情 報 保 護 學 會 誌
第 15 卷 第 2 號, 2005. 4

리눅스 보안운영체제를 위한 접근통제 프레임워크

김 정 순*, 이 재 서*, 이 승 용*, 김 민 수**, 노 봉 남***

요 약

기존 운영체제의 접근통제기법은 광범위한 사용자의 요구사항을 충족시키고, 강력한 시스템보안을 제공하기에 불충분하다. 최근의 보안운영체제는 다양한 접근통제기법을 지원하고, 섬세한 보안서비스를 제공할 수 있는 방향으로 연구되어 왔다. 본 논문에서는 현재까지 연구된 보안운영체제에 대한 연구내용과 핵심기술, 개발동향 및 평가기준, 그리고 다양한 접근통제기법을 지원하는 접근통제 프레임워크에 대하여 살펴본다.

Ⅰ. 서 론

보안운영체제는 기존의 운영체제 서비스에 접근통제, 암호화 등의 보안 서비스를 추가하거나 보완하여 구성된 신뢰할 수 있는 전산 환경(Trusted Computing Base : TCB)의 구현이다. 보안운영체제에 관한 대부분의 연구는 접근통제 메커니즘을 강화하는데 초점을 맞추고 있다. 보안운영체제의 동작 기반을 제공하는 보안 커널은 참조모니터의 구현이며, 접근통제는 시스템 내에서 발생하는 접근이 적절한지를 판단하는 과정이다. 참조모니터는 접근 발생 시 접근주체와 접근객체로부터 접근 결정에 필요한 정보를 추출하며 보안 규칙에 위배되지 않는지를 판단한다.[8]

정보보호기술이 점차로 능동형으로 진화하고 있으며, 컴퓨터 범죄의 증가로 인해 보안운영체제에 대한 필요성과 사용자 요구사항 및 강력한 시스템보안을 제공하기 위해 다양한 접근통제를 지원할 수 있는 접근통제프레임워크의 필요성이 증대되었다.

그러나 유닉스 및 리눅스 계열 운영체제의 접근통제기법은 주로 임의적 접근통제기법을 사용하고, 보안운영체제는 대부분 강제적 접근통제기법, 임의적 접근통제기법, 역할기반 접근통제기법 등 하나만을 구현한다. 이러한 방법은 광범위한 사용자의 요구사항을 충족시키고 강력한 시스템보안을 제공하기에는 불충분하다. 따라서 최근의 보안운영체제는 다양한 접근통제기법을 지원하고, 섬세한 보안 서비스를 제공할 수 있는 방향으로 연구되어 왔다.

본 고에서는 보안운영체제 개념, 보안운영체제의 개발 동향 및 평가기준에 대하여 살펴보고, 접근통제모델과 보안운영체제에 적용된 접근통제프레임워크들에 대하여 상세히 살펴본다.

Ⅱ. 보안운영체제

2.1 보안운영체제의 정의

보안운영체제는 기존의 운영체제 내에 내재되어 있는 문제점을 해결하기 위하여 보안 커널을 이식하고, 인증, 암호화 등의 보안 서비스를 추가하여 구현한 신뢰할 수 있는 전산 환경이다.[8,19] 보안운영체제는 참조모니터의 구현을 통해 사용자에 대한 식별 및 인증, 접근통제, 침입탐지 등의 기능을 통하여 커널 레벨로의 접근제어 및 알려지지 않은 공격에 대한 방어, 탐지 및 대응 등의 기능을 수행할 수 있다. 또한, 엄격하게 분리된 영역에 로그, 감사기록을 두어 보호함으로써 불법침입자가 침입의 흔적을 제거하는 행위를 방지하는 기능을 포함하고 있다.

2.2 보안운영체제의 요구 사항

보안운영체제를 개발하는데 있어서 고려되어야 할

본 연구는 정보통신부 대학 IT연구센터 육성, 지원사업의 연구결과로 수행되었습니다.
* 전남대학교 리눅스보안연구센터 ({cybersun, mirr1004, birch}@lsrc.jnu.ac.kr)
** 목포대학교 정보보호학과 (phoenix@mokpo.ac.kr)
*** 전남대학교 전자컴퓨터정보통신공학부 (bbong@jnu.ac.kr)

요구사항은 설계시 요구사항과 구현시 요구사항으로 나누어 볼 수 있다.[20] 설계시 요구사항으로 최소권한(least privilege), 보호의 경제성, 개방형 설계, 완전한 중재 및 조정 그리고 권한 분리, 최소공통, 사용의 용이성을 들 수 있으며, 구현시 고려사항으로는 식별 및 인증, 강제적 접근통제 및 임의적 접근통제, 객체 재사용 방지, 완전한 중재 및 조정을 들 수 있다.

위의 설계 및 구현시의 요구사항은 아래의 표 1에서 정리되어 있다.

〔표 1〕 보안운영체제 설계 및 구현시 요구사항

구 분	목 록
설계시 요구사항	☐ 최소권한 ☐ 보호 메커니즘의 경제성 ☐ 개방형 설계 ☐ 완전한 중재 및 조정 ☐ 허용에 기반한 접근 ☐ 권한 분리 ☐ 최소공통 메커니즘 ☐ 사용의 용이성
구현시 요구사항	☐ 식별 및 인증 ☐ 강제적 접근통제 ☐ 임의적 접근통제 ☐ 객체 재사용 방지 ☐ 완전한 중재 및 조정

2.3 보안운영체제의 구현 방법

보안운영체제는 기존 운영체제에 참조 모니터를 이용한 보안 커널을 이식하고 인증 및 암호화 등의 보안 서비스를 강화하여 구현한다.[21]

이 절에서는 보안운영체제라고도 할 수 있는 TCB에 대해서 설명하고 TCB를 이루는 보안 커널과 보안 커널의 핵심인 참조 모니터에 대해서 설명한다.

