IBM Resilient

Incident Response Platform Integrations

Cisco AMP for Endpoints Function V1.0.0

Release Date: March 2019

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

This guide describes the Cisco AMP for Endpoints Function.

Overview

Cisco AMP for Endpoints is a cloud-managed, endpoint security solution that prevents threats at the point of entry, then continuously tracks every file it allows onto the endpoints. It can also detect, contain, and remediate malicious files on an endpoint before damage can be done.

The Cisco AMP for Endpoints integration with the Resilient platform allows querying and updating of an AMP for Endpoints deployment. The returned results can be used to make customized updates to the Resilient platform, such as updating incidents, artifacts, data tables and so on.

The following type of queries can be executed:

- Get computer, computers or groups.
- Get computer trajectory.
- Get activity.
- Get event types or events.
- Get file lists, files in file list (get file list files).

The integration can also be used to make the following changes to a Cisco Amp for Endpoints environment:

- Add suspicious sha-256 values to a file list in the Cisco Amp for Endpoints environment, which can then be used to blacklist the related file.
- Delete sha-256 values from a file list in the Cisco Amp for Endpoints environment.
- Move a computer to a new group.

Paginated results

With Cisco AMP for Endpoints, an API query returns results to the client in paginated format so that responses are easier to handle. A query total result can thus be larger than will fit in a single paginated result. The package functions will attempt to retrieve the total results for a query up to a limit defined for the integration.

Limits

There are 2 types of limit enforcement in the Cisco AMP for Endpoints integration.

Query limits

Since some of the API queries can have a total number of results which can overwhelm the Resilient platform's ability to process them, the integration has in-built limits to the amount of results that are returned by the functions.

- A default global limit is set to 1000 results.
- The integration configuration parameter, `query_limit`, can be used to override the global results limit if required. There is no upper limit for this setting.
- A number of the integration functions have a `limit` parameter for the API call itself. If this value is set, it overrides both the global and `query_limit` values; however, there is an upper limit of 500 for this parameter, above which, an HTTP error is thrown.

Rate limits

The Cisco AMP for Endpoints API clients are only allowed to make a limited number of requests per hour. Each request returns a response with headers detailing the current rate limit status. If the limit is overrun, an HTTP 429 error is thrown.

```
X-Rate-Limit-Limit - Total allowed requests in the current period.
X-Rate-Limit-Remaining - Requests left.
X-Rate-Limit-Reset - Number of seconds before the limit is reset.
```

The integration uses the rate limit headers to prevent the integration from doing an overrun on this limit. In the event that a 429 error is thrown, the integration configuration setting, `max_retries`, is used to determine how many times the integration attempts a retry of the request.

There are a total of 12 functions supplied in the Resilient Cisco Amp for Endpoints package. There are also example rules, workflows and scripts in the customizations section of the package that demonstrate usage of the various functions.

The remainder of this document describes the included functions, rules, workflows, scripts and data tables. It also demonstrates how to configure and execute the example custom workflows.

Installation

Before installing, verify that your environment meets the following prerequisites:

- Resilient platform must be version 31 or later.
- You must have a Resilient account to use for the integrations. This can be any account that has the permission to view and modify administrator and customization settings, and read and update incidents. You must know the account username and password.
- You must have access to the command line of the Resilient appliance, which hosts the Resilient platform; or to a separate integration server where you will deploy and run the functions code. If you are using a separate integration server, you must install Python version 2.7.10 or later, or version 3.6 or later, and "pip". (The Resilient appliance is preconfigured with a suitable version of Python.)

Configure Cisco AMP for Endpoints

Cisco AMP for Endpoints is a cloud based solution which has different API versions and base URLs (based on geographic region). The integration has configuration settings to allow it to be set up correctly for the local environment.

Access to the Cisco AMP for Endpoints REST API is allowed by providing a client ID and API access token in the request. The access token is tied to a user account on the AMP for Endpoints console.

More information is available at: Cisco AMP for Endpoints documentation

And more specifically for the API: Cisco AMP for Endpoints REST API documentation

Install the Python components

The functions package contains Python components that are called by the Resilient platform to execute the functions during your workflows. These components run in the Resilient Circuits integration framework.

The package also includes Resilient customizations that will be imported into the platform later.

Complete the following steps to install the Python components:

1. Ensure that the environment is up-to-date, as follows:

```
sudo pip install --upgrade pip
sudo pip install --upgrade setuptools
sudo pip install --upgrade resilient-circuits
```

2. Run the following command to install the package:

```
sudo pip install --upgrade fn_cisco_amp4ep-1.0.0.tar.gz
```

Configure the Python components

The Resilient Circuits components run as an unprivileged user, typically named integration. If you do not already have an integration user configured on your appliance, create it now.

Complete the following steps to configure and run the integration:

1. Using sudo, switch to the integration user, as follows:

   ```
   sudo su - integration
   ```

2. Use one of the following commands to create or update the resilient-circuits configuration file. Use `-c` for new environments or `-u` for existing environments.

   ```
   resilient-circuits config -c
   ```

   or

   ```
   resilient-circuits config -u
   ```

3. Edit the resilient-circuits configuration file, as follows:

   a. In the [resilient] section, ensure that you provide all the information required to connect to the Resilient platform.

   b. In the [fn_cisco_amp4ep] section, edit the settings as follows (N.A.):

   ```
   [fn_cisco_amp4ep]
   base_url=https://api.amp.cisco.com/
   api_version=v1
   # The client id will be generated on the Cisco AMP for endpoints
   dashboard.
   client_id=<client id>
   # The api_token will be generated on the Cisco AMP for endpoints
   dashboard and will be will be in uuid format.
   api_token=<api token>
   # Settings for access to cisco website via a proxy
   #http_proxy=http':'http://proxy:80
   #https_proxy=https':'http://proxy:80
   # Query results global limit override for the integration global default
   which is set to 1000.
   #query_limit=1000
   # Max number of retry attempts on Rate Limit exception
   max_retries=3
   ```

Deploy customizations to the Resilient platform

The package contains function definitions that you can use in workflows, and includes example workflows and rules that show how to use these functions.

1. Use the following command to deploy these customizations to the Resilient platform:

   ```
   resilient-circuits customize
   ```

2. Respond to the prompts to deploy functions, message destinations, workflows and rules.

Run the integration framework

To test the integration package before running it in a production environment, you must run the integration manually, using the following command:

```
resilient-circuits run
…
2018-11-26 16:56:34,467 INFO [app] Configuration file: app.config
2018-11-26 16:56:34,469 INFO [app] Resilient server: <host>
2018-11-26 16:56:34,470 INFO [app] Resilient user: <acct>
2018-11-26 16:56:34,470 INFO [app] Resilient org: <org>
2018-11-26 16:56:34,471 INFO [app] Logging Level: INFO
…
2018-11-26 16:56:35,411 INFO [component_loader] Loading 12 components
2018-11-26 16:56:35,412 INFO [component_loader]
'fn_cisco_amp4ep.components.fn_amp_delete_file_list_files.FunctionComponent'
loading
…
2018-11-26 16:56:35,427 INFO [component_loader]
'fn_cisco_amp4ep.components.fn_amp_get_computers.FunctionComponent' loading
…
2018-11-26 16:56:35,439 INFO [actions_component]
'fn_cisco_amp4ep.components.fn_amp_delete_file_list_files.FunctionComponent'
function 'fn_amp_delete_file_list_files' registered to 'fn_cisco_amp'
…
2018-11-26 16:56:35,578 INFO [actions_component]
'fn_cisco_amp4ep.components.fn_amp_get_computers.FunctionComponent' function
'fn_amp_get_computers' registered to 'fn_cisco_amp'
2018-11-26 16:56:35,578 INFO [app] Components loaded
2018-11-26 16:56:35,686 INFO [actions_component] Subscribe to message
destination 'fn_cisco_amp'
2018-11-26 16:56:35,687 INFO [stomp_component] Subscribe to message destination
actions.202.fn_cisco_amp
…
```

The `resilient-circuits` command starts, loads its components, and continues to run until interrupted. If it stops immediately with an error message, check your configuration values and retry.

Configure Resilient Circuits for restart

For normal operation, Resilient Circuits must run continuously. The recommended way to do this is to configure it to automatically run at start up. On a Red Hat appliance, you can do this using a `systemd` unit file such as the one below. You might need to change the paths to your working directory and `app.config`.

1. The unit file must be named `resilient_circuits.service` To create the file, enter the following command:

   ```
   sudo vi /etc/systemd/system/resilient_circuits.service
   ```

2. Add the following contents to the file and change as necessary:

   ```
   [Unit]
   Description=Resilient-Circuits Service
   After=resilient.service
   Requires=resilient.service
   ```

   ```
   [Service]
   Type=simple
   User=integration
   WorkingDirectory=/home/integration
   ExecStart=/usr/local/bin/resilient-circuits run
   Restart=always
   TimeoutSec=10
   Environment=APP_CONFIG_FILE=/home/integration/.resilient/app.config
   Environment=APP_LOCK_FILE=/home/integration/.resilient/resilient_circuits.lo
   ck
   ```

   ```
   [Install]
   WantedBy=multi-user.target
   ```

3. Ensure that the service unit file is correctly permissioned, as follows:

   ```
   sudo chmod 664 /etc/systemd/system/resilient_circuits.service
   ```

4. Use the `systemctl` command to manually start, stop, restart and return status on the service:

   ```
   sudo systemctl resilient_circuits [start|stop|restart|status]
   ```

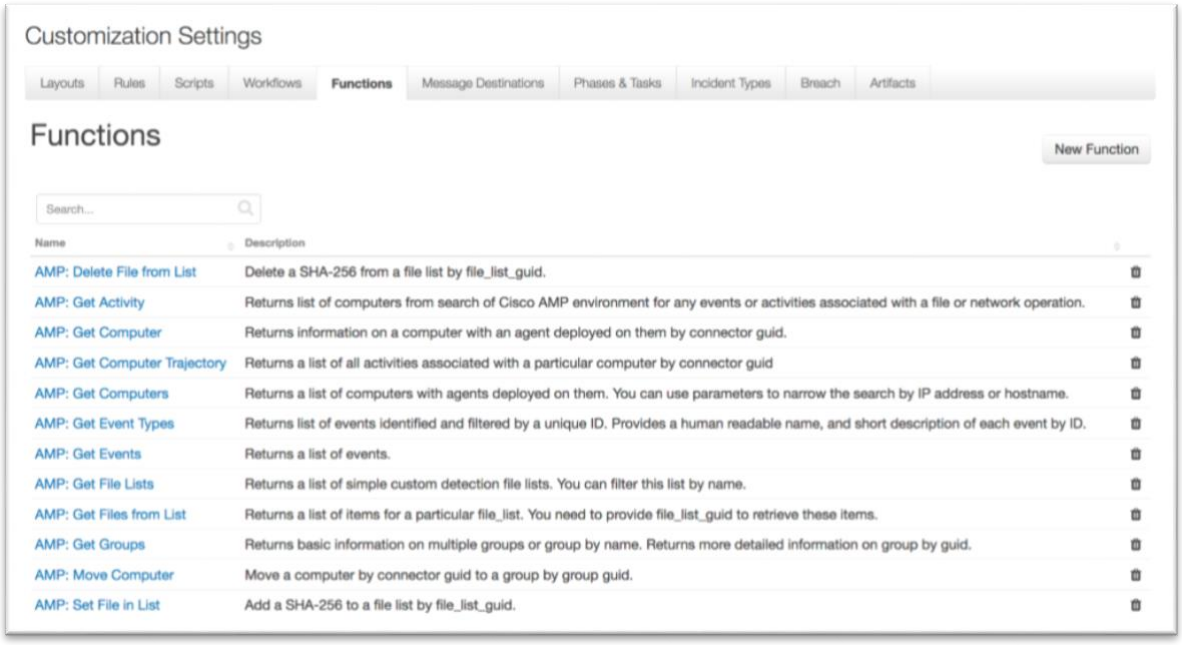
You can view log files for `systemd` and the `resilient-circuits` service using the `journalctl` command, as follows:

```
sudo journalctl -u resilient_circuits --since "2 hours ago"
```

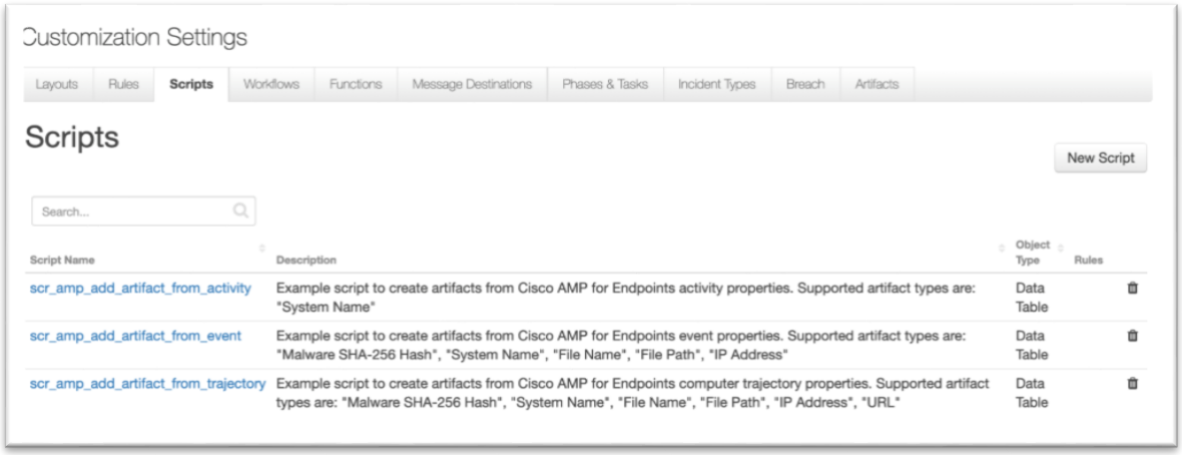
Function Descriptions

Once the function package deploys the functions, you can view the customizations in the Resilient platform Customizations Settings as shown below.

