IBM Resilient

Security Orchestration, Automation and Response Platform

Carbon Black Protection Integration V1.0.2

Release Date: September 2019

Resilient functions simplify development of integrations by wrapping each activity into an individual workflow component. These components can be easily installed, then used and combined in Resilient workflows. The Resilient platform sends data to the function component that performs an activity then returns the results to the workflow. The results can be acted upon by scripts, rules, and workflow decision points to dynamically orchestrate the security incident response activities.

This guide describes the Carbon Black Protection Integration.

Overview

This integration consists of 12 functions which call various APIs to perform different actions, such as retrieving approval request details, updating approval requests and deleting files. It also contains a polling component to create incidents in the Resilient platform that correspond to approval requests in Carbon Black Protection.

Installation

You download the function package to a Resilient integration server, and from there you deploy the functions and components to a Resilient platform. These procedures are provided in the Resilient Integration Server Guide (PDF).

The functions included this package have the following requirements, which are above and beyond those listed in the *Resilient Integration Server Guide*.

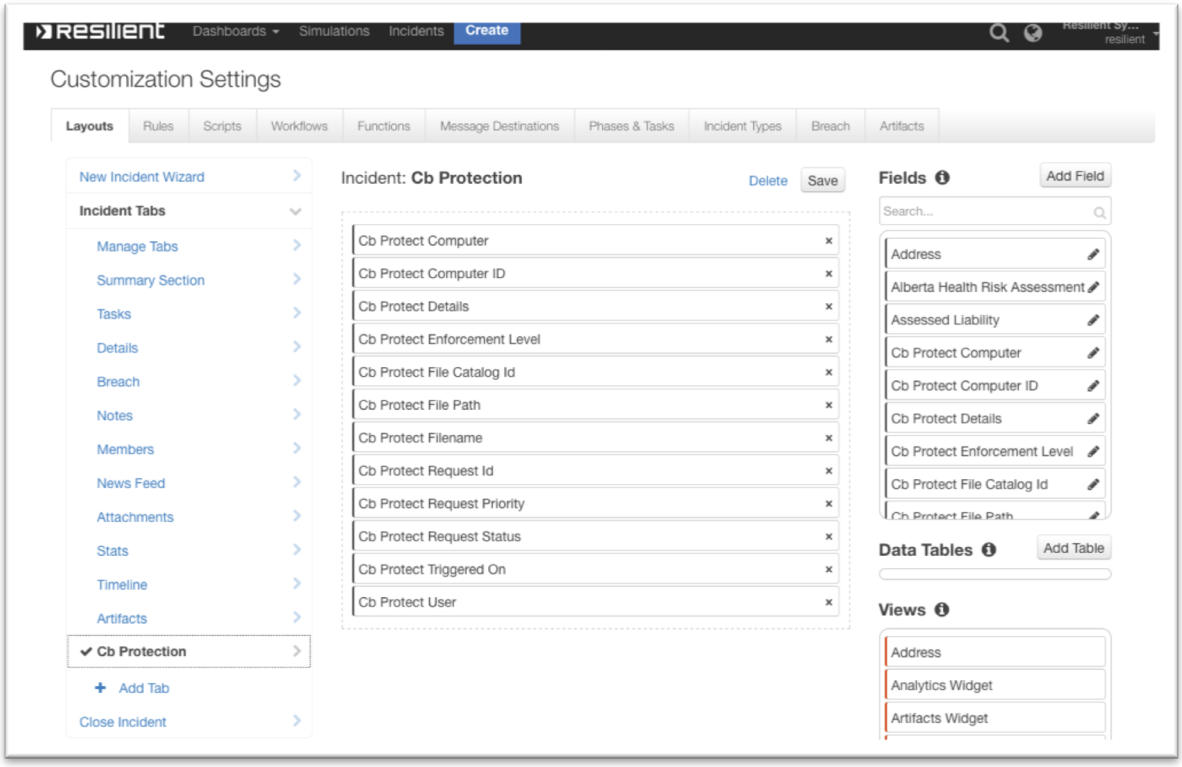
- Resilient platform is version 30 or later.
- Carbon Black Protection v8.1 or later.

After installing the package, Resilient Circuits creates a new section, [fn_cb_protection], in the app.config file. You need to edit the following settings in that section.

