다중등급 보안 리눅스 구현 및 시험평가
(Implementation and Evaluation of Multi-level Secure Linux)

손 형 길 † 박 태 규 †† 이 금 석 †††
(Hyung-Gil Shon) (Tae-Kyou Park) (Kuem-Suk Lee)

요 약 기존 침입차단시스템과 침입탐지시스템과 같은 외곽방어 개념의 보안대책은 전산망 내의 중요 서버를 보호함에 있어서 그 한계를 갖는다. 본 논문에서는 보안 리눅스 운영체제를 구현함에 있어 중요 요구사항으로 TCSEC B1급의 다중등급 보안 기능과 해킹 방어 기능을 정의하고, 이를 시스템 후킹을 통한 리눅스 커널 모듈로 구현하였다. 본 구현 시스템의 시험 평가로서 강제적 접근제어 및 해킹 방어 시험, 성능 측정을 실시하여 보안성 및 안정성, 가용성을 입증하였다. 구현된 보안 커널 기반의 리눅스 운영체제는 Setuid 이용 공격, 버퍼 오버플로우 공격, 백도어 설치 공격, 경유지 이용 공격 등의 해킹을 커널 수준에서 원천적으로 차단하며, 평균 1.18%의 성능 부하율을 나타내어 타 시스템보다 우수하다.

키워드 : 보안 리눅스, 보안 커널, 다중등급 보안

Abstract A current firewall or IDS(intrusion detection system) of the network level suffers from many vulnerabilities in internal computing servers. For a secure Linux implementation using system call hooking, this paper defines two requirements such as the multi-level security function of TCSEC B1 and a prevention of hacking attacks. This paper evaluates the secure Linux implemented in terms of the mandatory access control, anti-hacking and performance overhead, and thus shows the security, stability and availability of the multi-level secure Linux. At the kernel level this system protects various hacking attacks such as using Setuid programs, inserting back-door and via-attacks. The performance degradation is an average 1.18% less than other secure OS product.

Key words : Secure Linux, Security Kernel, Multi-level Security

1. 서 론

기존 침입차단시스템(firewall)과 침입탐지시스템(IDS)과 같은 응용 수준에서의 보안대책은 전산망 내의 중요 서버를 보호함에 있어서 그 한계를 갖는다. 이러한 응용 수준에서의 보안 대책은 근원적으로 중요한 정보를 담고 있는 인트라넷 내부 서버에 대한 보호를 위해서는 한계를 가지고 있으므로 충분한 보안 대책이 될 수 없으며, 컴퓨터 시스템의 운영체제 자체가 기본적으로 보안상 취약성을 내포하고 있는 상태에서는 응용 수준의 어떠한 보안 대책도 사상누각을 쌓는 결과를 초래할 것이라는 지적들이 제기 되었으며[1,2], 침입차단시스템과 침입탐지시스템, 보안 관제시스템 등을 갖추고도 불법적인 해킹을 당하여 중요한 정보들이 파괴, 유출되는 사건들이 빈번히 발생하고 있다[1,3,4,5]. 따라서 이러한 보안 한계를 극복하기 위해서는 보안 운영 체제의 활용이 서버 보안 측면에서 기본적인 전제가 되어야만 한다. 인터넷을 구성하는 기본 요소인 운영체제의 보안성 평가의 잣대는 미 국방성 표준 규정(DoD 5200.28-STD)인 신뢰성 컴퓨터시스템 평가기준(TCSEC, Trusted Computer System Evaluation Criteria)과 ISO 국제공통평가기준인 CC(Common Criteria)의 Multi-level Operating System Protection Profile (MLS-OS PP)[6]가 될 수 있다. 오픈 소스 운영체제인 리눅스는 개별적으로 서버를 운영하는 사람에서부터 클러스터링을 이용한 슈퍼컴퓨터까지 개인용, 교육 연구용, 국방용, 행정용등 광범위하게 그 활용분야 및 영역이 넓혀지고 있다. 그러나 리눅스 운영체제의 보안성은 TCSEC 기준으로 볼 때 C2급(사용자 식별 인증, 임의적 접근제어(DAC, Discre-

† 비 회 원 : 행정자치부 행정망운영과장
shonhg@gcc.go.kr
†† 종신회원 : 한서대학교 컴퓨터정보학과 교수
tkpark@hanseo.ac.kr
††† 정 회 원 : 동국대학교 컴퓨터공학과 교수
kslee@dgu.ac.kr
논문접수 : 2002년 11월 28일
심사완료 : 2003년 4월 21일

tionary Access Control), 객체 재사용 방지, 감사 추적 등의 기능)에 해당되는 보안 기능을 갖고 있다[7]. 따라서 리눅스를 중요한 정보처리용 서버로 사용하기 위해서는 C2급의 한계를 보완하여 B1급 수준의 운영체제로 개발하고자 하는 요구가 제기되고 있다. 본 논문에서는 C2급에 해당되는 기존 리눅스 운영체제(Kernel 2.4)에서 시스템 호출 후킹(System call Hooking) 기법과 LKM (Loadable Kernel Module) 방법을 이용하여 커널 수준에서 TCSEC B1급 이상에서 요구하는 다중등급 보안(MLS : Multi- level Security) 커널을 구현하였다.

본 논문 구성은 2장에서 보안 운영체제 개발을 위하여 리눅스의 해킹 취약성 분석 및 보안 관련 연구 동향을 분석 하고, 3장에서 보안 운영체제의 요구사항을 살펴보고, 4장에서는 보안 운영체제의 2가지 중요 요구사항 중 하나인 TCSEC B1급의 구현된 주요 기능을 설명하였다. 5장은 또 다른 중요 요구사항인 운영체제 수준에서의 해킹을 방지하기 위한 기능을 보이며, 6장에서 통합보안관리와 7장에서 앞서의 구현 기능에 대한 시험으로 강제적 접근 접근제어(MAC, Mandatory Access Control)와 성능 및 해킹방어 시험 결과에 대하여 설명한다.

2. 리눅스 취약성 및 보안대책 연구

현재 보급되고 있는 리눅스 운영체제(OS)의 취약성을 분류해보면 사용자 인증의 패스워드 공격, 버퍼 오버플로우 및 경주 조건(Race Condition) 공격 등을 통한 root 권한탈취 공격, 파일 소유자의 임의적 판단에 의한 접근제어, 데몬 공격, 서비스 거부 공격(Denial of Service), 바이러스 및 웜 감염, IP 스푸핑, 패킷 스니퍼링 등과 같은 다양한 유형의 취약점을 지니고 있다[8]. 그러므로 기존 네트워크 수준에서의 보안 방식으로는 이와 같이 다양한 운영체제의 내부적 취약성 방어에는 한계를 가질 수밖에 없다. 따라서 운영체제의 커널 수준에서 보안 기능을 탑재한 보안 커널(Security Kernel) 연구가 1980년 초기부터 미국에서 연구 개발되어 오고 있다[9,10,11,12]. DoD 정의에 의하면 보안 커널은 "컴퓨터 보안에 있어서 참조 모니터(Reference Monitor) 개념을 구현한 TCB(Trusted Computing Base)의 하드웨어, 펌웨어, 소프트웨어 요소이며, 모든 접근을 중재해야만 하고, 수정으로부터 보호되어야 하며, 작고 정확성이 검증되어야 한다[13]"고 정의하고 있다.

