www.ijsrnsc.org

Available online at www.ijsrnsc.org

IJSRNSC

Volume-1, Issue-4, Oct- 2013
Research Paper

Int. J. Sci. Res. in Network Security and Communication

ISSN: 2321-3256

Architecture for Network-Intrusion Detection and Response in open Networks using Analyzer Mobile Agents

M.Shivakumar[1*], R.Subalakshmi[2], S. Shanthakumari[3] and S.John Joseph[4]

[1*]*Dept. of IT, Sudharsan Engineering College, Anna university, India,*
[2]*Information Technology, Kuppam Engineering College, Andhra Pradesh, India*
[3]*Department of IT, Kuppam Engineering College, Andhra Pradesh, India*
[4] *Department of IT, Sudharsan Engineering College, Anna University, India*

Received: 12 September 2013 Revised: 17 September 2013 Accepted: 22 October 2013 Published: 30 October 2013

***Abstract*—**This paper describes the implementation of distributed agent architecture for intrusion detection and response in networked computers. Unlike conventional intrusion detection systems (IDS), this security system attempts to emulate mechanisms of the natural immune system using Java-based agents. These security agents monitor multiple levels like packet, process, system, user etc. of networked computers to determine correlation among the observed anomalous patterns, reporting such abnormal behavior to the network administrator and possibly taking some action to counter a suspected security violation. Here the focus is on the design aspects of such an intrusion detection system by integrating different artificial intelligence techniques and a mobile agent architecture. Here anomaly based intrusion mechanism will be taken into consideration with the help of agents.

Keywords- Component; Mobile Agents (MA), Intrusion Detection, Distributed Systems

I. INTRODUCTION *(HEADING 1)*

An intrusion is somebody (“hacker" or "cracker") attempting to break into or misuse your system. The word "misuse" is broad, and can reflect something severe as stealing confidential data to something minor such as misusing your email system for spam (though for many of us, that is a major issue!). With the emergence of Internet and the World Wide Web, the concept of Global village has taken its inception. There are facilities to virtually achieve any kind of information on the internet. All these advantages have been achieved because of networking computers and associated devices. There has been a rapid progress in this field. Along with this, there is the arms race between the intruders and people who provide security to the systems in networks. This project IDS (detection and protection) [2, 3] runs on the host machines and assists the network Administrators to detect several intrusion attacks and inform to the owner of the system and also provide security by blocking the malicious users based on their IP addresses.

The concept of creating an intrusion detection system was first proposed in 1980 by James Anderson [4]. However, the field did not take off until 1987 when Dorothy Denning published an intrusion detection model [10]. In 1988, at least three IDS prototypes were created [6] [20] [21]. In the following years, an ever-increasing number of research prototypes were explored. The US government, realizing that its computer systems were insecure, provided significant funding for research in IDSs. Hundreds of millions of dollars have probably been spent on IDS research within the last ten years. Because intrusion detection has become a mature industry and a proven technology, nearly all of the easy problems have been solved. No major breakthroughs in intrusion detection research have recently been made. Instead, commercial companies are mostly perfecting existing intrusion detection techniques. With the maturation of the intrusion detection field, traditional lines of intrusion detection research are having diminishing returns. Therefore, future intrusion detection research is expected to focus on relatively unexplored areas such as:

- Attack response mechanisms,
- Architectures for highly distributed intrusion detection systems,
- Intrusion detection inter-operability standards, and
- New paradigms for performing intrusion detection.

