Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a 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 1 of this series discusses how to implement an identity provider initiated single sign-on to Salesforce using a signed SAML assertion.

View more content in this series

Introduction

As more institutions and organizations offer services and collaborations online, employees need to access both on-premises as well as cloud-based applications to do their day-to-day work. This demands implementing a single sign-on (SSO) infrastructure enabling users to sign in once to access all authorized internal and external resources and applications. Organizations with a large Salesforce user base can leverage their existing SSO infrastructure to implement SSO to the Force.com platform, which supports Federated Identity Management powered by an external SSO identity provider. The Force.com platform supports both delegated and federated authentication for SSO.

This tutorial series focuses on federated authentication only, which has several advantages over delegated authentication and is widely accepted in the IT industry. Federated authentication does not validate the user’s actual password on the Force.com platform. Instead, the platform receives a Security Assertion Markup Language (SAML) assertion in an HTTP POST request. The SAML assertion has a limited validity period, contains a unique identifier, is digitally signed, and is optionally encrypted. If the assertion is still valid within its validity period, has an identifier that has not been used before, and has a valid signature from a trusted identity provider, the user is granted access to the requested resource or page. Using IBM® WebSphere DataPower (hereafter called DataPower) as the SSO identity provider, enterprises can implement a single registry of
user identities with a centralized management interface for all of their internal applications as well as external applications (for example, Force.com) that support federated single sign-on.

This series consists of the following four tutorials:

1. Part 1: Identity provider initiated SSO using a signed SAML assertion (this tutorial)
2. Part 2: Identity provider initiated SSO using an encrypted and signed SAML assertion
3. Part 3: Service provider initiated SSO using a signed SAML assertion
4. Part 4: Service provider initiated SSO using an encrypted and signed SAML assertion

How federated authentication works

SAML provides a secure, XML-based solution for exchanging user security information between an identity provider (such as DataPower) and a service provider (such as Force.com). Besides the identity provider initiated single-sign-on, SAML 2.0 also supports service provider initiated single-sign-on. This allows service providers like Force.com to query the identity provider for authentication. SAML 2.0 also adds a useful feature called "single-logout", which defines a mechanism for logging out all of the service providers quickly and easily.

Parties involved in an SAML transaction

There are three parties involved in an SAML transaction:

1. **Identity provider (the asserting party):** The identity provider is the authority system that provides the SAML assertion. In this tutorial series, DataPower is the identity provider.
2. **Service provider (the relying party):** The service provider is the system, in this case Force.com, which trusts the identity provider's user information and uses the data to provide access to the requested resource or page.
3. **User (the subject of the assertion):** This is the user who needs a single sign-on to Force.com.

Use cases for SAML

There are two important use cases for SAML:

1. **Identity provider initiated SSO:** The user directly requests the SSO login to the identity provider, logs in, and is then redirected (HTTP POST binding) to a predefined landing page at the service provider.
2. **Service provider initiated SSO:** The user requests access to a protected resource or page from the service provider. The service provider redirects (HTTP POST or REDIRECT binding) the request to the identity provider with an SAML authentication request. The identity provider validates the SAML authentication request, challenges the user for login credentials if the user is not logged in already, and sends the SAML assertion response to service provider using an HTTP POST binding. The service provider validates the SAML assertion and redirects to the resource or page that was originally requested.

Force.com supports both of these use cases. However, the scope of this tutorial (Part 1) is limited to the identity provider initiated SSO login only, which uses a signed SAML assertion.
The SAML assertion response from the identity provider contains the header information, the subject, and the assertion statements about the subject in the form of attributes and conditions. The SAML assertion used in this example contains a federated ID from the identity provider, which is unique within the Force.com platform and is mapped to a Force.com user, as shown in Figure 1 and Figure 2.

