Linux 시스템의 보안커널에 관한 연구

한명묵* · 이준환**

요 약

보안 운영체제인 SELinux는 임의 접근제어만을 제공하는 기존의 리눅스 커널에 리눅스 보안 모듈을 이용하여 강제적인 접근제어를 구현한 보안 시스템이다. 그러나 시스템 침입이 발생하였을 때, 침입 탐지 및 로깅만으로는 부족하다. 본 논문에서는 동적인 접근 제어를 활용하여 접근 위반 탐지 및 권한 제한을 수행하는 SELinux 보안 커널에 대해 제안한다. 시스템의 불법적인 접근이 발생할 시에 보안 검사 기능을 통해 이를 탐지하고 권한 제한 기능으로 동적으로 침입자에 대한 시스템의 권한을 변경하여 재침입을 방지할 수 있는 시스템을 실험을 통해 구현하였다.

A Study on Security Kernel of Linux System

Myung-Mook Han* · Jun-Hwan Lee**

ABSTRACT

SELinux, security operating system, is the security system which implements mandatory access control using linux security module on the traditional linux kernel supporting discretionary access control. But intrusion detection and logging are lacked when system intrusions are happened. This study proposes a SELinux security kernel which performs detection of access violation and privilege restriction using dynamic access control. It detects the intrusion using security check when the abnormal access of system is happened, and dynamically changes the system privilege for the intruder through privilege restriction. Finally we prevent reintrusion and explain the result of experiment.

Key words : SELinux, Security Kernel, Dynamic Access Control

* 경원대학교 IT대학
** (주)시큐브

1. 서 론

기존의 운영체제와 애플리케이션에서 메모리 공격, 레이스 컨디션과 같은 보안상의 결함이 존재하여 시스템 외부 또는 내부에서 사용자 권한이나 관리자 권한을 획득할 수 있는 많은 취약점이 존재한다. 실제로 이러한 취약점을 이용하여 많은 시스템이 해킹당해 왔다. 이러한 문제점을 운영체제 수준에서 원천적으로 해결할 수 있는 보안 운영체제가 연구, 개발되었다[1].

미국 국가안전보장국에서 주도로 개발한 오픈소스 보안 운영체제인 SELinux[2]는 임의 접근 제어(Discretionary Access Control, DAC)만을 제공하는 기존의 리눅스 커널에 리눅스 보안 모듈(Linux Security Modules, LSM)프레임워크[4]를 이용하여 강제적인 접근 제어(Mandatory Access Control, MAC)를 구현한 커널기반의 보안 시스템이다[3].

SELinux에서 제공하는 기능을 응용하여 실제 침입에 대한 침입 차단 시스템과 유사한 역할을 수행할 수도 있다[4]. 그러나 침입을 차단하였을 때 시스템을 보호하기 위해 침입자의 시스템 사용 권한을 없애거나 최소화할 수 있도록 접근 제어 정책을 동적으로 수정한다면 침입자의 침입을 탐지 및 차단하고 차후의 침입 시도까지 미리 예방할 수 있을 것이다.

본 논문에서는 동적인 접근 제어를 활용하여 접근 위반 탐지 및 권한 제한을 수행하는 SELinux 보안 커널에 대해 설명한다. 시스템의 불법적인 접근이 발생할 시에 보안 검사 기능을 통해 이를 탐지하고 권한 제한 기능으로 동적으로 침입자에 대한 시스템의 권한을 변경하여 재침입을 방지할 수 있는 시스템을 구현하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 SELinux에 대해 기술하고 제 3장에서는 제안하는 동적인 접근 제어를 활용하여 접근 위반 탐지 및 권한 제한을 수행하는 SELinux 보안 커널의 설계를 기술한다. 제 4장에서 실험과 성능 테스트를 하고, 마지막으로 결론으로 구성된다.

