CORBA 환경에서 역할-기반 접근제어 기법을 이용한 보안정책 관리

조 은정*, 이 금석
동국 대학교, 컴퓨터 공학과

Security Policy Management using Role-Based Access Control in CORBA Environment

Eunjung Cho*, Keumsuk Lee
Department of Computer Engineering, Dongguk university

요 약

분산 컴퓨팅 환경에서 응용 프로그램들은 다른 응용 프로그램들과 자원을 공유하고 통신하면서 보다 효율적으로 작업을 수행하게 된다 이러한 과정에서 침입자들에 의한 자원 손실을 막고 정보 무결성을 유지하는 것은 중요한 문제가 된다 따라서 분산 환경에서는 분산된 자원 사용에 대한 인증(Authentication) 및 인가(Authorization) 과정 등의 중요성이 증대되었다.

이질적인 시스템간의 분산 환경을 구축할 수 있는 미들웨어(Middleware)중 가장 활발히 연구되고 있는 CORBA에서는 보안 서비스를 제공해 주기 위해 COSS(Common Object Service Specification)3에서 보안 서비스를 정의했다 하지만 COSS3에서 정의한 객체 접근제어 기법만으로는 효율적인 보안정책 관리가 어렵다. 따라서 본 논문에서는 CORBA 보안 서비스 중 객체 접근제어를 위해 역할-기반 접근제어(RBAC Role-Based Access Control) 기법을 이용한 효율적이고 융통성 있는 보안정책 관리에 대해 논한다

1. 개 요

클라이언트/서버 환경이 확산됨에 따라 많은 응용 프로그램들이 개발되고 운영되면서 분산 서비스의 공통 부분이 될 수 있는 공통 기반 구조가 필요하게 되었다. CORBA[1][2]는 특히 이질적인 시스템을 통합하는 분산 객체 컴퓨팅(Distributed Object Computing) 환경을 구성할 수 있는 공통 기반 구조이다.

한편, 네트워크를 기반으로 한 분산 환경에서 외부로부터의 불법적인 침입은 전체 시스템의 운영과 무결성에 치명적인 문제를 일으킨다. 따라서, 분산되어 있는 자원에 대해 사용자가 접근을 요구할 때, 우선 사용자에 대한 인증 과정을 거친 후 인가된 사용자인지를 결정하는 과정이 필요하다.

이를 위해 가장 활발히 연구되고 있는 CORBA에서는 COSS3[1]에서 보안서비스를 정의하고 있다. 그러나 분산되어 있는 객체들에 대한 보안정책은 정책 변경, 삭제, 삽입 등의 관리비용이 많이 들고 보안정책의 충돌을 분석하기 어렵다. 또한 관리해야 할 조직들의 기능이 다양해지고 그 규모가 커짐으로써 조직의 특성에 맞는 보안정책 구현이 매우 어렵다. 따라서 COSS3에서 정의한 보안서비스를 구현하는 데 사용되는 접근제어 목록(Access Control List), 능력(Capability), 레이블-기반(Label-Based)과 같은 규칙 수준의 객체 접근제어 기법들보다는 역할/사용자(Individual), 역할/허가(Permission)의 관계를 분리하여 정책 관리를 추상화하는 역할-기반 접근제어[6][7][8]과 같은 정책 수준의 접근제어 기법이 필요하다.

본 논문에서는 CORBA 환경에서 제공해주는 보안서비스를 역할-기반 접근제어 기법을 이용하여 보안정책 관리를 용이하도록 하고, 또한 복잡한 보안정책을 융통성 있게 적용할 수 있도록 제안한다. 본 논문의 구성은 2장에서 CORBA 보안서비스를 소개하고, 3장에서는 보안서비스를 제공하고 있는 CORBA 제품군을 살펴보고, 역할기반 접근제어 기법을 살펴본다. 4장에서는 역할-기반 접근제어 기법을 이용한 CORBA 보안서비스를 설명하고 5장에서 결론 및 향후 연구를 기술한다.

