SecuROS[1]에서 개발된 사용자 및 프로그램 인터페이스

두 소 영[†]·고 종 국[†]·은 성 경[†]·김 정 녀[†]·공 은 배[††]

요 약

공개 운영체제인 Linux와 FreeBSD는 무료이면서 그 성능이 우수하여 많은 사람들이 사용하고 있다. 개방성은 중요한 특징이지만 공개된 커널의 원천코드는 자주 해커들에 의해서 공격당하는 문제점이 되기도 한다. 본 논문에서는 이러한 문제점을 해결하기에 가장 적합한 해결책인 보안운영체제 SecuROS에 대해서 설명하고, 보안운영체제의 활성화를 돕기 위해 필수적인 표준화된 사용자 및 프로그래머 인터페이스에 대해서 소개한다. 개발된 보안운영체제는 MAC과 ACL 접근통제를 사용하고 가장 일반적으로 사용되고 있는 POSIX 표준을 인터페이스 규격으로 따른다.

Secure User and Program Interface for SecuROS

So-Young Doo[†] · Jong-Gook Ko[†] · Sung-Kyong Un[†] ·
Jeong-Nyeo Kim[†] · Un-Bae Kong[††]

ABSTRACT

Many people use Linux and FreeBSD because it is freeware and excellent performance. The open source code is very important feature but it also has some problem which may be attacked by hackers frequently. This paper describes the SecuROS of secure operating system that is best solution to this problem and introduces user and programmer interface for active use of secure operating system. Developed secure operating system is composed of the access control method MAC and ACL and conforms to the POSIX which is universally used.

키워드 : 보안운영체제(Secure Operating System), 접근제어(Access control), 보안 명령어 및 라이브러리(Secure Utility and Library)

1. 서 론

공개 운영체제로 잘 알려진 Linux, FreeBSD 등은 무료 소프트웨어이면서도 그 성능이 우수하기 때문에 범용화되어 사용되고 있다. 개방성은 중요한 특징이지만 공개된 커널의 원천코드로 인하여 인터넷 서버, 파일서버, 메일서버 등으로 사용하는데 있어서 보안상의 문제점들이 발생되고 있다. 이러한 문제점 해결을 위한 방법으로 방화벽, 침입탐지 시스템, 보안운영체제[1] 시스템들이 제안되고 있다.

보안운영체제란 운영체제상에 내재된 보안상의 결함으로 인하여 발생 가능한 각종 해킹으로부터 시스템 보호를 위하여 기존의 운영체제 내에 보안기능을 추가한 운영체제를 말한다. 보안운영체제를 사용하여 얻을 수 있는 가장 큰 장점은 외부 공격자들 뿐만 아니라 내부 사용자들이 시스템 자원의 오용을 사전에 방지할 수 있다는 것이다.

SecuROS는 한국전자통신연구원 보안운영체제연구팀에서 지난 2년여동안 Linux와 FreeBSD를 대상으로 접근통제 기능을 추가하여 개발하고 있는 보안운영체제 시스템이다. 현재 Linux 기반의 보안운영체제 시스템은 MAC(Mandatory Access Control)과 ACL(Access Control List) 접근통제 기능을 추가하여 완료된 상태이다. FreeBSD를 기반으로 한 보안운영체제 시스템은 MAC, ACL, 그리고 RBAC(Role Based Access Control) 접근통제를 지원하도록 설계하였고 계속 진행중이다. 본 논문에서는 개발 완료된 Linux 기반의 보안운영체제 시스템에 대해 설명하고 이 시스템에서 제공되는 사용자 및 프로그램 인터페이스를 설명하고자 한다.

기존 유닉스(UNIX)계열 시스템에서 접근제어 단위는 소유자, 소유자의 그룹, 그외 사용자(Owner, Group, Other)로 구분하였고 그 범주에 따라 읽기,쓰기,실행(rwx)권한을 부여할 수 있었다. 또한, 슈퍼유저에게 모든 권한이 편중되어 있기 때문에 시스템을 악용하거나 파손하려는 사용자들은 시스템에 접근하여 무엇보다도 먼저 슈퍼유저가 되려고 시도하고 있다.

하지만, 개발된 보안운영체제 시스템은 ACL을 사용하여

---

1) SecuROS(SecuRe Operating System)은 한국전자통신연구원에서 개발한 보안운영체제

† 정 회 원 : 한국전자통신연구원 보안운영체제연구팀
†† 정 회 원 : 충남대학교 컴퓨터공학과 교수
논문접수 : 2001년 7월 10일, 심사완료 : 2001년 10월 17일

접근 단위를 소유자, 소유자의 그룹, 사용자, 그룹, 그외 사용자로 구분하여 보다 세분화된 접근 제어가 가능하도록 하였다. 또한,객체(파일, 프린터 등)와 주체(사용자, 프로세스 등)를 MAC에 해당하는 등급과 범주로 분류하고 객체에 할당된 등급과 범주에 해당하는 주체에 대해서만 접근을 허가한다. 제한된 범주내에서 정보를 사용하도록 함으로써 슈퍼유저라 할지라도 시스템의 모든 자원을 임의로 사용할 수 없도록 하였다.

