IBM Resilient

Incident Response Platform Integrations

Cisco Umbrella Investigate Function V1.0.0

Release Date: May 2018

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 Umbrella Investigate Function.

Overview

Umbrella Investigate is the interface to the security data collated by the Cisco Umbrella Investigate research team. The Cisco Umbrella Investigate REST API service allows for the querying of the Umbrella DNS database to show security events and correlations in their datasets. The Investigate REST API opens up the power of the Investigate classification results, correlation, and history and is based on the Umbrella global network, the world's largest security network.

The Cisco Umbrella Investigate integration with IBM Resilient allows querying of the Investigate datasets using their REST APIs and the returned results can be used to make customized updates to a Resilient instance such as updating incidents, artifacts, data-tables and so on.

There are 14 functions supplied in the Resilient Function package for Umbrella Investigate. The Functions interrogate the various REST APIs exposed by the Investigate service. There are also example workflows in the customizations section of the package which demonstrate usage of the Resilient Investigate Functions to update data tables.

The remainder of this document describes the included Functions, how to configure example custom workflows, and any additional customization options.

Installation

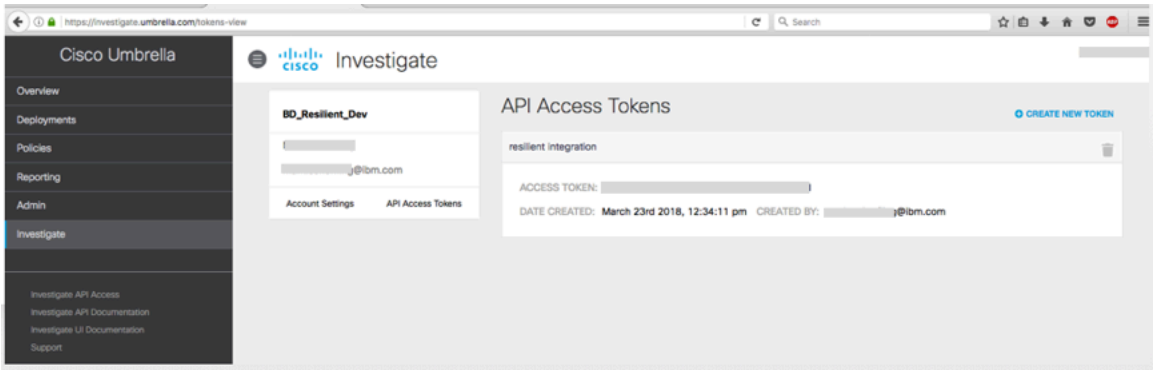
Before installing, verify that your environment meets the following prerequisites:

- Resilient platform must be version 30 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 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.)

Cisco Umbrella Investigate configurationThe Umbrella Investigate default base URL is https://investigate.api.umbrella.com/ .

You can override the base URL if required.

Access to the Cisco Umbrella Investigate REST API is allowed by providing an access token in the request. The access token is tied to a user account on the Umbrella platform.



More information is available here https://investigate-api.readme.io/docs/about-the-api-authentication.

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_umbrella_inv-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_umbrella_inv]` section, edit the settings as follows:

   ```
   base_url=https://investigate.api.umbrella.com/
   # The api_token will be supplied by Cisco will be in uuid format.
   api_token= abcd1234-a123-123a-123a-123456abcdef
   ```

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
```

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.

Configuration of 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]
```

