Journal of Computing and Information Technology - CIT 14, 2006, 3, 217–223
doi:10.2498/cit.2006.03.05

Multi-level and Secured Agent-based Intrusion Detection System

Adesina Simon Sodiya

Department of Computer Science, University of Agriculture, Abeokuta, Ogun State, Nigeria

Since Intrusion Detection System (IDS) has become necessary security tool for detecting attacks on computer network and resources, it is essential to improve previous designs. Recently many mobile agent-based IDSs have been designed, but there are still some drawbacks. Some of these drawbacks are low detection efficiency, high false alarm rate and agent security. A multi-level and secured IDS architecture based on mobile agent is presented in this work to correct these drawbacks. In order to make the design more efficient, the architecture is improved in such a way that intrusion detection at the lower level and the confirmation of intrusion detected take place at the upper level. The design also incorporates data mining strategy in the identification of intrusive actions. Implementing the new design using JAVA shows a better performance than previous designs.

Keywords: computer network, attacks, IDS, mobile agent.

1. Introduction

IDS came into existence because conventional intrusion prevention techniques are not adequate to prevent attacks on computer and network systems. Intrusion detection system (IDS) is defined as a component that analyses system and user operations in computer and network systems in search of activities considered undesirable from security perspective (Sodiya et. al., 2004). IDS detects some set of intrusions and executes certain predetermined actions when an intrusion is detected. (Helmer et. al., 2003). However, because of this great work, designing effective and efficient IDS has been the major focus of researchers in the field of computer security.

Since the concept of IDS was introduced in 1980 (Anderson, 1980), many IDSs have been designed and implemented for centralized systems. The centralized IDS architecture is usually faced with the following problems:

- Limited detection efficiency,
- High number of false positive alarms,
- Limited ability to analyse and detect distributed attacks,
- IDS security.

Though the initial problem performance is now better, the management and correlation of large amount of data in distributed environment towards effective intrusion detection is still a major problem.

Applying mobile-agent to intrusion detection design is a recent development and it is aimed at effective intrusion detection in distributed environment. Intrusion detection in distributed environment requires data gathering, analysis and detection at every unit of the network. It has been established that mobile agent would perform well with this task.

Mobile Agents (MAs) are autonomous programs that can act or work independently and perform different tasks. They are also defined as software entities which function continuously and autonomously in a particular environment and are able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment (Bradshaw, 1997). MAs are by nature autonomous, collaborative, self-organising and mobile and these features are not found in the previous centralized IDSs.

Recently, many MA-based IDSs have been designed. Some of the benefits, as stated by Jansen et. al., (1999), are:

- Overcoming network latency,
- Reducing network load,
- Asynchronous execution and autonomy,
- Structure and composition,
- Adapting dynamically,
- Robust and fault-tolerant behaviour,
- Scalability.

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 in detection performance, 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 modes of execution are still very low compared to native codes (Kruegel and Toth, 2002).

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

An architecture that is called Multi-level and Secured Agent-based Intrusion Detection System (MSAIDS) is presented in this work to correct some of these shortcomings.

The rest of this paper is organized as follows. Related works are presented in section 2. MSAIDS architecture is described in section 3. The implementation procedure and evaluation of the design are presented in section 4. In section 5, future work is presented and the work is concluded in section 6.

2. Related Works

Several MA-IDS architectures have been designed and implemented in different projects. An automatic tool was designed for generating mobile agents for distributed IDS in Wang et al., 2006. The real performance of the agents generated for IDS in terms of false positive and negative was not mentioned. The JAM project at Columbia University is a typical MA-IDS (Stolfo et al., 1997; Lee and Stolfo, 1998). The project used intelligent, distributed Java agents and data mining to learn models of fraud and intrusive behaviours that can be shared between organizations. Another similar project called Intrusion Detection Agent (IDA) was designed by Information-technology Promotion Agency (IPA) in Japan (Asaka et. Al., 1999). IDA is a hierarchical architecture that relies on mobile agents to trace intruders among various hosts. Another architecture that is capable of efficient and flexible distribution of analysis and monitoring tasks, as well as of integration of existing detection techniques, was presented in Deeter et. al., 2004. The work addressed the problems of bandwidth scalability, processing scalability, analysis delay and integration. An approach where decision making and distributed autonomous agents are charged with the role of investigating intrusions was implemented in Micael (de Queiroz et al., 1999). The approach was based on three main components known as sentinels, detachments and headquarters. It provided an interesting architecture but no implementation has been seen so far. Security Policy Adaptation Reinforced Through Agents (Sparta) is an IDS project sponsored by the European Union (Kruegel et al., 2002). It was primarily designed to detect security violations in heterogeneous networks. Most of these designs are still not perfect because they are still faced with one or more of the problems stated in section 1.

