SAML Installation and Integration Best Practices

Updated: 1/4/2016

Authors: Jonathan Jarvis, Pedro Teixeira, and João L. M. Pereira

Contents

Introduction

SAML is a specification that provides a means to exchange authentication and authorization of the “principal” (user) between an Identity Provider (IdP) and a Service Provider (SP). Once the plugin is installed, the Pentaho BA Server will become a SAML Service Provider, relying on the assertion to provide authentication, or both authentication and authorization for role assignment depending on the Identity Provider being used.

Getting Started

To begin, you will need:

- To select an IdP:

  Each IdP is different. They may vary by: signing key strengths, encryption requirements, secure socket layer endpoint protection, or sign out method capability.

- An installed Pentaho 6 BA Server

  You will need to install and configure the SAML plugin. Instructions for this can be found later in the document. Depending on the selected IdP, you might have to secure the BA Server for HTTPS communication.

- To decide on an authorization method for user role assignment:
  - Role provided in SAML assertion attributes – Varies based on IdP and attributes that can be provided
  - Hybrid method – Use an external service as a JDBC compliant database to read granted authorities for an authenticated NameID (SAML provided username used in BA Server)

IdP	Requires BA Server HTTPS (SSL)?	Authorization Attributes Supported?
Okta	No	Yes
SSOCircle	No	No
AD FS 2.0	Yes	Yes (with AD Schema)
AD FS 3.0	Yes	Yes (with AD Schema)
Keycloak	No	Yes

Configuring the BA Server for SAML Authentication

Installing the SAML Plugin and Required Files

*Note: Steps 2-4 should not be needed in Pentaho 6.1, where custom.properties already has correct dependendencies

1. Obtain the SAML Plugin Karaf Assembly (pentaho-saml-sample.kar file), logout.jsp, and applicationContext-spring-security-saml.xml
    a. Check “Additional resources” section for the location from where you can download logout.jsp and applicationContext-spring-security-saml.xml
    b. Check “Additional resources” section for the github location from where you can get the pentaho-saml-sample source code (to build the .kar file)
2. Shutdown the BA Server
3. Open $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/karaf/etc/custom.properties
    a. Add a specific version to the line containing “org.springframework.security.context, \” by changing it to:

       ```
       org.springframework.security.context; version\="2.0.8.RELEASE", \
       ```

    b. Directly below that line, add a new line containing another required dependency:

       ```
       org.springframework.security.ui; version\="2.0.8.RELEASE", \
       ```

4. Start the BA Server
5. Wait for the server to report it has started in $PENTAHO_HOME/server/biserver-ee/tomcat/logs/catalina.log
6. Place the pentaho-saml-sample.kar file in the $PENTAHO_HOME/server/biserver-ee/system/karaf/deploy/ folder. Once the plugin is installed, the $PENTAHO_HOME/server/biserver-ee/system/karaf/etc/pentaho.saml.cfg configuration file will be created
7. Shutdown the BA Server
8. Copy logout.jsp into the $PENTAHO_HOME/server/biserver-ee/tomcat/webapps/pentaho/ folder
9. Copy applicationContext-spring-security-saml.xml into the $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/ folder

Create a SAML Assertion Signing (and Encryption) Certificate and Keystore

1. Create a $PENTAHO_HOME/server/biserver-ee/saml/ folder
2. Open a terminal or command prompt, and make the newly created folder your working directory:

   ```
   cd $PENTAHO_HOME/server/biserver-ee/saml
   ```

3. Run the keytool command to generate a self-signed certificate. You may also obtain a signed certificate from a certificate authority if you wish. Ensure that the certificate uses a hash algorithm supported by your IdP (most likely SHA1 or SHA256).

   ```
   $PENTAHO_JAVA_HOME/bin/keytool -genkey -alias saml -keystore $PENTAHO_HOME/server/biserver-ee/saml/saml.keystore.jks -storepass changeit -keyalg RSA –keypass changeit
   ```

Notes:

- When prompted, fill out any information relevant to your organization
- If you're IdP only supports SHA1 signing, add "-sigalg SHA1WithRSA" as an argument
- Use a password other than "changeit" for –storepass and –keypass arguments
- The keystore password (-storepass) and key password (-keypass) do not need to be the same

Prepare the Pentaho Service Provider Metadata XML File

Having the Service Provider metadata file can simplify the process of obtaining the required Identity Provider metadata from the different IdP services. This section describes how to modify a template SP metadata file to match your BA Server installation.

