Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

Dipak K. Pal (dipakpal@in.ibm.com)  
Software Developer  
IBM  

13 May 2015

This four-part tutorial series describes a Salesforce® federated single sign-on solution using WebSphere® DataPower® as an identity provider. Part 2 describes how to implement an identity provider initiated single sign-on to Salesforce using an encrypted and signed SAML assertion.

Introduction

Assuming you have already read the first article in this series, this tutorial (Part 2) will not recap the fundamentals of the Salesforce federated single sign-on (SSO). This tutorial primarily focuses on how to implement the identity provider initiated SSO to Salesforce using IBM® WebSphere DataPower (hereafter called DataPower), powered by an encrypted and signed SAML assertion. Figure 1 and Figure 2 show a high level overview of this solution.

Figure 1. Overview of identity provider initiated SSO login to Salesforce

© Copyright IBM Corporation 2015  
Trademark
Single sign-on setup on the Force.com platform

This section provides the steps to implement the SSO setup in the Force.com platform. If you have already implemented an example illustrated by any of the tutorials in this series, you may not need to repeat the same steps again. However, make sure you have configured the additional setup required for this example.

Step 1: Create Salesforce user accounts

1. First, create a user account (Salesforce Administrator) at the Salesforce developer site if you do not have one already. Figure 3 shows the Salesforce user (Developer Edition) sign-up process.

Figure 3. Salesforce user account (Developer Edition) creation

2. Login to Salesforce by entering the credentials obtained during the sign-up process and create another user account. Make sure you enter a federation ID as shown in Figure 4 to enable the SSO login for this user account. Figure 5 shows the Salesforce user details page. The examples illustrated by this tutorial series use the following sample user account for the SSO setup. We highly recommend that you do not enable the SSO feature for your organization's Salesforce administrator because any outage of the identity provider (DataPower) application will prevent all administrative related activities.
User ID: sdfc.user@sdfc.com
Federation ID: sdfc.user@sso.sdfc.com

Figure 4. Enter a Federation ID to enable SSO

Figure 5. User account (Standard Platform User) details at Salesforce

Step 2: Create a self-signed or CA-signed certificate

Certificates are used for authenticated SSO with an external identity provider. You need this Salesforce certificate to encrypt the SAML assertion on the identity provider side. Once you create the certificate, download and save it to your local system. The examples illustrated by this tutorial series use the following self-signed certificate (Label: Salesforce SSO) shown in Figure 6 and Figure 7.
Step 3: Manage the Salesforce domain

Using "My Domain", you can define a custom Salesforce domain name. This domain name helps you better manage the login and authentication for your organization in several ways. Refer to the Salesforce documentation for domain management details. The examples illustrated by this tutorial series use the following domain name. We suggest that you use a suitable domain name for your organization. An example of a sample domain name is https://dipak-sdfc-dev-ed.my.salesforce.com/.
Step 4: Set up SSO settings at Salesforce

1. Login as Salesforce administrator and go to **Setup > Security Controls > Single Sign-On Settings**. Make sure the **SAML Enabled** checkbox is checked in the **Federated Single Sign-On Using SAML** tab, and click the **New** button in the **SAML Single Sign-On Settings** tab. Figure 9 and Figure 10 show the Salesforce SSO setup used by this tutorial. Prior to this SSO setup at Force.com platform, make sure you have already created the DataPower crypto key and associated certificate described in **Step 5**.

2. Upload the DataPower certificate as an **Identity Provider Certificate** such that the Force.com platform can validate the signature associated with the SAML assertion.

3. Use the self-signed or CA-signed certificate generated in **Step 2** for both the **Signing Certificate** and **Assertion Decryption Certificate** fields. This **Signing Certificate** is primarily used for the service provider initiated SSO where the SAML authentication request is signed by this certificate. The identity provider (for example, DataPower) encrypts the SAML assertion response by using this Salesforce certificate. Enter the appropriate values in the **Issuer** and **Entity Id** fields. The SAML assertion used in this tutorial is encrypted and contains the **Federation ID** of the login user in the **NameIdentifier** (**NameID**) field in the Subject statement.

4. Once you save the SAML SSO settings, note down the Salesforce login URL, for example: https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC&sc=0LEF0000000Kytb.
Configure DataPower as a single sign-on identity provider

This section shows how to develop the necessary DataPower artifacts that primarily consist of an HTTPS Front Side Handler, a Multi-Protocol Gateway that includes the required processing policy rules, and an AAA policy.

