The Design and Implementation of an Adaptive Intelligent Multi-Agents for Network Protection.

Ihekweaba Ogechi, M.Sc.[*]; Ihekweaba Chukwugoziem, Ph.D.; and H.C. Inyiama, Ph.D.

[1]Department of Computer Engineering, Michael Okpara University of Agriculture, Umudike, Nigeria.
[2]Department of Computer/Electronics Engineering, Nnamdi Azikiwe University, Awka, Nigeria.

E-mail: gozihekwaba@yahoo.com[*]

ABSTRACT

This paper presents the basic concept of network attacks, as well as their profiles, taxonomies, classification and identification frameworks. Further, current approaches for intrusion detection on networks were x-rayed though, with emphasis on Denial of Service (DoS) attacks. The concept of intelligent agents and subsequently adaptive systems was elucidated. Modern systems analysis and design tools were then applied in the development of a model for an adaptive intelligent multi-agent for DoS protection.

(Keywords: agent, dynamic host configuration protocol, access control list, context based access control, data flow diagram, agent oriented software engineering)

INTRODUCTION

Devices that connect into the corporate network through are frequently not in compliance with corporate policies. This scenario has laid networks bare to security breaches including application layer attacks, auto rotes, backdoors, man–in–the–middle attacks, network reconnaissance, packet sniffers, password attacks, brute force attacks, port redirection attacks, Trojan horse attacks, viruses, trust exploitation attacks, Denial of Service (DoS), and Distributed Denial of service (DDoS) attacks (Todd 2007).

Denial of Service attacks are a major cause of incorrect operations in the internet and is arguably one of the most serious threats that the internet community faces today (Valar, 2004). From the time it is detected and recovered from, the victim is virtually paralyzed and cannot respond to legitimate requests. For large commercial sites, this translates to losses of billions of dollars in magnitude (Aleifa et al., 2007)

Corporate networks, and the attacks used to exploit them, are so complex that no single mechanism can be relied upon to keep them secure. This has led to the concept of "Defense in Depth" (Todd, 2007).

The self – defending network based on adaptive intelligent agent monitoring, provides systems – based solutions that allow organizations to use their IT infrastructure in new ways to reduce windows of vulnerability, minimize the impact of attacks, and improve overall infrastructure availability and reliability (Cisco White paper 1992 – 2005). This helps create autonomous systems that can quickly react to an outbreak with little or no human interaction. This type of rapid response is required to thwart the latest forms of misuse that are much more virulent than their predecessors.

It is helpful to classify them in order to clarify the process of defending against DoS. Some technical writing categorize Denial of service attacks as flood attacks and malformed packets while others consider these as software exploits and flooding attacks. Single source and multi-source are categories that are based on the location of the observation point.

It is noticeable from available literature (Prashant et al., 2003) that there had been much research addressing automated classification of attacks. Attacks are usually classified based on header analysis, ramp–up behavior and spectral analysis. It is however suggested in the literature that this three – pronged approach is necessary to deal with an increasing level of difficulty in classifying attacks depending on the level of IP header spoofing present in the attack.

Typically, intrusion detection systems (IDS) can be classified based on two concepts; matching of

the previously seen and hence known anomalous patterns from an internal database of signatures or building profiles based on normal data and detecting deviation from the expected behaviors. The first approach is referred to as misuse detection and leads to signature Based IDS while the second is Anomaly Detection and leads to Behavior based IDS. The signature based systems have very high detection accuracy but they fail when an attack is previously unseen. On the other hand, behavior based IDS may have the ability to detect new unseen attacks but have the problem of low detection accuracy (Debar, 1992; Kumar et al.,1994; Ghosh, 1991).

Also based on the mode of deployment, the intrusion detection systems are classified as Network based, Host based, and Application based. Network based systems make a decision by analyzing the network logs and packet headers from the incoming and outgoing packet since they are deployed at the periphery of the network. Though they are easy to manage and give a centralized control, they have to work with limited information and are further constrained in case of encryption and network address translation. Host based systems monitor individual systems and uses system logs extensively to make any decision.

Fred Cohen published in 1984 that detection of computer viruses is undecidable and NP-hard (Cohen, 2004). In laymen's terms, this means that it is impossible to detect every type of an intrusion in every type of case, and that the resources needed to detect intrusions grow with the amount of network traffic. Paul Helman, et al. in 1992 used a scale of 0 to 1 to represent normal behavior (0) to misuse (1) (Helman, 1992).

The purpose of an intrusion detection system is to provide the rating for computer activities. According to (Jensen et al. 1999), authors have defined a set of desirable characteristics for IDS by focusing on two themes: Functional and performance requirements. The functional requirements include:

- IDS must continuously monitor and report intrusion.
- IDS should have a very low false alarm rate,
- IDS should provide enough information to repair the system in the case of detection of intrusion notice that these requirements lean on basic IDS goals. In fact, conventional IDS solutions focus only on alerting administrators without suggesting any corrective actions.
- IDS must detect and react to distributed and coordinated attacks. This detection feature is one of the most difficult because it needs a huge distributed amount of information in addition to the hard task of synchronization between different hosts.
- The IDS should be adaptive to network topology and configuration changes (Jensen et al., 1999).
- Intrusion should be detected in real – time as it should be reported immediately in order to minimize network damage.
- The IDS must be scalable in order to handle additional computational and communication loads.

Different types of agents reflect a set of properties, which common among them (Oliveira, 1998) are, autonomy, Co-operation, proactivity, (Labroid,1998; Bocan, 2004)., reactivity, adaptability, intelligence, flexibility and mobility. Having reviewed the properties of intelligent agents as presented in referenced literature (Oliveira, 1998), it can be concluded that intelligent agents provide a more coherent and flexible approach to network intrusion detection and security management.

