LIDS 를 이용한 리눅스 시스템 보안

o *정수진, ***김정녀, *박승규, *최경희 , **정기현
*아주대학교 정보 및 컴퓨터 공학부 , **아주대학교 전자 전기 공학부
***한국전자통신연구원
gamilla@cesys.ajou.ac.kr

Linux System Security Using LIDS

o * Soojin Chung, ***JeongNyeo Kim, *SeungKyu Park, *Kyung-Hee Choi, **Gi-Hyun Jung,
* Information & Computer Division, Ajou University.
**Devision of Electronical & Electronics Engineering, Ajou University.
***Electronics & Telecommunications Research Institute

요 약

리눅스의 활용 범위가 임베디드 시스템 혹은 웹 서버 등 다방면으로 넓어져 가면서 보안 문제도 이와 함께 중요한 연구 및 개발 과제로 떠오르고 있다. Huagang 은 Linux 의 침입 감지 및 방어를 위하여 LIDS(Linux Intrusion Detection and Defense System)을 개발/발표하였다. LIDS 는 시스템의 침입 감지 및 방어에 중점을 둔 시스템으로서 사용자별 파일을 보호하기 위한 개념이나 시스템 관리자를 비롯해 일반 사용자에 이르기까지 전체적인 보안을 만족시키기에는 여러 가지 부족한 면을 가지고 있다. 본 논문에서는 이러한 LIDS 의 불편한 점을 보완하여 LIDS 의 활용 가능성을 높이었다.

1. 서 론

리눅스는 관련된 모든 소스가 공개되어 있다는 장점이 있기 때문에, Web 서버 등 인터넷 환경에서 사용되는 시스템에서부터 정보 가전에 이르기까지 매우 많은 응용 시스템들이 점차 리눅스를 바탕으로 개발되고 있다. 리눅스의 사용 범위나 분야의 증가와 함께 자연히 리눅스의 보안 문제도 매우 중요한 연구나 개발과제가 되고 있다. 커널과 일반 프로그램들을 비롯한 거의 모든 것의 소스가 공개되어 있어 많은 리눅스 시스템에 대한 해킹이 이루어지고 있기 때문이다[1,8].

이러한 리눅스 시스템의 침입에 대한 대책으로 Xie Huagang 은 LIDS(Linux Intrusion Detection/Defence System) - 리눅스 커널 2.2.14 에 패치하여 사용하는 0.9.1 안정 버전 - (이하 LIDS 로 지칭)를 개발 발표하였다[4]. LIDS 는 커널을 생성할 때 설정된 사용자 이외의 다른 사용자(루트 사용자 포함)는 보호되는 파일에 대하여 LIDS 가 지정한 작업만 할 수 있게 허락한다. 또한 LIDS 관리자의 패스워드가 커널 코드에 삽입됨으로써 침입에 대한 방어 능력을 강화되었으나 패스워드의 변경이 불가능하다는 단점을 가지게 되었다. 또한 LIDS 의 관리자는 모든 사용자의 파일에 대한 절대적 권한을 가짐으로써 파일 보호 측면에서 너무 강화되는 권한을 가지게 된다. 본 논문에서는 LIDS 의 이러한 단점을 보강고자 한다.

2. IDS(Intrusion Detection System)

IDS 는 허가되지 않은 사용자의 침입과 허가된 사용자의 잘못된 사용을 감지하여 적절한 조치를 취해 시스템을 보호하는 역할을 한다.

LIDS 는 침입을 감지하는 방법과 감시하는 대상을 기준으로 다음과 같이 분류할 수 있다 [2].

침입을 감지하는 방법으로는 misuse detection model 과 anomaly detection model 로 나눌 수 있다. Misuse detection model 인 경우 이미 기존에 알고있는 침입 방법과 일치하는 것을 감지하는 것으로, 기존 침입방법에 대한 DB 화가 중요하다. 알려지지 않은 방법으로 침입하는 것은 감지 할 수 없는 단점이 있다. Anomaly detection model 은 사용자의 일반적인 특성을 기억하고 있다가 사용자가 기억되어 있는 습성과 다른 행동을 하면 아이디를 도용한 침입으로 간주하는 방식이다. 이 방식은 구현이 매우 어려우므로 현재 개발된 IDS 의 90%가 misuse detection 방식을 사용한다.

