다중등급보안 리눅스 기반의 RBAC 시스템 구현

김대중*, 김현정*, 김정래*, 박태규*, 조인구**, 임연호**

*한서대학교 전산학과, **티에스온넷(주)

Implementation of RBAC System on MLS-Linux OS

Dae-Joong Kim*, Hyun-Jung Kim*, Jung_Rae Kim*, Tae-Kyou Park*,

In-Gu Jo** and Yeon-Ho Im**

*Department of Computer Science, Hanseo Univ.

**TSonNet Inc.

요 약

역할기반 접근제어(RBAC : Role Based Access Control)는 임의적 접근제어와 강제적 접근제어에 비해 견고함과 유연성을 제공한다. 따라서 RBAC은 최근 금융시스템 및 병원시스템 등에서 많은 관심의 대상이 되고 있다. 본 논문에서는 안전성이 인증된 다중등급보안(MLS : Multi-Level Security) 리눅스를 이용하여 인터넷상에서 가상은행의 금융업무를 안전하게 처리할 수 있는 다중등급기반의 RBAC 시스템을 구현함을 보인다.

I. 서론

지금까지 연구된 접근제어 방법인 임의적 접근제어(DAC: Discretionary Access Control)와 강제적 접근제어(MAC: Mandatory Access Control)[1, 2]는 역할에 따른 자원 접근을 필요로 하는 일반 기업 환경에 적용함에 어려움이 따랐다. 따라서 견고하면서도 유연성이 제공되는 접근제어인 RBAC은 사용자별 역할이 분명한 금융 시스템과 병원 시스템 등에서 관심의 대상이 되고 있다[3,4,5,7]. 이 개념은 조직 수준에서 보안 관리를 증진시키기 위해서 사용자 식별 수준이 아닌 추상화 수준을 제공함으로써 조직의 기능 변화에 따라 역할과 관련한 연산의 삭제 및 추가 역시 유연성 있게 이루어진다. RBAC의 정책을 구현하기 위해서는 역할(role), 역할 계층(role hierarchy), 사용자-역할 관계, 역할-권한 관계, 제약조건 등과 같은 다양한 요소에 대한 세밀한 구성이 필요하다. 이는 접근 권한이 역할에 부여되고, 사용자는 적절한 역할에 소속됨으로써 역할의 수행에 필요한 최소 자원만을 접근 할 수 있도록 하는 것으로 관리자에게는 편리한 관리 능력을 제공한다. 또한 사용자를 정적·동적 의무분리로 접근제어를 규제할 수 있다. RBAC은 기업모델에 맞추어 기업 조직에 대한 다양한 접근제어를 제공하는 큰 장점이 있다. 미국 NIST(National Institution of Standards and Technology)에서 RBAC에 대한 보안 규격 1.0을 발표하였고, '98년 5월에 국제공통평가기준(CC : Common Criteria)이 발표되어 ISO를 통한 국제 평가기준제정 과정에 있으며, 이를 기반으로 한 역할 기반 접근제어 보호 프로파일(PP : Protection Profile)이 발표되었다[12]. 국내에서는 소수의 대학, 연구 그룹 등에서 모델에 대한 동향, 응용 수준에서 RBAC 모델에 대한 개념 및 구현 가능성을 학회에 소개하는 정도이다. 한편 국내에서 최근 인터넷 뱅킹, 사이버 트레이딩 등 인트라넷을 이용한 금융기관 및 일반 기업의 생존을 좌우하는 경쟁력의 핵심요소로 인트라넷 환경이 크게 자리 매김하고 있다. 이러한 환경에서 주요 서버 등에 보관되어 있는 고객정보, 금융관련 정보 등이 내부자의 불법적인 접근, 외부자에 의한 해킹 위협 등 다양한 위협에 노출되어 있다. 이와 같은 환경에서 운영체제의 커널 수준에서 사용자의 역할에 근거하여 내부 중요자료의 접근을 통제하고, 사용자의 역할 분리를 통해 사용자의 권한을 제한하고, 내부자의 이상행위 검출 등의 기능이 필요하다. "역할 기반 접근제어"는 이 같은 조

건을 만족하며 특히 금융분야의 정보처리 시스템에 적합한 것으로 널리 인식되고 있지만 주로 응용 수준에서 이루어지기 때문에 안전성 검증이 미약하다. 본 논문에서는 안전성이 수학적으로 검증된 강제적 접근제어인 MLS 시스템을 통해 RBAC 시스템의 역할을 보안등급(clearance)과 보호범주(category) 방식으로 매핑[8]하는 방식을 사용했다. [8]에서 그림 1과 같은 방법으로 RBAC 시스템을 MLS로 매핑하는 방식을 제시하고 있다. 이에 본 논문에서는 본 연구팀이 개발한 리눅스용 다중등급 보안(SecuOS/LX_MLS) 커널을 이용하여 안전한 금융 시스템을 위한 RBAC 모델을 설계하고 구현하였다.

