제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

리눅스 시스템의 보안 강화를 위한 LKM(Loadable Kernel Module) 설계 및 구현

김익수*, 김명호*
*숭실대학교 컴퓨터학과

e-mail:skycolor@ss.ssu.ac.kr

A Design and Implementation of Loadable Kernel Module for Enhanced Security on Linux System

Ik-Su Kim*, Myung-Ho Kim*
*School of Computing, Soong-Sil University

요 약

공격자는 시스템에 침입하기 위해 취약점을 수집하며 여러 공격방법을 통해 루트권한을 획득하게 된다. 루트권한을 획득한 공격자는 공격 시스템에 루트킷을 설치하여 침입에 대한 흔적을 숨기고 차후 침입을 위한 백도어를 남기게 되는데 최근 등장한 커널 기반의 루트킷은 시스템에 대한 침입 탐지를 어렵게 하고 있다. 이러한 공격에 대응하기 위해 침입탐지 및 차단을 위한 보안 시스템들이 많이 개발되어 왔지만 공격자들은 보안 시스템들을 우회하여 시스템에 침입하고 있다.

본 논문에서는 루트권한을 획득한 공격자의 불법행위를 막기 위해 시스템 보안 강화 LKM을 설계, 구현하여 중요 파일의 변조와 루트킷의 설치를 막고 공격자의 불법행위를 관리자에게 실시간으로 알릴 수 있는 방법을 제안한다.

1. 서론

시스템 공격자는 목표 시스템에 침입하기 위해 취약점을 수집하며 취약점을 이용한 여러 공격방법을 통해 루트권한을 획득하게 된다. 루트권한을 획득한 공격자는 자신의 침입 흔적을 숨기고 손쉽게 시스템에 다시 접근할 수 있도록 루트킷이라 불리는 백도어 및 트로이잔 프로그램 패키지를 설치하며 이러한 루트킷은 인터넷을 통해 쉽게 구할 수 있어 시스템 보안의 커다란 위협이 되고 있다[1]. 일반적인 루트킷들은 관리자의 세심한 로그 분석과 시스템의 현재 상태들을 조사함으로써 탐지가 가능하지만 커널 기반의 루트킷은 탐지가 매우 어렵기 때문에 공격자는 자신이 설치한 루트킷을 이용하여 손쉽게 시스템에 재침입하게 된다. 이와 같은 불법침입 행위를 막기 위해 개발된 보안 시스템들은 공격자의 침입을 어렵게 하지만 완벽하게 침입을 막을 수 없으며 루트권한을 획득한 공격자에 대해 어떠한 방어능력도 갖추지 못하고 있다.

본 논문에서는 기존의 보안 시스템들이 가지는 문제점을 극복하기 위해 보안 LKM을 설계 및 구현하여 공격자의 루트권한 획득에 따른 불법행위를 막고 실시간으로 관리자에게 불법행위를 알릴 수 있는 방법을 제안한다.

2. 관련연구

여러 공격방법을 통해 시스템 침입에 성공한 공격자는 차후에 쉽게 재침입하기 위해서 루트킷을 설치하게 된다. 2장에서는 공격자의 일반적인 시스템 침입방법과 시스템 침입에 성공한 공격자가 설치하는 루트킷, 공격자로부터 시스템을 보호하기 위해 개발

된 여러 보안 시스템에 대해 살펴보도록 한다.

2.1 일반적인 시스템 침입 방법

일반적으로 시스템 공격자는 (표 1)과 같은 방법들을 통해서 목표 시스템에 침입하게 된다.

취약점 정보 수집	포트 스캐너와 취약점 검색 도구를 사용하여 취약점 정보를 수집
버퍼오버플로우	SETUID 루트 프로그램이 수행중일 때 버퍼 오버플로우를 일으켜 루트권한 획득
사용자 도용	네트워크 상에 떠도는 패킷을 캡쳐하여 ID와 패스워드를 도용

(표 1) 시스템 공격 유형

취약점 정보를 수집하기 위해 공격자는 포트 스캐너와 취약점 검색도구를 사용하여 취약점을 수집하게 되며 취약점으로 노출된 데몬에 버퍼오버플로우를 일으켜서 루트쉘을 얻는다. 그리고 다른 시스템에 침입하기 위해서 시스템을 스니핑하여 사용자의 ID와 패스워드를 얻어낸다.

