通信情報保護學會誌
第9券 第2號 1999. 06

Web 환경을 중심으로 한 RBAC의 연구 동향

오 세종*, 박 석**

요 약

역할 기반 접근제어(RBAC)는 기업을 위한 응용 분야에서 특별히 주목을 받고 있는 보안 기술이다. 그 이유는 RBAC이 대규모의 네트워크로 연결된 응용 분야에서 보안 관리의 복잡성과 비용을 절감시켜 주기 때문이다. 본 기술 논문은 지금까지 연구된 RBAC의 연구 결과를 소개하고 향후의 과제에 대해 제시하는데 그 목적이 있다. 먼저 RBAC의 출현 배경과 기본 모델의 여러 요소들에 대해 소개한다. 그리고 RBAC에 관련된 연구 동향을 적용 시스템을 중심으로 분석한다. 특별히 Web 환경에서 RBAC의 적용에 초점을 맞춘다. Web 환경, 특별히 인트라넷에서의 보안 필요성을 제시하고 여기에 적용된 RBAC(인트라넷-RBAC)의 내용과 더불어 인트라넷-RBAC의 구현에 대해서도 소개한다. 마지막으로 향후에 추가적으로 연구되어야할 과제들에 대해 분석한다.

Ⅰ. 서 론

1970년대에 들어서면서 컴퓨터 시스템이 다수의 사용자에게 다수의 응용(application)을 제공하는 특성을 갖게되면서 데이타 보안 문제에 대한 관심이 높아지게 되었다. 시스템 관리자와 소프트웨어 개발자들은 권한이 있는 사용자들에게만 특정 데이타 또는 자원들이 제공되는 것을 보장하기 위한 서로 다른 종류의 접근제어(access control)를 구현하기 위해 노력하였다. 접근(access)이란 컴퓨터 내의 자원에 대해 어떤 작업(예 : 사용, 변경, 조회)을 할 수 있는 능력을 말한다. 접근제어(access control)는 그러한 능력을 가능하게 하거나 제한할 수 있는 수단이다. 컴퓨터에 기초한 접근제어는 어떤 사용자 혹은 어떤 프로세스가 시스템의 특정한 자원을 사용할 수 있는지에 대해 기술하며 어떤 유형의 접근형태(access type)가 허용되는지에 대해서도 기술한다. 지금까지 연구된 접근제어 기법으로 임의적 접근제어(discretionary access control)와 강제적 접근제어(mandatory access control)가 있다[1].

임의적 접근제어는 시스템내의 각 사용자(또는 사용자 그룹)와 각 객체에 대하여 사용자가 객체에 허용된 접근 모드를 기술하는 사용자의 식별(identification)과 인가에 기초하여 정보에 대한 사용자의 접근을 제어한다. 만일 사용자가 특정 모드로 객체에 접근할 수 있다는 것을 기술하는 권한(authorization)을 소유한다면 접근은 허락되고, 그렇지 않다면 거절된다. 임의적 접근제어의 유연성은 다양한 시스템과 응용에 적당하다. 이러한 이유로, 특히 상업적이거나 기업적인 환경에서 다양한 구현을 통하여 폭넓게 사용되어지고 있다.

강제적 접근제어는 각 정보에 결합된 비밀등급(classification level)과 사용자에게 부여된 인가등급(clearance level)을 사전에 규정된 규칙과 비교하여 그 규칙을 만족하는 사용자에게만 접근 권한을 부여하는 보안정책으로서, 군사적 환경과 같이 정보의 기밀성이 매우 중요시되는 환경에서 사용되고 있다. 강제적 접근제어는 비밀등급이 높은 객체 정보가 낮은 등급의 객체로 흐르는 것을 방지한다. 이러한 시스템에서 정보는 같은 등급의 보안 클래스 내 또는 위쪽으로 전달되게 된다.

위에서 논의한 임의적 접근제어와 강제적 접근제어 정책은 미국 국방성의 오렌지 북(Orange Book)에서 언급된 표준으로 인식되어 왔다. 그러나 이러한 고전적인 임의적 정책과 강제적 정책으로 다루지

* 서강대학교 컴퓨터학과(sejong@dblab4.sogang.ac.kr), ** 서강대학교 컴퓨터학과(spark@dblab.sogang.ac.kr)
※본 연구는 정보통신연구관리단 대학기초연구지원사업(C1-98-0744-00)의 지원으로 수행되었습니다.

못하는 많은 실제적인 요구들이 존재한다는 것을 보안관련 종사자들이 느끼게 되었다. 이에 따라 고전적인 임의적, 강제적 접근제어 정책에 대한 몇 가지 대안들이 제안되었다. 이러한 정책은 권한의 사용이나 할당에 있어서 제한을 기술하는(강제적 접근방법에서처럼) 가능성과 함께, 임의적 접근제어에서 처럼 사용자에게 객체에 대한 권한을 부여하는 것을 허용한다. 제안된 대안중 하나가 역할 기반 접근제어(Role Based Access Control : RBAC)이다.

RBAC 정책은 정보에 대한 사용자의 접근을 시스템내에서 사용자가 수행하는 행위에 기초하여 규정한다. 역할 기반 정책은 시스템내에서 역할의 식별을 필요로 한다. 역할은 특정 작업 행위와 관련된 책임과 행동의 집합으로 정의된다. 그리고, 사용자에게 허가된 모든 접근을 기술하는 대신에, 객체에 대한 접근 권한이 역할에 대하여 부여된다. 사용자는 역할을 부여받을 권한을 갖는다. NIST의 최근 연구[15]에 의하면 RBAC에 대해 역할이 많은 기업 조직과 정부 조직에서 유용한 접근 방법으로 확신하고 있다.

Ⅱ. RBAC의 개념

RBAC의 중심적인 개념은 사용자가 기업이나 조직의 정보 자원을 임의로 접근할 수 없도록 하는 것이다. 대신에 접근 권한이 역할(role)에 부여되고 사용자는 적절한 역할에 소속됨으로서 역할의 수행에 필요한 최소 자원만을 접근할 수 있도록 한다. 이러한 아이디어는 권한 관리를 매우 단순화 시켜주고 기업의 특정한 보안정책을 구현하는데 있어서 유연성을 제공하는 장점이 있다. 사용자는 그들의 업무적 권한과 책임에 따라 특정 역할의 구성원이 되며 접근 구조의 변경이 없이도 역할의 변경을 쉽게 할 수 있다[3].

RBAC 모델의 개념은 다음과 같이 잘 알려진 세 가지 보안 원리를 지원한다[2].

- 최소 권한 원칙(least privilege principle) : 역할 계층성을 이용하여 작업에 꼭 필요한 최소한의 허가 사항만을 역할에 배정하는 정책이다.
- 임무 분리(separation of duty) : 정보의 무결성을 침해하는 사기 행위나 부정 수단을 유발할 수 있는 작업은 상호 감시적인 역할로 지정하여 임무를 분리시켜 수행한다.
- 데이타 추상화(data abstraction) : 전형적인 운영체제나 응용시스템에서 사용되어졌던 데이타를 처리하는 read, write, execute 등의 연산 대신에 다양한 기능을 수행할 수 있고 명령어를 추상화시키는 상업적인 처리 명령어 credit(입금), debit(출금), transfer(이체), create account(계좌개설), delete account(계좌해지)등을 지원한다.

2.1 RBAC의 기본 모델

그림 1은 RBAC의 기본 모델을 보여준다. RBAC의 기본 모델은 사용자(U: user), 역할(R: role), 인가권한(P: permission), 세션(S: session)으로 구성되어 있다[2].

