Enabling hybrid SAML/JDBC Configuration

Pre-requisites

All tasks/steps at **Activating_SAML_sample_in_BA_server** document have been successfully completed.

Overview

This document explains how to change from a (default configuration of) SAML-only authentication/authorization mechanism to a hybrid SAML/JDBC one:

- **SAML for Authentication**: a 3rd-party identification service takes care of the user's authentication (i.e. ensuring a user is who it states it is by matching a username to a password);
    - Communication between BA-server and that 3rd party authentication service is made according to SAML protocol

- **JDBC for Authorization** (i.e. assigning a username with a set of roles/authorities) is JDBC-based
    - It is the user's responsibility to create the DB tables and populating them with the users and roles;
    - you are free to change those, but to do that you will also need to change the default DB connection values defined in `pentaho-solutions/system/applicationContext-spring-security-jdbc.properties`.

Preparing BA-server for hybrid SAML/JDBC

Note: The following are preparation tasks, i.e. **tasks that only need doing once**.

Step 1 of 3 | Create the user/authorities database tables

We will need to create 3 database tables, namely:

Table Name: USERS		
Column Name	**Column Type**	**Column Description**
username	VARCHAR(50)	The username
password	VARCHAR(50)	**Note**: this column value is not considered in a hybrid SAML/JDBC solution, as all authentication takes place in the 3rd party authentication service; you can fill this column with <empty string>, "ignored", …
enabled	VARCHAR(5)	'true' if user is enabled, 'false' otherwise

Table Name: AUTHORITIES		
Column Name	**Column Type**	**Column Description**
authority	VARCHAR(50)	The Pentaho Role ("Administrator", "Report Author", …)

Table Name: GRANTED_AUTHORITIES		
Column Name	**Column Type**	**Column Description**
username	VARCHAR(50)	The username
authority	VARCHAR(50)	The Pentaho Role

Note: Using Postgres as the database of choice ?

You can get a simple (for sample purposes) DB table creation/population script.

Next to this document, you should have a "`resources`" folder. Navigate to /hybrid_solution/database_scripts/postgres and use `create_populate_tables.sql`

Step 2 of 3 | Setting up the correct JDBC connection properties

1. Edit `pentaho-solutions/system/applicationContext-spring-security-jdbc.properties` and update the properties according to your chosen JDBC database

Property Key	Property Description
datasource.driver.classname	The fully qualified Java class name of the JDBC driver to be used
datasource.url	The connection URL to be passed to our JDBC driver to establish a connection
datasource.username	The connection username to be passed to our JDBC driver to establish a connection
datasource.password	The connection password to be passed to our JDBC driver to establish a connection
datasource.validation.query	The SQL query that will be used to validate connections from this pool before returning them to the caller. This query must be an SELECT statement that returns at least one row.
datasource.pool.max.wait	The maximum number of milliseconds that the pool will wait (when there are no available connections). For a connection to be returned before throwing an exception, or <= 0 to wait indefinitely. Default is -1
datasource.pool.max.active	The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit. Default value is 8.
datasource.max.idle	The maximum number of connections that can remain idle in the pool, without extra ones being destroyed, or negative for no limit. Default value is 8.
datasource.min.idle	The minimum number of active connections that can remain idle in the pool, without extra ones being created when the evictor runs, or 0 to create none. Default value is 0.

2. Save and close the file.

Step 3 of 3 | Enabling the Hybrid SAML/JDBC authorization

1. Edit `pentaho-solutions/system/karaf/etc/pentaho.saml.cfg`
2. Locate property '`authorization.provider`'
3. Modify its value to '`jdbc`'
4. Save and close the file.