2.3.1 TCB

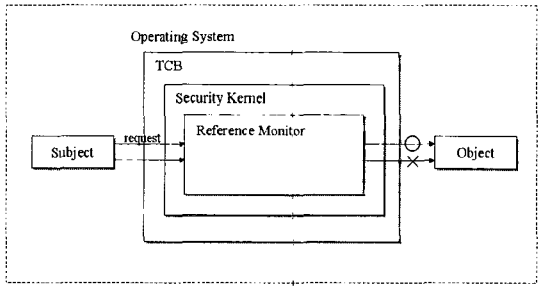
TCB는 시스템이 제공하는 보안 수준을 다루지는 않으며, 신뢰할 수 있는 수준만을 대상으로 다룬다. 전산 시스템이 단지 운영체제로만 구성되지 않기 때문에 TCB는 단지 운영체제만을 다루지 않고, 하드웨어, 소프트웨어, 구성요소 그리고 펌웨어 등 대부분을 다루고 있다. 마찬가지로 시스템의 모든 부분이 신뢰될 필요는 없기 때문에, 신뢰할 수 있는 수준의 평가 작업의 일부만이 TCB를 구성하고 있는 구조, 보안 서비스, 그리고 보증 메커니즘을 식별하는 것이고, 위험적인 활동에 대해서 어떻게 TCB가 보호되는지를 제시하여야 한다.

2.3.2 참조 모니터

참조 모니터는 안전하다고 여겨지는 추상적인 개념이다. 다음 장에서 설명할 보안 프레임워크들은 참조 모니터에 대한 구현이라고 할 수 있다. 참조 모니터에서는 악의적인 주체가 허가되지 않은 접근이나 변경을 하고자 할 때에 대상이 된 객체를 보호하기 위해 객체에 대한 모든 접근을 중재하는 것이다.

그림 1은 보안운영체제의 구조이다. TCB 내부에 보안 커널을 위치시키고 보안 커널 내부에 참조 모니터가 위치하고 있다. 참조 모니터는 보안 커널의 가장 중요한 부분으로 객체에 대한 접근통제, 감사, 식별 및 인증, 보안 매개변수 설정을 다른 보안 메커니즘과 데이터를 교환하면서 상호 작용을 한다.

이와 같은 보안 메커니즘에 의해 주체의 객체에 대한 접근요청은 정당할 때만 허용되어진다.



〔그림 1〕 보안운영체제의 구조

2.3.3 보안 커널

보안 커널은 운영체제만을 대상으로 하지 않는다. 하드웨어와 펌웨어, 소프트웨어 어플리케이션과 같은 모든 전산 환경의 요소들이 보안대상이며 참조 모니터, 감사 모듈, 식별 및 인증 모듈 등 다양한 보안 메커니즘을 통합하여 구성된다.

보안 커널은 완전해야하기 때문에 다양한 요구사항이 존재하게 된다. 다음의 보안 커널에 대한 세 가지는 주요 요구사항이다. 첫 번째, 참조 모니터 개념을 수행하는 프로세스와는 분리되어야 한다. 그리고 그들은 부정하게 조작될 수 없어야 한다. 두 번째, 참조 모니터는 주체와 객체 사이에서 발생하는 모든 행위를 대상으로 접근 통제하여야 한다. 참조 모니터를 우회하여 회피하는 것은 불가능해야 한다. 즉, 보안 커널은 완전하고 잘못될 수 없는 방식으로 구현되어야만 한다. 마지막으로, 완전하고 포괄적인 방식으로 입증되고, 시험될 수 있도록 충분히 작아야 한다.

2.4 보안운영체제의 개발 동향

운영체제 기술 발전의 흐름에 따라 보안운영체제 또한 기존의 통합 커널(Integrated Kernel : IK) 방식에서 마이크로 커널(Micro Kernel : MK) 방식으로 바뀌고 있다.

대표적인 보안운영체제로 미국에서 진행된 DTOS Distributed Trusted Operating System) 프로토타입 구현을 끝으로 종료된 Synergy 연구 과제의 후속으로 진행된 Flask 모델을 리눅스 시스템에 적용한 SELinux(Security Enhanced Linux)가 있다. 국내외에서 개발된 보안 기술은 운영체제의 발전에 따른 보안운영체제 개발 기술은 표 2와 같다.

SELinux는 NSA(National Security Agency)의 주도하에 여러 연구소에 의해 구현된 보안 리눅스 운영체제이다. SELinux의 가장 큰 특징은 Flask 구조를 사용하여 어떠한 운영체제에도 적용 가능하게 보안구조를 정의해 놓고 있어, 어떤 운영체제에서도 유연하게 잘 동작한다는 장점을 가지고 있다.

RSBAC(Rule Set Based Access Control)은 Abrams와 Lapadula에 의한 GFAC(Generalized Framework for Access Control)에 기초한 보안운영체제이다. 몇몇 커널 모듈에 기초해 접근 제어를 할 수 있는 유연한 시스템을 제공한다. 이외에도 Medusa DS9은 커널 수준에서 사용자 공간 인가 서버를 지원함으로써 리눅스를 확장한다.

최근에는 국내에서도 활발하게 보안운영체제 개발을 위해 노력하고 있는 상황이다. 국내의 대표적 보안운영체제로는 시큐브레인의 Hizard 및 시큐브의 TOS, 티에스온넷의 RedOwl 그리고 ETRI의 SecuROS 등이 있다. 현재 전남대학교에서도 보안운영체제를 위한 접근통제 프레임워크를 개발 중에 있다.