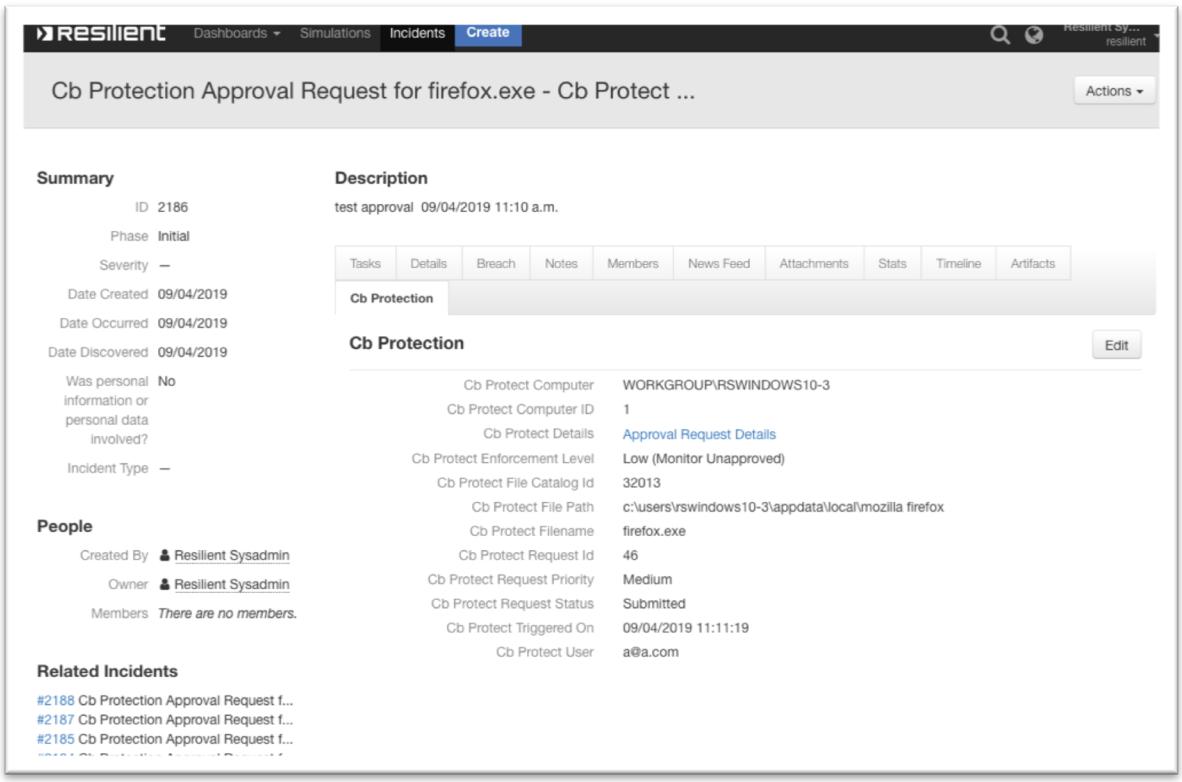
```
[fn_cb_protection]
# Name or IP address of your CbProtect server
server=10.200.1.1

# Access token issued by the CbProtect administrator
token= XXXX-XXXX-XXXX-XXXX

# If your CbProtect server has a self-signed TLS certificate, you cannot verify
it:
# verify_cert=false

# Interval (seconds) for automatic escalation of approval requests, set 0 to
disable
# Suggest 300 as a starting point, which will check CbProtect every 5 minutes
escalation_interval=0

# Optional: query for which requests to escalate; default is to escalate all
open approval requests
# escalation_query=resolution:0

# Optional: path to a custom template file for the escalated incident
# template_tile=/usr/integration/bit9_escalation.jinja

# Optional: set this to only escalate a single request ID, e.g. when testing a
custom template
# test_single_request=999
```

Create a Custom Layout

In order to view the status of the Carbon Black Protection integration, go to the Layout tab in Resilient and create an Incident Tab named Cb Protection. Drag the Cb Protect incident fields from the Fields column on the right to the Incident: CbProtection tab column in the middle of the page and then hit the Save button.

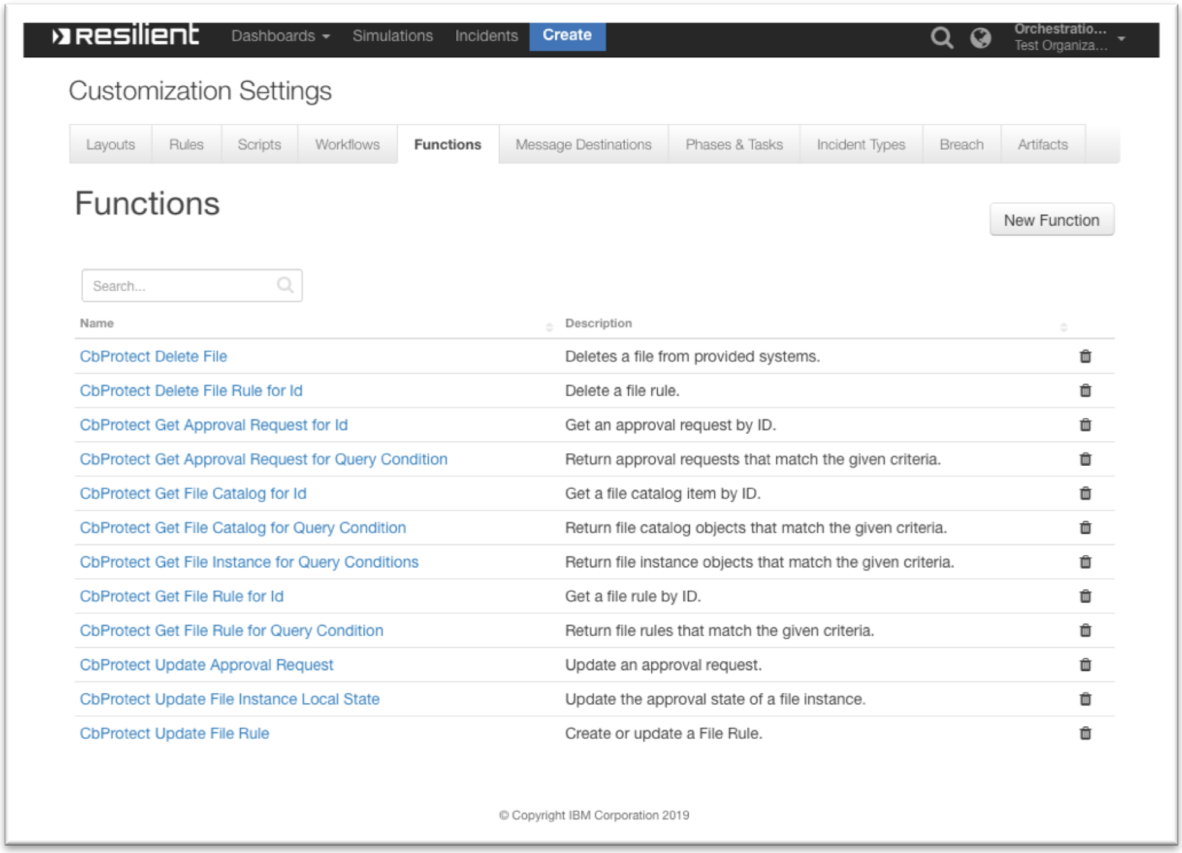


Now you can go to the Cb Protection tab and view that status of the Approval and Ban requests.



