IBM Resilient

Incident Response Platform Integrations

McAfee ePO Function V1.0.0

Release Date: April 2018

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 McAfee ePO Function.

Overview

The McAfee ePO function contains the ability to apply a tag to a system managed in ePO from the Resilient Platform.

This document describes the McAfee ePO function, its customization options, and how to configure it in custom workflows.

Installation

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,

```
sudo pip install --upgrade fn_mcafee_epo-<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.

Perform the following to configure and run the integration:

1. Using 'sudo', become the integration user.

   ```
   sudo su - integration
   ```

2. Create or update the resilient-circuits configuration file.

   ```
   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_mcafee_epo] section, edit the settings as follows:

   ```
   ePO_url=https://<your_epo_server>:<port>
   epo_username=<your_epo_username>
   epo_password=<your_epo_password>
   epo_trust_cert=[true|false]
   ```

   Use false for self-signed SSL certificates.

Deploy customizations to the Resilient platform

The package contains the function definition that you can use in workflows, and an example workflow and rule that show how to use the function.

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

```
resilient-circuits customize
```

Answer the prompts to deploy the function, message destination, workflow and rule. The following data will be imported.

```
Function inputs: mcafee_epo_systems, mcafee_epo_tag
Message Destination: McAfee ePO Message Destination
Function: McAfee Tag an ePO asset
Workflow: (Example) McAfee Tag ePO asset workflow
Rule: (Example) McAfee Tag ePO Asset Shut Down
```

Run the integration framework

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. Below shows a successful connection to the Resilient platform and loading of components

```
2018-04-10 12:05:09,686 INFO [app] Resilient server: 9.108.163.130
2018-04-10 12:05:09,687 INFO [app] Resilient org: TestOrg
2018-04-10 12:05:09,687 INFO [app] Logging Level: INFO
2018-04-10 12:05:09,688 WARNING [co3] Unverified HTTPS requests
(cafile=false).
2018-04-10 12:05:10,142 INFO [app] Components auto-load directory: (none)
2018-04-10 12:05:10,306 INFO [component_loader] Loading 1 components
2018-04-10 12:05:10,307 INFO [component_loader]
'fn_mcafee_epo.components.mcafee_tag_an_epo_asset.FunctionComponent'
loading
2018-04-10 12:05:10,308 WARNING [actions_component] Unverified STOMP TLS
certificate (cafile=false)
2018-04-10 12:05:10,309 INFO [stomp_component] Connect to
9.108.163.130:65001
2018-04-10 12:05:10,310 INFO [actions_component]
'fn_mcafee_epo.components.mcafee_tag_an_epo_asset.FunctionComponent'
function 'mcafee_tag_an_epo_asset' registered to
'mcafee_epo_message_destination'
2018-04-10 12:05:10,310 INFO [app] Components loaded
2018-04-10 12:05:10,312 INFO [app] App Started
2018-04-10 12:05:10,414 INFO [actions_component] STOMP attempting to
connect
2018-04-10 12:05:10,414 INFO [stomp_component] Connect to Stomp...
2018-04-10 12:05:10,415 INFO [client] Connecting to 9.108.163.130:65001
...
2018-04-10 12:05:10,437 INFO [client] Connection established
2018-04-10 12:05:10,537 INFO [client] Connected to stomp broker
[session=ID:resilient.localdomain-40775-1523276401752-5:3, version=1.2]
2018-04-10 12:05:10,538 INFO [stomp_component] Connected to
failover:(ssl://9.108.163.130:65001)?maxReconnectAttempts=1,startupMaxReco
nnectAttempts=1
2018-04-10 12:05:10,538 INFO [stomp_component] Client HB: 0  Server HB:
15000
2018-04-10 12:05:10,538 INFO [stomp_component] No Client heartbeats will
be sent
```

```
2018-04-10 12:05:10,538 INFO [stomp_component] Requested heartbeats from
server.
2018-04-10 12:05:10,539 INFO [actions_component] STOMP connected.
2018-04-10 12:05:10,641 INFO [actions_component] Subscribe to message
destination 'mcafee_epo_message_destination'
2018-04-10 12:05:10,642 INFO [stomp_component] Subscribe to message
destination actions.<orgID>.mcafee_epo_message_destination
```

