Agent Intrusion Detection Model In Attributed Environment

Jong-Geun Jeong, Chul-Won Kim, *Member, KIMICS*

Abstract—Firewall is not perfectly prevent hacker, Intrusion Detection System(IDS) is considered a next generation security solution for more trusted network and system security. We propose a agent IDS model in the different platforms that can detect intrusions in the expanded distributed host environment, since that is a drawback of existing IDS. Then we implement a prototype and verify validity. We use a pattern extraction agent so that we extract audit files needed in intrusion detection automatically even in other platforms.

Index Terms—Agent, IDS, Auditdata, Attribute

I. INTRODUCTION

Recently, the hacking technique is gradually distributed and becoming an automatic agent. It happened that well-known sites and important networks were hacked so that they had to stop their services. Hence, it is critical to protect internal information in e-commerce transactions and to provide a new system protection mechanism. Firewall can not protect external intrusions at all. a certain level, while it can not protect internal illegal actions. Active research is undergoing in internal and external intrusion detection system in real time. It is hard to have an effective detection system when it is designed for a single host system, since most internet sites and networks are the distributed multi-host systems.

It is necessary to introduce the suitable distributed environment IDS for detection these various attacks. Therefore in this paper, we propose a distributed intrusion detection system model using detection agent.

II. CLASSIFICATION IDS

A. Categorization

The objective of intrusion detection system is to detect illegal use or abuse not only from internal intruders but also external ones. There are two different ways to categorize it. One method is based on data source according to Computer Operations Audit and Security.

Technology (COAST) and the other is based on intrusion models. The former categorize audit data generated and collected from single host based systems, audit data generated and collected from multi host based systems, and network based system which uses collected packet data from the network to detect intrusion. The latter categorize an anomaly detection that detects abnormal behavior of users in a system and a misuse detection that detects intrusion from a system's known weak point or bug. Another classification is IBM Zurich Research Lab's. This classifies functional characteristics of intrusion detection systems from nonfunctional characteristics. Functional characteristics are based on intrusion detection methods, response methods to intrusions, collecting location of audit data whereas the non-functional ones are based on the frequency of monitoring.

B. Elements of technology

Intrusion detection system consists of 4 stages: data collection stage, data modification and extraction stage, intrusion analysis and detection stage, and report and response stage. The data collection stage gathers the generated information such as system usage, user information, and packets. Then uses them as audit data. The data modification and extraction stage converts into a audit data format when multi host systems are watched and extracts required information to judge an intrusion. The analysis and intrusion detection stage is a critical part and judges if this is an intrusion with analysis of audit data. The report and response stage produces a proper response automatically or reports to the system administrator.

C. Technology analysis

The implementation method of intrusion detection systems can be classified into 3 groups.

(1) real time watch and analysis technology of an intrusion
(2) collect and analysis technology of real time packets
(3) analysis technologies by analyzing audit

Real time IDS detects any illegal access or modification to non permitted files or modification to login programs. Effective method for real time IDS is monitoring and handling any illegal behavior caused from systems and devices of a network. Most behavior monitoring uses audit data provided from the operating system, whereas it has to use audit data from Web-server, Router, Firewall, TCP/IP port in order to detect illegal action in many different ways.

Real time IDS handles situations promptly to minimize the damage, since an intruder usually attempts to have

Manuscript received May 25, 2004.

J. G. Jeong is with the Computer Engineering Department, University of Honam, Gwangju, Korea,(corresponding author to provide phone:+82-62-940-5409, E_mail:jkjeong@honam.ac.kr)

C. W. Kim is with the Computer Engineering Department, University of Honam, Gwangju, Korea,(corresponding author to provide phone:+82-62-940-5403, E_mail:cwkim@honam.ac.kr)

administrator's privileges. Real time IDSs are classified into a single-host detection system and multi-host detection system. The former works in one system and it is not suitable for multi platform environment. The latter recognize the whole network and system as an agent that detects intrusions by collecting audit data from the distributed environment and analyzing it. The role of the agent is to collect and extract audit data after being installed in a multi-host system.