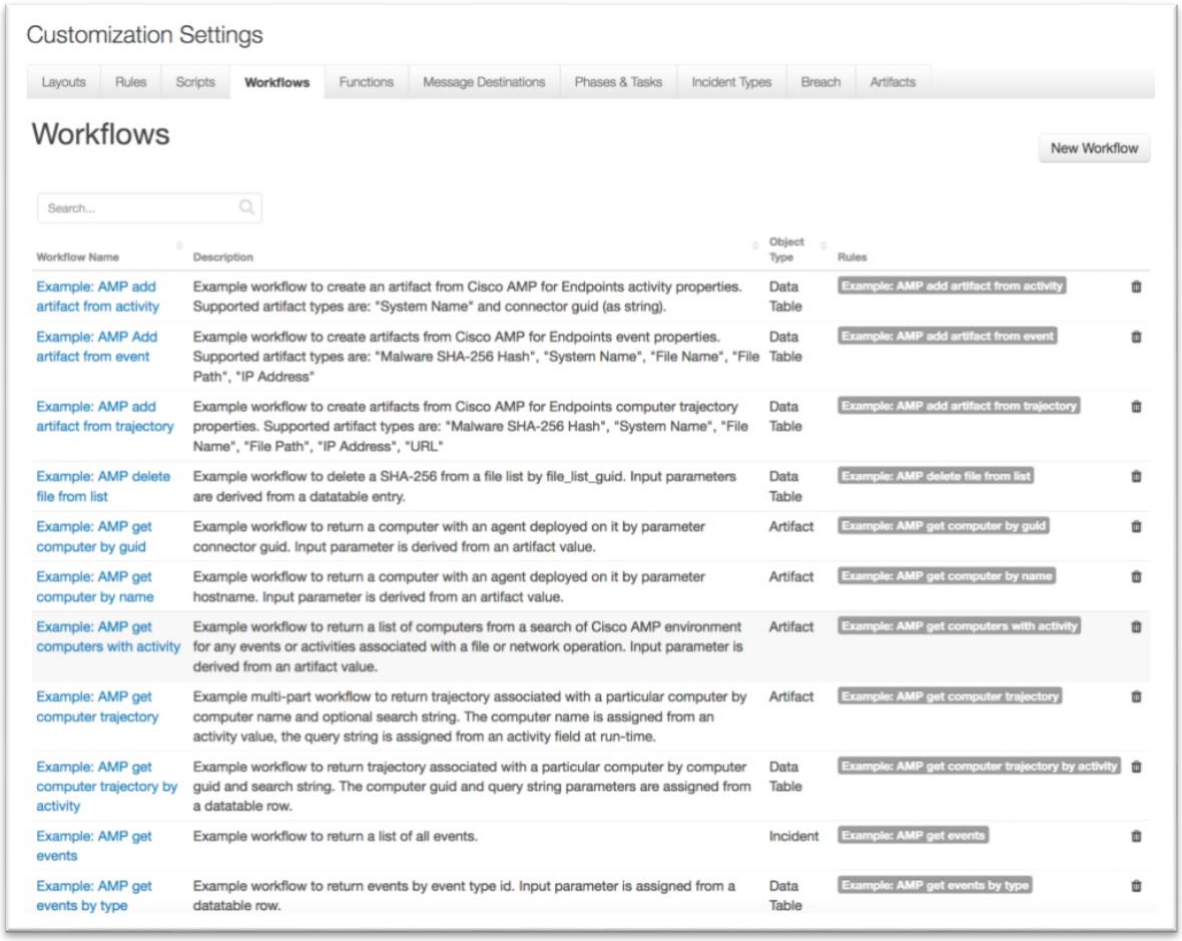
Functions

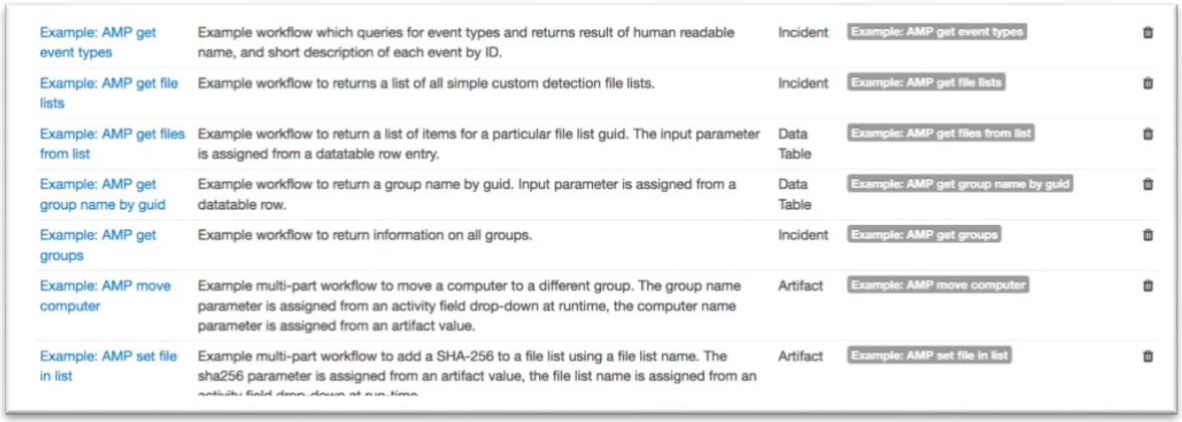


Scripts



Workflows





Functions and related components

The package includes example workflows, scripts, rules and data tables that show how you can use the functions, as shown in the following table. Resilient users can view the rules in the Rules tab, the workflows in the Workflows tab and scripts in Scripts tab and modify them as needed.

Query type functions (Query the Cisco AMP for Endpoints environment)

Function name	Workflows	Rules	Data Tables
AMP: Get Event Types	Example: AMP get event types	Example: AMP get event types	Cisco AMP Event Types
AMP: Get Events	Example: AMP get events Example: AMP get events by type	Example: AMP get events Example: AMP get events by type	Cisco AMP Events
AMP: Get Computer	Example: AMP get computer by guid	Example: AMP get computer by guid	Cisco AMP Computers
AMP: Get Computers	Example: AMP get computer by name	Example: AMP get computer by name	Cisco AMP Computers
AMP: Get Computer Trajectory	Example: AMP get computer trajectory Example: AMP get computer trajectory by activity	Example: AMP get computer trajectory Example: AMP get computer trajectory by activity	Cisco AMP Computer Trajectory
AMP: Get Activity	Example: AMP get computers with activity	Example: AMP get computers with activity	Cisco AMP Activity
AMP: Get File Lists	Example: AMP get file lists	Example: AMP get file lists	Cisco AMP SCD File Lists
AMP: Get Files from List	Example: AMP get files from list	Example: AMP get files from list	Cisco AMP File List files
AMP: Get Groups	Example: AMP get groups Example: AMP get group name by guid	Example: AMP get groups Example: AMP get group name by guid	Cisco AMP Groups

Update type functions (Make changes to the Cisco AMP for Endpoints environment)

AMP: Set File in List	Example: AMP set file in list	Example: AMP set file in list	N/A
AMP: Delete File from List	Example: AMP delete file from list	Example: AMP delete file from list	N/A
AMP: Move Computer	Example: AMP move computer	Example: AMP move computer	N/A

Scripts (Generate Resilient artifacts from Cisco AMP for Endpoints properties)

Script name	Workflows	Rules	Artifact types
Script: AMP add artifact from activity	Example: AMP add artifact from activity	Example: AMP add artifact from activity	"System Name" “Guid as string”
Script: AMP add artifact from event	Example: AMP Add artifact from event	Example: AMP Add artifact from event	"Malware SHA-256 Hash", "System Name", "File Name", "File Path", "IP Address"
Script: AMP add artifact from trajectory	Example: AMP add artifact from trajectory	Example: AMP add artifact from trajectory	"Malware SHA-256 Hash", "System Name", "File Name", "File Path", "IP Address", URL

Function inputs

Each function has a set of inputs, which you can view by clicking the function name in the Functions tab of the Resilient platform.

The Resilient functions use input parameters starting with `amp_`, examples include `amp_conn_guid`, `amp_file_list_guid` and `amp_q`. These are equivalent to the input parameters and qualifiers used in the REST API calls. Refer to Cisco AMP for Endpoints REST API documentation on the use of these inputs.

The following input parameter is used in the Resilient functions where the input can be one of several different types.

```
amp_q e.g. you can search on an IP Address, SHA256, file name or URL
```

Parameter `amp_start_date` uses a Datepicker to populate the value in the Workflow Input tab. This value is translated to a Unix epoch timestamp in milliseconds, by the Resilient platform. The Workflow Pre-Process Script accepts a Unix timestamp value in milliseconds.

Parameter `amp_event_type` accepts either a single integer or a list of comma-separated integers.

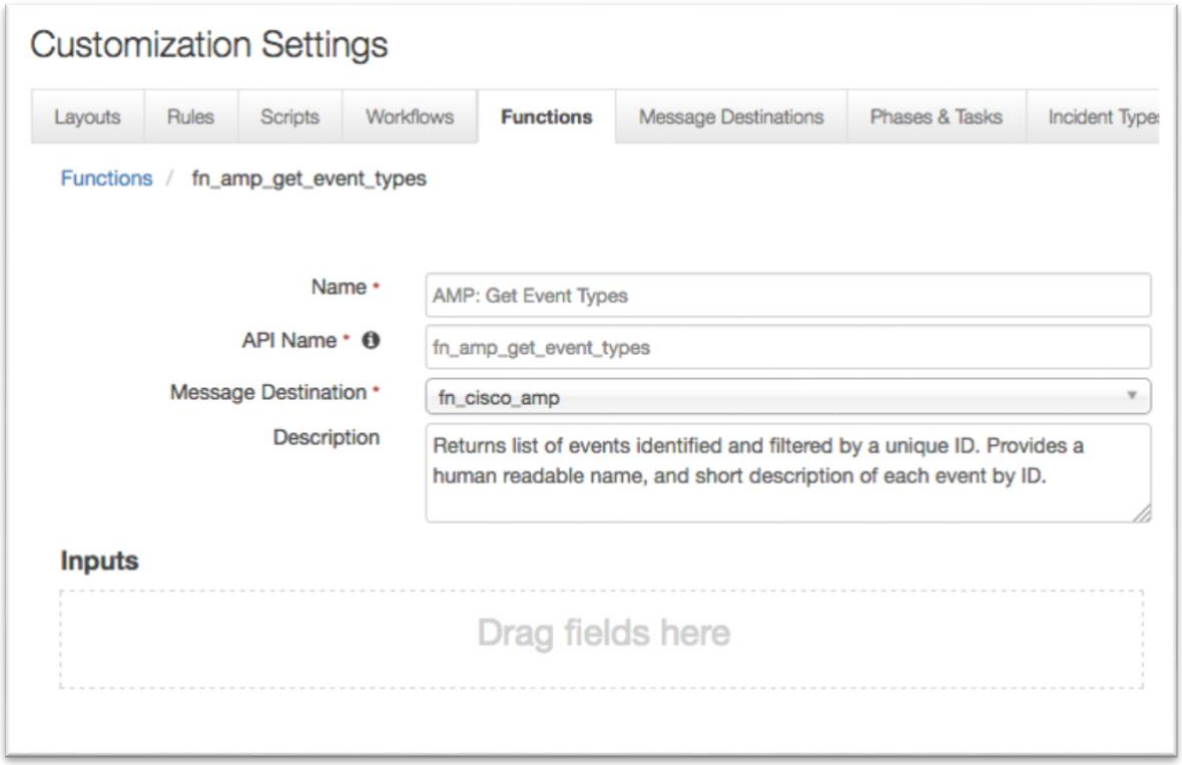
Function input assignment at run-time

A number of the example workflows assign function input values from activity fields at run-time.

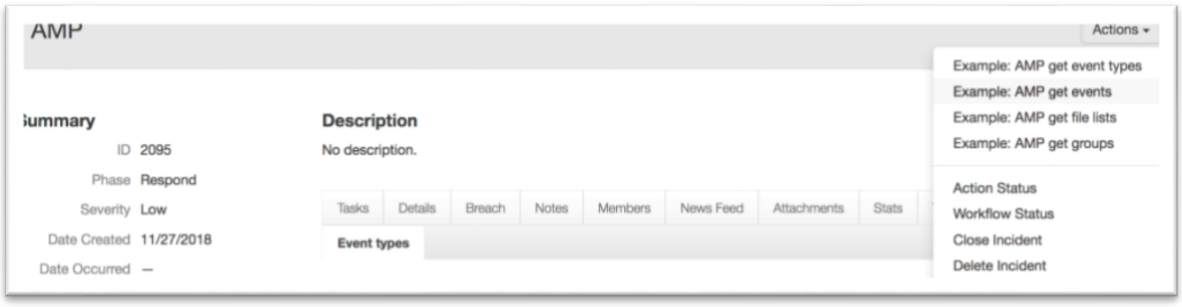
Customizations

AMP: Get Event Types

Returns list of events identified and filtered by a unique ID. Provides a human readable name, and short description of each event by ID.