II. MOBILE AGENT TECHNOLOGY

IDSs implemented using MAs is one of the new paradigms for intrusion detection. MAs are a particular type of software agent, having the capability to move from one host to another. A software agent can be defined as [7]: “… a software entity which functions continuously and autonomously in a particular environment … able to carry out activities in a flexible and intelligent manner that is responsive to changes in the environment … Ideally, an agent that functions continuously … would be able to learn from its experience. In addition, we expect an agent that inhabits an environment with other agents and processes to be able to communicate and cooperate with them, and

Corresponding Author: *Dr. M. Shivakumar*

perhaps move from place to place in doing so." Mobile agents have been a research topic of interest for several years, yet this research has for the most part remained within laboratories and has not experienced a wide-scale adoption by industry. The development of the World Wide Web application, however, has dramatically stimulated interest in this area of research by offering the possibility of a widely deployed application that could use mobile agent technology. The research community visualizes mobile agents launched via web browsers to gather information and interact with any node in the network. IBM and General Magic were early pioneers of this vision, [8, 15]. Concurrent with this effort, ARPA sponsored a Knowledge Sharing program. The KQML language [13] was developed under this program and remains one of the viable Agent Communication Languages (ACLs). This research area was reformulated in the '95-'96 time frame when Java was released by Sun Microsystems. Although Java is simply a new interpreted computer language, it is designed for network interactions and is a powerful enabling technology for mobile code. Web browsers were quickly "Java-enabled" and the IT community seemed convinced that mobile code would quickly become a reality. The Java language provided some system independence and considerable security features were included in the language and implementations. These are not unique features, of course, they simply were implemented better in Java than other languages and so Java became extremely popular. During this same period, numerous proposals for mobile agent implementations were fielded. For example, the Lava system [23, 14] was developed at North Carolina State University. This system focused on security problems and developed a simple security policy for applets. Mitre Corporation [11, 12] also pursued work in this area, developing authentication mechanisms and defining a taxonomy of security related problems. An important observation to make about most of the early work in this field is the assumption made by most researchers about a totally open system. That is, the security problems being addressed are those found in a system with open connectivity and with the maximum possible threats. Several researchers reached conclusions indicating that the paradigm was not useful since there were always certain threats that could not be adequately countered while maintaining a totally open system. Partly because of these conclusions, as well as well publicized attacks against early Java enabled systems, security related problems have hindered the widespread adoption of MA technology. Security architectures have been defined, but they contain too much residual risk for most applications. Recent work at the University of Tulsa, for example, proposes using mobile agents for data mining purposes. Such an application requires providers of information to keep their systems "open" to a multitude of users, most of whom are unknown to the host. A good overview of current mobile agent projects and technology is provided in [18].

However, relatively little work has been done on using a mobile agent architecture for the purpose of providing a security capability, such as intrusion detection. If a mobile agent architecture is designed for a specific purpose such as system administration or security function maintenance, then strong authentication may be enforced and the residual risk decreases significantly.

While MAs are an extraordinarily powerful tool, their implementation has been hindered by security considerations. These security considerations are especially critical for intrusion detection systems, with the result that most security research in this field has concentrated upon the architecture necessary to provide security for mobile agents. We claim that such negative results are not fatal to the proposed study since these security issues are likely to be addressed by the research community and there will be few authorized users of the MA-based IDSs within an organization.

A. Java Agents for Meta Learning

The Java Agents for Meta-Learning (JAM) project [17] at Columbia University, NY, applies meta-learning to distributed data mining, using intelligent agents. Intelligent agents employ artificial intelligence techniques to model knowledge and reasoning, as well as behavior, in multi-agent societies or domains. The design has two key components: local fraud detection agents that learn how to detect fraud and provide intrusion detection services within a single corporate information system, and a secure, integrated meta-learning system that combines the collective knowledge acquired by individual local agents. Data mining, like neural networks and other single-point learning applications, does not enable knowledge sharing among agents. The meta-learning approach attempts to overcome this limitation by integrating a number of separately learned classifiers embodied as remote agents.

