멀티 운영체제 기반의 파일 접근 제어 모듈 설계 및 구현*

소 우 영**

요 약

최근 각종 보안 침해 사고는 불특정 다수를 대상으로 발생하고 있으며, 이는 네트워크를 통한 정보의 공유가 가속화되면서 그 피해가 더욱 늘어나고 있다. 이러한 보안 침해 사고는 침입차단 시스템, 침입탐지시스템, 가상사설망 등 정보보호시스템의 활용으로 최소화가 가능하지만, 이는 전문가적인 지식이 필요하며 일반 사용자가 운영하기가 쉽지 않다. 본 논문에서는 일반 사용자가 쉽게 사용할 수 있도록 파일 접근 제어 모듈을 설계 및 개발하였고, 이는 공격에 대한 탐지가 아닌 운영체제상에서 공격에 대한 차단할 수 있다. 본 논문에서 구현한 파일 접근 제어 모듈은 윈도우환경 뿐만 아니라 리눅스 환경에서도 적용할 수 있으며, 다중 사용자가 사용하는 운영체제에 따라 파일에 대한 접근 제어를 함으로써 파일에 대한 접근제어, 무결성, 부인 방지를 할 수 있다.

Design and Implementation of File Access Control Module Based on Multi-Operating System

Wooyoung Soh**

ABSTRACT

Recently, various threat and security incident are occurred for unspecified individuals, and this problem increases as the rapid of information sharing through Internet. The using of Information Security System such as IDS, Firewall, VPN etc. makes this problem minimal. However, professional knowledge or skill is needed in that case, normal user can't operate the Information Security System. This paper designs and implements File Access Control Module(FACM) to use easily for normal user against malicious threats and attacks. The FACM can exclude from malicious threats and attacks based on operation system rather than detection of threats and attacks. The FACM is working not only Windows System but also Linux System, and the FACM has effect on access control, integrity and non-repudiation for a file with an access control over files on the each OS that are used by multi-user.

Key words : File Access Control, Security Kernel, Secure OS

---

* 이 논문은 2007년도 한남대학교 교비학술연구조성비 지원에 의하여 연구되었음.

** 한남대학교 컴퓨터공학과 교수

1. 서 론

최근 급속한 정보통신기술의 발달에 따라 정보시스템은 과거 인간이 상상하지 못했던 편리함과 신속성을 제공하고 있으나, 그에 따른 각가지 문제점들 또한 발생하여 많은 피해를 입고 있다.

침해사고를 예방하고 효과적인 대응방법을 마련하기 위해 침입차단기술, 침입탐지기술 등 여러 가지 정보보호기술들이 개발되어 왔다. 그러나 이와 같은 기술들은 알려진 취약점에 대한 예방과 탐지에 대해서는 좋은 결과를 보여주지만, 알져지지 않은 취약점이나 공격에 대해서는 적절한 대응이 쉽지 않은 단점이 있다. 또한 대개의 침해사고 피해 발생 시 중요한 서비스를 중단하게 되며 이 경우 매우 중대한 문제를 야기시킬 수도 있다. 이와 같이 알려지지 않은 취약점이나 공격에 의한 침해사고 대응방법이 요구된다.

침입탐지시스템이나 침입차단시스템이 네트워크를 통한 침입대응방법이었다면 본 논문에서 제시되는 파일 접근 제어는 시스템에서의 침해에 대한 대응 방법이다. 본 논문에서는 보안 운영체제에서 사용할 수 있는 파일 접근 제어 모듈을 설계 및 구현함으로써 알려지지 않은 새로운 침입 유형에 대해서 파일을 수정 및 삭제시키지 못하게 함으로써 침해 대응을 할 수 있다. 침입자가 관리자 권한인 Administrator 또는 root의 계정 권한을 획득하였더라도 보안 운영체제에서 사용되는 권한으로 중요한 시스템 파일이나 보안이 필요한 파일에 대한 접근을 차단한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구로써 보안 운영체제에 대하여 알아보고 제 3장에서는 본 논문에서 제시하는 모듈에 대한 설계 및 구현에 대하여 서술한다. 마지막으로 제 4장에서는 결론을 맺는다.