**Step 5: Manage the DataPower crypto key**

1. Generate a crypto key and a self-signed (or CA-signed) certificate by using the DataPower crypto tools, as shown in Figures 11 to 13, which are required for a secured HTTP (SSL) connection and signature generation.

2. Download the certificate from the `temporary://` directory once you create it. If you have already implemented the example illustrated in Part 1 of this tutorial series, you may reuse the same crypto configuration. Make sure you configure a crypto certificate and import the Salesforce certificate downloaded in **Step 2**, which is required to encrypt the SAML assertion.
3. Create a crypto identification credentials and add the DataPower crypto key (Figure 14), which is required for the SSL setup.
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

Figure 14. Configure crypto identification credentials with the DataPower crypto key

4. Optionally, create a crypto validation credentials and add client (for example, browser) certificates, as shown in Figure 15. You may not require a crypto validation credentials if you do not enforce the SSL client certificate validation. You can optionally add a Salesforce certificate too, which is not required for this example. However, it will be required if you implement a service provider initiated SSO to validate the signature contained by the SAML authentication request.

Figure 15. Configure crypto validation credentials

5. Create a crypto profile and add the crypto identification credentials and crypto validation credentials (optional), as shown in Figure 16.
6. Create an SSL proxy profile with the **Reverse** SSL Direction (Figure 17) and use the crypto profile created in the previous step.

**Figure 17. Configure the SSL proxy profile**

**Step 6: Create an HTTPS Front Side Handler**

Create an HTTPS Front Side Handler (Figure 18) to accept the SSO requests from the browser.
Figure 18. Configure an HTTPS Front Side Handler

Step 7: Create a Multi-Protocol Gateway
Create a Multi-Protocol Gateway with a dynamic back-end routing (Figure 19). Make sure you select **Non-XML** for both the request and response data type to deal with the HTTP form data and HTML data. The next step, Step 8, describes the Multi-Protocol Gateway Policy details.

Figure 19. Configure a Multi-Protocol Gateway

Step 8: Create a Multi-Protocol Gateway Policy

1. Create a Multi-Protocol Gateway Policy with following processing rules.
   - **First rule:** This rule deals with the favicon.ico request from the browser (Figure 20).
     - **Direction:** Client to server.
     - **Action:** Match
       - **Matching Type:** URL
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

URL Match: /favicon.ico
2. **Action**: Advanced [Set Variable]
   - **Variable Name**: var://service/mpgw/skip-backside
3. **Variable Assignment**: 1
4. **Action**: Result

**Figure 20. Multi-Protocol Gateway Policy rule to process favicon.ico request from the browser**

- **Second Rule**: This rule processes the SSO login request from the browser that is initiated by the user by entering the DataPower SSO login URL (Figure 21).

**Figure 21. Multi-Protocol Gateway Policy rule to process SSO login request from the browser**

**Direction**: Client to server.
1. **Action**: Match
   - **Matching Type**: URL
URL Match:/sso/salesforce/login

Note: Use an appropriate URL as per your organization's standard.

2. **Action:** AAA
   Create an AAA Policy as described in Step 9.

3. **Action:** Transform (Figure 22)
   Use the provided XSLT in the Download section of the tutorial (remove-specific-attribute.xsl) shown in Listing 1 to remove the @NotBefore attribute from the <SubjectConfirmationData> element as the Salesforce SAML Assertion Validator framework does not allow this attribute.

**Figure 22. Use XSLT in Transform action to remove an attribute from SAML Assertion XML**

Listing 1. XSLT [remove-specific-attribute.xsl] to remove a specific attribute from SAML Assertion XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0" extension-element-prefixes="dp"
    exclude-result-prefixes="dp" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:dp="http://www.datapower.com/extensions">
    <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="no"/>
    <xsl:strip-space elements="*"/>
    <xsl:template match="@* | node()">
        <xsl:copy>
            <xsl:apply-templates select="@* | node()"/>
        </xsl:copy>
    </xsl:template>
    <xsl:template match="/*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:protocol' and
        local-name()='Response']/*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:assertion' and
        local-name()='Assertion']/*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:assertion'
        and local-name()='Subject']/*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:assertion'
        and local-name()='SubjectConfirmation']/*[namespace-uri()=
        'urn:oasis:names:tc:SAML:2.0:assertion' and local-name()=
        'SubjectConfirmationData']/@NotBefore"/>
</xsl:stylesheet>
```

4. **Action:** Encrypt
This action encrypts the SAML assertion by using the Salesforce certificate. Create a document crypto map by specifying the following XPATH (Listing 2), as shown in Figure 23 and Figure 24.

**Listing 2. XPATH used at document crypto map**