사용자(user)와 역할(role) : 모델의 간략화를 위해서 사용자는 사람을, 역할(role)은 역할에 부여된 책임과 권한을 기술하는 조직내의 업무 기능(job function)의 이름으로 간주한다. 사용자는 컴퓨터 시스템을 통하여 시스템 내의 정보를 사용하는 객체로서 한 사용자는 한명의 사람에 대응된다. 역할은 접근제어 정책을 구현하는 중요한 의미적 구조이다. RBAC 시스템에서는 시스템 관리자가 회사나 조직의 업무 기능에 따라 역할을 생성하고 역할에 권한을 부여한다. 역할 계층(RH: role hierarchy)은 관련성이 있는 역할들간의 부분순서(partial order) 관계로서 정의되며 기업의 권한과 책임의 체계와 매우 유사하여 기업의 권한체계를 모델링하는데 매우 적합하다.

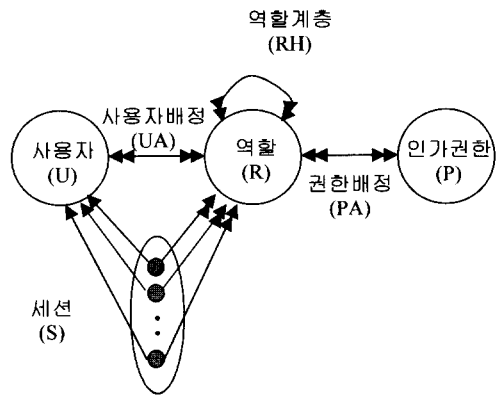


그림 1. RBAC의 기본 모델

인가권한(permission) : 인가권한은 시스템의 하나 또는 그 이상의 객체에 대한 특정 접근 모드(예 : read, write, update)의 승인을 나타낸다. RBAC에서의 인가권한(permission)은 권한허가(authorization), 접근권리(access right), 권한(privilege)과 같은 의미를 갖는다. 여기서 객체는 기업 또는 조직내의 정보시스템을 구성하고 있는 자료(data)나 시스템 자원(system resource)을 말한다. 인가권한은 네트워크 수준으로부터 특정 레코드의 특정 필드에 대한 접근 단위에 이르기까지 다양한 레벨, 다양한 범위로 주어질 수 있다.

세션(session) : 사용자는 시스템에 로그인(log in)등을 통해 그들이 가진 역할의 부분집합을 활성화할 때 세션을 형성한다. 각 세션은 하나의 사용자와 여러 개의 권한을 매핑(mapping)한다. 그림 1에서 이중 화살표는 다중 역할이 동시에 활성화한다는 것을 말한다. 사용자에게 사용 가능한 권한은 그러한 세션에 활성화된 모든 역할이 가진 권한의 합집합이다. 각 세션은 그림 1에서 보듯이 단일 화살표에 의해 지시되는 단일 사용자와 관련된다. 이러한 관계는 세션이 존속하는 동안 지속된다. 세션의 개념은 접근제어 영역에서 주체(subject)의 전통적 개념과 일치한다. 사용자는 워크스테이션의 화면에 나타난 여러 개의 윈도우와 같이 동시에 열려진 다중 세션을 가질 수 있다. 각 세션은 서로 다른 활성화된 역할을 가진다.

사용자 배정(user assignment)과 인가권한 배정(permission assignment) : 사용자 배정과 인가권한 배정은 다대다 관계이며 RBAC 모델에서 매우 중요한 구성요소이다. RBAC의 특징중의 하나는 사용자가 정보 객체들에 대해서 실행할 수 있는 연산들을 직접 사용자에게 부여하는 대신 조직의 업무 수행에 필요한 역할에 배정하고(인가권한 배정), 사용자는 해당 역할의 구성원이 됨으로써(사용자 배정) 정보 객체에 대해 지원하는 연산을 수행하도록 하는 것이다. 이러한 방법은 사용자와 정보 객체수가 많은 일반 기업 환경에서 권한의 관리를 매우 용이하게 수행할 수 있는 장점을 제공한다.

그림 2는 RBAC에 기초하여 영업부의 정보를 관리하는 간단한 예를 보여준다. (RBAC을 구현하는 방법은 연구자나 시스템마다 다를 수 있다.) 그림 2에서 사용자 '변창우'는 회사내에서 '영업부장'과 '보안 관리자'의 역할을 담당하고 있으며 그 역할로 인해 '판매리스트'와 '영업 실적' 그리고 '시스템 Log'에 접근할 수 있는 권한이 부여된다. 그림에는 나타나 있지 않지만 역할에서 영업부장-영업과장-영업사원은 역할 계층(role hierarchy)을 이룬다.

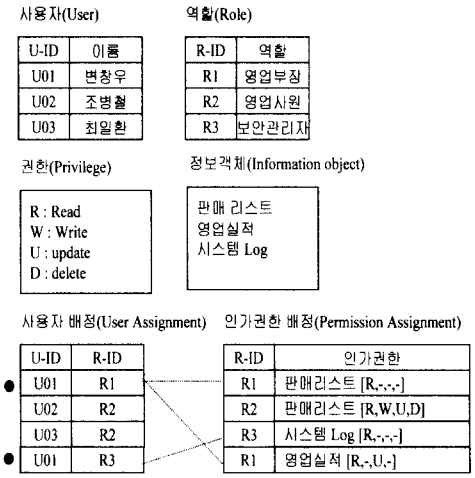


그림 2. RBAC의 간단한 적용 예제

2.2 RBAC에 관련된 연구동향

RBAC에 대한 연구는 1) RBAC을 정의 하고 RBAC의 각 구성요소들을 분석하는 분야, 2) RBAC을 WEB 환경에 적용하는 분야, 3) RBAC을 DBMS에 적용하는 분야, 4) RBAC에 대한 표준 보안규격을 제정하는 분야, 5) 기존 보안 환경에 RBAC을 적용하는 응용시스템 개발분야 등으로 나누어 볼 수 있다.

① RBAC에 대한 정의 및 각 구성요소 분석

R. Sandhu와 H. Feinstein은 데이타베이스 모델에 적용된 ANSI/SPARC 3계층 구조(3 tier architecture)를 RBAC에 도입하여 RBAC의 3계층 구조를 정의하였다[14].

R. Sandhu와 SETA corporation의 연구자들은 RBAC 모델을 네가지로 정리하였다[13].

초기의 RBAC 모델은 사용자(user)와 역할(role)이 많은 대형 시스템에서 중앙 집중화된 관리를 하는 것이 어렵고 많은 비용을 수반하는 단점을 지니고 있었다. 이를 해결하고자한 것이 ARBAC 97(Administrative RBAC '97) 모델이다. ARBAC97은 RBAC 그 자신을 RBAC의 분산 관리에 이용하는 개념이다. ARBAC97은 URA97,

PRA97, RRA97 이라는 이름으로 분야가 분리되어 연구되고 있다[6][7][8].

SBIR(Small Business Innovation Research) 프로그램은 산업계, 정부, 대학기관 등과 협력하여 RBAC과 RBAC의 응용을 정의하고 개발해오고 있다. 또한 NSA(National Security Association)와 Maryland 대학이 공동으로 안전하고, 효과적이며, 접근제어에 알맞는 메커니즘을 제공하는 RBAC의 형식적 참조 모델(formal reference model)을 개발하는 중이다[12].

② WEB 환경에 대한 RBAC의 적용

Barkley와 몇몇 연구자들이 WEB 환경에 RBAC의 개념을 도입하여 Web 문서에 대한 보안관리를 하기 위한 모델을 개발하고 이를 Unix와 NT환경에서 구현하는 연구를 하고 있다[11].

