Bidirectional Sync between SNOW & Sentinel.

Integration of Microsoft Sentinel with ServiceNow in Domain Separation

Table of Contents:

1. Introduction
2. Prerequisites
3. Architecture Overview in ServiceNow
4. Configuration Business Rules in ServiceNow
   1. Incident Creation
   2. Incident Update
   3. Incident Resolution

Introduction

This document provides a detailed guide to integrating Microsoft Sentinel with ServiceNow in a domain-separated environment. Domain separation in ServiceNow ensures data security and segregation across multiple tenants.

2. Prerequisites

ServiceNow:

* ServiceNow instance with domain separation enabled.
* Create a user for Microsoft Sentinel on ServiceNow.
* Appropriate roles and permissions for integration

General:

* Network connectivity between Microsoft Sentinel and ServiceNow.
* Get the end-point URL from Sentinel.

3. Architecture Overview in ServiceNow

The integration involves Microsoft Sentinel sending security incidents to ServiceNow. The incidents will be processed and recorded as tickets in ServiceNow within their respective domains using business rules.

* 1. Configuration Business Rules in ServiceNow

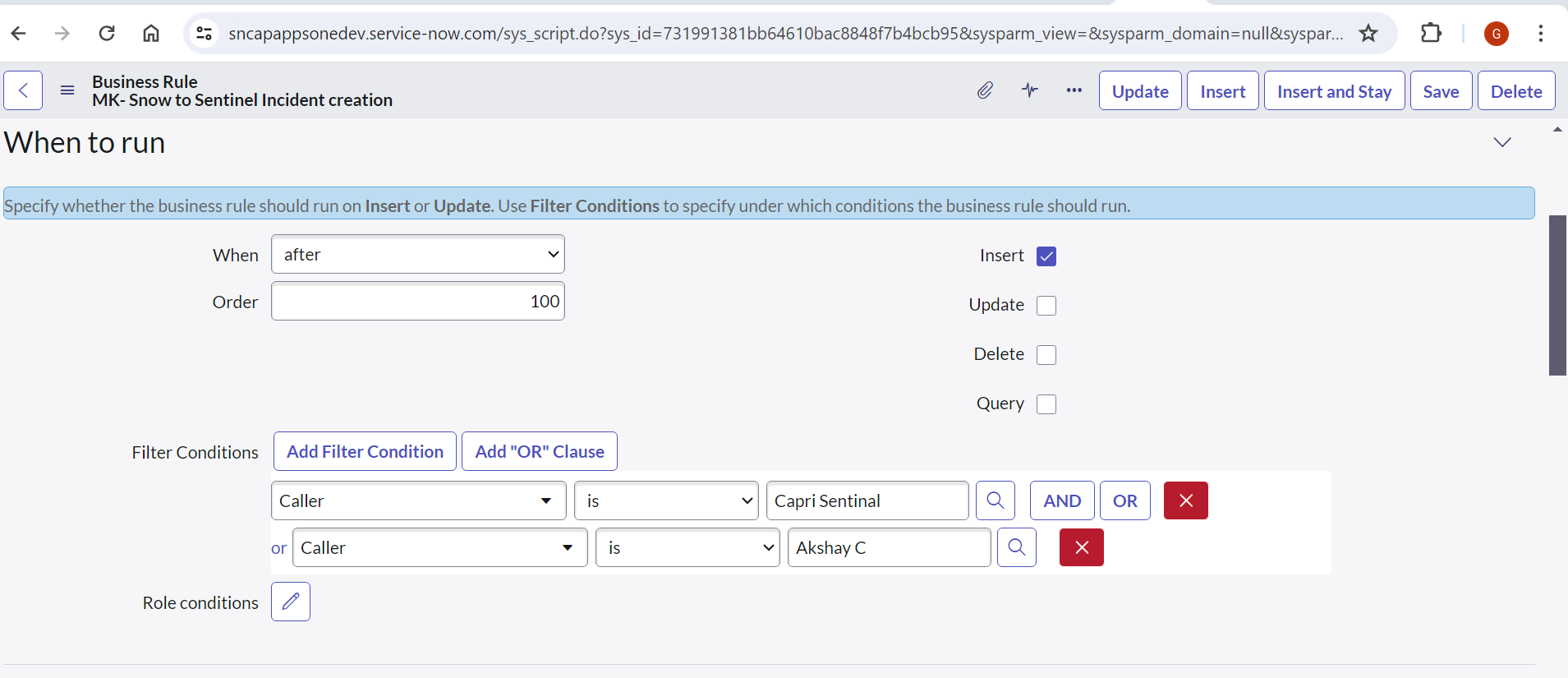
We created 3 Business Rules for this bidirectional sync between Snow & Sentinel.  
1) Insertion.

