데이터베이스를 이용한 실시간 감사·추적 시스템 구현

최형환* · 박태규* · 이윤희** · 조인구** · 임연호**

Implementation of a Realtime Audit Trail System using Database

Hyung-Hwan Choi*, Tae-Kyou Park*, Youn-Hee Lee**, In-Goo Jo** and Yeon-Ho Im**

요 약

기존의 리눅스 운영체제에서는 임의적 접근제어(DAC)에 의해서 자원의 접근을 통제하며, 이 때의 접근제어 정보를 로그 파일을 통한 정적인 감사 추적에 의존하고 있다. 따라서 본 논문에서는 DAC와 함께 강제적 접근통제(MAC) 기법을 구현하여 커널 수준에서 자원을 안전하고 강제적으로 통제할 수 있는 다중등급 보안(MLS) 시스템을 설계, 구현하였으며, 동적이며 실시간으로 감사 정보를 수집, 분석, 추적할 수 있도록 데이터베이스 연동을 통한 감사 추적 시스템을 설계하고 구현하였다. 데이터베이스 연동을 통한 실시간 감사 추적 시스템은 보안 관리자로 하여금 불법적 침입 및 자료의 유출에 대하여 실시간으로 대처할 수 있도록 한다. 본 논문에서는 이러한 리눅스 실시간 감사 추적 시스템을 설계하고 구현한 내용을 소개한다.

Keywords : 리눅스, 다중등급 보안, 데이터베이스, 실시간, 감사, 추적

1. 서 론

유닉스 계열 운영체제의 가장 큰 약점 중 하나는 시스템의 자원 관리 및 사용 권한이 루트(root)에게 너무 많이 집중되었다는 것이다. 따라서 시스템을 안전하게 관리 및 사용하기 위해서는 루트 권한의 분산 및 불법적인 루트 권한의 탈취를 막아야한다. 이러한 강력한 루트 권한을 침입자가 획득하기 위한 방법으로는 인증 과정에서 루트 패스워드로 시스템에 접속하는 것이다. 다른 방법으로는 적절한 인증 과정을 거치지 않고 루트 권한을 획득하는 방법이 있다. 본 논문에서는 리눅스의 커널 수준에서 MLS(Multilevel Security) 시스템을 위하여 MAC (Mandatory Access Control)를 구현하였으며, 이 MLS 시스템에 침입자가 부당한 방법으로 루트 권한 획득과 시스템 자원, 또는 중요 정보를 사용 및 변조하는 모든 과정을 실시간으로 모니터링(Monitoring)하여 보안 관리자로 하여금 적절히 대처하도록 하는 감사 추적 시스템을 구현하였다. 불법적으로 루트 권한을 획득하지 못하게 하는 방법으로는 실행시키는 시스템 호출(setuid, exec 등)의 수행을 제한[1]하는 것과 시스템 자원에 대한 일반 사용자의 접근을 제한하는 것, 그리고, 중요 문서에 보안 등급을 설정하여 접근을 제한하는 것 등이 있다[2,3]. 이러한 일련의 과정에서 나오는 정보는 커널 수준에서 채집하고, 실시간 감사 추적 시스템의 데몬(Daemon)을 통하여 데이터베이스에 즉시 저장한 후, 데이터베이스 질의어를 통하여 실시간으로 감사 추적 할 수 있다. 시스템에서 감시 대상이 되는 모든 시스템 자원과 중요 문서, 일반 사용자, 접근제어 리스트(ACL: Access Control List) 등은 보안 관리자 수준의 시스템에서 보호 및 관리가 가능하다. 즉 루트의 권한 중 일부 보안에 관련되는 부분은 보안 관리자가 루트의 권한에 영향을 받지 않고 관리할 수 있다.

본 논문의 구성은 제 2장에서는 TCSEC(Trusted Computer System Evaluation Criteria)에서의 감사 추적 요구사항에 대하여 기술하고, 제 3장에서는 리눅스 운영체제의 감사 추적과 로그 및 로그 파일에 대하여 설명하며, 제 4장에서는 MLS 시스템과 본 논문에서 제안하고 설계한 데이터베이스를 이용한 실시간 감사 추적 시스템에 대한 설명과 구현 사례를 본다.

2. 신뢰성 컴퓨터 시스템 평가 기준

미국의 경우 1983년 "Orange Book"[4]으로 불리는 신뢰성 컴퓨터 시스템 평가기준인 TCSEC[5] 초안이 제정되어, 1985년 미 국방성 표준인 DoD 5200.28-STD으로 채택되었다. 1998년 ISO/IEC/JTC1에서는 CC(Common Criteria) Version 2를 발표하였

* 한서대학교 정보보호연구실
** 티에스온넷(주)

다[6]. 이것은 접근통제, 무결성, 감사 추적, 신분확인, 키 관리 등의 정보보호 전 분야에 대한 포괄적인 보안요구 사항과, 요즈음 새로 등장하고 있는 보안 엔지니어링 개념을 도입한 보증 요구사항도 포함하고 있다. TCSEC에서는 <표 1>과 같이 평가 등급을 7단계로 분류하여 각 기관별 특성에 맞는 컴퓨터 시스템을 도입·운영하도록 권고하고 있다. 평가 등급에서 컴퓨터 시스템의 기능은 보안 정책과 책임성으로 구분된다. 감사 추적은 책임성의 기능에 속해 있으며, C2급 이상의 보안 시스템에는 필수적으로 구현이 되어 있어야 한다. 감사 추적은 운영체제, 응용프로그램, 또는 사용자의 활동에 관한 사건에 대한 일련의 기록이며, 개인의 책임성, 사건의 재구성, 침입 탐지, 그리고 문제 분석을 포함한 몇 가지 보안과 관련된 목적을 달성하는데 도움을 주는 수단을 제공한다.

