Security Audit System for Secure Router

So-Young Doo, and Ki-Young Kim

Secure OS Research Team, ETRI, Daejeon, Korea
(Tel : +82-42-860-5354; E-mail: {sydoo,kykim}@etri.re.kr)

Abstract: An audit tracer is one of the last ways to defend an attack for network equipments. Firewall and IDS which block off an attack in advance are active way and audit tracing is passive way which analogizes a type and a situation of an attack from log after an attack. This paper explains importance of audit trace function in network equipment for security and defines events which we must leave by security audit log. We design and implement security audit system for secure router. This paper explains the reason why we separate general audit log and security audit log.

Keywords: Security Audit, Secure Router, Security Log.

1. INTRODUCTION

Audit trace function is the important function that can analyze contents and a way of an attack after an attack of a system was performed. They can take *syslog* to use in a UNIX system for example with the most general thing during audit trace feature. The *syslog* operating in a basic operating system loaded by network equipment is recording an important accident to occur during operation. In these days, network equipment and security equipment are merged with one system and are born with security network equipment. A secure switch, a secure router, a secure gateway is the security network equipment which appended an intrusion detection function to a switch, a router, a gateway. It is to have appended a function to judge whether it is an attack packet before delivering a packet to an equipment native function with the internal or an external net. Compare to the existing network equipment, security network equipment generates large quantity and various important events. It is not impossible that they use *syslog* for recording these accidents, but it is not efficient to analyze left logs. Also, the protective way that important audit log related to security is deleted unlike other audit log contents, or cannot be modified must be got. We used security audit trace feature in order to solve these limits and we were systematic and designed a safe security audit function and implemented.

Configuration of this paper is that we look into study about a log generation and analysis of a system in a Chapter 2, we put requirements about security audit log into shape in a Chapter 3, explaining the contents that we implemented to a security router with a security audit tracing system proposing in this paper in a Chapter 4, and we are going to make a conclusion in a Chapter 5.

2. RELATED WORKS

The attack detection that is not new things used a log file and an analytical method. A case of the simplest form as swatch[1,2] and TkLogger[3], they compare with the representation which fitted contents of log file to grammar and detect a specific accident have been used generally. It is designed in order the ASAX(Advanced Security audit trail Analyzer on uniX)[4] which is a tool of a higher level uses a rule-based language to be called RUSSEL and deals with audit tracing, and to be satisfied with a TCSEC[5] C2 grade. There is the SAINT(A Security Analysis Integration Tool)[6] which was proposed in order to analyze a log and a report generated with various security tools. They generate a try and a warning message about this to find any pattern to generate a problem in a system in this tool and are going to present the solution how it is possible, but there are many problems that they compare to he target and must solve yet. A way to protect log file from a hacker is studied [7] and study devoted to what they easily show complicated and various log to have been happened [8]. NIDES(Next generation Intrusion Detection Export System) includes these audit trace feature to analyze a log of the self-security system that they developed with management[9.10.11]. Audit tracing log contents of NIDES are putting emphasis by systems operation pattern analysis of users. SRS(Secure Router System) is appended intrusion detection, packet filtering, and a trusted channel function to a PC-based Linux router system [12,13,14,15,16]. This system can substitute a simple form for IDS(Intrusion Detection System) and the firewall which composed the existing security network. Compare to having been the form that IDS and firewall were separated, and this system has the strong point which let you integrate security capability each other organically several kinds of.

This paper explains on a security audit tracing system. This paper separates audit information related to security to have been happened particularly, and to manage in a security router system. This paper separates security audit log from the existing system log and manage with a collection particularly. The reason is because it is convenient for management and analysis and a log of and can protect a log to have been happened more safely.

3. REQUIREMENTS OF SECURITY AUDIT

It is important that security auditing system analyzes a left log, but it is more important what kind of contents and how to record. Therefore, it is necessary a programmer uses provided audit tracing interface and classifies and records contents of a correct accident by a step.

This paper classified the following accidents according to priority and tried to summarize sensitive information about each accident for SRS. SRS has a packet filtering function, an intrusion detection function, system self-security capability, a virtual private network function.

3.1 Packet Filtering Function

Packet filtering function will write an access control list to a router, and it compares all packets of a port, a protocol, an address of packet with access control list. If this packet was

admitted, it is disposed of a packet with transmission to receipt place if not so. It is important events in this function that setting of a rule for packet filtering, a change and an elimination event, a user without privilege manipulated a rule, and input of a packet broken by a rule and the due out.

3.2 Intrusion Detection Function

Intrusion detection function analyzes header and content of all packets which flowed into a router, and detects a specific attack. It is important events in this function that policy setting, a change and elimination, a user without privilege having attempted in order to set up policy, and if it was judged an attack by policy.

