안전한 운영체제 접근제어 정책에 대한 보안성 및 성능 시험

김 정 녀[†] · 손 승 원[††] · 이 철 훈[†††]

요 약

SecuROS(Secure & Reliable Operating System) 시스템은 FreeBSD 4.3 운영체제 커널에 접근 제어, 사용자 인증, 감사 추적, 암호화 파일 시스템, 신뢰 채널 등의 보안 기능을 추가 구현하여 시스템에 발생 가능한 해킹을 방지하고 차단하는 시스템을 말한다. 본 논문에서는 SecuROS의 핵심 기술 중에 하나인 접근제어 기법을 기술하고, 해당 접근제어 정책인 DAC, MAC, RBAC의 구현 내용을 소개하며, 접근제어 정책의 적용에 따른 보안성과 성능 시험을 위한 도구 및 방법을 나타낸다. 기존의 운영체제 환경과 새로운 접근제어 정책을 적용한 환경 사이의 보안성 및 성능의 상관 관계를 기술한다.

Test on the Security and Performance on the Basis of the Access Control Policy Implemented by Secure OS

Jeong-Nyeo Kim[†] · Sung-Won Sohn[††] · Cheol-Hoon Lee[†††]

ABSTRACT

SecuROS(Secure & Reliable Operating System) prevents and blocks possible system cracking by implementing additional security functions in FreeBSD 4.3 operating system (OS) kernel, including access control, user authentication, audit trail, encryption file system and trusted channel. This paper describes access control technique, which is one of core technologies of SecuROS, introduces the implementations of DAC, MAC and RBAC, all of which are corresponding access control policies, and show security and results of performance measurement on the basis of application of access control policies. Finally, security and performance between conventional OS environment and environment adopting access control policy is described.

키워드 : 보안 운영체제(Secure Operating System), 접근 제어(Access Control), 보안성(Security), 성능(Performance)

1. 서 론

인터넷과 같은 네트워크 환경에서 유닉스가 가지는 개방성은 중요한 특징이지만, 컴퓨터 시스템내의 정보보호를 향상시키기 위한 기법은 현재 표준 유닉스에서는 매우 부족한 실정이다. 이에, 기존 유닉스 시스템의 취약점을 보완하는 패치 버전이나 업그레이드를 통한 임시 방편적인 방법보다는 원천적으로 운영체제 자체의 보안성을 강화한 안전한 운영체제의 필요성이 대두되고 있다. 또한 최근 들어 버퍼 오버플로우등과 같은 운영체제 자체의 보안상의 결함을 이용한 시스템 해킹이 늘고 있어, 비 인가된 사용자가 해당 시스템에 침입하여서 시스템내의 중요한 정보를 가로채거나 중요한 데이터가 들어 있는 파일들을 변경하는 등의 해킹 사고가 잇따라 발생하고 있다.

기존 유닉스 시스템의 취약점을 보완하는 패치 버전이나 업그레이드를 통한 임시 방편적인 방법보다는 원천적으로 새로운 안전한 운영체제 개발과 같은 근본적인 해결 방법이 바람직하다[2]. 또한 요즈음 들어서는 공개 소프트웨어 개념에 의해 리눅스를 비롯한 FreeBSD, OpenBSD 등 많은 운영체제 들이 공개되는 추세에 있어서 더 더욱 운영체제의 보안 결함을 이용하는 시스템 해킹 수법 들이 늘고 있다. 특히, 산업 사회를 거쳐 고도의 정보화 사회로 진입하면서, 고도의 각종 통신 수단이나 국가 기반 구축을 위한 시스템들이 더욱더 위험에 처해 있다. 이러한 환경에서 응용 프로그램 수준의 보안으로는 정보시스템의 완벽한 보안이 될 수 없음으로 운영체제 자체의 결함을 해결하고 시스템 차원의 정보보호 기능을 제공하는 안전한 운영체제 기술이 필요함을 인식하여야 할 것이다.

본 고에서는 이러한 운영체제 상에 내재된 보안상의 결함으로 인하여 발생할 소지가 있는 각종 해킹으로부터 시스템을 보호하기 위하여, 기존의 운영체제 내에 보안 기능을 추가한 안전한 운영체제 개념을 소개하고자 한다[1]. 안전한 운영체제는 시스템 사용자에 대한 다중 수준의 식별 및 인증, 강제적 접근 통제, 임의적 접근 통제, 역할 기반 접근 통제의 최소 권한 분리 기능 등의 보안 기능 요소들을 갖추어야 한다[4, 10]. 더 나아가서는 사용자의 시스템 사용 현황을 파악하기 위한 감사 추적 기능이나 시스템내

† 정 회 원 : 한국전자통신연구원 선임연구원
†† 정 회 원 : 한국전자통신연구원 책임연구원
††† 정 회 원 : 충남대학교 컴퓨터공학과 교수
논문접수 : 2003년 5월 21일, 심사완료 : 2003년 8월 13일

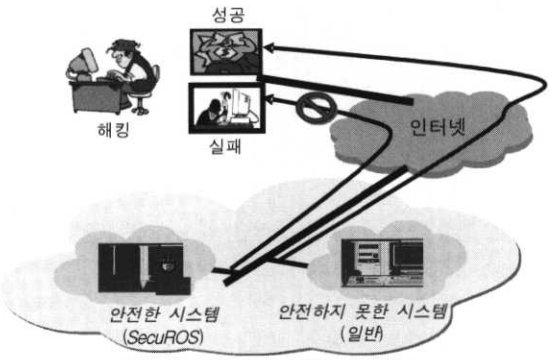
의 불법 정보 유출을 막을 수 있는 데이터의 안전한 저장 기능이나, 안전한 전송 기능까지 확장될 수 있다.