〔표 2〕 국외의 보안운영체제 개발 동향

보안운영체제	특 징
UNICOS MLS	☐ Cray UNICOS 9.0 기반의 다중수준 보안운영체제 ☐ TCSEC Strict B1 등급 지원 ☐ 참조 모니터 개념 구현
Synergy DTOS · Flask	☐ NSA 주도하에 개발된 강력하고 유연성 있는 차세대 운영체제 ☐ Secure Computing사 개발, 1997년 7월 완료 ☐ 수정된 Mach 커널과 외부 Security Server로 구현 ☐ 유연성, 투명성, 모듈성, 계층적 보안을 설계원칙으로 구현 ☐ 유타 대학의 Flux 프로젝트 결과인 Fluke 마이크로 커널에 보안기능 강화
Security Enhanced Linux	☐ Flask 과제 후속으로 리눅스 운영체제에 적용 ☐ 2000년 12월말 1차 공개
EROS	☐ EROS(Extremely Reliable OS) ☐ GNOSYS(구 KOS)의 세 번째 구현, TymShare와 Inc.에 의해 제작
RSBAC	☐ GFAC(Generalized Framework for Access Control) 모델을 리눅스 커널에 통합 ☐ 유연한 보안기능 제공
DTE	☐ TE의 더욱 강화된 형태 ☐ 사용자, 파일 , 네트워크 사이의 접근 조정 목적
LIDS	☐ MAC 구현 보안운영체제 ☐ 다양한 객체에 대한 접근 제어 가능
Medusa DS9	☐ 표준 리눅스 보안 구조 확장 ☐ 인가서버를 구현한 보안 구조

2.5 보안운영체제의 평가기준

정보화 역기능으로부터 주요 데이터를 보호하기 위해서는 안전성과 신뢰성이 검증된 정보보호시스템을 사용하여 조직의 정보보호 수준을 향상시킬 수 있도록 정보보호시스템 평가인증제도에 대한 필요성이 더욱 높아지고 있다.[23]

미국, 영국, 독일, 프랑스, 캐나다 등 선진국에서는 10여 년 전부터 정보화역기능의 위험을 방지하기 위해 제도적 일환으로 자국의 환경에 적합한 평가인증제도를 마련하여 정보보호시스템을 평가하고 있으며, 국내에서도 1998년 2월부터 평가인증제도를 시행중에 있다.

최근 선진국에서는 국제공통평가기준이 개발되어 국제표준(ISO/IEC 15408)으로 제정 되었으며, 이를 기반으로 각 국가에서 평가한 결과를 상호 인정하는 상호인정 협정이 체결 되었다.

2.5.1 TCSEC

미국은 1960년대 후반 컴퓨터 보안 연구를 시작하여 보안정책, 보안 요구사항 등 컴퓨터 시스템 보안을 위한 지침을 발표하였다. 그리고 1983년에 TCSEC(Orange Book)를 발표하였다. TCSEC의 등급은 C1, C2, B1, B2, B3, A1 의 6등급으로 구성되며, D등급은 부적합 판정을 의미한다.[18]

TCSEC의 요구사항은 보안정책, 책임성, 보증, 문서화로 이루어져 있으며, 등급체계는 A, B, C, D로 크게 구분되며, 등급단계에 따라 요구되는 보안기능

및 보증 요구사항들이 추가 된다. D급은 평가가 수행이 되었지만, 평가등급의 요구사항을 만족하지 못한 시스템을 위하여 존재하며, B등급에서는 가장 중요한 요구사항으로 보안레이블의 무결성을 보장하고, 가장 높은 등급 분류인 A는 정형화된 검증방법을 사용하는 것이 특징이다. TCSEC의 등급체계는 높은 등급으로 갈수록 보안기능요구사항과 보증요구사항이 체계적으로 증가한다.

2.5.2 ITSEC

ITSEC은 보안기능 부분과 보증 부분으로 이루어져 있고, TCSEC과는 달리 단일 기준으로 모든 정보보호제품을 평가하고자 하였기 때문에 보안기능은 제품이 사용될 환경을 고려하여 개발자가 보안기능을 설정하거나, TCSEC 혹은 ZSEIC에서 미리 정의한 보안기능을 사용하도록 하고 있다.

ITSEC의 특징은 시스템과 제품을 동일한 평가기준으로 평가하도록 되어있으며, 새로운 보안기능의 정의가 용이하고 등급의 평가는 보증 평가만으로 이루어지고 있다.

ITSEC은 TCSEC과의 호환을 위해 F-C1, F-C2, F-B1, F-B2 및 F-B3등 5가지와 독일의 ZSIEC의 보안기능을 이용한 F-IN(무결성), F-AV(가용성), F-DI(전송데이터 무결성), F-DC(비밀성) 및 F-DX(전송데이터 비밀성)등 총 10가지를 제공해 주고 있다.

ITSEC의 보안기능 등급체계는 TCSEC을 기본으로 하고 있으며, 등급은 E1(최저), E2, E3, E4, E5 및 E6(최고)의 6등급으로 나누고 있으며, E0등급은 부적한 판정을 의미한다. 효용성 보증 요구사항은 모든 시스템에 대해 공통으로 요구되는 것이며, 등급은 정확성 보증을 위해 요구되는 사항에 따라 결정된다.

2.5.3 CC

세계 각국의 평가 기준이 상이하여 평가에 소요되는 비용과 시간이 많이 소요되며, TCSEC, ITSEC, CTCPEC, FC등의 평가기준 통합의 필요성을 절감한 미국(NIST, NSA), 캐나다(CSE), 프랑스(SCSSI), 독일(BSI), 네덜란드(NL-NCSA) 및 영국(CESG) 등 6개국이 1993년 CC를 개발하기로 합의를 하였다. 1998년 5월 버전 2.0이 발표된 상황이며, ISO/IEC ITC1/SC27/WG3에서 표준화를 위한 노력도 병행하고 있다.

우리나라도 최근 국내 정보보호산업의 국제경쟁력을 강화하고 다양한 정보보호시스템을 평가하기 위해 지난 99년 6월 국제표준으로(ISO 15408)으로 채택된 CC를 국내 정보보호시스템 평가기준으로 수용하여 시행중에 있다.

