제1회 한국사이버테러정보전학회 춘계학술발표대회 (2004.5)

윈도우 기반의 보안 운영체제를 위한 파일 접근 제어 모듈 설계 및 구현

한석재, 김완경, 소우영
한남대학교 컴퓨터공학과

요 약

네트워크를 통한 정보의 공유가 가속화되면서 정보시스템은 다양한 보안 위협에 노출되어 있으며 각종 보안 사고가 사회적 문제로 나타나고 있다. 이에 따라 시스템을 안전하게 보호하기 위하여 침입차단시스템, 침입탐지시스템, 가상사설망 등의 다양한 보안 시스템들이 운영되고 있다. 이러한 보안 시스템은 전문가적인 지식이 필요하며 일반 사용자가 운영하기가 쉽지 않다. 본 논문에서는 공격에 대한 탐지가 아닌 운영체제상에서 공격에 대한 차단할 수 있는 보안 운영체제에 필요한 파일 접근제어 모듈을 설계 및 개발하였다. 본 논문에서 구현된 모듈은 윈도우 기반으로 하였으며 여러 사용자가 사용하는 윈도우 기반의 운영체제에서 파일에 대한 접근 제어를 함으로써 파일에 대한 무결성, 부인 방지를 할 수 있다.

Design and Implementation of File Access Control Module for Secure Operating System Using on Windows

Seak-Hae Han, Wan-Kyung Kim, Woo-Young Soh

ABSTRACT

With the rapid development of information sharing through network, IT system is exposed to various threatener and security incident are became a social problem. As a countermeasure, various security systems are been using such as IDS, Firewall, VPN etc.. But, expertise or expert is required to handle security system. In this paper, design and implementation of file access control module for secure operation system. The module, implemented in this paper, is based on Windows and has effect integrity and non-repudiation for a file.

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. 그러

본 연구는 과학기술부 지역협력연구사업 (R12-2003-004-01002-0) 지원으로 수행되었음

나, 이와 같은 기술들은 알려진 취약점에 대한 예방과 탐지에 대해서는 좋은 결과를 보여주지만, 알려지지 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 야기 시킬 수도 있다. 이와 같이 알려지지 않은 취약점이나 공격에 의한 침해사고 대응 방법이 요구된다.

침입 탐지 시스템이나 침입 차단 시스템이 네트워크를 통한 침입 대응 방법이었다면 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 삭제 시키지 못하게 함으로써 침해 대응을 할 수 있다. 침입자가 관리자 권한인 Administrator의 계정 권한을 획득하였더라도 보안 운영체제에서 사용되는 권한으로 중요한 시스템 파일이나 보안이 필요한 파일에 대한 접근을 차단한다.

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

2. 보안 운영체제

2.1 보안 운영체제

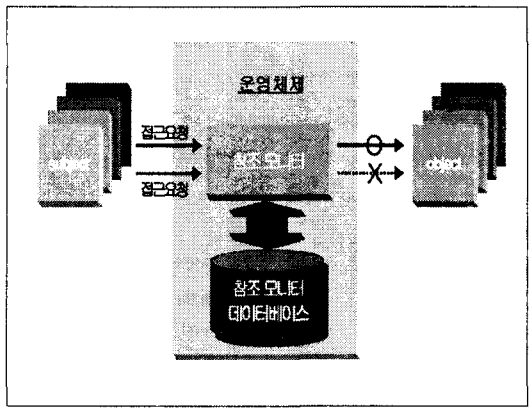
보안 운영체제란 기존의 커널에 보안 기능을 통합시킨 보안 커널(Secure Kernel)이 추가로 이식된 운영체제로 참조 모니터(Reference Monitor) 개념을 정의한 TCB(Trusted Computing Base)의 하드웨어, 펌웨어, 소프트웨어의 요소를 뜻한다[1].

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.



[그림 1] 보안 운영체제 개념도

참조 모니터 구현은 시스템 콜 엔트리(윈도우 NT의 경우 Service Table)에서 원래의 커널의 시스템 콜 주소를 저장한 후, 참조 모니터 함수의 주소를 시스템 콜 엔트리에 저장한다.

보안 커널의 보안 기능을 살펴보면 다음과

같다[3].