<표 1> 신뢰성 컴퓨터 시스템의 평가 기준 요구 사항

요구사항	D	C1	C2	B1	B2	B3	A1	기능
임의적 접근통제	■	□	□	□	□	□	□	
객체 재사용	■	■	□	□	□	□	□	
레이블	■	■	■	□	□	□	□	보
레이블 무결성	■	■	■	□	□	□	□	
레이블 부착 정보의 전송	■	■	■	□	□	□	□	안
다중등급 보안 장치로 전송	■	■	■	□	□	□	□	
단일등급 보안 장치로 전송	■	■	■	□	□	□	□	정
출력물에 대한 레이블	■	■	■	□	□	□	□	
강제적 접근통제	■	■	■	□	□	□	□	책
주체의 보안 레이블	■	■	■	■	□	□	□	
장치 레이블	■	■	■	■	□	□	□	
신분 확인	■	□	□	□	□	□	□	책
감사 추적	■	■	□	□	□	□	□	임
안전한 경로	■	■	■	■	□	□	□	성

□ 추가적인 요구사항은 없다.
□ 새로운 혹은 개선된 요구 사항이 존재한다.
■ 요구사항이 존재하지 않는다.

컴퓨터 시스템은 여러 가지 감사 추적 기능을 가지고 있으며, 각각은 특별한 유형의 활동을 담당하고 있다. 감사는 관리, 운영, 그리고 기술적 통제를 조사 분석하는 것이다. 감사는 감사 추적으로부터 컴퓨터 시스템의 활동에 대한 값진 정보를 얻을 수 있다. 감사 추적은 컴퓨터 시스템의 감사 기능성을 향상시키며, 시스템 및 응용 프로세스의 시스템 활동과 사용자 활동에 대한 기록을 유지한다. 또한 감사 추적은 적당한 도구와 절차를 함께 사용하여 보안 위반사항이나 성능문제, 응용프로그램의 결함을 찾는데 도움을 준다.

감사 자료는 보안 관리자만이 수정과 삭제가 가능하도록 보호 영역에 저장 관리되어야 한다. 감사 기능은 감사 정보의 수집, 보호, 그리고 분석 기능을 포함하여야하며 이러한 분석을 통하여 보안 위반 사건이 발생했을 때 자원의 피해 정도를 탐지할 수 있는 자료로 활용되어야 한다. 그리고 감사 기능은 보안 위반의 잠재성을 탐지하여 사전에 경고할 수 있도록 해야한다.[7]

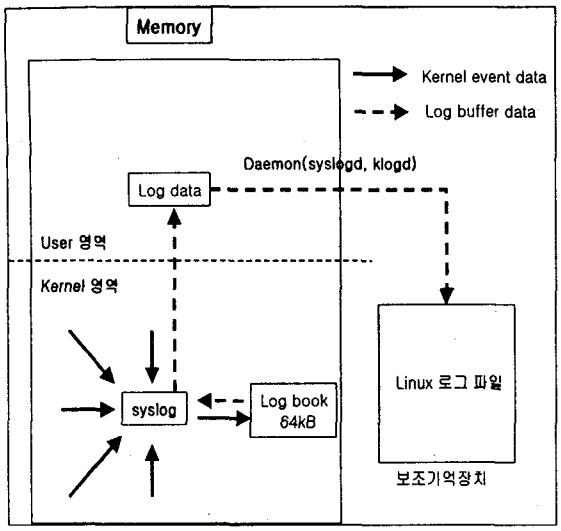
3. 리눅스 운영체제 감사 추적

시스템 보안 및 감사 추적 시스템에서 로그와 로그 파일은 빼놓을 수 없는 아주 중요한 것 중의 하나이다. 로그란, 컴퓨터 작동에 관한 기록, 상황의 변경, 스위치 선택, 입/출력 장치에 대한 사항, 콘솔에서 입/출력한 자료, 정지 상태나 원인 등에 관한 사항의 기록 등 컴퓨터 운용에 관계되는 모든 기록으로, 시스템 로그는 현재 시스템에서 어떤 일이 발생했는지 기록하는 일기장과도 같은 것이다. 이러한 모든 기록들은 로그 파일에 자동으로 기록되게 할 수 있다. 감사 추적이 잘 이루어지기 위해서는 이러한 로그 및 로그 파일의 설정이나 구성, 사건의 중요도 등이 잘 정의되어 있어야 한다. 로그나 로그 파일이 없는 감사 추적은 제 기능을 할 수 없기 때문이다.
리눅스 시스템에서 로그는 리눅스 시스템 자체에서 발생하는 여러 가지 일을 포함해 응용 프로그램의 수행 중에 발생하는 일과 프로토콜 계층에서 발생하는 일도 로그가 이뤄진다. 그러므로 시스템에서 발생하는 모든 일이 기록으로 남는다고 할 수 있다. 리눅스 로그 시스템은 사용자 영역과 커널 영역으로 구분되어지는데 모든 작업은 커널 영역에서 수행되므로 로그의 대부분이 커널에서 발생한다. 이렇게 발생된 로그는 직접 사용자 영역이나, 또는 로그 파일로 기록되는 것이 아니고, syslog()에 의해 관리된다. syslog()는 시스템의 "Log book"을 관리하고, "Log level"을 설정한다. Log book은 64KB 크기로 커널 영역의 한 메모리이다. 그리고 이것은 printk()라는 함수에 의해 데이터가 기록된다. Log level은 printk() 함수의 수행을 위한 우선 순위 등급이다. 그러므로, Log level보다 높은 우선 순위 메시지만이 콘솔로 printk()에 의해 나타나게 된다[8,9].

<표 2> printk() 함수에서의 로그 값 정의

```
#define LOG_BUF_LEN   (65536)
static char log_buf[LOG_BUF_LEN]
unsigned long log_size = 0;
static unsigned long log_start = 0;
static unsigned long logged_char = 0;
int sys_syslog(int type, char *buf, int len)
```

(그림 1)에서 보는 바와 같이 커널에서 발생하는 모든 로그 데이터는 syslog()의 제어 하에 커널 영역의 log book에 저장되고 다시 사용자 영역에 저장되며, 다시 데몬에 의해 보조 기억 장치에 로그 파일로 기록된다. 이러한 일련의 과정은 <표 3>, <표 4>, <표 5>에 나타난 각각의 명령어, 기능, 그리고 등급에 따라서 리눅스 로그 정보가 파일로 저장된다[9].



