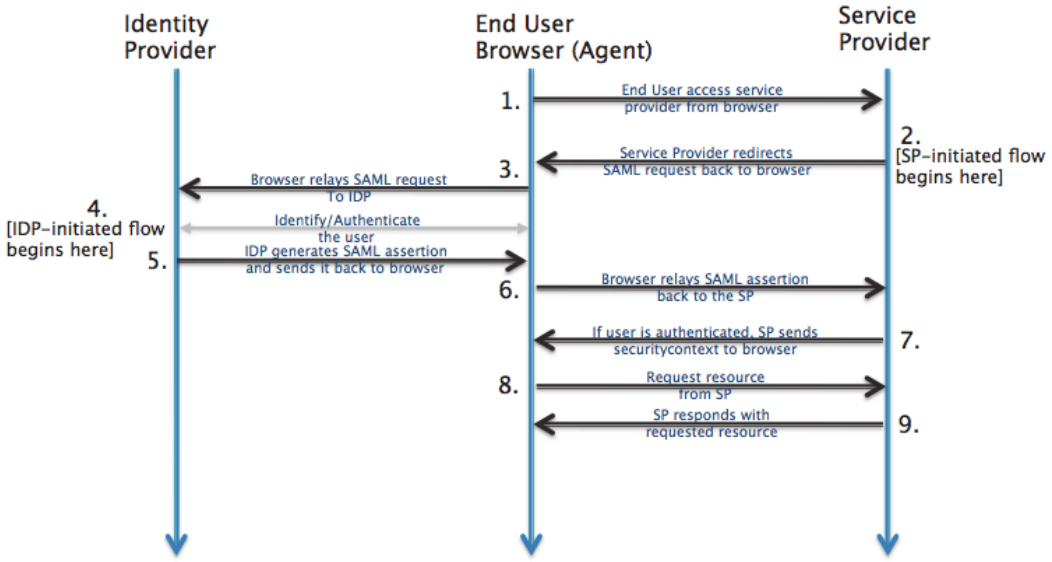
Registering in SSOCircle.com IdP

SAML: Overview

[1] SAML is mostly used as a web-based authentication mechanism as it relies on the browser being used as an agent that brokers the authentication flow. At high-level, the authentication flow of SAML looks like this:



Identity Provider	SSOCircle.com	The 3rd-party entity that takes care of the user's authentication; There are multiple such entities with SAML protocol support, such as OpenSSO, SSOCircle.com, OneLogin.com, Salesforce.com… For this example, we'll be using SSOCircle.com
End-user browser agent	Pentaho User	User that accesses BA-server via browser
Service Provider	BA-server	Pentaho BA Server

[1] http://developer.okta.com/docs/guides/saml_guidance.html

Registering in SSOCircle.com Identification Provider (IdP)

Note: Each user that intends to use BA-server needs to register itself first in SSOCirle.com and upload to its service providers list the “pentaho service provider” metadata

Step 1 of 2 | Get SSOCircle.com IdP and Pentaho SP metadata files

1. Have your chosen Service Provider (i.e. BA-Server) metadata xml file at hand

   Developer/QA only: if none is created yet, you can leverage on the already existing SP metadata file for Pentaho BA-Server. For this:
   a. Next to this document, you should have a “`resources`” folder
   b. Download “pentaho-sp.xml” and rename it to something more identifiable with SSOCircle (e.g. “pentaho-ssocircle-sp.xml”);

2. Go to http://idp.ssocircle.com/idp-meta.xml (it’s of public access)
   a. Download that IDP’s xml metadata and name it to something identifiable with SSOCircle (e.g. “ssocircle-idp.xml”)

Step 2 of 2 | Registering yourself in SSOCircle.com

1. Go to http://www.ssocircle.com
2. Top menu bar > “Sign In /Register” (last option on the right) > “Register”
3. Fill in all fields
   a. Be sure to provide a real email, as a confirmation email will be sent to you.
4. Once registration is done, login to ssocircle.com
5. Done.
6. (**optional**) Logout from ssocircle.com (left-side menu, 1st option)

Important: the following steps are meant to be done only once for each entity; reach out to the Support Team to check if you need to execute the following;

1. We need to submit a new service provider for “pentaho”, otherwise no incoming authentication requests from http://localhost:8080/pentaho will be allowed
2. In your user account page, select “Manage Metadata” (left-side menu, 9th option)
3. Select “Add new service provider”
   a. In the “Enter the FQDN of the service provider” field, type “localhost”
   b. In the “Attributes sent in assertion”, select all 3
   c. In the “Insert Metadata”:
      i. Open “pentaho-ssocircle-dev-sp.xml” and copy its contents **as-is**

ii. Paste it into the “Metadata” field

d. Click “Submit” button (below the Metadata area text field)

4. Wait a couple of seconds; If all went well, you should see a new registered service provider, called “pentaho”
5. (**optional**) Logout from ssocircle.com (left-side menu, 1st option)
6. Done.

Important Note

The service provider metadata we’ve just submitted is a **for-testing-purposes-only** metadata, adapted from spring-security-saml-sample to the pentaho webapp;

The only authentication requests that will be accepted in ssocircle.com will be the ones coming from a “http://localhost:8080/pentaho”.

If you are accessing BA-server via a remote IP (instead of localhost) or using a different port (rather than 8080), or even if you have changed the webapp name from “pentaho” to some other name, then most likely ssocircle will not allow those unless you update a service provider metadata to SSOCircle.com that matches your scenario.

Q&A

Q1 | Do I need a certificate to sign the authentication requests?

Yes.

For this sample, we are using a certificate provided by spring-security-saml, stored in a .jks (keystore file).

It’s already bundled in the saml-authentication-provider sample (jar:/security/keystore.jks).

You can get the original here: https://github.com/spring-projects/spring-security-saml/blob/1.0.1.RELEASE/core/src/test/resources/org/springframework/security/saml/key/keystore.jks

This certificate is used by some IdP’s, such as SSOCircle.com and okta.developer.com.

If you plan to connect to some other IdPs, then you must ensure you update the keystore file to include the certificate provided by that Identification Provider.