본 논문의 구성은 2장에서 안전한 운영체제의 개념과 필요성을 살펴보고, 본 연구실에서 개발한 SecuROS의 구조, 기능 그리고 현재 구현 정도를 소개한다. 3장에서 그 중 핵심 기술인 정책 기반의 접근제어 기술의 설계 및 구현 내용을 기술한다. 4장에서 안전한 운영체제의 보안성을 시험한 것으로 임의의 파일을 DAC, MAC, RBAC에 의해 비인가된 사용자의 접근을 막을 수 있는 SecuROS의 보안성을 분석하며, 5장에서는 보안성을 제공하면서 접근제어 기법에 따른 성능 시험의 결과를 소개한다. 마지막으로 앞으로 더 해야 할 연구의 방향을 제시한다.

2. 안전한 운영체제

2.1 안전한 운영체제 개념

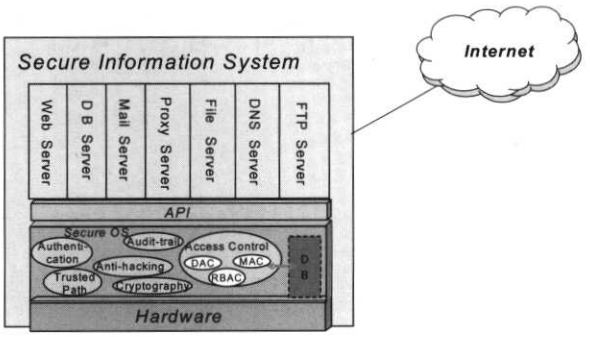
시스템에 대한 보안은 기본적으로 구조를 변경하지 않고 여러 가지 방법으로 개선될 수 있으나 아주 민감한 정보를 보호하고자 한다면, 강력한 개발 전략과 특별한 시스템 구조가 요구된다. 안전한 운영체제는 기존의 운영체제 내에 내재되어 있는 문제점을 해결하기 위하여 운영체제를 설계하는 기술로 운영체제 내에 보안 기능을 추가하여 운영체제의 결함을 이용한 시스템 해킹을 방지하기 위한 것이다. 안전한 운영체제는 개념적으로 (그림 1)과 같이 나타낸다.



(그림 1) 안전한 운영체제 개념도

2.2 안전한 운영체제 구조 및 기능

위와 같은 보안 운영체제의 기본 기능에 따라 보안 운영체제가 가져야 할 운영체제 구조를 기능 별로 구분한 것은 다음 (그림 2)와 같다. 보안 운영체제가 실행되는 정보시스템은 크게 인터넷 서버와 데이터 운용 서버 용도로 사용될 수 있다. 인터넷 서버의 경우는 웹 서버, FTP 서버, DNS 서버, Proxy 서버, Mail 서버 등을 들 수 있고, 데이터 운용 서버의 경우에는 DB 서버나 파일 서버로 쓰이는 경우이다. 그 이외에도 군이나 관공서 등에서 국가용 기능 시스템으로 활용이 가능하며, 다음과 같은 기능들을 제공한다.



(그림 2) 안전한 운영체제 구조도

2.2.1 사용자 인증

다중 수준 보안 정책의 사용자 인증 기능을 제공한다. 기존의 신분 기반 사용자 인증은 ID와 패스워드를 가지고 인증하므로 트로이 목마 취약성을 가지고 있으므로, 사용자가 갖는 등급과 범주, 역할 그리고 스마트카드 등에 의한 사용자 신분 확인 기능을 추가한다. 이는 강제적인 접근제어의 기준이 되는 보안 라벨(Security Label)에 의한 신분 확인과 역할 기반의 접근제어에 활용될 사용자 직무에 의하여 신분을 확인할 수 있다. 이에 추가로 스마트카드에 의하여 인증을 하도록 확장하였으며, 스마트카드 인증의 경우는 각 사용자를 식별할 수 있는 키를 저장한 스마트카드를 사용하여 신분을 확인하도록 되어 있다.

2.2.2 감사 추적

사용자의 정보 및 상태를 로그에 기록하고 분석할 수 있도록 하는 일종의 시스템 모니터링 기능이라 할 수 있다. 예를 들자면 사용자 인증에 실패한 경우에는 사용자 인증 정보, 날짜, 시간, 성공 여부, 시도 횟수 등을 기록하도록 한다. 또한 시스템 호출 수준의 사용자 처리 정보를 로그에 기록한다. 이는 setuid, setgid, link, read, write, exec 등과 같은 중요 시스템 호출을 사용하는 경우나 보안 관리자의 접근제어 속성을 변경하는 시스템 호출 처리 정보도 로그에 기록한다. 이는 객체에 대한 접근 여부와 시간 등을 기록하여 이상 상태가 발생하였을 때 추적 할 수 있도록 한다. 그 이외에도 로그의 기록을 분석하여 이상 상태를 알릴 수도 있다. 가장 중요한 것은 로그의 크기와 시스템의 성능을 위하여 Low, Medium, High 세 단계로 나누어 로그 기록이나 감사의 수준을 사용자가 정할 수 있도록 한다.

2.2.3 접근 제어

접근 제어 기능은 시스템 내의 자원(예를 들자면 파일, 파일 시스템, 장치, 메모리 등)에 대한 허가되지 않은 접근을 통제하여, 불법적인 자원의 사용, 노출, 수정, 파괴 등 불법적인 실행을 막는 것을 말한다. 접근 제어 정책은 다음과 같다.