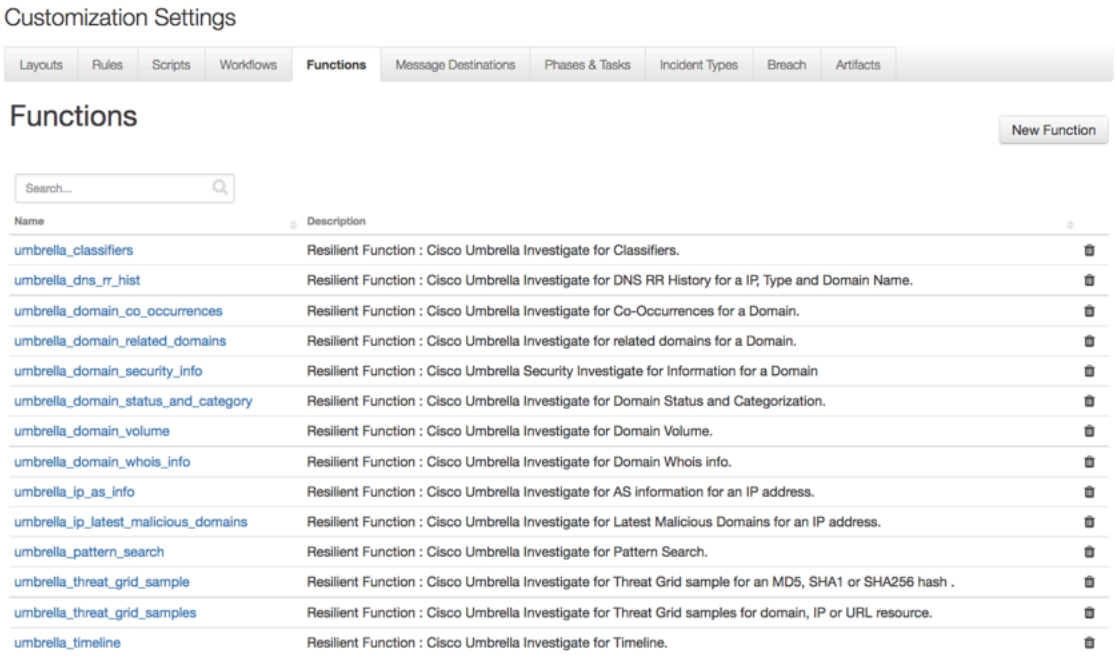
You can view log files for `systemd` and the `resilient-circuits` service using the `journalctl` command, as follows:

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

Customization Descriptions

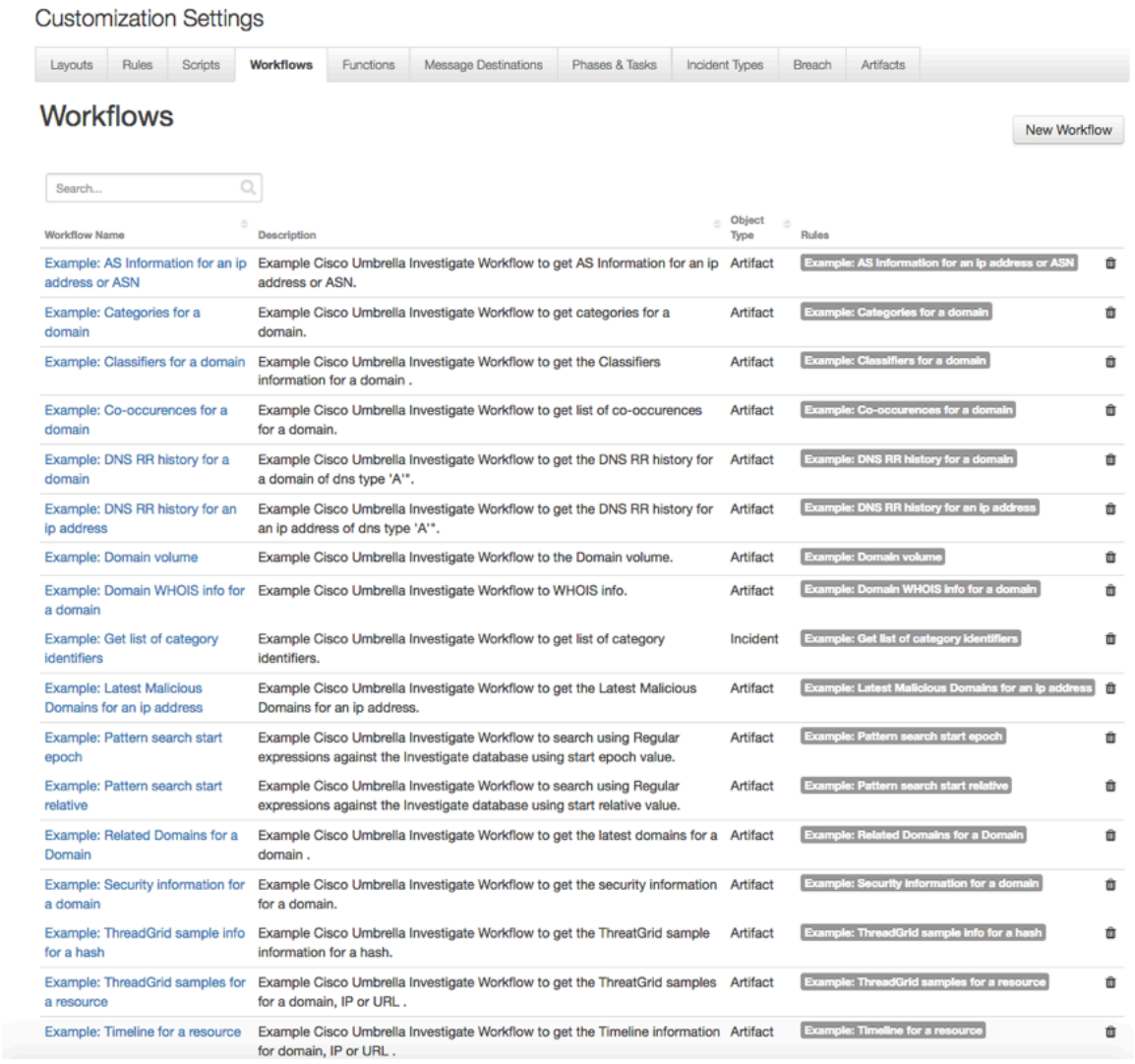
After the function package customizations are deployed to the Resilient instance, you can view the functions in the Functions tab in the Resilient platform, as shown in the following screenshot.

