리눅스 운영체제를 위한 보안 시스템 설계

박 진 석, 김 순 곤

The Design for Security System of Linux Operating System

JinSeok Park, SoonGohn Kim

요 약

본 논문은 기존 SELinux를 포함한 리눅스 보안 운영체제와 보안 모듈에 관한 선행 연구 분석을 통해 사용자 식별 인증, 주체·객체의 보안 권한 상속, 참조 모니터와 MAC 등급 처리, DB를 이용한 실시간 감사 추적이 적용된 리눅스 접근제어 보안 모듈을 다음과 같이 제안하였다. 첫째, 사용자 인증 시 접속 허용 IP를 판별하고 보안등급과 범주를 입력하게 하여 불법 침입자가 superuser(root) 권한을 획득하였다 하더라도 시스템 파괴가 불가능 하도록 설계 하였다. 둘째, 주체 및 객체의 보안 속성 상속을 통하여 주체가 보안이 설정된 객체에 접근할 때, 설정되어 있는 주체의 보안 정보와 객체에 설정되어 있는 보안 정보를 비교함으로써 접근 제어가 이루어지도록 하였다. 셋째, 커널상에서 현재 발생하는 모든 사건에 대해 참조모니터의 감사를 진행하며, 악의적인 목적으로 여러 객체에 접근하는 것을 사전에 차단하도록 하였다. 넷째, DB를 이용한 실시간 보안 감사 추적 시스템을 통해 각 행위와 관련된 보안 감사 자료는 보안 관리자에게 실시간으로 제공되기 때문에 긴급 상황이나 문제 발생 시 즉각 대처할 수 있도록 설계하였다.

Abstract

This paper reviews the current studies about the current secure OS, security module and SELinux, and suggests Linux access control module that uses the user discriminating authentication, security authority inheritance of subjects and objects, reference monitor and MAC class process and real-time audit trailing using DB. First, during the user authentication process, it distinguishes the access permission IP and separates the superuser(root)'s authority from that of the security manager by making the users input the security level and the protection category. Second, when the subjects have access to the objects through security authority inheritance of subjects and objects, the suggested system carries out the access control by comparing the security information of the subjects with that of the objects. Third, this system implements a Reference Monitor audit on every current events happening in the kernel. As it decides the access permission after checking the current MAC security attributes, it can block any malicious intrusion in advance. Fourth, through the real-time audit trailing system, it detects all activities in the operating system, records them in the database and offers the security manager with the related security audit data in real-time.

Keywords : Secure OS, Secure System, Linux Operating System

* 교신저자 중부대학교 정보과학과 박사과정(hwasok@joongbu.ac.kr)
** 중부대학교 컴퓨터학과 교수(sgkim@joongbu.ac.kr)
접수일자 : 2010년 9월 15일, 수정일자 : 2010년 10월 15일, 심사완료일자 : 2010년 12월 10일

Ⅰ. 서 론

기존의 리눅스에서 제공하는 접근제어와 로그를 생성하는 감사 기능 등 보안기능은 미국의 컴퓨터 신뢰성 평가기준인 TCSEC 기준으로 C2급에 해당한다. 다소의 강력한 보안 시스템이 연구용 시제품 혹은 매우 전문적인 제품으로 구현되어왔지만 시스템 관리자 및 일반 사용자에겐 아직도 어려운 도전으로 남고 있다. 즉, 기존의 시스템에서 어떻게 새로운 보안 기능들을 활용할 것인지를 쉽게 결정하지 못하고 있다. 특히 이러한 문제점을 해결하기위해서 2001년 미국 NSA의 Peter Loscocco가 리눅스 커널 정상회의(summit)에서 2000년 NSA를 중심으로 NAI Lab SCC(Secure Computing Co) MITRE가 공동으로 수행한 SELinux 연구 프로젝트의 결과를 리눅스 커널에 반영할 것을 제안 발표함으로써 LSM(리눅스 보안 모듈) 프로젝트가 태동하게 된 것이다. 본 논문에서는 SELinux를 포함한 리눅스 보안 운영체제와 보안 모듈에 관한 분석을 통해 TCSEC B1 등급 기준을 만족하는 리눅스 보안 운영체제의 차세대 접근제어 보안 모듈을 제안하여 리눅스 보안 운영체제가 나아갈 방향을 제시한다.