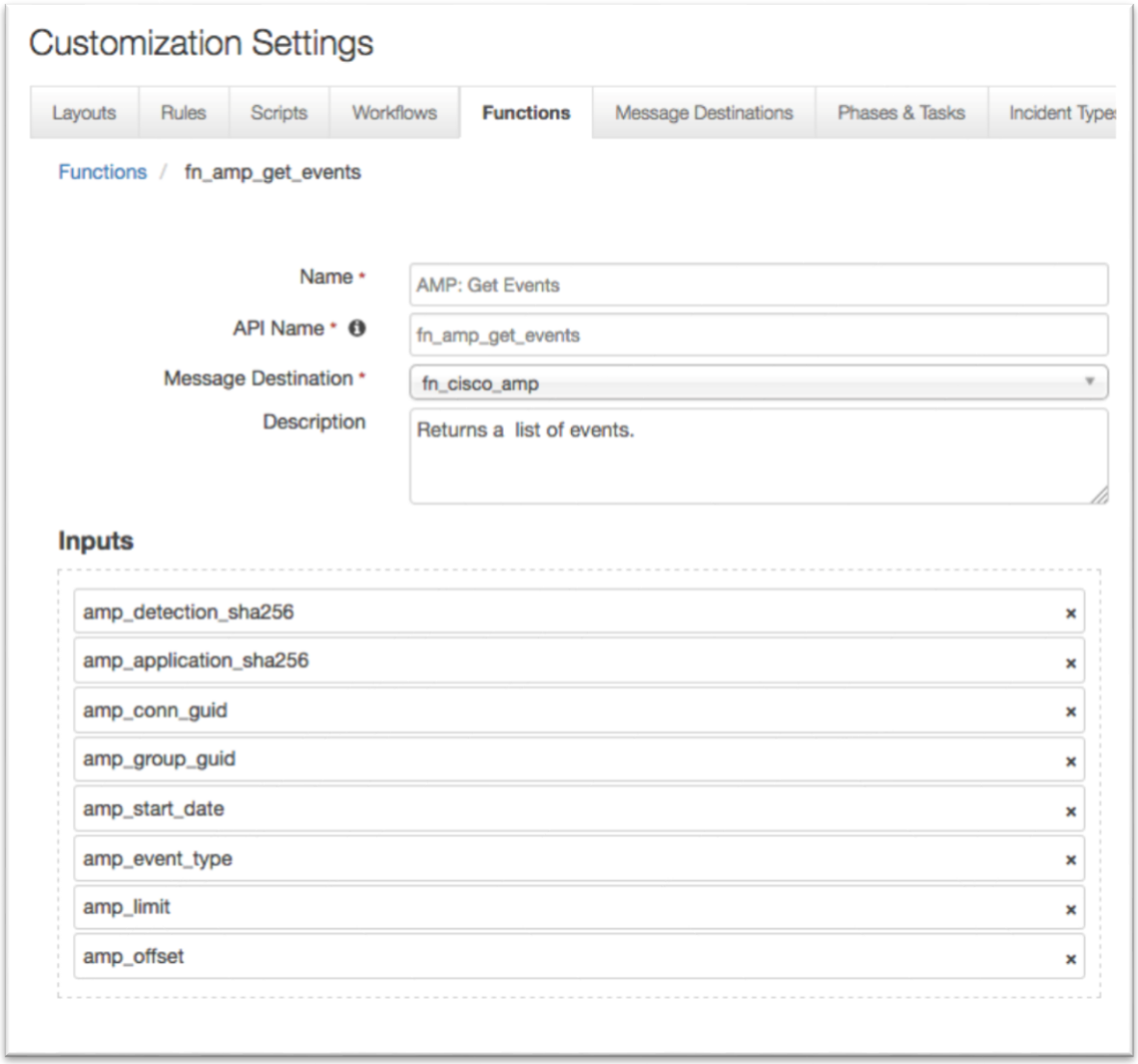


A Menu Item rule and workflow, both called “Example: AMP get event types”, are also included. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down menu of an incident.



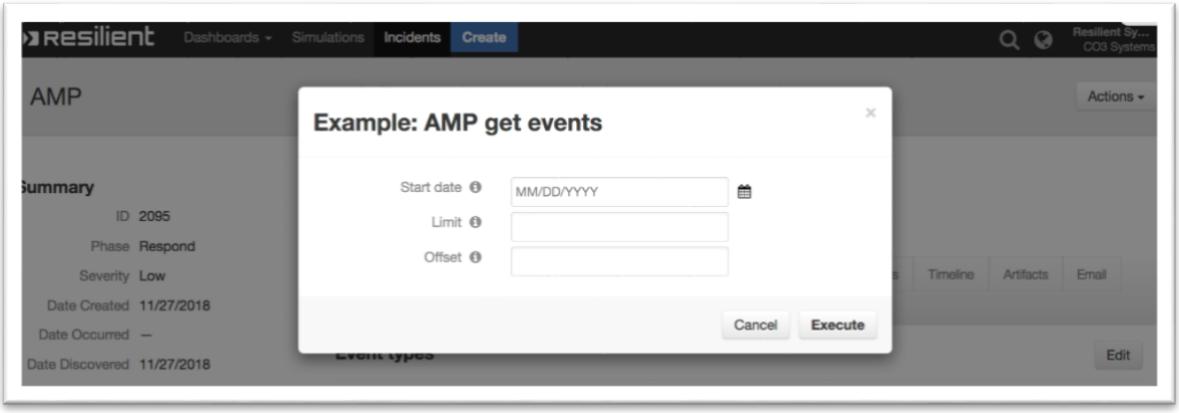
AMP: Get Events

Returns a list of events.

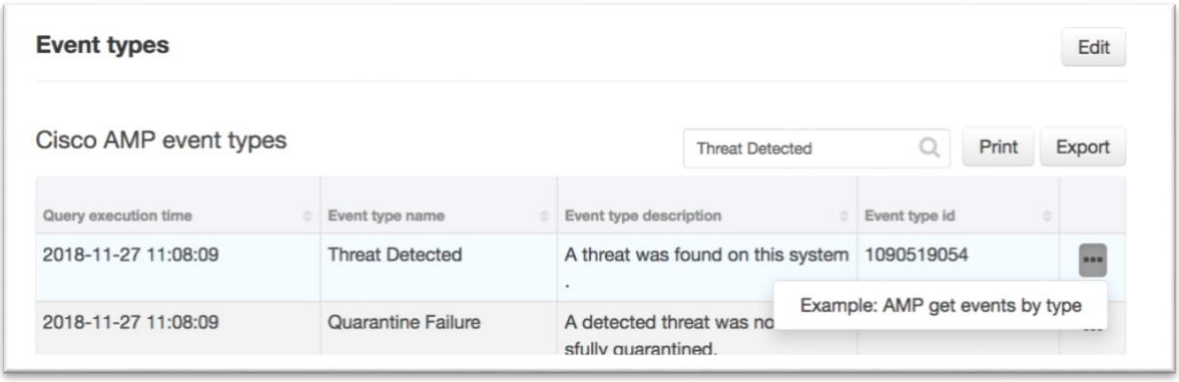


A Menu Item rule and workflow both named “Example: AMP get events” are included. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down menu of an incident.

This rule presents a dialog to the user where values for optional parameters `start_date`, `amp_limit` and `amp_offset` can be assigned at run-time. The parameters are assigned from activity fields.

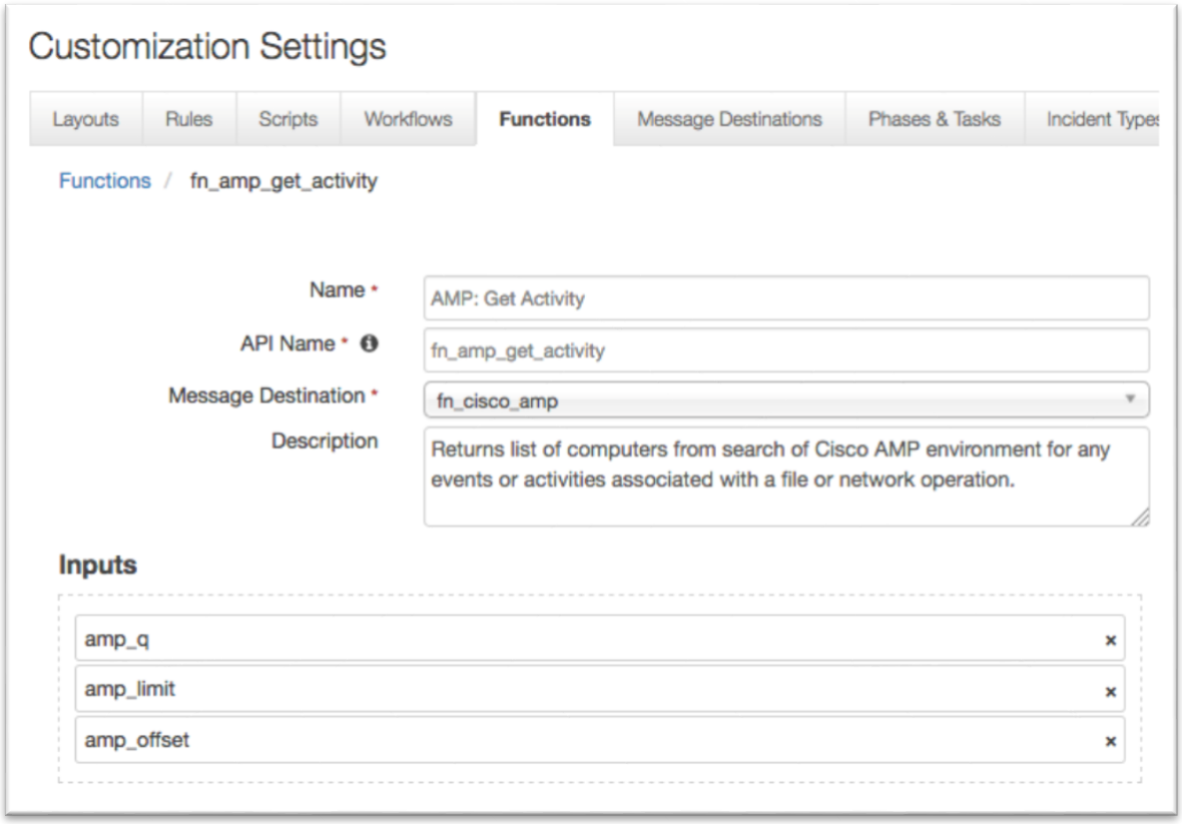


A Menu Item rule and workflow, both called Example: AMP get events by type”, are also included. A user can invoke the workflow by right-clicking on this rule from the Actions menu of a row in data table “Cisco AMP Event Types”. Event type ID input parameter is assigned from the data table row.

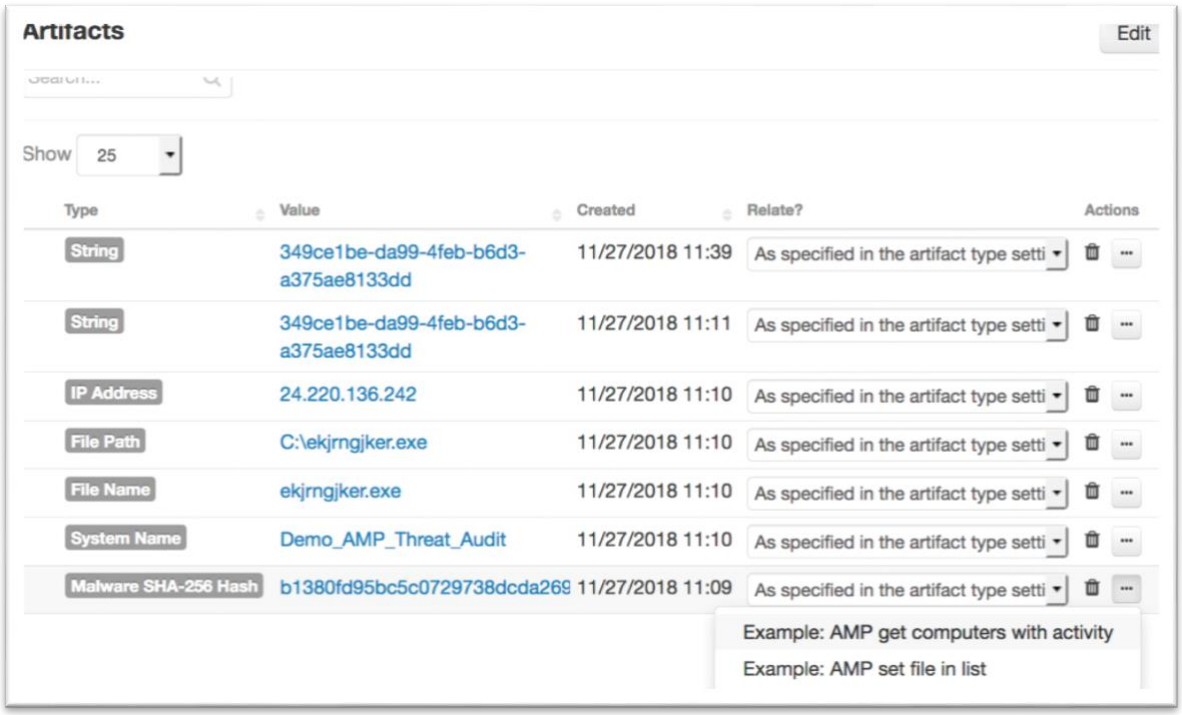


AMP: Get Activity

Returns list of computers from a search of the Cisco AMP environment for any events or activities associated with a file or network operation.

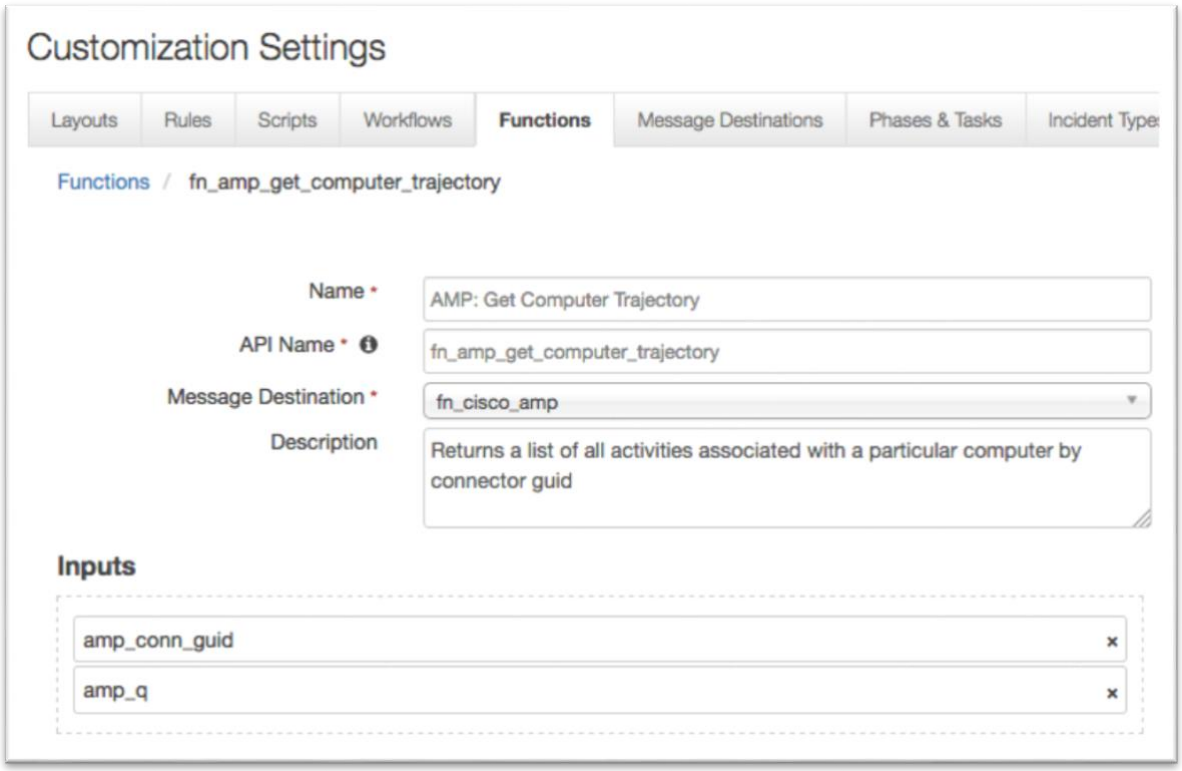


A Menu Item rule and workflow, both called “Example: AMP get computers with activity”, are included. A user can invoke the workflow by clicking the Actions icon for an artifact then selecting a rule. This rule is applicable for artifact types IP Address, Malware SHA-256 Hash, URL, and File Name. Query string input parameter is assigned from the artifact value.

