HITACHI
Inspire the Next

Connecting the PDI Client to a Secure Hadoop Cluster

HITACHI
Inspire the Next

Change log (if you want to use it):

Date	Version	Author	Changes

Contents

Overview

This document covers best practices on methods and strategies regarding the different options to execute processes and authenticate users with Big Data using the Windows operating system.

Our intended audience is Pentaho administrators or anyone with Pentaho Data Integration (PDI) experience who is interested in improving authentication setups.

Software	Version(s)
Pentaho	6.x, 7.x, 8.0

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

Before You Begin

Pentaho integrates Enterprise level security with Java Authentication and Authorization Service (JAAS). Included with JAAS are multiple implementation modules for authentication and authorization services; here we will focus on Kerberos (KRB5) implementation and the way that Pentaho interacts with it.

In Hadoop secure environments, it is often necessary to use unlimited encryption libraries. Make sure the Java Platform, Standard Development Kit (JDK) used by Pentaho Platform has installed the JCE.

PDI Kerberos Authentication

PDI uses Kerberos as a network authentication protocol for all component communications and for execution of processes. Kerberos uses a ticketing system to grant accesses.

This diagram illustrates how Kerberos interacts with PDI 6.x. Later versions of PDI support a dynamic (impersonation) Kerberos principal as well as a static one. Use Secure Impersonation to Access a Cloudera Cluster has information about Pentaho, Kerberos, and impersonation.

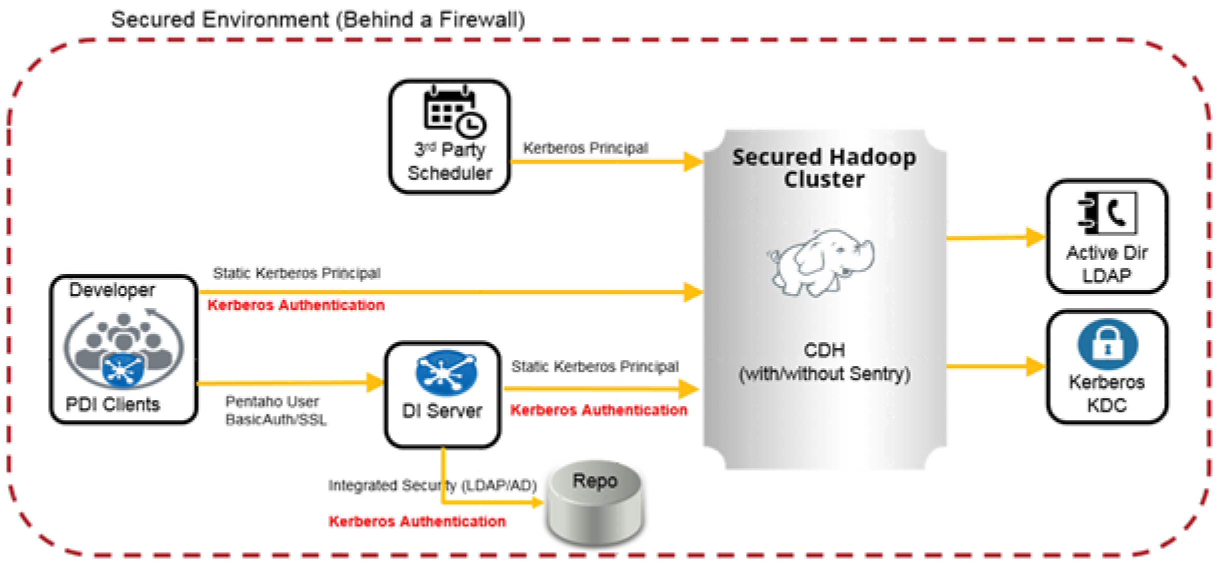

Figure 1: Kerberos Ticketing System for Pentaho 6.x

PDI provides Kerberos support for Cloudera, Hortonworks, and MapR, Amazon EMR, or MS Azure. Once the correct shim configuration is installed, there are two options to consider:

- Configuring Pentaho to Use One Static Kerberos Ticket
- Enabling Users to Use Their Own Kerberos Tickets

Whichever of these options you choose, follow it with the next section in this document by setting up the Kerberos client in Windows to make the Kerberos tickets available to Java applications.

Configuring Pentaho to Use One Static Kerberos Ticket

Shim configuration is closely tied to the Hadoop distribution in Pentaho, and is needed to connect to MapR, Cloudera, Hortonworks, Amazon EMR, or MS Azure. Implement Kerberos Authentication in the Pentaho Documentation has specific steps on how to use Kerberos to provide PDI client users access to your Hadoop cluster.

Before you configure the shim, make note of the Kerberos `principal` and `keytab` location or password, so that you can specify it in the `config.properties` file.