III. STRUCTURE OF REAL TIME IDS

Most IDS researched so far, classifies IDS into a model based and a data source based systems and uses the rule-based detection method. Because such IDSs detect intrusions by an ID process, it loads the system. Also if the process has a defect, it will degrade the system's effectiveness. In order to solve this problem we use a multi-agent to execute watch, collect, and data watch and detection. Also when IDS has no self-learning features, it cannot be flexible to handle a change in a system's environment or new types of intrusion. Real time IDS consists of Audit Record Collector, Task allocator, and Intrusion analyzer showed in figure 1. [9]

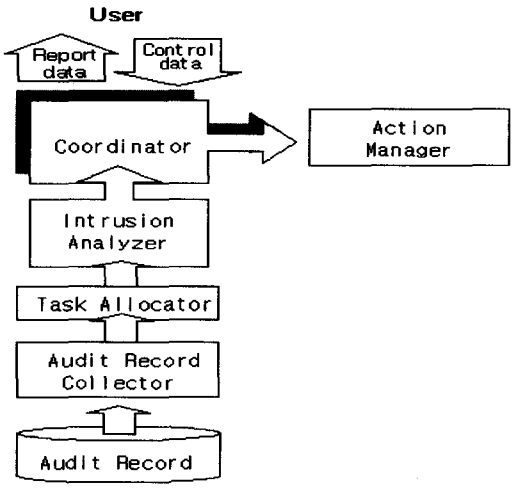


Fig. 1 Structure of real time IDS

(1) Audit Record Collector(ARC)

The ARC collects various audit data or packets from systems. It has administrator's privileges to collect the intrusion related data and must be capable of watching every status and extracting any required audit data.

(2) Task allocator

Task allocators distribute tasks adequately to judge modules in the intrusion analyzer according to the intrusion detection types. It improves performance of judging intrusions in the analyzer by processing audit data from ARC in a form required in the analyzer.

(3) Intrusion analyzer

This works as a simple analyzer, a complex analyzer, and an intelligent analyzer

IV. PROPOSED SYSTEM

Agents are the most suitable tools to watch networks or systems in distributed systems. It is ideal particularly because self-learning about intrusion information should be done automatically. Figure 2 shows a design of an automatic intrusion detection system using real-time pattern extracting agent.

We propose an automatic pattern extracting agent to not only study past intrusion types but detect and study new intrusion patterns[6]~[8].

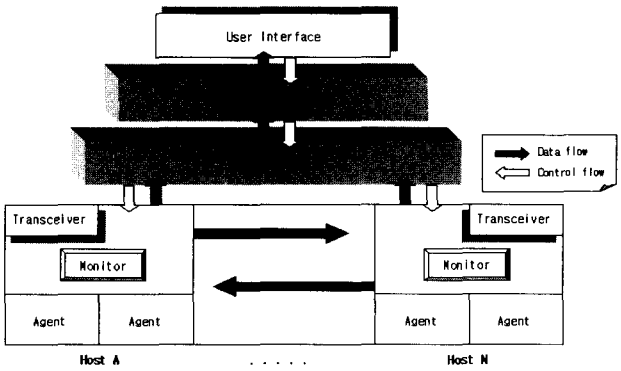


Fig. 2 Proposed System

The structure of the agent system consists of four parts: interface agent, pattern extracting agent, profile collecting agent, and process audit agent as shown in Figure 3.

The interface agent transfers a detection scenario produced at the intrusion detection server. It also sets a suitable environment for the target host. The pattern extraction agent selects only necessary audit data for a scenario in the ID server from the collected audit data at the profile collection agent.

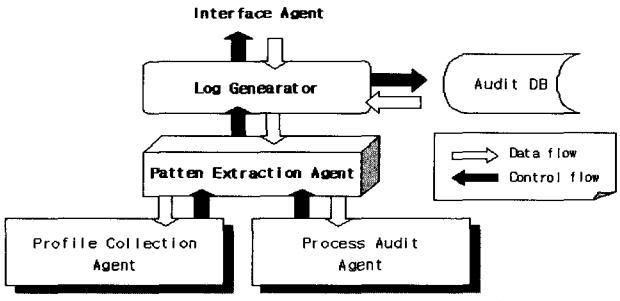


Fig. 3 Agent Structure

The collected audit data is retransferred to the ID server, when a new pattern is collected it is stored in a pattern database.

The profile collection agent and process audit agent collect events from target host: CPU usage, failed ID in login, and trial of certain port access, for the kernel.

The interface agent receives a scenario of ID system. It is ordered to collect information about current profiles and processes of a user to the pattern extraction agent. The following is an algorithm of a class of the pattern extraction agent.

```
Patt_extractor_agen
    rcv_scenario();
    request_profile();
    rcvprofile();
    request_process();
Store_pattern_DB                    (1)
    request_pattern();
    stroe_DB();
InterF_agen
    request_pattern();
    rcv_pattern();
    sendpattern();
```

When target systems are heterogenous, the problem is the format of audit files. In this paper, the standardization of extracted audit data is selected to solve the problem. Audit files from the pattern extract agent are regenerated in the standized format at the log generator.