보안운영체제에서 다양한 응용프로그램의 개발을 돕기 위해서는 응용프로그램 개발자들과 시스템 사용자들에게 공통된 인터페이스를 제공하는 것이 필수적이다. 개발된 시스템은 이미 개발되었거나 연구 개발중인 다른 보안운영체제 시스템에서도 가장 많이 적용하고 있는 POSIX 인터페이스를 지원하고 있다. POSIX 1003.1e[2]와 1003.2c[3] 중에서 MAC과 ACL에 해당하는 모든 인터페이스를 지원하여 사용자의 편리성과 프로그램의 이식성 및 확장성을 제공한다. 현재 POSIX 1003.1e와 1003.2c는 표준화 절차가 중단된 상태이다. 표준화 관여자에게 문의한 바로는 현재까지 정의된 이상의 내용은 제품 생산자 임의에 맡기는 것이 더 옳다고 판단했기 때문이라고 한다.

본 논문의 구성은 2장에서 현재까지 제안되거나 사용되는 대표적인 보안운영체제와 각각의 사용자 인터페이스에 대해서 설명하고, 3장에서 개발된 Linux 기반의 다단계 접근제어 보안운영체제의 동작과 구성에 대해서 설명하고, 4장에서 제공되는 사용자 인터페이스에 대해서 설명하고, 5장에서는 구현된 시스템에서 다단계 사용자인증의 동작 절차를 설명하고, 6장에서 결론과 향후 연구 방향에 대해 논의한다.

2. 관련 연구

1970년대부터 미국을 비롯한 선진국에서는 중요 정보 및 비밀 정보처리를 위한 컴퓨터 시스템의 필요성을 인식하여, 신뢰성을 평가할 수 있는 기준을 제정하고 보안운영체제를 연구하여 안전하고 신뢰성 있는 컴퓨터 시스템 개발에 많은 투자를 하고 있고, 1980년대 이후부터는 커널 내에 안전한 운영체제를 탑재한 컴퓨터 시스템을 보급하기 시작했다. 미국의 경우, 신뢰성 컴퓨터 평가 기준 TCSEC(Trusted Computer System Evaluation Criteria)[4] B1급 이상의 컴퓨터 시스템에서는 보안처리 모듈이 커널 내에 구현되고 있으며, B2급 이상의 평가를 받은 컴퓨터 시스템에 대해서는 해외로 수출을 금지하고 있다.

현재 개발된 보안운영체제로는 다음과 같은 것들이 있다. TCSEC D 등급은 보안이 거의 고려되지 않고 있는 PC나 Macintosh, C1 등급은 사용자가 서로 침범할 수 없도록 되어 있는 일반적인 유닉스 시스템, C2 등급은 보안과 관련된 정보를 로그로 남기는 등의 기능이 추가된 것으로 DEC의 OpenVMS, B1 등급은 객체에 보안등급 같은 것을 정의할 수 있도록 함으로써 낮은 등급의 사용자가 높은 등급의 객체에 접근할 수 없도록 하는 기능을 가진 것으로 IBM의 MVS/SP, HP의 HP-UX BLS, Unisys OS1100, Sun의 Trusted Solaris, Microsoft Windows NT, B2 등급 이상은 보안 정책이 시스템의 모든 객체에 대하여 각각의 접근 방법을 여러 가지로 정의할 수 있는 기능을 가지며 하드웨어에도 관련되어 있는 형태로 Trusted XENIX(B2)와 Trusted Information System의 TMach(B3)가 있다[5]. 그리고 Linux의 보안 패키지로 개발된 B1 등급의 Medusa[6], RSBAC[7], SELinux (Security Enhanced Linux)[8]가 대표적으로 알려져 있다. 국내에서 티에스온넷, 나일소프트, 시큐브등이 보안운영체제에 관한 연구를 진행하고 있다. 이 시스템들 중 현재 널리 알려진 대표적인 보안운영체제 시스템 중 Medusa, Trusted Solaris 8[9], SELinux의 시스템 동작과 사용자 인터페이스에 대해서 설명한다.

2.1 Medusa

Medusa는 Linux의 보안성을 강화하기 위한 하나의 패키지이다. Medusa는 보안관리자(constable), 커널패치(kernel patch), 디바이스 드라이버(device driver)로 구성되어 있다. 보안관리자는 사용자 혹은 프로세스가 실행하고자 하는 행위에 대하여 허가여부를 결정하는 결정자(Decider)역할을 한다. 커널 패치는 사용자 혹은 프로세스가 요청한 시스템 호출을 가로채어 보안관리자에게 허가여부를 묻는 집행자 역할을 한다. 보안관리자는 커널 내부에 존재하지 않고 사용자 프로세스로 존재하며, 커널과의 통신은 문자 디바이스(character device)를 이용한다.

