (ACL messages) which is the standard agent communication language from FIPA.
Anomaly Manager Agent was implemented as a datamining Classification algorithm (decision tree) to build a model for normal behavior during training and classify any deviation from the normal behavior in the test data [11].

7. CONCLUSION AND FUTURE WORK

This model is proposed to get rid of the drawbacks of most of the existing IDS also to satisfy as most as possible the requirements for secure, reliable, efficient IDS. Agents provide Efficiency, fault tolerance, extensible a scalable system [12]. Distributed Agents overcomes the Centralized monolithic analysis (single-point of failure). An implementation of the Anomaly Agent (off-line learning and testing) was introduced.
For the future work we will work on an extension of the proposed architecture using Genetic algorithms with datamining to detect novel attacks in the networks. We still have a challenge in the communication between Agents in different sites, which is a security challenge.

8. REFERENCES

[1] Housam Shaban Al-Allouni,” An Intrusion Detection Approach to Computer Networks”, Msc thesis, Military Technical College, 2003.

[2] Christopher Krugel, Thomas Toth, “Flexiable, Mobile Agent Based Intrusion Detection for Dynamic Networks”, In European Wireless, Italy, February 2002.

[3] Jai Sunder Balasubramaniyan, Jose Omar Garcia-Fernandez et. al, “An Architecture for Intrusion Detection using Autonomous Agents”, 14th Annual Computer Security Applications Conference, pages 13-24. IEEE Computer Society, December 1998.

[4] Mohamed Eid,”A New Mobile Agent-Based Intrusion Detection System Using Distributed Sensors", third FEA Student conference, American University of Beirut, 2005.

[5] G. Vigna and R. Kemmerer,"Netstat: A network based intrusion detection system ", 14th Annual Computer Security Applications Conference, 1998.

[6] Krishna Kavi, David C. Kung, Hitesh et. al.," Extending UML for Modeling and Design of MultiAgent Systems", 2nd Intl Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS2003), Portland, OR, May 3-10, 2003.

[7] Krishna M. Kavi, Mohamed Aborizka, and David Kung,"A Framework For Designing, Modeling and Analyzing Agent Based Software Systems", 5th International Conference on Algorithms and Architectures for Parallel Processing (IC3APP2K2), Beijing, China, Oct. 23-25, 2002.

[8] Christopher Krugel, Thomas Toth,"Sparta – A Mobile Agent based Intrusion Detection System", IFIP Conference on Network Security (I-NetSec), Kluwer Academic Publishers, Belgium, November 2001.

[9] James Odell, H. Van Dyke Parunak, " Representing Agent Interaction Protocols in UML", 22nd International Conference on Software Engineering (ISCE) pp. 121–140, 2001.

[10] Java Agent Development Environment, http://jade.tilab.com/.

[11] Wenke Lee," A Data Mining Framework for Constructing Features and Models for Intrusion Detection Systems", 1999 IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.

[12] Walter Brenner, Rudiger Zarnekow, "Intelligent Software Agents Foundations and applications", Springer, 1998.

[13] Michael Wooldridge."An introduction to MultiAgent Systems, John Wiley & Sons, February 2002.

[14] Mark Crosbie and Eugene Spafford,"Defending a computer system using autonomous agent", 18th National Information Systems Security Conference, Oct 1995.

[15] S. Staniford-Chen, S. Cheung, R. Crawford et. al.,"GrIDS: A graph based intrusion detection system for large networks", 19th National Information Systems Security Conference, October 1996.

[16] Philips A. Porras and Peter G. Neumann," EMERALD: Event monitoring enabling responses to anomalous live disturbances ", 20th National Information Systems Security Conference, 1997.

[17] S. R. Snapp, J. Brentano, G. V. Dias et. al., "DIDS (Distributed Intrusion Detection System)-Motivation, Architecture, and an early Prototype", 14th National Information Systems Security Conference, 1991.