Agent based Decentralized and Fault Tolerant Intrusion Detection System

Arjun Singh
Asst. Professor
Dept. of CSE
SPSU Udaipur

Surbhi Chauhan
Research Scholar
Dept. of CSE
Amity University, Noida

Kamal Kant
Asst. Professor
Dept. of CSE
Amity university, Noida

Reshma Dokania
Product Developer
BMC Software Pvt Ltd
Pune, India

ABSTRACT
An intrusion detection system has become a standard component of security infrastructure. Mobile agents transportable in network, gather information, evaluate and guide the alarm to network administrator. Advantage of having the mobile agent based approach in IDS that there is no centralized failure, less latency rate of data transmission and it has real time capability to generate the alarm against the intrusion. Despite of having many number of advantages of agent based IDS, it have some challenges like security of agent from attacker and high time to detection. which need to address. This paper present Mobile Agent Based decentralized and Fault Tolerant Intrusion Detection System to detect user anomalies in windows environment. This paper focus on the protection of mobile agent from malicious host.

Keywords
Machine Learning, data mining, Intrusion Detection System (IDS), Denial of Service (Dos)

1. INTRODUCTION
With the advancement and growing need of computer applications and network technologies, the security problems are becoming more complicated in information system day by day. There are many first line defense technologies exit like firewall, user authentication and cryptography, but these steps are not enough to prevent the miss use of the resources and information. To reduce the misuse of the network resources, second line defense technologies should be used viz. Intrusion Detection System.

An intrusion is defined as attempt to compromise the integrity, confidentiality, and availability of the resources. Intrusion detection is an act to detect the intrusion i.e. to find out the unwanted access, manipulating and disabling the system (DoS attack) either by authorized or unauthorized user through the network.

Any Intrusion Detection System has some inherent characteristics by focusing on functional and performance requirements [21]. An ideal type IDS should have very false alarming rate, high detection rate, corrective action against the attack, dynamic adaption and scalability. Additionally IDS can be classified as anomaly and policy detection [21]. Anomaly detection analyzes the normal activity patterns of every authorized user and any activity pattern is treated as an intrusion. On other hand Policy detection technique is a signature based model for known attack and any activity that matches with signature of known attacks is treated as an intrusion. However Defects exists in both anomaly and Policy detection Techniques. Error rate can be determined by these defects. Error rate is also known as False Alarm rate. Source of data or information derived depends on Intrusion Detection System tools. There is a family of Intrusion Detection System tools such a HIDS (Host Based Intrusion Detection System) that exploits information from a single host, NIDS (Network Based Intrusion Detection System) that exploits information from a whole segment of a network and HYIDS (A Hybrid Intrusion Detection System that exploits information from both HIDS and NIDS). Data gathered from HIDS (Host Based Intrusion Detection System), NIDS (Network Based Intrusion Detection System), HYIDS (Hybrid Based Intrusion Detection System) include:

- Audit trails are logs of event in a network environment.
- Network Packets are unit of information.

Response Mechanism can be characterized into Active Intrusion Detection System and Passive Intrusion Detection System [22].Notifying of an Intrusion to the administration comes into category of Passive Intrusion Detection System. Active Intrusion Detection System only Detect the Intrusion. Many Intrusion Detection Systems have been implemented for Centralized System. Single point of failure in centralized system introduces the distributed technology. Mobile Agent plays a vital role. Mobile agent is a software agent, having the capability to migrate from one host to another [20].

2. BACKGROUND
Jude Shavlik et al implemented Winnow-based algorithm [9], a machine-learning approach, for detecting intrusions on individual computer. Their method worked by monitoring properties of computer system. The algorithm gathered and analyzed hundreds of Windows2000 system properties each second. The properties included the load on the CPU, process performance, network performance and traffic, disk activity, monitoring of registry locations, system files, process performance , the programs that are currently running, system API's invoked, etc. The algorithm created models that represented each particular computer's range of normal behavior. Parameters that determined when the alarm should be raised, due to abnormal activity, were set on a per – computer basis, based on an analysis on the training data. The hypothesis provided some insight into which system properties played the most valuable role in creating statistical profiles of computer. The authors gathered 200 system properties and computed 1500 features out of these properties ever second. The hypothesis generated per – computer based profile with respect to

