韓國컴퓨터情報學會 論文誌
第15卷 第12號, 2010. 12.

2010-15-12-4-11

직무기반 접근제어를 사용하는 웹기반 응용 시스템의 시스템 아키텍처 설계

이 호*

A System Architecture Design for Web-Based Application Systems using Role-Based Access Control

Ho Lee*

요 약

현재 널리 사용되고 있는 웹 서버 기반 보안 체계에서는 아직도 사용자 기반 접근제어를 많이 사용하고 있다. 웹과 직무기반 접근제어의 통합은 인터넷 기반 시스템의 보안을 강화할 수 있는 효과적인 수단이 될 수 있다. 직무기반 접근제어를 웹을 기반으로 하는 응용 시스템에서 사용하기 위해서는 이를 위한 시스템 아키텍처가 마련되어야 한다. 본 논문에서는 직무기반 접근제어를 웹기반 응용 시스템에 접목하기 위한 시스템 아키텍처를 제안한다. 제시한 시스템 아키텍처는 크게는 시스템 구성 및 시스템 동작으로 구성되는데, 먼저 직무기반 접근제어 엔진이 사용하는 인증서에 대해 기술하고, 사용풀 방식의 시스템 아키텍처를 제안하며, 직무 서버를 중심으로 하는 전체 시스템의 구성에 대해 언급한다. 그리고 웹기반 응용 시스템에서 어떻게 직무기반 접근제어가 수행되는지에 관한 시스템 동작 방식을 제시한다. 마지막으로 제안한 시스템 아키텍처에 대한 타당성 입증을 위해서 시스템 아키텍처에 대해 분석을 한다.

Abstract

Among web-based systems being widely used now, there are so many systems which are still using an user-level access control method. By successfully applying role-based access control(RBAC) to web-based application systems, we can expect to have an effective means with reinforced security for Internet-based systems. In order to apply RBAC to web-based application systems, we should come up with a system architecture for it. I proposed a system architecture which is needed to apply RBAC to web-based application systems. The proposed system architecture is largely composed of system composition and system functioning. For details, firstly, a certificate used by RBAC is specified. Secondly, a system architecture using a user-pull method is proposed and overall system components are mentioned with a role server being centered. Then,

• 제1저자 : 이호
• 투고일 : 2010. 12. 01, 심사일 : 2010. 12. 10, 게재확정일 : 2010. 12. 12.
* 한국재활복지대학 컴퓨터정보보안과 교수

I showed how the system architecture can work to carry out RBAC on web-based application systems. Lastly, the analyses on the proposed system architecture are described for the purpose of proving its feasibility.

▸ Keyword : RBAC(Role-Based Access Control), 직무기반접근제어(RBAC), 접근제어(Access Control)

Ⅰ. 서 론

웹은 인터넷 상에 웹 사이트를 구축하여 관공서의 민원 서비스, 대학의 학사 행정 서비스 및 기업의 전자 상거래 서비스 등을 가능하게 하는 중요한 기반 기술이다. 웹, 운영체제, 데이터베이스 시스템의 통합이 지속적으로 증가하고 있는 것으로 보아서 대학 관공서 및 기업 차원의 컴퓨팅을 위해서 앞으로도 다양한 기술 및 구성 요소를 통합하여 웹 환경에서 동작하는 다양한 용도의 응용 시스템을 구축하는 경향이 지속될 것이라는 예측이 가능하다. 그런데, 현재 널리 사용되고 있는 웹 서버 기반의 보안 체계에서는 아직도 사용자 수준의 인증이나 사용자 기반의 접근제어를 많이 사용하고 있다.

웹과 직무기반 접근제어의 성공적인 통합은 사회에서 필요로 하는 각종 대규모 인터넷 기반 시스템의 보안을 강화할 수 있는 효과적인 수단을 제공할 수 있다. 직무기반 접근제어를 웹을 기반으로 하는 응용 시스템에서 사용하기 위해서는 이를 위한 시스템 아키텍처 구성 방안이 마련되어야 한다.

이 논문에서는 직무기반 접근제어를 대규모 웹기반 응용 시스템에 적용하기 위한 시스템 아키텍처 설계 방안을 제안한다.

Ⅱ. 관련 연구

1. 직무기반 접근제어

직무기반 접근제어(RBAC)에서의 직무란 접근제어의 근간을 이루는 의미론적 구성 개념이다. 직무기반 접근제어에서 시스템 관리자는 조직에서 수행하는 업무에 기초하여 직무들을 정의하고, 직무에 접근허가를 부여하며, 업무 책임 및 자격을 고려하여 사용자들을 특정 직무에 할당 한다[1]. 직무에 따라 자원을 액세스 할 수 있는 사람 및 자원을 액세스할 수 있는 정도가 정해진다.

직무기반 접근제어가 아니고는 어떤 접근허가를 어느 사용자에게 부여해야 할지 결정하기가 곤란하다[2]. [그림 1]은 직무기반 접근제어의 참조 모델간의 관계를 보여주며 [그림 2]는 참조 모델들의 특성을 나타낸다. RBAC0는 기본 모델로서 직무기반 접근제어를 위한 최소한의 요구 조건을 가진다. 진화된 모델인 RBAC1과 RBAC2는 RBAC0를 포함하면서 RBAC1은 직무계층이 부가되어 있고 RBAC2는 제약조건이 부가되어 있다. RBAC2는 RBAC1을 포함한다고 말할 수 도 있다. 통합 모델인 RBAC3은 RBAC0, RBAC1, RBAC2의 특성들을 모두 포함하는 모델이다[3].