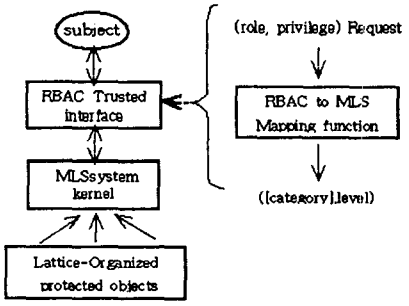

그림 1 MLS-RBAC 매핑방식

II. 다중등급보안 리눅스

본 연구팀에서 개발한 다중등급 보안 커널[13,14,15]을 이용하여 RBAC을 구현 구현하게 되면 다음과 같은 장점을 가지게 된다. 첫째로 RBAC 시스템 개발 및 관리의 비용을 최소화 할 수 있다. RBAC을 MLS로 매핑하는 방식을 통하여 RBAC 시스템을 기존 다중등급 보안 커널 상에서 수행시킴으로 새로운 RBAC 시스템을 구축하기 위해 큰 투자를 요구하지 않으며, 기존 MLS 보안 관리체계를 함께 이용함으로써 관리의 효율성을 기할 수 있다. 둘째로 신뢰된 프로세스를 사용하여 RBAC을 구현함으로 안전하고 높은 보증수준을 유지하는 RBAC 시스템을 제공할 수 있다. MLS 정책은 다중등급 주체(multi-level subject)와 객체간(subject-to-object)의 상호작용을 지원하는 강제적 접근 모델의 한 형태이다. 이는 주체 보안 레이블(subject label)과 객체 보안 레이블(object label)을 부여하므로 DAC와 동시에 MAC을 사용해 강제적 접근제어를 시행한다. 기 구현된 다중등급 보안 커널에서는 모든 주체(프로세스)와 객체(정규파일, 디렉토리, 특수 장치파일) 레이블을 부여하며, 이는 수정된 BLP(Bell & LaPadula) 모델에 따른 강제적 접근제어를 통해 보안레이블을 비교하면서 이루어진다. 또 새로운 프로세스의 생성(fork)에 따른 레이블의 상속화, 파일에 대한 커널 모드의 암호화/복호화, 데이터베이스를 이용한 실시간 감사 추적 시스템 뿐 만 아니라 스마트카드를 이용한 사용자 인증, 출력될 문서에 대한 보안 표시 정보 강제적 출력, root의 권한 제한, 보안 응용프로그램 인터페이스 등을 제공한다[14]. 보안 레이블의 정보는 주체가 객체에 대한 접근을 요청할 때 보안 서버의 정책에 따라서 접근 허가 여부를 결정하는데 사용된다[6].

III. MLS를 이용한 Web 기반의 RBAC 시스템 설계 및 구현

1. JNI(Java Native Interface)를 이용한 사용자 인증

자바언어로 작성된 프로그램들과 기존의 자바가 아닌 프로그래밍 언어, API Took Kit과 프로그램들간의 통합을 쉽게 할 수 있도록 JNI가 제공된다[9]. JNI가 자바 가상 머신 내에 포함됨으로써, 자바 가상머신이 호스트 운영체제 상의 입/출력, 그래픽스, 네트웍 그리고 쓰레드 같은 기능들을 작동하기 위한 로컬 시스템 호출을 수행할 수 있도록 한다. 또 JNI는 Java Native Code(C, C++)와 연결할 수 있는 프로그래밍 인터페이스로 제작된 JDK의 일부 API이다. 웹을 이용한 RBAC에서의 인증 방법은 기존의 Standalone 방식의 인증 방법과는 달리 인증 시 기존의 ID와 password를 이용한 방식뿐만 아닌 역할을 추가로 인증 받아야 한다. 이를 위해 서버에는 JSP(Java Server Page)를 사용하였으며 웹에서 시스템의 패스워드(/etc/passwd)파일과 쉐도우(/etc/shadow)파일을

