CBD개발방법에 의한 CORBA 보안정보관리 Component 설계

The Design of CORBA Security Information Management Component Base on CBD

김재열, 송미영
한국한의학연구원 한의학정보화사업단

Kim jae-yeol, Song mi-young
Korea Institute of Oriental Medicine

요약

분산객체 처리 환경에서 CORBA를 기반으로 한 응용 객체간에 전달되는 정보에 대하여 보안을 유지하기 위해 요구되는 보안 정보를 관리하는 기법을 CBD설계방법으로 제안하고 이들을 관리하기 위한 보안정책들의 객체들을 효율적으로 관리하기 위한 방안을 제시하였다.

I. 서 론

최근 정보처리 기술 중 하나인 분산객체 처리 환경은 여러 통신망으로 연관된 자원들을 공유하는 분산처리 환경과 상속, 다양화, 캡슐화, 재사용 등의 장점을 지닌 객체 지향 처리 환경을 결합한 것으로 주어진 환경 조건을 기반으로 융통성 있게 통신 분야에 신뢰성 있는 새로운 형태로 주목을 받고 있다. 그러나 정보처리 환경은 정보의 공유가 확산됨과 동시에 정보보안의 역기능적인 측면의 부작용도 증가하여 심각한 문제로 대두되고 있다.

분산객체 처리 환경에서 정보보안을 지원하기 위한 기술들이 많이 개발 중에 있고 분산 객체 처리 환경에서 CORBA를 기반으로 하는 응용 객체간의 정보 보안을 지원하기 위하여 OMG CORBA Security 명세서를 제안하고 있다[6][10].

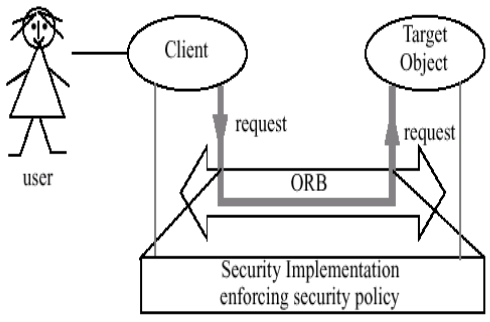
또한 OMG CORBA Security 명세서에 있는 보안 정보 관리 부분만 생각해 볼 때 각각 보안 정보 관리를 하기 위한 정책 정보와 보안 정보의 관리가 어렵고 단순한 테이블 형태로 존재하여 각 테이블을 단순 조인하는 검색으로 이루어지고 있다. 본 논문에서는 보안 정보관리의 정확성, 무결성, 효율성을 기하기 위한 Repository 활용 방안을 제시하였다.

II. CORBA 보안 구조

1. 보안참조 모델

보안 참조 모델은 안전성을 필요로 하는 시스템이 보안 정책을 어디서, 어떻게 시행하는가를 기술한다. 보안 정책은 객체를 접근할 수 있는 조건, 사용자 또는 Principal이 누구이며 허용된 일이 무엇이고 그들의 권한을 위임할 수 있는지 여부를 보여주는 인증에 대한 정보, 객체간 통신의 안전성 품질(Quality of Protection 등), 보안 관련 행위들에 대하여 어떤 책임이 요구 되는지 등을 정의한다.

보안 정책에는 접근 제어 정책(Access Control Policy), 감사정책(Auditing Policy), 인증 정책(Authentication Policy), 보안 호출 정책(Security Invocation Policy), 부인 봉쇄 정책(Non-Repudiation Policy), 위임 정책(Delegation Policy) 등이 있을 수 있다[2].



▶▶ 그림 1. 객체 시스템을 위한 보안모델[2]

CORBA를 기반으로 한 객체 시스템의 보안 모델은 <그림 1>에서 보는 바와 같이 나타낼 수 있다. 모든 작업은 반드시 보안 정책을 시행하는 적절한 구현 모듈을 거쳐서 수행 되도록 하고(by-pass impossible), 보안 기능들은 그 자체가 도중에 불법으로 변경되거나 절취 되지 않도록 안전성을 유지해야 하며, 보안 정책에 의해 항상 호출될 수 있어야 한다.

