ServiceNow Application Developer

Securing Applications Against Unauthorized Users > Exercise: Access Controls

In this exercise, you will create and debug Access Controls to allow *NeedIt* users to see only the *NeedIt* records where they are the *Requested for*.

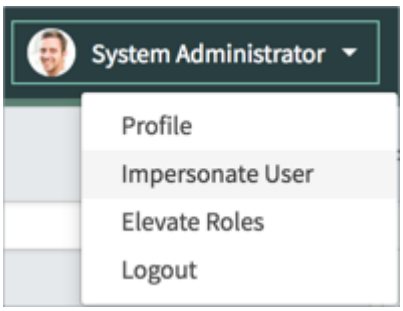
Preparation

1. In Studio, use the Application Explorer to open **Data Model > Table > NeedIt**.

2. Copy the value in the *Name* field to the clipboard.

3. Impersonate *Beth Anglin*.

   a. In the main ServiceNow browser window (not Studio), open the **User menu** and select the **Impersonate User** menu item.

   b. In the *Impersonate User* dialog, type **Beth** in the *Search for user* field.

   c. Select **beth.anglin**.

4. In the Application Navigator, type **NeedIt** in the *Filter navigator* field. Note that Beth can see only two modules: *Create New* and *My NeedIt Requests*. Beth cannot see the *Open* or *All* modules.

5. In the Application Navigator, paste the value from the clipboard and type **.list** at the end of the name. It will look something like this:

6. Press the **<return>** key on the keyboard. What happens?

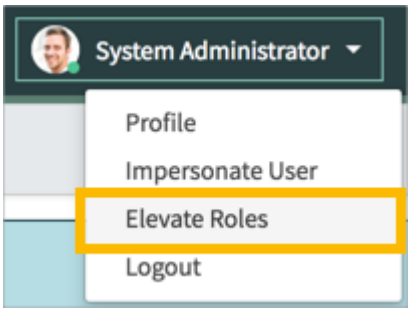
Typing a table name followed by *.list* in the Application Navigator *Filter navigator* field opens the list of table records. In the previous exercise, you removed Beth's access to the *All* and *Open* modules. If Beth knows how to use *.list*, she can still see the records you did not intend for her to access. As you can see, module permissions alone cannot protect a table's records. Use Access Controls to protect table records.

Examine Existing Access Controls

1. Impersonate the **admin** user.

2. Elevate Security privileges.

   a. Open the **User menu** in the banner and select the **Elevate Roles** menu item.



   b. In the *Elevate Roles* dialog select **security_admin**.

   c. Click the **OK** button.

   d. If Studio was already open, you may need to reload Studio using the browser's reload button for Studio to detect the elevated privileges.

3. If the *NeedIt* application is not open in Studio from the last exercise, open it now.

   a. In the main ServiceNow browser window use the Application Navigator to open **System Applications > Studio**.

   b. In the *Select Application* dialog, click the **NeedIt** application.

4. In the Application Explorer, locate **Access Controls > Access Control**.

5. Open the *create* Access Control and examine the configuration. Note the table, field (if any), description, and the role.

6. Open the *read*, *write*, and *delete* Access Controls and examine the configurations.

▶ **QUESTION**: The descriptions for the Access Controls all say **Default access control**. When and how were the default Access Controls created?

Modify a Default Access Control

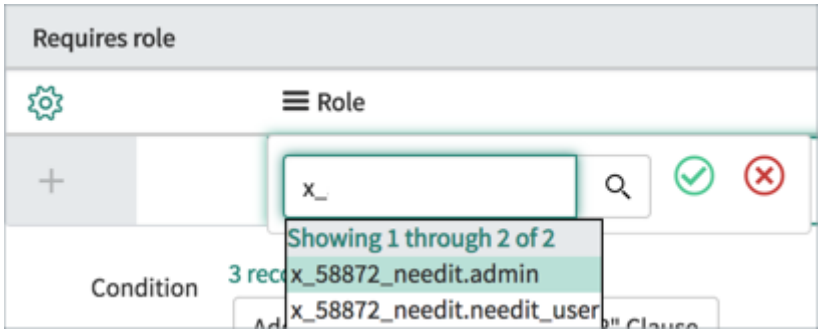
In this part of the exercise, you will modify the default read Access Control to grant full read access only to the application's admin role.

1. In the Application Explorer, open **Access Controls > Access Control > x_<app scope>_needit_needit (read)**.

2. Scroll to the *Requires role* list and click the **Mark for deletion** button (❌) for the *x_<app scope>_needit.needit_user* role.