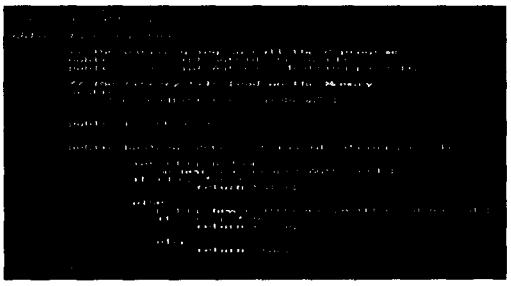

그림 2 : JNI를 사용한 loginbean

사용해 로그인 할 수 있도록 JNI를 통해 로그인 과정을 구현하였다. 그림 2는 JNI의 구현 부분이다. 그림 3은 가상은행 모델을 설계해 Web 기반의 RBAC에서의 사용자 로그인 방법을 나타낸다.

클라이언트가 서버로 접근되어 인증되는 경우는 다음과 같다.

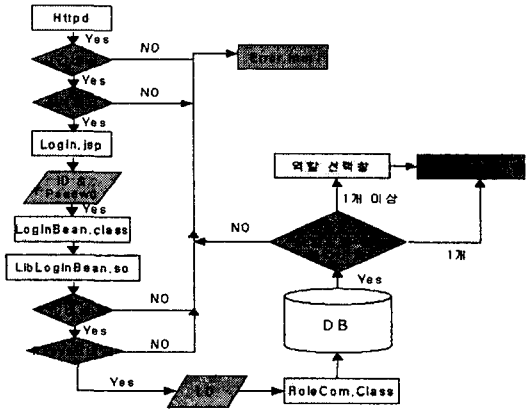

그림 3 : Web 기반의 RBAC에서의 사용자 로그인 방법

첫 단계로 외부로부터 시스템에 접속하려는 사용자는 사용자 ID와 password 입력하게 되며 제약 조건(시간과 공석제약)을 확인하게 된다. 만약 제약 조건에 맞지 않으면 에러 메시지를 표시한다. 제약 조건이 맞게 되면 ID와 password값을 가지고 login.jsp로 이동한다.

둘째 단계로 login.jsp에서는 JNI를 이용해 시스템에 있는 ID와 password를 비교하여 정상적인 사용자인지를 확인하게 된다. 만약 정상적인 사용자인 경우 사용자의 역할을 얻기 위해 ID 값을 갖고 DB의 UA(User Assignment)와 비교해 사용자의 역할을 할당받는다. 이 때 역할이 둘 이상이면 그림 4와 같이 역할 선택 창이 나타난다.

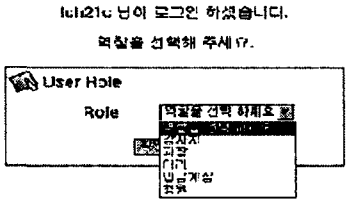

그림 4 : Web기반의 RBAC 역할 선택창

마지막으로 역할에 맞는 권한 할당을 받아야 하는데 이는 RoleCom.class에서 역할을 가지고 DB의 PA(Permission Assignment)테이블에서 권한을 부여받는다. 이로 인해 역할에 맞는 권한으로 접근제어가 이루어진다.

이러한 보안인가 정보들은 사용자가 시스템에 로그인되어 있는 동안에만 유지되며, 로그아웃을 할 경우에는 정보들이 소멸된다.

2. Web 기반의 RBAC 접근제어

그림 5는 일반적인 RBAC에 기초하여 영업부의 정보를 관리하는 간단한 예를 보여 준다. 그림에서 사용자 '김대중'은 회사 내에서 '영업부장'과 '보안관리자'의 역할을 담당하고 있다. 그 역할로 인해 '판매리스트'와 '영업 실적' 그리고 '시스템 log'에 접근 할 수 있는 권한이 부여된다. 그러나 본 논문에서는 MLS-Kernel을 사용하여 RBAC의 접근제어를 구현하였는데, 이 시스템의 경우 MLS에서 사용하는 보안등급과 보호범주의 개념을 사용하게 된다. 이 시스템의 PA 테이블은 기존의 RBAC의 권한을 배정하는 PA 테이블과 다른 접근 방식을 사용한다. 즉, 일반적인 RBAC인 경우, 역할에 따른 정보객체에 대해 접근 할 수 있는 정보 객체에 인가 권한을 주게되지만, MLS-Kernel의 경우 보안등급과 보호범주를 사용하여 강제적 접근제어로 역할과 정보객체에 접근 여부를 판가름하게 된다.

