Authentication for HiveServer2

You can configure authentication for in-bound client connection to HiveServer2. Clients of HiveServer 2 include beeline and odbc/jdbc client applications.

Credentials are submitted from the HiveServer2 clients to HiveServer2 as plain text. To secure the credential transmission, MapR supports SSL encryption for HiveServer2. For information about how to configure encryption, see Hive Encryption.

HiveServer2 supports the following authentication methods:

- MapR-SASL Authentication
- LDAP Authentication using OpenLDAP
- Pluggable Access Modules (PAM) Authentication
- Custom Authentication
- Kerberos Authentication

MapR-SASL Authentication

MapR-SASL is available starting with the 1504 release of Hive 0.13 and Hive 1.0. However, the configuration requirements for MapR-SASL differ based on the version of Hive that you have installed:

- As of Hive 0.13-1510, Hive 1.0-1510, and Hive 1.2-1510, MapR-SASL and PAM are enabled by default on a secure cluster; no configuration is required. Complete the steps below if you want HiveServer2 to only accept MapR-SASL authentication.
- In Hive 0.13-1508 and Hive 1.0-1508, MapR-SASL is not the default and must be configured.
- In Hive 0.13-1504 and Hive 1.0-1504, MapR-SASL is the default authentication method when the cluster is secure. No configuration is required.

1. Configure the following property in hive-site.xml on each node where HiveServer2 is installed:

Property	Value
hive.server2.authentication	MAPRSASL

```
<property>
  <name>hive.server2.authentication</name>
  <value>MAPRSASL</value>
</property>
```

2. Restart HiveServer2 to apply these changes.

```
maprcli node services -name hs2 -action restart -nodes <comma separated list of
nodes>
```

LDAP Authentication using OpenLDAP

1. Configure the following properties in the `hive-site.xml` file on each node where HiveServer2 is installed:

Property	Value
`hive.server2.authentication`	LDAP
`hive.server2.authentication.ldap.url`	<The access URL for your LDAP server>
`hive.server2.authentication.ldap.baseDN`	<The base LDAP DN for your LDAP server. For example, `ou=People,dc=mycompany,dc=com`.>

```
property>
  <name>hive.server2.authentication</name>
  <value>LDAP</value>
</property>
<property>
  <name>hive.server2.authentication.ldap.url</name>
  <value><LDAP URL></value>
</property>
<property>
  <name>hive.server2.authentication.ldap.baseDN</name>
  <value><LDAP Base DN></value>
</property>
```

2. Restart HiveServer2 to apply these changes.

```
maprcli node services -name hs2 -action restart -nodes <comma separated list of
nodes>
```

Pluggable Access Modules (PAM) Authentication

The configuration requirements for PAM differ based on the version of Hive that you have installed:

- As of Hive 0.13-1501, Hive 1.0-1510, and Hive 1.2-1510, MapR-SASL and PAM are enabled by default on a secure cluster; no configuration is required.
- In Hive 0.13-1508 and Hive 1.0-1508, PAM is the default authentication method for HiveServer2 on a secure cluster; no configuration is required.
- In Hive 0.13-1504 and Hive 1.0-1504, PAM is not the default authentication method and therefore it requires the following configuration steps.

1. Configure the following properties in the hive-site.xml on the hiveserver2 node:

Property	Value
`hive.server2.authentication`	`PAM`
`hive.server2.authentication.pam.services`	<A comma-separated list of pam module>

```
<property>
   <name>hive.server2.authentication</name>
   <value>PAM</value>
</property>
<property>
   <name>hive.server2.authentication.pam.services</name>
   <value>login,sudo</value>
   <description>comma separated list of pam modules to verify</description>
</property>
```

2. Restart HiveServer2 to apply these changes.

```
maprcli node services -name hs2 -action restart -nodes <comma separated list of
nodes>
```

Custom Authentication

You can configure HiveServer2 to use custom authentication.

1. Create a custom Authenticator class derived from the following interface:

```
public interface PasswdAuthenticationProvider {
  /**
   * The Authenticate method is called by the HiveServer2 authentication layer
   * to authenticate users for their requests.
   * If a user is to be granted, return nothing/throw nothing.
   * When a user is to be disallowed, throw an appropriate {@link
AuthenticationException}.
   *
   * For an example implementation, see {@link LdapAuthenticationProviderImpl}.
   *
   * @param user - The username received over the connection request
   * @param password - The password received over the connection request
   * @throws AuthenticationException - When a user is found to be
   * invalid by the implementation
   */
  void Authenticate(String user, String password) throws AuthenticationException;
}
```

   The attached `SampleAuthenticator.java` code has an example implementation that has stored usernames and passwords.
2. Configure the following properties in the `hive-site.xml` file on each node where HiveServer2 is installed:

Property	Value
`hive.server2.authentication`	`CUSTOM`
`hive.server2.custom.authentication.class`	<The authentication class name. For example, `hive.server2.custom.authentication.class`>

```
<property>
<name>hive.server2.authentication</name>
<value>CUSTOM</value>
</property>

<property>
<name>hive.server2.custom.authentication.class</name>
<value>hive.test.SampleAuthenticator</value>
</property>
```

3. Restart Hiveserver2 to apply the changes:

```
maprcli node services -name hs2 -action restart -nodes <comma separated list of
nodes>
```

Kerberos Authentication

You can configure HiveServer2 to use Kerberos authentication.

