안전한 리눅스 운영체제를 위한 접근제어 설계 및 구현

고종국° 두소영 은성경 김정녀
한국전자통신연구원(ETRI)
{jgko, sydoo, skun, jnkim}@etri.re.kr

Design of Access Control for Secure Linux OS and its Implementation

Jong-Gook Ko°, So-Young Doo, Sung-Kyung Un, Jung-Nyu Kim
Electronics and Telecommunications Research Institute

요 약

본 논문은 최근에 그 수요가 증가하고 있는 리눅스를 기반으로 하여 안전한 운영체제의 설계 및 구현에 대하여 기술한다. 안전한 운영체제를 위해 사용되는 보안 기능들은 MAC(Mandatory Access Control), DAC(Discretionary Access Control), 그리고 SOP(Separate of privilege)과 같이 파일,디렉토리, 그리고 디바이스와 같은 시스템 자원에 대한 접근을 제어하는 기능과 감사 추적(Auditing) 기능, 그리고 사용자 인증기능 등이 있다. 접근제어란 컴퓨터 자원, 통신자원, 정보자원 등에 대한 허가되지 않는 접근을 막는 것이다. 본 논문에서는 리눅스 운영체제의 보안을 위해 필요한 접근 제어 기능과 다중 레벨의 사용자 인증기능의 설계 및 구현에 대해 기술한다. 또한 접근제어기능 구현은 국제 표준화인 POSIX 1003.1e 을 기준으로 하였다.

Keywords : 접근제어,Mandatory Access Control, Discretionary Access Control, 감사추적,사용자 인증
Role Based Access Control

1. 서론

인터넷을 비롯한 분산 통신망 환경의 급격한 확산에 따라 많은 보안 취약점이 노출되고 이를 이용한 해킹 수법이 갈수록 지능화 되어가고 있다. 통신망의 거대와 정보자산의 고급화, 사용자 집단의 다양화는 공격기회의 증가, 해커수의 증가, 그리고 공격에 의한 피해의 대규모화를 초래하므로 정보시스템에 대한 보안은 시급한 문제로 대두되고 있다. 공개 운영체제로 잘 알려진 리눅스, FreeBSD 등은 시스템의 가격 대 성능 비와 안전성 측면에서 우수하고, 원천코드가 공개되어 있어 유닉스에 친숙한 사용자로 하여금 웹 서버, 메일 서버, DB 서버 등으로 많이 사용되고있다. 또한 공개 운영체제들은 다양한 환경에서 업무 처리 능력이 뛰어나며 유연성 및 확장성, 신뢰성이 뛰어나서 서버용 운영체제로 많이 사용되었으나 커널의 원천 코드가 공개되어 있어서 해커들에게 이용되어 네트워크 운영체제로서의 보안상 문제점들이 부각되었다. 이러한 문제점들을 해결하기 위한 방법으로 방화벽이나 침입탐지 시스템 등의 개발이 많이 이루어 지고 있는 상태이다. 하지만 보다 근본적이고, 원천적인 보안을 위해 운영체제 내에 보안 기능을 통합시킨 보안 커널을 추가로 이식하는 보안 운영체제 기술 개발의 필요성이 증가하고 있다[1][2]. 본 논문에서는 운영체제 레벨에서의 보안기능을 수행하는 접근 방식으로서 공개 운영체제 리눅스를 기반으로 한 보안 운영체제를 위해 접근 제어기능과 다중레벨의 사용자인증 기능의 설계 및 구현에 대해 기술한다. 2 장에서는 관련연구에 대해 소개하고 3 장에서는 접근제어기술과 다중레벨의 사용자 인증의 설계 및 구현에 대해 기술한다. 4 장에서는 결론 및 향후 과제에 대해 설명한다.

