IBM Resilient

Incident Response Platform Integrations

Cisco Enforcement Function V1.0.1

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 Cisco Enforcement Function.

Overview

Cisco Umbrella Enforcement is a DNS-based solution for securing an Enterprise's ports and protocols. This Functions-based integration package accesses the Cisco Enforcement APIs to perform the following operations:

- Request the list of domains retained within Cisco Enforcement
- Add a domain to Cisco Enforcement
- Remove a domain from Cisco Enforcement

The remainder of this document describes each included function, how to configure them in custom workflows, and any additional customization options.

Installation

App Host

All the components for running this integration in a container already exist when using the App Host app.

To install,

- Navigate to Administrative Settings and then the Apps tab.
- Click the Install button and select the downloaded file: app-fn_cisco_enforcement-x.x.x.zip.
- Go to the Configuration tab and edit the app.config file, editing the settings for Cisco Enforcement.

Config Required Example Description

Config	Required	Example	Description
url	Yes	https://s-platform.api.opendns.com/1.0	Base URL for all API calls to Cisco Enforcement
api_token	Yes	11111-2222-3333-44444	API token used for API calls
protocol_version	Yes	1.0a	Version of API used.
provider_name	Yes	Security Platform	Required settings for API integrations
https_proxy	No	https:/proxy_host.com:<port>	Optional https proxy
http_proxy	No	http:/proxy_host.com:<port>	Optional http proxy

Integrations Server

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

- Resilient platform is version 30 or later.
- You 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 need to 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 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.)

Install the Python components

The functions package contains Python components that will be 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.

Ensure that the environment is up to date,

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

To install the package, you must first unzip it then install the package as follows:

```
sudo pip install --upgrade fn_cisco_enforcement-<version>.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.

Perform the following to configure and run the integration:

1. Using sudo, become the integration user.

```
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.
   a. In the [resilient] section, ensure that you provide all the information needed to connect to the Resilient platform.
   b. In the [fn_cisco_enforcement] section, edit the settings as follows:

```
# change as necessary
url=https://s-platform.api.opendns.com/1.0

# replace with api token to your subscription
api_token=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx

# these settings need not change
protocol_version=1.0a
provider_name=Security Platform Deploy customizations to the Resilient
platform
# Uncomment to specify proxies needed
#https_proxy=
#http_proxy=
```

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

Deploy these customizations to the Resilient platform with the following command:

```
resilient-circuits customize
```

Answer 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 with 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 restartability

For normal operation, resilient-circuits must run continuously. The recommend way to do this is to configure it to automatically run at startup. On a Red Hat appliance, this is done using a systemd unit file such as the one below. You may need to change the paths to your working directory and app.config.

The unit file should be named ‘resilient_circuits.service’:

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

The contents (review and edit 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.
lock
```

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

Ensure that the service unit file is correctly permissioned:

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

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

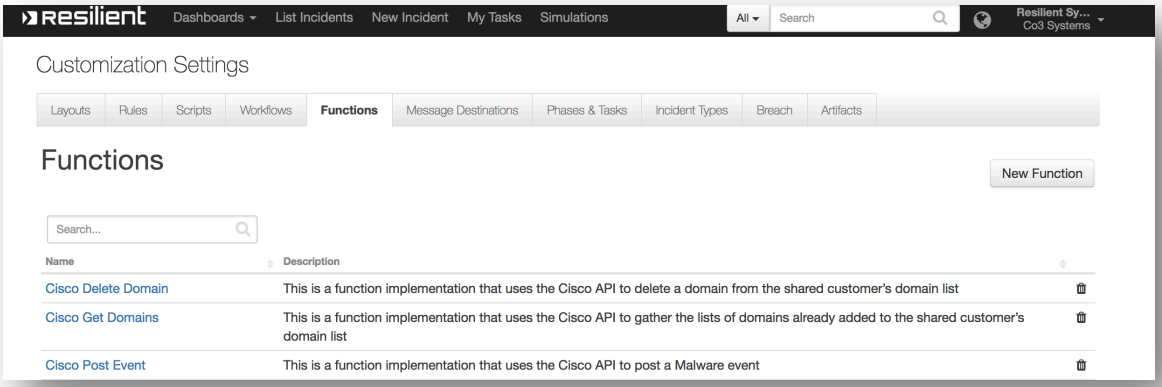
```
sudo systemctl resilient_circuits [start|stop|restart|status]
```

Log files for systemd and the resilient-circuits service can be viewed through the journalctl command:

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

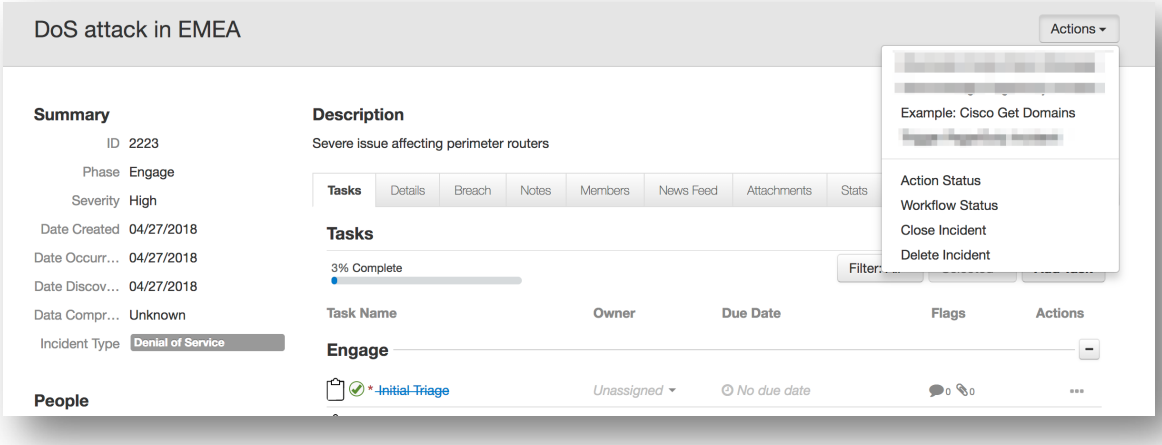
Function Descriptions

Once the integration package deploys the function(s), 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 and modify these workflows and rules for your own needs.



Cisco Get Domains

This incident level function will request a list all domains managed with Cisco Enforcement. The example workflow saves the results in a data table for later review and action.