different users. The argument against this hypothesis is that, collecting huge number of properties and computing more features would itself consume many of the CPU cycles.
The Jingju [15] present the dynamic clonal selection algorithm .It can reduce False positive rate with high detection . In this process the dynamic clonal selection algorithm is improved. The Autonomous Agents for Intrusion Detection (AAFID) project [2] makes use of multiple layers of agents organized in a hierarchical structure with each layer performing a set of intrusion detection tasks. AAFID uses only static agents and is deprived of some of the benefits mobile agents can offer. The characteristics and issues of IDSs for pervasive computing environments were discussed by PradeepKannadiga et al [12], and very high-level mobile agent based architecture suitable for a pervasive computing environment is proposed. In [12], mobile agents are divided into thin and thick mobile agent, and are capable of operating in resource constrained pervasive computing devices and normal computing nodes respectively. The IDS described in [3] is made of several layers of agents. Each layer sends information to the layer above it. The bottom layer is called surveillance agents that move to every hosts and collect intrusion related data in order to send the data to the upper layers for analysis and response. The IDS discusses how IDS with multiple smaller components are better than a single monolithic IDS module. Pradeep Kannadiga and Mohammad Zulkernine [13] implemented distributed IDS called DIDS which addressed scalability problem and fault – tolerance using mobile agents. Static agent in each host detected suspicious activities and mobile agent visited every host gathering traces of attacks from static agent there. Such traces received from various static agents were aggregated and correlated to detect similar suspicious activities which resulted in decentralized data analysis making the IDS more scalable. Mobile agents executed assigned tasks even when disconnected from controller module that created it. Hence, the failure of controller module does not stop the currently ongoing IDS tasks and made the system more reliable. The argument against this hypothesis is that, the mobile agents' carrying traces of attacks could get disturbed while traveling across the network would end up with undefined results at the succeeding hosts. Moreover, an activity of a user which was considered anomaly need not be anomalous behavior of another user. Yoshinori Okazaki etal adopted DP Matching schema in UNIX environment [18]. It generated process profiles using thesystem calls to detect anomalous activities. Process profile is otherwise called as Program Profile. The profile consisted of three types of profile such as base profile, suit profile and daemon profile. The features in the approach used multiple parameters. The base profile was a collection of system calls in normal situation, where the type of a system call and its ranking of occurring frequency in that host were recorded. The suit profile which was also a record of system calls recorded when suit program was executed with exec ( ). The daemon profile recorded sequence of system calls in daemon process. A daemon process waited for a call from external processes. This profile started recording when a client connected to a daemon. In the proposed work, three system parameters corresponding to user application were monitored to build program profile of each application with respect to user. Mo Xiu-Liang, Wang Chun-dong and Wang Huai-bin [20] propose A distributed intrusion detection System. This system detects intrusion with superior performance and saves network resources. Mobile Agent Environment, Data Analysis and distributed Sensors are the Intrusion Detection Component
Debapriyay Mukhopadhyay and Satyajit Banerjee [5] argued that the anomaly detection system proposed in [9]works by profiling the normal "system behavior" instead of normal "user behavior". The authors in [5] discussed "user profiling" through Bayesian Network. The following information was gathered to analyze to build user profile in Windows environment. Helmer et al [7] designed and implemented an IDS prototype based on mobile agents with core components at the centralized server system. The agents travel between monitored systems in a network, obtain information from data cleaning agents, classify and correlate information, and report the information to the detecting server and database via mediators. In case of centralized architecture a single point failure or a busy state of the detecting component, the detection rate will drop down. Moreover, in such architectures since all monitored system communicates to a single centralized detecting server the network traffic in that segment will increase. To avoid such single point failure and reduce the network traffic at the network segment PengNing et al [14] designed and implemented a decentralized research prototype IDS named coordinated attacks response and detection system (CARDS), which aims at detecting distributed attacks that cannot be detected using data collected at any single place. CARDS adopted a signature-based approach. It decomposed global representations of distributed attacks into smaller units that correspond to the distributed events indicating the attacks, then execute and coordinate the detection tasks in the places where the corresponding events were observed. Tao Peng et al [16] discussed the challenges faced by Distributed IDS using Cumulative Sum Algorithm and a machine learning approach. They proposed a robust scheme to monitor local statistics and then decide when to share information so that both communication overheads among the distributed detecting system and detecting delay were minimized. Cansian et al [1] presented an attack signature model which works on intrusion signature handling and analyzing, from storage to manipulation. Using the model, the process of storing and analyzing information about intrusion signature would become less difficult. The argument here is that the misused detection. technique works only with set of known attacks and the testing set, and fails the detect any new attack with would not have been occurred earlier and not in the set of known attacks i.e. it cannot detect unknown attacks. Liu Jianxiago, Li Lijunan[19] proposed models which adopts decentralized distributed system. This Model detects intrusion in real time with Flexibility and Expansibility