2. SELinux

2.1 주요 기능

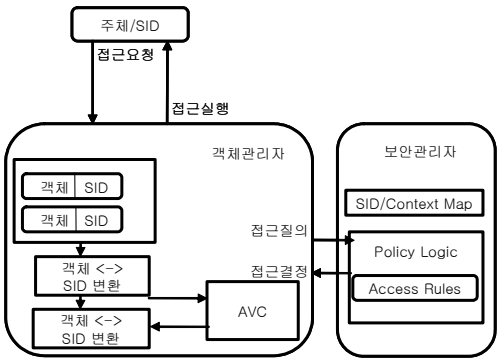
- 자원에 대한 여러 가지 접근 통제 수행 : 유형 강화(Type Enforcement TE), 역할 기반 접근 통제 (Role Based Access Control, RBAC), 다중 등급 보안 (Multi Level Security, MLS)와 같은 접근 통제 정책을 지원한다.
- 커널 보호 : 공격자에 의한 커널 수정을 방지하기 위해 모듈 삽입이나 커널에 대한 직접 쓰기에 대한 보호를 할 수 있다.
- 침입에 대한 피해 범위의 감소 : 프로세스 실행시 필요한 최소한의 권한만 할당하여 피해 발생시 프로세스나 시스템의 끼치는 영향을 최소화한다.
- 감사 기능 : 시스템에 일어나는 보안 규칙 위배에 대한 정보는 파일로 기록되어 감사 자료로 활용할 수 있다.

이러한 기능들은 리눅스 커널 2.6버전부터 제공되는 리눅스 보안 모듈(Linux Security Module, LSM) 프레임워크를 이용하여 삽입 가능한 커널 모듈로 개발되었고, 모듈 삽입을 통해 기존에 제공되던 DAC 접근 제어 기능들을 대체한다[5].

2.2 구조

SELinux는 Flask 프로젝트[6]의 후속 프로젝트로 Flask 구조를 리눅스 커널에 이식한 시스템이다. (그림 1)은 SELinux의 구조를 간략하게 보여준다.

객체 관리자(Object Manager)는 시스템에 제어 동작을 제공하고, 보안 서버에서 특정 보안 정책에 대한 보안 결정이 이루어지면 보안 결정에 대한



(그림 1) SELinux의 구조

수행을 담당한다. 또한 AVC(Active Vector Cache) 관리 정책의 적용과 변경 등의 기능을 제공한다.

보안 서버(Security Server)는 보안 정책을 결정하고, SID(Security ID)와 보안 컨텍스트(Security Context)간의 매핑을 유지하고, 새로 생성되는 객체에 SID를 부여한다

2.3 문제점

침입자는 버퍼 오버플로우 기법과 같은 메모리 조작을 통하여 프로세스에 시스템 쉘을 실행하는 등의 악성 코드를 삽입할 수 있다. 그러나 SELinux가 제어하는 시스템에서는 삽입한 악성 코드의 실행에 필요한 권한을 획득할 수 없어서 이와 같은 공격으로 인한 피해를 막을 수 있다. 또한 SELinux는 이러한 불법 접근에 대한 감사 정보를 기록할 수 있어서 감사 자료로 사용할 수 있다.

SELinux의 기능을 응용하여 비정상적인 접근에 대한 로그를 기록할 수 있다. 그렇지만 커널의 로그 정보에는 침입에 대한 증거가 될 수 있는 비정상적인 접근에 대한 로그와 다수의 일반적인 시스템 관련 로그가 같이 저장되고, 어떠한 종류의 로그가 위험한지 커널내에서 판단할 수 없다. 따라서 침입에 대한 좀 더 효과적인 대응을 할 수 있도록 어떠한 종류의 접근 정보가 위험한지를 커널스스로 판단할 수 있도록 할 수 있게 하는 기능의 구현이 필요하다.

3. 제안하는 보안커널

3.1 기본 구조

본 논문에서 제안하는 시스템에서는 SELinux의 접근 제어 언어를 확장하여 어떠한 종류의 접근 정보가 위험한지를 판단하여 Conditional Policy와 연동할 수 있도록 불리언 변수를 설정하는 '보안 검사'와 그에 대한 대응으로 Conditional Policy의 조건부 정책을 이용하여 일반 접근 제어 리스트(Normal Access Control List)를 제한된 접근 제어 리스트(Restricted Access Control List)로 변경하는 '권한 제한' 기능을 추가하여 동적으로 접근 제어 정책을 적용할 수 있는 시스템을 설계하였다.

