Technical Document: ServiceNow to SailPoint Role Request Workflow

May 14, 2025

Contents

1 Overview

This document outlines the process of handling role requests initiated from ServiceNow to SailPoint IdentityIQ (IIQ) for provisioning Z accounts (privileged accounts in Active Directory) and ZC accounts (Azure cloud accounts). The workflow involves creating or modifying accounts, assigning roles, and managing AD groups or Azure cloud groups based on the request type. The process leverages SailPoint workflows, rules, and custom objects to automate account provisioning and role assignment.

2 Process Flow

2.1 ServiceNow Initiates Role Request

ServiceNow sends a role request to SailPoint via a REST API call. The request is a POST request to the SailPoint endpoint `http://silvmspiqd01.ent.dir.labor.gov:8080/identityiq`. The JSON payload contains the necessary parameters for the role request.

2.1.1 JSON Payload Example

```
{
  "workFlowArgs": {
    "ritm": "RITM0540493",
    "identityName": "May_User4",
    "email": "May_User4@dev.dol.gov",
    "accountType": "ZC",
    "role": "Desktop - Tier 3 Support",
    "requestType": "Add/Modify",
    "launcher": "spadmin"
  }
}
```

- **ritm**: Request ticket number (e.g., RITM0540493).
- **identityName**: The name of the user (e.g., May_User4).
- **email**: User's email address (e.g., May_User4@dev.dol.gov).
- **accountType**: Type of account (e.g., Z for AD privileged accounts, ZC for Azure accounts).
- **role**: The business role to assign (e.g., Desktop - Tier 3 Support).
- **requestType**: Type of request (e.g., Add/Modify, New Account Request, Delete Account).
- **launcher**: The user or system initiating the request (e.g., spadmin).

2.2 SailPoint Workflow: ESD Role Request Workflow

The SailPoint workflow `ESD Role Request Workflow` (filename: `ESD Role Request Workflow.xml`) is triggered upon receiving the REST API call. This workflow processes the JSON payload and orchestrates the provisioning process.

2.2.1 Key Steps in the Workflow

1. **Extract Parameters**: The workflow extracts the parameters from the JSON payload (e.g., `identityName`, `role`, `accountType`, `requestType`).
2. **Generate Provisioning Plan**:
   - The workflow references the rule `ESD-12 Role Request Plan Generation` (filename: `ESD-12 Role Request Plan Generation.xml`) to create a provisioning plan.
   - The rule takes inputs such as `roleIds`, `identityName`, `requestType`, and `accountType` to generate a provisioning plan for role assignment or account creation.
3. **Account Selection**:
   - The workflow uses the rule `ESD-12 Z Account Selection` (filename: `ESD-12 Z Account Selection.xml`) to determine if the user already has a Z account in AD or a ZC account in Azure.
   - If no account exists, a new account is created with a specific naming convention.
4. **Provision Account and Assign Roles**:
   - Based on the provisioning plan, SailPoint provisions the account in AD (for Z accounts) or Azure (for ZC accounts) and assigns the appropriate roles.
5. **Post-Provisioning**:
   - For Z accounts, a PAM (Privileged Access Management) account is created (covered in a separate document).
   - For ZC accounts, no PAM account creation is required.
6. **Notify ServiceNow**: The workflow notifies ServiceNow of the provisioning result.

2.3 Rule: ESD-12 Role Request Plan Generation

This rule is responsible for generating the provisioning plan based on the request type (`New Account Request`, `Add/Modify`, `Delete Account`).

2.3.1 Logic

- **Inputs**:
  - `roleIds`: List of role IDs to assign.
  - `identityName`: Name of the identity (e.g., May_User4).
  - `requestType`: Type of request (e.g., Add/Modify).
  - `nativeId`: Native identity for the account (if applicable).
- **Custom Object Reference**:
  - The rule references the custom object `ESD-12 Custom` (filename: `ESD-12 Custom.xml`) to retrieve configuration values such as `Z_FORMAT`, `Z_AD_OU`, `ZC_ROLE_NAME_FILTER`, and `Z_ROLE_NAME_FILTER`.
- **Provisioning Plan Creation**:
  - For `New Account Request`:

        - Creates a new account request for the application `IIQ`.
        - Assigns roles based on the `roleIds`.
        - Differentiates between Z roles (on-premises) and ZC roles (cloud) using filters (`Z_ROLE_NAME_FILTER` and `ZC_ROLE_NAME_FILTER`).
    - For `Add/Modify`:
        - Modifies an existing account by adding or removing roles.
    - For `Delete Account`:
        - Deletes the specified account.