- 임의적 접근 제어(DAC)
  신분기반의 접근 제어 정책으로 주체나 또는 그들이 속해 있는 그룹들의 신분 즉 ID에 근거하여 객체에 대한

접근을 제한하는 기법으로 트로이 목마의 취약성을 가진다. 메커니즘으로는 액세스 제어 리스트(Access Control List), 권한 리스트(Capability List) 등이 있는데, SecuROS에 액세스 제어 리스트를 이용하여 구현하였다.

- 강제적 접근 제어(MAC)
  규칙 기반의 접근 제어 정책으로 보안 라벨이라고 하는 보안등급과 범주에 의한 강제적인 접근 제어 방식으로 계급 체계가 있는 군 또는 공공기관에서 사용될 수 있다. BLP(Bell & LaPadula) 접근 제어 모델[3, 5]적용하여, 해당 등급의 자료만 접근 하도록 하며, 상위 등급의 자료는 읽기를 막고(No Read Up), 하위 등급의 자료는 쓰기를 금지하는(No Write Down) 형태의 등급별 접근 제어와 해당되는 부서와 같은 범주에 따라 접근을 제어하는 방식이다.

- 직무기반 접근 제어(RBAC)
  DAC과 MAC 혼합 형태의 접근 제어 정책으로 DAC과 MAC 방식의 단점을 해결하기 위해 직무 또는 역할 기반의 접근 제어 방식으로 상업적인 환경에 적합하다. 자원에 대한 사용자의 접근을 개별적인 신분이 아니라 조직 내에서의 개인 직무에 따라 결정하는 것으로 이는 보안 관리의 유연성 및 효율성을 제공하는 장점을 갖는다.

2.2.4 암 호

암호 기능은 정보의 비밀성을 제공하기 위하여 사용되는 것으로 암호화와 복호화에 쓰이는 키의 상이 여부에 따라 암호 방식도 나뉘어 진다. SecuROS에서는 DES, Blowfish 등과 같이 널리 사용되는 암호화 방식을 이용한다.

SecuROS에서 암호 기술은 인증 과정이나, 데이터를 보호하기 위하여 사용될 수 있으며, 그 이외에도 하드디스크를 가로채 가도 파일을 읽을 수 없도록 하는 디스크에 암호화해서 저장하는 암호화 파일 시스템용 암호화나 네트워크상의 패킷 가로채기에 의한 정보 유출을 막기 위한 데이터 전송을 위한 암호화에 사용된다.

2.2.5 암호화 파일 시스템

시스템내의 백업 관리자와 같은 경우 시스템내의 모든 객체들을 읽을 수 있어야 백업을 할 수 있을 것이나, 시스템내의 중요 데이터나 백업 관리자가 보면 안 되는 중요한 파일들도 있을 것이다. 이를 위하여 백업 관리자는 시스템내의 모든 객체를 읽기는 가능하나 쓰기는 할 수 없도록 하며 인가되지 않은 사용자 이면 파일이 암호화 된 채로 보이도록 하는 암호화 파일 시스템 기능이 필요하다. 그 이외에도 하드디스크 분실시에도 해당되는 데이터가 암호화되어 저장되어 있으므로 암호키 값이 없는 경우에는 데이터를 읽을 수 없으므로 하드디스크 분실시에도 정보 유출이 되지 않는다는 장점이 있다.

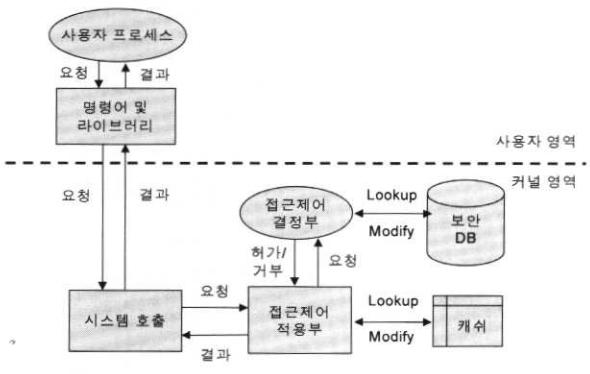
2.2.6 신뢰 경로

신뢰 경로 기술은 크게 두가지로 나누어 볼 수 있다. 사용자와 시스템간의 신뢰 경로 기술로 사용자가 시스템에 로그인할 때 또는 특정 사용자가 패스워드를 변경할 때 사용자가 입력하는 패스워드 등을 가로채기 할 수 있다. 이때는 신뢰할 수 있는 경로를 제공하여야 하므로 본 시스템에서는 사용자에게 로그인 역할을 주어 로그인 역할이 아닌 사용자는 접근을 허가 하지 않는 형태로 신뢰 경로 기술을 제공한다. 둘째는 시스템과 시스템간의 신뢰 경로 기술로 이는 Telnet이나 Ftp 등과 네트워크를 이용하는 경우 로그인 시나 데이터 전송시에 데이터를 가로채기 할 수가 있다. 이때는 신뢰할 수 있는 신뢰 채널 기능을 제공하여야 하므로 암호화하여 전송하고 복호화하여 받아 들일 수 있는 SecuROS간의 암호화된 신뢰 채널 기능을 제공한다.

3. 접근제어 설계 및 구현

3.1 접근제어 설계 및 구현

SecuROS에서의 접근 제어 기능은 크게 세 가지로 구현하였다. 그 중 DAC과 MAC은 TCSEC 기준[9]을 만족하기 위하여 구현하였으며, RBAC은 상업적인 환경에 적합하므로 구현하였다.



