

# ServiceNow security ACLs

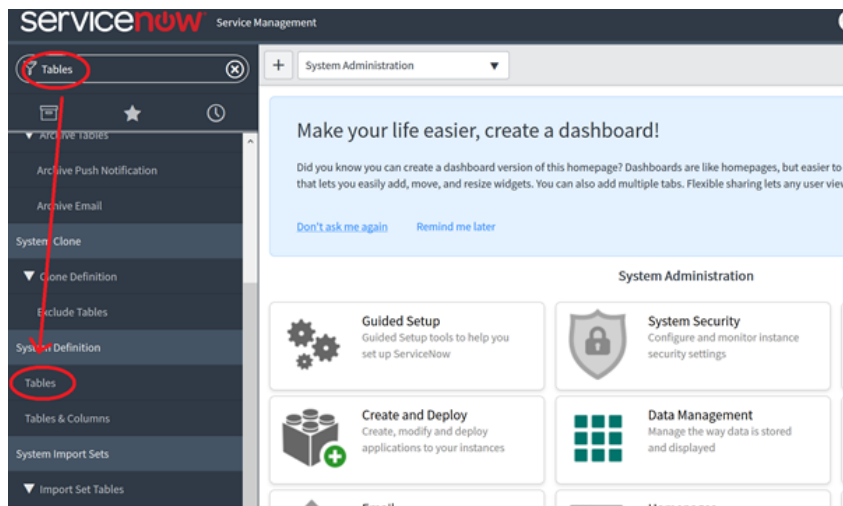
This section describes how to configure a ServiceNow server to allow the Aspire ServiceNow connector to retrieve security ACLs.

In order to follow the steps in this guide, a user with enough permissions to create custom tables and scheduled jobs in the ServiceNow server is required.

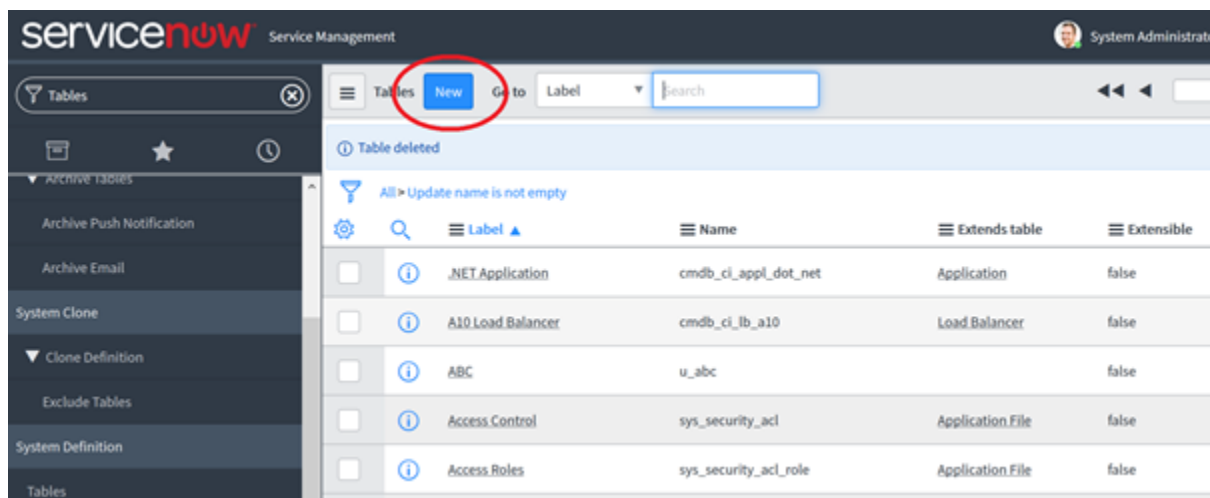
## Custom tables creation

Two tables are required for the configuration of the server: An “ACL Tables” table and a “ACL Table Users” table:

- Log in to your ServiceNow server instance.
- In the “Filter Navigator” write the word “Tables”.



- Under “System Definition”, select the option “Tables”.
- In the “Tables” page, click on “New”.



- In the “Tables – New Record” page, specify “ACL Tables” as the label of the table. The system will automatically assign the name of the table to “u\_acl\_tables”. It is very important that the name of this table be exactly that name so please double check it.

servicenow Service Management

Tables

Table New record

A table is a collection of records in the database. Each record corresponds to a specific instance of data. Applications use tables and records to manage data and processes. [More Info](#)

\* Label ACL Tables

\* Name u\_acl\_tables

Extends table

- At the bottom of "Tables – New Record", insert a new column with the name "Table". Set "Type" to "String" and the "Display" value to "true", then click the "Submit" button.

Columns Controls Application Access

Table Columns Search for text Search

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
X	Table	String				true
+	Insert a new row...					

Submit Cancel

- Back to the "Tables" page, browse to the newly created "ACL Tables" table and set its "Extensible" attribute to "true".

