

IBM Resilient



Security Orchestration, Automation and Response Platform

Carbon Black Protection Integration V1.0.2

Release Date: July 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

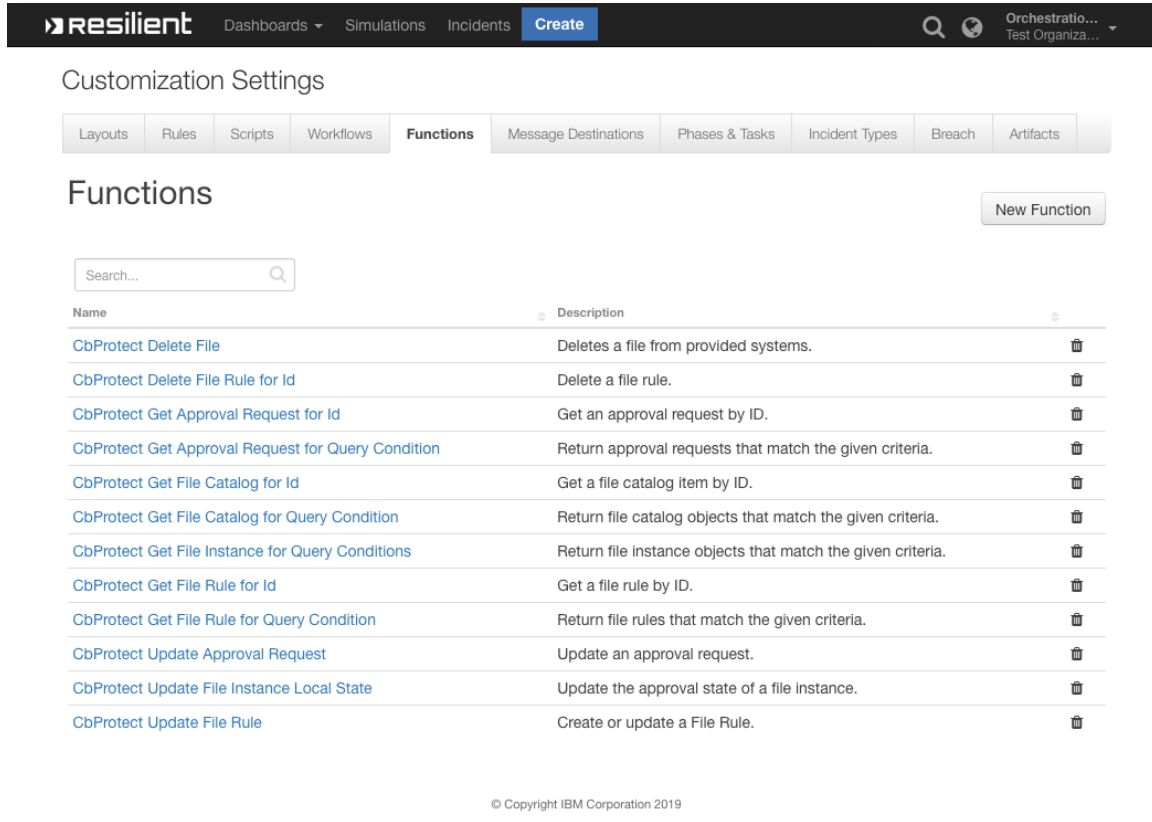
# Optional: query for which requests to escalate; default is to escalate all
open approval requests
# escalation_query=resolution:0

# 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
```

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.



The screenshot shows the Resilient platform interface. The top navigation bar includes the Resilient logo, a search icon, and a dropdown menu for 'Orchestration... Test Organiza...'. The main navigation bar has tabs for 'Dashboards', 'Simulations', 'Incidents', 'Create', 'Message Destinations', 'Phases & Tasks', 'Incident Types', 'Breach', and 'Artifacts'. The 'Functions' tab is selected, displaying a 'Customization Settings' header and a 'Functions' section. A 'New Function' button is in the top right. A search bar is located above a table of functions. The table has two columns: 'Name' and 'Description'. It lists 13 functions, each with a trash icon for deletion. The functions are all related to CbProtect operations.

Name	Description
CbProtect Delete File	Deletes a file from provided systems.
CbProtect Delete File Rule for Id	Delete a file rule.
CbProtect Get Approval Request for Id	Get an approval request by ID.