(그림 1) 리눅스 로그 시스템

<표 3> syslog level

Level	Function
LOG_CRIT	log a critical message
LOG_DEBUG	log a debug level message
LOG_EMERG	log an emergency message
LOG_ERR	log an error message
LOG_INFO	log an informational message
LOG_NOTICE	log a notice message
LOG_WARNING	log a warning condition

<표 4> syslog type

type	Function
0	close the log book NOP return = 0
1	open the log book NOP return = 0
2	read from the log book
3	read to the last 4KB of messages in the ring buffer
4	read & clear last 4KB of messages in the ring buffer
5	clear ring buffer
6	disable printk's to console
7	enable printk's to console
8	set log level of messages printed to console

<표 5> syslog facilities

Facilities	Function
LOG_AUTHPRIV	specifies that the current message is type AUTH (a security, authentication, or authorization notification).
LOG_CRON	specifies a clock daemon (cron/at)message.
LOG_DAEMON	specifies a system daemon message.
LOG_KERN	specifies a kernel message.
LOG_LPR	specifies a line printer daemon message.
LOG_MAIL	specifies a mail subsystem message.
LOG_NEWS	specifies an Usenet news message.
LOG_SYSLOG	specifies an internal syslog message.
LOG_USER	specifies a generic user message.
LOG_UUCP	specifies a UUCP message.

기록된 리눅스 로그 파일은 (그림 2)에서 보는바와 같이 로그 파일이 존재하지만, 그 중에서 리눅스 시스템에서 관리자가 항상 주의를 요하며, 본 논문에서 구현한 실시간 감사 추적 시스템에 적합한 로그 데이터베이스를 구현하는데 필요한 정보를 가지고 있는 로그 파일인 wtmp, lastlog, messages, httpd log, xferlog에 관해서 간단히 설명하면 다음과 같다.

- wtmp: 시스템이 처음 실행되어 현재까지의 모든 사용자 접속 정보 기록, 사용자의 접속 위치 기록, 접속 상태 기록, 이진파일로 기록, "last" app 사용.
- lastlog: 리눅스 시스템에 존재하는 모든 사용자의 마지막 접속 기록, 이진 파일로 기록, "lastlog" app 사용.
- messages: syslogd(app)와 klogd(ker)에 의해 시스템의 모든 정보 기록, 네트워 서비스 사건 기록, PAM 내용 기록.
- xferlog: FTP 프로토콜을 통한 자료의 송/수신 정보 기록, 텍스트 파일.
- httpd logs: 웹서버에 관련된 로그 기록, access_log, error_log 파일로 존재.

```
[root@flamy log]# l
total 380
-rw-r--r--    1 root     root        11609 Sep 19 14:57 boot.log
-rw-------    1 root     root         7751 Sep 19 22:30 cron
-rw-r--r--    1 root     root         3492 Sep 19 14:56 dmesg
-rw-r--r--    1 root     root            0 Jan  3  2000 htmlaccess.log
drwxr-xr-x    2 root     root         4096 Sep 16 04:02 httpd/
drwx------    2 root     root         4096 Dec 14  1999 iptraf/
-rw-r--r--    1 root     root       148044 Sep 19 22:32 lastlog
-rw-------    1 root     root         1397 Sep 19 14:57 maillog
-rw-------    1 root     root        60065 Sep 19 22:37 messages
-rw-r--r--    1 root     root            0 Sep  2 04:02 netconf.log
drwxrwxr-x    3 news     news         4096 Jan  3  2000 news/
-rw-------    1 root     root            0 Nov 22  1999 pacct
-rw-------    1 root     root         5634 Sep 16 04:03 procmail
drwx------    2 root     root         4096 Jan  3  2000 samba/
-rw-------    1 root     root            0 Nov 22  1999 savacct
-rw-------    1 root     root         1274 Sep 19 22:32 secure
-rw-r--r--    1 root     root          672 Aug 16 13:59 sendmail.st
-rw-------    1 root     root            0 Sep 16 04:02 spooler
drwxr-x---    2 squid    squid        4096 Jan  3  2000 squid/
-rw-------    1 root     root            0 Sep  2 04:02 sudo.log
-rw-------    1 root     root           31 Aug  4 04:02 sudo.log.1.gz
-rw-------    1 root     root           31 Jul 25 19:08 sudo.log.2.gz
-rw-------    1 root     root           31 Jan  3  2000 sudo.log.3.gz
-rw-------    1 root     root            0 Nov 22  1999 usracct
drwxr-xr-x    2 uucp     uucp         4096 Jan  3  2000 uucp/
drwxr-xr-x    2 root     root         4096 Jan 10  2000 vbox/
-rw-rw-r--    1 root     utmp       135168 Sep 19 22:32 wtmp
-rw-r--r--    1 root     root         9945 Sep 19 16:32 xdm-error.log
-rw-r--r--    1 root     nobody      61942 Sep 19 16:24 xferlog
[root@flamy log]#
```

