CORBA 기반의 보안 플랫폼과 그 응용

나 중 찬† · 김 영 균† · 김 경 범†† · 김 명 준†††

요 약

본 논문에서는 인터넷에 기반한 분산 객체 시스템 환경에서 존재하는 보안 위협에 대처할 수 있는 SCAP(Security platform for CORBA based APplication)이라는 보안 플랫폼을 구축한다. SCAP은 OMG(Object Management Group)에 의해 발표된 CORBA 보안 규격을 지원하며, 대규모 네트워크에 유용한 공통의 안전한 상호운용성 레벨 2(Common Secure Interoperability Functionality level 2 : CSI Level 2)을 지원하며, 주요 네 개의 블록(인증 블록, 보안 설정 블록, 접근 제어 블록, 보안 정보 관리 블록)으로 구성되어 보안 서비스를 제공한다. 그리고 이를 구현하는데 있어서 기존의 보안 기술을 이용하여 CORBA 보안 규격을 따른 외부 서비스를 제공하도록 하였으며, 객체 중개자 역할을 하는 상용적인 CORBA 제품의 원천코드 없이 인터셉터(interceptor) 기법을 실현하는 방법과 보안 서비스를 위해 요구되는 Current 객체를 확장하는 방법을 추구하였다. 끝으로 웹 서버와 정보 제공자 환경에서 SCAP 보안 플랫폼을 이용하여 보안 서비스의 제공 및 활용을 살펴보았다.

A Security Platform based on CORBA and its Application

Jung-Chan Na† · Young-Kyoon Kim† · Kyeong-Beom Kim†† · Meong-Joon Kim†††

ABSTRACT

This paper proposes a security platform, called SCAP(Security platform for CORBA based APplication), to cope with potential threats in a distributed object system. SCAP supports CORBA security specification announced by OMG. SCAP is comprised of four functional blocks, which co-work with ORB to provide security services: Authentication Block, Association Block, Access Control Block, and Security Information Management Block. It is designed to support Common Secure Interoperability Functionality Level 2, which is useful for large-scale intra-, or inter-network based applications. Actual security services, which are dependent on supporting security technology, will be provided as external security service for replaceability. Implementation issues such as how to simulate an interceptor mechanism using a commercial ORB product without source code, and how to extend Current object required for security services are also described. At the end of the paper, the SCAP applied to the web environment is described to show its practical utilization.

1. 서 론

최근 정보 기술은 인터넷에 의해 상호 연결되는 다수 컴퓨터 시스템들로 구성된 분산 처리 시스템으로 발전하여 왔다. 더욱이 객체 지향 기술이 분산 시스템에 적용되기 시작됨으로써, 분산처리와 객체지향 기술이 통합된 분산 객체 처리 시스템은 기존에 개발된 소프트웨어의 재사용을 증가시키는 동시에 다양한 응용 개발의 효율성을 향상시킬 수 있는 새로운 정보 기술로 발전하였다.

그러나 분산 객체 처리 시스템은 전통적인 시스템에 비해 보다 많은 보안 침해 가능성이 존재한다. 이러한 위협은 주로 기밀에 해당되는 중요한 객체 정보에 대

† 정 회 원 : 한국전자통신연구원 인터넷서비스연구부 선임연구원
†† 정 회 원 : 한국전자통신연구원 인터넷서비스연구부 책임연구원
††† 종신회원 : 한국전자통신연구원 인터넷서비스연구부 책임연구원
논문접수 : 1999년 10월 4일, 심사완료 : 1999년 11월 10일

한 불법적인 접근, 사용자 위장 침입, 통신망의 도청, 통신 선로 상의 데이터 변조 등과 같은 다양한 보안 공격 등이 있다[1]. 그리고 중앙집중 시스템에서의 보안 기술은 분산 객체 시스템 환경에 적용하는데 있어서 적합치 않다. 따라서 본질적인 분산 객체 지향 성질을 이용하는 새로운 보안 기술이 요구된다.

분산 처리 시스템의 보안 서비스를 지원하기 위한 보안 기술은 Kerberos[2], Security Framework for POSIX Open System Environment[3], OSF DCE(Open System Foundation Distributed Computing Environment)[4], SESAME(Secure European System for Applications in a Multivendor Environment)[5] 등과 같이 많은 연구와 구현이 진행되었다. 많은 실제 사용자 환경에서의 Kerberos 이용은 실제로 지역 네트워크 환경 또는 폐쇄된 이용자 그룹 내에서 인증 서비스와 안전한 분산 응용을 구축하는데 있어서 기반을 제공하는 능력이 증명되었다. 그러나 Kerberos는 순전한 대칭 키 분배와 사용자의 신원에 기반한 인증 모델을 지원하는 현재의 한계에 의해 제약을 받는다. POSIX에서는 안전한 개방형 환경을 위하여 보안 프레임워크 및 인터페이스를 제공하였지만, 구현이 용이한 안전한 객체 모델 및 구조를 제공하지는 않았다. DCE 보안 서비스와 SESAME는 비대칭 키 분배와 사용자의 권한에 기반한 인증 모델을 지원하고 있어 분산 처리 환경에 적합한 보안 서비스를 제공하지만, 기존에 개발된 소프트웨어 재사용을 증가시키는 객체 지향 개념이 없다.

분산 객체 시스템은 일반적인 보안 위협 요소 이외에도 객체의 상속 특성으로 인해 새로운 요소들이 계속적으로 생성 소멸 변경되며, 보안 정책도 기업 환경의 변화에 따라 계속 변하는 특성을 갖는다. OMG(Object Management Group)[6]에서 발표한 CORBA(Common Object Request Broker Architecture)는 이질적인 분산 환경에서 응용 객체들간의 상호운용성과 다중 객체 시스템의 상호 연결성을 제공하기 위해 규정된 객체 요구 중개자의 표준 구조를 갖고 있음으로써 분산된 객체들 사이의 투명한 요구와 응답을 제공하는 분산객체 컴퓨팅을 지원해 준다. 또한 CORBA 기반의 분산 처리 응용들은 다양한 보안 위협들로부터 안전성이 보장될 수 있는 보안 서비스의 필요성이 요구되는데, 이를 위해 OMG는 CORBA 기반 응용의 보안성을 유지하는데 핵심적인 보안 기능과 보안 인터페이스의 표준 안을 제시하였다[1, 7]. 기존에 존재하는 다양한 보안 기술들은 CORBA 보안 규격에 따른 외부 보안 서비스를 제공하기 위해 이용될 수 있는데, 대표적인 예는 공통 보안 인터페이스로써 GSS-API(Generic Security Service-Application Programming Interface)[8]기술을 이용하여 보안 객체를 구현하는 부담을 덜 수 있다. 또한 CORBA 보안 규격은 ORB간의 안전한 상호운용성을 위해 CSI 규격[9]을 발표함으로써 새로운 분산 객체 응용을 위한 보안 플랫폼으로서 인식되고 있다.