2.2 루트킷

공격자는 시스템을 침입한 후 침입 흔적을 제거하고 차후에 재침입하기 위해 루트킷이라 불리는 백도어와 트로이잔 프로그램 패키지를 설치하게 된다. 이러한 루트킷은 크게 애플리케이션 수준의 루트킷과 커널 기반의 루트킷으로 분류된다. 애플리케이션 수준의 루트킷에는 lrk(Linux Rootkit), t0rn Kit과 같은 루트킷이 널리 알려졌으며 이 루트킷이 시스템 상에 설치되면 공격자가 원하는 파일들과 프로세스들을 숨길 수가 있다. 즉, 공격자가 이들 루트킷에 포함되어 있는 변조된 ls, ifconfig, find, ps와 같은 프로그램들로 원래의 시스템 프로그램을 대체하게 되면 다른 사용자들은 특정 파일과 프로세스를 볼 수가 없게 된다. knark과 같은 LKM 루트킷은 기능상 애플리케이션 수준의 루트킷과 비슷하지만 메모리 상에 커널 모듈 형태로 존재하게 된다. LKM 루트킷은 정상적인 시스템 콜을 가로채서 공격자가 만든 시스템 콜 함수가 실행되도록 함으로써 특정 파일과 프로세스들을 숨기게 된다.

2.3 루트킷 탐지도구

일반적으로 루트킷을 탐지하기 위해서는 로그파일의 분석, 변조되지 않은 시스템 프로그램과 의심되는 프로그램의 비교 분석을 통해 어렵지 않게 찾을 수 있으며 루트킷 탐지 도구인 chrootkit을 사용하여 루트킷의 설치여부를 알 수 있다. 그러나 LKM 루트킷은 커널의 시스템 콜 함수를 가로채기 때문에 시스템 명령으로는 루트킷을 발견하기가 어려우므로 Kernel 정보를 분석할 수 있는 도구인 kstat나 carbonite를 사용하여 루트킷을 탐지해야 한다[2].

루트킷 탐지도구는 공격자의 시스템 침입이 이루어진 이후에야 루트킷의 설치 여부를 알 수 있으며 공격자가 시스템을 침입하고 루트킷을 설치하지 않는다면 루트킷 탐지도구는 무용지물이 된다. 그리고 이러한 루트킷 탐지도구들은 루트킷 설치를 막을 수 없고 공격자의 루트권한 획득을 통한 불법행위 문제를 해결할 수가 없다.

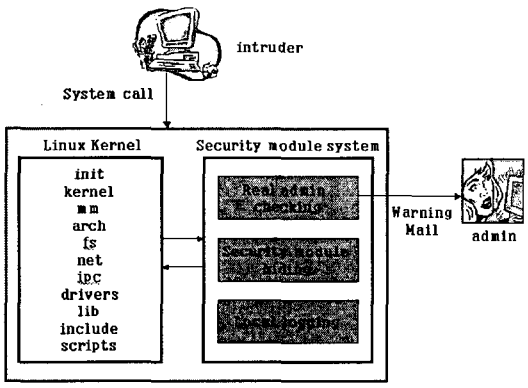
2.4 침입탐지 및 차단 시스템

최근 공격자들의 불법침입에 따른 피해를 최소화하기 위해 방화벽과 침입탐지시스템이 개발되어 왔다. 방화벽은 패킷필터링을 통해서 외부로부터의 불법적 트래픽 유입을 막으며 허가되고 인증된 트래픽만을 허용함으로써 내부 네트워크에 있는 전산자원을 보호하고 내부정보들이 외부로 유출되는 것을 방지하는 기능을 가진다. 침입탐지시스템은 여러 침입방법에 대한 규칙을 자체적으로 내장하여 침입행동을 실시간으로 감지, 제어할 수 있는 기능을 제공한다. 이러한 방화벽은 사전에 미리 IP 주소나 포트 등을 등록하여 시스템 접근을 허용하거나 막는 정적인 방법이며 침입탐지시스템은 침입에 대한 대응이 실시간 차단이 아니라 공격자와 관리자에게 경고메일을 보내는 방법이므로 침입탐지 이후에 생기는 불법 행동에 대한 커다란 위험이 따르게 된다.

3. 시스템 강화를 위한 보안모듈 설계 및 구현

앞서 기술했듯이 공격자는 공격시스템에서 루트권한을 획득한 후 공격도구와 흔적을 숨기기 위해 커널 기반의 루트킷을 설치하게 된다. 따라서 본 논문에서는 루트권한을 획득한 공격자의 불법행위를 막기 위한 모듈을 설계, 구현한다.