Functions

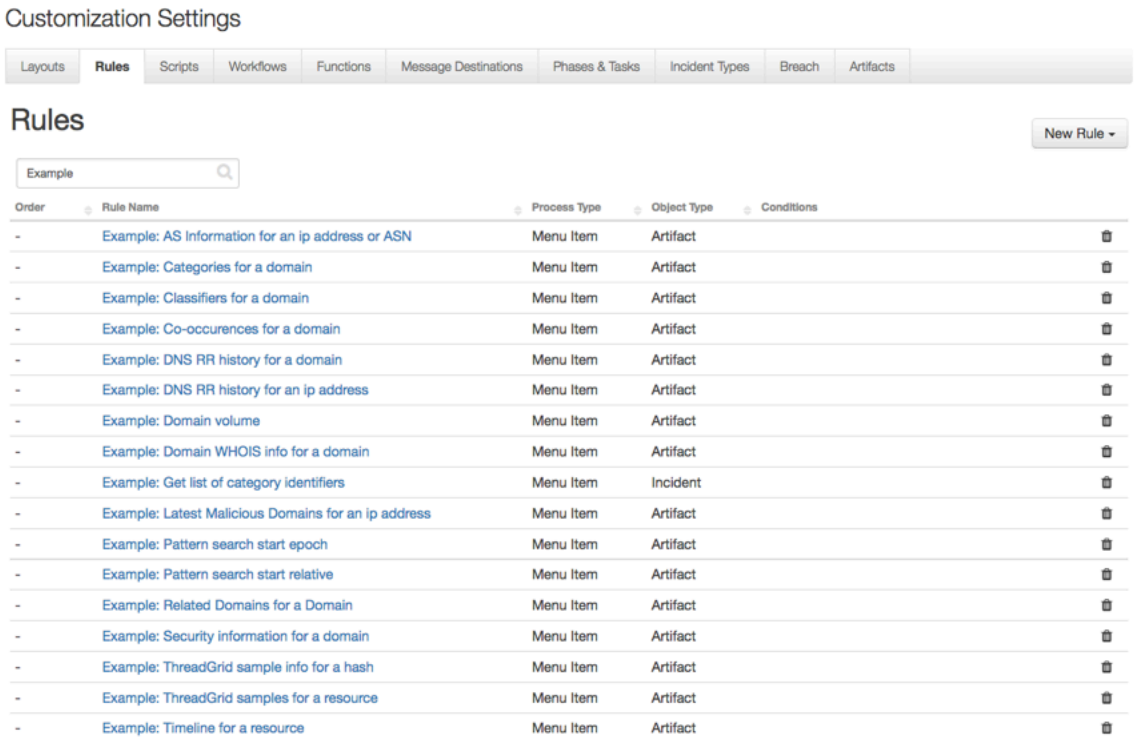


The package also includes example workflows, rules and data tables that show how you can use the functions. The Resilient user can copy and modify these Resilient objects for their own needs.