Ⅱ. 관련연구

1. 보안 운영체제의 정의

보안 운영체제란 컴퓨터 운영체제 상에 내재된 보안상의 결함으로 인하여 발생 가능한 각종 취약성으로 부터 시스템을 보호하기 위하여 운영체제 내에 보안기능을 위한 보안커널(Security Kernel)을 이식한 운영체제를 일컫는다. 따라서 보안 커널이 이식된 운영체제는 참조모니터의 구현을 통해 컴퓨터 사용자에 대한 식별 및 인증, 강제적 접근통제, 임의적 접근통제, 재사용 방지, 침입탐지 등의 보안 기능을 통하여 커널 레벨로의 접근제어 및 알려지지 않은 공격에 대한 방어, 탐지 및 대응 등의 기능을 수행할 수 있다. 더불어 엄격하게 분리된 영역에 로그, 감사기록을 두어 보호함으로써 불법 침입자가 침입의 흔적을 제거하는 행위를 방지하는 기능을 포함하고 있어야 한다.

2. 보안 운영체제의 요구 사항

1) 보안 운영체제의 설계시 요구 사항

보안 운영체제를 구현하기 위해서는 설계시에 운영체제가 유지하는 정보에 대한 기밀성(confidentiality), 무결성(integrity), 가용성(availability)을 보장하도록 하여야 한다. 따라서 특권의 최소화(Least of Privilege), 메커니즘의 경제성, 완전한 중재, 허가 기반, 특권의 분리, 사용의 용이성 같은 요소를 고려해야 한다.

2) 보안 운영체제의 기능적 요구사항

첫째, TCSEC B1급 운영체제에서 요구되는 주요 기능, 즉 사용자 식별 및 인증 기능, 주체와 객체에 대한 비밀 등급 및 보호 범주의 부여, 임의적 접근제어 및 강제적 접근제어 기능의 제공, 감사 및 추적 기능이 필요하다.

둘째, 리눅스 보안 운영체제에서 현실적으로 필수 불가결한 기능적 요구사항은 각종 Application software나 System program의 취약성을 이용한 해킹 공격으로부터 방어하는 기능이 필요하다.

Ⅲ. LSM과 SELinux의 접근제어 방법

1. LSM의 구조 및 인터페이스

1) LSM 기본구조

LSM 기본구조는 다음의 3가지 목표를 만족하도록 했다. 첫째, 다른 보안 모델을 사용하는 경우에는 단지 다른 커널 모듈을 적재하면 되도록 일반적일 것, 둘째, 개념적으로 단순하고 최소한으로 영향을 미치며 효율적일 것, 셋째, 선택 가능한 보안 모듈로서 기존의 POSIX. le capability를 지원할 수 있을 것 등이다. 여러 가지 접근제어가 가능하면서 이러한 3가지 목표를 만족시키기 위해서 LSM은 커널의 내부 객체(task, inode, open file 등)에 대한 접근이 가능 하도록 하는 방법을 채택하게 되었다. 사용자 프로세스가 시스템 호출을 수행할 때 우선 자원을 찾아 할당하기 위해서 리눅스 커널의 기존 로직을 거치며, 오류 체킹을 수행한 후 기존의 UNIX DAC에 넘기게 된다. 커널이 내부 객체에 접근을 시도하기 직전에 LSM hook이 “이 접근이 당신에게는 올바른가?”라는 질문을 부여하기 위해서 당신이 원하는 모듈로 out-call을 만들게 된다. 이 모듈이 이러한 정책적인 질문을 처리하고 "yes" 또는 "no" 중 하나를 되 돌려준다. LSM 모듈의 조합은 다른 문제이다. 한편, 어떤 보안 정책들은 명시적으로 상호 충돌이 발생될 수 있기 때문에 여러 가지 보안 정책이 하나의 총체적 경우로 구성될 수 없는 경우가 있다. 다른 한편, 여러 가지의 다양한 보안 정책을 조합하여 구성하는 것이 분명 바람직하다. 여기서 LSM은 효과적으로 모듈 작성자에게 모듈을 "stack"할 수 있도록 방법을 제공한다. 즉, 첫 번째 적재된 모듈은 LSM 인터페이스를 다음에 적재될 모듈에게 전달할 수 있게 된다. 첫 번째 모듈은 두 번째 모듈에서 특정한 일 을 수행 후 되돌아와 접근 결정을 종합적으로 조합 할 책임을 지게 된다.