CbProtect Get Approval Request for Query Condition	Return approval requests that match the given criteria.
CbProtect Get File Catalog for Id	Get a file catalog item by ID.
CbProtect Get File Catalog for Query Condition	Return file catalog objects that match the given criteria.
CbProtect Get File Instance for Query Conditions	Return file instance objects that match the given criteria.
CbProtect Get File Rule for Id	Get a file rule by ID.
CbProtect Get File Rule for Query Condition	Return file rules that match the given criteria.
CbProtect Update Approval Request	Update an approval request.
CbProtect Update File Instance Local State	Update the approval state of a file instance.
CbProtect Update File Rule	Create or update a File Rule.

© Copyright IBM Corporation 2019

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.

The screenshot displays the Resilient platform interface for customizing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and a 'Create' button. The 'Customization Settings' section is active, showing tabs for 'Layouts', 'Rules', 'Scripts', 'Workflows', 'Functions', 'Message Destinations', 'Phases & Tasks', 'Incident Types', 'Breach', and 'Artifacts'. The 'Workflows' tab is selected, and the workflow '(Example) CbProtect Get Approval Request' is being edited.

Workflow Details:

- Name:** (Example) CbProtect Get Approval Request
- API Name:** cbprotect_get_approval_request
- Description:** When a Carbon Black approval request is received, fetch the details including the File Catalog ID, and create hash artifacts.
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:58
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Get Approval Request

Workflow Diagram:

The workflow diagram shows a sequence of steps:

- Start:** A circle icon representing the start of the workflow.
- Decision:** A diamond icon with an 'X' inside, indicating a decision point.
- Function:** A box labeled 'CbProtect Get Approval Request for Id' with a 'f(x)' icon. A callout box explains: 'Read full details of the approval request. Update the incident with the file catalog item.'
- Function:** A box labeled 'CbProtect Get File Catalog for Id' with a 'f(x)' icon. A callout box explains: 'Get the file information and write hash artifacts to the incident.'
- Decision:** A diamond icon with an 'X' inside, indicating a decision point.
- End:** A circle icon representing the end of the workflow.

Input Parameters:

Input Parameter	Value
bit9_approval_request_id	

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.

The screenshot shows the 'Customization Settings' page in the Resilient interface. The 'Workflows' tab is selected, showing a workflow named '(Example) CbProtect Get Appr Req for Q \'fileName:notepad.exe\''. The workflow details include:

- Name:** (Example) CbProtect Get Appr Req for Q 'fileName:notepad.exe'
- API Name:** example_cbprotection_query_approval_request
- Description:** Queries for approval requests based on the provided query.
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:57
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Get Approv...

The workflow diagram shows a sequence of steps:

- Start your workflow here** (Start node)
- Queries for approval requests with the file name notepad.exe** (Function node: CbProtect Get Approval Request fo...)
- A note is created with the results** (Note node)
- End node** (Circle node)

The workflow is configured with the following input parameter:

Input Parameter	Value
bit9_query	fileName:notepad.exe

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:

The screenshot displays the Resilient platform interface for customizing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and a 'Create' button. The 'Customization Settings' section is active, showing tabs for 'Layouts', 'Rules', 'Scripts', 'Workflows', 'Functions', 'Message Destinations', 'Phases & Tasks', 'Incident Types', 'Breach', and 'Artifacts'. The 'Workflows' tab is selected, and the specific workflow '(Example) CbProtect Approve File Globally and Close Request' is being edited.

The workflow details include:

- Name:** (Example) CbProtect Approve File Globally and Close Request
- API Name:** cb_protect_approve_file_globally_and_close_request
- Description:** Approve the file hash globally, then mark the approval request as Closed (Approved).
- Object Type:** Incident

Metadata on the right indicates the workflow was created by the 'Orchestration Engine' on 06/19/2019 at 11:57.

The workflow diagram shows a sequence of steps:

- Start node (circle) leading to a decision diamond.
- Decision diamond: 'no request id, or'. If true, it leads to an end node (circle). If false, it proceeds to the next step.
- Step 1: 'CbProtect Get File Catalog for Id' (function icon).
- Step 2: 'CbProtect Update File Rule' (function icon).
- Step 3: 'CbProtect Update Approval Request' (function icon).