(그림 3) 접근제어 적용/결정 구조도

사용자 응용 프로그램이 명령어 및 라이브러리를 통하여 커널에 접근 요청을 하면 커널내의 접근제어 적용부에서는 해당 접근 요청에 대하여 접근이 가능한지 여부를 (그림 3)과 같이 커널내의 접근제어 결정부에게 요청한다. 접근제어 결정부에서는 해당 프로세스가 해당 자원에 접근 권한이 있는지를 검사하는 기능을 수행한다. 이 때 DAC, MAC, RBAC에 대한 접근 검사를 수행하며 요청된 접근에 대한 허가가 있는지 검사하여 하나라도 거부되면 접근이 허락되지 않는다. 접근제어 결정부에서는 보안 관리 데이터베이스에 저장되어 있는 보안 정보를 가지고 MAC, DAC, RBAC 보안 정책에 근거하여 해당 사용자가 자원에 접근 권한이 있는지 검사하여 접근 가능 여부를 알려준다. 보안 관리 데이터베이스는 DAC, MAC, RBAC 관련한 보안 관리 정보를 저장

하며, 특정 디렉터리에 저장이 되어있어 보안 관리자 만이 접근할 수 있도록 하였다. 접근제어 적용부에서는 접근제어 결정부로부터 받은 접근 허가/거부 결과를 사용자 프로세스에게 반환하며, 빠른 처리를 위하여 보안 관리 DB의 내용을 캐쉬에 두어 성능을 높였다

3.1.1 DAC

DAC은 신분기반의 접근 제어 정책으로 ID/Passwd 기반의 인증을 거쳐서 해당 ID의 사용자 접근을 객체의 접근제어 리스트의 permission에 의하여 통제하는 기법으로 기존의 유닉스 보다는 좀더 fine-grain하게 접근을 통제하였다.

```
• Linux permission
  rwxr--r--1john   research   89  Oct 6 22 : 30  demo
• Access Control List of file demo
  ACL_USER_OBJ : : rwx      == the owner of file
  ACL_USER : bob : rwx
  ACL_USER : fred : r-x
  ACL_GROUP_OBJ :: r--      == the group of file
  ACL_GROUP : manager : rw-
  ACL_GROUP : staff : r-x
  ACL_OTHER :: r            == the other
  ACL_MASK :: rwx
```

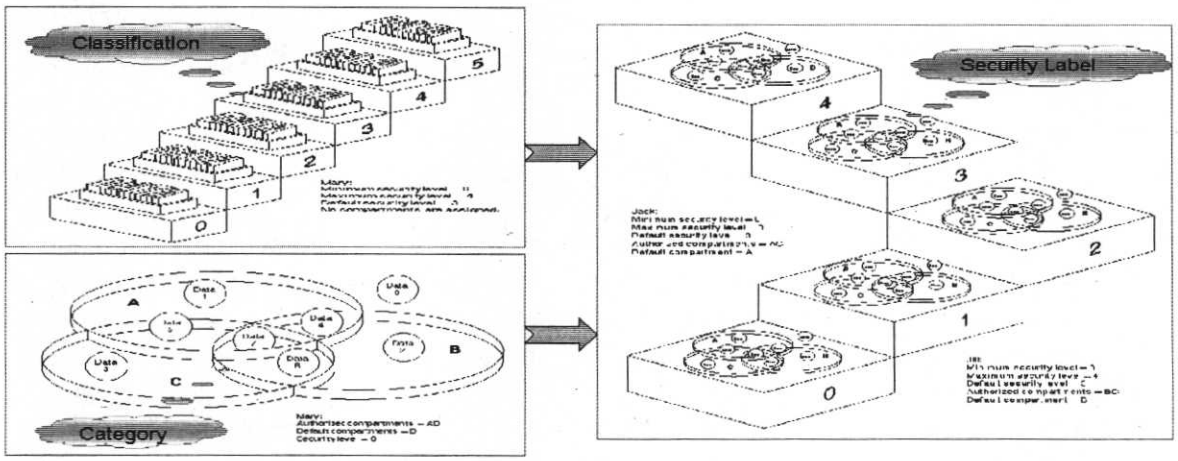
(그림 4) DAC 구현

DAC 메커니즘의 경우에는 기존의 리눅스에서 처럼 객체의 소유자, 소유 그룹, 다른 사용자들로 지정된 해당 객체의 접근제어 리스트를 좀 더 정밀하게 나누어 접근을 통제하는 것으로 소유자 이외의 개별적인 사용자별로, 소유 그룹 이외의 특정 그룹에 따라 정해진 접근을 통제하는 형태로 확장을 시켰다. 각 객체마다 그 객체에 접근할 수 있는 주체들의 접근 속성(읽기, 쓰기, 그리고 실행) 리스트를 가지며, 한 객체에 추가로 16개까지의 접근 속성을 부여할 수 있다. DAC을 검사할 때는 아이노드 번호에 의해 인덱싱된 ACL 정보를 찾아 접근 권한을 확인한다. 이는 기본 IEEE POSIX P1003.1e, 2c인 Security Extension 표준 규격에 따라 정의된 DAC 기능을 제공한다. SecuROS에서 제공하는 DAC은 (그림 4)와 같다.

3.1.2 MAC 구현

MAC은 규칙을 기반으로 한 강제적인 접근제어 정책으로 시스템 내의 사용자 각각에 등급과 범주를 두어서 해당 등급이나 범주에 따라 접근을 통제하는 기법이다. BLP(Bell & LaPadula) 모델[3, 5]을 기반으로 커널내에 구현하였으며 시스템 내의 등급은 크게 5가지, 범주는 64개까지 구분하여 사용 가능하다.