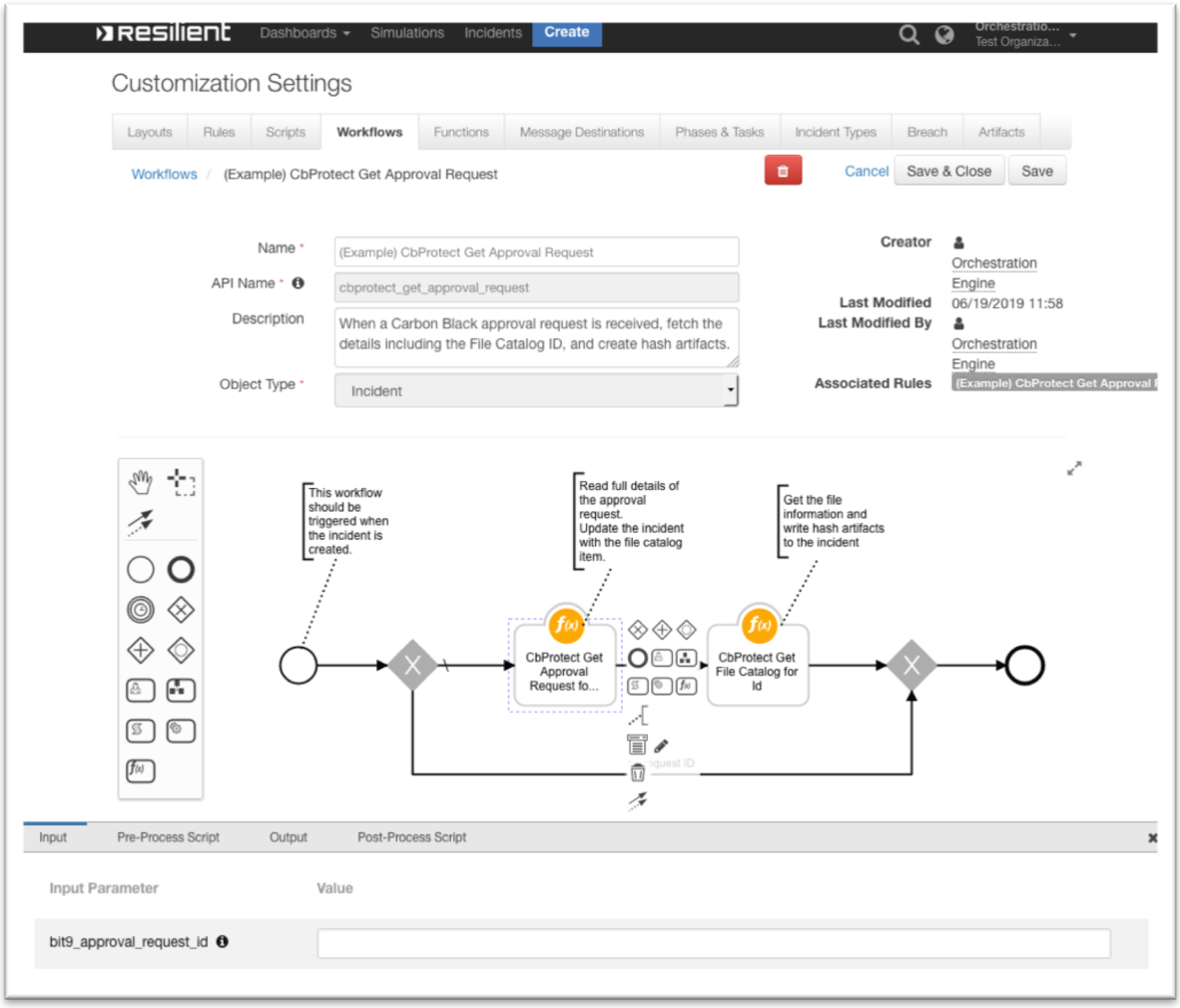
Function Descriptions

Once the function package deploys the functions, you can view them in the Resilient platform Functions tab, as shown below. The package also includes example workflows and rules that show how the functions can be used. You can copy these workflows and rules for your own needs.