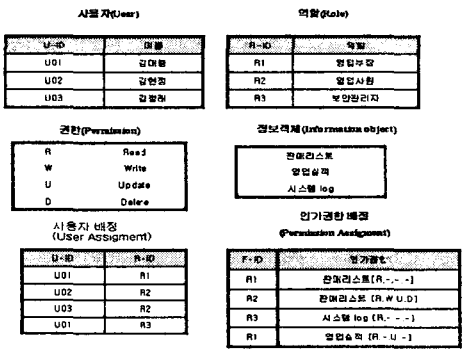

그림 5 : 일반적인 RBAC 접근제어

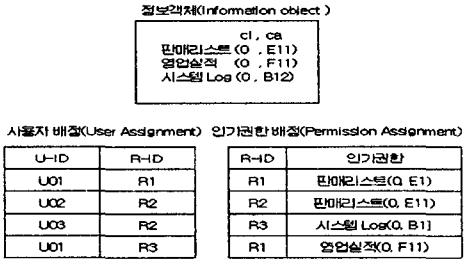

그림 6 : MLS-Kernel RBAC의 접근제어 예

그림 6은 일반적인 RBAC과 같은 일을 했을 경우의 정보 객체와 PA 테이블에서 보안등급 관계이

다. 본 구현에서는 기존의 RBAC의 PA 테이블의 권한 부분에 MLS의 보호범주만을 mapping시킴으로써 다중등급 보안을 이용한 접근제어가 이루어진다. 따라서 MLS-Kernel을 사용한 Web 기반의 RBAC 시스템의 파일의 접근은 수정된 BLP 모델에 따라 제어된다. 조직에서 중요한 파일에 대한 읽기, 쓰기에 대한 접근제어 예는 다음 표 1과 같다. 표 1에서 보호범주 정보가 설정된 각 파일에 대한 zpzg라는 ID의 사용자에 대한 각 파일의 읽기/쓰기 행위의 결과이다. 이때 사용자 zpzg의 보호범주는 E11(과장)인 경우이다.

표 1 : 보호범주만을 사용하는 Web 기반의 RBAC의 읽기/쓰기

파일		사용자		읽기결과	쓰기결과
파일명	범주	ID	범주		
secret1	E1	zpzg	E11	denied	denied
secret2	E11	zpzg	E11	accept	accept
secret3	E112	zpzg	E11	accept	denied

표에서 secret1, secret2, secret3 파일 중에서 secret1 파일은 사용자 zpzg보다 범주의 범위가 넓기 때문에(즉, super-set) 읽기 행위가 denied 되는 것을 알 수 있다. 그러나 secret2와 secret3은 보호 범주가 하위의 관계(즉, sub-set)에 있기 때문에 읽기 행위가 가능하다. 이는 보호 범주가 포함 관계에 있는 경우에만 읽기 접근이 가능함을 알 수 있다. 쓰기의 경우는 오직 보호범주가 같은 경우에만 가능하다. secret2의 파일만이 zpzg와 보호범주가 E11로 같으므로 쓰기 행위가 가능하고 나머지 경우는 모두 거부된다. 이는 조직 내에서 파일의 소유자가 임의로 다른 사람에게 조직내의 기밀 문서를 복사하는 등의 불법적인 행위를 차단한다. 물론 보호범주가 0인 파일은 쓰기가 가능하다. 그러나 이 경우엔 그 파일을 쓴 사용자의 보호범주가 파일에 설정된다. 본 논문에서는 은행 모델[16]을 가상으로 설정하여 시범적으로 구현했으며, 그림 7은 가상은행에서 각 부서별 사원들의 역할에 보호범주를 나타낸 그림이다. "E1, E11, E12, E121…" 은 보호범주를 의미한다. 본 논문의 웹 구현 접근제어에서 DB를 이용하는 제어는 다른 방식을 사용한다. 은행 업무처리에 필요한 데이터는 DB에 넣어 관리하게 된다. DB에 접근하는 마지막 프로세스는 사용하는 DB에 따라 달라진다. 본 논문에 사용된 Mysql은 Mysqld 데몬이 객체인 DB 테이블 데이터 파일에 접근하여 서비스하게된다. 그러나 Mysqld 데몬은 프로세스 동작이 아닌 하나 이상의 프로세스에 각 프로세스마다 멀티 쓰레드로 동작하기 때문에 주체와 객체와의 접근제어를 위한 레이블이 쉽지 않다. 본 논문에서 구현된 접근제어는 웹을 이용한 접근제어 방식으로 그림 8에서 보여준다.

