Script Use Cases

Dec 22, 2016 | Tags: script

The following use-cases involves setting Custom Rules and Scripts.

Note: Custom fields can be added from the Customization Menu -> Layouts -> Add Field.
Note: Custom rules can be added from the Customization Menu -> Rules -> Add Rule
Note: Custom scripts can be added from the Customization Menu -> Scripts -> Add Script

1. Set Incident Owner based on Incident Type

First, you will need to create an **Automatic Rule**. To do so, navigate to **Customization Settings -> Rules** and select **New Rule.** Set the **object type** to **Incident**. Next, select **Add New** under the **condition** section set the condition to: **Incident Type contain*s* Other**. Save the rule and navigate to the **Scripts** tab and select **New Script.** Set the object type to **Incident** and enter the following in the codebox:

```
if "Other" in incident.incident_type_ids:
  incident.owner_id = "gina@gmail.com"
```

Hit save, navigate back to the rule you created, select **Edit**, then click **Add New** in the **Ordered** section. Select **Run Script** and then select the script you created from the drop down menu.

This script will automatically set an incident's owner to **"gina@gmail.com"** if the incident type is equal to **Other.**

---

2. Set Incident members based on Incident Type

First, you will need to create an **Automatic Rule**. To do so, navigate to **Customization Settings -> Rules** and select **New Rule.** Set the **object type** to **Incident**. Next, select **Add New** under the **condition** section set the condition to: **Incident Type *contains* Other**. Save the rule and navigate to the **Scripts** tab and select **New Script.** Set the object type to **Incident** and enter the following in the codebox:

```
if "Other" in incident.incident_type_ids:
  incident.members= ["ak@gmail.com", "ak2@gmail.com"]
```

Hit save, navigate back to the rule you created, select **Edit**, then click **Add New** in the **Ordered** section. Select **Run Script** and then select the script you created from the drop down menu.

This script will now automatically set the members of an incident to: ak@gmailcom and ak2@gmail.com if an incident has type **Other**.

3. Set Incident Owner based on Incident Phase

Create an Automatic Rule on the Incident object. Set Phase is changed to Complete condition. Create following incident script and add to the rule. This will set incident owner to "ak2@gmail.com" if incident phaseis "Complete".

First, you will need to create an **Automatic Rule**. To do so, navigate to **Customization Settings -> Rules** and select **New Rule.** Set the **object type** to **Incident**. Next, select **Add New** under the **condition** section set the condition to: **Phase *is set to* Complete**. Save the rule and navigate to the **Scripts** tab and select **New Script.** Set the object type to **Incident** and enter the following in the codebox:

```
if incident.phase_id == 'Complete':
  incident.owner_id = ak2@gmail.com
```

Hit save, navigate back to the rule you created, select **Edit**, then click **Add New** in the **Ordered** section. Select **Run Script** and then select the script you created from the drop down menu.

This script will now automatically set the owner of an incident to "ak2@gmailcom" if that incident is in phase: **Complete.**

4. Conditionally required fields based on conditions/activities

First, you will need to create an **Automatic Rule**. To do so, navigate to **Customization Settings -> Rules** and select **New Rule.** Set the **object type** to **Incident**. Next, select **Add New** under the **condition** section set the condition to: **Incident Type *contains* Malware**. Save the rule and navigate to the **Scripts** tab and select **New Script.** Set the object type to **Incident** and enter the following in the codebox:

This will force you set "City" for incident when you create an incident whose type contains "Malware". Otherwise it will throw an error - The Rule 'setRequiredFieldBasedOnConditons' is unable to update the Incident 'JonIncident' because: Incident type for Malware needs city name

```
if ("Malware" in incident.incident_type_ids) and (not incident.city)  :
  helper.fail("Incident type for Malware needs city name")
```

Hit save, navigate back to the rule you created, select **Edit**, then click **Add New** in the **Ordered** section. Select **Run Script** and then select the script you created from the drop down menu.

This script will force you to set the **City** field when creating a new incident **IF** Malware is one of that incident's types. An error is thrown if the creation is attempted without a value for **City.** Note: the same field can be optional for certain incident types and mandatory for others.

5. Task notes visible on incident notes

This is an example script that will copy Task level notes and add them to the Incident Notes Widget.

Navigate to an incident and create a data table called "tasknotes" with three columns - "taskid"(Number), "taskname"(Text) and "notetext"(Text Area, Rich Text) and place it under incident Notes Tab section. The "notetext" holds the text of the task note. Create following task note script.

You can create a Menu Item rule on the Note object and run the rule on a task note. Ideally create an Automatic Rule that runs every time a task note is created, but currently the Automatic Rule option is not available for this case.

```
if task is not None:
  row = incident.addRow("tasknotes")
  row.taskid = task.id
  row.taskname = task.name
  row.notetext = note.text.content
```

6. Task attachments on incident attachments

A script that will show all Task Level Attachments in the Incident Level Attachments Widget or that will create a data table placed on the Incident level Attachments tab and hyperlink to all Task level attachments. The data table should have a column to capture the Task the attachment was attached to.