2. CORBA 보안서비스

기존의 분산 시스템에서는 보안서비스를 제공하기 위해 외부 보안관리 응용프로그램을 사용하기 때문에 외부 보안관리 응용프로그램과의 통신이 필요하다. 그러나 CORBA에서 제공하는 객체 보안서비스(Object Security Service)는 별도의 외부 보안 응용프로그램 없이 ORB(Object Request Broker)자체에서 보안서비스를 제공해 준다. 따라서 CORBA 환경에서는 외부 보안 관리 응용프로그램과의 통신이 없으므로 성능의 향상을 기대할 수 있다[1]. 또한 COSS3에서는 ORB 자체에서 제공하는 보안서비스 외에 따로 보안서비스를 규정하여 사용할 수 있도록 하고 있다.

COSS3에서 정의한 보안서비스를 요약하면 다음과 같다[1][2].

① 보안 기능성(Security Functionality)은 두 레벨로 구성된다. 레벨 1에서는 ORB 자체에서 제공되는 보안서비스(접근제어, 감사)를 이용하기 때문에 서비스 객체 구현 및 서비스 객체 사용이 보안정책과 독립적으로 이루어질 수 있다. 그러나 레벨 2

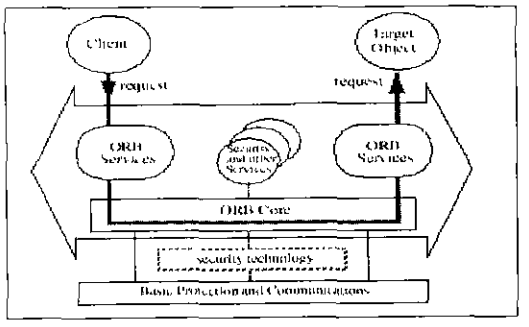
에서는 응용프로그램 수준에서 보안서비스를 정의하여 사용할 수 있을 뿐만 아니라 외부 보안 관리 프로그램 등을 포함할 수 있다.

② CORBA 환경에서 보안서비스를 구현하기 위해서 사용되는 보안 기법들은 필요에 따라 다양하게 선택되어질 수 있다. 이러한 보안 대체성(Security Replaceability)을 이용하여 융통성 있는 보안정책을 제공할 수 있다.

③ GIOP(General Inter-ORB Protocol)/IIOP(Internet Inter-ORB Protocol) 를 통해 서로 다른 ORB간에 안전한 통신을 제공한다.

그러나 CORBA 환경에서는 객체 접근제어시 목록, 능력, 레이블-기반-접근제어 기법 등을 이용한다. 하지만 이러한 기법들을 대규모 분산 환경에서는 사용할 경우 발생하는 보안정책의 충돌을 분석하기 어려울 뿐만 아니라 보안정책이 변경된 경우 이를 적용하기가 매우 어렵다.

(그림 1)은 CORBA 보안서비스를 간략히 나타낸 것이다. 좀더 자세히 CORBA 보안서비스의 동작과정을 살펴보면, 서비스를 요구하는 객체를 "클라이언트 객체"라 하고, 클라이언트 객체가 요구한 서비스를 제공해 주는 객체를 "서버 객체"라 한다. 일반적으로 CORBA 환경에서는 인증 작업을 위해서 커버러스(Keberos)같은 제 3의 인증 서버를 이용한다. 인증된 클라이언트 객체의 서비스 요청은 ORB내부에서 제공되는 보안정책에 따라 접근이 제어된다. 즉, 인증 작업을 거친 클라이언트 객체는 인증 서버로부터 유일한 인증 ID를 제공받는다 클라이언트 객체는 인증 ID를 받은 후 원하는 서버 객체와 통신하기 위한 보안 토큰을 전달받는다. 클라이언트 객체가 부여받은 인증 ID는 ORB를 통해 서버 객체에게 자동적으로 알려진다. 마지막으로 서버 객체는 접근제어 목록을 참조하여 클라이언트가 요청한 서비스에 대한 접근 결정(Access Decision)을 수행한다.