III. INTRUSION DETECTION AGENT SYSTEM

The Information-technology Promotion Agency (IPA) in Japan, is developing an IDS called the Intrusion Detection Agent system (IDA) [5]. The IDA is a multi-host based IDS. Instead of analyzing all of the users' activities, IDA works by watching specific events that may relate to intrusions, referred to as Marks Left by Suspected Intruder (MLSI). If an MLSI is found, IDA gathers information related to the MLSI, analyzes the information, and decides whether or not an intrusion has occurred. The IDA system relies on mobile agents to trace intruders among the various hosts involved in an intrusion and to gather information. The architecture is hierarchical, with a central manager at the root and a variety of agents at the leaves. A sensor is an agent that resides at a node in search of MLSIs. Upon discovery of such information, the sensor notifies the manager who dispatches a tracing agent to the host. The tracing agent initiates an information-gathering agent to collect related information at

the host, before moving onto any other site identified as a suspected point of origin. The manager collects and integrates the results from the information-gathering agent as they return. Possible duplication caused by multiple sensors detecting the same intrusion is resolved through a message board at each monitored host. The developers indicate that the resulting multiagent system is an efficient and effective way for detecting intrusions.

A. IDS Requirements

At least one past effort has identified desirable characteristics for an IDS. In [9], the authors indicate that, regardless on what mechanisms an IDS is based, it must do the following:

- Run continuously without human supervision,
- Be fault tolerant and survivable,
- Resist subversion,
- Impose minimal overhead,
- Observe deviations from normal behavior,
- Be easily tailored to a specific network,
- Adapt to changes over time, and
- Be difficult to fool.

We have developed a similar set of requirements along two themes: functional and performance requirements.

B. Functional Requirements

As the network-computing environment increases in complexity, so do the functional requirements of IDSs. Common functional requirements of an IDS being deployed in current or near-term operational computing environments (see Appendix A for more information on the operational environments envisioned) include the following:

- The IDS must continuously monitor and report intrusions.
- The IDS must supply enough information to repair the system, determine the extent of damage, and establish responsibility for the intrusion.
- The IDS should be modular and configurable as each host and network segment will require their own tests and these tests will need to be continuously upgraded and eventually replaced with new tests.
- Since the IDS is assigned the critical role of monitoring the security state of the network, the IDS itself is a primary target of attack. The IDS must be able to operate in a hostile computing environment and exhibit a high degree of fault-tolerance and allow for graceful degradation.
- The IDS should be adaptive to network topology and configuration changes as computing elements are dynamically added and removed from the network.
- Anomaly detection systems should have a very low false alarm rate. Given the projected increase in network connectivity and traffic, simply decreasing the percentage of overall false alarms may not be sufficient as their absolute number may continue to rise.
- The IDS should be able to learn from past experiences and improve its detection capabilities over time. A self-tuning IDS will be able to learning from false alarms with the guidance of system administrators and eventually on its own.
- The IDS should be able to be easily and frequently updated with attack signatures as new security advisories and security patches become available and new vulnerabilities and attacks are discovered.
- Decision support tools will be necessary to help system administrators respond to various attacks. The IDS will be required not only to detect anomalous events, but also to take automated corrective action.
- The IDS should be able to perform data fusion and be able to process information from multiple and distributed data sources such as firewalls, routers, and switches. As real-time detection demands push networked-based solutions to re-programmable hardware devices that can download new capabilities, the IDS will need to be able to communicate with the hardware-based devices.
- Data reduction tools will be necessary to help the IDS process the information gathered from data fusion techniques. Data mining tools will be helpful in running statistical analysis tools on archived data in support of anomaly detection techniques.
- The IDS should be capable of providing an automated response to suspicious activity. Rapid changes in network conditions and limited network administration expertise make it difficult for system administrators to diagnose problems and take corrective action to minimize the damage that intruders can cause.
- The ability to detect and react to distributed and coordinated attacks will become necessary. Coordinated attacks against a network will be able to marshal greater forces and launch many more and varied attacks against a single target. These attacks can be permutations of known attacks, be rapidly evolving, and be launched at little cost to the attackers.
- Distributing the computational load and the diagnostic capabilities to agents scattered throughout the network adds a level of fault-tolerance, but it is often necessary for the system administrator to have control over the IDS from a central location.
- The IDS should be able to work with other Commercial Off-the-Shelf (COTS) security tools, as no vendor toolset is likely to excel in or to provide complete coverage of the detection, diagnosis, and response responsibilities. The IDS framework should be able to integrate various data reduction, forensic, host-based, and network-based security tools. Interoperability and conformance to standards will further increase the value of the IDS.