2. 관련 연구

보안 운영체제에 대한 연구와 개발은 미국을 비롯한 많은 선진 외국에서 정부 차원 또는 상업용으로

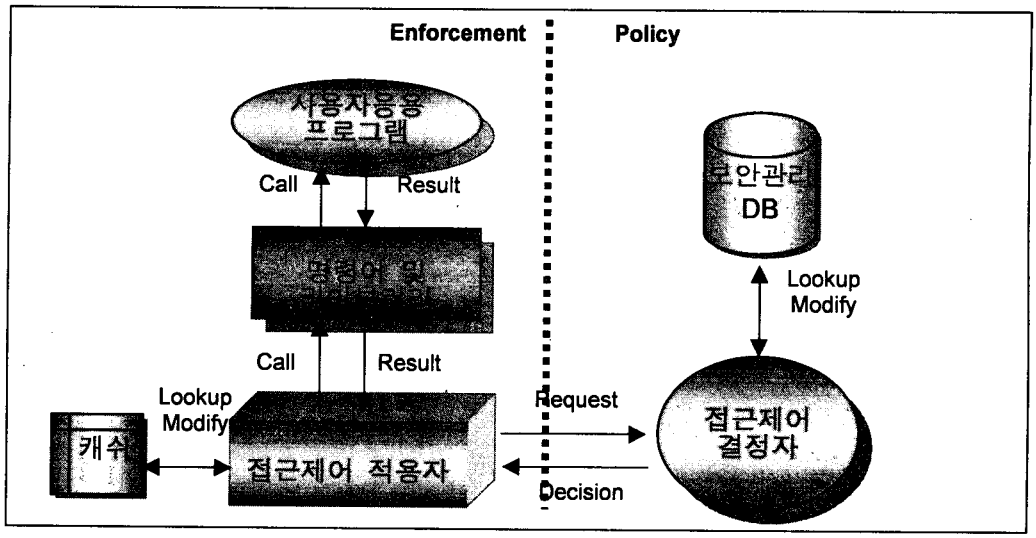

그림 1. 보안 운영체제 기능적 구성도

이루어져 왔으나 공개 운영체제인 리눅스나 FreeBSD 운영체제의 경우 커널 수준에서의 보안에 관한 연구는 그 취약성에 비하여 상당히 부족한 상황이다. 다만, 현재 진행되고 있는 커널 수준에서의 보안에 관한 프로젝트로는 미국의 Arizona 대학의 HOSANA 프로젝트로 공개 운영체제인 리눅스 운영체제에 네트웍의 IP 보안 프로토콜을 중심으로 다단계 보안 기능을 추가한 연구이다. Rule Set Based Access Control(RSBAC)[3]은 본격적으로 리눅스 운영체제에 접근 제어기능을 추가한 것으로 MAC, DAC[4], 그리고 RBAC[5]등의 기능들을 각 모듈별로 제공하고 있다. RSBAC 은 현재 리눅스 커널에 보안요소를 첨가한 오픈 소스 코드이고 각각의 보안 모듈에 기초한 유동적인 접근제어 시스템을 제공한다. 이 시스템은 파일과 디렉토리와 같은 컴퓨터 자원에 대한 접근을 각각의 접근제어 모듈에 근거하여 접근 허가 여부를 결정한 후 수행해 나가는 접근 제어 시스템이다. 접근 제어 시스템 중에서 접근 여부를 판단하여 알려주는 결정부(Access Decision Facility) 와 결정된 접근 권한을 적용시키는 적용부(Enforcement) 로 나뉜다. 모든 보안과 관련된 시스템 콜은 ADF(Access Decision Facility) 를 호출하는 보안 적용 코드가 추가된다. 그리고 ADF 는 차례로 각각의 활성화된 보안 모듈들을 호출하고 그 모듈들로부터 받은 결과를 종합하여 접근 허용 여부 결정을 내린다. 모든 주체와 객체에 대한 접근 정보는 완전히 차단된 디렉토리들에 저장되고 각각의 마운트된 장치에 하나씩 놓이게 된다. 그리고 이 디렉토리에 대한 접근은 RSBAC 시스템에서 제공하는 특정 시스템콜을 통해서만 접근이 가능하다. RSBAC 에서는 보안 관리자 또는 보안관리자 역할에 속하는 사용자만이 파일이나 디렉토리와 같은 시스템 자원의 보안 정보를 획득 및 설정을 할 수 있지만 본 프로젝트에서는 파일의 소유자등과 같이 해당 자원에 접근 권한이 있는 사용자도 보안 정보를 획득 및 설정 할 수 있도록 하여 접근제어 기능에 좀더 유연성과 효율성을 제공하였다.

3. 접근제어 기능 및 사용자 인증기능의 설계 및 구현

3.1 보안 운영체제 구성 및 접근제어 동작 내용

그림 1 은 보안 운영체제의 구성도를 나타내고 있다. 사용자 응용 프로그램은 명령어 및 라이브러리를 통하여 커널에 접근요청을 보낸다. 커널에서는 해당 접근 요청에 대하여 먼저 캐쉬 영역에서 해당 프로세스와 시스템 자원에 대하여 이미 결정했던 접근제어 정보가 있는지를 조사하여 없는 경우 접근제어 결정 결정자를 호출하는 역할을 한다. 접근제어 결정자는 사용자 영역에서 수행되고 해당 프로세스가 해당 자원에 접근권한이 있는지를 검사하는 기능을 수행한다. 접근제어의 동작흐름을 보면 먼저, 사용자응용 프로그램에서 명령어 및 라이브러리를 통해 보내진 접근 요청을 커널에서 받아 캐쉬에서 해당 프로세스와 해당 자원에 대해 접근제어 정보가 있는지를 먼저 확인하고 없는 경우 접근제어 결정자를 호출한다. 접근제어 결정자는 보안 관리 데이터베이스에 저장되어있는 보안 정보들을 가지고 MAC 과 ACL 의 보안 정책에 근거하여 해당 사용자가 해당 자원에 접근권한이 있는지를 검사하여 그 결과를 다시 커널에 보낸다 커널은 접근제어 결정 결정자로 부터 받은 접근 결정 정보를 가지고 사용자 응용 프로그램이 해당 자원에 접근 할 수 있게 하거나 또는 접근 할 수 없음을 알린다.

