SDN Contribution

Configuring Single Sign-on for SAP NetWeaver Application Server Java™ with IBM Tivoli Federated Identity Manager using SAML 1.0

Applies to:

- SAP NetWeaver Application Server Java 04 SP14
- IBM Tivoli Federated Identity Manager 6.0

Summary

This article describes how to configure single sign-on (SSO) for SAP NetWeaver Application Server Java™ with IBM® Tivoli® Federated Identity Manager using Security Assertion Markup Language (SAML) 1.0

Created on: 21 March 2006

Author Bio

Peter Tuton is a Staff Software Engineer in the Tivoli Security Integration Factory Team based on the Gold Coast, Australia. In this role, Peter is the Technical Lead for the team developing integration solutions for the IBM Tivoli Access Manager software suite.

Table of Contents

Introduction 2

Overview 2

Configuring TFIM 4

Step 1: Creating a federation 4

Step 2: Creating a partner 6

Step 3: Restart WebSphere 7

Step 4: Restart WebSEAL 7

Configuring SAP NetWeaver AS Java 7

Step 1: Changing the Startup Mode for the SAML Service 7

Step 2: Creating a destination 8

Step 3: Configuring the SAML parameters 8

Step 4: Adjusting the Login Module Stack for using SAML 9

Testing the configuration 9

Resources 9

Disclaimer and Liability Notice 10

Introduction

SAP NetWeaver Application Server Java provides for the ability to use the SAML protocol to sign on to its applications (for example, SAP NetWeaver Portal). Tivoli Federated Identity Manager (TFIM) 6.0 can be used as an assertion source (identity provider) for federated single sign-on to SAP AS-Java applications.

This article describes how to configure single sign-on for SAP AS-Java with TFIM using the SAML 1.0 protocol and the Browser/Artifact profile. It assumes TFIM and SAP AS-Java are installed and running.

You can get an overview of Tivoli Federated Identity Manager at

http://www.ibm.com/software/tivoli/products/federated-identity-mgr/

Overview

Security Assertion Markup Language (SAML) is a standard produced by the Security Services Technical Committee (SSTC) within the Oasis Standards Organization. SAML consists of two distinct pieces of functionality: The SAML assertion (used to transfer information about a user) and the SAML protocol (the means of exchanging a SAML assertion). Full details on SAML are available from: http://www.oasis-open.org/committees/security

SAML 1.0 and 1.1 (both ratified as standards) define push-based protocols, meaning that the SSO request is initiated from the identity provider and pushed to the service provider. SAML provides for:

- Browser/POST profile
- Browser/Artifact profile

The difference between these two is how the actual security information (vouch for token) is exchanged between an identity provider (in this case TFIM) and service provider (SAP AS-Java).

With a Browser/POST profile, a SAML assertion (vouch or token) is included in the response that is sent to the service provider as part of an HTML form. This is a front channel exchange of the SAML assertion.

With a Browser/Artifact profile, a pointer to the SAML assertion (called an artifact) is included in the query string of an HTTP 302 redirect to the service provider. The service provider in turn issues a direct SOAP/HTTP request back to the identity provider, exchanging the artifact for the actual SAML assertion.

SAP NetWeaver AS Java supports the Browser/Artifact profile, as illustrated in Figure 1.