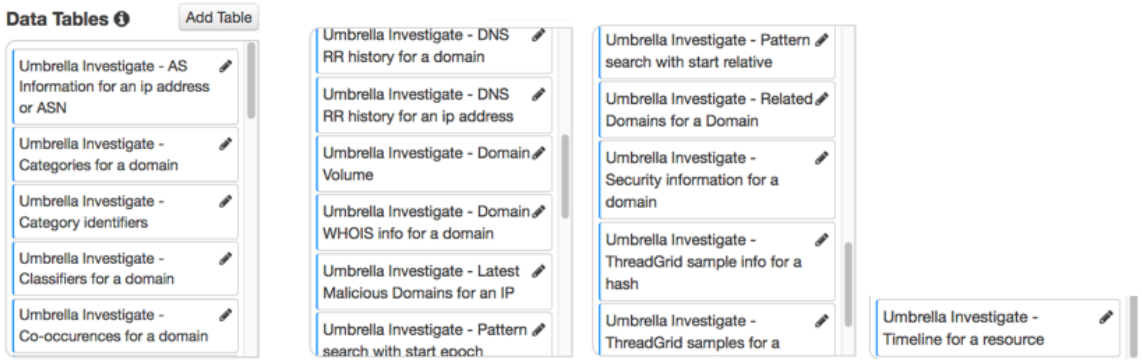
Workflows


Customization Settings
Layouts
Rules
Scripts
Workflows
Functions
Message Destinations
Phases & Tasks
Incident Types
Breach
Artifacts
Workflows
New Workflow
Search...
Workflow Name
Description
Object Type
Rules
Example: AS Information for an ip address or ASN
Example Cisco Umbrella Investigate Workflow to get AS Information for an ip address or ASN.
Artifact
Example: AS Information for an ip address or ASN
Example: Categories for a domain
Example Cisco Umbrella Investigate Workflow to get categories for a domain.
Artifact
Example: Categories for a domain
Example: Classifiers for a domain
Example Cisco Umbrella Investigate Workflow to get the Classifiers information for a domain .
Artifact
Example: Classifiers for a domain
Example: Co-occurences for a domain
Example Cisco Umbrella Investigate Workflow to get list of co-occurences for a domain.
Artifact
Example: Co-occurences for a domain
Example: DNS RR history for a domain
Example Cisco Umbrella Investigate Workflow to get the DNS RR history for a domain of dns type 'A'".
Artifact
Example: DNS RR history for a domain
Example: DNS RR history for an ip address
Example Cisco Umbrella Investigate Workflow to get the DNS RR history for an ip address of dns type 'A'".
Artifact
Example: DNS RR history for an ip address
Example: Domain volume
Example Cisco Umbrella Investigate Workflow to the Domain volume.
Artifact
Example: Domain volume
Example: Domain WHOIS info for a domain
Example Cisco Umbrella Investigate Workflow to WHOIS info.
Artifact
Example: Domain WHOIS info for a domain
Example: Get list of category identifiers
Example Cisco Umbrella Investigate Workflow to get list of category identifiers.
Incident
Example: Get list of category identifiers
Example: Latest Malicious Domains for an ip address
Example Cisco Umbrella Investigate Workflow to get the Latest Malicious Domains for an ip address.
Artifact
Example: Latest Malicious Domains for an ip address
Example: Pattern search start epoch
Example Cisco Umbrella Investigate Workflow to search using Regular expressions against the Investigate database using start epoch value.
Artifact
Example: Pattern search start epoch
Example: Pattern search start relative
Example Cisco Umbrella Investigate Workflow to search using Regular expressions against the Investigate database using start relative value.
Artifact
Example: Pattern search start relative
Example: Related Domains for a Domain
Example Cisco Umbrella Investigate Workflow to get the latest domains for a domain .
Artifact
Example: Related Domains for a Domain
Example: Security information for a domain
Example Cisco Umbrella Investigate Workflow to get the security information for a domain.
Artifact
Example: Security information for a domain
Example: ThreadGrid sample info for a hash
Example Cisco Umbrella Investigate Workflow to get the ThreatGrid sample information for a hash.
Artifact
Example: ThreadGrid sample info for a hash
Example: ThreadGrid samples for a resource
Example Cisco Umbrella Investigate Workflow to get the ThreatGrid samples for a domain, IP or URL .
Artifact
Example: ThreadGrid samples for a resource
Example: Timeline for a resource
Example Cisco Umbrella Investigate Workflow to get the Timeline information for domain, IP or URL .
Artifact
Example: Timeline for a resource

