ServiceNow Project Report

Title: Prevent User Deletion if Assigned to an Incident

Abstract

In an IT Service Management (ITSM) environment, user records are critical for maintaining accountability and tracking incident ownership. Deleting a user who is still assigned to open incidents can result in broken references, data loss, and workflow disruption.

This project implements a safeguard within ServiceNow that prevents the deletion of users actively assigned to incidents that are not yet closed, ensuring data integrity and operational reliability.

---

1. Introduction

Overview:
ServiceNow is a cloud-based ITSM platform used to automate workflows, manage incidents, and improve service delivery. Administrators can accidentally delete users who are still assigned to open incidents, which can break data links and disrupt ongoing tasks.

Objective:

- Prevent deletion of users with active incident assignments.
- Maintain accountability and data integrity.
- Provide user-friendly alerts for administrators.

2. Problem Statement

Currently, the system allows administrators to delete users even if they are assigned to open incidents. This leads to:

- Broken references in the incident table
- Loss of accountability
- Workflow disruptions

The project aims to implement a **Business Rule** that blocks user deletion when unresolved incidents exist and displays an informative message.

3. System Design

Process Flow:

1. Administrator attempts to delete a user.
2. A Business Rule triggers **before deletion**.
3. The system checks the incident table for open incidents assigned to the user.
4. If active incidents are found:
   - Deletion is aborted
   - An error message is displayed
5. If no active incidents exist, deletion proceeds normally.

Logical Architecture:

- **Tables Involved:**
  - `sys_user` – stores user records
  - `incident` – stores incidents assigned to users
- **Relationship:** Incidents reference users via the `assigned_to` field.

4. Implementation

Steps:

1. Navigate to **System Definition** → **Business Rules**.
2. Create a new Business Rule with the following configuration:
   - **Name:** Prevent User Deletion if Assigned to an Incident
   - **Table:**`sys_user`
   - **When to run:** Before Delete

        - **Active:** True

3. Add the following script:

```
(function executeRule(current, previous /*null when async*/) {

    // Query for active incidents assigned to this user
    var inc = new GlideRecord('incident');
    inc.addQuery('assigned_to', current.sys_id);
    inc.addQuery('state', '!=', '7'); // 7 = Closed
    inc.query();

    // Prevent deletion if any active incidents exist
    if (inc.hasNext()) {
        gs.addErrorMessage('Cannot delete user: Active incidents are
still assigned to this user.');
        current.setAbortAction(true);
    }

})(current, previous);
```

---

5. Testing and Results

Test Scenarios:

- **User with active incidents:** Deletion is blocked; error message displayed.
- **User with only closed incidents:** Deletion allowed.
- **User with no incidents:** Deletion allowed.

Result:
The Business Rule works as expected in all cases, maintaining system integrity and preventing accidental deletion.

---

6. Advantages

- Ensures **data integrity** and consistency.
- Prevents accidental deletion of critical users.
- Enhances accountability for incident ownership.
- Easy to configure and maintain.

---

7. Limitations

- Only checks incidents; other tables like Problem or Change are not included.
- Requires manual reassignment of open incidents before deletion.

---

8. Future Enhancements

- Extend validation to Problem, Change, and Task modules.
- Add automated reassignment of open incidents before deletion.
- Create a dashboard showing users blocked from deletion with their assigned incidents.

---

9. Conclusion

This project successfully implements a safeguard in ServiceNow to prevent the deletion of users who are assigned to open incidents. By using a **Before Delete Business Rule**, the system maintains data integrity, ensures accountability, and supports uninterrupted ITSM operations.

---

10. References

- ServiceNow Documentation: https://docs.servicenow.com
- ITIL v4 Guidelines for Incident Management
- ServiceNow Developer Community Articles