Ⅲ. 접근통제모델

접근통제 정책은 1985년 미국 국방성에서 작성된 TCSEC에 의해 크게 임의적 접근통제 정책과 강제적 접근통제 정책의 두 가지로 분류된다.[18] 임의적 접근통제 정책에서는 정보객체 소유자가 임의적으로 자신이 소유하고 있는 정보객체에 대한 접근허가 여부가 결정되는 방식이며, 강제적 접근통제 정책은 미리 정의된 엄격한 규칙에 의해 정보가 저장된 객체에 대한 사용자의 접근허가 여부를 판단하는 특징을 가진다. 또한 Sandhu는 1996년 이 두 정책 모두 실제 기업 환경에 적용되기에는 부적합한 특성이 있다고 언급하고 역할기반 접근통제 정책을 제안했다. 역할기반 접근통제 정책은 기업 환경뿐만 아니라 데이터베이스, 운영체제 등에 적용될 수 있는 매우 유연한 접근통제 정책으로, 임의적 또는 강제적 접근통제 정책보다 정보에 대한 추상적인 접근통제와 효율적인 접근권한 관리를 수행할 수 있다는 장점을 가지고 있다.

3.1 임의적 접근통제(DAC)

임의적 접근통제 정책은 정보객체에 대한 사용을 요청하는 사용자의 신원(identification)에 근거를 두고 접근허가를 결정하는 정책이다. 여기서 '임의적'이라는 용어는 정보객체에 대한 접근권한을 보안 관리자의 판단에 의해 부여하거나 철회할 수 있다는 것을 의미한다. 임의적 접근통제 정책 중 대표적인 예가 정보객체의 소유자에 대한 임의적 접근통제 방법이다. 이 방식은 접근권한의 통제가 정보객체의 소유자에 의해 다른 사용자에게 허가되거나 철회할 수 있으며, 정보객체의 소유자에 의해 접근통제가 이루어짐으로써 정보보호 기능이 분산되어 실행됨을 의미하며, 중앙집중형의 접근통제가 용이하지 않는 특성을 가진다.[7] 또한 이 정책은 강제적 접근통제 정책보다 유연한 정보보호 메커니즘을 제공할 수 있는 장점이 있으나, 이 정책 역시 상업 환경에 적용에 부적합한 특성을 가지고 있다.[16]

3.2 강제적 접근통제(MAC)

강제적 접근통제는 시스템 보안관리자에 의해 부여

된 사용자와 정보객체의 보안 등급에 의해 정보에 대한 접근허가 여부를 결정한다. 여기서, 사용자에게 부여되는 보안 등급을 인가 등급(clearance level), 정보객체에 주어지는 보안 등급을 비밀 등급(secrecy level)이라고 한다. 정보객체에 주어지는 보안 등급은 정보객체에 포함된 정보가 불법적으로 누출되었을 때 입게 되는 손해의 정도, 즉 정보의 중요도를 나타낸다. 한편, TCSEC에서는 규칙기반(rule-based) 접근통제 정책과 동일한 개념으로 강제적 접근통제 정책을 정의하고 있다.

강제적 접근통제 정책을 구현하기 위한 주요 모델로서는 정보의 비밀성 보장을 위해 정보의 엄격한 일방향 흐름 통제를 위한 Bell-La Padula(BLP) 모델[3,4]과 정보의 무결성을 위한 Biba 모델[5,6] 등이 있다.

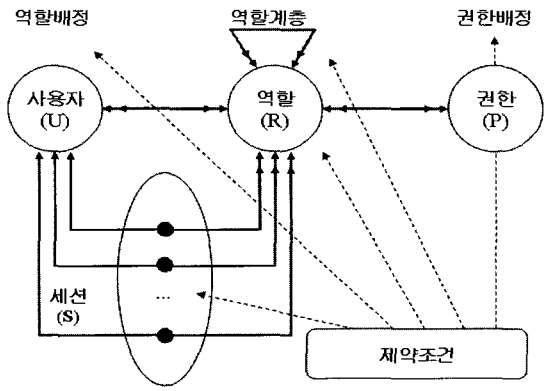
강제적 접근통제 정책은 접근통제 판단의 기준이 되는 보안 레이블을 사용자와 정보객체에 부여하고, 접근통제 규칙을 정의하는 기능이 제한된 수의 보안관리자에 의해 관리되므로 시스템의 보안관리가 중앙집중형으로 이루어진다. 그러나 BLP, Biba 모델이 적용되기 위해서는 시스템을 구성하는 사용자와 정보객체에게 계층적 구조를 가지는 보안 등급이 일관성 있게 부여될 수 있는 환경이어야 하며, 사용자와 정보객체의 수가 많아지고 다양한 보안 특성을 가지게 되는 다른 환경에서는 적용이 용이하지 않는 문제점을 가지고 있다.

3.3 역할 기반 접근통제(RBAC)

역할 기반 접근통제 모델은 정보에 대한 연산을 수행할 수 있는 권한들을 사용자에게 직접 할당되지 않고, 주어진 기업 환경에서 정의된 역할에 대해서만 배정한다. 따라서 사용자가 원하는 정보에 대한 연산을 수행하기 위해서는 먼저 해당 정보에 대한 연산을 실행할 수 있는 권한을 가진 역할의 소속원이 되어야 한다. 사용자를 특정 역할의 소속원으로 배정하는 권한은 미리 정해진 보안 관리자들에 의해 수행될 수 있다. 그리고 전체 시스템의 보안 관리를 위한 통제가 몇 명의 보안관리자에 의해 이루어지게 되므로, 임의적 접근통제 정책에서 발생할 수 있는 접근권한 통제의 어려움을 해결할 수 있다.

그림 2는 역할기반 접근통제 모델의 주요 구성요소와 구성요소 간의 관계를 나타내고 있다.[16]

이처럼 역할기반 접근통제 모델에서는 권한 관리를 사용자와 정보 객체간의 관계로 인식하는 대신 기업



〔그림 2〕 역할기반 접근통제 모델의 구성요소와의 관계

환경에서의 역할과 정보 객체간의 관계로 설정, 관리함으로써 사용자와 정보 객체의 수가 대단히 많은 실제의 기업 환경에 매우 적합한 특성을 제공한다.[2,16] 또한 최소권한 원칙(least privilege principle), 임무분리(Separation of Duty : SoD), 차료 추상화와 같은 주요 보안 원칙들 역시 지원하고 있다.