Z. Tari 와 S. Chan는 WEB 환경중에서도 기업의 정보인프라 구축에 많이 이용되는 인트라넷 환경에서 RBAC을 적용하기 위한 모델을 발표하였다[9].

이 분야에 대해서는 다음절에서 자세히 소개한다.

③ DBMS에 대한 RBAC의 적용

RBAC은 주요 상업용 DBMS 제품에도 적용되었다. R. Sandhu는 대표적인 DBMS 제품에 구현된 RBAC의 사양을 분석하였다[4]. RBAC을 적용한 대표적인 제품으로 다음과 같은 것이 있다.

- Informix Online Dynamic Server Version 7.2
- Sybase Adaptive Server release 11.5
- Oracle Enterprise Server Version 8.0

제품별로 RBAC의 사양이 적용된 내용이 조금씩 다른데, 요약된 결과는 표 1과 같다.

R. Sandhu와 G. Ahn은 Window NT의 그룹(group) 개념에 RBAC의 개념을 추가하여 대규모 시스템에 적합한 접근제어 방법을 제안하였다[5].

R. Sandhu는 ARBAC에서 PRA(permission role assignment) 분야를 Oracle에 적용하는 방법을 제시하였다[16].

이와 같이 RBAC이 상용 DBMS나 OS상에서 구현된 사례는 RBAC이 현실문제의 해결에 유용한 방법론임을 뒷받침해 준다.

④ RBAC에 대한 표준 보안규격을 제정

RBAC이 하나의 보안 솔루션으로 적용될 수 있기 위해서는 RBAC의 표준 보안규격이 필요하다. 미국의 NIST(National Institute of Standards and Technology)는 RBAC에 대한 보안규격 1.0을 1998년도에 발표하였다[17]. RBAC의 표준에 관한 연구는 앞으로도 계속될 전망이다.

표 2. 상용 DBMS에 구현된 RBAC의 사양

Feature	Informix	Sybase	Oracle
역할 수여자(role grantee)가 역할을 다른 사용자에게 부여(grant)할 수 있는 능력	Yes	No	Yes
하나의 사용자 세션에서 여러 역할의 동시 활성화	No	Yes	Yes
하나의 사용자 세션에 대해 초기 활성화 역할 집합(default active role set)의 정의	No	Yes	Yes
역할 계층의 구현	Yes	Yes	Yes
역할에 대한 정적 의무분리 구현	No	Yes	No
역할에 대한 동적 의무분리 구현	(Yes)	Yes	No
하나의 역할에 지정될 수 있는 최대, 최소 사용자수(cardinality)의 정의	No	No	No
DBMS 시스템 권한 (system privilege)을 역할에 부여	No	Yes	Yes
DBMS 객체 권한 (object privilege)을 역할에 부여	Yes	Yes	Yes

⑤ RBAC을 적용한 응용시스템 개발

RBAC에 대한 이론적인 연구와 더불어 이를 현실의 문제에 적용하려는 연구가 진행되고 있다. 몇몇 연구기관들이 이미 공감대가 형성된 요구 분야에 대해 RBAC을 포함시키는 실험을 하고 있다. RBAC은 SESAME(Secure European System for Applications in a Multi-vendor Environment)의 분산 시스템과 데이타베이스 언어 SQL3에서 보안 모델의 일부분으로 통합되었다. 또한 OMG의 CORBA 사양이 OMG에 의해 정의된 분산 객체 기술을 사용할 수 있는 접근제어 메커니즘의 예로서 RBAC을 사용하였다[12].

NIST는 RBAC을 의료 시스템(health care system)에 구현하는 프로젝트를 진행하였다[15].

Ⅲ. Web환경에서 RBAC의 적용

World Wide Web의 개방성과 확장성, 사용의 용이성은 기업 활동의 중요한 전략으로서 Web 기술이 주목받게 하는 요인이 되고 있다. 기업들이 경쟁적으로 개설하고 있는 홈 페이지는 고객과 기업을 이어주는 접촉점일 뿐만 아니라 쇼핑몰 등에서 볼 수 있는 것처럼 중요한 영업 활동의 장이 되고 있다. 이와 같이 불특정 다수를 대상으로 하는 분야뿐만 아니라 특정 기업이나 조직의 구성원들이 내부적으로 정보를 공유하기 위해 Web 기술을 적용하는 경우가 있는데 이를 일반적으로 인트라넷이라고 부른다. Web이라고 하는 개방된 환경이 실제 이용에 있어서 보안의 문제가 많이 따르는 것처럼 인트라넷도 그러하며, 이에 대한 연구가 활발히 이루어지고 있다. Z. Tari와 S. Chan은 RBAC을 인트라넷의 보안문제에 적용하는 연구를 통해 인트라넷-RBAC(I-RBAC)의 개념을 소개하였다[9].

3.1 인트라넷에서 보안의 필요성

인트라넷은 일반적으로 TCP/IP로 연결된 인터넷의 장점을 기업 내부 업무에 적용하는 네트워크로 정의된다. 기존의 기간 시스템과 Web 관련 신기술을 결합해 시스템을 구축하는 새로운 개념의 전산환경으로 볼 수 있다. 멀티미디어 환경하에서 전세계의 어느 곳에서든 정보 공유가 가능한 인트라넷은 세계화를 추구하는 대기업들에게는 매력적임에 틀림이 없다. 그러나 인트라넷이 본격적으로 도입되었을 때 기업의 비밀이 외부로 유출될 가능성이 있다. 인트라넷에는 세 개의 보안 위협 영역이 존재하는데 그것은 저장장소, 접근 그리고 전송의 영역이다. 저장장소 보안은 하나 이상의 서버들 안에 존재하는 물리적 자원들에 대한 보호를 말하고, 접근 보안은 인트라넷에서 이용할 수 있는 논리적 자원들에 대한 접근제어 및 사용자 인증에 관심을 둔다. 전송 보안은 정보가 네트워크를 통해 전달될 때의 정보 보호와 관련이 있다. 전송 보안을 위해 암호화 기법, 전자 서명 등이 이용된다. 기본적으로 인트라넷을 다른 개방된 네트워크로부터 보호하기 위해서는 방화벽(firewall)이 사용된다. RBAC은 접근 보안 분야에 적용될 수 있으며, 네트워크 객체가 인트라넷 자원들에 접근하기 위해 적절한 권한을 갖는지를 결정하는 것과 네트워크 객체들의 권한을 어떻게 검사해야 하는지를 결정한다.

3.2 인트라넷 환경에서 RBAC의 적용 특징

인트라넷을 위한 RBAC(인트라넷-RBAC)은 여러 부분에서 기존의 RBAC과 차이가 있다.

첫번째로 역할(role)은 인트라넷 내에서 지역적 혹은 전역적 인가권한들 중 하나를 참조한다. 지역적 인가권한(local permission)은 하나의 네트워크 객체가 각각의 서버들 안에서 이용할 수 있는 서로 다른 네트워크 객체들 상에서 가지는 권한(privilege)들의 집합을 말한다. 전역 인가권한(global permission)은 전체 인트라넷 서버들 위에서의 권한들의 집합이다. 지역과 전역 인가권한들 사이의 중요한 차이는 인가권한 집합의 크기에 있는데 그것이 각각 네트워크 객체와 서버들일 수 있다는 점에서의 차이이다.

