SDN Contribution

Getting Started: Security Assertion Markup Language (SAML)

Applies to:

SAP NetWeaver Web Application Server (WebAS)

Summary

This document provides an introduction to the OASIS Security Assertion Markup Language (SAML).

Created on: 2 May 2006

Author Bio

As a Standards Architect with SAP's Industry Standards team, Martin Raepple works in the area of standardization and interoperability testing of new Web Services technologies, focusing on message security and identity management. Martin is a frequent speaker at conferences and author of books and articles relating to information security, integration middleware and J2EE development.

Introduction

In a typical IT infrastructure, user's security information is distributed across multiple servers (database, LDAP, proprietary user stores) and the resources (JSP, BSP, servlets) which a user accesses also are hosted on multiple servers (sometimes in multiple DNS domains). With internet browsers as the standard client for many applications, user navigation needs to be seamless with respect to security. Users need to be authenticated and authorized by applications as users access those applications.

OASIS SAML enables XML-based exchange of security information related to a user between servers over HTTP. The information for authentication and authorization can be exchanged using SAML in the back end without users noticing the exchange. SAML acknowledges that each platform has its own mechanism for authentication and authorization. Consequently, user security information is exchanged in a standard XML-based structure. The basic entity of security information is known as 'assertion'. An assertion is a statement made by a trusted authority. There are three types of assertion types

- Authentication: It presents a fact that the user was authenticated at a particular time and with a particular authentication method. It is analogous to the SAP Logon Ticket technology.
- Attribute: It presents a fact that the user has these attribute values. Using this assertion type, a trusted authority can vouch user's attributes.
- AuthorizationDecision: It presents a fact that the user is authorized to access a particular resource.

When a SAML-enabled server (known as Service Provider) receives an HTTP request for a protected resource, it starts with the identification of the user. The incoming HTTP request will either have a SAML assertion or a pointer to it. In the case of a pointer, the Service Provider will resolve the pointer with an associated Identity Provider to get a SAML assertion. Once the user identity is established, the user needs to be authorized for the protected resource. There are a number of possible variations:

- Local authorization: The incoming user is mapped to a local user. Then authorization is done against the local user.
- Attribute-based authorization: The requested resource can only be accessed by those users which satisfy a certain criteria (for example, spending limit > $10K). This user attribute information (user's spending limit) is kept somewhere else (Attribute Authority). The Service Provider can request the Attribute Authority for the user's SAML attribute assertion (with spending limit). Based on the received SAML attribute assertion, the Service Provider can make an authorization decision.
- Distributed Authorization: Sometimes a user requests for a resource (let’s call it Composite Resource or CR) that is a collection of other resources (lets call it Sub Resource(s) or SR) hosted on different servers. In this scenario, the CR can identify the user and then ask individual SRs for authorization for this user. Each SR can return SAML authorization decision assertions to the CR. The CR can present the result based on the SAML authorization decision assertions.

It should be noticed that the above list is to illustrate some examples of SAML. SAML provides a generic framework that can be used in any scenario where the decision point (for authentication and authorization) and the information point (for providing user security information) are not the same entity.

SAML 1.0

SAML 1.0 was ratified as OASIS Standard in November 2002. SAML 1.0 enables three usage scenarios.

- Single Sign On (SSO): SSO plays an important role in cross-application navigation. In SAP context, SAP Logon Ticket is a popular SSO mechanism. SAML enables SSO between multiple applications, where one application is the identity provider and other applications are service providers. User information is exchanged in an XML-based SAML assertion. SAML assertions are analogous to SAP Logon Tickets. There are two browser-based SSO profiles, Browser/Artifact Profile and Browser/POST Profile, that specify the interactions between user, identity provider and service provider to exchange user security information to allow SSO.

- Remote Authorization: When a service provider receives a user request with a user's security context for a protected resource, it can invoke an external authorization service to evaluate if the user is allowed to access this resource.

- Attribute Based Authorization: When a user navigates from one application to another application, an authorization check needs to be made to evaluate if the user is authorized to access the requested resource. The authorization check can be done based on the user's attribute information traveling in SAML assertion. This allows for a centralized repository for user attributes that can be used across multiple applications.

Besides these usage scenarios, SAML also provided input to WS-Security SAML Token Profile, which specifies how SAML assertions are processed within SOAP messages.

SAML 1.1

SAML 1.1 was ratified as OASIS Standard in September 2003. SAML 1.1 is a minor revision to SAML 1.0 for bug fixing and clarifications.

SAML 2.0

SAML 2.0 is a major upgrade with respect to SAML 1.0/1.1 and was ratified as an OASIS Standard in March 2005. The SAML 2.0 effort delivers on the following goals:

- Address issues and enhancement requests that have arisen from experience with real-world SAML implementations and with other security architectures that use SAML

- Adding support for features that were deferred from previous versions of SAML

- Develop an approach for unifying various identity federation models found in real-world SAML implementations and SAML-based security architectures

SAML 2.0 enables the following scenarios:

- Identity Federation: Before SAML can be used to enable SSO for a Principal, the Identity Provider and Service provider must be able to share between themselves some identifier for that Principal; otherwise, they would have no means by which to query or assert to the other the authentication status of that Principal. Identity federation (or account linkage), which is a process by which, after explicit Principal opt-in, the Identity Provider and Service Provider exchange between them a unique and opaque (a random string that, in and of itself, can not be linked to the Principal) identifier for that Principal. Subsequent communications between the Identity Provider and Service Provider on behalf of this Principal use this opaque identifier.

- Authentication Context: One particular area in which Liberty extended SAML is a concept Liberty refers to as Authentication Context; in other words, information added to the SAML Authentication Assertion regarding the details of the technology used for the actual authentication action (e.g.

password versus biometric), what processes were followed in the issuance of the identity (e.g. face-to-face meeting versus Web self-registration), and other characteristics that may be relevant to the consumer of the SAML Authentication Assertion

- Permission Based Attribute Sharing: Allow services describing an individual's attributes (e.g., profiles, wallet, location) to participate in on-line transactions initiated by the individual. With this functionality, it is ensured that user attributes are released to only to those entities for which the user has given explicit consent.
- Pseudonyms for principals: In a real world scenario, where you have multiple Identity and Service Providers, there is a need for a mechanism by which Principals can be uniquely identified across multiple domains. At the same time, Principal privacy demands that a Principal not be required to have a single identifier across all domains. The Identity Provider and Service Provider in a federation need to remember each other's handle for the Principal, so they create entries in their Principal directories for each other and record each other's handle for the Principal.
- SSO with Attribute Exchange: This can be used to achieve a kind of federation without using an account-linking model
- Enhanced Client Profile: This SSO profile is for enhanced clients that have information about the relevant identity provider for a given principal and service provider. This is primarily targeted for mobile devices.

Disclaimer and Liability Notice