1. If you do not already have an SP metadata file, copy the text below into a unix formatted file called pentaho-sp.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" ID="pentaho" entityID="pentaho">
    <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="false"
                        protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
        <md:KeyDescriptor use="signing">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                    <ds:X509Certificate><!-- REPLACE COMMENT WITH BASE64 SIGNING CERTIFICATE --></ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>
        <md:KeyDescriptor use="encryption">
            <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                    <ds:X509Certificate><!-- REPLACE COMMENT WITH BASE64 ENCRYPTION CERTIFICATE --></ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </md:KeyDescriptor>
        <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
        <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
        <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat>
        <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
        <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName</md:NameIDFormat>

        <!-- MODIFY LOCATION ATTRIBUTES WITH PROTOCOL, DOMAIN, and PORT -->
        <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                                Location="https://localhost:8443/pentaho/saml/SingleLogout"/>
        <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
                                Location="https://localhost:8443/pentaho/saml/SingleLogout"/>
        <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
                                Location="https://localhost:8443/pentaho/saml/SSO" index="0" isDefault="true"/>
        <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
                                Location="https://localhost:8443/pentaho/saml/SSO" index="1"/>
    </md:SPSSODescriptor>
</md:EntityDescriptor>
```

2. Move or copy your SP metadata file (pentaho-sp.xml if using a file from step 1) to $PENTAHO_HOME/server/biserver-ee/saml/pentaho-sp.xml

3. Locate the XML tag entries for “<md:SingleLogoutService>” and “<md:AssertionConsumerService>,” and replace the values of the “Location” attribute (bolded in blue above) with the appropriate protocol (http/https), domain name, and port that the BA Server is running on.
4. Export the contents of your saml signing (and additional encryption certificate if you generated one) certificate to a base64 representation using the following keytool command:

   ```
   $PENTAHO_JAVA_HOME/bin/keytool -exportcert -keystore $PENTAHO_HOME/server/biserver-ee/saml/saml.keystore.jks -storepass changeit -alias saml -rfc
   ```

5. The command in the previous step should print out the certificate data, which looks like:

   ```
   -----BEGIN CERTIFICATE----
   CERTIFICATE
   DATA
   PAYLOAD
   -----END CERTIFICATE-----
   ```

   Copy the content of the certificate data payload (not the begin or end line) into the appropriate “<ds:X509Certificate>” entry tag. The “use” attribute of the parent “<md:KeyDescriptior>” tag defines if you’re dealing with the signing or encryption certificate. The same certificate data can be used in both spots.

Setup a Pentaho Application/Party Trust/Client in the IdP and Obtain the IdP Metadata XML

Since each IdP is different, the process of defining Pentaho as an SP within a selected IdP is different. When configuring the SP in the IdP, the entity id is “pentaho” in all lower case. The IdP may have a tool to help import the Pentaho SP using the pentaho-sp.xml file configured in the previous section. Similarly, the process of obtaining IdP metadata is different for every Identity Provider. Once obtained and installed, the file instructs the BA Server where to redirect authentication requests. Examples of tested Identity Providers are provided in the “Identity Provider Configuration, Metadata, and IdP Specific Client Configuration” section later in this document.

Save the IdP metadata file as $PENTAHO_HOME/server/biserver-ee/saml/saml-idp.xml

Configure the pentaho.saml.cfg File

At this point, there is a saml folder with an IdP metadata xml file, the Pentaho SP metadata xml file, and a saml keystore with the signing certificate (and possibly separate encryption certificate). When editing the pentaho.saml.cfg file, please note that absolute paths (no variables) must be listed in the file. For the purposes of demonstration, assume that $PENTAHO_HOME = /pentaho6. Also, if editing these files for use in a Windows environment, use the forward slash “/” in lieu of the backslash “\” for paths, e.g. C:/pentaho6

1. Shut down the BA Server if it is running
2. Open the /pentaho6/server/biserver-ee/pentaho-solutions/system/karaf/etc/pentaho.saml.cfg file for editing.

Setting IdP Properties