2) LSM 인터페이스

LSM 인터페이스는 150여개의 큰 함수 테이블이다. 이 함수들은 기본적으로 전통적인 super-user DAC 정책을 구현하는 호출로 알려져 있다. 제공하고 있는 인터페이스는 정책 등록(Policy Registration) 방법을 제공하며, 여러 가지의 Hook을 위한 인터페이스를 제공하고 있다. 정책 등록 방법에 있어 LSM 인터페이스는 callback method (security_ops) 구조로 구현된다. 보안 모듈은 집행하는 보안 정책에 따라 callback을 구현 할 책임이 있다. 부팅할 때 security_ops 구조가 디폴트 callback으로 초기화되며 이 callback은 전통적인 superuser 시맨틱을 구현하게 된다. 보안 모듈은 동적으로 적재가능 모듈(dynamically loadable module) 또는 정적으로 커널에 링크되는 모듈로 만들어 질 수 있다. 이 모듈은 동적 적재 모듈 경우에는 모듈 적재 시에 초기화 되고, 정적 링크 모듈의 경우에는 do_initcalls()을 수행하는 동안 초기화된다. 초기화하는 동안 보안 모듈은 register - security()를 호출함으로써 LSM 기본구조에 이 callback을 등록해야만 한다. security_ops 구조를 디폴트 superuser 정책에 되돌려 주기 위해서 이 모듈을 제거(unload)할 때는 unregister_security()를 호출해야 한다. LSM 기본구조는 한 순간에 오직 하나의 주가 되는 보안 정책만을 알고 있다. 하나의 보안 정책이 LSM 기본구조에 등록되게 되면 새로운 보안 정책 의 등록은 불가하다. 어떤 경우에는 여러 가지 보안 정책을 조

합하는 것도 가능하다. 그러나 이 경우 LSM은 새로운 모듈을 다른 모듈과 함께 "stack"할 수 있도록 하며, 기본 구조는 오직 하나의 security-ops 만을 알고 있을 뿐이다. 추가적으로 보안 정책을 등록하기 위해서는 후속 모듈은 mod_reg- security()를 이용해서 주가되는 모듈에 등록해야 한다. 이러한 방법이 LSM 기본구조를 단순하게 해주며 주가 되는 보안 모듈에 복합적 보안 정책을 추가하게 된다.

2. SELinux의 보안체계 및 구조

1) SELinux의 접근제어 방법

SELinux는 MAC에 기반을 두고 접근제어를 한다. 보안 정책을 주체(사용자, 프로세스)나 객체에 대하여 관리 차원으로 규정할 수 있고, 커널에 SELinux을 구현하면 프로세스와 객체를 제어할 수 있다. 또한, 권한은 인증된 사용자가 아닌 이용 가능한 모든 보안 관련 정보에 의해 결정되어 진다.

2) SELinux의 보안정책 및 적용

SELinux에서 정책은 목표 정책(Targeted Policy)과 엄격 정책(Strict Policy)로 나눈다. 엄격 정책은 기본 설치 이외의 프로세스에 대해 단일의 엄격한 제한을 하는 것으로 전문적인 지식이 없는 사용자에게는 많은 어려움이 따른다. 목표 정책은 기본적인 제한을 표준 리눅스 보안에 의해 통제되도록 하고, 일부 프로세스에 대해서만 SELinux의 정책으로 접근을 제어하는 것을 말한다.

SELinux 보안 정책은 사용자, 프로그램, 프로세스 그리고 이들의 동작 대상인 파일과 디바이스를 포함한 시스템 전체, 즉, 모든 주체와 객체에 대한 접근 허가(access permissions)를 기술한다(그림 1 참조).