본 논문의 목적은 분산 객체 처리 시스템에서 안전성 위협에 대한 보호를 하기 위하여 SCAP이라는 보안 플랫폼을 제안하며, 이의 응용으로써 웹 상에서 웹 서버와 정보 제공자 환경에서 보안 서비스를 제공하기 위한 응용 수준의 구성을 제안한다.

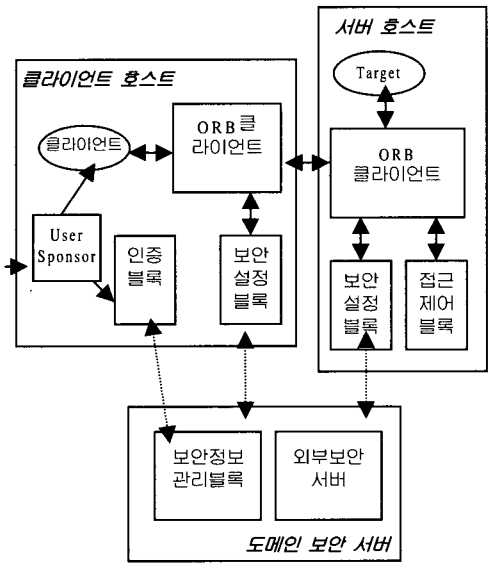
2장에서는 SCAP설계 시 고려사항과 전체적인 구조를 나타내며, 3장에서는 SCAP의 각 기능 블록에 대한 설계 내용을 기술한다. 4장은 SCAP 구현 시 고려사항을 기술하며, 구현된 SCAP과 분산 환경에서 보안 서비스를 제공하기 위한 여러 플랫폼들과 비교 분석하였다. 5장은 SCAP을 기반으로 한 웹 서버와 정보 제공자 환경에서 SCAP 보안 플랫폼을 이용하여 보안 서비스의 제공 및 활용을 살펴보았으며, 마지막으로 결론을 기술한다.

2. SCAP 개요

SCAP 시스템은 분산 객체 환경인 CORBA를 기반한 응용들에 대한 보안 서비스를 제공하는 보안 플랫폼이며, 초고속 응용 서비스를 안전하게 보호할 수 있는 활용 가능한 플랫폼이다. 따라서 SCAP은 인증, 접근 제어, 보안 통신, 그리고 보안 정책 관리와 같은 보안 서비스를 지원하여 독립적 또는 CORBA에 연동된 시스템들에게 보안성을 제공하며, 대규모 인트라넷이나 혹은 광대역 기업 통신망을 근간으로 하는 응용들에 유용한 CORBA 보안 서비스 규격의 CSI Level 2를 지원하도록 설계한다[15]. 실질적인 보안 서비스는 인증 알고리즘, 암호화, 키 생성 및 분배 등을 자체적으로 해결하는 다양한 시스템들에 종속적이기 때문에, CORBA 보안 서비스 규격에서 제안하였듯이 SCAP은 이를 언제든지 대체할 수 있는 외부 보안 서버로 구현한다. 이를 위해서 CSI-ECMA(European Computer Manufacturers Association) 프로토콜을 지원하는 SESAME의 신뢰성 있는 제삼자(Trusted Third Party : TTP) 구

성요소를 선택하여, 이를 SCAP의 외부 보안 서버로 활용한다.

SCAP의 전체적인 운용구조는 다음 (그림 1)과 같다.



(그림 1) 보안 플랫폼의 운영 구조

CORBA 보안 플랫폼은 클라이언트/서버 구조를 기반으로 하며, 도메인 보안 서버(domain security server) 시스템, 클라이언트 호스트 시스템, 서버 호스트 시스템으로 구성된다. SCAP 개발의 기술적 전략 차원에서 권한 위임, 보안성 상호운영, 보안 감사, 부인 봉쇄(non repudiation)와 같은 보안 기능들은 SCAP 시스템의 현재 버전에는 포함되지 않으며, 추후 차기 버전에서 지원될 예정이다.

도메인 보안 서버 시스템에는 보안 정책 관리 블록과 외부 보안 서버가 포함된다. 보안 정책 관리 블록은 보안 관리자가 도메인의 보안 정책을 관리할 수 있는 인터페이스와 관리 기능을 제공하며, 그리고 인증 정보, 사용자 및 역할 속성 등과 같은 보안 정보를 저장 관리한다. 외부 보안 서버는 인증 서비스, 권한 속성 서비스, 보안 설정 서비스, 암호화 지원 기능 등과 같은 실제적인 보안 기능을 제공한다. 물리적인 구성 측면에서의 외부 보안 서버는 인증 서버(authentication Server : AS), 권한 속성 서버(Privilege Attribute server : PAS), 키 분배 서버(Key Distribution Server : KDS), 인증 기관(Certificate Authority : CA)으로 이루어진 SESAME의 TTP이다.

TTP를 구성하는 각 요소들이 처리하는 기능을 정리하면 다음과 같다. 먼저, AS는 클라이언트 사용자를 대신하는 principle에 대한 인증을 수행하며, PAS는 인증된 principle이 갖는 접근 권한을 부여한다. KDS는 PAC(privilege attribute certificate)와 정보 교환 시 이용되는 암호화 키를 principle에게 제공한다. CA는 공개 키를 포함하고 있는 디렉토리 인증서(directory certificate)를 발행하여, CA를 신뢰하고 있는 개체가 만들어낸 시그니쳐(signature)가 정당한지를 확인할 수 있게 해준다.