대부분의 응용 객체들은 보안 정책이 어떻게 되어 있는지 또는 내부적으로 어떻게 처리되는지 알지 못한다(unaware of security). 사용자는 응용 클라이언트를 호출하기 전에 사용자로서의 자격을 인증 받고, 계속해서 보안 서비스가 자동으로 수행된다. 어떤 응용 객체들은 시스템이 제공하는 보안 정책의 통제를 받고 있지만 그 자신이 보안 기능을 수행하지는 않으며. 어떤 응용 객체는 자신이 정한 보안 정책(예를 들면, 그들 자신의 데이터에 대한 접근 제어나 보안 관련 감시 활동)들을 수행한다[4][5][10].

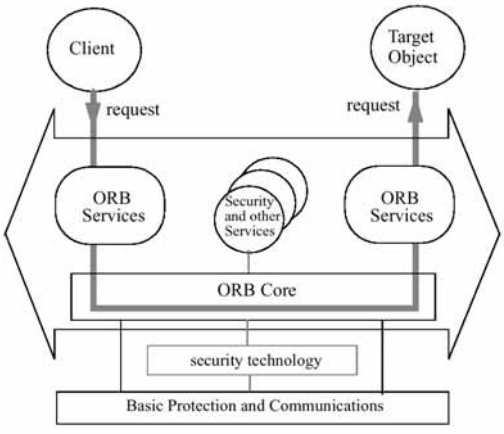
ORB는 보안 정책을 위배하지 않는 정당한 요구에 대해서는 최소한 올바른 처리를 해주도록 해야 하며 보안 정책에 의해 요구된 보안 서비스를 수행해야 한다.

보안 모델은 일반적으로 보안 정책들의 특정 집합을 정의한다. OMA(Object Management Architecture)는 서로 다른 시장의 요구를 만족하는 다양한 보안 정책을 광범위하게 지원해야 하므로 단일 보안 모델의 제안은 적합하지 못하고 많은 다른 종류의 정책들을 만들어 낼 수 있는 기본 골격(framework)을 제공하는 보안 참조모델(또는 meta-policy)을 정의하는 것이 필요하다. Meta-policy는 보안 구조에서 제공하는 추상화 된 인터페이스와 가능한 보안 기능들을 정의하고 유연성 있는 지침을 제공한다[2].

2. 보안구조 모델

CORBA에서 object invocation시 보안 서비스를 제공하기 위한structural model은 아래 와 같이 4개의 구성 요소로 구성된다.

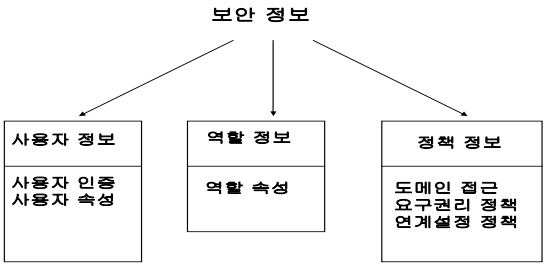
- Application-level components
- 특정 보안 technology에 독립적인 보안 서비스 components
- 특정 보안 technology 처리 components
- Basic protection & communication components



▶▶ 그림 2. Structural model 의 구성[2]

III. 보안정보 관리 기법 및 설계

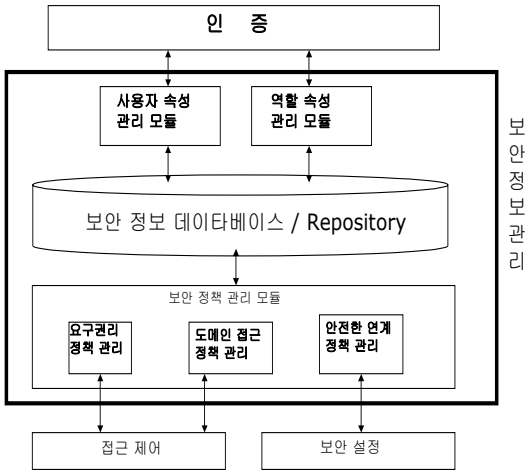
CORBA Security 환경에서 관리해야 하는 보안 정보는 사용자의 인증 정보 및 속성 정보와 역할 속성 정보, 접근 제어를 위한 정책 정보, 안전한 연계를 설정하기 위한 정책 정보로 구성 할 수 있다.