감시하는 대상으로 분류하면 host based system 과 network based system, router based system 으로 나눌 수 있다. *Host based system* 은 해당 host 하나만을 감시한다. 즉 IDS 가 설치된 host 에 들어오는 요청의 침입 여부를 구별한다. 자기 자신만을 지키므로 다른 방법에 비해 가장 믿을 만하지만 대규모의 네트워크를 관리 할 때는 host 의 개수 뿐만 아니라 종류도 매우 다양하므로 설치와 관리의 한계가 있다는 단점이 있다. Network based system 는 네트워크 상에 돌아다니는 packet 의 형식이나 내용을 관찰하여 침입을 위한 것인지의 여부를 추론한다. 이 방식은 대규모의 네트워크에서 설치를 하여 관리하기는 쉬우나 가끔 중요한 패킷을 놓쳐 검사하지 못할 경우가 발생하기도 한다. *Router based system* 은 라우터에 설치되어 라우터에 물려있는 네트워크의 하부구조를 보호한다. 이 경우는 아예 해당 네트워크에 진입하는 곳을 막는 것이기 때문에 침입을 감지하는 데 있어서 매우 효과적이나 상당히 많은 데이터가 오고 가는 네트워크에서 라우터에 무거운 부담이 된다.

3. LIDS (Linux IDS)

LIDS 는 리눅스 커널에 패치되어 동작하는 침입 감지/보호 시스템(IDS)으로 2 장에서 설명한 IDS 의 기준으로 분류하면 misuse 방식의 host-based system 이다.

LIDS 의 목적은 시스템 내에서 모든 일을 할 수 있는 root 의 권한에 제한을 두는 것이다[4,7]. 현재 모든 리눅스 시스템에서는 해커가 root 의 권한을 획득하면 시스템을 마음대로 조작할 수 있다. 이에 LIDS 를 사용하면 root 의 권한을 획득한 해커의 공격을 효과적으로 차단할 수 있다. 리눅스 시스템에서 system call 은 실행 중에 capability 를 확인하여 실행시킨 프로세스가 적합한 자격이 있는지를 검사하는데 root process 는 모든 capability 를 가지고 동작하기 때문에 root 는 모든 system call 을 호출 및 사용을 할 수 있는 것이다. LIDS 는 이러한 capability 를 조작하여 root 가 하는 일을 제한한다. 예를 들어, LIDS 상태에서 CAP_NET_ADMIN capability 를 제거하면 아무리 root 라 하더라도 마음대로 네트워크 설정을 변경할 수 없다. 이런 식으로 LIDS 를 사용하면 capability 를 이용해서 중요한 프로세스들, 특히 init 프로세스를 부모로 둔 daemon 프로세스들을 함부로 죽일 수 없게 할 수 있다. 그리고 중요한 설정/로그기록 파일들 및 실행 파일들을 변경하거나 삭제하는 것을 막는다. (파일을 read-only 나 append-only 로 설정하여 보호한다.) 마지막으로 모듈을 임의로 추가하거나 삭제하지 못하게 하고, disk, memory, I/O port 등에 직접적인 접근을 막는 일 등을 할 수 있다.

4. LIDS 의 단점 및 해결책

LIDS 의 단점은 password 를 변경하기가 매우 불편하다는 것과 일반 사용자의 관점에서는 생각하지 않았다는 점이었다. 이 두 가지 아쉬운 점을 보안하기 위해 아래와 같은 해결방법을 생각해 보았다.

4.1 password 변경 문제

LIDS 모드와 capability 등을 설정 및 해제하는데 사용되는 LIDS 의 password 는 커널 컴파일 시에 지정하게 되어 있다. 따라서 사용하다가 잊어버릴 경우 또는 해커에게 비밀번호가 노출되었을 경우에는 다시 커널을 컴파일해서 비밀번호를 새로 지정해야 한다. 이는 매우 불편하고 비효율적이다.

이를 보안하기 위해 부팅시에 비밀번호가 적힌 파일을 읽어 커널 내에 저장하는 방법이 있다. 시스템이 부팅되어서 시스템 종료 때까지 동일한 password 를 유지하여 안정성을 주되 변경이 가능하도록 하는 방법이다.