Name	Label	Type	Extensible	Last Modified
ACL	cmdo_ci_id_ace	Load Balancer	false	2018-07-14 14:28:54
ACL Endpoint	cmdb_ci_endpoint_acl	Endpoint	false	2018-07-14 14:22:16
ACL Tables	u_acl_tables		true	2018-09-20 07:34:39
Action	ecc_action		false	2018-07-14 14:19:03

- In the "Tables" page, click on "New" again.
- In the "Tables – New Record" page, specify "ACL Table Users" as the label of the table. The system will automatically assign the name of the table to "u\_acl\_table\_users". It is very important that the name of this table be exactly that name so please double check this one as well.
- In the "Extends Table" option browse and select the table we created previously ("ACL Tables").

Applications use tables and records to manage data and processes. [More Info](#)

\* Label

\* Name

Extends table

Create

Add n

- At the bottom of "Tables – New Record", insert a new column with the name "User". Set "Type" to "String" and the "Display" value to "true", then click the "Submit" button.

Columns Controls Application Access

Table Columns Search for text Search

Dictionary Entries

	Column label	Type	Reference	Max length	Default value	Display
<input type="button" value="X"/> <input type="button" value="P"/>	User	String				true
<input type="button" value="+"/>	Insert a new row...					

Submit Cancel

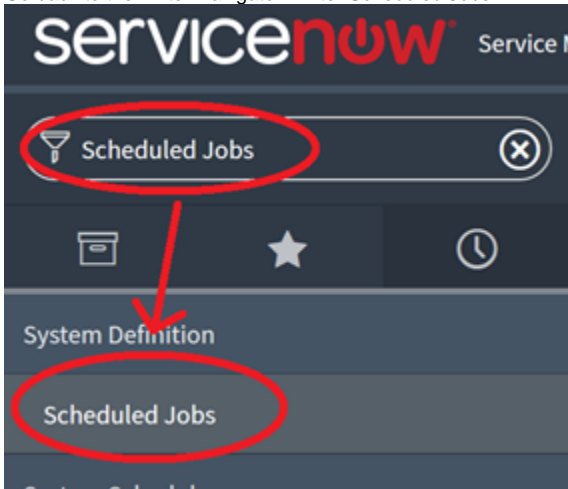
- You should now have two tables in the "Tables" page: "ACL Tables" and "ACL Table Users". Verify the details circled in red and continue.

	Label	Name	Extends table	Extensible	
<input type="checkbox"/>	ACL Table Users	u_acl_table_users	ACL Tables	false	2018
<input type="checkbox"/>	ACL Tables	u_acl_tables		true	2018
<input type="checkbox"/>	Action	ecc_action		false	2018

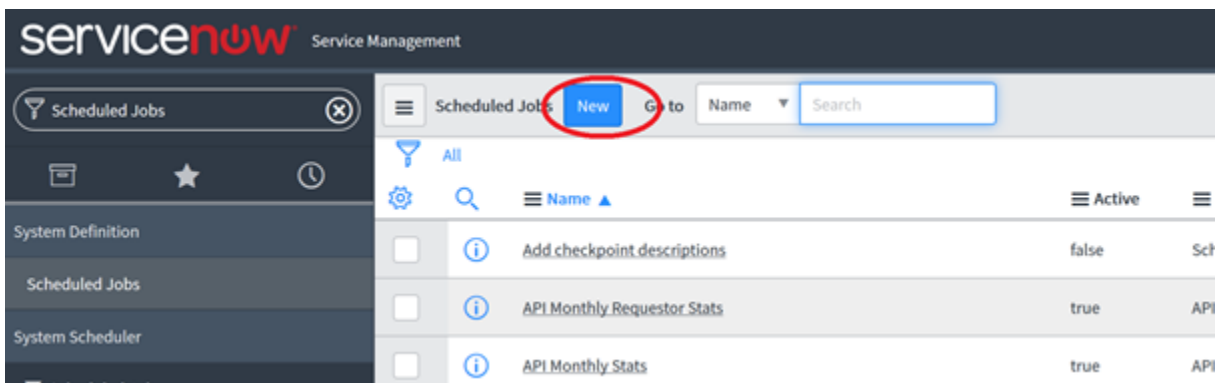
## Scheduled job creation

We need a script to fill the ACLs tables and keep them updated. This script will be run with a ServiceNow Scheduled Job:

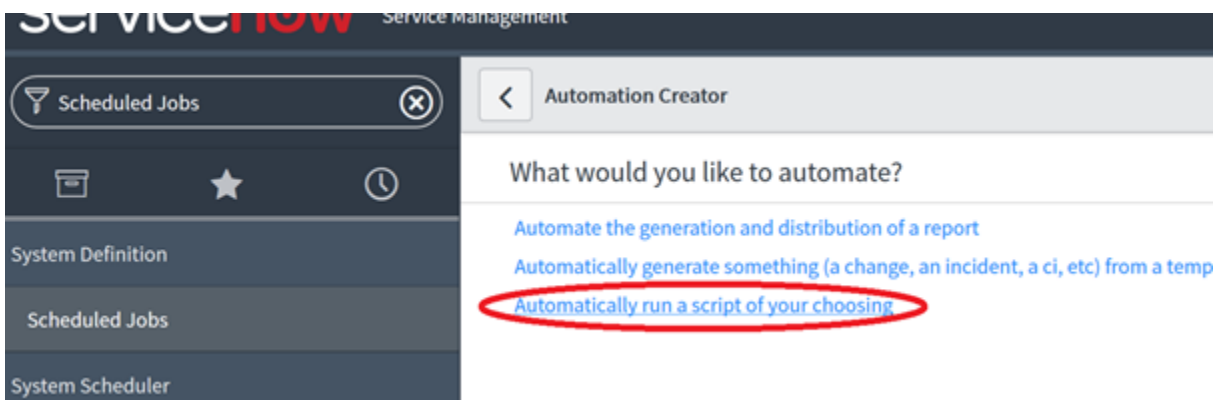
- Go back to the "Filter Navigator" write "Scheduled Jobs".