- 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 자유 재량에 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 관련 사건 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

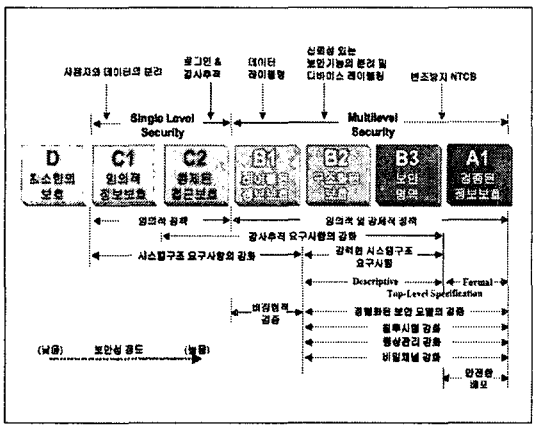
보안 커널의 구현 방법은 커널을 새로이 구현하는 방법과 모듈방식으로 만들어 모듈을 커널 속에 심는 방법(LKM: Loadable Kernel Module)등 두 가지방법이 있으며 통합 커널 기반 방식은 기존 운영체제의 모든 기능을 포함하며 API는 커널 서비스를 이용하게 하며 이와는 대조적으로 마이크로 커널 방식은 기존 통합 커널을 최소화하고 시스템을 최대한 모듈화 한다.

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 지침서. 컴퓨터 시스템의 보안을 효과적으로 평가하기 위해 6개의 기본 요구 사항을 정의하였으며, 그 사항을 만족시키는 수준에 따라 7가지의 평가 등급을 제시하였다[4].



[그림 2] TCSEC 평가 기준

3. 설계 및 구현

3.1 개발 환경

본 논문에서 구현한 모듈은 윈도우 2000 Professional을 기본 운영체제로 사용하였으며 . 또한 테스트를 위하여 Administrator와 구별하여 또 다른 계정을 하나 두었다. 프로그램 이름은 ACL (Access Control List)이라 두었으며 DLL 라이브러리의 이름은 ACLLib 라 하였다. 모듈에서 필요로 하는 함수는 Visual C++ 6.0을 이용하여 DLL로 작성하였고 GUI 는 델파이 6.0을 사용하였다.

3.2 구현

윈도우 환경에서의 파일에 대한 보안 객체를 읽고 쓰는 것은 aclapi를 이용하여 사용하면 된다. NTFS 파일 시스템에서는 파일에 보안 설명자를 두었으며 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.

다음은 ACE에 대한 읽는 함수의 일부분을 보여준다.

```
if (GetSecurityInfo(hFile, SE_FILE_OBJECT,
OWNER_SECURITY_INFORMATION |
DACL_SECURITY_INFORMATION, &pOwner,
NULL, &pDacl, NULL, (LPVOID *)&pSD) !=
ERROR_SUCCESS)
        return -1;

CloseHandle(hFile);

//소유자 정보
cbName = 0;
cbDomain = 0;
LookupAccountSid(NULL, pOwner, NULL, &cbName,
```

```
NULL, &cbDomain, &peUse);

Name = (char *)malloc(cbName);
Domain = (char *)malloc(cbDomain);

LookupAccountSid(NULL, pOwner, Name, &cbName,
Domain, &cbDomain, &peUse);

strcpy(Entry->Name, Name);
strcpy(Entry->Domain, Domain);

free(Name);
free(Domain);

nAce = 0;
//DACL 정보
GetExplicitEntriesFromAcl(pDacl, &nAce, &pEntry);
Entry->Count = (int)nAce;

for(i = 0; i < (int)nAce; i++) {
   cbName = 0;
   cbDomain = 0;
   LookupAccountSid(NULL,
pEntry[i].Trustee.ptstrName, NULL, &cbName,
NULL, &cbDomain, &peUse);

   Name = (char *)malloc(cbName);
   Domain = (char *)malloc(cbDomain);

   LookupAccountSid(NULL,
pEntry[i].Trustee.ptstrName, Name, &cbName,
Domain, &cbDomain, &peUse);

   strcpy(Entry->AceName[i], Name);
   Entry->AccessMode[i]
AccessModeToDword(pEntry[i].grfAccessMode);
   Entry->Permission[i]
pEntry[i].grfAccessPermissions;
```

위의 함수로부터 각 파일에 대한 소유자 정보와 도메인 정보, DACL 정보를 얻을 수 있다.

