BEA WebLogic Server R
SAML 2.0 and SAML Token Profile 1.1 Support

Version: 10.3 Tech Preview

Document Date: October 2007

Table of Contents

Overview of SAML 2.0 and SAML Token Profile 1.1 Support

WebLogic Server 10.3 includes broad support for SAML 2.0, including support for the SAML 2.0 Web Single Sign-On (SSO) profile and the Web Services Security (WS-Security) SAML Token profile 1.1.

The SAML 2.0 Web SSO profile is part of the core set of SAML 2.0 standards, and specifies how SAML 2.0 assertions and protocols should be used to provide browser-based single sign-on between and among Identity Provider (IdP) sites and Service Provider (SP) sites.

The SAML Token profile is part of the core set of WS-Security standards, and specifies how SAML assertions can be used for Web Services security. WebLogic Server 10.3 supports SAML Token Profile 1.1, including support for SAML 2.0 and SAML 1.1 assertions. SAML Token Profile 1.1 is backwards compatible with SAML Token Profile 1.0.

SAML 2.0 Components

Support for SAML 2.0 is provided by several different components areas in WebLogic Server 10.3 including:

- SAML 2.0 Security Providers specifically the SAML2 Credential Mapping provider and the SAML2 Identity Assertion provider
- SAML 2.0 Single Sign-On Services
- Web Services support for SAML Token Profile 1.1

SAML 2.0 Security Providers

The new SAML2 Credential Mapping provider and SAML2 Identity Assertion provider generate and consume, respectively, SAML 2.0 assertions.

At least one of the providers must be configured in order to use of SAML 2.0 in WebLogic Server 10.3.

Note: For some uses of SAML 2.0, the SAML authentication provider must also be configured. The SAML authentication provider enables “virtual user” functionality for both the SAML 2.0 and SAML 1.1 identity asserters.

SAML 2.0 Single Sign-On Services

WebLogic Server 10.3 can be configured to act as a SAML 2.0 Identity Provider (IdP), Service Provider (SP), or both. When acting as an IdP, the SAML2 Credential Mapper must be configured, so that the IdP can produce assertions. When acting as an SP, the SAML2IdentityAsserter must be configured, so that the SP can consume assertions.

SAML 2.0 Single Sign-On Services are configured on a per-server basis. To enable SAML 2.0 SSO for a cluster, configure the SSO services on each server in the cluster.

Web Services Security SAML Token Profile 1.1

WebLogic Server 10.3 Web Services now supports SAML Token Profile 1.1. This feature includes support for SAML 2.0 and SAML 1.1 assertions, and is backwards compatible with SAML Token Profile 1.0

SAML tokens are configured for a web service through use of the appropriate WS-SecurityPolicy assertions.

> **Note:** SAML Token Profile 1.1 is supported only through WS-SecurityPolicy. The earlier "WLS 9.2 Security Policy" supports SAML Token Profile 1.0/SAML 1.1 only.

When using SAML Token Profile, the appropriate SAML security providers must be configured (either the SAML 2.0 or SAML 1.1 Credential Mapping or IdentityAssertion providers) depending on the desired SAML version(s) and assertion usage.

Configuring SAML 2.0 Security Providers

In order to use SAML 2.0, you must configure the SAML2 Credential Mapping provider, the SAML2 IdentityAssertion provider, or both. The SAML Authentication provider maybe configured along with the SAML2 Identity Assertion provider if support for "virtual users" is desired.

General Guidelines

This section provides some general information on configuring the SAML 2.0 security providers:

- Providers can be configured through the WebLogic Server Administration Console or by using the WebLogic Scripting tool (WLST).
- Weblogic Server must be rebooted after configuring security providers. Providers are not active, and provider management operations are not available, until the server is rebooted. Multiple providers can be configured at the same time, however, and the server rebooted only once.
- When multiple authentication providers are configured, their execution proceeds according to the rules for a JAAS Authentication service. This means that their behavior is affected by the configured provider ordering, and by the JAAS control flags configured for each provider. An understanding of JAAS Authentication is essential to understanding the behavior of the SAML2 Identity Assertion provider and SAMLAuthentication provider.

Using WLST to Configure SAML Providers

When using WLST to configure the SAML security providers, remember the following:

- Use Edit mode to add or remove providers, or to modify provider settings: Navigate to the Edit tree and use the startEdit(), save(), and activate() commands.
- Do not use Edit mode when invoking management operations on the providers (for example, adding, removing, or modifying SAML partner entries).
- Whether editing provider configuration or invoking management operations, the first step is to get the realm object:

  ```
  realm = cmo.getSecurityConfiguration().getDefaultRealm()
  ```

- Method calls on the security realm can be used to create, lookup, or remove Credential Mapping and Authentication providers.