This configuration lets Pentaho get a ticket from the Kerberos Key Distribution Center (KDC) server. All interactions coming from either the server or the PDI client will use the specified ticket. This is the preferred method for Pentaho Servers or for a specific PDI client dedicated to a purpose, and is acceptable if you are using one Kerberos ticket for all activities.

Enabling Users to Use Their Own Kerberos Tickets

If your situation requires that all PDI activities use the individual user's Kerberos ticket previously obtained by Linux or Windows, this means that Pentaho software will not issue a Kerberos ticket itself.

However, if that individual user's ticket is valid, and is already in the proper cache area, it will be used to authenticate all the activities within the cluster.

There are certain limitations to this approach and some important items that need to be considered:

Multiple JVM Processes

When combining this approach with external schedulers using kitchen or pan commands, if multiple ETL processes are executed at the same time, each process will start a new JVM process. This could cause a problem based on machine resources available.

Ticket Lifetime Period

Make sure that the ticket lifetime period will cover the amount of time that the requested job or transformation will take to execute all tasks.

If the process is an orchestration process, the ticket needs to remain valid for the entire duration of the orchestration process. Orchestration processes include things such as:

- Querying to push some more work to the cluster
- Waiting until the transformation work is finished
- Submitting or pushing any additional activities

Ticket Cache Area

Do not attempt to change the authentication batch on the fly (while a JVM is using it). Instead, change it at the beginning of the processes.

We recommend that you set up a specific ticket cache location for each process if you have multiple processes that need to execute as same users.

You can do this either by setting the `KRB5CCNAME` environmental variable, or ensuring each process is executed by a different Unix user.

Setting Up the Kerberos Client in Windows

Setting up the Kerberos client in Windows makes Kerberos tickets will be available to Java applications. You can choose any Windows-compatible Kerberos client for this purpose. We recommend the MIT Kerberos Client application:

1. Download and install your chosen Kerberos client.
2. Copy a valid `krb5.ini` or `krb5.conf` configuration file and place it in a location such as `C:\kerberos\`.
3. From the desktop's **Control Panel**, search for **environment variables** and open the **Environment Variables** editor.
4. Add a system environment variable called `KRB5_CONFIG` that points to the full path of the `krb5.ini` or `krb5.conf` file you created.
5. Create another system environment variable called `KRB5CCNAME` to specify the directory where the Kerberos cache will be stored, such as `C:\kerberos\krbcache`.
6. Open the Kerberos client and issue a ticket.
7. Type your Kerberos principal (`your_username@REALM`) and your password.

Pentaho YARN Cluster and Authentication

A PDI on YARN cluster is a way to make Hadoop YARN start PDI Carte servers. This makes it possible to use your cluster to distribute large processing transformations. Our documentation on Big Data On-Cluster Processing has more information about setting up YARN clusters for Pentaho. You can find details on these topics in the following sections:

- Authentication and Kerberos Integration
- Testing the Authentication in the PDI Client

Authentication and Kerberos Integration

For authentication and Kerberos integration, the YARN package that is uploaded and distributed by Hadoop Distributed File System (HDFS) to the containers must be set up with a valid Kerberos static configuration such that it will not use the executing user's configuration. This is necessary because the container will not have a Kerberos ticket in the default cache location.

If the shim configuration specifies a `keytab` file for authentication, make sure that this `keytab` file is accessible from any data node or container where Carte server is running.

We recommend placing the `keytab` file on all Hadoop nodes under the same directory location, for example, in `/etc/pentaho/carte.keytab`.

Testing the Authentication in the PDI Client

To test the authentication in the PDI client, run a transformation with a step that connects to the Hadoop cluster. You should have `read` and `write` access to your home directory on the Hadoop cluster.

1. Create a new transformation. If needed, make one that connects to the Hadoop cluster:
   a. Drag the **Generate Rows** step to the canvas.
   b. Right-click the step and choose **Edit...**.
   c. Indicate a limit (the number of rows you want to generate), then put in field information, such as the name of the field, type, and a value.
   d. Click **Preview** to ensure that data is generated, then click the **Close** button to save the step.

2. Drag a **Hadoop File Output** step onto the canvas, then create a hop between the **Generate Rows** and **Hadoop File Output** steps.
   a. Right-click the **Hadoop File Output** step and choose **Edit...**.
   b. In the **Filename** field, indicate the path of the file that will contain the output of the **Generate Rows** step. The path should exist on the Hadoop cluster.
   c. Make sure that you indicate an extension such as `txt`, create a parent directory, and add filenames to the result.
   d. Click the **OK** button, then save the transformation.

3. Run the transformation. If there are errors, correct them, and rerun it.
4. After it finishes running, open a terminal window.

You will now be able to view the results of the output file on the Hadoop filesystem.

For example, if you saved your file to a file named `test.txt`, you could type a command like this: `hdfs fs -cat /user/pentaho-user/test/test.txt`.