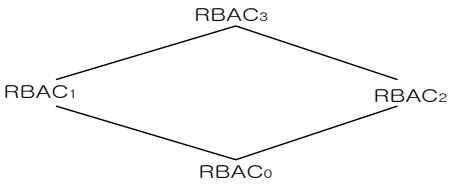

그림 1. 직무기반 접근제어 참조 모델간의 관계
Figure 1. Relationship between Role-Based Access Control Models

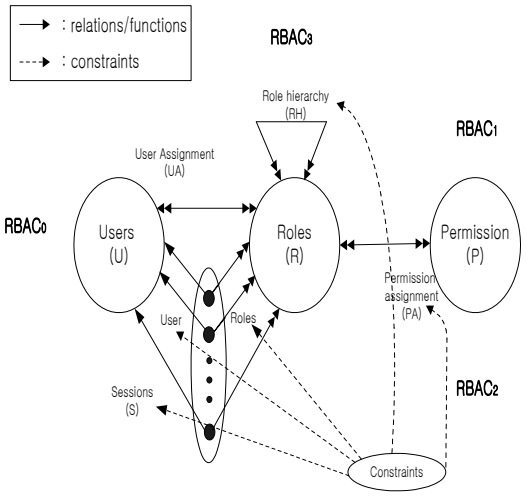

그림 2. 직무기반 접근제어 참조 모델
Figure 2. Role-Based Access Control Reference Mo

2. 공개키 인증서 X.509 v3

공개키 암호화에서 공개키와 개인키는 인증기관(CA)에 의해 같은 알고리즘을(RSA) 사용하여 동시에 만들어진다. 개인키는 요청자에게만 주어지며, 공개키는 모든 사람이 접근할 수 있는 디렉터리에 디지털 인증서의 일부로서 공개된다. 개인키는 절대로 다른 사람과 공유되거나 인터넷을 통해 전송되지 않는다. 사용자는 누군가가 공개 디렉터리에서 찾은 자신의 공개키를 이용해 암호화한 텍스트를 해독하기 위해 개인키를 사용한다.

ITU와 ISO가 1988년에 표준안을 공표하고 IETF에 의해 채택된 X.509는 현재 공개키 인증서의 데이터 포맷으로 가장 널리 사용되고 있는데 지정된 인증기관의 이용을 전제로 한다. X.509는 공개키와 주체(인증서 피발급자, 사람이나 엔터티)를 바인딩하기 위하여 사용되어 왔는데 공개키와 인증서 피발급자 간의 바인딩의 진실성 보장을 위해서 인증기관이 인증서에 전자 서명하도록 되어있다. X.509 v3은 확장 필드를 지원하는데 특정 커뮤니티에서만 고유하게 사용하는 정보 전달을 목적으로 이 필드를 새로이 정의하여 사용할 수 있다. 각 확장 필드는 플래그에 critical이나 non-critical로 반드시 지정하도록 되어 있는데 인증서를 사용하는 시스템에서는 critical 확장 필드를 해석할 수 없는 경우에는 인증서를 무조건 거부하도록 되어있고 non-critical 확장 필드를 해석할 수 없는 경우에는 무시할 수 있도록 되어 있다.

3. SSL 프로토콜

SSL 프로토콜은 Netscape Navigator 브라우저와 함께 소개된 후에 웹에서 지배적으로 사용하는 프로토콜로 자리 잡고 있다. 이 프로토콜은 트랜스포트 계층에서 동작하기 때문에 TCP를 사용하는 어떤 프로그램이라도 SSL 연결을 설정하는 것이 가능하다. 이 SSL 프로토콜은 웹 서버와 브라우저 간의 암호화된 연결을 확립할 수 있는 안전한 수단을 제공한다. 또한 SSL은 X.509 인증서를 이용하여 웹 서버와 브라우저 간의 인증 서비스도 지원한다. 서버 인증서는 클라이언트가 특정한 웹 서버를 인증할 수 있는 수단인데, 웹 브라우저는 서버의 공개키를 사용하여 웹 서버와 안전한 TCP 연결을 교섭할 수 있다. 웹 서버는 선택적으로 자신의 클라이언트의 인증서 내용을 확인함으로써 사용자 인증을 할 수 도 있다.

III. 시스템 설계

1. 설계 관련 기본 사항

1.1 설계 목표

다음과 같은 설계 목표를 가지고 시스템을 설계한다.

(1) 시스템은 사용 가능한 기존의 COTS(commercial off-the-shelf) 기술을 통합하여 구성한다.
(2) X.509 v3 공개키 인증서를 이용한다.
(3) 인증서를 직무 서버로부터 받아 웹기반 응용 시스템에 제출하는 사용자풀 방식을 적용한다.
(4) 기존의 웹기반 응용 시스템의 변경 정도를 최소화하고 웹 브라우저에는 전혀 영향이 없도록 한다.
(5) RBAC 엔진은 웹기반 응용 시스템과는 다른 시스템에서도 독립적으로 동작할 수 있게 한다.
(6) RBAC 모델을 토대로 실제 구현 모델인 RBACi(RBAC for Implementation)를 정의하고 이 모델을 적용한 시스템을 설계한다.
(7) 웹기반 응용 시스템의 보안을 위해서 사용자 인증, 데이터 전송을 위한 네트워크 보안, 접근제어 등의 보안 서비스를 시스템 구성에 반영한다.
(8) 클라이언트-서버 인터랙션에는 HTTP 및 SSL 프로토콜을 서버-서버 인터랙션에는 CORBA의 객체 지향 프로토콜인 IIOP를 사용한다.
(9) 시스템 구성 요소인 웹 서버와 RBAC 간의 인터페이스를 명확히 한다.