Configuring the SAML2 Credential Mapping Provider

To configure a SAML2 Credential Mapping provider:

1. Create a SAML2 Credential Mapping provider (class name `com.bea.security.saml2.providers.SAML2CredentialMapper`) using the Administration Console or WLST.
2. Configure the Credential Mapping provider's issuer URI for the Credential Mapping provider using the `setIssuerURI()` method.
3. To use an assertion signing key other than the default SSL identity key for the server, configure the key's `alias` and `passphrase` using the `setSigningKeyAlias()` and `setSigningKeyPassPhrase()` methods. The signing key (and corresponding certificate) must be configured in the SSL keystore for the server.

Configuring the SAML2 Identity Assertion Provider

To configure a SAML2 Identity Assertion provider:

1. Create a SAML2 Identity Assertion provider (class name `com.bea.security.saml2.providers.SAML2IdentityAsserter`) using the Administration Console or WLST.
2. There are no other required configuration entries.

Configuring the SAMLAuthenticator

To configure the SAML Authentication provider

1. Create a SAML Authentication provider (class name `weblogic.security.providers.saml.SAMLAuthenticator`) using the Administration Console or WLST.
2. There are no other configuration entries.

Configuring SAML 2.0 Partners

After the appropriate providers have been configured and the server has been rebooted, you can configure SAML 2.0 partner entries.

You must configure a SAML partner entry for each entity with which you want to use SAML single sign on (SSO) or SAML Token Profile. Partner entries can be configured using the WebLogic Server Administration Console or using WLST. This document describes only the WLST method.

Types of SAML 2.0 Partners

SAML partners are either IdP partners (representing an Identity Provider partner) or SP partners (representing a Service Provider partner). Partners are further distinguished as WebSSO partners or WSS partners. Altogether, there are four partner types:

- `WebSSOIdPPartner`
- `WebSSOSPPartner`
- `WSSIdPPartner`
- `WSSSPPartner`

General Guidelines

- All SAML 2.0 partner entries must be given a name. For Web SSO partners, this name is arbitrary but should not contain spaces. For WSS partners, this name must be the endpoint of the web service with which the partner entry is associated.
- SAML 2.0 partners are ignored if their Enabled attribute is set to `false`. This setting allows you to save partial or incorrect configurations without affecting on-going SSO or WSS operations. When the Enabled attribute is set to `true`, validation occurs and prevents an incorrectly configured partner from being saved. Enabled partners become available for use by the SAML 2.0 security providers.

Configuring Identity Provider (IdP) Partners

IdP partners are configured on the SAML v2 IdentityAssertion provider, and represent Identity Providers from whom the identity asserter will accept and validate assertions.

To create an IdP partner:

1. Start WLST and get the SAML v2 IdentityAssertion provider.
2. To create a WebSSOIdPPartner, you can either:
    a. Call the `newWebSSOIdPPartner()` method of the Identity Assertion provider. Configure the various settings, including instantiating new Endpoint objects by calling `newEndpoint()` or `newIndexedEndpoint()` and setting these endpoints on the partner object.
    b. Call the `consumeIdPPartnerMetadata()` method to generate a WebSSOIdPPartner object from SAML 2.0 Metadata provided by the IdP partner organization. This is the recommended method for creating a WebSSOIdPPartner.
3. To create a WSSIdPPartner, call `newWSSIdPPartner()` and configure the desired settings on the partner object. WSS partners cannot be created from Metadata because the SAML 2.0 standard does not specify Metadata for WSS use cases.
4. Use the `loadCertificate()` method to load a certificate. The resulting certificate object can be set in a partner configuration as appropriate.
5. In all cases, save the newly created partner object by calling `addIdPPartner()` and passing in the partner object.

Additional methods exist to list, fetch, modify, and remove IdP partners.

Configuring Service Provider (SP) Partners

SP partners are configured on the SAML v2 Credential Mapping and represent Service Providers for whom the Credential Mapping will generate assertions.

To create an SP partner:

1. Start WLST and get the SAML v2 Credential Mapping provider.
4. To create a WebSSOSPPartner, you can either:
    a) Call the `newWebSSOSPPartner()` method of the Credential Mapping provider. Configure the various settings, including instantiating new Endpoint objects by calling

`newEndpoint()` or `newIndexedEndpoint()` and setting these endpoints on the partner object.

b) Call the `consumeSPPartnerMetadata()` method to generate a WebSSOSPPartner object from SAML 2.0 Metadata provided by the SP partner organization. This is the recommended method for creating a WebSSOSPPartner.