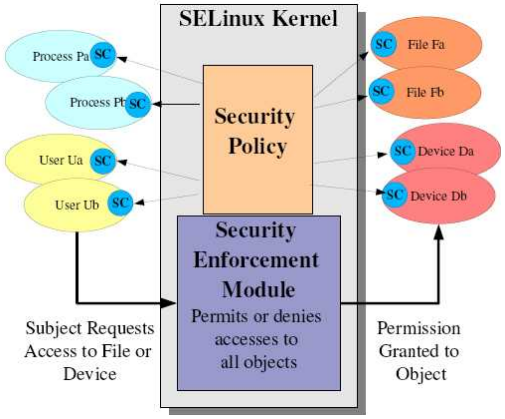

그림 1. SELinux의 보안 정책
Fig. 1 The security policy of SELinux

Ⅳ. 리눅스 접근제어 모듈 설계

1. 3단계 사용자 식별 인증

기존의 표준 UNIX와 Linux 시스템에서는 불법적인 목적으로 시스템에 들어온 침입자가 superuser(root) 권한을 획득하게 되면 시스템의 모든 객체들을 자신의 불법적인 목적에 맞게 조작하는 것이 가능하였다. 이러한 문제점 해결을 위해서 본 논문에서는 3단계의 과정을 거치는 사용자 식별 인증 모듈의 설계를 통하여 허가되지 않은 불법 침입자를 완전 차단하고자 한다. 기존 유닉스 및 리눅스 시스템에서 사용하는 사용자 정보들은 passwd 파일과 shadow 파일에 그대로 유지되며 사용자의 보안 정보는 이와 분리하여 별도로 저장된다. 사용자의 보안 정보를 저장하기 위해서 본 제안 시스템에서는 별도의 디렉토리와 파일을 유지하도록 하며, 사용자의 보안 정보를 안전하게 보호하기 위해 이 디렉토리는 root 보다 상위의 보안 관리자만이 접근할 수 있도록 보안 권한을 설정하여 관리한다.

보안 커널을 기존의 사용자 인증 정보(사용자 식별자, 암호)에 보안등급과 보호 범주를 추가하여 사용하도록 하며, 또한 시스템 관리자 계정인 root의 상위 수준에 존재하고 /etc/passwd에 존재하지 않는 보안 관리자라는 가상의 계정을 설정한다. 따라서 보안 관리자는 우선 root 인증을 거쳐 다시 보안 관리자 인증을 거치며, 요구되는 보안 등급과 보호 범주로 로그인을 변경하지 않은 상태에서는 통합 보안 관리 시스템 디렉토리에 접근할 수 없도록 설계한다. 또한 일반 사용자도 보안 관리자를 통해서만 최소, 기본 및 최대 보안등급과 보호 범주가 설정될 수 있으며, 보안등급이 없는 일반 사용자가 서버에 접속 시에는 보안등급과 보호 범주가 (0, 0)으로 설정되어 보안등급이 부여된 폴더나 파일에 접근할 수 없게 설계하는 것이 핵심이다 (그림 2 참조).

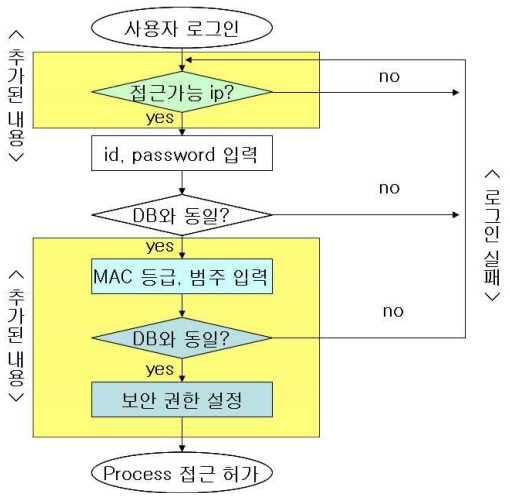

그림 2. 3단계의 사용자 식별 인증
Fig. 2. The three stages of user identification authentication

첫째, 네트워크 상에서의 비인가된 사용자나 해킹을 시도하는 모든 활동을 원천적으로 봉쇄하기 위해 접근을 시도하는 사용자의 IP 주소가 시스템에 보관(DB)된 접속 허가 IP 주소와 일치하는지를 판별한다. IP 주소가 일치하면 사용자가 로그인(ID, password) 정보 화면을 보여주고, 만약 IP 주소가 일치하지 않는다면 오류 메시지를 보여준다

둘째, ip 주소 확인이 끝나면 사용자의 ID와 password를 입력받아 DB에 저장된 내용과 비교하여 일치하면 로그인을 허가하고 불일치하면 오류메시지를 보여준다