Using lightweight agents was first presented in Autonomous Agents for Intrusion Detection (AAFID) (Balasubramaya et. al., 1998). This is another classical IDS with agents used mainly as a means for structuring the intrusion collection components into a set of lightweight software components which can easily be reconfigured. Another approach that implemented lightweight technique was presented by Helmer et. al., (2003). The approach was intended to use lightweight agents so as to improve high time to detection. Each lightweight agent was used to carry out little IDS task in order to improve efficiency. Using lightweight agents typically has the problem of performance, especially when different operating platforms are involved in the distributed environment. Therefore, all these works need improvement.

3. MSAIDS Design Methodology

The significant interests of this design are:

a. ***Improving IDS performance:*** An IDS that is functionally correct, but detects attacks too slowly, is of little use (Jansen et. al., 1999). It is necessary to improve time-to-detection of MA-IDS.

b. ***Detection of autonomous attacks:*** The aim of MSAIDS is to provide an architecture where autonomous attacks are efficiently detected.

c. ***Reduction in false alarm:*** The work also aims at reducing false alarm rates effectively.

d. ***IDS agents security:*** Since the role of IDS is to monitor and ensure security of the network, the IDS itself is a primary target of the attacks. It is important for IDS agents to operate in hostile environment and still exhibit a high degree of fault-tolerance and performance. A major consideration of this work is to present an architecture that provides agents protection.

3.1. System Architecture

The components of MSAIDS architecture are shown in Figure 1 below.

The architecture provides a methodology where intrusion detection is done at two levels viz: the Lower Level Detection (LLD) and Upper Level Detection (ULD).

a. Lower Level Detection (LLD)

The main focus of the LLD is to have some agents to first detect intrusions independently and subsequently report back to the ULD for further investigation and confirmation of intrusions. At this level, there are two categories of agents: the data agents and processing agents. There are four data agents and they move around the nodes in the network to collect associated information from application messages, authentication events, system calls, TCP connections and others. There are two processing agents working at this level and they are known as Node Agents. The first node agent (Node-1-agent) is responsible for the construction of the first level database. The agent collects the data from the data agents to build a database with the information collected. This agent is also in charge of data cleansing, classification and formatting.

The second agent (Node-2-agent) is responsible for data mining and first level intrusion detection. It applies data mining algorithm as presented in Sodiya et al., 2004, on the information in the data warehouse and communicates the possibility of intrusions to the interface agent through the Alarm Agent (AA).

b. Upper Level Detection (ULD)

This level is also known as confirmation level and it is involved in separate intrusion detection process. As shown in Figure 1, there are low-level agents that are responsible for separate data collection from different sources. It is essential not to rely on the data gathered by node-1-agent in the LLD. These low level agents gather data from the data agents and inform the Controller and Protector (CP) about the nature of data gathered. The CP acts as the facilitator agent. The CP passes the data to the data mining agent for storage and mining activities. The data mining agent then applies data mining algorithm on the database so as to extract patterns or associations of intrusive events. If there is any intrusion suspected, it is reported by the Interface Agent (IA) to the Site Security Officer (SSO).

The database at the ULD provides another view of the knowledge and activity of the monitored distributed network.

The interesting thing about this architecture is that if there is no signal or alarm from LLD, the ULD does not check intrusions. The low level agents just constantly gather data and these data are used to update the ULD database. A full intrusions check is not initiated if there is no trigger from the LLD. This enables few agents to be active at the same time and ease the problem of protocol and language definitions associated with the communication between agents.

