Lab Guide

Access Control Rules:

Securing Your Instance

Arnez Nisperos & Jason Lewko

Default Login / Password:

admin / Knowledge16

itil / Knowledge16

employee / Knowledge16

This
Page
Intentionally
Left
Blank

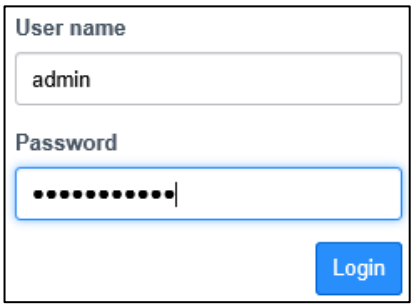
Lab Goal

Lab 1.0
Work with Roles

An instance uses access control rules, also called access control lists (ACLs), to control what data users can access and how they can access it. ACL rules require users to pass a set of requirements in order to gain access to particular data.

Log on to Your Training Instance

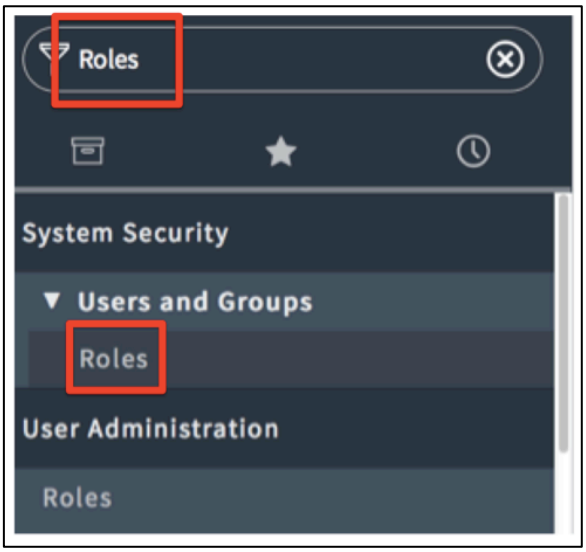
1. Navigate to the unique instance URL provided to you.
2. Log in using the **admin** credentials provided.



©2016 ServiceNow, Inc. All rights reserved.

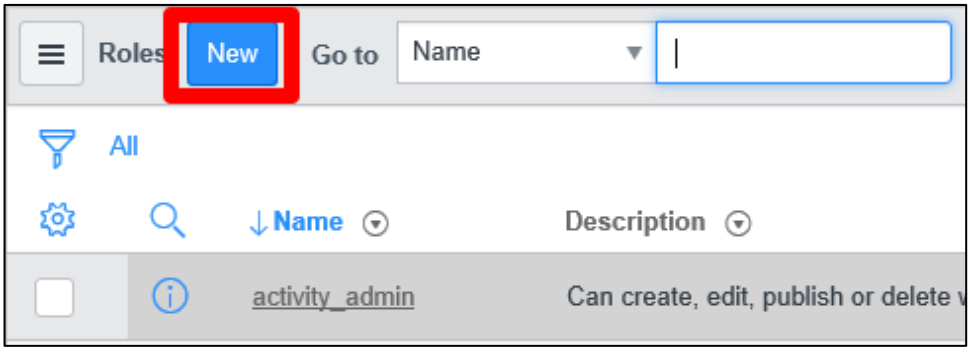
Create a New Role

1. Navigate to **System Security > Users and Groups > Roles**.



Note: In subsequent labs, this step is abbreviated to **Application Name** > **Module Name**.

2. Click **New**.



3. Name the role, **Role1**.



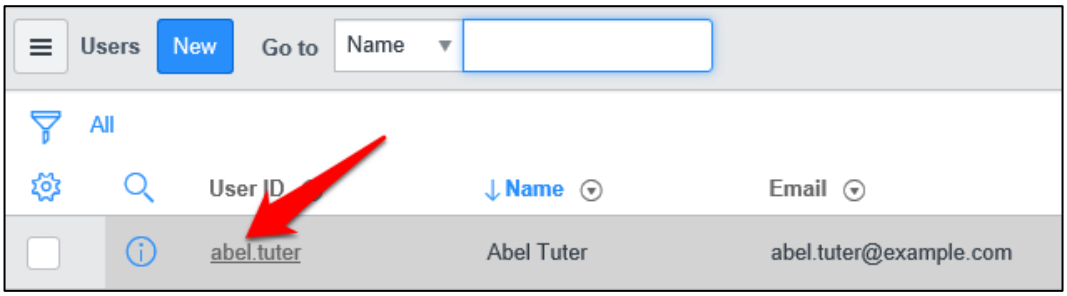
©2016 ServiceNow, Inc. All rights reserved.

4. Click **Submit**.

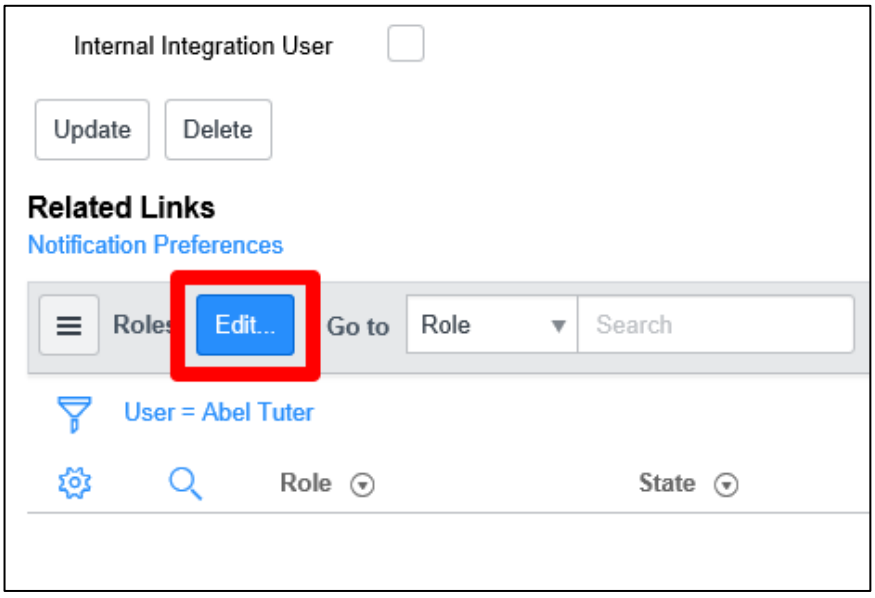
Apply the Role to a User

1. Navigate to **User Administration > Users**.
2. Click on **abel.tuter**.

   If Abel Tuter is not at the top of the list, sort the list of users by name by clicking on the **Name** column header.



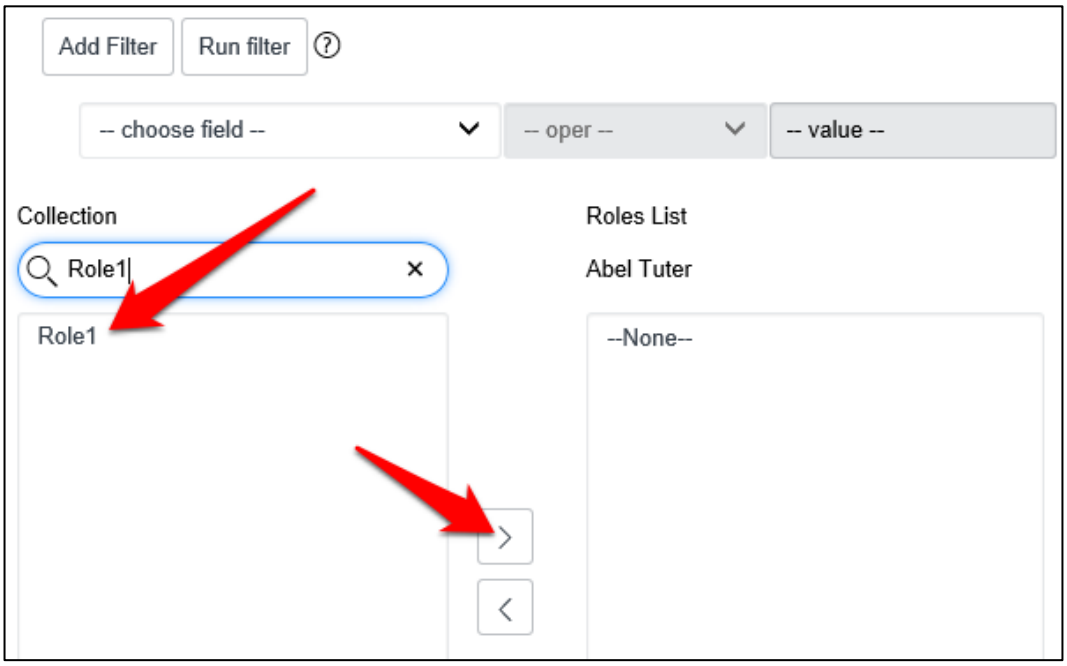
3. In the Roles related list, click the **Edit** button.



4. Enter **Role1** in the **Search** field.

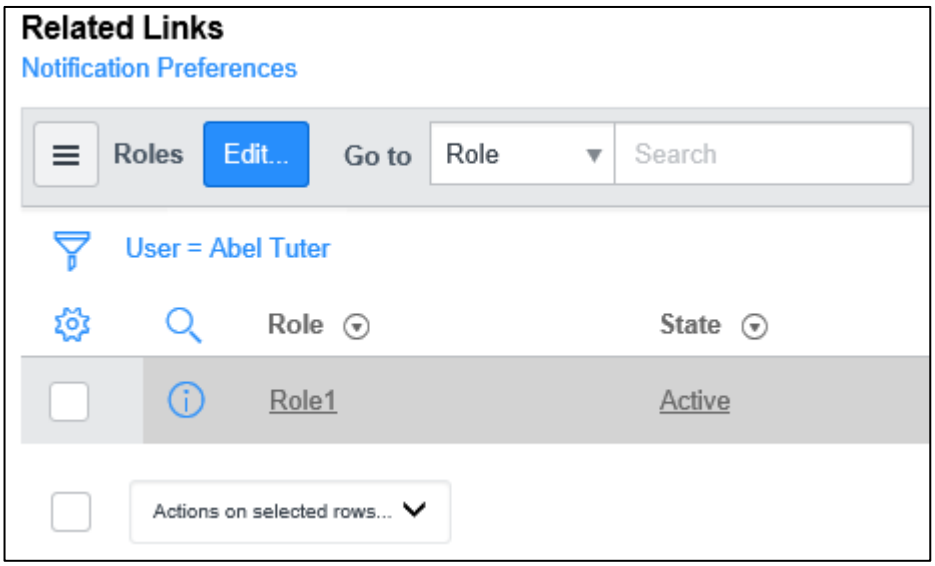
©2016 ServiceNow, Inc. All rights reserved.

5. Select **Role1** in the list, and click the **Add** button to move it into the Roles List collection.



6. Click **Save**.

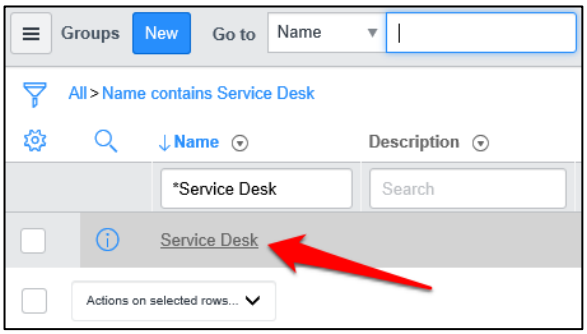
7. Verify that Abel has the **Role1** role in the Roles related list.



©2016 ServiceNow, Inc. All rights reserved.

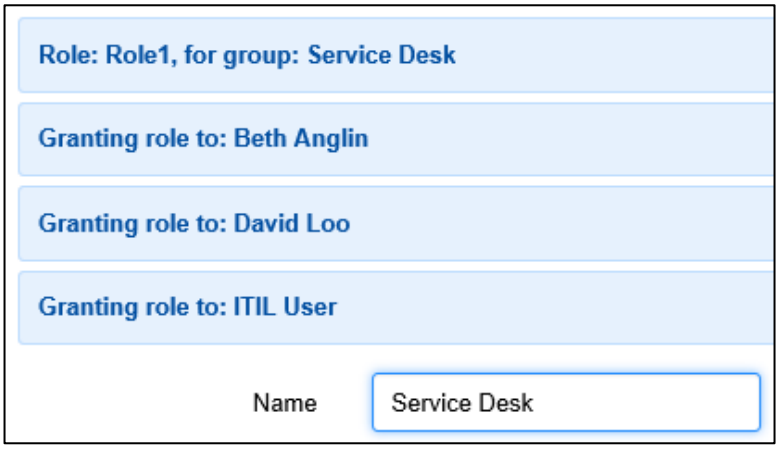
Apply the Role to a Group

1. Navigate to **User Administration > Groups**.
2. Open the **Service Desk** record.



3. In the Roles related list, click **Edit**.
4. Type **Role1** in the **Search** field.
5. Select **Role1** in the list, and click the **Add** button to move it into the Roles List collection.
6. Click **Save**.

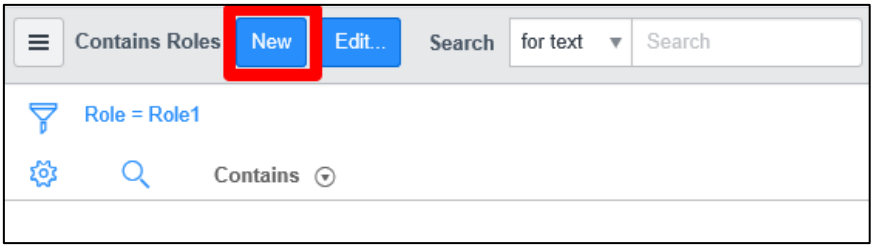
   Note: The new roles are automatically applied to members. The top of the browser displays messages indicating the application of new roles to members.



