Configure for Hive with Ranger

Contents:

This section describes how to ensure that the Trifacta® platform is configured correctly to connect to Hive when Ranger is enabled for Hive. Ranger provides role-based authorization for Hive and other Hadoop components on the Hortonworks platform.

Ranger effectively functions as a whitelist of URI's; by default, access is denied for any object in Hive. When a URI is requested, Ranger checks HDFS for permissions for the authenticated user. If HDFS denies access, then Ranger checks its defined set of URI's for the permission and, if a match is found, grants access for the authenticated user.

- For more information, see *http://hortonworks.com/blog/best-practices-for-hive-authorization-using-apache-ranger-in-hdp-2-2/*

Pre-requisites

Before you begin, please verify that your enterprise has deployed both Hive and Ranger according to recommended configuration practices. For more information, please consult the documentation that was provided with your Hadoop distribution.

NOTE: Before you begin, you must integrate the Trifacta platform with Hive. See *Configure for Hive*.

Secure Impersonation with Trifacta platform and Hive with Ranger

Secure impersonation ensures consistent and easily traceable security access to the data stored within your Hadoop cluster.

NOTE: Although not required, secure impersonation is highly recommended for connecting the platform with Hive.

Since secure impersonation for the combination of HiveServer2 and Ranger is not supported by Ranger, you must apply the following additional configuration changes to the Trifacta platform to enable secure impersonation in the environment:

1. Enable the platform with secure impersonation. See *Configure for Secure Impersonation* for details.
2. Add the hive service user `hive` to the Unix or LDAP group `[os.group` (default=`trifacta`)`]`.
3. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`. For more information, see *Platform Configuration Methods*.
4. Set the following parameter:

```
"hdfs.permissions.userUmask" = 027
```

5. Ensure that the Unix or LDAP group has read access to the Hive warehouse directory, as described in the following section. For more information, see *http://hortonworks.com/blog/best-practices-for-hive-authorization-using-apache-ranger-in-hdp-2-2/*.

Users and Groups for Ranger

When the Trifacta platform is enabled with secure impersonation and submits requests to Hive, the following steps occur:

1. The platform authenticates as the `[hadoop.user.principal` (default=`trifacta`)`]` user through Kerberos.
2. The Hive server authorizes access to the underlying table through Ranger as the Hadoop principal user assigned to `[hadoop.user.principal]`.

   > **NOTE:** This Hadoop principal is the user that should be configured through policies in Ranger to have the appropriate privileges.

3. The Hive server executes access to the physical data file on HDFS as the Unix user `hive`, which should be part of the group `[hadoop.group` (default=`trifactausers`)`]`.

> **NOTE:** Since Ranger assigns access to databases, tables, and columns to Unix users and groups, a common practice is to assign the Hadoop principal users (used by Trifacta users) to dedicated Unix groups that are separate from the Unix group `[os.group` (default=`trifacta`)`]` use within Ranger. Ranger should not grant any privileges and roles to the Unix group `[os.group` (default=`trifacta`)`]`.

> **NOTE:** In UNIX environments, usernames and group names are case-sensitive. Please verify that you are using the case-sensitive names for users and groups in your Hadoop configuration and Trifacta configuration file.

Policies in Ranger

In Ranger, you can configure access through policies. A Ranger **policy** is a combination of:

- Specified database, table, or tabled column
- Permissions associated with that specified object.
- Assignment of permissions to individual users or groups

Required Permissions

> **NOTE:** In general, to manage access through Ranger, permissions to underlying Hadoop components such as HDFS or Hive should be minimized within those components. All permissions in Ranger are additive, which means that you should be careful about overlapping users and groups.

The following components require these permissions at a minimum to be assigned to the Hadoop principal:

Component	Permissions
HDFS	Read, Write, Execute
Hive	Select, Update. Create (for Hive publishing)

Configuration

> **NOTE:** The following configuration is required for integration of HDP with Hive publishing when Ranger is enabled.

1. In the Ambari console, navigate to the following: **HDFS > Configs > Advanced > Advanced ranger-hdfs-plugin-properties**.
2. Set the following to `true`: **Enable Ranger for HDFS**.
3. From the left nav bar, navigate to the following: **Ranger > Configs > Ranger Plugin tab**.
4. Set the following to `true`: **Hive Ranger Plugin**.
5. Please verify that the other Ambari properties are set to integrate Hive through Ranger. For more information, see the HDP documentation.
6. Restart the HDP cluster.
7. Open Ranger.
8. In the policies area, create the following two policies:

   ```
   trifacta_policies
   hive_warehouse
   ```

9. Set the following properties on these policies:
   a. Policy Type: `Access`
   b. Enabled: `true`
   c. Resource path:
      i. For `trifacta_policies`, set this value to either of the following values:

      ```
      /trifacta
      /trifacta/queryResults
      ```

      ii. For `hive_warehouse`, set this value to the location of the Hive warehouse. The following is the default value:

      ```
      /user/hive/warehouse
      ```

   d. Recursive: `true`

   > **NOTE:** Policies must be recursive.

   e. Audit Logging: `yes`
   f. Allow conditions:
      i. Select group: `Hadoop`, `Trifacta`
      ii. Select user: `Trifacta`
      iii. Permissions: `Read`, `Write`, `Execute`
10. Save the policies.

Verify Operations

After you have completed your configuration changes, you should restart the platform. See *Start and Stop the Platform*.

To verify platform operations, run a simple job. For more information, see *Verify Operations*.