SQL Sentry Role Based Security

Last Modified on 18 May 2021

Introduction

To provide a more secure environment and allow non system administrators to take advantage of SQL Sentry's features, roles are placed on the SQL Sentry database during its installation or upgrade. Users are placed in these roles to allow them access to the features they need, while restricting access to features that may be above and beyond their responsibility.

Setting up Role Based Security

Role Based Security is configured through T-SQL statements or by using SSMS to set up database roles in SQL Server. There are two role-based security models to choose from:

- **Allow All**: Create a user on the SQL Sentry database, and add them to the **allow_all** role. This provides full access to the SQL Sentry database. From here, add the user to any of the custom **deny_** roles to restrict that user's access to the different functions of SQL Sentry. Typically, there's a role to deny updating the specified information, and one to deny reading the information at all.
- **Least Privilege**: Create a user on the SQL Sentry database, and add them to the **allow_least_privilege** role. This provides a minimal level of access, with the culmination of all the explicit **deny_update** roles available. From here, add the user to any of the custom **allow_** roles to expand that user's access to the different functions of SQL Sentry.

Roles

Warning: The roles starting with **db_** are SQL Server default roles placed on every database. Using these roles in the SQL Sentry database may cause unpredictable behavior.

⛔**Unsupported:** The **allow_readonly** role has been deprecated. You should switch to the **least privilege** model if you are currently using **allow_readonly**.

Allow All Model

Use the **allow_all** model to create users with full access to all of SQL Sentry's features, then remove access to any desired role(s) by applying the appropriate **deny_update** role. The following roles are available:

Role	Description
allow_all	Provides full access to SQL Sentry's features. Place all non-sa users in this role, then add **deny roles** from this table to restrict access.
deny_actions_read	Denies the ability to view all **General**, **Failsafe**, **Audit**, and **Custom Condition** actions.
deny_actions_update	Denies the ability to make changes to any actions, but allows the viewing of those settings, making them **read-only**.
deny_appsettings_update	Denies any changes made under the SQL Sentry **Monitoring Service** > **Settings** node.
deny_contact_update	Denies the ability to update information for individual users, but allows viewing the information, making it **read-only**.
deny_contactgroup_update	Denies the ability to update group information, making it **read-only**.
deny_customconditions_update	Denies the ability to enable, disable, create, or edit advisory conditions.
deny_eventchain_read	Denies the ability to view **Event Chain** information.
deny_eventchain_update	Denies the ability to make changes to event chains.
deny_fragman_manual_analyze	Denies the ability to manually execute **analyze fragmentation now** through Indexes tab.
deny_fragman_manual_defrag	Denies the ability to manually execute **defragment now** through Indexes tab.
deny_quick_trace	Denies the ability to run a QuickTrace™.
deny_settings_connection_read	Denies the ability to view information under the **Settings** tab at the instance level for the specified instance type.

Role	Description
deny_settings_connection_update	Denies the ability to make changes under the **Settings** tab at the instance level for the specified instance type.
deny_settings_object_read	Denies the ability to view information under the **Settings** tab at the object level.
deny_settings_object_update	Denies the ability to make changes under the **Settings** tab at the object level.
deny_settings_source_read	Denies the ability to view source information from the **Settings** tab.
deny_settings_source_update	Denies the ability to make changes to **Source** information from the **Settings** tab.
deny_site_update	Denies changes made to **Site Configuration**.
deny_watch_connection	Denies the ability to watch or stop watching an instance.
deny_watch_object	Denies the ability to watch or stop watching an individual object.

Allow Least Privilege Model

Use the **allow_least_privilege** model to create users with basic access to SQL Sentry's features, then add access to any desired role(s) with the appropriate **allow_update** role. The following roles are available:

Role	Description
allow_least_privilege	Provides access to read and view SQL Sentry's features, and denies update permissions to any update action listed in this table. You can place any non-sa users in this role, and then add **allow roles** to allow access. For example, users that are assigned the *allow_least_privilege* role can create custom event views in the SQL Sentry Client, but can not make any changes to the monitoring service settings without being assigned to the appropriate allow_update role from this table (*allow_appsettings_update* in this case).
allow_actions_update	Allows the ability to make changes to any actions.
allow_appsettings_update	Allows any changes made under the **SQL Sentry Monitoring Service > Settings** node.
allow_contact_update	Allows the ability to update information for individual users.
allow_contactgroup_update	Allows the ability to update group information.
allow_customconditions_update	Allows the ability to enable, disable, create, or edit advisory conditions.

Role	Description
allow_eventchain_update	Allows the ability to make changes to event chains.
allow_fragman_manual_analyze	Allows the ability to manually execute **analyze fragmentation now** through Indexes tab.
allow_fragman_manual_defrag	Allows the ability to manually execute **defragment now** through Indexes tab.
allow_quick_trace	Allows the ability to run a QuickTrace™.
allow_settings_connection_update	Allows the ability to make changes under the **Settings** tab at the instance level for the specified instance type.
allow_settings_object_update	Allows the ability to make changes under the **Settings** tab at the object level.
allow_settings_source_update	Allows the ability to make changes to **Source** information from the **Settings** tab.
allow_site_update	Allows changes made to site configuration.
allow_watch_connection	Allows the ability to watch or stop watching an instance.
allow_watch_object	Allows the ability to watch or stop watching an individual object.