〈표 1〉 Medusa 환경설정파일 변수

변 수	내 용	변 수	내 용
vs	객체의 가상공간비트맵	target_vs	대상 객체의 가상공간 비트맵
vss	프로세스의 가상공간 비트맵	target_vss	대상 프로세스의 가상공간 비트맵
vsr	프로세스가 읽을 수 있는 가상공간 비트맵	target_vsr	대상 프로세스가 읽을 수 있는 가상공간 비트맵
vsw	프로세스가 쓸 수 있는 가상공간 비트맵	target_vsw	대상 프로세스가 쓸 수 있는 가상공간 비트맵
proact	보안관리자로부터 확인이 필요한 동작	target_proact	대상 프로세스가 보안관리자로 부터 확인이 필요한 동작
fsact	보안관리자로부터 허가가 필요한 파일시스템 내용	target_fsact	대상프로세스가 보안관리자로부터 허가가 필요한 파일시스템 내용

Medusa는 파일 시스템에 대한 접근제어로 가상파일시스템(Virtual File System)의 모든 파일에 가상공간비트맵(Virtual space bitmap)을 할당한다. 비트맵은 32bit로 이루어져 있고, 각 비트(bit)는 특정 가상공간을 지정한다. 시스템 호출(fork, exec, signal, execute set uid program, setuid, capability) 검사등의 시스템 호출을 통제하기 위하여 오퍼레이션 비트맵을 할당한다. 시스템 호출을 요청한 프로세스와 대상 파일(또는 다른 프로세스)의 가상공간비트맵에 공통부분이 있는지 확인하고 공통부분이 있는 경우에만 동작이 허가된다. 보안관리자는 환경설정파일에 기록된 내용을 바탕으로 허가여부를 결정하게 되는데 환경설정파일에 기술되는 변수 몇 가지를 정리하면 <표 1>과 같다.

2.2 Trusted Solaris 8

SUN에서는 지난 2000년말 Trusted Solaris 8을 내놓았다. Trusted Solaris 8은 TCSEC 기준 B1+급을 만족하고 있다. ACL, MAC, RBAC을 지원한다. Trusted Solaris 8에서는 모든 객체와 프로세스의 보안 속성값을 CMW(Compartmented Mode Workstation) 레이블과 변경 조작이 가능한 정보 레이블인 비밀등급(Sensitivity Label)로 정의한다.

Trusted Solaris 8은 프로세스가 시스템 보안 규칙에 의해서 금지되는 동작을 수행할 수 있도록 하는 특권 처리, 동작을 수행하기 이전에 사용자 인증을 통해 권한이 있는 사용자의 경우에만 사용을 허가하는 처리, CMW 레이블을 사용하여 데이터를 분류하고 접근 관리하는 처리 그리고, SLD(Single Level Directory)와 하나 이상으로 구성된 MLD(Multi Level Directory)를 조작하는 사용자 인터페이스를 제공한다. 몇 가지 시스템 호출과 그 동작 내용을 정리하면 <표 2>와 같다.

2.3 SELinux

SELinux는 운영체제의 기밀성과 무결성을 강화하여 개발되었다. NSA(National Security Agency)와 SCC(Secure Computing Corporation)가 개발한 것으로 강력하고 유동적인 유형에 따른 실행(Type Enforcement)을 바탕으로 한 MAC, RBAC를 지원한다. 이 구조는 Mach와 Fluk으로부터 발전된 것으로 FLASK(Flux Advanced Security Kernel)라고 불린다. 이 구조를 Linux에 통합하여 구현한 것이다. Linux 커널 2.2.2와 2.2.17에 구현되었고 보안서버를 커널내에 서브 시스템으로 구현한 것이 특징이다. 접근제어 정책은 각 주체는 영역레이블(domain label)을 가지고, 각 객체는 유형레이블(type label)을 가지고 있어서 연산은 'Allow<domain, type, operation>'과 같은 설정 규칙에 따라 결정된다. 유형의 실행범위는 그룹화되어 있고, 각 역할(Role)들은 상하 계층적 구조를 가지고 상위 역할은 하위 역할 모두를 상속받는다. SELinux에서 제공하는 시스템 호출 몇 가지를 정리하면 <표 3>과 같다.

3. SecuROS

개발된 보안운영체제 시스템에서는 정보와 자원에 대한 접근이 사용자별로 허가된 범위 내에서만 가능하다. 따라서 파일, 유틸리티, 디바이스등을 사용자가 오용할 수 없도록 방지할 수 있다. 또한, 슈퍼유저에게 편중되어 있는 시스템 관리 능력을 분배하여 다중 역할을 총괄하여 발생하던 문제점을 해결하였다[10].