**Figure 1. Identity provider initiated SSO login to Salesforce**

![Identity provider initiated SSO login to Salesforce](image)

**Figure 2. Identity provider initiated SSO login process**

![Identity provider initiated SSO login process](image)

**Single sign-on setup on the Force.com platform**

A step-by-step guide for single sign-on setup at Force.com platform is described in this section.

**Step 1: Create Salesforce user accounts**

1. First, create a user account (Salesforce Administrator) at the [Salesforce developer site](https://developer.salesforce.com), if you do not have one already. Figure 3 shows the Salesforce user (Developer Edition) sign-up process.
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 a Salesforce user detail page. 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 single sign-on feature for your organization's Salesforce administrator because any outage of the identity provider (DataPower) application will prevent all admin-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

1. Certificates are used for authenticated single-sign-on with an external identity provider. You may not require a Salesforce certificate for an identity provider initiated SSO if the SAML assertion generated by the identity provider is not encrypted.

2. Once you create the certificate, download and save it into your local system. 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 (see Figure 8). A custom 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. 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-sdfe-dev-ed.my.salesforce.com/.
Step 4: Set up SSO settings at Salesforce

1. Login as a Salesforce administrator and go to Setup > Security Controls > Single Sign-On Settings. Make sure the SAML Enabled checkbox is checked at the Federated Single Sign-On Using SAML tab, and click the New button at 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 a DataPower crypto key and the 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 as "Signing Certificate". Enter the appropriate values for the Issuer and Entity Id fields. The SAML assertion used in this example is not encrypted and it contains the Federation ID of the login user at the NameIdentifier (NameID) field in the Subject statement.

4. Once you save the SAML single signon settings (Figure 9), note the Salesforce login URL (for example, https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC).
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a signed SAML assertion

Configure DataPower as single sign-on identity provider

The following steps describe how to develop the necessary DataPower artifacts that primarily consist of a 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 using the DataPower crypto tools, as shown in Figure 11 and Figure 12, which are required for a secured HTTP (SSL) connection and signature generation. Download the DataPower Certificate from the `temporary://` directory once you create it.
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a signed SAML assertion

2. Create a crypto certificate as shown in Figure 13 and import the Salesforce certificate downloaded in Step 2. However, you can skip creating this crypto certificate because the SAML assertion generated by this example is not encrypted.

Figure 13. Configure crypto certificate and import the Salesforce Certificate

3. Create a Crypto Identification Credentials with the DataPower crypto key, which is required for the SSL setup (Figure 14).
Figure 14. Configure crypto identification credentials with the DataPower crypto key

4. Optionally, create a crypto validation credentials and add required 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 associated with 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 (Figure 16). However, you can skip adding the crypto validation credentials if you do not enforce the client (for example, browser) SSL certificate validation.
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 a HTTPS front side handler (Figure 18) to accept the SSO requests from the browser.
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 as well as the HTML data. The next step, Step 8, describes the Multi-Protocol Gateway Policy details.

Step 8: Create a Multi-Protocol Gateway policy
Create a Multi-Protocol Gateway Policy with the following processing rules:

- **First rule:** This rule deals with the favicon.ico request from the browser (Figure 20).
  - **Direction:** Client to Server.
  - **Action:** Match
Figure 20. Multi-Protocol Gateway Policy rule to process the favicon.ico request from the browser

- **Second rule:** This rule processes the SSO login request from the browser that is initiated by the user using the DataPower SSO login URL directly (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 based on your organization’s standard.

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

3. **Action:** Transform (see Figure 22)  
   Use the XSLT (remove-specific-attribute.xsl) shown in Listing 1 to remove the @NotBefore attribute from the <SubjectConfirmationData> element because 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**

   ![Configure Transform Action]

   ![WebSphere DataPower XSL2]

   ![Transformer Action]

   ![XML Style Sheet]