To model intrusion detection, the intelligent agent must combine the cognitive (knowledge – based) to reason about complex attacks with reactive capabilities (stimulus response) to react to the environments changes. Thus, an agent has three functions: an event filtering function, interaction, and a deliberation function as shown in Figure 1.

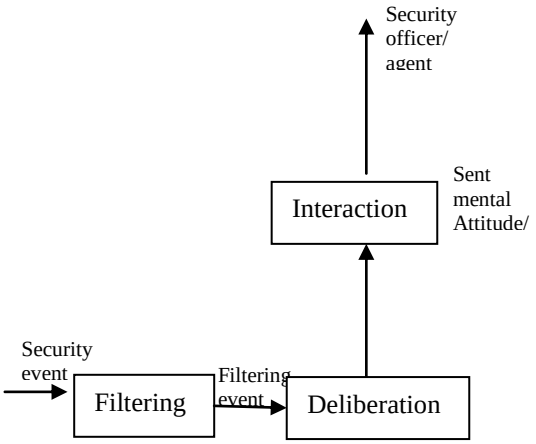

Figure 1: Interaction between Agent and Function.

THE DIANA AGENT ARCHITECTURE

The DIANA architecture being the shell of the proposed intelligent agent strategy for network protection as proposed by this work, literature referencing this architecture (Asaka, 1999), was considered as substantial for the understanding of this work. The DIANA agent architecture consists of two main components: the brain, which is responsible for managing agent skills and skills, which provide the agent with capabilities and behaviors. The Agent's brain (Figure 2) offers two types of necessary facilities for the agent operations: local and inter agent facilities.

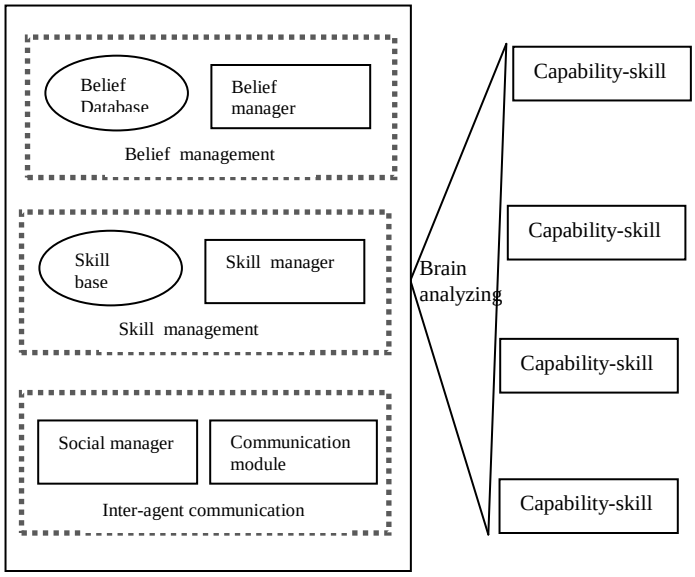

Figure 2: DIANA Agent Architecture.

The main role of the brain is to manage both agent's belief database and agent's skill base. "An agent belief expresses its expectation about the current state of the world and about the likelihood of a course of action achieving certain effects" (Muller, 1996). Beliefs hold network management information as well as information about the agent itself and the other agents. These beliefs can be accessed concurrently by several skills, therefore, the Belief Manager maintain the integrity and coherent access to the Belief Database.

Skills can be downloaded dynamically into the agent inside its skill base. The main role of the skill manager is to check the availability of pre-requisite skills required by newly loaded skills and if they not yet loaded, it must search for them either locally or on distant agents. It is also responsible for the disposal of un-useful skills to keep the agents size as small as possible. During its operations, the skill can update or delete existing beliefs or create new ones. A skill operation may depend on belief created by other skills, and the skill manager is therefore in charge dispatching asynchronously these beliefs to the interested skills in a transparent way. It holds all the necessary information about the skill in the base.

The Brain Analyzer is responsible for the parsing of the messages that the brain receives, either from the skill or from the inter-agent communication.

Both the communication module, which is responsible for managing interaction with the other agents and the social manager, which holds information about the other agents, support inter-agent communication facilities to the agent.
The idea of distributing the intrusion detection system using agent software is not entirely new. However, most of the related works emphasize static agents instead of mobile one. Applying mobile agent technology to IDS has been carried out within only a few research projects.

In 1999, a project at the information-Technology Promotion Agency (IPA) in Japan involved an intrusion Detection Agent (IDA) system (Bernades, et al., 2000). IDA is a classic hot-based system that relies on mobile agents mainly to trace intruder among the various hosts involved in an intrusion. In the same year the project MICAEL (Trapethi, et al., 2002) pursued a more ambitious aim where the entire system is based on adaptive intelligent agents. Nevertheless, only the architecture description has been presented and no details have followed so far.

In 2000, an IDS framework based on mobile intelligent agents has been described in (Trapethi et al., 2002). Unfortunately, detection is dealt with superficially. In 2002, Wooldridge describes an IDS designed as a mobile application that roams the network to detect attacks and track intruder. IMA-IDS are a distributed intrusion detection system using mobile and intelligent agents.

For the system analysis described in this work, in particular, the use of case driven object oriented software engineering (OOSE) approach is utilized, with extension for agent oriented analysis using methodology for engineering system of software agents (MESSAGE). This hybridization of OOSE and MESSAGE is an

effective methodology for specifying and analyzing agent based systems.

