韓國 컴퓨터情報學會 論文誌
第9卷 第1號, 2004. 3.

2004-9-1-2-3

컴퓨터 포렌식스를 지원하는 보안 감사/추적 모듈 설계

고 병 수*, 박 영 신**, 최 용 락***

A Design of Secure Audit / Trace Module to Support Computer Forensics

Byoung-Soo Koh *, Young-Shin Park **, Yong-Rak Choi ***

요 약

현재 거의 모든 운영체제는 여러 웹 서비스를 지원하기 위하여 운영체제 수준의 보안성을 제공하고 있다. 하지만, 많은 부분에서 운영체제 수준의 보안성이 취약한 것이 사실이다. 특히, TCSEC(Trusted Computer System Evaluation Ceiteria)에서 정의한 B2 레벨 이상의 보안성을 만족시키기 위한 Security Kernel 레벨에서의 감사/추적 기능이 필요하다. 이를 위해 시스템 콜 호출시 감사 자료를 생성하고, 모든 이벤트에 대해 동일한 포맷의 감사 자료를 생성하여 추후 역추적하는데 필요한 정보를 제공토록 한다. 본 논문에서는 안전한 증거 확보를 위하여 기존 리눅스 커널에 대해 변경 없이 적용 가능한 LKM(Loadable Kernel Module) 기법을 이용한 감사/추적 시스템 모듈을 제안하였다. 이는 침입탐지시스템과 같은 외부 감사 자료등을 동시에 활용할 수 있는 인터페이스를 제공하고, 시스템 관리자와 보안 관리자를 구분하여 역할기반의 안전한 시스템을 제공한다. 이러한 자료등은 추후 법적 대응이 가능한 컴퓨터 포렌식스의 자료로 활용하고자 한다.

Abstract

In general, operating system is offering the security function of OS level to support several web services. However, it is true that security side of OS level is weak from many parts. Specially, it is needed to audit/trace function in security kernel level to satisfy security more than B2 level that define in TCSEC(Trusted Computer System Evaluation

*, ** 대전대학교 컴퓨터공학과 대학원
*** 대전대학교 컴퓨터공학부 교수
- 본 결과물은 산업자원부의 출연금 등으로 수행한 지역전략산업 석박사 연구인력 양성사업의 연구결과입니다.

Criteria). So we need to create audit data at system call invocation for this, and do to create audit data of equal format about almost event and supply information to do traceback late. This paper proposes audit/trace system module that use LKM(Loadable Kernel Module) technique. It is applicable without alteration about existing linux kernel to ensure safe evidence. It offers interface that can utilize external audit data such as intrusion detection system, and also offers safe role based system that is divided system administrator and security administrator. These data will going to utilize to computer forensics' data that legal confrontation is possible.

▸ Keyword : 컴퓨터 포렌식스 / 역추적 / 침해사고대응 / LKM / LSM

Ⅰ. 서 론

인터넷 웜 바이러스나 정보전, 사이버 테러 등의 보안위협요소는 갈수록 증가하는 추세에 있으며, 이러한 위협요소들은 네트워크나 시스템뿐만 아니라 디지털 컨텐츠 보호에도 많은 위협이 되고 있다. 그리고 오늘날의 인터넷 구조는 서로 다른 레벨에서 다양한 공격지점이 발생할 수 있는 환경을 갖고 있다. 네트워크와 운영체제, 그리고 운영체제 위에서 실행중인 어플리케이션 등은 여러 취약점들을 노출하며 수많은 보안 이슈들이 제시되어 왔다. 하지만 대부분의 보안 업체들은 네트워크 레벨이나 어플리케이션 레벨에서 보안이 중심이 되어 가장 중요할 수 있는 서버레벨에 대해 소홀한 것이 사실이다. 미국 등의 선진국에서는 현재 시스템 보안을 위한 안전한 운영체제에 대해 상당한 투자가 이루어지고 있으며, 이를 통해 각종 보안 서버 관리에 대한 표준화가 이루어지고 있다[1][3].

