IBM Resilient

Incident Response Platform Integrations

PagerDuty Function V1.0.0

Release Date: April 2018

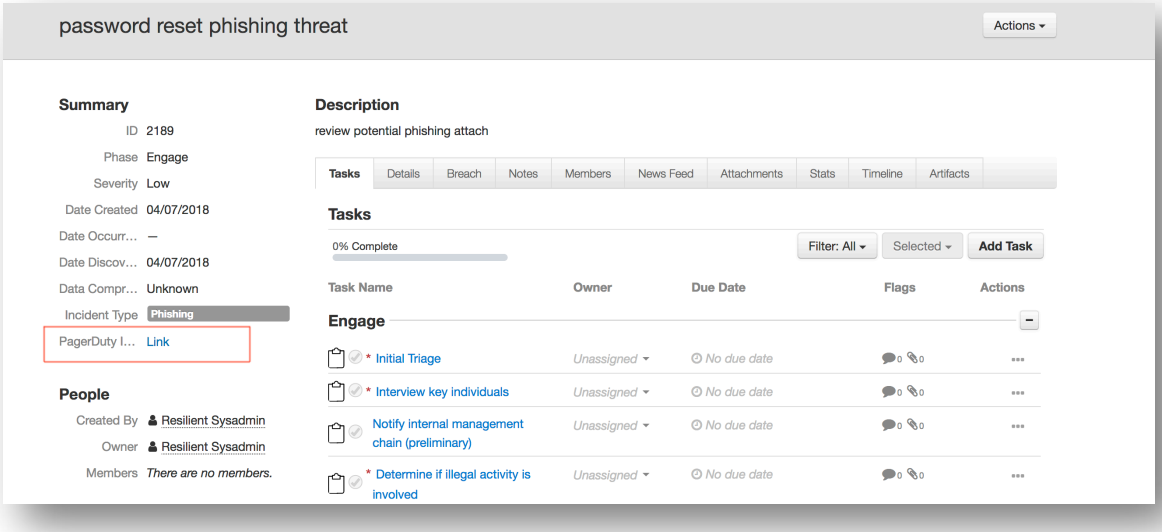
Resilient Functions simplify development of the integrations by sending data from the Resilient platform to a remote program that performs an activity then returns the results to the function. The results can be acted upon by a script and the result of that becomes a decision point in the Resilient workflow.

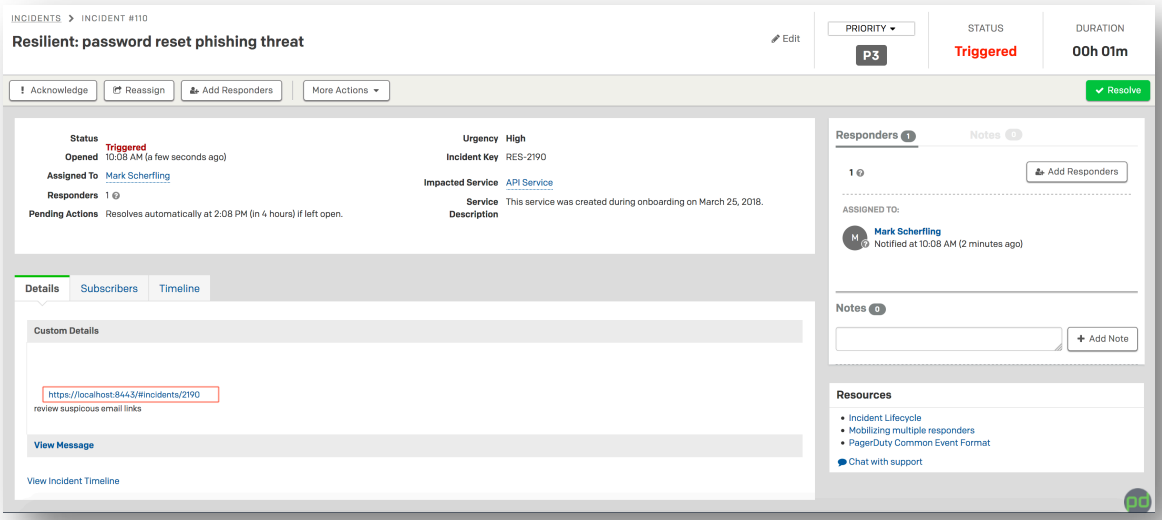
This guide describes the PagerDuty Function.

Overview

The PagerDuty integration with the Resilient platform allows for the tracking of Incidents as PagerDuty Incidents. Bidirectional links are saved to allow for easy navigation between the applications.

This integration allows for the creation of PagerDuty incidents, adding notes, and transitioning incidents when the corresponding Resilient incident is closed. The following screenshots show how this would appear in an incident.