3. MOBILE AGENT BASED DISTRIBUTED INTRUSION SYSTEM

The proposed intrusion detection system "Mobile Agent Based Distributed Intrusion Detection system" is designed by keeping in mind the notation of flexibility, scalability and reliability. The aim of this System is to detect anomalous usage of legitimate applications by authorized users in Windows environment and to implement a fault – tolerant architecture which can continue providing detection service

even in case of failure of one or more detecting servers. The main objective of the IDS is to propose protection mechanism of mobile agent from malicious host. There exists no architecture that continues providing detection services even in case of busy state or failure of one more detecting server.

Hence this Mobile Agent based Distributed Intrusion Detection System have high detection rates in real time. Fig 1 shows the internal architecture of the IDS component at each detecting server. The components at the detecting servers are as follows: Server Agent, Agent Repository, Current usage, Detection engine and Profile builder.

(i) Server Agent: The responsibility of this agent is to accept request from hosts where users have proved authentication. It then also activates the mobility of the mobile agent to each monitored host. The agent will also periodically collect current user process information from every authenticated local agent at the clients. In the case of either no response or if the user logs out, the agent will activate the profile builder to create normal activity models of that user for the next session on an assumption that the user has logged out.

(ii) Agent Repository: This is the storage area where the mobile agent resides. The server agent activates the mobility of agent from this repository.

(iii) Current Usage: The current running user application of the user at the client is periodically sent to the detecting system. When the user logs in for less than 15 times, the current usage considered as the TRAINSET Otherwise, TESTSET. In either of the case, the information is considered to be the user's history.

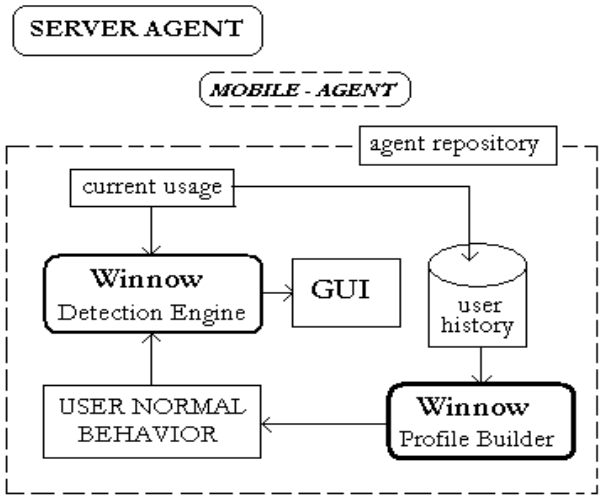

Fig.1: Internal architecture of the IDS component at each detecting server

iv). Detection Engine: This agent works only if the particular user has logged for more than 15 times. For every period of 10 seconds the information in current usage is filtered and sent to the TESTET. The anomaly detection engine uses Winnow-based algorithm to detect user anomalies. The algorithm uses the normal pattern of each application corresponding to the user from the TUNESET and compares it periodically with the corresponding current running " application" information in the TESTSET.

(v). Profile Builder: The responsibility of this agent is to build the normal usage pattern for each user. The agent uses Winnow-based algorithm to build such patterns. The recent 10-15 log information of a user in the TRAINSET is considered for this process. The pattern generated is considered as the profile of that user for the next session and are placed in the TUNESET, user normal behavior corresponding to the user.

The authentication of a user is taken care by the traditional Windows authentication mechanism. The components at the detecting servers are as follows: *Local agent, List of detection servers and Mobile agent.*
(i). Local Agent: The core responsibility of this agent is to communicate to the server agent at the detecting server, receive and activate the functionality of the *mobile agent* ,and to terminate the mobility of the agent .The additional functionality of this agent is to communicate to another detecting server if connected detecting server is busy or failed to respond.
(ii). List of Detecting Servers: Each host contains an array of IP addresses and an array of corresponding port numbers of all the detecting servers in the environment. The order of this list varies for different host and is independent of the user.

