The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 16, No. 6, pp.55-63, Dec. 31, 2016. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2016.16.6.55
JIIBC 2016-6-7

SOA 기반 ESB 환경에서 내부 종단 사용자 위협 대응을 위한 보안 아키텍처 제안

Security Architecture Proposal for Threat Response of insider in SOA-based ESB Environment

오시화*, 김인석**

Shi-hwa Oh*, In-seok Kim**

요 약 많은 기업에서 방대한 데이터를 안정적으로 처리하고 업무시스템을 통합하기 위하여 SOA(service oriented architecture) 기반의 ESB(enterprise service bus) 모델을 적용하고 있다. 그러나 SOA 구축을 위한 기존 웹 서비스 기술은 안전하게 데이터를 교환하기에는 한계가 있어 웹 서비스 보안 기술의 표준화가 진행되고 있지만, 실질적인 적용이 미흡한 상황이다. 이와 같은 환경으로 구축된 대규모 업무시스템을 사용하는 내부의 종단 사용자는 다양한 조직과 역할로 구성된다. 종단 사용자가 규정된 일정한 권한을 넘어 인가되지 않은 정보를 취득하여 개인의 이익이나 악의적인 목적으로 이용하고자 하는 경우 기업은 외부의 공격보다 더 큰 피해를 입을 수 있다. 본 논문은 종단 사용자가 이용하는 웹 서비스 기술의 보안 위협을 식별하여 대응 할 수 있는 보안 아키텍처를 제안하고자 한다.

Abstract SOA(service oriented architecture) based ESB(enterprise service bus) model is widely adopted in many companies for the safe processing of enormous data and the integration of business system. The existing web service technologies for the construction of SOA, however, show unsatisfactory in practical applications though the standardization of web service security technologies is in progress due to their limitations in safe exchange of data. Internal end users using a large business system based on such environment are composed of the variety of organizations and roles. Companies might receive more serious damage from insider threat than that from external one when internal end users get unauthorized information beyond the limits of their authority for private profit and bad purposes. In this paper, we propose a security architecture capable of identifying and coping with the security threats of web service technologies arouse from internal end users.

Key Words : SOA, ESB, insider threat, security architecture

Ⅰ. 서 론

최근 많은 기업에서 회사 전체에 걸쳐 분리되어 운용되고 있던 시스템을 통합하고 방대한 데이터를 조합하여 새로운 비즈니스 정보를 생성하는 방향으로 구축되고 있다. 이와 같은 요구사항을 수용하기 위하여 SOA(service oriented architecture) 기반의 ESB (enterprise service bus) 모델 기술이 적용되고 있다. 분할된 업무시스템 조

*정회원, 고려대학교 정보보호대학원 정보보호학과
**정회원, 고려대학교 정보보호대학원(교신저자)
접수일자: 2016년 10월 23일, 수정완료: 2016년 11월 23일
게재확정일자: 2016년 12월 9일

Received: 23 October, 2016 / Revised: 23 November, 2016
Accepted: 9 December, 2016
**Corresponding Author: iskim11@korea.ac.kr
Center for Information Security Technologies(CIST), Korea University, Korea

각 단위들을 느슨하게 연결(loosely coupled)해 완성된 업무시스템을 만드는 SOA의 주요 특징으로 유연성과 재사용성을 들 수 있다[1]. 이러한 SOA를 구현하고자 하는 기업들 사이에서, 연계 주체 간 데이터 지향, 이벤트 구동 및 서비스 품질을 보장하며 상호작용을 지원할 수 있는 인프라로 ESB가 유용하게 사용되고 있다[2].

내부자에 의한 보안사고는 지속적으로 발생하고 있으며 국내에서는 옥션, GS 칼텍스, 농협, KT 등 대기업조차 피해를 입고 있다. 이러한 보안사고는 특히, 통합된 업무시스템을 사용하는 내부사용자에 의하여 발생하는 경우 정보유출의 피해 규모가 매우 커질 수 있다. CERT의 2011 CyberSecurity Watch Survey에 따르면, 내부자에 의한 피해가 외부자에 의한 것보다 많은 것을 알 수 있다. 내부자 사고의 유형은 인증되지 않은 접속에 의한 기업 정보 손상이 63%로 가장 높으며, 고의적이지 않은 기업 정보 유출이 57%로 그 다음으로 많은 비율을 차지하고 있다[3]. 두 유형 모두 접근권한 및 계정관리, 정보관리에 관련된 것으로 내부자의 행위를 관리하는 것과 연관된 것이다. 따라서 기업이 준비하고 있는 SOA 기반의 업무시스템 구축에서 고려해야 할 중요한 요소는, SOA를 구현하는 웹 서비스의 보안 취약점을 이용한 내부 사용자의 정보유출 가능성을 예측하고 이에 대응할 수 있는 조치를 취하는 것이다.

본 논문에서는 SOA를 구현하는 ESB 인프라 환경을 이용하여 다수의 내부 사용자에 의한 정보유출 사고를 방지할 수 있도록 보안 아키텍처를 구성하는 방안을 제안하고자 한다.