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:

```
[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:

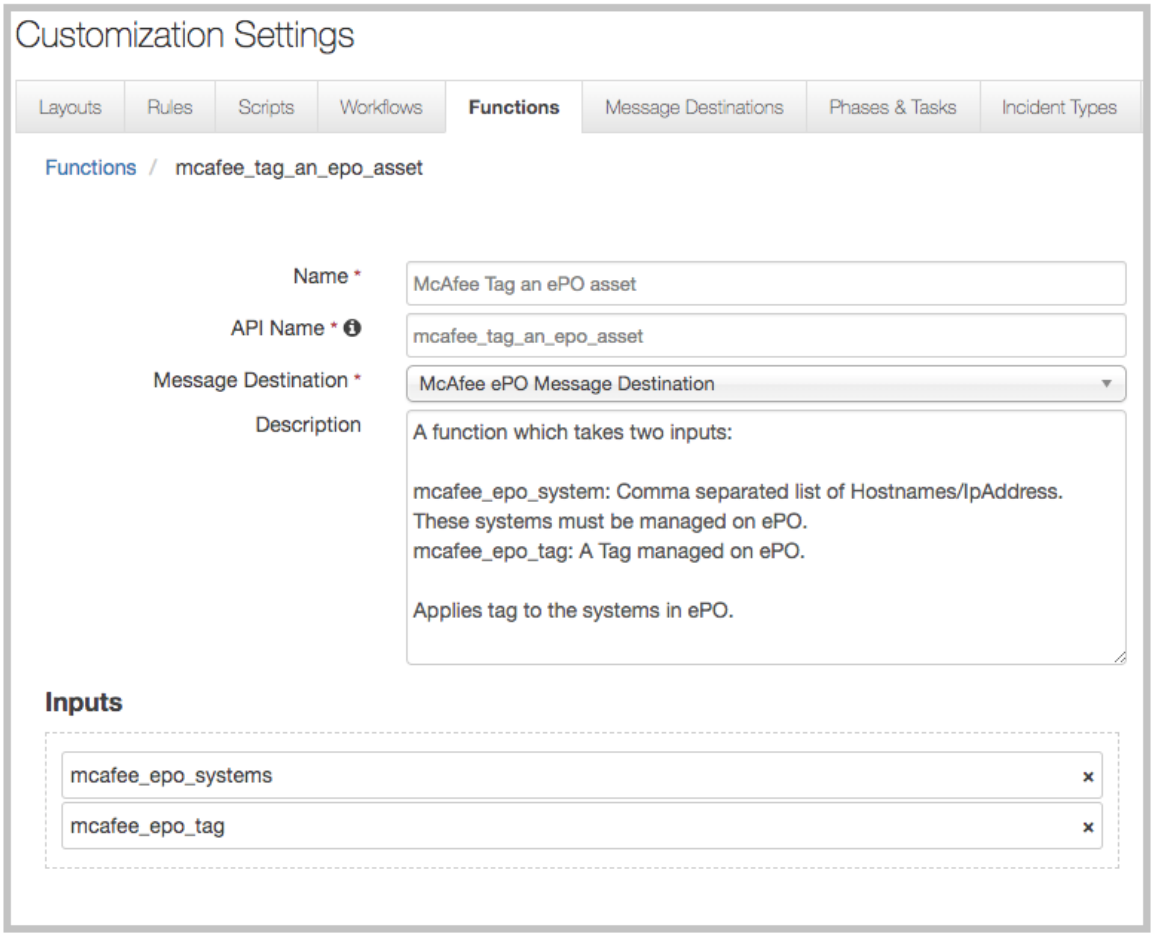
```
sudo systemctl resilient_circuits [start|stop|restart|status]
```

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 Description

Once the function package deploys the function, you can view it in the Resilient platform Functions tab. You can see the function details by clicking its name, as shown in the following screenshot.



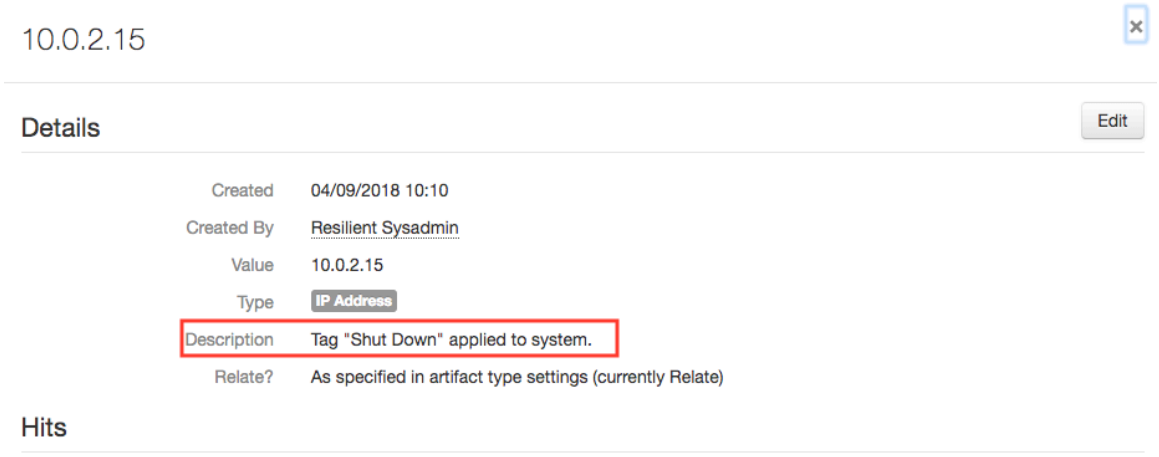
This function applies a tag to a system in ePO. Given a system or a list of the system's hostname's/IP addresses along with a tag, the function applies the tag to the provided systems in ePO. The function assumes that the list of systems and the provide tag are present and managed within ePO.

The function package also includes an example workflow and rule that show how the functions can be used. You can copy and modify these workflows and rules for your own needs.

The function response returns the function inputs, and a list of systems along with the tag similar to the following.

```
{
  "Systems": "10.0.2.15",
  "Tag": "Shut Down"
}
```

The out of the box example is triggered from an artifact of type IP Address or System Name and will set a description of the artifact to **Tag "<Tag used>" applied to system.**



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.