Ⅳ. 접근통제 프레임워크

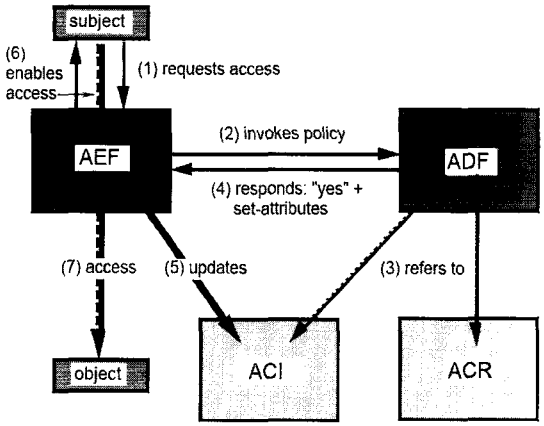
4.1 GFAC와 RSBAC

4.1.1 GFAC 프레임워크

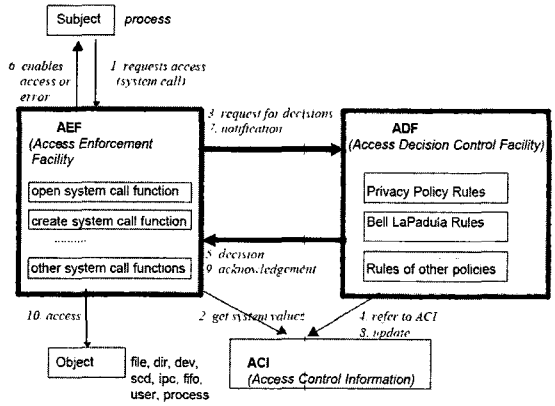
GFAC(Generalized Framework for Access Control) 프레임워크는 ISO에서 정의된 WDACF (Working Draft on Access Control Framework)의 개념을 이용하여 만들어진 프레임워크이다.[13] GFAC 프레임워크의 모든 접근통제는 기초적인 개념들의 집합에 기반하고 있다. GFAC 프레임워크는 ISO 표준인 WDACF로부터 전문용어 및 개념을 활용하였다.[9,13]

GFAC 프레임워크는 그림 3과 같이 크게 판결(adjudication)부분인 ADF(Access Control Decision Facility)와 수행(enforcement)부분인 AEF (Access Control Enforcement Facility)으로 구성된다. AEF는 TCB 역할에 대응하고, ADF는 TCB의 일부인 보안정책을 구체화하는 접근통제 규칙들에 대응한다.

주체의 객체로의 접근은 AEF에 의해 모니터 되고 AEF는 ADF에게 접근에 대한 허용 여부를 요청하게 된다. ADF는 주체 및 객체에 정의되어진 접근통제 규칙을 이용하여 결정을 내리고 그 결과를 AEF에게 응답하게 되고 AEF는 ADF의 결과를 적용시킨다.[13]



〔그림 3〕 GFAC 프레임워크 구조도



〔그림 4〕 RSBAC 프레임워크 구조도

ACI(Access Control Information)는 주체, 프로세스, 객체에 대한 정보를 보관한다. 이외에도 ACR (Access Control Rules)은 IF-THEN로 이루어진 논리적 상태들의 집합을 보관한다.[11]

4.1.2 RSBAC

RSBAC(Rule Set Based Access Control)은 지난 2000년에 GFAC 프레임워크를 기반으로 개발된 아주 강력하고 유연한 기능을 가진 리눅스 기반 공개용 접근통제 프레임워크이다. 즉, RSBAC은 GFAC 프레임워크를 리눅스 시스템에 적용한 것이다. RSBAC의 구조는 그림 4와 같이 ADF와 AEF로 구성된 GFAC 프레임워크와 같다.

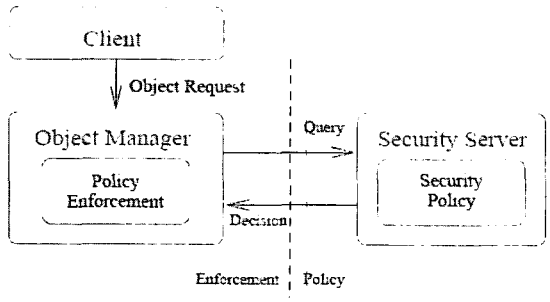
ACI는 주체, 프로세스, 객체에 대한 모든 자원들의 보안 속성을 저장하게 된다. AEF, ADF, ACI는 모두 보안정책과는 독립적으로 구성된다.[11]

주체의 객체에 대한 접근이 요청되면 AEF는 이를 수신한 후 ACI에서 해당 주체, 객체 등에 보안 속성을 획득한다. 이렇게 획득된 보안 속성과 함께 ADF로 허용 여부에 대한 결정을 의뢰하게 되고 ADF는 ACI의 정보와 보안 정책을 이용하여 허용 여부를 판단하고 ACI의 값을 수정한 후 허용 여부 결정을 AEF에게 보내게 된다. 이렇게 보내진 허용여부는 AEF에 의해서 집행되게 된다.

4.2 Flask와 SELinux

4.2.1 Flask 프레임워크

Flask(Fluke Advanced Security Kernel)는 유타(Utah)대학의 Flux 프로젝트의 결과인 Fluke 마이크로 커널에 보안 기능을 강화한 새로운 버전이다. DARPA (Defense Advanced Research Project Agency)의 지원을 받으며 유타대학교 Flux 팀과 미 국방부(Department of Defense : DoD)와 협력으로 개발 진행 중이며 DoA(Department of Army)와 AFRL(Air Force Research Laboratory)에 의해 관리된다.[17]



〔그림 5〕 Flask 프레임워크 구조도

기본적인 Flask 구조는 그림 5와 같이 Object Managers와 Security Server로 구성된다. 다른 프레임워크와는 달리 클라이언트-서버 관계로 표현된 것이 특징이다. 하지만 기능적으로는 일반적인 보안 프레임워크와 비슷한 형태이다. Object Manager는 GFAC 프레임워크의 AEF와 같이 주체와 객체 사이에서의 보안 정책을 집행하는 부분이며 Security Server는 ADF와 같이 해당 주체와 객체 사이의 접근 허용 여부를 판단하는 부분이다.

