# 커널 백도어 공격 탐지 및 복구시스템 설계에 관한 연구

# (A Study for Detection of the Kernel Backdoor Attack and Design of the restoration system)

전 완 근*, 오 임 걸**
(Wan-Keun Jeon, Im-Geol Oh)

**요 약** 커널 백도어로부터 공격이 탐지되는 즉시 해킹 피해를 입은 시스템에서 증거 손실을 최소화하고 안전하고 신뢰할 수 있는 증거 보존, 그리고 신속하게 대응하도록 함으로써 시스템 피해를 최소화할 수 있는 백업 및 분석시스템을 설계 구현한다. 본 논문에서는 삭제된 로그파일을 복원하고 복원된 파일과 하드디스크의 이미지를 분석하여 해커의 위치를 찾을 수 있다.

**핵심주제어** : 커널 백도아, 백업시스템, 분석시스템, 포렌직, 침입자

**Abstract** As soon as an intrusion is detected by kernel backdoor, the proposed method can be preserve secure and trustworthy evidence even in a damaged system. As an experimental tool, we implement a backup and analysis system, which can be response quickly, to minimize the damages. In this paper, we propose a method, which can restore the deleted log file and analyze the image of a hard disk, to be able to expose the location of a intruder.

**Key Words** : Kernel Backdoor, Backup system, Analysis System, Forensic, Intruder

## 1. 서 론

정보보호를 위한 대표적인 도구인 침입차단 시스템과 침입탐지 시스템은 현재 많이 사용되고 있다. 하지만 해킹 기술의 발전은 이러한 도구들을 계속적으로 무력화하고 있다[1].

결국 보안과 해킹의 양면적인 특성은 상호보완적인 관계로 발전해가고 있으며 공격이 일어난 사건을 다루는 컴퓨터 포렌식(forensic) 분야의 중요성이 부각되고 있다.

컴퓨터 포렌식 분야는 컴퓨터 데이터에 대한 보존, 검출, 분석, 문서화에 대한 내용들을 다루며[2, 3], 영역에 따라 크게 법집행(Law enforcement), 정보전(Information warfare), 중요산업기반 보호의 영역으로 나누어진다[4]. 그 중 전통적으로 법집행의 영역은 컴퓨터 포렌식 분야의 중심이었으며 증거가 소멸되지 않기 위한 무결성의 검증, 원본 이미지의 훼손 방지 등 법적인 관점이 중심이 되었다. 그러나 군대나 민간에서의 컴퓨터 포렌식은 공격이 발생되었는지의 여부를 빨리 알아서 공격의 위협을 최대한 빨리 제거하는 것을 강조한다[4].

* 디지털 포렌직 센터
** 한서대학교 인터넷공학과

커널 백도어(Backdoor)는 기존의 백도어의 동작 형식과 프로그램 형태가 완전히 다르며 이와 같은 형태의 백도어를 만들려면 리눅스 커널(linux kernel)[5-9]에 대한 완벽한 이해를 수반하여야 하기 때문에 해커의 프로그래밍 능력 또한 매우 높다고 볼 수 있다. 이와 같이 고난도의 기술을 가진 해커를 검거한다는 것은 현재까지는 거의 불가능하였다.

로그 기록을 기반으로 하는 기존의 역추적 또는 호스트 기반 침입탐지시스템은 로그가 삭제된 경우에는 더 이상의 분석과 침입자 추적에 힘이 들었기 때문이다.

이처럼 삭제된 파일에 대한 복구하는 기능을 기존의 컴퓨터 커널 백도어을 탐지하는 도구[10]들에는 없어 침입자를 추적하는 데에는 한계가 있으며 공격방법이나 공격 툴을 추정하는 데 그쳐 심층적인 분석이 불가능 하였다.

본 논문에서는 이와 같은 문제점을 해결하기 위해서는 삭제된 로그기록 뿐만 아니라 교활한 해커에 의해 심어놓은 공격 툴 등과 같은 바이너리 파일복구기능을 추가하였으며 커널 백도어 공격 탐지시점의 침입증거를 확보하여 침입자를 추적하고 그 피해를 최소화하도록 신속하게 대응하는 백업 시스템 및 분석시스템을 설계하고 구현한다.

## 2. 관련연구

### 2.1 백업시스템 설계

백업은 실제 피해시스템에서 분석을 시작한 시점 이후로 데이터가 손상되지 않았다는 전제가 바탕이 되어야 하므로 피해시스템의 데이터 획득 및 분석 과정에서 조작상의 문제, 시스템 자체의 오류 등으로 인해 시스템의 정보가 수정이 되어버리는 경우를 대비하기 위한 절차이다.

백업은 침입탐지시스템에서 커널 백도어의 침입을 인지한 후 이루어지는데 분석을 위한 첫 번째 단계로 커널 백도어 침해탐지시스템으로부터 백업 요청이 오면 바로 피해시스템에 대한 백업에 들어가게 되는데 시스템, 네트워크와 같은 휘발성정보 정보추출을 위한 1차 백업과 상세정보추출을 위한 2차 백업으로 나뉘어 이루어지도록 설계하였다.

원격에서 이루어진 백업 이미지에 대하여 해쉬값(Hash Value)을 부여하여 백업 이미지의 무결성을 확보되도록 하였다.

#### 2.1.1. 1차 백업

1차 백업과정에서는 침입자 추적을 빠르게 하기 위해 커널 백도어 공격 탐지시점 당시의 커널 레벨 및 응용프로그램상의 현재 운영 중인 프로세스 및 네트워크 연결 정보 등을 수집하도록 하였다. 이렇게 수집된 프로세스 정보와 네트워크연결 정보 등 공격 IP 등을 확인할 수 있는 자료를 추출하여 분석시스템으로 보내져 IP 조회를 통하여 침입자 추적을 할 수 있도록 한다.

커널에서 직접 원천적인 데이터를 추출하는 방법을 사용하여 변조되지 않은 정보를 수집함으로서 침입자가 피해시스템에 설치한 루트킷(root kit)과 같은 악성 프로그램과 프로그래밍에 의해 변조된 정보를 가져오게 될 가능성을 줄이도록 하였다.

또한 일반적으로 중요한 운영체제정보와 같은 시스템 정보 및 환경 설정 내용, 로그 파일, 데이터 디렉터리, 기타 응용 프로그램 관련 파일 등도 1차 백업 시 수집된다.