Rules



Data tables

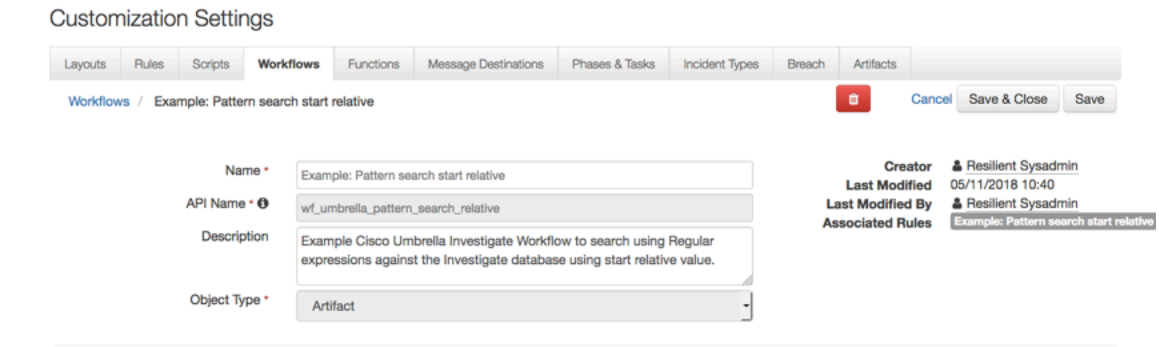


Function arguments

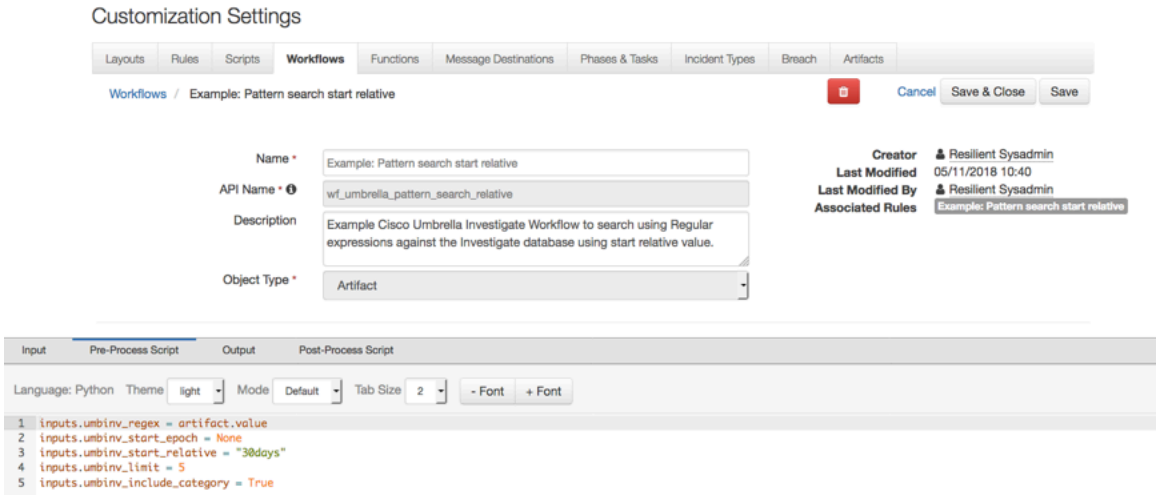
Refer to the Cisco Umbrella API documentation on the use of the Umbrella Investigate arguments. The Resilient Functions all use input parameters starting with *umbinv_* examples include `umbinv_domains`, *umbinv_showlabels* and *umbinv_status_endpoint*. These are equivalent to the parameters used in the REST API call. (c.f. https://investigate-api.readme.io/docs/introduction-to-cisco-investigate/).