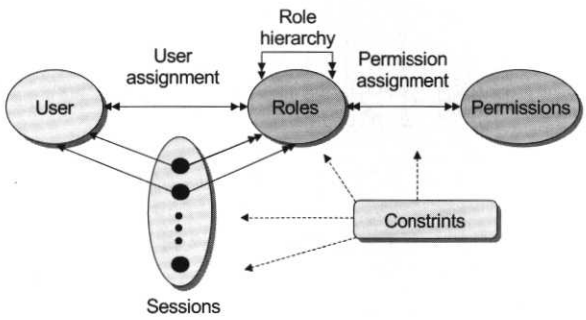
MAC 메커니즘의 경우는 실생활에 비유하면 비밀 취급 인가를 갖는 것을 구현하는 것으로 사용자의 등급을 나누어 등급별 트로이 목마 문제를 해결한 것을 말한다. 등급은 다음과 같이 사용자의 등급에 따라 동일한 등급의 주체는 동일한 등급의 객체를 읽기/쓰기 할 수 있도록 하는 것을 기본 원칙으로 하여 해당 등급의 주체가 본인이 가진 등급 보다 높은 등급의 객체를 읽을 수 없고(No Read Up), 해당 등급의 주체가 본인이 가진 등급보다 낮은 등급의 객체를 쓸 수 없는(No Write Down) 규칙에 의한 접근을 통제하는 방식이다. 이는 BLP(Bell & Lapadula) 모델을 준수하는 것으로 계층적인 관계의 접근은 통제가 가능하나 동일한 등급 내의 다른 그룹간의 상호 관계에 의한 접근 통제가 필요하므로, 부서 또는 범주라 하여 해당되는 범주에 속한 주체만이 해당 객체를 접근할 수 있도록 하는 범주에 의하여 접근 통제까지 한다. 보안 데이터베이스에 있는 MAC 정보는 객체에 해당하는 아이노드와 등급, 범주로 구성되어 있다. 해당 프로세스가 접근하려는 객체의 아이노드 정보를 이용하여 MAC 정보를 찾아낸 후에 해당 프로세서의 등급과 범주 내용을 비교하여 접근 권한 여부를 확인한다. SecuROS에서 제공하는 MAC은 (그림 5)와 같다.



(그림 5) MAC 구현

3.1.3 RBAC 구현

직무 기반의 접근 제어 정책인 RBAC은 상업적인 환경에서 가장 좋은 접근제어 정책으로 NIST 표준[4]인 RBAC 모델 [10]을 기반으로 커널 수준에서 구현하였다. RBAC 메커니즘의 경우는 위의 DAC과 MAC의 혼합된 형태로 상업적인 환경에 가장 적합한 접근제어 방식으로 해당 주체의 권한을 최소화한 최소 권한(Minimum Privilege) 분리 원칙으로 사용자의 역할이나 직무에 최소화된 권한을 부여하고 그 역할이나 직무에 따라 접근을 통제하는 방식을 말한다. RBAC에서는 임의의 역할에 대한 접근속성을 가지는 객체에 주체가 접근하기 위해서는 그 해당 역할에 주체가 멤버가 되어야만 접근할 수 있도록 하였다. 보안 관리자 역할과 백업관리자 역할 등을 두고 특정 보안 관리 명령어나 보안 데이터베이스 파일과 같은 중요 정보는 보안 관리자 역할에 속해 있는 사용자 만이 접근 할 수 있도록 하였다. 루트일지라도 보안 관리자 역할을 가지고 있지 않으면 접근 할 수 없다. 본 시스템에서는 현재 16개까지의 역할을 만들 수 있다. 역할 관리에 있어서 역할을 생성하고 삭제할 때 삭제되는 역할과 관계가 있는 모든 객체들의 퍼미션 정보를 모들 삭제해야 하는 관리상의 문제가 발생할 수 있다. 하지만 본 시스템에서는 해당 역할이 가지고 있는 생성 번호와 객체가 가지고 있는 역할의 생성 번호를 비교하는 방식으로 관련된 모든 객체들의 퍼미션 정보를 삭제하지 않고도 유효성 검사를 통해 관리가 가능하다. RBAC 기반의 접근 제어는 (그림 6)과 같다.



(그림 6) RBAC 구현

4. 접근제어에 의한 보안성 시험

4.1 MAC에 의한 보안성 시험

MAC 메커니즘의 구현으로 시스템내의 자원을 등급과 범주에 의해 보호 할 수 있다. 예를 들면 MAC에 의한 보안성 시험으로 /secure_data/secret_mac 파일에 등급을 세팅한다. 먼저 setfmac 명령어를 이용하여 파일의 등급 세팅한 후에 등급이 0인 staff 사용자가 읽을려고 하면 "permission denied"가 나오면서 접근이 거부 되는 것을 볼 수 있다. 3등급의 사용자인 manager가 0등급인 파일에 쓰기를 하면 쓰기가 거부되는 것을 볼 수 있다. 또한 범주에 따라 범주를 management 로 세팅한 후에 management가 세팅되지 않은 사용자가 읽을려고 하면 같은 상태로 접근이 거부 됨을 볼 수 있다.

