# 보안 시스템을 위한 RBAC 설계

정회원 김 성 열*

# Design of the RBAC for Security Systems

Seong Ryeol Kim* *Regular Member*

요 약

본 논문은 보안 시스템을 위한 RBAC 설계에 관한 논문이며, 보안 시스템의 관심은 지난 몇 년 동안 지대하다. 그러나 현재의 보안 구조들은 부족하며 많은 문제점이 있다. 따라서 새로운 보안 시스템을 위한 RBAC의 설계를 제안한다. RBAC는 새로운 기술로 커다란 망기저 응용(network based application) 내에서 보안관리 비용과 복잡성을 줄일 수 있다는 이유로 특별한 상용 응용들에서 많은 관심이 증가되고 있다. 본 논문에서 설계 제안되는 RBAC는 서버 소프트웨어의 수정 없이 상업적으로 이용 가능한 응용 서버들과 연결할 수 있다.

ABSTRACT

This paper presents the design of the RBAC (Role-Based Access Control) for security systems. Security system interest have exploded in the last few years. But, on existing security frameworks are lack and many problems. Therefore, we propose to design of a RBAC for a new security systems. RBAC is a new technology that is attracting increasing attention, particularly for commercial applications, because of its potential for reducing the complexity and cost of security administration in large network based applications. The RBAC components can be linked with commercially available application servers, and require no modification of the server software.

## I. 서 론

컴퓨터 정보 통신망을 이용하는 분산 정보 시스템들이 최근에 폭 넓게 개발되어 사용되고 있다. 그러나 이러한 시스템들은 업무들의 운영과 수행, 통신망상에서 업무들의 접근과 통제에 대한 보안상의 문제가 커다란 문제점으로 대두되고 있다. 따라서, 본 논문에서는 보안 시스템을 위한 RBAC를 설계 제안한다. 그리고 분산 정보 시스템에 적용할 수 있도록 RBAC의 특징과 구현 환경에 대하여 기술한다.

RBAC는 특별히 상업적 응용에 적용할 수 있다는 기대[1]로 전형적인 임의 접근 제어(discretionary access control : DAC)와 위임 접근 제어(mandatory access control : MAC) 정책을 대신할 수 있다. RBAC의 기본적인 배경[2],[3]은 기업이나 기관의 조직 구조에 자연스럽게 접근하여 하부조직의 제어와 전형적인 접근제어 리스트 등과 같은 보안정책[4]을 강조하여 명세화를 요구할 수 있다는 데 있다.

RBAC의 보안은 조직 구조에 부합되는 단계별로 관리[1],[6]된다. 각각의 사용자는 하나 또는 하나이상의 임무가 할당되며, 각각의 임무에는 여러 임무들 중에서 사용자들에게 허가된 하나 또는 하나이상의 특권들로 할당[4]된다.

RBAC의 보안 관리는 고유 임무에 종사자를 할당하여 종사자가 특정 작업을 수행하도록 하는 결정된 행위들로 구성한다. 임무계층과 상호 배타적 임무들에 의해 설명되는 복잡성은 보안관리를 더욱 쉽게 하기 위한 RBAC 소프트웨어에 이용된다.

RBAC는 하나의 정책기술[5]로 RBAC 모형에 의해 연결된 직무 정책의 분할 즉, 대부분의 구조적 응용 속에서 미묘하게 분할되어 있지만 RBAC는 직무 분할이 자연스럽게 다양한 보안정책에 의해 구현할 수 있다. 이러한 직무 분할은 정적 직무 분

* 청주대학교 컴퓨터정보공학과(srkim@comnet.chongju.ac.kr)
논문번호 : 98398-0902, 접수일자 : 1998년 9월 2일

할과 동적 직무 분할로 나누어 구현한다.

## Ⅱ. RBAC의 설계

RBAC의 설계에 있어 직무, 임무와 특권의 관계는 그림 1과 같으며, 임무들은 계층적으로 구성할 수 있다.

보안 시스템에 적용할 RBAC의 설계는 기본적인 직분, 임무와 특권과 각각의 동작인 특권접근, 직분의 일치성, 임무할당, 임무인가, 특권인가, 임무계층 직무분할 등에 대하여 정형적으로 정의한다.