There are three ways to identify the IdP metadata XML file. It can be specified to be read from a URL, a filesystem path, or from a jar on the classpath (saml.idp.metadata.url, saml.idp.metadata.filesystem, and saml.idp.metadata.classpath respectively). Only one method should be enabled at any time, and the non-used properties should be commented out with a number “#” sign at the front of the line.

1. Comment the entries for saml.idp.metadata.url and saml.idp.metadata.classpath if they are uncommented
2. Uncomment the entry for saml.idp.metadata.filesystem if it is commented
3. Change the value (after the equal sign) to the path of your IdP metadata XML file:

   saml.idp.metadata.filesystem=/pentaho6/server/biserver-ee/saml/saml-idp.xml

You will also need to set the saml.idp.url property, which is used to select the proper EntityDescriptor from the referenced saml-idp.xml file.

1. Open $PENTAHO_HOME/server/biserver-ee/saml/saml-idp.xml
2. Locate the “<EntityDescriptor>” tag, and copy the value of the “entityId” attribute:

   <EntityDescriptor ID=”…” entityID=”**http://the-idp-url** “ xmlns=

3. In pentaho.saml.cfg, set the value of the saml.idp.url property to the copied value

   saml.idp.url=http://the-idp-url

Finally, there are some properties to adjust based on the IdP that is selected. The settings used with tested Identity Providers are available in the “Identity Provider Configuration, Metadata, and IdP Specific Client Configuration” section of this document. These settings are:

- use.global.logout.strategy – **true**/false controls if a Single Logout endpoint can be used
- ensure.incoming.logout.request.signed – **false**/true – ensures incoming logout requests are signed by IdP
- ensure.outgoing.logout.request.signed=**true**/false – ensures outgoing logout request are signed by SP
- ensure.outgoing.logout.response.signed=**true**/false – ensures outgoing logout responses are signed by IdP

Setting SP Properties

Similar to the IdP settings, there are three ways to specify the Service Provider document. Choose from URL, filesystem, or classpath using the saml.sp.metadata.url, saml.sp.metadata.filesystem, or saml.sp.metadata.classpath properties. This example will use the filesystem, since earlier instructions directed you to save the pentaho-sp.xml file.

1. Comment out the entries (prefix the line with “#”) for saml.sp.metadata.url and saml.sp.metadata.classpath if they’re uncommented
2. Uncomment the line for saml.sp.metadata.filesystem if it is commented
3. Change the value of the saml.sp.metadata.filesystem to:

   saml.sp.metdata.filesystem=/pentaho6/server/biserver-ee/saml/pentaho-sp.xml

The other property that is configurable for the Service Provider is saml.sp.metadata.entityId. It has a case sensitive value that defaults to “pentaho”. Editing this value is not recommended. The value has to match the entityId in the SP metadata XML file and the party trust configured in the IdP.

Setting the Keystore Properties

All of the certificates needed for signing, encryption, and communication with SAML servers need to be in a single keystore file. Similar to IdP and SP configuration, the keystore can be specified with a URL, filesystem path, or classpath entry using the saml.keystore.url, saml.keystore.filesystem, or saml.keystore.classpath properties.

1. Comment the saml.keystore.url and saml.keystore.classpath properties with "#" if they are uncommented
2. Uncomment saml.keystore.filesystem if it is commented with "#"
3. Change the value of saml.keystore.filesystem to:
   saml.keystore.filesystem=/pentaho6/server/biserver-ee/saml/saml.keystore.jks

4. Locate the saml.keystore.default.key property, and change it to match the alias of your saml signing certificate. The path setup before referenced a self-signed certificate aliased as "saml," which would be configured as:

   saml.keystore.default.key=saml

5. Locate and set the keystore password with the saml.keystore.password property. This should match the password used as the –storepass argument when you obtained a SAML signing certificate in the "Create a SAML Assertion Signing (and Encryption) Certificate and Keystore" section of this document.

   saml.keystore.password=**changeit**

6. If any of your certificate private key passwords include the colon ":" character, change the saml.username.password.delimiter.char property to a valid delimiter character not included in any of the key passwords.

7. Provide a comma separated list of username<delimeter>password to allow the Pentaho SAML plugin to read private keys in the saml.keystore.private.username.passwords property:

   saml.keystore.private.username.passwords=saml:changeit,saml2:changeit

Enable the SAML Plugin

1. Shut down the BA Server if it is running
2. Open $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/pentaho-spring-beans.xml for editing
3. Locate the line containing "<import resource="applicationContext-spring-security-jdbc.xml" />" and add the following line below it:

```
<import resource="applicationContext-spring-security-saml.xml" />
```

4. Save and close the file
5. Open $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/security.properties for editing
6. Change the provider on the first line to "saml". Once completed, the line will read:

   provider=saml

7. Save and close the file

8. Start the BA Server and attempt to authenticate. The system should redirect to the Single Sign On page (if not already logged in), and upon successful credential authentication, the user's Name ID provided by SAML will appear in the top right hand corner of the page.

Note: If you wish to disable the SAML plugin, simply shut down the BA Server, then comment out the line added to pentaho-spring-beans.xml and change the provider in security.properties to its previous value.

Configuring the Authorization Method (User Role Assignment)

SAML Attribute for Role Assignment

By default, the SAML plugin expects user role information to be passed as an attribute in the SAML assertion. Not all Identity Providers are able to store and pass custom attributes. If this is the case with the IdP you have selected, jump directly to the "Hybrid Role Assignment via JDBC" section below.

There are properties in $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/karaf/etc/pentaho.saml.cfg file that control which attribute to obtain role information from, as well as how to parse the granted authority from it. If you make any changes to the file, start by shutting down the BA Server.

The first property is **authorization.provider**. This property is configured with "saml" as its default value, which tells the plugin to obtain the user role from a SAML assertion attribute. If this setting is being used and a user does not have any particular attribute, a value of "Authenticated" is assigned to the user.

```
authorization.provider=saml
```

The **saml.role.related.user.attribute.name** property is used to specify the attribute that contains the user role information provided in the SAML assertion. If the IdP does not allow a custom property name, or you already have (and wish to use) an attribute with user roles for other tools, specify the custom attribute name in this properties' value.

```
saml.role.related.user.attribute.name=Pentaho Role
```

The **saml.role.related.user.attribute.prefix** property stores a prefix for any role granted to the authenticated user. Since many tools may consume roles in a single custom attribute from the IdP, a prefix should be added to separate the different Service Providers. For example, the user might only be granted an Authenticated role for service provider X, while they should have the privileges as an Administrator in Pentaho. The attribute passed back (named according to the saml.role.related.user.attribute.name property) may contain a value like "X:Authenticated Pentaho:Administrator". The default value for this field is "Pentaho:", and roles should be created in the IdP using this value as the prefix if applicable.