II. 관련 연구

1. SOA 및 웹 서비스

SOA는 느슨한 결합의 (loosely coupled), 상호연동 할 수 있는(interoperable) 서비스들의 조합으로 어플리케이션 개발을 가능하게 하는 정보시스템 아키텍처이다. SOA는 다양한 서비스로 구성되어 있으며 개방된 표준을 사용하여 서비스간의 의사소통을 하는 아키텍처로 볼 수 있다. 현재 SOA를 구현하기 위한 기술로 가장 주목 받고 있는 것이 웹 서비스이다.

웹 서비스는 서비스 지향 원리를 웹 기반으로 구현한 것이다. 웹 서비스는 네트워크 및 관련 표준을 통해 운영체계(OS: operating system) 및 프로그램 언어에 상관없이 상호운영이 가능하도록 해주는 표준화된 소프트웨어 기술이다. 웹 서비스는 다음의 특징을 갖는다[4].

- 플랫폼 독립적이다 : 웹 서비스는 유연한 어플리케이션 구조를 가지고 있기 때문에 서비스 공급자나 수요자가 특별한 기능을 추가하기 위해 새로운 플랫폼을 사용하지 않아도 되며, 플랫폼 선택도 자유롭다.
- 디바이스 및 위치 독립적이다 : 웹 서비스를 통해 PC, PDA, 핸드폰 등 다양한 유무선 디바이스를 통해 시간 및 장소에 상관없이 웹 서비스에 접근이 가능하다.
- 동적인 기능(dynamic function)이다 : 다양한 기능들을 적절한 서비스 제공자로부터 찾을 수 있고, 실시간으로 연계시킬 수 있으며, 서비스 제공자와 수요자의 역할이 고정되어 있지 않다.
- 상호 운용성을 제공한다 : 상호 작용하는 서비스는 표준화된 다양한 메시지 교환 형태를 이용하여 인터페이스하기 때문에 이기종 환경에서 상호운용성을 제공할 수 있다.

웹 서비스는 “서비스 제공자(Service Provider)”, “서비스 사용자(Service Consumer)”, “서비스 등록저장소(Service Broker)” 로 구성되어 있다. 서비스 제공자는 생성된 자원을 개방형 표준에 의해 서비스화 하여 등록저장소에 공개한다[5].

웹 서비스는 SOAP, WSDL, UDDI 등의 주요 표준 기술로 이루어진다. 웹 서비스의 메시지는 주로 XML (extensible mark-up langage)이 사용된다. SOAP (simple object access protocol)은 XML 프로토콜 표준으로 Web Services의 요청 및 응답에서 사용되는 XML 메시지 형식과 통신 프로토콜과의 바인딩 방법을 정의하고 있다. XML과 HTTP(hyper text transfer protocol) 등을 기본으로 하여 다른 컴퓨터에 있는 데이터나 서비스를 호출하기 위한 통신규약이다. WSDL (web service description language)은 웹 서비스에서 제공하는 기능들을 외부에서 이용할 수 있도록 그 사용방법을 알려주는 인터페이스 언어로 XML 기반으로 작성된다. 웹서비스가 수행하는 작업, 호출 가능한 메소드, 메소드에 전달해야 하는 파라미터 타입, 사용되는 바인딩 프로토콜 등 기술정보들을 정의한다. UDDI (universal description, discovery, and integration)는 개방형 표준과 비독점적 기술을 기반으로 개발된 전역 비즈니스 레지스토리이다.

이 레지스토리를 이용하여 다양한 웹서비스를 쉽게 검색하여 사용할 수 있다[6]. 연계주체 간 데이터 지향과 이와 관련된 표준 기술의 관계를 보여주는 웹 서비스 아키텍처를 그림 1에 나타내었다.

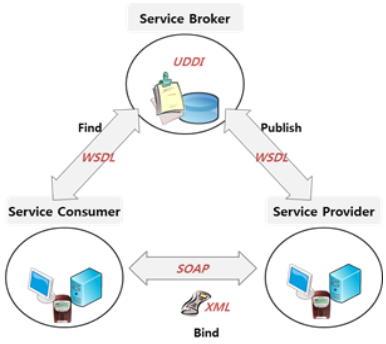

그림 1. 웹 서비스 아키텍처
Fig. 1. Web service architecture

2. 웹 서비스 보안

그림 2.는 MS와 IBM에서 제안하고 표준화를 위해 작업 중인 서비스 보안 기술의 체계도(roadmap)이다[4].

WS-Security(web service security)는 SOAP 메시지에 대한 어플리케이션 단계 보안에 대한 내용을 기술하고 있는 것으로 바이너리 보안 토큰을 인코딩하는 방법과 X.509 인증서나 커버러스(Kerberos) 티켓 등을 사용하는 방식 등을 정의하고 있다[7][8].

그림 2 에서 서비스 보안 기술로 XML Signature[9], XML Encryption[10] 스펙을 포함하여 표준 기술을 정의하고 있다.