▶▶ 그림 3. 보안 정보 관리 객체 환경에 있는 보안 정보

사용자 인증 정보는 사용자의 암호와 비밀 키 등이 있고, 사용자 속성 정보는 사용자의 역할 이름과 속성권한 등이 있다. 또한 역할 속성 정보는 이 역할에 존재하는 타겟과 위임 여부 정보 등이 있다. 접근 제어 정보는 사용자가 응용 객체에 기술된 인터페이스의 특정한 메소드를 실행하는 여부를 나타낸 요구권리(RequiredRights) 정책 정보와 사용자의 권한 속성에 특정한 권리를 결합시킨 형태로 권한 속성과 위임 플래그와 권리로 구성되는 도메인 접근 정책(DomainAccessPolicy) 정보 등이 있고, 응용 클라이언트와 응용 서버사이에 데이터를 주고받는데 요구되는 데이터의 보안 수준을 나타내는 안전한 연계를 설정하는 정책(SecureInvocationPolicy)정보가 있다.

보안 정보 권리 구조는 사용자 속성 관리 모듈, 역할 관리 모듈, 접근제어 정책 관리 모듈, 안전한 연계 설정 정책 관리 모듈로 구성된다.



▶▶ 그림 4. 보안 정보 관리 구성요소[6]

사용자 인증 및 속성 관리 모듈은 사용자 데이터베이스(Repository)에 정의되어 있는 사용자 인증 정보 및 속성 정보를 관리하며, 역할 속성 관리 모듈은 역할 데이터베이스(Repository)에 정의 되어 있는 역할 속성 정보를 관리한다. 이들 두 관리 모듈은 사용자 인증을 위하여 아래와 같은 관리 정보를 제공하는 인터페이스를 갖는다[6][7].

```
//User authentication and attribute management
        void UserAuthentication:get_user_auth();

//Role privilege management
        void UserPrivilegeAttribute::add_user();
        void UserPrivilegeAttribute::delete_user();
        void UserPrivilegeAttribute::replace_user();
        void UserPrivilegeAttribute::get_user();
```

보안 정책관리 모듈은 접근 제어를 위한 정책 정보와 안전한 연계 설정을 위한 정책 정보를 관리하는 인터페이스와 객체에 대하여 사용자가 접근이 허용되는 지를 결정하거나 응용 클라이언트와 응용 서버 사이의 안전한 연계 설정을 위하여 요구되는 아래와 같은 각 정책 정보를 제공하는 인터페이스를 갖는다.

```
//RequiredRights Policy management
      void RequiredRights::get_required_right();
      void RequiredRights::set_required_right();

// Domain Access Policy management
      void DomainAccessPolicy::grant_right();
      void DomainAccessPolicy::get_right();
      void DomainAccessPolicy::replace_right();
      void DomainAccessPolicy::revoke_right();

//Security Invocation Policy management
      void SecurityInvocationPolicy::get_association();
      void SecurityInvocationPolicy::set_association();
```

CORBA 보안의 다른 서비스(인증, 접근제어, 안전한 연계 설정 서비스)는 자신의 기능을 수행하기 위해서 우선 원격에 있는 보안 정보 관리 객체에 바인딩하며[1], 이후 적절한 인터페이스를 이용하여 보안 정보 데이터베이스(Repository)로부터 보안 정보를 얻어 온다. 또한 이러한 보안 정보 관리에 대한 관리는 특정한 관리자 계정을 두어 관리 하도록 한다.