SCAP의 클라이언트와 서버 시스템은 인증 블록, 보안 설정 블록, 접근 제어 블록, ORB로 구성된다. 인증 블록은 인증을 요청한 주체가 누구인지를 검증하기 위해서 principle을 식별하고 인증하며, 보안 설정 블록은 클라이언트와 서버 사이에 보안 통신로를 설정하여, 교환되는 메시지의 비밀성과 무결성을 보장한다. 그리고 접근 제어 블록은 서버 응용 서비스를 요청한 principle이 응용 객체에 접근할 수 있는지를 결정하는 접근 제어 기능을 제공하며, 이는 대행자의 권한 속성과 접근할 응용 객체의 제어 정보를 바탕으로 이루어진다.

보안 서비스를 제공하는 SCAP의 동작 과정은 다음과 같다. 클라이언트 호스트에서 인증이 이루어지기 위해서는 사용자가 자신의 식별자와 암호, 그리고 역할 정보를 스폰서(user sponsor)에 제출해야 하며, 스폰서는 이러한 정보를 이용하여 사용자 인증을 위해 인증 블록을 호출한다. 인증 블록에서 실제의 인증 처리는 외부 보안 서버에 존재하는 AS와 PAS에서 처리된다. 즉 인증 처리 과정에서 인증 블록은 AS로부터 전달받은 PAC을 바탕으로 사용자 신임장(credentials)을 생성하며, 사용자 신임장은 추후에 보안 설정 서비스가 제공될 때 활용된다. 성공적으로 인증이 완료된 후에 사용자는 클라이언트 응용 프로그램을 수행시키고, 응용 프로그램은 서비스 요청을 클라이언트 호스트 ORB에 전달한다. 클라이언트 호스트의 ORB는 보안 문맥(secure context)를 생성하기 위해 보안 설정 블록을 호출하며, 보안 설정 블록에서 만든 보안 문맥을 서버 호스트의 ORB에 보안 토큰(security token) 형태로 전달한다. 클라이언트 요청을 받은 서버 호스트의 ORB는 서버에 존재하는 보안 설정 블록을 호출하여 서비스를 요청한 클라이언트와 보안 설정 처리를 수행하기 위한 서버 호스트의 보안 문맥을 생성하고,

외부 보안 서버에 존재하는 KDS로부터 키 정보를 수신한 후에, 그 결과를 클라이언트 호스트의 ORB에 보내면 클라이언트와 서버 사이에 안전한 보안 통신로의 설정이 완료된다. 이러한 안전한 보안 통신로가 설정되면, 클라이언트 호스트의 ORB는 클라이언트의 서비스 요청 메시지를 보안 설정 블록을 통해 암호화시켜 서버에 전달하며, 서버 호스트의 ORB는 서버의 보안 설정 블록을 통해 수신된 메시지를 해독하고, 요청 서비스가 서버 응용의 객체 서비스를 이용할 수 있는 지를 판단하기 위해 접근 제어 블록을 호출한다. 접근 제어 블록에서 접근 허용 결정이 이루어지면, 실제로 클라이언트의 서비스 요청 메시지가 서버의 응용 객체에 전달된다.

3. CORBA기반 보안 플랫폼의 설계

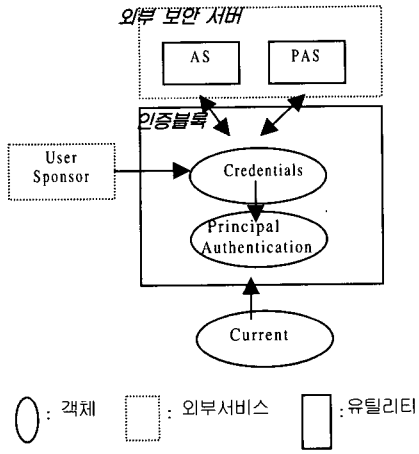
본 장은 SCAP의 구성 요소인 인증블록, 보안설정블록, 접근제어블록, 보안정보관리 블록에 대한 운영 알고리즘을 기술한다.

3.1 인증 블록[19]

사용자 인증은 클라이언트에 위치한 스폰서 프로그램이 PA 객체의 메소드를 호출함으로써 시작되며, PA 객체는 외부 보안 서버에 위치한 AS에 사용자 식별자와 암호 또는 비밀 키를 전달하면서 실제로 인증 처리를 요청하게 된다. 인증 요청을 받은 AS는 정당한 사용자인지를 확인한 후에 PAS 티켓과 사용자 비밀 키로 암호화되어 있는 세션 키(session key)를 반환한다. 만약, 사용자가 자신의 역할을 기술한 경우에, PAS에서 사용자에 한정된 권한 속성을 얻기 위해서 PA 객체는 AS에서 반환한 PAS 티켓과 세션 키를 사용하여 PAS에 권한 속성을 포함하는 PAC을 요청한다. 이런 단계가 마무리되면 PA 객체는 반환된 PAC을 기반으로 하나의 Credentials 객체를 생성한다.

위의 인증블록의 기능을 제공하기 위하여 (그림 2)와 같이 PA(principle authentication) 객체와 Credentials 객체로 구성된다[19].

Credentials 객체는 대행자의 보안 속성들 즉, 인증된 사용자 식별자, 권한, 보안 설정을 위한 정보를 유지하며, 또한 권한 정보를 접근할 수 있는 메소드를 제공한다. 사용자가 인증된 후에는 반드시 하나 이상의 신임장 객체가 존재해야 한다.

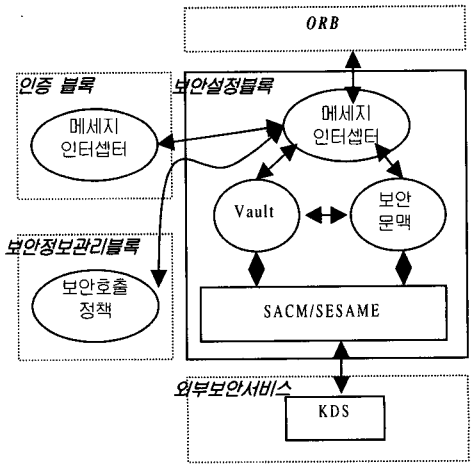


(그림 2) 인증 블록의 구조

이러한 Credentials 객체에 대한 참조를 위하여 Current 객체를 유지한다. Current 객체는 클라이언트와 서버 모두에 존재하며, 클라이언트 또는 서버의 현재 활동을 표현하는 객체이다. 따라서 이 객체는 현재 수행중인 활동에 결합된 신임장 객체를 접근하기 위한 인터페이스를 지원한다.