#### 2.1.2. 2차 백업

2차 백업에서는 파일시스템에 대한 심층 분석을 하기 위한 디스크 이미지를 백업 서버로 전송하도록 한다. 2차 백업에서는 피해시스템의 증거를 훼손하지 않고 복사된 정보를 분석하기 위하여 피해시스템의 하드 디스크 전체에 대하여 파티션 별로 분석 시스템으로 복사한다.

파일시스템에서 1024 바이트 단위로 이미지를 복사한 후 백업시스템에 전송한다. 백업된 전체 이미지를 MD5[11], SHA[12]등의 무결성 검사 프로그램을 이용하여 백업시스템에서 분석시스템으로의 전송 전과 후 변조유무를 확인을 위해서 해쉬값을 부여한다.

### 2.2 분석시스템

### 2.2.1 분석시스템의 개요

컴퓨터 포렌식에 있어서 증거물 분석과정은 시스템의 종류, 범죄의 유형, 분석도구 등에 따라 그 방법이 매우 다르다[13]. 분석내용은 기본적으로 행위자, 시간, 내용 등의 순으로 파악하고, 가능하면 사건을 재현할 수 있도록 하여야 한다[13, 14].

분석시스템의 가장 큰 목표는 해킹 침입 및 공격 받은 시스템을 분석하고 남겨진 증거들을 기반으로 공격방법을 추론하고 공격자의 위치를 추적하는 것이다.

백업시스템에서 1차와 2차 백업을 거쳐서 시스템/네트워크 정보, 하드디스크의 전체 이미지 파일 등에 해쉬값이 부여된 후 분석시스템으로 전송된다. 분석시스템에서는 백업시스템으로부터 이송된 이미지 파일에 대한 무결성 검사과정을 거친 후 복구 과정과 1차 및 2차 분석을 통하여 찾아낸 공격시스템의 IP 주소에 대한 추적조회와 삭제된 파일이나 디렉토리 정보, 숨겨진 폴더나 파일등을 찾아내고, 네트워크 정보와 시스템 정보를 이용하여 추적을 위한 정보를 추출하는 중요한 역할을 하게 된다.

### 2.2.2 분석시스템의 기능

분석기능은 포렌식 시스템에서 가장 중요한 역할을 하며 아래와 같은 기능 등이 필요하다.

(1) MD5. SHA등을 이용하여 무결성 검사 기능

증거물의 획득-이송간의 수집된 증거가 위조 또는 변조되지 않아야[15] 정확한 분석을 할 수 있다. MD5 도구를 이용하여 증거물 획득 및 획득이후의 해쉬값의 동일성여부를 체크한다.

(2) 복구기능

복구기능을 이용하여 하드디스크에서 공격자가 고의로 은닉하거나 삭제한 로그파일이나 공격툴 등을 찾아 낼 수 있도록 하였다.

일반적으로 파일의 생성, 변경 및 삭제가 빈번히 발생하고, 강력해지는 공격도구들은 침입 및 공격의 흔적을 모두 제거하는 방법들이 이용되고 있기 때문에 삭제된 부분을 복원하기 위한 파일시스템의 복구는 기본적이고 가장 중요한 기능이다. 앞으로 이 복구기능의 침입자를 찾는데 있어서 더욱 더 중요해 질 것이다.

(3) 데이터 추출 및 검색기능

분석기능은 삭제되거나 지워진 파일에서 복구된 파일과 백업된 전체이미지, 네트워크정보 등으로부터 특정 파일이나 문자열 등을 검색하여 정보를 추출하는데 이용된다.

(4) Whois 서비스 자동조회기능[16-19]

네트워크 및 시스템 로그 분석과정을 통하여 나오는 모든 네트워크 정보나 스캔 공격을 시도한 접속로그 등에 결과가 피해시스템에 남겨져 있을 수 있으며 그중 IP 주소정보는 침입자를 추적할 수 있는 중요한 단서가 되며 이를 Whois DB에 자동조회를 통하여 침입자의 해킹을 시도하는 위치를 파악한다.

### 2.2.3 분석절차

해킹탐지시스템에서 커널 백도어의 공격이 탐지될 경우에 실시간으로 시스템 내에 네트워크정보와 메모리정보 등과 같은 휘발성 정보를 자동으로 백업한 후 백업 및 분석시스템으로 정보를 전달한다. 분석시스템은 전달받은 정보 등에 대한 분석을 하여, 피해시스템의 피해정도를 파악하고 침입자의 위치를 추적한다.

(1) 1차 분석

1차 분석에서는 주로 시스템, 네트워크, 휘발성 정보 등을 주로 분석하며 IP 주소, 운영체제정보, 비정상적인 프로세스, 네트워크 정보 등을 백업시스템으로부터 전송받아 분석시스템의 Whois 조회기능을 이용하여 IP 조회를 통하여 침입자의 위치를 추적한다.

(2) 2차 분석

2차 분석은 정밀분석으로 HDD 이미지나 복구된 로그파일로부터 삭제되거나 숨겨진 데이터나 파일이나 디렉토리 정보 등을 추출하여 공격 루트

킷이나 백도어 기타 공격에 이용된 S/W등을 통하여 공격방법이나 공격도구 등을 점검한다.

## 3. 백업시스템 구현

백업시스템에서는 침입탐지시스템으로부터 원격 백업요청이 오면 침입자추적에 필요한 수사정보를 수집하여 분석시스템으로 넘겨주는 기능을 수행한다. 백업은 침입을 인지한 후 분석대응을 위한 첫 단계라 할 수 있다. 침해사고 분석시 얻어지는 모든 정보가 법적인 증거로서의 효력을 갖기 위해서는 우선, 실제 피해시스템에서 분석을 시작한 시점 이후로 데이터가 손상되지 않았다는 전제가 바탕이 되어야 하므로 피해시스템의 데이터 획득 및 분석 과정에서 조작상의 문제, 시스템 자체의 오류 등으로 인해 시스템의 정보가 수정이 되어버리는 경우, 그리고 분석을 시작할 당시의 프로세스와 메모리 정보, 시스템 자원 사용 정보 등의 휘발성 정보에 대한 백업이 이루어지지 않은 경우 침해사고 분석 결과에 치명적인 영향을 줄 수 있다.

본 논문에서 구현된 백업시스템은 KRCERT (Korea Computer Emergency Response Team)에서 해킹피해시스템에 대하여 실제로 행하여지는 분석 작업의 순서측면에서 접근하여 백업 알고리듬을 구현하였다.