Flask 시스템의 목표는 다양한 보안 정책을 수용할 수 있는 유연성(flexibility)과 투명성, 모듈성, 계층적 보안을 설계 원칙으로 하고 있다. 다양한 보안 정책을 수용하기 위해서는 고수준 함수가 사용하는 저

수준함수에 대한 적절한 접근 통제가 이루어져야 하며, 접근 권한은 보안 정책과 정확히 일치해야 한다. 정책은 정적이지 않아야 하며, 정책의 변화 또는 동적 정책을 위하여 기존에 허용된 접근 권한에 대한 폐지가 이루어져야 한다.

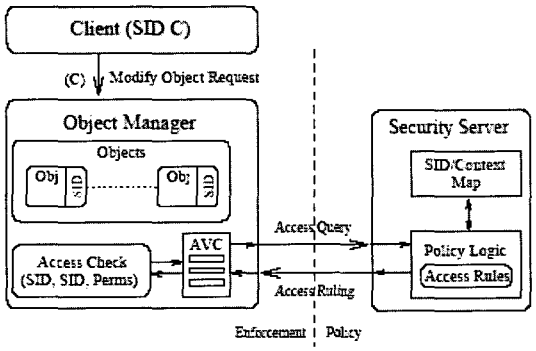
4.2.2 SELinux

SELinux 프레임워크의 보안 구조는 어떠한 운영체제에도 적용이 가능하게 보안 구조를 정의하고 있는 Flask 구조와 매우 흡사하다.

SELinux에는 전통적인 강제적 접근통제의 구현사항보다 더 진보된 강제적 접근통제 모델이 구현되어 있다. SELinux는 전통적인 강제적 접근통제의 문제점을 극복하기 위해 보안 정책 집단과 그것을 행하는 메커니즘을 분리시켜서 다양한 강제적 접근통제 정책을 가능하게 하는 좀 더 유연한 방법과 사용하기 편리한 방법을 제공한다.[17,24]

SELinux의 보안구조는 그림 6처럼 Security Server, Object Manager, AVC(Access Vector Cache)로 구성되어 있다. Security Server는 선택된 보안정책모델에 따른 보안정책을 결정한다. 여기서 보안 정책은 보안서버의 코드와 보안 데이터베이스를 말한다. 이 데이터베이스를 변경함으로서 MAC 보안 모델과 다른 RBAC과 같은 다른 보안 모델로도 변경이 가능한 유연성을 제공한다.[17,24]

Object Manager는 보안서버에게 주체(클라이언트)로부터 온 요청된 객체에 대한 허용여부를 질의하는 역할을 한다. 또한 객체에 대한 SID를 매핑으로 얻어서 보안서버에 보내고 Security Server로부터 보안 정책을 받는다. 그런 후에 새로운 객체에 대한 레이블을 얻고 객체에 대한 접근 결과를 얻게 된다.[17,24]



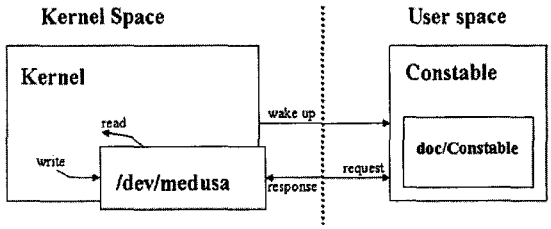
〔그림 6〕 SELinux 프레임워크 구조도

Object Manager 내부에 공유된 라이브러리 형태로 되어있는 AVC(Access vector cache)는 해당 객체의 SID에 대한 보안서버의 결과를 캐싱(caching)하는 기능을 한다. 내부에 발생하는 많은 오퍼레이션이 보안서버의 결정결과에 종속적이므로 AVC를 통해 보다 연산을 빠르게 참조할 수 있게 하였다. 그리고 이 AVC는 보안 정책 변경시 보안서버에 의해 새로 변경 되게 된다.[17,24]

4.3 Medusa DS9

Medusa DS9는 표준의 리눅스 보안 구조를 확장함으로써 리눅스 운영체제를 전체적으로 보안을 개선하는 하나의 패키지이다. 커널수준에서 사용자 공간 인가서버(authorization server)를 지원하고 모든 사용자 공간의 어플리케이션이라도 접근할 수 있도록 하였다.[15]

어떤 주체의 객체에 대한 접근을 실행하기에 앞서, 커널은 인가서버에 확인을 구한다. 인가서버는 접근요청을 허락하거나 차단한다. 이 방법은 거의 모든 보안구조(security architecture)에 사용될 수 있으며, 인가 서버가 적절히 구성될 때, 매우 섬세하게 접근권한들을 결정할 수 있고, 정당한 감사를 행할 수 있다.



〔그림 7〕 Medusa DS9 구조도

그림 7은 두 개로 이루어진 Medusa의 구조도이다. 커널 영역과 사용자 영역의 Constable 사이의 통신은 /dev/medusa 특별한 장치를 사용한다. 커널이 인가에 대한 확인을 필요로 할 때, 이 장치에 자료를 쓰고 현재의 프로세스를 수면상태(sleep)로 만들고 Constable을 깨운다. 깨어난 Constable은 /dev/medusa로 부터 자료를 읽고, doc/Constable에서 논의되어 결정되는 하나의 응답을 선택하여 커널에 되돌려 보내고 수면상태가 된다. 커널은 자료를 얻고, 수면상태의 프로세스를 깨우고 접근요청에 대한 결과를 결정한다.