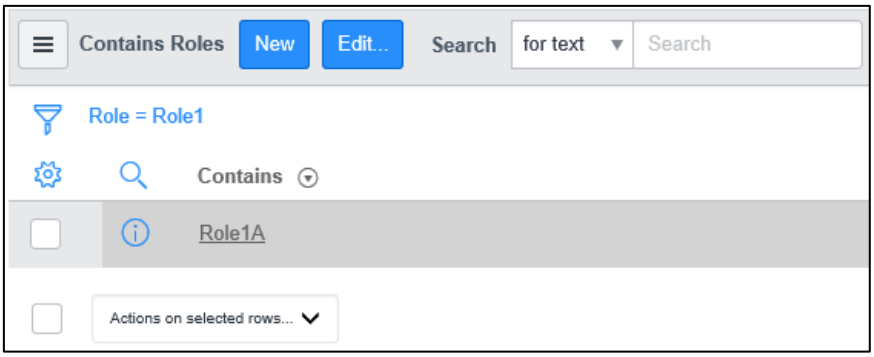
©2016 ServiceNow, Inc. All rights reserved.

Create a Nested Role

1. Navigate to **User Administration > Roles**.
2. Open the **Role1** record.
3. In the Contains Roles related list, click **New**.



4. In the resulting form, name the role **Role1A**.
5. Click **Submit**.



Lab Success Verification

1. Navigate to **User Administration > Users**.
2. Open the **abel.tuter** record.

©2016 ServiceNow, Inc. All rights reserved.

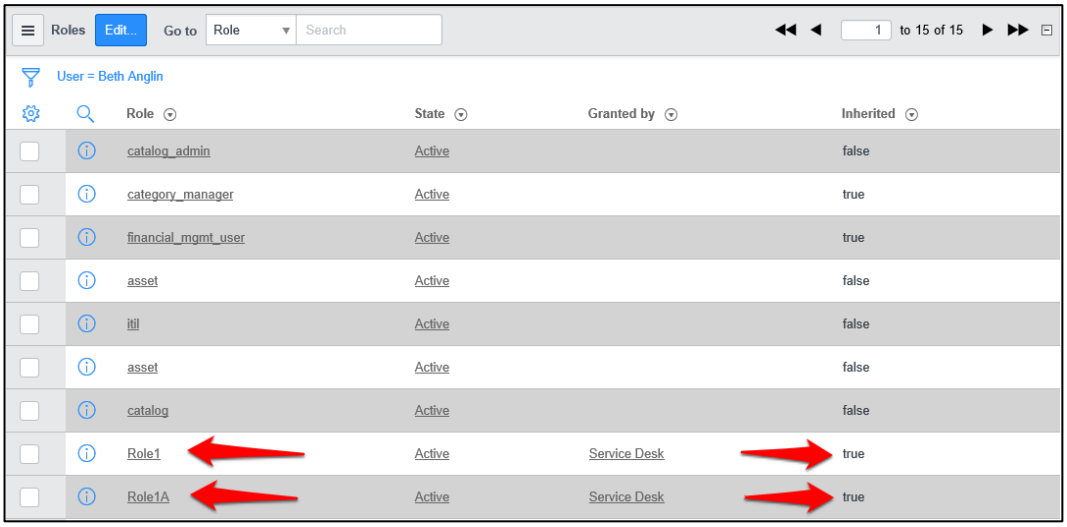
3. Abel should have two roles now in the Roles related list. **Role1A** should display **true** in the Inherited column.



4. Navigate to **User Administration > Users**.

5. Locate and open the **beth.anglin** record.

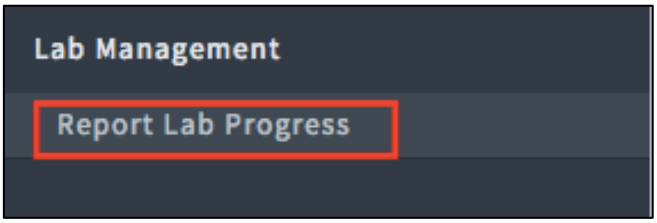
6. Beth should also have those roles, in addition to others. Both roles should display **true** in the Inherited column, granted by her membership in the Service Desk group.



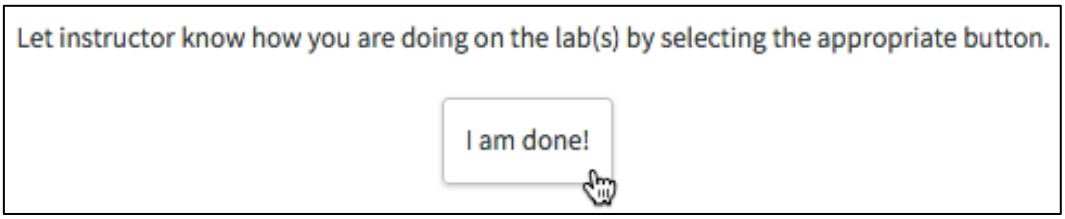
©2016 ServiceNow, Inc. All rights reserved.

Progress Report

1. Navigate to **Lab Management> Report Lab Progress**.



2. Click **I am done!**



©2016 ServiceNow, Inc. All rights reserved.

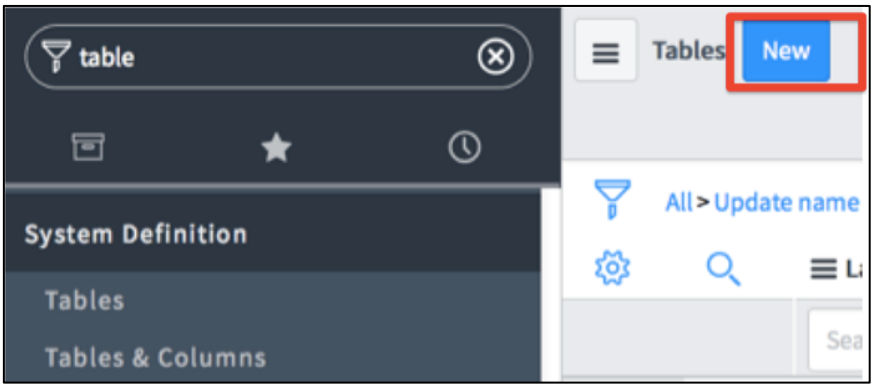
Lab Goal

This lab explains how to create a workspace for using Access Control Rules in ServiceNow. The first part of the lab explains how to create a new table to use in the workspace. The second part implements role-based visibility of an application menu item.

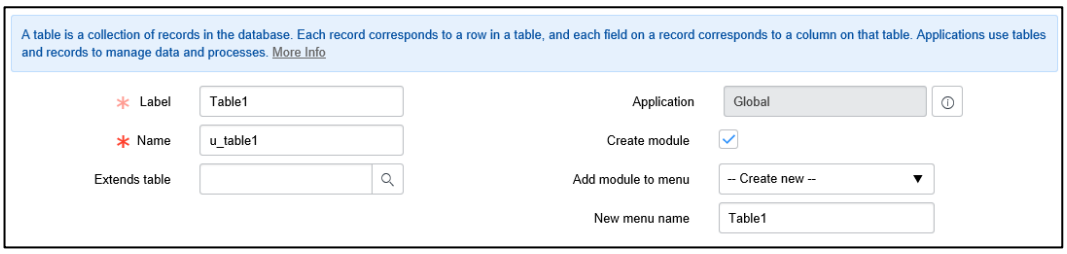
Lab 2.0
Set Up

Create a New Table

1. Navigate to **System Definition > Tables & Columns**.
2. Click **Create Table**.



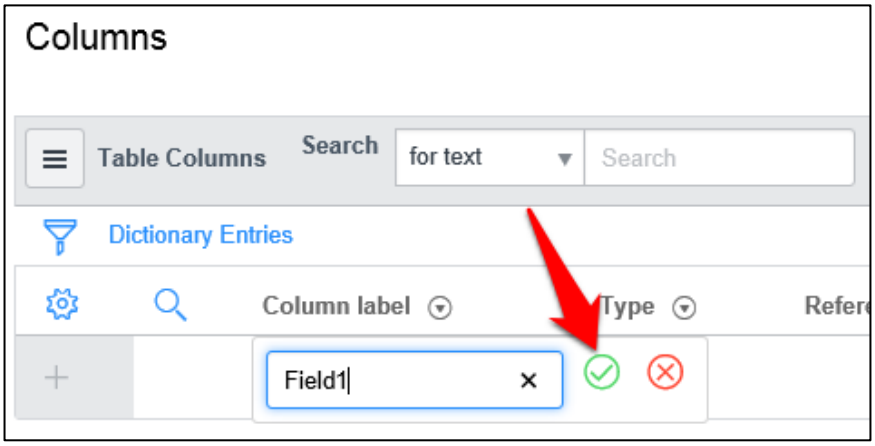
3. Enter **Table1** in the **Label** field. After the cursor leaves this field, the **Name** field populates automatically.



4. In the **Table Columns** list editor, double-click on ***Insert a new row…***

©2016 ServiceNow, Inc. All rights reserved.

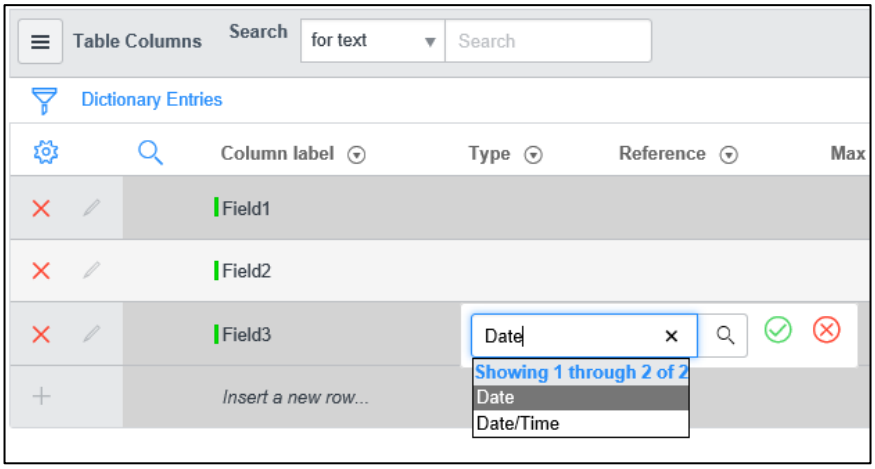
5. Enter **Field1** in **Column label**. Click the **green check mark** button to insert the field name into the table.



6. Add **Field2**, following the directions in the step above.

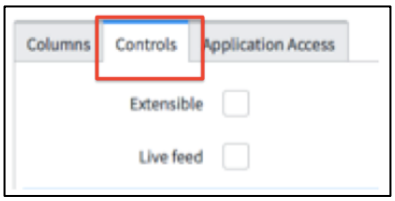
7. Add a final column, **Field3**.

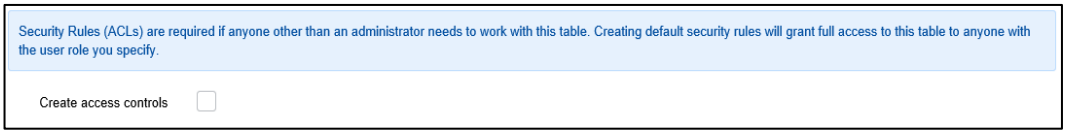
8. Double-click the **Type** column on the **Field3** row, and select **Date**. You can type **Date** and select it from the type-ahead list, or click on the **magnifying glass** to see a list of all possible types.



©2016 ServiceNow, Inc. All rights reserved.

9. By default, ServiceNow offers to create access control automatically. Since this is an exercise to understand how access control building works, scroll down and clear the **Create access controls** check box in the Controls section. Clearing that check box hides the **User role** field.





10. Click **Submit**.

11. After a few moments the navigation pane refreshes and **Table1s** appears as a menu option.



Create a Record

1. Navigate to **Table1 > Table1s**.

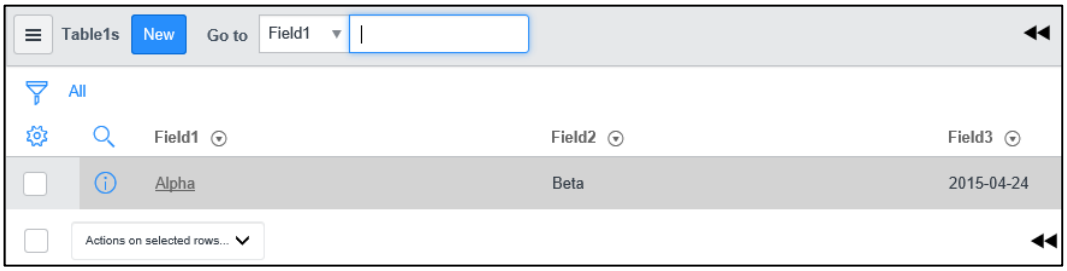
©2016 ServiceNow, Inc. All rights reserved.

2. Click the **New** button to create a new record and enter the field information listed below.

   Field1: **Alpha**
   Field2: **Beta**
   Field3: Click on the **calendar** decoration and select **Tomorrow's date**.