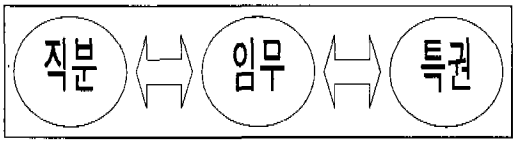


그림 1. 직분, 임무와 특권의 관계

RBAC 설계시 사용할 기본적인 변수는 다음과 같이 정형적으로 정의한다.

s : 직분(subject)
i, j : 임무(role)
p : 특권(privilege)
u : 사용자(user)

이러한 변수들을 사용하여 다음과 같은 직분, 임무, 접근특권, 직분의 일치성, 임무할당, 직무분할 등의 정의에 사용한다.

**직분 :**

U[s] = 직분 s에 관련된 사용자 u
R[s] = 인가된 직분 s를 위한 임무들의 집합
A[s] = 직분 s를 위한 수행중인 임무들의 현재 리스트

**임무 :**

M[i] = 임무 i를 위한 인가된 사용자들
P[i] = 임무 i를 위한 인가된 특권들
E = 각기 다른 상호 배제된 임무(i, j)들 집합

**특권접근 :**

X[s, p] = 참(true)일 때 특권 p를 수행할 수 있는 직분 s

변화가 없을 때에 RBAC 시스템에 의해 이전 상태가 계속 유지된다.

**직분의 일치성 :**

사용자들과 수행중인 사용자들의 행위와의 일치성

$$(\forall s, u, i)|U[s]=u : u\in M[i] \Leftrightarrow i\in R[s] \quad (1)$$

**임무할당 :**

하나의 직분이 수행할 수 있는 하나의 특권으로 직분을 선택하거나 하나의 동작임무를 할당할 때에 수행한다.

$$(\forall s, p) : X[s, p] \Rightarrow A[s]\neq\varnothing \quad (2)$$

**임무인가 :**

하나의 직분의 동작임무는 직분을 위해 인가되어야 한다.

$$(\forall s) : i\in A[s] \Rightarrow i\in R[s] \quad (3)$$

**특권인가 :**

하나의 직분은 하나의 특권을 수행해야 하는데, 특권은 직분의 현재 행위의 임무를 위하여 인가된다.

$$(\forall s, p)(\exists i) : X[s, p] \Rightarrow i\in A[s] \wedge p\in P[i] \quad (4)$$

식 (2)와 식 (3)은 하나의 직분은 하나의 특권을 수행할 수 있는 규칙으로 특권은 이러한 행위의 임무를 위하여 인가된다.

**임무계층 :**

임무가 인가되었거나 행위 임무 집합을 포함하거나, 임무들 하위에 부분순서집합(poset)이 포함되어 있다면 임무들은 부분 순서 집합으로 구성된다.

$$(\forall i, j, s) : (i\in A[s] \wedge i\geq j \Rightarrow j\in A[s]) \wedge (i\in R[s] \wedge i\geq j \Rightarrow j\in R[s]) \quad (5)$$

**직무의 분할 :**

정적 임무 분할은 관리자가 임무인가 작업을 동시에 상호배제 규칙을 수행할 수 있으나, 동적 직무 분할은 사용자가 하나의 세션을 위해 임무를 선택해야 한다. 이러한 정적 직무 분할의 정의는 식 (6)과 같이 정의하며, 동적 직무 분할의 정의는 식 (7)과 같이 정의한다.

$$(\forall u, i, j)[i \neq j : u \in M[i] \wedge u \in M[j] \Rightarrow (i, j) \notin E \quad (6)$$

$$(\forall u, s, i, j)[i \neq j : u \in M[i] \wedge u \in M[j] \Rightarrow$$

$$i \in A[s] \wedge j \in A[s] \Rightarrow (i, j) \notin E \quad (7)$$

이상과 같이 보안 시스템에 사용할 RBAC를 정형적으로 설계하였다.

정의된 RBAC는 기존의 프로그램 언어 즉, Visual BASIC, Visual C++, 4GL, CGI, ASP 또는 JavaScript 등을 이용하여 구현할 수 있다.

## Ⅲ. 보안 시스템의 RBAC 적용

보안 시스템의 RBAC 적용은 응용 서버 상에서 설계 제안된 RBAC의 구현 방법에 대하여 기술한다. 응용 서버에는 응용 소프트웨어들과 브라우저에 대한 특별한 요구사항이 필요 없기 때문에 임의 응용 소프트웨어나 브라우저라도 특별한 응용 서버에서도 사용할 수 있다. 따라서 응용 서버의 이용 가치를 높여줄 수 있는 응용 소프트웨어나 브라우저라면 모두 사용할 수가 있다.

이러한 RBAC를 적용하기 위한 모형은 그림2와 같다.