A. Standardization of audit data

Existing IDS are not suitable to support heterogeneous systems because of their reliance characteristics. Therefore this proposed system maintains a consistent log audit resource structure which is standardized log resource from the log data analyzer. By using log filter, log resources are collected and analysed from the log data analyzer according to the scenario. The generation structure is shown in figure 4.

After collecting the required log information from an individual operating system's log analyzer, extracts only necessary log fields in a log process through the log analyzer of each operating system, then transforms it in the next process to the standard format. During this procedure, the log process generates the necessary audit data in the standard format.

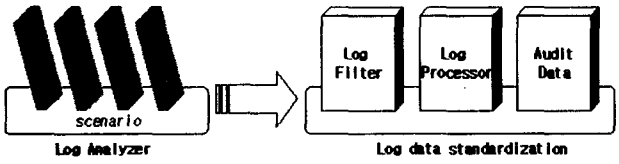


Fig. 4 Audit data standard format generation structure using log filter

In order to test the proposed standard audit resources, we use two different operating systems, SUN and AIX. We standardise the log files generated from those operating systems. Following data structure 2 shows the standard format of audit data transferred through the log processor. In order to form a standardised format items of a scenario generated through the scenario generator are necessary.

```
struct std_audit_data
unsigned long     tseq;
char              hostname[32];
char              remotehost[32];
char              ttyname[16];          (2)
char              cmd[18];
char              jobname[16];
char              dellog;
char              errlogin;
char              errorflagd;
time_t            timestamp;
long              syscall;
long              errno;
char              port;
```

B. Scenario generator

The scenario generator is the system login that anomaly detection and misuse detection need to analyse. The login record provides important security information to an administrator. Thus, a tool such as Swatch has been developed. However, it is weak at systematic analysis whereas it has an informing functionality instantly when it detects a preset pattern. The proposed scenario generator in this paper is able to accomplish not only basic log analysis but also comprehensive problem analysis. This research improves overall analysis function through a number of scenarios. The detecting patterns generated at the scenario generator are collected in the standard format of audit data like figure 8. We collected and analysed every log data produced from four different scenarios to design the realtime scenario generator. The structures of scenarios are as follows.

(1) scenario 1: reading a file that is restricted to read
(2) scenario 2: trying to write to a file that is restricted to write
(3) scenario 3: trying to access a directory that is restricted to read and write
(4) scenario4: deleting an important file by the user who has failed in login frequently

Algorithm 3 shows the structure of the scenario generator.

```
Scenario_Generator{
    Receive_scenario();
    Generate_scenario();            (3)
    Send_scenario();
    Update_scenario();}
```

It transports a necessary field to the scenario generator to create a scenario that helps detect an intrusion at the ID engine from user interface. The scenario generator creates a scenario depending on received fields and sends the fields of the scenario to agents that reside in host systems. When a new scenario field is sent, the scenario is updated.

C. decision engine structure

The roles of the intrusion decision engine are to decide an intrusion as a crucial part of IDS and to report a countermove. Report on an intrusion goes to the system administrator immediately with a warning message as shown in figure 5. It collects log data from agents in distributed host systems, transforms this to standard audit data, then decides if it is an intrusion. Decision of intrusion in this paper is done when the ratio of an applied pattern in a scenario and extracted pattern is threshold, 80%.

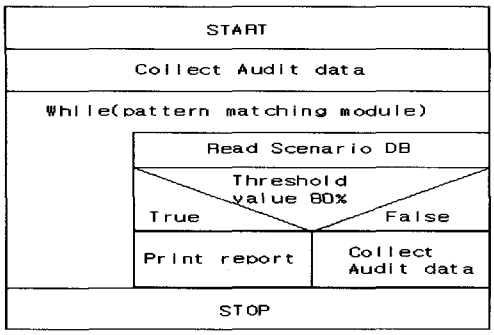


Fig. 4 Intrusion decision structure

V. PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

Commands related to the four scenarios are used as experiment resources in this paper. These resources are commands, files, or directories in UNIX and threshold is used to judge intrusions. When the threshold is high, accuracy of intrusion detection will be high, and, false positive rate is low. overall intrusion detection ratio will be low and it might regard an intrusion as normal usage that will lead to a critical result. However, if the threshold is set too low, accuracy will be low while the detection ration will be high and false positive rate is high. Therefore, the intrusion engine can choose adaptive value of threshold. Table 2 shows the list of security related files or directories used in this research.