리눅스의 B1급 보안 커널 관련 연구는 미국 NSA의 seLinux[14], NPS의 MLS-Linux[15], Wirex의 Immunix[16] 등이 발표되었으나, 기존의 이러한 연구의 구현 방식은 커널 소스를 직접 수정하는 방식으로서 커널의 패치나 버전 변경에 따라 재개발이 수시로 필요하거나 커널의 재 컴파일이 필요하여 시스템을 중단시켜야 하는 등의 문제와 개발비용이 커지고 기존 응용 프로그램의 호환성이 부족해지는 문제점을 내포하고 있다. 한편 미국의 경우, B2급에 대해서는 전략적으로 대외 수출을 통제하고 있다[17]. 국내의 국책 연구소, 소수 벤처기업과 대학 등에서도 대부분 커널 직접 수정방식으로 연구 개발을 해오고 있는 실정이다. 그러나 본 논문에서 구현한 방식인 후킹과 LKM 방식은 기존 시스템 호출을 수정 없이 인터셉트함으로써 커널 재 컴파일이 불필요하고, 시스템의 중단 없이 커널 모듈을 삽입, 제거가 가능하므로 적은비용으로 호환성을 유지하므로 앞서 언급한 문제점을 해결할 수 있다. 이러한 후킹 방식을 이용한 B1급 리눅스 보안커널은 아직 발표된 사례가 없으며, 리눅스 보안 제품으로서 후킹방식으로 구현된 제품은 본 논문의 7장에서 벤치마킹한 CA사의 "eTrust Access Control"[18]이 있으나 C2급으로 알려져 있다.

3. 보안 리눅스 운영체제 요구사항

리눅스 보안 커널을 개발함에 있어 중요하게 고려한 2가지 요구사항은 설계를 위한 일반적 요구사항과 구현을 위한 기능적 요구사항이다. 설계를 위한 요구사항은 일반적인 보안 시스템 개발에서 요구하는 것과 유사하게 고려될 수 있으며, 구현시의 기능적 요구사항은 중요한 정보에 대하여는 보안 등급을 부여하여 강제적 접근제어를 수행하는 TCSEC B1 클래스의 기능적 요구사항과 OS 자체에서 개발 시부터 가지는 해킹 취약성을 방어하는 해킹방지 기능이 현대에서는 필수적으로 요구된다.

3.1 보안 운영체제의 설계 요구사항

보안 운영체제를 구현하기 위해서 설계 시에 OS가 유지하는 정보에 대한 기밀성, 무결성, 가용성을 보장하도록 하여야 한다[19]. 따라서 다음과 같은 요소를 고려하여야 한다. ① 특권의 최소화(Least of Privilege) : 부적절하거나 악의가 있는 공격으로부터 피해를 최소화하기 위해 root를 포함한 사용자와 프로그램은 가능한 최소한의 권한을 사용하여 운영되어야 한다. ② 메커니즘의 경제성 : 보호 시스템의 설계는 작고 검증이 가능하고 처리가 빨라야 한다. 그 이유는 보안성에 대한 완전한 분석, 시험, 검증이 가능하며, 부가적인 처리로 인한 성능상의 오버헤드가 최소가 되어야 하기 때문이다. ③ 완전한 중재 : 모든 접근 시도는 통제되어야 하고,

우회하려는 어떠한 시도도 접근 통제가 가능해야 한다. ④ 허가기반 : 기본적으로 접근은 거부되어야만 한다. 일반적으로 설계 시 접근불가 항목을 정의하기 보다는 접근가능하게 설계하는 오류로부터 벗어나야 한다. ⑤ 특권의 분리 : 이상적으로, 객체(Object)에 대한 접근은 사용자 인증과 보안등급, 또는 암호 키 등과 같이 한 가지 이상 조건들을 사용하여야 한다. 이러한 방식으로 보호 메커니즘을 우회한 프로세스도 보호하고자 하는 대상에 접근하지 못하게 한다. ⑥ 사용의 용이성 : 사용자 및 보안 관리자가 쉽게 사용하고 쉽게 보안 설정 사항 등을 통제하며 로그 정보를 이용함으로써 보호 메커니즘을 우회 할 가능성을 제거한다.

3.2 보안 운영체제의 기능적 요구사항

전통적으로 군사적 목적 및 비밀 수준의 중요 정보를 보호하기 위하여 적용되었던 보안 운영체제 개발은 TCSEC의 B1급 이상의 요구사항을 적용하고 있다. 다시 말해서 CC의 EAL4 등급이상의 MLS-OS PP가 적용되고 있다. 부가적으로 TCSEC의 요구사항에 맞게 구현된 시스템은 시스템 내부의 중요한 정보를 효율적으로 보호하였지만, 시스템의 성능을 저하시키는 하나의 요인이 되기도 한다. 따라서 이러한 성능 부하율을 최소화 하도록 설계되고 구현되어야 한다. 한편 최근 들어 폭발적으로 증가하는 인터넷 사용 추세를 감안할 때 보안 운영체제에 대한 필요성은 급증할 것으로 예측되는 바, 군사용, 금융기관 등의 중요 정보처리용과 일반사용자를 위한 공통적인 보안 운영체제의 개발이 필요하며, 이러한 요구사항을 바탕으로 보안 리눅스를 개발하기 위한 기능적 요구사항은 사용자 식별 및 인증, 강제적 접근제어, 레이블 보안, 감사 추적, 보안 관리, 해킹 방지 등으로 크게 정의하고 특수한 기능들은 사용자가 선택적으로(optional) 적용할 수 있도록 해야 한다.

4. TCSEC B1급 기능 구현

4.1 구현된 보안 리눅스 운영체제 개요

구현된 보안 리눅스 시스템은 (그림 1)에서와 같이 커널 수준과 사용자 수준으로 크게 구분된다. 커널 수준은 보안 커널로서 시스템 호출을 후킹하여 보안 기능을 수행하는 참조 모니터와 커널 프로세스를 통제하여 해킹을 방지하는 해킹방지(Anti-Hacking) 모듈로 구성된다. 또한 커널 수준의 모든 보안 기능은 TCB로 통칭된다. 사용자 수준에서는 MLS 가 적용되는 사용자 프로세스, 사용자 식별 인증 모듈과 감사 추적을 위한 로그 수집 데몬 모듈로 구성된다.