2) Update

3) Resolution

### 1. Incident Creation

1. Navigate to Business Rules:
   * Navigate to System Definition > Business Rules.
   * Click on New to create a new business rule.



This is the script for insertion.

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

    try {

        var CorrelationId = String(current.correlation\_id);

        var IncidentNumber = String(current.number);

        var Description = String(current.description);

        var request = new sn\_ws.RESTMessageV2();

        var requestBody = {

            "number": CorrelationId,

            "ResultNumber": IncidentNumber,

            "Description": Description

        };

        request.setRequestBody(JSON.stringify(requestBody));

        request.setEndpoint('https://prod2-26.germanywestcentral.logic.azure.com:443/workflows/05c10435eb984d699a3e8c19e00b650f/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=Tj9CxZwTnM\_BVnMAKsD9UkYN\_enl2xB0QEIDsLpNXac');

        request.setHttpMethod('POST');

        request.setRequestHeader("Accept", "application/json");

        request.setRequestHeader('Content-Type', 'application/json');

        var response = request.execute();

        var responseBody = response.getBody();

        var httpStatus = response.getStatusCode();

        var parsedData = JSON.parse(responseBody);

        gs.log(response.getBody());

    } catch (ex) {

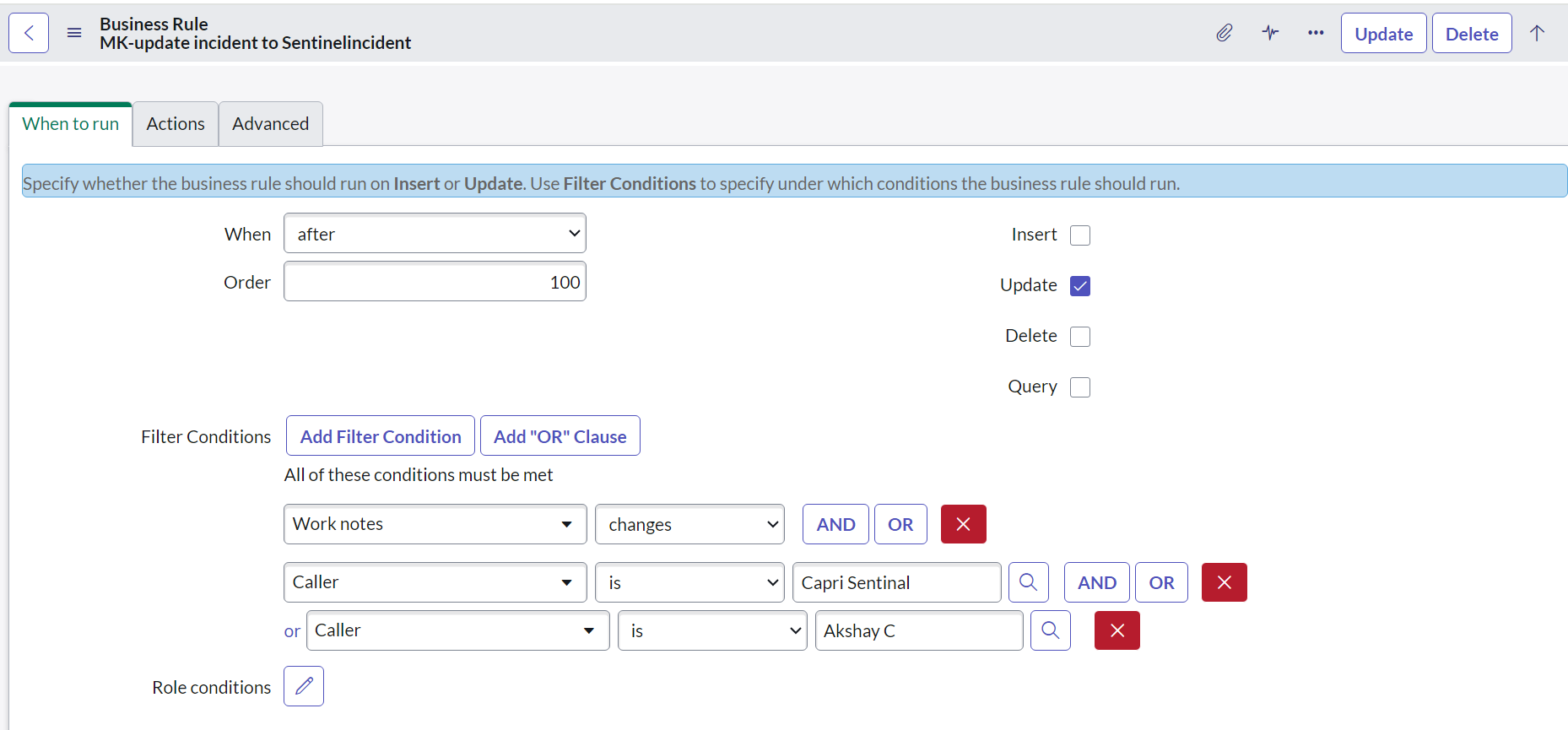
        var message = ex.message;