두번째로 지역-역할(local role)들은 각각의 서버들 안에서 네트워크 객체들 위의 지역적 권한들과 관련이 있는 특수화된 역할이다. 반면에, 전역-역할(global role)들은 네트워크 서버 상에서 전역적 권한을 갖는 일반적인 기업의 역할들을 말한다. 예를 들어 웹 서버 내에서 지역-역할(local role)들은 웹 관리자(Web administrator), 웹 에디터(Web Editor) 등일 수 있고 이러한 지역-역할은 이러한 웹 서버 안에서 특별한 작업들을 수행하기 위해 특정한 그들만의 직업 기능을 가질 것이다. 이러한 지역-역할과 관련된 지역적인 권한은 웹 서버 내에서 어떠한 웹 페이지를 판독, 기록, 수정 그리고 생성하기 위한 권한(privilege)들이다. 법인 회사 안에서 전역-역할은 사장, 부장, 과장 그리고 평사원 등으로 구성되어 진다. 이와 같은 전역-역할들에는 서로 다른 서버들 안에서 서로 다른 지역-역할들을 가진 인트라넷의 자원들을 접근하기 위해 전역 권한들이 주어진다.

세번째로 인트라넷 환경 내에서 보안정책의 유연한 실행을 제공하기 위해 두 가지 형태의 역할 계층(role hierarchy)을 포함한다. 지역-역할 계층들은 개개의 서버들의 자원을 사용하기 위한 지역적 혹은 전역적 네트워크 객체들의 권한들을 표현한다. 전역-역할 계층은 전체 인트라넷을 통과하는 자원의 접근성과 연관되어 있다. 계층의 각 타입은 서로 다른 정책들(지역적인 정책들과 전체 인트라넷을 위한 전역적인 정책들)을 실시하기 위해 적당한 메커니즘을 제공한다.

마지막으로 전역(global) 보안 정책들을 효율적으로 실시하기 위해서, 네트워크 객체의 권한허가를 검사하는데 이용하는 전역(global) 보안 데이타베이스 정보를 여러 서버에 중복하여 관리하는 중복 메커니즘(replication mechanism)을 사용한다.

3.3 인트라넷-RBAC 개요

본 절에서는 인트라넷-RBAC의 핵심 개념인 네트워크 객체, 인가권한, 역할에 대해 간단히 소개하고 역할 정보를 저장하는 지역-역할 데이타베이스와 전역-역할 데이타베이스의 구조에 대해 설명한다. 또한 예제를 통해 지역-역할과 전역-역할이 어떻게 매핑되는지를 보여준다.

(1) 인트라넷-RBAC의 핵심 개념

인트라넷-RBAC의 핵심 개념은 네트워크 객체(network object), 인가권한(permission), 역할(role) 이다.

● 네트워크 객체 : 네트워크 객체는 데이타를 저장하고 특별한 서비스를 수행하거나 혹은 네트워크 하드웨어의 부분을 나타내는 네트워크 개체(entity)로 정의될 수 있다. 개체는 사람, 사용자들과 같이 능동적일 수 있고 데이타베이스 서버, 웹 서버와 같이 수동적일 수도 있다. 네트워크 객체는 네트워크에서 유일한 식별자를 갖고 그것에 부여된 인가권한(permission), 특성(property), 그리고 값(value)을 갖는다. 인가권한이란 객체가 인트라넷 자원들에 관계되어 가지는 권한(privilege)들의 집합이고, 특성은 네트워크 객체의 특성들을 설명하는 이름, 생성일, IP주소와 같은 속성들(attributes)의 집합이다. 값(value)은 객체의 상태를 나타낸다. 즉 객체의 특성들의 각 속성을 위한 실제의 데이타 값을 말한다. 보안 정보는 하나의 리스트로서 네트워크 객체에 붙여질 수 있는데 이것은 접근권한이 주어진 믿을만한(trusted) 네트워크 객체들의 그룹을 기록하고 있다. 이것이 접근제어 리스트(ACL : Access Control List)이다. 표 2는 네트워크 객체의 표현 예를 보여준다. 표 2에서 'S1000'이라는 네트워크 객체는 'Name' 이라는 특성에대해 'File Object' 라는 값을 갖는데 이는 'S1000'이 'File Object'임을 나타낸다. 'S1000'은 'j_mail', 'j_report' 라는 객체에 대해서는 생성, 삭제, 읽기, 쓰기 권한을 가지며, 'j_project'라는 객체에 대해서는 읽기 권한만을 갖는다. 'PK130'의 접근제어 리스트에 있는 '*'는 'PK130'이 모든 객체에 읽기와 쓰기 권한을 가지고 접근할 수 있음을 나타낸다.

● 인가권한(Permission) : 인가권한이란 네트워크 객체가 할 수 있는 일을 결정하는 권한(privilege)들의 종류를 묘사하는 속성들의 집합이다. 인트라넷 관리자가 네트워크 객체에 권한들을 지정한다. 인트라넷-RBAC에서는 단순성을 위해 수퍼바이저(S), 생성(C), 삭제(D), 읽기(R), 쓰기(W), 실행(X)의 여섯 가지 권한을 가정한다.

● 역할(role) : 역할은 접근제어에 대한 상위 레벨의 표현이다. 그것은 하나의 네트워크 객체 혹은 그것들의 그룹과 관련이 있을 수 있고 서로 다른 인가권한들과 연관되어 있다. 동일한 역할이 할당된 모든 네트워크 객체는 그 역할과 연관된 인가권한을 가지고 같은 특권들을 공유한다. 그 역할과 ACL(Access Control List)안에 있는 접근 가능한 네트워크 객체들과 연관된 모든 인가권한들은 사용자들이 네트워크 객체에 접근할 수 있는 권한이 있는지를 검사하는데 사용되며 역할 테이블(role table)안에 기록된다. 표 3은 역할 테이블의 예를 보여준다. 인트라넷-RBAC에서도 RBAC에서와 같이 역할계층이나 전용역할(private role)을 지원한다. 인트라넷-RBAC이 RBAC과 다른 점은 인트라넷-RBAC의 역할은 지역-역할(local role)과 전역-역할(global role)로 나누어진다는 것이다. 지역-역할은 하나의 서버 영역에 국한된 역할을 말하며 전역-역할은 인트라넷 전체의 영역 수준에서 부여되는 역할을 말한다. 각각은 지역-역할 데이타베이스와 전역-역할 데이타베이스에 그 정보가 저장된다.

(2) 지역-역할 데이타베이스(Local Role Database)

지역-역할 데이타베이스(local role database)는 인트라넷 내에서 한 서버 내의 네트워크 객체들의 접근 정보를 저장한다. 지역-역할 데이타베이스

표 3. '쓰기(write)' 인가권한 도메인 테이블

네트워크 객체 / 역할	Mail-object (from Mail Server)	S1000 (from File Server)
(r_{11}) Project	1	1
(r_{12}) Administrator	1	1
(r_{13}) Marketing	1	0

표 4. 네트워크 객체의 표현 예

네트워크 객체	특성 (property)	값 (value)	인가권한	ACL
S1000	Name	File Object	[-,C,D,R,W,-]	j_mail, j_report
S1000	Name	File Object	[-,-,-,R,-,-]	j_project
PK130	Name	admin.	[-,-,-,R,W,-]	*

는 하나의 역할이 접근할 수 있는 모든 네트워크 객체들을 표현하기 위해 서로 다른 많은 수의 인가권한 도메인 테이블(permission domain table)들을 포함하고 있다. 하나의 인가권한 도메인 테이블은 특정 인가권한 도메인 안에서 어떤 역할들이 네트워크 객체들을 접근 할 수 있는지를 나타낸다. 하나의 인가권한 집합은 여섯 가지 타입의 권한에 관한 정보를 포함하고 있으며, 하나의 지역-역할 데이타베이스는 인가권한의 각 타입에 대해 하나씩(수퍼바이져, 생성, 삭제, 읽기, 쓰기, 그리고 실행) 여섯 개의 권한 도메인 테이블들을 갖는다. 표 4는 표 3의 역할 테이블에 대한 '쓰기' 인가권한 도메인 테이블을 나타내며 Marketing 역할은 Mail-object에 대해서는 쓰기 권한을 갖지만 S1000에 대해서는 쓰기 권한이 없음을 표현한다. 이러한 인가권한 도메인 테이블은 역할 테이블로부터 만들어질 수 있다. (표에서 r_{11}, r_{12},...은 지역-역할을 간단히 나타내기 위한 역할의 ID이다).