Role Based Example

Scenario

You have a junior DBA that needs to use SQL Sentry's **Calendar** view to check for any failures or long running jobs overnight, but you don't want them to make changes to any of SQL Sentry's settings.

Solution

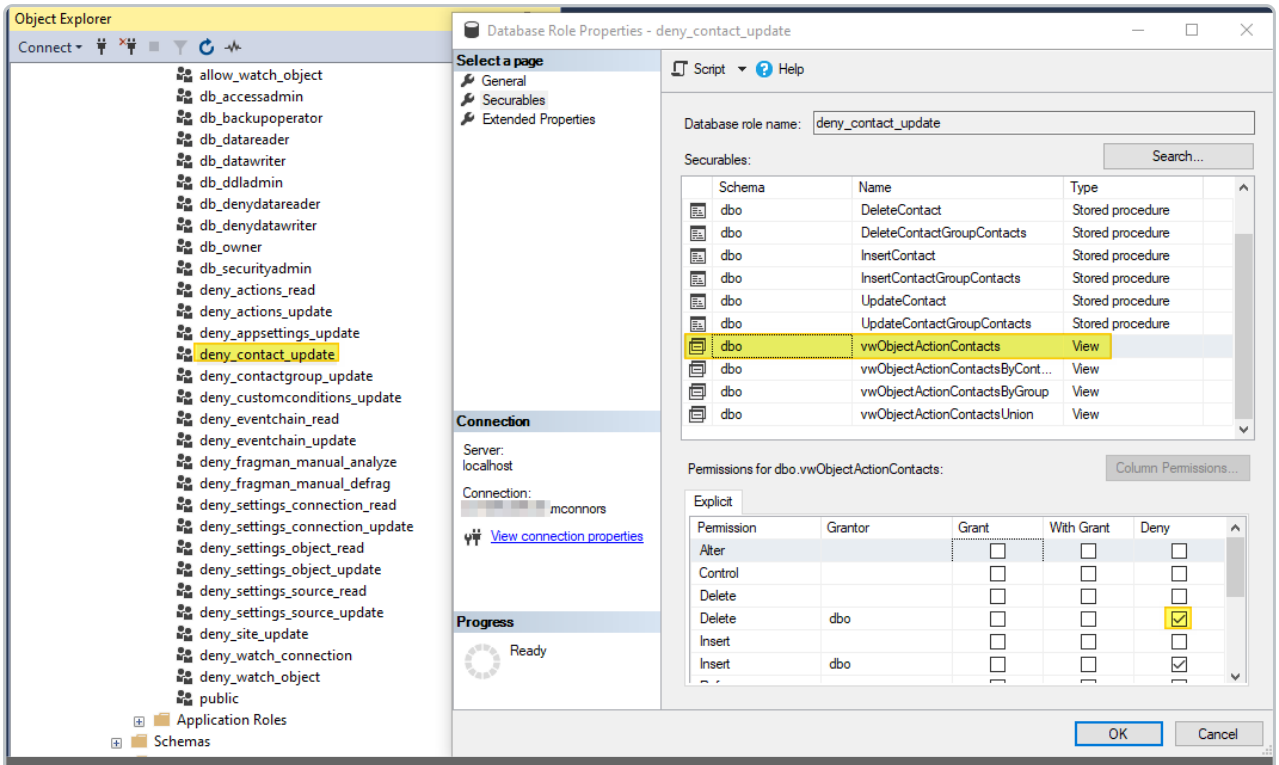
Add their login as a **User** on the SQL Sentry database. Place that user in the **allow_all** role. This ensures the user has access to all the information they need while being explicitly denied any information specified in the additional roles assigned to them. Finally, for this example, you may want to add this user to all **deny_ roles** except the ones ending in **_read**. This denies changes to any settings along with the ability to **Watch** or **Stop Watching** an instance or object.

It's important to remember that logins using **SQL Server Authentication** must be specified in the SQL Sentry client instance information. To specify SQL Server Authentication, select **File** > **Connect to Installation.** Uncheck the box marked **Integrated Windows Authentication**, and then enter the user's login and password. Restart the SQL Sentry client to apply the settings. These new settings remain in effect on this SQL Sentry client until explicitly changed.

Viewing Role Details

To view the full details behind any role:

1. Open **SSMS**
2. Navigate to your **SQL Server instance > Databases > SQL Sentry (or SentryOne) > Security > Roles > Database Roles**
3. Select a **role** and view **properties**
4. Select **Securables**



Example of looking at role details in SQL Server Management Studio