### 3.1 1차 백업

침입에 대한 정보를 분석하기 위해 대부분의 경우 일단 시스템을 종료시키거나 네트워크에서 격리시키는 등의 조치를 하게 된다. 이러한 과정에서 공격자의 로그인 상태정보, 네트워크 연결 정보, 커널 모듈정보 등 중요한 정보가 손상되게 되므로 피해시스템을 격리하기 전 현재 시스템의 상태를 완벽하게 보존하여야 한다. 이러한 휘발성 정보를 백업하기위해 시스템 명령어를 사용하는 방법과 커널에서 원천적인 정보를 획득하는 방법을 모두 병행하여 사용하였다.

(1) 시스템 명령어를 이용한 정보수집

```
Step 1. 백업시스템은 침입탐지시스템에서 요청한 원격 백업요청을 수락한다.
Step 2. 백업시스템에서 원격으로 피해시스템에 연결한다.
Step 3. 정보수집모듈을 수행한다.
    if (1차 백업) then
  {
    시스템명령어를 이용한 시스템 및 네트워크 정보수집;
    커널레벨 프로세스 및 네트워크 정보수집;
    중요 파일 백업;
    1차 수집결과를 분석시스템에 전송; }
    else if (2차 백업) then
  {
    전체 HDD 이미지 백업;
    이미지에 해쉬값 생성 및 부여;
    이미지를 분석시스템에 전송; }
    else Step 4.
Step 4. 백업을 종료한다.
```

시스템명령어를 이용하여 피해시스템의 호스트명, OS, 분석 당시 시스템 시간, IP 주소, MAC 주소 정보를 수집한다. 주로 프로세스 정보와 네트워크 연결 정보위주로 수집을 한다. 다음은 시스템 명령어를 이용하여 정보 수집 시 필요한 시스템 명령들이다.

- hostname : 피해시스템의 호스트이름
- uname-s : 피해시스템의 운영체제와 버전 정보
- date : 피해시스템에 접속하여 정보수집 날짜와 시간
- ifconifg : NIC의 네트워크 설정 정보
- netstat -rn : 라우팅 테이블 정보
- lsof -P -i -n : 열려있는 네트워크 소켓 정보
- lsof : 현재 실행되는 프로세스가 사용하는 파일 정보
- who : 현재 로그인한 사용자 정보
- arp : arp 캐쉬 정보
- ps : 현재 실행되고 있는 프로세스 정보

(2) LKM(Loadable Kernel Module)을 이용한 정보수집

시스템 명령을 사용하여 그 결과를 백업하는 방법에서는 조사할 수 있는 시스템 프로그램은 루트킷 또는 악성프로그램에 의해 변조된 정보를 가져

올 가능성이 크다.

그러나 시스템의 커널에서 직접 원천적인 데이터를 추출하는 방법을 사용하면 변조되지 않은 정보를 원천에서 얻을 수 있다. 본 논문에서는 시스템 커널에서 변조되지 않은 프로세스와 네트워크 정보를 추출하기 위해 LKM을 이용하여 각각 구현하였다.

• LKM 프로세스 정보수집모듈

프로세스 정보수집모듈은 커널의 일부분으로 동작하며 커널과 동일한 권한을 가진 LKM를 통해 커널의 프로세스 정보에 접근하여 숨겨진 프로세스 정보를 추출하고, 시스템명령어를 이용하여 나온 정보와 비교함으로써 숨겨진 프로세스를 찾아내도록 하였다.

프로세스 구조체는 task_struct에 정의되어 있으며, 환형이중 연결리스트구조를 가지고 있어 init_task로 접근하면 모든 프로세스에 대한 정보를 얻을 수 있다. <그림 3-1>은 프로세스 정보수집모듈을 나타내고 있다.

```
task_struct *proc_task;
proc_task = &init_task;
do
{
   // task_struct에서 정보추출
      .....
      proc_task = proc_task->next_task;
   //다음 프로세스의 task_struct를 얻음.
} while( proc_task->pid != init_task.pid );
```

<그림 3-1> LKM 프로세스 정보수집모듈

• LKM 네트워크 정보수집모듈

네트워크 정보수집모듈은 공격자가 설치하였을지도 모를 커널 루트킷을 고려하여 커널 레벨에서 네트워크 연결정보 등을 수집할 수 있도록 구현하였다. 커널이 관리하는 각 프로세스의 네트워크 구조체인 소켓은 프로세스 내에서 열려진 파일로 취급되므로 열려진 파일을 추적하면 네트워크 정보에 접근할 수 있다.

<그림 3-2>은 task_struct로부터 socket까지를 추적해가며 현재 연결된 네트워크 정보를 추출하는 것을 나타내고 있다.

```
* /dev/kmem을 연다.
kd=open("/dev/kmem", O_RDONLY);
  . ..
* struct task_struct *init_task의 주소를 찾는다.
for (j = 0, s = syms; j < ret; ++j, ++s)
 {
   if(strstr((char *)syms+ s->name, sym_name))
   {
     init_task_addr = s->value;
     break; } }
* 모든 프로세스에 대해 task_struct를 추적한다.
for(i=1; i < NR_TASKS; i++)
 {
   * /dev/kmem으로부터 각 프로세스의
      task_struct를 읽는다.
     read (kd, &taskk, sizeof (struct
     task_struct));
   if(taskk.files)
    {
      * 열려진 파일들에 대해서 멤버변수를 추적
      * struct sock *sk를 추출
      * (struct *task_struct process).(struct
         files_struct *files)
      * (struct file** fd).(struct dentry
         f_dentry)
      * (struct inode *d_inode)
      * (union u).(struct socket socket_i)
      * (struct sock *sk)
      * struct sock *sk에서 네트워크 정보를
        접근 }
    addr=(unsigned long)taskk.next_task;
  }
```

<그림 3-2> LKM 네트워크 정보수집모듈

이렇게 추출된 프로세스와 네트워크 정보는 온라인상의 정보로서 휘발성이라는 특징을 가진 매우 중요한 정보이다. 이는 초기 온라인상의 상태값으로 의미뿐만 아니라 실제 변조의 위험으로부터 벗어나 시스템에서 중요한 정보로 취급되어진다. 또한 숨겨진 프로세스나 숨겨진 네트워크를 연결 정보를 찾아 공격자의 경로를 추적할 수 있는 중요한 정보가 된다.