다음은 파일에 대한 ACE 객체를 쓰는 함수의 일부분이다.

```
GetNamedSecurityInfo(FileName,
                     SE_FILE_OBJECT,

DACL_SECURITY_INFORMATION,
                     NULL,
                     NULL,
                     &ExistingDacl,
                     NULL,
                     &psd);

BuildExplicitAccessWithName(&explicitaccess,
                            Trustee,
                            AccessMask,
                            option,
                            InheritFlag);

// add specified access to the object
SetEntriesInAcl(1,
                &explicitaccess,
                ExistingDacl,
                &NewAcl);

// apply new security to file
SetNamedSecurityInfo(FileName,
                     SE_FILE_OBJECT,  //  object
type

DACL_SECURITY_INFORMATION,
                     NULL,
                     NULL,
                     NewAcl,
                     NULL);
```

위의 소스는 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);

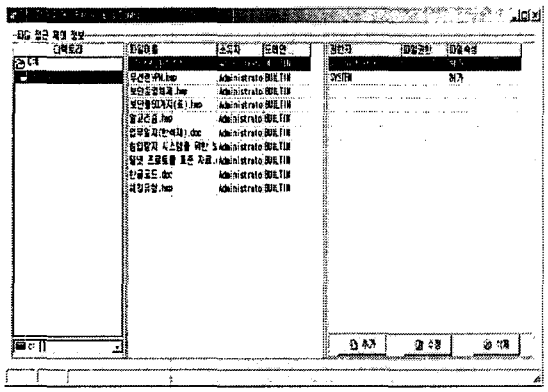
위에서 구현된 DLL을 통하여 ACL 프로그램을 작성하였으며 GUI 화면을 구성하는 프로그램은 델파이로 작성되었다.

다음은 델파이에서 ACE에 대한 정보를 읽는 함수를 호출하는 것을 보여준다.

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

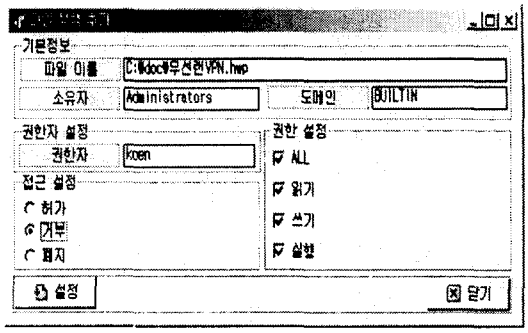
다음은 델파이에서 ACE에 대한 정보를 쓰는 함수를 호출하는 것을 보여준다.

```
Ret := SetSecurity(PChar(FileNameEdit.Text), PChar(TrusteeEdit.Text), Access, Permission);
```



[그림 3] ACL 프로그램

[그림 3]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.

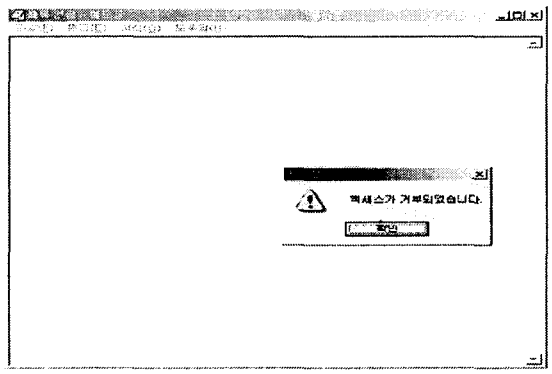


[그림 4] 객체추가 화면

[그림 4]는 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 5] 엑세스 거부 화면

해당 파일을 접근 시 엑세스가 거부되었다는 메시지창을 볼 수 있다.

4. 결 론

전 세계가 정보 통신의 급속한 발전으로 컴퓨터 및 인터넷의 사용이 급격히 증가함에 따라 정보 처리의 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 침해사고에서도 나올수 있으며 이는 보안 운영체제로써 대응할 수 있다.

본 논문에서는 이러한 보안 운영체제에서 사용되는 파일 정책 제어 모듈을 설계 및 구현함으로써 보안 운영체제에서 사용할 수 있도록 하였다.

본 논문에서 구현된 모듈은 보안 운영체제를 위한 파일 정책을 제어함으로써 시스템에 대한 보안을 강화할 수 있다.

참고문헌

[1] 김재명, 홍기융, 홍기완, "Secure OS 보안정책 및 메커니즘", 정보보호학회지, 13권 4호, 2003

[2] 이홍섭, 이철원, 이정효, 박정호, "정보통신 기반구조 보호를 위한 보안 커널 개발 동향", 정보보호학회지, 8권 4호, 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

한 석 재

2002년 ~ 현재 한남대학교 컴퓨터공학과 박사과정

김 완 경

2003년 ~ 현재 한남대학교 컴퓨터공학과 석사과정

소 우 영

1992년 ~ 현재 한남대학교 컴퓨터공하과 교수