```xml
```

**Figure 23. Encrypt SAML Assertion**

![Configure Encrypt Action](image1)

**Figure 24. Document crypto map for encryption**

![Configure Document Crypto Map](image2)

5. **Action:** Sign

This action signs the encrypted SAML assertion XML (Figure 25). Use the following XPATH in Listing 3.

**Listing 3. XPATH in SAML Assertion XML needs to be signed**

```xml
//*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:protocol' and local-name()='Response']
```
6. **Action**: Transform (Figure 26)

Upon successful authentication, this **Transform Action** generates an HTML page, which includes an HTML FORM with an HTTP POST Action, and contains a base-64 encoded SAML Assertion XML as a hidden parameter. Figure 36 shows a sample HTML page responded by the DataPower after successful authentication. Use the XSLT (**SSO-Generate-HTML-Idp.xsl**) shown in Listing 4 and also provided with this tutorial in the **Download** section. Make sure you update the XSLT with the appropriate Salesforce login URL (XSLT Variable: `sdfcLoginURL`) obtained in Step 4.
Listing 4. XSLT [SSO-Generate-HTML-Idp.xsl] to generate HTML page

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" extension-element-prefixes="dp"
exclude-result-prefixes="dp" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:dp="http://www.datapower.com/extensions">
<xsl:output method="html" omit-xml-declaration="yes"/>
<xsl:param name="sdfcLoginURL"
select="'https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC
&sc=0LEF0000000Kytb'"/>
<xsl:variable name="signedSamlSerialized">
<dp:serialize select="/" omit-xml-decl="yes"/>
</xsl:variable>
<xsl:variable name="base64Samldata"
select="dp:encode($signedSamlSerialized, 'base-64')"/>
<html>
<head>
<title>Login to Salesforce</title>
<meta http-equiv="Cache-Control" content="no-cache"/>
<meta http-equiv="Pragma" content="no-cache"/>
<meta http-equiv="Expires" content="-1"/>
<meta http-equiv="X-UA-Compatible" content="IE=8"/>
<style type="text/css">
.body{background-color: #3EB1D0;}
.data{
  background-color: #C6EDEC;
  margin-top: 10%;
  margin-left: 20%;
  margin-right: 20%;
  font-family: verdana;
  color: #0A6DA8;}
.headline{font-size: 50px;}
.button{
  background-color: #F5DAA3;
  color: #0D75AA;
  font-size: x-large;
}
</style>
</head>
<body class="body">
<div align="center" class="data">
<form method="POST" enctype="application/x-www-form-urlencoded">
<xsl:attribute name="action">
<xsl:value-of select="$sdfcLoginURL"/>
</xsl:attribute>
<div class="headline">
<i><b>Salesforce Login</b></i>
</div>
<div><h2>Welcome<span><i><b>
<xsl:value-of select="dp:variable('var://context/WSM/identity/username')"/></b></i>
</span></h2></div>
<div>
<h4>
<p>Clicking Submit button will redirect you to Salesforce home page</p>
<p>Please contact <a href="mailto:support@example.com">support@example.com</a>
if you experience any issues during Salesforce Login</p>
</h4>
</div>
<input type="hidden" name="idpConfig.recipient">
<xsl:attribute name="value">
<xsl:value-of select="$sdfcLoginURL"/>
</xsl:attribute>
</form>
</div>
</body>
</html>
```
7. **Action:** Advanced [Set Variable] (see Figure 27)
   
   **Variable Name:** var://service/mpgw/skip-backside
   
   **Variable Assignment:** 1

   **Figure 27. Advanced action to skip back-end routing**

8. **Action:** Result (Figure 28)
Step 9: Create an AAA policy

1. Create an AAA policy to authenticate the user's credentials and to generate an SAML assertion as a result of successful authentication. The example illustrated by this tutorial uses a basic HTTP authentication method and validates the user's credentials using a DataPower AAA info file shown in Listing 5. However, you can implement an appropriate authentication method as per your organization's standard. Figures 29 to 32 describe this AAA policy configuration detail. If you have already implemented the example illustrated in Part 1 of the series, you can just reuse the same AAA policy without any changes.
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