2. 보안 운영체제

2.1 보안 운영체제

보안 운영체제란 기존의 커널에 보안 기능을 통합시킨 보안 커널(Secure Kernel)이 추가로 이식된 운영체제로 참조 모니터(Reference Monior) 개념을 정의한 TCB(Trusted Computing Base)의 하드웨어, 펌웨어, 소프트웨어의 요소를 뜻한다[1].

보안 운영체제의 기능을 살펴보면 다음과 같다.

- 사용자에 대한 식별 및 인증
- 강제적 접근 통제(MAC : Mandatory Access Control)
- 임의적 접근 통제(DAC : Discretionary Access Control)
- 재사용 방지(Object Reuse Prevention)
- 침입 탐지(Intrusion Detection)

참조 모니터는 보안 커널[2]의 가장 중요한 부분으로 참조 모니터의 기능을 살펴보면 다음과 같다.

- 객체에 대한 접근 통제 기능 수행한다.
- 감사, 식별 및 인증, 보안 매개 변수 설정 등과 같은 다른 보안 매커니즘과 데이터를 교환하면서 상호 작용한다.

2.2 보안 커널의 보안 기능[3]

보안 커널의 보안 기능은 각각 다음과 같은 역할을 수행한다.

- 식별 및 인증 : 고유한 사용자 신분에 대한 인증 및 검증
- 강제적 접근 통제 : 사용자의 접근결정에 대해 고정된 보안 속성을 보안 관리자 또는 운영체제에 의해 정해진 엄격한 규칙에 따라 자동적으로 부여함으로써 사용자의 자유재량에 상관없이 강제적으로 접근 통제

- 임의적 접근 통제 : 사전에 보안 정책이나 보안 관리자에 의해 개별 사용자에게 합법적으로 부여한 한도내의 재량권에 따라 사용자가 그 재량권을 적용해 접근 통제
- 객체 재사용 방지 : 메모리에 이전 사용자가 사용하던 정보가 남아 있지 않도록 기억 장치 공간을 깨끗이 정리
- 완전한 중재 및 조정 : 모든 접근 경로에 대한 완전한 통제
- 감사 및 감사 기록 축소 : 보안 관련사건 기록의 유지 및 감사 기록의 보호

막대한 양의 감사 기록에 대한 분석 및 축소

- 완전한 경로 : 패스워드 설정 및 접근 허용의 변경 등과 같은 보안 관련 작업을 수행 할때 안전한 경로 제공
- 침입 탐지 : 정상적인 시스템의 사용 패턴을 분석하고, 비정상적인 사용이 발생했을 때 이에 대한 경보 제공

2.3 보안 커널 구현 방법

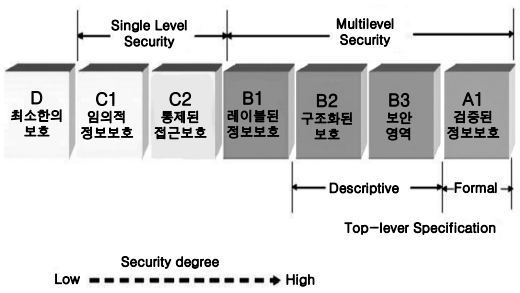
보안 커널의 구현 방법은 커널을 새로이 구현하는 방법과 모듈방식으로 만들어 모듈을 커널 속에 심는 방법(LKM : Loadable Kernel Module) 등 두 가지 방법이 있으며 통합 커널 기반 방식은 기존 운영체제의 모든 기능을 포함하며 API는 커널 서비스를 이용하게 하며 이와는 대조적으로 마이크로 커널 방식은 기존 통합 커널을 최소화하고 시스템을 최대한 모듈화 한다.