(그림 1) CORBA 보안서비스의 구조적인 모델

3. 관련 연구

기존의 CORBA 제품들은 보안서비스를 어떻게 구성하고 사용하고 있는지 살펴보고 본 논문에서 제안한 시스템의 효율적인 보안정책 관리를 위해 적용한 역할-기반 접근제어 개념을 살펴본다.

3.1 CORBA 제품군

CORBA 제품들은 COSS3에서 정의된 보안서비스 중 일부를 구현하여 객체에 대한 접근제어를 제공한다.

BEA의 ObjectBroker는 커버러스와 같은 제 3의 인증 패키지를 이용하여 보안서비스를 제공하고, DNS Tech. 에서는 Java에서 제공하는 보안 구조와 통합하여 보안서비스를 제공한다[3].

그러나 BEA와 DNS의 제품에 비해 보다 충실히 COSS3 보안서비스를 제공하는 Iona의 Orbix[2]는 2장에서와 같이 보안 기능성 레벨 1에서 보안서비스를 제공하고 있다.

Orbix는 GSS-API(Generic Security Service-API)[5]로 커버러스, SESAME 등의 외부 보안 소프트웨어를 이용하여 인증 작업을 수행하고, 커버러스를 이용할 경우 DES(Data Encryption Standard)로 메시지를 암호화한다 마지막으로 CORBA에서 인트셉터(Interceptor)의 역할을 하는 Orbix 필터(Filter), 변환자(Transformer)를 이용하여 CORBA 보안서비스 레벨 1의 기능 즉, 인산/인터페이스/서버간의 ACL을 참조하여 객체 접근제어를 제공해 주고 있다.

위의 제품들은 COSS3에 기반 하여 보안서비스를 제공해 주고 있으나 2장에서와 같이 보안정책 관리 측면에서의 많은 부담을 기지고 있다.

3.2 역할-기반 접근제어 (RBAC)

시스템이 대규모화되고 다양해지면서 조직들은 그 조직 특성에 적합한 복잡한 보안정책을 필요로 하게 되었고, 보안정책의 일관성 유지 및 보안정책의 변경을 실제 시스템에 적용하기 위한 비용이 높아졌다. 2장에서 언급한 기존의 접근제어 목록, 능력, 레이블-기반 접근제어 기법들은 규칙 수준에서 접근제어 서비스를 제공하기 때문에 위와 같은 요구를 만족시키기 어렵다.

기존의 접근제어 기법에서는 각 사용자에게 권한을 할당하는 반면, 역할-기반 접근제어 기법에서는 필요한 역할(Role)과 그 역할이 수행할 수 있는 연산을 보안정책에 맞게 정의한 후, 실제 사용자들에게 각자 역할을 할당하는 기법이다. 따라서 역할-기반 접근제어 기법은 보안정책 관리자에게 사용자와 자원 접근 권한의 관계를 독립적으로 생각할 수 있게 함으로써, 추상화된 보안정책의 적용이 가능하다. 예를 들면, 은행에서 역할 R1-"P1' A객체를 수정할 수 있다"이 보안정책 변경으로 인하여 "P1'' A객체를 단지 읽을 수만 있다"로 바뀐다면, 역할 R1에 해당하는 사용자에 대해서는 변경 작업이 필요 없다 그러나 기존의 접근제어 기법을 이용하는 경우 동일한 기능을 하는 사용자들은 그룹 단위로 관리되는데, 정책이 변경되면 각 사용자가 가진 권한에 대해서도 변경이 되어야 한다. 또한 역할-기반 접근제어 기법을 이용하면 복잡한 보안정책도 추상화하여 표현할 수 있기 때문에 보다 효율적으로 보안정책을 관리할 수 있다[7]. 예를 들면, "병원에서 역할-간호원은 오후 10부터 다음날 아침 7시까지는 환자 카드를 작성할 수 있다"는 보안정책이 필요한 경우, 접근제어 목록 기법과 같이 기존의 방식에서는 해당하는 모든 객체에 대하여 보안정책을 표현해야 하기 때문에 보안정책의 충돌을 분석하기 어려울 뿐만 아니라 보안정책 관리자의 작업 오류도 발생할 수 있다.