Figure 30. AAA policy – Extract identity from request

Figure 31. AAA policy – Authenticate the user

Figure 32. AAA policy – Authorize a request
2. Configure the AAA Access Control Policy to generate an SAML assertion response upon successful authentication. Figure 33 shows an AAA Access Control Policy page enabling an SAML assertion. Make sure you use the same SAML Issuer value (for example, https://datapowersso.ibm.com/) configured from the Salesforce SSO settings (Step 4). Also set the Entity ID value from the Salesforce SSO settings (Step 4) in the SAML Audience field in the AAA Access Control Policy page. Use the Salesforce login URL for both the SAML Recipient and Response Destination fields. Create an SAML attribute definitions as shown in Figure 34.

**Figure 33. AAA policy – Generate SAML assertion response**

![AAA policy – Generate SAML assertion response](image)

**Figure 34. AAA policy – SAML attributes**

![AAA policy – SAML attributes](image)

**Listing 5. Sample AAA Policy Info file**

```xml
<?xml version="1.0" encoding="utf-8"?>
<AAAInfo xmlns="http://www.datapower.com/AAAInfo">
  <FormatVersion>1</FormatVersion>
  <!-- Add your SAML attributes here -->
</AAAInfo>
```
<Filename>local:///AAAInfoPolicy.xml</Filename>
<Summary>AAA Policy Info XML file</Summary>
<Authenticate>
<Username>dipakpal</Username>
<Password>dipakpal-password</Password>
<OutputCredential>dipakpal@sso.sdfc.com</OutputCredential>
</Authenticate>
<Authenticate>
<Username>sdfc.user</Username>
<Password>sdfc-password</Password>
<OutputCredential>sdfc.user@sso.sdfc.com</OutputCredential>
</Authenticate>
<Authenticate>
<CustomToken>https://dipak-sdfc-dev-ed.my.salesforce.com</CustomToken>
<OutputCredential>saml.auth.request.issuer</OutputCredential>
</Authenticate>
<Authenticate>
<DN>CN=Alice, O=DataPower, L=Cambridge, ST=MA, C=US</DN>
<OutputCredential>admin</OutputCredential>
</Authenticate>
<Authenticate>
<IPHost>127.0.0.1</IPHost>
<OutputCredential>admin</OutputCredential>
</Authenticate>
<Authenticate>
<IPHost>::1</IPHost>
<OutputCredential>admin</OutputCredential>
</Authenticate>
<Authenticate>
<Username>guest</Username>
<Password>guest</Password>
<OutputCredential>user</OutputCredential>
</Authenticate>
<Authorize>
<InputCredential>dipakpal@saml.sdfc.com</InputCredential>
<InputResource>/sso/*</InputResource>
<Access>allow</Access>
</Authorize>
<Authorize>
<InputCredential>sdfc.user@sso.sdfc.com</InputCredential>
<InputResource>/sso/*</InputResource>
<Access>allow</Access>
</Authorize>
<Authorize>
<InputCredential>saml.auth.request.issuer</InputCredential>
<InputResource>/sso/*</InputResource>
<Access>allow</Access>
</Authorize>
<Authorize>
<InputCredential>guest</InputCredential>
<InputResource>/sso/salesforce</InputResource>
<Access>deny</Access>
</Authorize>
<Authorize>
<InputCredential>admin</InputCredential>
<InputResource>/sso/salesforce</InputResource>
<Access>allow</Access>
</Authorize>
<Authorize>
<InputCredential>admin</InputCredential>
<Version>3</Version>
<SerialNumber>0</SerialNumber>
<SignatureAlgorithm>sha1WithRSAEncryption</SignatureAlgorithm>
<Issuer>C=US, ST=MA, L=Cambridge, O=DataPower, CN=Alice</Issuer>
<NotBefore>2002-11-23T01:15:33Z</NotBefore>
<NotAfter>2012-11-23T01:15:33Z</NotAfter>
<Subject>C=US, ST=MA, L=Cambridge, O=DataPower, CN=Alice</Subject>
Step 10: Test the single sign-on

Enter your identity provider (DataPower) SSO login URL (for example, https://<dp-host:port>/sso/salesforce/login) in the web browser as shown in Figure 35. Upon successful authentication, DataPower responds back with an SSO re-direct (HTTP POST binding) page as shown in Figure 36, which includes an HTML FORM with an HTTP POST action, and contains a base-64 encoded SAML assertion XML as a hidden parameter. It requires the user to submit the HTML page by clicking the Login to Salesforce button, which results in submitting an HTTP POST request to the Salesforce login URL (for example, https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC&sc=OLEF0000000Kytb).