```
saml.role.related.user.attribute.prefix=Pentaho:
```

Hybrid Role Assignment via JDBC

Some Identity Providers do not provide a mechanism to pass custom attributes for role information. Also, some organizations may want to store authentication and authorization data in different systems. If either two of these scenarios match your requirements, Hybrid Role Assignment should be used. The SAML plugin can rely on non-SAML spring security beans that have been configured in Pentaho to provide user roles/authorization. This section is dedicated to using the JDBC method for BA Server Authorization after successful SAML Authentication.

1. Shut down the BA Server
2. Open $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/karaf/etc/pentaho.saml.cfg for editing, and change the **authorization.provider** to jdbc:

   authorization.provider=jdbc
3. Follow the standard procedure for setting up JDBC authorization:
   a. Choose and acquire an RDBMS to store Pentaho authorization information with. This may be stored in an additional set of tables on the same system hosting the Pentaho repository
   b. Update $PENTAHO_HOME/server/biserver-ee/pentaho-solutions/system/applicaitonContext-spring-security-jdbc.properties with the configuration of your RDBMS
   c. Add Authorities, Users, and Granted_Authorities tables
   d. Populate the Authorities, Users, and Granted_Authorities tables

      IMPORTANT: The username in the Users table and Granted_Authorities table must exactly match (case sensitive) the “NameID” attribute passed in the SAML assertion. The value passed is configured during the trusted party/client setup in the IdP. Some example values could be an e-mail address, an Active Directory User-Principal-Name, or a sAMAccountName.

4. Start the BA Server and login to test the authorization

Identity Provider Configuration, Metadata, and IdP Specific Client Configuration

Okta

Some IdPs have a feature to upload the SP Metadata XML file to configure your service provider. However, Okta does not currently have this feature, so you’ll have to export the signing certificate that will be imported during configuration.

Export the Signing Certificate

Execute the following command to place the signing public key in a cer file:

> *$PENTAHO_JAVA_HOME/bin/keytool -exportcert -keystore $PENTAHO_HOME/server/biserver-ee/saml/saml.keystore.jks -storepass **changeit** -alias saml –rfc > $PENTAHO_HOME/server/biserver-ee/saml/saml.signing.cert.cer*

Identity Provider Configuration

Add the Application

1. Sign into an Okta account with Administrative privileges
2. Enter the “Admin” section of the site
3. Click the “Add Applications” shortcut
4. Click the “Create New App” button
5. Fill in the “App name” field with “pentaho”
6. Optionally add a logo, and leave the visibility settings unchanged
7. Click “Next”
8. Fill in the basic settings form as follows:

Field	**Value**
Single sign on URL	**http://localhost:8080**/pentaho/saml/SSO
Recipient URL and Destination URL checkbox	Checkbox selected
Audience URI	pentaho
Default Relay State	<empty>
Name ID format	EmailAddress
Application Username	Okta username

9. Update the **blue portion** in the URL above to match the domain and port of the BA Server
10. Click on “Show advanced settings,” and fill in these values:

Field	**Value**
Response	signed
Assertion Signature	signed
Signature algorithm	RSA- SHA256
Digest algorithm	SHA256
Assertion encryption	unencrypted
Enable Single Logout	Checkbox: checked
Single Logout URL	**http://localhost:8080**/pentaho/saml/SingleLogout
SP Issuer	pentaho
Signature Certificate	Upload the certificate exported above, which should be located at: $PENTAHO_HOME/server/biserver-ee/saml/saml.signing.cert.cer
Authentication Context	PasswordProtectedTransport
Honor Force Auth.	Yes
Saml Issuer ID	http://www.okta.com/${org.externalKey}

11. Update the **blue portion** of the Single Logout URL above to match the domain and port of the BA Server
12. If you wish to authorize users with roles in the Okta SAML assertion, add this “Group Attribute Statement”

Name	**Name format**	**Dropdown**	**Filter**
Pentaho Role	Unspecified	StartsWith	Pentaho:

Note: “Pentaho Role” is merely a name set for the attribute that will carry the Group list; you are free to use a prefix that best suits your needs. Be sure to update pentaho.saml.cfg accordingly

Property	Value	Note
saml.role.related.user.attribute.name	Pentaho Role	Name of the attribute that carries this user’s Group list

13. Click “Next”
14. Check the radio button “I'm an Okta customer adding an internal app”
15. Enable the checkbox “This is an internal app that we have created”
16. Click “Finish”

    You should be redirected to the “Sign On” page of the newly created “pentaho” application. If not, you can get there by following these steps:

    i. In the top menu bar, select Applications > Applications
    ii. Select “pentaho”
    iii. Click “Sign On”

17. Click the “Identity Provider metadata” hyperlink, and download the resulting file as $PENTAHO_HOME/server/biserver-ee/saml/okta-idp.xml

Create Groups for Pentaho Roles (If Using for Authorization, Otherwise Skip to Add Users)

1. In the top menu bar, select Directory > Groups
2. Add the following three groups:

Name	Description
Pentaho:Administrator	Pentaho BA-server’s Administrator Role
Pentaho:Power User	Pentaho BA-server’s Power User Role
Pentaho:Report Author	Pentaho BA-server’s Report Author Role

Note: “Pentaho:” is merely a prefix used at each group; you are free to use a prefix that best suits your needs. Be sure to update pentaho.saml.cfg accordingly