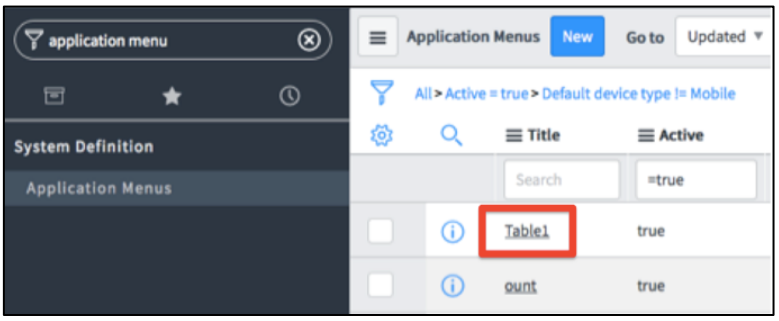


3. Click **Submit**. The screen returns to the Table1 list.



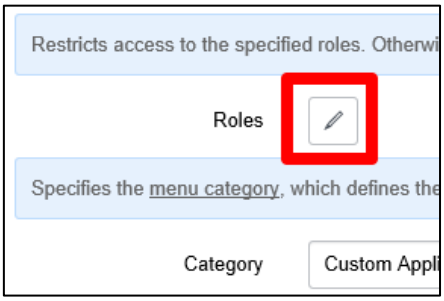
Require a Role to See the New Module

1. Navigate to **Applications Menus**. Select **Table1**.

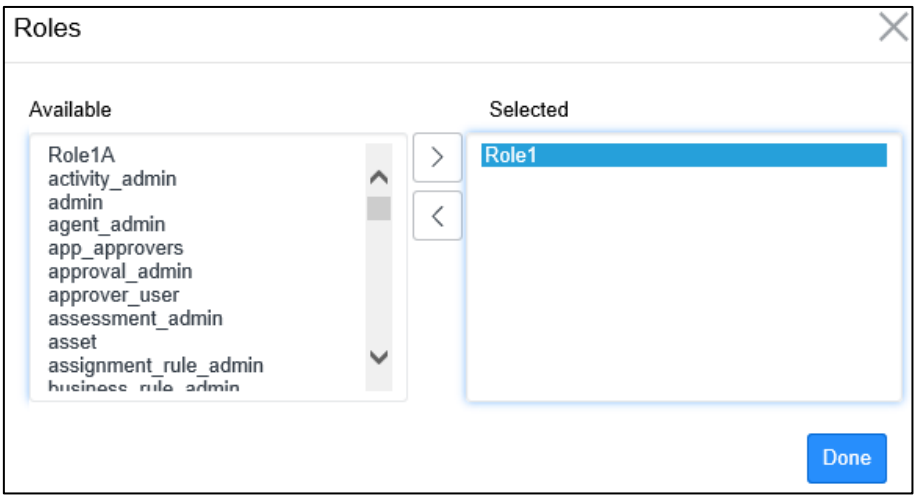


©2016 ServiceNow, Inc. All rights reserved.

2. In the **Roles** field, click on the **edit** (**pen**) button.



3. Select **Role1** from the list and use the **right arrow** (**>**) to move it to the **Selected** column.

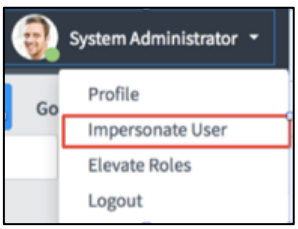


4. Click **Done**.

5. Click **Update**.

Lab Success Verification

1. Click the **Impersonate User** in the user name dropdown above the navigation pane.

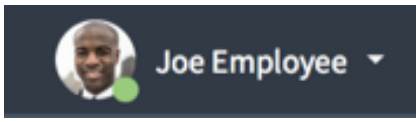


©2016 ServiceNow, Inc. All rights reserved.

2. Impersonate **Joe Employee** by typing the name in the reference field and selecting **Joe Employee**. Click **OK**.



3. Joe Employee should not see the Table1 menu.





4. To verify Joe Employee has no access, enter **u_table1.list** (do not finish with the Enter key) in the **Type filter text** field at the top of the navigation pane. You should see a security constraints message.



5. Impersonate **Abel Tuter**. Abel should see the **Table1** menu.

©2016 ServiceNow, Inc. All rights reserved.

6. Navigate to **Table1 > Table1s** to see the record list. Abel currently has the same view of Table1 as Joe Employee.



7. Return to the System Administrator user by using the Impersonation function.

   Note: In this, and in subsequent labs, instead of using Impersonation, you can use separate sessions or another browser and log in as Abel Tuter. To do this, set a password for Abel that you can use for the lab in **User Administration > Users**. Then, either open a different browser to your lab instance, or right-click on the browser's shortcut and start a private session. On Chrome, this is called "New incognito window." On Firefox, it is called "New private window." On Internet Explorer, it is called "Start InPrivate Browsing." In the new browser window, log on as **abel.tuter** with the password you specified.

©2016 ServiceNow, Inc. All rights reserved.

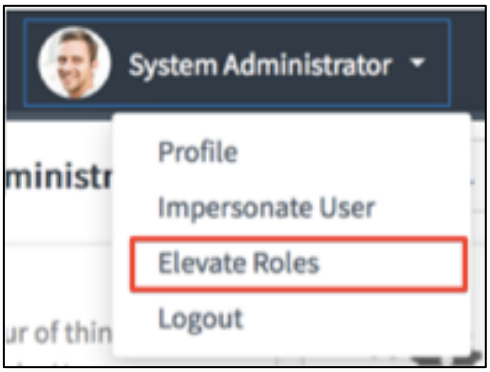
Lab Goal

Abel cannot see records in the new table due to the default "deny System Property and Access Control Rules" that exist on ServiceNow instances. This lab focuses on how to create rules to address data visibility.

Lab 3.0
Table Access Control Rule

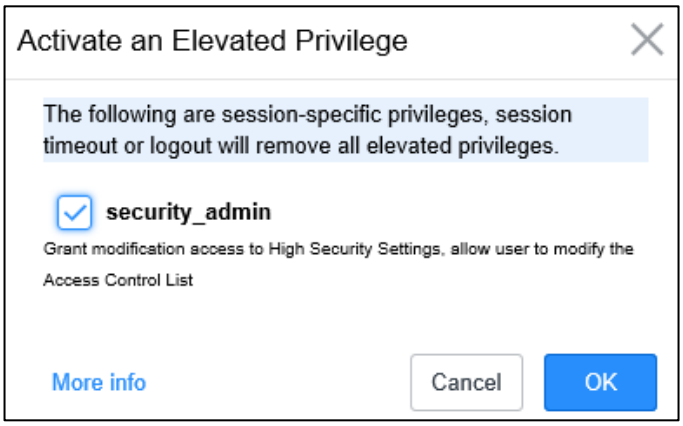
Elevate Privileges

1. Click the **user name** dropdown. Click on **Elevate Roles**. If you are currently impersonating a different user, use the Impersonate User to become **System Administrator**.



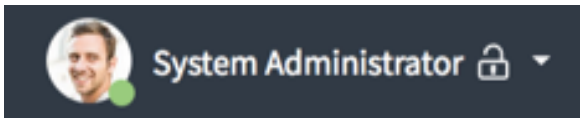
©2016 ServiceNow, Inc. All rights reserved.

2. Check the option box next to **security_admin**, and click **OK**.



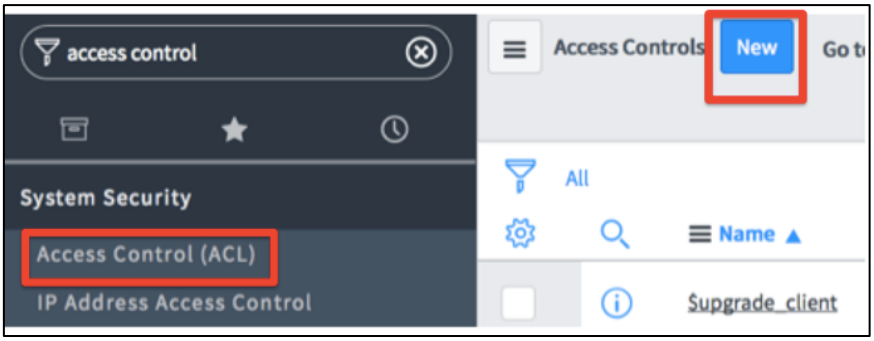
Note: Any time you Impersonate another user, the session drops elevated privileges. Elevate the privileges again to work with security rules.

3. Your session has elevated privileges now. Note the unlocked icon next to your user name.



Create an Access Control Rule

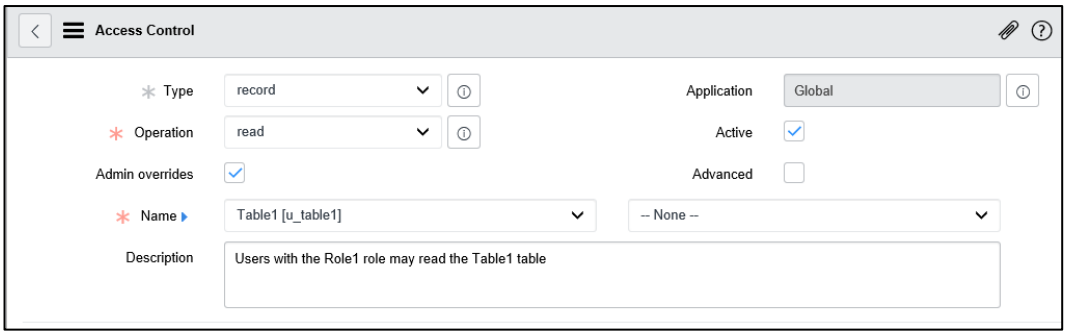
1. Navigate to **System Security > Access Control (ACL)**.
2. Click **New**.



©2016 ServiceNow, Inc. All rights reserved.

3. Enter the following information into the fields:

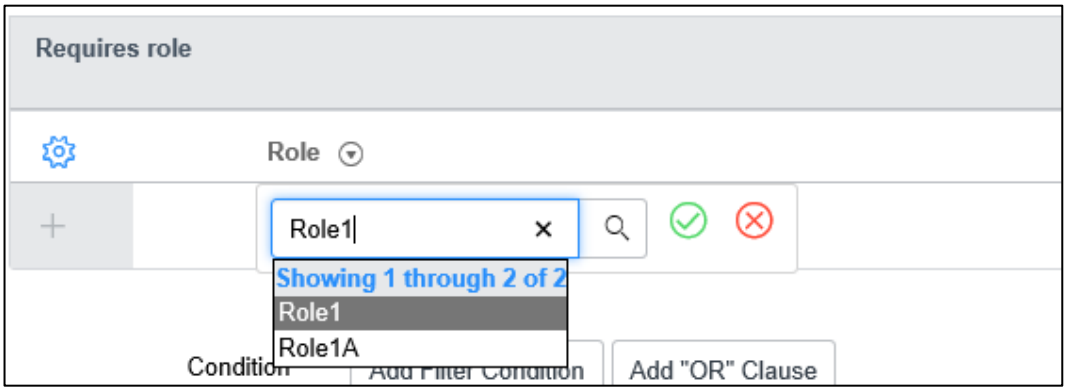
   Operation: **read**
   Name: **Table1 [u_table1], -- None --**
   Description: **Users with the Role1 role may read the Table1 table**



4. In the Definition section, in the **Requires role** list editor, double-click **Insert a new row...**.

5. Add **Role1**, and click the **green** check mark button.

6. Click **Submit**.

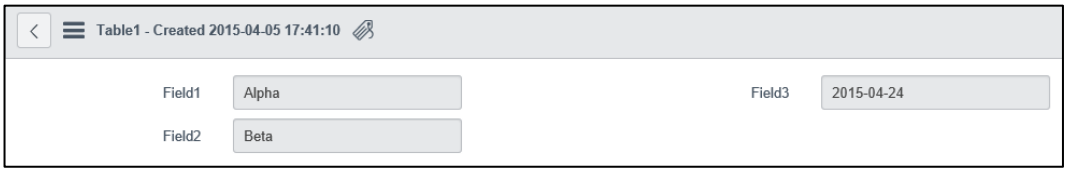


Note: Access control rules evaluate the **Conditions**, **Script**, and **Requires role** related list to determine if the rule allows access. If any of those individual pieces do not have data, they automatically evaluate to **true**. An access control rule with no **Condition**, **Script**, or **Required role** allows anyone to use the operation that the rule would be securing.

©2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. As System Administrator, navigate to **Table1 > Table1s**. Note one created record.
2. As Abel Tuter, navigate to **Table1 > Table1s**. Note the same record. The user Abel, however, cannot edit the record fields.
3. Click on the **record** to open it. Note the grayed out fields, indicating that those fields are read-only. Note that Abel also has no **Submit** button.



©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

Abel can read records on the new table now. This lab explains how to create more rules that further constrain his view.

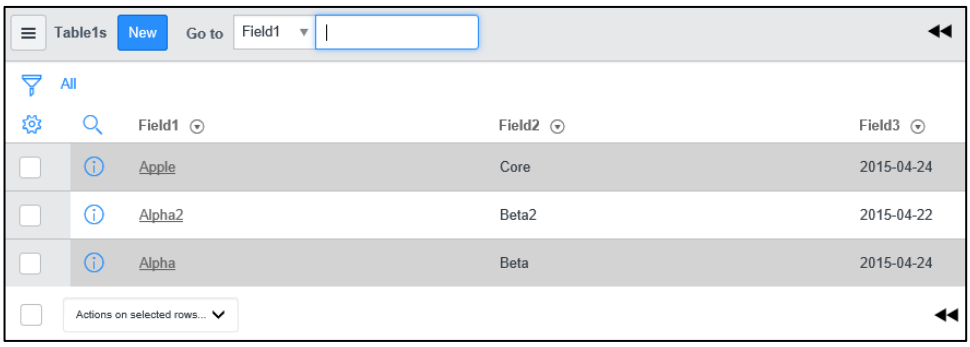
Lab 4.0
Field Access Control Rules

Create a New Record in the Table

1. As System Administrator, navigate to **Table1 > Table1s**.
2. Create and submit two new records using the field data listed below.

   Field1: **Alpha2**
   Field2: **Beta2**
   Field3: **Yesterday's date**

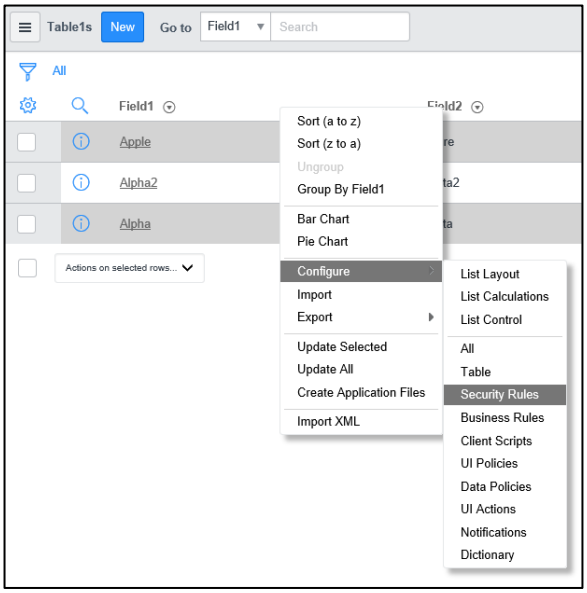
   Field1: **Apple**
   Field2: **Core**
   Field3: **Tomorrow's date**



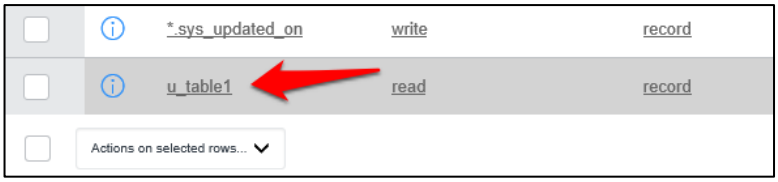
©2016 ServiceNow, Inc. All rights reserved.