3. Double-click **Insert a new row...**.

4. Type **x_** in the search field and select the **x_<app scope>_needit.admin** role from the list.



5. Click the **Save** button ( ).

6. Click the **Update** button to save the changes.

Create an Access Control

In this part of the exercise, you will create an Access Control to allow users with the *x_<app scope>_needit.needit_user* role to view records where they are the *Requested for*.

1. Create an Access Control.

   a. In Studio, click the **Create Application File** link.

   b. In the *Filter...* field enter the text **Access** OR select **Access Control** from the categories in the left hand pane.

   c. Select **Access Control** in the middle pane as the file type, then click the **Create** button.

2. Configure the Access Control:

   Type: **record**

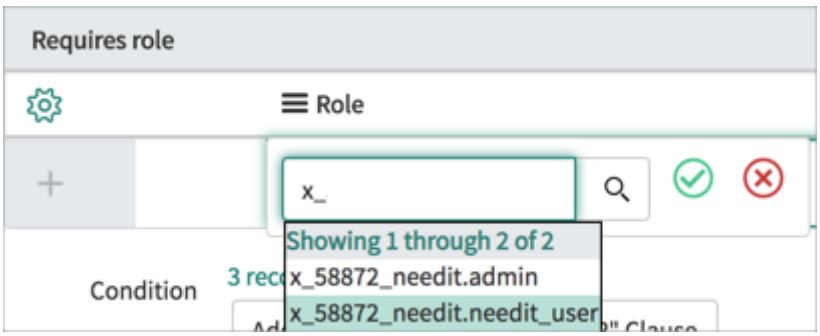
   Operation: **read**

   Name: **[NeedIt] [--None--]**

3. Add the *x_<app scope>_needit.needit_user* role to the Access Control.

   a. Scroll to the *Requires role* list and double-click **Insert a new row...**.

   b. Type **x_** in the search field and select the **x_<app scope>_needit.needit_user** role from the list.



   c. Click the **Save** button ( ).

4. Add a condition requiring the currently logged in user to be the *Requested for* value.

   a. Scroll to the *Condition* configuration.

   b. Configure the condition: **[Requested for] [is (dynamic)] [Me]**

5. Click the **Submit** button.

6. Examine the *Access Control Configuration Watcher* to see if other Access Controls are affected.

7. Click the **Continue** button.

Testing

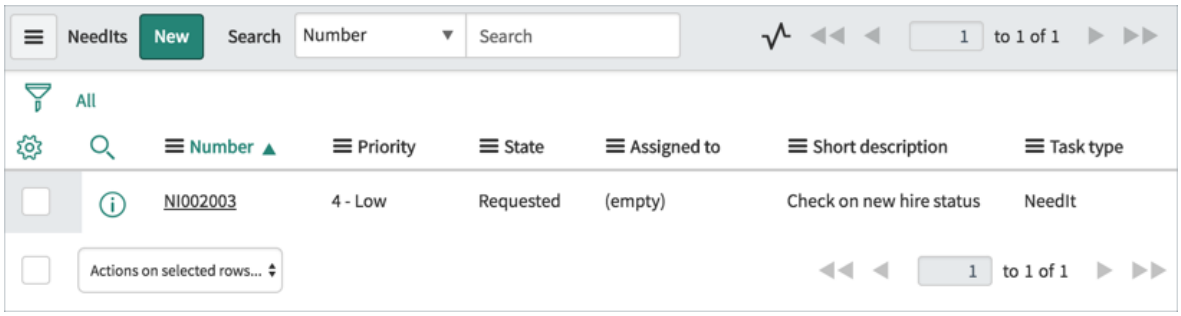
1. In Studio, switch to the *NeedIt* table tab. If the *NeedIt* table tab is not still open, use the Application Explorer to open **Data Model > Table > NeedIt**.

2. Copy the value in the *Name* field to the clipboard.

3. Impersonate *Beth Anglin*.

4. In the Application Navigator, paste the value from the clipboard and type **.list** at the end of the name.

5. Press the **<return>** key on the keyboard. Only the record where Beth is the *Requested for* should display.



6. Open a record from the list and verify that Beth is the *Requested for*. If not, debug and re-test.

7. Impersonate *Fred Luddy*.

8. In the Application Navigator *Filter navigator* field, paste the value from the clipboard followed by **.list**. Press the **<return>** key.

9. Fred should be able to see all the *NeedIt* request records without constraint. If not, debug and re-test.

10. Impersonate the *System Administrator*.