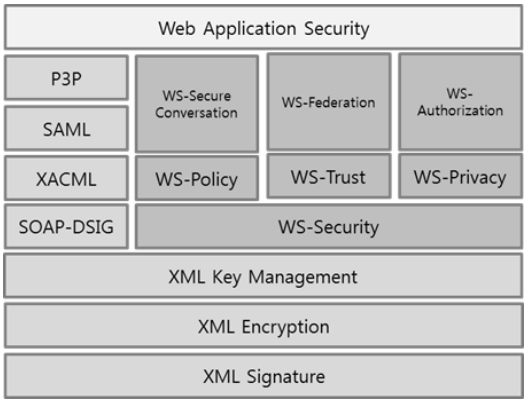

그림 2. 웹 서비스 보안 기술 로드맵
Fig. 2. Web service security specification roadmap

XML Signature

메시지의 무결성은 XML Signature를 통해 보장된다. SOAP 내부의 특정 부분에 저장하여 이에 송신자의 전자서명을 계산하고 전송하는 방법으로 메시지의 무결성을 보장한다. 대상이 되는 부분의 해쉬 함수 계산결과와 이를 송신자의 비밀키를 이용하여 계산한 전자서명 결과를 함께 포함한다. 그림 3에 XML Signature의 예를 나타내었다.

```
<Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo>
  <CanonicalizationMethod Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/>
  <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/">
    <Transforms>
      <Transform Algorithm="http://www.w3.org/2006/12/xml-c14n11"/>
    </Transforms>
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
    <DigestValue>dGhpcyBpcyBub3QgYSBzaWduYXR1cmUK.../DigestValue>
  </Reference>
</SignedInfo>
  <SignatureValue>...</SignatureValue>
  <KeyInfo>
   <KeyValue>
     <DSAKeyValue>
       <P>...</P><Q>...</Q><G>...</G><Y>...</Y>
     </DSAKeyValue>
   </KeyValue>
  </KeyInfo>
</Signature>
```

그림 3. XML 전자서명 예제
Fig. 3. XML signature example

XML Encryption

메시지의 기밀성 보장은 XML Encryption을 통해 보장된다. 그림 4 에 예로 보인 XML Encryption 기술은 메시지를 암호화하고 표현하는 방법을 명시한다. 암호문을 사용하여 이루어지며, 송신자 전송 전에 메시지를 암호화 하고 수신자가 공유키 정보를 사용해서 이것을 해독하게 된다.

Non-encrypting an XML Element	<?xml version='1.0'?> <PaymentInfo xmlns='http://example.org/paymentv2'> <Name>John Smith</Name> <CreditCard Limit='5,000' Currency='USD'> <Number>4019 2445 0277 5567</Number> <Issuer>Example Bank</Issuer> <Expiration>04/02</Expiration> </CreditCard> </PaymentInfo>
Encrypting an XML Element	<?xml version='1.0'?> <PaymentInfo xmlns='http://example.org/paymentv2'> <Name>John Smith</Name> <EncryptedData Type='http://www.w3.org/2001/04/xmlenc#Element' xmlns='http://www.w3.org/2001/04/xmlenc#'> <CipherData> <CipherValue>A23B45C56</CipherValue> </CipherData> </EncryptedData> </PaymentInfo>

그림 4. XML 암호화 데이터 예제
Fig. 4. XML encryption data example

3. ESB

ESB는 표준 웹서비스 기반 동적 라우팅 및 변환 기술을 기반으로 SOA를 지원하는 미들웨어 플랫폼이다. 서비스, 응용 프로그램 및 자원을 통합하고 연결하는 미들웨어의 새로운 패턴이며 SOA의 핵심부분이다. ESB 아키텍처는 버스에 기반을 두고 있는데 버스는 SOAP, HTTP 및 JMS(java message service) 등 표준을 기반으로 데이터 배달 서비스를 제공 한다[11]. 소프트웨어 서비스와 어플리케이션 컴포넌트 간의 연동을 위한 백본 역할을 수행한다. ESB는 보안 기능으로 메시지 전송 시 보안기능, 전자서명, SOAP메시지 암호화 기능, 사용자 인증, 서비스 권한관리, 사용자 인증시스템 연동 기능을 제공한다[12]. 이러한 ESB의 논리 구조를 그림 5에 나타내었다.