4.2 RBAC에 의한 보안성 시험

RBAC 메커니즘의 구현으로 시스템내의 자원을 역할에 의해 보호할 수 있다. 시스템내의 /secure_data/secret_rbac 파일을 액세스 할 수 있는 역할을 세팅한다. 먼저 setfrole 명령어를 이용하여 파일을 액세스 할 수 있는 manager 역할을 지정한 후에 해당 역할이 아닌 staff으로 로그인 하여 파일을 액세스 하려고 하면 접근이 거부됨을 볼 수 있다. 예를 들면 웹 서버로 활용되는 경우를 들 수 있다. 웹 관리자 역할인 경우에만 웹 페이지 파일들을 읽거나 변경할 수 있도록 하고, 웹엑세스 역할인 경우에는 웹 서버내의 파일들을 읽기만 하도록 하는 때에 효율적이다.

5. SecuROS 기능 및 성능 분석

5.1 기능 분석

SecuROS에 구현된 접근제어의 기능은 상용 Secure OS인 SUN Microsystems사의 Trusted Solaris와 비교해 볼 수 있다. Trusted Solaris는 SUN Microsystems사에서 개발한 Secure OS로 비교한 버전은 2.5.1이다. 각 기능별 비교는 다음과 같다.

- 임의적 접근 제어(DAC) : Trusted Solaris의 경우에는 기존의 유닉스 Permission bit에 의한 ACL 기능만 제공하나, SecuROS는 기존의 유닉스 Permission bit를 확장한 좀더 Fine-grained 한 DAC 기능을 제공한다. 기존의 경우에는 해당 객체의 소유자, 그룹, 그 이외의 사용자로 접근제어를 할 수 있었다면, SecuROS에서는 소유자 이외의 다른 개별 사용자, 해당 그룹 이외의 다른 그룹별로 좀더 나누어서 접근제어가 가능하다는 것이다. 이는 기존의 DAC 보다 좀더 유연성을 제공하면서 보안성을 높일 수 있다는 장점이 있으며, TCSEC 규격과 IEEE POSIX P1003.1e, 2c 표준을 준수하여 표준화된 API를 제공한다.
- 강제적 접근 제어(MAC) : Trusted Solaris의 경우에는 로그인시에 정한 등급에 따라 접근을 통제하는 것으로 기본적으로 등급에 따른 접근제어만 가능하다. SecuROS에서는 로그인시에 입력한 등급과 범주에 따라 접근제어가 가능하다. 등급에 따른 접근제어는 사용자의 계층(수직)적인 분류에 따른 방법이라면 범주에 따른 접근제어는 수평적인 관계 내에서의 분류에 따른 방법이라 할 수 있다. 예를 들자면 동일한 등급이라 하더라도 소속이 다른 경우라면 이에 따른 접근 제어가 필요하다는 것이다. 이는 동일 등급의 사용자에게 유연성을 주면서 강화된 접근제어 방식을 제공한다. 또한 DAC과 마찬가지로 TCSEC 규격과

IEEE POSIX P1003.1e, 2c 표준을 준수하여 표준화된 API를 제공한다.

- 직무 기반 접근 제어(RBAC) : 직무 기반의 접근 제어는 Trusted Solaris의 경우에도 제공을 하고 있으나 이는 사용자 계정을 사용하여 직무와 매핑시킨 단순 직무 기반의 접근 제어 기법이다. 한 계정에 하나의 역할만 정해져 있어서 유연성이 부족하며, 직무의 계층도 없어서 직무 상하간의 계층적인 제어도 할 수 없다는 단점이 있다. SecuROS에서의 RBAC은 NIST의 표준을 준수하여[4] 커널내에 구현한 것으로 한 사용자가 16개까지의 역할을 가질 수 있고 로그인시에 사용자가 입력한 역할에 따라 접근제어가 이루어지므로 유연성을 제공한다. 또한 직무간의 계층적인 제어도 가능하다.

5.2 성능 분석

5.2.1 성능 분석 환경

성능 분석 도구는 두가지로 나누어서 하였다. 첫째는 유닉스 시스템의 성능상 병목 현상을 측정하여 주는 lmbench라는 성능 측정 도구를 사용하는 것으로 FreeBSD 환경에서 컴파일하여 사용한다. 시스템 지연이나 프로세서와 메모리, 네트워크간의 데이터 대역폭을 측정할 수 있는 마이크로 벤치마크 프로그램으로 나누어져 있다. 그 중 File System Latency와 Disk bandwidth를 측정한다. 둘째는 파일 입출력 지연시간 및 처리량 측정하는 프로그램을 작성하여 측정한다. 그 시나리오는 1K, 10K, 100K의 다른 파일에 대하여 파일 관련 작업(생성, 삭제, 열기, 닫기, 읽기)을 1번 수행하는데 걸리는 시간을 측정하는 것으로 해당 시스템 호출을 이용하여 작성하고 실행 전과 후의 시스템 시간을 측정하여 시간 차이를 구하는 방식으로 측정하였다. 이 두가지 성능 분석 도구를 이용하여 크게 기존의 FreeBSD 4.3 운영체제 환경[14], DAC만 적용한 환경, DAC과 MAC을 함께 적용한 경우, DAC과 RBAC을 함께 적용한 환경 등 네 가지 환경으로 나누어서 성능을 측정하였다.