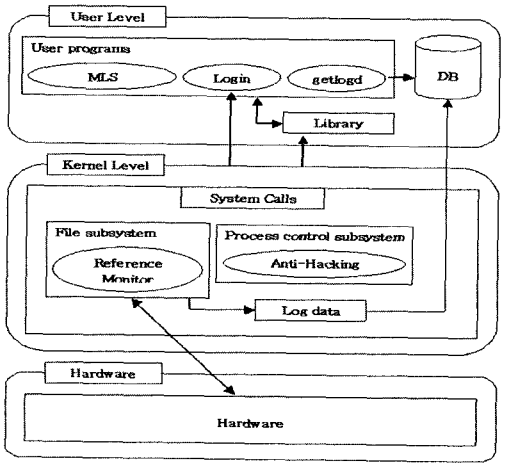


그림 1 보안 리눅스 OS 구성도

4.2 사용자 식별 인증

본 논문에서 구현된 보안 커널은 기존의 사용자 인증 정보(사용자 ID, 패스워드)에 보안등급과 보호범주를 추가하여 사용하도록 하였다. 또한 시스템 관리자 계정인 root와 별도로 root의 상위 수준에 존재하는 보안관리자(secadmin)라는 가상의 계정(/etc/passwd에 존재하지 않음)을 설정하였다. 따라서 이 보안관리자는 우선 root 인증을 거쳐 (그림 2)처럼 보안관리자 인증을 거치며, 요구되는 보안등급과 보호범주로 로그인을 변경하지 않은 상태에서는 통합보안관리시스템인 "cmm" 디렉토리에 들어갈 수가 없다. 보안관리자는 "secpv" 명령어를 통하여 현재 로그인 shell의 보안등급과 보호범주를 확인할

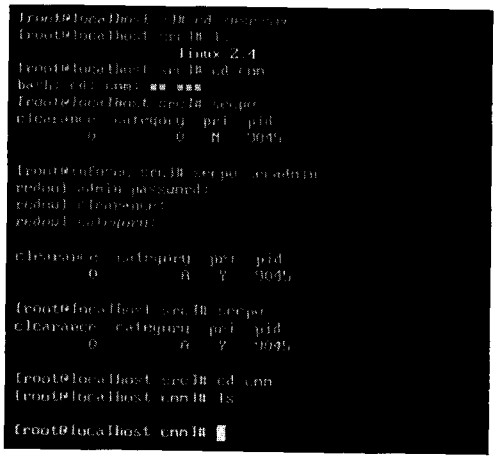


그림 2 보안 관리자 인증 화면

수 있으며 재설정도 가능하다. 또한, "secpv secadmin" 명령어를 이용하여 보안관리자의 보호등급과 보호범주를 설정할 수 있다. 또한 일반 사용자도 보안관리자를 통해서만 최소, 기본(default) 및 최대 보안등급과 보호범주가 설정될 수 있다.

보안 등급이 없는 일반 사용자가 서버에 접속 시에는 보안등급과 보호범주가 (0,0)으로 설정이 되어 보안등급이 부여된 폴더나 파일에 접근할 수 없게 된다. 예로 (그림 3)은 사용자 자신의 보안등급과 보호범주로 접속하기 위해서 "secpv 자신의 계정" 명령으로 허가된 등급범위 내에서 보안등급과 보호범주를 얻을 수 있음을 보인다.

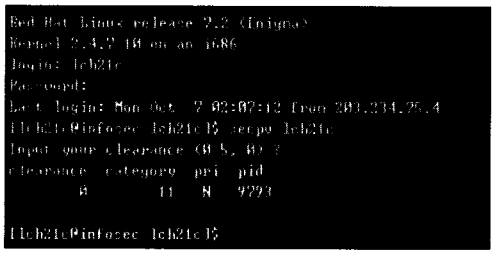


그림 3 사용자 인증 화면

4.3 주체 및 객체의 보안 레이블 상속

보안 운영체제의 주요한 요구사항이라 볼 수 있는 보안 레이블 상속은 주체가 새로운 객체 생성 시 주체에 레이블된 보안등급과 보호범주가 그대로 상속되며, 자식 프로세스가 생성될 때마다 또한 부모 프로세스의 보안속성이 자동으로 상속된다. (그림 4)는 사용자가 파일 생성 시 사용자의 보안 레이블이 파일에 상속됨을 보인다.

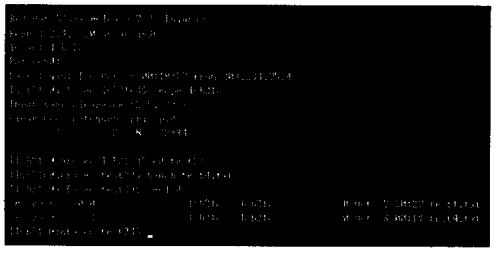


그림 4 보안 레이블 상속 화면

4.4 참조 모니터와 강제적 접근제어

보안 운영체제는 커널 내부에 구현된 참조 모니터(Reference Monitor)에 의해 파일 등 모든 자원에 대한 접근 통제가 이루어지며, 이를 토대로 권한이 없는 사용자에 의한 불법적인 접근 또는 외부로부터 악의적인 목적으로 시도되는 해킹은 운영체제 커널에서 강제적으로 차단된다. 이는 (그림 5)와 같이 주체의 보안등급, 보호범주와 객체의 레이블, 보호범주를 비교하여 강제적으로 접근이 제어된다. 본 구현에서는 주체와 객체간의 접근제어가 수정된 BLP(Bell-LaPadula)모델[20]에 의하여 행하여진다.

즉, ① 주체 S는 객체 O를 오직 Clearance(S) >= Clearance(O)일 경우에만 read할 수 있다(Simple Security : SS-Property) ② 주체 S는 객체 O를 오직 Clearance(S) = Clearance(O)일 경우에만 write(delete 포함)할 수 있다(수정된 Star : *-Property).

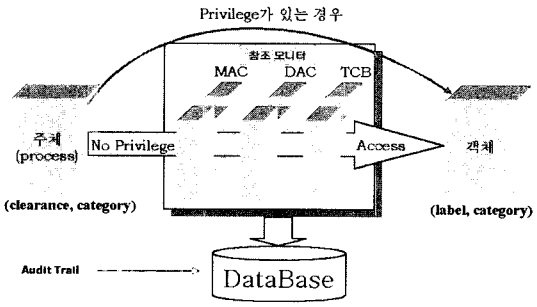


그림 5 참조 모니터 구성도

보안레이블은 강제적 접근 제어를 위한 필수적인 메커니즘이다. 이러한 보안 레이블은 주체가 객체에 접근을 요청할 때마다 보안 정책에 따라서 접근 허가 여부를 결정하는 데 사용된다.