이를 구현하기 위해 현재 커널 내부에 define 이 되어있는 password 를 커널에서 변수로 선언하고, 부팅 때마다 암호화된 password 를 저장된 파일에서 읽도록 한다. password 를 저장하고 있는 파일은 RipeMD160 을 이용해 encryption 된 값을 저장하고 root 만이 볼 수 있으며 LIDS 로 보호되어야 한다. 또한 제공되는 관리 프로그램인 lidsadm 에 비밀번호를 변경하기 위한 옵션을 추가하여 password 를 저장하는 파일을 관리한다.

LIDS 비밀번호를 변경하기 위해서는 우선 기존 비밀번호를 사용해서 LIDS 모드를 해제하고(password 설정파일이 LIDS 에 의해 보호되고 있으므로) 비밀번호를 변경한 후, 시스템을 rebooting 시켜야 한다.

Password 변경을 위해 LIDS 에서 수정된 코드는 아래 그림들과 같다.

그림 1 은 커널 부팅 시 파일 시스템을 mount 한 후 LIDS password 가 저장된 파일(/etc/lidspassword.conf)로부터 password 를 읽어와 커널 내 변수(LIDS_RMD160_PASSWD)에 저장하는 부분이다.

그림 2 는 커널 내에서 lidsadm 에 의해 불려지는 함수로 입력된 비밀번호와 커널 내에 저장된 비밀번호(커널 부팅 시 읽어 놓은 비밀번호)가 같은지 확인하는 부분이다.

그림 3 은 시스템 내에 LIDS 의 password 를 저장하고 있는 설정파일이다. lidspasswd.conf 는 LIDS 에 의해 read-only 로 보호된다. 파일에 저장되는 비밀번호는 그림 4 에 해당되는 코드에 의해 생성된다.

그림 4 는 LIDS 와 관련된 값들을 설정/관리하는데 사용되는 lidsadm 에 password 를 변경하기 위해 새로 생성된 함수이다.

```
#define LIDS_PASSWD_FILE "/etc/lidspasswd.conf"

//비밀번호 초기값 저장
char LIDS_RMD160_PASSWD[170] =
"bafc0d1a9d5aed88395021b8fe77a43e4b3a444c";;

static void __init do_basic_setup(void)
{
        :
    filesystem_setup();
    mount_root();

#ifdef CONFIG_LIDS
    {
      int lidsfd, rval=0;
      if((lidsfd=open(LIDS_PASSWD_FILE, O_RDONLY, 0))>=0)
      {
         if((rval=read(lidsfd, LIDS_RMD160 _PASSWD, 170))>0)
             LIDS_RMD160_PASSWD[rval]='\0';
         close(lidsfd);
      }
    }
#endif
        :
}
```

그림 1 linuxinit/main.c - 부팅 시 password 저장

```
extern char LIDS_RMD160_PASSWD[170];
        :
int lids_proc_locks_sysctl(ctl_table *table, ....
{
   if (write) {
      :
#ifdef CONFIG_LIDS_ALLOW_SWITCH
      :
   if ( ((lids_first_time)&&(!locks.passwd[0]))||
     (!strncmp(rmd160sig,LIDS_RMD160_PASSWD,
 strlen(rmd160sig))) ) {
        :
   }
   else
     lids_security_alert("incorrect password ..");
        :
```

그림 2 linux/fs/lids.c - 입력된 password 확인

```
# ls -l /etc/lidspasswd.conf
-rw------- 1 root root 530 Aug 18 01:29 /etc/lidspasswd.conf

# cat lidspasswd.conf
7c633229fff4ed400b55b6085d62021617f387a2
```

그림 3 /etc/lidspasswd.conf

```
#define LIDS_PASSWD_FILE "/etc/lidspasswd.conf"

void lids_change_passwd()
{
  char passwd[64];
  int times=0;
  int ok;

  while(times < 3) {
     times++;
     if ((ok=read_rmd160_passwd(passwd,1,2))==0)
        break;
  }
  if((fd=open(LIDS_PASSWD_FILE,
   O_WRONLY|O_CREAT|O_TRUNC, S_IRUSR|S_IWUSR))<0)
  {
     printf("password file open error\n");
     return ;
  }
  write(fd, passwd, 64);
  close(fd);
}
```

그림 4 lidsadm 의 password 변경 함수

4.2 일반 사용자를 위한 보안

현재 LIDS 는 시스템에 관한 보안만을 위한 것으로 일반 사용자가 자신의 파일을 보호하고 싶을 때는 해결책이 없다.