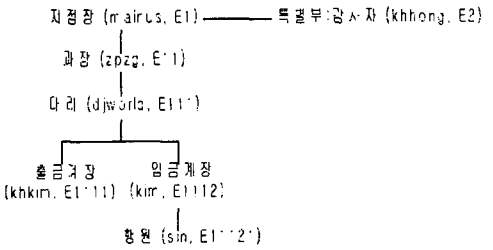

그림 7 : 가상은행 조직의 역할별 보호범주

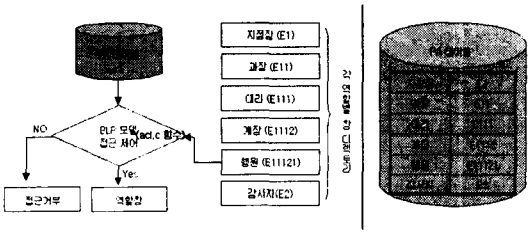

그림 8 : 역할과 객체의 접근제어 방법과 PA 테이블

PA 테이블에서 사용자가 역할에 대한 권한 값을 가져오게 되며 주체가 객체인 각 역할의 디렉토리로 접근하려 할 때 역할의 권한 값을 MLS 시스템 역할의 주체 보안 레이블 값으로 넘겨주게 되며 객체가 되는 역할의 창이 담긴 디렉토리에서는 그 디렉토리의 i-node 값의 보안 레이블 값을 읽어와 커널의 수정된 acl.c 부분에서 BLP모델을 이용한 접근제어를 하게된다. 즉, 권한 E1을 가진 지점장은 권한 E2인 감사자 역할 창으로 들어 갈 수 없게 되는 것이다.

3. Web기반의 RBAC 보안관리자

보안 관리자는 MLS 시스템과 RBAC 시스템에서 가장 중요한 역할 및 역할-권한을 설정하는 사람으로 이전의 시스템 관리자(root)의 중요한 권한을 나누어 사용하게 된다. 본 논문에서의 보안 관리자는 MLS 보안 관리자가 될 수도 있으며, 사용자-역할, 역할-권한과 제약 조건 설정을 위한 작업을 할 수 있다. 역할을 할당하기 위해서 보안 관리자는 사용자의 ID에 대한 역할을 할당해 UA 테이블에 저장한다. 역할에 대한 권한을 설정하기 위해 보안 관리자는 PA테이블을 사용하게 된다. 권한은 접근할 객체에 대한 주체의 MLS-Kernel 접근제어 방법을 사용하여 주체와 객체에 레이블링을 하게 된다. 일반적인 역할의 사용자와는 다른 경로를 가지게 된다. 예를 들면 일반 사용자들은 http://123.234.23.34를 사용해 로그인을 하지만

보안 관리자의 경우는 http://123.234.23.34/admin 같은 경로로 로그인을 한다.

4. RBAC 적용 가상 은행모델 구현

1) 지점장 역할

본 가상은행에서 지점장 역할은 고객 계좌 조회(개인, 전체)와 고객정보 조회, 그리고 직원 정보를 관리 할 수 있다. 조회는 고객의 계좌번호와 성명 순으로 가능하다. 직원정보의 경우 추가, 수정, 삭제, 조회 등을 할 수 있게 된다.

2) 행원 역할

행원 역할은 기본적으로 고객에 대한 은행의 서비스 부분을 전담한다. 고객의 입/출금, 이체, 조회 고객의 신규 및 삭제의 업무를 담당한다. 입금은 무통장, 수표, 일반으로 나뉘고, 현금과 수표를 동시에 처리가 가능하다. 입금을 하게 되면 DB에 저장되기 전에 한번 더 확인 창이 뜬다. 이 때 취소를 누르게 되면 모든 입력상태는 초기화가 된다. 출금의 경우 고객의 계좌번호 외에 비밀번호를 입력하여 비밀번호가 틀린 경우엔 경고 메시지가 나타나고 맞을 경우 기본 정보와 통장의 잔고가 나타난다. 이 때의 동작은 입금의 방법과 유사하다. 조회의 경우 고객의 기본적인 정보를 보여주고, 기간을 선택하여 조회 할 수 도 있다. 신규의 경우 고객의 신상명세를 입력하면 계좌 번호는 자동 할당이 된다. 고객 계좌의 삭제는 계좌의 잔고가 없을 경우에만 가능하다.

