Security for SOA and Web Services

Applies to:

Enterprise SOA, Web Services, SAP NetWeaver.

For more information, visit the SOA Management homepage.

Summary

Service oriented architecture (SOA) enables loosely coupled applications to be assembled from a set of internal and external services (web services) that are distributed over a connected infrastructure. The distributed nature of SOA makes addressing security concerns a critical success factor. The primary concern in SOA is to establish an interoperable framework that enables security for services, applications, and users in a trusted environment and complies with established corporate policies. The standards and techniques to provide security in a SOA are evolving rapidly. This article will outline the security concerns and related standards in SOA.

Author: Dipak Chopra

Company: SAP America

Created on: 25 March 2008

Author Bio

Dipak Chopra is a Principal Enterprise Architect in National Competency Center in SAP America. He has extensive experience on SOA platforms, technology and initiatives. Over last seven years, he has actively worked with customers, partners and consortia towards enabling critical components of SOA infrastructure.

Table of Contents

Introduction and Approach

It is assumed that the reader has basic knowledge of web services and security.

Web services are characterized by open standards (SOAP, WSDL, UDDI), a variety of implementation platforms (J2EE, .NET, ABAP), and client devices. The presence of such diverse IT systems makes it necessary to take an evolutionary approach that complements the existing infrastructure to address a plethora of security concerns, which poses one of the most significant challenges to realize the great potential of the integration and interoperability of business applications and systems.

This paper outlines the security issues in an SOA, the need for new security architecture, as well as related security standards.

Security Issues

In the conventional enterprise application security realm, there is an underlying concept of a trusted computing base (TCB). The TCB provides mechanisms for enforcing a security policy that protects the resources within the controlled environment. This is equivalent to providing a security perimeter that protects valuable resources.

Web services don't have a clear notion of a security perimeter. SOA is a new approach to distributed computing, where application functionality is abstracted as business services that are location-independent and discoverable on the network. Web services architecture allows service composition, which may engage many different service providers distributed across different platforms and enterprises. In such an open environment, distinguishing legitimate service requests from illegitimate ones becomes a challenge.

A secure SOA should address the following security issues:

Message Security

A message may have to hop through various intermediaries (e.g. Middleware such as the SAP® Process Infrastructure) to reach the final destination. It becomes critical to maintain the *confidentiality* of the sensitive information so that no unauthorized entity can gain access to it. It may also be important to evaluate whether or not the message was not modified in transit and ensure that the *integrity* of the message is maintained. In most of the B2B scenarios, it becomes important to establish the *message authentication*, which guarantees that the message was created by the claimed identity. In addition, *non-repudiation* is also an important requirement in B2B applications. Non-repudiation is about the guarantee that the sender can not later repudiate and claim that he never sent the message.

Trust

As mentioned above, the intent of SOA is about the loose coupling of services. This loose coupling makes the issue of trust quite explicit. The WS-Security specification defines trust as, "the characteristic that one entity is willing to rely upon another entity to execute a set of actions." The trust issue has two aspects; one is to specify the trust policies and the second is to broker trust between different security domains (also known as spheres of trust).

Distributed Policies

Corporations want to make sure that any communication with the outside is compliant with the policies they've established. Policies are attached to all the related entities, client, service, and discovery. Policies are used to describe a broad range of service requirements and capabilities. For example, an organization may use a policy to define security requirement such as encryption. Policies are validated at the time of interaction and within the context of the interaction.

Identity Management

A client uses an identity (most of the time it is a user identity) to gain access to the service it needs. A typical SOA solution is distributed over multiple security domains and there could be several identities attached to a single user in different security domains. This poses some problems with traditional security approaches. The security infrastructures may vary among the various backend systems, so users may need to be

authenticated for each system. The other problem is related to the SOA service composition layer, which might be calling many different atomic services falling under different security domains. The absence of overall security context makes it difficult to associate multiple user identities.