미국의 경우, 신뢰성 컴퓨터 평가 기준(TCSEC : Trusted Computer System Evaluation Criteria) B1급 이상의 컴퓨터 시스템에서는 안전한 OS의 구현은 대부분 보안 커널(Security Kernel)로 구현하고 있으며, B2급 이상의 평가를 받은 컴퓨터 시스템에 대해서는 해외로 수출을 금지하고 있다[1].

본 논문에서는 보안 커널 수준의 역할기반을 이용하여 시스템의 여러 로그 파일을 보호하고, 이러한 안전한 로그 정보를 이용하여 마킹 기반의 IP 역추적을 통한 보안 감사/추적 모듈을 제안하였다. 수집된 증거 정보들은 제안된 보안 정책에 따라 생성되며, 무결성을 제공받는 증거 자료를 기반으로 제안된 IP 역추적 모듈을 이용하여 추적이 가능하다. 여기에 추후 침해사고 대응을 위한 컴퓨터 포렌식스 자료로도 활용할 수 있도록 하였다.

본 모듈의 개발 환경은 리눅스를 기반으로 하였고 클라이언트는 윈도우즈 계열 운영체제를 기반으로 설계하였다. 2장에서는 컴퓨터 포렌식스에 관한 관련 연구 내용을 기술하였으며, 3장에서는 보안 커널 기반의 추적/감사 모듈을 설계하였고, 4장에서는 3장에서 설계한 모듈을 구현하여 리눅스 운영체제에 적용해 보았다. 5장 결론에서는 본 논문의 주된 성능과 향후 연구 내용을 서술하였다.

Ⅱ. 배경 연구

1. 컴퓨터 포렌식스

오늘날 비즈니스 커뮤니케이션의 70%가 전기, 전자적으로 이루어지고 있으며, 이에 따른 모든 순기능과 역기능에 대하여 결정적인 거래증거는 컴퓨터와 네트워크 안에 있다. 이러한 환경으로부터 디지털 전자 컨텐츠의 모든 접근행위에 대하여 전자적 증거물을 수집분석 및 역추적 등의 절차를 수행하고 법적 증거물 제시와 적절한 대응조치를 할 수 있도록 새롭게 출현한 기술이 컴퓨터 포렌식스이다[2].

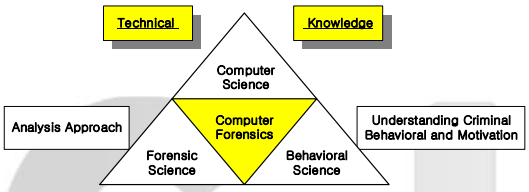

그림 1. 컴퓨터 포렌식스 영역
Fig. 1 Computer Forensics domain

ESM(Enterprise Security Management)과의 차이점에 대해 의문점을 갖는 사람들도 있으나, ESM은 보안제품간의 상호호환성을 바탕으로 관리할 수 있는 제한된 영역을 통제하는 시스템이다. 반면에, 컴퓨터 포렌식스는 침해사고 대응 방법을 위해 여러 문제점들을 분야별로 구분시켜 놓음으로써 향후 침해사고 발생시 증거를 획득, 보존하여 법적 대응이 가능하도록 하는 시스템이라 할 수 있다. 또한 포렌식스는 (그림 1)과 같이 컴퓨터 및 행동과학적 측면의 다양한 기술과 지식을 필요로 한다.

2. 포렌식스을 이용한 시스템 분석 방법

컴퓨터 포렌식을 이용한 시스템 분석 방법에는 증거 보존 및 분석을 위한 시스템 분석 방법과 해킹과 같은 공격 흔적을 찾기 위한 무결성 도구를 이용한 방법, 그리고 공격 기법을 분석하여 침해 여부를 판단하고 로그 파일 분석 및 복구를 통한 증거 수집을 이용한 방법 등이 있다. 분석 시스템을 이용한 분석방법은 공격 흔적을 보존하기 위해 수행

