Pentaho, Linux, and Microsoft Active Directory Authentication with Kerberos
Change log (if you want to use it):

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Author</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Contents

Overview.............................................................................................................................................................. 1
Before You Begin............................................................................................................................................ 1
Setting Up the Domain Controller and SQL Server ....................................................................................... 2
  Configuring Domain Controller..................................................................................................................... 2
  Configuring Linux and Pentaho .................................................................................................................... 4
  Configuring Pentaho Data Integration (PDI) Connection............................................................................. 5
Pentaho Server Configuration .......................................................................................................................... 7
  Verifying Database Configuration................................................................................................................ 7
  Configuring Pentaho Server's Manage Connections................................................................................... 8
  Configuring context.xml............................................................................................................................... 8
Related Information........................................................................................................................................... 9
Finalization Checklist...................................................................................................................................... Error! Bookmark not defined.
Overview

Many enterprises require that all their connections to different databases be authenticated through a Kerberos ticket. This applies to all JDBC-based connections such as hibernate, quartz, and jackrabbit. This also applies to solution connections, which are configured in the context.xml, through Manage Data Sources in the Pentaho User Console (PUC), and in a job or a transformation.

Some of the topics discussed here include Windows authentication, Kerberos, Microsoft SQL Server, Linux, kinit, klist, and the Pentaho Servers.

The intention of this document is to speak about topics generally; however, these are the specific versions covered here:

<table>
<thead>
<tr>
<th>Software</th>
<th>Version(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pentaho</td>
<td>6.x, 7.x</td>
</tr>
</tbody>
</table>

The Components Reference in Pentaho Documentation has a complete list of supported software and hardware.

Before You Begin

Before beginning, use the following information to prepare for the procedures described in the main section of the document.

You will need:

1. A Pentaho Server that is running in a Linux environment.
2. An active directory (AD) domain controller that is reachable by both the database and the Pentaho server.
3. A database configured correctly to accept connections with Kerberos tickets from the domain controller.
4. A JDBC driver that is used to connect to the database supports Kerberos authentication.
5. One single AD user with access to all the required databases.
6. The user running the Pentaho Server to be able to create a Kerberos ticket through kinit with the credentials for the enabled user.
7. A valid ticket with an external tool that generates and renews the ticket when needed.

All tests and examples were developed to connect to the MS SQL Server 2012 R2. For other databases, follow the instructions provided by its documentation on configuring the JDBC properties, such as class, URL, etc.
Setting Up the Domain Controller and SQL Server

In this section, we will go over the steps to set up your domain controller (Active Directory) and SQL server.

You can find details on these topics in the following sections:

- Configuring Domain Controller
- Configuring Linux and Pentaho
- Configuring Pentaho Data Integration (PDI) Connection

Configuring Domain Controller

Make sure that the hostname is the same as the one used in the `setspn` command (either host only or FQDN), or the connection will fail with a Defective token detected (Mechanism level: AP_REP token id does not match!) error.

1. Install and configure the domain controller and SQL server to work together:

   ![Figure 1: SQL Server Properties](image)

   Figure 1: SQL Server Properties
2. In the domain controller, create a service principal name (SPN) with the `setspn.exe` utility, with the AD user running the SQL server service and its hostname. Be sure to configure your client with the same information.

   ```bash
   setspn.exe -A MSSQLSvc/eagsqlserver:1433 PENTAHOEAG\mssqlserver
   setspn.exe -L mssqlserver
   ```

   ![Figure 2: setspn Utility](image)

3. Create a new login in the SQL server for the AD user who has access to the data sources.

4. Create six databases. Make sure the new login is the owner of all the Pentaho Server databases: jackrabbit, hibernate, and quartz.