마이크로 커널이란 범용 운영체제로서 사용될 것을 염두에 둔 운영체제에 있어서 커널의 조립성은 확장성이라는 면에 초점을 두고 연구, 개발되어 왔다. 특히 상업용 시스템의 경우 전통적으로 동적으로 적재 가능한 확장 모듈, 예를 들어 디바이스 드라이버, 설계 당시에는 고려치 않았던 새로운 서브시스템(subsystem), 새로운 파일 시스템 등을 지원할 수 있는 경로, 즉 인터페이스를 제공하는 방식을 택해왔다. 이것은 일종의 일체형 커널 모델로서 매우 현실적인 접근 방식이기는 하지만, 커널의 안정성(reliability)에 대해서는 특별한 대책이 없는 것이 또한 현실이다. 이에 확장성에 대한 기존의 방식과는 전혀 다른 새로운 모델이 제시되었는데 이것이 마이크로 커널 모델이다. 운영체제가 지원해야 하는 커널의 기능이 하나의 커다란 커널에 모아져 있던 일체형 커널과 달리, 마이크로 커널은 커널을 최소화하고 커널 외부에 필요한 기능을 제공하는 서버를 구현하는 접근 방법을 택하고 있어, 기본적으로 커널은 주소 공간(address space) 관리, 프로세스간 통신(IPC), 그리고 기본적인 스케줄링 기능만을 제공한다는 것이다. 디바이스 드라이버를 포함한 모든 기능들은 서버형태로 사용자 모드에서 수행하면서 커널 입장에서는 다른 사용자 응용 프로그램과 완전히 동일하게 취급된다. 또한 각 서버들은 자신만의 주소 공간을 갖고 있기 때문에 각 서버에 의해 지원되는 시스템 요소들은 상호간의 간섭으로부터 보호될 수 있게 된다. 이와 같은 마이크로 커널의 경우 1980년 후반 도입된 이후 폭넓은 범위에서의 유연성과 확장성 지원 등 많은 장점으로 인해 매우 각광을 받았으나, 서버의 기능을 이용하기 위한 빈번한 IPC에 의한 오버헤드와 구현상의 불합리성으로 인해 매우 낮은 성능을 보여왔다. 이에 새로운 방향에서 마이크로 커널을 접근하려는 노력이 많이 있다.

보안 커널은 운영체제 기술 발전의 흐름에 따라 보안 운영체제 또한 기존의 IK(Integrated Kernel : 통합 커널) 방식보다는 MK(Micro Kernel) 방식으로 개발 경향이 변하고 있다.

TCSEC(Trusted computer security evaluation criteria)은 미국 국립 컴퓨터 보안 센터(NCSC)가 1985년 발간한 안전한 컴퓨터 시스템을 위한 평가 지침서로 컴퓨터 시스템의 보안을 효과적으로 평가하기 위해 6개의 기본 요구 사항을 정의하였으며, 그 사항을 만족시키는 수준에 따라 7가지의 평

가 등급을 제시하였다[4].



(그림 1) Evaluation level of TCSEC

3. 다중 운영체제 기반의 파일 접근 제어 모듈(FACM) 설계 및 개발

3.1 파일 접근 제어 모듈(FACM) 개발 목록 정의

3.1.1 보안 설명자

보안 오브젝트는 보안 설정에 따라 사용자별로 사용 권한이 달라진다. 이런 엑세스 제어가 오브젝트별로 가능하려면 오브젝트 자체에 보안 설정이 기억되어 있어야 한다. 예를 들어 Admin.txt 파일에는 User1은 읽을 수만 있고, User2는 읽고 쓸 수 있다는 보안 설정정보가 기억되어 있다면 이런 보안 설정 정보를 보안 설명자라고 한다. 보안 설명자는 보안 설정 정보를 기억하는 일종의 구조체이며, 다음과 같은 정보들로 구성되어 있다.

- 소유자의 SID : 오브젝트를 소유하고 있는 사용자의 SID이다. 소유자는 보안 설명자의 DACL 정보와는 상관없이 오브젝트에 대한 모든 권한을 가진다.
- 소유자의 그룹 SID : 소유자가 속한 그룹의 ID이며, 이 정보는 타 운영체제와의 호환성을 위해 존재하며 윈도우즈 환경에서는 큰 의미가 없다.
- DACL : 사용자별 권한 정보의 목록이다. 누가 이 오브젝트를 읽을 수 있는지 또는 쓰기가 거부되는지 등의 정보를 기억하고 있다. 보안 설명자의 가장 핵심이 되는 중요한 정보이다.
- SACL : 오브젝트를 엑세스할 때 기록할 감사 정보를 기억한다. 누가 이 오브젝트에 대해 어떤 동작을 할 때 이벤트 로그에 기록을 남기도록 하는 정보가 들어있다.

3.1.2 보안 식별자

보안 식별자(Security Identifiers)는 윈도우 NT와 2000에서 사용자나 그룹 같은 객체를 유일하게 식별해주는 값으로 사용되는 식별자를 의미한다. 사용자에게 할당된 SID는 사용자나 그룹들이 수행하는 프로세스나 시도하는 행위에 따라붙는 접근 허용 토큰의 일부이다. SID는 버전, 도메인 정보, 사용자 정보 등을 포함하고 있으며 이진 포맷으로 되어 있다.

