IBM Resilient

Incident Response Platform Integrations
Calendar Invite Function V1.0.0
Release Date: September 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 Calendar Invite Function.

Overview

fn_calendar_invite function takes the input incident id and retrieves the members and the owner of the incident and their email addresses. The function creates an email and calendar ICS file and sends an email to all incident members using an SMTP server inviting them to a meeting to discuss the incident.

Included in the package is an example workflow that uses the fn_calendar_invite function and an example rule for creating the fn_calendar_invite workflow menu item. Both the example workflow and rule are each called "Example: Calendar Invite".

Installation

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

- Resilient platform is version 31 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."x", where "x" is 10 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 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
```

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_calendar_invite] section, edit the settings as follows:

```
[fn_calendar_invite]

# Setup the email information for the sender of the calendar_invite email

email_username=jimmy@example.com

email_password=l33t

email_nickname=Resilient Meeting Organizer

email_host=mail.example.com

email_port=25
```

Deploy customizations to the Resilient platform

This Resilient Function package provides a function fn_calendar_invite, an example workflow that invokes the fn_calendar_invite function, a message queue and rules for creating the the fn_calendar_invite menu item.

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

```
resilient-circuits customize
```

1. 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 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.

Configure Resilient Circuits for restart

For normal operation, Resilient Circuits must run <u>continuously</u>. 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.

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.
lock
```

```
[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 function, 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.
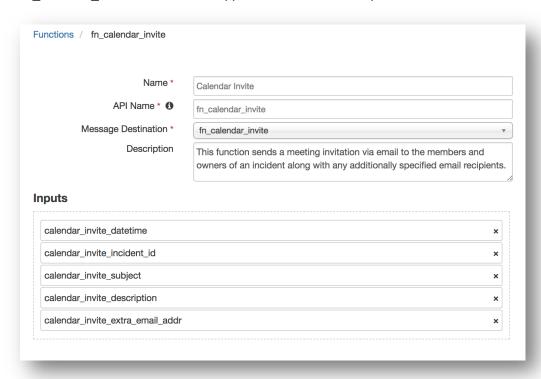
fn_calendar_invite :

This Resilient Function package provides a function fn_calendar_invite that takes as input:

- an incident id

- meeting date and time information

- email subject text

- meeting description

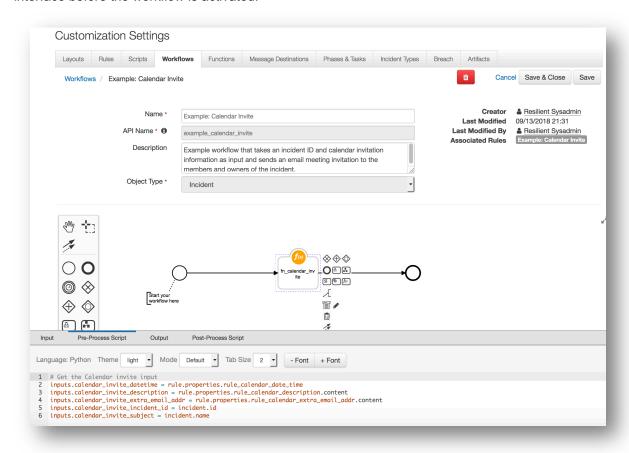- an additional list of email addresses to send meeting invitation

The fn_calendar_invite function takes the input incident id and retrieves the members and the owner of the incident and their email addresses. The function creates an email and calendar ICS file and sends email to all incident members using an SMTP server inviting them to a meeting to discuss the incident.

The fn_calendar_invite Functions tab appears in Resilient as depicted below:
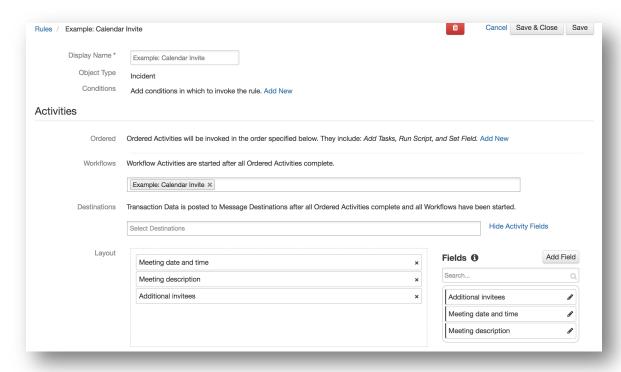
Example: Calendar Invite Workflow:

The incident id, the meeting date/time and description and the email subject are passed to the workflow in the pre-processor script. Note how the meeting date/time, description and additional email addresses are retrieved from the rule activity fields "rule.properties.", which is a new feature starting in Resilient V31. The Resilient user will manually enter this information through the user interface before the workflow is activated.
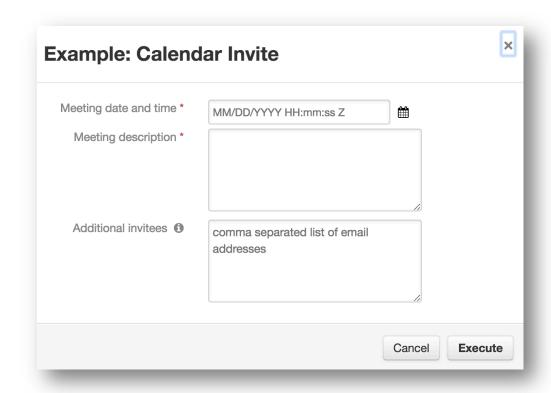
Example: Calendar Invite Rule:

Below is a screenshot of the Rules tab for the Example: Calendar Invite rule. The Example: Calendar Invite rule will display the Example: Calendar Invite menu item is Incident Actions menu.

The Example: Calendar Invite workflow is activated when the Example: Calendar Invite menu item is selected from the Actions menu on the incident tab. The following pop-up menu allows the user to input the information that is passed to the workflow. If the workflow completes successfully, the calendar invite email is sent to all members and the owner of the incident and to any email addresses entered by the user in the rules activity pop-up menu.

Example: Calendar Invite   ✕

| Meeting date and time * | MM/DD/YYYY HH:mm:ss Z | 📅 |

Meeting description *

Additional invitees ⓘ   comma separated list of email
                        addresses

                                                    Cancel   **Execute**

Resilient Platform Configuration

The Resilient fn_calendar_invite package uses an SMTP email server to send the calendar invite email.

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.