주체에 대한 보안레이블 구현은 프로세스마다 존재하는 Process Control Block인 task_struct의 마지막 부분에 추가하였으며, 객체의 보안 레이블은 리눅스 파일 시스템의 모든 폴더, 파일마다 하나씩 i-node를 가지고 있으므로, Ext2 파일 시스템 내에 예약되어 있는 i-node 구조[21]의 일부분을 사용하였다. (그림 6)은 i-node 구조에 보안레이블을 추가한 모양을 보인다.

4.5 감사 추적

기존의 감사 추적 기능은 정적으로 기록된 사건(event) 로그 파일을 별도의 감사 추적 프로그램에서 읽어 감사자에게 보여주었으나 이 경우 실시간 감사추적이 불가능할 뿐만 아니라 중요한 사건 정보의 상세정보 출력이 어려웠다. 본 논문에서는 데이타베이스를 이용하여 실시간으로 감사추적을 할 수 있도록 하였으며, 로그정보를 데이타베이스에 실시간으로 저장할 수 있도록 구성하였다. (그림 7)은 이러한 DB를 이용한 실

type/permissi	user(uid)	file size
access time		time of creation
time of modification		time of deletion
group(gid)	link counter	no of block
file attribute		reserved(OS-dependent)
12 direct blocks		
one stage indirect block		two stage indirect block
three stage block		file version
file acl		directory acl
fragment address		reserved(OS-dependent)
예약필드(32비트):보안 등급 필드		예약필드(32비트):카테고리 정보 필드

그림 6 수정된 i-node 구조

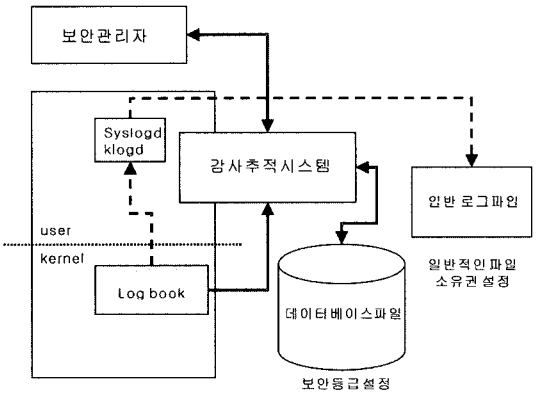


그림 7 다중등급 보안 커널 감사 정보의 흐름

시간 감사 정보의 흐름을 보여준다. 사용자에 부여된 보안속성과 또 사용자가 접근하고자 하는 자원 객체 사이에 강제적 접근제어가 커널 내에서 진행되기 때문에, 시스템 내의 모든 행위는 커널 내의 참조 모니터를 통해서 수집이 가능하며 이를 통해서 세세한(fine-grained) 접근정보에 이르기까지 감사 작업이 이루어지게 된다. 이러한 로그 데이타베이스 파일 또한 보안등급이 설정되어야하며 보안관리자를 제외한 root나 일반 사용자는 접근이 통제되어 읽기, 쓰기가 불가능하다. 이를 통하여 해커의 불법적인 접근 후 자신의 흔적을 제거하는 것을 방지할 수 있다.

(그림 8)의 감사 화면은 왼쪽의 질의문 조건, 통합보안관리 대상 서버 에이전트(agent) 선택 리스트와 오른쪽의 감사자료 화면으로 구성된다. 각각의 서버로부터 수집된 로그 정보가 너무 방대하기 때문에 좀 더 쉬운

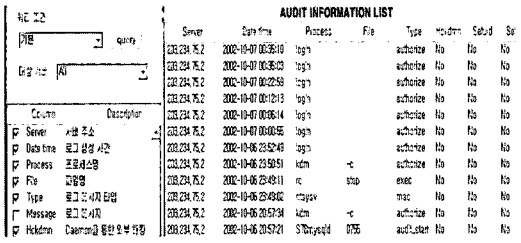


그림 8 감사 정보 목록

검색을 위하여 관리자가 선택한 조건에 따라 검색할 수 있도록 하였다. 감사자료 검색 조건은 기본(default), 프로세스별, 날짜별, 사용자별, 보안등급별, 보호범주별, 접속 IP별 등으로 구분된다. 또한, 각 서버, 프로세스, 사용자 및 IP 주소별로 발생한 감사 발생 건수를 (그림 9)와 같이 그래픽 화면으로 보여줌으로써 보안관리자 측면에서의 편리한 조회와 침입 발생시 알람기능에 의한 통보로 보다 빠른 대응을 가능케 하였다.

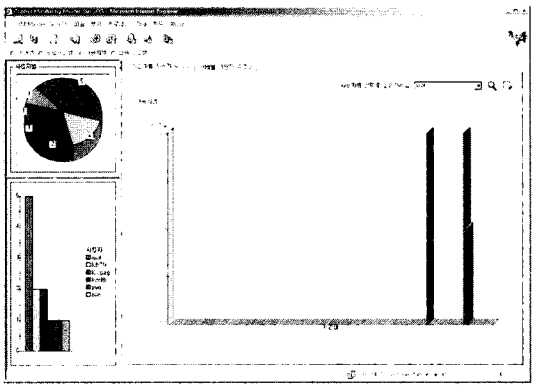


그림 9 통계 조회 창

5. 해킹 방지 기능 구현

보안 커널의 TCB(Trusted Computing Base)에서는 해킹을 통한 주요 파일의 불법적인 변조, 탈취 등을 차단하고, 불법적인 root 권한 획득, 데몬 공격, 바이러스 및 백 도어 등의 불법실행 등 다양한 해킹위협으로부터 시스템을 보호한다. 리눅스 운영체제의 취약성을 이용한 공격 및 내부 범죄로부터의 주요파일 보호가 이 곳에서 이루어진다. 커널 수준의 해킹차단으로 setuid, setgid 프로그램의 취약성을 이용한 root shell 탈취 시도 시 이를 자동으로 탐지하여 원천 봉쇄하게 된다. 또한, 주요 daemon에 대한 공격, 시스템 파일 변조, 불법실행 파일을 통한 해킹 시도 등을 차단하여 바이러스 및 백

도어 공격으로부터 시스템을 보호한다. 본 구현 연구에서는 여러 가지 보안 기능을 제공하여 보안관리자로 하여금 이들 기능 중 자신의 보안 정책에 맞게 선택적으로 적용할 수 있도록 하였다. (그림 10)은 여러 가지 보안기능 설정 옵션을 보여 주고 있다. 각 보안기능에 대해 보안관리자만이 옵션을 "enable" 또는 "disable" 할 수 있도록 하였는데 "enable"은 해당 기능의 적용을 의미하고 "disable"은 적용되지 않음을 의미한다. 보안 기능별로 몇 가지의 주요한 옵션만을 살펴본다.