(3) 전역-역할 데이타베이스(Global Role Database)

인트라넷 안에서 전역적 권한들은 전역-역할들과 전역-역할 계층에 의해 구체화되고 관리된다. 모든 인트라넷 서버들은 주어진 네트워크 객체가 하나의 연산을 수행하기 전에 권한들을 검사하기 위해 각 구성요소 서버 위에 지역-역할 데이타베이스를 유지하고 있다. 그러나 네트워크 객체는 다른 서버들 상에서 다른 연산들을 시도할 수가 있다. 그래서 네트워크 객체는 서로 다른 자원들의 사용을 허락하는 지역(local)과 전역(global)적 권한들을 가지고서 전역적으로 행동한다. 그러한 네트워크 객체를 전역 네트워크 객체(global network object)라고 부른다.

설계 관점에서, 전역과 지역 네트워크 객체들 사이의 중요한 차이점은 전역 네트워크 객체는 전체 인트라넷을 통해서 유일한 식별자를 갖는다는 것이다. 지역 네트워크 객체는 단지 대응하는 지역 서버에게만 알려진 식별자를 갖는다.

인트라넷 자원들의 보안 접근 권한을 위임하기 위한 전역-역할(global role)을 소개한다. 전역-역할은 인트라넷 관리자로 하여금 다수의 서버들을 통해서 네트워크 객체들을 위한 권한을 지정할 수 있도록 한다. 이러한 역할들은 개개의 지역-역할들로 설명된다. 그래서 전역-역할은 다수의 서버들 안에서 다수의 지역-역할들을 가질 수 있다. 또한 모든 적당한 지역-역할들의 집합은 전역적 네트워크 객체의 전역적 권한들을 정의할 수 있다. 아래에 표 5는 지역-역할의 측면에서 전역-역할들의 정의를 설명하고 있으며, 전역-역할 R_1은 GS_1 서버에 있는 지역-역할 r_{11}, r_{12}과 GS_3 서버에 있는 지역-역할 r_{31}의 합집합으로서 정의됨을 나타낸다.

그림 3은 전역-역할과 지역-역할과의 관계를 설명한다. 전역-역할은 각 인트라넷 서버를 위한 지역-역할들의 리스트를 포함하고 있다. 각각의 지역-역할은 자신이 속한 서버의 네트워크 객체들을 위한 권한 집합을 포함하고 있는 각각의 ACL을 갖고 있다. 하나의 서버는 이러한 네트워크 객체를 하나 이상 포함하고 있다. 전역-역할에서 Is-a는 전역-역할 계층(global role hierarchy)이 존재함을 나타낸다. 전역-역할 계층은 인트라넷을 위한 전반적인 논리적인 권한 계층을 명시한다. 계층은 역할들과 함께 보안 관리를 위한 높은 레벨의 추상화를 제공하고 권한지정 작업을 단순화시킨다.

표 5. 역할 테이블

역할	인가권한	ACL
Project	[-,C,D,R,W,X]	Mail-object
Administrator	[-,C,D,R,W,X]	Mail-object
Marketing	[-,-,-,R,W,X]	Mail-object
Project	[-,-,-,R,W,X]	S1000
Administrator	[-,C,D,R,W,X]	S1000
Marketing	[-,C,D,R,-,X]	S1000

표 6. 전역-역할 테이블의 예

전역-역할	전역 인가권한	ACL (접근제어 리스트)
R_1	r_{11}, r_{12}	GS_1
R_2	r_{11}, r_{13}	GS_1
R_3	r_{11}, r_{15}	GS_1
R_1	r_{31}	GS_3

지역-역할 데이타베이스처럼 전역-역할 데이타베이스(global role database)는 어떠한 인트라넷 서버로부터도 접근이 가능한 모든 네트워크 객체들을 기록한다. 전역-역할 데이타베이스 내에서 전역-역할 테이블은 해당 전역-역할들을 위한 모든 권한들을 저장한다. 형식적으로 하나의 전역-역할 테이블은 지역-역할 테이블과 비슷하다. 그것은 트리플 triple(rg,pg,lg)로 표현될수 있는데 rg는 전역-역할들을 명시하고, pg는 전역적 권한들을 명시하며, lg는 ACL을 의미한다. 전역-역할 데이타베이스의 정보 접근 혹은 수정작업을 중앙으로 집중화하는 것으로 인한 비효율성을 피하기 위해서 시스템은 전역적인 보안 정보의 이용성(availability)을 증가시키기 위해서 각 서버마다 전역-역할 데이타베이스를 중복(replicate)해서 갖도록 한다. 이러한 중복 관리는 하나의 서버가 작동하지 않는 상황에서도 RBAC 메카니즘이 정상적으로 실행되도록 하는 장점이 있지만 한 복사본이 수정될 때 다른 복사본들과의 일관성(consistency)문제가 발생한다. 이러한 문제는 전체 순서화(total ordering)라고 불리는 알고리즘으로 해결한다.

(4) 전역-역할과 지역-역할 데이타베이스의 매핑

그림 4는 인트라넷상에서 전역 역할과 지역 역할이 어떻게 매핑되는지의 예를 보여준다. 그림에서 GRH는 전역-역할 계층을 나타내며 서버 GS_1과 서버 GS_2에 중복하여 존재함을 알 수 있다. LRH는 지역-역할 계층을 나타내며 각 서버마다 고유의 정보를 가지고 있다. GRH와 LRH 밑에 있는 도식의 의미는 다음과 같다.

- R_1:{r_{11},r_{12}}{GS_1}{r_{31}}{GS_3}

  R_1은 전역-역할로서 서버 GS_1에 있는 지역-역할 r_{11}, r_{12}과 서버 GS_3에 있는 지역-역할 r_{31}로서 정의된다.
- r_{11}:{-,-,-,R,W,X}{no_{11},no_{12}}

  r_{11}은 서버 GS_1에 존재하는 지역-역할로서 네트워크 객체 no_{11}, no_{12}에 대해 읽기, 쓰기, 실행 권한을 갖는다.