3.2 보안 설정 블록[18]

보안 설정 블록은 클라이언트/서버 사이의 보안 통신로를 설정하는 보안 문맥의 초기화와 관리 기능을 제공한다[18]. 다음에 있는 (그림 3)과 같이 보안설정 블록은 메시지 인터셉터(message interceptor) 객체,



(그림 3) 보안 설정 블록의 구조

Vault 객체, 보안 문맥 객체, 보안 설정 문맥 모듈(Secure Association Context Module : SACM)로 구성된다.

메시지 인터셉터는 ORB와 객체 서비스들 사이의 상호작용 기능을 제공한다. 이 인터셉터는 클라이언트가 ORB와 바인딩되는 시점에 객체 참조에 기술된 서버 응용 객체와 상호 보안 통신을 위해 보안 문맥을 설정한다. 따라서 클라이언트는 서버 객체 호출 시에는 서버로 전달되는 메시지를 보호하기 위해 바인딩시 설정했던 보안 문맥을 사용하기 위하여 이 객체를 호출한다. 또한 클라이언트와 마찬가지로 서버 호스트의 메시지 인터셉터도 보안 문맥을 설정하여 이를 활용한다.

클라이언트가 서버 객체에 대해 서비스를 요청할 때, 메시지 인터셉터는 이를 필터링하여 서버 객체와 서비스를 요청한 클라이언트 객체 사이에 이전에 보안 문맥이 설정되어 있는지를 검사한다. 보안 문맥이 존재하면 이를 활용하여 서버와 보안 통신이 수행 가능하며, 만일 존재치 않는 경우에는 인증 블록에 유지되는 신임장 객체와 보안 정보 관리 블록에서 관리하는 보안 설정 정책 정보를 이용하여 클라이언트와 서버 사이에 보안 설정을 수행한다.

클라이언트 메시지 인터셉터는 클라이언트와 서버 응용에서 보안 통신을 위해 요구되는 보호 수준 등과 같은 여러 보안 정보를 요청하기 위해 Valut 객체를 호출한다. 호출된 Vault 객체는 SACM을 통해서 Credentials 객체와 세션 키를 획득한다. 외부 보안 서버인 SESAME의 SACM은 보안 서버에 있는 KDS와 상호 연결되어 세션 키를 획득하고, 그리고 암호화 메시지 전송 또는 수신된 암호 메시지의 복호화 기능을 활용할 수 있는 GSS-API를 제공하는 모듈이다. Valut 객체는 Credentials 객체를 이용하여 새로운 보안 문맥 객체를 생성한 후, 서버로 보내질 보안 토큰과 생성된 보안 문맥 객체를 반환한다. 인터셉터는 보안 설정 메시지를 구성하는데, 이 메시지에는 보안 토큰이 포함된다. 서버 호스트에서 보안 설정 메시지를 받으면 서버호스트 메시지 인터셉터는 보안 토큰과 서비스 요청 메시지를 분리하고, 적절한 보안 문맥을 관리하는 서버호스트 Vault 객체를 이용하여 보안 토큰을 처리한다.

보안 설정이 이루어지고 난 후, 메시지 인터셉터는 현재 설정된 보안 문맥에 기술된 보호 수준에 따라서 서비스 요청과 반환 메시지에 대해 비밀성과 무결성을 제공하는 메시지 보호를 수행한다.

3.3 접근 제어 블록[17]

접근 제어 블록은 인가되지 않은 사용자가 서버 응용 객체에 정의된 연산들을 수행치 못하도록 방지하는 기능을 제공한다[17]. 인증 처리로 생성된 신임장 객체와 서버 응용 객체의 제어 정보인 도메인 접근 제어 정책, 요구 권리 정책, 그리고 연산을 수행하는 과정에 관련된 문맥 정보를 기반으로 접근 제어 블록은 사용자의 연산 수행 여부를 검사하는 접근 제어 기법을 실행한다.

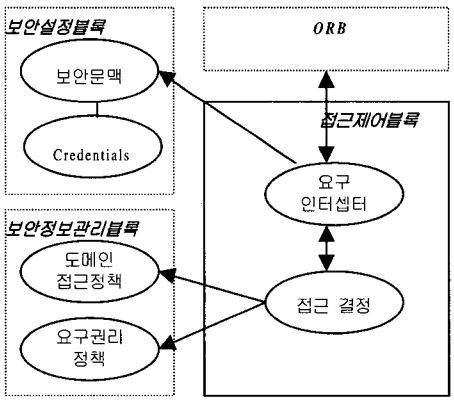
접근 제어 인터셉터는 클라이언트에서 서버 응용으로 전달되는 모든 요청 메시지를 중간에서 가로채고, 이 요청이 정당한지를 검사하기 위해서 접근 결정 객체의 접근 결정 메소드를 호출하는 일을 담당한다. 또한 접근 결정 객체의 결과에 따라서 연산 수행 요청 메시지를 서버 응용 객체로 전달하거나, 혹은 접근 불가로 인한 오류를 발생시켜 메시지를 반환시킨다.