(그림 2) 리눅스 로그 파일

4. 데이터베이스를 이용한 감사 추적 시스템

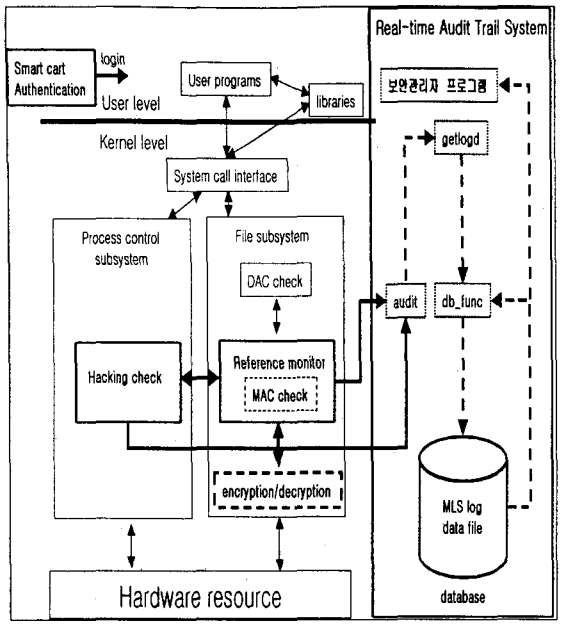
가. 다중등급보안 리눅스 시스템

침입자의 침입을 좀더 근본적으로 차단하기 위해 기존의 리눅스 시스템에 MLS[10]를 구현하고, 인증 과정에서 스마트 카드[11]를 이용한 인증을 구현하며, 그리고 기존의 응용 프로그램을 대신하는 보안 응용 프로그램[12]를 추가할 수 있도록 API를 제공하여 침입자가 궁극적으로 원하는 것을 얻지 못하게 하는 강화된 보안 시스템을 구현한 것이 MLS 리눅스 시스템이다. MLS를 구현하는데 있어 필수적으로 필요한 기술적 사항은 MAC이다[13]. 본 논문에서 MAC에 사용된 보안 모델은 주체와 객체의 보안등급과 비밀 등급에 따라 엄격하게 접근을 통제할 수 있는 BLP(Bell & LaPadula)[14] 모델이다. 이 모델을 본 시스템에 적합하게 수정된 BLP모델을 구성하였다. 또한 시스템 관리자와 보안 관리자의 역할을 분리하여 원천적인 보안 관리에 만전을 기하였으며, 응용프로그램계층에서의 암호화 수준을 커널 계층에서의 암호화 방식[12]으로 구현하여 비밀 수준의 모든 파일과 문서가 자동적으로 사용자에 투명하게 암호화되어 저장된다. 사용자, 파일, 그리고 모든 자원에는 MLS 시스템에 적용되는 보안 등급[6]이 설정되어 있어 MAC가 가능하도록 구현되었다.

나. DB를 이용한 실시간 감사 추적 시스템

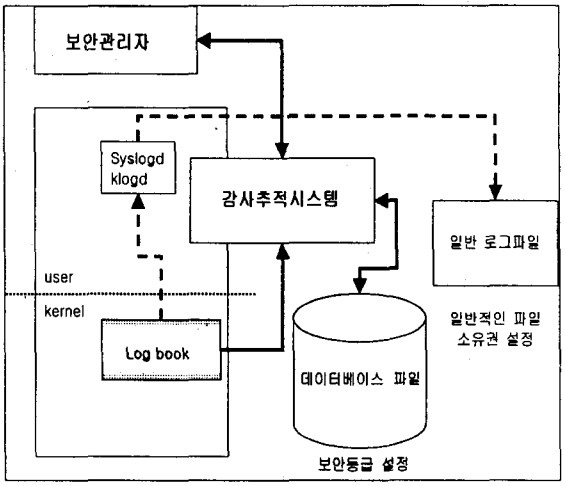
1) 설계

일반적으로 로그를 파일로 저장하는 것은 너무 많은 기록매체의 낭비를 요하므로, 이 로그 정보를 데이터베이스화하여 체계적으로 관리 기록하면 자원의 낭비를 줄이고, 관리의 효율적인 측면과 감사 추적의 용이함 등의 장점을 기대할 수 있다. 기존의 로그와 로그 파일은 일관성이 없이 서로가 난립된 가운데, 어느 것 하나 완전한 정보를 가지고 있지 않아 어떠한 정보를 분석하려고 한다면 여러 개의 로그 파일을 조사 분석해야한다는 어려움이 있다. 그리고 로그 정보를 실시간으로 분석하기가 어렵다는 단점이 있다. 이에 이러한 모든 정보를 어떤 사건이 발생하는 즉시, 하나의 데이터베이스에 실시간으로 저장하여 일관성 있게 관리 유지한다면 비용뿐 아니라 보안 문제의 개선과 감사 추적시스템의 유지/관리가 더욱 편리할 것이다.



(그림 3) MLS 리눅스 감사 시스템 구조

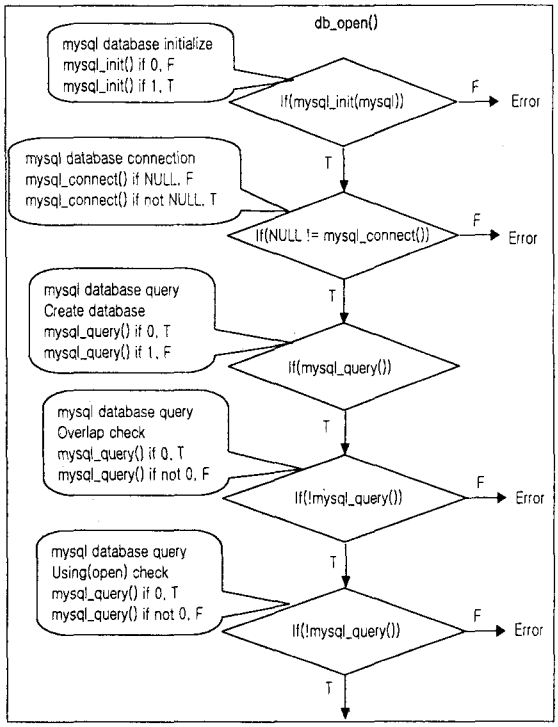
MLS 시스템에서 무엇보다도 중요한 것은 MAC이다. MAC는 자체로서도 만족할만하다고 할 수 있으나, 좀더 보안을 생각한다면 MLS 시스템에 적합한 실시간 감사 추적 시스템을 추가하야 좀더 확실한 보안 시스템으로 만드는 것이다. 그래서 MLS 리눅스 시스템에 데이터베이스를 이용하여 실시간으로 감사·추적을 할 수 있도록 시스템을 설계하였다.