MESSAG is an AOSE methodology which builds upon current software engineering practices covering analysis and design of multi-agent system (MAS).

SYSTEMS ANALYSIS

In the analysis of the existing system, it was necessary to consider its effectiveness as it leveraged on the configuration and continual update of access control list (ACL) in the routing subsystem of the network security infrastructure based on some security policies.

This system is based on the detection of unauthorized access, network attacks, and rogue services via through an intrusion prevention system (IPS).

The components of this system include:

- The perimeter defense module (employing packet filtering).
- Access – control list (ACL).
- Authentication proxy (This authenticates inbound users and outbound users).
- The context based access control (CBAC). The job of this module of the security system is to scrutinize any and all traffic that is attempting to come through the firewall so it can find out about and control the state information for TCP and UDP session. The logical topology of this system is depicted in Figure 3.

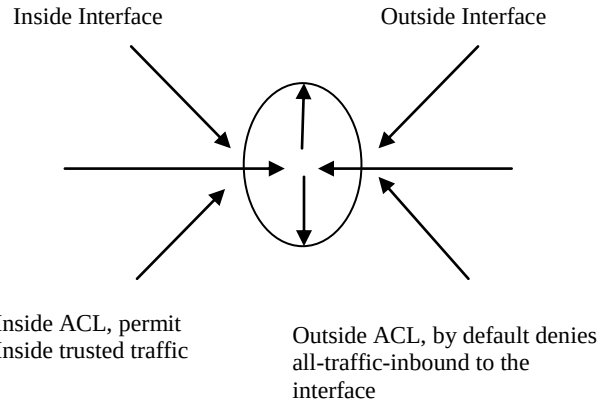

Figure 3: Logical Topology of the CBAC.

The existing system can be analyzed by employing a process model to represent it. A process–centered technique is ideal for analyzing this system since the overall function of this system for network defense is the integration of the process modules of perimeter filtering that employs open system interconnectivity (OSI) layer three features, access – list control (ACL) processing, and conditional forwarding process.

Modern structured analysis, being a process–centered technique, is used to model the requirements for this system. Structured analysis employ data flow diagram (DFD) to analyze systems.

In the working of this system, the effectiveness of the system to mitigate attacks depends on the traffic filtering policy adopted by the network administrator. This is reflected on how the network administrator configures the access – list control. The access– list control is represented as a process (i.e., circles) is the structured analysis data flow diagrams (DFD). The processing of the access– list control (ALC) gives rise to data (represented as either solid or dashed arrows in the DFD).

In this case, the data could mean permit or deny signature sent to the firewall subsystem. From the architecture of the existing system that is based on the IPS (intrusion prevention system), the system uses a monolithic architecture than centralized intrusion detection and attack prevention. The access control list is configured and deployed on the centralized firewall. The centralized defense scheme suffers from a number of problems:

The deliberation module is a major component of the multi agent system. The deliberation module interacts directly with information resources mainly different specialized database entries. The databases are: The agent detection goals, Agent beliefs, Agent intention, and Agent knowledge base.

The intelligent agents do not interact directly with these databases but does so through the deliberation module as depicted in Figure 6, agents access and update these databases as the multi-agent system runs.

In the block diagram (Figure 4), the traffic filter is a component of the system that filters incoming traffic. This component constitutes the filtering

function of the intelligent agents. This filtering function access the detection goal through the deliberation module.

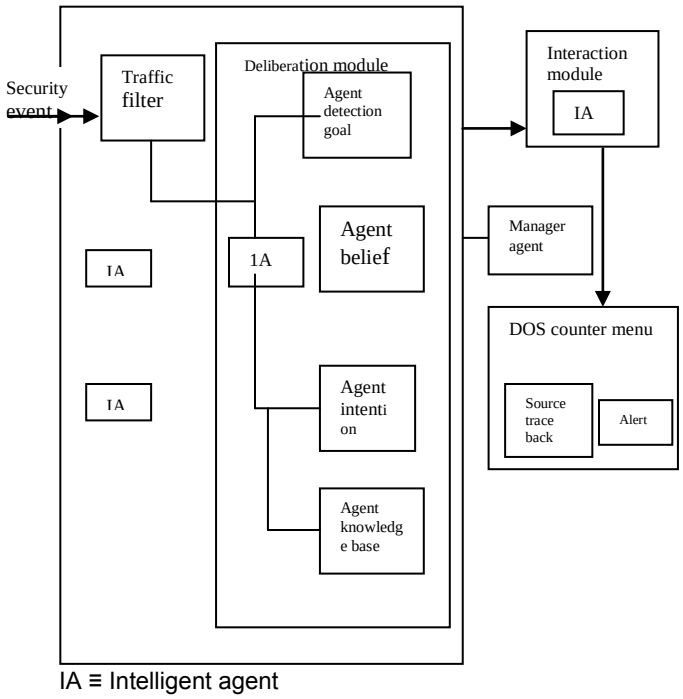

Figure 4: Block Diagram Overview of Proposed System.

The Agent detection goal is a database of security event classes. A security event is only collected when the event matches the event classes specified by the detection goals.

The agent belief is a collection of filtered events. What we can call likely attacks, or possible attacks. An intelligent agent interacts with other intelligent agents to get specialized help or other functionality as shown by the IA symbol. The manager performs some sort of co-ordination function. In doing this it also interacts with the agent knowledge base to get information about the types and specialized functions, and about other agents within the multi-agent system. It coordinates inter-agent interactions. The structure of interaction with other specialized agents is depicted in Figure 5. This collector agent will be interested in a set of event categories.