Modify the Table Read Rule

1. As an Elevated admin, right-click in the column headers for the list view for **Table1** and select **Configure > Security Rules**. This gives a filtered view of the access control rules in the ServiceNow instance, showing only the ones that apply to **Table1**.



2. Click the **u_table1** read rule created in the previous lab. It is at the bottom of the page.



3. In the **Definition** section, click on the **Condition** selector indicating choose field, and select **Field3**. For the operator, select **after**, and in the value field, select **Yesterday**.



4. Click **Update**.

©2016 ServiceNow, Inc. All rights reserved.

5. You will see the Execution Plan. Click **Continue**.



©2016 ServiceNow, Inc. All rights reserved.

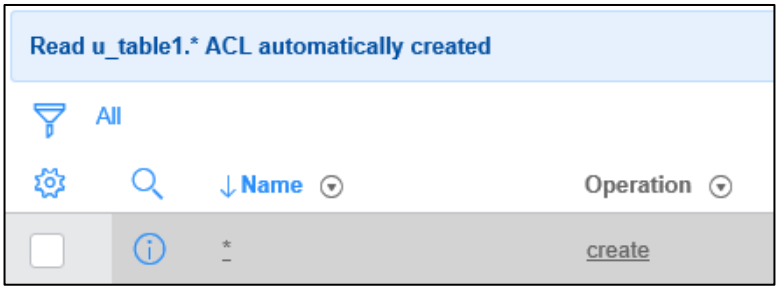
Create a Field Access Control Rule

1. Create and submit a new rule, using the following condition values.

   Operation: **read**
   Name: **Table1 [u_table1], Field2**
   Condition: **Field2, contains, Beta**

   The condition value is case-sensitive. Be sure it matches the values used in the records.

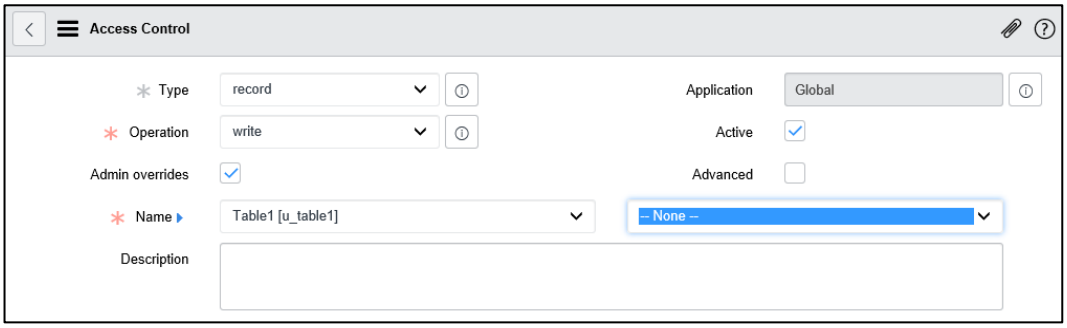
   Note: After you submit the rule, notice the newly automatically created **u_table1.*** read rule, which automatically has **Role1** in the **Requires role** related list. This appears in a display message at the top of the page after the new rule is submitted. The new rule you created will be on the next page of the list.



Create a Table Rule for the Write Operation

1. Create a new rule, using the following condition values.

   Operation: **write**
   Name: **Table1 [u_table1], -- None --**



2. In the **Requires role list editor,** double-click **Insert a new row...**

©2016 ServiceNow, Inc. All rights reserved.

3. Add **Role1**.

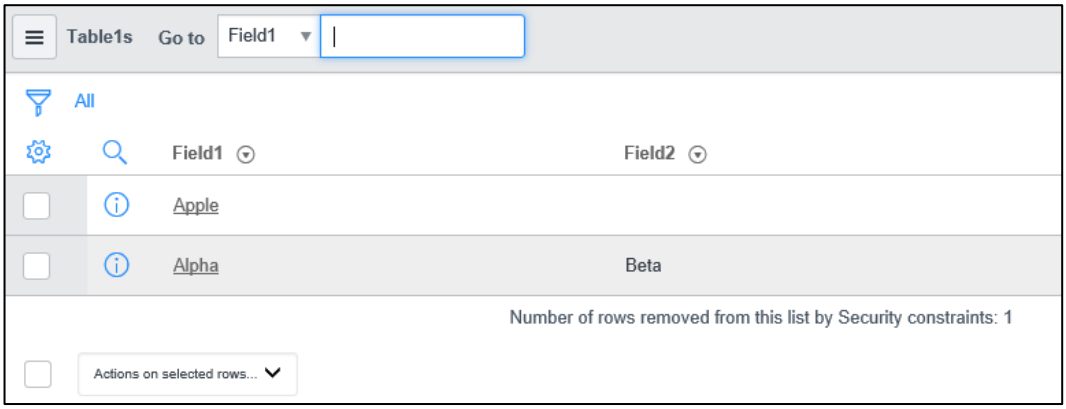


4. Click **Submit**.

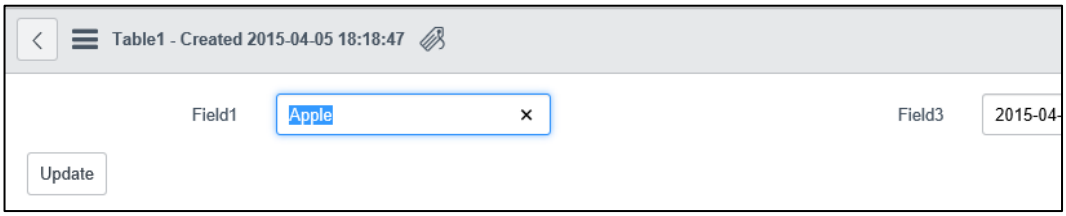
Lab Success Verification

1. As System Administrator, Navigate to **Table1 > Table1s**. Note three created records, and all fields are visible.

2. As Abel Tuter, Navigate to **Table1 > Table1s**. Note two records, and the return of the "**Number of rows removed from this list by security constraints**" message. One of the records has a blank **Field2**.



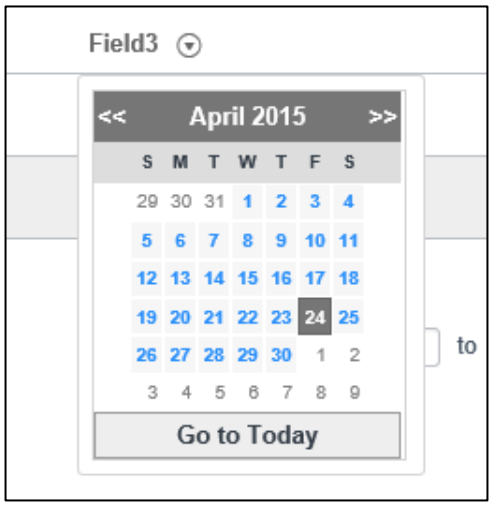
3. Click to open the **Apple** record. Note that the form hides **Field2**. Also, Abel can edit the record fields.



©2016 ServiceNow, Inc. All rights reserved.

4. Return to the list view by clicking the back arrow icon at the top of the record.

5. Double-click in the **Field3** column on one of the visible records. Note that Abel can edit those fields in the record form as well as in the list view.



©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

ServiceNow provides list editing and view capabilities as part of the platform. These capabilities can be secured with list-specific operations. After applying an ACR, constraint messages appear. This lab explains a way to clean up the view, using **Before Query** rules. These are not precisely security rules, but they function in a similar manner.

Lab 5.0
Lists and Query Rules

Create a List Edit Rule

1. Abel is able to change the contents of the records. You do not wish for anyone to be able to modify the third field (the date field) from the list view. As an Elevated Admin, Navigate to **Table1 > Table1s**.

2. Click to open a record.

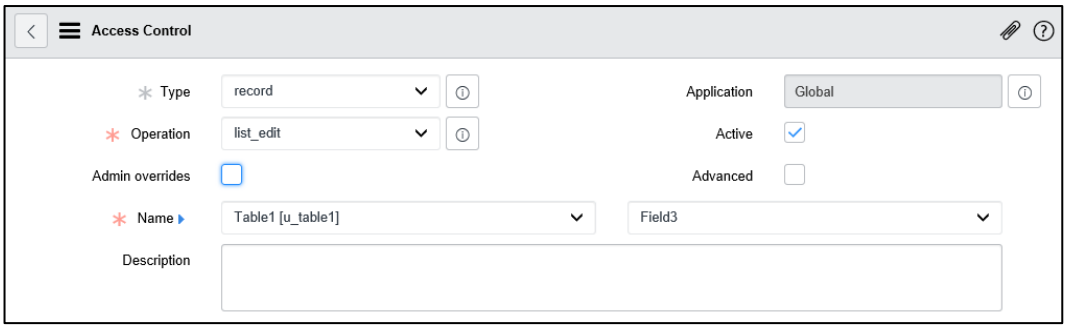
3. Right-click the **Field3** label, and select **Show Security Rules**. Note that this list is filtered even further, and shows only the access control rules for this specific field.



©2016 ServiceNow, Inc. All rights reserved.

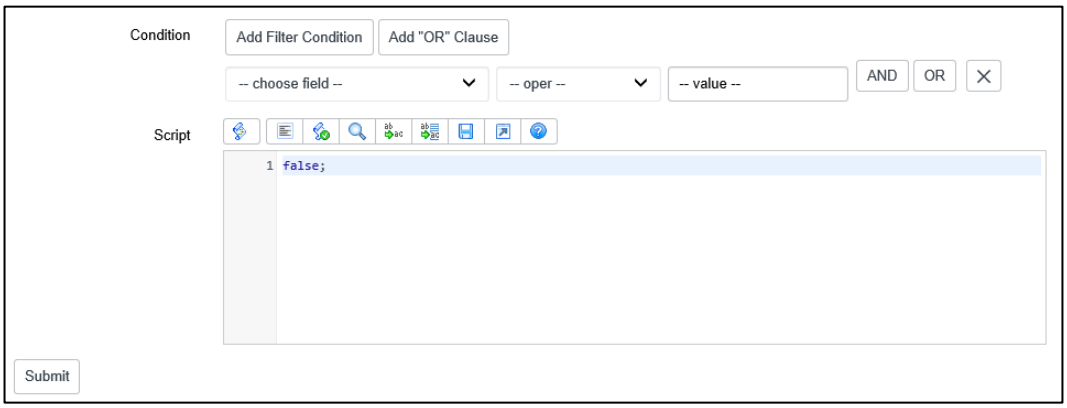
4. Create a new rule using the field and condition information below:

   Operation: **list_edit**
   Name: **Table1 [u_table1]**, **Field3**
   Admin overrides: **uncheck**



5. Check the **Advanced** box. Doing so exposes the Script field in the Definition section.

6. Enter **false;** in the Script field.

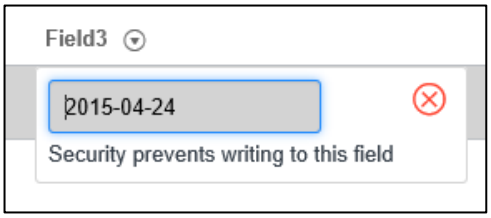


7. Click **Submit**.

©2016 ServiceNow, Inc. All rights reserved.

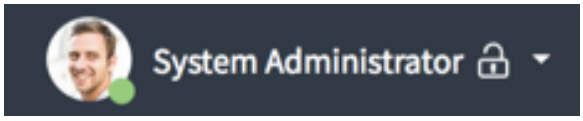
Verify the list_edit Rule

1. Navigate to **Table1 > Table1s**.
2. Double-click **Field3** on any of the records. The **Security prevents writing to this field** message appears.

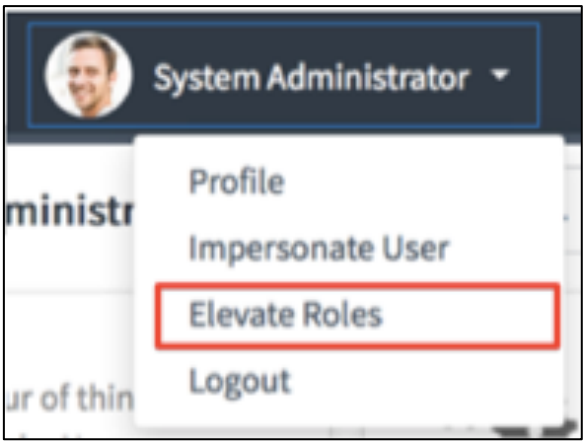


Remove Field2 from the List Layout

1. Remove your credential elevation if you are still **Elevated**. If the lock icon next to the user name is open, you are still Elevated.

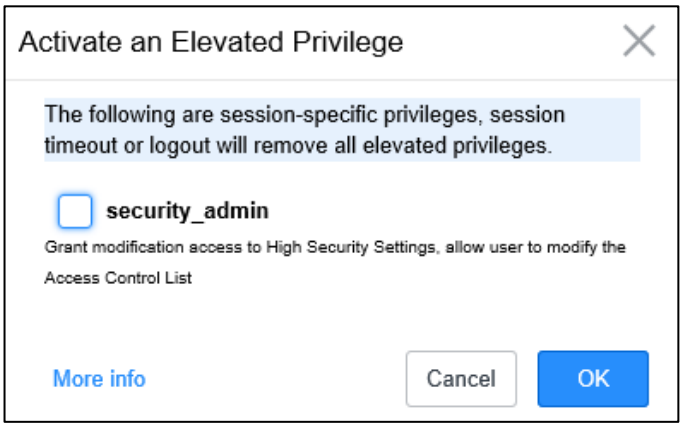


2. Click the **down arrow** and click on **Elevate Roles**.



©2016 ServiceNow, Inc. All rights reserved.