3.3 System Self-Security Capability

System self-security capability is access control processing technique, in which a role based access control[17] processing function is added to a kernel region to determine whether access to the resources of the system is permitted or denied on the basis of access control rules and databased attributes (for example, read, write, execute, inherit, etc.) It is important events in this function that they finish and booting of a router, start and quit by all functions that modularization of a router is made, identification about administrator approaching a router and authentication, if approach remotely identification and mutual authentication, router resources and access control rule setting about information, change and elimination, start of communication protocols provided in a router and end, and setting of the user identity confirmation related data that can approach a router, change and elimination.

3.4 Virtual Private Function

Virtual private network is to provide the trusted channel that can deliver all data delivered between terminals connected to a router safely. It is important events in this function that trusted channel setting change and elimination, trusted channel setting of a user without privilege and elimination, a change try, the communication protocol which violated transmitted data integrity, there is abnormal quit or re-connection of a trusted channel to have been set up.

Besides, users with authority set, delete, and modify audit level setting for each functions of security router. Saved information for a security audit log record are subject and identifier information of the object, the accident occurrence date and time, a form of an accident, result of an accident success or failure, a reason to be unsuccessful in case of accident that failed.

We can regard the audit object as the contents that must be considered besides this by security audit log, before-after of events selection, data compression of an accident, multiple audit tracings, physical save media.

In this paper we record audit log, after an accident was happened, data do not compress. We substitute log file to another file and create the new file, if file size exceeds. As for the audit tracing data, an administrator with authority which is different to system administrator can set up an audit level for security and we use role-based access control for system access control and protecting the audit log.

4. SECURITY ADUIT TRAIL SYSTEM

A security audit trail system is composed of an audit trail module and audit trail daemon. An audit trail module carries out a function to attract audit information, and audit trail daemon carries out a function to save accident information in a file. An audit trail module is developed with a Linux module form and is equipped with necessary convenience and to be able to do a removal if unnecessary. We are showing a configuration of audit trail system with other functions by a Figure 1.

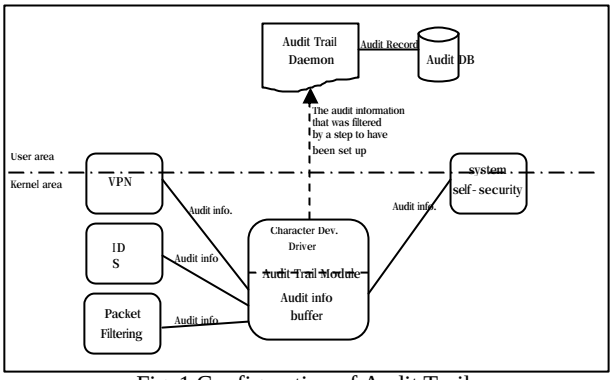

Fig. 1 Configuration of Audit Trail

Audit trail daemon is carried out in application layer and exchanges data with an audit trail module through a character device. We support GUI (Graphic User Interface) viewer to see a saved audit log file [18].

A log for audit tracing is separated by an each function and can set up a different step according to an each function. We can set up on a security router system self-security function, a packet filtering function, an intrusion detection function, a trusted channel function, and audit tracing function. A audit level is 8 which are emergency event (SEC_EMERG), need reaction immediately (SEC_CRIT), general error event (SEC_ERR), warning event (SEC_WARNING), normal but important situation (SEC_NOTICE), information message (SEC_INFO), debug message (SEC_DEBUG).

The emergency situation is the most serious state, and debug message is the lowest state. It includes higher levels if he selected a level.

The occurrence can get a log by two interfaces. One is generated inner functional call for the only kernel functions and the other is system call to record the logs for application programs.

It is characteristics in a security router it is done modularization, and to have implemented each function.

They did in order they used the way that was used if they used a thing number to specify for information transmission between modules, and to be able to communicate log information. They showed a way to record an important accident of kernel inner functions by a Figure 2 with a simple form.

register_aud(int (*func)), mod_aud_get_func( )

It is a form it will define a function in the kernel inside, and to connect *func* function to *local_audit_aud_write_sys* function in an audit trail modular initialization step after export got this function done. We use *int sys_aud_write_sys (int block_id, int aud_level, char * fmt, …)* system call for an audit tracing log to be happened in application program and