- IDS data often requires additional analysis to assess any damage to the network after an intrusion has been detected. Although the anomalous event was the first detected, it may not be the first attempt to gain unauthorized access to the network. Post event analysis will be needed to identify compromised machines before the network can be restored to a safe condition.
- The IDS itself must also be designed with security in mind. For example, the IDS must be able to authenticate the administrator, audit administrator actions, mutually authenticate IDS devices, protect the IDS data, and not create additional vulnerabilities.

C. Perfromance Requirements

An IDS that is functionally correct, but that detects attacks too slowly is of little use. Thus we must enumerate several performance requirements for IDSs. The IDS performance requirements include:

- To the extent possible, anomalous events or breaches in security should be detected in real-time and reported immediately to minimize the damage to the network and the loss or corruption of confidential information.
- The IDS must not place undue burden or interfere with the normal operations for which the systems were bought and deployed to begin with. This requirement makes it necessary for the agents to be cognizant of the consumption of network resources for which they are competing. There is a tradeoff between additional levels of security monitoring and the performance penalty to be paid by other applications.

The IDS must be scalable. As new computing devices are added to the network, the IDS must be able to handle the additional computational and communication load.

IV. MOBILE AGENTS FOR INTRUSION DETECTION

For mobile agents to be useful for intrusion detection, it is necessary that many, if not all, hosts and network devices are installed with an MA platform. This is not a far-fetched assumption because an MA platform is general-purpose software that enables organizations to implement many different applications. If MAs become popular, every new host may come preinstalled with a MA platform just as today most personal computers come bundled with a Java interpreter in the web browser. Contrast this to many IDS schemes that assume that a host-based IDS is installed on every host. It is generally too expensive to install a proprietary solution (like a host-based IDS) on every host in a network, but it is not unusual to install a general-purpose interpreter (like an MA platform and Java virtual machine) on every host.

A. Advantages

A number of advantages of using mobile code and mobile agent computing paradigms have been proposed [16,22]. These advantages include: overcoming network latency, reducing network load, executing asynchronously and autonomously, adapting dynamically, operating in heterogeneous environments, and having robust and fault-tolerant behavior. This section examines these claims and evaluates their applicability to the design of ID systems.

V. INNOVATIONS IN INTRUSION DETECTION SYSTEMS

Intrusion detection systems are less than perfect. [19] outlines a number of shortcomings of currently deployed IDSs, which are summarized as follows:

- No generic building methodology,
- Lack of efficiency,
- Lack of portability among monitored environments,
- Limited flexibility (includes tailor ability, scalability, and dynamic re-configurability),
- Limited upgradability of detection techniques,
- Difficult maintenance of rule sets,
- No performance and coverage benchmarks, and
- No good way to test effectiveness.

Developers continue to solve some of these shortcomings through the refinement of existing techniques, but some shortcomings are inherent in the way IDSs are constructed. While mobile agents can help improve IDSs in many areas, they offer no help in others. For example, the ability of an IDS to detect attacks from a single vantage point, by looking at information from a single host, a single application, or a single network interface (i.e., single point detection), is the primary problem facing IDS manufacturers. Mobile agent technology cannot enhance the ability of an IDS to perform single point detection of attacks or reduce false positive rates. Moreover, in most cases, mobile agent technology slows down the ability of an IDS to process events thereby actually decreasing its detection ability. This is a severe limitation for single point IDSs attempting to evaluate events in real time. This does not mean that MAs are not useful to IDSs. MAs can solve several major problems with IDSs, but more importantly, as discussed below, they can provide IDSs with performance benefits and heretofore unseen capabilities. For example, the mobility of agents make them ideal for detection schemes that follow a “cop on the beat,” “immune system,” or other model.