   ![Figure 3: Database Login Properties](image)
## Configuring Linux and Pentaho

1. Download the latest [MS SQL Server JDBC driver](#).
2. Unzip and place the SQL JDBC .jar file in the lib Pentaho class path:
   - **Pentaho Server**: pentaho-server/tomcat/lib
   - **Client tools (PDI)**: data-integration/lib/
3. Create a `/etc/krb5.conf` file with the details of your domain controller.
   ```
   $ cat /etc/krb5.conf
   [libdefaults]
   default_realm = PENTAHOEAG.LOCAL
dns_lookup_realm = false
dns_lookup_kdc = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
   
   [realms]
   PENTAHOEAG.LOCAL = {
     kdc = 10.100.14.101
     admin_server = 10.100.14.101
   }
   
   [domain_realm]
   .pentahoeag.local = PENTAHOEAG.LOCAL
   pentahoeag.local = PENTAHOEAG.LOCAL
   ```
4. Create a new Kerberos ticket with the `kinit` command and run the `klist–e` command to see all encryption types.
   ```
   $ kinit pentahouser@PENTAHOEAG.LOCAL
   Password for pentahouser@PENTAHOEAG.LOCAL:
   $ klist -e
   Ticket cache: FILE:/tmp/krb5cc_1000
   Default principal: pentahouser@PENTAHOEAG.LOCAL
   
   Valid starting       Expires              Service principal
   03/15/2016 15:10:42  03/16/2016 01:10:42
   krbtgt/PENTAHOEAG.LOCAL@PENTAHOEAG.LOCAL
   ```
5. If the encryption type is `aes256` (Etype (skey, tkt): aes256-cts-hmac-shal-96), then Java Cryptography Extension (JCE) unlimited strength is required:
   a. Download the [JCE zip file](#) from Oracle.
   b. Back up the `US_export_policy.jar` and `local_policy.jar` from the `lib/security` directory of the Java 8 runtime environment (JRE) (`/usr/lib/jvm/java-8-oracle/jre/lib/security`).
   c. Overwrite the `US_export_policy.jar` and `local_policy.jar` files with the ones included in the zip file.
Configuring Pentaho Data Integration (PDI) Connection

1. Make sure a Kerberos ticket is active, and then start PDI.
2. Create a new job or transformation.
3. In the View tab, right-click on Database Connections, and choose New.
4. In the General section, create a new database connection. Choose MS SQL Server (Native) as the Connection Type, and Native (JDBC) as the Access.
5. Configure the connection with the details listed in the figure and table:

![Database Connection General Section](image)

**Figure 4: Database Connection General Section**

**Table 1: Database Connection Details**

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Name</strong></td>
<td>This is the hostname of the SQL server. The hostname must be the same as the one used in the setspn command (either host only or FQDN), or the connection will fail with a “Defective token detected (Mechanism level: AP_REP token id does not match!)” error.</td>
</tr>
<tr>
<td><strong>Database Name</strong></td>
<td>This is the name of the database. The user who generated the Kerberos ticket must be able to access this database.</td>
</tr>
<tr>
<td><strong>Port Number</strong></td>
<td>Use 1433, or whatever port you configured in the MS SQL Server.</td>
</tr>
<tr>
<td><strong>Instance Name</strong></td>
<td>Leave this blank.</td>
</tr>
<tr>
<td><strong>User Name</strong></td>
<td>Leave this blank.</td>
</tr>
<tr>
<td><strong>Password</strong></td>
<td>Leave this blank.</td>
</tr>
</tbody>
</table>
6. Click on the **Options** section, and create a new parameter named **authenticationScheme** with a value of **JavaKerberos**:

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Integrated Security</td>
<td>Check this box.</td>
</tr>
</tbody>
</table>

![Figure 5: Options Tab - New Parameter](image)

7. Click on **Test**. A success message should appear:

![Figure 6: Database Connection Test](image)
Pentaho Server Configuration

In the following section, you will find instructions to configure your Pentaho Server.