사용자들은 접근 초기부터 보안 레벨을 가지고서 접근하게 된다. 레벨은 MAC을 기반으로 구성하여 등급과 범주값을 의미한다. 이 보안 레벨에 따라 프로세스의 레벨이 결정되고 해당 프로세스는 상위레벨에 속한 정보에 접근할 수 없으며 하위레벨에 속한 정보를 만들 수 없게 된다. 슈퍼유저의 경우에도 모든 레벨의 모든 정보를 조작할 수 있는 것이 아니고 특정 보안레벨이 설정되어 있는 정보는 보안관리자의 허가를 받은 후에만 접근할 수 있다.

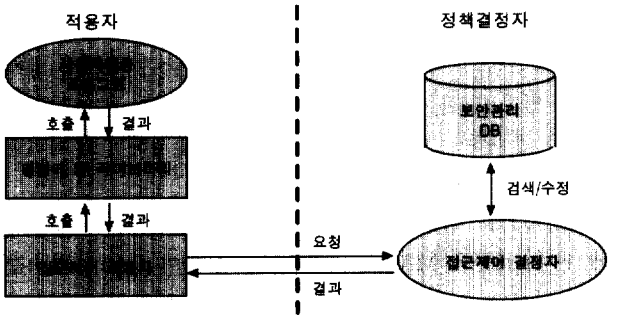
〈표 2〉 Trusted Solaris의 시스템 호출

시스템호출	동 작 내 용	시스템호출	동 작 내 용
facl	파일의 ACL 정보 읽기	chkauth	사용자 인증 검사
fsetfpriv	파일에 대한 권한 변경	fgetfprive	파일에 정의된 권한 읽기
setpattr	프로세스에 대한 보안 특성 플래그 변경	getpattr	프로세스에 대한 보안 특성 플래그 읽기
setcmwlabel	해당 파일의 CMW 레이블 변경	getcmwlabel	해당 파일의 CMW 레이블 읽기
setclearance	프로세스의 비취인가설정	getclearance	프로세스의 비취인가 읽기

〈표 3〉 SELinux 시스템 호출

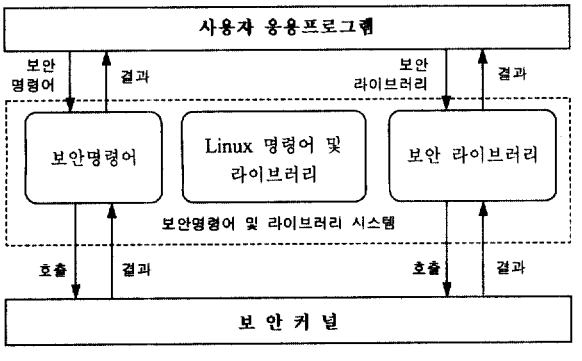
시스템 호출	동 작 내 용	시스템 호출	동 작 내 용
security_load_policy	새로운 규칙을 구성하기 위해 읽기	chsid	파일의 보안 식별자를 변경
connect_secure	소켓의 연결을 초기화	execve_secure	프로그램 실행
fstat_secure	파일의 상태 읽기	security_get_sids	동작하는 보안 식별자 읽기
security_sid_to_context	보안식별자에 해당하는 보안정보 읽기	security_transition_sid	객체에 레이블 부여를 위한 보안식별자 계산
semsid	세마포어에 관련된 보안식별자 찾기	stat_secure	파일의 상태 설정

SecuROS는 Linux 운영체제에 보안모듈을 추가한 형태이다. (그림 1)은 보안운영체제의 구성도를 나타내고 있다. 보안운영체제의 동작은 우선 사용자 응용 프로그램에서 명령어 및 라이브러리를 사용하여 커널에 객체의 접근 요청을 보낸다. 커널의 접근제어 적용자는 해당 접근 요청을 접근제어 결정자에게 전달한다. 접근제어 결정자는 사용자 영역에서 수행되고 해당 프로세스가 해당 자원에 접근권한이 있는지를 검사하는 기능을 수행한다. 접근제어 결정자는 보안 관리 데이터베이스에 저장되어 있는 보안 정보들을 가지고 MAC과 ACL의 보안 정책에 근거하여 해당 사용자가 해당 자원에 접근권한이 있는지를 검사하여 그 결과를 접근제어 적용자에게 보낸다. 커널은 접근제어 결정자로부터 받은 접근 결정 정보를 가지고 사용자 응용프로그램이 해당 자원에 접근할 수 있게 하거나 또는 접근할 수 없음을 말한다. 이와 같이 모든 객체에 대한 접근이 접근제어 적용자와 접근제어 결정자에 의해서 판단된 후 실행되므로 임의적인 사용자의 오용은 발생하지 않는다. 또한 보안관리데이터베이스는 보안관리자에 의해서만 설정되고 사용되 수 있도록 하여 일반사용자는 접근할 수 없다.