보통의 리눅스 시스템에서 root 는 일반 사용자의 파일도 모두 수정이나 삭제가 가능하다. 이는 일반 사용자가 자신의 개인정보나 중요한 비밀사항을 파일로 저장하고자 할 때 불안한 일임에 틀림 없다. 해커가 root 의 권한을 얻었을 때, 시스템과 관련된 파일들은 LIDS 가 지켜주겠지만 LIDS 로 보호하기에는 자주 사용되고 수정되는 일반 사용자의 개인적이고 중요한 파일의 경우는 보호가 불가능하다. 관리자에게 부탁해 자신의 파일을 LIDS 로 read-only 나 append-only 로 설정해 놓을 수 있지만 이렇게 하면 자신이 작업하며 수정할 수가 없다.

이를 수정/보안 하기 위해 파일을 열거나 삭제하거나, 또는 파일의 소유자를 변경하는데 사용되는 system call 인 sys_open 과 sys_chown, sys_unlink 의 코드에 파일의 소유자가 root 가 아닐 경우 root 가 파일을 열어 볼 수 없고 삭제할 수 없도록 또는 소유자를 root 로 바꿀 수 없도록, 접근을 제한하는 코드를 삽입하였다. LIDS 가 설정되어 있을 때는 위의 세 가지 system call 을 사용할 때 root 도 일반사용자의 파일을 함부로 수정하거나 변경시키지 못한다.

추가된 소스 코드 중 sys_open 을 살펴보면 그림 5 와 같다.

```
asmlinkage int sys_open(const char * filename, int
      flags, int mode)
{
     char * tmp;
     int fd, error;
#ifdef CONFIG_LIDS
     struct stat filestat;
     stat(filename, &filestat);
     if(getuid()==0)
       if(filestat.st_uid != getuid())
         return ;
#endif
      :
```

그림 5 sys_open – 일반 사용자의 권한 증가

5. 결론

이 논문에서는 시스템의 침입을 감지하는 LIDS 를 사용할 때 비밀번호를 변경하지 못하는 불편을 개선하고 일반 사용자의 파일에 대한 미흡한 점을 개선해 좀 더 시스템 관리자와 사용자에게 만족스러운 시스템으로 만들었다.

LIDS 는 해당 host 를 침입으로부터 지키는데 효과적인 시스템이다. 하지만 LIDS 를 설치했다고 해서 보안에 대해서 안심해서는 안 된다. 어떠한 보안 시스템도 모든 해킹으로부터 완벽하게 안전할 수는 없기 때문이다. 시스템에 대한 보안과 편리성은 서로 trade-off 관계이기 때문에 사용자에게 편리성을 제공하기 위해서는 어느 정도 이상의 보안 설정 강도를 높일 수 없다[8]. 또한 해커가 LIDS 의 비밀번호를 알아낼 수도 있다. 따라서 시스템 관리자는 항상 보안에 신경을 쓰고 비밀번호를 자주 변경하고 자료를 백업하며 log 파일을 수시로 살펴야 할 것이다. 그리고 일반 사용자 역시 자신의 중요한 자료를 자주 백업해 두도록 해야 할 것이다.

6. Reference

[1] PLUS (포항공대 유닉스 보안 연구회) "Security Plus for UNIX" ISBN: 89-314-1490-0

[2] Robert Durst, Terrence Champion, Brian Witten, Eric Miller and Luigi Spagnuolo, "testing and evaluating computer intrusion detection systems." *Commun. ACM* 42, 7 (Jul. 1999), Pages 53 – 61

[3] Xie Huagang, "Build a security system with LIDS" http://www.lids.org

[4] Xie Huagang, "LIDS Hacking HOWTO" http://www.lids.org

[5] Philippe Biondi, "LIDS-Howto http://www.lids.org

[6] Intrusion Detection http://www.cerias.purdue.edu/coast/intrusion-detection/detection.html

[7] David "Del" Elson "Intrusion detection on linux" http://www.securityfocus.com/focus/ids/articles/linux-ids.html

[8] kevin Fenzi, Dave Wreski , "Security-HOWTO" http://kldp.org/HOWTO/html/Security/Security-HOWTO-10.html