See the Investigate Function in the workflows: *Example: Pattern search start relative*. Review the *Input* and/or *Pre-Process Script* tabs when editing the function within a workflow for the execution settings.

Input tab



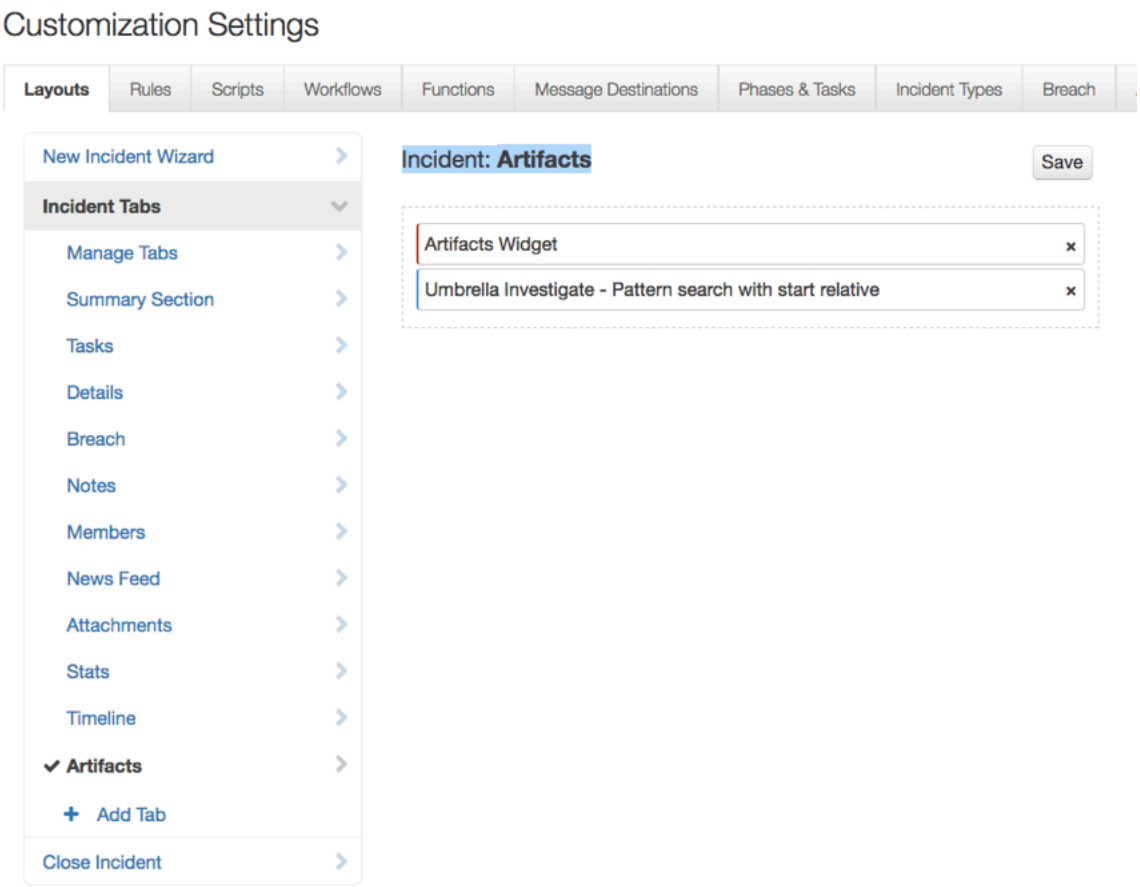
Pre-Process Script tab



Before using a workflow

- Change the pre-defined value in either the *Input* or *Pre-Processing Script* tab for your environment (Note: Definitions in the *Pre-Processing Script* tab will over-ride any *Input* tab settings.)
- Add the required data-table to the incident artifacts tab. (Note: Most of the workflows are configured for Artifact object type with the exception of the workflow *Example: Get list of category identifiers* which is configured for Incident object type.)