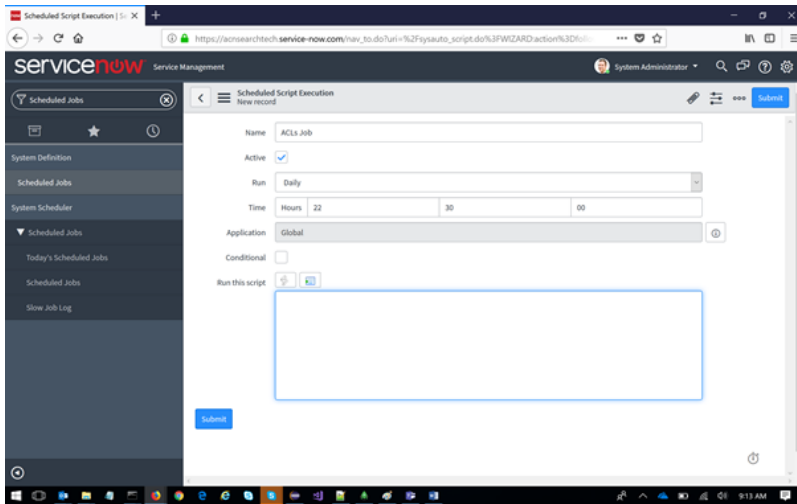
- Under "System Definition", select the option "Scheduled Jobs".
- In the "Scheduled Jobs" page, click on "New".



- On the "Automation Creator" page, select the "Automatically run a script of your choosing".



- On the "Scheduled Script Execution – New Record" page, specify a name for the job and set a schedule according to your needs or the need of your client. The script may be a long running script so plan accordingly.



- On the "Run this script" section, copy and paste the following script:

```
// Retrieve all Knowledge Bases
var kbs_record = new GlideRecord('kb_knowledge_base');
kbs_record.addQuery('active',true);
kbs_record.query();

// Admin user is stored to restore it after impersonations
var adminUser = gs.getSession().getUserName().toString();

var user_record = null;
var kb_record = null;

var acl_tables_record = new GlideRecord('u_acl_tables');
var acl_table_users_record = new GlideRecord('u_acl_table_users');

var kb_id = null;

// Each KB is inserted in the ACL Tables table
while(kbs_record.next()) {
    kb_id = kbs_record.sys_id.toString();

    acl_tables_record.initialize();
    acl_tables_record.addQuery('u_table', kb_id);
    acl_tables_record.query();

    // If the knowledge base record is not on the table, we add it
    if (!acl_tables_record.next()){
        acl_tables_record.u_table = kb_id;
        acl_tables_record.insert();
    }
}
```

```

// For each KB, we verify if users are allowed to access it. If they do, a record is inserted in the ACL
Table Users table

user_record = new GlideRecord('sys_user');
user_record.addQuery('active',true);
user_record.query();
while(user_record.next()) {

    var impersonateSuccess = gs.getSession().impersonate(user_record.user_name);

    if (impersonateSuccess){
        kb_record = new GlideRecord('kb_knowledge_base');

        acl_table_users_record.initialize();
        acl_table_users_record.addQuery('u_table', kb_id);
        acl_table_users_record.addQuery('u_user', user_record.user_name);

        // If the user has permissions, we add the record
        if (kb_record.get(kb_id) && kb_record.canRead()) {
            acl_table_users_record.query();

            // If the record is not already in the table, we add it
            if (!acl_table_users_record.next())
            {
                acl_table_users_record.u_table = kb_id;
                acl_table_users_record.u_user = user_record.user_name;
                acl_table_users_record.insert();
            }
        }

        // If the user has no permissions, we try to remove the record
        else
        {
            acl_table_users_record.deleteMultiple();
        }
    }

    gs.getSession().impersonate(adminUser);
}
}

```

- Now click on the “Submit” button.

```
// Each KB is inserted in the ACL Tables table
while(kbs_record.next()) {
    kb_id = kbs_record.sys_id.toString();

    acl_tables_record.initialize();
    acl_tables_record.addQuery('u_table', kb_id);
    acl_tables_record.query();
}
```



Submit

- Congratulations! Your ServiceNow server is now configured to be used with our ServiceNow connector. The connector will be able to query the table 'u\_acl\_table\_users', using the Knowledge Base ID that every crawled document has, the result is a list of users with read permissions for that Knowledge Base.