- End node (circle).

Annotations for the steps:

- Step 1: 'Get the file hash'.
- Step 2: 'Set the file rule for this hash - fileState=2 (approved), - policyId=0 (global)'.
- Step 3: 'Update the approval request, marking it as Closed - Resolved - Approved'.

A note at the start of the workflow states: 'This incident workflow will usually be triggered manually.'

Below the diagram, the 'Input' section lists parameters for the workflow:

Input Parameter	Value
bit9_approval_request_id	
bit9_approval_request_resolution	
bit9_approval_request_resolution_comments	
bit9_approval_request_status	

bit9_file_delete: CbProtect Delete File

This function deletes a file from one or all computers using Carbon Black Protection. Set the bit9_file_action input to delete by file hash or file name (choose file hash to set catalog ID). Set the bit9_computer_id input for a specific computer, or use "0" to select all computers. Then, depending on the chosen file action, set the catalog ID, file hash, or file name. The following is an example of this function in the (Example) CbProtect Delete File By Hash workflow:

The screenshot displays the Resilient interface for customizing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and 'Create'. The 'Create' tab is active, showing a 'Customization Settings' page for the workflow '(Example) CbProtect Delete File By Hash'.

Workflow Details:

- Name:** (Example) CbProtect Delete File By Hash
- API Name:** example_cbprotect_delete_file
- Description:** Deletes a file by hash of type SHA1, SHA256 and MD5 from all systems.
- Object Type:** Artifact
- Creator:** Orchestration Engine
- Last Modified:** 06/20/2019 16:24
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Delete File By

Workflow Diagram:

The diagram shows a workflow starting with a 'Start your workflow here' node, followed by a 'CbProtect Delete File' function (labeled 'f(n)'). The function is annotated with 'Update inputs with the hash value for the file you wish to delete' and 'A note is created with the results'. The workflow ends with a 'Note' node.

Input Parameters:

Input Parameter	Value
bit9_file_action	DeleteFileByHash
bit9_computer_id	
bit9_file_catalog_id	
bit9_file_hash	
bit9_file_name	

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:

The screenshot displays the Resilient console interface for editing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and 'Create'. The 'Customization Settings' section is active, showing the workflow details for '(Example) CbProtect Approve File Globally and Close Request'.

Workflow Details:

- Name:** (Example) CbProtect Approve File Globally and Close Request
- API Name:** cb_protect_approve_file_globally_and_close_request
- Description:** Approve the file hash globally, then mark the approval request as Closed (Approved).
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:57
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Approve File Globally and Close Request

Workflow Diagram:

The workflow diagram illustrates the process flow:

- Start:** A circle icon representing the start of the workflow.
- Decision:** A diamond icon with an 'X' that checks for 'no request id, or no file catalog id'. If true, it leads to an end point.
- Function:** A rounded rectangle icon labeled 'CbProtect Get File Catalog for Id' with a callout 'Get the file hash'.
- Function:** A rounded rectangle icon labeled 'CbProtect Update File Rule' with a callout 'Set the file rule for this hash - fileState=2 (approved), - policyIds=0 (global)'.
- Function:** A rounded rectangle icon labeled 'CbProtect Update Approval Request' with a callout 'Update the approval request, marking it as Closed - Resolved - Approved'.
- End:** A diamond icon with an 'X' representing the end of the workflow.

Input Parameters:

Input Parameter	Value
bit9_file_catalog_id	

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 'trust:1' 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>.

Trust:1 represents trust of this file (0-10). Special value of -1 is reserved for unknown.

The screenshot shows the 'Customization Settings' page in the Resilient interface. The 'Workflows' tab is selected, and the workflow '(Example) CbProtect Get File Catalog for Query 'trust:1'' is being edited. The settings include:

- Name:** (Example) CbProtect Get File Catalog for Query 'trust:1'
- API Name:** example_cbprotection_query_file_catalog
- Description:** Given a "trust:1" query condition, return details based on the file catalog results.
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:56
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Get File Catalog

The workflow diagram shows a sequence of steps:

- Start your workflow here** (Start node)
- Queries the file catalog for files with a Trust level of 1** (Function node: CbProtect Get File Catalog for Query 'trust:1')
- A note is created with the results** (Note node)
- End node** (End node)

The workflow is configured with the following input parameters:

Input Parameter	Value
bit9_query	trust:1

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:

Resilient Dashboards Simulations Incidents **Create** Orchestratio... Test Organiza...