중인 프로세스 상태 및 포트, 현재 네트워크정보, 사용자와 터미널에 대한 로그 정보, 현재 사용자, 현재까지 변경된 모든 파일 등을 조사해야 한다. 무결성 도구를 이용한 시스템 변조 유무 확인 방법은 공격기법 및 웜바이러스(바이러스 포함)등을 분석하는 방법, 공격자가 삭제한 파일이나 데이터를 복구하여 증거를 수집하는 방법, 로그파일을 점검함으로써 파일의 유출 및 도난 여부를 확인하는 방법, 디스크 복구를 통한 증거 수집 방법등이 있다. 현재 컴퓨터 침입에 대응하고 법적 효력을 갖도록 하는 무결성을 보장하는 디지털 증거를 확보하여, 물리적 증거물과 동일한 효력을 발생할 수 있도록 레포트화 할 수 있는 기술이 필요하다.

3. TCP 연결 역추적 기술

공격자는 침입시도의 성공 여부와 상관없이 최대한 자신에 대한 정보를 감추기 위해 여러 다른 시스템들을 경유하는 공격을 수행한다. 이러한 우회공격에서 경유된 시스템들로부터 정보를 획득하여 이를 바탕으로 실제 공격자의 위치를 역추적하는 시스템을 우회공격 근원지 역추적 혹은 연결체인 역추적 기술이라 한다[4]. 이러한 TCP 연결 역추적 기술은 <표 1>과 같이 호스트 기반 연결 역추적 기술과 네트워크 기반 연결 역추적 기술로 분류된다.

표 1. TCP 연결 역추적 기술 분류
Table. 1 Technique classification of TCP chain traceback

	수동적인 방법	능동적인 방법
호스트 기반	DIDS CIS AIAA	Caller ID
네트워크 기반	Thumbprint-based Timing-based Sequence Number-based	IDIP, CITRA, AN-IDR SWT MTBS(제안 방식)

III. 보안 커널 기반의 감사/추적 모듈 설계

본 논문에서는 보안 커널 기반의 감사 및 추적을 위하여, 일반적으로 파일의 소유권 변경이 가능한 로그 파일을 대신, 커널 레벨에서의 로그 엔진을 구현하였다. 그리고 해당 보안 정책을 적용한 후 안전한 데이터베이스로 저장되어 컴퓨터 포렌식스 자료와 추적 시스템의 기본적인 자료로 사용하도록 설계하였다. (그림 2)는 보안 커널기반의 감사/추적 시스템 구성을 나타내고 있다. 이 시스템은 리눅스 운영체제에 다음과 같은 기능의 구현을 고려하였다.

- 운영체제 수준에서의 강제적 접근 통제(Mandatory Access Control)
- root 권한의 제한
- 커널 모드의 안전한 추적/감사 기능 제공
- 시스템 관리자와 보안 관리자의 역할 분리
- 네트워크에 의한 접근 통제

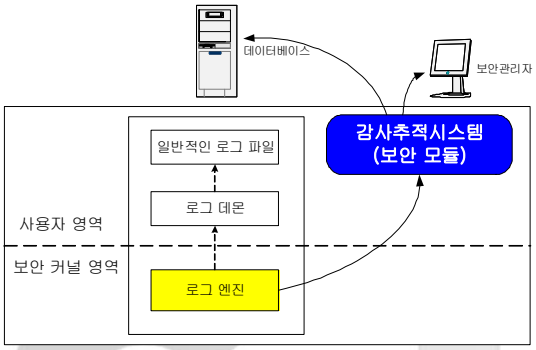

그림 2. 감사/추적 시스템 구성도
Fig. 2 Composition diagram of adult/trace system

1. 로그 엔진

커널 모듈에서 보내는 로그 정보를 DB파일에 저장하기 위해서 커널영역과 유저영역사이에 통신 인터페이스로 Character Device를 사용한다. 커널 모듈에서 나오는 로그 정보를 로그 데몬이 Character Device를 3초 단위로 로그 메시지가 있는지 확인한다. 메시지가 있다면 실시간으로 DB파일에 저장한다. 이러한 커널 레벨의 로그 저장 방식은 실시간으로 저장하여 능동적인 대응에 필요한 추적 및 감사 자료로 활용되며, 컴퓨터 포렌식스를 위한 기초 자료로 사용된다. (그림 3)는 해당 알고리즘을 나타내며 (그림 4)는 Character device에 로그가 저장되는 과정을 도식화한 다이어그램이다.