1.2 접근제어 메커니즘

직무기반 접근제어 메커니즘의 일처리 과정은 다음과 같다[3].

(1) 관리 단계(Administration Phase)

관리 단계는 사용자 및 객체의 보안 속성을 생성하거나 유지하는 단계인데 관리자만이 사용할 수 있는 프로그램 툴을 사용하여 주로 시스템 초기 설치 시에 행해진다.

(2) 세션 단계(Session Phase)

세션 단계는 세션 확립, 세션 특성 변경 및 세션 종료로 구성된다. 세션이란 주체라 불리는 프로세스의 집합인데, 여기서 주체는 특정 사용자를 대신하여 행위를 하게 된다. 세션 확립은 사용자 인증, 단●복수의 주체 생성, 사용자 보안 속성을 각 주체에 바인딩하기로 구성된다.

(3) 시행 단계(Enforcement Phase)

시행 단계는 접근허가를 받기 위하여 주체의 보안 속성을 객체의 보안 속성들과 비교하는 단계이다. 이 단계는 주체가 객체 접근을 시도할 때마다 발생하게 되며, 세 단계 중에서

가장 빈번히 발생한다.

관리 단계에서는 주체가 객체를 접근할 수 있는 규칙을 규정하고 세션 단계에서는 사용자 인증 후에 주체를 생성하게 되며 시행 단계에서는 주체가 사용자(직무) 이름으로 객체를 접근한다.

1.3 세션과 사용자 정보

관리 단계에서 정의된 사용자 보안 속성 값은 세션 단계에서 주체의 보안 속성 결정의 기준이 되므로 항상 최신 값을 유지해야 한다. 그러기 위해서 집중(centralized) 사용자 정보 방식을 적용하는데 이는 보안 정보가 포함된 사용자 정보를 단일 서버 상에 유지하고 세션 프로세싱 동안에 그 서버만을 액세스하도록 한다. 이 방법은 세션 프로세싱 동안에 항상 최신의 사용자 보안 정보를 보장 받을 수 있으나 사용자 속성 정보가 보관된 단일 서버의 장애 시에는 세션 프로세싱이 불가능해 진다. 사용자의 보안 정보가 모든 서버에 분산 유지되는 분산 사용자 정보 방식은 정보의 일관성(consistency) 유지에 문제가 있을 수 있어서 사용하지 않았다[4].

2. 인증서 설계

2.1 인증서 형식 정의

사용자의 속성 정보가 보안 서비스를 이용해 웹상에서 안전하게 전달될 수 있어야만 속성 정보를 기반으로 인증이나 접근제어를 할 수 있다.

X.509 v3의 확장 필드를 이용하여, SSL 등과 같은 기존의 표준안과 호환성을 유지하면서도 웹상에서 안전한 속성정보 전달이 가능할 수 있도록 하기 위해서 주체 식별자와 접근제어 정보를 같이 포함하는 새로운 인증서(BC) 형식을 정의한다[5]. 인증서 구성 항목 중에서 확장 필드를 이용하여 접근제어에 필요한 정보를 전달할 수 있도록 확장 필드의 형식 및 내용을 <표 1>에서와 같이 정한다.

표 1. X.509 v3 확장 필드의 형식 및 내용
Table 1. Format and Contents of X.509 v3 Extensions

형 식	내 용	Critical
RoleIdentifier ::= SEQUENCE SIZE(1,..MAX) OF SEQUENCE{ RoleID INTEGER RoleSpecifier BIT STRING OPTIONAL}	- 사용자에게 부여된 직무 타이틀 - 복수 개의 직무 지정 가능	No
SecurityLevel ::=SEQUENCE SIZE(1,..MAX) OF SEQUENCE{ SecurityLevel INTEGER SecuritySpecifier BIT STRING OPTIONAL}	- 직무에 부여된 보안성 등급 - 복수 개의 보안성 등급 가능	No
IntegrityLevel ::=SEQUENCE SIZE(1,..MAX) OF SEQUENCE{ IntegrityLevel INTEGER IntegritySpecifier BIT STRING OPTIONAL}	- 직무에 부여된 무결성 등급 - 복수 개의 무결성 등급 가능	No

2.2 인증서 발행

[그림 3]에서와 같은 절차를 통해 인증서를 발급하는 방식을 사용한다[6]. 직무 서버에서는 사용자 인증 후에 해당 클라이언트의 직무와 접근제어 속성 정보를 포함하는 클라이언트 인증서(공개키를 포함)를 발급한다. [그림 3]은 이를 위한 상세 과정을 보여준다[7].

2.3 웹 서버의 인증서 정책

웹기반 응용 시스템도 같은 인증기관이 발급하는 인증서를 사용함으로 클라이언트가 직무 서버에서 발급 받은 인증서를 도메인 상의 어떤 웹 서버에서도 사용 가능하다.