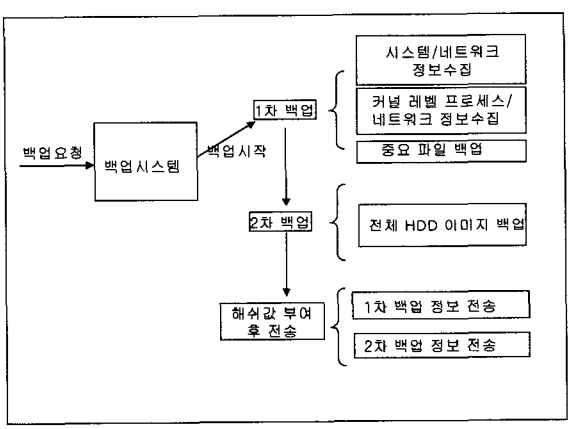
### 3.2 2차 백업

2차 백업에서는 피해시스템의 증거를 훼손하지 않고 복사된 정보를 분석하기 위하여 피해시스템의 하드디스크 전체에 대하여 파티션별로 분석시스템으로 복사한다. 백업단계에서 각각의 파티션을 비트 단위로 이미지 백업을 수행하도록 한다.

### 3.3 이미지 해쉬값 생성

백업된 전체 이미지를 MD5등의 프로그램을 이용하여 분석시스템으로의 전송하기 전에 무결성 확보를 위해서 해쉬값을 부여한다.

<그림 3-3>은 피해시스템에서 정보수집기능을 수행하는 프로그램의 모듈 구성과 동작 흐름도이다.




<그림 3-3> 백업 흐름도

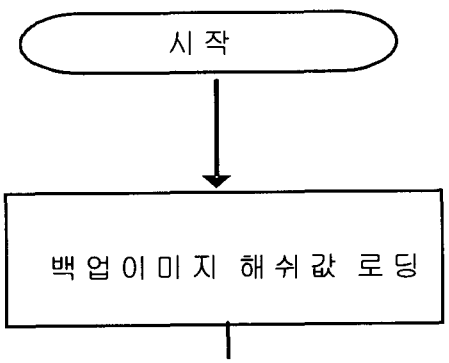
## 4. 분석시스템 구현

분석시스템은 백업시스템에 의해 이루어진 피해시스템에 대한 1차, 2차 백업자료(시스템 및 네트워크정보, 디스크 이미지 복사본)에 대한 분석이 이루어진다. 분석은 먼저 백업시스템으로 받은 이미지의 위·변조 확인작업을 거친 후 1차와 2차로 나뉘어 이루어진다. 이 과정을 통하여 침입자의 IP 주소와 침입방법, 침입도구 등을 밝혀낸다.

분석알고리듬은 KRCERT[20]에서의 사용하는 방법을 적용하여 구체적인 피해시스템에 대한 증거추출에 있어서 활용도를 높였다. 또한, 피해시스템의 자원을 이용하지 않고 분석시스템의 자원을 이용하여 정확하게 분석할 수 있도록 하였다.

### 4.1 증거파일 위·변조 검사

증거파일에 대한 분석에 앞서 백업시스템으로부터 전송되기 전과 동일한 파일인지를 확인하여야 한다. <그림 4-1>은 증거물의 무결성을 확보하기 위해 해쉬 및 오류검증알고리듬을 이용하여 원본 디스크이미지와 디스크이미지의 해쉬값이 일치하는 지를 간단히 확인한 후에 변조가 안 되었다는 것을 검증하는 과정이다.




<그림 4-1> 백업이미지 무결성 검사

### 4.2 1차 증거분석 및 침입자 추적

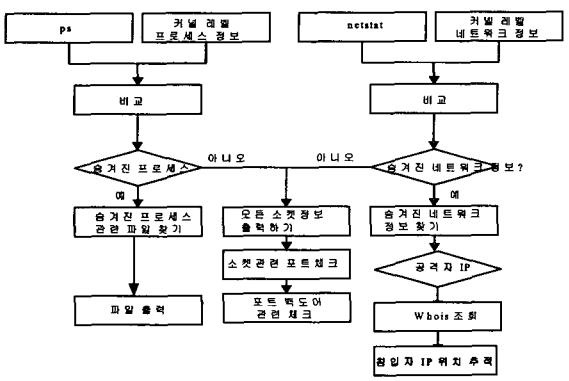
1차 증거분석과정에서는 커널로부터 추출한 네트워크 및 프로세스 정보와 응용 계층에서 추출한 결과를 비교함으로서 루트킷 탐지, 백도어 탐지, 이상 프로세스 탐지 등의 다양한 정보를 수집하고 최종적으로 침입자의 IP 주소를 알아내어 Whois DB를 이용하여 위치를 찾아내는 기능을 하도록 구현하였다.

<그림 4-2>은 원천 휘발성 정보인 프로세스 및 네트워크 정보와 시스템 명령어를 이용한 정보를 비교하는 모듈의 동작 방식을 사용하고 있다.

프로세스 분석은 시스템 명령어 'ps' 와 원천 프로세스 정보를 비교하여 숨겨져 있는 프로세스를 검출하고, 숨겨진 프로세스가 사용하는 파일의 정보를 수집하여 증거로 저장한다. 숨겨진 정보가 없다면 모든 태스크의 소켓정보를 검출하고 그 소켓의 포트를 검출하게 된다.

네트워크 역시 시스템 명령어인 'netstat' 와 원

천 네트워크 정보를 비교하여 숨긴 네트워크 연결을 검출하고 침입자의 IP 주소를 파악한다. 파악된 IP 주소는 Whois DB 조회를 통하여 자동으로 침입자의 위치를 판별하게 된다. (그림 4-10)에서와 같이 피해시스템 분석 결과 침격자의 것으로 추정되는 의심스러운 네트워크 주소에 대하여 Whois 조회를 통하여 위치를 알아내도록 하였다.




<그림 4-2> 1차 증거분석 흐름도

### 4.3 2차 증거자료 분석

백업 1차 과정을 통해 수집된 프로세스나 네트워크 증거자료 분석을 통하여 시스템관련정보와 네트워크 정보검색을 통하여 침입자의 접속 IP 주소에 대한 Whois 조회를 통하여 위치를 추적할 수가 있었다. 2차 증거자료 분석에는 1차 분석단계에서 추출한 IP 주소에 대한 침입시간대를 로그분석을 통하여 추가적으로 확인한 후 시스템 침입하기 전이나 후에 일어난 공격행위, 침입방법, 취약점, 공격 툴 등을 구체적인 증거자료를 찾도록 구현하였다.