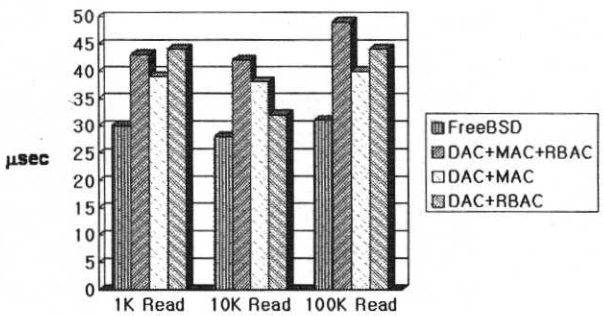
5.2.2 성능 측정

5.2.2.1 lmbench 도구

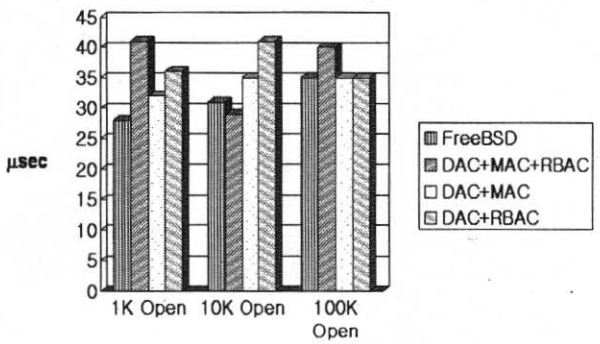
lmbench 도구를 사용하는 경우는 다음 <표 1>, <표 2>와 같이 두가지 경우로 성능이 측정이 되었다.

5.2.2.2 파일 입출력 지연시간 및 처리량 측정 도구

파일 입출력 지연시간 및 처리량을 측정하는 도구를 사용하는 경우는 (그림 7)과 (그림 8)과 같이 성능이 측정되었다.



(그림 7) 파일 읽기 처리율



(그림 8) 파일 열기 처리율

〈표 1〉 시스템 호출 시험

Test Env.	Null call	Null I/O	Stat	Open/Close	Select TCP	Sig inst	Sig hndl	Fork prok	Exec proc	Sh proc
FreeBSD	1.13 μsec	2.30 μsec	6.50 μsec	8.76 μsec	10.4 μsec	1.67 μsec	3.50 μsec	254 μsec	1052 μsec	2004 μsec
DAC + MAC	1.15 μsec	2.28 μsec	9.08 μsec	11.3 μsec	10.7 μsec	1.68 μsec	3.45 μsec	264 μsec	1101 μsec	2061 μsec
DAC + RBAC	1.14 μsec	2.28 μsec	18.5 μsec	22.6 μsec	10.7 μsec	1.67 μsec	3.43 μsec	265 μsec	1162 μsec	2150 μsec
DAC + MAC + RBAC	1.42 μsec	2.58 μsec	19.4 μsec	25.3 μsec	10.8 μsec	1.80 μsec	4.02 μsec	287 μsec	1209 μsec	2212 μsec

〈표 2〉 컨텍스트 스위치 시험

No. Programs	2	2	2	8	8	16	16
Data Seg Size	0K	16K	64K	16K	64K	16K	64K
FreeBSD	5.530 μsec	7.350 μsec	11.9 μsec	15.5 μsec	48.8 μsec	19.0 μsec	49.2 μsec
DAC + MAC	5.700 μsec	7.43 μsec	12.3 μsec	15.8 μsec	49.5 μsec	18.1 μsec	49.7 μsec
DAC + RBAC	4.810 μsec	6.440 μsec	11.7 μsec	15.2 μsec	48.5 μsec	17.5 μsec	48.9 μsec
DAC + MAC + RBAC	5.640 μsec	7.330 μsec	12.2 μsec	17.0 μsec	53.2 μsec	18.5 μsec	55.1 μsec

5.2.1 성능 분석 결과

lmbench 도구에 의한 성능 분석 결과를 보면 기존의 Free BSD 보다 성능은 저하되며, 모두 적용하는 것보다 DAC + MAC 적용이나 DAC + RBAC 적용이 성능 저하가 적은 것으로 나타났다. 또한 두 가지 경우가 각 경우에 따라 조금씩 우위를 보이는 면이 있는 것을 알 수 있다. 파일 입출력 지연시간 및 처리량을 측정하는 도구를 이용한 성능 분석 결과에 의하면 기존 FreeBSD의 경우 보다는 파일 읽기의 경우 평균 114.3%, 파일 열기의 경우 평균 114.6% 정도임으로 평균 114.5%의 처리율이 나타난다. DAC만 적용한 경우에는 DAC + MAC 적용 경우나, DAC + RBAC 적용 경우보다 성능은 좋으나 보안성 측면에서 떨어지며, DAC + MAC 적용 경우와 DAC + RBAC 적용의 경우는 보안성은 더 우수하나 DAC + MAC 적용의 경우가 DAC + RBAC 적용의 경우 보다 약 7% 정도 성능이 더 우수하다. MAC을 적용하는 경우가 성능면으로 좋다는 것을 알 수 있다.

6. 결론 및 향후 연구 방향

본 고에서는 안전한 운영체제의 개념과 안전한 운영체제의 필요성을 기술하고 FreeBSD 4.3 커널을 기반으로 한 안전한 운영체제의 기술을 소개하였다. 또한 안전한 운영체제 내의 접근제어 기법인 DAC, MAC, RBAC 등의 구현 내용을 바탕으로 보안성과 성능을 측정하였다. 본 결과는 네가지 측면에서 의미를 가질 수 있다. 첫째는 TCSEC과 POSIX 표준을 준수한 DAC, MAC 구현과 NIST 표준인 RBAC 모델을 커널내에 구현하였다는 데에 큰 의의를 둘 수 있다. 둘째로는 커널 수준의 접근 제어나 시스템 관리자인 루트의 권한을 분산 시킴으로써 시스템의 주요 파일이나 장치들을 보호할 수가 있으며, 무엇보다 보안성이 증명된 BLP 모델이나 RBAC 모델을 구현하여 보안성 측면에서 우수하다고 할 수 있다. 셋째로는 접근제어 등과 같은 보안이 구현된 안전한 운영체제의 성능을 측정할 수 있었다는 것이다. 접근제어 성격상 파일을 열기, 읽기, 쓰기 등과 같은 데서 성능 비교가 필요한데, 이를 위한 파일 입출력 지연시간 및 처리량을 측정하는 성능 측정 도구를 구현하였다는 것이다. 마지막으로 접근제어 기법의 조합에 따라 성능도 달라지므로, 보안의 필요에 따라 동적으로 접근제어 정책을 조합하여 사용할 수 있음을 알 수 있다. 또한 이러한 성능 분석 결과를 바탕으로 이러한 보안성과 성능을 만족할 수 있는 새로운 접근제어 메커니즘을 연구하여야 할 것이다.