2. 보안 감사 모듈

보안 모듈은 커널 소스 수정이 필요 없는 커널 모듈 형식으로 커널에 애드-온 되도록 설계하였다. 보안 모듈은 참조 모니터링을 통한 강제적 접근 통제 구현과 capability를

통한 네트워크 서비스 프로그램 제어를 제공하여 접근 통제 기능이 강화되었다.

```
memset(object_path, '\0', PATH_MAX);
 if( SecuOS_Check_Base(dentry,SECUOS_WRITE) > 0 ) {
  sprintf(error_msg,"rmdir error no permission
SECUOS_WRITE");
  secuos_log(2,__FUNCTION__
SecuOS_Absolute_Path(dentry, object_path), error_msg);

  if(!error_flag)
      path_release(&nd);
   putname(name);
   return -1;
}
```

그림 3. 로그 파일 DB 저장 알고리즘
Fig. 3 DB store algorithm to log file

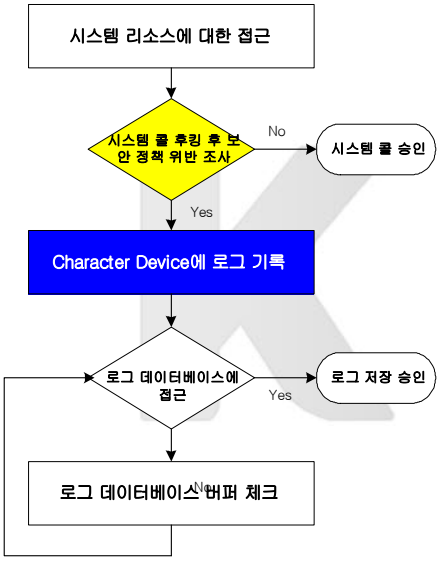

그림 4. Character Device 로그 저장
Fig. 4 Characte Device log restore

참조 모니터링은 임의의 프로세스가 시스템 자원에 접근할 경우 실제시스템 콜을 후킹(hooking)한 시스템 콜이 호출되고, 후킹된 시스템 콜에서는 접근 시도한 주체가 객체에 대해 요청한 권한이 있는지 체크하게 된다. 시스템 콜 뿐만 아니라 커널에서 심볼 테이블에 등록한 파일 오퍼레이션 심볼에 대해서도 후킹을 시도 한다.

시스템 콜을 후킹 후, 그 시스템 콜을 호출한 주체(Subject)가 보안 정책에서 해당 권한이 명시되어 있으면 원래의 시스템 콜을 호출할 수 있도록 해주며, 보안 정책에 명시되어 있지 않으면 그 주체에 해당하는 정보(level, time, uid, gid 등)를 Charater Device에 쓰게 되며, 상주해 있는 로그 데몬이 Character Device에 쓰여진 로그 정보를 DB 파일에 저장하게 된다.

보안 정책은 주체나 객체에 대한 접근 권한과 root 권한을 가진 사용자로 하여금 시스템이나 접근 기록을 변조 또는 삭제되지 않도록 설정하였다.

3. 보안 추적 모듈

제안된 MTBS(Marking TraceBack System) 모듈은 로그 데이터베이스에 저장된 자료를 기반으로, 해당 IP를 추적하여 침입자의 경로를 알아내는 모듈이다.

MTBS는 크게 응답패킷 Capture 모듈, 마킹 Write 모듈, 마킹 패킷 Send 모듈로 구성된다. 또한, 각 경유하는 호스트의 IP 주소를 저장하기 위한 DB와 분산된 MTBS에서 경유 호스트와 공격자의 정보를 알리기 위한 AMR(Alert Message Report)이 존재한다. (그림 5)는 MTBS의 흐름도를 나타낸다.