- **SID의 내용**
  - 사용자와 그룹 보안 설명자
  - 48비트 ID 인증
  - 수정 수준
  - 다양한 하위 권한 값

3.1.3 엑세스 토큰

보안 오브젝트가 허용되지 않은 사용자로부터 보호되기 위해서는 엑세스하고자 하는 사람의 신분에 대한 정보와 보안 설명자를 비교해야 할 필요가 있다. 여기서 사용자의 신분에 대한 정보를 엑세스 토큰이라고 하며 보안 설명자와 함께 윈도우즈 운영체제의 보안 체계를 구성하는 중요한 요소이다.

3.1.4 ACL

접근 제어 목록(Access Control List)은 개별적

인 보안 정보 조각인 접근 제어 항목(ACE)의 배열이다. SID와 ACL의 비교를 통해 자원에 대한 접근가능 여부가 판단된다[5, 6].

- 임의 액세스 제어목록(DACL) : 개체의 보안설명자에서 특정 사용자와 그룹에 대해 사용권한을 허가하거나 거부하는 목록
- 시스템 액세스 제어목록(SACL) : 개체의 보안설명자에서 사용자나 그룹마다 감사해야할 이벤트를 지정하는 목록

3.1.5 ACE

ACL은 ACE(Access Control Entry)의 배열이며 ACE는 ACL의 배열요소이다. ACL은 ACE를 전혀 가지지 않을 수도 있고 복수개의 ACE를 가질 수도 있다. ACE는 실질적인 보안 정보를 가지는 요소라고 할 수 있으며, 보안 설명자나 ACL은 ACE를 담기 위한 그릇일 뿐이다. 보안 오브젝트의 보안 설정을 알고 싶거나 변경하고 싶다면 ACE를 읽고 편집해야 한다. Windows 2000에는 여섯 가지 유형의 ACE가 있으며, 이 여섯 가지 유형의 ACE는 파일 시스템 개체, Active Directory 개체 등 보안을 적용할 수 있는 모든 개체에 적용되는 일반 ACE와 Active Directory 개체에만 적용되는 개체 ACE가 있다. 일반 ACE와 개체 ACE는 기본적으로는 동일하며 상속 및 개체 액세스에 대해 제공하는 제어 밀도에만 차이가 있다.

3.1.6 엑세스 권한

ACE의 정보는 누구에게 어떤 엑세스 권한을 허가 또는 금지 할 것인가를 지정한다. 각 보안 오브젝트에 적용되는 엑세스 권한은 오브젝트별로 다양하다. 읽기, 쓰기, 삭제 등의 공통적인 엑세스 권한이 있고, 추가, 속성변경, 이동, 복사, 종료, 질의, 우선순위 변경 등등 오브젝트별로 고유한 엑세스 권한들도 있다. 이렇게 다양한 엑세스 권한의 조합을 표현하기 위해 32비트 정수인 엑세스 마스크가 사용된다. 엑세스 마스크의 각 비트는 엑세스 권한과 일대일로 대응되므로 엑세스 마스크는 엑세스 권한들을 멤버로 가지는 비트 필드형의 구조체라 할 수 있다.

3.2 윈도우 기반의 FACM 설계 및 개발

3.2.1 개발 환경

본 논문에서 구현하는 모듈은 윈도우 2000 Professional을 기본 운영체제로 사용하며, 또한 테스트를 위하여 Administrator와 구별하여 또 다른 계정을 하나 생성한다. 프로그램 이름은 ACL(Access Control List)이라 하고, DLL 라이브러리의 이름은 ACLLib라 한다.

모듈에서 필요로 하는 함수는 Visual C++ 6.0을 이용하여 DLL로 작성하고, GUI는 델파이 6.0을 사용한다.

3.2.2 설계 및 개발

윈도우 환경에서의 파일에 대한 보안 객체를 읽고 쓰는 것은 aclapi를 이용하여 사용하고, NTFS 파일 시스템에서는 파일에 보안 설명자를 두며 보안 설명자는 두개의 ACL(Access Control List)로 구성되어 있다. ACL은 개별적인 보안 정보조각인 ACE(Access Control Entry)의 배열이다.