셋째, 접근제어 정보로는 MAC의 등급(clearance)과 범주(category)를 사용하며, 사용자가 보안 관리자로부터 미리 할당 받은 등급과 범주를 입력할 수 있도록 한다. 입력된 내용이 보안 DB에 저장된 내용과 동일하거나 허가 범위 내에 있다면, 사용자에게는 요청된 접근제어 속성값을 가지는 프로세스가 할당된다. 그러나 사용자의 등급, 범주 중 한 가지라도 다른 값을 입력하게 된다면 사용자 인증은 오류 메시지와 함께 종료되고 이 내용은 log에 기록이 된다.

보안 DB의 내용은 접속 가능한 IP 주소와 사용자에 따른 MAC의 최소 등급과 최대 등급, 범주가 저장되도록 설계한다.

2. 주체 및 객체의 보안 권한 상속

기존의 표준 UNIX나 Linux 시스템은 superuser(root) 권한을 획득하면 모든 파일에 대해 수정 및 삭제가 가능하다. 그러나 보안 속성을 가진 어떤 사용자가 새로운 파일을 생성하거나 기존의 파일을 수정할 경우 사용자의 보안 속성이 자동으로 파일에 부여 된다면 보안 관리 프로세스 상의 취약성을 제거할 수 있다. 이에 본 논문에서는 주체 및 객체에 대한 보안 권한이 자동으로 상속되는 모듈을 제안하고자 한다. 보안 권한 상속이 이루어지는 주체와 객체의 대상은 다음과 같다.

첫째, 로그인 사용자에 MAC 보안등급(clearance), 범주(category)를 부여하고, 프로세스에 사용자의 보안 등급과 범주를 child 프로세스 생성시 부모 프로세스의 보안 권한을 상속하여 모든 주체에 대해 보안 인가 권한이 부여되도록 한다.

둘째, 일반 디렉토리(regular file), 디렉토리(directory), 장치 구동 파일(device special file)인 consol, terminal, printer, Single level device인 floppy, CD-ROM, tape, Special mechanism인 socket 등의 모든 객체에 대해 보안등급과 범주 레이블(label)이 부여되도록 한다.

그림 3은 인증된 사용자 프로세스가 생성 또는 클론 방식을 이용하여 프로세스를 생성할 때 각 보안 프로세스 정보 테이블에도 사용자의 MAC등급과 MAC범주 수록을 위한 저장 공간을 자동으로 확보할 수 있도록 하고, 확보된 task_struct 저장 공간에는 사용자의 보안 허가 등급과 보호 범주 정보가 복사되어 상속되도록 하는 과정이다.

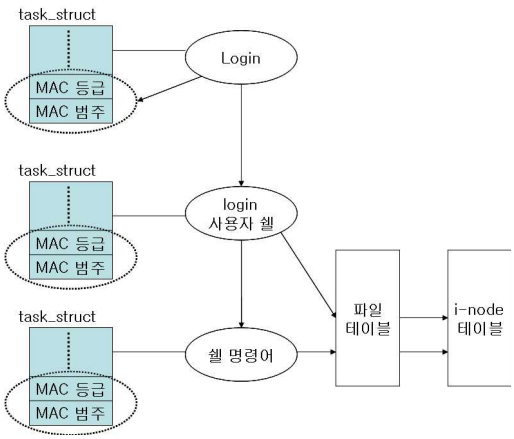

그림 3. task_struct 저장 공간의 보안 권한 상속
Fig. 3. The security authority inheritance of task_struct

즉, 주체(프로세스)가 보안이 설정된 객체에 접근할 때, 이미 설정되어 있는 주체의 보안 정보와 접근하려고 하는 객체의 보안 정보를 비교함으로서 접근 제어가 이루어지도록 하는 것이다.

3. DB를 이용한 실시간 감사 추적

기존의 리눅스 운영체제에서의 감사 추적은 정적으로 기록된 사건(event) 로그 파일을 이용하여 별도의 감사 추적 프로그램을 통해서 가능하였다. 이 경우에는 보안 관리자 또는 감사인(auditor)에 의한 실시간 감사 추적이 불가능 할 뿐만 아니라 다양한 사건 정보의 출력이 어려웠다.