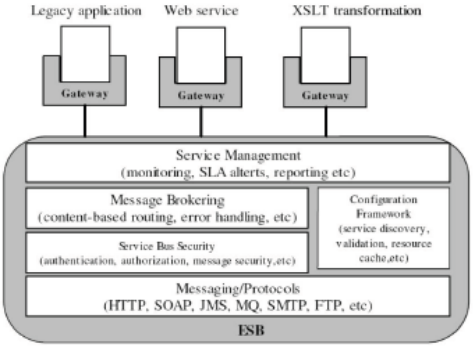

그림 5. ESB 논리 아키텍처
Fig. 5. ESB logical architecture

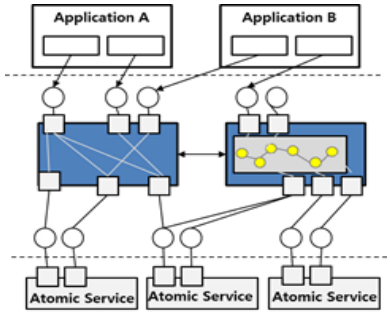

그림 6. ESB 기반의 SOA 어플리케이션
Fig. 6. SOA application based ESB

ESB 기반 SOA Application은 XML 기반의 메시지를 각 컴포넌트에 전송하여 처리가 가능하다. ESB 기반 SOA Application에 대한 설명은 그림 6과 같다. ESB는 Function 중심의 단위 서비스를 연결 조합하게 되고, 이러한 ESB를 포함하는 Application A, B, C 등은 SOA Application (복합 서비스)의 형태로 동작하게 된다[4].

4. TLS/SSL

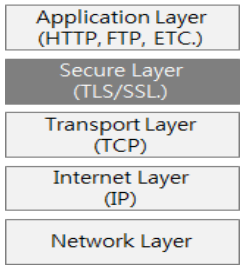

그림 7. TLS/SSL 프로토콜
Fig. 7. TLS/SSL protocol

TLS/SSL(transport layer secure/secure sockets layer)은 암호화 기술을 이용하여 안전한 통신서비스를 제공하는 암호 규약으로, 전달되는 모든 데이터가 암호화 되며 데이터 일부분만 암호화 할 수는 없다. TLS/SSL은 기밀성, 무결성, 인증, 부인방지를 보장하며 다음의 2 가지 단계를 통해 이루어진다[13].

- 인증은 통신 동안 이용될 암호화 파라미터의 데이터 교환과 협상을 포함한다. 이 단계는 비대칭 암호화와 X509 디지털 서명을 이용한다. 인증과 부인방지를 보장한다.
- 전송되는 패킷에 대하여 교환 데이터의 대칭키 암호화와 MAC (message authentication code) 계산과 검증을 한다. 기밀성과 무결성을 보장한다.

TLS/SSL은 TCP/IP 계층에서 전송계층과 응용계층 사이에서 동작하며, 독립적인 프로토콜로서, 이것이 포함된 통신 과정에의 전송계층 구조를 그림 7에 나타내었다.

Ⅲ. 내부 종단사용자의 정보 유출 위협

내부 사용자는 기업 내 네트워크에 접속 가능하며 중요 시스템 서버 및 데이터 등의 정보에 대해 합법적인 접근 권한을 가지고 언제든지 컴퓨터 및 네트워크의 구성, 프로그램, 데이터 등의 정보를 열람하거나 변경할 수 있는 고용인을 의미한다[14]. 정규 직원은 물론 계약직 직원 및 협력업체 직원 등이 이에 포함된다. 내부 사용자는 기

업이 정한 보안 수준에 따라 PC 보안부터 서버 보안까지 기술적 보안뿐 아니라 물리적, 관리적 보안을 준수하도록 통제된다.

SOA 기반의 ESB 환경 업무시스템은 회사 전체에 분리되어 운용되고 있는 시스템을 통합하고 방대한 데이터를 조합하기 위하여 구축되므로 당연히 많은 내부 사용자가 이용하고 있다. 내부사용자는 접근할 수 있는 통합된 업무시스템에서 다양한 역할과 권한을 부여받게 된다. 하지만, SOA 기반의 ESB 환경 업무시스템에서 일부 종단 사용자는 시스템의 취약점을 이용하여 인가된 정보 이상의 정보를 취득하고자 불법적인 시도를 하기도 한다.

다수의 내부 사용자 중에서 개인의 이익이나 악의적인 의도로 시스템에 위협을 가하지 않도록 안전하게 정보를 관리하는 것은 무엇보다 중요하다. 따라서 이와 같은 업무시스템에서 정보 유출을 방지하기 위하여 부여된 권한을 넘어서는 접근을 시도하는 내부 종단 사용자를 공격자로 정의하고, 공격자가 취할 수 있는 정보 유출 위협을 식별하고, 그 대응을 위한 보안 아키텍처를 설계하는 것이 필요하다.

1. 공격 방법

SOA기반에서 웹 메시지에 대한 보안을 적용할 수 있는 기술은 여러 가지가 있지만, 가장 널리 쓰이는 기술은 SSL이다. ll.4에 서술한 것처럼 TLS/SSL 프로토콜은 전송구간을 암호화하여 데이터를 전송하지만, 보안 기술의 한계가 있다. 이 과정에서 나쁜 의도의 내부 종단 사용자는 프록시 서버와 프록시 툴을 이용한 MITM (man-in-the-middle) 공격을 통하여 그림 8과 같이 정보를 스니핑할 수 있다.