지역-역할과 전역-역할이 어떻게 작용하는지를 이해하기 위해 응용 서버(GS_1)의 Tom이라는 사용자가 SQL 서버(GS_3)에 있는 인사화일(no_{32})을 읽으려고 시도하는 경우를 가정해보자. 먼저 Tom이 시스템에 로그인할 때 Tom이 합법적인 사용자인지를 확인해야 하는데 이는 네트워크 객체 테이블에서 Tom에 관한 정보를 확인함으로써 이루어진다. Tom이 합법적인 사용자임이 판명되면 접근 하고자 하는 인사화일(no_{32})이 다른 서버에 있으므로 전역-역할 테이블을 참조하여 Tom에게 부여된 전역-역할을 찾는다. (만일 Tom이 접근 하고자하는 네트워크 객체가 응용 서버(GS_1) 내에 있다면 지역-역할을 참조하면 된다.) 만일 Tom에게 아무런 전역-역할도 부여되어 있지 않다면 Tom은 다른 서버의 객체에 접근할 수 없다. Tom에게 R_2 라는 전역-역할이 주어졌다면 R2로부터 지역-역할 r_{11}, r_{13}, r_{33}이 유도될 수 있다. 시스템은 지역-역할 r_{11}, r_{13}, r_{33}의 유효성을 검증하기 위해 서로 다른 서버들의

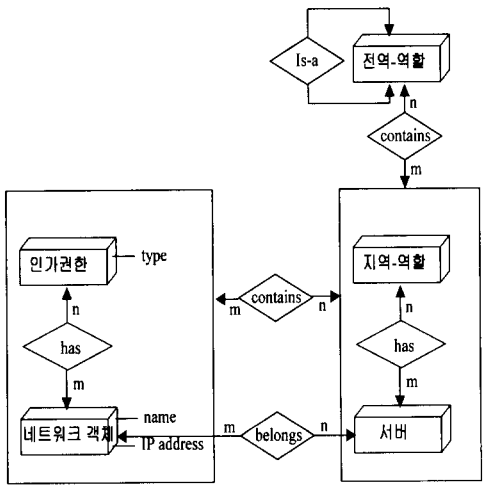


그림 3. 전역-역할의 주요 요소와 지역-역할과의 관계

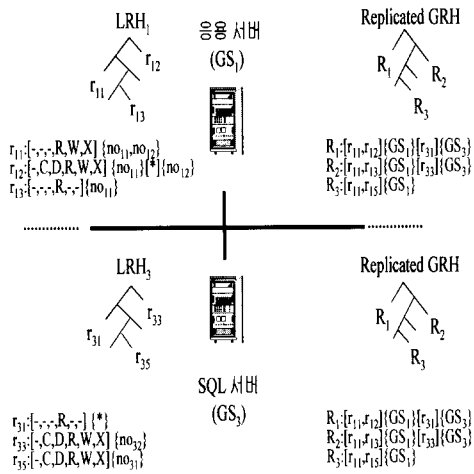


그림 4. 전역-역할과 지역-역할의 매핑 예제

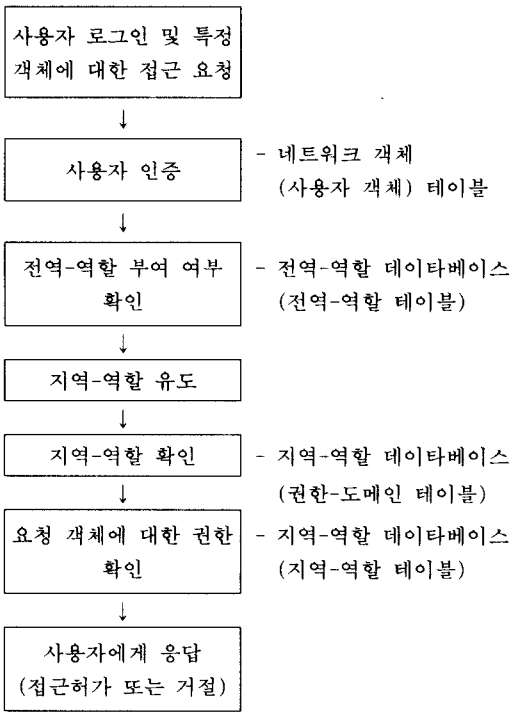


그림 5. 특정 객체에 대한 사용자의 접근과정

지역-역할 데이타베이스에서 권한 도메인 테이블들을 참조한다. 이 과정에서 SQL 서버(GS_3)의 지역-역할 r_{33}이 네트워크 객체 인사화일(no_{32})에 대한 읽기 권한을 가지므로 Tom이 인사화일(no_{32})을 읽는 것이 허용된다. 이상의 과정을 도식적으로 정리하면 그림 5와 같다.

Ⅳ. 인트라넷-RBAC의 구현

4.1 에이전트 기반의 인트라넷-RBAC 구현

제안된 인트라넷을 위한 RBAC은 에이전트(agent)를 기반으로 구현될 수 있다[9]. 에이전트란 인트라넷의 자원들을 이용, 접근, 수정하기 위해 사용자의 지역 그리고 전역적 권한을 검사함으로써, 서로 다른 보안 절차들을 구현한 능동적인 네트워크 객체들이다. 단순하게 에이전트는 인트라넷 환경 안에서 특별한 작업들을 수행할 수 있는 능동적 개체로 취급된다. 에이전트들은 두개의 주요 구성요소들을 포함하는데 인터페이스(interface)와 구현부분(implementation)이다. 인터페이스는 인트라넷 내에서 시행될 필요한 보안 절차들을 설명하는데, 예를 들어 이러한 절차들은 사용자 인증, 전역적 권한 검사, 지역적 권한 검사 등을 포함하고 있다. 에이전트의 구현부분은 인트라넷 서버 내에서 특별한 에이전트의 인터페이스를 지원하기 위해 어떻게 보안 절차들이 수행될 수 있는지를 설명한다.

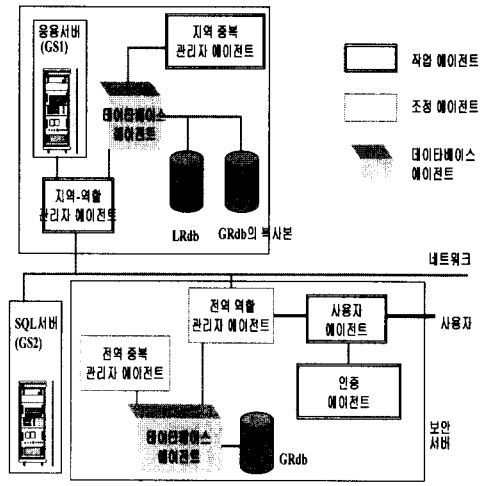


그림 6. 인트라넷-RBAC의 구현시 필요한 에이전트와 구조도

모든 요구된 보안 절차들을 시행하기 위한 하나의 에이전트를 사용하기보다는 현재 인트라넷-RBAC의 경우, 인트라넷을 안전한 상태로 유지하는 데에 서로 다른 책임이 있는 서로 다른 에이전트들을 사용해서 설계하였다. 여기서는 에이전트를 세 가지 타입으로 구별하는데 즉, 조정 에이전트(coordination agent), 작업 에이전트(task agent) 그리고 데이타베이스 에이전트(database agent)이다. 그림 6은 이러한 에이전트들을 보여주고, 그림 7은 그들간의 상호작용을 설명하고 있다.

그림 5와 그림 6에서 조정 에이전트는 보안정책 시행과 사용자 인증을 포함해서 인트라넷의 여러 다른 활동들을 관리한다. 전역-역할 관리자(global role manager)는 보안 활동을 조정하고 전역-중복관리자(global replica manager)는 네트워크에 문제가 발생했을 때 서로 다른 서버에 있는 역할 데이타베이스의 유용성을 유지한다. 작업 에이전트는 인트라넷에서 사용자 질의의 최적화와 사용자 권한의 검사와 같은 활동에 대한 책임이 있다. 지역-역할 관리자(local role manager)는 지역 사용자들의 권한을 검사하는 보안 절차들을 수행하며 지역-중복 관리자(local replica manager)는 역할 데이타베이스의 서로 다른 복사본들을 일관성 있게 유지한

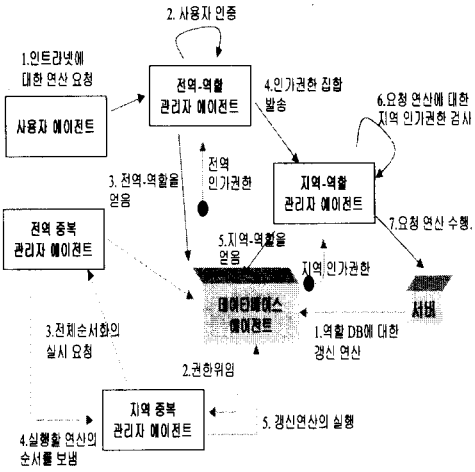


그림 7. 에이전트간의 상호작용

다. 데이타베이스 에이전트는 역할 데이타베이스 내에 있는 지역-역할과 전역-역할을 수정하는 것과 같은 전형적인 데이타베이스의 기능을 수행한다.