(그림 1) 보안운영체제 기능적 구성도

4. SecuROS에서 개발된 사용자 인터페이스



(그림 2) 보안 명령어 및 라이브러리 시스템 구성도

IEEE 표준안 POSIX 1003.1e, 1003.2c는 각각 보안 명령어와 보안 라이브러리에 요구되는 내용을 정의한 것이다. 1003.1e에서는 보안 라이브러리를 정의하고 있으며, 1003.2c는 개방형 시스템의 접근제어리스트, 권한 분리, 강제적인 접근제어등에 대한 보안 유틸리티를 정의하고 있다. 정의하고 있는 내용은 ACL, 감사(Audit), 자격(Capability), MAC, 꼬리표(Information Labeling)로 나뉘어 있다.

개발된 내용은 이중 ACL과 MAC의 모든 명령어와 라이브러리이다. 이 내용을 간략히 설명하면 다음과 같다. (그림 2)는 보안운영체제에서 보안 명령어 및 라이브러리 시스템의 동작 위치를 나타낸 것이다.

4.1 ACL

기존 Linux 시스템에서는 각 객체의 접근 단위를 소유자, 소유자의 그룹, 그외(user, group, other)로 주체를 나누고 이들 주체에 읽기·쓰기·실행(rwx)이라는 권한을 주고 있는데 이것을 좀 더 세분화한 것이 ACL이다. 즉, 특정 사용자나 그룹에게 추가로 다른 값을 정의할 수 있다. ACL라이브러리와 유틸리티에서는 ACL에 있는 정보를 읽거나 생성, 수정 및 삭제하는 기능들을 다룰 수 있는 인터페이스를 제공한다.

〈표 4〉 ACL 명령어 및 라이브러리

명 령 어	동 작 내 용
Setfacl	파일에 acl값을 설정
Getfacl	파일에 설정된 acl 값 읽기
엔트리 조작 라이브러리	
acl_create_entry	acl 엔트리 추가
acl_delete_entry	acl 엔트리 삭제
acl_get_entry	acl 엔트리 값 읽기
acl_set_perms	acl 엔트리의 퍼미션 설정
acl_get_perms	acl 엔트리의 퍼미션 읽기
acl_clear_perms	acl 엔트리의 퍼미션 삭제
acl_get_qualifier	acl 엔트리의 qualifier 읽기
acl_set_qualifier	acl 엔트리의 qualifier 설정
acl_get_tag_type	acl 엔트리의 tag type 읽기
acl_set_tag_type	acl 엔트리의 tag type 설정
ACL 조작 라이브러리	
acl_get_file	파일의 acl 읽기
acl_set_file	파일의 acl 설정
acl_get_fd	파일 디스크립터의 acl 읽기
acl_set_fd	파일 디스크립터의 acl 설정
ACL 형식 변환 라이브러리	
acl_to_text	acl을 text로 변환하기
acl_from_text	text로 들어온 내용을 acl로 변환하기

예를 들어 test라는 파일의 ACL을 읽어보면 다음과 같다. 특별한 설정이 없는 경우 일반적인 유닉스 파일 접근제어와 동일하다. 추가적인 설정은 파일의 소유주와 보안관리자 만이 할 수 있다.

```
[secureos] getfacl test
user::rw-
group::rw-
other::r
```

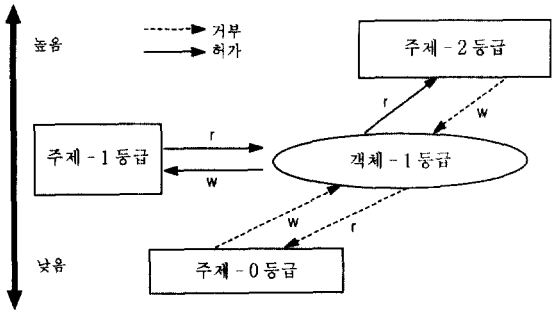
여기에 사용자 번호 500이 test 파일에 대해 읽기, 쓰기, 실행을 할 수 있는 권한을 추가해 보도록 하자. -m 옵션은 추가를 의미한다.

```
[secureos] setfacl -m user : 500 : rwx
[secureos] getfacl test
user : : rw-
group : : rw-
other : : r--
mask : : rwx
user : 500 : rwx
```

이와 같이 유닉스 시스템에서 제한적이던 권한 설정을 특정 사용자 혹은 특정 그룹으로 확장하는 개념이 ACL 이다. 물론 이러한 설정은 파일 소유자 및 보안관리자에 의해서만 가능하다. ACL에 사용되는 명령어와 라이브러리는 <표 4>와 같은 종류가 있다.

4.2 MAC

MAC은 기존 Linux에서 사용되지 않던 개념으로 모든 주체와 객체에 등급과 범주를 설정하는 것이다. 등급은 높고 낮은 값을 가지고 범주는 다수의 그룹으로 생각할 수 있다. (그림 3)은 MAC의 등급 처리에 대한 설명이다. 하위등급은 상위등급의 객체를 읽을 수 없으며(No Read Up), 상위등급은 하위등급에 쓸 수 없다(No Write Down)는 것이 기본 규칙이다. 예를 들어, 어떤 객체가 1등급을 가지고 AB라는 범주로 설정되어 있다면 1등급보다 낮은 등급(예 : 0등급)으로는 이 객체에 접근하여 읽기할 수 없으며, 상위등급(예 : 2등급)으로는 이 객체에 접근하여 쓰기 할 수 없다. A또는 B라는 범주에 속하지 않은 주체 또한 이 객체에 접근할 수 없다.