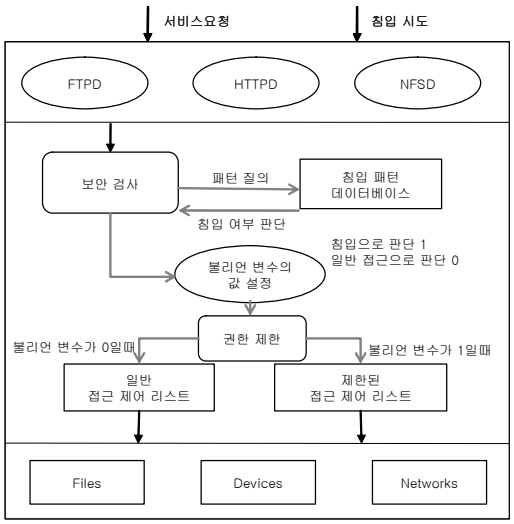
(그림 2)에서는 비정상적인 접근이 발생하였을 시에 피해를 예방하기 위해 "보안 검사"와 "권한 제한"을 수행하여 침입자에게 제한된 접근 제어가 적용되는 과정을 보여준다.

3.1.1 보안 검사

보안 검사는 어떠한 종류의 접근 정보가 위험한지를 침입 탐지 데이터베이스를 참조하여 판단하게 된다. 침입 탐지 데이터베이스에 위험하다고 판단할 수 있는 접근 정보를 기술하기 위하여 SELinux의 접근 제어 언어에 'detect'란 문법을 추가하였다. 이 문법은 SELinux의 "allow" 문법과 사용 방법이 유사하다.

(그림 3)은 침입자가 ftp 데몬을 버퍼 오버플로우 방식으로 공격하여 시스템 쉘을 실행하는 공격을 정의한 구문이다.

만약 'detect' 구문에 해당하는 접근이 발생하면 '권한 제한'을 통하여 불리언 변수의 값이 true로 설정된다.



(그림 2) 강화된 보안 커널의 동작 과정

```
bool ftpd_write_forbid false;
detect ftpd_t shell_exec_t:file { execute } { ftpd_write_forbid true }
```

(그림 3) detect 구문의 정의

이러한 보안 검사 기능에서 탐지할 수 있는 것은 다음과 같다.

- 비정상적인 쉘 실행, 관리자 권한 획득 시도.
- 설정파일, 로그파일등의 변조나 삭제.
- 자원들에 대한 허가되지 않은 접근

3.1.2 권한 제한

제 3.1.1절에서는 보안 검사를 통하여 모든 접근에 대한 접근 위반 여부를 판단하였다. 만약 그 과정에서 접근 위반이 탐지되어지면, 해당 침입에 대한 피해를 최소화하기 위해 “권한 제한”을 실행하여 제한된 접근 제어 정책을 적용한다. 이는 보안 검사의 결과에 따라서 해당 접근에 대한 접근 정책이 동적으로 달라질 수 있다는 것을 의미한다.

권한 제한은 침입이 발생하였을 때 침입자에게서 시스템에 대한 권한을 축소하는 기능을 수행한다. 제한된 접근 제어 정책의 정의를 위하여 보안 검사과정에서 그 결과에 따라 값이 true, false로 설정되는 불리언 변수와 SELinux의 Conditional Policy를 이용하게 된다.

(그림 4)는 권한 제한을 정의한 조건 표현문이다.

```
bool ftpd_write_forbid false;
detect ftpd_t shell_exec_t:file { execute } { ftpd_write_forbid true }
if (ftpd_write_forbid)
{
  allow ftpd_t user_home_t:file { read, getattr, lock, ioctl };
}
else
{
  allow ftpd_t user_home_t:file { create, ioctl, read, getattr, lock, write,
                                  setattr, append, link, unlink, rename };
}
```

(그림 4) 권한 제한을 정의한 조건 표현문

접근 위반이 탐지되기 전에는 불리언 변수의 값이 false가 되어 파일에 대한 쓰기, 읽기, 삭제 등의 기능을 수행할 수 있지만 보안 검사에 의해 접근 위반이 탐지되면 불리언 변수의 값은 true로 설정되어 그에 해당하는 접근 정책이 적용되게 된다. 따라서 파일에 대한 생성, 쓰기, 삭제등의 권한은 삭제되고 읽기와 같은 최소한의 권한만 허가되어 ftp 데몬의 업로드를 제한하게 된다.