A. Network Intrusion:

A deliberate attempt to enter a network and break the security of the network and thus breaking the confidentiality of the information present in the systems of the network. The person who tries to attempt such an action is called as an Intruder and the action can be termed as Network Intrusion. The network administrator is supposed to protect his network from such persons and this software can help his in his efforts.

B. Intrusion detection systems (IDS)

An Intrusion Detection System (IDS) is a system that is responsible for detecting anomalous, inappropriate, or other data that may be considered unauthorized occurring on a network. An IDS captures and inspects all traffic, regardless of whether it's permitted or not. Based on the contents, at either the IP or application level, an alert is generated. An IDS can be classified based on the input data or the detection mechanism. Depending on the data used by an IDS, the IDS can be classified into a network IDS or host-based IDS.

C. Network IDS:

A network IDS analyzes the data transmitted over a network. A network IDS can protect a big network, a LAN, or a single host. Data used by a network IDS includes, packet header data, packet statistics, and application layer payload data. Various network statistics such as rate of incoming packets, rate of failed connections, and average session length can also be used for detection purposes.

D. Host-based IDS:

A host-based IDS is deployed on the host machine to be protected. Various host-related data like commands executed, CPU usage, hard-disk access, memory usage, audit logs and others can be used by a host-based IDS to detect an intrusion. Sequence of system calls executed by a program can also be used for the detection of an intrusion.

E. Based on the detection mechanism, an IDS can be classified into a misuse IDS or an anomaly IDS.

- *isuse IDS (or Signature IDS):*

A misuse detector uses known patterns of attacks called signatures to catch intrusions. A misuse IDS generates signatures from a given set of attacks. While monitoring, it checks if an attack pattern is present in the monitored data and takes appropriate action when a signature is matched. Hence, misuse IDSs can only detect known attacks.

- *Anomaly IDS:*

An anomaly detector records the normal usage patterns of the system. Any system usage which deviates significantly from the normal profile is considered a possible intrusion and an alarm is raised. Unlike a misuse IDS, an anomaly IDS does not require knowledge of attack patterns and thus can possibly detect new attacks.

- *Need for an IDS:*

Intrusion detection devices are an integral part of any network. The internet is constantly evolving, and new vulnerabilities and exploits are found regularly. They provide an additional level of protection to detect the presence of an intruder, and help to provide accountability for the attacker's action.

- *Four different types of attacks have been identified which makes the need for an IDS critical.*

✓ Denial of service

Network-based denial-of-service [1, 2, 3] attacks are one of the easiest types of attacks. It often requires little effort to fully consume resources on the target computer, to starve the target computer of resources, or to cause critical services to fail or malfunction. Internal corporate networks typically do not have internal filtering defenses against common denial-of-service attacks, such as flooding.

✓ Threat to Confidentiality

Some viruses attach themselves to existing files on the system they infect and they send the infected files to others. This can result in confidential [1] information being distributed without the author's permission.

✓ Modification of contents

Intruders might be able to modify news sites, produce bogus press releases, and conduct other activities, all of which could have economic impact.

✓ Masquerade

A masquerade [1, 2, 3] takes place when one entity pretends to be a different entity. Authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges. Any system connected to the internet and providing TCP-based network services (such as a Web server, FTP server, or mail server) is potentially subject to this attack. Note that in addition to attacks launched at specific hosts, these attacks could also be launched against your routers or other network server systems if these hosts enable (or turn on) other TCP services (e.g., echo).The consequences of the attack may vary depending on the system; however, the attack itself is fundamental to the TCP protocol used by all systems.

VI. PURPOSE AND SCOPE OF THIS SYSTEM.

✓ Purpose of the system:

The purpose of the system is to detect certain well-known intrusion attacks on the host system and display warnings to the user and also store information regarding the IP addresses and allow the traffic based on that information.