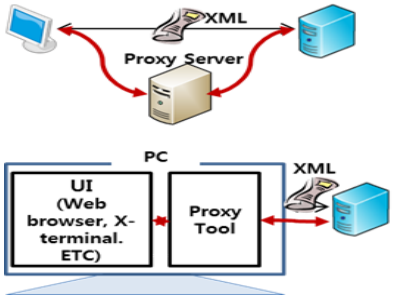

그림 8. 프록시 서버 또는 프록시 툴을 이용한 XML 메시지 스니핑
Fig. 8. XML message sniffing using proxy server or proxy tool

2. 위협

SOA 기반의 업무시스템을 이용하는 종단 사용자는 공격기술이 높은 해킹이기보다는 공개된 해킹 방법을 이용하는 낮은 수준의 공격방법으로 기존 업무시스템의 취약점을 이용하는 경우가 대부분이다. SOA 기반의 SOAP 방식으로 구현된 업무시스템에서 예상할 수 있는 위협은 다음과 같다.

사용자 ID/PW 탈취

내부 종단 사용자의 경우는 내부 사용자 간의 권한에 따라 접근 가능한 정보의 수준이 상이하다. 예를 들면, 관리자 수준에서 접근할 수 있는 정보는 팀원들이 접근할 수 있는 정보보다 중요한 정보인 경우가 많다. 따라서 동일 공간에서 관리자의 권한을 탈취하여 본인에게 인가되지 않은 정보를 취득하기 위한 불법적 시도가 가능하다. 간단하게는 사회공학적 공격방법으로 동일 공간에서 ID/PW 알아내는 것이 가능하며, 또는 인터넷에 공개되어 있는 해킹 기술을 이용할 수 있는 IT 기술을 조금 알고 있는 사용자라면 ID/PW 쉽게 탈취 할 수 있다.

XML Message 가로채기

XML Message를 가로채서 중요 정보를 획득하는 소극적 공격이 있다. 보안성 향상 목적으로 업무시스템의 사용자 출력을 최소화하기 위한 정보 마스킹과 같은 조치가 취해지기 전에 MITM 공격으로 원본 데이터를 획득할 수 있다.

XML Message 위변조

XML Message를 스니핑만 하는 것이 아니라 적극적인 위변조를 통하여 필요한 정보를 획득하는 것이다. 정보 조회를 위한 일련번호, 서비스번호 등 조회 조건으로 가능한 항목을 변조하여 인가되지 않은 정보를 획득할 수 있다. 이와 같은 조작을 통한 위변조가 이루어질 경우, 용이하게 정보를 획득할 수 있어, 대량의 정보 유출 통로가 될 수 있다.

Ⅳ. 내부 종단사용자의 정보 유출 방지를 위한 보안 아키텍처 제안

본 장에서는 3장에서 식별한 내부 종단사용자의 정보

유출 위협에 대응하기 위한 보안 아키텍처를 제안한다.

1. 아키텍처 설계 대상 범위

본 논문에서 제안하고자 하는 보안 아키텍처는 SOA 기반의 ESB 환경에서의 업무시스템을 이용하는 종단 사용자에 의한 잠재적인 정보유출 위협에 대응하기 위한 것이다. 업무시스템을 이용하는 종단 사용자는 내부자에 준하여 PC 보안 통제를 갖추고 있으며, 일정한 권한을 가지고 업무시스템에 접속할 수 있는 권한자에 해당한다. 즉, 보호대상과 공격자는 다음과 같다.

- 보호 대상 : SOA 기반의 ESB 환경으로 구축된 업무 시스템의 처리 정보
- 공격자 : 업무시스템을 이용하여 정보를 처리하는 내부망의 종단 사용자

2. 종단 사용자 서비스 요청 일원화

제안하는 보안 아키텍처를 효과적으로 구축하기 위하여 종단 사용자를 통제하는 ESB 환경에서의 일원화된 관리체계가 필요하다. ESB 서버가 서비스를 요청/응답하는 중간 매개체 역할을 하게 됨으로써 보안 통제의 기본 Layer가 구현될 수 있는 적정 위치가 된다. ESB 서버가 다양한 서비스를 매개하는 기본적인 역할만으로도 많은 부하가 불가피한 상황에서 보안 공통 기능의 부여는 또 다른 추가 부담이 될 수 있을 것이다. 그러나 기업의 정보처리 시스템을 체계적으로 관리하기 위한 소프트웨어 아키텍처로 ESB를 중심으로 서비스를 매개하도록 구성한 경우 빠짐없는 보안통제를 구현하여 관리할 수 있는 최적은 ESB 서버에 위치할 수밖에 없다. 또한 ESB 제품에서 지원하는 보안 표준화 기능을 활용할 수 있을 것이므로 일원화 관리체계의 구축에 따른 서버의 추가 부담을 최소화 할 수 있을 것이다.

3. 보안 아키텍처

본 논문에서 제안하는 보안 아키텍처는 사용자의 서비스 요청과 채널 시스템에서의 요청이 인입되는 ESB 서버에 구현된다. ESB 서버의 보안 공통 처리부를 구성하여 종단 사용자의 정보 유출을 탐지 및 방지할 수 있는 보안 안전성을 검증하도록 한다. 보안상 안전한 서비스 요청에 한하여 비즈니스 시스템의 처리부로 이관하여 응답을 생성 및 제공할 수 있도록 구성한다.

