Informatica How-To Library

INFORMATICA

How to Configure Security for SQL Data Services

Abstract

You can configure security for SQL data services that are deployed in an application through pass-through and column level security. Pass-through security allows end users to view virtual table data based on their user credentials instead of the connection credentials. Column level security restricts end user access to virtual table columns for certain users or groups. This article explains how to configure pass-through and column level security for deployed SQL data services.

Supported Versions

- Informatica Data Services 9.0 - 9.0.1

Table of Contents

Overview

Informatica Developer users create SQL data services. An SQL data service is a virtual database that end users can query. Create an SQL data service to define uniform views of data. To make the data available for end users, deploy an application that contains the SQL data service. End users can query the virtual tables in an SQL data service through a third-party client tool as if the virtual tables were physical tables.

An SQL data service contains the virtual schemas and the virtual tables or stored procedures that define the database structure. If the virtual schema contains virtual tables, the SQL data service also contains virtual table mappings that define the flow of data between the sources and the virtual tables. When an end user queries a virtual table, the Data Integration Service uses the virtual table mapping to transform the data it retrieves from the sources. A virtual table mapping can combine data from different sources to create a single view of the data.

When the Data Integration Service retrieves source data, it uses the connections to the source database defined in the Developer tool or Administrator tool. Each source connection contains a user name and password. The Data Integration Service inherits permissions on the source database from the connection user name. For example, if the connection user name has read access on the Salary column in the Employee table, the Data Integration Service has read access on the Salary column. All end users that query the SQL data service also have read access on the Salary column.

Because database tables can contain sensitive information, you might want to restrict data access for end users that query SQL data services. You can control end user access to virtual table data through pass-through security and column level security. If you want end users to view source table data based on their user credentials instead of the connection credentials, use pass-through security. If you want to restrict access to certain columns for users or groups, use column level security.

Pass-through Security

Pass-through security is the ability to connect to an external source with the client user credentials instead of the credentials from a connection object. When the Data Integration Service retrieves virtual table data from a source, it connects to the source with the client user credentials. Therefore, users with different credentials have different levels of access to the data in a virtual table.

Use pass-through security to ensure that only end users that are authorized to view sensitive data can see the data. If end users query an SQL data service, and the query retrieves rows from a source they cannot view, the query fails.

Configure pass-through security for a connection in the Administrator tool. Enable pass-through security for the connection in each Data Integration Service that uses the connection.

Example

A large organization wants to restrict access to customer data. The organization manages users, groups, and permissions within an LDAP directory service. The organization wants to use the existing security infrastructure to ensure that employees in different groups can only view data that they are authorized to view.

The organization wants to create a virtual table, Customers, that combines data from the Insured, Prospect, and Agency relational tables. The three tables exist in different databases. The organization uses the LDAP service to restrict access to the source databases that contain the Insured and Prospect tables. Employees in the sales group can view data in the Prospect table, but employees in the billing group cannot.

The database that contains the Agency table does not contain sensitive information, so the organization does not restrict access to it.

The organization wants to apply its existing security settings to the Customer virtual table.

To apply security to the Customer virtual table, employees in the organization perform the following tasks:

1. An administrator for the organization uses the Administrator tool to create connections to the source databases. The administrator configures each connection with a default user name and password. The default user can access all tables in the source databases.
2. A developer for the organization uses the Developer tool to create the Customer virtual table in an SQL data service. The developer adds the SQL data service to an application and deploys the application to a Data Integration Service.
3. The administrator uses the Administrator tool to configure security for the connections. The administrator selects the Data Integration Service and configures the connections for the Insured and Prospect tables for pass-through security. The administrator does not configure pass-through security for the connection to the Agency table database.
4. The administrator enables and starts the application.

When the application is running, employees in the organization can use a third-party client tool to run SQL queries against the Customer virtual table.

An employee in the sales group runs a query that selects all customers in a region. When the Data Integration Service connects to the Insured and Prospect tables, it replaces the connection user name and password with the client user name and password. Because employees in the sales group can view all tables, the query returns customers from the Insured, Prospect, and Agency tables.

An employee in the billing group runs the same query. Because employees in the billing group cannot view the Prospect table, the query fails.

Pass-through Security with Data Object Caching

To use data object caching with pass-through security, you must enable caching in the pass-through security properties for the Data Integration Service.

When you deploy an SQL data service, you can choose to cache the logical data objects in a database. You must specify the database in which to store the data object cache. The Data Integration Service validates the user credentials for access to the cache database. If a user can connect to the cache database, the user has access to all tables in the cache. The Data Integration Service does not validate user credentials against the source databases when caching is enabled.

For example, you configure caching for the EmployeeSQLDS SQL data service and enable pass-through security for connections. The Data Integration Service caches tables from the Compensation and the Employee databases. A user might not have access to the Compensation database. However, if the user has access to the cache database, the user can select compensation data in an SQL query.