보안 설명자는 두 개의 ACL을 가지고 있으며 하나는 엑세스 권한 목록인 DACL(Discretionary ACL)이며 나머지 하나는 감사 기록 작성을 통제하는 SACL(System ACL)이다. DACL은 여러 개의 ACE로 구성되며 각 ACE는 누가 이 오브젝트에 대한 어떤 권한을 가지는지에 대한 정보를 표현한다.

본 논문에서는 이러한 각각의 파일에 대한 ACE에 대하여 읽고 쓰고 수정함으로써 각 파일에 대한 보안 정책을 설정한다.

ACE에 대한 읽는 함수로부터, 각 파일에 대한 소유자 정보와 도메인 정보, DACL 정보를 얻을 수 있으며, ACE 객체를 쓰는 함수를 다음과 같은

함수로 export 되어 DLL 라이브러리를 통해 사용되어 진다.

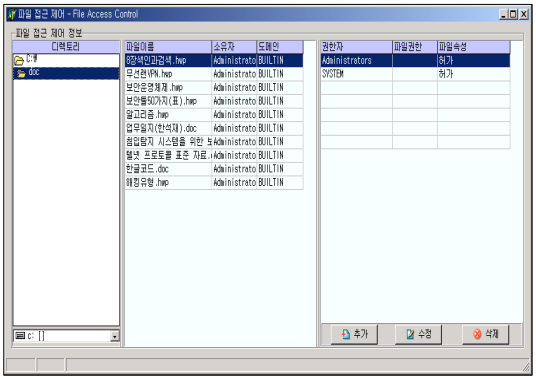
```
extern “C” __declspec(dllexport) int
GetSecurity(char *filename, ACLEntry *Entry) ;

extern “C” __declspec(dllexport) int SetSecurity
(char *FileName, char *Trustee, DWORD
AccessMode, DWORD Permission) ;
```

위에서 구현된 DLL을 통하여 ACL 프로그램을 작성하였으며 GUI 화면을 구성하는 프로그램은 델파이로 작성되었다. 델파이에서 ACE에 대한 정보를 읽는 함수 호출과 ACE에 대한 정보를 쓰는 함수 호출은 다음과 같다.

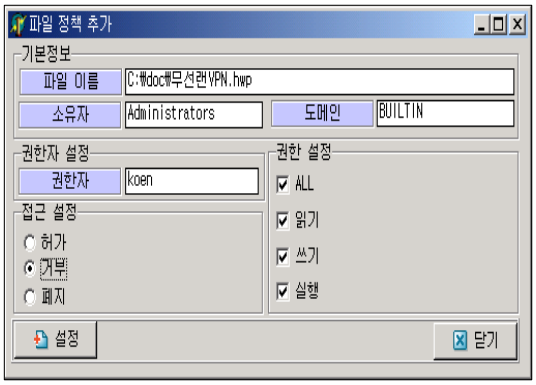
```
if GetSecurity(PChar(FolderName + '\' +
FileName), @Entry) < 0 then Exit ;
        // R_fucntion call

Ret := SetSecurity(PChar(FileNameEdit.Text),
PChar(TrusteeEdit.Text), Access, Permission)
        //W_fuction call
```



(그림 2) ACL Program

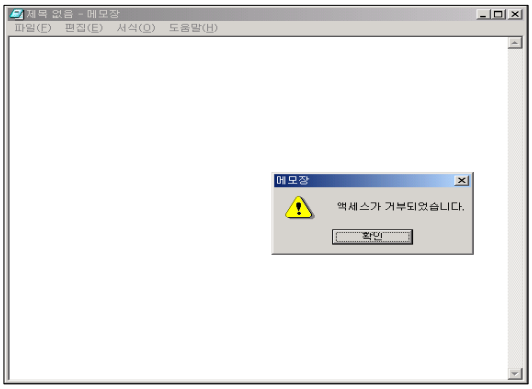
(그림 2)에서 드라이버와 디렉토리를 선택하여 파일을 선택하면 각 파일에 대한 ACE 객체를 볼 수 있으며 각 파일에 대한 ACE 객체에 대해 추가, 수정 및 삭제할 수 있다.



(그림 3) Screen of Object Append

(그림 3)은 파일에 대한 ACE 객체를 추가하는 화면이다. 테스트를 위해 koen이란 Administrator가 아닌 계정을 만들고 파일에 대한 정책을 설정하였다.