3. Deselect **security_admin**.

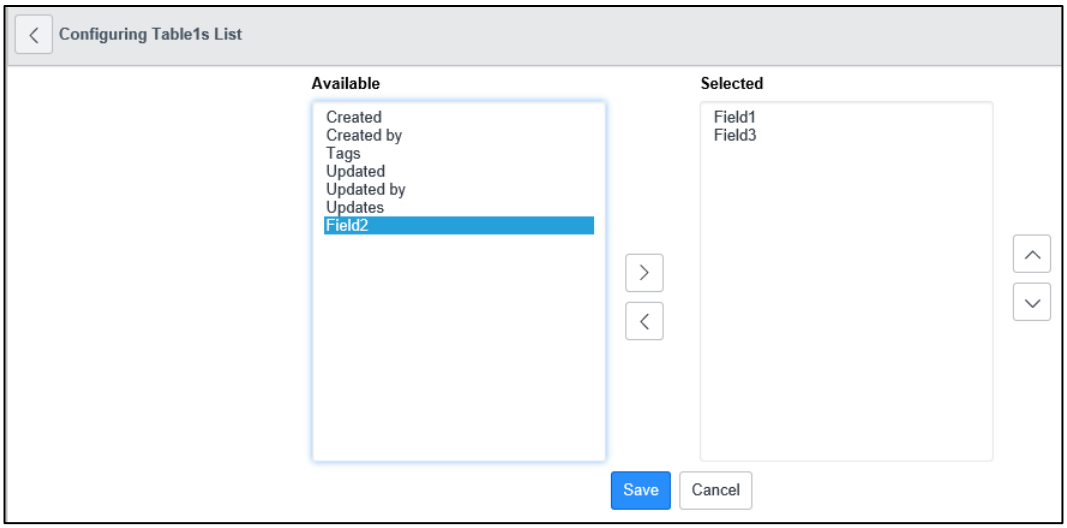


4. Click **OK**.

5. Navigate to **Table1 > Table1s**.

6. Right-click the list header and select **Configure > List Layout**.

7. Double-click **Field2** to remove it from the list layout.

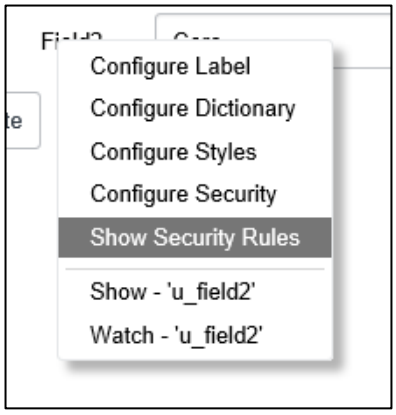


8. Click **Save**.

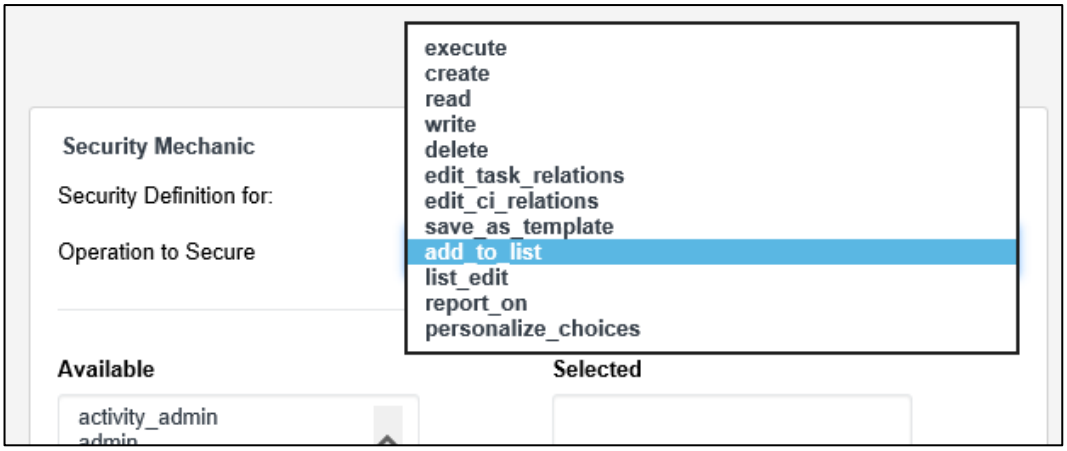
©2016 ServiceNow, Inc. All rights reserved.

Prevent a Field from Being Visible in a List View

1. Click to open one of the records.
2. Right-click the **Field2** label, and select **Configure Security**. This opens the **security_mechanic** UI page.

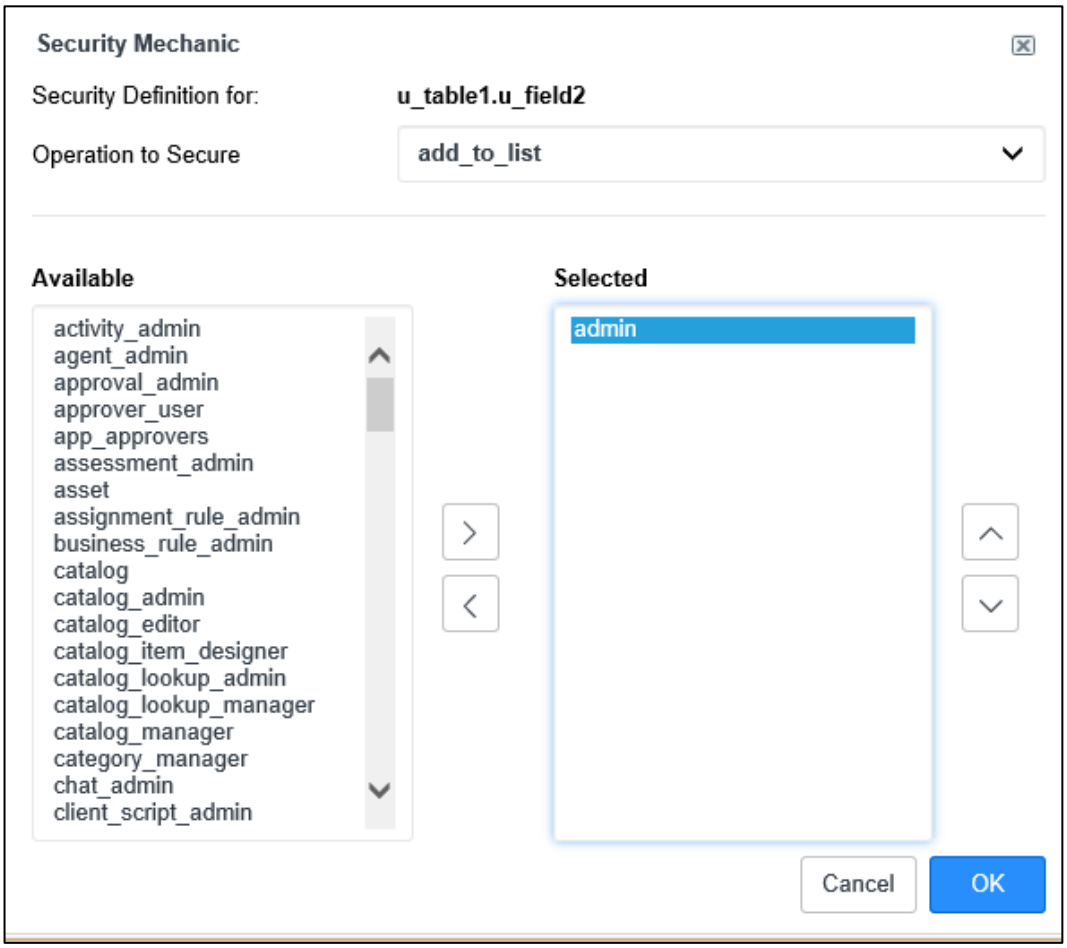


3. Select **add_to_list** in the **Operation to Secure** drop-down list.



©2016 ServiceNow, Inc. All rights reserved.

4. Select **admin** and move it to the **Selected** bucket.



5. Click **OK** to create the new rule.

6. When **Save complete** appears, highlighted in **green** at the bottom of the UI page, click the close button (**X**) in the upper-right corner of the page to exit. Note that using this method creates an access control rule without being an Elevated user.

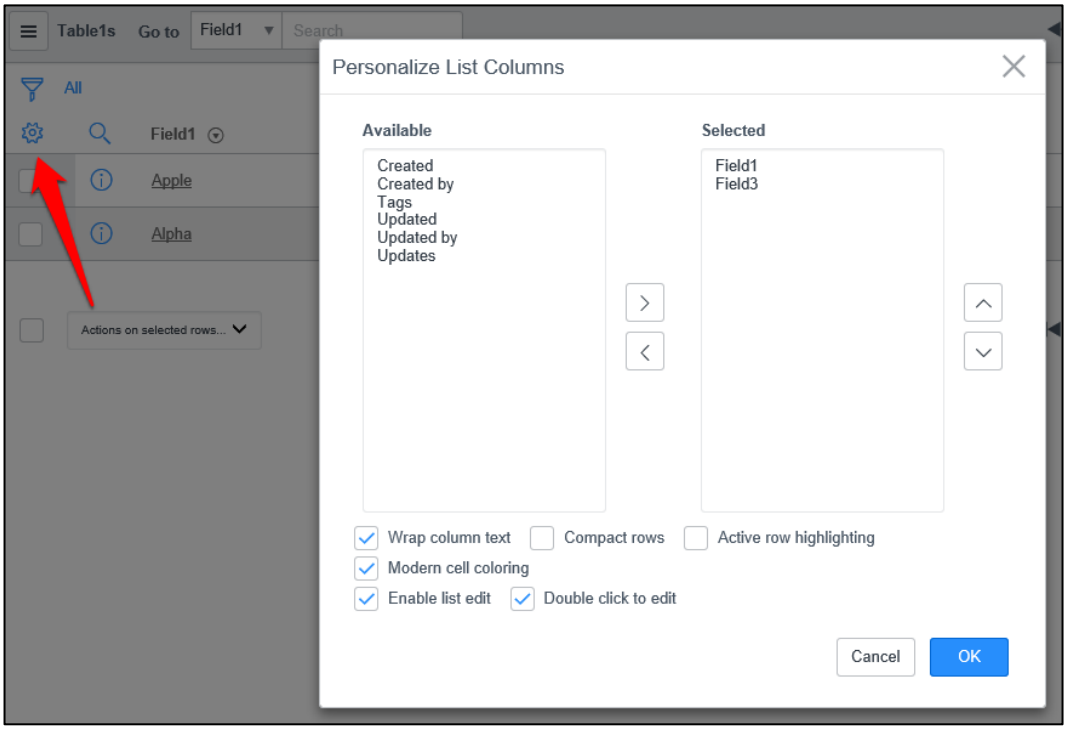


Verify the add_to_list Rule

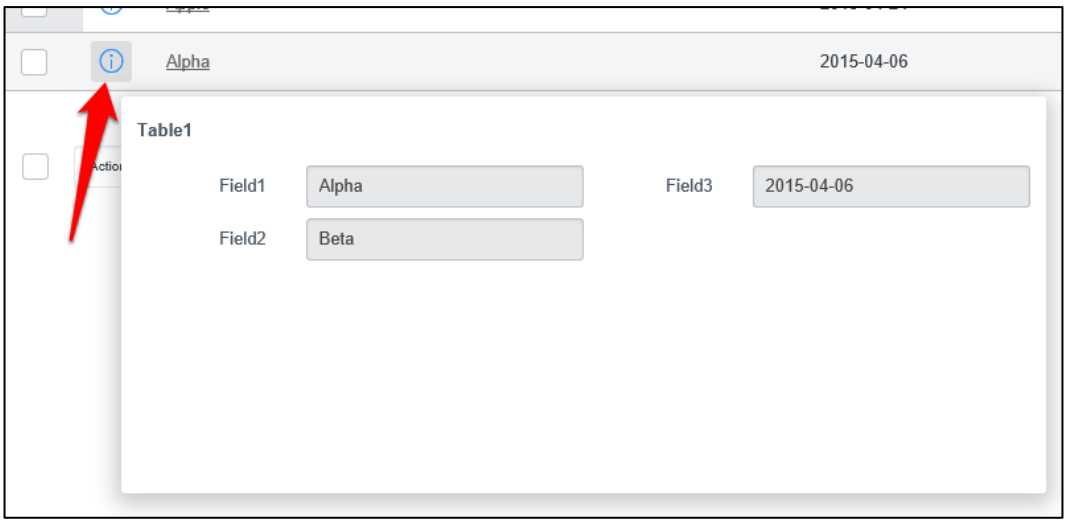
1. As Abel, navigate to **Table1 > Table1s**. Note that the number of columns has changed.

©2016 ServiceNow, Inc. All rights reserved.

2. Click the **gear wheel** in the upper-left of the list to personalize the columns. Note that **Field2** is not available. Click the **Cancel** button.



3. Hover over the **Alpha** record decoration, or click to open the record. Note that **Field2** is still visible. It is not visible on the **Apple** record, as it does not meet the requirements for the field read rule you created.