When you configure pass-through security, the default is to disallow data object caching for data objects that depend on pass-through connections. When you enable data object caching with pass-through security, verify that you do not allow unauthorized users access to some of the data in the cache. When you enable caching for pass-through security connections, you enable data object caching for all all pass-through security connections.

Adding Pass-through Security

Select the connections that use pass-through security.

1. In the **Administrator tool**, select the Data Integration Service.
2. Click the **Properties** view.
3. Edit the pass-through security options.

   The **Edit Pass-through Security Options** dialog box appears.
4. Optionally, click **New**to create a connection.
5. To choose pass-through connections, click **Select**. You can select multiple connections at a time.
6. Select **Allow Caching** to allow data object caching for the SQL data services that use the connections.
7. Click OK.

You must recycle the Data Integration Service to enable caching for the connections.

Column Level Security

Column level security is the ability to deny access to individual columns in a virtual table. When end users query columns that they are not authorized to view, the Data Integration Service returns substitute data values, null values, or an error.

To view virtual table data, end users must have select permission on the virtual table or SQL data service. Configure permissions for SQL data services and virtual tables through the Administrator tool. Configure column level security through infacmd.

Example

A financial services organization needs to make the Customer virtual table available to its employees. One column in the Customer virtual table contains credit card numbers. The organization must ensure that employees in the billing department can see the credit card numbers, while employees in other departments see a substitute value of 0000-0000-0000-0000.

To secure the credit card information, employees in the organization perform the following tasks:

1. A developer for the organization uses the Developer tool to create an SQL data service that contains the Customer virtual table. The developer creates an application that contains the SQL data service and deploys the application to a Data Integration Service.
2. An administrator for the organization uses the Administrator tool to create groups for the departments that needs to access Customer virtual table. The administrator creates two groups, Billing and NonBilling. The Billing group contains the employees authorized to view the credit card numbers. The NonBilling group contains employees authorized to view the Customer table, but not the credit card numbers.

3. The administrator disables the Data Integration Service that runs the application.
4. The administrator runs the infacmd sql UpdateColumnOptions command on the CardNo column in the Customer virtual table. The administrator sets the return value to 0000-0000-0000-0000.
5. The administrator runs the infacmd sql SetColumnPermissions command on the CardNo column in the Customer virtual table. The administrator specifies NonBilling as the group to deny permissions to.
6. The administrator enables and restarts the Data Integration Service.

When the application is running, employees in the organization can use a third-party client tool to run SQL queries against the Customer virtual table.

An employee in the Billing group runs a query that selects one customer by name. The employee sees all of the customer information including the customer credit card number. An employee in the NonBilling group runs the same query. The employee sees the all of the customer information, but the credit card number appears as 0000-0000-0000-0000.

Restricted Columns

When you configure column level security, set a column option that determines what happens when a user selects the restricted column in a query. You can substitute the restricted data with a default value. Or, you can fail the query if a user selects the restricted column.

For example, an Administrator denies a user access to the salary column in the Employee table. The Administrator configures a substitute value of 100,000 for the salary column. When the user selects the salary column in an SQL query, the Data Integration Service returns 100,000 for the salary in each row.

Run the infacmd sql UpdateColumnOptions command to configure the column options. You cannot set column options in the Administrator tool.

When you run infacmd sql UpdateColumnOptions, enter the following options:

ColumnOptions.DenyWith=*option*

Determines whether to substitute the restricted column value or to fail the query. If you substitute the column value, you can choose to substitute the value with NULL or with a constant value. Enter one of the following options:

- ERROR. Fails the query and returns an error when an SQL query selects a restricted column.
- NULL. Returns null values for a restricted column in each row.
- VALUE. Returns a constant value in place of the restricted column in each row. Configure the constant value in the ColumnOptions.InsufficientPermissionValue option.

ColumnOptions.InsufficientPermissionValue=*value*

Substitutes the restricted column value with a constant. The default is an empty string. If the Data Integration Service substitutes the column with an empty string, but the column is a number or a date, the query returns errors. If you do not configure a value for the DenyWith option, the Data Integration Service ignores the InsufficientPermissionValue option.

To configure a substitute value for a column, enter the command with the following syntax:

```
infacmd sql UpdateColumnOptions -dn empDomain -sn DISService -un Administrator -pd Adminpass -sqlds
employee_APP.employees_SQL -t Employee -c Salary -o ColumnOptions.DenyWith=VALUE
ColumnOptions.InsufficientPermissionValue=100000
```

If you do not configure either option for a restricted column, default is not to fail the query. The query runs and the Data Integration Service substitutes the column value with NULL.

Adding Column Level Security

Configure column level security with the infacmd sql SetColumnPermissions command. You cannot set column level security from the Administrator tool.

An Employee table contains FirstName, LastName, Dept, and Salary columns. You enable a user to access the Employee table but restrict the user from accessing the salary column.