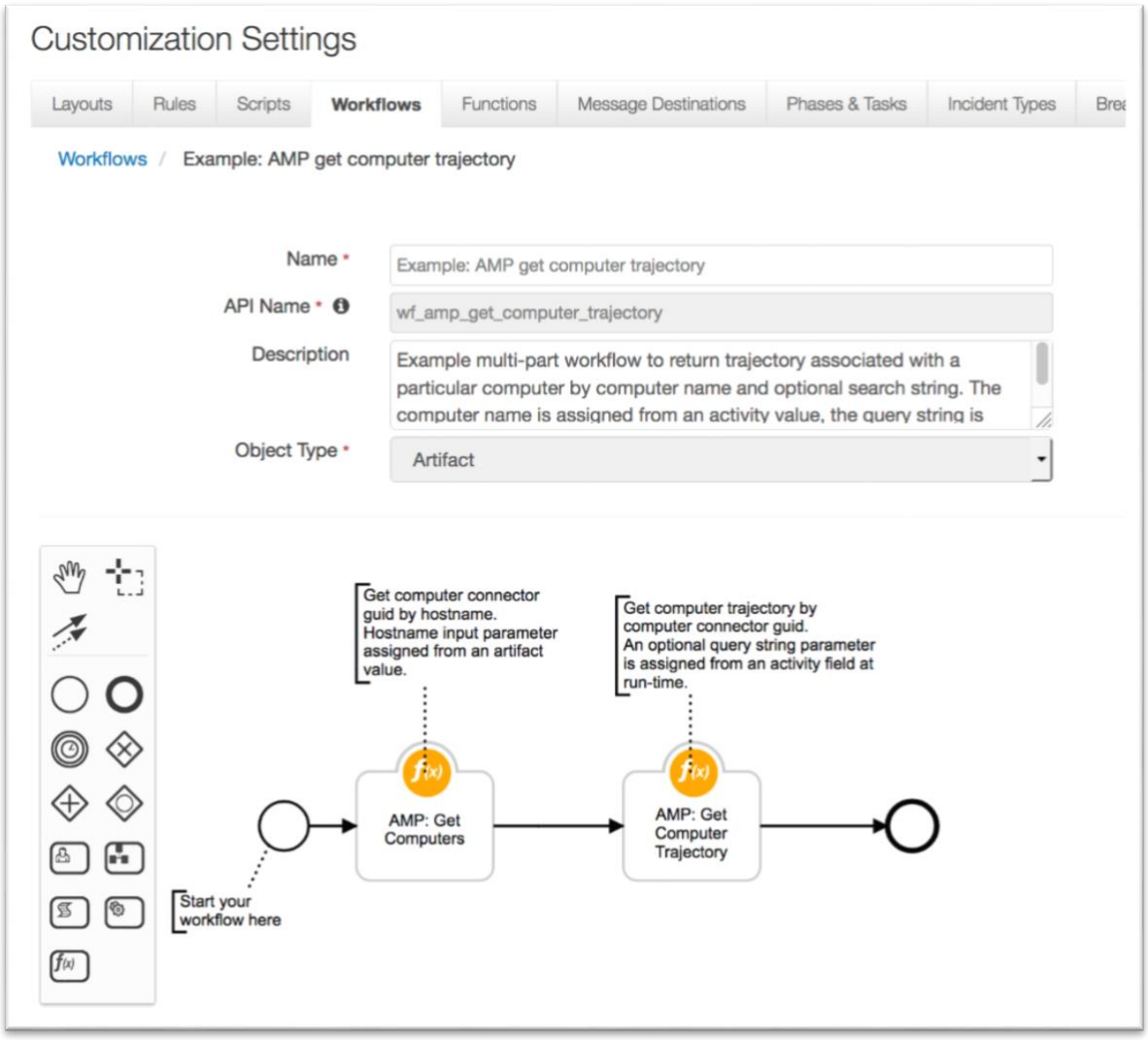


AMP: Get Computer Trajectory

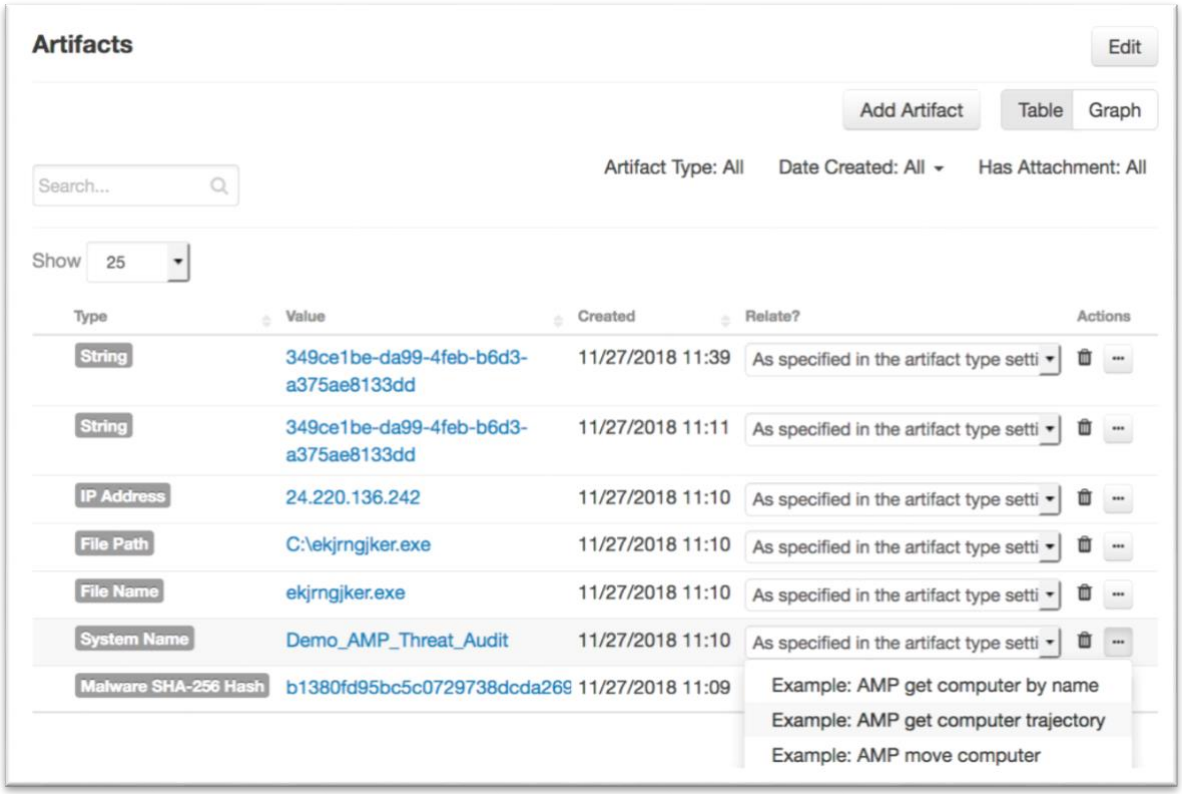
Returns a list of all activities associated with a particular computer by connector guid.



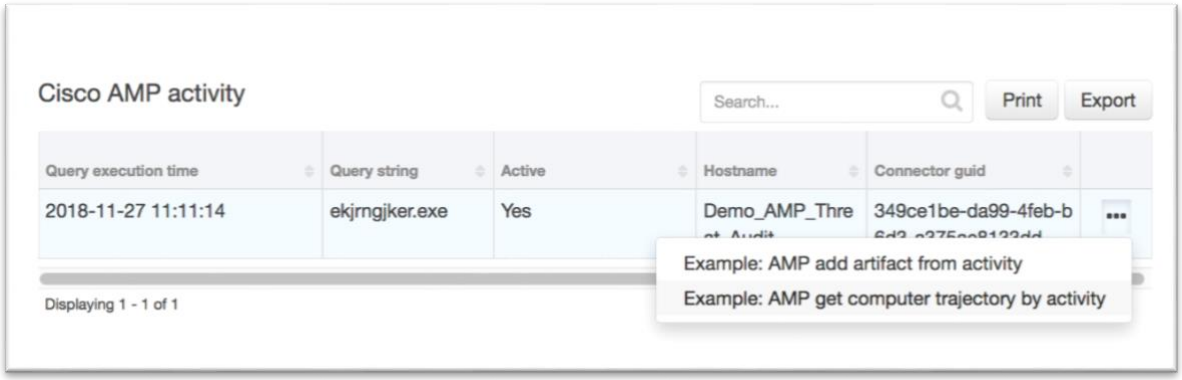
A Menu Item rule and workflow, both called “Example: AMP get computer trajectory”, are included. This is a multi-step workflow where the first function is used to provide the input value for connector guid in the second function.



A user can invoke the workflow by clicking the Actions icon for an artifact then selecting a rule. This rule is applicable for artifact types DNS Name and System Name.

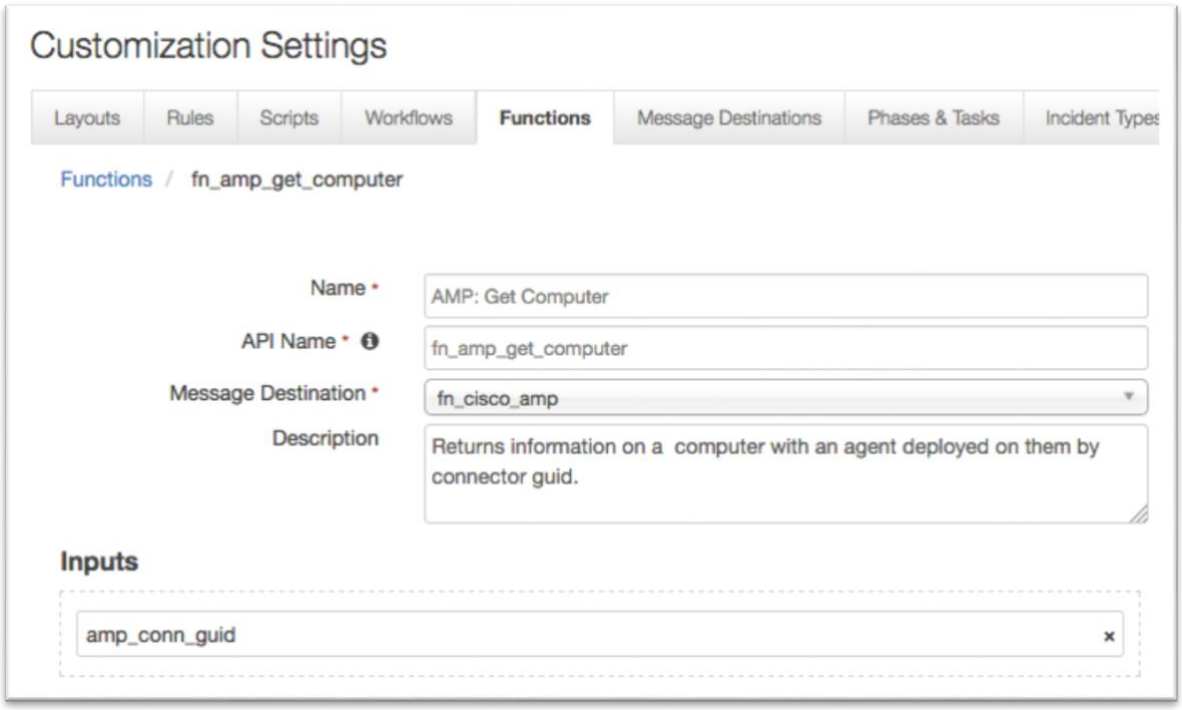


A Menu Item rule and workflow, both called “Example: AMP get computer trajectory by activity”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions menu of a row in data table “Cisco AMP Activity”.

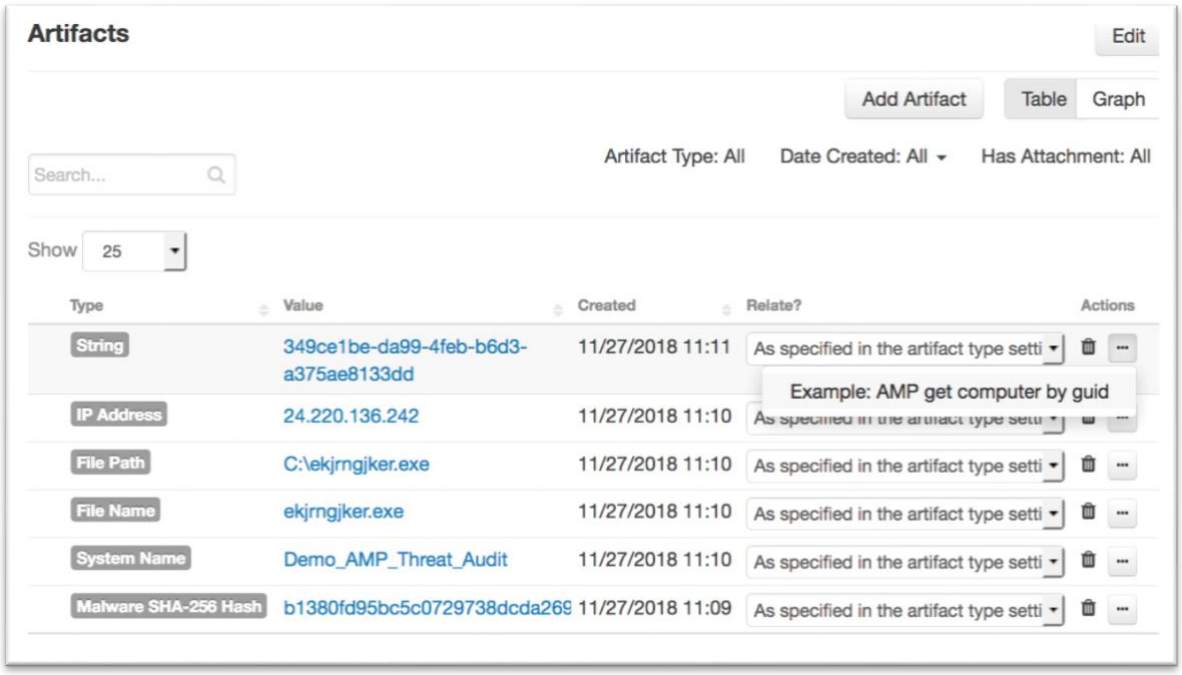


AMP: Get Computer

Returns information on a computer with an agent deployed on them by connector guid.