공격자의 시스템 공격에 대응하기 위한 보안모듈의 구성은 (그림 1)와 같다.



(그림 1) 시스템 보안 강화를 위한 모듈 구성도

일단 보안 모듈이 시스템에 로드되면 시스템 사용자의 모든 명령에 대해 Real admin checking 부분에서 사용자가 요구하는 명령이 적합한지 결정을 하게 된다. 일반 사용자의 시스템 명령은 모두 허가되지만 루트 권한의 사용자 명령은 제한을 하게 된다. 즉, 모듈을 로드한 루트권한의 사용자는 시스템의 모든 명령을 수행할 수 있지만 그 외의 루트권한 사용자의 명령에 대해서는 서비스를 거부하게 된다. 실제 시스템 관리자가 모듈을 로드하게 되면 모듈에서는 모듈을 로드한 터미널의 tty값을 저장하게 되며 이후의 모든 루트권한의 명령은 tty값과 일치할 경우에만 시스템 명령을 서비스하게 되고 tty값이 틀린 경우에는 서비스를 거부하게 된다. 로컬 혹은 리모트 접속을 통해 시스템 관리자가 루트권한의 명령을 수행하기 위해서는 보안 모듈의 패스워드를 입력함으로써 모듈내에 저장된 tty값이 시스템 관리자의 현재 터미널에 대한 tty값으로 변경되게 된다[3, 4, 5]. 커널이 시스템 명령을 처리하기 위해서는 execve 시스템 콜 함수를 호출하게 되는데 이 시스템 콜 함수 내부에서 우선적으로 시스템 보안 강화 모듈이 저장된 tty값과 서비스를 요청한 터미널의 tty값을 비교하게 된다. 만일 비교된 값이 일치하게 되면 execve 시스템 콜은 해당 명령을 수행하기 위해 커널내의 함수인 do_execve 함수를 호출하여 서비스를 수행하고 값이 일치하지 않을 경우 execve 시스템 콜 함수는 do_execve 함수의 호출없이 에러를 리턴하게 된다.

보안 모듈은 자체 모듈에 관한 정보를 외부에 숨기기 위해서 lsmod 명령에 의한 결과와 /proc 파일 시스템의 정보를 거짓으로 알리게 된다. 즉, 실제 모듈명을 다른 이름으로 변조함으로써 다른 사용자가 보안 모듈을 찾거나 언로드 하는 것을 막게 한다.

```
int init_module()
{
    orig_sys_write=sys_call_table[__NR_write];
    sys_call_table[__NR_write]=sec_sys_write;
}
void cleanup_module()
{
  sys_call_table[__NR_write]=orig_sys_write;
}
int sec_sys_write()
{
    char hide[]="sec_module";
    ......
    if(strstr(kernel_buf, hide)!=NULL)
    {
       //특정 스트링을 변조하는 부분
       ......
    }
    ......
    return orig_sys_write(fd, buf, count)
}
```

(그림 2) 특정 스트링을 변조하기 위한 코드

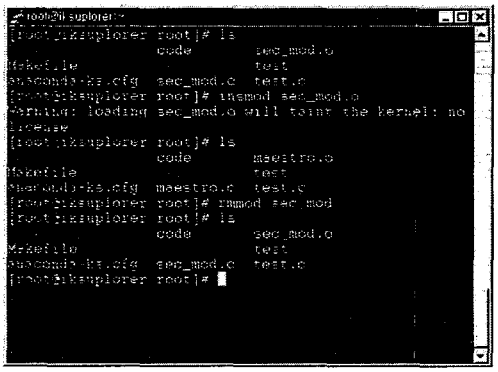
(그림 2)은 보안 모듈에 대한 정보를 변조하기 위한 코드로서 시스템 콜 함수의 주소 리스트를 가지고 있는 sys_call_table에서 write 시스템 콜 함수의 주소를 보안모듈의 시스템 콜 함수의 주소로 변경하게 된다. 커널은 ls와 lsmod와 같은 명령을 서비스하기 위해 write 시스템 콜 함수를 호출하게 되는데 보안모듈이 로드된 후에는 이 서비스를 위해 보안모듈 내의 write 시스템 콜 함수를 호출하게 된다. write 시스템 콜 함수가 호출되면 함수의 인자값과 변조할 스트링을 비교하여 서로 일치할 경우 인자값을 변조함으로써 일반 사용자들은 터미널 상에서 보안모듈에 관한 정보를 볼 수 없게 된다.