4. 역할-기반 접근제어접근제어 기법을 이용한 보안정책 관리

COSS3에서 정의한 보안 대체성을 이용하면 역할-기반 접근제어 기법을 CORBA 환경에서 보안서비스로 이용할 수 있다

3장에서와 같이 CORBA 제품들은 접근제어 수행시 규칙 수준에서 객체에 대한 접근제어를 하기 때문에 보안정책 관리에 많은 어려움이 있다.

따라서 본 논문에서는 CORBA 환경에서 객체의 보안정책 관리에 정책 수준의 역할-기반 접근제어 기법을 적용하여 분산되어 있는 객

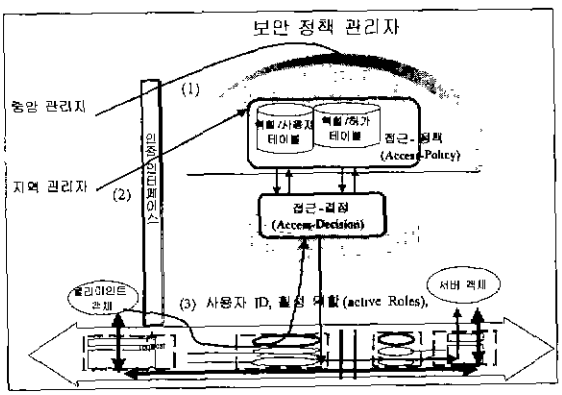
체들에 대한 보안정책의 정책 변경, 삭제 및 삽입의 관리를 효율적으로 하고 다양한 기능의 조직 특성에 맞는 보안정책 구현에 융통성을 제공하는 시스템을 제안한다.

4.1 가정

본 시스템 구현을 위한 CORBA 환경은 CORBA 스펙(Spec) 2.0에 기초한다. 또한 시스템 구현 환경은 하나의 ORB 내에서 클라이언트 객체와 서버 객체간의 접근제어를 지원하는 것으로 가정한다 또한 보안정책 관리자는 역할/허가(Permission) 관계를 관리하는 "중앙 보안정책 관리자: 중앙 관리자"와 역할/사용자(Individual) 관계를 관리하는 "지역 보안정책 관리자: 지역 관리자"로 나눈다. 이러한 권한의 분리로 관리자의 권한 남용을 막을 수 있다. 중앙 관리자는 각 조직에 필요한 역할과 연산을 정의 및 관리하고, 지역 관리자는 각 지역(부서)별로 필요한 역할들에 사용자를 할당 및 변경 관리를 한다 [8]. 한 사용자는 복수의 역할을 가질 수 있고, 사용자가 서버 객체를 사용하기 위해서는 인증 과정에서 자신이 원하는 역할을 선택해야 한다. 또한 사용자는 인증 과정시 자신이 선택한 활성 역할(Active Role) 이외의 권한을 가질 수 없다.

4.2 보안정책 관리자(Security Policy Manager)

(그림 2)의 보안정책 관리자는 크게 세 부분으로 구성된다. 첫째, 인증 인터페이스(Authentication Interface) 는 관리자와 사용자의 인증 서비스를 제공한다. 둘째, 보안-정책(Access-Policy)은 중앙 관리자와 지역 관리자에 의해 조직에 필요한 역할과 그 역할이 수행할 수 있는 연산과의 관계를 나타내는 역할/허가 테이블과 역할과 사용자의 관계를 나타내는 역할/허가 테이블을 유지하여 접근-결정(Access-Decision)에게 정보를 제공한다. 셋째, 접근-결정(Access-Decision)은 클라이언트 객체가 서버 객체에 접근을 요구할 때 정의된 접근-정책을 이용하여 접근의 허가 여부를 담당한다