Property	Value	Note
saml.role.related.user.attribute.prefix	Pentaho:	(Optional) some attribute values may come with a prefix that allows them to be identified between separate contexts (examples: 'Pentaho:Report Author', 'Zendesk:CTools Support', 'Office365:Contributor C1', 'Pentaho:Authenticated', ...)

Assign Users to the “pentaho” Application

1. In the top menu bar, select Applications > Applications
2. Click the “Assign Applications” button
3. Select “pentaho” from the “Applications” section on the left

4. Select any users that should be allowed to authenticate to Pentaho using SAML from the "People" section on the right
5. Click "Next"
6. Click "Confirm Assignments"

Add Users to Groups for Role Assignment (if Using for Authorization, Otherwise Skip)

1. In the top menu bar, select Directory > People
2. Select the user you wish to authorize with a role
3. Click on the "Groups" tab under the user name
4. Start typing "Pentaho" in the "Groups" text box
5. Click "Add" on any roles/groups you wish to add the user to

IdP Specific Configuration in pentaho.saml.cfg

Property	**Value**	**Note**
saml.idp.metadata.filesystem	$PENTAHO_HOME/server/biserver-ee/saml/okta-idp.xml	This setting cannot take variables like $PENTAHO_HOME. Ensure that you use the absolute path to the IdP Metadata XML
use.global.logout.strategy	false	As of the writing of this document, global logouts did not work through Okta
ensure.incoming.logout.request.signed	false	
ensure.outgoing.logout.response.signed	true	
ensure.outgoing.logout.request.signed	true	
saml.role.related.user.attribute.name	Pentaho Role	Only needed for authorization/role assignment
saml.role.related.user.attribute.prefix	Pentaho:	Only needed for authorization/role assignment

AD FS (Active Directory Federation Services) 2.0, 3.0

AD FS 2.0 runs as a web application under IIS 7 for Windows Server 2008, and 3.0 runs as its own service in Windows 2012. The setup for both systems is configured through a wizard. AD FS needs to be installed and configured to use SSL for Server Communications and Pentaho (the Service Provider) also needs to be running securely over HTTPS. The process below assumes that Active Directory Domain Services (AD DS) and Active Directory Federation Services (AD FS) have already been installed and configured with certificates.

Identity Provider Configuration for Authentication

Adding and Configuring a Relying Party Trust

1. Enter the AD FS Management Snap-in Utility
2. Expand "Trust Relationships" in the tree on the left side of the window
3. Right click "Relying Party Trusts" and click "Add Relying Party Trust…"
4. Click "Start"

5. Choose the “Import data about a relying party from a file” radio option
6. Click “Browse” and select $PENTAHO_HOME/server/biserver-ee/saml/pentaho-sp.xml (copy to this machine if necessary)
7. Click “Next”
8. Type “pentaho” as the Display name
9. Optionally type “pentaho” in the Notes
10. Click “Next”
    a. In AD FS 3.0, you will be prompted with “Configure Multi-factor Authentication Now?”
    b. Select the “I do not want to configure multi-factor authentication settings for this relying party trust at this time” radio option
    c. Click “Next”
11. Select “Permit all users to access this relying party”
12. Click “Next”
13. The next page reviews all the settings that were extracted from the pentaho-sp.xml service provider metadata XML file. No changes are required
14. Click “Next”
15. Click “Finish” to exit the wizard
16. You will now see a “pentaho” Relying Party Trust listed in the middle pane on the page

If you are using a SHA1 algorithm for your assertion signing key (setting defaults to SHA-256):

1. Right click on the “pentaho” Relying Party Trust
2. Select “Properties” option from the context menu
3. Click on the “Advanced” tab
4. Select the “SHA-1” Secure hash algorithm option

Configuring Assertion Claim Attributes

1. Right click on the “pentaho” Relying Party Trust
2. Select the “Edit Claim Rules…” context menu option
3. Select the “Issuance Transform Rules” tab
4. Click the “Add Rule…” button, which will start a wizard
5. Select “Send LDAP Attributes as Claims” from the dropdown box
6. Click “Next”
7. Name the rule “pentaho claim rules”
8. Select “Active Directory” from the Attribute store dropdown box
9. Add a claim mapping from the LDAP Attribute “E-mail-Addresses” to an Outgoing Claim Type of “E-mail Address”
10. Add a claim mapping from the LDAP Attribute “User-Principal-Name” to an Outgoing Claim Type of “Name ID”

    Note: The value of the mapped “Name ID” attribute is what users must type into the username field on the Pentaho BA Server login page. The User-Principal-Name is recommended because it includes both the user account name and the account domain (ex. user@domain.com). However, in a single domain scenario, administrators might want to use the SAM-Account-Name LDAP Attribute, which only provides the user account name (ex. user).

11. If you are going to use AD FS to provide the Pentaho BA Server with Role information, add an additional mapping from the LDAP Attribute “Token-Groups Qualified by Long Domain” to an Outgoing Claim Type of “Pentaho Role”, which will have to be typed into the combo box
12. Click “Finish”
13. Click “Ok” to exit the Edit Claim Rules dialog

Exporting the IdP Medata XML File

1. Replace **your.adfs.domain** with your AD FS host domain name in the following URL:

   https://**your.adfs.domain**/federationmetadata/2007-06/federationmetadata.xml