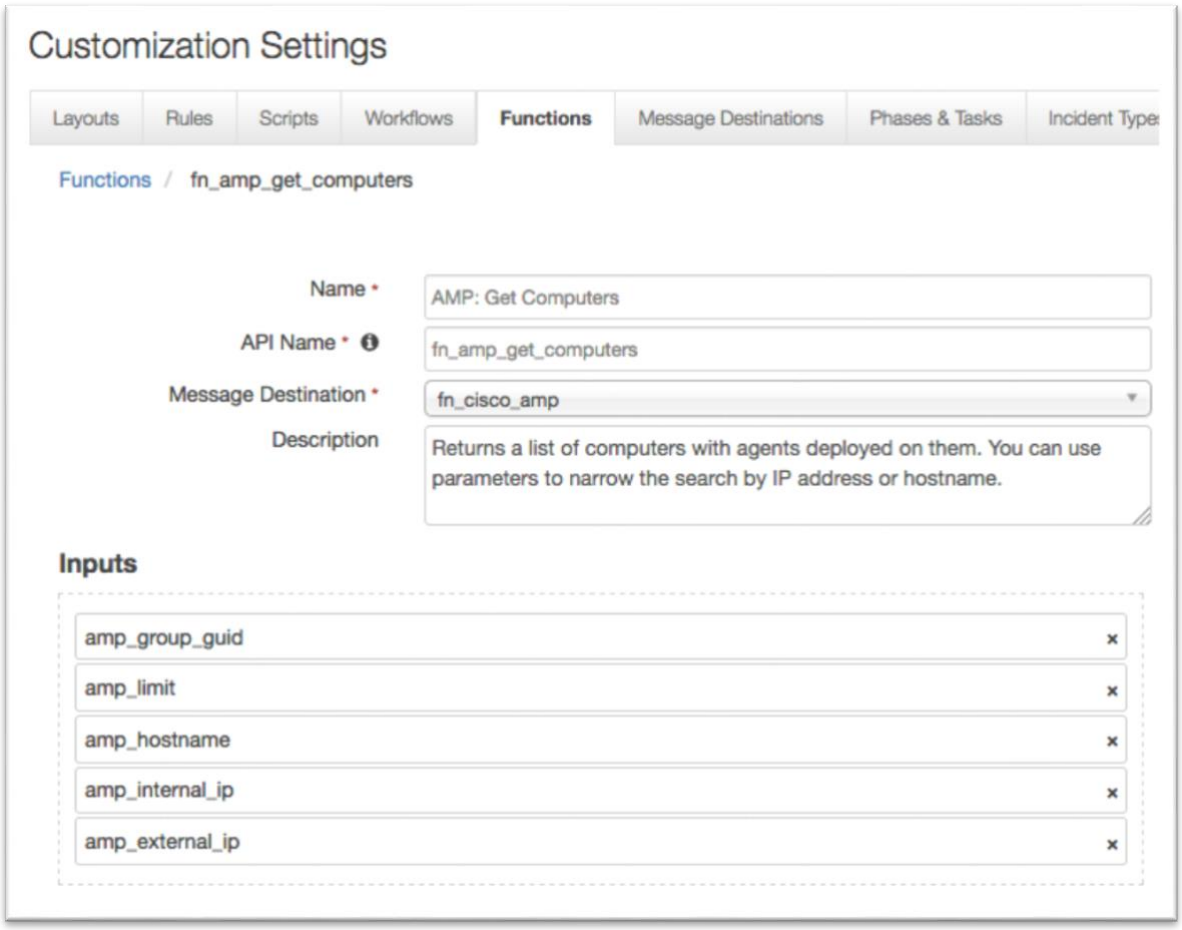


A Menu Item rule and workflow, both called "Example: AMP get computer by guid", are included. A user can invoke the workflow by clicking the Actions icon for an artifact then selecting the rule. This rule is applicable for artifact type String. The artifact value must be a valid Cisco AMP endpoint connector guid or the function will fail.

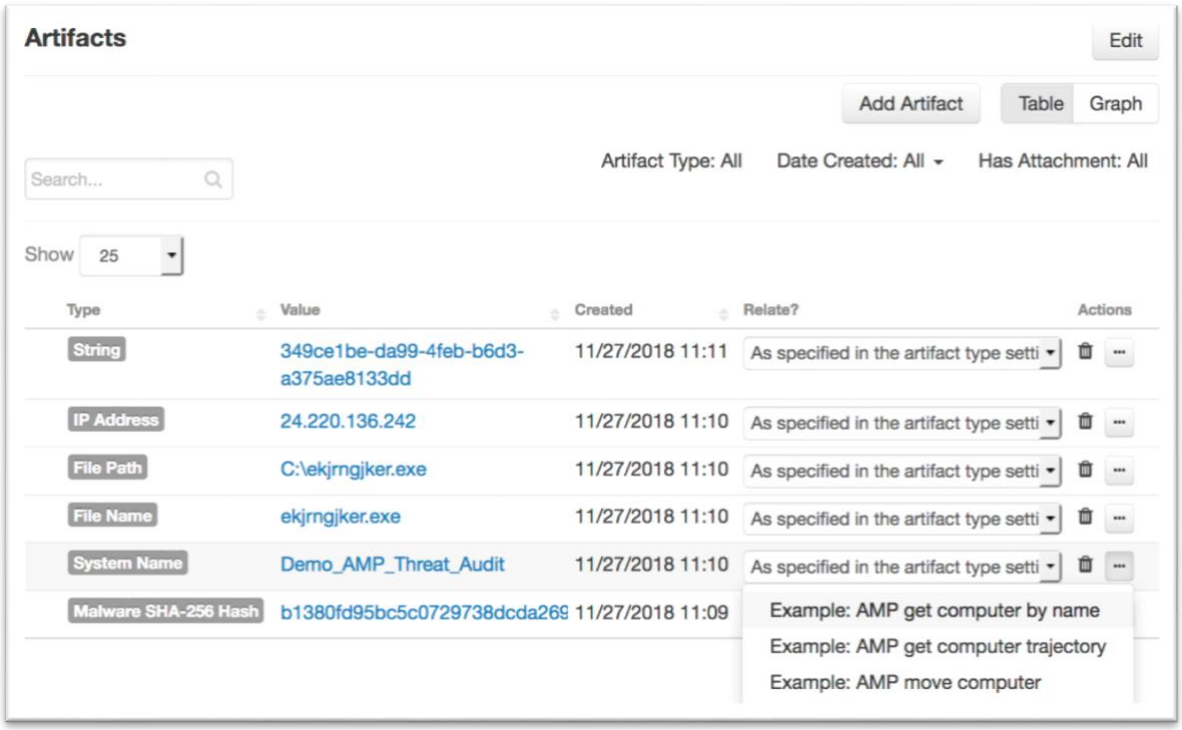


AMP: Get Computers

Returns a list of computers with agents deployed on them. You can use parameters to narrow the search by IP address or hostname.

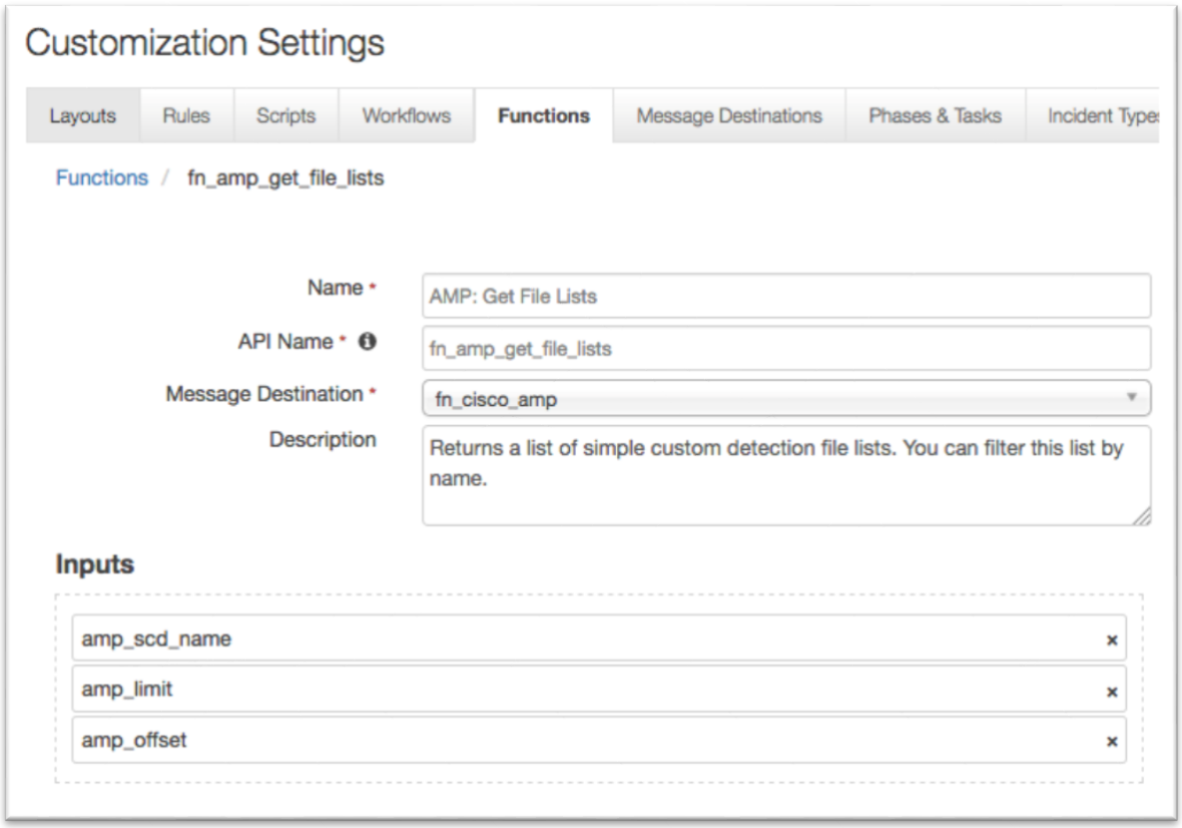


A Menu Item rule and workflow both called "Example: AMP get computer by name" are included. A user can invoke the workflow by clicking the Actions icon for an artifact, then selecting the rule. This rule is applicable for artifact types DNS Name and System Name.

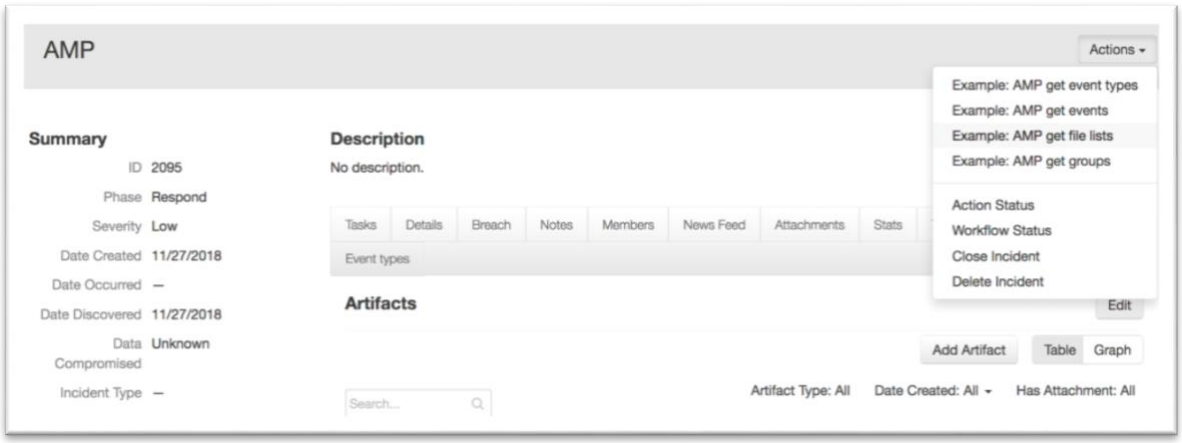


AMP: Get File Lists

Returns a list of simple custom detection file lists. You can filter this list by name.

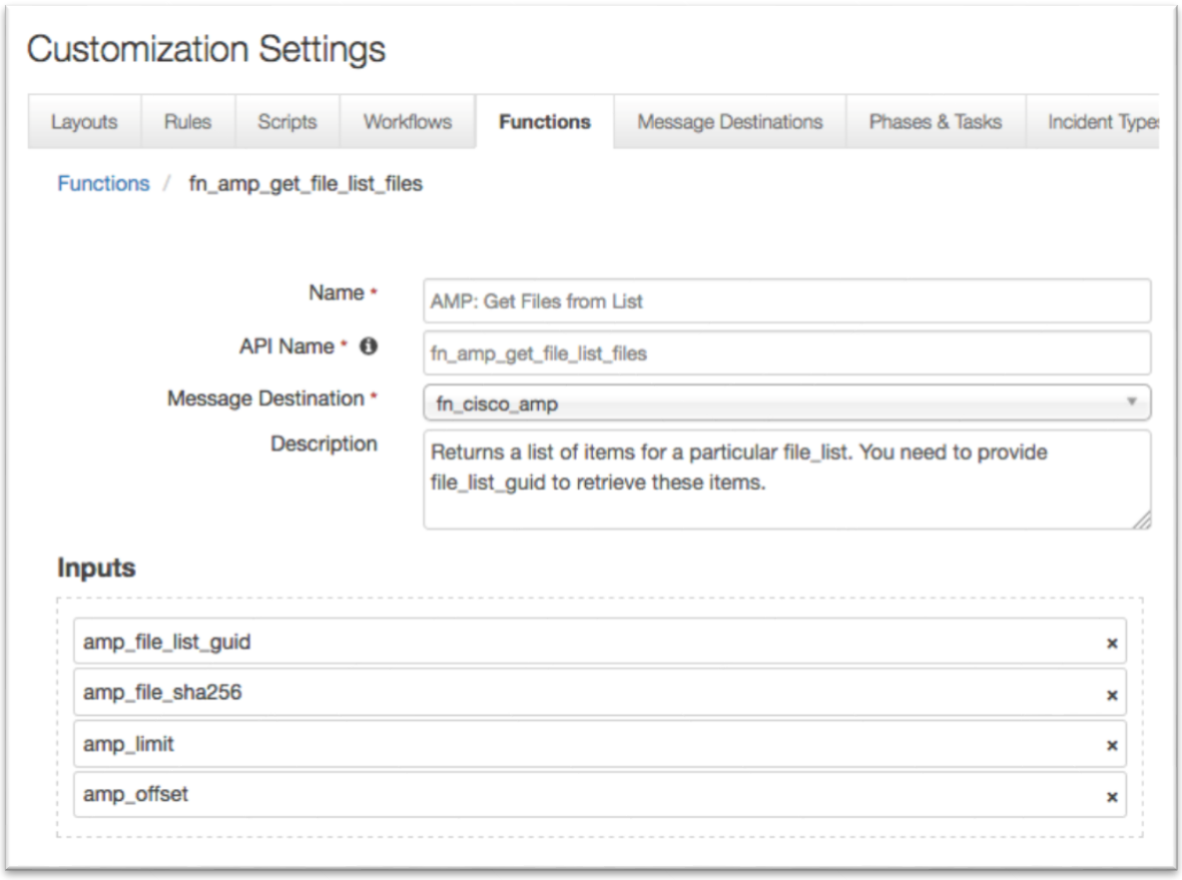


A Menu Item rule and workflow, both called “Example: AMP get file lists”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down menu of an incident.