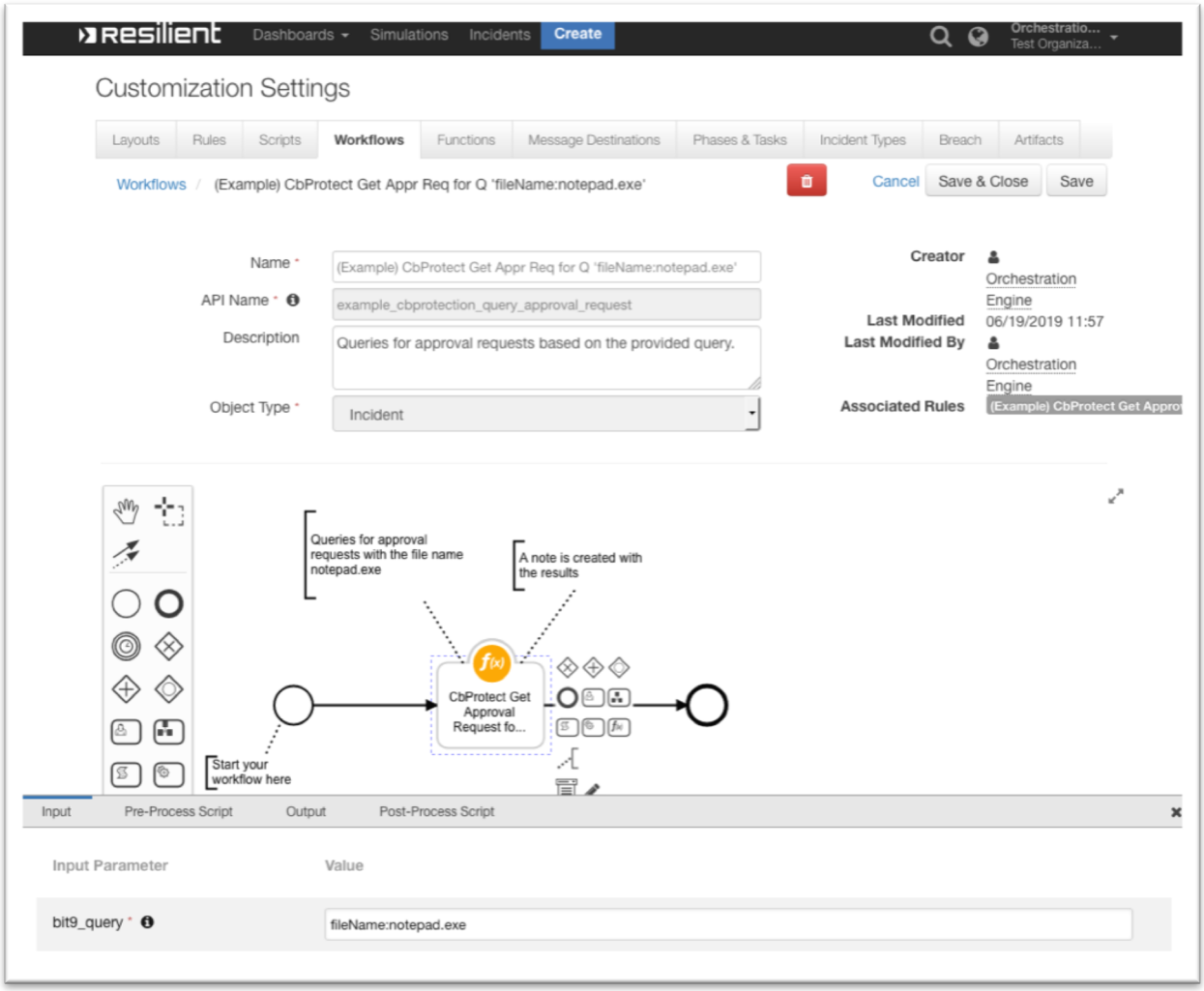
bit9_approval_request_get: CbProtect Get Approval Request for Id

Given an approval request's ID, the function returns the details of the approval request. The function takes one input, bit9_approval-request_id, which is a number. The following is an example of this function in the (Example) CbProtect Get Approval request workflow.



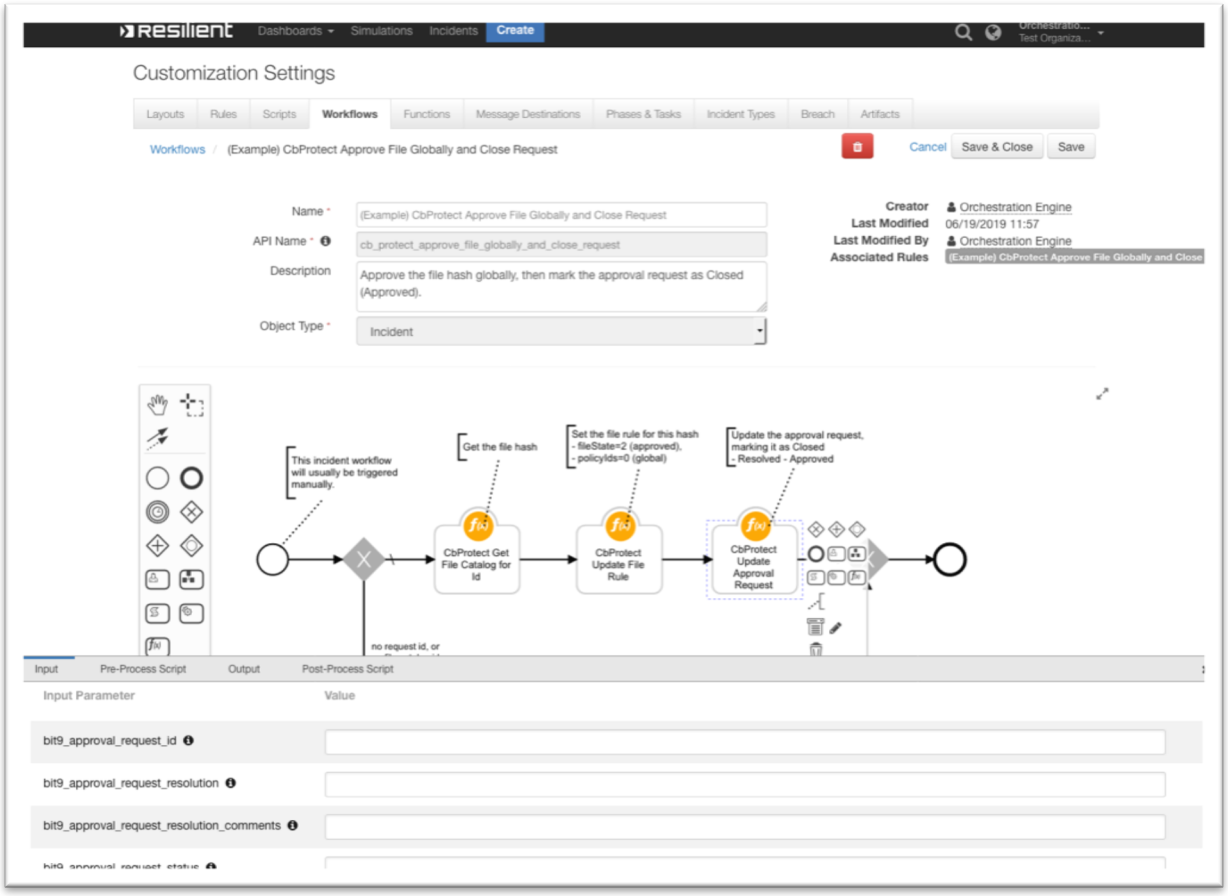
bit9_approval_request_query: CbProtect Get Approval Request for Query Condition