        gs.log('Message'+message);

    }

})(current, previous);

2. For update the incident - MK-update incident to Sentinel incident



This is the script for the update.

{

    var CorrelationId = String(current.correlation\_id);

    var IncidentNumber = String(current.number);

    var WorkNotes = String(current.work\_notes.getJournalEntry(1));

    var request = new sn\_ws.RESTMessageV2();

    var requestBody = {

        "number": CorrelationId,

        "ResultNumber": IncidentNumber,

        "Sentinel Incident Comment Section": WorkNotes

    };

    request.setRequestBody(JSON.stringify(requestBody));

    request.setEndpoint('https://prod2-26.germanywestcentral.logic.azure.com:443/workflows/05c10435eb984d699a3e8c19e00b650f/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=Tj9CxZwTnM\_BVnMAKsD9UkYN\_enl2xB0QEIDsLpNXac');

    request.setHttpMethod('POST');

    request.setRequestHeader("Accept", "application/json");

    request.setRequestHeader('Content-Type', 'application/json');

    var response = request.execute();

    var responseBody = response.getBody();

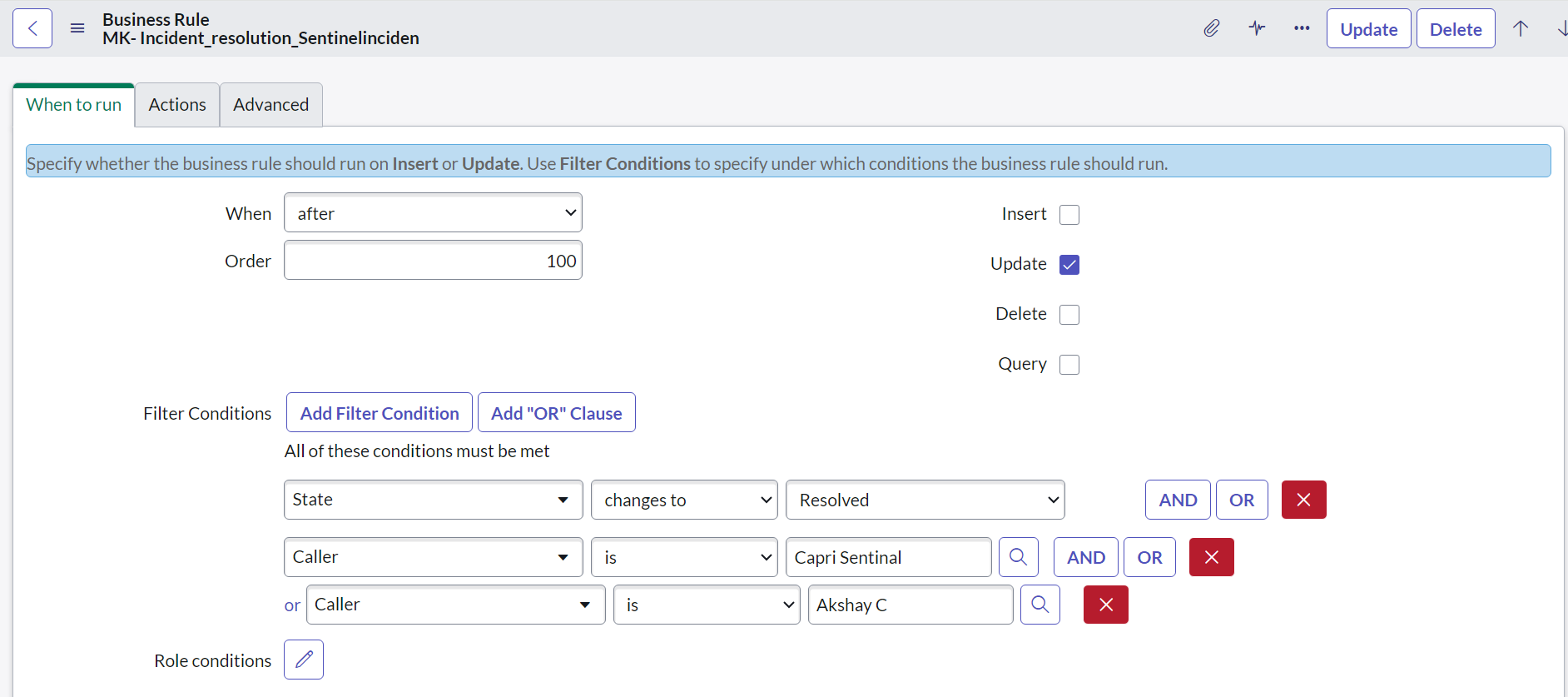
    var httpStatus = response.getStatusCode();

    var parsedData = JSON.parse(responseBody);

    gs.log(response.getBody());

}

3. For resolution - MK- Incident\_resolution\_Sentinelincident



{

    var WorkNotes = String(current.work\_notes.getJournalEntry(1));