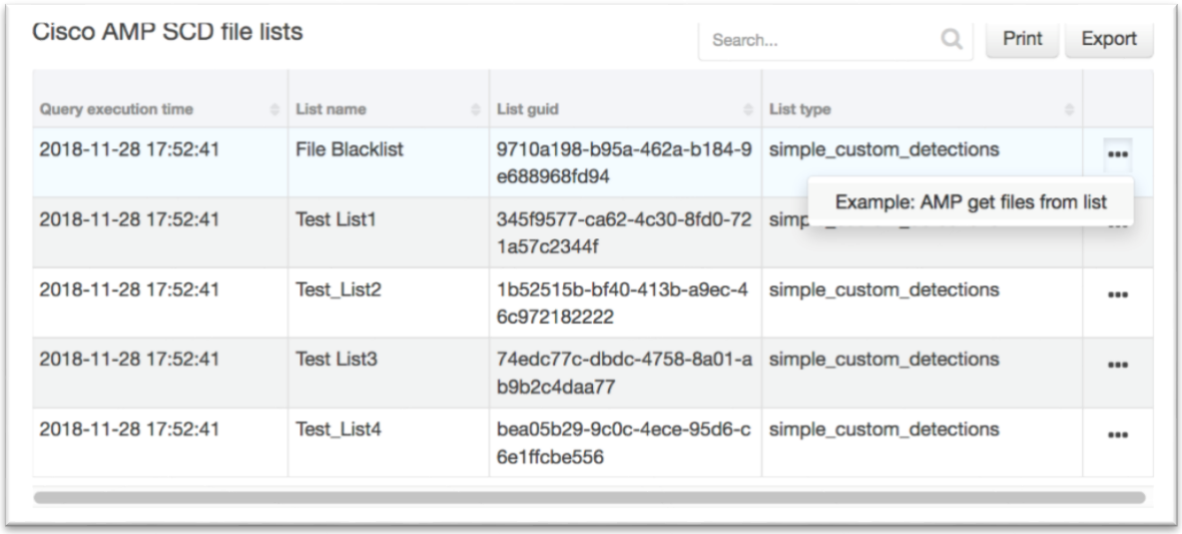


AMP: Get Files from List

Returns a list of items for a particular file_list. You need to provide the file_list_guid to retrieve these items.

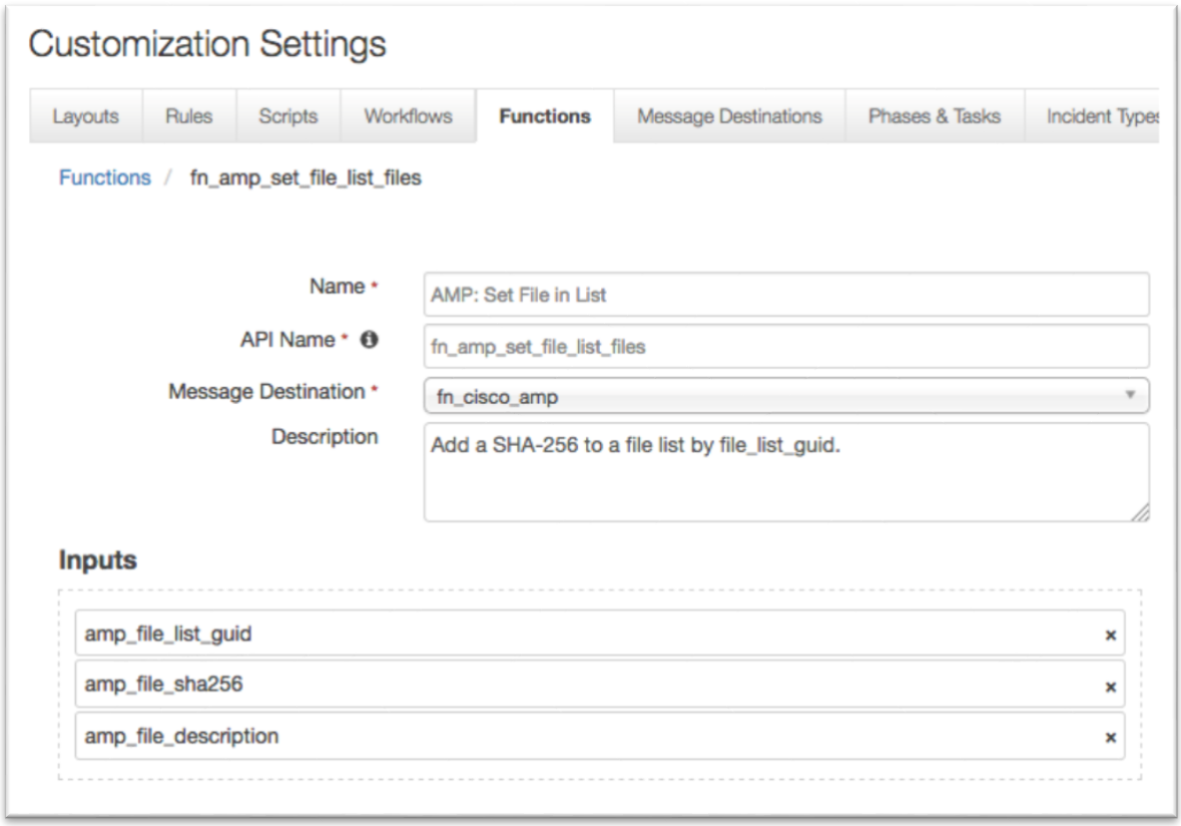


A Menu Item rule and workflow, both called “Example: AMP get files from list”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions menu of a row in data table “Cisco AMP SCD File Lists”.

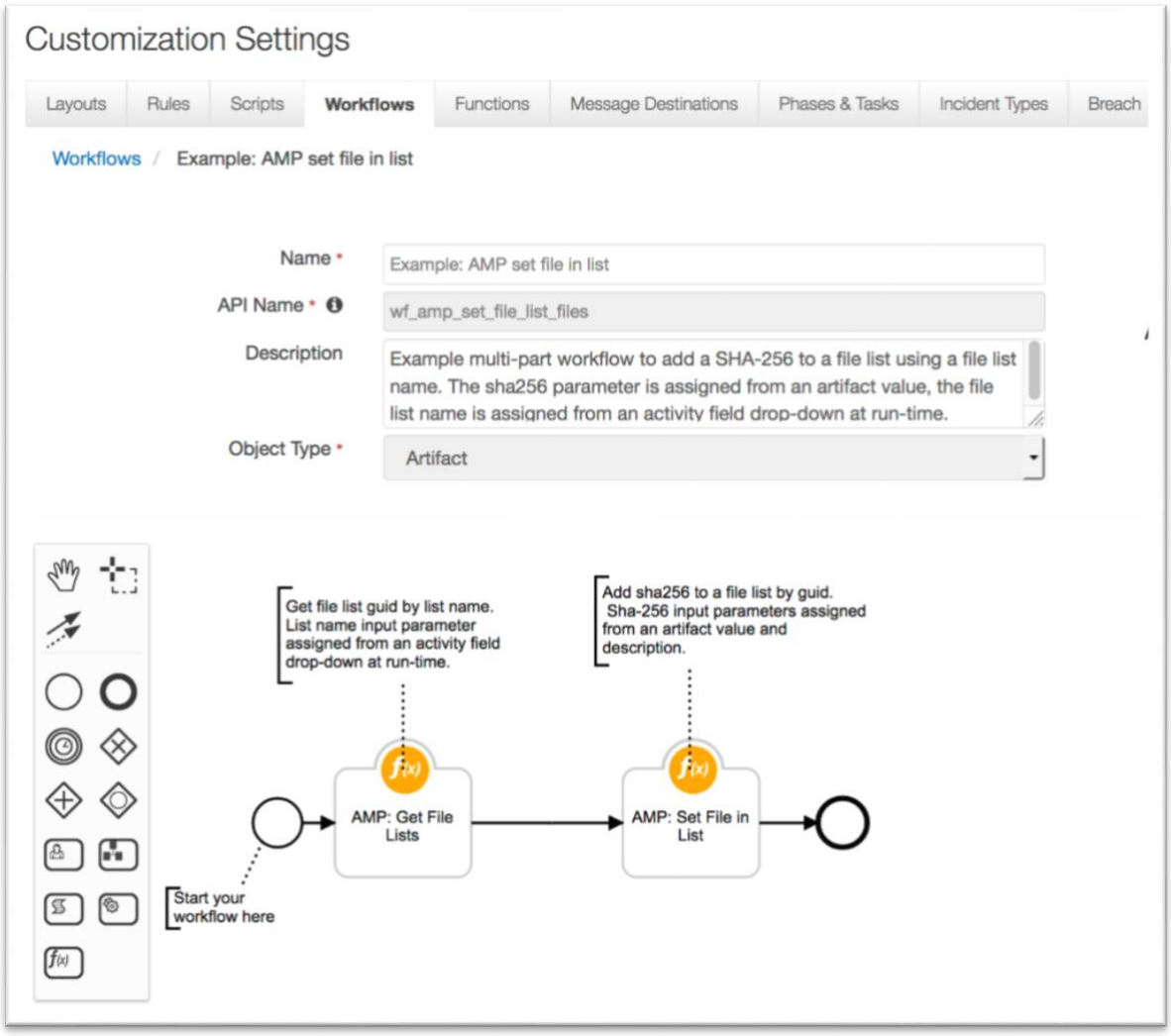


AMP: Set File in List

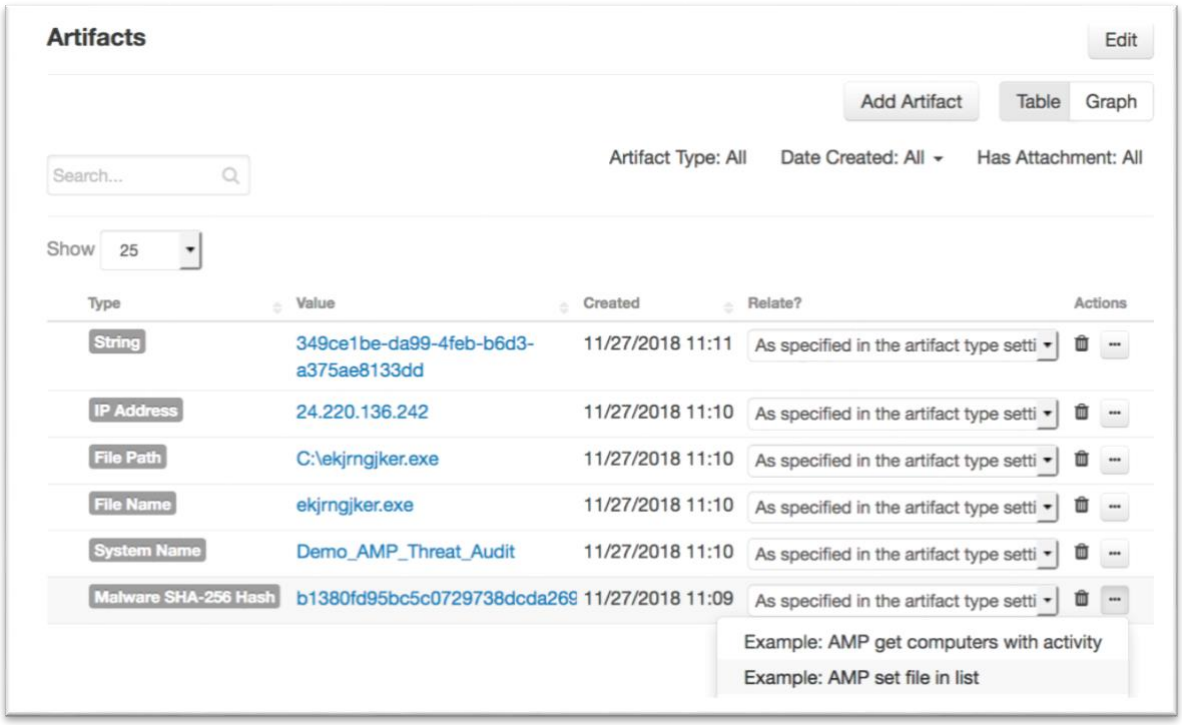
Adds a SHA-256 to a file list by file_list_guid.



A Menu Item rule and workflow, both called “Example: AMP set file in list”, are included. This is a multi-step workflow where the first function is used to provide the input value for file list guid in the second function.