3) 감사자 역할 창

감사자는 은행에서 일어난 모든 금액 및 계좌, 직원 정보의 변경 사항 등을 감독하는 사람으로 모든 현금의 이동 유무를 파악하기 위한 권한이 필요하게 된다. 그러기 위해 행원, 계장, 대리, 은행장 등의 역할 테이블에 대한 감사 활동을 할 수 있다. 예를 들어 행원에 대한 감사로 고객 계좌 테이블에 대한 읽기 감사를 할 수 있으며 지점장의 경우 직원 변경 유무들을 조사해 불법적인 흐름을 감지 할 수 있다.

4) 보안 관리자 역할

RBAC 시스템에서는 시스템 관리자와 보안 관리자로 나뉘어져 역할 분담이 가능하게 된다. 시스템 관리자는 시스템 상에 사용자를 추가 할 수 있지만 그 사용자에 따른 역할을 할당하지는 못한다. 이는 보안 관리자의 몫이다. 이렇듯 두 관리자를 두어 보안의 투명성을 제공한다. 보안 관리자는 MLS 시스템이나 RBAC 시스템에서 가장 중요한 주체 및 역할에 대한 권한을 설정하는 사람으로서 보안 관리자는 역할-권한과 제약 조건 설정을 할 수 있다. 역할-권한의 동작으로 역할 추가, 역할 삭제, 역할 할당 및 권한 설정 등의 권한을 가지고 있다. 역할 추가 행위는 그림 9에서 보여준다. 이 경우 새로 추가된 역할은 PA 테이블에 추가되며 역할 삭제의 경우도 사용된다. 이 때도 PA 테이블에서 삭제된다. 역할 할당은 시스템에 허가된 사용자에게 역할을 할당하는 것으로 그림 10은 'lee1'이란 사용자에게 새로운 역할인 감사자를 할당하는 것을 보여주고 있다. 이 때의 역할 할당은 UA 테이블에 저장된다.

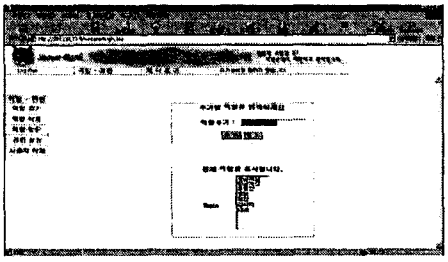

그림 9 : 보안관리자 역할 추가 창

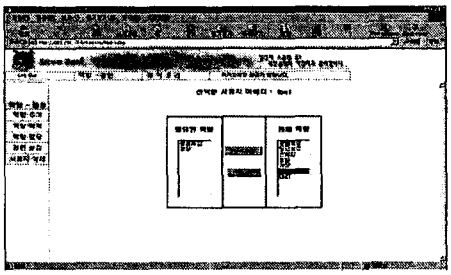

그림 10 : 사용자 역할 할당/삭제 창

그림 11은 역할에 권한을 설정해주는 창으로 권한은 주체와 객체의 BLP 모델에 의해 매핑 된다. 사용자 삭제의 경우 사용자를 UA 테이블에서 제거하는 것으로 사용자가 더 이상 조직 내에 존재할 필요가 없을 경우에 사용한다. 사용자 삭제시 주의 할 점은 사용자의 모든 보안 정보가 의도하지 않게 사라질 위험이 있으므로 삭제 될 경우 경고 표시를 확인 후 삭제한다. 또한 본 논문에서 구현한 가상 은행 모델에서는 제약 조건을 시간 제약과 공석제한을 두었다. 시간제한은 평일에 09:00~17:00까지 토요일은 09:00~13:00까지로 제한하였다. 그리고 업무 연장을 해야 할 경우 시간의 범위를 입력하게 되면 그 시간까지 업무를 연장 할 수 있게된다. 공석의 경우 휴가, 출장, 연수로 나누어져, 보안관리자가 공석인 사용자의 ID를 입력하므로 공석의 제한 조건을 실행 할 수 있다.