① sec_mac : 강제적 접근제어를 구현하는 기능으로 "enable"로 설정을 하여야만 모든 보안 레이블이 부여된 객체에 접근시 강제적 접근제어가 적용된다. root 계정일지라도 강제적 접근제어가 적용되기 때문에 권한이 없으면 명령의 실행이 거부된다.

② sec_no_auth_no_exec : "enable"로 설정하면 이후 생성되는 모든 실행파일은 디폴트로 실행이 불가능해진다. 실행을 하기 위해서는 보안관리자로부터 권한부여 받아야만 한다.

③ sec_kill_control : 원격에서 접속한 사용자에 대해서 보안관리자로부터 권한을 부여받지 않으면 kill과 같이 시스템에 중대한 영향을 미치는 명령어들을 실행할 수 없도록 한다.

④ sec_exec_ro_control : 읽기 전용으로 적용된 실행파일에 대한 변조 방지 적용 여부를 결정한다. 읽기 전용 파일에 대해 쓰기가 불가능하므로 해킹을 목적으로 한 파일의 변조를 차단한다. 이 기능은 바이러스의 이식이나 백도어 설치에 의한 침입을 차단하는데 효과적으로 사용될 수 있다.

⑤ sec_trace_setuid_attack : setuid 프로세스에 대해 버퍼 오버플로우 공격을 감지하고 이를 방어하는 옵션이다. 이 옵션을 사용하면 root의 권한 탈취를 방어할 수 있다.

⑥ sec_trace_daemon_attack : root 권한으로 실행되는 데몬들을 대상으로 취약점을 공격하여 불법적인 접근을 탐지하고 이를 방어할 수 있도록 한 옵션이다.

⑦ sec_web_no_exec_cgi : 웹 서비스를 위해 실행되는 데몬들을 대상으로 취약점을 공격하여 root 권한을 취득하는 행위를 방어한다.

⑧ sec_web_exec_control : web상에서의 명령어 실행 여부를 결정한다. 이것이 "disable"되어 있는 경우에는 모든 명령어를 실행시킬 수 있다.

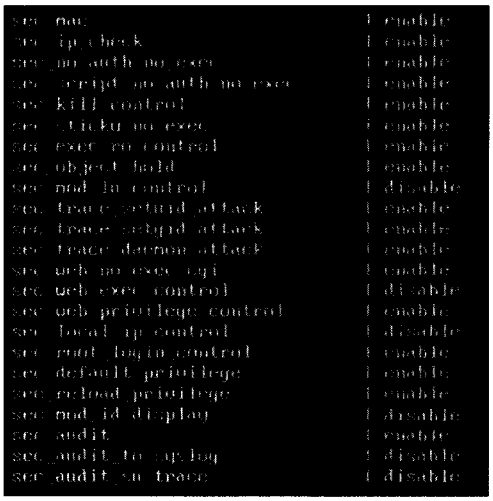


그림 10 보안기능 적용 화면

6. 통합보안관리

보안관리자가 네트워크로 분산되어 존재하는 다수의 보안관리 대상 서버를 한곳에서 원격으로 관리 및 통제할 수 있는 ESM(Enterprise Security Management) 기능이다. 통합관리 시스템은 지리적으로 분산된 다수의 각 서버에 대하여 사용자의 보안 속성 및 보안등급의 설정과 실시간 보안관제 기능을 제공하기 때문에 불법적인 침입에 대한 신속한 대응이 가능하게 된다. 보안관리자의 GUI를 통해 조직의 보안 정책을 간편하게 설정 및 변경할 수 있도록 하였다. (그림 11)은 CMM (Control and Monitoring Master)의 전체 구성도를 나타낸 것이다. 보안통합관리는 각 서버의 정보 수집과 명령을 수행하는 Agent와 각각의 Agent로부터의 정보를 관리하는 통합관리 서버, 그리고 통합관리 서버에 연결하여 수집된 정보를 그래픽 화면으로 보안관리자에게 보여주거나 통합관리 서버를 통하여 각 Agent에게 명령을 실행하는 Manager로 구성된다. Manager는 웹 환경에서 Java로 구현하였으며 플랫폼에 독립적로 Windows 뿐만 아니라 Unix, Linux 등에서도 사용 가능하도록 하였다. 이때 통합관리 서버와 Manager 사이에는 SSL

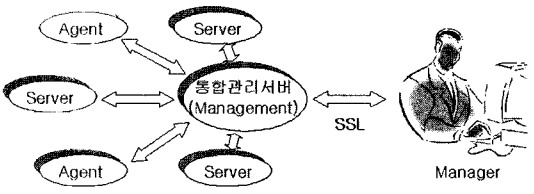


그림 11 CMM 구성도

(Secure Socket Layer) 2048비트 암호화가 지원된다. 통합서버에 접속이 이루어지면 관리자의 로그온 창이 표시되며 보안관리자로 로그 온 하기 위해서는 사용자 식별과 인증과정이 필요하다. 이 과정은 앞서 사용자 식별 인증에서 언급한 바와 같이 서버의 root 사용자 확인과 보안 관리자 확인의 두 단계를 거친다. (그림 12)는 CMM 접속 시 보안관리자 인증 화면이다.

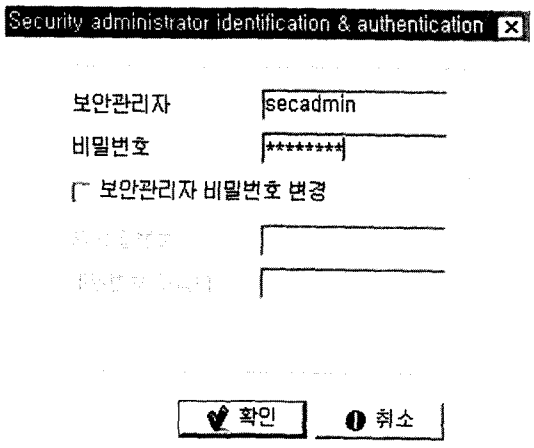


그림 12 CMM 보안관리자 인증 화면

7. 시험 평가

7.1 접근제어 시험

모든 주체와 객체에 보안등급과 보호범주가 주어진 상황에서 각각의 사용자별로 레이블링을 하여 강제적 접근제어 시험을 하였다. (그림 13)과 같이 "test"라는 폴더는 보안등급 0, 보호범주 2로 설정이 되어있다. <0, 2> 권한을 갖고 있는 사용자의 경우 접근에 성공하였으며, (그림 14)는 보호범주가 다르기 때문에 접근에 실패한 경우를 보이고 있다.