접근 결정 객체에서 연산의 수행 여부를 결정하기 위해서는 무엇보다도 먼저 접근 제어 인터셉터가 서버 호스트에 생성되어 있는 보안 문맥 객체를 통하여 클라이언트 사용자에게서 받은 신임장에 기술된 사용자 권한 정보를 획득해야 한다. 이 정보가 접근 결정 객체에 전달되면 아래에 있는 접근 규칙을 이용하여 최종적인 접근허용 여부를 결정한다.

```
if (there is no DomainAccessPolicy object )
   permit request_access;
else {
   if ( user privileges are given )
   // search a tuple, which has user privilege in
   // DomainAccessPolicy;
   if ( there is no tuple )
      reject request_access;
      // get a granted rights from the tuple;
      // gr = the granted rights from the tuple;
      if ( there is no RequiredRights object )
         permit request_access;
      else {
         // search a tuple, which includes the interface
         // and the operation that the requestor want to
         // access in RequiredRights object;
         if ( found ) {
            // rr = the required rights;
            // get the right_combinator in the tuple;
            switch ( right_combinator ) {
               case SecAllRights :
                  if ( gr contains all element of rr )
                     permit request_access;
                  else
```

```
                reject request_access;
            case SecAllRights :
                if ( gr contains at least an element of rr
                )
                    permit request_access;
                else
                    reject request_access;
            }// switch statement
        }
    else
        reject request_access
    } // fi RequiredRights
}// fi DomainAccessPolicy
```

위의 접근제어블록의 기능을 제공하기 위하여 (그림 4)와 같이 접근 제어 인터셉터 객체와 접근 결정 객체로 구성된다[19].



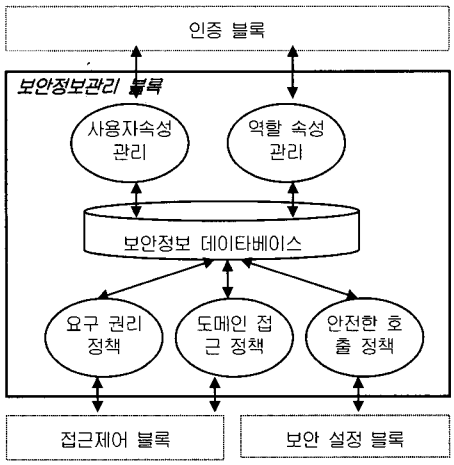
(그림 4) 접근 제어 블록의 구조

CORBA 보안 서비스 표준에 의하면 접근 제어 기능은 서버 객체 호출 시 ORB 수준에서 시행 또는 응용 프로그램 자체에 의해서 수행될 수 있지만, SCAP의 접근 제어 기능은 서버 응용 객체에 위치한 ORB에서 자동적으로 접근 제어가 수행되도록 설계한다. 또한 접근 제어 블록에서 접근 결정시에 이용하는 접근 규칙은 POSIX.6[10]을 기반으로 한다. 결과적으로 이 블록의 접근 제어 기능은 사용자의 권한, 서버 도메인의 보안 정책과 요구 권리 정책, 그리고 POSIX.6 접근 규칙을 바탕으로 서버 호스트의 ORB에서 자동적으로 시행되는 특징을 갖는다.

3.4 보안 정보 관리 블록[16]

보안 정보 관리 블록의 기능은 두 가지 부류로 구분되는데, 첫번째는 보안 관리자가 보안 정책을 설정하고 관리하기 위한 기능이며, 두 번째는 다른 블록들에게 보안 정책 정보를 제공하는 기능이다[16].

(그림 5)와 같이 이 블록은 세가지 부분으로 구성된다.



(그림 5) 보안 정보 관리 블록의 구조

사용자 인증 정보, 사용자 권한 및 역할 권한 정보 등과 같은 보안 정보는 외부 보안 서버에 존재하는 보안 정보 데이터베이스에서 관리된다.

사용자 정보 관리 모듈은 인증 블록에서 이용하는 사용자 인증 정보, 사용자 속성, 역할 속성 등을 관리하는데 필요한 인터페이스를 제공한다. 보안 정책 관리 모듈은 다시 세 부분으로 나누어지는데, 각기 요구 권리 정책, 도메인 보안 정책, 보안 설정 정책을 관리하기 위한 부분들이다. 요구 권리와 도메인 보안 정책은 접근 제어 블록에서 접근 결정을 위해 필요한 정보이며, 보안 설정 정책은 보안 설정 블록에서 보안 통신 시에 메시지 보호의 수준을 결정하는 필요한 정보이다.

보안 정책 정보를 관리하는 사용자는 반드시 보안 관리자로써 시스템에 로그인하여, SCAP 제어 프로그램을 수행시켜야 한다. 제어 프로그램은 도메인 정의, 도메인의 구성원 관리, 사용자 권한 속성과 역할 권한 속성, 그리고 보안 정책들을 관리하는데 필요한 기능들을 제공하며, 메뉴 방식의 명령어 인터페이스로 이루어져 있다. 보안 관리자는 이 제어 프로그램을 통해서만 보안 정보 데이터베이스를 새롭게 생성, 수정, 삭제할 수 있다.

4. 구 현

본 장에서는 SCAP의 각 기능 블록을 구현하는데 있어 구현상의 제기된 문제점과 이를 해결하는 방안 및 구현시의 고려 사항들을 살펴보며, 구현된 SCAP 보안 플랫폼과 다른 보안 플랫폼과의 비교 분석하고자 한다.

4.1 구현 환경

시스템은 Ultra Sun Sparkstation의 Solaris2.5에서 Sun C++ 컴파일러에 의해 개발되었다. SCAP을 구현함에 있어 CORBA 보안 규격에서 제시하고 있는 외부 보안 기술은 SESAME V4의 제삼자 서버를 이용하였으며, SCAP의 각 구성 요소들에 대한 통신 통로는 IONA사의 Orbix 2.1 MT[11]를 이용하였다.

4.2 SCAP 구현 시 고려 사항

CORBA 보안 규격에서 제시하고 있는 기능 수준 1을 제공하기 위해서는 인터셉터 기능을 지원하는 CORBA 규격을 따르는 통신 통로와 신임장과 같은 정보를 갖고 있는 Current 객체를 지원해야 한다.

이를 위한 구현 환경으로 CORBA 버전 2.0을 기반으로 한 ORB 제품으로 IONA사의 Orbix 2.1 MT를 선택하였다. 인터셉터는 하나 이상의 ORB 서비스 수행을 담당하고 있으며, 논리적으로 보안설정블록과 접근제어블록에서의 클라이언트 객체와 목적 객체 사이의 호출 통로를 놓여 지는데, 구현을 위해 채택한 Orbix 제품은 CORBA 보안 서비스를 제공하는데 필요한 인터셉터 객체를 지원하지 않는 대신에 Filter와 Transformer라는 인터셉터와 유사한 기능을 제공하고 있다. Filter는 CORBA의 요구 수준(request level)의 인터셉터와 유사하며, Transformer는 메시지 수준(message level)의 인터셉터이다.