4.2 CGI 기반의 RBAC 구현

J. F. Barkley는 CGI를 기반으로 Web 서버상에서 RBAC의 구현(RBAC/Web)에 관한 논문을 발표하였다[11]. 인트라넷-RBAC을 전제로 구현되지는 않았지만 인트라넷-RBAC의 구현에도 충분히 적용될 수 있다. RBAC/Web은 역할과 역할 계층, 제약조건을 포함하며 각 사용자는 유일한 ID를 부여받는다. RBAC/Web에서 권한(privilege)은 GET, HEAD, PUT과 같은 HTTP 메소드(method)로써 구현된다. RBAC/Web의 주요 구성요소는 표 6과 같다.

사용자가 RBAC/Web 서버를 이용하는 과정은 그림 8과 같다.

① 사용자는 일반 Web 브라우저를 통해 URL을 입력하여 Web 서버에 접속한다. 이 과정은 RBAC/Web의 제어 대상이 아니다. 그러나 RBAC에 의해 제어되는 URL을 이용하기 위해서는 먼저 RBAC 세션을 개설해야 한다.

② RBAC 세션이 사용자에게 열리면 사용자는 활성-역할-집합(ARS : current active role set)을 선택한다. 이를 통해 사용자에게 특정 역할이 할당되며, RBAC에 의해 제어되는 URL 상에서

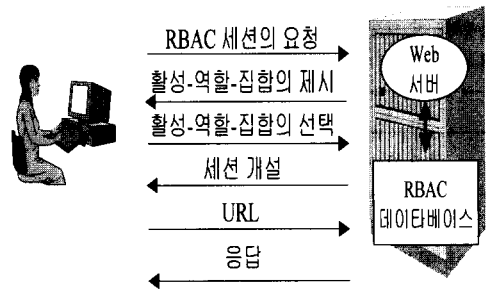


그림 8. RBAC/Web의 사용 과정

수행할 수 있는 연산(operation)이 결정된다. 이 활성-역할-집합은 사용자가 다른 활성-역할-집합을 설정할 때까지 유효하다.

③ 권한을 얻은 사용자는 원하는 URL에 접속하여 자신의 권한 범위 내에서 필요한 연산을 수행하며 Web 서버는 CGI를 통해 사용자의 권한과 요구된 연산을 비교하여 서비스 여부를 판단한다.

④ 작업이 끝나면 사용자는 RBAC 세션을 종료한다.

CGI 방식은 기존의 Web 서버를 변경하지 않고도 Web 환경에 RBAC을 구현할 수 있다는 점에서 장점이 있으며 4.1에서 제시한 에이전트 개념의 실제 구현 수단으로서 사용될 수도 있다. 현재 RBAC/Web은 UNIX와 NT환경에서 구현중에 있

표 6. RBAC/WEB의 구성요소

Database	사용자와 역할의 연결, 역할 계층, 사용자/역할의 제약조건, 현재 활성화된 역할들, 역할과 연산(operation)의 관계에 대한 정보를 저장하는 화일들의 집합으로 Cache에 상주한다.
Database Server	위의 Database를 지원하는 HOST. 위의 화일들은 Admin Tool에 의해 관리되며 이 화일들에 변동이 있을 때 Database server는 Web server에게 변경된 내용을 통보한다.
API Library	Web server 와 CGI 가 RBAC/Web database에 대한 접근할 때 이용되는 명세(specification). C 등의 언어로 작성됨
CGI	RBAC은 CGI 로 구현되며 Web server에 의해 이용된다. 이때 Web Server는 아무런 변경을 필요로 하지 않는다.
Session Manager	RBAC 세션을 관리. 사용자의 활성-역할-집합(ARS)을 생성하거나 제거 한다.
Admin Tool	Server 관리자로 하여금 사용자, 역할, 허가된 연산, 사용자와 역할의 연계, 할과 권한의 연계등을 정의하고 변경할 수 있도록 지원되는 도구. Web 브라우저를 통해 Admin Tool을 사용한다

다.

V. 인트라넷-RBAC의 평가 및 연구과제

제안된 인트라넷-RBAC 접근제어 모델은 인트라넷 내에서 효율적인 접근 관리 방법을 제공한다. 기업 인트라넷의 보안은 대체로 접근제어 관리에 의존한다. 제안된 인트라넷-RBAC의 한가지 장점은 기업 인트라넷의 서로 다른 보안 접근 정책들을 만족시킬 수 있는 유연성에 있다. 이것은 사용자의 역할 지정에 근거한 효율적인 사용자 권한 메커니즘을 포함시킴으로서 인트라넷-RBAC 내에서 이루어진다. 그리고 서로 다른 서버들을 위한 책임을 구조화하는 메커니즘을 제공하기 위해 에이전트 기반의 인트라넷-RBAC 구현이 이용될 수 있다. 침입자들로부터 인트라넷 서버에 있는 자원들을 보호하기 위한 보안 에이전트들은 익명에 의한 접근을 제거할 수 있고, 따라서 어떤 불법적이거나 안전하지 않은 네트워크 자원들의 이용을 중단시킬 수 있다.

인트라넷-RBAC의 구현을 위해서 제안된 에이전트 모델은 분산 환경에서의 보안 관리를 위한 적합한 구조를 가지고 있다. 이러한 에이전트 모델은 아직 개념적으로만 정의되었기 때문에 실제 구현을 위해서는 추가적인 연구가 필요하다. CGI 기반의 RBAC/Web은 일반적인 Web 환경에서 RBAC을 실제적으로 구현한 모델로서 기존의 Web 서버를 그대로 이용해서 구현할 수 있는 장점이 있다. 인트라넷-RBAC의 실제적인 구현에 있어서 RBAC/Web에서 제시된 기술들이 그대로 이용될 수 있다고 판단된다.

인트라넷-RBAC은 여러 장점 더불어 다음과 같은 한계점들을 가지고 있는데 이는 인트라넷-RBAC의 계속적인 연구과제이다.