그림 4는 기존의 상용화된 보안 운영체제 제품들의 DB를 이용한 실시간 감사 정보 시스템이다.

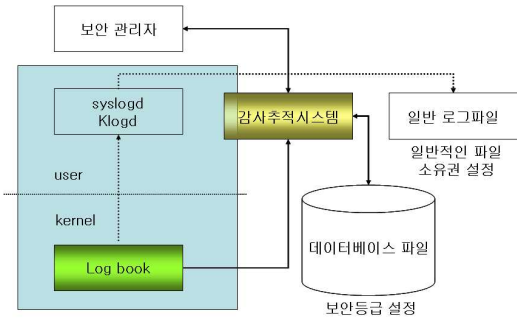

그림 4. Log book을 활용한 감사 정보 시스템
Fig. 4. An existing Audit Information System

보안 관리자는 SQL 질의어를 통해서 저장된 정보를 실시간으로 원하는 정보만을 볼 수 있다. 사용자에게 부여된 보안 속성과 사용자가 접근하고자 하는 객체 사이에 강제적 접근 제어가 커널 내에서 진행되기 때문에, 시스템 내의 모든 행위는 커널 내의 참조 모니터를 통해서 수집이 가능하다. 이를 통해 세세한 접근 정보에 이르기까지 감사 작업이 이루어지게 된다.

이러한 로그 DB 파일 또한 보안등급이 설정되어야 하며, 보안 관리자를 제외한 root나 일반 사용자는 접근이 통제되어 읽기, 쓰기가 불가능하다. 이를 통해 해커의 불법적인 접근 후 자신의 흔적을 제거하는 방법을 방지할 수 있다.

본 논문에서는 기존의 실시간 감사 정보 시스템의 기능을 세가지 영역으로 더 세분화하여 다음과 같이 제안한다.

첫째, 커널에 로그 정보를 데이터베이스에 연결할 수 있게 프로그램을 설계하고,

둘째, 시스템 호출을 이용하여 커널 영역의 메모리인 log book에서 로그 정보를 가져오는 audit 함수를 설계하며,

셋째로 커널과 시스템 호출을 이용하여 사용자 영역에서 기존의 klogd나 syslogd의 역할을 대신 수행하는 getlogd를 구성한다.

이 보안 감사 추적 시스템은 사용자가 컴퓨터를 사용하는 동안의 모든 행위를 운영체제 내부에서 감지하여 데이터베이스에 기록하고, 각 행위와 관련된 보안 감사 자료를 보안 관리자에게 실시간으로 제공하게 된다.

또한, 사용자에 부여된 보안속성과 사용자가 접근하고자 하는 자원 사이에 강제적 접근제어가 커널 내에서 진행되기 때문에 시스템내의 모든 행위는 커널 내에서 감사 작업이 이루어지게 된다. 감사 작업이 일회성으로 그치지 않고 이후 진행을 추적함으로써 시스템 내의 해킹시도와 같은 작업이 발생 되었을 경우에 대비하여 추적 감시 기능이 제공된다. 추적된 기록에 대해서는 시스템 관리자가 참고하도록 하여 차후에 새로운 보안 정책을 결정할 수 있는 기본 자료로 사용할 수 있을 것이다.

그림 5는 본 논문에서 제안하는 실시간 감사 추적 시스템의 구조도이다.

Ⅴ. 결 론

본 논문에서는 기존 SELinux를 포함한 리눅스 보안 운영체제와 보안 모듈에 관한 선행 연구 분석을 통해 사용자 식별 인증, 주체·객체의 보안 권한 상속, 참조 모니터와 MAC 등급 처리, DB를 이용한 실시간 감사 추적이 적용된 리눅스 접근제어 보안 모듈을 다음과 같이 제안하였다.