   **Listing 1. XSLT [remove-specific-attribute.xsl] to remove a specific attribute from the 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:copy>
   <xsl:apply-templates select="@* | node()"/>
   </xsl:copy>
   </xsl:stylesheet>
   ```

   Use the XSLT shown in Listing 1 to remove the @NotBefore attribute from the SAML assertion XML.

4. **Action:** Sign (see Figure 23)
This action digitally signs the SAML assertion XML (XPATH: /*[namespace-uri()='urn:oasis:names:tc:SAML:2.0:protocol' and local-name()='Response']) by using the DataPower crypto key and certificate generated in Step 5.

**Figure 23. Sign SAML assertion XML**

5. **Action**: Transform (see Figure 24)
   
   This **Transform** action generates a HTML page that includes an HTML FORM with an HTTP POST Action and contains a base-64 encoded SAML assertion XML as a hidden parameter. Figure 34 shows a sample HTML page responded by DataPower after a successful authentication. Use the XSLT (**SSO-Generate-HTML-Idp.xsl**) shown in Listing 2 for this action. Make sure you update the XSLT with the appropriate Salesforce login URL (XSLT Variable: `sdfcLoginURL`) obtained in Step 4.

**Figure 24. XSLT Transformation to generate HTML response**
Listing 2. 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:template match="/">
    <xsl:param name="sdfcLoginURL" select="'https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC'"/>
    <xsl:variable name="signedSamlSerialized">
      <dp:serialize select="/" omit-xml-decl="yes"/>
    </xsl:variable>
    <xsl:variable name="base64Samldata" select="dp:encode($signedSamlSerialized,'base-64')"/>
    <html>
      <head>
        <title>Login to Salesforce</title>
        <meta http-equiv="Cache-Control" content="no-cache"/>
        <meta http-equiv="Pragma" content="no-cache"/>
        <meta http-equiv="Expires" content="-1"/>
        <meta http-equiv="X-UA-Compatible" content="IE=8"/>
        <style type="text/css">
          .body{background-color: #3EB1D0;}
          .data{
            background-color: #C6EDEC;
            margin-top: 10%;
            margin-left: 20%;
            margin-right: 20%;
            font-family: verdana;
            color: #0A6DA8;
          }
          .headline{font-size: 50px;}
          .button{
            background-color: #F5DAA3;
            color: #0D75AA;
            font-size: x-large;
          }
        </style>
      </head>
      <body class="body">
        <div align="center" class="data">
          <div class="headline"><i><b>Salesforce Login</b></i></div>
          <div><h2>Welcome</h2></div>
          <div><h4>
            Clicking Submit button will redirect you to Salesforce home page
          </h4></div>
          <div class="button" align="center">
            <input type="hidden" name="idpConfig.recipient">
            <xsl:attribute name="value">
              <xsl:value-of select="$sdfcLoginURL"/>
            </xsl:attribute>
            <input type="submit" value="Login"/>
          </div>
          <p>Please contact support@example.com if you experience any issues during Salesforce Login</p>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a signed SAML assertion

6. **Action:** Advanced [Set Variable] (see Figure 25)
   **Variable Name:** var://service/mpgw/skip-backside
   **Variable Assignment:** 1

   **Figure 25. Advanced Action to skip back-end routing**

![Configure Set Variable Action](image)

7. **Action:** Result (see Figure 26)
**Step 9: Create an AAA policy**

1. Create an AAA policy to authenticate the user and to generate an SAML assertion as the result of a successful authentication. Examples illustrated by this tutorial series use a basic HTTP authentication method and validate user credentials using the DataPower AAA info file as shown in Listing 3. However, you can implement an appropriate authentication method per your organization’s standard. Figures 27 to 30 describe the AAA policy configuration details.
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a signed SAML assertion

Figure 28. AAA Policy – Extract identity from the request

Figure 29. AAA Policy – Authenticate the user
Figure 30. AAA Policy – Authorize a request

Listing 3. Sample AAA Policy Info file