✓ Scope of the system:

The system frames certain rules based upon the input given by the user. It then allows traffic inwards or outwards based upon the rules. The system also detects certain well-known attacks and gives warnings to the user. The rules defined by the system are intact can be used by the Agents who will be always monitoring the network taking care of deviations happening in the network. The Agents here I am suggesting may be as stated below,

- Memory Agent
- JVM Agent
- System Agent
- Thread Agent

In general I am speaking about Mobile Agents, where in my view the Agents are mobile in nature sniffing the network traffic, and the remote host with certain hierarchical relationships, that may be has show below.

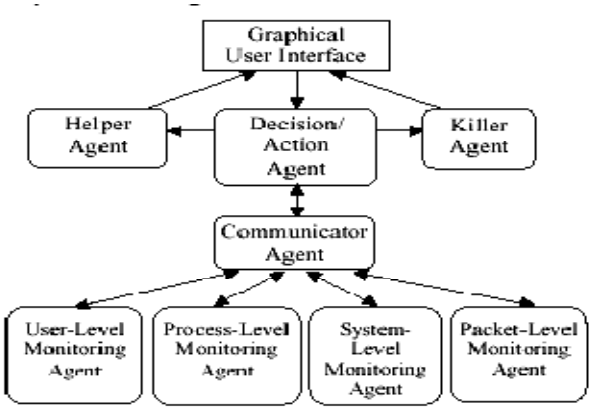

Figure 1. Agent Hierarchy

As per my view, the Advantage of this system over Traditional IDS is no need to miss even a single new attack , and since it is an automated system, the impact of Agents implementation, the presence of the Network/Host Administrator is no more required always.

VII. CONCLUSION AND FUTURE WORK

When a hacker attacks a system, the ideal response would be to stop his activity before he can cause any damage or gain access to any sensitive information. This would require recognition of the attack as it takes place. These signatures needed to be updated by the vendors on a regular basis in order to protect from new types of attacks. However, no detection system can catch all types of intrusions and each model has its strengths and weaknesses in detecting different violations in networked computer systems. Recently, researchers started investigating techniques like artificial intelligence autonomous agents and mobile agent architectures for detecting intrusion in network environment. Most existing intrusion detection systems either use signature based or anomaly based intrusion detection system, here the technique mobile agent architecture has been implemented.

Future work deals mainly with Analyzer agents. We aim to study a set of statistical and behavior models in order to develop a new one for describing a "correct" and an "attack free" system behavior. We believe that these models will be more efficient when coupled with other analyzer such as signature-based systems.

REFERENCES

[1]. William Stallings, “Cryptography and Network Security”, Principles and Practices, Third Edition.

[2]. D. E. Denning, "An intrusion-detection model". IEEE Transactions on Software Engineering, Vol. SE-13 (No. 2):222-232, Feb. 1987.

[3]. Stephen Northcutt, Judy Novak, “Network Intrusion Detection”, Third Edition, Pearson Education 2003.

[4]. Anderson, James P., “Computer Security Threat Monitoring and Surveillance,” Technical Report, James P. Anderson Co., Fort Washington, PA, April1980.

[5]. Amoroso, Edward, Intrusion Detection, Intrusion.net Books, Sparta, New Jersey, 1999. [ASAK99] M.Asaka, S.Okazawa, A.Taguchi, and S.Goto, "A Method of Tracing Intruders by Use of Mobile Agents," INET'99, June 1999.

[6]. Bauer, David S. and Koblentz, Michael E., “NIDX: An Expert System for Real-Time Network Intrusion Detection,” Proceedings of the Computer Networking Symposium, pp. 90-106, April 1988, Washington, DC.

[7]. Jeffrey M. Bradshaw, “An Introduction to Software Agents,” In Jeffrey M. Bradshaw, editor, Software Agents, chapter 1. AAAI Press/The MIT Press, 1997.