The design considered the importance of providing adequate security to agents because mobile code generally introduces a number of secu-

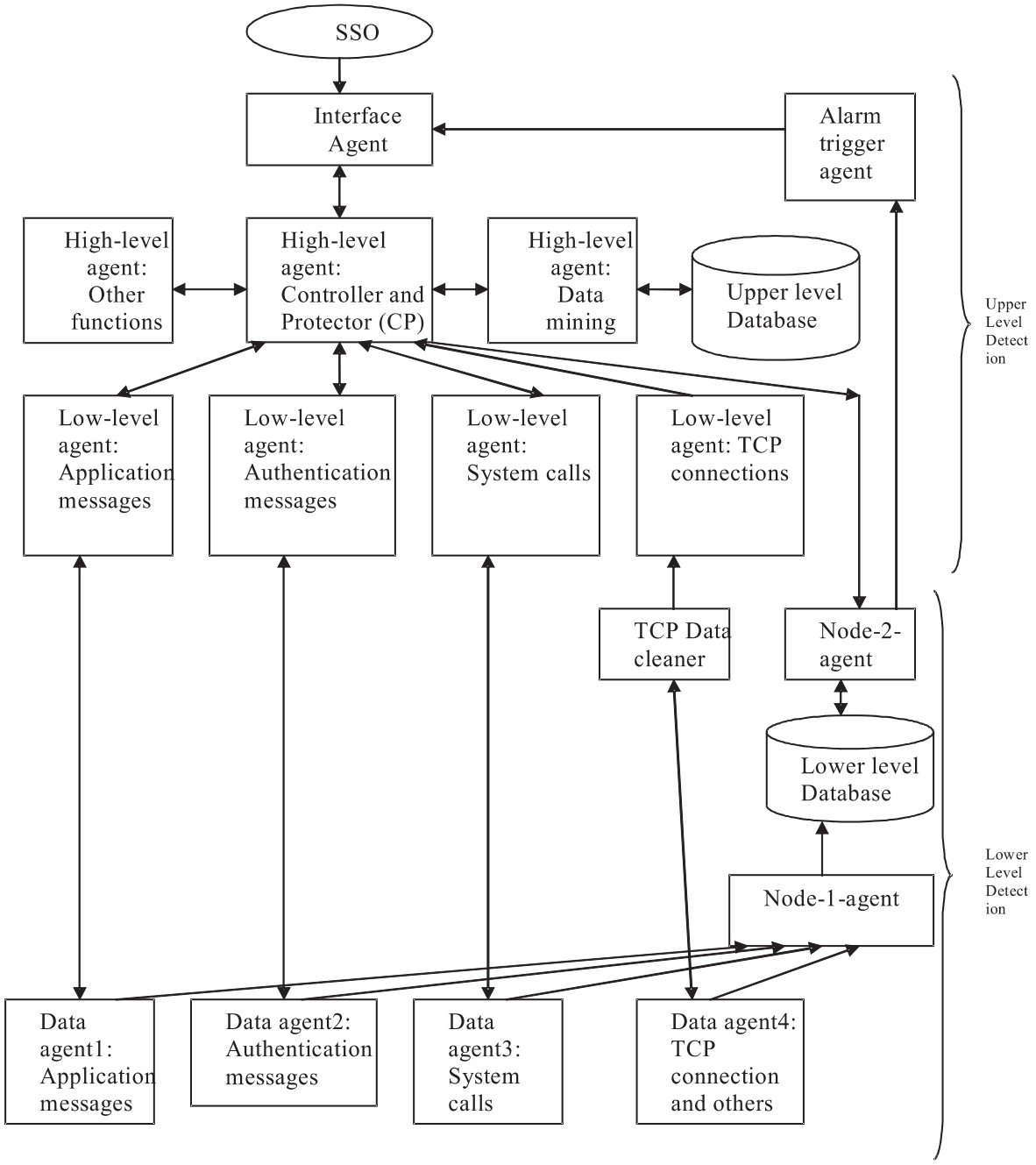

Fig. 1. MSAIDS Architecture.