4. 실험 및 성능 테스트

강화된 보안커널 구현을 위하여 리눅스 커널 소스와 libsepol, checkpolicy 소스를 수정하였다. 사용한 커널 버전과 libsepol, checkpolicy 버전은 다음과 같다.

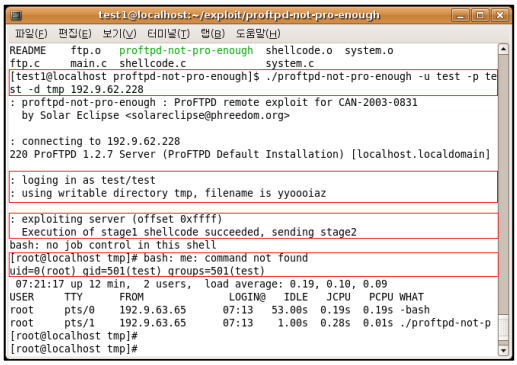
- Linux kernel 2.6.12
- Checkpolicy-1.24
- Libsepol-1.6

4.1 실험

이번 절에서는 취약점이 있는 ftp 서버 프로그램을 설치하고 버퍼 오버플로우 공격을 하여 원격에서 루트 권한을 획득할 수 있는 공격을 테스트한다. 테스트에 사용한 ftp 서버 프로그램은 proftpd-1.2.7 버전이고 이 버전은 ascii upload 버퍼 오버플로우 공격에 취약점이 존재한다.

4.1.1 일반 리눅스 시스템

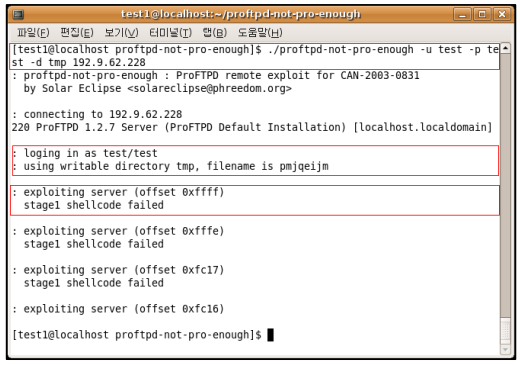
(그림 5)는 일반 리눅스 시스템에서 ftp 서버를 실행하고 원격에서 익스플로잇을 통하여 루트 권한을 획득하는 과정을 보여준다. 시스템에 대한 루트 권한을 획득하기 위하여 익스플로잇을 실행하여 원격 FTP 서버에 버퍼 오버플로우 공격을 수행한다. 공격을 수행하는 과정에서 공격에 필요한 ascii 파일을 업로드 한다. 이 때 공격자는 ftp 서버에 공격 파일을 업로드하기 위한 쓰기 권한이 필요하다. 다음 단계에서 버퍼 오버플로우 공격으로 시스템 쉘을 실행한다. 그 결과 공격자에 의해 원격으로 FTP 서버의 루트 권한 쉘을 획득한 것을 볼 수 있다.



(그림 5) 루트 권한 획득

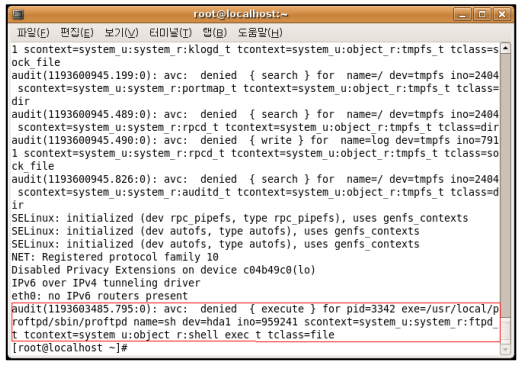
4.1.2 SELinux 시스템

SELinux가 적용된 서버 시스템에서 제 4.2.1절과 같은 공격을 시도하였다. (그림 6)은 SELinux 서버 시스템에서 버퍼 오버플로우 공격을 시도한 과정을 보여준다. 공격시에 ascii 파일 업로드에는 성공하였지만(쓰기 권한 필요) 루트 권한을 얻기 위해 버퍼 오버플로우 공격을 통해 쉘을 실행하는 과정은 실패하는 것을 볼 수 있다.