3.2 접근제어 정책

본 프로젝트에서는 접근제어 결정자에서 접근 권한 결정을 위한 보안 정책으로 MAC, DAC 그리고 SOF 정책을 기반으로 하고 있다. 또한, 접근 제어 구현에

있어서 RSBAC 과 달리 POSIX 표준화에 따르고 있다. BLP 모델[6]의 ss-property 와 *-property 원칙을 따르는 MAC 에서는 각 주체(프로세스)와 각 객체(시스템 자원:파일,디렉토리,디바이스등)들에게 등급과 범주를 주고 이 등급과 범주를 기준으로 낮은 수준의 등급과 범주를 갖는 주체가 높은 수준의 등급과 범주를 가지는 객체를 읽을 수 없도록 하거나 높은 수준의 등급과 범주를 갖는 주체가 낮은 수준의 등급과 범주를 가지는 객체를 쓸 수 없도록 하여 인가되지 않는 정보의 유출을 막아 기밀성을 보장하도록 하였다. DAC 에서는 각 객체마다 그 객체에 접근할 수 있는 주체들의 접근 속성(읽기,쓰기,그리고 실행) 리스트들을 가진다. 기존의 리눅스에서 제공하는 user, group 그리고 other 와 함께 확장된 user , group 리스트를 제공하도록 하였다. 확장된 user, group 란 실제 객체의 소유자 와 객체가 속해있는 그룹이외에 임의의 다른 사용자 또는 임의의 다른 그룹에 그 객체에 대한 접근 속성 부여 할 수 있도록 한 것이다. 그림 2 는 /home 디렉토리 객체에 대한 접근제어 리스트의 예를 나타낸다.

TAG	ID	PERMISSION
user	100	r w x
group	500	r w
other		r
TAG	ID	PERMISSION
extended_user	110	r w
	113	r w
	115	r x
extended_group	501	r w
	504	r
	505	r

그림 2. 접근제어 리스트 예

한 객체마다 소유하는 접근제어 리스트의 총 엔트리는 POSIX 표준에 따라서 16 개로 정하였다. 권한 분리(SOP) 기능에서는 막강한 시스템관리자(루트)의 권한을 분리 시키려는 목적에서 시스템의 사용자 영역을 시스템관리자(루트), 보안관리자, 그리고 일반 사용자로 나누었다. 보안 정보들을 보관하는 보안 데이터베이스의 접근 및 수정은 보안 관리자만이 할 수 있고 시스템 관리자나 일반 사용자들은 보안 정책(MAC,DAC)에 기준 하여 권한이 있는 경우에만 접근 및 수정을 할 수 있다. 이러한 방식으로 각 사용자들을 분리하여 역할에 기반한 접근제어 방식으로 확장될 수가 있다. 또한, 루트의 권한을 축소시켜서 루트 권한이 해킹 되었을 때에도 피해를 줄일 수 있다.

3.3 다중레벨의 사용자 인증

본 프로젝트에서는 다중레벨의 사용자 인증과정을 위해 사용자가 시스템에 로그인 시 기존의 사용자 아이디와 패스워드 입력 이외에 사용자의 등급 및 범주를 입력하도록 하여 사용자 인증을 하도록 하였다. 사용자에게 허용되는 최대 등급, 최소 등급 그리고 최대 범주 값을 부여하여 사용자가 자신에게 주어진 해당 범위 안에서 로그인 할 수 있도록 하였다. 그래서 해당 범위 안에서 사용자가 여러 등급으로 로그인하여 작업할 수 있도록 하였다.

4. 결론 및 향후 과제

본 논문에서는 안전한 리눅스 운영체제를 위해서 접근제어의 설계 및 구현에 대한 내용을 기술하였다. 접근제어 기능을 수행하기 위해 필요한 보안 정책으로는 Mandatory Access Control(MAC) 과 Discretionary Access Control(DAC), 그리고 SOP(Separate Of Privilege) 들의 정책들을 기반으로 하였다. 사용자 인증 시 기존의 아이디와 패스워드 이외에 사용자의 등급 및 범주 값을 이용하여 인증 시에 사용되도록 하고 허용된 범위 내에서 사용자가 여러 등급을 로그인할 수 있도록 하였다. 그러나, 보안 관리 데이터 베이스 파일에서 보안 정보들을 읽거나 쓸 경우 초래되는 성능 저하 문제와 보안 관리 데이터베이스의 백업 문제들은 앞으로 개선되어야 할 문제들이다.

참고문헌

[1]Kevin Brady. "Integrating B2 Security into a UNIX System", Proceedings of UniForum Conference, 1992.

[2]Mark Funkenhauser. "B1 TUNIS : A Kernel for a Secure UNIX System", Canadian Computer Security Conference, 1989.

[3] "Rule Set Based Access Control", http://www.rsbac.de/

[4] Ingrid M. Olson. Marshall D. Abrams. "Computer Access Control Policy Choices", Computer & Security. Vol. 9. pp.699-714. 1990.

[5]R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. "Role based access control models", IEEE Computer, 29(2), February 1996.

[6]Roos Lindgreen, Herschberg I. S. "On the Validity of the Bell-Lapadula Model", Computer & Security. Vol. 13. pp.317-338. 1994.