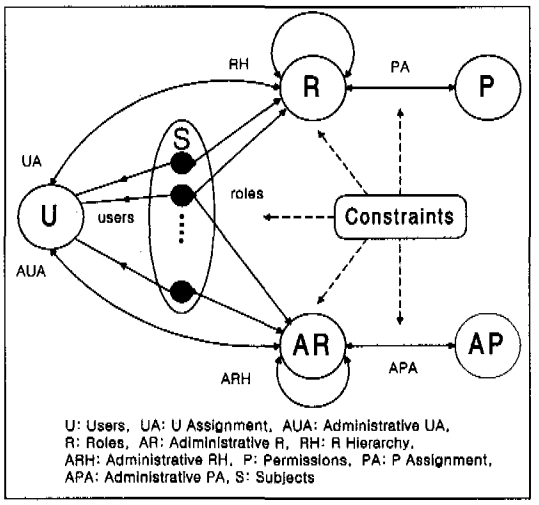


그림 2. 응용 서버를 위한 RBAC 모형

UNIX 환경의 응용 서버는 독립적인 응용 서버로 사용하는 방법과 UNIX 웹 서버와 같이 인터넷을 사용하는 방법[2]이 있다. 또한, NT용 응용 서버는 원시 코드를 변경하거나 서버 내부의 환경을 변경할 필요 없이 사용 가능하다.

가장 간단한 방법은 RBAC 응용 CGI로 대신하는 것이다. RBAC 응용 CGI는 현재의 UNIX 서버의 원시코드 수정 없이 사용할 수 있다. RBAC URL들은 RBAC 응용 CGI에 의해 처리되고 웹 서버를 통하여 전달된다.

응용 서버는 Netscape, NCSA, CERN 또는 Apache 서버들을 위한 UNIX 서버와 Internet Information Server, WebSite, 또는 공급자를 위한 Windows NT 환경하에서 구현된다.

이러한 응용 서버의 구성요소는 표 1과 같으며, UNIX 환경에서의 응용 서버 구성요소이며, Windows NT 환경은 NT에 포함되어 있는 보안 기술이 RBAC 구현에 적합하기 때문에 NT 버전은 단지 데이터베이스, 세션 관리자와 관리도구 요소를 사용한다.

표 1. 응용 서버 구성요소

| 구성요소 | 구성 내용 및 기능 |
|---|---|
| 데이터베이스 | 사용자와 임무들 사이의 관계, 임무계층, 사용자/임무 관계의 내용, 현재의 활동 임무들과 임무와 기능 사이의 관계 등을 상세히 기록한 파일들이다. |
| DB Server | 사용자와 임무들 사이의 관계, 임무계층, 사용자/임무 관계의 내용, 현재의 활동 임무들과 임무와 기능사이의 관계 등을 정의한 파일들을 신뢰할 수 있게 복사할 수 있는 호스트들이다. 이러한 파일들은 관리도구에 의해 생성되고 유지된다. 파일들이 변경될 때에 DB 서버는 응용 서버의 캐쉬의 내용을 수정할 수 있도록 통보한다. |
| API Library | 응용 DB를 접근하기 위해 응용 서버와 CGI 등에 의해 사용되는 명세서로, API는 응용 서버 구현시 RBAC를 추가할 수 있다. API 라이브러리는 응용 API 구현시 사용되는 C++, Perl, Java 라이브러리 등을 사용한다. |
| CGI, ASP, Java | 원시코드의 수정 없이 일반적으로 현존하는 응용 서버를 사용하기 위하여 CGI, ASP, Java 등을 RBAC 구현에 사용하며, 응용 CGI, ASP, Java 등은 응용 API 라이브러리를 사용한다. |
| Session Manager | RBAC 세션을 관리하는 관리자로 응용 세션 관리자는 사용자의 현재 활동임무의 집합을 생성하거나 삭제한다. |
| 관리도구 | 사용자, 임무들의 생성과 허가된 기능들, 사용자와 임무와의 관계와 허가된 기능의 임무들, 사용자/임무 관계의 내용명세, DB의 유지 등을 서버 관리자에게 허용한다. 관리자는 웹 브라우저를 통하여 응용 관리도구에 접근할 수 있다. |

RBAC 응용의 환경파일은 사용자의 임무들에 준하여 접근제어를 준비하는 동안에 파일 이름들에 의해 URL들로 연결된다. RBAC 응용 CGI의 설치는 웹서버의 설치와 유사하다.

RBAC 응용 CGI는 상대적으로 간단히 설치할수 있으나, 사용하는 동안에 웹서버 내에서 직접 접근제어를 수행할 수 없다.