증거분석 모듈에서는 다음과 같이 4단계로 이루어진다.

- 이미지 복구
- 로그파일 분석
- 바이너리파일의 무결성 검사
- 공격 툴 및 공격도구 검색

#### 4.3.1 복구된 로그파일 분석

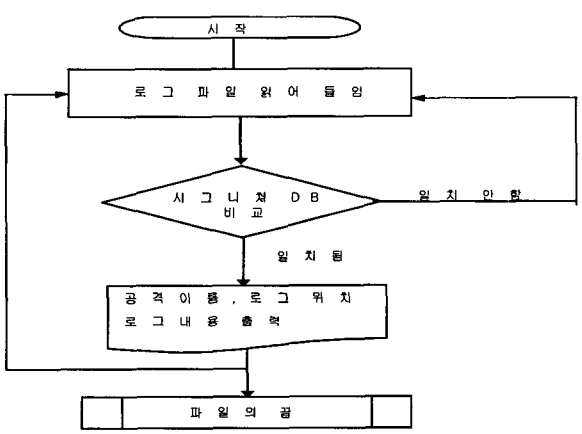
해킹당한 피해시스템에서의 침입증거분석에 있어서 로그기록은 매우 중요한 단서가 된다. 해커는 시스템에 침입한 후 관련된 로그들을 삭제하기 때문에 침입관련 증거자료 등을 찾아내기가 매우 힘이 들다. 이와 같이 대부분 로그기반의 분석방법에 한계를 가지게 된다. 그러나 포렌식 기법의 하나인 삭제된 데이터나 파일을 복구하여 살려내면 로그기록을 확보하여 분석을 할 수가 있다. 즉 복구된 로그파일을 분석하면 침입자, 침입방법, 취약점 등을 알아 낼 수가 있다.

공격시점은 IP 추적이나 공격 행위 등을 추가적으로 알아내기에 반드시 알아내야할 정보이다. 시스템에 남겨진 로그정보를 분석하면 침입자의 공격시간대를 추적할 수 있다.

이 시간정보는 커널 백도어 탐지모듈에서 탐지된 시간정보에 신뢰성을 강화하는 역할을 한다.

- 로그 검사

로그검사는 문자열 검색을 사용하여 피해시스템의 로그파일에서 공격시도로 추측되는 엔트리를 추출하는 모듈이다. 로그파일에 있는 문자열을 라인 단위로 검사하여 피해시스템의 로그파일에 특징과 일치하는 문자열이 발견되면 해당 로그 엔트리를 출력하는 모듈이다. <그림 4-3>은 알려진 공격을 탐지하는 모듈의 흐름도이다.



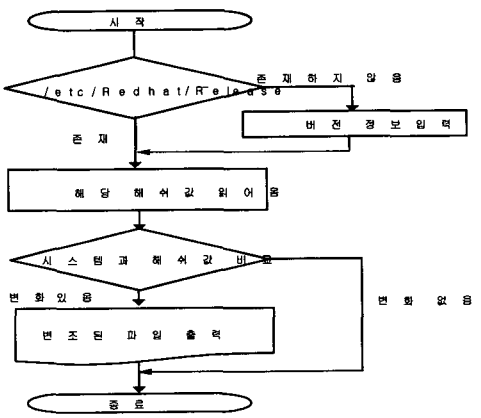

<그림 4-3> 알려진 공격탐지모듈 흐름도

#### 4.3.2 바이너리 파일의 무결성 검사

해커는 시스템에 침입한 후 자신의 흔적을 숨기거나 백도어를 설치하기 위해서 다양한 바이너리

파일을 변조한다. 따라서 본 논문에서는 리눅스 시스템에서 바이너리 파일을 가지고 있는 /bin, /sbin, /usr/bin, /usr/sbin 디렉터리 아래의 바이너리 파일의 무결성을 검사한다. 바이너리 파일 무결성 검사모듈에서는 MD5 해쉬값과 파일의 stat 구조체 값을 유지함으로써 변조 여부를 검사한다.

<그림 4-4>는 모듈의 바이너리파일의 무결성 검사흐름도이다.




<그림 4-4> 바이너리 파일 무결성 검사 흐름도

- 바이너리 변조 검사

피해시스템의 바이너리 변조 유무를 검사하기 위한 모듈이며 'md5sum' 이라는 프로그램을 사용하여 아래와 같이 간단히 구현하였다.

```
md5sum -c /md5_result.txt | grep FAILED
```

md5sum 프로그램의 '-c' 옵션은 인자로 주어지는 md5 체크섬 데이터 파일과 리스트에 있는 파일에 대한 무결성 검사를 하고 두개의 결과 값이 다른 바이너리를 출력해준다. md5 체크섬 데이터 파일은 다음과 같은 방법으로 네트워크에 연결되지 않은 새로 설치된 시스템에서 간단하게 수집하였다.

```
#cd /bin
#md5sum * >> /md5_result.txt
```

### 4.3.3 공격 툴 및 프로그램 검색

공격 툴 및 알려진 루트킷이 생성하는 디렉터리 또는 파일이 피해시스템에 존재여부를 확인한다.

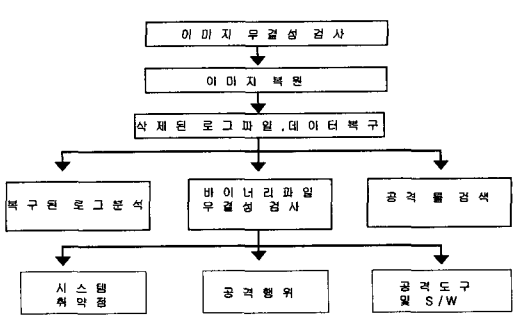
다음은 공격 프로그램을 찾는 검색모듈의 알고리듬이다.

```
dir_find()
{
  if [ -d $1 ] then
  echo "!!WARNING!! suspicious
  dir $1 found..."
  file
}
file_find()
{
  if [ -f $1 ] then
  echo "!!WARNING!! suspicious file
  $1 found..."
  file
}
# 의심스러운 디렉토리
dir_find /tmp
dir_find /var/run/.tmp
dir_find /dev/...
dir_find /bin/...

# 의심이 가는 파일들
file_find /dev/ptyy
file_find /dev/ptyu
file_find /dev/ptyq
```

또한 검색모듈에서는 다음과 같은 기능도 수행한다.