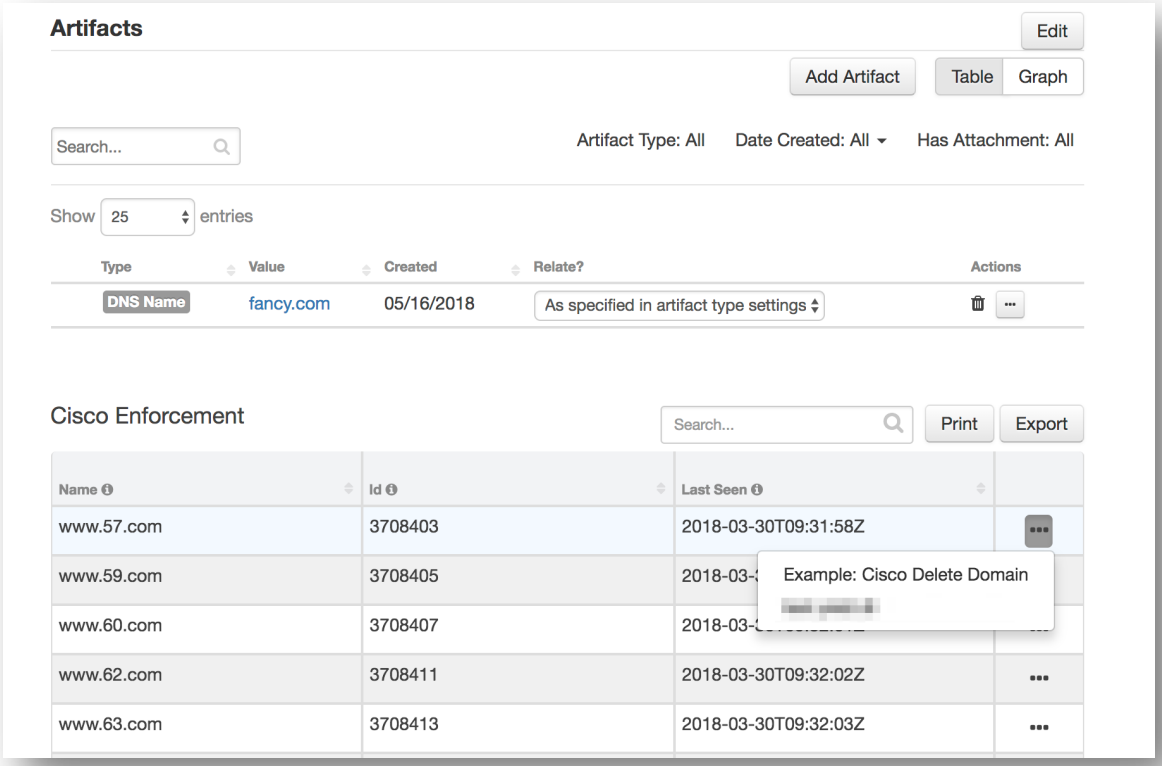


Cisco Add Domains

This function will add a domain to Cisco Enforcement. The example rule and workflow reference an artifact of type URL, URI Path and DNS Name.

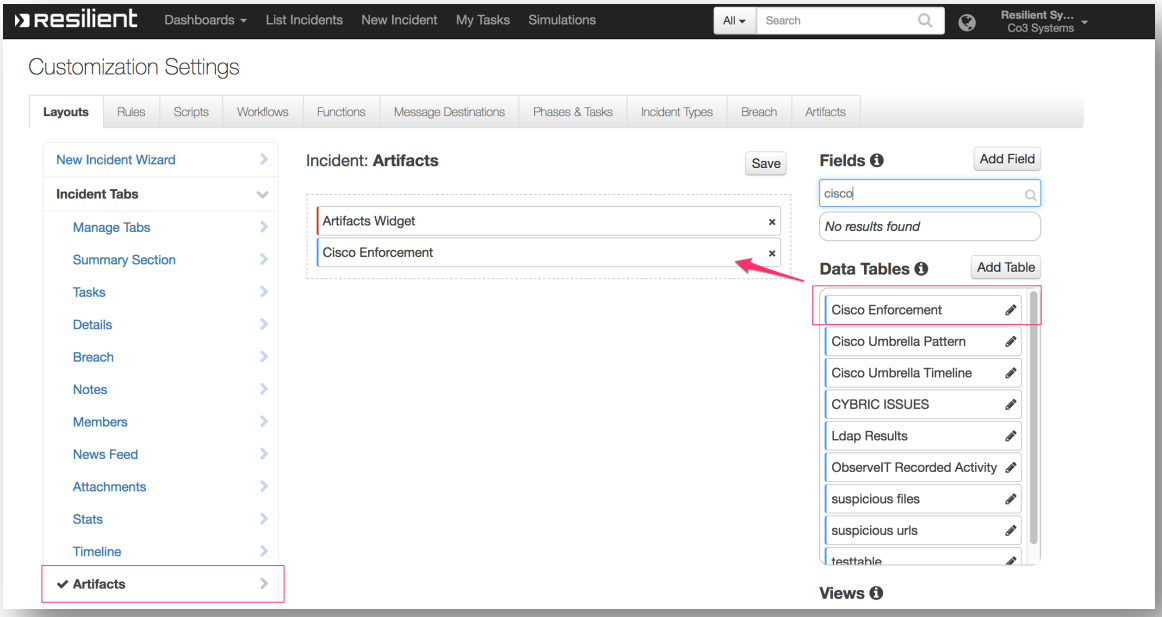
Cisco Delete Domain

This function will delete a domain in Cisco Enforcement. The example workflow references the domain entry collected from the Cisco Get Domains function.



Resilient Platform Configuration

In order to display the list of domains with artifacts, navigate to Layouts under Customize Settings and drag the Cisco Enforcement data table to the Artifacts layout and save. This table is shared among all artifacts and is accumulative with every invocation of the Cisco Get Domains function.



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

Use the IBM Support portal to open a case on this app. Also reference the Resilient Community for any discussion between customers and IBM.

History

Version	Date	Notes
1.0.1	Sept. 2020	App Host and Proxy support added
1.0.0	June 2018	Initial release