RBAC 응용을 사용하기 위한 다른 방법으로는 RBAC 접근 결정을 위한 RBAC 응용 API를 UNIX 응용 웹서버 상에서 수정해야 한다. URL은 웹서버 환경파일과 파일 이름과 URL이 연결되는 RBAC 제어 URL로 구성된다.

넷스케이프와 아파치와 같은 UNIX 환경의 웹서버는 단계별로 운영되도록 환경 파라메타에 의해 치환되거나 보강된다. 이와 같은 방법으로 서버의 원시 코드 수정 없이 웹서버의 동작의 변경이 가능하다.

이러한 응용 웹서버를 위하여 RBAC 응용 API는 적절한 환경 수정 파라메타와 호출 순서는 간단한 사전준비로 통합될 수 있다.

### 1. 인증

RBAC는 하나의 접근제어 메카니즘으로 현존하는 웹 인증과 비밀성 서비스들과 결합하여 사용할 수 있다.

이러한 방법들은 사용자명/패스워드, SSL (Secure Socket Library), SHTTP (Secure HTTP)와 PCT (Private Communication Technology Protocol) 등을 사용할 수 있다.

사용자 확인정보는 응용 서버에 의해 RBAC 응용에 전달된다. 응용서버 관리자에 의해 설정되며, 기밀 자료 전송의 준비와 사용자 확인정보의 인증은 응용 서버의 책임이다.

### 2. 최종 사용자의 사용방법

응용 웹 서버의 최종사용자의 접근은 기본적으로 요청한 URL의 접근이 RBAC 응용에 의해 제어될 수 없을 때는 일반적인 접근과 동일하다. 그러나 RBAC 적용으로 강화된 응용 웹 서버의 접근은 RBAC 세션이 설정되어야 사용자 접근이 가능하다.

RBAC 세션의 설정 방법으로 최종 사용자의 접근 허용은 하나의 현행의 행위 임무의 집합(ARS : Action Role Set)으로 할당한다. ARS는 허가된 기능들에 의해 결정된다. 최종 사용자는 제어된 URL들이 RBAC 응용 상에서 수행될 수 있다. 최종 사용자의 접근은 새로운 ARS가 설정될 때까지 허용된다. 따라서 임의로 RBAC 세션 변경할 수 있어 유연성이 보장된다.

사용자는 DSD 관계를 갖고 있는 임무들로 할당된다. 세션 관리자는 할당된 임무의 집합의 부분을 선택하기 위하여 사용자들을 활성화된다면 사용자들은 세션 내에서 사용할 수 있다.

사용자들은 임무선택 요구와 임의의 DSD 관계를 위배하지 않는 부분집합의 리스트를 제공한다. 임무선택을 최소화하기 위하여 리스내의 부분집합들은 사용자들에게 할당된 임무들의 사용 가능한 부분집합들로 DSD 관계를 위배하지 않는 상위 부분집합들로부터 얻을 수 있다.

한번의 임무선택은 RBAC 세션에서 이루어지는데, RBAC 세션은 ARS에 위치한 모든 인증된 임무들 즉, 임무들에 상속되는 할당된 임무들에 의해 설정 가능하다.

사용자에게 할당된 임무들이 DSD 관계가 아니라면 RBAC 세션은 ARS에 있는 모든 인증된 임무들을 자동적으로 설정한다.

일반적으로 하나의 RBAC 세션은 하나의 인증된 최종 사용자를 요구한다. 최종 사용자로부터 인증이 삭제되었다면 접근은 거부된다. 그러나 최종 사용자의 인증과 RBAC 세션의 확립은 완전히 운영에서 분리된다. 따라서 RBAC 응용은 임의 인증 기술이라도 사용될 수 있다.

본 논문에서 제안한 그림 2의 RBAC 모형은 상속관계와 제안된 2개의 특징이 대립적인 임무들 사이의 DSD 관계는 응용 개발에 영향을 받지 않는다.

예를 들면 임의 임무 R은 임무 R'을 상속하고, 임의 임무 R과 임무 R'은 DSD 관계를 공유한다. 이러한 예에서 사용자가 임무 R이 RBAC 세션 내에 설정하고자 할 때, ARS 내에 이미 할당된 임무들의 존재에 대한 의문이 발생할 것이다. 두 개의 관계들은 ARS 내에서 동시에 수용될 수 없다. 따라서 이러한 의문의 대답은 정의된 예의 임무 관계가 중복 허가되지 않음으로 해소될 수 있다. 그러므로 RBAC 응용을 구현하면서 모든 경우와 일관성을 보장해야 한다.