4.4 DTE

DTE(Domain and Type Enforcement)는 Boebert, Kain에 의해 제안된 테이블기반(table-oriented) 접근제어 메커니즘으로 TE(Type Enforcement)의 더욱 강화된 형태이다. 많은 접근제어 스키마와 같이 TE는 능동적인 엔티티들인 주체들과 수동적인 엔티티들인 객체들의 집합(collection)으로 시스템에 접근하고 있다.[12]

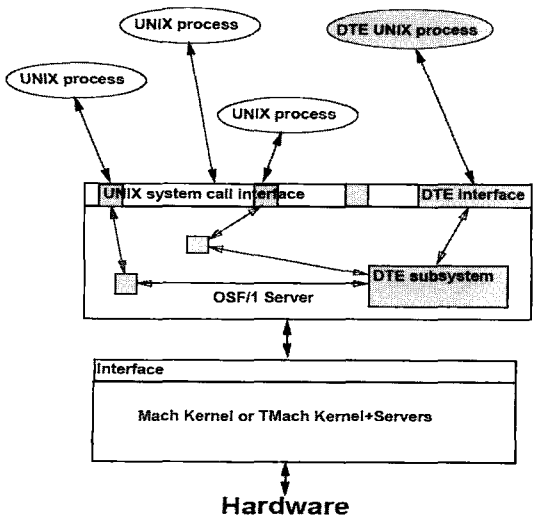
DTE는 사용자, 파일, 네트워크 사이에서 접근을 조정하는 것을 목적으로 개발되었다.

그림 8은 DTE 시스템의 프로토타입 구조도이다. 일반적인 커널 시스템 콜은 DTE interface를 통하여 보안 서비스를 요청한다. 커널 시스템 콜 이외에도 DTE는 DTE-aware 어플리케이션을 위하여 20개의 새로운 시스템 콜을 제공한다. 새롭게 추가된 DTE 시스템 콜은 시스템 보안 설정 및 모니터링 등을 제공하고 이외에도 보안과 관련된 기능을 제공한다.[12]

4.5 LSM

LSM(Linux Security Module)은 2001년 NSA에서 SELinux를 발표했을 때, 이를 리누즈 토발즈가 capabilities code migration 기능을 포함하여 제시한 프레임워크를 구현한 것이다.

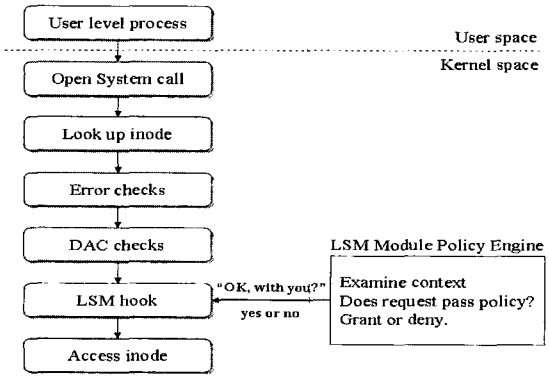
LSM 매커니즘은 서로 다른 접근제어개념이나 보안 정책들을 커널에서 적재 가능한 모듈로써 구현하는데 편의성을 제공하기 위한 것이다. 하지만, 실제로 특정한 모듈을 의미하는 말이 아니며, 어떤 보안 기능도 제공하지 않는다. 단지 보안 모듈이 지원하는 것은 기초구조만 제공한다. LSM 인터페이스의 기초적인 추상적 개념은 내부의 커널 객체들로의 접근을 조정하는 것이다.



〔그림 8〕 DTE 시스템 구조도

기존에 커널에서 접근제어를 구현한 프로젝트들의 대부분이 커널 패치형태로 구현하거나, LKM(Loadable Kernel Module)형식의 개발이 대부분이었다. 이러한 방식의 접근들은 서로 다른 접근제어를 두 가지 이상 적용하거나, 두 가지 이상의 보안정책을 적용할 필요가 있는 시스템에서 적용하기에는 문제점이 많았다.[11]

LSM을 이용하는 기법은 커널의 함수들과 그 함수들 내부에서 접근제어에 해당하는 연산을 담당할 후킹함수(hooking function)을 제작한 후 그 후킹함수를 더미함수의 자리에 등록하는 방식으로 구현한다.



〔그림 9〕 LSM 구조도

그림 9는 LSM의 구조도이다. 기존의 시스템 콜의 루틴에 LSM hook 부분이 추가되어 있는 구조를 나타내고 있다. 앞에서 설명한 것처럼 이렇게 추가된 hook 부분에 아무런 일도 하지 않는 더미함수(dummy function)가 연결되게 된다. LSM을 이용한 보안운영체제 구현방식은 더미함수들을 보안관리자가 의도하는 정책을 적용할 수 있는 함수로 대체하여 구현한다.

V. 결 론

본 고에서는 보안운영체제의 개념과 구현방법을 기술하고 보안운영체제의 접근동향 및 평가동향에 대하여 살펴보았다. 또한, DAC, MAC, RBAC과 같은 접근통제 모델과, 현재까지 연구된 통합 접근통제 프레임워크에 대하여 살펴보았다.

정보보호기술이 점차로 수동형에서 능동형으로 진화하고 있으며, 보안운영체제가 정보보호기술의 핵심으로 주목받고 있다. 운영체제 기술 발전의 흐름에 따라 보안운영체제는 기존의 통합 커널 방식에서 마이크로 커널 방식으로 바뀌고 있으며, 사용자들의 폭 넓은 요구사항과 강력한 시스템 보안을 제공하기 위해 다양한 접근통제를 지원할 수 있는 접근통제 프레임워크에 대한 연구가 진행되었다.

지금까지 연구된 SELinux, RSBAC과 같은 보안운영체제는 유연한 접근통제정책과 섬세한 보안 서비스를 지원하고, 사용자가 쉽게 사용할 수 있도록 개발되어 왔다. 향후 보안운영체제는 식별인증, 접근통제, 침입방지, 자체보호, 보안감사, 보안관리 등의 다양한 보안기능을 고려해서 연구되어야 한다. 특히, 유비쿼터스 환경에 적합한 임베디드 보안운영체제에 대한 연구는 현재 활발히 이루어지고 있다.