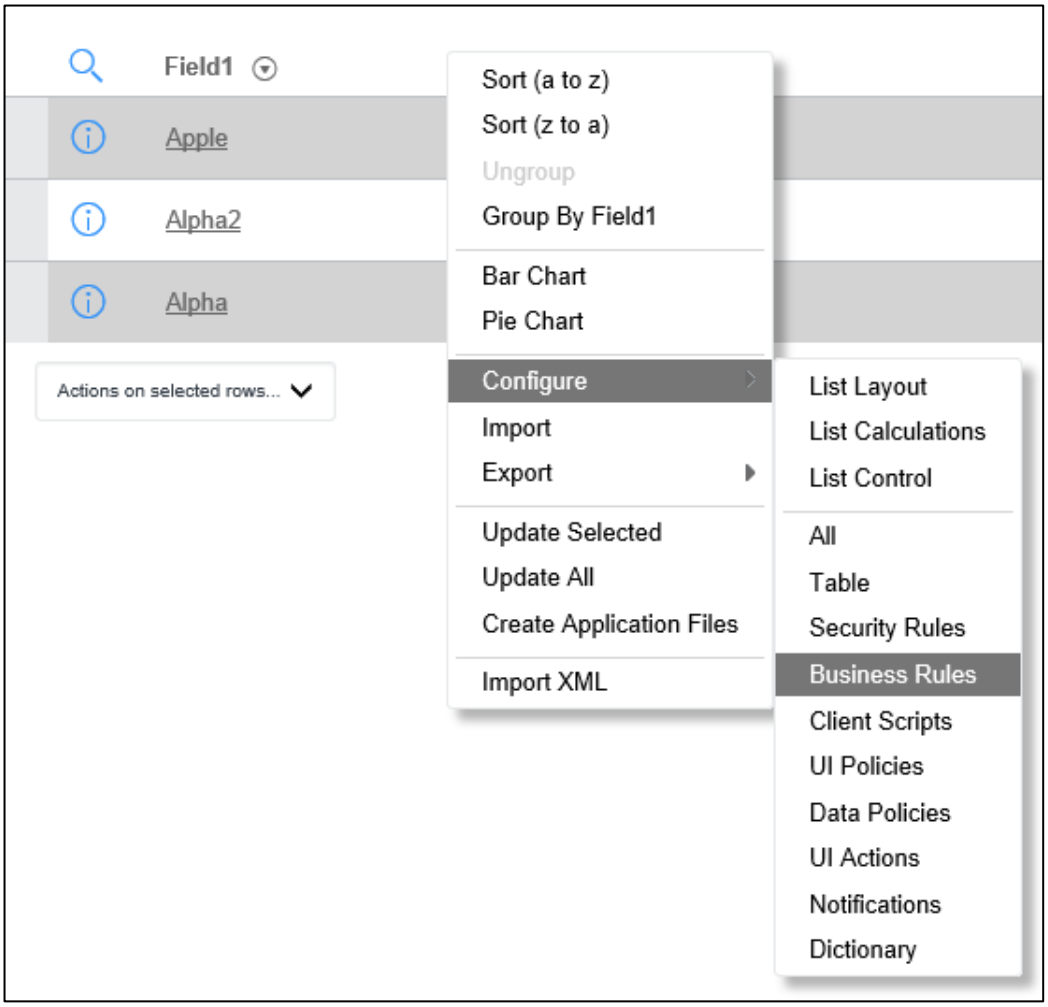


©2016 ServiceNow, Inc. All rights reserved.

Create a Before-Query Business Rule

1. To hide the “Number of rows removed from this list by Security constraints” message, navigate to **Table1 > Table1s** as **System Administrator**.

2. Right-click on a column header and select **Configure > Business Rules**.



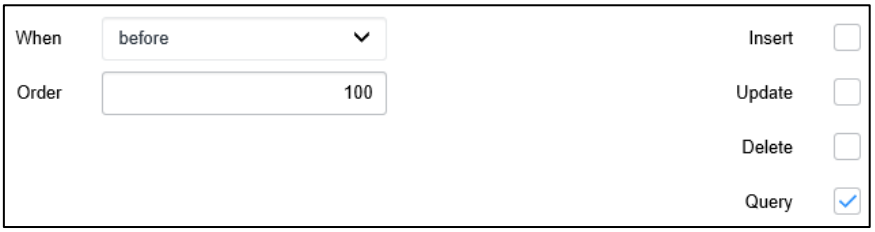
©2016 ServiceNow, Inc. All rights reserved.

3. Click **New** and enter the field information listed below:

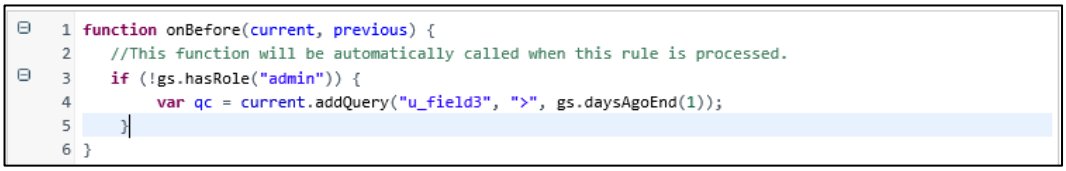
   Name: **u_table1 query**
   Advanc**ed: checked**
   When: **before**
   Query: **checked**
   Script: **Cut and paste the s**cript inside the supplied function block.

```
if (!gs.hasRole("admin")) {
        var qc = current.addQuery("u_field3", ">", gs.daysAgoEnd(1));
    }
```





Note: If you have the tabbed view enabled, the **When** field and **Query** check box are in the **When to run** tab. The **Script** field is in the **Actions** tab.



```
function onBefore(current, previous) {
   //This function will be automatically called when this rule is processed.
   if (!gs.hasRole("admin")) {
        var qc = current.addQuery("u_field3", ">", gs.daysAgoEnd(1));
    }
}
```

4. Click the **Check Syntax** button to ensure the script is correct.



   The script adds a query filter to any query for records that remove records, which would be blocked by the read rule created earlier. It checks to see if the user has the role admin first, then filters to add only records in which the date in Field3 is newer than yesterday.

5. Click **Submit**.

©2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

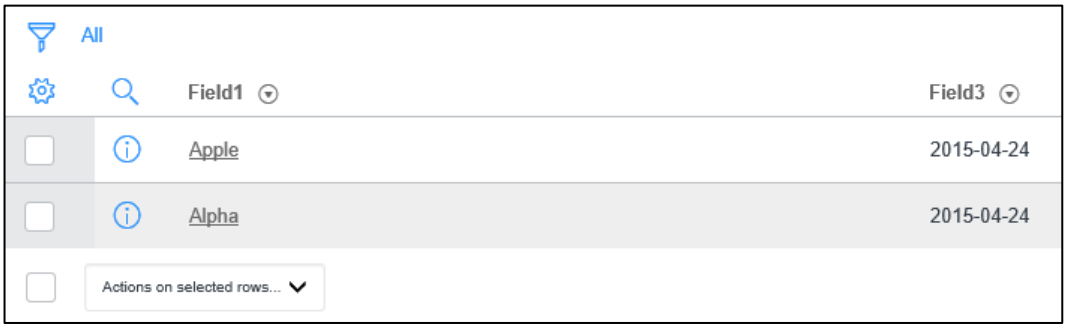
1. As System Administrator, navigate to **Table1 > Table1s**. Note the three records, now with only two columns.

2. Double-click **Field3** on one of the records. You cannot edit **Field3** from the list view, because the **Admin overrides** field was unchecked for that rule.

3. Click the **Personalize List cog** at the top-left of the list view. As System Administrator, you can add Field2 to the columns in the list view.

   Note: If you add **Field2** to the default list view using **Configure > List Layout** from the list context menu, it appears to all users in their list views, even if an **add_to_list** rule would prevent those users from adding the field.

4. As Abel Tuter, navigate to **Table1 > Table1s**. Note the list view no longer displays the security constraint message. Abel sees two columns in the list view. Clicking the Personalize List cog does not show the **Field2** column in the Available list.



©2016 ServiceNow, Inc. All rights reserved.

Lab Goal

This lab focuses on the **report_on** access control rule. This rule has practical applications beyond security. Limiting the tables that users can report on simplifies the user interface and limits potential performance issues.

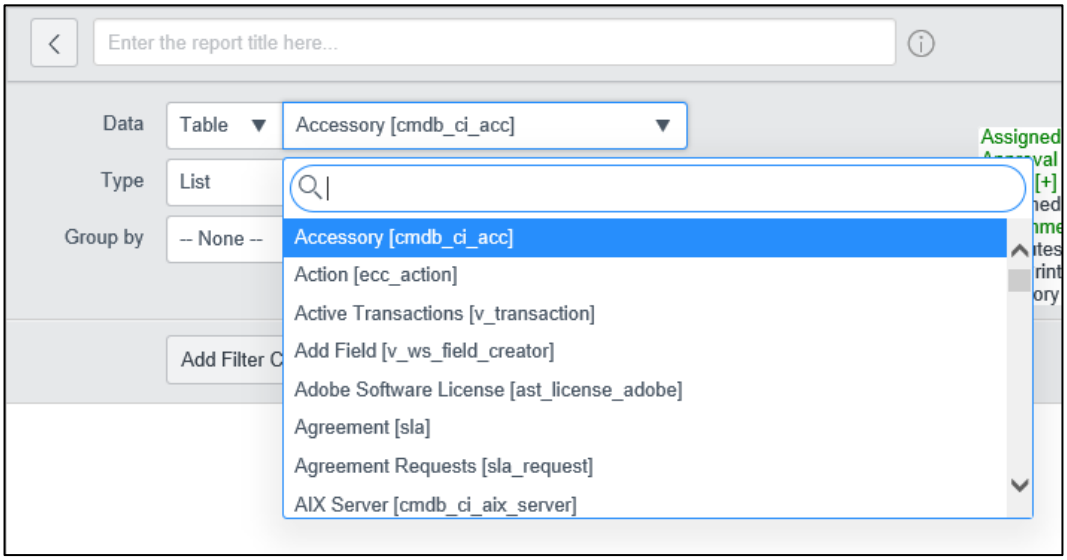
Lab 6.0 Practical Blacklist: Reports

The Practical Example

1. Log in as **ITIL** user with the provided credentials.
2. Navigate to **Reports > View / Run**.
3. Click **Create a report**.



4. Look at the tables available in the **Data** drop-down list. There are numerous tables, many of which are not of interest to users who want to report on tables or applications relevant to them.

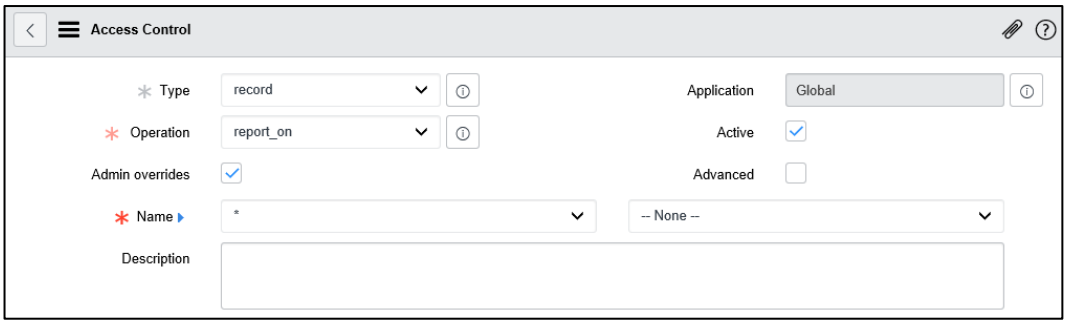


©2016 ServiceNow, Inc. All rights reserved.

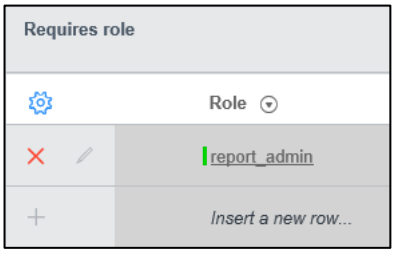
5. As an Elevated Admin, navigate to **System Security > Access Control (ACL)**.

6. Create a new rule, using the field information below:

   Operation: **report_on**
   Name: *****, **-- None --**



7. Add **report_admin t**o the **Requires role** list editor.



8. Click **Submit** to return to the Access Controls list view.

9. Create and submit a new rule for **Incident**, using the field information below:

   Operation: **report_on**
   Name: **Incident [incident]**, **-- None --**
   Requires Role: **itil**

10. Create and submit a new rule for **Problem**, using the field information below:

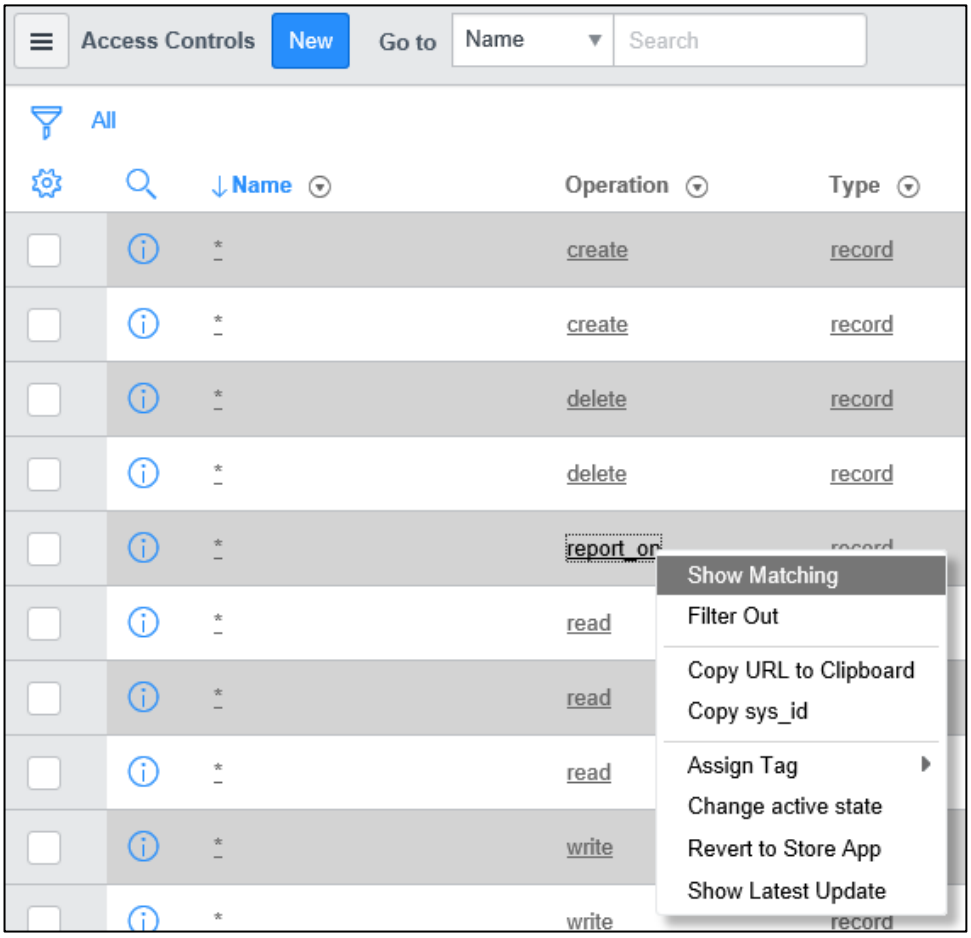
    Operation: **report_on**
    Name: **Problem [problem]**, **-- None --**
    Requires Role: itil

11. Create and submit a new rule for **Change Request**, using the field information below:

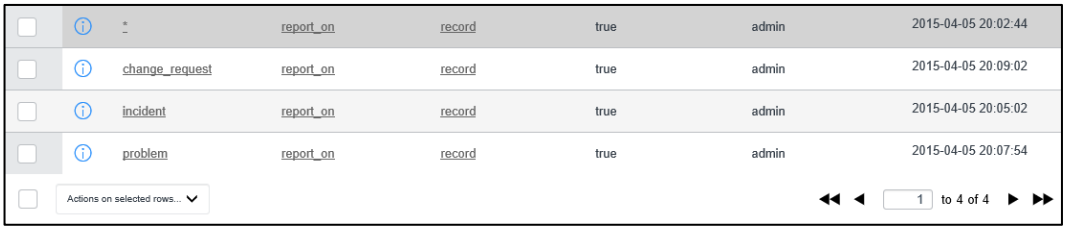
    Operation: **report_on**
    Name: **Change Request [change_request], -- None --**
    Requires Role: **itil**

®2016 ServiceNow, Inc. All rights reserved.