- 한가지 중요한 문제는 역할들 사이의 일관성과 관련된다. 이것은 지역-역할과 전역-역할 데이타베이스 측면에서 표현될 수 있다. 인트라넷이 정적인 환경이 아니고 동적으로 변화해 가는 환경이기 때문에 이에 따른 역할들의 변화는 새로운 보안 요구사항들을 반영하면서, 서로 다른 데이타베이스들의 일관성을 유지하기 위해 관리되어야 한다. 이 문제의 해결책은 정적 그리고 동적인 임무분리에 기초를 두게 될 것이다.
- 위와 관련이 있는 문제로서 인트라넷에 지역적으로 멀리 떨어져 있는 여러 서버들이 포함되고 다국적으로 흩어져 있는 수많은 조직들이 이를 이용하게 될 경우, 많은 수의 지역-역할, 전역- 역할들이 필요하게 되고 역할 그 자체만을 관리하는데도 많은 비용을 필요하게된다. 이를 해결하기 위해 ARBAC(Administrative RBAC)의 개념이 인트라넷-RBAC에 포함될 필요가 있다.
- 인트라넷-RBAC에서는 신뢰성과 유용성을 위해 동일한 전역-역할 데이타베이스를 여러 서버에 중복해서 관리하는 방법을 제안하고 있다. 그리고 중복된 복사본들 사이의 일관성은 전체 순서화(total ordering)이라는 알고리즘에 의해 유지될 수 있다고 제안한다. 그러나 많은 수의 서버에 존재하는 복사본들의 관리는 분산 환경에서 해결하기 매우 어려운 문제중의 하나이다. 일어날 수 있는 가능한 모든 상황들이 고려되어야하며 인트라넷-RBAC의 환경에 적합한 방법론이 선택되어야 한다.
- 인트라넷-RBAC에서는 사용자(user)와 사용자가 접근하고자하는 객체들이 모두 네트워크 객체라는 단일 자료구조로서 표현되고 관리되기 때문에 기존의 RBAC 모델에서 「사용자-역할-인가권한-접근 대상객체」라는 연결성이 모호하다 특히 사용자-역할 배정(user roll assignment) 부분이 혼동을 일으킬 소지가 많다. 따라서 사용자에 대해서는 별도로 관리하는 것을 고려할 필요가 있다.
- 이제까지 연산의 병행수행 제어(concurrency control)에 관한 이슈는 인트라넷-RBAC 내에 언급되지 않았었다. 인트라넷 환경에서는 서로 다른 사용자들에 의해 실행되는 다수의 병행 연산들이 있을 수 있다. 문제는 특별히 병행수행 연산들이 인트라넷 내에서 같은 정보를 수정할 때 발생한다. 이 문제의 해결책은 분산 데이타베이스 내에서 행해졌던 연구로부터 빌려올 수 있는데, 일관성 만족을 위한 로킹(locking)과 타임 스탬프(time-stamp) 프로토콜 같은 것이 있을 수 있다.

VI. 결 론

인트라넷은 최근 기업의 생존을 좌우하는 경쟁력의 핵심 요소로 자리매김을 하고 있다. 특히 세계화를 지향하는 대기업들에게는 더욱 절실하다. "1천대 기업 중 90%가 인트라넷을 구축중이거나 운영중"이라는 최근 미국의 전문 시장조사 업체의 보고서에서 보이듯 인트라넷의 도입은 전 세계적인 추세이다. 이러한 인트라넷에서의 가장 큰 문제는 보안이라고 할 수 있다. 인트라넷 보안은 지금까지는 방화벽과

같이 주로 외부의 침입자들로부터 시스템을 보호하는 일에 초점이 맞추어져 있었다. 그 결과 내부에 있는 다수의 사용자가 동일한 정보 시스템을 이용하게 될 때 개인, 부서 혹인 기업 차원의 기밀 정보를 효과적이고 안전하게 보호하면서도 사용자에게는 불편을 최소화시키는 보안 문제가 대두되었다. RBAC은 이에 대한 하나의 대안으로 주목받고 있는데 그 이유는 RBAC이 가진 보안 체계가 일반 기업이나 조직의 업무 형태와 유사하여 적용이 쉽고 유연성이 뛰어나기 때문이다.

인트라넷-RBAC은 인트라넷상에서 발생하는 보안문제에 RBAC을 적용한 것으로 지역-역할과 전역-역할을 구분하고 이를 적절히 관리함으로서 보안관리에 대한 모델을 제공하고 있다. 에이전트 모델과 CGI 방법이 인트라넷-RBAC을 구현하기 위한 수단으로 제시되었다. 인트라넷-RBAC은 아직은 모델을 제시한 단계이기 때문에 이론적으로 보완되어야할 부분이 많고 구현기술의 측면에서도 연구가 부족한 실정이지만 기업의 인트라넷 환경에서 발생하는 보안문제를 해결하기 위한 유용한 방법으로서 자리잡게될 전망이다. 따라서 국내에서도 이에 대한 지속적인 연구가 필요하다.

참 고 문 헌

[1] E. G. Amoroso, Fundamentals of Computer Security Technology, Prentice Hall Inc., 1994.

[2] R. S. Sandhu and P. Samarati, "Access Control : Principles and Practice", IEEE Communication Magazine, 9, 1994, pp. 40-48.

[3] D. Ferraio, J. Cugini and R. Kuhn, "Role-based Access Control(RBAC): Features and motivations", Proc. of 11th Annual Computer Security Application Conference, 1995.12.

[4] C. Ramaswamy and R. Sandhu, "Role-Based Access Control Features in Commercial Database Management Systems", NISSC, 1998.

[5] R. Sandhu and Gail-Joon Ahn, "Group Hierarchies with Decentralized User Assignment in Windows NT", ASTED-CSE, 1998.

[6] R. Sandhu, V. Bhamidipati, E. Coyne, S. Ganta, and C. Youman, "The ARBAC97 model for role-based administration of roles : Preliminary description and model", Proc. of the second ACM Wrokshop on rRole-Based Access Control, ACM, 1997.11.

[7] R. Sandhu and V. Bhamidipati, "The URA97 Model for Role-Based User-Role assignment", Proc. of IFIP WG 11.3, 1997.8.

[8] R. Sandhu and Q. Munawer, "The RRA97 Model for Role-Based Administration of Role Hierarchies", ACSAC, 1998.

[9] Z. Tari and Shun-Wo Chan, "A Role Based Access Control For Intranet Security", IEEE INTERNET COMPUTING Magazine, 1997.10.

[10] R. Sandhu and J. S. Park, "Decentralized User-Role Assignment for Web-based Intranets", Proc. of 3rd ACM Workshop on Role-Based Access Control, 1998.10.

[11] J. F. Barkely, A. V. Cincotta, D. F. Ferraiolo, S. Gavrilla and D. R. Kuhn, "Role Based Access Control For the World Wide Web", 20th NCSC, 1997.4.

[12] "An Introduction to Role_Based Access Control", SEM security news, http://www. semshred.com/sem_secrbac.htm.

[13] R. Sandhu, E. Coyne, H. Feinstein, and C. Younman, "Role-Based Access Control Models", IEEE Computer Magazine Vol. 29, 1996.2.

[14] R. Sandhu and H. Feinstein, "A Three Tier Architecture For Role-Based Access Control", Proc. of 17th NIST-NCSC, 1994.10.

[15] NIST의 RBAC관련 프로젝트, http://hissa.ncsl.nist.gov/rbac/.

[16] R. Sandhu and V. Bhamidipati, "An Oracle Implementation of the PRA97 Model for Permission-Role

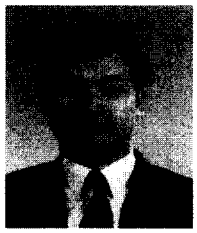
Assignment", Proc. of 3rd ACM wrokshop on Role Based Access Control, 1998.
[17] Common Criteria : Role-Based Access Control (RBAC) Protection Profile (v1.0), http://csrc.nist.gov/cc/pp/pplist.htm.

오 세 종(Se-jong Oh)

1989년 2월 : 서강대학교 전자계산학과 졸업
1991년 2월 : 서강대학교 전자계산학과 석사
1997년9월~현재 : 서강대학교 컴퓨터학과 박사과정

<관심분야> 데이터베이스 보안, 병행수행제어

박 석(Seog Park) 정회원



1978년 2월 : 서울대학교 계산통계학과 졸업
1980년 2월 : 한국과학기술원 전산학과 석사
1983년 2월 : 한국과학기술원 전산학과 박사1983년~ 현재 : 서강대 컴퓨터학과 교수
1989년~ 1991년 University of Virginia 방문교수

<관심분야> 실시간 데이타베이스, 보안 데이타베이스, 주기억장치 데이타베이스, 멀티미디어 데이타베이스, 트랜잭션 관리, 병행수행제어