Add data table artifact tab



Relationships between Rules, Workflow Functions and data tables.

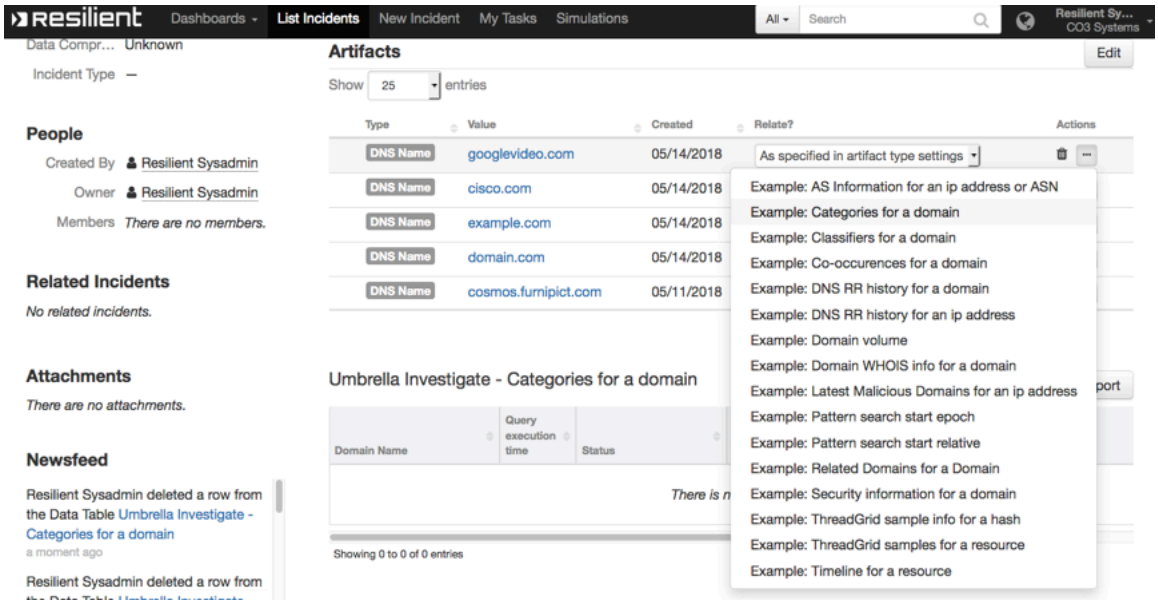
The example workflows each has a function and a data-table associated with it as shown in the following table.

Rule	Workflow	Function	Data table
Example: AS Information for an ip address or ASN	*Example: AS Information for an ip address or ASN*	*umbrella_ip_as_info*	*Umbrella Investigate - AS Information for an ip address or ASN*
Example: Get list of category identifiers	*Example: Get list of category identifiers*	*umbrella_domain_status_and_category*	*Umbrella Investigate - Category identifiers*
Example: Categories for a domain	*Example: Categories for a domain*	*umbrella_domain_status_and_category*	*Umbrella Investigate - Categories for a domain*
Example: Classifiers for a domain	*Example: Classifiers for a domain*	*umbrella_classifiers*	*Umbrella Investigate - Classifiers for a domain*
Example: DNS RR history for a domain	*Example: DNS RR history for a domain*	*umbrella_domain_co_occurrences*	*Umbrella Investigate - Co-occurences for a domain*
Example: DNS RR history for a domain	*Example: DNS RR history for a domain*	*umbrella_dns_rr_hist*	*Umbrella Investigate - DNS RR history for a domain*
Example: DNS RR history for an ip address	*Example: DNS RR history for an ip address*	*umbrella_dns_rr_hist*	*Umbrella Investigate - DNS RR history for an ip address*
Example: Domain volume	*Example: Domain volume*	*umbrella_domain_volume*	*Umbrella Investigate - Domain Volume*
Example: Domain WHOIS info for a domain	*Example: Domain WHOIS info for a domain*	*umbrella_domain_whois_info*	*Umbrella Investigate - Domain WHOIS info for a domain*
Example: Latest Malicious Domains for an ip address	*Example: Latest Malicious Domains for an ip address*	*umbrella_ip_latest_malicious_domains*	*Umbrella Investigate - Latest Malicious Domains for an IP*
Example: Pattern search start epoch	*Example: Pattern search start epoch*	*umbrella_pattern_search*	*Umbrella Investigate - Pattern search with start epoch*
Example: Pattern search start relative	*Example: Pattern search start relative*	*umbrella_pattern_search*	*Umbrella Investigate - Pattern search with start relative*
Example: Related Domains for a Domain	*Example: Related Domains for a Domain*	*umbrella_domain_related_domains*	*Umbrella Investigate - Related Domains for a Domain*