- /dev에 존재하는 정규 파일 검사
- 파일명이 "." ".." 으로 시작하는 파일 검사
- 디렉터리 명에 공백문자를 포함하는 디렉터리 검사
- /etc/passwd 파일의 root 이외의 계정에 UID가 0인 계정 검사

<그림 4-5>은 2차로 수집된 증거자료에 대한 분석흐름도이다.



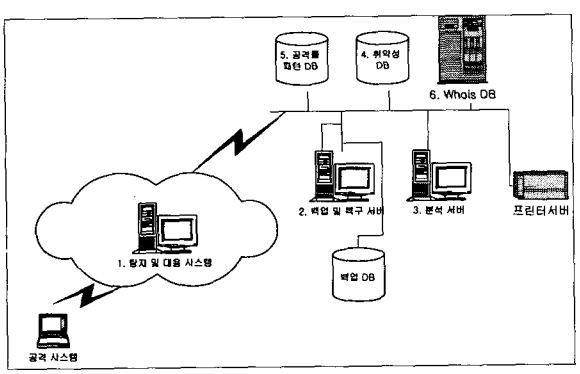

<그림 4-5> 2차 증거자료 분석흐름도

먼저 하드디스크와 이미지의 해쉬값을 비교하여 그 결과값이 일치하면 이미지를 복원한다. 또한 변조되거나 삭제된 로그파일을 복구하여 로그파일에 남겨진 시그니쳐를 확인하여 피해 시스템이 가지고 있는 취약성정보를 확인한다. 즉 침입방법을 분석하기위해 시스템내의 로그정보를 기반으로 공격을 당한 원인을 분석하게 된다. 복구된 데이터에 대한 바이너리 파일의 무결성 검사등을 통하여 루트킷, 해킹도구, 악성 프로그램의 존재유무 등을 찾아내어 현재 시스템에 피해상황을 판단하게 된다.

## 5. 시험분석 및 결과

### 5.1 시험환경

시험환경은 <그림 5-1>과 같이 탐지 및 대응시스템을 피해시스템에 설치하고, 공격시스템에서 목표시스템으로 해킹공격을 하면 침입탐지기에서 침입상황을 탐지하여 침입사실을 통보하도록 한다. 동시에 해킹피해시스템에서 이미지 백업을 받아 분석을 할 수 있도록 환경을 구축하여 시험하였다.



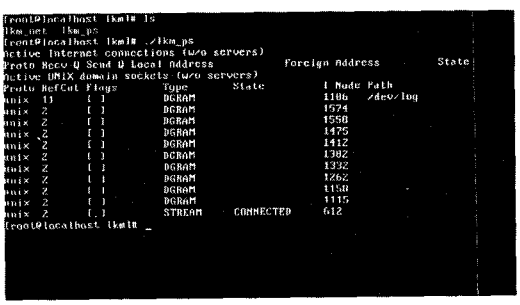

<그림 5-1> 시험분석 환경 구성도

### 5.2 백업시스템 시험분석

#### 5.2.1 1차 네트워크 및 프로세스 정보백업

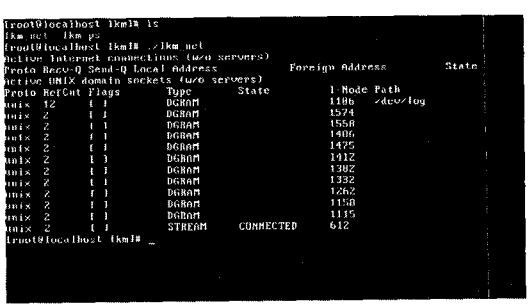
리눅스 시스템의 경우 공격자가 설치하였을 지도 모를 커널 루트킷을 고려하여 커널 레벨에서 프로세스 및 네트워크 연결정보 등을 수집한다. 피해시스템에 설치되는 탐지 모듈은 피해시스템에서 백업 서버로 디스크 이미지를 전송하기 이전 단계에서 커널 레벨에서 현재 구동중인 프로세스와 연결된 네트워크 세션 정보를 LKM(lkm_ps, lkm_net)을 이용하여 추출한다.

<그림 5-2>는 커널 레벨의 프로세스 정보 추출을 위한 모듈을 로딩한 후 추출한 결과를 나타내고 있다.




<그림 5-2> 커널 레벨 프로세스 정보추출

<그림 5-3>은 프로세스 정보 추출모듈과 마찬가지로 LKM 형태로 구현하였고 현재의 네트워크 정보를 추출한 결과를 나타내고 있다.



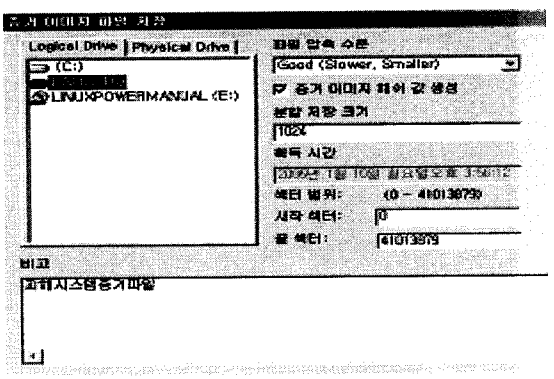

<그림 5-3> 커널 레벨 네트워크 정보추출

위의 결과에서처럼 추출된 프로세스와 네트워크 정보는 온라인상의 정보로서 휘발성이라는 특징을 가진 매우 중요한 정보이다. 이는 초기 온라인상의 상태 값으로 의미뿐만 아니라 실제 변조의 위험으로부터 벗어나 시스템에서 중요한 정보로 취급되어진다. 또한 숨겨진 프로세스나 숨겨진 네트워크를 연결 정보를 찾아 공격자의 경로를 추적할 수 있는 중요한 단서가 된다.