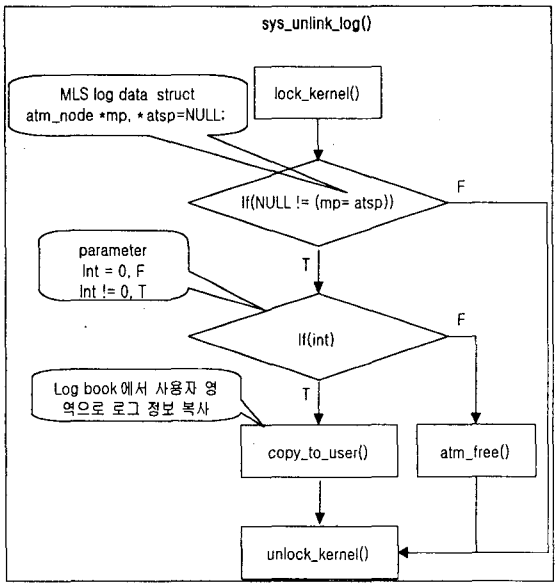
(그림 4) MLS 시스템에서의 로그 정보 흐름

따라서, 감사 추적 시스템은 침입의 탐지 및 자료의 유출에 대해서도 실시간으로 확인할 수 있으며, 이러한 일련의 과정이 데이터베이스에 실시간으로 기록되고, 보안관리자에게 실시간으로 전달될 수 있게 된다.

(그림 4)에서는 이러한 로그 정보의 흐름을 간략하게 표현하였다. 이러한 감사 추적 시스템의 로그 정보는 MLS 시스템에 적합하게 정의된 정보들을 저장한다. 보안 관리자는 저장된 정보를 언제든지 실시간으로 질의하여 원하는 정보만을 가져올 수 있다. 본 논문에서 구현한 감사 추적 시스템의 감사 정보 테이블은 29개의 항목으로 구성되었다. 감사 추적 시스템의 구성은 크게 세 가지로 구성 되어있다. 첫째, 커널에 다중등급보안 로그 정보를 데이터베이스에 연결하여 사용할 수 있게 mysql[15,16] 데이터베이스 함수를 사용하여 db_open() 함수, db_close() 함수, 그리고 db_query() 함수를 커널 함수로 설계하였다.



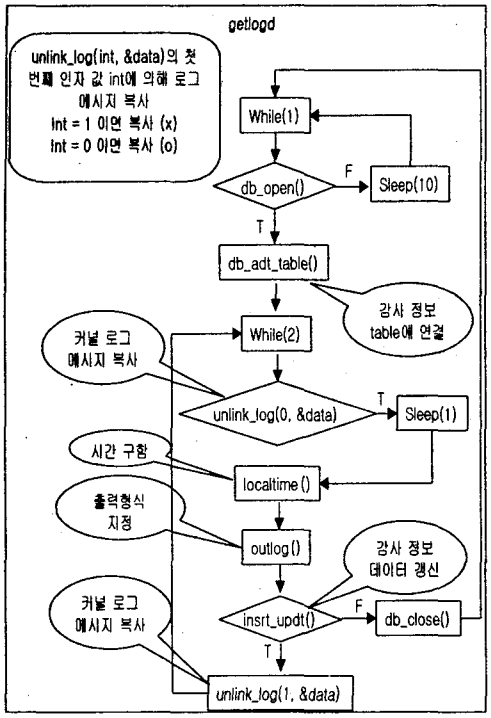
(그림 5) 커널 함수



(그림 6) Audit 함수

둘째, 시스템 호출을 이용하여 커널 영역의 메모리인 log book에서 로그 정보를 가져오는 audit함수를 설계하였다.

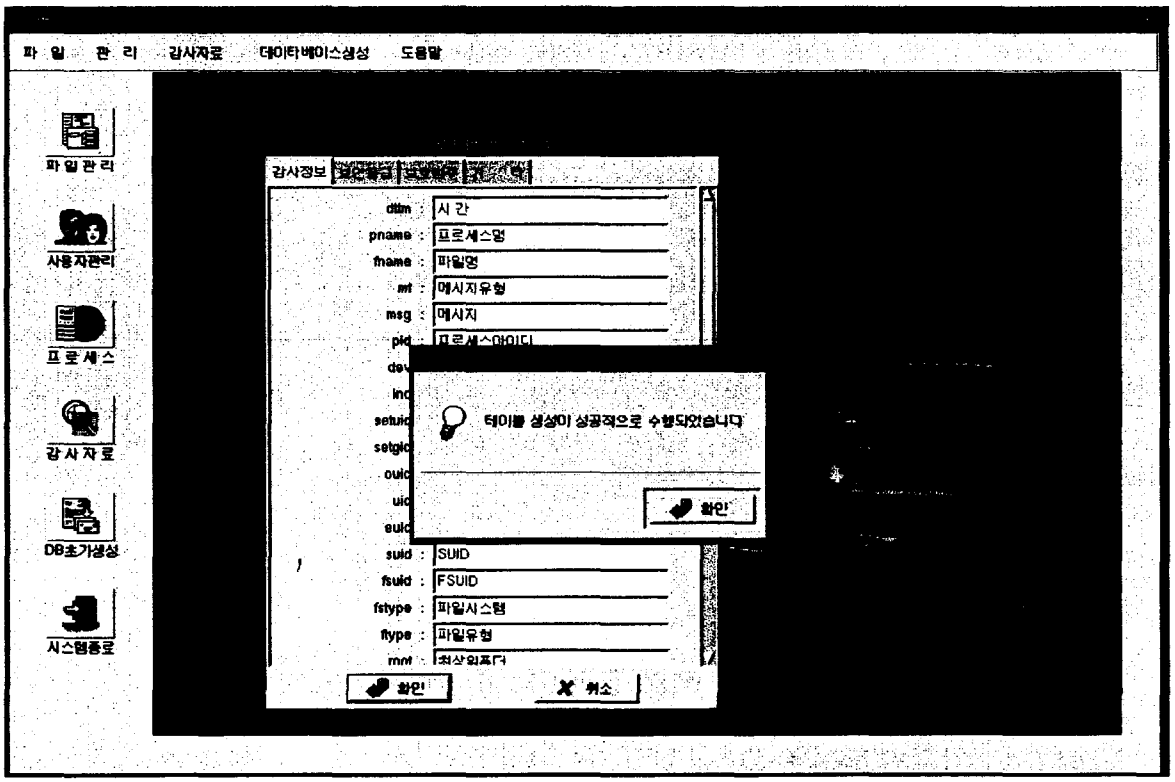
셋째, 커널과 시스템 호출을 이용하여 사용자 영역에서 기존의 klogd, syslogd 데몬의 역할을 대신 수행하는 실시간 감사 추적 데몬인 getlogd를 설계하였다.