3.1 INTERNAL ARCHITECTURE OF THE IDS COMPONENT

The authentication of a user is taken care by the traditional Windows authentication mechanism. Fig 2 shows the internal architecture of the IDS components at each host being monitored.
The components at the detecting servers are as follows: Local agent, List of detection servers and Mobile agent.

(i). Local Agent: The core responsibility of this agent is to communicate to the server agent at the detecting server, receive and activate the functionality of the *mobile agent* ,and to terminate the mobility of the agent .The additional functionality of this agent is to communicate to another detecting server if connected detecting server is busy or failed to respond.

(ii). List of Detecting Servers: Each host contains an array of IP addresses and an array of corresponding port numbers of all the detecting servers in the environment. The order of this list varies for different host and is independent of the user.

(iii). Mobile Agent: Mobile agent technology is very good for detecting Intrusion in the distributed environment. Mobile agents are independent from the platform of the host .Mobile agent run on the agent platform In order to prevent the network from damage when a part of IDS fails. Mobile agents execute autonomously .The system can be reconfigured because of dynamic nature of the Mobile agent. This agent is originally activated from detecting server and later its functionality is activated by the local agent. The responsibility of this agent is to activate a tool called *pslist.exe which* collects both system and user processes. The system processes are considered as noise and hence need to be removed. The *default.txt* which contains all the system processes is then used to filter such noisy information.

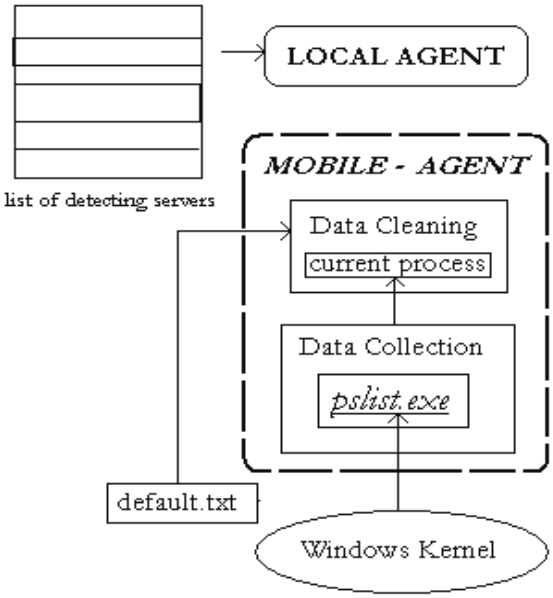

Fig. 2: Internal architecture of the IDS components at each host being monitored

4. PREVENTION ACTION FOR MOBILE AGENT DURING INFORMATION GATHERING

The Mobile Agent Based Distributed System adopts the traditional 'user name – password' security mechanism of Windows 2000 Server. Once when a user logs into any of the monitored hosts in the network, after proving authenticity, local agent at that host sends request to the detecting system. Fig 3: Message exchanged during Information gathering. The detecting system is selected based on the random list available with the local agent. The request carries the name of the user, system details and login time. On receiving the request from various hosts, the detecting system checks for the details of the users with its database. Incase if there is no entry for that user then a new table is created with the user' name. The detecting system then activates the mobility of agent towards each active host. The functionalities of the agent and hence the rest of the System is explained below. The modules in this System were divided based on the generic frame work of any IDS models:

i. Information Gathering
ii. Analysis Engine
iii. Report Manager.

Information Gathering: Information Source collects current running application details from the kernel and cleaning the collected data. The agent at the host receives an executes the mobile agent locally. The mobile agent in turn executes PsList.exe. This PsList.exe is a freely downloadable process status tool which interacts with kernel and takes control over the kernel on user request. The PsList.exe, in real time, collects all processes running at the kernel. The mobile agent then filters system related processes, and sends the name of the application processes and related details to the detecting system. The details include handle count, CPU time and elapsed time. The detecting system stores this information in database corresponding to the user. The local agent then destroys the execution of the mobile agent at that host and listens for the new mobile agent. The detecting system waits for a period of 10 seconds and again activates the mobile agents. This is repeated until there is at least one active host in the network. The Mobile agent then transmits this information to the server agent. The agent finally destroyed at the user-end.