Rule	Workflow	Function	Data table
Example: Security information for a domain	*Example: Security information for a domain*	*umbrella_domain_security_info*	*Umbrella Investigate - Security information for a domain*
Example: ThreadGrid sample info for a hash	*Example: ThreadGrid sample info for a hash*	*umbrella_threat_grid_sample*	*Umbrella Investigate - ThreadGrid sample info for a hash*
Example: ThreadGrid samples for a resource	*Example: ThreadGrid samples for a resource*	*umbrella_threat_grid_samples*	*Umbrella Investigate - ThreadGrid samples for a resource*
Example: Timeline for a resource	*Example: Timeline for a resource*	*umbrella_timeline*	*Umbrella Investigate - Timeline for a resource*

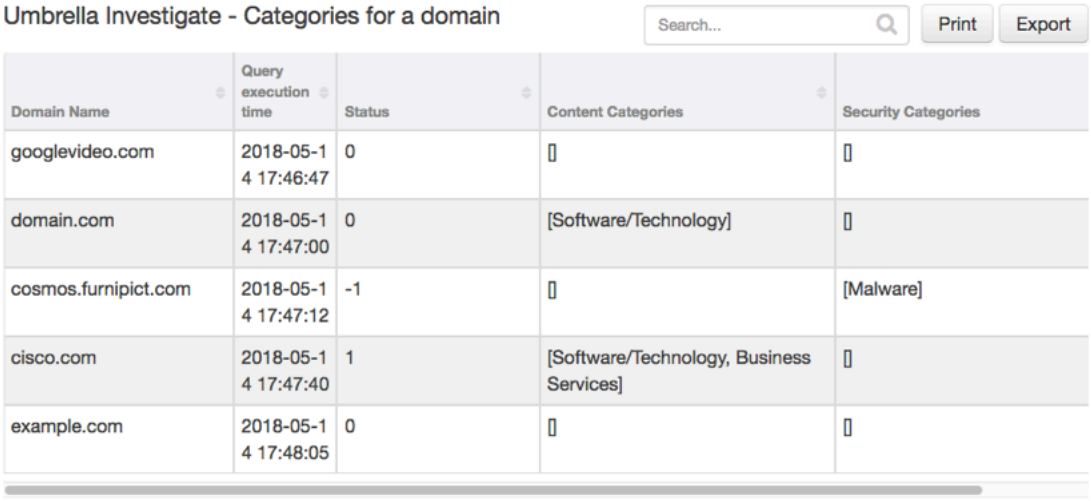
Workflow execution

To run a Cisco Umbrella Investigate query, click on the Actions icon for an Artifact, then select a rule and click on the rule. This will execute the corresponding workflow against that particular Artifact. In the following example the user executes the rule *Example: Categories for a domain* and the corresponding data table will get updated as shown below, where the artifact values are domain names.



Data table *Umbrella Investigate - Categories for a domain* (api name *umbinv_categories_for_a_domain*) will get updated with an entry for each domain that the rule/workflow is run against.

Note: Some of the Workflows with add more than one row per artifact for each execution.



Umbrella Investigate - Categories for a domain

Search... Print Export

Domain Name	Query execution time	Status	Content Categories	Security Categories
googlevideo.com	2018-05-14 17:46:47	0	[]	[]
domain.com	2018-05-14 17:47:00	0	[Software/Technology]	[]
cosmos.furnipict.com	2018-05-14 17:47:12	-1	[]	[Malware]
cisco.com	2018-05-14 17:47:40	1	[Software/Technology, Business Services]	[]
example.com	2018-05-14 17:48:05	0	[]	[]

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.