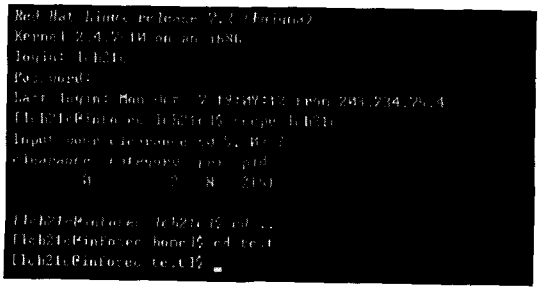


그림 13 접근시도 성공 화면

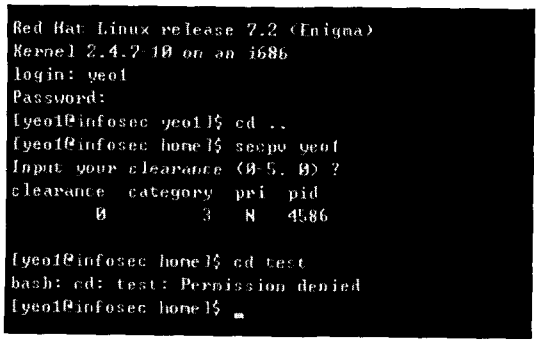


그림 14 접근시도 실패 화면

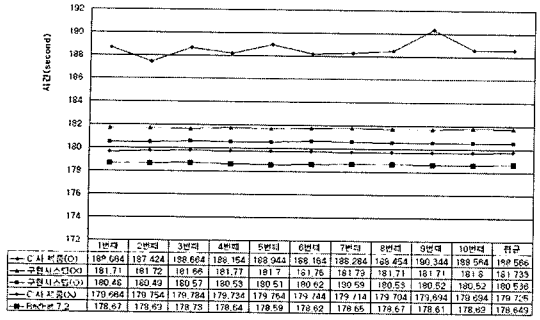
7.2 성능 시험

기존 리눅스 커널에 부가적으로 보안 커널 기능을 구현함으로써 발생되는 부하율을 벤치마킹하기 위해서 본 시험에서는 ① 운영체제 보안 제품을 설치하지 않은 RedHat Linux 7.2 ② 사용자로부터 서버의 자원들을 접근제어 목록(ACL, Access Control List)을 통하여 서버의 자원을 보호하는 LKM 방식의 CA사의 "eTrust Access Control" 제품[18] ③ 본 연구에서 구현한 구현 시스템 등 3가지를 비교 평가하였다. 표 1과 같은 동일한 환경의 시험용 리눅스 서버에서 성능평가를 실시하였으며, 시험 방법은 실제 인터넷 웹사이트(www.yahoo.co.kr)의 메인 화면을 시험용 리눅스 서버에 각기 다른 파일로 다운로드하여 저장하였으며, 이 저장된 파일을 50개 단위로 증가시켜가며 각 10회씩 검색시간을 측정하여 평균값을 구하였다. 검색 처리는 시험용 리눅스 서버의 "localhost"에서 C언어로 작성된 Batch 프로그램을 이용하여 File Open, File Read, File Close를 수행하였다.

(그림 15)는 보호 대상 파일 400개에 대하여 각 100회씩 10회에 걸쳐 처리 시간(단위는 초)의 평균값을 나타낸다. "C사제품(X)"은 보호하고자 하는 객체를 ACL(접근제어 목록)에 추가하지 않은 상태를 의미하며, "구현시스템(X)"은 보호를 원하는 객체에 레이블링을 하지 않은 상태를 나타낸다. 여기에서 구현시스템의 경우 레이블링을 한 경우가 레이블링을 하지 않았을 때보다 검

표 1 성능 시험 환경

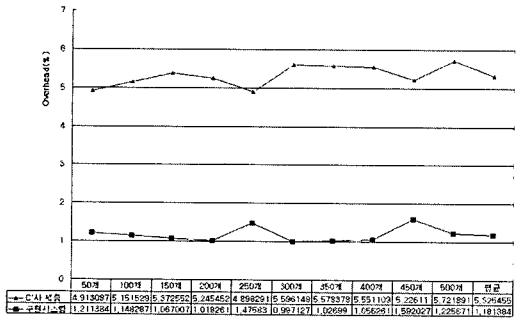
CPU	IBM PC 셀러론 1.0G
RAM	256MB
HDD	20GB
O·S	RedHat Linux 7.2
Kernel	2.4.7-10



	1번째	2번째	3번째	4번째	5번째	6번째	7번째	8번째	9번째	10번째	평균
C사 보안(O)	189.684	187.424	188.664	188.164	188.944	188.164	188.284	189.454	190.344	188.564	188.566
구현시스템(O)	181.71	181.72	181.66	181.77	181.7	181.76	181.79	181.71	181.71	181.8	181.735
구현시스템(X)	180.46	180.49	180.57	180.53	180.51	180.62	180.59	180.53	180.52	180.52	180.538
C사 보안(X)	179.684	179.754	179.784	179.734	179.764	179.744	179.714	179.704	179.694	179.694	179.726
Redhat 7.2	178.67	178.69	178.75	178.64	178.59	178.69	178.65	178.67	178.61	178.69	178.649

그림 15 파일 400개 경우 처리시간(sec)

색 시간이 약간 차이가 있는 것을 볼 수 있다. 그 이유는 레이블링을 하지 않았을 경우에는 쓰기(Write) 권한이 주어져 보안 정보의 상속 과정이 처리되기 때문이다. 본 시험에서는 각 경우에 대하여 공히 각 파일들에 읽기 권한만을 부여하여 측정하였다. 보안 기능에 따른 성능상의 부하율 비교표는 (그림 16)과 같다. 구현 시스템의 경우에는 프로그램사용자 주체와 50개 단위로 모든 보호대상 객체에 레이블링을 하였고, C사 제품의 경우에도 모든 보호 대상 객체를 50개 단위로 ACL 목록에 추가하여 처리 시간을 측정을 하였다. (그림 16)의 비교표에서 보는바와 같이 C사 제품은 평균적으로 5.325%의 부하율이 측정되었으며, 본 구현시스템은 평균적으로 1.182%의 부하율(overhead)이 측정되었다. 이와 같은 성능상의 부하율의 차이는 접근제어 설계에 따른 구조적인 차이에 기인한다. 즉, 본 구현의 경우, 파일을 open할 때마다 일반 Linux OS에서 DAC(permission bit)를 check하듯이, 주체인 프로세스가 객체인 파일에 접근할 때 주체 프로세스의 보안 레이블과 그 객체의 inode에 있는 보안 레이블을 비교(강제적 접근제어)하기 때문에 1~2%의 부하율이 소요된다. 이러한 부하율은 AT&T의 System V/MLS에서 발표된 바 있다[13]. C



	50개	100개	150개	200개	250개	300개	350개	400개	450개	500개	평균
C사 보안	4.913087	5.151529	5.372552	5.245452	4.898291	5.596149	5.578378	5.651109	5.22611	5.721891	5.325465
구현시스템	1.211384	1.149267	1.067007	1.018261	1.47583	0.997127	1.02699	1.056281	1.592027	1.225671	1.181384

그림 16 Secure OS 성능 부하율 비교표

사 제품의 경우 ACL에 접근제어 대상 파일을 목록에 필요시마다 접근 허가권과 함께 추가해 주어야 하며, 프로세스가 파일을 open 할 때마다 ACL 접근제어 목록을 탐색(search)하여 해당 파일에 대한 접근허가를 결정하게 된다.