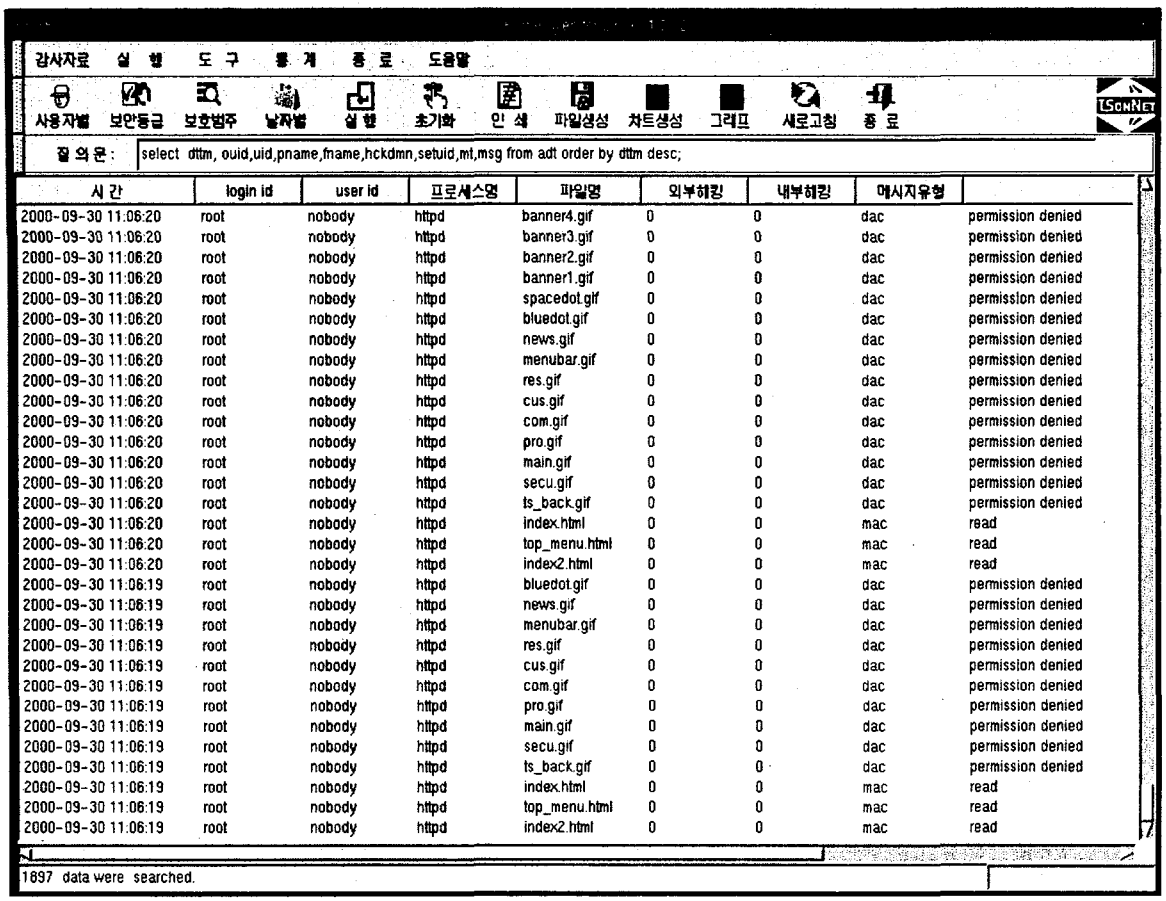
(그림 7) 실시간 감사 추적 데몬

2) 구현

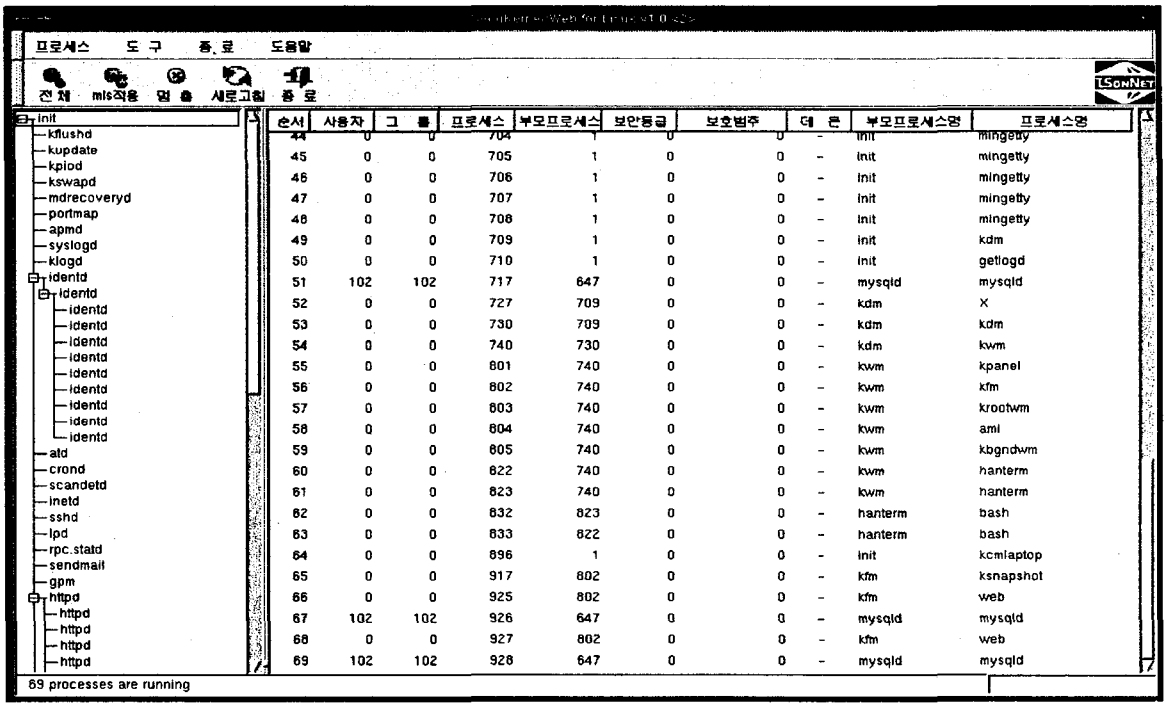
MLS 리눅스 시스템에 접속하기 위해서는 사용자 인증이 안전하게 수행되어야 한다. 사용자 인증 화면 초기에 사용자 ID(Identifier)와 PIN(Personal Identifier Number)을 입력하여 사용자 인증을 실시한다. 이때 보안등급에 관련된 정보는 스마트 카드를 이용하여 입력한다. 이 감사 추적 시스템 사용자는 시스템 관리자가 아닌 보안 관리자의 사용자 인증을 필요로 한다. 인증이 성공적으로 수행되면, 보안 등급에 따라 권한이 설정되고, 감사 추적 시스템 데이터베이스를 생성한다. 각각의 항목을 선택하여 데이터베이스에 보안 정보 테이블을 생성할 수 있다. (그림 8)은 이러한 작업을 수행한 예로서 보안 관리자만이 테이블을 생성하고, 초기화 및 갱신 등의 작업을 할 수 있다. 초기에 생성된 데이터베이스 테이블은 시스템이 작동 중에는 언제든지 실시간으로 정보의 갱신이 가능하며, 항상 모니터링이 가능하다. 이 감사 정보 데이터가 생성이 되면 감사 추적 시스템에서 감사 정보를 이용하여 분석과 추적을 할 수 있는데, 이러한 모든 정보는 감사 정보에서 확인할 수 있다. 감사 정보는 감사 추적 시스템 화면에서 감사 정보를 선택하여 모든 감사 정보를 볼 수 있는데 이 정보는 (그림 9)에 나타난 것과 같다. 감사 정보에는 시간, 프로세스, 사용자, 접근한 파일, 해킹 상태, MAC 실시 여부, 그리고 메시지 유형 등의 정보 테이블이 있다. 간단히 설명하면 메시지 유형에는 DAC와 MAC가 있다. 그리고 read나 permission denied 등은 성공 여부를 나타내는 메시지이다. 이러한 정보를 이용하여 보안 관리자는 항상 시스템의 보안을 확인할 수 있다.