- **Output**:
    - Returns a `ProvisioningPlan` object that SailPoint uses to execute the provisioning.

2.3.2 Key Code Snippet

```
if (requestType.equalsIgnoreCase(newAccoutValue)) {
  AccountRequest onPremacctReq = null;
  AccountRequest onCloudacctReq = null;
  attrOpr = ProvisioningPlan.Operation.Add;

  for (String roleId : roleIds) {
    AttributeRequest attrReq = new AttributeRequest();
    attrReq.setOp(attrOpr);
    Attributes attr = new Attributes();
    attr.put("assignment", true);
    attrReq.setArgs(attr);
    attrReq.setName("assignedRoles");
    Bundle roleObj = context.getObjectByName(Bundle.class, roleId);
    attrReq.setValue(roleObj.getName());

    if (roleObj.getName().toLowerCase().endsWith(zRoleNameFilter)) {
      if (onPremacctReq == null) {
        onPremacctReq = new AccountRequest();
        onPremacctReq.setOperation(AccountRequest.Operation.Modify);
        onPremacctReq.setApplication("IIQ");
        onPremacctReq.add(attrReq);
      } else {
        onPremacctReq.add(attrReq);
      }
    } else if (roleObj.getName().toLowerCase().endsWith(
       zcRoleNameFilter)) {
      if (onCloudacctReq == null) {
        onCloudacctReq = new AccountRequest();
        onCloudacctReq.setOperation(AccountRequest.Operation.Modify);
        onCloudacctReq.setApplication("IIQ");
        onCloudacctReq.add(attrReq);
      } else {
        onCloudacctReq.add(attrReq);
      }
    }
  }
  plan.setSource("LCM");
  if (onCloudacctReq != null) {
    plan.add(onCloudacctReq);
  }
  if (onPremacctReq != null) {
    plan.add(onPremacctReq);
  }
}
```

2.4 Rule: ESD-12 Z Account Selection

This rule determines whether the user already has a Z account in AD or a ZC account in Azure. If no account exists, it creates a new account with a specific naming convention.

2.4.1 Logic

- **Inputs**:
  - identity: The identity object of the user.
  - links: List of existing accounts (links) associated with the identity.
- **Custom Object Reference**:
  - Retrieves Z_FORMAT (e.g., "Z-") and Z_AD_OU (organizational unit in AD) from the custom object ESD-12 Custom.
- **Account Selection**:
  - Checks if the user has an existing Z account by looking for an account with sAMAccountName starting with Z_FORMAT.
  - If no Z account exists, creates a new account with:
    - sAMAccountName: Z-<firstname><lastname> (e.g., Z-MayUser4).
    - Distinguished Name (DN): cn=Z-<firstname><lastname>,<Z_AD_OU> (e.g., cn=Z-MayUser4,ou=
- **Output**:
  - Returns a Link object representing the selected or newly created account.

2.4.2 Key Code Snippet

```
Link selectedLink = null;
Custom custObj = context.getObjectByName(Custom.class, "ESD-12 Custom");
Map appMap = custObj.getAttributes();
String zFormat = appMap.get("Z_FORMAT");
String zOu = appMap.get("Z_AD_OU");

if (links != null) {
  for (Link link : links) {
    String id = link.getAttribute("sAMAccountName").toString();
    if (id.toLowerCase().startsWith(zFormat)) {
      selectedLink = link;
    }
  }
}

if (selectedLink == null) {
  String idenName = identity.getName();
  idenName = zFormat + idenName;
  String natId = "cn=" + idenName + "," + zOu;
  Link _nlink = new Link();
  _nlink.setNativeIdentity(natId);
  selectedLink = _nlink;
}

return selectedLink;
```

2.5 Custom Object: ESD-12 Custom

The custom object `ESD-12 Custom` (filename: `ESD-12 Custom.xml`) stores configuration values used by the rules.