This function takes one input, bit9_query which is a query string, and returns the approval requests that match the given query condition. The following is an example of this function in the (Example) CbProtect Get Appr Req for Q 'fileName:notepad.exe' workflow. You can set a different query condition following the guidelines https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#query-condition and review the all approval request properties to query here https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#approvalrequest. fileName:notepad.exe represents name of the file on the agent.



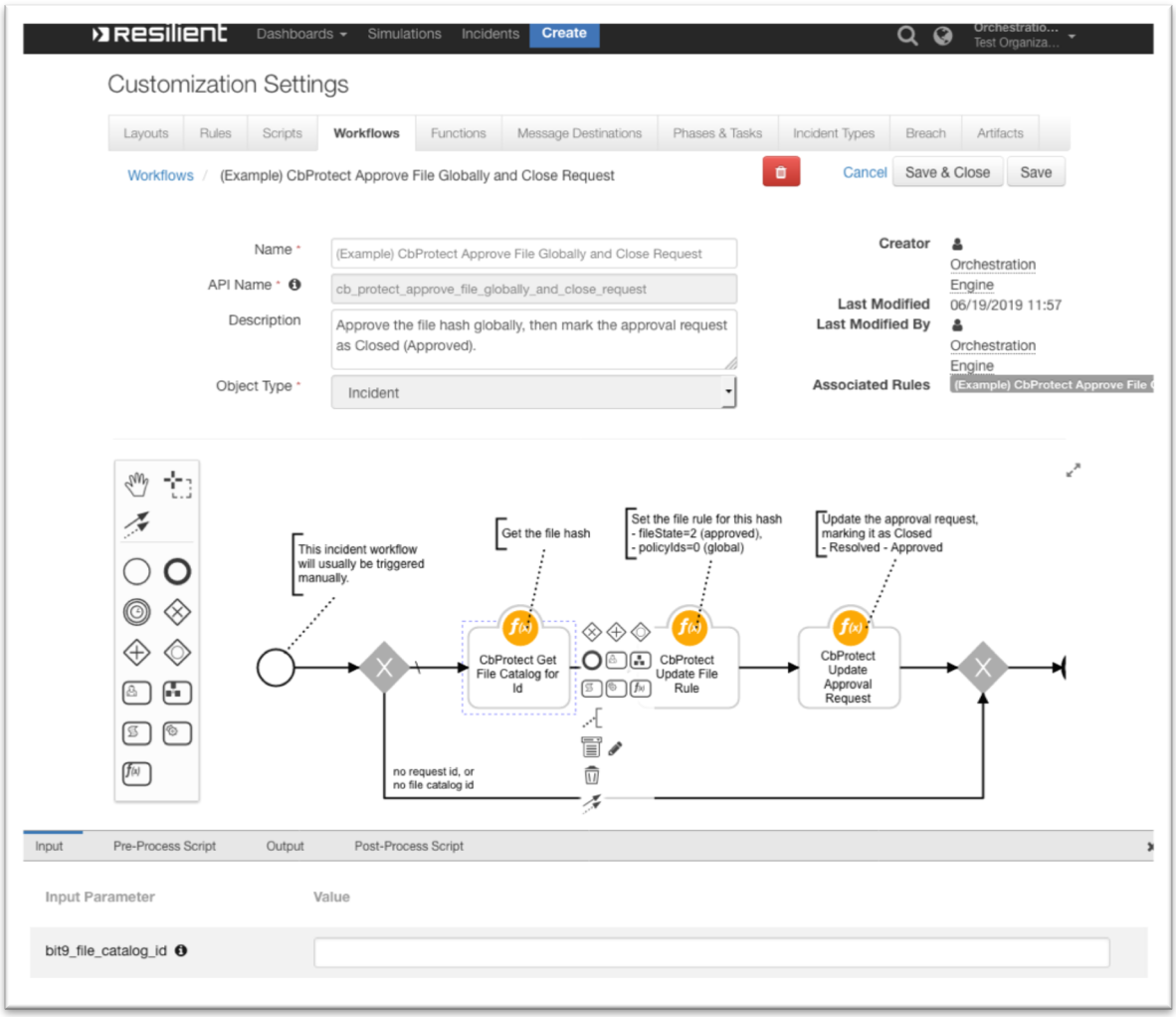
bit9_approval_request_update: CbProtect Update Approval Request