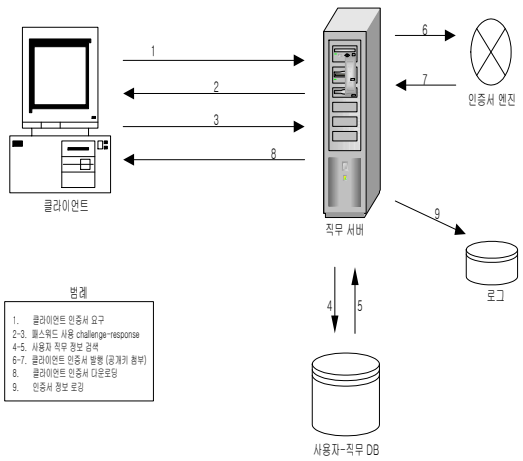

그림 3. 인증서 발행 절차
Figure 3. Procedure for Issuing Certificates

3. 시스템 아키텍처

3.1 사용자 정보 획득 방법

사용자의 속성 정보를 웹을 이용하여 획득할 수 있는 방법에는 두 가지 방식이 있는데 사용자풀 방식(UPS)과 서버풀 방식(SPS)이다[8]. UPS에서는 사용자가 직무 서버로부터 속성 정보를 받아서 접근허가를 얻기 위하여 웹 서버에 보낸다. 반면에, SPS에서는 클라이언트는 사용자 인증을 위한 정보를 웹 서버에 제시하면 웹 서버가 사용자 속성 정보를 직무 서버로부터 획득한 후에 사용자를 위한 접근허가를 결정한다. SPS는 사용자 입장에서는 편리한 반면에 웹 서버 입장에서는 UPS보다 부담(load)이 있다고 할 수 있다.

이 논문에서는 UPS를 사용하여 [그림 4]에서처럼 클라이언트가 직무 서버로부터 사용자 속성 정보가 포함된 인증서를 받아서 웹 서버로 보낸다.

3.2 시스템 구성

시스템은 [그림 4]와 같이 RBAC 관리 툴, 직무 서버, 웹 기반 응용 시스템의 세 가지 구성 요소로 구성된다.

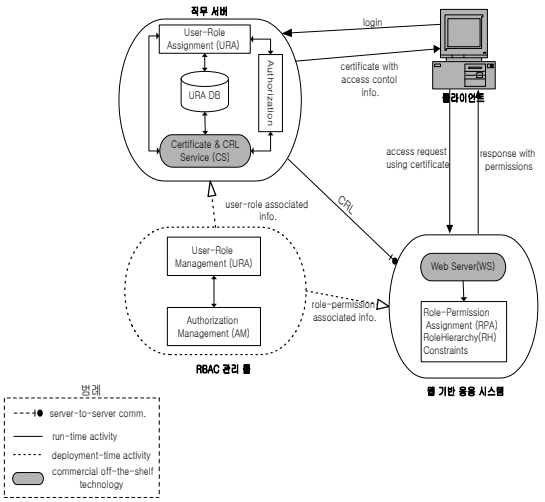

그림 4. 사용자-풀 방식의 시스템 아키텍처
Figure 4. System Architecture using User-Pull Method

3.2.1 RBAC 관리 툴(RMT)

RMT는 다음의 일을 수행하는 소프트웨어 툴로서, RBAC 시스템 관리자가 주로 시스템의 설치 시에 이용하여 RBAC의 운영에 필요한 각종 정보를 관리한다. RMT로 만든 정보 중에서 사용자-직무관련 정보는 직무 서버로 직무-접근허가 관련 정보는 웹 서버로 보내진다.

(1) 사용자-직무 관리(URM)

직무의 생성●변경●삭제 및 사용자를 단일 또는 복수의 직무에 할당●해지 등으로 인하여 발생하는 사용자-직무 관련 정보를 처리한다.

(2) 접근허가 관리(AM)

직무 서버에 저장되는 직무나 웹 서버에 저장되는 객체에 부여된 보안 속성 정보의 생성●변경●삭제를 처리한다.

3.2.2 직무 서버

직무 서버는 RBAC이 적용되는 도메인 상에서 클라이언트를 위하여 다음의 일을 처리하는 전용(dedicated) 서버이다.

(1) 사용자 인증

클라이언트가 로그인을 요청하면 클라이언트가 RBAC이 적용되는 도메인의 웹 서버를 접근할 수 있는 자격이 있는지를 인증한다.

(2) 사용자-직무 할당(URA)

인증된 사용자의 직무를 URA DB에서 검색하여 해당 직무를 사용자에게 할당한다[9].

(3) 인증서 서비스(CS)

클라이언트가 웹 서버 접근을 요구할 때 필요한 사용자의 직무 및 보안 속성 정보를 포함하고 있는 인증서를 발행한다. 또한 주기적으로 인증서 해지 목록(CRL)을 게시하여 웹 서버들이 CRL을 입수하여 클라이언트 인증 절차 수행 시에 해지된 인증서를 사용치 못하도록 하는 보안 서비스도 제공한다.

(4) URA DB

사용자가 부여 받은 직무에 관한 데이터베이스를 유지한다.

3.2.3 웹기반 응용 시스템

웹기반 응용 시스템은 웹을 기반으로 하는 응용 프로그램이 실행되는 서버로서 응용 프로그램이 실행되기 위해서 웹 서버가 설치된다. 클라이언트가 웹 서버의 자원(객체)에 접근을 요구하면 RPA, RH 및 Constraints가 같이 접근허가 결정에 관여한다.