(그림 3) MAC 등급 처리 규칙

예를 들어 test라는 파일의 MAC을 읽어본 결과는 다음과 같다. 모든 객체는 특별한 설정이 이루어지기 전까지 등급과 범주 모두 0을 가지고 있게 된다.

```
[secureos] getfmac test
class : 0, category : 0
```

MAC을 설정할 때는 현재 프로세스의 MAC보다 높게 설정할 수 없다. 따라서 현재 프로세스의 MAC값과 동일하게 설정하기 위해 현재 프로세스의 mac값을 getpmac으로 읽어보면 다음과 같다. 등급은 십진수로 표시되며 숫자가 클수록 높은 등급을 의미한다. 즉, 1등급 보다 2등급이 상위 등급이다. 십진수로 표시하지 않고 Top Secret, Secret 등의 명시적인 이름으로 표현할 수도 있다. 범주는 2진수로 표시되며 현재는 64개 정도의 범주를 표시할 수 있다. 여기서 각각의 1은 1비트를 의미하는데 실제로는 회사의 부서나 팀 또는 기타 그룹으로 명시될 수 있다. 시스템의 등급과 범주의 최대값은 변경 가능하다.

```
[secureos] getpmac
class : 3, category : 111
```

현재 프로세스의 MAC이 3등급과 범주는 111로 설정되어 있으므로 test파일도 3등급에 111범주로 설정한 후 getfmac으로 결과를 확인한 내용은 다음과 같다.

```
[secureos] setfmac 3 : 111 test
[secureos] getfmac test
class : 3, category : 111
```

만약 test 파일을 허용범위가 벗어나는 5등급에 11111범주로 설정하면 다음과 같은 오류 메시지가 출력되고 test 파일의 MAC은 원래 값을 그대로 유지하게 된다.

```
[secureos] setfmac 5 : 11111
setfmac : Incorrect MAC information
[secureos] getfmac test
class : 3, category : 111
```

MAC을 처리하는 명령어와 라이브러리는 <표 5>와 같다.

〈표 5〉 MAC명령어 및 라이브러리

명 령 어	동 작 내 용
setpmac	프로세스의 MAC 설정
getfeacl	파일의 effective acl 읽기
라이브러리	동 작 내 용
mac_dominate	MAC의 비교 우위 결정
mac_equal	MAC이 동일한지 비교
mac_from_text	문자열을 MAC으로 변환
mac_glb	두 개의 MAC을 가능한 가장 낮은 MAC으로 조합
mac_lub	두 개의 MAC을 가능한 가장 높은 MAC으로 조합
mac_to_text	MAC을 문자열로 변환
mac_get_fd	파일 디스크립터의 MAC 읽기
mac_set_fd	파일 디스크립터의 MAC 설정
mac_get_file	파일의 MAC 읽기
mac_set_file	파일의 MAC 설정
mac_get_proc	프로세스의 MAC 읽기
mac_set_proc	프로세스의 MAC 설정

4.3 추가된 명령어 및 라이브러리

ACL과 MAC 라이브러리를 사용하여 보안 명령어를 구현하는 과정에서 IEEE 표준안에 포함되어 있지 않아서 불편하였던 몇 가지 명령어와 라이브러리를 추가 개발하였다.

4.3.1 settpmac

현재 프로세스의 MAC을 설정

setpmac *label*

label : MAC을 문자열로 표시한 것

4.3.2 getfeacl

ACL은 설정되어 있는 내용과 mask가 설정되어 있는 경우 실제 적용되는 내용이 다를 수 있다. mask가 적용된 ACL을 Effective ACL이라고 하는데 이 값과 실제 설정된 ACL을 구분하여 사용할 때 필요하다.

getfeacl *file*

file : 파일의 위치와 이름

4.3.3 acl_entry_to_text

ACL 엔트리를 문자열로 변환하는 라이브러리이다. ACL의 특정 엔트리만을 출력하고자 할 때 필요있다.

char *acl_entry_to_text(acl_entry_t acl_entry, ssize_t *len_p)

acl_entry : 문자열로 반환하고자 하는 엔트리의 위치
len_p : 반환되는 문자열의 크기를 가리키는 값의 위치
반환값 : 엔트리를 문자열로 변환한 내용이 저장된 위치

4.3.4 acl_entry_from_text

문자열로 입력된 내용을 ACL 엔트리로 변환하는 라이브러리이다. ACL에 추가로 엔트리를 입력하고자 할 때 필요하다.