내부 사용자의 위협을 대응하는 보안 아키텍처를 그림 9에 나타내었다. 보안 공통 처리부(security layer)의 구성요소는 다음과 같다. 사용자가 정당한지를 판단하는 “User Authentication” 부분, XML 형식 서비스에서 발생되는 정보 유출 및 정보 위변조 등을 방지하기 위한 Web Service 보안을 처리하는 “Web Service Security” 부분, 서비스 별로 적정한 사용자의 요청인지를 판단하는 “Service Authorization” 부분, 그리고 사용자의 처리 로그를 분석하여 사후 탐지를 할 수 있도록 로그를 관리하는 “Log Management” 부분이 주요 구성요소이다.

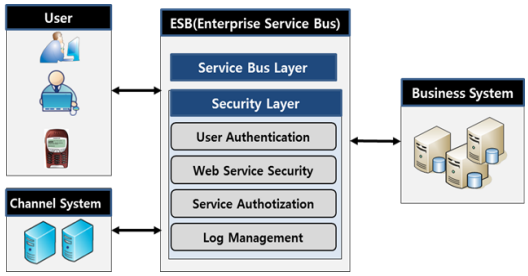

그림 9. 내부 사용자 위협에 대응하는 보안 아키텍처
Fig. 9. Security architecture for insider threat

가. 사용자 인증(User Authentication)

시스템에 접속할 수 있는 정당한 사용자인지 확인하기 위하여 크게 2 가지 부분으로 나누어 관리되어야 한다.

첫 번째는 사용자가 정당한 인가자인지 검증하는 것으로, 이 단계에서는 회사 전체 인사관리 시스템과 연동하여 소속, 직책, 사번등의 인사정보를 확인한다. 이 경우는 일반적으로 ID/PW를 기준으로 검증하게 된다. 그러나 고객정보를 처리하는 시스템과 같이 정보 처리 시스템의 중요도가 높은 경우, 사용자 인증에 추가적인 방법을 도입한다. 즉, OTP, 사설인증서 등의 다중요소 인증을 구현하여 사용자 인증 보안성을 제고한다.

두 번째는 세션의 인증 관리가 필요하다. 비인가 사용자가 자신이 속한 LAN 구간에서 인증된 세션을 가로채기 하여 이를 통해 정보를 획득하기 위한 서비스를 요청하는 것을 방지하기 위한 방법이다. 이를 위해, 사용자가 인증을 하였던 IP와 MAC(media access control)을 해당 사용자의 자산 관리 시스템-IP 및 MAC 관리 시스템-을 통해 인증하며, 해당 IP와 MAC 대상 사용자에 대한 서비스 토큰을 생성하여 관리한다. 서비스 토큰은 일정시간마다 갱신하여 사용자가 갱신된 서비스 토큰으로 요청

할 수 있도록 한다. 또한, 일정시간 이상 서비스 요청이 없는 경우 서비스 토큰을 만료시켜 세션을 비활성화 한다.

나. 웹 서비스 보안(Web Service Security)

XML을 기반으로 하는 SOAP 메시지에 대한 보안 취약점을 대응하는 방안으로 Application Layer에서 기밀성, 무결성을 보장하는 방안으로 구성한다. 메시지의 기밀성을 위한 XML Encryption이나 메시지의 무결성을 보장하기 위한 Message Digest를 구현하기 위하여 Encryption Key를 관리하는 것은 중요하다. 기업에 도입되는 ESB 솔루션은 WS-Security 표준화를 준수하여 XML의 전자서명, 암호화, 키관리 기능을 포함하고 있겠으나 대응하는 사용자 인터페이스 구현 기술에서 WS-Security 표준화 기술을 지원하지 않는다면 XML 보안기능을 자체적으로 구현하는 것이 필요하다. Web service 보안의 구성요소들을 그림 10.에 나타내었다.

- XML Encryption : UI(user interface)와 ESB 구간에서 전송되는 XML 형식의 데이터에 중요 정보를 암호화하여 평문 데이터를 암호문 데이터로 대체하여 전송한다. XML 메시지 전체에 대한 암호화는 성능 저하 문제를 야기 시킬 수 있어 중요 정보에 한하여 암호화 한다.
- Message Digest : XML 메시지에 대한 Hash 값을 산출하여 XML 메시지와 함께 전송하는 방법이다. 메시지를 위변조하는 경우, UI와 ESB 양단에서 계산된 Hash 값이 서로 달라 위변조를 탐지하게 되며, 오류를 발생하여 정상적인 요청 및 응답을 전달하지 않도록 한다.
- Key Management : XML Encryption 및 Message Digest에 이용되는 암호화 키를 관리하는 방안이다. 대칭키를 이용한 암호화 및 Hash 의 salt 값을 이용하는 경우, UI와 ESB 구간에 키를 분배하고 관리하는 것이 필요하다.
- Field Filtering : Business system에서 생성하여 전달되는 메시지 데이터를 ESB에서 필터링하여 UI에 전달되는 항목을 제한한다. 사용자에게 전달되는 메시지에 최소한의 정보만을 공개하는 방안으로 정보 유출을 최소화 할 수 있다.