실제 관리자 외의 다른 사용자가 루트권한으로 명령을 내리게 되면 침입에 대한 불법행위로 간주하고 그에 해당하는 적절한 로그를 남기게 되며 실제 시스템 관리자에게 메일을 전송함으로써 실시간으로 침입행위를 알 수 있도록 한다[6].

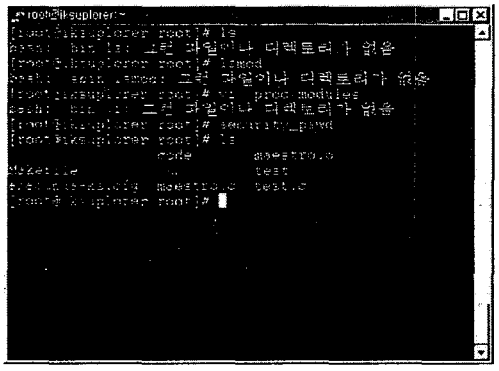
기존의 보안 시스템들은 침입을 탐지하고 외부로의 불법적인 접근을 막기 위한 솔루션을 제공했다. 하지만 이들 보안 시스템의 문제점은 공격자가 시스템 침입에 성공하고 루트권한을 획득했을때에는 어떠한 대응도 할 수 없다는 것이다. 최근 발생하는 공격들이 버퍼오버플로우에 의한 루트쉘의 획득이기 때문에 본 논문에서 구현한 시스템 보안 강화 모듈은 이러한 문제점을 극복할 수 있다.

4. 실험

시스템 보안 강화 모듈의 실험은 Linux kernel 2.4.13에서 수행되었다.



(그림 3) 실제 시스템 관리자의 터미널



(그림 4) 시스템 관리자의 외부 로긴 터미널

(그림 3)에서 볼 수 있듯이 실제 시스템 관리자가 시스템 보안 강화 모듈을 로드하게 되면 관리자는 모든 명령에 대한 서비스를 받을 수 있다. 하지만 (그림 4)에서와 같이 다른 터미널로 로긴한 후에는 루트권한의 모든 명령에 대해 서비스를 받을 수가 없다. 관리자가 외부에서 혹은 다른 터미널에서 시스템을 사용하기 위해서는 시스템 보안 모듈의 패스워드를 입력하게 되면 루트권한으로 시스템 명령을 정상적으로 수행할 수가 있다.

5. 결론 및 향후과제

최근 시스템 공격자들은 시스템의 취약점을 이용한 버퍼오버플로우를 통해 루트쉘을 얻는다. 루트권한을 얻은 공격자는 시스템에 남겨진 자신의 공격흔적을 제거하고 차후의 공격을 위한 루트킷을 설치하게 되며 더 나아가서는 시스템 자체를 파괴하게 된다.

본 논문에서는 루트권한을 획득한 공격자의 불법행위를 막기 위한 시스템 보안 강화 LKM을 구현하여 이러한 공격에 효과적으로 대응할 수 있다. 일단 시스템 보안 강화 LKM이 시스템 상에 로드되면 루트권한을 획득한 공격자는 루트권한으로 시스템 명령을 수행할 수 없으며 시스템 명령 수행시에 실시간으로 실제 관리자에게 메일이 전달되기 때문에 관리자는 공격자의 불법행위를 즉각적으로 탐지할 수 있게 된다.

현재 구현된 시스템 보안 강화 모듈은 아직 공격자에 대한 IP를 추적하는 기능이 없기 때문에 공격자에 대해 직접적인 대응을 할 수 없다. 따라서 향후에는 공격자의 IP를 추적하여 적절히 대응하는 기능을 구현할 것이다.

참고문헌

[1] 이현우, 김영직, 전숙 "UNIX 피해시스템 분석" (2002)

[2] 이계찬, 이현우, "LKM 루트킷 탐지" (2002)

[3] Alessandro Rubini, Jonathan Corbet, "Linux Device Drivers"

[4] Ori Pomerants, "Linux Kernel Module Programming Guide", (1999)

[5] 이호, "Linux Kernel Programming", (2000)

[6] 이호, "Advanced Module Programming" (2001)

[7] Daniel P. Bovet, Marco Cesati, "Understanding The Linux Kernel"

[8] http://linuxkernel.org

[9] http://packetstormsecurity.org

[10] http://www.securitymap.net