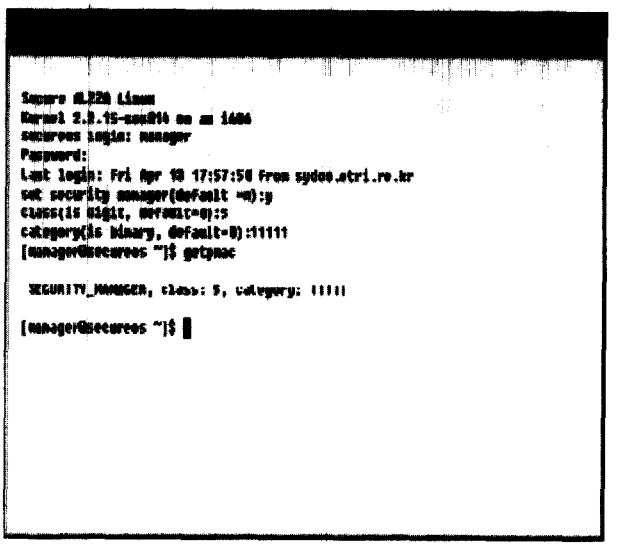
acl_entry_t acl_entry_from_text(const char *buf_p)

buf_p : 문자열이 저장된 위치
반환값 : 문자열을 엔트리로 변환한 엔트리의 위치

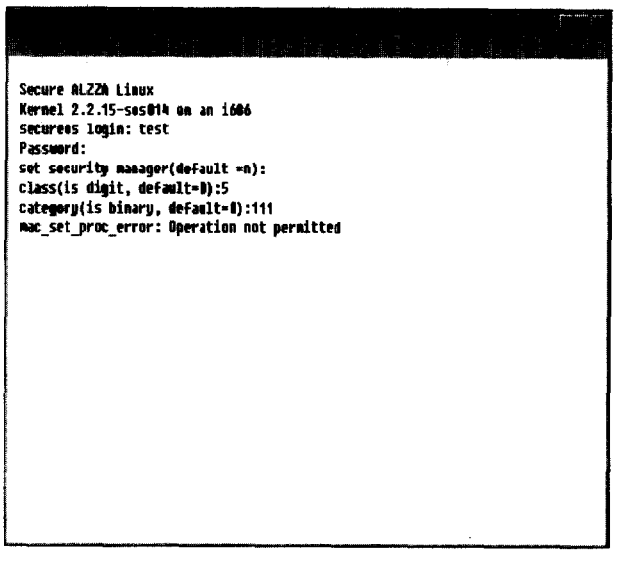
5. SecuROS의 다단계 사용자인증

개발된 시스템에서는 다단계 사용자인증을 제공하고 있다. 이 기능은 개발된 시스템을 효율적으로 사용할 수 있도록 사용자들이 로그인할 때 사용자의 등급과 범주를 묻고 프로세스를 그 등급과 범주 내에서만 작업할 수 있도록 제한하는 기능이다. 처리절차는 다음과 같다. 예를 들어 manager라는 아이디가 보안관리자로 등록되어 있고 등급과 범주를 5 : 11111과 같이 사용할 수 있음이 시스템에 설정되어 있다고 하자. (그림 4)는 이 절차를 수행한 결과이다.

등급과 범주를 벗어나는 파일이나 디바이스의 접근은 허가되지 않는다. 뿐만 아니라 사용자에게 설정된 범위를 벗어나는 시스템 접근도 거부된다. (그림 5)는 test라는 사용자가 범주 3과 등급 111로 시스템에 접근할 수 있도록 미리 정의되어 있는데 등급 5로 접근한 경우의 결과를 나타낸 것이다. 이 경우 사용자는 시스템에 접근할 수 없으며 이 내용은 시스템의 보안 로그에 남게 된다.



(그림 4) 보안운영체제 로그인 성공



(그림 5) 보안운영체제 로그인 실패

다단계 사용자 접근제어 기능은 사용자가 임의로 중요한 상위정보를 하위등급으로 유출하거나 하위등급의 사용자가 상위등급의 자료를 읽어 가는 일들을 방지할 수 있게 하여 시스템의 안전성을 높이게 된다.

6. 결 론

개발된 SecuROS는 Linux 기반의 다단계 접근제어가 제공되는 보안운영체제이고 그 위에 POSIX에서 정의한 ACL과 MAC에 관련된 모든 보안 명령어 및 라이브러리를 보안운영체제 사용자 인터페이스로 제공한다. SecuROS는 슈퍼유저의 기능을 분산시키는 기능과 시스템 자원의 접근제어를 강화하는 역할을 수행하고 있다. 보안 명령어 및 라이브러리는 표준화된 자료를 근거로 했다는 중요한 특징을 가지고 있어서 응용프로그램 개발자에게 편리하고 효율적인 환경을 제공한다. 또한 개발 중 필요가 요구되는 보안 명령어 및 라이브러리를 추가로 개발하였다.