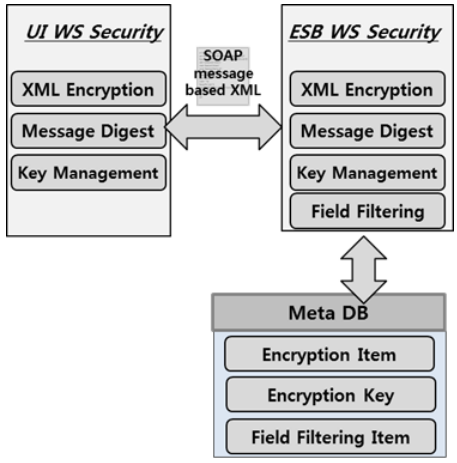

그림 10. 웹 서비스 보안 구성
Fig. 10. Web service security component

다. 권한 인증(Service Authorization)

SOA 기반의 업무시스템에서 재활용성 특징에 따라 다양한 어플리케이션에서 서비스를 호출하고, 각 어플리케이션의 사용자 역할은 상이한 체계로 관리된다. 그래서 SOA 기반의 시스템에서 권한 관리는 매우 복잡하다. ESB로 인입되는 서비스 각각에 대한 서비스 단위로 권한을 인증하는 체계가 필요하다. 우선, 종단 사용자가 이용하는 서비스와 채널시스템에서 사용하는 서비스는 구분이 되어야 한다. 채널시스템에서 사용하는 대량 데이터 처리 서비스에 대한 권한이 종단사용자에게 부여되지 않도록 해야 한다. 그러나 권한 인증을 위하여 종단 사용자와 채널시스템에 대한 통합된 권한 관리 체계를 유지하는 것을 제안한다.

복잡한 시스템에서 권한 인증을 효과적으로 검증하기 위하여 호출되는 서비스는 그림 11 에 나타낸 것처럼 서비스 속성과 사용자 속성으로 관리가 되어야 한다. 서비스 속성에는 어플리케이션명, 메뉴명, 컴포넌트명, 제한된 조건 등이 포함되어야 한다. 사용자 속성에는 사용자가 속한 조직과 사용자의 역할을 포함하여야 한다. 서비스의 재활용성을 높이며, 엄격한 서비스 권한 관리를 위한 서비스 속성과 사용자 속성을 통한 관리 체계를 구성하는 것은 시스템의 통합과 서비스 증가에 유연하게 대응하는 중요한 요소이다.

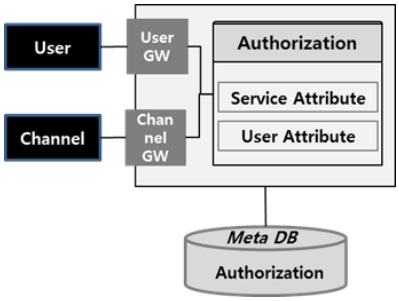

그림 11. 사용자 권한 구성
Fig. 11. Authorization Componen

라. 로그 관리(Log Management)

로그 관리는 ESB를 통해 요청/응답 처리의 모든 로그를 수집하고, 수집된 로그를 기업의 전체 관리 목적에 따라 운용되고 있는 로그 시스템으로 재분배하는 역할을 한다. 이에 관한 ESB 통합 로그 관리 시스템 사이의 로그 전달 관계를 그림 12 에 나타내었다. 정보보호 관련법에 따라 개인정보의 기록 및 보관이 법제화 되어 있어, 기업에서는 일정 보유기간 동안 로그를 기록 보관하는 시스템을 별도 구축하는 경우가 많다. 이와 같은 시스템으로 로그를 중앙 집중화함으로써 로그의 유실을 막고, 위변조를 방지할 수 있는 체계가 마련되고 있다. 또한, 고객정보를 처리하는 시스템의 경우에는 보다 엄격한 로그 관리가 필요하다. 고객정보 처리 이용 내역의 통지 및 정보주체의 권리보장을 위하여 고객정보 처리 내역을 관리하는 로그시스템에 로그를 필터링하여 분배하는 것이 필요하다. 사전에 모든 정보 유출 위협을 탐지 및 방지하는 것은 성능문제 등의 이유로 구현할 수 없으나 사후 로그를 모니터링 하여 위협을 탐지할 수 있는 시스템이 구축되어 있는 사례가 많다. 이와 같은 처리 로그의 모니터링을 통하여 정보 유출 등의 보안 사고를 신속히 인지할 수 있도록 구성하여 대형 정보보안 사고를 예방하는 효과를 기대할 수 있다. 이와 같이 로그의 수집기능을 일원화하여 기업에서 관리하고 있는 다양한 로그를 생성하는 비용을 줄이고, 빠짐없이 로그를 기록 및 모니터링하여 법 준수와 내부 사용자 불법행위를 탐지하여 시스템의 보안 수준을 제고할 수 있다.