To restrict the user from the salary column, disable the Data Integration Service and enter an infacmd similar to the following command:

```
infacmd sql SetColumnPermissions -dn empDomain -sn DISService -un Administrator -pd Adminpass -sqlds
employee_APP.employees -t Employee -c Salary gun -Tom -dp SQL_Select
```

The following SQL statements return NULL in the salary column:

```
Select * from Employee
Select LastName, Salary from Employee
```

The default behavior is to return null values.

Column Options

The following table describes the infacmd sql UpdateColumnOptions options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option or the environment variable INFA_DEFAULT_DOMAIN. If you set a domain name with both methods, the -dn option takes precedence.
-ServiceName -sn	service_name	Required. Name of the Data Integration Service where the application is deployed.
-UserName -un	user_name	Required. User name to connect to the domain. You can set the user name with the -un option or the environment variable INFA_DEFAULT_DOMAIN_USER. If you set a user name with both methods, the -un option takes precedence.
-Password -pd	password	Required. Password for the user name. The password is case sensitive. You can set a password with the -pd option or the INFA_DEFAULT_DOMAIN_PASSWORD environment variable. If you set a password with both methods, the password set with the -pd option takes precedence.
-SecurityDomain -sdn	security_domain	Required if you use LDAP authentication. Name of the security domain that the user belongs to. You can set a security domain with the -sdn option or the environment variable INFA_DEFAULT_SECURITY_DOMAIN. If you set a domain name with both methods, the -sdn option takes precedence. Security domain is case sensitive.
-ResilienceTimeout -re	timeout_period_in_seconds	Optional. Amount of time in seconds that infacmd attempts to establish or re-establish a connection to the domain. You can set the resilience timeout period with the -re option or the environment variable INFA_CLIENT_RESILIENCE_TIMEOUT. If you set a the resilience timeout period with both methods, the -re option takes precedence.
SQLDataService -sqlds	sql_data_service	Required. Name of the SQL data service with the virtual table.

Option	Argument	Description
Table -t	schema.table	Required. Name of the virtual table. Enter table in the following format: <schema_name>.<table_name>
Column -c	column	Column name.
Options -o	options	Required. Enter each option separated by a space. To view current options, run the infacmd sql ListColumnOptions command. ColumnOptions.DenyWith= - ERROR. Fails the query and returns an error. - NULL. Returns null values for a restricted column in each row. - VALUE. Returns a constant value in place of the restricted column in each row. Configure the constant value in the InsufficientPermissionValue option. ColumnOptions.InsuffiicientPermissionValue = <constant> Substitutes the restricted column value with a constant value. The default is an empty string. If you do not configure ColumnOptions.DenyWith the Data Integration Service ignores the InsufficientPermissionValue option.

Column Level Security Options

The following table describes infacmd sql SetColumnPermissions options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. Name of the Informatica domain. You can set the domain name with the -dn option or the environment variable INFA_DEFAULT_DOMAIN. If you set a domain name with both methods, the -dn option takes precedence.
-ServiceName -sn	service_name	Required. Name of the Data Integration Service where the application is deployed.
-UserName -un	user_name	Required. User name to connect to the domain. You can set the user name with the -un option or the environment variable INFA_DEFAULT_DOMAIN_USER. If you set a user name with both methods, the -un option takes precedence.
-Password -pd	password	Required. Password for the user name. The password is case sensitive. You can set a password with the -pd option or the environment variable INFA_DEFAULT_DOMAIN_PASSWORD. If you set a password with both methods, the password set with the -pd option takes precedence.
-SecurityDomain -sdn	security_domain	Required if you use LDAP authentication. Name of the security domain that the user belongs to. You can set a security domain with the -sdn option or the environment variable INFA_DEFAULT_SECURITY_DOMAIN. If you set a domain name with both methods, the -sdn option takes precedence. Security domain is case sensitive.
-ResilienceTimeout -re	timeout_period_in_seconds	Optional. Amount of time in seconds that infacmd attempts to establish or re-establish a connection to the domain. You can set the resilience timeout period with the -re option or the environment

Option	Argument	Description
		variable INFA_CLIENT_RESILIENCE_TIMEOUT. If you set a the resilience timeout period with both methods, the -re option takes precedence.
-SQLDataService -sqlds	sql_data_service	Required. Name of the SQL data service with the virtual table.
-Table -t	schema.table	Required. Name of the virtual table. Enter table in the following format: `<schema_name>.<table_name>`
-Column -c	column	Name of the column to update.
-GranteeUserName| GranteeGroupName -gun|-ggn	grantee_user_name| grantee_group_name	User or group name to deny permissions to.
-GranteeSecurityDomain -gsdn	grantee_security_domain	Required if you use LDAP authentication and you are granting user permissions. Name of the security domain that the user belongs to.
-DeniedPermissions -dp	denied_permissions	Required. Enter SQL_Select to restrict a user from including the column in a SELECT.

Authors

Lori Troy
Senior Technical Writer

Ellen Chandler
Principal Technical Writer

Lalitha Sundaramurthy
Senior Product Manager