응용 객체 클라이언트는 응용 객체를 호출할 때에는 정보에 대하여 보안을 유지하기 위하여 필요한 보안 정보를 얻는다. 각 보안 정보의 유형에 따라 적절한 보안 정보 제공 객체를 호출한다. 보안 제공 객체는 사용자 정보, 역할 정보, 도메인 접근 정책 정보, 요구권리 정책 정보, 안전한 연계 설정 정책 정보 등의 객체가 있다.

[요구권리]

요구권리는 보안정보관리 메커니즘을 통해서 각각 거쳐야 할 Object들의 정보를 Repository에서 Meta-policy-data 정보를 얻고 추가, 삭제 등의 작업을 수행 하게 된다. Domain들은 서로 같은 Policy를 가지고 있을 수 있다. 이렇게 중복된 Policy의 변경 혹은 삭제로 인한 여러 개 그리고 다차원적인 Domain을 별도로 관리 하면서 같이 변경하여 줄 수 있고, 만약 변경된 domain 과 Policy를 적용하여 요구권리를 하였을 때는 정상적인 운용이 가능해진다. 따라서 Repository를 사용하게 되면 domain 과 Policy들을 별도로 관리할 수 있게 인터페이스를 추가하여 만들어 줄 필요가 없게 된다.

[안전한 연계설정관리]

연계설정관리도 요구권리와 같은 방식으로 보안정보관리에 필요한 각각의 Object의 정보를 Repository에서 Meta-policy-data 정보를 획득한다. 각각의 객체들끼리의 연계설정에서 오는 Version정보유지는 과거에 연계를 지어준 정보가 존재 하지 않음으로 해서오는 반복적인 작업과 History 정보를 무시하고 같은 레코드에 Update하는 낭비를 줄일 수 있다. 보안 설정 모듈을 통해서 들어오는 객체들은 접근할 때 마다 연계설정을 수시로 해주어야 한다. 보안 객체의 History 정보로 인한 반복적인 작업이 없어지고 또한 매번 Update에서 발생되는 트랜잭션도 감소한다.

[도메인 접근정책]

도메인 접근정책도 앞에서 제시한 요구권리와 연계설정관리 메커니즘과 같은 방식으로 Meta-policy-data 정보를 Repository에서 추가, 삭제 등의 형식으로 관리되어진다. Principal의 순수성을 검증할 수 있도록Level를 두어 앞 단계에서 어떤 작업으로 인증 되어 거쳐 왔는지를 알 수 있고 신임할 수 있는지 여부를 판단 할 수 있다. Domain과 Policy 들 사이의 연관 관계성을 나타낼 수 있고 그들 간의 스키마 관리도 가능해진다.

IV. 결 론

본 논문은 분산객체 처리 환경에서 보안 서비스를 제공하기 위해 필요한 보안 정보들을 관리하기 위한 기법을 제시하고 설계 하였다. 본 논문에서 제시한 보안정보관리 기법은 CORBA Security 명세서를 따라 설계 하였다.

▌참 고 문 헌▐

[1] OMG. "CORBAservice : Common Object Specification", 1996
[2] OMG. "CORBA Security service Specification",1995
[3] ISO/IEC, "Information technology IRDS framework", ISO/IEC, 1993
[4] Randy Otte, "Understanding CORBA", Prentice Inc., 1996
[5] Thomas J. Mowbray, "The Essential CORBA", Wiley & Sons Inc., 1995
[6] 나중찬, "SCAP에서의 보안 정보관리", 제1회 개방형보안기술과 응용 워크샵, 통신정보보안학회, 1997
[7] 송영기, "초고속정보통신 기반 안전성지원 플랫폼개발", 제1회 개방형보안기술과응용워크샵,통신정보보안학회, 1997
[8] 한국전산원, "IRDS 서비스 표준 연구", 한국전산원, 1995
[9] 염태진, 박재형, 리자, 김기봉, 진성일, "분산 환경에서의 IRDS 기반 정보저장소 설계 및 구현," 한국정보처리학회, 1998
[10] 이권일, "분산객체환경에서의 보안서비스 구현", 통신정보보안학회논문지, 1998