rity issues. The security threat to mobile agents are classified into four categories: agent-to-agent, agent-to-platform, platform-to-agent and other-to-agent (Helmer et al., 2003). MSAIDS maintains security of agents by using asymmetric (Public and Private key pair) cryptosystem when they move around the network. This technique is also used in Sparta. In addition to this, agents' states are normally recorded and authenticated before they are initiated. This is done so as to monitor deviation from the normal state and role when carrying out their responsibilities on the network. When an agent is initiated, the CP stores the size and structure of the mobile code. The CP also ensures proper communication and delivery of service among agents. The agents in MSAIDS are protected in such a way that the attributes of a particular agent are visible only to the group of agents it will communicate with.

3.2. Agents Communication

The eXtensible Markup Language (XML) is used in MSAIDS as agent communication language. Communication in this work requires informing and requesting rather than invoking. An agent contacts another agent in order to obtain or deliver information. Agents' knowledge of the service to render is contained within the configuration file. An agent's request for information or service is defined or encoded in XML format and it is transported to the provider using SOAP – Simple Object Access Protocol (Box et al., 2000). The providing agent then processes the details of the request and returns the information or service, also in XML format. The communication between agents is clear and tidy. The communication method supports multiple agents with different roles. The agent communication method of MSAIDS is described in Figure 2 below.

3.3. Data Mining Process

As seen in Figure 1, data are collected from different sources within the distributed nodes to the low-level database, which is then used for intrusion detection by node-2-agent. Database is also constructed by the CP-agent using the data gathered by the specific data gathering agents. In both levels of detection, the same data mining algorithm is implemented to detect intrusion. A modified apriori algorithm, as proposed in Sodiya and Longe (2005), is used to discover intrusive patterns by both the data mining agent and node-2-agent.

All valid or permissible rules and their thresholds are specified within the Permissible Rules Definition (PRD). The PRD is a component of the CP. So, the two mining agents relate with the CP from time to time to get the valid rules. The PRD allows the SSO to formulate and describe rules for normal events for system calls, authentication, TCP connection, etc., and for their relationships. The patterns discovered are then compared with the rules in the PRD to monitor the deviation from valid or normal events. If the node-2-agent finds a deviation, it signals the alarm agent, which then communicates the possibility of intrusion to the IA. The IA automatically informs the SSO of the first level intrusion suspected. As mentioned earlier, the IA will then signal the CP to initiate a confirmatory check.