(1) 직무-접근허가 할당(RPA)

사용자의 객체에 대한 접근 요구에 대해서 사용자의 직무에 따른 보안 속성과 웹 서버 자원(객체)의 보안 속성을 비교하여 접근허가를 결정한다[10].

(2) 직무 계층(RH) 및 Constraints

웹 서버마다 독립적으로 정한 원칙에 따라서 접근허가 결정에 적용되는 직무 계층 및 RBAC 구성 요소에 대한 제약조건을 결정한다.

3.3 시스템 구성에 사용된 COTS

웹 클라이언트로는 마이크로소프트사의 Explorer, 인증 서버로는 마이크로소프트사의 Certificate Service, 웹 서버로는

마이크로소프트사의 Internet Information Server를 시스템 구성에 포함한다. 이 논문에서 제안한 RBACi 엔진을 중심으로 이러한 COTS 구성 요소를 통합하여 시스템을 구성한다.

4. 시스템 동작

4.1 안전한 정보 전달 메커니즘

[그림 5]는 사용자의 자원 접근 요구를 RBAC 시스템이 처리하는 절차를 보여주는 개념도이다. 일의 수행 절차는 다음과 같다.

① 클라이언트가 인증에 의하여 합법적인 사용자임을 확인
② 사용자는 도메인 내에서 직무 서버로부터 자신에게 할당된 직무 정보를 획득
③ 사용자가 자신의 직무를 사용하여 웹 서버에 접근을 요구하면 웹 서버는 사용자의 직무 정보를 이용하여 웹 서버와의 트랜잭션을 허가

이 과정에서 문제가 되는 것은 클라이언트가 제시하는 직무 정보를 웹 서버가 어떻게 신뢰할 수 있느냐는 것이다. 즉, 악의적인 사용자가 위조된 직무 정보를 이용해 웹 서버로의 접근을 시도했을 때 이를 어떻게 방지할 수 있느냐 하는 문제이다. 이 문제점을 해결하기 위한 방안으로 3장 2절에서 정의한 인증서를 이용한다. 즉 사용자의 ID, 직무, 접근제어 정보를 포함하는 인증서를 사용자풀 방식을 이용하여 전달함으로써 직무 서버와 웹 서버들 간에 속성 정보 전달을 위한 별도의 채널이 없이도 SSL 같은 기존의 프로토콜과의 호환성을 유지하면서 안전하게 직무 정보의 전달을 할 수 있다.

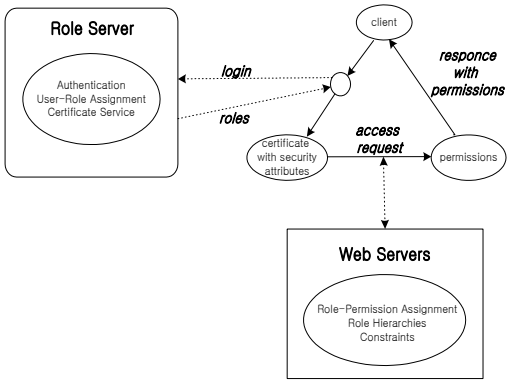

그림 5. 안전한 직무 정보 전달 메커니즘
Figure 5. Secure Mechanism for Role Info. Transfer

4.2 직무기반 접근제어의 동작 메커니즘

[그림 6]은 인증서 발행, 웹 서버에서의 처리 과정 및 RBACi 엔진의 일 처리 과정을 보여준다. 클라이언트가 RBAC이 적용되는 도메인의 웹 서버와 트랜잭션을 하고자 하는 경우에는 다음의 과정을 거친다.

① 세션의 시작 단계에서 클라이언트는 직무 서버와의 연결을 설정
② 직무 서버가 사용자를 인증
③ 직무 서버는 URA 데이터베이스에서 사용자의 직무를 검색
④ 사용자의 ID와 속성들(직무 포함)을 확장 필드에 포함시켜서 인증서 생성
⑤ 인증서는 클라이언트로 전달되어 클라이언트 시스템에 저장

따라서 사용자는 인증서의 유효 기간이 만료되기 전까지는 직무 서버로부터 인증서를 재발급 받지 않아도 된다. 이것은 사용자가 인증서 유효 기간 동안에는 인증서에 포함된 직무 정보를 언제나 사용할 수 있다는 것을 의미한다. 사용자 ID, 직무 및 접근제어 정보는 동일한 인증기관에 의해 전자 서명되는 방법을 사용한다. 사용자는 자신의 클라이언트 시스템에 복수개의 인증서를 보관하고 있을 수 도 있다. 클라이언트가 특정한 웹 서버를 그 서버의 URL을 사용하여 액세스하는 순간에 클라이언트의 웹 브라우저와 웹 서버는 SSL 프로토콜을 이용하여 서로를 인증하게 된다. 즉, 클라이언트의 브라우저가 웹 서버로부터 X.509 인증서를 받아 내용을 확인한 후 일치하는 인증서를 자신의 시스템에 저장해둔 인증서 중에서 선택하여 이를 웹 서버로 보낸다. 웹 서버는 클라이언트로부터 수신한 인증서를 검사하여 사용자 인증을 하게 되는데 이때 웹 서버의 로컬 캐시에 저장되어 있는 인증서 해지 목록(CRL)을 먼저 확인하여 수신한 인증서가 이미 해지된 것인지를 확인하고 이상이 없는 경우에만 인증서의 유효 기간과 전자 서명된 정보의 이상 여부를 확인할 수 있다. 이러한 일련의 과정을 정상적으로 통과하면 웹 서버는 인증서가 포함하고 있는 직무 및 접근제어 정보를 신뢰하여 이 정보를 응용 시스템의 RBACi 엔진에게 전달한다. 그러면 RBACi 엔진은 직무기반 접근제어 메커니즘을 적용하여 주체의 객체에 대한 접근허가를 결정한다.