따라서 본 논문에서는 인터셉터 기능을 지원하기 위하여 Orbix의 Filter와 Transformer를 이용하였으며, 보안설정블록을 위한 MessageFilter와 Message-Transformer라는 보안 인터셉터와 접근제어블록의 RequestInterceptor-Filter와 RequestInterceptor를 구현하였으며, 구현된 클래스들은 다음과 같다.

- MessageFilter
  통신 주체간에 보안 세션을 초기화하기 위한 요구를 가로채어 보안문맥을 생성하는 Vaulr 객체에게 넘겨준다.

- MessageTransformer
  전송되는 메시지를 가로채어 보안 통신을 보장하기 위해 SecurityContext 객체에게 전달한다.

- RequestInterceptorFilter
  응용 서버에 대한 모든 요구를 객체 요구 중개자 내부에서 필터링한다.

- RequestInterceptor
  서버쪽에 생성되어 있는 보안문맥 객체가 관리하는 사용자의 권한을 요청하고, 반환된 권한을 입력변수로 하여 접근 허용 여부를 수행하는 메소드를 호출한다.

보안 설정 블록을 위한 메시지 인터셉터와 접근 제어 블록을 위한 요구 인터셉터는 이 필터 기법을 통하여 시작될 수 있다. 필터 기법을 이용하여 ORB와 상호동작하는 외부 인터셉터는 마치 ORB에 내장된 인터셉터처럼 보이게 된다.

또한 구현 시 다른 고려사항은 Current 객체는 CORBA 보안 규격에 따라 클라이언트 객체와 목적 객체에서 수행 문맥에 대한 정보를 가지고 있어야만 한다. 특히 이것은 신임장과 같은 보안 정보를 가지고 있으며, 이 정보를 접근할 때 이용된다. 그러나 현재의 Orbix 버전은 Current 객체를 지원하지 않는다. 따라서 이를 해결하기 위해 Current 객체는 필요할 때마다 외부적으로 생성시키도록 하였다.

4.3 구현 검증

SCAP 구현에 대한 검증은 사용자 인증, 응용에 대한 사용자 접근 제어 기능, 보안 정책 및 사용자의 보안 정보를 관리하는 기능이 정해진 규격대로 동작하는지를 확인하는 작업을 수행하였다.

검증을 수행하기 위하여 통신망으로 두 대의 워크스테이션과 사용자의 PC 환경을 구성하였다. 시험 환경의 논리적인 구성은 응용 클라이언트 시스템, 응용 서버 및 보안 서비스를 제공하는 SCAP 서버 시스템을 위치하도록 구성하였다. 또한 응용 클라이언트와 응용 서버 시스템은 IONA Orbix 2.1MT를 설치하여 통신통로로 이용하였다.

구현된 SCAP 기능 수행의 정확성을 검증하기 위한 검증항목과 수행결과를 요약하면 <표 1>과 같다.

〈표 1〉 기능 검증항목 및 수행 결과

검증대상	검증항목	검증 결과
인증	등록되지 않은 사용자	인증 오류
	등록된 사용자	인증 성공
	사용자 권한 정보 획득	식별자 권한과 역할권한 획득
보안 문맥 설정	단순 보안 문맥 생성	보안 문맥 생성 성공
	암호화 처리	암호화된 문자열 판독 불가
	다중 보안 문맥 생성	다중문맥 생성 성공
접근 제어	접근 제어 정책 부재	인증된 사용자 문서 서버 접근 불가
	접근 제어 정책 존재	인증된 사용자 문서 서버 모두 접근 허용

4.4 SCAP과 다른 보안 서비스 플랫폼과의 비교

본 절은 본 논문에서 설계한 SCAP과 다른 보안 서비스 플랫폼과의 기능적 측면에서 비교한 결과를 설명한다.

〈표 2〉 SCAP과 여러 다른 플랫폼과의 비교

	SCAP	DCE	POSIX	SESAME
관련 단체	ETRI	OSF	ISO, IEEE	ECMA
모델	객체지향 모델	절차적	운영체제	절차적
특성	구조적	보안서버	계층적	보안서
인증 기법	기법에 독립	Kerberos V5	기법에 독립	Kerberos V5
접근 제어 기법	push/pull	push	push/pull	Push
암호 기법	기법에 독립	DES, MD5 등	기법에 독립	RSA, MD5 등
키	독립	대칭적	대칭적, 비대칭적	대칭적, 비대칭적
감사 기능	있음		있음	있음
위임 기능	있음	있음	있음	있음
부인 봉쇄 기능	있음		있음	있음
인 터 페이스	ORB보안 객체	안전한 RPC, GSS-API	POSIX-API	Extended GSS-API

<표 2>은 CORBA 보안 규격을 따른 SCAP을 상용의 분산처리 시스템을 제공하는 DCE 보안 서비스, 안전한 개방형 환경에 있어서 보안 플랫폼과 인터페이스를 제시한 POSIX, ECMA 프로젝트 하에 개발된 SESAME를 여러 관점에서 비교한 것이다.

5. SCAP 응용

인터넷을 기반으로 하는 웹 서버와 정보제공자는 분산객체처리 환경을 포함하는 이기종 환경에서 운용되고 있으므로 일반 사용자 또는 웹 브라우저는 분산객체 응용에 대한 실제 통로로써 IIOP(Inter Internet-ORB Protocol)을 이용하여 웹 환경을 접근할 수 있다. 또한 인터넷을 기반으로 하는 클라이언트와 서버 사이에 상호 교환되는 데이터의 무결성과 비밀성을 보장하며, 정당한 사용자임을 증명하기 위한 인증 기능이 절대적으로 필요하며, 이를 제공하기 위하여 데이터 교환 시에 데이터에 대한 안전성을 제공하는 통신 프로토콜인 SSL(Secure Socket Layer)을 이용하거나[12] 또는 응용 수준에서 안정성을 전담하는 제삼자 서버(Trusted Third Parties)를 이용하여 안전성을 제공하고 있다[13, 14].