Navigate to an incident and create a data table called "taskattachments" with three columns - "taskid"(Number), "taskname"(Text) and "attachment"(Text Area, Rich Text) and place it under incident Attachments Tab section. The "attachment" column holds information on attachment name. Create following task attachment script. You can create a Menu Item rule on the Attachment object and run the rule on the task attachment. Ideally create an Automatic Rule that runs every time a task note is created, but currently the Automatic Rule option is not available for this case.

```
if attachment is not None:
  row = incident.addRow("taskattach")
  row.taskid = task.id
  row.taskname = task.name
  row.attachment = attachment.name
```

7. Orphaned incidents/auto add group

The following incident script checks members of an incident and if there is no group in the member list add "AKGroup" to the member list. Create an Automatic Rule on the Incident object without a condition and add the script to it.

```
#data structure of incident.members is not list, set and array
x = list(incident.members)
contains = False
for member in x:
  if "@" not in member:
    contains = True
    continue
if contains == False:
  x.append('AKGroup')
  incident.members = x
```

8. Calculated fields - Set Value of a Field based on the Value of another field

This following is an example script that will set the Value of a Field based on the Value of a previously selected Field Value.

Use Case: A Threat Score is made up of 3 threat variables values; Confidentiality/Integrity/Availability. If each of the three Fields are answered 1, then Threat Score == High. If all are answered 3 then Threat Score == Low.

Note: Custom fields can be added from the Customization Menu -> Layouts -> Add Field.
Note: Custom rules can be added from the Customization Menu -> Rules -> Add Rule

Note: Custom scripts can be added from the Customization Menu -> Scripts -> Add Script

Create three custom fields - "confidentiality", "integrity" and "availability" with "Select" type and add "1", "2" and "3" to each of them. Create a custom field called "threatscore" with "Select" type and add "Low", "Medium" and "High". Place all four custom fields under incident Detail Tab.

Create the following incident script and run on an incident:

```
if (incident.properties.confidentiality == '1') and
(incident.properties.integrity == '1') and (incident.properties.availability
== '1') : incident.properties.threatscore = 'High'
```

Next, Create an Automatic Rule with object type: **Incident** with any of the following conditions and select **Run Script** in the **Ordered** section and select your script from the dropdown menu.

1 Confidentiality is changed
2 Integrity is changed
3 Availability is changed

This will set the incident's "threat score" field to high in the event that any of those fields are changed.

The following examples require the use of custom scripts with the Resilient API and cannot be used with the built-in script editor of the web interface.

9. Time Stamps/User Stamps - Duration Task was open (date/time initiated vs. date/time closed)

Calculate duration from a task open to close.

On an incident, Create a custom data table with three columns - "taskid"(Number), "taskname"(Text) and "duration"(Number).

```
from java.util
import Date
if task.status == 'C':
 row = incident.addRow("taskduration")
 row.taskid = task.id
 row.taskname = task.name
 row.duration = (task.closed_date - task.init_date) / 60000 # in min
 task.description = str(task.closed_date - task.init_date)
```

Next, Create an Automatic Rule with object type: **Task** with the condition:
Status *is changed to* Closed
and select **Run Script** in the **Ordered** section and select your script from the dropdown menu.

10. Time Stamps/User Stamps - Duration Incident was active (date/time created vs. date/time closed)

Calculate duration since an incident open to close.

Create a custom field with Number type and name it "durationmin" and place it under incident Details Tab **(select Incident Tabs at the Layouts page and drag the custom field into Details)**.

Create following incident script:

```
from java.util import Date
now = Date()
incident.properties.durationmin = (now.time - incident.create_date) / 60000 #in min
```

Next, Create an Automatic Rule with object type: Incident with the following condition:
Status *is Changed to* Closed
and select Run Script in the Ordered section and select your script from the dropdown menu.

This script will now calculate the duration in minutes that an incident was open.

11. Time Stamps/User Stamps - Date/Time Incident last modified)

This script will capture an incident's last modified time and set it as a field.

Create a custom field with Text type and name it "lastmodified". Place it under incident Details Tab.
(select Incident Tabs at the Layouts page and drag the custom field into Details).

Create the following incident script:

```
from java.util
import Date now = Date()
incident.properties.lastmodified = str(now)
```

Next, create an Automatic Rule on the **Incident** object without a condition and add the script to the rule. When incident is being modified the time stamp is captured and set to the "lastmodified" field.

12. Milestones - When Incident Owner Changed

The following script creates a milestone for an incident when the owner is changed.

Create a custom field with "Text" type and name it "originalowner".

Create following incident script.

```
from java.util import Date
orig_owner = incident.owner_id
if incident.properties.originalowner and not incident.properties.originalowner:
  orig_owner = incident.properties.originalowner
incident.addMilestone('Incident Owner Changed', 'From: ' + orig_owner + ' To: ' +
incident.owner_id, Date())
incident.properties.originalowner = incident.owner_id
```

Create an Automatic Rule on the **Incident object** with the **condition** of: **Owner is changed**.
Under the **Ordered** section, select **Run Script** and select your script from the drop down menu.

This will create a Milestone whenever incident owner is changed.
The milestone title will be "Incident Owner Changed" and description "From: admin@co3sys.com To: ak@gmail.com ".
The date will be the milestone creation time.

13. Milestones - When Task Completed

The following script creates a milestone when a task is marked as **completed**.

Create the following **task** script.

```
from java.util import Date
if task.status == 'C':
  incident.addMilestone('Task Completed', 'Task ID: ' + str(task.id) + ' Task Name: ' + task.name,
Date())
log.info('test {}'.format(task.status))
```

Create an Automatic Rule on the Task object and set the condition to:
Status *is changed to* Closed.
Under the **Ordered** section, select **Run Script** and select your script from the drop down menu.

This will create a Milestone when a task is closed.

The milestone title will be "Task Completed"' and description "Task ID: 2251453 Task Name: MarcusTask". The date will be milestone creation time.