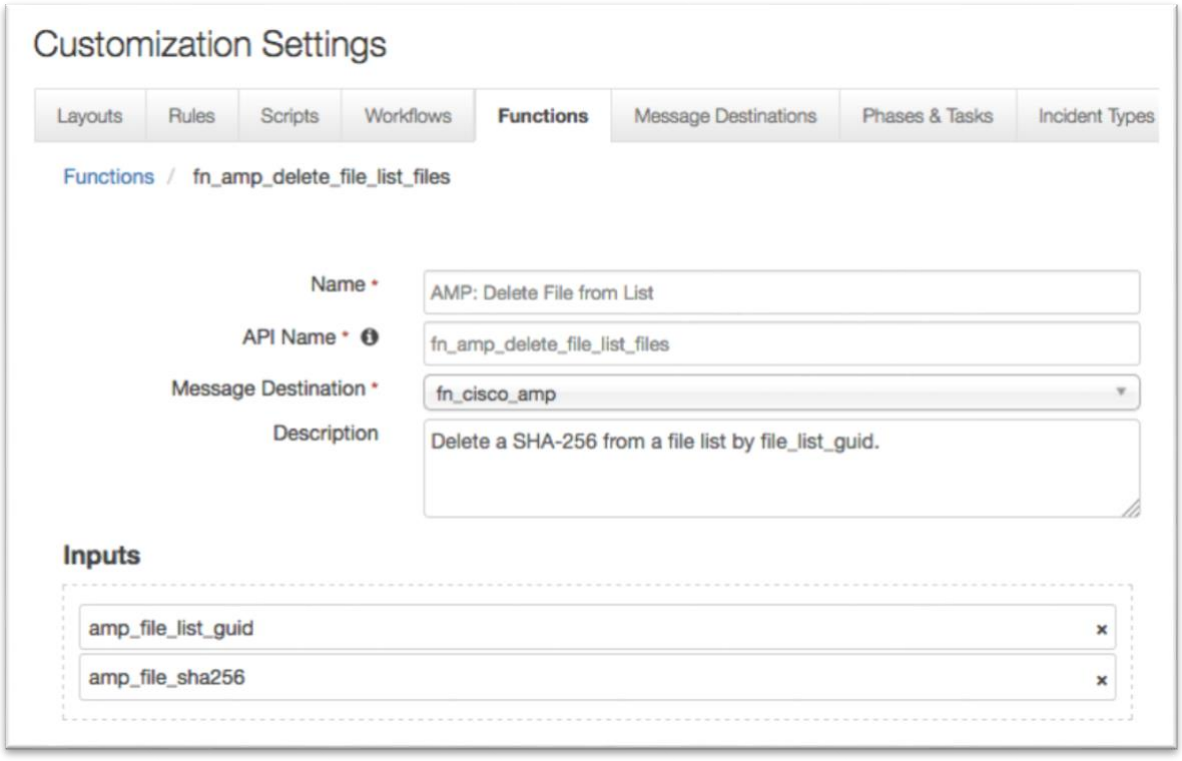


A user can invoke the workflow by clicking the Actions icon for an artifact then selecting a rule. This rule is applicable for artifact type Malware SHA-256 Hash.

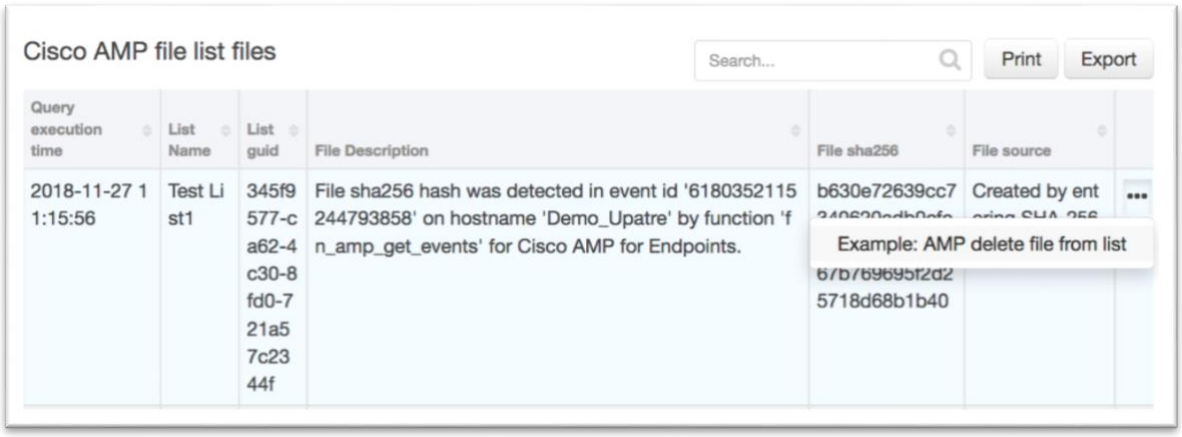


AMP: Delete File from List

Deletes a SHA-256 from a file list by file_list_guid.

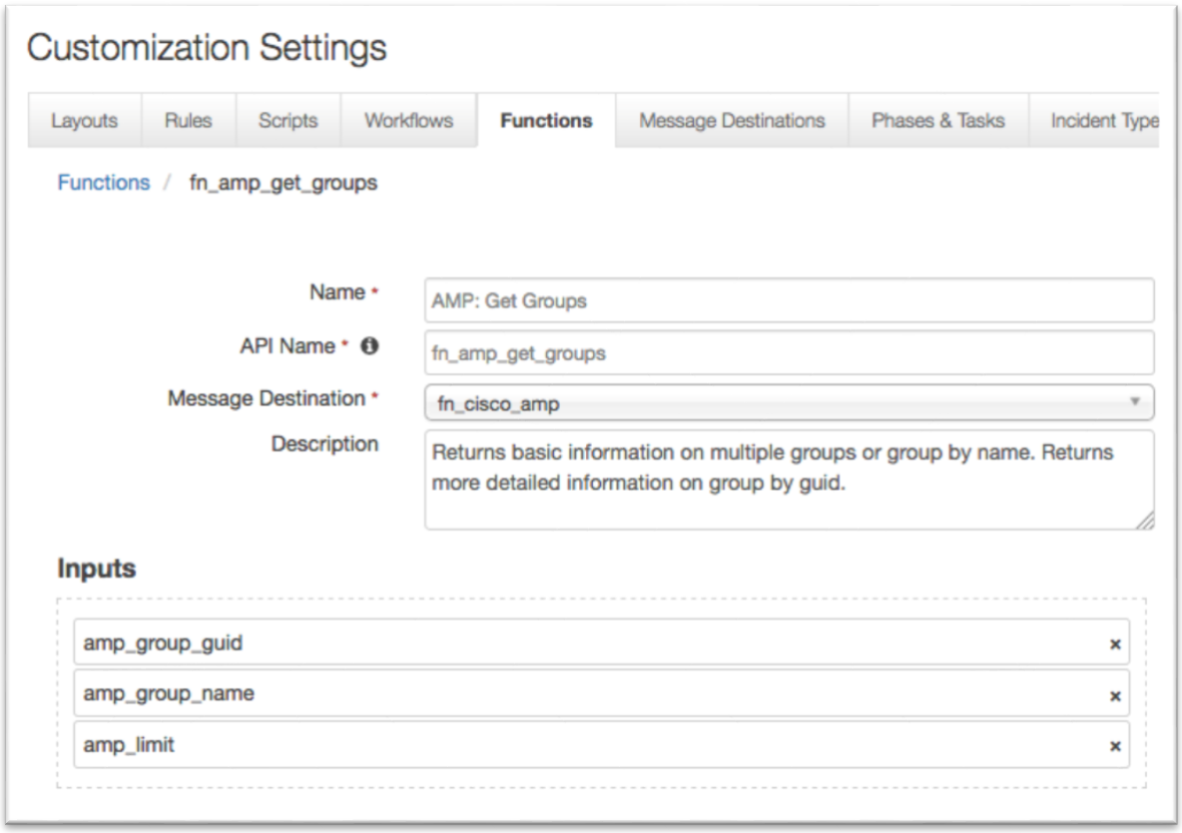


A Menu Item rule and workflow, both called “Example: AMP delete file from list”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions menu of a row in data table “Cisco AMP File List files”.

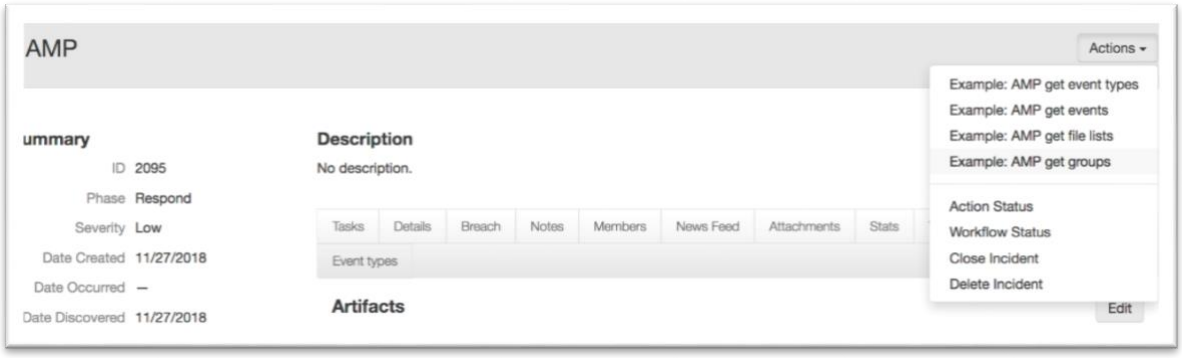


AMP: Get Groups

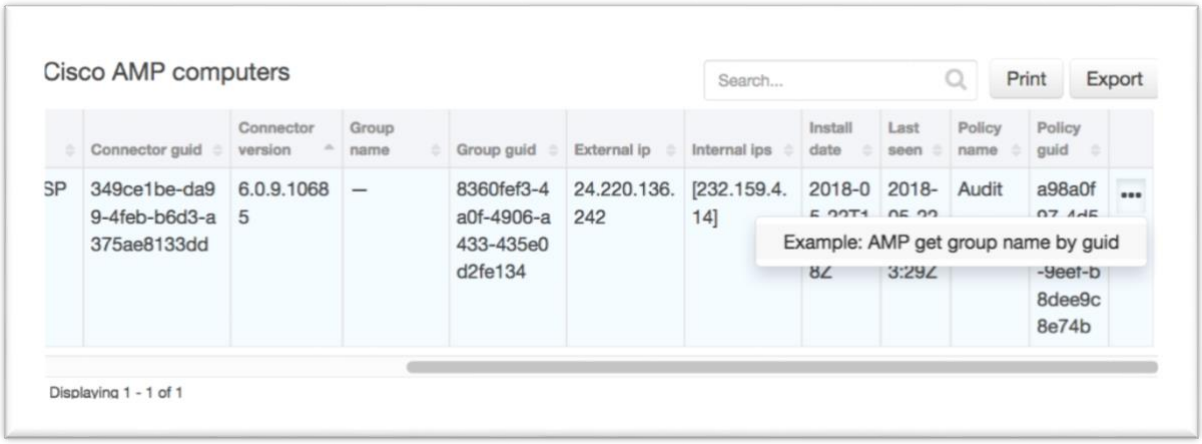
Returns basic information on multiple groups or group by name. Returns more detailed information on group by guid.



A Menu Item rule and workflow, both called “Example: AMP get groups”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions drop-down menu of an incident.

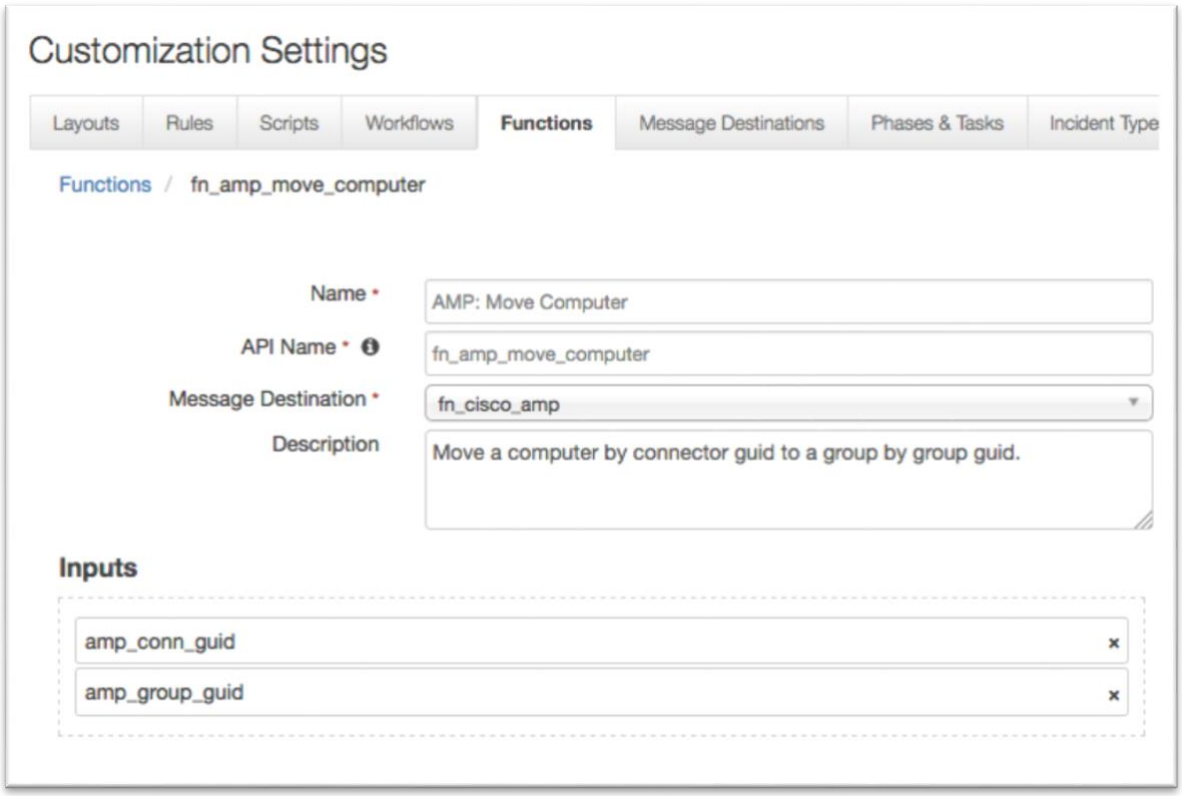


A Menu Item rule and workflow, both called “Example: AMP get group name by guid”, are included. A user can invoke the workflow by right-clicking on this rule from the Actions menu of a row in data table “Cisco AMP Computers”.