This function accepts as input a request ID, approval request resolution, comments, and status. With these, it updates an approval request. The following is an example of this function in a workflow:



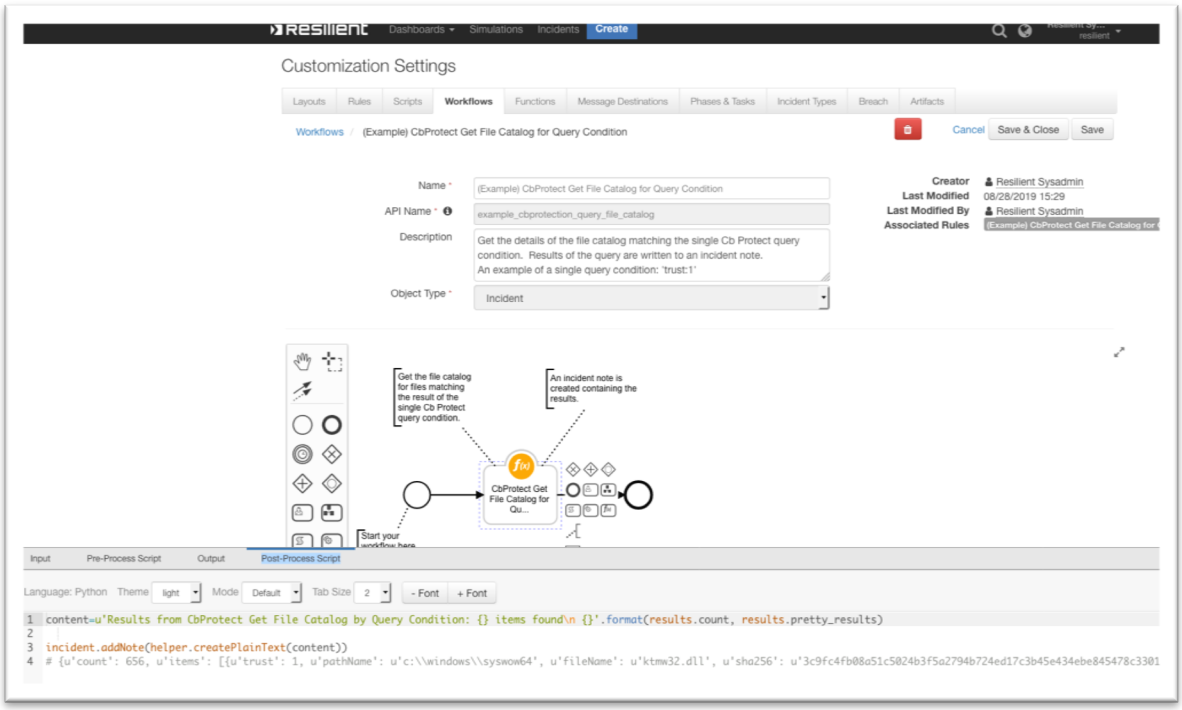
bit9_file_catalog_get: CbProtect Get File Catalog for Id

Returns the file catalog details based on the catalog ID provided. The following is an example of this function in the (Example) CbProtect Approve File Globally and Close Request workflow:



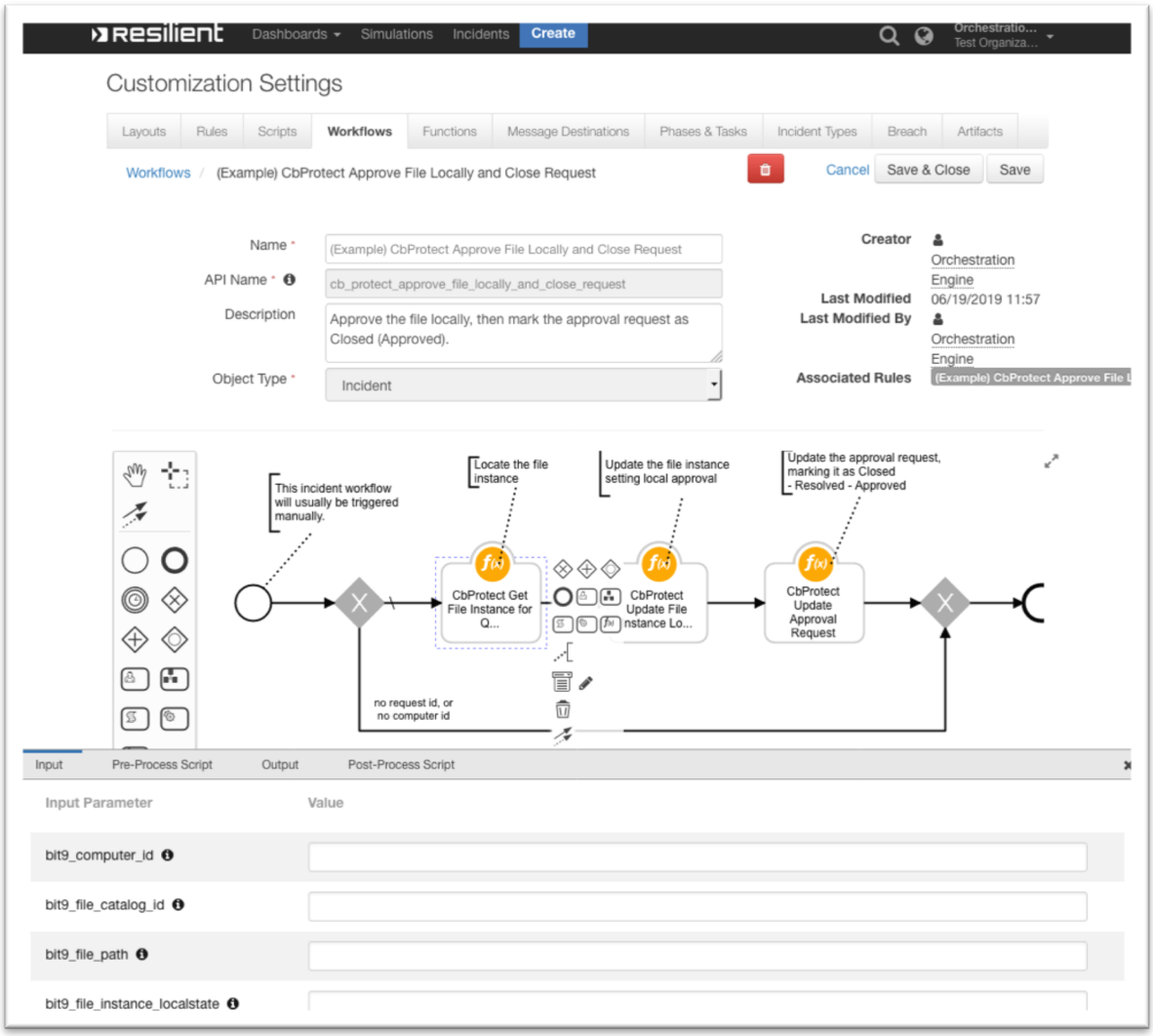
bit9_file_catalog_query: Cbprotect Get File Catalog for Query Condition