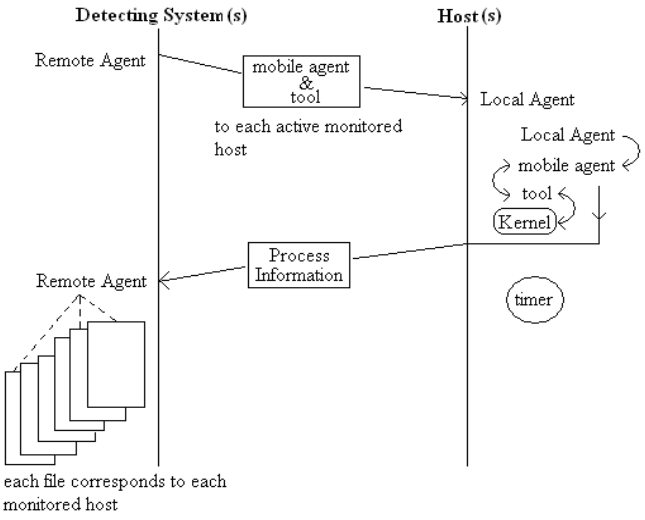

Fig 3: Message exchanged during Information gathering

Mobile agents can be protected from malicious host in information Gathering Modules.

Analysis Engine: This module is considered as the heart of the system. This module is further divided into three phases:

i. Initial Tuning,
ii. Parameter Tuning
iii. Parameter Testing

(i). Initial training: The first few login sessions of every user irrespective of the host would be considered as the training period for that user. The mobile agent periodically visits every host and reports user's activities at hosts to the detecting system. The detecting system maintains a separate temporary flat file *'current usage'* for every user at each active host. The information that would be gathered during a session, for every period of 10 seconds, will be maintained in this file. Once a user completes a session, the current usage of that that user would be analyzed to update '*userhistory'*. The handle value of each application is computed based on the maximum of handle value from the information collected during every interval of 10 seconds. The CPU and Elapsed time of each application are computed as follows. When the reported time of an application is greater than its time reported at previous interval then it means that the application is still running and hence history is not updated.If the reported time is less then it would be concluded that this application is stopped and then another similar application is opened. And if there is no report sent for that application then it would be concluded that this application is stopped. In either of the latter cases the greatest time will be updated in the history.

(ii). Parameter Tuning: Once after a user complete his training phases, information gathered will be used to build his profile. This profile of every user carries the threshold values of parameters which determine when an alarm should be generated. Instead using existing profiling algorithm (like Self Organized Map, Neural Networks, Fuzzy Login, Data Mining etc.,) which would in turn increase the load on the CPU in real time detection, in this System only the maximum and average of parameters will be computed. At the end of every session the maximum of handle count and average of the CPU and elapsed time of each application during the recent past will be computed to update the profile table. These values will be the thresholds for each application during the next session. Since thresholds are computed for each application, this approach is called as *'per application based profile'* or *'Program Profiling'*. The scope of the threshold is limited only for the next session. Latter, the thresholds are built using the history of information collected during recent five sessions.

(iii). Parameter Testing: During the testing phase, the detecting servers have with it the profile of users at active hosts. The mobile agents periodically visited the active hosts and report the activities of users at various hosts. The agent, for every 10 seconds, collects information such as the User name, Host IP address, Time, Name of the applications that were then currently running at that host, Number of simultaneous but same applications, Time since the applications were activated and Active time spent on each application. The last four parameters are instantaneously compared with the corresponding thresholds at the profile of that user. If currently received values exceed the threshold values then intrusion alert messages will be displayed at the detecting server.

Report Manager: The purpose of this module is to generate report for each intrusions launched by any user. An alert is given to the administrator along with the name of application, type of intrusion, system and user name, time and date of intrusion. This information is also stored in a file, which can later be used for computer forensic.

5. CONCLUSION

"Mobile Agent Based Intrusion Detection System" was proposed to identify anomalous usage of legitimate applications by authorized users in Windows environment. Mobile agents were used to collect application related parameters that were then currently running in the kernel at various hosts. The local agent then destroys the execution of the mobile agent at host and listens for the new mobile agent. We have proposed Preventive action for Mobile Agent during Information Gathering

6. REFERENCES

[1]. Adriano M. Cansian, Artur R. A. da Silva and Marcelo de Souza, "An Attack Signature Model to Computer Security Intrusion Detection", IEEE, pp: 1368-1373, 2002.

[2]. Balasubramaniyan. J, J. O. G.Fernandez, D. Isacoff, E., H. Spafford, and D. Zamboni, " An Architecture for Intrusion Detection using Autonomous Agents", Technical report no. TR 98-05, Purdue University, USA, 1998.