최근 들어서는 버퍼 오버플로우 뿐만 아니라 분산 서비스 거부 공격 등과 같은 해킹 기법들에 의해 시스템을 halt 상태로 만들기도 하는데,이를 위하여 시스템 차원의 감사 추적 기능이나 시스템 모니터링 기능으로 탐지하고 차단할 수 있도록 연구하여야 할 것이다.

이에 덧붙여서 현재 안전한 운영체제 시스템은 범용화되어 사용되지 않기 때문에 그 안전성이나 성능을 평가하는 것 또한 쉽지 않다. 국내에는 아직 안전한 운영체제 시스템을 평가하는 기준이 마련되지 않은 상태이다. 국제 평가 기준을 따르면서도 국내 상황에 알맞은 기준이 마련되어 시스템의 안전성과 성능을 판별할 수 있어야 하며, 외국 제품이나 기술과 차별화 되는 기술 보유의 기틀이 마련되어야 하겠다.

참 고 문 헌

[1] J. G. Ko, J. N. Kim and K. I. Jeong, "Access Control for Secure FreeBSD Operating System," *Proc. of WISA 2001, The Second International Workshop on Information Security Applications*, 2001.

[2] Peter A. Loscocco, Wtephen D. Dmalley, Patric A. Muckelbauer, Ruth C. Taylor, S. Jeff Truner, John F. Farrel, "The Inevitablity of Failure : The Flawed Assumption of Security in Modern Computing Environments," National Security Agency, 1997.

[3] Bell, David Elliott and Leonard J. La Padula, "Secure computer system : Unified exposition and multics interpretation," MITRE Technical Report 2997, MITRE Corp, Bedford, MA, 1975.

[4] David F. Ferraiolo, Ravi Sandu and Serban Gavrila, "A Proposed Standard for Role-Based Access Control," ACM transaction on Information and System Security, Vol.4, No.3, pp.224-274, Aug., 2001, http://csrc.nist.gov/rbac/.

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

[6] UNICOS Multilevel Security (MLS) Features Users Guide, SG-21111 10.0, http://rcs21.urz.tu-dresden.de:80/ebt-bin/nph-dweb/dynaweb./@Generic_BookTextVie.

[7] http://www.hpcc.gov/pubs/blue97/nsa/secureos.html.

[8] http://www.cs.utah.edu/flux/fluke/html/linux.html.

[9] DOD 5200.28-STD, "Department of Defense Trusted Computer System Evaluation Criteria," December, 1985.

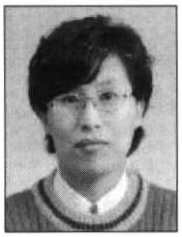
[10] D. Ferraolo and R, Kuhn, "Role-Based Access Control," Proceeding of the 15th National Computer Security Conference, 1992.

[11] R. Graubart, "Operating System Support for Trusted Applications," Proceedings of the 15th National Computer Security Conference, 1992.

[12] M. Harrison et al., "Protection in Operating Systems," Communications of ACM 19(8), August, 1976.

[13] Secure Computing Corporation, "Assurance in the Fluke Microkernel : Formal Security Policy Model," Technical report MD A904-97-C-3047 CDRL A003, March, 1998.

[14] FreeBSD 4.3-RELEASE Source Code.



김 정 녀

e-mail : jnkim@etri.re.kr

1987년 전남대학교 전산통계학과(학사)

1995년~1996년 Open Software Foundation Research Institute 공동 연구 파견(미국)

1998년~2000년 충남대학교 컴퓨터공학과 석사(공학석사)

1988년~현재 한국전자통신연구원 선임연구원(팀장)

2000년~현재 충남대학교 컴퓨터공학과 박사과정

관심분야 : 인터넷 정보보호, Secure OS, 네트워크 보안

손 승 원

e-mail : swsohn@etri.re.kr

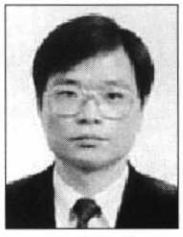
1984년 경북대학교 전자공학과(학사)

1994년 연세대학교 컴퓨터공학과 석사(공학석사)

1999년 충북대학교 컴퓨터공학과 박사(공학박사)

1991년~현재 한국전자통신연구원 책임연구원(부장)

관심분야 : 이동인터넷 보안, 정보보호, 네트워크 보안



이 철 훈

e-mail : chlee@ce.cnu.ac.kr

1983년 서울대학교 전자공학과(학사)

1983년~1986년 삼성전자 컴퓨터개발실 연구원

1988년 한국과학기술원 전기 및 전자공학과 석사(공학석사)

1992년 한국과학기술원 전기 및 전자공학과 박사(공학박사)

1992년~1994년 삼성전자 컴퓨터사업부 선임연구원

1994년~1995년 Univ. of Michigan 객원연구원

1995년~현재 충남대학교 컴퓨터공학과 부교수

관심분야 : 운영체제, 병렬처리, 결함허용 실시간 시스템 등