You can find details on the following topics:

- Verifying Database Configuration
- Configuring Pentaho Server's Manage Connections
- Configuring context.xml

Verifying Database Configuration

Follow Pentaho’s instructions to Use PostgreSQL as Your Repository Database. The changes described below must apply to the standard configuration for MS SQL Server described in the Use MS SQL Server as Your Repository Database documentation.

1. You should have already created the following databases and granted ownership to the AD user who will be using them in the Kerberos ticket:
   - jackrabbit
   - quartz
   - hibernate
   *Do not run the scripts as explained in this section; create_jcr_sqlServer.sql and create_repository_sqlServer.sql are not required.*

2. Modify create_quartz_sqlServer.sql by removing the first Begin and End block, and add USE quartz, before the first IF EXISTS statement.

3. Run the create_quartz_sqlServer.sql file, as explained in the Pentaho documentation.

4. Open the following files with any text editor:
   - pentaho-solutions/system/hibernate/sqlserver.hibernate.cfg.xml
   - pentaho-solutions/system/jackrabbit/repository.xml
   - tomcat/webapps/pentaho/META-INF/context.xml

5. In all three files, verify the following:
   - The hostname for the SQL server connection is the same as the one used in the setspn command.
   - Any reference to the connection URL has both the integratedSecurity=true and authenticationScheme=JavaKerberos properties.

   ```
   jdbc:sqlserver://eagsqlserver:1433;DatabaseName=hibernate;integratedSecurity=true;authenticationScheme=JavaKerberos
   ```

   - Make sure that any reference to the username and password for the database connection is set to empty.
Configuring Pentaho Server's Manage Connections

Once the Pentaho Server is correctly configured and running, create a new connection through the **Manage Connections** option in the Pentaho User Console (PUC). Here are the steps:

1. Log in to the PUC.
2. Go to **File Menu > Manage Data Sources**.
3. Add a new connection by clicking on the **Tools** icon (gear icon) and selecting **New Connection**.
4. Create a new database connection with the following details:

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name</td>
<td>This is the hostname of the SQL server. The hostname must be the same as the one used in the <code>setspn</code> command (either host only or FQDN), or the connection will fail with a “Defective token detected (Mechanism level: AP_REP token id does not match)” error.</td>
</tr>
<tr>
<td>Database Name</td>
<td>This is the name of the database. The user who generated the Kerberos ticket must be able to access this database.</td>
</tr>
<tr>
<td>Port Number</td>
<td>Use 1433, or whatever port you configured in the MS SQL Server.</td>
</tr>
<tr>
<td>Instance Name</td>
<td>Leave this blank.</td>
</tr>
<tr>
<td>User Name</td>
<td>Leave this blank.</td>
</tr>
<tr>
<td>Password</td>
<td>Leave this blank.</td>
</tr>
<tr>
<td>Use Integrated Security</td>
<td>Check this box.</td>
</tr>
</tbody>
</table>

5. Click **Options**, and then create a new parameter named `authenticationScheme` with a value of `JavaKerberos`.

Configuring `context.xml`

1. Stop the Pentaho Server.
2. Open the `context.xml` located in `tomcat/webapps/pentaho/META-INF` for the Pentaho Server or `tomcat/webapps/pentaho-di/META-INF` with any text editor and:
   a. Add the new connection definitions within the `<Context>` tags.
   b. Make sure the hostname for the SQL server connection is the same as the one used in the `setspn` command.
   c. Make sure any reference to the connection URL has both properties set:
      `integratedSecurity=true` and `authenticationScheme=JavaKerberos`.

   ```
   jdbc:sqlserver://eagsqlserver:1433;DatabaseName=di_hibernate;integratedSecurity=true;authenticationScheme=JavaKerberos
   ```

3. Make sure that any reference to the `username` and `password` for the database connection is set to `empty`. 
Related Information

Here are some links to information that you may find helpful while using this best practices document:

- Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 7 Download
- MS SQL Server JDBC Driver
- Pentaho
  - Components Reference
  - Use MS SQL Server as Your Repository Database
  - Use PostgreSQL as Your Repository Database