5. To create a WSSSPPartner, call `newWSSSPPartner()` and configure the desired settings on the partner object. WSS partners cannot be created from Metadata as the SAML 2.0 standard does not specify metadata for WSS use cases.
6. Use the `loadCertificate()` method to load a certificate. The resulting certificate object can be set in a partner configuration as appropriate.
7. In all cases, the newly created partner object must be saved by calling `addSPPartner()` and passing in the partner object.

Additional methods exist to list, fetch, modify, and remove SP partners.

Configuring SAML 2.0 Single-Sign On (SSO)

SAML 2.0 SSO services must be configured on each server where you want SAML 2.0 IdP and/or SP services to run.

This configuration can be done through the WebLogic Server Administration Console or by using WLST to modify the SingleSignOnServicesMBean. In the Administration console, these settings can be found under Server Configuration. The SingleSignOnServicesMBean is a child of the ServerMBean.

Various settings allow you to:

- Enable or disable IdP and SP services independently
- Indicate which SAML 2.0 binding(s) should be used for SSO
- Indicate whether SSL must be used for SSO requests
- Configure signing keys and certificates

  Note: If signing keys and certificates is not configured, the SSL identity key and certificate for the server are used. If specific aliases and passphrases are configured, they are looked up in the SSL keystore of the server.

Configuring and Using SAML Token Profile 1.1

The following sections provide information on configuring and using SAML Token Profile 1.1.

Configuring SAML Token Profile 1.1

To use SAML Token Profile 1.1:

1. Make sure that the SAML providers you need are configured and add the appropriate partner entries.

   Notes: You will need to configure both SAML 1.1 and SAML 2.0 security providers if you want to enable both versions of SAML for use with the SAML Token Profile. This document does not describe configuration of SAML 1.1 providers or partner entries, as

SAML 1.1 is not a new feature and existing documentation describes how to configure SAML 1.1 for use with Web Services.

When configuring SAML 2.0 partner entries, you must use the endpoint URL of the target web service as the name of the partner for both WSSIdPPartner and WSSSPPartner entries. Generally, you should specify the URL as `https` if SSL will be used but there is a known issue that requires SSL endpoints to be specified as `http` in some cases.

2. If you will be using policies that involve signatures related to SAML assertions (for example, SAML Holder-of-Key policies) where the key referenced by the assertion is used to sign the message, or Sender-Vouches policies where the sender's key is used to sign the message, you need to configure keys and certificates for signing and verification.

   Note: These keys/certificates are *not* used to create or verify signatures on the assertions themselves. Creating and verifying signatures on assertions is done using keys and certificates configured on the SAML security providers.

   a) Configure a PKI Credential Mapping provider on the sending side, and populate it with the keys/certificates to be used for signing:

   ```
   pkiCM.setKeypairCredential(
       type=<remote>, protocol=http,
       remoteHost=hostname, remotePort=portnumber,
       path=/ContextPath/ServicePath,
       username, Boolean('true'), None,
       alias, passphrase)
   ```

   The first (`String`) parameter is used to construct a Resource object that represents the endpoint of the target web service. The `userName` parameter is the user on whose behalf the signed WS message will be generated. The `alias` and `passphrase` parameters are the alias and passphrase used to retrieve the key/certificate from the keystore configured for the PKI Credential Mapping provider. The actual key and certificate should be loaded into the keystore before creating the `KeypairCredential`.

   b) Add the same certificates to the Certificate Registry on the receiving side, so they can be validated by the WS security runtime:

   ```
   reg.registerCertificate(certalias, certfile)
   ```

Using SAML Token Profile 1.1

Once the necessary SAML providers, PKI Credential Mapping provider, and Certificate Registry are configured, applications can use SAML Token Profile 1.1.

SAML Token Profile 1.1 is supported only for use with WS-SecurityPolicy 1.2. To use SAML Token Profile 1.1:

- Select an appropriate policy from the following supplied list:
  - `Wssp1.2-2007-Saml2.0-SenderVouches-Wss1.1.xml`
  - `Wssp1.2-2007-Saml2.0-SenderVouches-Wss1.1-Asymmetric.xml`
  - `Wssp1.2-2007-Saml2.0-HolderOfKey-Wss1.1-Asymmetric.xml`
  - `Wssp1.2-2007-Saml2.0-Bearer-Https.xml`
  - `Wssp1.2-2007-Saml1.1-SenderVouches-Wss1.1.xml`

- `Wssp1.2-2007-Saml1.1-SenderVouches-Wss1.0.xml`
- `Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.1-Asymmetric.xml`
- `Wssp1.2-2007-Saml1.1-HolderOfKey-Wss1.0.xml`

  You can also write your own policy though BEA recommends using one of the supplied policies.

- Code or configure your application to use the policy through policy annotations, policy attached to the application's WSDL, or runtime policy configuration.