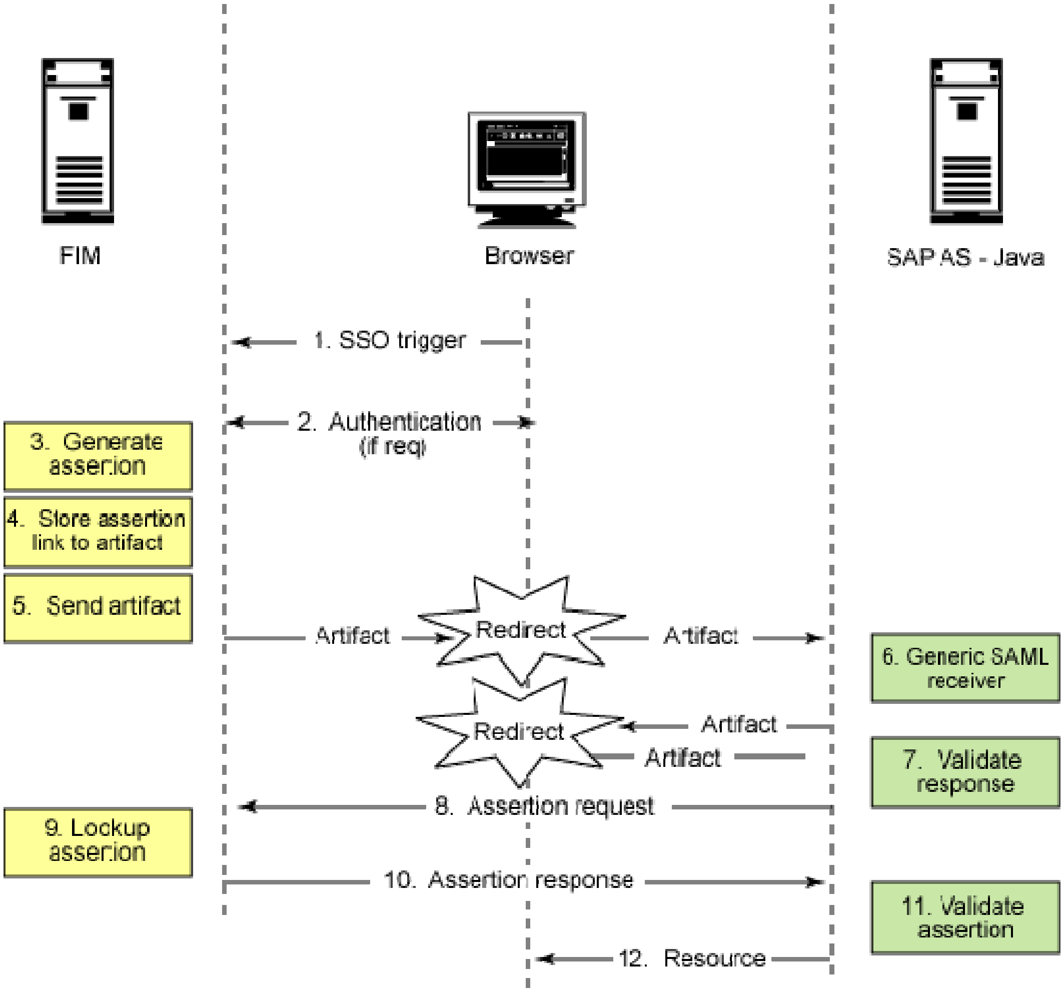

Figure 1. SAML SSO: Browser/Artifact

The process for when a user accesses an SAP NetWeaver AS Java application configured for SAML authentication through TFIM is as follows:

1. The user navigates to an SAP NetWeaver AS Java application via TFIM causing an SSO trigger. The application's target URL is contained in the request.
2. The user is required to authenticate to TFIM, if not already authenticated.
3. TFIM generates an assertion for the request.
4. TFIM links the assertion to an artifact in its assertion cache.
5. TFIM returns the artifact to the browser (contained in a query string parameter along with the original target URL) and instructs the browser to redirect to the application.
6. Optionally, a generic SAP SAML receiver servlet is used as the SAML receiver instead of the application. The generic SAML receiver simply redirects the browser to the application with the artifact. This allows for TFIM to provide SAML SSO to any SAP NetWeaver AS Java application configured for SAML on the one SAP NetWeaver AS Java environment. This article describes a configuration using the generic SAML receiver.
7. The application is configured with the SAP SAML Login Module. The SAML Login Module evaluates the artifact by validating the response. The source site (TFIM) is determined using information contained within the artifact.
8. The SAML Login Module generates an assertion request (using SOAP) and sends it to TFIM. This request contains the artifact originally generated by TFIM.
9. TFIM performs a lookup in its assertion cache for the appropriate assertion using the artifact as a lookup key.
10. The assertion response is returned from TFIM to the SAML Login Module.
11. The SAML Login Module analyses the assertion and authenticates the user.
12. Assuming successful authentication, the resource is returned to the user.

Configuring TFIM

This section covers the steps required to create and configure a SAML 1.0 federation partner in TFIM.

Step 1: Creating a federation

Perform the following tasks to create a new federation that uses the SAML 1.0 SSO protocol.

1. Open the Integrated Solutions Console.
2. Select **Tivoli Federated Identity Manager** > **Federation Management** > **Federation**.
3. Click **Create**.... The *General Information* page appears.
4. Enter a name for the federation, for example, `SAP NetWeaver AS Java`.
5. Select **Identity Provider** as your role.
6. Click **Next**. The *Contact Information* page is displayed.

7. Enter the appropriate company and contact information.

8. Click **Next**. *The Federated User Lifecycle Management* page is displayed.

9. Select **SAML 1.0**.

10. Click **Next**. The *Point of Contact Server* page is displayed.

11. Enter the appropriate **Point of Contact** - this is the machine with the TFIM SPS installed. This article will assume the value `TFIM.myexployer.com`.

12. Secure communications should remain selected unless you want to debug the communication in a non-production environment.