Table 2 Security related file/directory

Item	File/directory
Read file	/var/log, /usr/etc, vferg, acct, lastlog, sulog, utmp, wtmp, pacct syslog.con
Write file	/.rhosts, /usr/local/*
Read dir	/usr/adm, /var/adm, /var/log, /usr/sbin
Write dir	/var, /etc, /usr/bin/dev, /etc/security

Figure 5 shows the difference of various threshold values in a graph. the lower threshold gives higher detecting ratio and shorter time required. The higher threshold brings higher accuracy, lower detecting ratio, and longer detecting time.

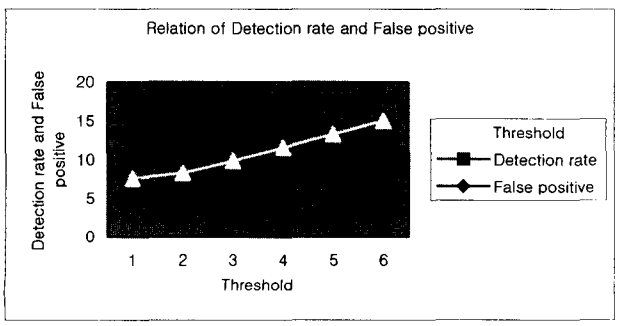


Fig. 5 Intrusion detection ratio variation and false positive following by critical value

The powerful aspects of the proposed system are extracting the required audit data for the intrusion detection in an agent stage; minimizing the work load of host systems by transforming the extracted data to the standard format; and allowing the value of threshold to be adjusted in order to respond to situations accordingly.

VI. CONCLUSION

The proposed pattern extracting agent system collects the required data to detect intrusions to distributed systems in real time and maximises the efficiency of detection in heterogeneous systems by standardisation of audit files. An intrusion scenario is generated in the scenario generator in order to collect necessary data to detect intrusions in an agent. It sends the data to the agent so that the agent can respond immediately and collect necessary data only. When multi intrusions occur in multi host systems, it loads the systems and slows the performing speed in the standardisation of audit file. Therefore we minimise loads to a systems when the system judges intrusions by using a unified format of audit data through the standardisation. However, continuing research about automated update of scenario and intrusion techniques is required to protect systems from various intrusions of hackers.

REFERENCE

[1] T. Lane and C. E. Brodley. "Temporal sequence learning and data reduction dor anomaly detection". In Processsing of the fifth ACM Conference on Computer and Communications Security, pages 150-158, 1998.

[2] W. Lee, R. Nimbalkar, K. Yee, S.Patil, P. Desai, T. Tran, and S. J. Stolfo. "A data mining and CIDF based approach for detecting novel and distributed intrusions". In Proceedings of the 3rd International Workshop on Recent Advances in intrusion Detection, October 2000.

[3] W. Lee, S. J. Stolfo. "Data mining approaches for intrusion detection". In Proceeding of the 1998 USENIX security Symposium, 1998.

[4] W.Lee, S.J.Stolfo, and K.Mok, "A Data Mining Framework for Building Intrusion Detection models", 1999 IEEE Symposium on security and Privacy, 1999.

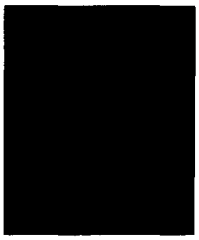
[5] Sandeep Kumar, gene Spafford. "A Pattern Matching Model for Misuse Intrusion Detection," Proceedings of the 17th National Computer Security Conference, October 1994.

[6] T. lane and C. E. Brodley. "Detecting abnormal : Machine learning in computer security", Technical Report TR-ECE 97-1, Prudue University, West Lafayette, IN, 1997.

[7] Jai Sundar B. Spafford E, "Software Agents for Intrusion Detection," Technical Report, Purdue University, Department of Computer Science, 1997.

[8] Crosbie M, Spafford E, "Defending a Computer System using Autonomous Agents," Technical Report, Purdue University, Department of Computer Science, 1996.

[9] Wenke Lee, Salvatore J.Stolfo, Philip k.Chan, "Real Time Data Mining-based Intrusion Detection". In proceedings of IEEE symposium on research in security and privacy, 2000.



Jong-Geun Jeong
Received his M.s. and Ph.D degrees in Department of Computer Science from Chosun University, Gwangju, Korea, in 1997 and 2002, respectively. In 2004, he joined the Honam University, Korea, Where he is presently a visiting professor.

Chul-Won Kim
Received his M.S. and Ph.D. Degrees in Department of Computer Engineering from KwangWoon University, Seoul, Korea, in 1984 and 1996, respectively. From 1988 to present, he is a professor, Department of Computer Engineering, Honam University.