12. In the list view, filter the list for the **report_on** operation.
    Right-click the **report_on** field for the *** rule** and select **Show Matching**.



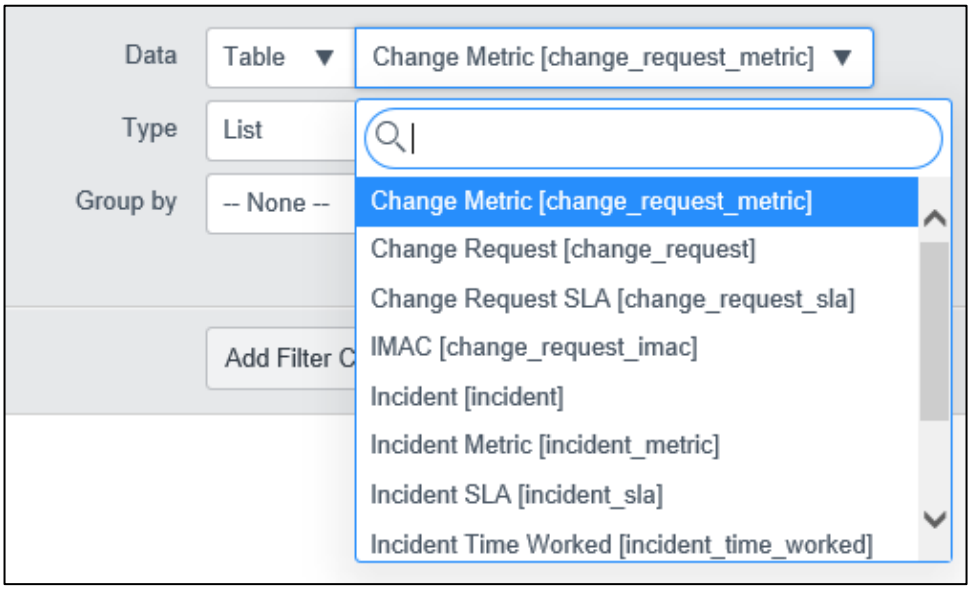
13. Four rules appear. If they do not match the values in the screenshot, you can edit them from here.



©2016 ServiceNow, Inc. All rights reserved.

Lab Success Verification

1. Log in as **ITIL** user.
2. Navigate to **Reports > View / Run**.
3. Click **Create a report**.
4. Note the tables available in the **Data** drop-down list. There now are fewer tables. The rules apply to any tables extended from the table, allowing the ITIL user to report on related tables as well.



Note: This is an example exercise. Before putting it into production, understand your reporting users and needs. It is easy to add reporting functionality for users based on group membership. This default configuration makes it possible for admins and users with the **report_admin** role to create reports on any table. This change does not prevent a user from being able to render any report already created and present in the report catalog.

©2016 ServiceNow, Inc. All rights reserved.

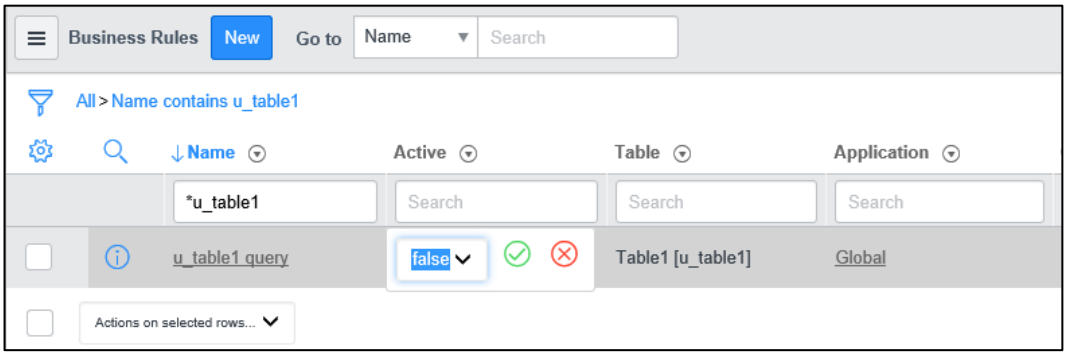
Lab Goal

ServiceNow provides tools to help figure out why you can or cannot see fields and records. This lab explains how the Debugging Access Control Rules is a server-side debug function.

Lab 7.0 Debugging Rules

Disable the Business Rule

1. As System Administrator, navigate to **System Definition > Business Rules**.
2. Search for **u_table1 query**.
3. Double-click the **Active** column.
4. Change the **Active** field to **false**.



Note: List edits save immediately.

Enable Security Rule Debugging

1. Navigate to **System Security > Debugging > Debug Security Rules**.
   A script complete message appears.

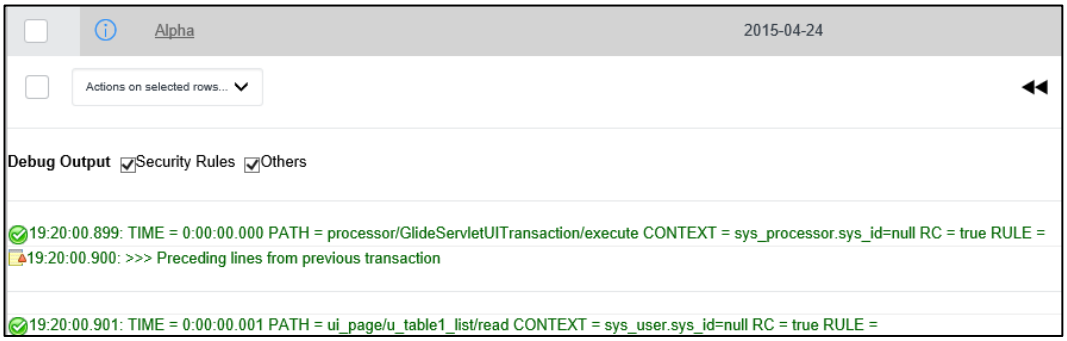
```
[0:00:00.016] Script completed in scope global: Debug Security Rules
```

©2016 ServiceNow, Inc. All rights reserved.

Examine Debug Output

2. If the user role is currently Elevated, remove the **security_admin** privileges.

3. Navigate to **Table1 > Table1s**. The Debug script has appended log data to the end of the content form.



4. Scroll down. Near the end of the output, there is a line with a red **X** icon in it. There is little that is restricted from the admin user.

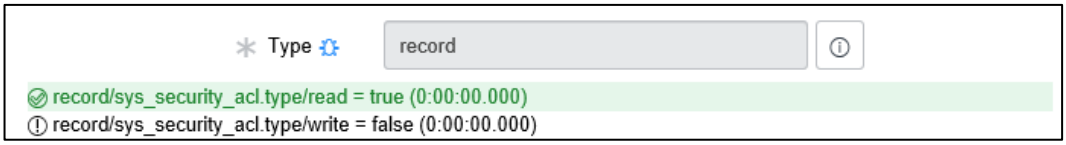


Note: The first icon in the line indicates success or failure overall for an operation. The three icons before the link correspond to the three evaluations that must be true before a rule evaluates to true: the **required roles**, the **condition builder** for the rule, and the **script** for the rule. The security rules are designed to fail fast. If a rule reaches a failure, it does not continue evaluating the rest of the conditions for that particular rule.

5. Click the **record/u_table1/read** link on that line.

6. A new window opens for the Access Control Rule you created. Note that the fields in the rule are read-only because the user is not currently Elevated.

7. Click the **bug** icon next to the **Type** field. There are check and exclamation notes. The green statement indicates that the System Administrator is allowed to read this field. The next line indicates that **write** evaluates to **false**. This is not **read-only** because of a dictionary field. It is **read-only** because there is an access control rule that makes it **read-only**.



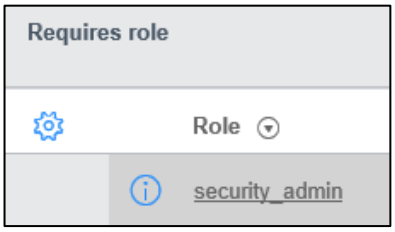
©2016 ServiceNow, Inc. All rights reserved.

8. Scroll down to the output, and continue down to the first entry that starts with a red **X**. This is the first denied operation, and should be deleted. Continue scrolling down until you see **record/sys_security_acl/write**. Click that link.



9. A new window opens for the system default access control rule for the access control table write operation. Review the record. Note that the **admin override** field is not checked. The **Requires roles** related list contains **security_admin**.





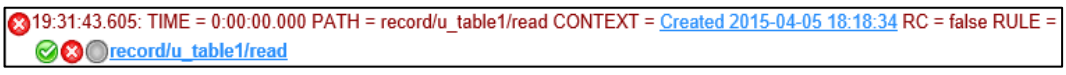
10. Close the two Access Control Rule windows just created.

Debug Other Users

1. Impersonate Abel Tuter. Debugging is session-specific. In order to see debug output, you must Impersonate a user, rather than open another browser.

2. Navigate to **Table1 > Table1s**.

3. We disabled the business rule that filters user queries automatically. The Security constraints message therefore reappears. Scroll down the debug output. Note that there are many more blocked operations for Abel. Most of them are **create** and **delete** operations. We did not create rules for those operations. The last one is a read operation, however, for **record/u_table1/read**.



Note: As Abel, you cannot follow that link to the Access Control Rule. Abel does not have permissions to read that table. If you try to follow the link, the debug output shows blocked operations for nearly everything.

©2016 ServiceNow, Inc. All rights reserved.

4. Scroll to the top and open the **Apple** record.

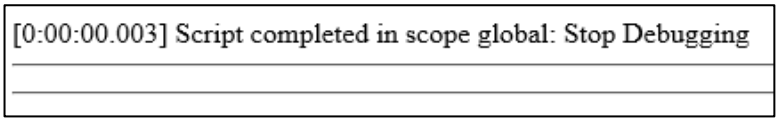
5. Remember Abel cannot see the **Field2** value on this record because the rule you created disabled the read operation when the value did not contain Beta. Scroll down. Locate the line that corresponds to that rule. It shows a failed evaluation and contains a **PATH = record/u_table1.u_field2/read** attribute.



Disable Debugging

1. Impersonate **System Administrator**.

2. Navigate to **System Security > Stop Debugging**. A script complete message appears.



©2016 ServiceNow, Inc. All rights reserved.

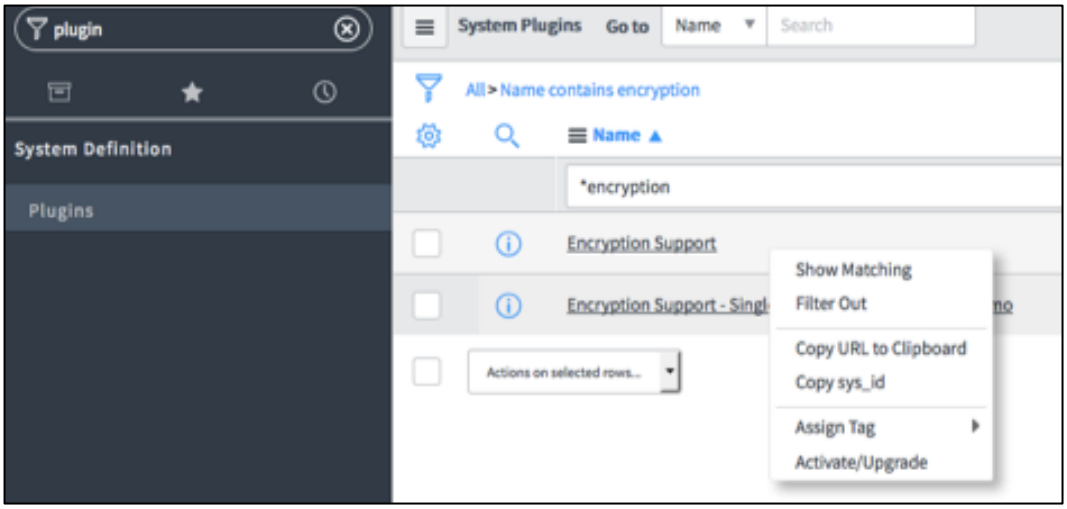
Lab Goal

This lab explains the Encryption plugin, which supports an additional method for securing data within a ServiceNow instance.