200개 파일의 경우, 부하율 계산식의 예는 다음 식 (1)과 같다.

$$1.018\% = (89.782-88.877) / 88.877 \times 100 \quad (1)$$

7.3 해킹 시험

본 구현의 중요한 요구사항 중 하나인 해킹 방지 기능을 시험하기 위해서 몇 가지 실제적인 해킹 프로그램을 이용하여 시험을 실시하였다. 시험은 로컬시스템보다는 네트워크 취약성을 이용한 공격방법을 선택하였다. 대표적인 해킹 방법인 setuid를 이용한 불법적인 root 권한 탈취 경우, sendmail을 이용한 시험을 보면, (그림 17)은 목표 서버에 계정을 가지고 있는 사용자가 불법적으로 root 권한을 취득하는 화면이다.

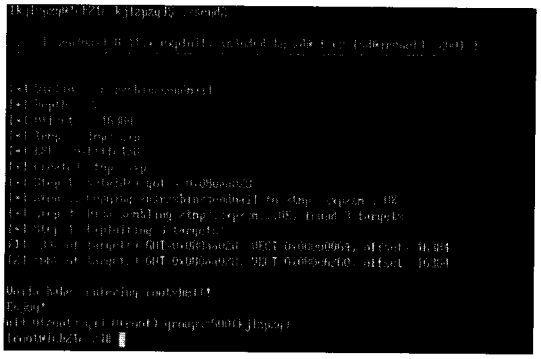


그림 17 sendmail 해킹 화면

(그림 18)은 서버에 본 논문의 보안 기능을 적용한 후 접속한 화면으로 root 계정을 얻지 못하는 것을 알 수 있다.

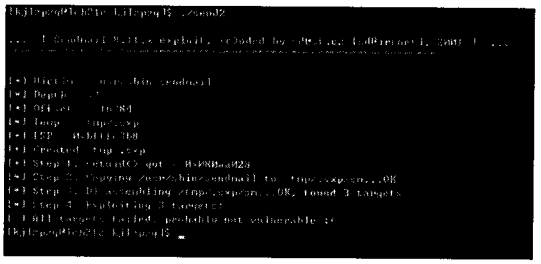


그림 18 sendmail 해킹 방어 화면

(그림 19)는 백도어를 이용한 해킹화면으로 용이하게 root 권한을 얻어낼 수 있다. 백도어는 해커나 일반 사용자가 root 권한을 획득한 후 다음에 재접속 시 쉽게 들어오기 위한 일종의 뒷문이다. 관리자가 모르는 곳에 백도어 실행 파일을 숨겨놓고 다음에 쉽게 root 권한을 취득할 수 있다. 백도어는 setuid 취약성을 이용한 방법으로 이렇게 정상적인 방법이 아닌 비정상적으로 root 권한을 취득하는 경우 본 논문의 구현시스템의 경우에는 (그림 20)과 같이 root 권한을 얻을 수는 있지만 일체의 명령어가 실행이 되지 않는다.

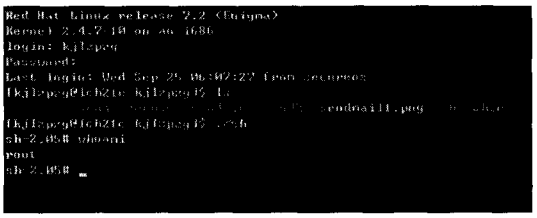


그림 19 백도어를 이용한 해킹 화면

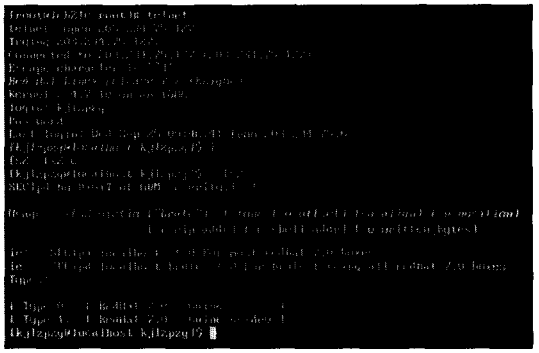


그림 20 백도어 해킹 방어 화면

버퍼오버플로우는 시스템 해킹 기법으로 가장 널리 알려진 기법이다[19]. 버퍼 오버플로우는 크게 스택 오버플로우와 heap 오버플로우 두 가지 방식으로 나누어진다. 다음 해킹 기법은 스택 오버플로우의 일종인 Format String Attack을 이용하였다. FSA은 C 프로그램의 printf( ) 함수의 취약성을 이용한 해킹 기법이다. (그림 21)은 해킹 경유지를 이용하여 목표 서버를 공격하여 root 권한을 취득하는 화면이다. 공격자는 일단 중간 경유지로 쓰일 서버에 계정이 있거나 또는 다른 해킹 방법으로 계정을 하나 만든다.

그 후 그곳에 해킹소스를 컴파일하여 저장시켜 놓는

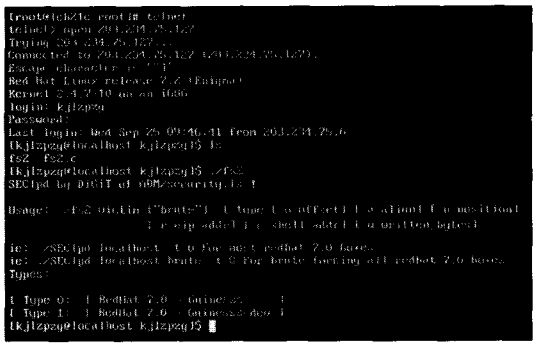


그림 21 해킹 경유지 이용 화면

다. 목표 서버의 IP만 알고 있다면 계정의 유무에 상관없이 open port를 통하여 공격할 수 있다. (그림 22)는 프린트포트(lp port)를 이용하여 침투한 후 root 권한을 취득한 화면이다. (그림 23)은 본 구현에서 포트별 취약성으로부터의 해킹을 방어한 화면이다.