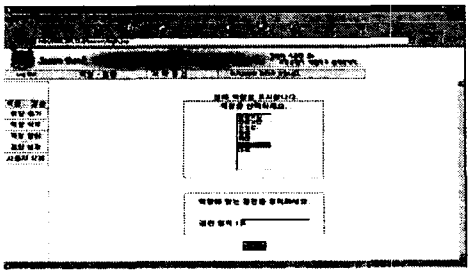

그림 11 : 역할에 대한 권한 설정 창

IV. 결 론

본 논문에서는 다중등급 보안을 위해 기존의 임의적 접근제어에 강제적 접근제어를 리눅스 운영체제의 커널에 추가한 MLS-Kernel을 기반으로 한 RBAC 가상 은행모델을 설계하고 구현하였다. 즉, 강제적 접근제어를 위해 BLP 모델을 적용해 참조 모니터 기능을 하도록 한 다중등급 보안 리눅스 커널에서 보호범주만을 사용하여 웹에서 가상은행 모델을 설정하여 시범 업무를 구현하였다. 다중등급 보안 커널의 강제적 접근제어는 웹 상에서 사용자의 역할에 따른 차별적인 보안등급을 통해 자원을 안전하게 보호할 수 있다. 즉 종래의 보안등급이 없는 사용자 또는 보안 등급이 맞지 않는 사용자가 임의적으로 보안등급을 갖는 파일, 디렉토리, 장치에 대한 읽기, 쓰기, 실행 등의 접근을 BLP 모델에 기초하여 커널 모드에서 부정행위를 원천적으로 차단할 수 있게된다. 이와 같이 구현된 RBAC 시스템은 안전한 커널을 기반으로 했기 때문에 커널 수준에서 시스템 보안을 가능하게 하며, 이를 이용한 응용 시스템 혹은 본 논문에서와 같이 웹 상에서의 보안을 강화할 수 있는 장점이 있다. 향후 연구가 더 필요한 부분으로는, 다중등급보안 네트워크 프로토콜 기능을 커널에서 지원토록 함으로써 프로세스간의 참조모니터 역할을 할 수 있는 보안 게이트웨이가 필요하다.

참고문헌

[1] David D. Clark and David R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," Proceedings of the 1987 IEEE Symposium on Security and Privacy, 1987, pages 184~194.

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

[3] David F. Ferraiolo, Janet A.Cugini, D. Richard Kuhn "Role-Based Access Control (RBAC): Features and Motivations," Proceedings of the 11th Annual Computer Security Applications Conference, 1995, pages 241~248.

[4] Ravi Sandhu, Edward J.Coyne, Hal Feinstein & Charles, Youman "Role-Based Access Control Models," IEEE Computer, Volume 29, Number 2 / Feb, 1996, pages 38~47.

[5] Ravi Sandhu, "Transaction Control Expressions For Separation Of Duties," Proceedings of 4th Aerospace Computer Security Applications Conference, Dec, 1988.

[6] Charles W. Flink II and Jonathan D. Weiss, "System V/MLS Labeling and Mandatory Policy Alternatives," Proceedings of USENIX-Winter'89, pp.413-427, 1989.

[7] Ravi Sandhu and Venkata Bhamidipati, "The ARBAC97 Model for Role-Based Administration of Roles: Preliminary of Second ACM Workshop on Role-Based Access Control," Nov, 1997.

[8] D. Richard Kuhn, "Role Based Access Control on MLS Systems without Kernel Changes," Proceedings of the Third ACM workshop on Role Base Access Control ACM, 1998.

[9] Sheng Liang, The Java Native Interface - Programmer's Guide and Specification, Addison Wesley, Dec, 1999.

[10] Charles L. Smith, Edward J. Coyne, Charles E. Youman and Srinivas Ganta, "A Marketing Survey of Civil Federal Government Organizations to Determine the Need for a Role-Based Access Control(RBAC) Security Product," NIST & SETA, small Business Innovation Research (SBIR), Jul, 1996.

[11] 박태규, 임연호, 커널 기반의 보안 리눅스 운영체제 구현, 정보보호학회 논문지 제 11권 제 4호, 2001.8.

[12] 김현정, 박태규 외, 다중등급 보안커널 구현과 보안 API, 한국정보보호학회 종합학술발표회 논문집, 제 10권, 1호, 2000. 11.

[13] 티에스온넷(주), SecuOS/LX_MLS 보안 관리자 사용자 설명서 v1.0, 2001. 5.

[14] H 은행 관리 규정집, H은행, 2000. 10.