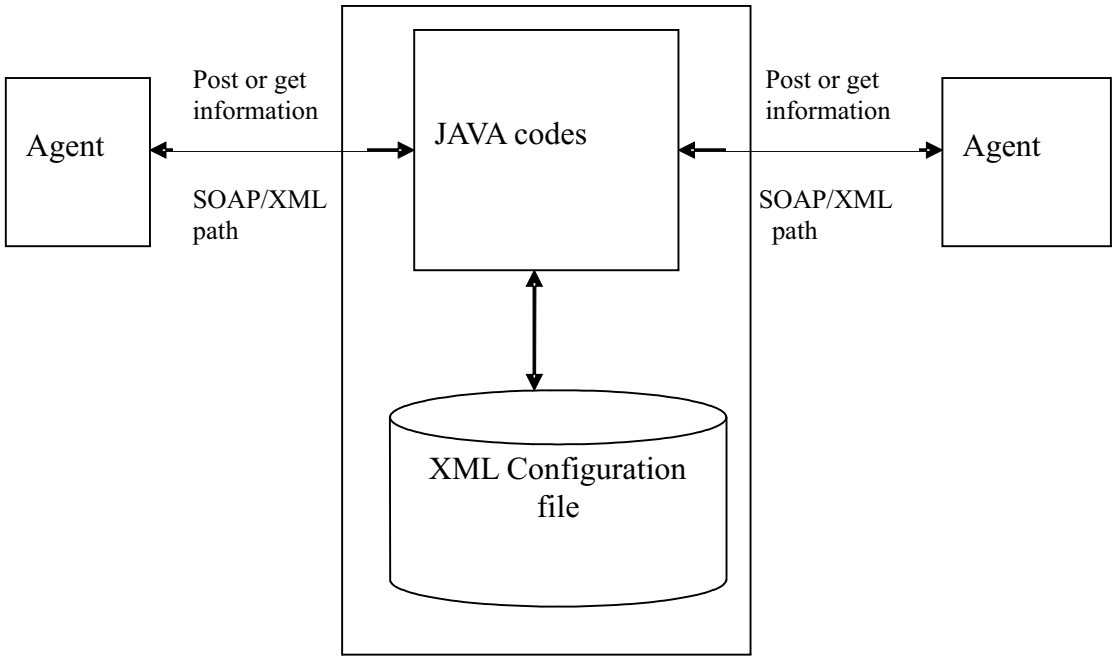

Fig. 2. MSAID Agent Communication Architecture using XML.

4. Implementation and Evaluation

MSAIDS was implemented using Java Development Kit 1.1.5. Some of the reasons for using JAVA for the implementation are:

- It provides high security for development and implementation,
- It supports easily agent development,
- It is platform independent,
- Its object orientation assists in structuring agents and their behaviour.

4.1. Data Source

Data were collected from the Computer Unit, University of Agriculture, Abeokuta, Nigeria, which has very large volume of network traffic. The data were classified as normal and intrusive. The data consisted of previous 3 months system calls, authentication and TCP connections. The normal data were used to encode rules in the PRD. This was done by implementing the mining algorithm separately on the normal data to identify these patterns or rules. 189 rules were encoded in the PRD.

4.2. Evaluation

A five-workstation network environment was used as the test bed. Summary of the results after implementing the design is as follows:

a. Intrusion Detection Ability

The system was able to detect most of the intrusive events introduced at different nodes. The gathered intrusive actions were introduced at different times and performance of the system was monitored. Using related data sources, as in other previous designs, the analysis is given as follows:

	Data sources	Intrusions initiated	Percentage detected
1	System calls	3694	98%
2	Logins	11006	97%
3	TCP connections	7700	100%

This shows a better performance than in some previous works, such as Li (2004), in which 94.4% detection rate was observed.

b. False Alarm Rate

The analysis of false alarm rate is given as follows:

	Data sources	False Alarm Rate
1	System calls	0.05
2	Logins	0.16
3	TCP connections	0.0

This is lower than 0.83% reported on system calls in Helmer et al., (2003). MSAIDS has also shown an improvement on false alarm.

c. System Capacity Evaluation

It took an average of 0.14 seconds to report an intrusion at the LLD and 0.75 seconds at the ULD. Intrusions are detected very fast in MSAIDS and it is believed that the system will work well even when the traffic is high.

5. Future work

Mobile agent-based intrusion detection technology is still undergoing a lot of improvement. Using other AI techniques apart from data mining should still be implemented for IDS design. Increasing the detection efficiency and security should still be the major focus of researchers in IDS. It might also be interesting to compare different MA-IDSs using the same data and under the same environment.

6. Conclusion

The design of MSAIDS has shown that MA technology is an efficient tool for building IDS infrastructure. Since MSAIDS has shown an improvement in comparison with some previous works, multi-level intrusion detection using mobile agents has proven to be efficient. The approach presented also improves scalability of the system because new hosts on the network would not automatically cause additional traffic

to the IDS. This design would help system administrators and SSO to spot and defend from attacks as well as to assist them in developing better protection and countermeasures for their systems in recognizing new attacks.

References

[1] ANDERSON J.P., *Computer security threat monitoring and surveillance*, Technical report, James P. Anderson co, Box 42, Fort Washington, February 1980.

[2] ASAKA, M., OKAZAWA, S., TAGUCHI, A. AND GOTO, S., *A method for tracing intruders by use of mobile agents*, INET'99, June 1999.