In the analysis the system is modeled as a multi – agent organization. This organizational model is made up of two models: the roles model and the interaction model.

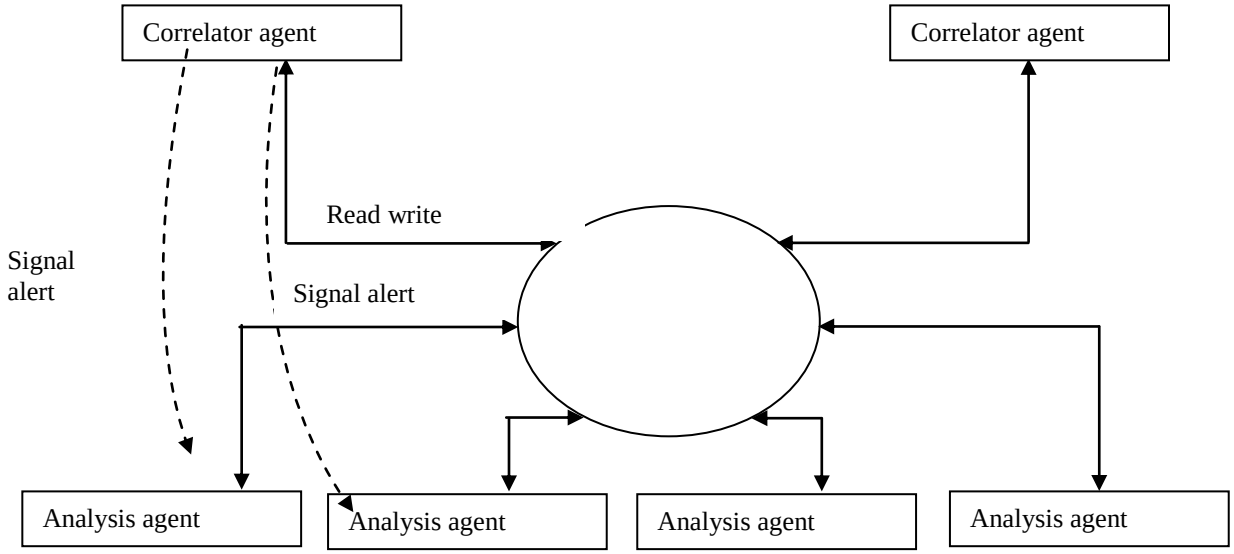

Figure 5: Manager Agent interacting with Specialized Agents.

The roles-model identifies the key roles in the system, it is an abstract description of an entity's expected function and is characterized by two types of attributes; permissions and responsibilities:

The links between the roles are represented in the interaction model. This model consists of a set of protocols definition, one for each type of inter – role interaction.

The abstract entities used in the analysis concept are depicted in Figure 6. In the diagram, the system is analyzed as being composed of roles which in turn are composed of attributes: permissions, responsibilities, protocols. The responsibilities are "liveness responsibility" and "safety responsibilities".

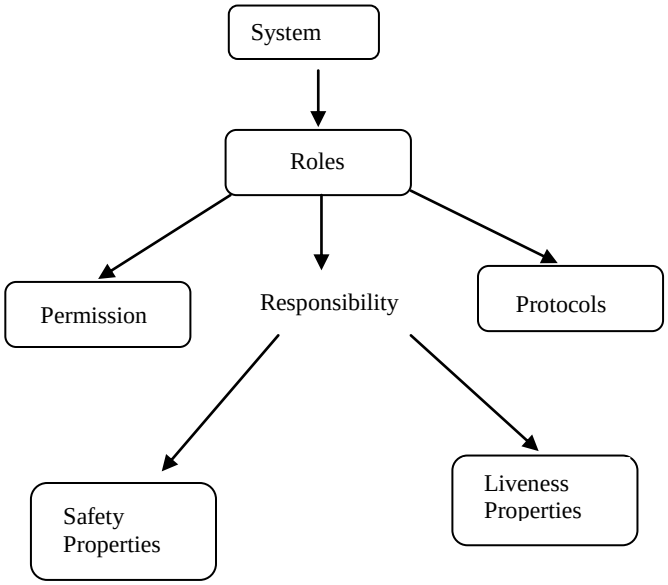

Figure 6: The Analysis Concepts.

The design process involves generating basically three models: agent model, service model and acquaintance model.

The agent model is used to document the various agent types that will be used in the system being developed, while the service model: identifies the services associated with each agent role, as it also specifies the main properties of the services.

Similarly, the acquaintance model simply defines the communication links that exist between the agent types. An agent acquaintance model is simply a graph with nodes in the graph corresponding to agent types and arcs in the graph corresponding to communication pathways.

The relationship between the analysis concepts and design concepts of the agent's organizational concepts and methodology employed in this work is depicted in Figure 7.

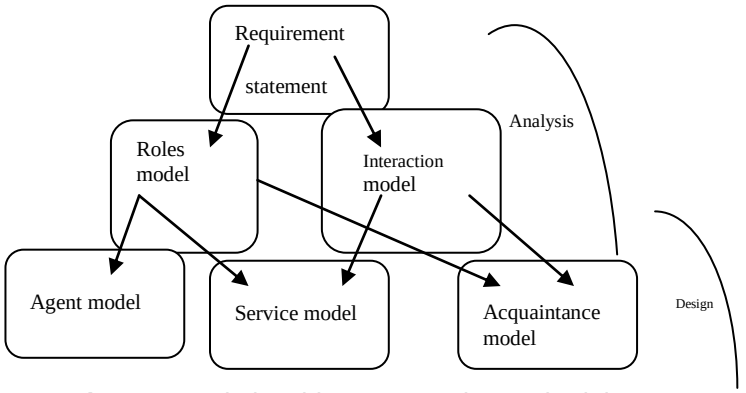

Figure 7: Relationship Between the Methodology Models. Three other agents are required.