[3]. Bernardes, M.C and dos Santos Moreira, E., "Implementation of an intrusion detection system based on mobile agents", Proceedings of the International Symposium on Software Engineering for Parallel and Distributed Systems, pp. 158-164 June 2000.

[4]. Chan. P. C and Victor K. Wei, "Preemptive Distributed Intrusion Detection using Mobile Agents", Proceedings of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02), 2002.

[5]. Debapriyay Mukhopadhyay and Satyajit Banerjee, "User Profiling for Host Based Anomaly Intrusion Detection in Windows NT", Emerging Applications Information Technology, pp.193-196, 2006.

[6]. Guy Helmer, Johnny S.K . Wong, Vasant Honavar and Les Miller, "Automated discovery of concise predictive rules for intrusion detection", The Journal of Systems and Software, vol. 60, pp: 165–175, 2002.

[7]. Guy Helmer , Johnny S.K. Wong, Vasant Honavar, Les Miller and Yanxin Wang, "Lightweight agents for intrusion detection", The Journal of Systems and Software, vol. 67, pp: 109-122, 2003.

[8]. Gorodetski. V, and Kotenko, "The Multi-agent Systems for Computer Network Security Assurance: Frameworks and Case Studies", Proceedings of the 2002 IEEE International Conference on Artificial Intelligence Systems (ICAIS'02), 2002.

[9]. Jude Shavlik and Mark Shavlik, "Selection, Combination and Evaluation of Effective Software Sensor for Detecting Abnormal Computer Usage", Proceedings of the Tenth International Conference on Knowledge Discovery and Data Mining, pp. 276-285, 2004.

[10]. Kymie M. C. Tan and Roy A. Maxion, " Determining the Operational Limits of an Anomaly-Based Intrusion Detector", IEEE Journal on Selected Areas In Communications, vol. 21, 2003

[11]. Ozgur Depren, Murat Topallar, Emin Anarim, M. Kemal Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks", Expert Systems with Applications, vol. 29, pp: 713–722, 2005.

[12]. Pradeep Kannadiga, M. Zulkernine, and S. Ahamed, "Towards an Intrusion Detection System for Pervasive Computing Environments", Proceedings of the International Conference on Information Technology (ITCC), Las Vegas, Nevada, USA, April 2005.

[13]. Pradeep Kannadiga and Mohammad Zulkernine, "DIDMA: A Distributed Intrusion Detection System Using Mobile Agents", Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks, 2005.

[14]. Peng Ning, Sushil Jajodia, Xiaoyang Sean Wang, "Design and Implementation of a decentralized prototype system for detecting distributed attacks", IEEE Transactions on Computer Communications, vol. 25. pp: 1374-1391, 2002.

[15] Jing Xu,Yongzhong Li " A New Distributed Intrusion Detection Model Based on immune Mobile Agent Proceedings of IEEE ,2009

[16]. Tao Peng , Christopher Leckie and Kotagiri Ramamohanarao , "Information Sharing for Distributed Intrusion Detection Systems", Journal of Network and Computer Applications, 2005.

[17]. Mo Xiu-liang, Wang Chun-dong, Wang Huai-bin "A Distributed Intrusion Detection System Based on Mobile Agents" Proceeding in IEEE, 2009

[18]. Yoshinori Okazaki and Izuru Sato, "A New Intrusion Detection Method based on Process Profiling", Proceedings of the IEEE Symposium on Applications and the Internet, 2002.

[19] Liu Jianxiao 1, Li Lijuan 1 "Research of Distributed Intrusion Detection System Model Based on Mobile Agent" Proceedings of IEEE International Forum on Information Technology and Application,2009

[20] MO Xiu-liang, WANG Chun-dong , WANG Huai-bin"A Distributed Intrusion Detection System Based on Mobile Agents" Proceedings of IEEE,2009

[21] Nita Patil, Chhaya Das, Shreya Patankar,Kshitija Pol "Analysis of Distributed Intrusion Detection Systems using Mobile Agents" Proceedings of IEEE First International Conference on Emerging Trends in Engineering and Technology ,2008

[22] Saidat Adebukola Onashoga , Adebayo D. Akinde, Adesina Simon Sodiya " A Strategic Review of Existing Mobile Agent-Based Intrusion Detection Systems" Proceeding of Issues in Informing Science and Information Technology Volume 6, 2009