[3] BALASUBRAMANIYA, J., GARCIA-FERNANDEZ, J.O., ISACOFF, D., SPAFFORD, E.H. AND ZAMBONI, D., *An architecture for intrusion detection using autonomous agents*, Department of Computer Science, Purdue University, Coast TR 98-05, (1998), `http://www.cs.purdue.edu/coast/coast-library.html`.

[4] BOX, D., EHNEBUSKE, D., KAKIVAYA, G., LAYMAN, A., MENDELSOHN, N., NIELSEN, F., THATTE, S., AND WINER, D., *Simple Object Access Protocol (SOAP) 1.2*, (2000), `http://www.w3.org/TR/2000/NOTE-SOAP-20000508/`.

[5] BRADSHAW, M.J., *An introduction to Software agents*, In Jeffrey M. Bradshaw, Editor, Software agents, Chapter 1, AAAI press, The MIT press, (1997).

[6] DEETER, K., SINGH, K., WILLSON, S., FILIPOZZI, L., AND VUONG, S., *APHIDS: A Mobile Agent-Based Programmable Hybrid Intrusion Detection*, (2004), `http://www.cc.gatech.edu/grads/k/ksingh/publication/aphids_cameraready.pdf`.

[7] DE QUERIOZ, J.D., DA COSTA CARMO, L.F.R. AND PIRMEZ, L., Micael: An autonomous mobile agent system to protect new generation networked applications, In *2nd annual workshop on Recent Advances in Intrusion Detection*, September 1999.

[8] HELMER, G., WONG J.S.K., HONAVAR V., MILLER L. AND WANG, Y., Lightweight agents for intrusion detection, *Journal of Systems and Software*, (2003), `www.elsevier.com/locate/jss`.

[9] JANSEN W., MELL, P., KARYGIANNIS, T. AND MARKS, D., *Applying Mobile Agent to Intrusion Detection and Response*, NIST Interim Report (IR) – 6416, (1999).

[10] KRUEGEL, C. AND TOTH, T., *Applying Mobile Agent Technology to Intrustion Detection*, Technical Report Number TUV-1841-2002-31, Technical University of Vienna, (2002).

[11] KRUEGEL, C., TOTH, T. AND KIRDA, E., *Sparta – A Mobile Agent-based Intrusion Detection System*, Technical Report Number TUV-1841-2002-24, Technical University of Vienna, (2002).

[12] LEE, W., AND STOLFO, Data Mining Approaches for Intrusion Detection, In *Pro-ss98, pub-USENIX*, (1998).

[13] SODIYA, A.S., LONGE H.O.D. AND AKINWALE A.T., A New Two-tiered Strategy to Intrusion Detection, *Emerald Journal of Information Management and Computer Security*, Vol. 12, No. 1, (2004).

[14] SODIYA, A.S. AND LONGE H.O.D., An Improved Two-tiered Strategy to Intrusion Detection, *Emerald Journal of Information Management and Computer Security*, Vol. 13, No. 1, (2005).

[15] STOLFO, S.J., PRODROMIDIS, A.L., TSELEPIS, S., LEE, W., FAN, D., AND CHAN, P.K., JAM: Java agents for meta-learning over distributed databases, In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Newport Beach, CA, USA, August 1997.

[16] WANG, W., BEHERA, S. R., WONG, J., HELMER, G., HONAVAR, V., MILLER, L., LUTZ, R., AND SLAGEL, M., Towards the Automatic Generation of Mobile Agents for Distributed Intrusion Detection System, *Journal of Systems and Software*, 79 (2006), pp. 1–14, `www.elsevier.com/locate/jss`.

[17] WEISS, G. *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, London, 2001.

Received: October, 2005
Accepted: May, 2006

Contact address:
A. S. Sodiya
Department of Computer Science
University of Agriculture, Abeokuta
Ogun State, Nigeria
e-mail: `sinaronke@yahoo.co.uk`

SODIYA ADESINA SIMON has PhD in Computer Science and presently lectures in the department of Computer Science, University of Agriculture, Abeokuta, Ogun State, Nigeria. He has published in both local and international journals. His areas of interest are Computer Security, Software Engineering and Artificial Intelligence. He is married with children.