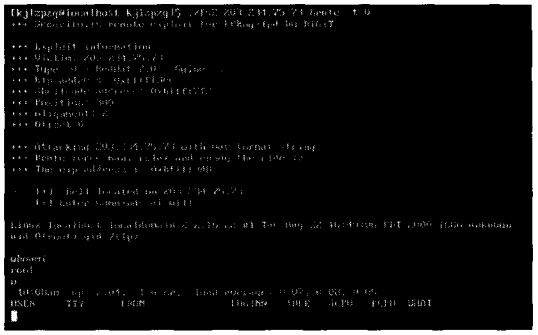


그림 22 버퍼 오버플로우 해킹 화면

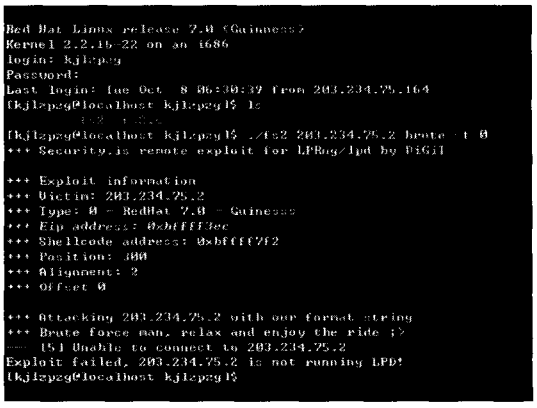


그림 23 열린 port 해킹 방지

8. 결 론

리눅스 운영체제는 오픈 소스와 저렴한 비용 등의 장점으로 인터넷 서버로서 그 활용이 광범위하며 보급 또한 날로 증가하고 있다. 이와 더불어 날로 다양해져가는 해킹기법 및 불법적 사용 시도에는 매우 취약한 면을 보이고 있다. 이러한 상황에서 기존의 방화벽이나 IDS 같은 보안제품이나 응용 프로그램으로 보안성을 확보하기에는 한계가 있다. 이에 본 논문에서는 보안 커널을 TCSEC B1급의 기능적 요구사항과 해킹방지의 2가지 요구사항을 중심으로 리눅스 보안 모델을 설계하고 시스템 콜 후킹 방법으로 이 기능들을 성공적으로 구현하였다. 본 논문에서 구현한 보안 리눅스 운영체제는 Kernel 2.4에서 구현되었으며, TCSEC B1 등급 기준에서 요구하는 대부분의 기능을 만족한다. 강제적 접근제어 시험, 성능 시험 및 해킹 시험을 통하여 그 보안성과 실용성 그리고 안전성을 확인하였다. 본 보안 리눅스 운영체제는 향후 계속적인 연구를 통하여 다양한 환경에서의 성능평가, 취약성 분석, B2/B3 등급으로의 상향화 연구개발이 필요할 것이다. 또한, DoS(Denial of Service) 공격 등 다양한 해킹방지에 대한 계속적인 연구가 필요할 것이다.

참 고 문 헌

[1] Peter A. Loscocco et al., The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments, 21st NISSC, 1998.

[2] Dixie B. Baker, Fortresses Built Upon Sand, ACM Proc. of the New Security Paradigms Work-shop, 1996.

[3] Sue Hildreth, ASP Security: Why Firewall Are Not Enough, http://www.ebizQ.net, 2001.2.

[4] Thomas H. Ptacek et al., Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, NAI Lab., 1998.1.

[5] Http://www. police.go.kr 경찰청 사이버테러 대응센터 보도자료

[6] ISO/IEC 15408 Common Criteria, common-criteria.org 1999. 8.

[7] D. D. Downs et al., "Issues in Discretionary Access Control," Proc. of IEEE Symposium on Security and Privacy, pp. 208-218, 1985.

[8] Charles P. Pfleeger, Security in Computing, PTR, 1997.

[9] DoD, Trusted Computer System Evaluation Criteria, DoD 5200.28. STD, 1985.

[10] ISO/IEC JTC1/SC27, Information Technology - Security Techniques - Security Information Object, N2315, 1999.

[11] http://www.radium.ncsc.mil/tpep/epl/

[12] http://www.cs.utah.edu/flux/fluke

[13] Charles W. Flink II et al., "System V/MLS Labeling and Mandatory Policy Alternatives," Proc. of USENIX-Winter'89, pp. 413-427, 1989.

[14] Security Enhanced Linux, http://www.nsa.gov/selinux

[15] Paul C. Clark, Policy-Enhanced Linux, 23rd NISSC, 2000.

[16] Immunix, http://immunix.org/

[17] Federal Register/Vol.65, No.10/Rules & Regulation(Part III : Dept. of Commerce, Bureau of Export Administration, Revision to Encryption Items;Interim Final Rule, Jan. 14, 2000).

[18] Computer Associates, eTrust Access Control for UNIX, 2001.

[19] IEEE Std 1003.2c-Draft standard fot Information Technology Portable Operating System Interface(POSIX) Part 2: Shell and Utilities : Protection and Control Interfaces.

[20] Bell. D. and Lapadula, "Secure Computer System: Mathematical Foundations and Model," MITRE Report MTR 2547, v2 Nov 1973.

[21] R. Magnus et al, LINUX KERNEL INTERNALS, 1999.

손 형 길

1988년 2월 국제대학교 전산통계학과(학사). 1993년 2월 연세대학교 산업대학원 전자계산학과(석사). 1998년 3월~현재 동국대학교 컴퓨터공학과 박사과정. 1982년 3월 총무처 정부전자계산소 행정자치부 정부전산정보관리소. 2001년 10월~현재 행정자치부 행정망운영과장. 관심분야는 보안운영체제, 분산시스템관리, 통신망관리

박 태 규

1980년 10월 경북대학교 전자계산기공학과(공학사). 1989년 8월 충남대학교 전산학과(이학석사). 1996년 2월 성균관대학교 정보공학과(공학박사). 1981년 2월~1982년 12월 한국국방연구원 연구원 1982년 12월~1992년 2월 한국전자통신연구원 선임연구원. 1997년 1월~1998년 1월 University of Western Sydney(Post-doc). 1992년 3월~현재 한서대학교 컴퓨터정보학과 교수. 관심분야는 보안운영체제, 네트워크 보안

이 금 석

1971년 2월 서울대학교 공과대학 응용수학과(공학사). 1978년 2월 한국과학원 전자계산학과(이학석사). 2001년 3월 건국대학교 대학원 컴퓨터정보통신학과(공학박사). 1973년 9월~1981년 2월 한국과학기술연구소 전산센터 선임기술원. 1981년 3월~현재 동국대학교 정보산업대학 컴퓨터공학과 교수 1999년 9월~현재 동국대학교 정보통신연구소 소장. 관심분야는 운영체제, 컴퓨터 시스템 성능평가, 보안운영체제, 분산시스템 관리, 소프트웨어 품질평가