An SAML assertion contains a federation ID (for example, sdfc.user@sso.sdfc.com) from the identity provider. Salesforce validates the message integrity using the embedded signature in the SAML assertion against the DataPower certificate, which is already uploaded during the SSO
setup process. Upon a successful signature validation, it decrypts the encrypted assertion and processes the SAML assertion statement, extracts and validates the Federation ID, and redirects to the landing page after a successful validation process. Figure 37 shows a sample Salesforce landing page after a successful SSO login attempt.

Figure 37. Sales landing page after successful SSO login

![Salesforce landing page after successful SSO login](image)

**Step 11: Debug and troubleshoot guide**

We assume you are familiar with common DataPower debugging and troubleshooting techniques, therefore this tutorial will primarily focus on Salesforce.

You can capture the base64 encoded SAML assertion data using **Developer Tools** in **Google Chrome** as shown in Figure 38 and Figure 39. Though you can get this data from the HTML page source, using **Developer Tools** in **Google Chrome** or **Internet Explorer** will help you analyze HTTP events including their request/response header and data structure. You can decode (base-64) this data by using any online tool (such as [http://www.string-functions.com/base64decode.aspx](http://www.string-functions.com/base64decode.aspx)) to get an SAML assertion XML. Listing 6 shows a sample encrypted and signed SAML assertion XML generated by this example.

Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

Figure 38. Start Developer Tools in Google Chrome

Figure 39. Capture base-64 encoded SAML assertion XML from HTML

Listing 6. Sample SAML assertion XML (encrypted and signed)

```xml
<samlp2:Response Version="2.0" ID="SAML-12b7a404-a031-4ecf-a163-5aaa553774b"
Destination="https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC"
xmlns:samlp2="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml2:Issuer
    xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
    https://datapowersso.ibm.com/
  </saml2:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod
        Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
      <SignatureMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <Reference URL="#SAML-12b7a404-a031-4ecf-a163-5aaa553774b"
        <Transforms>
```
Login to Salesforce as an administrator and validate the base-64 encoded SAML assertion to check whether the SSO identity provider (DataPower) generated the correct SAML assertion as expected by Salesforce. Figures 40 to 43 show the SAML assertion validation using the Force.com SAML Assertion Validator Framework.
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion

Figure 40. Start the SAML Assertion Validator Framework in Salesforce

Figure 41. Enter the base-64 encoded SAML assertion into the Salesforce SAML Validator
While testing the SSO login, you can check the login history at Force.com from a separate login as a Salesforce administrator. Figure 44 shows a login history page on the Force.com site, which can help you troubleshoot SSO login failures.
Conclusion

Organizations that have fewer Salesforce users may not require this single sign-on solution as they can continue to use the standard Salesforce login process. However, organizations with a large number of Salesforce users can leverage this solution by implementing a DataPower based SSO infrastructure for all of their internal applications as well as cloud-based and external applications (Force.com), which support Federated Identity Management using an external SSO identity provider.

To continue with this series, see the following tutorials:

- Part 1: Identity provider initiated SSO using a signed SAML assertion
- Part 3: Service provider initiated SSO using a signed SAML assertion
- Part 4: Service provider initiated SSO using an encrypted and signed SAML assertion
Downloads

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code sample</td>
<td>code_sample_part2.zip</td>
<td>7KB</td>
</tr>
</tbody>
</table>
Resources

- Single Sign On (SSO) Basics
- SAML 2.0 Basics
- Single Sign On with SAML on Force.com
- IBM WebSphere DataPower Knowledge Center
- Hands-on Training: Enable Single Sign-on with SAML and Salesforce Identity
- Single Sign-On Best Practices (Salesforce)
- Setting up Single Sign-On (Salesforce)
- developerWorks WebSphere zone
- developerWorks WebSphere DataPower discussion forum
About the author

Dipak K. Pal

Dipak K. Pal works as an Integration Designer in the IBM Global Business Services team. He has more than 10 years of IT experience in enterprise integration, cloud integration, service-oriented architecture, and J2EE-based application design and development. He is currently working on a cloud integration project, integrating several cloud applications with various on-premise applications using IBM WebSphere Cast Iron, IBM Integration Bus, and IBM WebSphere DataPower.

© Copyright IBM Corporation 2015
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)