위에서와 같이 해당파일이 Administrator이지만 권한자를 Administrator가 아닌 계정으로 거부로 모든 권한을 설정하였다. 설정에 성공한 후 윈도우를 Administrator로 로그인 한 후 해당 파일에 접근하였다.



(그림 4) Screen of Access denied

해당 파일을 접근 시 엑세스가 거부되었다는 메

시지창을 확인할 수 있다.

3.3 리눅스 기반의 FACM 설계 및 구현

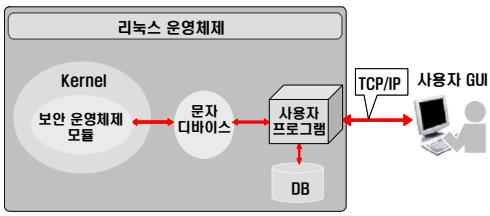
3.3.1 개발 환경

본 논문에서 구현한 모듈은 RedHat Linux 7.3을 기본 운영체제로 사용한다. 구현언어는 C언어를 사용하고, 사용자 GUI 환경은 Delphi 6.0을 사용하여 윈도우 운영체제에서 사용할 수 있도록 설계 및 개발한다.

3.3.2 설계 및 개발

본 논문에서 구현된 프로그램은 크게 3개로 되어 있으며 보안 운영체제 커널 모듈, 커널과 사용자사이의 데이터를 주고받을 수 있는 프로그램과 사용자 GUI 환경 프로그램으로 구성한다. 접근 제어를 위한 보안 운영체제 모듈은 커널 모듈 방식으로 되어 있으며 Loadable 할 수 있도록 하고, 이러한 방식은 커널 소스에 삽입하여 재 컴파일을 통하여 Rebuild하는 방식보다 개발하기 쉬우며 사용할 시도 Loadable 할 수 있는 장점이 있다. 보안 운영체제에서 사용되는 커널 모듈은 사용자 환경에서 접근 할 수 없으므로 문자 디바이스(Character Device)를 통하여 데이터를 주고 받을 수 있도록 하였으며 이러한 문자 디바이스를 접근하기 위한 프로그램을 개발한다. 또한 GUI 환경은 관리자가 사용하기 쉽도록 윈도우 환경으로 개발하였으며 리눅스 운영체제와의 데이터를 주고받을 수 있도록 TCP/IP 통신을 사용한다.

보안 운영체제를 위한 커널 모듈은 Load, Unload 할 수 있는 Loadable한 모듈로 개발되었으며 커널의 시스템 콜을 후킹(Hooking)하는 방식으로 되어 있다. 불법 침입으로 허가받지 않은 사용자가 root 관리자의 권한을 얻어 특정 라이브러리나 파일 등에 접근하려고 할 시 시스템 콜을 사용하게 된다. 이때 시스템 콜을 보안 운영체제 모듈에서 중간에 검증하게 된다.

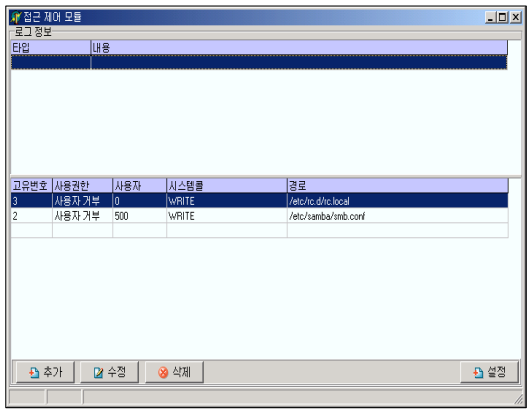


(그림 5) 구성도

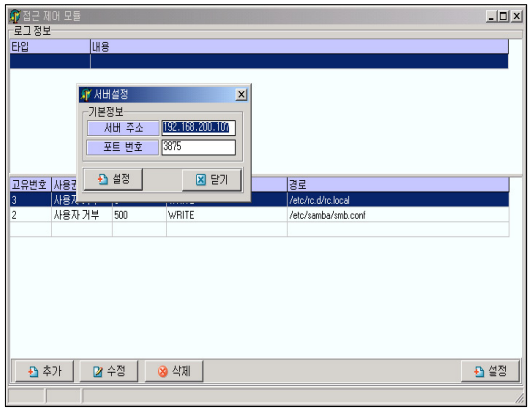
다음은 시스템 콜을 검증하는 방식이다.

```
extern void *sys_call_table[] ;
    //시스템 콜 테이블 정의
int (*orig_mkdir)(const char *pathname, int
mode) ;
    //mkdir 시스템 콜의 원래 시스템 콜 포인터
int (*orig_open)(const char * filename, int flags,
int mode) ;
    //open 시스템 콜의 원래 시스템 콜 포인터
int (*orig_read)(unsigned int fd, char * buf,
size_t count) ;
    //read 시스템 콜의 원래 시스템 콜 포인터
int (*orig_write)(unsigned int fd, char * buf,
size_t count) ;
    //write 시스템 콜의 원래 시스템 콜 포인터
```