Setup

The following lists the system requirements:

- Python version 2.7.10 or later, or version 3.6 or later
- Resilient Circuits and Resilient Python libraries version 30.0 or later
- Resilient platform version 30.0 or later

Perform the following to install and configure the function:

1. Ensure the environment is up to date:

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

2. Install the required software for the function (if not already installed):

   ```
   sudo pip install fn_pagerduty-<version>.tar.gz
   ```

3. Add the function to the Resilient platform:

   ```
   resilient-circuits customize
   ```

   You are prompted to answer prompts to import functions, message destinations, etc.

4. From the account used for Integrations, use one of the following commands to configure the PagerDuty settings. Use –c for new environments or –u for existing environments:

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

   OR

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

5. Edit the .resilient/app.config file and section [pagerduty]:

   ```
   api_token=<api_token from PagerDuty Service setup>
   verifyFlag=[True|False]
   ```

   Use False for self-signed SSL certificates.

After completing the configuration steps, enter the `resilient-circuits run` command. The following is an example of the resulting messages indicating the successful connection to the Resilient platform and the loading of the PagerDuty integration modules:

```
$ resilient-circuits run

2018-04-07 12:38:04,164 INFO [app] Configuration file:
/Users/Integration/.resilient/app.config

2018-04-07 12:38:04,165 INFO [app] Resilient server: <host>

2018-04-07 12:38:04,165 INFO [app] Resilient user: <acct>

2018-04-07 12:38:04,165 INFO [app] Resilient org: Co3 Systems

2018-04-07 12:38:04,165 INFO [app] Logging Level: INFO

…

2018-04-08 10:03:35,720 INFO [component_loader]
'fn_pagerduty.components.pd_create_note.FunctionComponent' loading

2018-04-08 10:03:35,721 INFO [component_loader]
'fn_pagerduty.components.pd_create_incident.FunctionComponent' loading

2018-04-08 10:03:35,723 INFO [component_loader]
'fn_pagerduty.components.pd_transition_incident.FunctionComponent' loading

…

2018-04-08 10:03:35,739 INFO [actions_component]
'fn_pagerduty.components.pd_create_note.FunctionComponent' function
'pagerduty_create_note' registered to 'pagerduty'

2018-04-08 10:03:35,740 INFO [actions_component]
'fn_pagerduty.components.pd_create_incident.FunctionComponent' function
'pagerduty_create_incident' registered to 'pagerduty'

2018-04-08 10:03:35,900 INFO [actions_component]
'fn_pagerduty.components.pd_transition_incident.FunctionComponent' function
'pagerduty_transition_incident' registered to 'pagerduty'

…

2018-04-08 10:03:36,016 INFO [actions_component] Subscribe to message destination
'pagerduty'

2018-04-08 10:03:36,019 INFO [stomp_component] Subscribe to message destination
actions.201.pagerduty

…
```

Resilient Platform Configuration

In the Customization Settings section of the Resilient platform, you can verify that the following PagerDuty specific functions, workflows and rules are available in the Resilient platform by clicking their respective tabs.

- Functions
  - **PagerDuty Create Incident**. Creates a PagerDuty incident including an incident's title, description and severity level. A custom incident field provides a link back to PagerDuty. See the corresponding workflow which defines which issue service and escalation policy to use.
  - **PagerDuty Create Note**. Creates a PagerDuty note based on adding a corresponding note to a Resilient incident. The note's description field is referenced.
  - **PagerDuty Transition Incident**. Transitions a PagerDuty incident. See the corresponding workflow which defines the PagerDuty status to use.

- Workflows. Some modifications are needed for your PagerDuty environment as indicated below.
  - **PagerDuty Open Incident**. Edit the PagerDuty Open Incident function for the specific pd_service and pd_escalation_policy fields for your PagerDuty use.
  - **PagerDuty Create Note**.
  - **PagerDuty Transition Incident**. Review the PagerDuty Transition Incident function for the specific status used in pd_status required in your workflow. The post-processing script by default uses 'resolved'.
- Rules. Operate on an Incident or an Incident's notes. If you wish to change rules between automatic and manual menu items, new rules referencing the same workflows need to be created.
  - **Trigger PagerDuty Incident**. Manual action.
  - **Create PagerDuty Note**. Automatic action if a PagerDuty incident is linked.
  - **Update PagerDuty Incident**. Can be used to acknowledge PagerDuty incidents or change the priority.
  - **Resolve PagerDuty Incident**. Automatic action if a PagerDuty incident is linked.

Operation

The default operation is to create a PagerDuty incident from a menu action manually. This can be changed to an automatic rule with additional custom conditions. All other rules trigger automatically once a PagerDuty incident is linked.

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.