    var CloseNotes = String(current.close\_notes);

    var ResolvedBy = String(current.resolved\_by.getDisplayValue());

    // var Description = current.short\_description.replace(/(\r\n|\n|\r|['"])/gm, ", ");

    var CorrelationId = String(current.correlation\_id);

    var IncidentNumber = String(current.number);

    var IncidentStatus = String(current.incident\_state.getDisplayValue());

    var request = new sn\_ws.RESTMessageV2();

    var requestBody = {

        "number": CorrelationId,

        "ResultNumber": IncidentNumber,

        "Status": IncidentStatus,

        "Sentinel Incident Comment Section": WorkNotes,

        "ClassificationComment": CloseNotes,

        "ClosedBy": ResolvedBy

    };

    request.setRequestBody(JSON.stringify(requestBody));

    request.setEndpoint('https://prod2-26.germanywestcentral.logic.azure.com:443/workflows/05c10435eb984d699a3e8c19e00b650f/triggers/manual/paths/invoke?api-version=2016-10-01&sp=%2Ftriggers%2Fmanual%2Frun&sv=1.0&sig=Tj9CxZwTnM\_BVnMAKsD9UkYN\_enl2xB0QEIDsLpNXac');

    request.setHttpMethod('POST');

    request.setRequestHeader("Accept", "application/json");

    request.setRequestHeader('Content-Type', 'application/json');

    var response = request.execute();

    var responseBody = response.getBody();

    var httpStatus = response.getStatusCode();

    var parsedData = JSON.parse(responseBody);

    gs.log(response.getBody());

}