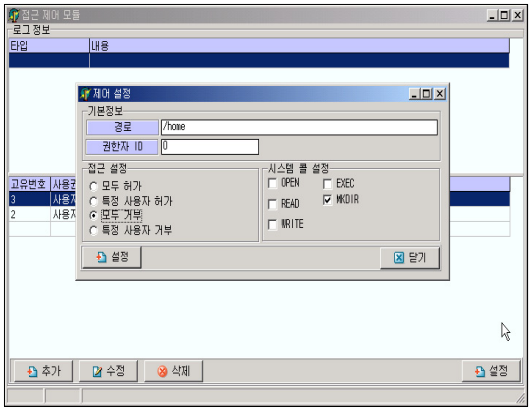
사용자 프로그램은 커널 모듈과 사용자 사이의 데이터를 주고받을 수 있도록 개발 하며, 커널과 사용자 사이에는 디바이스를 통하여서만 접근 가능하므로 커널 모듈과 사용자 사이에 데이터를 주고받을 수 있도록 하기 위하여 문자 디바이스(Character Device)를 두어 사용한다. 사용자 프로그램은 사용자 GUI와 데이터를 주고받기 위해 TCP/IP 소켓 통신을 한고, 문자 디바이스와 통신은 일반적인 파일 접근 방식과 같다. 또한 사용자 프로그램은 접근 제어 정책을 데이터베이스를 사용하여 저장하며 데이터베이스는 리눅스 설치 시 기본으로 설치되는 GDBM을 이용한다. GDBM는 특정 데이터베이스 사용시 해당 데이터베이스를 새로 설치해야하는 번거로움과 관련 테이블이 관계 형일 필요가 없고 트랜잭션과 같은 기능을 수행 할 필요가 없다는 특징을 가지고 있다.



(그림 6) 사용자 GUI 화면



(그림 7) 서버 정보 설정 화면



(그림 8) 정책 추가 화면

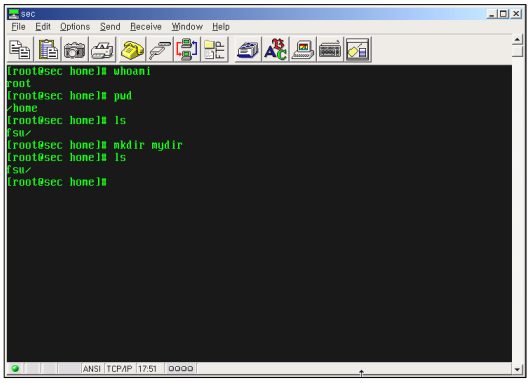
(그림 6)은 사용자 GUI 화면으로 접근 제어 정책을 설정하며 커널 모듈에서 로그 발생 시 로그를 나타낸다. 메뉴는 접근 제어 정책에 대한 추가, 수정, 삭제 메뉴가 있으며 서버 정보 설정에 대하여 있다.

(그림 7)과 (그림 8)은 각각 서버 정보에 대한 설정 화면과 정책 추가 화면을 나타낸다.

위의 설정에서는 /home 디렉토리 밑에 모든 사용자에 대한 mkdir 시스템 콜을 거부하도록 한 것이다. 위와 같이 되었을 경우 root인 관리자의 권한을 획득하더라도 /home 밑에 디렉토리를 만들 수 없도록 한 것이다.

다음은 위와 같이 설정한 후 /home 디렉토리 밑에 새로운 디렉토리를 만든 결과로써, whoami를 통하여 어떤 권한인지를 보여주고 그림은 pwd를 통해 현재 디렉토리를 보여주고 ls를 통해 현재 디렉토리의 파일과 디렉토리를 보여준 후 mkdir을 통하여 /home 밑에 디렉토리를 만든 다음 ls를 통하여 현재 디렉토리의 결과를 보여준다.