Correlator agent: this agent will carry specific information, called critical and send it to another agent called the analyzer agent.

Analyzer agent: analyzer agent are the engine of this proposed agent system. Several kinds of analysis such as classical signature detection, anomaly detection is integrated in this agent.

The Manager agent: agent gathers collected information and distributes it to analyzer agents. The manager agent corresponding to the manager Agent role has the protocols.

- AccessAgentDetectionGoal()
- AccessAgentBelief()
- Access Agentintention()
- AccessAgentKnowledgeBase()
- CallAnalyzerAgent()
- GatherInfo()

The other schema's for the role corellator and role analyzer agents were also articulated for the design phase.

ROLE SCHEMA:	Manager Agent
DESCRIPTION:	Interacts with agent knowledge base to gather information, distributes information to analyzer agents, co-ordinates into agent interactions, and interacts with detection goals & agent beliefs.
PROTOCOLS:	AccessAgentDetectionGoal(),AccessAgentbelief(), AccessIntention(), AccessAgentknowledgeBase(), CallAnalyzerAgent(), Gather info(), TestAgentBelief()
PERMISSIONS:	Reads & Access List of Intentions Security Events Network Traffic Filtered Events Agent Knowledge Base Security Event Category Detection Goals Security – Counter- Measures Critical Information
RESPONSIBILITIES: LIVEVESS:	Manager Agent = (TestAgentBelief//AccessAgent DetectionGoal//AccessAgentIntention(), AccessAgent knowledgebase() //CallAnalyzerAgents, GatherInfo)ˢ

Figure 8: Schemas for Role Manager Agent.

Level 1 Analysis:

Moving from level 0 to level 1, analysis focuses on the system itself identified at a glance the main piece of functionality required (seen as roles and /or types of agents). The approach followed in the analysis carried out in this chapter is to consider only roles initially and to define what agent will populate the system and what roles each agent will play at the beginning of the design process.

SYSTEMS DESIGN

The design of this system involved the processes of; Data modeling, Data structures and Databases.

Data modeling methods make use of the entity relationship diagram. Data modeling identifies the data objects their attributes and their relationship using graphical notation.

Data object is a representation of almost any computer information that must be understood by the software.

By composite information is meant something that has a number of different properties or attributes.

- A queue in combination within a hash table is used to buffer network traffic data while carry out traffic filtration.
- To facilitate the exchange of data between the manager agent and analyzer agent, vector and hash tables are used.
- The list – of – intentions data object is designed and represented as a java class with no methods i.e., a data class with object serialization. This is equivalent to struct in long ways like C/ C++:
- lass list – of – intentions {
- Public string intention –key;
- Public string description;
- Public string type;
- Public string action;
- Public int attack – ID;
- Public string attack profile

The no method class and structure is similarly used to implement security– event data structure, filtered – Event data structure, critical–link data structure, alert data structure and security counter– measures.