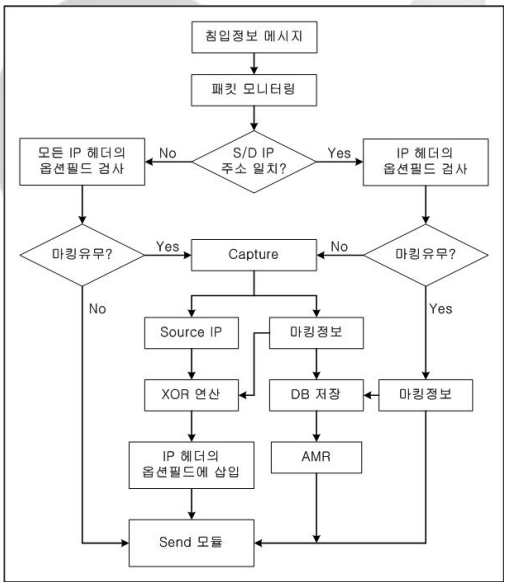

그림 5. MTBS의 흐름도
Fig. 5 Flow diagram of MTBS

Ⅳ. 감사/추적 모듈 구현

1. 감사를 위한 로그 정보 수집

서버측의 에러 로그의 정보를 기반으로 침입자를 역추적할 수 있는 자료로 활용하며, 침해사고 대응을 위한 컴퓨터 포렌식스의 중요한 자료가 된다. 이 로그 파일은 커널기반의 안전한 파일로 저장되게 되며, 무결성 보장을 위하여 별도의 데이터베이스에도 추가적으로 저장된다.

2. 보안 정책 설정

보안 정책 정보는 TCSEC에서 정의한 C2 레벨 이상의 보안성을 만족시키기 위한 감사 자료 생성 기능을 제공하며, 모든 감사 이벤트에 동일한 포맷의 감사 자료를 생성한다. 또한 보안 모듈이 올라가 있는 리눅스 서버에 설정된 보안 관련 정책을 보여준다. 리스트 창에 나타난 보안 정책은 일반 정책과 프로세스 보호 관련 정책, 그리고 Bind 서버에 관련한 정책으로 구분지어 나타내어진다. 기존 보안 정책을 나타내 주는 것뿐만 아니라, 보안 관리자가 보다 쉽게 정책을 추가하거나 삭제, 수정을 가능하게 해준다. (그림 6)은 보안 정책의 설정 내용이다.

```
555064:769:/bin/login:1:0:229841:769:/etc/shadow:0-0
555064:769:/bin/login:7:0:767060:769:/var/log/lastlog:0-0
391825:769:/usr/sbin/sshd:1:0:229841:769:/etc/shadow:0-0
391825:769:/usr/sbin/sshd:16:0:-1:10:CAP_NET_BIND_SERV
ICE:0-0
.....
343198:769:/usr/bin/procmail:16:0:-1:10:CAP_NET_BIND_SE
RVICE:0-0
327938:769:/root/project/log_daemon/secuos_logd:16:0:-1:3
1:CAP_PROTECTED:0-0
```

그림 6. 보안 정책 설정
Fig. 6 Security policy creation

3. 역추적을 통한 증거 수집

감사 모듈에 의해 저장된 로그 정보를 기반으로 공격에 대한 응답패킷을 Capture하기 위해 침입정보에서 Source IP와 Destination IP의 주소를 교환하여 MTBS[1]의 응답패킷의 정보에 입력한다. 이것은 우회공격의 연결특성에 따라 공격에 대한 응답패킷은 공격경로의 역방향으로 전달되기 때문이다. 실험 환경은 (그림 7)과 같다.