13. Click **Next**. The *SAML Data* page appears.

14. The **SOAP Endpoint** value is automatically created. Modify this value if required.

15. Select a **key** for signing the SAML responses.

16. Click **Next**. The *Security Assertion Markup Language (SAML) Module Configuration* page appears.

17. Accept the default values (unless you know you require a different value).

18. Click **Next**. *The Identity Mapping* page appears.

19. This page defines how a TFIM identity is mapped to an SAP NetWeaver AS Java identity. Listing 1 provides a simple identity mapping rule that maps all TFIM users to the 'Administrator' SAP user. Copy this sample mapping to a file and enter the name of the file in **XSLT File Containing Identity Mapping Rule**. Refer to the IBM Redbook listed in the resources section below for more details on how to create identity mappings.

20. Click **Import File**. The content of the file appears.

21. Click **Next**. The *Summary* page appears.

22. Note the value of the **Succinct Provider ID** and review the remaining values.

23. Click **Finish**. Do not restart the IBM WebSphere® software at this time.

24. Click **Done** or click **Add partner**... to immediately move to the next step.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:stsuuser="urn:ibm:names:ITFIM:1.0:stsuuser" version="1.0">
<xsl:strip-space elements="*" />
<xsl:output method="xml" version="1.0" encoding="utf-8" indent="yes" />
<!--
Initially we start with a copy of the document.
-->
<xsl:template match="@* | node()">
<xsl:copy>
<xsl:apply-templates select="@* | node()" />
```

```
</xsl:copy>

</xsl:template>

<!--

This template replaces the entire Principal element with one that

contains just the value of 'Administrator' and the data type appropriate for
SAML.

-->

<xsl:template match="//stsuuser:Principal">

  <stsuuser:Principal>

        <stsuuser:Attribute name="name"

        type="urn:oasis:names:tc:SAML:1.0:assertion#emailAddress">

              <stsuuser:Value>Administrator</stsuuser:Value>

        </stsuuser:Attribute>

  </stsuuser:Principal>

</xsl:template>

<!--

This template builds a new AttributeList. This involves adding an
AuthenticationMethod attribute

to meet SAML requirements. We assume this is always the "password"
mechanism, regardless of what

the TAM credential actually says.

-->

<xsl:template match="//stsuuser:AttributeList">

  <stsuuser:AttributeList>

        <!-- First the authentication method attribute -->

        <stsuuser:Attribute name="AuthenticationMethod"