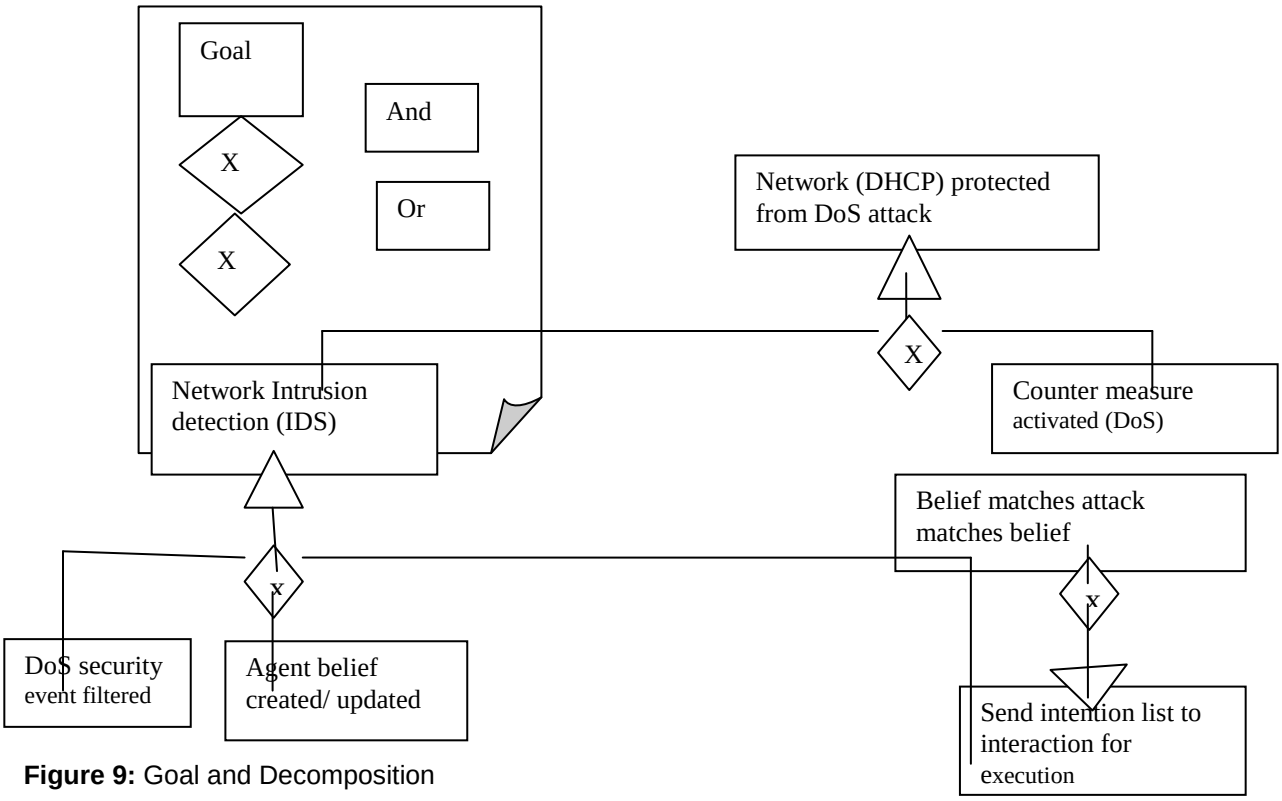

Figure 9: Goal and Decomposition

Event – ID	Event – name	Event – stamp	Security	Priority bit

- List of intentions

Intention – key	Descriptor – string	Type	Action	Attack – ID	Attack profile

- Security – event

Event – ID	Event – name	Event – stamp	Security	Event – class

Attributes

Event – class	Protocol	Event – Signature	Source – IP

- Filtered – event attributes event

Event – Protocol	Event – Signature	Event- handler	Queue – priority

▪ Critical info attributes event

Intention – Info –	Timestamp	Source	Data	Degree	Alert – level

- Alert event attributes event

Action	Event – id	Event – name	Alert – ID	Los – ID

Security – counter - measures Data attributes for

Intention –ID- signature	Event – signature	Interaction – handler	Los – ID	Security – event category

Figure 10: Data Attitude for the Object.

The very first step in designing a database application is to understand what data is to be stored in the database. The information gathered in the requirement analysis step is used to develop a high – level description of the data to be stored in the database along with the constraint known to hold over this data.

The system processing narrative identifies the knowledge base. The knowledge base constitutes a database for coordinating all the agent, for tracking security events, intrusion attempts, what is known about particular attack profiles, the agent available and running when the system start up.

The knowledge Base has the attributes:

Agent- into	No- of – attempts	Failed – attempts	Successful – attempt

Attack- source	Counter – measures	Number of agents – running	Target	Attack – signature	Timestamp

Employing data base analysis and design concepts, the knowledge base tuple is formally designed by the entity

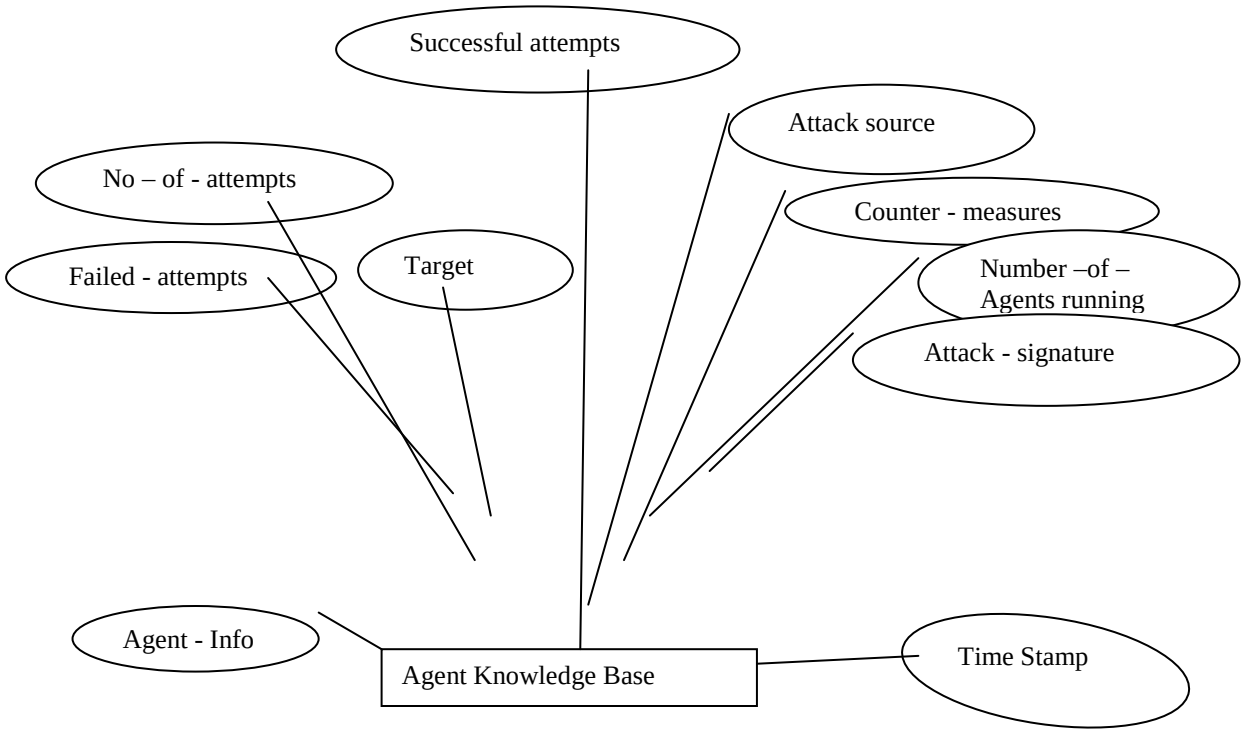

Figure 11: Agent Knowledge Base

Based on the entity set in using MySQL as database system, SOL 192 syntax uses the following construct to realize the data base:

```
CREATE DATABASE Knowledge Base {
Data file 'Drive – letter\ location\ ……\
Control – file
Log file
.
.
.
}

CREATE TABLE KNOWLEDGE BASE {
Number – of – attempts; Number,
Failed – attempts; Number,
Success – fail – attempt; Number,
Target;              char (30),
Attack – source;    char (30),
Counter – measures; char (30),
Number – of – agents running; number,
Agent – ID number primary key;
Attack – signature BLOB,
Timestamp Date time,
        };
```

Having established the data architecture and overall systems architecture of the agent based system, the component level design translates the design model into operational software.
Here the design is represented at a level of abstraction that is close to code. It establishes the algorithmic detail required to manipulate data structures effect communication between software components via their interfaces and implement the processing algorithmic allocated to each component.