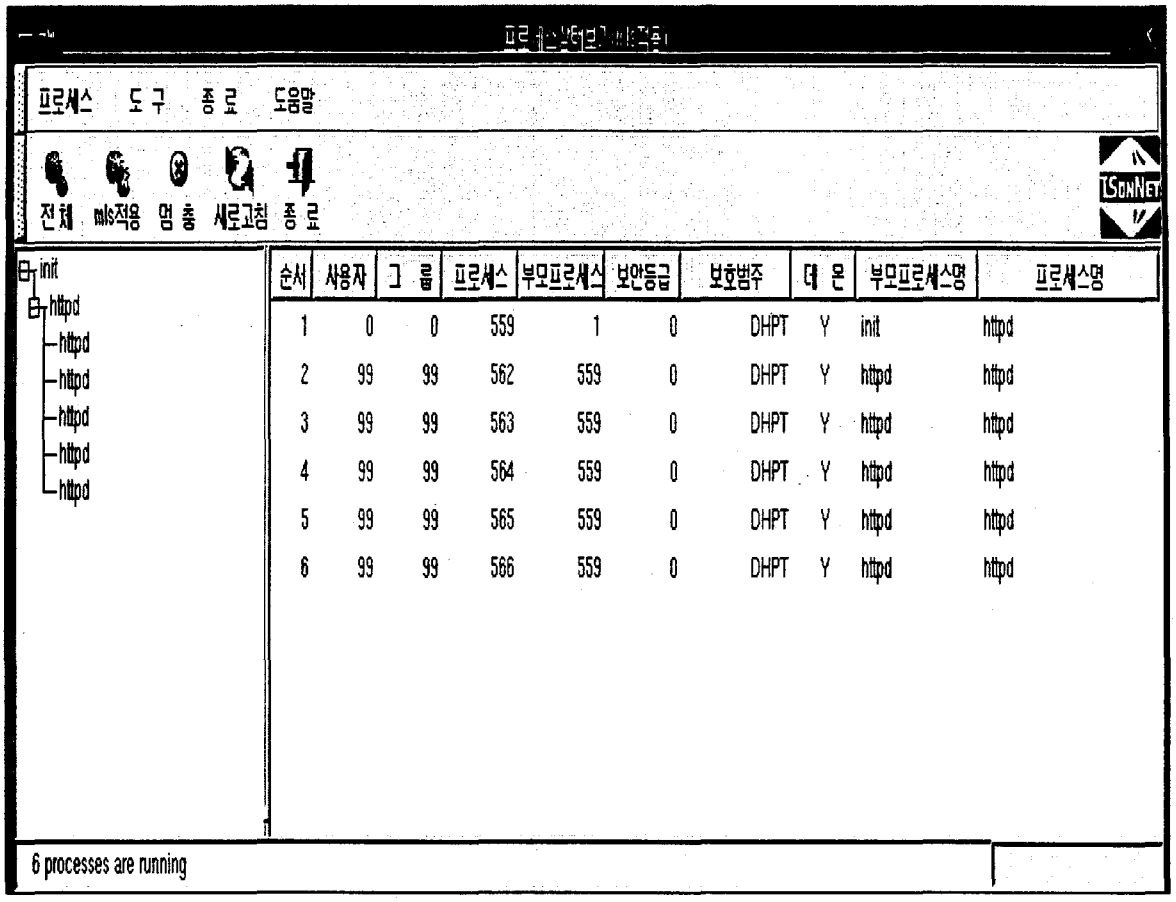
(그림 8) 감사 정보 테이블 생성



(그림 9) 감사 정보 출력 화면



(그림 10) 전체 프로세스에 대한 감사 정보



(그림 11) 실행중인 프로세스 감사 정보

감사 정보는 각각의 사용자, 프로세스, 보안 등급 별로 감사 정보를 이용할 수 있다. 또한 특정 정보를 원하는 경우, 보안 관리자는 감사 정보 테이블에 저장된 특정정보만을 가져올 수 있다. 즉, 현재 사용중인 프로세스의 정보를 분석하고자 한다면, 감사 추적 시스템을 통하여 데이터베이스를 검색하고 조건에 만족하는 항목을 수집하여 실시간으로 분석할 수 있다.

(그림 10)은 현재 시스템에 존재하는 모든 프로세스를 나타내고 있다. 데이터베이스의 프로세스 정보를 모두 출력한 것이다. 모든 프로세스에서 특정 프로세스인 httpd에 관한 프로세스의 정보를 원할 경우, 데이터베이스에 질의를 하여 httpd에 관한 현재 실행중인 프로세스만의 정보만을 출력시킬 수 있다. (그림 11)은 이러한 예를 보인 것인데, 현재 실행 상태, 보안 등급, 부모프로세스, 프로세스 명, 그룹, 사용자 등 기존에 정의한 감사 정보가 모두 나타나고 있다. 원하는 정보는 추가로 언제든지 변경이 가능하다.

5. 결 론

본 논문에서는 DAC와 MAC의 정책을 통하여 MLS 리눅스 시스템을 구현하여 이 리눅스 시스템에 실시간으로 감사 추적이 가능하도록 데이터베이스화 하여 그 기능을 추가하였다. 로그의 실시간 감사는 기존의 리눅스 시스템에 보안을 위하여 그 기능을 추가 구현하였다. 실시간 감사는 실시간으로 침입을 감지하고 자료의 유출 및 변경을 감지할 수 있어 보안기능을 강화할 수 있다는 장점이 있다. 이러한 감사 추적 시스템을 사용자 계층이 아닌 커널 계층에서 구현하여 보안 기능을 원천적으로 강화하였다. 따라서, 시스템 내부 및 네트워크를 통한 외부의 침입과 자료의 변경 등에 실시간으로 대처할 수 있을 것이며, 기존의 리눅스에서 제공하는 로그 정보보다 좀 더 구체적이고 다양한 정보를 정의하여 관리할 수 있다. 또한 감사 추적에 필요한 로그 정보는 언제든지 보안 관리자에 의해 동적으로 재구성할 수 있으며, 사용자가 용이하게 사용이 가능하도록 인터페이스는 X-Window를 활용하여 구현하였다.

참 고 문 헌

[1] 문한구, 루트 권한 감시를 통한 유닉스의 보안 강화, 서울대학교, 1996.