MapR clusters do not provide Kerberos infrastructure. The tips in this section assume a Linux-based Kerberos environment, and the specific commands for your environment may vary. Consult with your Kerberos administrator for assistance.

Configuring HiveServer 2 to use Kerberos

Enabling HiveServer to use Kerberos authentication requires following steps on each node where HiveServer 2 is installed:

1. Create a Kerberos Identity and keytab.
   You can use the following commands in a Linux-based Kerberos environment to set up the identity and update the keytab file:

   ```
   # kadmin
       : addprinc -randkey username/<FQDN@REALM>
       : ktadd -k /opt/mapr/conf/hive.keytab username/<FQDN@REALM>
   ```

   The `hive.keytab` file must be owned and readable only by the `mapr` user.

2. Configure the following properties in hive-site.xml on each node where hiveserver2 is installed:

Property	Value
hive.server2.authentication	KERBEROS
hive.server2.authentication.kerberos.principal	<HiveServer2 Principle. For example, mapr/FQDN@REALM>
hive.server2.authentication.kerberos.keytab	<The keytab file for the HiverServer2 principle. For example, /opt/mapr/conf/hive.keytab>

   ```
   <property>
     <name>hive.server2.authentication</name>
     <value>KERBEROS</value>
     <description>authenticationtype</description>
   </property>
   <property>
     <name>hive.server2.authentication.kerberos.principal</name>
     <value>mapr/FQDN@REALM</value>
     <description>HiveServer2 principal. If _HOST is used as the FQDN portion, it
   will be replaced with the actual hostname of the running instance.</description>
   </property>
   <property>
     <name>hive.server2.authentication.kerberos.keytab</name>
     <value>/opt/mapr/conf/hive.keytab</value>
     <description>Keytab file for HiveServer2 principal</description>
   </property>
   ```

3. Reconfigure following options in `env.sh` (`/opt/mapr/conf/env.sh`) on each node where hiveserver2 is installed:

Existing Configuration	Required Configuration
`MAPR_HIVE_SERVER_LOGIN_OPTS="-Dhadoop.login=maprsasl_keytab"` `MAPR_HIVE_LOGIN_OPTS="-Dhadoop.login=maprsasl"`	`MAPR_HIVE_SEVER_LOGIN_OPTS="-Dhadoop.lo` `MAPR_HIVE_LOGIN_OPTS="-Dhadoop.login=hy`

These configuration are listed in the portion of the file that begins with `if [ "$MAPR_SECURITY_STATUS" = "true" ];`

4. Restart HiveServer2 to apply these changes.

```
maprcli node services -name hs2 -action restart -nodes <comma separated list of
nodes>
```

Configuring HiveServer 2 Clients to use Kerberos when Authenticating with HiveServer2

- On each node where HiveServer2 clients (not including Beeline) are installed, reconfigure the following option in env.sh (`/opt/mapr/conf/env.sh`):

Existing Configuration	Required Configuration
`MAPR_HIVE_LOGIN_OPTS="-Dhadoop.login=maprsasl"`	`MAPR_HIVE_LOGIN_OPTS="-Dhadoop.login=hybrid"`

This configuration is listed in the portion of the file that begins with `if [ "$MAPR_SECURITY_STATUS" = "true" ];`

- On each node where Beeline is installed, reconfigure the following option in beeline.sh ($hive_home/bin/ext/beeline.sh):

Existing Configuration	Required Configuration
`HADOOP_OPTS="$HADOOP_OPTS${MAPR_HIVE_LOGIN_OPTS}"`	`HADOOP_OPTS="$HADOOP_OPTS${KERBEROS_LOGIN_OPTS}"`

For more information, see Connecting to Hive.

The MAPR_HIVE_LOGIN_OPTS and MAPR_HIVE_SERVER_LOGIN_OPTS were added in 1504 release of Hive 0.13 and Hive 1.0. If you have Hive 0.13 from a prior release, you do not need to configure these properties. Instead, set MAPR_ECOSYSTEM_LOGIN_OPTS and MAPR_ECOSYSTEM_SERVER_LOGIN_OPTS to `"-Dhadoop.login=hybrid"` in /opt/mapr/conf/env.sh.