4.3 인증서 해지 목록 처리 메커니즘

웹 서버가 사용자가 제시한 인증서를 받았을 때 그 인증서의 해지 여부를 확인하기 위하여 인증서 해지 목록(CRL) 배포 시기에 맞춰서 인증기관으로부터 복사한 최신의 인증서 해지 목록을 항상 자신의 로컬 캐시에 보관한다. 직무 서버의 구성 요소인 'Issue BS & CRL'은 CRL을 인증서 배포 지

점에 게시하고 웹 서버는 인증서 해지 목록 입수를 위해서 'Issue BS & CRL' 기능과 주기적으로 트랜잭션 한다.

4.4 웹 서버에서의 직무기반 접근제어 방법

웹 서버에서의 접근허가는 웹 사이트를 방문하는 모든 사용자들을 대상으로 하여 특정한 페이지를 볼 수 있는지, 스크립트를 실행할 수 있는지, 사이트에 정보를 업로딩할 수 있는지 등의 접근권한 결정을 필요로 한다[11].

사용자가 웹 서버에 접근을 시도할 때 웹 서버는 사용자의 접근허가 결정을 하기 전에 인증서의 보안 속성 정보를 처리하는 프로세스들을 먼저 실행하고 나서 RBACi 엔진이 접근제어를 수행하도록 한다.

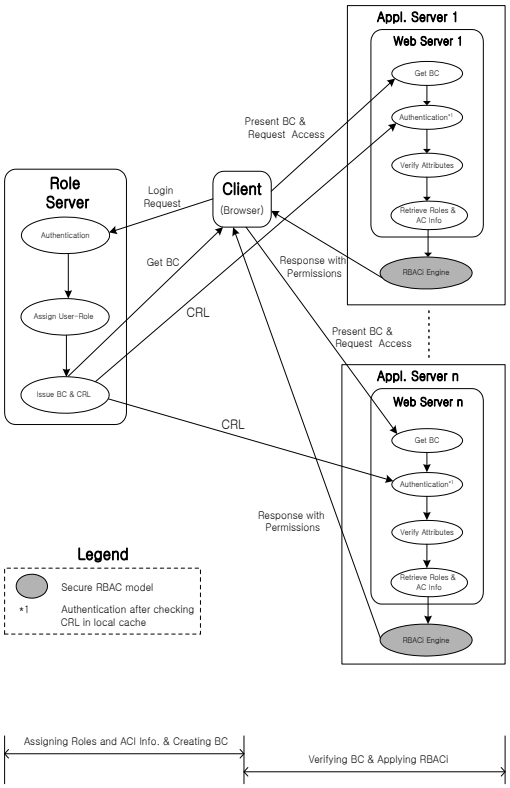

그림 6. RBAC의 웹 상에서의 동작 메커니즘
Figure 6. Operating Mechanism of RBAC on Web

마이크로소프트사의 운영체제인 Windows Server 200x에서 지원하는 파일 시스템인 NTFS에는 서버에 계정을 가지고 있는 주체인 특정 사용자 또는 사용자 그룹이 객체인 파일 및 폴더들에 대해서 불법적인 접근을 할 수 없도록 제어할 수 있는 기능이 있다. 웹 서버 소프트웨어인 IIS를 사용하여 NTFS가 제공하는 파일 시스템 서비스들과 웹 서버의 보안 특성을 적절히 구성함으로써 웹 상에서의 불법적인 사용자 접근으로부터 웹 서버의 정보(객체)를 보호할 수 있다.

4.4.1 직무와 직무 계정의 매핑

웹 서버는 클라이언트의 인증서에서 얻은 직무 정보를 접근제어 메커니즘 적용에 사용하기 때문에 사용자 ID 정보는 사용하지 않는다. [그림 7]은 직무와 직무 계정의 매핑에 대한 예를 보여준다. 응용 시스템(웹 서버 및 RBACi 엔진을 포함)에 [그림 7]의 각 직무에 해당하는 직무 계정(DIR, PL1, PL2, PE1)을 생성하고 각 직무를 해당 직무 계정에 매핑한다. 웹 서버의 인증서 매핑 기능을 이용하여[12] [그림 7]의 직무 계층의 각 직무를 해당 직무 계정에 매핑한다. 즉, 인증서 확장 필드의 직무 정보를 사용하여 웹 서버 시스템의 직무 계정에 매핑할 수 있다[13]. 사용자(주체)인 Director가 SSL 프로토콜을 사용하여 자신의 클라이언트 인증서(직무 DIR을 포함)를 웹 서버로 보내서 서버의 인증을 통과하면 Director는 서버 시스템에 생성된 직무 계정 DIR로 매핑된다. 웹 서버 상에서 각 직무 계정이 부여 받는 특정한 객체에 대한 접근허가는 웹 서버의 RBACi 엔진의 접근제어 결정에 따른다.