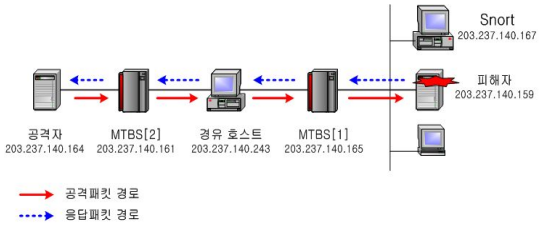

그림 7. 실험 환경
Fig. 7 Test environment

(그림 8)은 MTBS[1]에서 Capture한 응답패킷에 Source IP 주소와 옵션필드를 XOR 연산하여 그 결과 값을 IP 헤더의 옵션필드에 새롭게 마킹한 결과이다. 마킹한 패킷은 Send 버튼을 통해 목적지 주소로 보낸다.

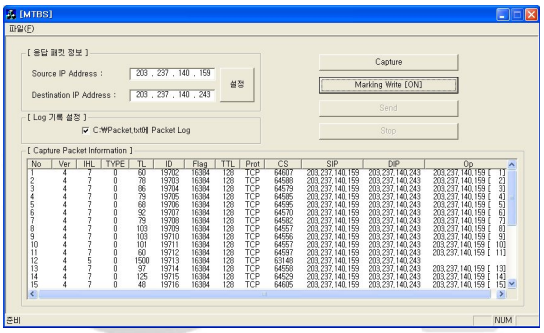

그림 8. MTBS[1]에서 Capture한 패킷을 마킹한 결과
Fig. 8 The result of marking packet

MTBS[2]에서는 Source IP 주소와 옵션필드의 마킹여부를 검사하여 마킹이 존재하는 경우 이를 Capture한다. Capture된 옵션필드에 다시 경유 호스트의 주소를 마킹하기 위해 MTBS[2]는 응답패킷에서 Source IP 주소와 옵션필드를 XOR 연산하여 그 결과 값을 IP 헤더의 옵션필드에 새롭게 마킹한다.

그리고, Capture Packet Information에서 Destination IP 주소를 파악할 수 있으므로 마킹한 결과를 목적지 주소로 보낸다. 그리고 MTBS[2]는 Source IP 주소가 전송되는 패킷들을 모니터링 하여 옵션필드가 존재하는지 검사한다. 만약 존재하는 경우 전송했던 패킷의 옵션필드와 일치하는지 검사한다.

표 2. SWT와 제안 방식의 성능 비교
Table. 2 Performance comparison of SWT and proposal method

항 목	SWT	MTBS
침입정보	IDS와 연동하여 침입탐지 정보를 전달 받음	IDS와 연동하여 침입탐지정보를 전달 받음
서비스 타입	연결형 서비스	연결형 서비스
마킹 방법	가상 null 스트링문자를 사용하여 마킹	Source IP와 옵션필드를 XOR하여 결과 값을 마킹
마킹 정보	다른 문자들과 구별되는 랜덤한 값	모든 경유 호스트들의 주소를 XOR 연산한 값
마킹 위치	패킷의 데이터 영역을 사용	IP 헤더의 옵션필드를 사용
암호화된 연결	암호화 된 연결인 경우 역추적이 불가능	SSL과 SSH를 이용하는 연결에서도 역추적 가능
적용 환경	Active 네트워크	현재의 네트워크에 적용
실시간 역추적	실시간 역추적 가능	실시간 역추적 가능

위의 <표 2>는 현재 가장 활발히 연구가 진행되고 있는 SWT와 본 논문에서 제안한 MTBS를 비교한 것이다.

V. 결론

리눅스 운영체제는 전 세계의 개발자들이 상호 협력하여 개발하고 있는 공개 소스 기반의 운영체제다. 따라서 유사 운영체제에 비해 더욱 안전한 보안 기능이 구비되어야함에도 불구하고 감사 자료 생성 기능에 있어 매우 취약한 상황이다. 즉, 리눅스 운영체제에 대한 C2레벨 이상의 보안성을 만족시키기 위한 기본 기능 중의 하나인 감사 자료 생성 기능은 접근 제어나 기타 공개 프로젝트 등에서 성공적인 결과를 보여주고 있지 못하다.