        type="urn:oasis:names:tc:SAML:1.0:assertion">


  <stsuuser:Value>urn:oasis:names:tc:SAML:1.0:am:password</stsuuser:Value>

        </stsuuser:Attribute>

  </stsuuser:AttributeList>

</xsl:template>

</xsl:stylesheet>
```

Listing 1. Sample identity mapping

Step 2: Creating a partner

Perform the following steps to create a new partner for the SAP NetWeaver AS Java federation. Ignore the first four tasks if the **Add partner**... button was selected in Step 1 above.

1. Using the Integrated Solutions Console (ISC), select **Tivoli Federated Identity Manager** > **Federation Management** > **Partners**.
2. Click **Create**.... The *Select Federation* page appears.
3. Select the federation created in Step 1.
4. Click **Next**. The *Partner Information* page appears.
5. Enter the appropriate company and contact information.
6. Click **Next**. The *SAML Data* page appears.
7. Enter the **Provider ID**. For a default installation of SAP NetWeaver AS Java, this value should be: `http://sapserver:50000`. Where `sapserver` is the name of the SAP NetWeaver AS Java machine and 50000 is the port on which the SAP NetWeaver AS Java instance is listening.
8. Type the **Assertion Consumer Service URL**. For a default installation of SAP NetWeaver AS Java, this value should be `http://sapserver:50000/saml/receiver`.
9. Ensure the **Partner uses Browser POST profile for Single Sign-On** is not selected.
10. Click **Next**. The *Configure Security Token* page appears.
11. Deselect the **Enable the Signing of Assertions** option.
12. Click **Next**. The *Identity Mapping* page appears.
13. Simply click **Next** as we will use the identity mapping created for the federation. The *Summary* page appears.
14. Review the settings and click **Finish** when complete.
15. Click **Enable Partner**.

Step 3: Restart WebSphere

In the ISC, click **Restart WebSphere** to enable the changes to the TFIM configuration.

Step 4: Restart WebSEAL

Restart WebSEAL to enable the new TFIM partner.

Configuring SAP NetWeaver AS Java

This section covers the steps required to configure SAP NetWeaver AS Java to support SAML assertions.

Step 1: Changing the Startup Mode for the SAML Service

Perform the following steps using the Configuration Adapter in the Visual Administrator to change the startup mode for the SAP SAML Service.

1. Select **Server** > **Services** > **Configuration Adapter**.

2. Expand **Configurations - cluster_data - server - cfg - services**.

3. Switch to edit mode. Click **Yes**.

4. Select **Propertysheet tc~sec~saml~service-runtime** and click the pencil representing **Show the details of the selected node**. The *Change Configuration* page appears.

5. Select **start-up mode**. The **Change property entry** page appears.

6. In the **Custom** field, enter the value always and click **Apply custom**. You return to the *Change Configuration* page.

7. Click **OK**.

8. Restart the J2EE™ Engine server process.

Step 2: Creating a destination

Perform the following steps using Destinations in the Visual Administrator to create a new destination. The destination defines the parameters in order to connect the TFIM federation.

1. Expand **Destinations** - **HTTP**.

2. Click **New** and enter a name for the new destination, for example, `TFIM`.

3. Click **OK**.

4. Enter the URL of the TFIM SOAP Endpoint. The value can be retrieved by viewing the properties of the federation in the ISC. Using the example values above, the entry contains: `http://TFIM.myemployerx.com/TFIM/sps/SAP_AS-Java/saml/soap`.

5. Select the appropriate **Authentication** mechanism. The mechanism selected will depend on your deployment scenario. For testing purposes, assuming WebSEAL is used as the authentication server, the `sec_master` credentials could be used with `BASIC` authentication.

6. Click **Save**.

Step 3: Configuring the SAML parameters

Perform the following steps using the Configuration Adapter in the Visual Administrator to change the SAML parameters.

1. Expand **Configurations** > **SAML** > **Configuration**.

2. Switch to Edit mode. Click **Yes**.

3. Select **PartnersInbound** and click **Create** a node below the selected node. The *Create* page appears.

4. Enter a **Name** for the partner, for example, `TFIM`.

5. Click **Create**. A new node is created.

6. Expand the new node.

7. Double-click **DestinationName**. Enter the name of the destination created in the previous step.

8. Double-click **SourceID**. Enter the value of the TFIM federation Source ID **prefixed with B64**: . This is the Source ID you noted earlier when you configured the identity provider federation on the TFIM system.

9. For testing purposes, the PermitInsecureConnections parameter, located under **Configurations** > **SAML** > **Configuration - Settings** can be set to `true`. In a production environment, this value should be set to `false`.

Step 4: Adjusting the Login Module Stack for using SAML

Perform the following steps using the **Security Provider** service in the **Visual Administrator** to adjust the login modules that apply to the application that is to be configured for SAML assertions. Perform these steps for each template or application that is to support SAML assertions, e.g. the basic template.

1. Select the **Authentication** tab.

2. Click **Add New**. The *Available Login Modules* page appears.

3. Select **SAMLLoginModule**. Click **OK**. The SAML Login Module is added to the end of the list of Login Modules.

4. Select SAMLLoginModule and click **Modify**. The *Edit Login Module* page is page is displayed.

5. Set the Position to `1`.

6. Ensure the Flag is set to `SUFFICIENT`.

7. Click **OK**.

Testing the configuration

Perform the following tasks to test the configuration of SSO using SAML assertions. This test performs sign-on to the SAP NetWeaver Portal. If SAP NetWeaver Portal is not available the User Administration servlet (`/useradmin/userAdminServlet`) can be used along with the Basic authentication template.

1. Adjust the Login Module Stack for the **ticket** template as outlined in Step 4: Adjusting the Login Module Stack for using SAML.

2. Navigate to the SSO trigger. Using the example throughout this article, the SSO trigger would be: `http://TFIM.myemployerx.com/TFIM/sps/SAP_AS-Java/saml/login?TARGET=http://sapserver:50000/irj/portal`.

3. Authentication to TFIM is required because the user has not yet authenticated.

The Portal welcome page should be displayed for the *Administrator* SAP user.

Resources

- Refer to the Federated Identity Manager online documentation for complete details on how to configure SAML federations: http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/index.jsp?toc=/com.ibm.tivoli.TFIM.doc/toc.xml

- The Redbook Federated Identity Management and Web Services Security with IBM Tivoli Security Solutions covers important aspects of utilizing the Tivoli integrated identity management architecture in order to build and deploy the Tivoli Federated Identity Management and Web Services Security components: http://www.redbooks.ibm.com/abstracts/sg246394.html?Open

- The article Using SAML Assertions for Single Sign-On (SAP Knowledge Warehouse) provides complete information on how to configure SAP NetWeaver AS Java for SAML assertions: http://help.sap.com/saphelp_nw04/helpdata/en/94/695b3ebd564644e10000000a114084/content.htm

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.