2.5.1 Key Attributes

- `Z_FORMAT`: Prefix for Z accounts (e.g., "Z-").
- `Z_AD_OU`: Organizational unit for Z accounts in AD (e.g., `ou=PrivilegedAccounts,dc=ent,dc=dir,dc=labor`
- `ZC_ROLE_NAME_FILTER`: Filter for ZC roles (e.g., "_zc").
- `Z_ROLE_NAME_FILTER`: Filter for Z roles (e.g., "_z").
- `NEW_ACC_REQ_FORM_STR`: Value for new account requests (e.g., "New Account Request").
- `ADD_ROLES_FORM_STR`: Value for adding roles (e.g., "Add/Modify").
- `REMOVE_ROLES_FORM_STR`: Value for removing roles (e.g., "Remove Roles").
- `DELETE_ACC_FORM_STR`: Value for deleting accounts (e.g., "Delete Account").

2.6 Account Provisioning

2.6.1 Z Account (Active Directory)

- **Account Creation**:
  - If the user does not have a Z account, a new AD account is created with:
    - `sAMAccountName`: `Z-<firstname><lastname>` (e.g., Z-MayUser4).
    - DN: Based on the `Z_AD_OU` from the custom object.
  - The AD provisioning policy assigns the user to the appropriate AD groups tied to the IT roles associated with the business role (e.g., Desktop - Tier 3 Support).
- **Role Assignment**:
  - For `Add/Modify` requests, the provisioning plan adds or removes AD groups as needed.
- **Post-Provisioning**:
  - A PAM (Privileged Access Management) account is created (details in a separate document).

2.6.2 ZC Account (Azure)

- **Account Creation**:
  - If the user does not have a ZC account, a new Azure account is created with:
    - UPN (User Principal Name): `zc-<firstname><lastname>@domain` (e.g., zc-MayUser4@dev.dol.gov
  - The provisioning plan assigns the user to Azure cloud groups tied to the IT roles associated with the ZC business role.
- **Role Assignment**:
  - For `Add/Modify` requests, the provisioning plan adds or removes Azure groups as needed.
- **Post-Provisioning**:
  - No PAM account creation is required for ZC accounts.

2.7 Notify ServiceNow

The workflow checks the provisioning result using the `Check Provisioning Success` step (from `ESD Role Request Workflow.xml`). If successful, it notifies ServiceNow of the outcome.

2.7.1 Key Code Snippet

```
if (project != null) {
  logger.debug("Project Executed in identity:\n" + project.toXml());
  List<ProvisioningPlan> plans = project.getPlans();
  if (plans != null) {
    for (ProvisioningPlan plan1 : plans) {
      List<AccountRequest> accReqsts = plan1.getAccountRequests();
      if (accReqsts != null) {
        for (AccountRequest accReq : accReqsts) {
          ProvisioningResult res = (ProvisioningResult) accReq.getResult();
          String provStatus = res.getStatus();
          logger.debug(accReq.getApplication() + " Provisioning status is: " + res.getStatus());
          if (provStatus != null && (provStatus.equalsIgnoreCase(ProvisioningResult.STATUS_COMMITTED) || provStatus.equalsIgnoreCase(ProvisioningResult.STATUS_QUEUED))) {
            provSuccessful = "true";
          } else {
            provSuccessful = "false";
            return provSuccessful;
          }
        }
      }
    }
  }
}
return provSuccessful;
```

3 Summary

- **ServiceNow** initiates a role request via a REST API call to SailPoint with a JSON payload.

- The `ESD Role Request Workflow` processes the request, generates a provisioning plan using the `ESD-12 Role Request Plan Generation` rule, and selects or creates an account using the `ESD-12 Z Account Selection` rule.
- **Z Accounts** are provisioned in AD with a `Z-<firstname><lastname>` naming convention, assigned AD groups, and followed by PAM account creation.
- **ZC Accounts** are provisioned in Azure with a `zc-<firstname><lastname>` UPN, assigned cloud groups, and do not require PAM account creation.
- The provisioning result is logged and communicated back to ServiceNow.

This process ensures automated and secure provisioning of privileged accounts while adhering to organizational policies and role-based access control.