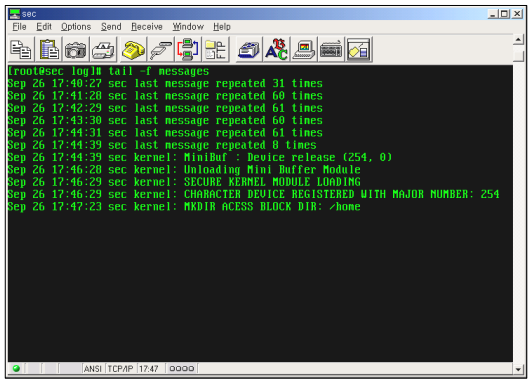
아래 결과에서와 같이 root 권한으로 /home 밑에 디렉토리를 만들더라도 디렉토리가 생성되지 않음을 볼 수 있다.



(그림 9) 구현 결과 화면

다음 (그림 10)은 printk 커널 함수를 통하여 콘솔로 보여주게 한 화면이다. 다음과 같이 콘솔을

사용하지 않을 시 다음과 같이 /var/log/message를 보면 된다.



(그림 10) 실행 결과 화면

4. 결 론

전 세계가 정보 통신의 급속한 발전으로 컴퓨터 및 인터넷의 사용이 급격히 증가함에 따라 정보처리의 편의성이 증대되는 한편 컴퓨터 보안에 취약한 일반 사용자들은 정보보호상의 다양한 문제에 처하고 있다.

인터넷을 통한 불법침입으로 시스템의 자원 및 중요한 자료들이 위협 당하고 있고 때로는 치명적인 손실을 입고 있어 인터넷 상에서의 보안 서비스에 대한 필요성이 절실히 요구되는 실정이다[3]. 이러한 보안 서비스에 대한 필요성은 시스템의 침해 사고에서도 나올 수 있으며 이는 보안 운영체제로써 대응할 수 있다.

본 논문에서는 전문적인 보안 지식이 없는 일반 사용자 또한 쉽고 편하게 사용할 수 있도록 GUI 환경을 제공하고, 윈도우기반 뿐만 아니라 리눅스 운영체제 기반에서도 활용 가능하도록 개발하였다. 이는 다중 사용자가 이용하는 단말에서 프라이버시가 보장되어야 하는 개인용 파일이나 데이터에 대한 접근을 방지함으로써 해당 파일이나 데이터에 대한 무결성 및 부인 방지를 제공할 수 있다.

향후, 본 논문에서 개발한 다중 운영체제기반 FACM은 해당 운영체제에 대한 모듈이 아닌 하나의 모듈로 통합하여, 더욱 효과적인 프로그램으로 개발해야 할 것으로 사료된다. 이는 비단 윈도우나 리눅스 기반뿐만 아니라 유비쿼터스 컴퓨팅 환경에서의 여타 OS에서도 적용 가능할 수 있도록 계속적인 연구가 필요할 것으로 사료된다.

참 고 문 헌

[1] 김재명, 홍기융, 홍기완, “Secure OS 보안정책 및 메커니즘”, 정보보호학회지, 제13권, 제4호, 2003.

[2] 이홍섭, 이철원, 이정효, 박정호, “정보통신 기반구조 보호를 위한 보안 커널 개발 동향”, 정보보호학회지, 제8권, 제4호, 1998.

[3] 소우영 외 3인, “컴퓨터 통신 보안”, 그린출판사, pp. 603-606

[4] San Jose, “Common Criteria Solutions”, Security Lab, http://www.fact-index.com/t/tc/tcsec.html.

[5] Chris, Prosise and Kevin, Mandia., “Incident Response : Investigating Computer Crime”, McCraw-Hill, p. 371, 2001.

[6] 신철우, “Windows 2000 Server”, 영진출판사, 2000.

소 우 영

1979년 중앙대학교 전자계산학과 (이학사)
1981년 서울대학교 전자계산학과 (이학석사)
1990년 미국매릴랜드대학 전자계산학과(공학박사)
1996년 한국전자통신 초빙연구원
1991년~현재 한남대학교 컴퓨터공학과 정교수
2003년~현재 한남대학교 정보통신교육원장