임무들 사이의 상속(계승), SSD, DSD 관계들은 항상 일관성을 갖도록 하여, 운영시에 다른 관계보다 우선하는 관계 즉, RBAC 응용들의 동작내에 존재하는 예외는 없어야 한다. 따라서 현재의 수행시에 숨겨진 규칙은 없어 RBAC 응용 관리자를 보증

한다.

모든 수행 동작시간 내에 RBAC 응용 DB는 일관성을 갖고 항상 RBAC DB 내에 정의된 모든 임무의 관계에 의하여 동작할 수 있다.

RBAC 응용에 의해 구현되는 응용 웹서버에서 허가된 동작은 HTTP 프로토콜 정의에 정의된 방법에 의해 수행한다. 이 방법들은 GET, HEAD, PUT, POST 등이 있다. RBAC 응용은 HTTP 방법의 URL를 수행하기 위한 임무 속에서 사용자 대행 능력으로 제어할 수 있다.

## Ⅳ. 결 론

서버상에 존재하는 각종 응용들의 수행 및 접근, 영업 활동들을 위한 응용들에 대한 정보의 사용자 접근은 필요에 따라 적시에 통제되어야 한다. 따라서 RBAC 응용의 목적은 이러한 접근제어 서비스들을 준비하기 위한 것이다. 이것은 새롭고 더욱 정교한 응용 즉, 현존하거나 발생가능한 위험 상황이 존재하지 않도록 하고, 다른 자원과 정보의 접근 허락을 위하여 응용 서버의 사용을 가능하게 된다. 설계 제안된 RBAC의 장점은 이것을 지원할 수 있는 관리상의 능력을 가지고 있다는 것이다. 인증 자료의 관리는 비용의 재 발생과 처리 부담을 줄일 수 있다.

설계 제안된 RBAC 하에서 사용자들은 임무에 근거한 책임과 권한을 갖는 자격을 얻을 수 있음으로서 작업 할당에 따라 새로운 임무에 대한 사용자의 자격을 쉽게 재 설정 또는 취소할 수 있다. RBAC에서 사용자들의 개별적인 근거에 따른 자격 규정으로는 수행을 허락하지 않으며, 모든 동작들은 임무에 관련된다. 새로운 동작에 관련된 임무는 조직상의 기능의 변경에 따라 임무를 설정 또는 삭제할 수도 있다. 이러한 기본적인 개념은 임무들을 모든 사용자를 위한 개별적인 근거로 직접적으로 특권의 갱신없이 갱신할 수 있다는 특권의 관리와 이해를 단순화할 수 있다는 장점을 가지고 있다.

제안된 RBAC는 서버에 구현하기 위해서 서버 원시 코드의 수정 없이 현재의 시스템에 통합하여 구현할 수 있어 실질적으로 모든 서버들에 이식할 수 있다는 장점을 가지고 있다.

## 참 고 문 헌


[1] R. Sandhu, E. J. Coyne, and C. E. Youman, editors. Pro. of the First ACM Workshop on Role Based Access Control. ACM, 1996.

[2] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role based access control models. IEEE Computer, Vol.29, No.2, 1996.

[3] D. Ferraiolo, J. Cugini, and D. R. Kuhn. Role based access control: Features and motivations. In Annual Computer Security Applications Conference. IEEE Computer Society Press, 1995.

[4] S. H. von Solms and I. van der Merve. The management of computer security profiles using a role oriented approach. Computers and Security, Vol. 13, No. 8, 1994.

[5] D. Ferraiolo and D. R. Kuhn. Role based access control. In 15th National Computer Security Conference. NIST/NSA, 1992.

[6] D. F. Sterne. A TCP subset for integrity and role-based access control. In 15th National Computer Security Conference. NIST/NSA, 1992.



김 성 열(Seong Ryeol, Kim) 정회원

1954년 6월 9일생

1982년 2월 : 숭실대학교 전자계산학과(공학사)

1987년 2월 : 숭실대학교 대학원 전자계산학과(공학석사)

1992년 2월 : 숭실대학교 대학원 전자계산학과(공학박사)

1982년 1월~1984년 2월 : 한국전력공사 전자계산소 근무

1984년 3월~1990년 8월 : 오산대학 전자계산과 조교수

1990년 9월~1999년 현재 : 청주대학교 컴퓨터정보공학과 부교수

1997년 1월~1998년 1월 : 호주 QUT ISRC 객원교수

<주관심 분야> 컴퓨터 정보통신, 컴퓨터 보안, 분산 객체시스템