현재 모든 주체 및 객체에 대한 접근제어 설정은 텍스트를 사용하고 있으나 GUI(Graphic User Interface)를 사용하여 보다 편리한 설정기능을 지원하도록 개발중이다. 또한, 보안운영체제 사용자들이 필수적으로 사용하는 암호 라이브러리를 추가적으로 지원하여 암호화를 위하여 다른 라이브러리를 추가 설치해야 하는 번거로움을 줄이고자 한다. 이 외에도 관리자의 편리를 도모하기 위한 관리 인터페이스와 주요 보안 인터페이스의 사용을 조사하여 적절한 조치를 취할 수 있도록 보안 명령어 및 라이브러리 모니터링 툴을 추가하는 것도 고려중이다.

보안운영체제 시스템에서 제공하는 보안 사용자 및 프로그램 인터페이스는 접근제어 방법에 따라 많은 차이를 보인다. 또한 접근제어 방법은 어떤 것이 우위에 있다고 판단되기보다는 응용프로그램 및 적용분야에 따라 적합한 것을 사용하게 된다. 현재 보안운영체제 시스템은 범용화되어 사용되지 않고 있기 때문에 그 성능을 평가하는 것 또한 쉽지 않다. 국내에는 아직 보안운영체제 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 차별화되는 기술 보유의 기틀이 마련되어야 하겠다.

참 고 문 헌

[1] 홍승필, 고제욱, "정보보안 기술과 구현", pp.293, 파워북, 1998.

[2] IEEE Std 1003.1e - Draft standard for InformationTechnology-Portable Operating System Interface(POSIX) Part 1 : System Application Program Interface(API)- Protection, Audit and Control Interfaces.

[3] IEEE Std 1003.2c - Draft standard for Information Technology-Portable Operating System Interface(POSIX) Part 2 : Shell and Utilities : Protection and Control Interfaces.

[4] "DoD Trusted Computer System Evaluation Criteria," http : // 147.51.219.9/otd/c2protect/isso/DOD/520028std/5200_28std1.htm#1.0.

[5] "Eveluated Product List by Vendor," http : // www.radium.ncsc.mil/tpep/epl/epl-by-vendor.html.

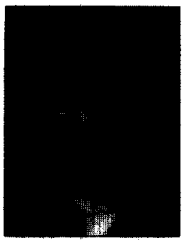
[6] "Rule Sset Based Access Control," http : // www.rsbac.org/.

[7] "Medusa," http : // www.medusa.formax.sk.

[8] "Security-Enhanced Linux," http : // www.nsa.gov/selinux/.

[9] "Trusted Solaris 8," http : // www.sun.com/trusted-solaris/.

[10] Jong-Gook Ko, So-young Doo, Sung-Kyung Un, and Jeong-Nyeo Kim, "Design and Implementation for Secure OS based on Linux," WISA2000, Vol.1 No.1. pp.175-181.



두 소 영

e-mail : sydoo@etri.re.kr

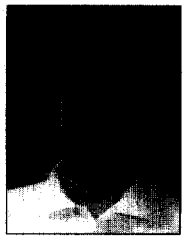
1992년 군산대학교 정보통신공학과 졸업 (학사)

1994년 충남대학교 컴퓨터공학과 졸업 (석사)

1994~1997년 대우 고등기술연구원

2000~현재 한국전자통신연구원

관심분야 : 정보보호, 암호프로토콜, 네트워크보안, 고속통신프로토콜



고 종 국

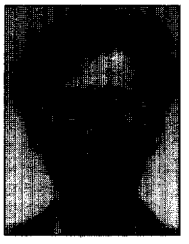
e-mail : jgko@etri.re.kr

1998년 전북대학교 전산학과 졸업(학사)

2000년 광주과학기술원 정보통신공학과 졸업(석사)

2000년~현재 한국전자통신연구원

관심분야 : 정보보호, 네트웍 프로그래밍, 운영체제



은 성 경

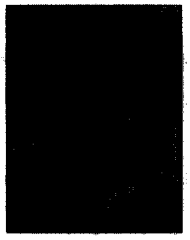
e-mail : skun@etri.re.kr

1991년 전북대학교 컴퓨터공학과 졸업 (학사)

1993년 포항공과대학 전자계산학과 졸업 (석사)

1993~현재 한국전자통신연구원

관심분야 : 디지털전송, 시스템보안, 시스템평가



김 정 녀

e-mail : jnkim@etri.re.kr

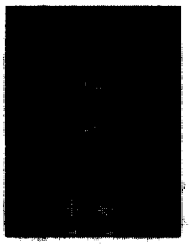
1987년 전남대학교 전산통계 학과 졸업(학사)

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

2000년 충남대학교 대학원 컴퓨터공학과(석사)

1988년~현재 한국전자통신연구원 보안운영체제연구팀장(선임연구원)

관심분야 : 운영체제, 분산 처리, 고장 감내, 시스템 보안



공 은 배

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

1978년 서울대학교 계산통계학과 졸업(학사)

1981년 서울대학교 계산통계학과 졸업(석사)

1995년 Oregon State Univ. 전산학과 졸업(박사)

1996년~현재 충남대학교 컴퓨터공학과 정교수

관심분야 : 암호학, 기계학습, 생물정보학