can record. If a user or administrator carried out system call in addition to a security audit in this paper, they generate all events and leaved to log file.

```
int init_module(void)
{
    ……
    register_aud(local_audit_aud_write_sys);
    ……
}
int local_audit_aud_write_sys
(int block_id, int aud_level, char *log, ...)
{
    ……
}


/* auditmodule.c */

int init_module(void)
{
    ……
    register_aud(local_audit_aud_write_sys);
    ……
}
int local_audit_aud_write_sys
(int block_id, int aud_level, char *log, ...)
{
    ……
}
/* audit_kernel.c */
int (*aud_get_func)(int block_id, int aud_level,
                                        char *log, ...);
int register_aud
(int (*func)(int block_id, int aud_level, char *log, ...))
{
   aud_get_func = func;
   if(aud_get_func == NULL){
        printk("auditmodule not yet loaded\n");
        return 1;
    }else {
        printk(" auditmodule loaded \n");
        return 0;
    }
}
int mod_aud_get_func(int block_id, int aud_level,
                                          char *fmt,...)
{
    ……
    returncode =
      (*aud_get_func)(block_id, aud_level,"%s",fmt);
   if(returncode == 0 ){
        return 0;
   }else if(returncode == 1){
        return 0;
    }else {
        printk("func called error = %d \n", returncode);
        return returncode;
    }
}
EXPORT_SYMBOL(reguster_aud);
EXPORT_SYMBOL(mod_aud_get_func);
```

Fig.2 Functions for Security Audit Trail

If a user or administrator carried out system call, a log is generated. Security audit system intercepts this system call and generates an audit information record and saves it when application program call system call. Also, we provide functional diagram controlling that someone approaches resources and information if they have no privilege by role-based access control for system self-security on all system call in a security router.

Contents of log information used in a developed audit tracing system are including the following host name, accident name, the accident occurrence is visual, accident subject information, application program information, an audit step, audit information, the results, occurrence order.

5. CONCLUSIONS

This paper relates to a security audit system for secure router which provides IDS, Firewall, VPN and self-security capability. We support the various audit level and interface for security audit for secure router. This paper explained the aim of audit log contents to satisfy a TCSEC B2 grade. This paper separates the general log to security log why is for efficiency of protection and analysis audit log.

We study continue the function of automatic analysis and the method of more efficient left log contents.

REFERENCES

[1] Stephen E. Hansen and E.Todd Atkins. “Centralized system monitoring with Swatch,” UNIX Security Symposium, Sep.1992, pp.105-117.

[2] Stephen E. Hansen and E.Todd Atkins. “Automated system monitoring and notification with swatch,” LISA, 1993, pp.145-155.

[3] Doug Hughes.TkLogger. Program available at “ftp://coast.cs.purdue.edu/pug/tools/unix/tklogger.tar.Z

[4] N. Haller, B.Charlier, A. Mounji, and I. Mathieu, “ASAX: Software architecture and rule-based language for universal audit trail analysis,” ESORICS, Nov. 1992, pp.435-450.

[5] S.Chokhani,“Trusted Products Evaluation,” Communications of the ACM, July. 1992, pp.64-76.

[6] D.Zamboni, “SAINT: A Security Analysis Integration Tool,” SANS, May1996, pp.3-15.

[7] M. Bellare and B. Yee, "Forward integrity for secure audit logs," tech. rep., Computer Science and Engineering Department, University of California at San Diego, Nov., 1997.

[8] J. Hoagland, C. Wee, K.N. Levitt , "Audit Log Analysis Using the Visual Audit Browser Toolkit". U.C. Davis Computer Science Department Technical Report CSE-95-11, 1995

[9] T. Lunt, “Automated Audit Trail Analysis and Intrusion Detection: A Survey,” National Computer Security Conf. , Oct. 1988.

[10] R.Jagannathan, T.F.Lunt, F.Gilham, A.TRamaru, C.Jalali, P.Neumann,T.D.Garvey, and J.Lowrance. “Requirements Specifications: Next Generation Intrusion-Detection Export System (NIDES),” Technical Report, SRI International, Sep.1992.

[11] T.F.Lunt, “Detecting Intruders in Computer Systems,” Conference on Auditing and Computer Technology, 1993.

[12] B.H. Jeong, J.N. Kim, S.W. Son, C.H. Park,

"Krnel-level Intrusion Detection System for Minimum Packet Loss," ICACT. Feb.2004. pp.207-212.

[13] B.H Jeong, J.N Kim, J.S Jang, "The Trends of Router Security," Vol. 1154, 2004.7., pp.1-11.

[14] M.H Han, J.N Kim, S.W Son,,"The Security Mechanism in VPN," NCS, Dec. 2003, pp.97-100.

[15] J.D Lim, J.N Kim, "Conformance Test for Open Internet Key Exchange Protocol Implementation," KICS Summer Conf., July. 2004, pp.166.

[16] S.Y Doo, J.N Kim, "Secure Management System for Network Equipment," CEIC, Dec.2004, pp.219-222.

[17] R.Sandhu, D.Ferraiolo and R.Kuhn. "The NIST Model for Role Based Access Control: Towards a Unified Standard," Proceedings, 5th ACM Workshop on Role Based Access Control, 2000.

[18] S.H Jo, S.K Eun, J.N Kim, "Policy-based Security Management for secure networking of Router," KICS Fall Conf., Nov.2004, pp.175.