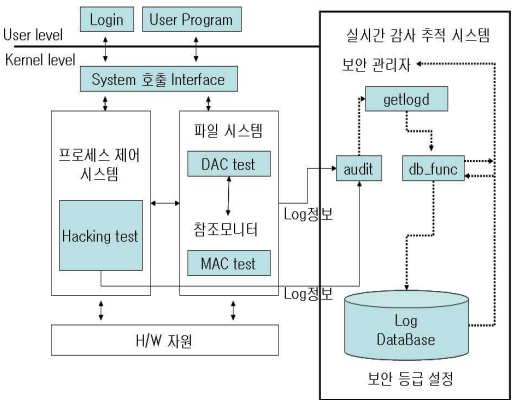

그림 5. DB를 이용한 실시간 감사 추적 시스템
Fig. 5. Real-time audit trailing system using DB

첫째, 사용자 인증 시 접속 허용 IP를 판별하고 보안등급과 범주를 입력하게 하여 superuser(root)와 보안 관리자의 권한을 분리하였다. 인증 시 사용자의 접속 ip를 판별하여 허가되속 i은 ip의 접속을 사전에 차단하며, 보안 관리자로부터 사전에 할당받은 등급과 범주root만 시스템 , 보안가능하기를 입력하여 불법 침입자가 superuser(root) 권한을 획득하였다 하더라도 시스템 파괴가 불가능 하도록 설계 하였다.

둘째, 주체 및 객체의 보안 속성 상속을 통하여 주체가 보안이 설정된 객체에 접근할 때, 설정되어 있는 주체의 보안 정보와 객체에 설

정되어 있는 보안 정보를 비교함으로서 접근 제어가 이루어지도록 하였다.

셋째, 커널상에서 현재 발생하는 모든 사건에 대해 참조모니터의 감사를 진행하며, 현재 설정되어 있는 MAC 보안 속성을 확인한 후에 접근 허용여부가 결정되기 때문에 악의적인 목적으로 여러 객체에 접근하는 것을 사전에 차단되도록 하였다.

넷째, DB를 이용한 실시간 보안 감사 추적 시스템을 통해 사용자가 컴퓨터를 사용하는 동안의 모든 행위는 운영체제 내부에서 감지되고 데이터베이스에 기록되며, 각 행위와 관련된 보안 감사 자료는 보안 관리자에게 실시간으로 제공되기 때문에 긴급 상황이나 문제 발생 시 즉각 대처할 수 있도록 설계하였다.

이상과 같이 제안된 리눅스 접근제어 보안 모듈이 실제로 구현된다면 TCSEC B1 등급 기준에서 요구하는 대부분의 기능을 만족시키게 될 것이다. 더불어 root 권한을 제한하여 접근 권한이 없을 때에는 보안등급 파일에 접근을 차단하며, DB를 이용하여 실시간으로 감사 추적이 가능하여 악의적인 해킹시도를 원천적으로 차단할 수 있을 것이다.

참 고 문 헌

[1] 김정순, 이재서, 이승용, 김민수, 노봉남, 리눅스 보안운영체제를 위한 접근통제 프레임워크, 정보보호학회지 제15권 제2호, 2005. 4.

[2] 손형길, 박태규, 이금석, 다중등급 보안 리눅스 구현 및 시험평가, 한국정보과학회 논문지, 2003. 06.

[3] 전준상, 정성재, 소우영, SELinux 서버의 정책 설정에 관한 고찰, 정보보호학회충청지부 Practices : SELinux, November, 2003

[6] S. Smalley. Configuring the SELinux policy. NAI Labs Report #02-007, available at www.nsa.gov/selinux, June 2002.

[7] Stephen Smalley and Timothy Fraser, A Security Policy Configuration for theSecurity-Enhanced Linux, NAI Labs Technical Report, February 2001. 추계, 2005. 10.

[4] 정보보호시스템 평가 인증 가이드, 한국정보보호센터, 2004.12.

[5] Dan Wlash, Elevating Security Best

저자약력

박 진 석(Jin-Seok Park) **정회원**

1995년 2월 : 충남대학교 기계공학교육학과 (공학사)
2006년 2월 : 한남대학교 정보보호학과 (공학석사)
2008년 2월 : 중부대학교 정보과학과 박사과정 수료

<관심분야> : 리눅스 보안, 학습평가시스템

김 순 곤(Soon-Gohn Kim) **종신회원**

1999년 8월 : 전북대학교 전자계산기공학과 (공학박사)
1987년 2월 : 동국대학교 전산교육학과 (교육학석사)
1995년 3월 ~ 현재 : 중부대학교 컴퓨터학과 교수

<관심분야> : 정보보호, 데이터베이스, 멀티미디어