Interoperability

Although interoperability is not a security issue per se, it is a very important factor in making a typical SOA solution work. Most of the web services specifications provide a number of mechanisms to do the same thing, which lead to interoperability issues. For example, a sender can encrypt data using Triple DES, AES 128, AES 192, or AES 256 algorithms and the receiver can only decrypt AES 128.

Security Standards and Technologies

Is a web services security layer really required? To answer this question, one needs to look into the security concerns and try to address them using existing security technologies. The industry has a set of existing and widely accepted transport-layer security mechanisms, such as SSL (Secure Sockets Layer), TLS (Transport Layer Security). Despite their popularity, SSL and TLS (which is a minor update of SSL) have some limitations when it comes to web services.

- SSL is designed to provide point-to-point security, which is not enough for web services because end-to-end security is required. In a typical web services environment where XML-based business documents are routed through multiple intermediaries, it becomes difficult for those intermediaries to participate in security operations in an integrated fashion.
- SSL-based communication provides security (confidentiality, integrity) at the transport level rather than at the message level. As a result, messages are protected only while in transit on the wire. For example, sensitive data received on SSL channel, once persisted, is not generally protected unless one applies some encryption technology.
- Finally, SSL does not provide element-wise signing and encryption. For example, if there is a large purchase order XML document, and there is a need to sign or encrypt only a credit card element, signing or encrypting only that element with SSL is not possible.

All of these limitations make SSL and TLS unsuitable for most of the security needs of web services. In the last couple of years, this gap has been identified by the industry and considerable effort has been invested to provide a new security architecture to address the needs of SOA. It is important to realize that the new service oriented security architecture must adhere to the essential characteristics of SOA.

Organizations have existing security infrastructures in place, which protect the resources on diverse platforms. The fundamental goal of SOA is to enable the existing infrastructures to interoperate. The solution is to enable a security spanning layer over existing security infrastructures with a SOA.

SOA enables organizations to build a set of reusable security services that can be used by business applications. The resulting reusability ensures consistent security policies across platform with reduced development costs. For example, an authentication function can be offered as service. Due to the distributed nature of SOA, it is important that the standards-based architecture is adopted, which delivers interoperability and lets the infrastructure operate across organizational boundaries. In the last couple of years, much work has been done to create security standards, which are very promising.

Message Security

Message security is about providing message confidentiality, integrity, non-repudiation, and exchanging security credentials between web service client and web service.

WS-Security is an OASIS standard. WS-Security describes enhancements to SOAP messaging to provide message integrity, message confidentiality, and message authentication. WS-Security uses XML Signature to provide message integrity and message authentication and uses XML Encryption to provide confidentiality. WS-Security also provides a general-purpose mechanism for associating security tokens with messages. Examples of security tokens are X.509 certificate, SAML assertion.

XML Encryption is a W3C recommendation. It ensures confidentiality of XML information transfers. XML Encryption allows the parts of an XML document to be encrypted while leaving other parts open. WS-

Security provides processing rules for using XML Signature for SOAP messages. JSR 106 (XML Digital Encryption APIs) defines a standard set of APIs for XML digital encryption services.

XML Signature is a W3C recommendation. It ensures message integrity and authentication. WS-Security provides processing rules for using XML Signature for SOAP messages. Signature can be applied over parts of an XML document. JSR 105 (XML Digital Signature APIs) defines a standard set of high-level implementation-independent APIs for XML digital signature services.

As mentioned above, in B2B applications, sometimes there is a need for message non-repudiation. Non-repudiation is required due to malicious senders who can later disavow having created and sent a particular message. Resolving non-repudiation issue requires message authentication and sender authentication simultaneously. This can be achieved by using XML Signature for message authentication and SSL-based sender authentication.

Trust

In a distributed environment like SOA, the two sides (client and service) need to establish trust before they can interact with each other.

WS-Trust is an OASIS standard. WS-Trust defines extensions to WS-Security to provide mechanisms to get security tokens and to establish trust relationships. For example, a client can send only X.509 security tokens and the web service can accept only SAML security tokens. WS-Trust provides a protocol to get the SAML security token by presenting the X.509 security token. By doing so, WS-Trust resolves the token format mismatch; trust between client and web service can be established. This provides a great benefit as different security infrastructures can interoperate with each other without significant changes.