#### 5.2.2 디스크 이미지 백업

디스크 이미지 백업모듈에서는 준비된 백업시스

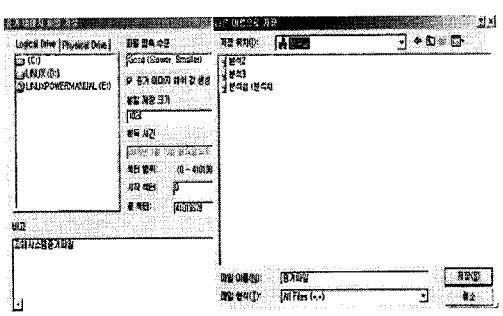
템에 피해시스템의 디스크 이미지를 파티션 별로 선택적으로 백업서버로 전송한다. 리눅스 시스템의 경우 다양한 파일시스템을 지원하기 때문에 피해 시스템의 OS에 구애받지 않고 복사된 파일시스템을 마운트해서 사용 가능하다. <그림 5-4>은 이미지 백업 모듈에서 백업이 필요한 파티션을 선택하도록 하는 화면이다.




<그림 5-4> 디스크 이미지 백업

### 5.2.3 백업 이미지 분석시스템으로 전송

각 모듈의 실행 결과는 네트워크를 통해 분석서버로 전송이 되는데 모듈 실행 결과 이전에 최초로 전송되어야 하는 정보는 피해시스템 정보와 사용자가 입력한 사건정보 그리고 사용자가 선택한 모듈에 대한 정보들이 기록되는 사건 정보 데이터이다. <그림 5-5>은 백업한 이미지 파일을 분석서버에 전송하는 화면을 나타낸 것이다.

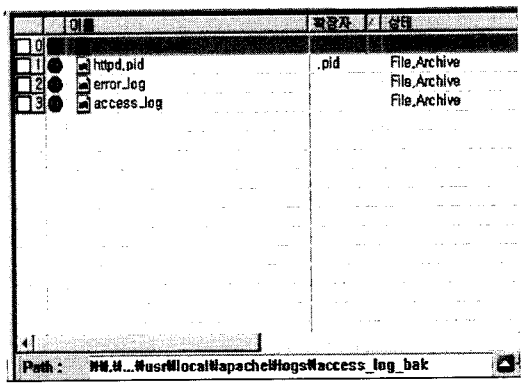


<그림 5-5> 백업 이미지 분석서버 전송

## 5.3 분석시스템 시험분석

### 5.3.1 복구데이터 분석

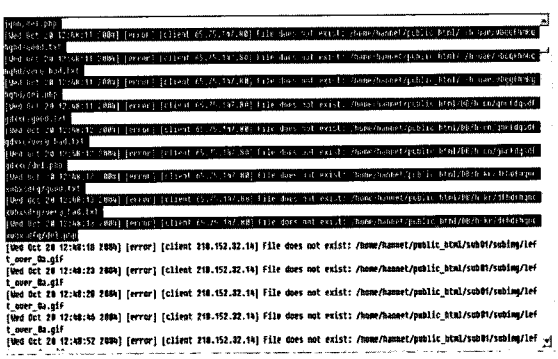
삭제된 파일, 데이터 복구, 지워진 로그파일은 복구하여야 한다. 복구된 데이터는 데이터 조각마다 모두 파일로 하나씩 만들어진다. 실제 복구된 데이터 중에서 로그파일을 확인해 보면 <그림 5-6>와 같이 제대로 복구가 되었음을 확인해 볼 수 있다.




<그림 5-6> 지워진 로그데이터 복구

### 5.3.2 로그파일 분석

<그림 5-7>과 같은 공격 형태의 로그가 남는 경우 로그에서의 알려진 공격을 탐지하기 위해 특정 흔적을 추출해야 한다. 로그에서 공격을 탐지하는 행위는 그것 자체만으로 크게 도움이 되지는 못한다. 하지만 로그가 나타난 흔적을 중심으로 시간대를 추측할 수 있는 근거를 마련하게 된다.



<그림 5-7> 공격흔적 로그

<그림 5-7>에서의 공격흔적로그에서 보면 2004년 10월 20일경에 12시 48분경에 수차례 접근을 시도하였던 것을 볼 수 있다. 또한 공격자의 침입자의 IP 주소가 65.75.147.80이라는 것을 확인하였다.

### 5.3.3 침입자 추적

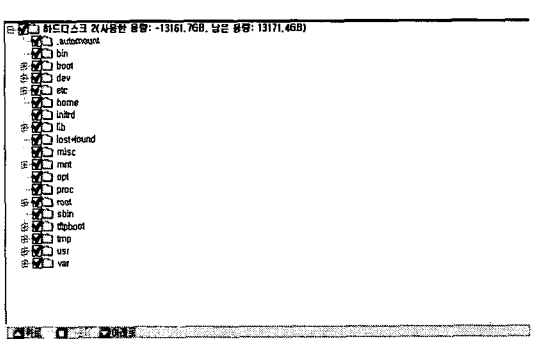
<그림 5-8>에서와 같이 피해시스템으로부터 수집된 정보를 분석한 결과 공격자의 것으로 추정되는 네트워크 정보를 Whois 서비스를 통하여 위치를 추적하였다. 그 결과 불법침입자가 접근한 불법 호스트는 미국 소재지의 개인사용자인 것으로 나오는 것을 확인할 수 있다.

```
Search for :
Search results for: NET-65-75-147-80-1

  CustName:   J. Nassif
  Address:    Private Residence
  Address:    Private Residence
  City:       Tulsa
  StateProv:  Ok
  PostalCode: 74115
  Country:    US
  RegDate:    2004-02-26
  Updated:    2004-02-26

  NetRange:   65.75.147.80 - 65.75.147.87
  CIDR:       65.75.147.80/29
  NetName:    NET-MANAGED
  NetHandle:  NET-65-75-147-80-1
  Parent:     NET-65-75-128-0-1

  NetType:    Reassigned
  Comment:
  RegDate:    2004-02-26
  Updated:    2004-02-26
```

<그림 5-8> 침입자 추적 결과

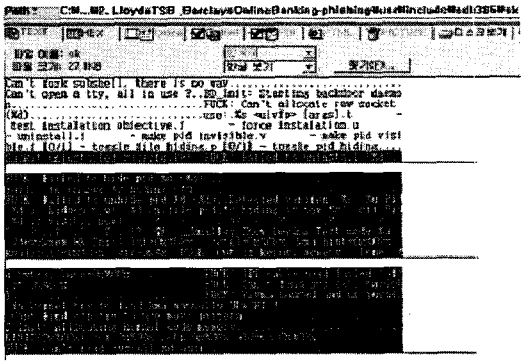
### 5.3.4 공격도구 찾기

백업된 이미지를 통해 지워진 파일들을 이미지별로 리포트 해주며 블록 단위로 파일형태에 따라 리포트 해주게 된다<그림 5-9>. 지워진 파일 블록 하나를 선택하였을 경우는 파일 내용을 헥사코드와 아스키 코드로 확인할 수 있다.