2. Visit the URL, and download the resulting XML as $PENTAHO_HOME/server/biserver-ee/saml/adfs-idp.xml

Export the AD FS Signing Certificate

1. Enter the AD FS Management Snap-in Utility
2. Expand the “Service” option in the tree on the left pane
3. Select the “Certificates” option underneath “Service”
4. Double click on the “Token-signing” certificate to open a Certificate dialog
5. Select the “Details” tab
6. Click the “Copy to File…” button
7. Select the option to export the certificate as a “Base-64 encoded X.509 (.CER)” file
8. Click “Next”
9. Save the file as adfs.signing.cer and transfer it to $PENTAHO_HOME/server/biserver-ee/saml/adfs.signing.cer

Import the AD FS Signing Certificate into the SAML Keystore

1. Enter a command prompt
2. Replace keystore passwords/paths, and run the command below:

   *$PENTAHO_JAVA_HOME/bin/keytool -import –alias adfs -keystore $PENTAHO_HOME/server/biserver-ee/saml/saml.keystore.jks -storepass **changeit** -file $PENTAHO_HOME/server/biserver-ee/saml/adfs.signing.cer*

Install Java JCE Unlimited Strength Security in Your JRE (Required once per JRE)

This step is required so the JVM can use larger key sizes for the certificate exported from AD FS. Without installing JCE Unlimited Strength, you may notice InvalidKeyException errors in the BA Server Tomcat logs.

1. Download the appropriate JCE Unlimited Strength archive for the version of the Java Runtime hosting Tomcat for the BA Server from Oracle
2. Follow the instructions to install, which will be packaged with the download in a README file

IdP Specific Configuration in pentaho.saml.cfg

Property	**Value**	**Note**
saml.idp.metadata.filesystem	$PENTAHO_HOME/server/biserver-ee/saml/adfs-idp.xml	This setting cannot take variables like $PENTAHO_HOME. Ensure that you use the absolute path to the IdP

		Metadata XML
use.global.logout.strategy	true	
ensure.incoming.logout.request.signed	false	
ensure.outgoing.logout.response.signed	true	
ensure.outgoing.logout.request.signed	true	
saml.role.related.user.attribute.name	Pentaho Role	Only needed for authorization/role assignment
saml.role.related.user.attribute.prefix	Pentaho_	(Optional) some attribute values may come with a prefix that allows them to be identified between separate contexts (examples: 'Pentaho:Report Author', 'Zendesk:CTools Support', 'Office365:Contributor C1', 'Pentaho:Authenticated', ...) **Please do not use ":" in the prefix as AD FS does not accept it**

Attribute Setup for Principal Role Assignment (Authorization)

Once a user is authenticated via SAML assertion, they will be assigned the default role of "Authenticated" in the BA Server. If you wish to manage Pentaho roles within Active Directory, follow the steps below to create groups with additional roles, and assign them to users. The role information will be passed by AD FS in the SAML assertion with the "Token-Groups Qualified by Long Domain" attribute configured above. The example below is setup to create a Pentaho Administrator group.

Creating Groups for Pentaho Roles

1. Open the "Active Directory Users and Computers" management console snap-in
2. On the tree in the left pane, expand your domain you wish to add the additional groups in
3. Right click on "Users"
4. In the context menu, select New -> Group
5. In the "New Object – Group" window that opens, type "Pentaho_Administrator" as the Group name

Note: The group name must match the "Pentaho_" prefix, "Pentaho_" is merely a prefix used at each group; you are free to use a prefix that best suits your needs. Be sure to update pentaho.saml.cfg accordingly

6. Ensure that Group scope is "Global"
7. Ensure that Group type is "Security"
8. Click "Ok"

Adding Users to Groups for Pentaho Roles

1. Open the "Active Directory Users and Computers" management console snap-in
2. On the tree in the left pane, expand your domain with the users you wish to add Pentaho roles to
3. Select the users you wish to assign specific roles to
4. Right click on the selected users, and choose "Add to a group…" from the context menu
5. Type the group names (created in the previous section) desired for the users

6. Press the “Check Names” button to ensure the group names match the groups that were created
7. Press “OK” to confirm the users

Addendum for AD FS 2.0

AD FS 2.0 runs as a web application inside of IIS. A basic authentication is used by default (although it does not work with some browsers), but a form based authentication option (similar to AD FS 3.0 default) exists.

Disable Windows Extended Protection – Basic Authentication

If you are using basic browser authentication, disabling Windows Extended Protection is required to allow all browsers to sign into Pentaho using SAML.

1. Access the Internet Information Services Manager snap-in
2. Select your domain in the tree on the left panel of the page
3. Double click on the “Authentication” icon that appears in the list on the right pane in the “IIS” section
4. Right click on “Windows Authentication”
5. Select “Advanced Settings…” in the context menu
6. Change the Extended Protection dropdown to the “Off” setting
7. Restart the services as described in the “Restart AD FS Services” section

Enable AD FS Form Authentication

Form based authentication is enabled by editing a configuration file within IIS.

1. In the Windows Explorer, access C:\inetpub\adfs\ls, or the adfs\ls directory where AD FS 2.0 is installed
2. Open the web.config file for editing
3. Locate the microsoft.identityServer.web entry of the XML that looks like this:

```
<microsoft.identityServer.web>
  <localAuthenticationTypes>
    <add name="Integrated" page="auth/integrated/" />
    <add name="Forms" page="FormsSignIn.aspx" />
    <add name="TlsClient" page="auth/sslclient/" />
    <add name="Basic" page="auth/basic/" />
  </localAuthenticationTypes>
```

4. Comment out the “<add name=”Integraged”…” line by surrounding it with <!-- and -->

```
<!-- <add name="Integrated" page="auth/integrated/" /> -->
```

5. Restart the services as described in the “Restart AD FS Services” section

Restart AD FS Services – Recommended After Changes

1. Enter the Services Management Console snap-in

2. Locate the "AD FS (2.X/3.X) Windows Service"
3. Right click on the service and select the "Restart" option
4. If using AD FS 2.0, restart the Web Application Pool
    - Access the Internet Information Services Manager snap-in
    - Expand the domain hosting AD FS in the tree from the left pane
    - Click on "Application Pools"
    - Right click on the "ADFSAppPool" option in the list
    - Select "Recycle…" from the context menu

Keycloak

Keycloak is an Identity Broker provided by JBoss. An Identity Broker is able to forward credentials from a login request to multiple trusted Identity Providers or Identity Servers. Keycloak can act as a fully-fledged standalone IdP, or it can broker requests to other SAML IdPs, OpenID Connect services, or various social networks. This provides a way to authenticate using multiple trusted services while the BA Server is configured to use a single service. In this context, Keycloak acts as the IdP for the Pentaho BA Server (SP), and Keycloak acts as a Service Provider (SP) to other Identity Providers (IdPs).

Identity Provider Configuration

1. Login to the Keycloak Administration Console
2. If desired, select or add a realm other than "Master" on the top left of the page
3. Click on the "Clients" option on the left side of the page
4. Click the "Create" button on the top right side of the client list on the right side of the page
5. Click the "Import" button on the top of the form
6. Select the $PENTAHO_HOME/server/biserver-ee/saml/pentaho-sp.xml file
7. Press the "Save" button
8. Replace your.keycloak.domain and master with your realm in the following URL, then download the contents as $PENTAHO_HOME/server/biserver-ee/keycloak-idp.xml

    https://**your.keycloak.domain**/auth/realms/**master**/protocol/saml/descriptor

IdP Specific Configuration in pentaho.saml.cfg

Property	Value	Note
saml.idp.metadata.filesystem	$PENTAHO_HOME/server/biserver-ee/saml/keycloak-idp.xml	This setting cannot take variables like $PENTAHO_HOME. Ensure that you use the absolute path to the IdP Metadata XML
use.global.logout.strategy	true	While this value is valid, Keycloak may not be able to forward logout requests to IdPs that do not support global logout
ensure.incoming.logout.request.signed	false	
ensure.outgoing.logout.response.signed	true	
ensure.outgoing.logout.request.signed	true	

Additional Resources

Pentaho SAML Reference Implementation:

Github:

https://github.com/pentaho/pentaho-engineering-samples

- Source Code: /Samples for Extending Pentaho/Reference Implementations/Security/SAML 2.0
- Resources: /Samples for Extending Pentaho/Reference Implementations/Security/SAML 2.0/documentation/resources