Object oriented concepts are employed in the design of the system components. Unified modeling language (UML) notation was used for the description of the component design.

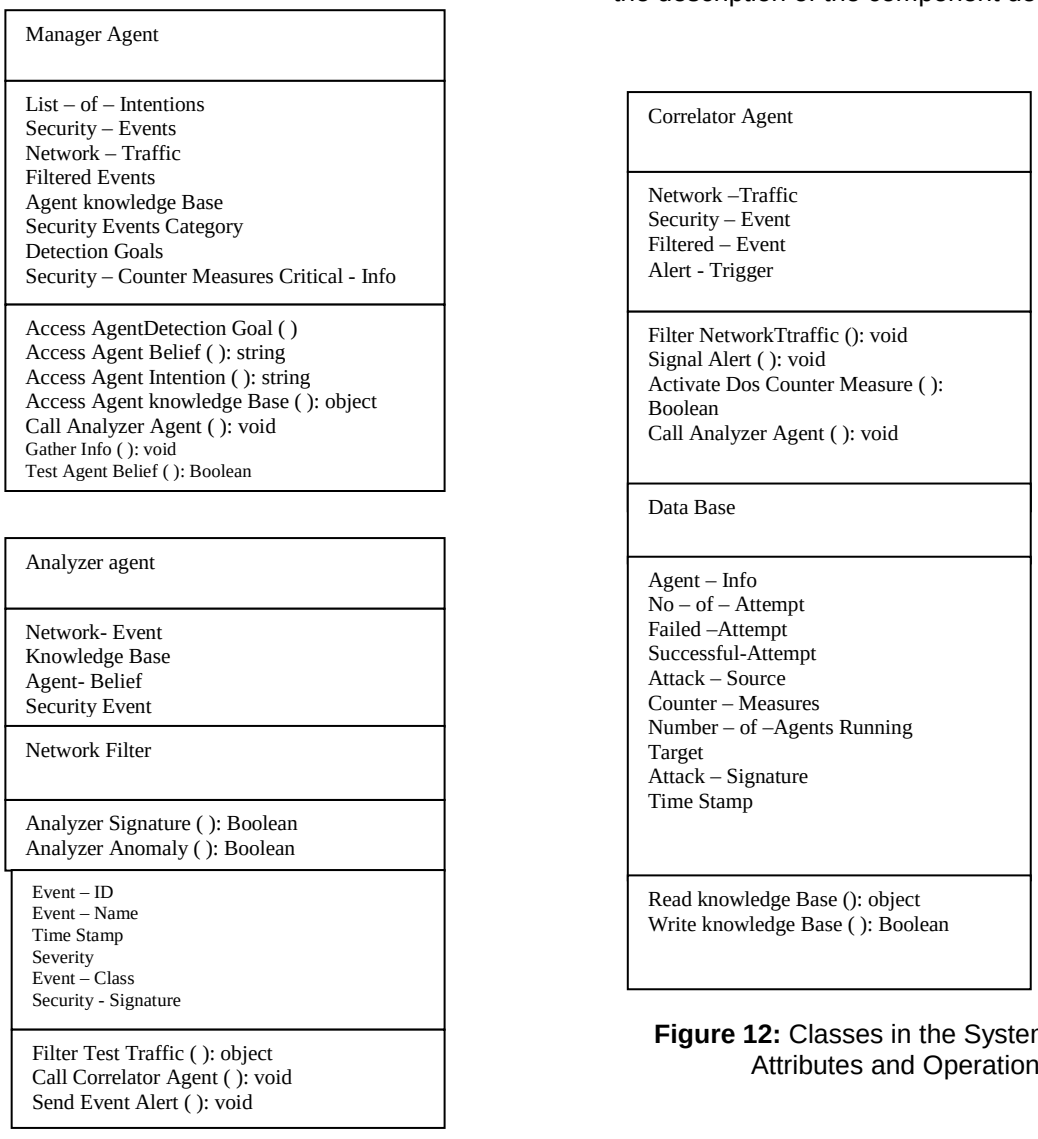

Figure 12: Classes in the System showing Attributes and Operations.

The codes that realize the systems components identified in the last section is generated in the JAVA programming language. The components are programmed as JAVA classes using the java rich API library.

For the agent classes, the classes that implement the intelligent agent classes need to run autonomously and concurrently.

Equally necessary for the agent classes is a communication infrastructure to communicate with other agents. Java multi-threading capability is used to realize the concurrent execution of the agents while the Java Remote Method Invocation (RMI) API is used to program support for inter agents' communication.

In the generation of the source code, the agent classes implement the java run able interface. The code skeleton for the manager agent:

```
Import     java. Util
Import     java. rmi. Remote.

Class ManagerAgent implements runnable extend unicate remote object
{
.
.
.
}
```

The key word extend here is used to support inheriting from the java unicaste remote object class, this is in order to support the inter agent communication.

```
Code skeleton for correlator agent
Class Correlator Agent implements runnable extend unicaste remote object.
{
.
.
}
```

The complete source code was generated using the java programming language.