<그림 5-9> 파일구조

<그림 5-10>은 피해시스템의 이미지에서 공격도구로 SucKit 커널 루트킷을 사용한 흔적을 발견한 모습을 보여준다.



<그림 5-10> 커널 루트킷 SucKit

## 6. 결 론

커널 백도어로부터 공격이 탐지되는 즉시 네트워크와 프로세스 정보를 확보하고, 인멸 및 손실될 수 있는 잠재된 컴퓨터 범죄의 증거들을 백업하는 이미징 모듈이 동작하게 구현하였다. 이때 1차적인 정보만을 수집 후 2차 하드 전체 이미징 작업이 이루어지기 전에 대응모듈이 동작하여 보다 더 빠르게 피해시스템에 대한 복구와 대응을 할 수 있도록 하였다.

1차 정보수집이 끝난 다음은 대응모듈이 동작하도록 하여 피해시스템에 커널 백도어의 공격으로부터 입은 피해여부에 따라 허가되지 않은 모듈 실행중지 및 제거, 변경된 시스템 콜 테이블 원상복원, 변경된 모듈 리스트 복원, 변경된 프로토콜 핸들러 원상복원 등의 기능을 수행하도록 하였다.

1차 정보수집과 유형별 대응이 끝난 다음은 2차 하드 이미징 파일을 분석자가 지정한 안전한 백업 서버로 옮겨진 후 해쉬값을 부여하여 무결성을 확보하도록 하였다. 분석시스템으로 옮겨진 후 피해시스템의 이미지를 이용하여 삭제된 파일을 복구하고 침입자에 의하여 의도적으로 숨겨진 정보를 검색하는 모듈을 실행할 수 있다. 이때 리눅스 시스템에서 지워지거나 변조된 파일의 내용을 복구하고 숨겨진 공간에서의 정보를 검색하는 기능을 수행할 수 있다.

또한 해킹 도구 검색 모듈과 피해분석 모듈은 커널 루트킷과 일반 루트킷을 탐지하고, 침입자에 의하여 변조된 파일들과 설치된 해킹 도구들을 검색하도록 하였다. 커널에 대한 직접접근을 통하여

수집된 정보를 분석 및 비교하여 커널의 변조, 시스템 명령어의 변조 등의 상황을 판단하여 정확한 프로세스 및 네트워크 상태를 분석한다. 침입자의 IP주소를 기준으로 Whois 조회를 하여 공격자의 위치를 추적하도록 하였다.

본 논문에서 제안된 방법은 삭제된 로그파일을 복원하고 복원된 파일과 하드디스크의 이미지를 분석하여 해커의 위치를 찾을 수 있도록 하고 탐지시스템을 두어 커널 백도어가 탐지되는 즉시 해킹 피해를 입은 리눅스 시스템에서 증거 손실을 최소화하고 안전하고 신뢰할 수 있는 증거 보존, 그리고 신속하게 대응하도록 함으로써 시스템 피해를 최소화하여 시스템 관리자 또는 관련 수사기관들에게 많은 도움을 주고자 하였다.

## 참 고 문 헌


[1] 한국전자통신연구원, 차세대 해킹 기술 및 네트워크 안정성 분석 연구 보고서, 2001. 12

[2] J. R. Vacca, Computer Forensics: Computer Crime Scene Investigation, Charles River Media, 2002

[3] Albert J. Marcella and Robert S. Greenfield, Cyber and Preserving Evidence of Computer Crimes, Auerbach, 2002

[4] Gary Palmer, "A road map for digital forensic research," Digital Forensics Research Workshop, 2001, 2002, 2003.

[5] Daniel P. Bovet, Macro Cesati, "Undestanding the Linux Kernel", O'Reilly, 2000.

[6] M. Beck, H. Bohme, M Dziadzka, U Kunitz, R. Magnus, D. Verworner, "Linux Kernel Internal, 2nd Ed", Addison-Wesley, 1997.

[7] R. Card, E. Dumas, F. Mevel, "The LINUX KERNEL Book", John Wiley & Son, 1998.

[8] F. Butzen, C. Hilton, "The Linux Network", The M&T Books Slackware Series, 1998.

[9] A. Rubini, "Linux Device Driver", O'Reilly, 1998.

[10] Brian Carrier, "Open source digital forensics tools: the legal argument," @stake, Oct., 2003

[11] http://www.ietf.org/rfc/rfc1321.txt

[12] http://www.faqs.org/rfcs/rfc3174.html

[13] Kevin Mandia & Chris Prosise, "Investingating Computer Crime", Incident Response, Osborne/McGraw-Hill, 2001

[14] Peter gutmann, Secure Deletion of Data from Magnetic and Solid-State Memory, 6th USENIX Security Symposium, 1996

[15] http://www.cai.co.kr/solutions/product/etrust/security_infomation_mgmt/network_forensic/index.asp

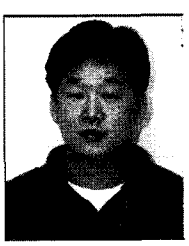
[16] http://www.arin.net/tools/whois_help.html

[17] http://whois.nic.or.kr

[18] http://www.apnic.net/apnic-bin/whois.pl

[19] http://www.ripe.net/perl/whois

[20] http://www.krcert.org


**전 완 근** (Wan-Keun Jeon)

- 정회원
- 1998년 2월 : 한서대학교 컴퓨터정보학과(이학사)
- 2000년 2월 : 한서대학교 대학원 정보보호공학과(공학석사)
- 2005년 2월 : 한서대학교 대학원 정보보호공학과 (공학박사)
- 2000년 3월 ~ 2006년 9월 : 한국정보 보호진흥원 선임연구원
- 관심분야 : 정보보호, 무선 인터넷 보안



**오 임 걸** (Im-Geol Oh)

- 정회원
- 1983년 2월 : 인하대학교 수학과 (이학사)
- 1986년 2월 : 인하대학교 수학과 응용수학전공(이학석사)
- 1993년 8월 : 인하대학교 통계학과 (이학박사)
- 2000년 8월 ~ 현재 : 인하대학교 컴퓨터공학과 박사과정
- 1995년 3월 ~ 현재 : 한서대학교 인터넷공학과 부교수
- 관심분야 : 컴퓨터 보안, 암호학, 컴퓨터 통신