Configure Security for Relational Connections

Contents:

You can apply the following Trifacta® platform features to relational connections to ensure compliance with enterprise practices.

User Security

Connection Security Levels

Connection Security Level	Description
Private	Private connections are created by individuals and are by default accessible only to the individual who created them.
Private and shared	Optionally, they can be shared by individuals with other users. **NOTE:** If needed, credential sharing can be disabled. See below.
Global	Global connections are either created by administrators or are private connections promoted to global by administrators.

Credential Sharing

By default, users are permitted to share credentials through the application. Credentials can be shared in the following ways:

- A user can create a private connection to a relational database. Through the application, this private connection can be shared with other users, so that they can access the creator's datasets.
- When sharing a flow with another user, the owner of the flow can choose to share the credentials that are necessary to connect to the datasets that are the sources of the flow.

As needed, credential sharing can be disabled.

NOTE: If enterprise policy is to disable the sharing of credentials, collaborators may need to be permitted to store their source data in shared locations.

Tip: Credential sharing can be disabled by individual users when they share a connection. The connection is shared, but the new user must provide new credentials to use the connection.

Steps:

To disable credential sharing at the global level:

1. Login to the application as an administrator.
2. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta-conf.json`

   . For more information, see *Platform Configuration Methods*.
3. Locate the following parameter. Set this property to `false`:

   ```
   "webapp.enableCredentialSharing": true,
   ```

4. Save your changes and restart the platform.

Technical Security

The following features enhance the security of individual and global relational connections.

Encryption Key File

Relational database passwords are encrypted using key files:

- **Passwords in transit:** The platform uses a proprietary encryption key that is invoked each time a relational password is shared among platform services.
- **Passwords at rest:** For creating connections to your relational sources, you must create and reference your own encryption key file. This encryption key is accessing your relational connections from the web application.

This encryption key file must be created and installed on the Trifacta node. For more information, see *Create Encryption Key File*.

SSL

You can enable SSL by adding the following string to the Connect String Opts field:

```
?ssl=true;
```

> **Tip:** Some connection windows have a Use SSL checkbox, which also works.

Configure long load timeout limits

For long loading relational sources, a timeout is applied to limit the permitted load time. As needed, you can modify this limit to account for larger load times.

You can apply this change through the *Admin Settings Page* (recommended) or

`trifacta-conf.json`

. For more information, see *Platform Configuration Methods*.

1. Locate and edit the following parameter:

   ```
   "webapp.connectivity.longLoadTimeoutMillis": 120000,
   ```

2. Save your changes and restart the platform.

Property	Description
`longLoadTimeoutMillis`	Max number of milliseconds to wait for a long-loading data source. The default value is `120000` (2 minutes).

For additional relational configuration settings, see *Configure Data Service*.

Enable SSO authentication

Relational connections can be configured to leverage your enterprise Single Sign-On (SSO) infrastructure for authentication. Additional configuration is required. For more information, see *Enable SSO for Relational Connections*.