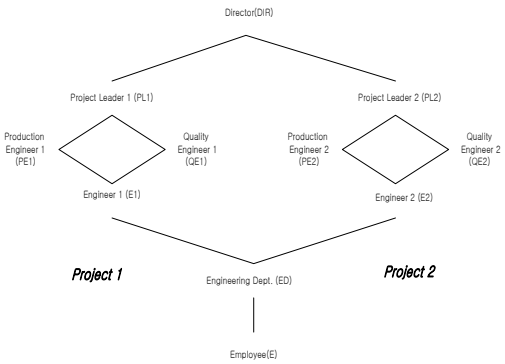

그림 7. 직무와 직무 계정의 매핑
Figure 7. Mapping between Roles and Role Accounts

4.4.2 웹 서버의 직무 계층 지원

여기서는 웹 서버의 RBACi이 직무 계층과 관련하여 직무에서 객체로의 접근제어를 수행하는 방법을 제시한다. [그림 8]은 [그림 7]의 직무 계층을 지원하기 위하여 웹 서버에서 RBACi의 접근제어 메커니즘을 어떻게 적용할 수 있는지를 보여준다. 웹 서버에 [그림 7]의 직무 계층상의 각 직무를 위한 디렉터리를 생성하는데 이는 주체의 접근 대상인 객체가 된다. 그리고 각 직무 계정(주체)이 직무 계층을 위해서 구성

한 디렉터리(객체)에 대해 특정한 접근허가를 받도록 응용 시스템(웹 서버)의 객체 보안 속성을 결정한다[5]. 이렇게 함으로써 RBACi 엔진이 주체와 객체의 보안 속성 정보를 서로 비교하여 최종적인 접근허가를 결정할 수 있다. 예를 들어, 어떤 사용자가 ProjectLeader1 직무를 부여 받는다고 가정하면, 그 사용자는 PL1 계정의 디렉터리에 접근허가를 부여받는 동시에 PE1, QE1, E1, ED, E 계정의 디렉터리에 대한 접근허가도 획득할 수 있도록 각 직무 계정에 할당된 디렉터리의 보안 속성 정보를 결정해줘야 한다[14].

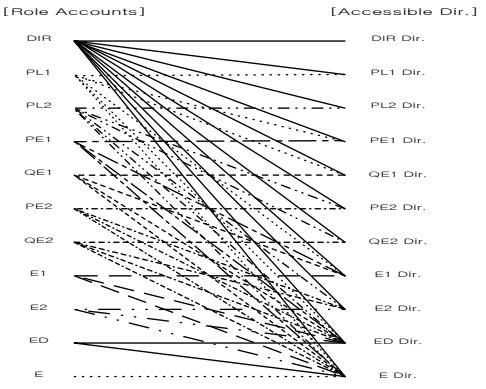

그림 8. 직무 계정에서 객체로의 접근허가 할당
Figure 8. Assigning Role Accounts to Objects

IV. 비교 분석

<표 2>는 시스템 아키텍처를 비교 분석한 표인데 이 표에서 열두 가지 항목에 대해 비교한 결과를 보면 제안한 시스템 아키텍처가 기존 시스템 아키텍처에 비해 우수함을 알 수 있다. 다음은 <표 2>의 비교 항목에 대한 자세한 설명이다.

①②기존의 사용자기반 접근제어를 위한 아키텍처가 아니고, 웹 환경에서 보안이 보장되는 직무기반 접근제어를 위한 아키텍처이다.

③ 사용자풀 방식 아키텍처를 사용함으로써 사용자 보안 속성 정보를 획득하기 위한 별도의 채널 없이 직무 서버만을 이용하여 사용자의 보안 속성 정보를 획득할 수 있다.

④⑤전담 서비스를 하는 직무 서버를 시스템 아키텍처에 도입함으로써 사용자 인증, 보안 정보 검색, 인증서 발급 기능 등을 웹 서버로부터 분리시켜 웹 서버는 접근제어를 이용한 웹 서비스에만 전념할 수 있게 하여 웹 서비스 성능을 개선했다.

표 2. 시스템 아키텍처 비교 분석
Table 2. Comparative Analysis of System Architectures

비교 항목		기존 시스템	제안한 시스템
①	접근제어	사용자기반	직무기반
②	시스템 환경	서버 환경	웹 환경
③	보안정보 획득	SPS	UPS
④	직무서버	전용 서버	웹서버에 통합
⑤	웹서버	직무서버 포함	전용 서버
⑥	인증서 사용	일회성	재사용
⑦	인증서 발급자	직무서버	CA
⑧	시스템 통합기술	전부 개발	COTS 사용
⑨	RBAC	시스템 통합	시스템 통합/분리
⑩	사용자 정보	웹서버에 분산	직무서버에 집중
⑪	배포 데이터 처리	수작업	RMT 툴
⑫	웹환경 보안	취약함	강력함

⑥ 직무 서버로부터 발급 받은 인증서를 클라이언트는 자신의 컴퓨터에 보관해두고 인증서 만료 기간 이전에는 재발급 없이 재사용할 수 있게 하였다.

⑦ 직무 서버가 발행하는 인증서를 사용해 웹 서버를 접근함으로써 위장 사용자가 위조된 직무를 사용하여 서버를 불법 접근하는 것을 차단할 수 있다.