Distributed Policies

In a typical SOA, where the client and the service may not be in the same security domain, policies enforce security rules on the outgoing (client side) and incoming (service) messages.

WS-Security Policy is an OASIS standard. It describes how senders and receivers can specify their security requirements and capabilities. For example, a service can specify that it requires SAML token and signed message in the incoming SOAP request. WS-Security Policy is based on WS-Policy (a W3C Recommendation). WS-Policy is fully extensible and does not place limits on the types of requirements and capabilities that may be described. It also defines a mechanism for attaching or associating service policies with SOAP messages.

Identity Management

As SOA environments are typically highly decentralized in nature, identity management becomes a significant challenge for web services. Identities can be stored in many directories as well as many different types of directories, including proprietary username/password repositories, LDAP, Active Directory, and X.509 certificate stores. An additional challenge is that SOA may have requests that result in additional requests to many different applications at once. An SOA-ready service may be composed of many service operations from many different services that each has their own identity. As part of a single transaction, many different services may be touched whether in parallel or in serial. Being able to authenticate and be authorized across all of these systems seamlessly improves the user experience as well as performance - driving the need for a federated identity solution for SOA environments.

SAML (Security Assertion Markup Language) is an OASIS Standard. SAML provides a framework for the exchange of security credentials amongst disparate systems and applications in an XML-based format. WS-Security provides a SAML Profile that defines mechanisms to use SAML 1.1 within the context of SOAP messages. The SAML 2.0, which is now an OASIS Standard, incorporated much of Liberty Alliance's work in an effort to become a unified standard for identity federation, which adds account linking, Single Sign-on/off, and improved facilities for establishing trust between organizations.

WS-Federation is an OASIS initiative. WS-Federation defines mechanisms to allow different security domains to federate by brokering trust of identities, attributes, or authentication between participating web services. It is based on WS-Security, WS-SecurityPolicy, and WS-Trust.

Interoperability

In general, wide spread of adoption of security standards increases interoperability, but even different implementations of the same specification may not interoperate in some cases. Core specifications (WS-Security, SAML, etc.) are designed in a way to be extensible and provide number of options for doing the same thing.

WS-I Basic Security Profile is a specification from WS-I to promote interoperability in the context of Web Service security. WS-I isn't a standards development organization, but it works closely with a number of standards bodies - W3C, OASIS - to promote and utilize the right set of technologies in business scenarios. The Basic Security Profile provides guidance for the use of web services security standards and technologies in the development of interoperable web services. The Basic Security Profile is an interoperability profile that addresses transport security, SOAP messaging security, and other security considerations. The profile provides specific security requirements, which can be tested on sender as well as receiver side.

Summary

Despite the number of complex issues surrounding security in the SOA, the emerging security standards offer a lot. A significant amount of work has been done in designing security standards in a modular, complementary, and evolutionary manner. As the domain of security standards is becoming mature, major web services platform vendors (SAP, IBM, Microsoft, Sun) have started adopting these technologies. The future of security standards revolve around three major topics: first the standardization of various WS-* specifications (WS-Trust, WS-Policy, WS-Federation) at least at the functional level; second, the convergence of the identity federation specifications, SAML and WS-Federation; and third, the increased focus on interoperability.

Related Content

Getting Started: Web Services Security (WS-Security)

Getting Started with WS-I Basic Security Profile

Getting Started with XML Encryption

Getting Started with XML Signature

Getting Started: Web Services Trust (WS-Trust)

Web Services Security Cookbook: Using Web Services Technology in Mixed IBM & SAP Environments

Web Services Security Interoperability between SAP NetWeaver 7.0 and Microsoft Windows Communication Foundation (WCF) – Part I

Web Services Security Interoperability Between SAP NetWeaver 7.0 and Microsoft Windows Communication Foundation – Part II

Web Services Security Interoperability Between SAP NetWeaver 7.0 and Microsoft Windows Communication Foundation – Part III

For more information, visit the SOA Management homepage.

Copyright