[2] 한국정보보호센터, 정보통신 기반구조 보호를 위한 보안 서버 모델 연구, 연구수행기관: 한서대학교, 1999 .

[3] 최형환, 이회선, 최용호, 홍승표, "L4 마이크로 커널 기반 L4Linux용 보안 서버 설계 및 구현," 통신정보보호학회 학술발표회 논문집, 1999.11. pp 97 ~ 107

[4] DoD, Orange Book, DoD, 1985.

[5] DoD, Trusted Computer System Evaluation Criteria, DoD 5200.28.STD,1985.

[6] ISO/IEC JTC1/SC 27, Information Technology-Security Techniques-Security Information Objects, N2315, 1999.

[7] 조규민, 황영석, 이경구, "정보보호시스템 평가 기준 보안 기능 요구사항 분석," 통신정보보호학회 학회지, 제10권 제3호, 2000. 9.

[8] Anonymous, Maximum Linux Security, SAMS, 2000.

[9] Michael Beck 외, Linux Kernel Internals second edition, Addison-Wesley, 1998

[10] 홍기융, 이철원, 이형수, 박태규, "MLS OS를 위한 엑세스 제어 메커니즘 연구," 제2회 정보보호와 암호에 관한 워크숍, 1990.9.

[11] 권현조, 원동호, "스마트 카드 데이터 보호를 위한 접근 통제 모델 분석," 통신정보보호학회 학회지, 제10권 제3호, 2000. 9.

[12] 김현정, 다중등급보안 리눅스 시스템 개발과 보안 인터페이스, 통신정보보호학회지,2000.11.

[13] 이철원, 홍기융, 김학범, 오경희, 심주걸, "다중등급보안정책을 지원하는 침입차단 시스템의 설계," 제7회 통신 정보 합동 학술 대회 (JCCI '97) 논문집, pp. 59~ 63, 1997. 4.

[14] Bell. D. & Lapadula, "Secure Computer System: Mathematical Foundations and Model," MITRE Report MTR 2547, v2 Nov 1973.

[15] 문태준, 권용찬, 윤형렬, MySQL 매뉴얼, http://kldp.org/

[16] 이상용, MySQL 튜토리얼, http://kldp.org/