Customization Settings

Layouts Rules Scripts **Workflows** Functions Message Destinations Phases & Tasks Incident Types Breach Artifacts

Workflows / (Example) CbProtect Approve File Locally and Close Request Cancel Save & Close Save

Name * (Example) CbProtect Approve File Locally and Close Request

API Name * cb_protect_approve_file_locally_and_close_request

Description Approve the file locally, then mark the approval request as Closed (Approved).

Object Type * Incident

Creator Orchestration Engine

Last Modified 06/19/2019 11:57

Last Modified By Orchestration Engine

Associated Rules (Example) CbProtect Approve File Lo

This incident workflow will usually be triggered manually.

Locate the file instance

Update the file instance setting local approval

Update the approval request, marking it as Closed - Resolved - Approved

no request id, or no computer id

Input Pre-Process Script Output Post-Process Script

Input Parameter	Value
bit9_computer_id ⓘ	
bit9_file_catalog_id ⓘ	
bit9_file_path ⓘ	
bit9_file_instance_localstate ⓘ	

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:

The screenshot displays the Resilient platform interface for customizing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and a 'Create' button. The 'Customization Settings' section is active, showing tabs for 'Layouts', 'Rules', 'Scripts', 'Workflows', 'Functions', 'Message Destinations', 'Phases & Tasks', 'Incident Types', 'Breach', and 'Artifacts'. The selected workflow is '(Example) CbProtect Approve File Locally and Close Request'.

Workflow Details:

- Name:** (Example) CbProtect Approve File Locally and Close Request
- API Name:** cb_protect_approve_file_locally_and_close_request
- Description:** Approve the file locally, then mark the approval request as Closed (Approved).
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:57
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Approve File Lo...

Workflow Diagram:

The workflow diagram illustrates the process flow:

- Start:** A circle icon representing the start of the workflow.
- Decision:** A diamond icon with an 'X' representing a decision point. A note indicates: 'This incident workflow will usually be triggered manually.'
- Function:** A rounded rectangle icon with an 'f' and 'id' representing a function. The function is 'CbProtect Get File Instance for Q...'. A note above it says: 'Locate the file instance'.
- Function:** A rounded rectangle icon with an 'f' and 'id' representing a function. The function is 'CbProtect Update File Instance Lo...'. A note above it says: 'Update the file instance setting local approval'.
- Function:** A rounded rectangle icon with an 'f' and 'id' representing a function. The function is 'CbProtect Update Approval Request'. A note above it says: 'Update the approval request, marking it as Closed - Resolved - Approved'.
- Decision:** A diamond icon with an 'X' representing a decision point. A note below it says: 'no request id, or no computer id'.
- End:** A circle icon representing the end of the workflow.

Input Parameters:

Input Parameter	Value
bit9_file_instance_id	
bit9_file_instance_localstate	

bit9_file_rule_delete: CbProtect Delete File Rule for Id

Given a file rule ID, deletes the file rule from Carbon Black. The following is an example of this function in the (Example) CbProtect Delete File Rule for Id 1 workflow:

The screenshot shows the 'Customization Settings' page in the Resilient interface. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and a 'Create' button. The main content area is titled 'Customization Settings' and features a tabbed interface with 'Workflows' selected. The workflow name is '(Example) CbProtect Delete File Rule for Id 1'. The settings include:

- Name:** (Example) CbProtect Delete File Rule for Id 1
- API Name:** example_cbprotection_delete_file_rule
- Description:** Deletes a file rule given a file rule id.
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/20/2019 16:02
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Delete File Rule

Below the settings is a workflow diagram. It starts with a 'Start your workflow here' node, followed by a 'CbProtect Delete File Rule for Id' function node. A callout box points to the function node with the text 'Update input with the id you wish to delete'. The diagram also shows a 'Post-Process Script' node and a 'Start your workflow here' node. The workflow is divided into four sections: Input, Pre-Process Script, Output, and Post-Process Script.