Returns file catalogs and their details from a provided query string. The following is an example of this function in (Example) CbProtect Get File Catalog for Query Condition workflow. You can set a different query condition following the guidelines https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#query-condition and review the all file catalog properties to query here https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#filecatalog.



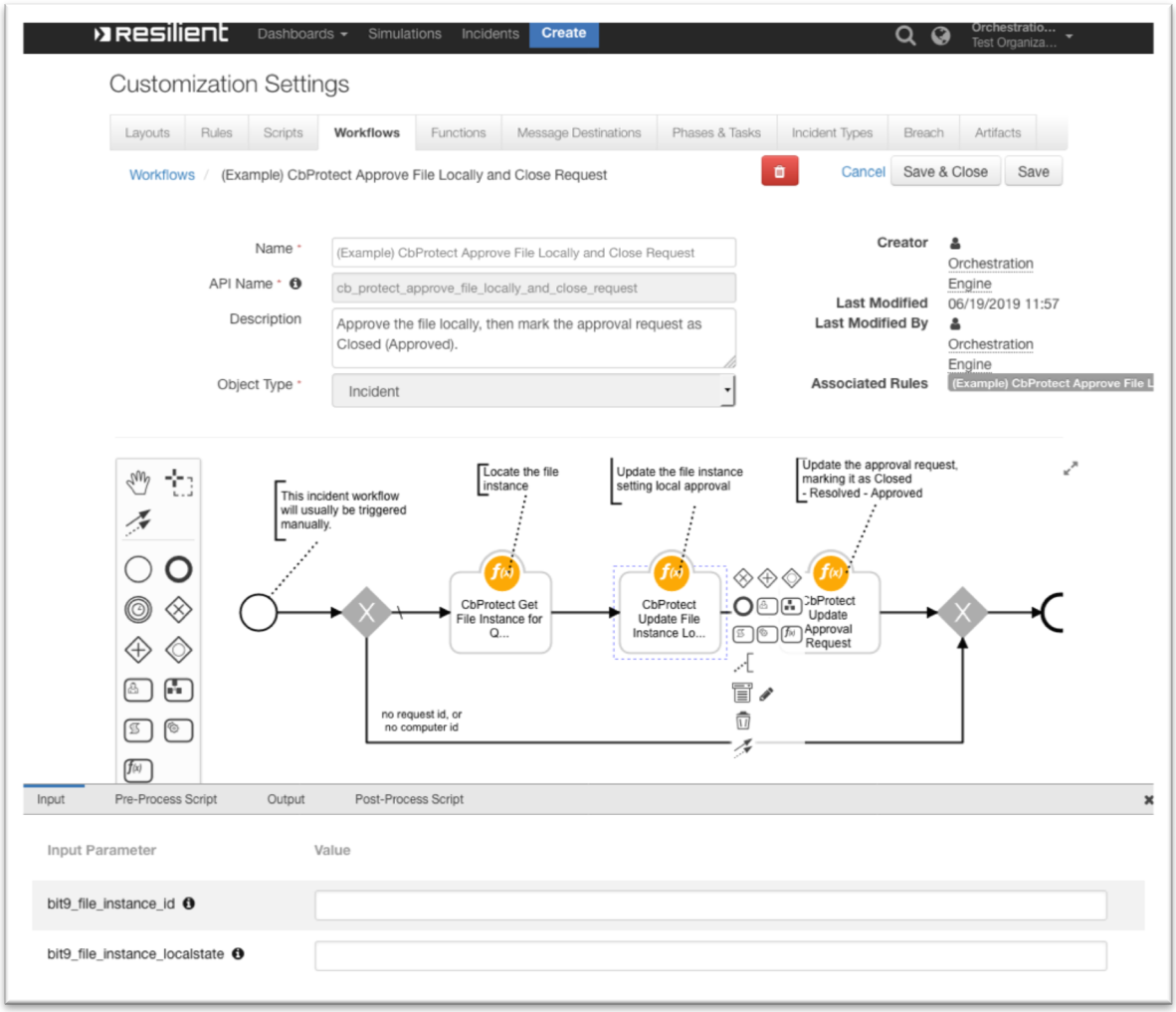
bit9_file_instance_query: CbProtect Get File Instance for Query Conditions