[8]. Chess, D., B. Grosof, C. Harrison, D. Levine, C. Parris, G. Tsudik, “Itinerant Agents for Mobile Computing,” IBM Research Report, RC 20010, March 1995. <URL: http://www.research.ibm.com/massdist>

[9]. Mark Crosbie and E. H. Spafford, “Active Defense of a Computer System Using Autonomous Agents,” Department of Computer Sciences, Purdue University, CSD-TR-95-008, 1995.

[10]. Denning, Dorothy E., “An Intrusion Detection Model,” IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, pp. 222-232, February 1987.

[11]. Farmer, W.M., J.D. Guttman, and V. Swarup, “Security for Mobile Agents: Authentication and State Appraisal,” Proceedings of the 4th European Symposium on Research in Computer Security (ESORICS ’96), pp. 118-130, September1996.

[12]. Farmer, W.M., J.D. Guttman, and V. Swarup, “Security for Mobile Agents: Issues and Requirements,” Proceedings: National Information Systems Security Conference, pp. 591-597, October 1996. <URL: http://csrc.nist.gov/nissc/1996/papers/NISSC96/paper033/>

[13]. Finin, T., R. Fritzson, D. McKay, and R. McEntire. “KQML as an Agent Communication Language,” Proceedings of the Third International Conference on Information and Knowledge Management (CIKM ’94), ACM Press, Nov. 1994.

[14]. Hansoty, Jatin N., “LAVA: Secure Delegation of Mobile Applets,” Master’s Thesis North Carolina State Univ., 1997. <URL: http://shang.csc.ncsu.edu:80/lava.html >

[15]. Harrison, C.G., D.M. Chess, A. Kershenbaum, “Mobile Agents: Are they a good idea?,” IBM Research Report, March 1995.

[16]. Danny Lange and Mitsuru Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley, 1998.

[17]. W. Lee, S.J. Stolfo, and K. Mok, “A Data Mining Framework for Building Intrusion Detection Models,” Proceedings of the IEEE Symposium on Security and Privacy, 1999. <URL: http://www.cs.columbia.edu/~sal/JAM/PROJECT/>

[18]. Marreale, P., “Agents on the Move,” IEEE Spectrum, April 1998, pp. 34-41.

[19]. Stefano Martino, “A Mobile Agent Approach to Intrusion Detection,” Joint Research Centre-Institute for Systems, Informatics and Safety, Italy, June 1999.

[20]. Michael M. Sebring et al., “Expert Systems in Intrusion Detection: A Case Study,” Proceedings, 11th National Computer Security Conference, pp. 74-81, October 1988.

[21]. Stephen E. Smaha, "Haystack: An Intrusion Detection System," Fourth Aerospace Computer Security Applications Conference, Orlando Florida, pp. 37-44, December 1988.

[22]. Jonathan Smith, "A Survey of Process Migration Mechanisms," Operating Systems Review, 22(3), ACM Special Interest Group on Operating Systems, pp. 28-40, July 1988.

[23]. Wu, S.F., M. S. Davis, J. N. Hansoty, J. J. Yuill, S. Farthing, J. S. Webster, X. Hu. "LAVA: Secure Delegation of Mobile Applets," Technical Report 96/42, Center for Advanced Computing and Communication, North Carolina State Univ., Raleigh, NC , October 1996.

AUTHOR PROFILES

Dr. M. ShivaKumar, Professor, Department of IT, Sudharsan Engineering College, Pudukkottai, T.N, India.he has published papers in various conferences (National & international) has good academic line of experience and published papers in various conferences (National & international).

Ms.R.Subalakshmi, Currently Working has Associate Professor in the Department of IT, Kuppam Engineering College, Kuppam, having profound knowledge in research and area of interest is Network Security, Software Engineering, operating System Etc.,

Mrs.S.Shanthakumari , Associate Professor, Dept. of CSE, Kuppam Engineering College, Kuppam, is a research scholar, carrying her research in the field of Network Security.

Mr.S.JohnJoseph, Currently working as a Assistant Professor in the Department of IT, Sudharsan Engineering college, Pudukkottai, is having profound experience in teaching and is currently a research scholar in the field of network security.