(그림 6) 루트 권한 획득 실패

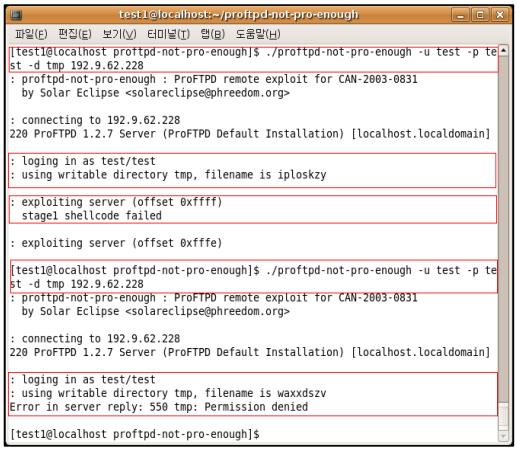
(그림 7)은 침입자가 버퍼 오버플로우 공격을 실패한 후 시스템이 남긴 로그를 보여준다. pid 3285를 가리키는 /usr/local/proftpd/sbin/proftd 프로세스가 시스템 쉘을 실행하려 하였지만 해당 프로세스는 ftpd_t의 타입으로 지정되어 있어서 shell_exec_t 타입의 파일을 실행하는 것을 실패하였다는 것을 볼 수 있다.



(그림 7) 침입 탐지 로그

4.1.3 강화된 보안 커널 시스템

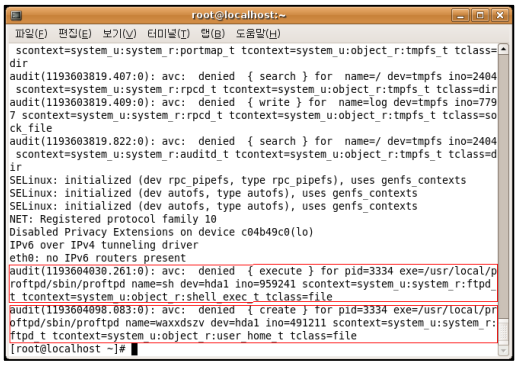
마지막으로 본 논문에서 제안한 강화된 보안 커널 시스템상에서 같은 공격을 시도하였다. 그 과정은 (그림 8)과 같다. 먼저 공격시에 ascii 파일 업로드에 성공하였지만 루트 권한을 얻기 위해 버퍼 오버플로우 공격을 통해 쉘을 실행하는 과정은 실패하는 것을 볼 수 있다.



(그림 8) 공격과 재공격의 실패

여기서 SELinux 시스템의 경우와 다른 점은 공격을 재시도 하였을 때이다. 첫 번째 공격 시도시 보안 커널 시스템에서 이를 탐지하고 쉘코드를 실행하는 것을 차단하는 과정에서 '보안 탐지'가 실행되어지고 그에 따라 설정한 불리언 변수의 값이 true로 변함에 따라 침입자의 권한이 제한되어 파일을 읽을 수는 있지만 새로 생성하거나 삭제하는 등의 행동을 할 수가 없다.

(그림 9)는 강화된 보안 커널 시스템이 접근위반을 탐지하고 권한 제한을 실행한 로그를 보여준다. 첫 번째 공격 시도 이후 침입자가 파일 생성에 실패한 것을 볼 수 있는데, 이는 보안 커널이 해당 공격을 탐지 한 후에 '보안 검사' 과정과 '권한 제한' 과정을 통하여 침입자의 권한을 제한하였기 때문이다.