⑧ COTS 기술을 아키텍처에 반영하여 시스템을 구성함으로써 실제 시스템 구현 시에 구성 요소 모두를 새로이 구현할 필요성을 없앴다.

⑨ RBACi 엔진을 시스템 구성에 포함시킬 수 도 있고 직무에 객체 접근허가를 직접 부여하는 경우에는 RBACi 엔진을 시스템에 포함치 않고도 구현이 가능토록 한 모듈러한 아키텍처이다.

⑩ 사용자 정보를 단일 직무 서버에서 집중 관리함으로써 사용자 속성 정보의 관리를 단순화 시켰으며 정보의 분산으로 인한 불일치 문제를 원천적으로 방지했다.

⑪ 시스템 설치 시에 주로 사용할 수 있는 RMT를 시스템 아키텍처에 도입함으로써 관리 단계의 처리를 편리하게 했다.

⑫ 인증서 메커니즘, CRL 메커니즘, 웹을 기반으로 하는 안전한 직무기반 접근제어 모델, SSL을 결합함으로써 보안이 취약한 웹 환경에서 보안이 보장될 수 있는 시스템 아키텍처를 구성했다.

V. 결론

현재 널리 사용되고 있는 웹 서버 기반의 보안 체계에서는 아직도 사용자 수준의 인증이나 사용자 기반의 접근제어를 많이 사용하고 있다. 웹과 직무기반 접근제어의 통합은 각종 대규모 인터넷 기반 시스템의 보안을 강화할 수 있는 효과적인 수단을 제공할 수 있다. 직무기반 접근제어를 웹을 기반으로 하는 응용 시스템에 적용하기 위해서는 이를 위한 시스템 아키텍처 구성 방안이 마련되어야 하는데 이 논문은 이러한 필요를 충족시키는데 기여할 수 있을 것으로 기대된다. 4장에서 비교 분석한 것처럼 이 논문에서 제안한 시스템 아키텍처는 여러 가지 장점을 갖는 설계로서 시장(market)의 COTS 기술과 결합시킨다면 어렵지 않게 응용 시스템을 구축할 수 있을 것으로 사료된다.

아울러 이 논문에서 제안한 시스템 아키텍처는 서론에서 언급한 활용 분야 이외에도 통합 고등 교육 기관에서 유비쿼터스 캠퍼스를 구축 시에 디지털 교육지원 시스템, 원격 속기 지원 시스템 또는 장애 학생 서비스 지원 시스템 등의 보안체계에도 응용할 수 있을 것으로 기대되어 이에 관한 후속 연구가 필요하다고 본다.

참고문헌

[1] A. Schaad, "Detecting Conflicts in a Role-Based Delegation Model", Proc. of the 17th Annual Conference on Computer Security Applications, pp. 117-127, Dec. 10-14, 2001.

[2] Ravi S. Sandhu, Edward J. Coyne, "Role-Based Access Control Models", IEEE Computer, pp. 38-47, Feb. 1996.

[3] Bandmann O, Dam M, Firozabadi B S, "Constrained Delegation", Proc. of IEEE Symposium on Security and Privacy, pp. 131-140, 2002.

[4] Joon S. Park, Ravi Sandhu, "Decentralized User-Role Assignment for Web-based Intranets", Proc. of ACM on RBAC, pp. 1-12, 1998.

[5] Joon S. Park, RaviSandhu, "RBAC on the Web by Smart Certificates", Proc. of ACM on RBAC, pp. 1-9, 1999.

[6] Gail-Joon Ahn, Ravi Sandhu, Myong Kang, Joon Park, "Injecting RBAC to Secure a Web-based Workflow System", Proc. of ACM on RBAC, pp. 1-10, 2000.

[7] Andreas Schaad, Jonathan Moffett, Jeremy Jacob, "The RBAC of a European Bank", Proc. of ACM on RBAC, pp. 3-9, 2001.

[8] Joon S. Park, "Secure Attribute Services on the Web", Ph.D Thesis, George Mason University, Aug. 1999.

[9] 이호, 정진욱, "안전한 인터넷 사용을 위한 접근 제어 메커니즘 설계", 한국컴퓨터정보학회 논문지, 제 5권 제 3호, 84-90쪽, 2000년 7월.

[10] 이호, "웹기반 응용을 위한 직무기반접근 제어 모델의 설계", 한국사이버테러정보전학회 정보보증논문지 제 2권 제 2호, 59-66쪽, 2002년 12월.

[11] Elisa Bertino, Silvana Castano, Elena Ferrari, "On Specifying Policies for Web Documents with an XML-based Language", Proc. of ACM on RBAC, pp. 57-65, 2001.

[12] Microsoft, "Internet Information Server", IIS Help File with Windows 2000 Advanced Server, pp. 1-8, 2000.

[13] Serban I. Gavrila, John F. Barkley, "Formal Specification for RBAC User/Role and Role/Role Relationship Management", Proc. of ACM 3rd Workshop on RBAC, pp. 81-90, 1998.

[14] 이호, 정진욱, "통합 접근 제어를 위한 시뮬레이션 모델 설계", 한국컴퓨터정보학회 논문지, 제 9권 제 4호, 5쪽, 2004년 12월.

저 자 소 개

이 호

2002년 2월 :
성균관 대학교 대학원 정보공학과
(공학 박사)
2002년 3월 - 현재 :
한국재활복지대학 컴퓨터정보보안과
교수
관심분야 : 정보보호, 컴퓨터통신