따라서 본 장은 보안 통신 프로토콜이 변경됨에 따라 웹 도구들의 변경되어야 하는 문제점을 방지하기 위해 SCAP을 기반으로 한 응용 수준에서 웹과 정보제공자 사이에서 분산객체 기반의 응용들에 대해 보안 서비스를 제공하고자 하였다.

5.1 요구사항

웹 서버와 정보 제공자 사이의 보안 서비스를 제공하기 위하여 다음과 같은 요구사항을 고려하였다.

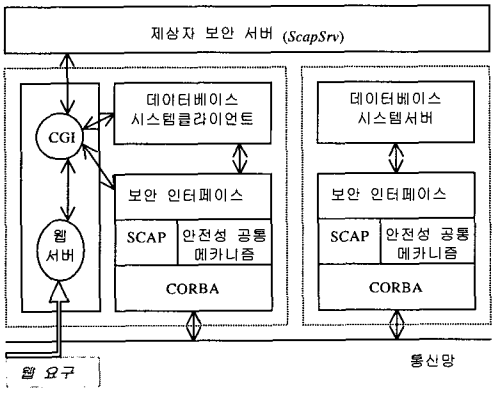
- 안전한 웹과 데이터베이스 연동을 위하여 기존의 웹 서버의 기능 및 브라우저와 HTTP 통신 규약의 성장에 따른 부담을 가능한 회피하기 위하여 웹 서버 및 브라우저를 변경하지않고 그대로 사용하기 위하여 CGI를 기반으로 하는 응용 서버 방식을 추구한다.

- 프로세스의 생성 소멸을 가능한 최소화하기 위하여 데이터베이스 관리 시스템의 서버와 응용을 데몬 프로세스로 두도록 한다.

- 안전성 기능을 지원하는 제삼자 서버 방식

5.2 구 조

웹 서버는 웹 클라이언트의 요구를 분석한다. 만약 데이터베이스 접근 요구이면, 데이터베이스 시스템 클라이언트는 데이터베이스 시스템 서버에게 이를 전달하고, 그 처리 결과를 웹 서버에 전달 및 통신상에서 암호화된 정보로 대체함으로써 통신상에서 보안 서비스를 제공한다. 이를 위한 구조는 (그림 6)과 같다.

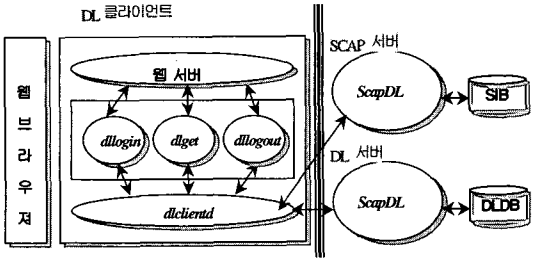


(그림 6) 안전한 웹과 데이터베이스 시스템 구조

(그림 6) 구조는 객체의 기본적인 표현과 서비스 요구에 대한 기본적인 통신을 제공하는 CORBA를 기반으로 하여 웹 서버와 데이터베이스 시스템 서버와의 안전한 보안 통로 설정을 위한 키 생성, 데이터 기밀성 또는 무결성을 제공하기 위한 메시지 보호, 접근 제어 등의 기능을 제공하는 SCAP 보안플랫폼을 이용하였다[5, 7]. 보안 인터페이스는 데이터베이스 클라이언트와 사용자 인증을 위한 CGI에 사용하였다.

5.3 구 현

(그림 7)은 이전의 안전한 웹 서버과 데이터베이스 시스템 구조에서 정보 이용자의 웹 브라우저를 통해 필요한 정보를 얻는 과정을 나타낸 것이다.



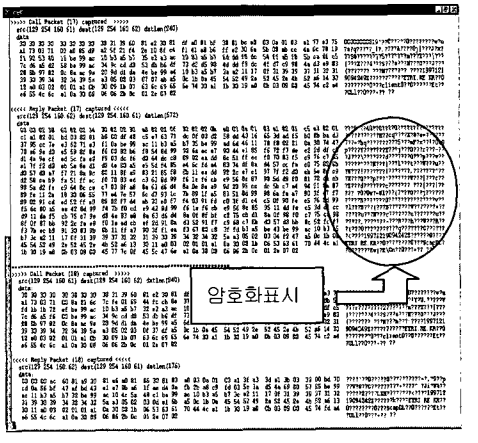
(그림 7) 안전한 웹과 정보제공자 시스템 동작

웹 서버와 정보제공자 사이의 보안 서비스를 위하여 특징적인 주요 CGI는 다음과 같다.

- 사용자 로그인(*dllogin*) CGI
  데이터베이스 시스템의 클라이언트 데몬에 사용자의 로그인을 알림
- 데이터베이스 자료 요구(*dlget*) CGI
- 사용자 로그아웃(*dllogout*) CGI
  데이터베이스 시스템의 클라이언트에 사용자 로그아웃을 알림

정보 이용자는 웹 브라우저를 통하여 사용자 로그인 및 로그아웃을 요구하면, *dllogin* 및 *dllogout*은 FIFO를 이용하여 데이터베이스 클라이언트 데몬에게 사용자 인증을 요구한다. *dlget*은 웹 서버의 서비스 요구문을 데이터베이스시스템 클라이언트에게 FIFO를 이용하여 전달하며, 그 결과 값을 FIFO로 받아 웹 서버에 전달한다. 자료를 전달할 때에 사용되는 FIFO 이름은 데이터베이스 시스템 클라이언트의 프로세스 번호를 사용하였다. 데이터베이스 클라이언트 데몬(*dlclientd*)은 자신을 복제하여 자식 프로세스를 생성 후 요구 유형에 따라 제삼자 보안 서버 또는 데이터베이스 시스템 서버와 접촉한다. 만약 사용자 로그인이 성공하면 데이터베이스 시스템 서버와의 연결을 시도한다.

5.4 수행 결과



(그림 8) 암호화 여부를 나타내는 모니터링 화면

두 시스템 사이에 전달되는 메시지들을 조사함으로써 암호화 여부를 확인할 수 있도록 하였다.

(그림 8)은 SCAP의 인터셉터를 적절하게 호출하여 메시지를 암호화하고 복호화하는 기능에 대한 수행 결과이다. SCAP 관리자는 자료의 기밀성과 무결성을 보장하기 위하여 SCAP 관리 프로그램을 이용하여 보안 서비스 종류를 미리 정하도록 하였다. 따라서 메시지의 기밀성과 무결성 제공 여부는 모니터링 프로그램을 수행함으로써 파악할 수 있도록 하였다.