본 논문의 결과를 통해 리눅스 운영체제에서도 유사 운영체제와 동일한 수준의 감사 자료 생성기능이 설계 목적을 만족하면서 추가될 수 있는 가능성을 확인하였다. 이 시스템은 리눅스 운영체제에 대해 기존 커널의 변경없이 LKM(Loadable Kernel Module)기법을 통해 동적으로 로딩이 가능하도록 설계되었기 때문에 커널의 재수정이나 시스템의 리부팅이 없이 적용될 수 있다. 또한 침입 탐지 시스템과 같은 외부의 프로세스가 용이하게 생성한 감사 자료를 활용할 수 있는 인터페이스를 가지고 있으며 실제 리눅스 운영체제에 대한 침입 시도 탐지의 가능성을 확인하였다. 이러한 감사 자료를 바탕으로 마킹 기반의 역추적 시스템 구현이 가능하고, 추후 발생할 수 있는 보안침해 사고에 대비한 컴퓨터 포렌식스 자료로의 사용은 상당히 효율적이라 할 수 있다. 현재의 모든 운영체제나 네트워크는 소극적 방어이기 때문에 제안 시스템에서는 컴퓨터 포렌식스나 실시간 역추적을 제공함으로써 능동형 방어 기술의 근본적인 기법을 제공할 수 있다.

LSM(Linux Security Module)은 리눅스 운영체제에 대한 다양한 보안 정책을 융통성있게 지원하기 위해 커널 패치 형식으로 제안된 범용 프레임워크이다. LSM 프레임워크와 연동하는 LKM 기반의 보안 모듈은 접근 제어 용도가 주를 이루고 있다. LKM 기반의 감사 용도의 보안 모듈도 LSM 프레임워크와 연동된다면 현재의 LSM에서 제공하는 여러 후크들과 보안 필드들을 이용할 수 있도록 지속적인 연구가 필요할 것이다.

참고문헌

[1] 박태규, 임연호, "리눅스 커널 기반의 안전한 OS 개발", (2001)

[2] 고병수, 박영신, 최용락, "보안침해사고 대응을 위한 컴퓨터 포렌식스 기술동향", 한국인터넷정보학회지, 4권 1호, pp.37-46(2003)

[3] Federal Register, Vol.65, No.10, Rules & Regulation(Part III : Department of Commerce, Bureau of Export Administration, Revision to Encryption Items ; Interim Final Rule), (2000)

[4] 박영신, 고병수, 최용락, "침입자 역추적을 위한 IP 헤더 마킹기법에 관한 연구", 한국인터넷정보학회 추계학술대회, 제4권 2호, p323-326

[5] Gery Herman, "Operating Systems Security for midrange and Large Computer : Overview", DATAPRO, (1996)

[6] http://www.radium.ncsc.mil/tpep/process/procedures.html, Overview of the Trusted product Evaluation Program(TPEP).

[7] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman, "Role-Based Access Control Models", IEEE Computer, Volume 29(1996)

[8] http://www.ccic.gov/pubs/blue97/nsa/secureos.html, Secure Operating System Development.

[9] Morrie Gasser, "Building a Secure Computer System", Van Nostrand Reinhold Company Inc., (1988)

저 자 소 개

고 병 수

2000년 호남대학교 컴퓨터공학과 (공학석사)

현 재 대전대학교 컴퓨터공학과 (박사수료)

<관심분야> Secure OS, 컴퓨터포렌식스, PKI 응용

박 영 신

2004년 대전대학교 컴퓨터공학과 (공학석사)

현 재 대전대학교 컴퓨터공학과 (박사과정)

<관심분야> 역추적, 컴퓨터 포렌식스

최 용 락

1989년 중앙대학교 전자계산학과 (박사)

1982년 3월~1986년 1월 한국전자통신연구원 선임연구원

현 재 대전대 컴퓨터공학부 교수

<관심분야> 컴퓨터통신보안, 컴퓨터 포렌식스, DRM