Input Parameter	Value
bit9_file_rule_id	1

bit9_file_rule_get: CbProtect Get File Rule for Id

Given a file rule ID, returns the details of the file rule. The following is an example of this function in the (Example) CbProtect Get File Rule for Id 1 workflow:

The screenshot displays the 'Customization Settings' page in the Resilient platform. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and a 'Create' button. The main content area is titled 'Customization Settings' and features a tabbed interface with 'Workflows' selected. The workflow name is '(Example) CbProtect Get File Rule for Id 1'. The API Name is 'example_cbprotection_file_rule_get'. The Description is 'Given a file rule id, returns back the details of the rule.' The Object Type is 'Incident'. The Creator is 'Orchestration Engine' and the Last Modified date is '06/20/2019 15:57'. The Last Modified By is 'Orchestration Engine'. The Associated Rules are listed as '(Example) CbProtect Get File Rule for Id 1'.

The workflow diagram shows a sequence of steps: 'Start your workflow here' (represented by a circle), followed by 'Update input with the id of file rule you wish to query' (represented by a circle), then the main function 'CbProtect Get File Rule for Id' (represented by a circle with an 'f' icon), and finally 'A note is created with the results' (represented by a circle). The workflow is connected to a 'Post-Process Script' step.

The bottom section of the page shows the 'Input' tab with a table of input parameters:

Input Parameter	Value
bit9_file_rule_id	1

bit9_file_rule_query: CbProtect Get File Rule for Query Condition

Given a query string, returns details of the file rules that match the query. The following is an example of this function in the (Example) CbProtect Get File Rule for Query 'name: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 file rule properties to query

<https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#filerule>.

fileName:notepad.exe represents the file name associated with this rule.

Customization Settings

Layouts Rules Scripts **Workflows** Functions Message Destinations Phases & Tasks Incident Types Breach Artifacts

Workflows / (Example) CbProtect Get File Rule for Query 'name:notepad.exe'

Name * (Example) CbProtect Get File Rule for Query 'name:notepad.exe'

API Name * example_cbprotection_query_file_rule

Description Given a "name:notepad.exe" query condition, returns back the details of the file rules which match the query condition.

Object Type * Incident

Creator Orchestration Engine

Last Modified 06/19/2019 11:56

Last Modified By Orchestration Engine

Associated Rules (Example) CbProtect Get File Rule for Query 'name:notepad.exe'

Workflow Diagram:

- Start your workflow here
- Queries for file rules where the name contains notepad.exe
- CbProtect Get File Rule for Query...
- A note is created with the results

Input Parameter	Value
bit9_query *	name:notepad.exe

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:

The screenshot displays the Resilient console interface for customizing a workflow. The top navigation bar includes 'Dashboards', 'Simulations', 'Incidents', and 'Create'. The 'Workflows' tab is selected, showing the workflow '(Example) CbProtect Approve File Globally and Close Request'.

Customization Settings:

- Name:** (Example) CbProtect Approve File Globally and Close Request
- API Name:** cb_protect_approve_file_globally_and_close_request
- Description:** Approve the file hash globally, then mark the approval request as Closed (Approved).
- Object Type:** Incident
- Creator:** Orchestration Engine
- Last Modified:** 06/19/2019 11:57
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Approve File G

Workflow Diagram:

The workflow diagram illustrates the process flow:

- Start:** A circle icon representing the start of the workflow.
- Decision:** A diamond icon with an 'X' that checks for 'no request id, or no file catalog id'. If true, it bypasses the next steps and goes to the end.
- Function 1:** 'CbProtect Get File Catalog for Id' (labeled 'f(x)').
- Function 2:** 'CbProtect Update File Rule' (labeled 'f(x)'). A callout box specifies: 'Set the file rule for this hash - fileState=2 (approved), - policyids=0 (global)'.
- Function 3:** 'CbProtect Update Approval Request' (labeled 'f(x)'). A callout box specifies: 'Update the approval request, marking it as Closed - Resolved - Approved'.
- End:** A circle icon representing the end of the workflow.

Input Parameters:

Input Parameter	Value
bit9_file_rule_id	1
bit9_file_catalog_id	
bit9_file_rule_name	
bit9_file_rule_description	
bit9_file_rule_filestate	
bit9_file_rule_sourcetype	5
bit9_file_rule_policyids	
bit9_file_rule_hash	

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/
```