Returns file instance objects that match the given criteria from the inputs. The following is an example of this function in the (Example) CbProtect Approve File Locally and Close Request workflow:



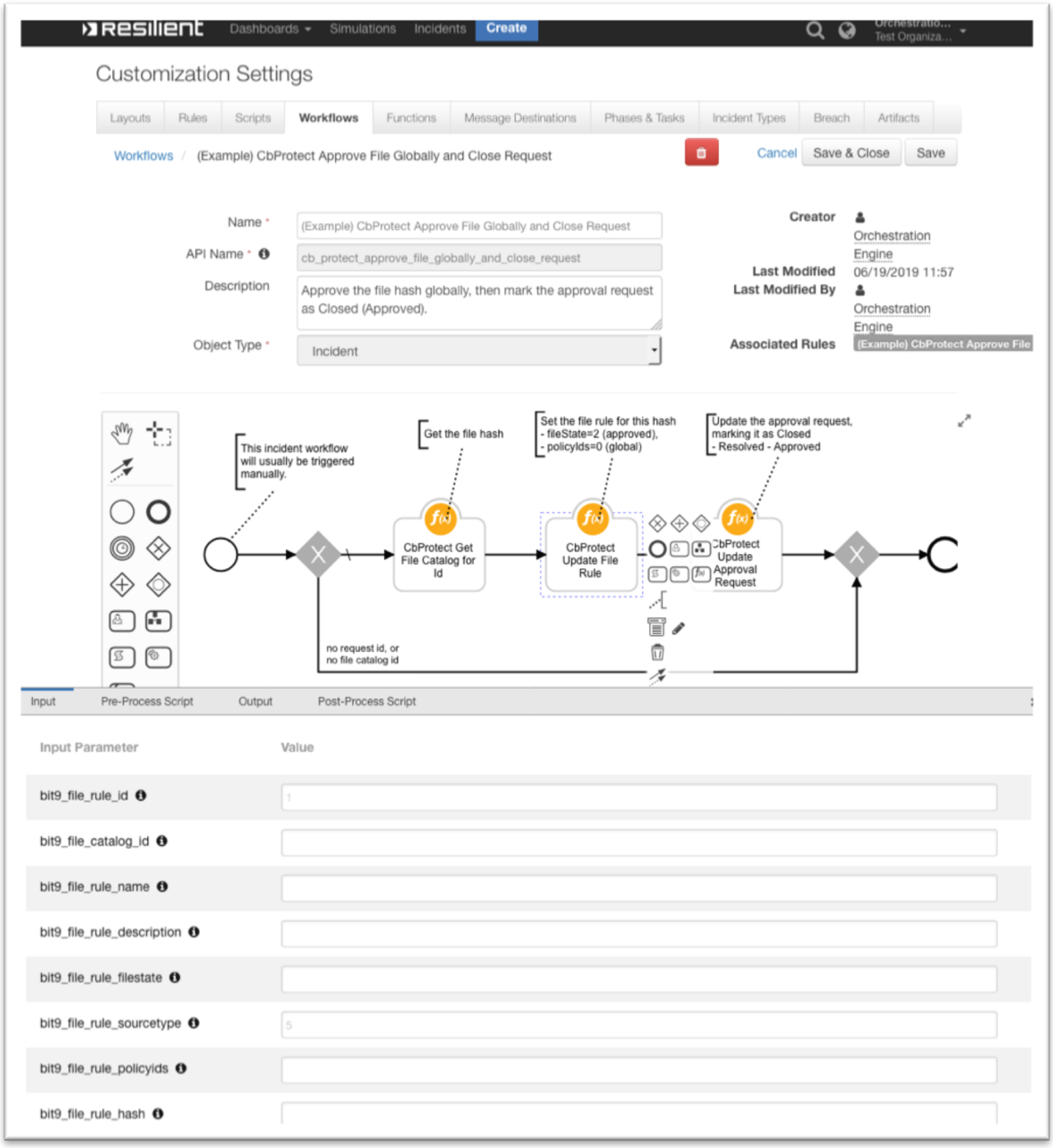
bit9_file_instance_update: CbProtect Update File Instance Local State

Updates a file instance's local approval/banned setting. This function has inputs for the file instance ID and the local state (for example, approved = 2). The following includes an example of this function in the (Example) CbProtect Approve File Locally and Close Request workflow:



bit9_file_rule_update: CbProtect Update File Rule

This function updates a file rule in Carbon Black based on the data passed as inputs. The following is an example of this function in the (Example) CbProtect Approve File Globally and Close Request workflow:



bit9_file_delete: Cb Protect Delete File

This function deletes a file by hash or filename using a Cb Protection API call based on the data passed as inputs. The Cb Protection API call that is used in the function is a beta solution and should not be used in production.

Carbon Black Protection Resilient Polling Component

This integration contains a polling component that automatically escalates approval requests into the Resilient platform. To enable this feature, the `escalation_interval` variable in the app.config file must be set to an integer greater than 0. This integer represents the interval in number of seconds for the automatic escalation of approval requests. It is recommended to start at 300, which checks every 5 mins.

You can also set optional values, such as `escalation_query`, which escalates approval requests that match the query; if not set, it defaults to all open approval requests. In addition, you can set `template_file` to the location of a custom jinja template file; if not set, the default template file is used. To create your own custom jinja file, you should use the default jinja file as a reference. This file can be found when expanding the package in the following directory:

`fn_cb_protection-<version#>/fn_cb_protection/data/`