Lab 8.0 Encryption Support

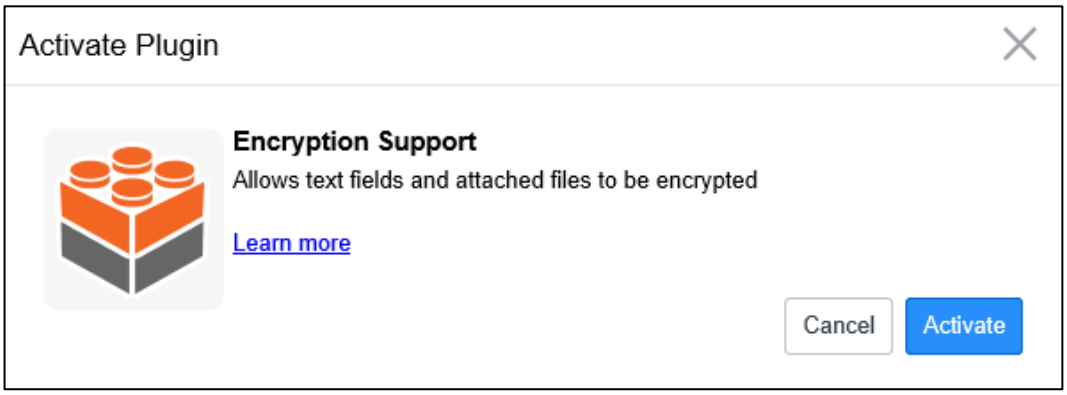
Activate the Plugin

1. As **System Administrator**, navigate to **System Definition > Plugins**.
2. Search for the **Encryption Support** plugin.
3. Right-click the plugin and select **Activate/Upgrade**.



©2016 ServiceNow, Inc. All rights reserved.

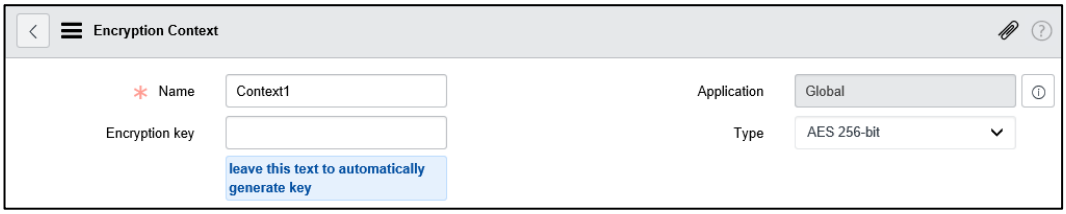
4. Click the **Activate** button in the resulting UI page.



5. When the activation completes, the UI page indicates success.

Create an Encryption Context

1. Navigate to **System Security > Users and Groups > Encryption Contexts**.
2. Click **New**.
3. Enter **Context1** in the **Name** field.
4. Select **AES 256-bit** from the **Type** drop-down list.



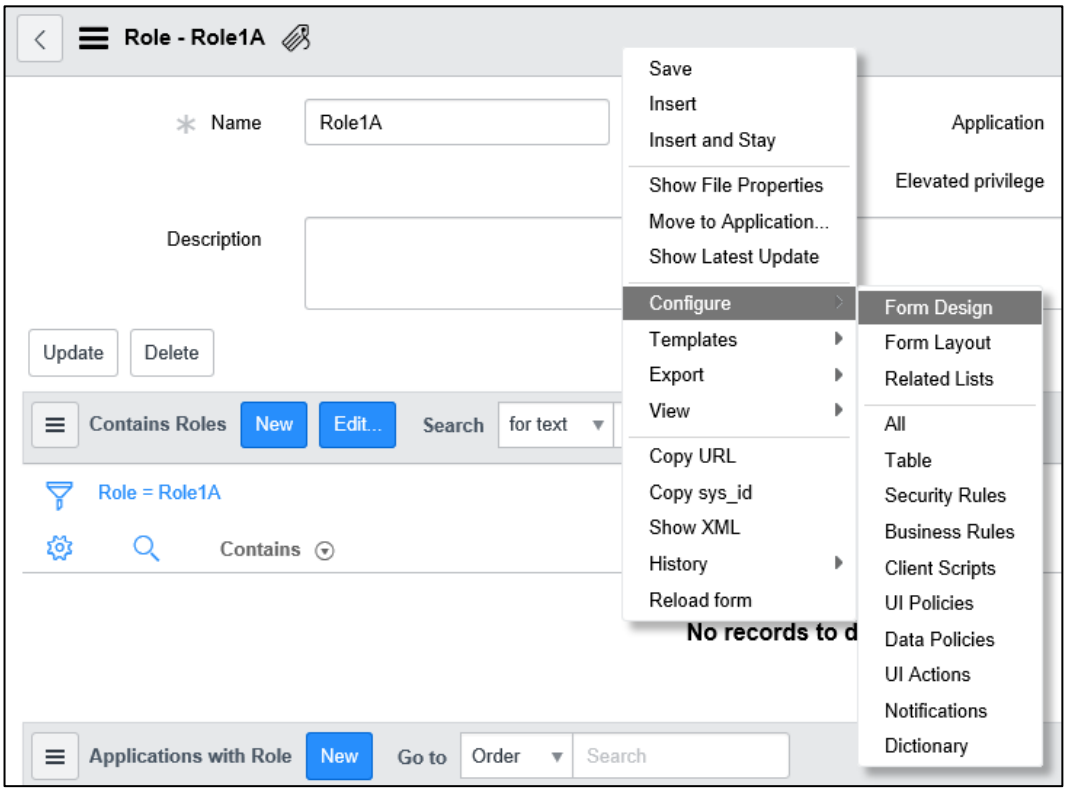
5. Click **Submit**.

Add the Encryption Context to a Role

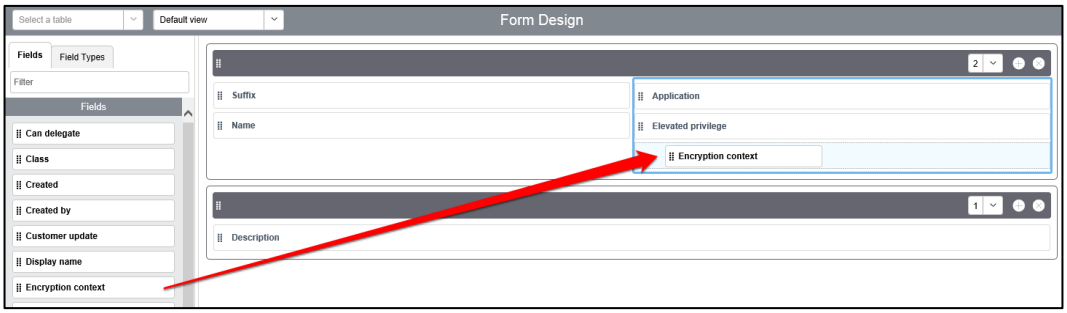
1. Navigate to **User Administration > Roles**.
2. Search for **Role1A**, and open the record.

©2016 ServiceNow, Inc. All rights reserved.

3. Right-click in the header bar and select **Configure > Form Design**. A new window opens.



4. On the **Fields** tab, drag the Encryption context field and drop it below the **Elevated privilege** field.



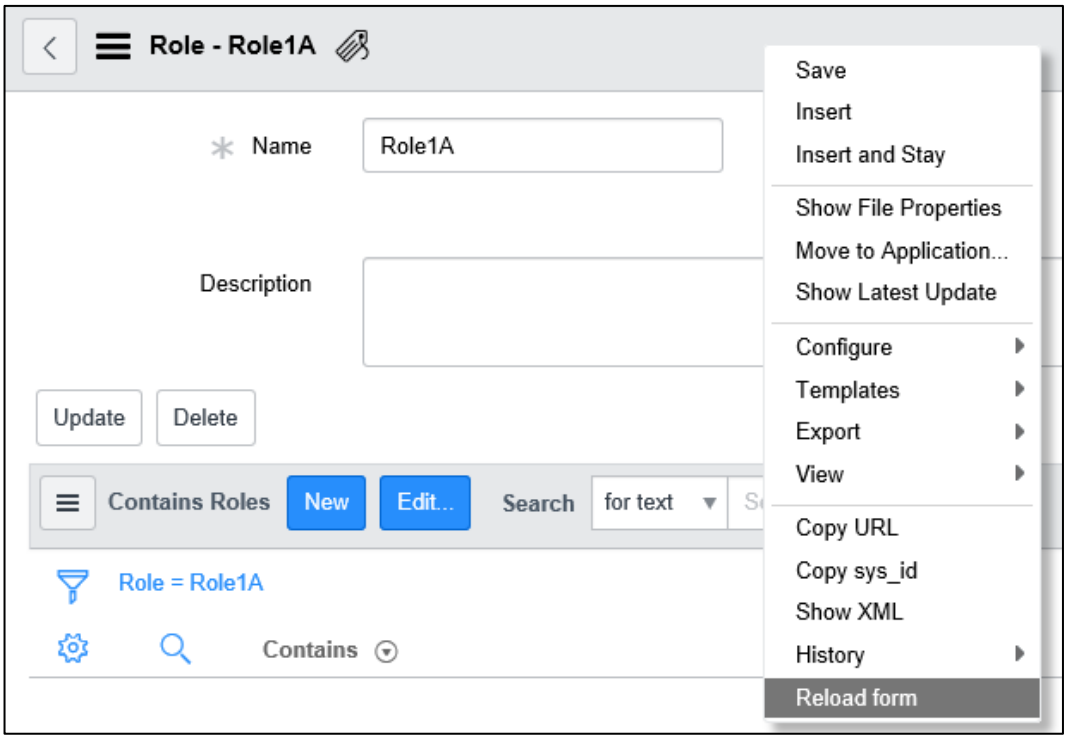
©2016 ServiceNow, Inc. All rights reserved.

5. Click **Save**.



6. Close the **Form Design** window.

7. In the original window, right-click the header and select **Reload form**.



©2016 ServiceNow, Inc. All rights reserved.

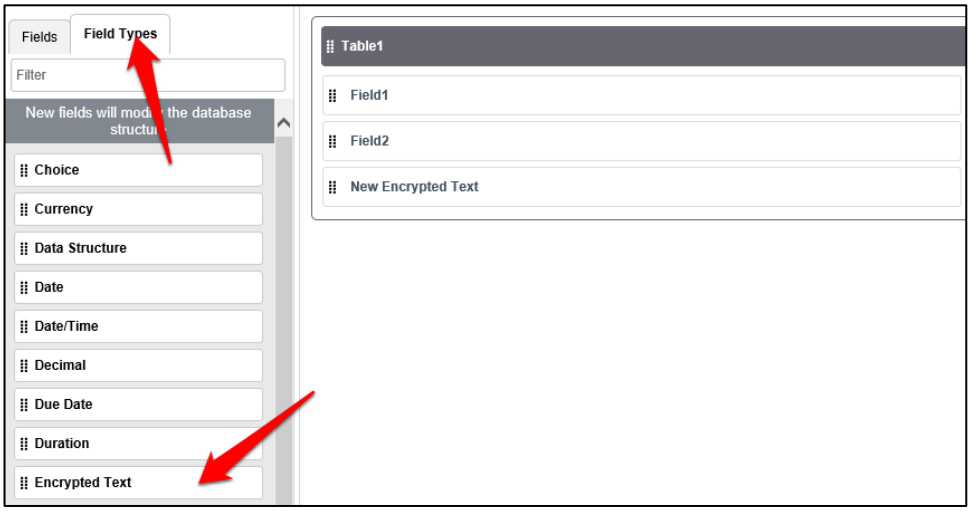
8. In the **Encryption Context** reference field, select **Context1**.



9. Click **Update**.

Add a New Encrypted Field to the Table

1. Navigate to **Table1 > Table1s**.
2. Open a record.
3. Right-click in the header bar and select **Configure > Form Design**.
4. In the navigation pane of the **Form Design** window, select the **Field Types** tab.
5. Double-click **Encrypted Text**.



6. Drag the **New Encrypted Text** field to below **Field3**.

©2016 ServiceNow, Inc. All rights reserved.

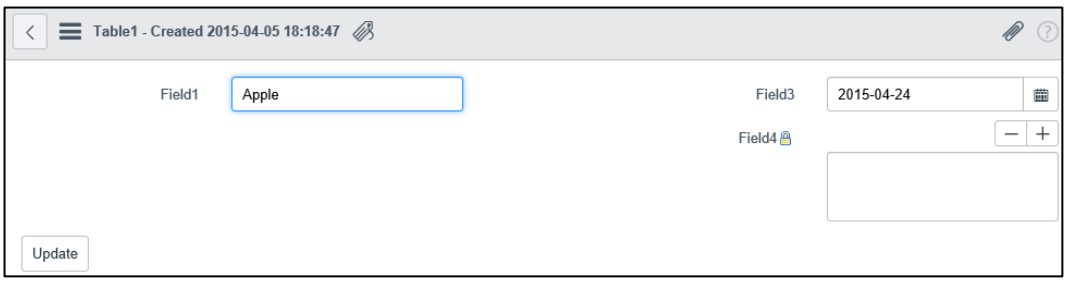
7. Click the **New Encrypted Text** label to be able to edit its label property. Enter **Field4**.



8. Click **Save**.
9. Close the **Form Design** window.
10. In the original window, right-click in the header and select **Reload form**. Note that the form does not show the new field as System Administrator.

Lab Success Verification

1. Impersonate **Abel Tuter**.
2. Navigate to **Table1 > Table1s**.
3. Open a record. Note that you are unable to see **Field4**. Your admin account does not inherit encryption contexts from impersonated accounts.
4. Log on in a **new browser session** as Abel Tuter.
5. Navigate to **Table1 > Table1s**.
6. Open a record. It displays the **Field4** encrypted text field.



©2016 ServiceNow, Inc. All rights reserved.