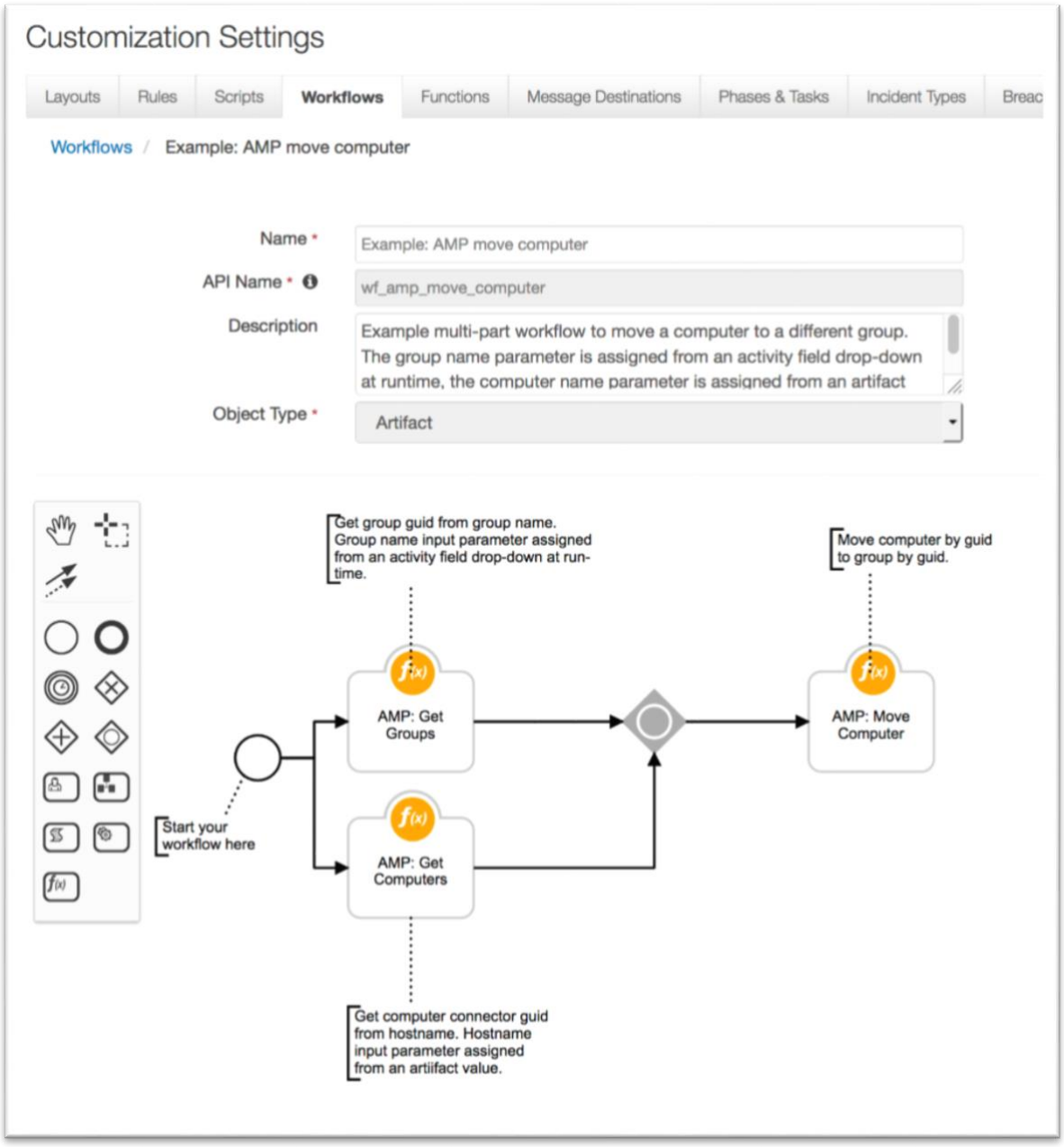


AMP: Move Computer

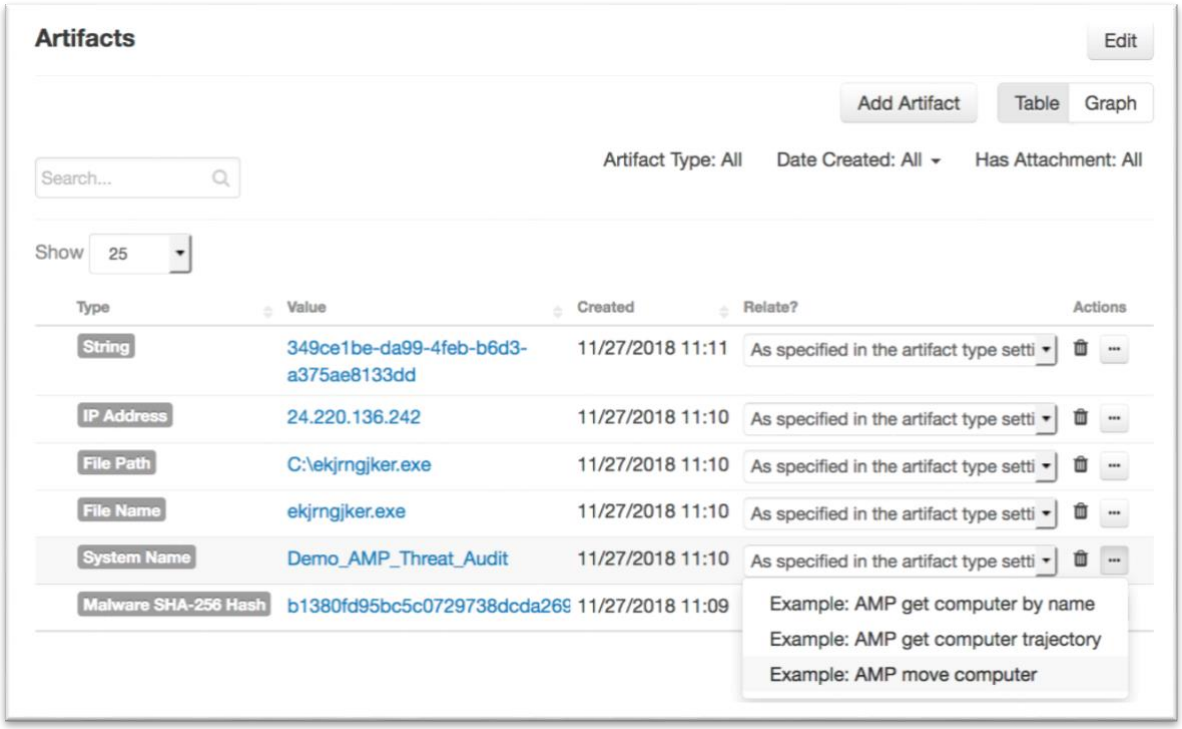
Moves a computer by connector guid to a group by group guid.



A Menu Item rule and workflow, both called “Example: AMP move computer”, are included. This is a multi-step workflow where two functions are used to provide input values for connector guid and group guid for the third function.



A user can invoke the workflow by clicking the Actions icon for an artifact then selecting the rule. This rule is applicable for artifact types DNS Name and System Name.

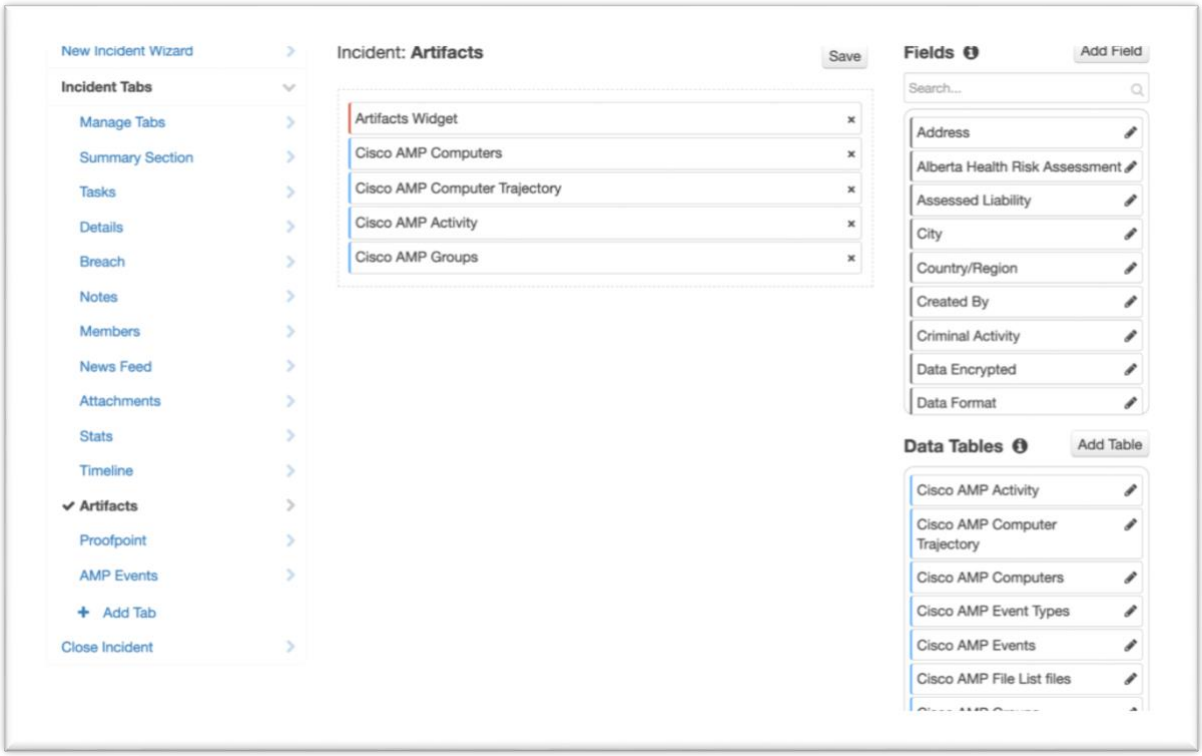


Resilient Platform Configuration

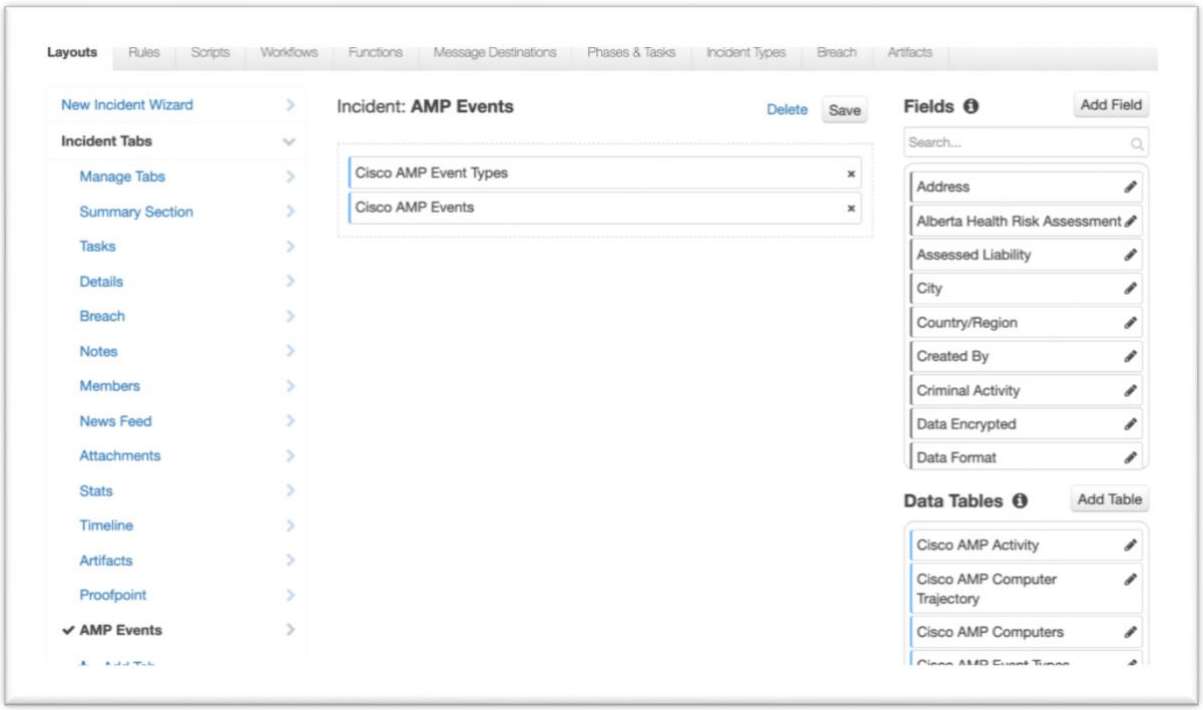
To display results, users need to manually add the provided data tables to the existing tabs (such as Artifacts) or create a new tabs:

1. Navigate to the Customization Settings and select the Layouts tab.
2. Select Artifacts.
3. Drag data tables to your Artifacts tab.
4. Click **Save**.

The following screenshot shows Cisco AMP for Endpoints data tables added to the Artifacts tab.



The following screenshot shows the **Cisco AMP Event Types and Event** data tables added to a new custom **AMP Events** tab.



It's suggested that you use logic groupings of data tables so not to burden one tab, such as Artifacts. The Artifacts Widget in Views can be replicated in multiple tabs but with different data tables.

Example Tabs:

Tab	Widgets and Data Tables
AMP Events	Cisco AMP Event Types Cisco AMP Events
Artifacts	Artifacts Widget Cisco AMP Computers Cisco AMP Computer Trajectory Cisco AMP Groups
AMP Files	Artifact Widget Cisco File List files Cisco SCD File Lists

Troubleshooting

There are several ways to verify the successful operation of a function.

- Resilient Action Status

  When viewing an incident, use the Actions menu to view Action Status. By default, pending and errors are displayed. Modify the filter for actions to also show Completed actions. Clicking on an action displays additional information on the progress made or what error occurred.

- Resilient Scripting Log

  A separate log file is available to review scripting errors. This is useful when issues occur in the pre-processing or post-processing scripts. The default location for this log file is: `/var/log/resilient-scripting/resilient-scripting.log`

- Resilient Logs

  By default, Resilient logs are retained at `/usr/share/co3/logs`. The `client.log` may contain additional information regarding the execution of functions.

- Resilient-Circuits

  The log is controlled in the `.resilient/app.config` file under the section `[resilient]` and the property `logdir`. The default file name is `app.log`. Each function will create progress information. Failures will show up as errors and may contain python trace statements.

Support

For additional support, contact support@resilientsystems.com.

Including relevant information from the log files will help us resolve your issue.