(그림 9) 위반행위 탐지, 권한제한의 로그

4.2 성능 테스트

IDS시스템의 성능 테스트를 위하여 Unix/Linux용 벤치마크 유틸리티인 LMBENCH[7]를 이용하여 성능 테스트를 진행하였다. IDS를 적용한 커널과 그렇지 않은 커널과의 성능 비교를 위하여 성능 테스트는 다음 커널상에서 진행되었다.

- Linux Kernel 2.6.12 [SELinux 미적용]
- Linux Kernel 2.6.12 [SELinux 적용]
- Linux Kernel 2.6.12 [강화된 SELinux]

LMBENCH 유틸리티를 이용하여 파일 연산 시스템 콜 호출과, 프로세스 생성에 대한 성능 테스트를 수행하였다. 테스트 항목은 다음과 같다.

- 파일 연산 시스템 콜(lat_syscall)
  - read(), write(), stat(), fstat(),
  - open(), close()
- 프로세스 생성 (lat_proc)
  - fork+exit(), fork+exec(),
  - fork+shell

이 벤치마크는 다음의 시스템에서 수행되었다.

- CPU : AMD Athlon XP 2500(1.8Ghz)
- RAM : 1GB(512MB * 2)

• DISK : 60GB EIDE DISK

벤치마크의 결과는 다음 <표 1>과 같다.

〈표 1〉 벤치마크 테스트

		SELinux 미적용	SELinux	강화된 SELinux
lat_syscall [μsec]	read	0.48	0.52	0.56
	write	0.48	0.77	0.80
	stat	1.13	2.01	2.06
	fstat	0.58	0.81	0.85
	open/close	1.87	3.14	3.34
lat_proc [μsec]	fork+exit	1551.44	1578.23	1578.30
	fork+execve	3929.16	4243.88	4331.91
	fork+/bin/sh	7300.21	7444.11	7446.23

5. 결 론

본 논문에서는 SELinux의 기능을 확장하여 리눅스 커널의 강화된 보안 시스템을 제안하였다. 기존의 네트워크 기반 침입 탐지 시스템과 다르게 시스템 성능에 끼치는 영향이 적고, 기존의 서명으로 한 탐지 방법과 다르게, 시스템에 대한 허가되지 않은 접근 및 불법적인 권한 획득과 같은 공격을 효과적으로 탐지하고 막을 수 있을 뿐만 아니라 침입자가 침입 시도시 시스템에 대한 권한을 축소함으로서 차후의 침입 시도까지 미리 예방할 수 있다는 것을 볼 수 있다. 향후 침입에 대한 실시간 감사 자료 생성 및 더욱 세밀한 보안 제어와 속도 개선에 대한 지속적인 연구를 진행하고자 한다.

참 고 문 헌

[1] Charles P. Pereeger, “Security In Computing”, Prentice Hall, 2nd ED, 1997.
[2] Security-Enhanced Linux, http://www.nsa.gov/selinux.
[3] Linux security modules, http://lsm.immunix.org.
[4] Frank Mayer, Karl MacMillan, and David Caplan, “SELinux by Example : Using Security Enhanced Linux”, Prentice Hall, 2006.
[5] T. Horie, K. Masumoto, Y. Miyamoto, T. Harada, and K. Tanaka, “Dynamic Access Control for Operating System Kernel”, Joho Shori Gakkai Kenkyu Houkoku, Vol. 126, pp. 25-30, 2003.
[6] The Flux Research Group, http://www.cs.utah.edu/flux/html/flash.html.
[7] LMBENCH, http://www.bitmover.com/lmbench/.

한 명 묵

1980년 연세대학교 공과대학 (공학사)
1987년 뉴욕공과대학교 컴퓨터공학과(공학석사)
1997년 오사카시립대학교 정보공학부(공학박사)
1998년～현재 경원대학교 IT대학 부교수

이 준 환

2001년 경원대학교 경제학과 입학
2005년 경원대학교 컴퓨터공학과 졸업
2005년 경원대학교 일반대학원 전자계산학과 입학
2008년 경원대학교 일반대학원 전자계산학과(공학석사)