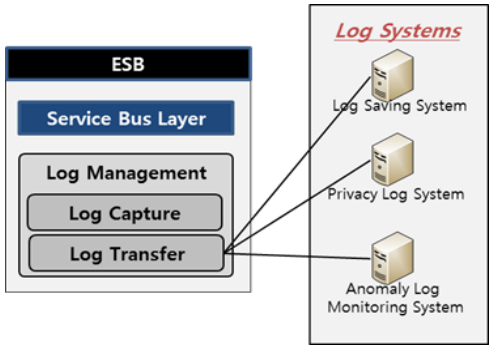

그림 12. 로그 관리 구성
Fig. 12. Log Management Component

V. 결론

본 논문은 SOA 기반의 ESB 환경에서 업무시스템을 이용하는 내부사용자에 의한 불법적인 정보유출을 예측하여 대응하기 위한 보안 아키텍처를 제안하였다. 제안하는 보안 아키텍처의 적용 및 구현을 통해 내부 종단 사용자에 의한 정보 유출 위험의 실질적 감소가 가능할 것으로 기대한다. 기업들은 다양한 업무시스템을 통합하는 과정에서, 웹 서비스 표준의 보안 기술을 이용하여 본 연구가 제안하는 보안 아키텍처를 구현할 수 있다. 제안된 보안 아키텍처의 구현을 위하여 시스템 아키텍처 및 솔루션 선정의 과정에서부터 고려되어야 할 것이다.

본 논문에서는 SOA 기반의 업무시스템의 위협을 내부 종단 사용자로 제한하여 ESB를 이용한 대응 방안을 제안하였으나, 향후에는 SOA 기반의 ESB 환경 전반의 위협을 고려한 총체적인 위험 대응 방안을 모색하는 것이 필요하다.

References

[1] Erl Thomas, "Service-oriented architecture : a field guide to integrating XML and Web services," 2007.

[2] Yan Liu, Ian Gorton, and Liming Zhu. "Performance Prediction of Service-Oriented Applications based on an Enterprise Service Bus," 2007 IEEE International Computer Software and Applications Conference (COMPSAC 2007), 2007.

DOI : https://doi.org/10.1109/COMPSAC.2007.166
[3] 2011 CyberSecurity Watch Survey. CERT. 2011
[4] Chol Hong Im, Do Seok Hong, and Jeong Joon Choi, "A Study of a Scheme to Assess and Improve ESB-based SOA Applications from the S/W Architecture Perspective," Korea IT Service article 5 (2006): 169-178.
[5] Won-kyu Park, Young-bum Park, "Design and Implementation of SOA based S/W Services for Dynamic Behavior of Embedded System", The Journal of The Institute of Webcasting, Internet and Telecommunication VOL. 10 No. 4
[6] Bae Hyun Kim, In Te You. (2005.6). Web Service Security Technology. Review of Korean Society for Internet Information, 6(2), 16-23.
[7] http://api.epeople.go.kr/guide/
[8] Eun-Mi An, Jeong-yong Byun. Beneficial Web Service Security with WS-Policy. Korea Information Science Society 1(1), 131-135, 2007.12.
[9] W3C, "XML Signature Syntax and Processing," Recommendation February 2002.
[10] W3C, "XML Encryption Syntax and Processing," Candidate commendation March 2002
[11] Kim, Youn-deok, "A Design of Secure Key Exchange Protocol and Framework for SOA based ESB Environment", Department of Computing, Graduate School of Soongsil University, 2013.06
[12] Yoon-Ho Kim, "Design and Implementation of Lightweight ESB Bus Engine for Service Oriented Architecture", The Journal of The Institute of Internet, Broadcasting and Communication (JIIBC), Vol. 14, No. 6, pp.131-137, Dec. 31, 2014
DOI : https://doi.org/10.7236/JIIBC.2014.14.6.131
[13] https://www.simple-talk.com/dotnet/net-framework/tlsssl-and-net-framework-4-0/
[14] Salvatore et. al., "Insider Attack and Cyber Security Beyond the Hacker," Springer, 2008.

저자 소개

오 시 화 (정회원)

- 1999년 2월 : 광운대학교 전자통신공학과 졸업
- 2014년 3월 ~ 현재 : 고려대학교 정보보호대학원 정보보호학과 석사과정
- 2015년 7월 ~ 현재 : KTDS 보안센터

<주 관심분야 : 전자금융보안, 정보보호관리체계, 보안 아키텍처>

김 인 석 (정회원)

- 1973년 : 홍익대학교 전자계산학과(학사)
- 2003년 : 동국대학교 정보보호학과(석사)
- 2008년 : 고려대학교 정보경영공학과 (박사)
- 2009년 ~ 현재 : 고려대학교 정보보호대학원 교수

<주 관심분야 : 전자금융보안, IT감사, 전자금융법규>