```xml
<?xml version="1.0" encoding="utf-8"?>
<AAAInfo xmlns="http://www.datapower.com/AAAInfo">
  <FormatVersion>1</FormatVersion>
  <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>
</AAAInfo>
```
2. Configure this AAA Access Control Policy to generate an SAML assertion response as a result of the successful authentication. Figure 31 shows the AAA Access Control Policy page enabling the SAML assertion. Make sure you use the same SAML Issuer (such as https://datapowersso.ibm.com/) configured during the Salesforce SSO settings (Step 4). Enter the Entity ID value from the Salesforce SSO settings (Step 4) on the "SAML Audience" field. Use the Salesforce login URL for both the "SAML Recipient" and "Response Destination" fields. Create the SAML attribute definitions as shown in Figure 32.
Step 10: Test the single sign-on

Enter your identity provider (such as DataPower) SSO login URL (such as https://<dp-host:port>/sso/salesforce/login) in the web browser as shown in Figure 33. Upon successful authentication, DataPower responds back with a SSO re-direct (HTTP POST binding) page as shown in Figure 34, which includes an HTML FORM with an HTTP POST Action and contains a base-64 encoded SAML assertion XML and RelayState as hidden parameters. It requires a user interaction to submit the HTML page by clicking the Login to Salesforce button, which results into submitting an HTTP POST request to the Salesforce login URL (such as https://dipak-sdfc-dev-ed.my.salesforce.com?so=00DF0000000ghLC).
Salesforce validates the message integrity using the embedded signature in the SAML assertion XML against the DataPower certificate, which is already uploaded during the SSO setup process. Upon successful signature validation, it processes the SAML assertion statement, extracts and validates the federation ID, and finally redirects to the page that was originally requested. Figure 35 shows the Salesforce landing page after a successful SSO login attempt by this example.

**Figure 33. SSO login attempt by the user through the web browser**

![Certificate Error]

**Figure 34. SSO re-direction (HTTP POST binding) page after successful authentication**

![Salesforce Login]
Step 11: Debug and troubleshoot guide

Assuming you are familiar with the DataPower common debugging and troubleshooting process, this tutorial primarily focuses on Salesforce.

You can capture base-64 encoded SAML assertion XML by using the Developer Tools in Google® Chrome (see Figure 36 and Figure 37). Though you can get this data from the HTML page source, using Developer Tools 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) to get an SAML assertion XML. Listing 4 shows a sample SAML assertion XML.
Figure 36. Start Developer Tools at Google Chrome

Figure 37. Capture base-64 encoded SAML assertion XML from HTML

Listing 4. Sample SAML Assertion XML (signed)

```xml
<samlp2:Response Version="2.0" ID="SAML-88fe1652-686b-4ef6-b5d8-80643ddbd791"
    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 URI="#SAML-88fe1652-686b-4ef6-b5d8-80643ddbd791"/>
        </SignedInfo>
        <Transforms>
            <Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#normalizeSpace"/>
        </Transforms>
    </Signature>
</samlp2:Response>
```
Login to Salesforce as an administrator and validate the base-64 encoded SAML assertion to check whether the SSO Identity Provider (DataPower) generates the correct SAML assertion as
expected by Salesforce. Figures 39 to 41 show the SAML assertion validation using Force.com's SAML Assertion Validator Framework.

**Figure 38. Start SAML Assertion Validator Framework at Salesforce**

![Start SAML Assertion Validator Framework at Salesforce](image)

**Figure 39. Enter base-64 encoded SAML assertion into the Salesforce SAML Validator**

![Enter base-64 encoded SAML assertion into the Salesforce SAML Validator](image)
Implementing Salesforce federated single sign-on with WebSphere DataPower, Part 1: Implementing an identity provider initiated SSO using a signed SAML assertion
While testing the SSO login, you can monitor the login history at Force.com from a separate login as administrator. Figure 42 shows a login history page at the Force.com site, which can help you troubleshoot the SSO login failures.

Figure 42. Login history page at Force.com

Conclusion

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

To continue with this series, see the following tutorials:

- Part 2: Identity provider initiated SSO using an encrypted and 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
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/)