Known Issues and Solutions

Here are some known issues with these procedures, and solutions or workarounds for them:

YARN Standby Mode Causes PDI MapReduce Job Slowdown

In this situation, PDI MapReduce jobs can seem to take forever. This can happen because when in the "Configuring Pentaho MapReduce job to use Kettle (PDI) Installation" stage on High Availability (HA), the active YARN Resource Manager (RM) goes into standby mode and is replaced by another RM.

Solution: Change the `yarn.resourcemanager.principal` property to reflect this RM replacement.

Virtual File System (VFS) Objects Error While Browsing HDFS

If you encounter an error about resolving VFS objects while connecting to the cluster to browse HDFS contents, it is most likely related to Java's inability to find the value of the `KRB5_CONFIG` variable.

Solution: If this happens, the workaround is to pass the variable in the command-line when the PDI client is launched. You may also want to check that you have the correct `krb5.ini` file in place.

Incorrect Principal Errors when Switching Environments

You may get "incorrect principal" errors when switching between development/QA/production environments. This may happen when a configuration is left unchanged, and when trying to connect to a cluster server using a different `REALM` for the Kerberos ticket.

Solution: Make sure the cluster configuration files for the connecting cluster are in place so that you can switch between development, QA, and production clusters.

YARN Failed to Renew Token Error

You may get an error message saying "YARN failed to renew token with renewer nobody." This issue is related to the way YARN replaces the `_HOST` macro in the `yarn.resourcemanager.principal` property. It inserts the user's Windows `hostname` instead of the `hostname` of the currently active YARN Resource Manager.

Since the user's Windows `hostname` is not whitelisted in the `AUTH_TO_LOCAL` rules in `core-site.xml`, the renewer `yarn/<windows host>@<REALM>` is mapped to a user called nobody and the ticket is never renewed.

Solution: A workaround is to override the value by editing the `yarn-site.xml` file (in the PDI client's Hadoop-configurations folder), and changing the value of `yarn.resourcemanager.principal` from `yarn/_HOST@<REALM>` to `yarn/<current active RM>@<REALM>`.

PDI MapReduce Jobs Fail While Running from Windows

In this situation, you may see PDI MapReduce jobs failing. Upon inspecting the logs, you see several log messages with a stale progress indication before an error occurs.

Solution: To fix this:

1. Make sure that the `mapreduce.app-submission.cross-platform` property is set to `true` in the cluster. The default is `false`, and that setting will cause all Windows job submissions to be rejected by YARN, as they come from a different operating system.
2. Re-deploy the client configuration and update the PDI shim.

Appendix

Here is some supplemental information around AES or JCE password encryption.

Apply AES Password Encryption

There are two ways to secure passwords in PDI: Kettle obfuscation and AES. Kettle obfuscation is the default method, but we recommend using the Advanced Encryption Standard (AES) for increased security.

The password security method you choose is applied to all passwords, including those in database connections, transformation steps, and job entries.

Apply AES Password Encryption in the Pentaho Documentation has more information on how to do this.

Install Java Cryptography Extension (JCE) Unlimited Strength

The Kerberos Key Distribution Center (KDC) configuration includes an AES-256 encryption setting. If an export level of AES-256 encryption is required, you will need to install the JCE files.

1. Download the JCE for the currently supported version of Java from the Oracle site.
2. Read and follow the installation instructions that are included with the download. Copy the JCE jars to the `java/lib/security` directory where PDI is installed on the Linux client machine.

Related Information

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

- Pentaho
    - Apply AES Password Encryption
    - Components Reference
    - Connect to a Hadoop Cluster with the PDI Client
    - Implement Kerberos Authentication
    - Pentaho Big Data On-Cluster Processing
    - Use Secure Impersonation to Access a Cloudera Cluster
- External Sites
    - Java Authentication and Authorization Service (JAAS) Reference Guide
    - Java Cryptography Extension (JCE)
    - Introduction to Kerberos Authentication
    - MIT Kerberos Client

Finalization Checklist

This checklist is designed to be added to any implemented project that uses this collection of best practices, to verify that all items have been considered and reviews have been performed.

Name of the Project:____

Date of the Review:____

Name of the Reviewer:____

Item	Response	Comments
Did you configure Pentaho to use one static Kerberos ticket?	YES_______ NO_______	
Did you enable users to use their own Kerberos tickets?	YES_______ NO_______	
Did you make sure the ticket lifetime period will cover the amount of time the requested job or transformation will take to execute all tasks?	YES_______ NO_______	
Did you set up a specific ticket cache location for each process?	YES_______ NO_______	
Did you set up the Kerberos client in Windows?	YES_______ NO_______	
Did you set up the YARN package with a Kerberos static configuration?	YES_______ NO_______	
Did you test authentication in the PDI client?	YES_______ NO_______	
Did you apply AES password encryption?	YES_______ NO_______	
Did you install JCE unlimited strength?	YES_______ NO_______	