참 고 문 헌

[1] Amon O., Simone F.H., "The 'Rule Set Based Access Control' (RSBAC) Framework for Linux," *Karlstad University Studies*, 2001.

[2] Baldwin, R.W., "Naming and Grouping Privileges to Simplify Security Management in Large Databases," *IEEE Symposium on Computer Security and Privacy*, 1990.

[3] Bell, D.E., and La Padular, L.J., *Secure Computer Systems: mathematical foundations. Technical Report M74-244*, MITRE Corp., vol.1-2, 1974.[10] vol.1-2, 1974.

[4] Bell, D.E., and La Padular, L.J., *Secure Computer Systems: unified exposition and Multics interpretation*, MITRE Corp., 1975.

[5] Biba, K.J., *Integrity Considerations for Secure Computer Systems, MTR-3153*, MITRE Corp., 1977.

[6] Biba, K.J., *Integrity Considerations for Secure Computer Systems*, U.S. Air Force Electronic Systems Division.

[7] Castano, S., et al., *Database Security*, Addition-Wesley, pp. 18-34, 1994.

[8] D. Gollmann, "Computer Security," *John Wiley & SONS*, 1999.

[9] Int'l Standards Organization(ISO), "Working Draft on Access Control Framework," *ISO/IEC JTC 1/SC 21 N5045*, 1990.

[10] ITU-T SG/7 & Working Parties, "Final text for recommendation X.812 Information Technology-Open Systems interconnection Security framework for open systems: Access control framework," 1995.

[11] James M., Stephen S., Greg K.H., "Linux Security Modules: General Security Support for the Linux Kernel", *USENIX Security Symposium*," 2002.

[12] Lee B., Daniel F.S., David L.S., Kenneth M. W., and Sheila A.H., "A Domain and Type Enforcement UNIX Prototype," *Fifth USENIX UNIX Security Symposium*, 1995.

[13] L.LaPadula, "Rule-Set Modeling of Trusted Computer System", *Essay 9 in: M.Abrams, S.Jajodia, H. Podell, "Information Security - An integrated Collection of Essays," IEEE Computer Society Press*, 1995.

[14] Magdalena B., Hari B., Jon S., Mike S., "The Medusa Distributed Stream-Processing System," MIT Computer Science and Artificial Intelligence Lab, 2003.

[15] Medusa DS9, http://medusa.fornax.sk

[16] Sandhu, R.S., "Role Hierarchies and Constraints for Lattice Based Access Controls," *Processing of the 4th European Symposium on Research in Computer Security*, Sep. 1996.

[17] SELinux, http://www.nsa.gov/selinux

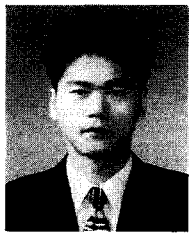
[18] U.S. Department of Defense, *Department of Defense Trusted Computer System Evaluation Criteria*, DOD 5200. 28-STD, National Computer Security Center, Dec. 1985.

[19] 김정녀, 손승원, 이철훈, "안전한 운영체제 접근제어 정책에 대한 보안성 및 성능 시험", 정보처리학회논문지 D 제10-D권 제5호(2003.8).

[20] 김정녀, 정교일, 이철훈, "리눅스 시스템의 버퍼

오버플로우 공격 대응 기법", 정보처리학회논문지 A 제8-A권 제4호(2001. 12).
[21] 손형길, 박태규, 이금석, "리눅스 운영체제 기반의 보안 커널 구현", 정보처리학회논문지 C 제10-C권 제2호(2003.4)
[22] 신욱, 이동익, 김형천, 강정민, 이진석, "보안운영체제를 위한 확장된 역할기반 접근통제 기법", 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호(2003. 11).
[23] 한국정보보호센터, "정보보호시스템 평가·인증 가이드", 2000. 12.
[24] 홍철호, "오픈 소스 보안운영체제의 성능평가에 대한 연구", 한국정보보호학회, 2002.

〈著 者 紹 介〉

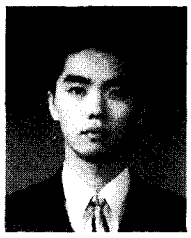


김 정 순 (Jung-sun Kim)

1998년 2월 : 전남대학교 전산학과 졸업
2000년 2월 : 전남대학교 전산학과 석사
2000년 3월~현재 : 전남대학교 전산학과 박사과정
〈관심분야〉 정보보호, 보안운영체제

이 재 서 (Jae-seo Lee)

2004년 2월 : 전남대학교 컴퓨터정보학부 졸업
2004년 3월~현재 : 전남대학교 전산학과 석사과정
〈관심분야〉 침입탐지시스템, 보안운영체제

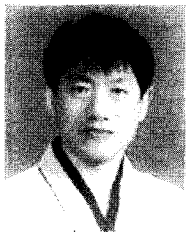


이 승 용 (Seung-yong Lee)

1995년 2월 : 전남대학교 전산학과 졸업
1997년 2월 : 전남대학교 전산통계학과 석사
2004년 8월 : 전남대학교 정보보호협동과정 박사
2005년 3월~현재 : 전남대학교 객원교수
〈관심분야〉 보안운영체제, 유비쿼터스 컴퓨팅 보안

김 민 수 (Minsoo Kim)

1993년 2월 : 전남대학교 전산통계학과 졸업
1995년 2월 : 전남대학교 전산통계학과 석사
2000년 2월 : 전남대학교 전산통계학과 박사
2005년 3월~현재 : 목포대학교 정보보호학과 교수
〈관심분야〉 침입탐지시스템, 보안운영체제, 데이터마이닝



노 봉 남 (Bong-nam Noh)

1978년 2월 : 전남대학교 수학교육과 졸업
1982년 2월 : 한국과학기술원 전산학과 석사
1994년 2월 : 전북대학교 전산통계학과 박사
1983년~현재 : 전남대학교 전자컴퓨터정보통신공학부 교수
〈관심분야〉 컴퓨터 네트워크 보안, 사이버 사회와 윤리