(그림 2) 역할-기반 접근제어 기법을 이용한 보안정책 관리시스템 구조

(그림 2)의 보안정책 관리자의 동작을 살펴보면 다음과 같다

(1) 역할/허가 관리; 인증된 중앙 관리자는 조직의 목적에 맞는 역할, 역할이 어떤 객체에 어떤 연산을 수행 할 수 있는지를 정의하고, 유지한다.

(2) 역할/사용자 관리, 인정된 지역 보안 관리자는 각 지역(부서)에서 관리할 수 있는 역할들을 그 지역의 사용자에게 할당, 회수한다. 다음의 원칙에 따라 사용자에게 역할을 할당한다.
- 사용자는 작업을 수행하기 위해 가져야 하는 권한 이외에 다른 권한을 가질 수 없다.
- 의무 분리 정책 또는 권한 충돌: 사용자가 한 역할을 가지고 작업을 수행할 때 다른 역할의 일원이 될 수 없다.
- 역할의 일원이 될 수 있는 사용자의 수를 초과할 수 없다

(3) 인증된 사용자가 서버 객체의 서비스를 제공받기 위해 활성 역할(Active role)과 서버 객체에서 수행할 연산을 지정한다. 접근-결정(Access-Decision)은 주어진 활성 역할을 조사하여 권한여부를 조사하고 연산의 허가/취소를 결정한다. 만약 아래와 같다면 사용자는 원하는 객체에 접근 할 수 있다
- 사용자가 서버 객체에 접근을 위해 요구한 역할에 권한 있다
- 요구한 역할이 사용자의 다른 활성 역할과 충돌이 없다
- 서버 객체에서 수행할 연산이 권한여부가 확인된 활성역할에 속한다.

5 결론 및 향후연구

본 논문에서는 CORBA 보안서비스의 객체 접근제어에 역할-접근제어 기법을 적용하여 하나의 조직에서 필요한 보안정책 구현을 용이하게 하였고, 보안정책의 충돌을 분석할 수 있다. 또한 복잡한 보안정책을 융통성 있게 구현 할 수 있게 하였다.

이질적인 시스템들간의 상호 운영성이 주요한 관심이 되는 요즈음 다른 보안 기법을 가지는 ORB간의 안전한 통신이 필요하다. 그러나 CORBA 보안서비스에서도 다른 보안 기법들을 사용한 ORB간의 상호 운영성을 제시하지 않고 있다. 앞으로 다른 보안 기법들의 호환성에 관한 연구가 진행되어야 할 것이다.

참고 문헌

[1] OMG, "CORBAservices: Common Object Services Specification," 1998
http://www.omg.org/corba/sectrans.htm#sec

[2] Jonn Siegel, "CORBA fundamentals and programing," John Wiley & Sons, Inc, 1996

[3] Konstantin Beznosov, "CORBASEC Frequently Asked Questions and Answers," /http://www.bhssf.org/IT /Project s/cpr/security/CORBASEC-FAQ/, 1998

[4] IONA TechnologiesI, "OrbixSecurity White Paper Part 1 of 2," http://www.iona.com/products/security/index.html

[5] J, Linn, "Generic security service application program interface," Geer Zolot associates, 1993

[6] David F. Ferraiolo and Richard Kuhn, "Role-based access control," Proceedings of the 15th NIST-NSA National computer security congerence, 1992

[7] Ravi S. sandhu, Edward J.Coync, Hal L. Feinstein and Charles E. Youman, "Role_-Based Access Control Models," IEEE computer, Colume 29, number 2, Feb 1996

[8] David F. Ferraiolo, J Cugini and Richard Kuhn, "Role-Based Access Control: Features and Motivations," National Instityte of standards and technoloty, 1995