RESULTS

The network security software developed was deployed on a host computer in a test network environment to protect the DHCP server from a Denial of Service (DoS) attack staged from a work station on the test network environment. The test network environment setup consists of four computers configured appropriately, It was discovered that the software was able to mitigate attacks that were generated from the workstations.

The key abilities of this adaptive network solution is that they:

- Remained active all times.
- Performed unobtrusively.
- Minimized propagation of attacks.
- Quickly responded to attacks.

CONCLUSION

The system design presentation moved progressively from data modeling (data information architecture) to more implementation specific representation that was programmed into the computer system.

The successful design development implementation and testing of the proposed multi agent based network production software has made available a formidable tool for mitigating against DoS network security attacks.

REFERENCES

1. Cisco System, Inc, 2006. "Core Element of the Cisco Self – Defending Network strategy". White paper 1992 – 2005.

2. Lammle, T. 2007. *Cisco Certified Network Associate Study Guide Sixth Edition*. "Chapter 10". 611 –613.

3. Hassan, A., J. Haidiman, and C. Papadopoulos. 2007. "A Framework for Classifier Denial of Service Attack". USC / Information Science Institute. Jan 2007.

4. Dewan, P., P. Dasgopts, and V. Karamcheti. 2003. "Defending against Denial of service Attacks Using Secure Name Resolution".

5. Spring, N., R. Mahajan, and D. Watherall. 2002. "Measuring ISP Topologies with Rocket Fuel". In: *Proceedings of ACM SIGCOM*. 02 August 2002.

6. Debar, H., M. Becker, and D. Sibon. 1992. "A Neural Network Component for an Intrusion Detection System". In: *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*.

7. Kumar, S. and E.H. Spafford. 1994. “An Application of Pattern Matching in Intrusion Detection”. In: Technical report CSDTR-94-013, Purdue University.

8. Ghosh, K. 1991. “Learning Program Behaviour Profiles for Intrusion Detection”. In: *Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring*.

9. Cohen, F. 2004. “Computer Viruses: Theory and Experiments”. 7th DOD / NBS Computer Security Conference: Gaithersburg, MD. September 24-16.

10. Helman, P., G. Liepins, and W. Richords. 1992. “Foundation of Intrusion Detection”. *The IEEE Computer Security Foundation Workshop V*.

11. Jensen, W., P. Mell, T. Karygiannis, and D. Marks. 2009. ” Applying Mobile Agents to Intrusion Detection and Response”. Technical Report, NIST interim report.

12. Oliveira, R. 1998. “Network Management with Knowledge of Requirement: Use of software Agents”. Ph.D. thesis.

13. Asaka, M., S. Okasawa, and A. Taguchi, 1999. “A Method of Tracing Intruders by Use of Mobile Agents”. In: INET’99.

14. Muller, J.P. 1996. “The Design of Intelligent Agents- A Layered Approach”. LNA1 State-of-The-Art Survey. Springer: Berlin, Germany.

15. Bernades, M.C. and E. Dos Santos Mareira. 2000. “Implementation of an Intrusion Detection System Based in Mobile Agents”. *International Symposium on Software Engineering for Parallel and Distributed System*.

16. Trapethi, A., T. Ahmed, S, Patlak, M. Carney, M. Koka, and P. Dokas.2002. “Active Monitoring of Network System using Mobile Agents”. Technical Report. University of Minnesota.

17. Wooldridge, M. and N.R. Jennirys. 1995. “Intelligent Agents: Theory and practice”. *Knowledge Engineering Review*.

18. Labroid, H. 1998. ”Error Control in Wireless ATM Networks”. Thesis. 1998.

19. Valer Bocan. 2004. “Threshold Puzzle: The Evolution of DoS- Resistant Authentication”. *PERIODICAL POLITEHNICA, Transaction for Automatic Control and Computer Science*. 49

20. Valar Bocan. 2008. “Development in DoS Research and Mitigating Technologies”.. Department of Computer Science and Engineering, Politehnica University of Timiora, Romannia.

ABOUT THE AUTHORS

Engr. (Mrs) Ogechi Ihekweaba is a Lecturer in the Department of Computer Engineering, Michael Okpara University of Agriculture, Umudike, Abia State, Nigeria. She is an ex student of Ido-Ani Federal Government Secondary School, Ondo State. Holds a Bachelor’s degree (B.Eng.) in Computer Science and Engineering, a Master’s Degree (M.Eng) in Computer Science & Engineering and she is at the verge of completing a Doctorate degree (Ph.D.) in Computer Engineering. Her area of specialization is Network Security and Computational Intelligence. She served as an Engineer with a Telecommunication outfit, CSAT COM Ltd. As a lecturer, she headed the Computer Science Department of OSISATECH. She is currently the SIWES and Seminar Coordinator for the Department of Computer Engineering, Michael Okpara University of Agriculture, Umudike. She has several publications, and is a member of professional bodies which include: Nigeria Computer Society (NCS), Computer Professionals of Nigeria (CPN), and Nigeria Society of Engineers (NSE). She is also a COREN registered Engineer.

SUGGESTED CITATION

Ogechi, I., I. Chukwugoziem, and H.C. Inyiama. 2014. “The Design and Implementation of an Adaptive Intelligent Multi-Agents for Network Protection”. *Pacific Journal of Science and Technology*. 15(2):132-143.

Pacific Journal of Science and Technology