6. 결 론

본 논문에서는 CORBA를 기반으로 실행되는 응용들에 대한 보안성을 지원하는 보안 플랫폼을 제안하고, 이를 설계 구현하였다. 구현된 보안 플랫폼은 인트라넷 또는 인터넷 통신 환경과 같은 분산 객체 시스템이 내재하고 있는 다양한 보안 위협들에 효과적으로 대처할 수 있는 CORBA 보안 서비스 표준을 따르고 있으며, 크게 네 가지의 기능 블록들(인증, 보안 설정, 접근 제어, 보안 정보 관리)로 구성하였다. 그리고 제시된 보안 플랫폼은 세상에서 널리 이용되고 있는 IONA Orbix CORBA 제품과 보안 서버인 SESAME V4를 구현 환경으로 활용하였다.

추후에는 현재 구현된 버전의 SCAP에서 메시지 인터셉터와 접근 제어 인터셉터를 ORB 외부에 위치시켰으나, 이를 ORB 내부로 이전시키기 위한 문제와 플랫폼의 개발 전략상 표준 CORBA 보안 서비스에는 정의되어 있으나 현재 구현 버전에서 생략되어 있는 권한 위임, 부인 봉쇄, 보안 감사 등의 기능들을 보강할 예정이다.

참 고 문 헌

[1] Object Management Group, *CORBA Security*, OMG Document Number 95.12.1, Dec., 1995.

[2] R.Oppliger, "*Authentication Systems for Secure Networks*," Artech Hous, pp.29-62 and pp.115-142, 1996.

[3] IEEE, *Draft Guide to the POSIX Open System Environment : A Security Framework*, IEEE Doc. Number No.13, Aug., 1995.

[4] OSF, *Open Software Foundation Training Course*, OSF DCE System Administration Course, Student Guide, Vol.1.0, Dec., 1992.

[5] T.Parker and D.Pinkas, *SESAME V4 Overview*, SESAME Issue 1, Dec., 1995.

[6] OMG, *The Common Object Request Broker : Architecture and Specification*, 2.0ed., Jul., 1995.

[7] OMG Security Working Group, "*OMG White Paper on Security*," OMG Doc. No.94.4.16, April, 1994.

[8] J.Linn, "*Generic Security Service Application Programming Interface*," Internet RFC1508, Sep., 1993.

[9] OMG, *Common Secure Interoperability Specification*, OMG Doc. No.orbos/96.06.20, Jun., 1996.

[10] POSIX, "*Protection, Audit, and Control Interface*," IEEE P1003.6.1, 1995.

[11] IONA, *Orbix Reference Guide*, IONA Technologies Ltd., 1997.

[12] Fredric J. Hirsch, "*Introducing SSL and Certificates using SSLeay*," http://www.camb.opengroup.org/RI/www/prism/www

[13] OSF, *Web-based WandD server administration*, 1997.

[14] Simens Nixdorf Company, *TrustedWeb Deployment Guide Ver.1.1*, 1997.

[15] Choi,Rak Man, Na, Jung Chan, Lee, Kwon Il, and Han Woo Young, "*Design of Security Platform for CORBA based Application*," Lecture Notes in Computer Science, Vol. 1334, pp.980198, 1997.

[16] Na, Jung Chan, Kim, Eun Mi, and Ine So Ran, "*An Implementation of Security Information Manager for SCAP*," Proc. Of the Asia-Pacific Network Operations and Management Symposium, pp 177-184, 1997.

[17] 김영균, 송영기, 인소란, "*CORBA 응용을 지원하는 보안 플랫폼의 접근 제어 기능 구현*", 한국정보과학회 추계학술 발표 논문집 24권 2호, pp.423-426, 1997.

[18] 이권일, 송영기, 인소란, "*CORBA 환경에서 보안 설정 기능의 설계 및 구현*", 한국정보과학회 추계학술 발표 논문집, pp.1360-1364, 1997.

[19] 한우용, 최락만, "*CORBA 인증 객체의 설계 및 구현*", 제1회 개방형 보안 기술과 응용 워크샵, 1997.

나 중 찬

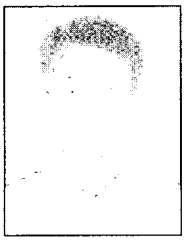
e-mail : njc@etri.re.kr

1986년 충남대학교 계산통계학과 학사

1989년 숭실대학교 전산학과 석사

1989년~현재 한국전자통신연구원 컴퓨터소프트웨어연구소 선임연구원

관심분야 : 실시간 시스템, 분산시스템, 인터넷 기반 S/W



김 영 균

e-mail : kimyoung@etri.re.kr

1991년 전남대학교 전산통계학과 학사

1993년 전남대학교 전산통계학과 이학석사

1995년 전남대학교 전산통계학과 이학박사

1989년~현재 한국전자통신연구원 컴퓨터소프트웨어연구소 선임연구원

관심분야 : 데이터베이스 시스템, 정보보호, 인터넷 기반 S/W

김 경 범

e-mail : kbkim@etri.re.kr

1981년 인하대학교 전자공학과 학사

1983년 인하대학교 전자공학과 석사

1983년~현재 한국전자통신연구원 컴퓨터소프트웨어연구소 책임연구원, 전자계산기 기술사

관심분야 : 분산 시스템, 정보 보호, 분산시스템 관리, 인터넷 기반 S/W

김 명 준

e-mail : joonkim@etri.re.kr

1978년 서울대학교 계산통계학과 학사

1980년 한국과학기술원 전산학 석사

1986년 프랑스 낭시(Nancy) 제1대학교 응용수학과 및 전산학과 박사

1980년~1981년 아주대학교 종합연구소 연구원

1981년~1986년 프랑스 낭시 전산학 연구소(CRIN) 연구원

1986년~현재 한국전자통신연구원 컴퓨터소프트웨어연구소 인터넷서비스연구부장

관심분야 : 데이터베이스 시스템, 트랜잭션 처리, 분산 시스템, 인터넷 기반 S/W