Incident Response Platform Integrations

Risk Fabric Function Version 1.0.0

Release Date: December 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 whicj then becomes a decision point in the Resilient workflow.

Overview

The Risk Fabric integration with the Resilient platform allows for the querying of risk ratings for artifacts such as IP addresses, computer endpoints, and users. Risk models, event scenarios, and action plans can be pulled into Resilient and created as incidents, and then fully mitigated or classified.

Action Plans and Risk Models on Open Incidents Page

Manually perform classifications and mitigation actions on risk models, event scenarios, and action plans using rules, or automatically using advanced workflows.

Rule-based Classifications and Mitigation Actions

Advanced Workflows for Classifications and Mitigation Actions

Setup

The following lists the system requirements for using Resilient with Risk Fabric:

* 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
* Risk Fabric version 6.5.1 or later

Perform the following procedure 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

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

sudo pip install fn\_risk\_fabric-<version>.tar.gz

1. Add the function to the Resilient platform:

resilient-circuits customize

You are prompted to answer prompts to import functions, message destinations, and so on.

1. From the account used for Integrations, use the following command to configure the Risk Fabric settings.

resilient-circuits config env\_option

In the preceding command, env\_option is the environment option. Use –c for new environments or –u for existing environments.

1. Edit the .resilient/app.config file and section [fn\_risk\_fabric] as follows:

server=risk\_fabric\_URL

username=risk\_fabric\_api\_user

password=risk\_fabric\_api\_password

In the preceding commands, use the Risk Fabric URL, API user name, and API user password.

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 Risk Fabric 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: <org>

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

…

2018-04-07 12:38:05,418 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_host\_risk.FunctionComponent' loading

2018-04-07 12:38:05,419 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_ip\_risk.FunctionComponent' loading

2018-04-07 12:38:05,420 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_user\_risk.FunctionComponent' loading

2018-04-07 12:38:05,421 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_risk\_model\_instances.FunctionComponent' loading

2018-04-07 12:38:05,422 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_risk\_model\_instance\_details.FunctionComponent' loading

2018-04-07 12:38:05,423 INFO [component\_loader] 'fn\_risk\_fabric.components.get\_action\_plans.FunctionComponent' loading

2018-04-07 12:38:05,424 INFO [component\_loader] 'fn\_risk\_fabric.components.set\_event\_classifications.FunctionComponent' loading

2018-04-07 12:38:05,425 INFO [component\_loader] 'fn\_risk\_fabric.components.set\_event\_mitigations.FunctionComponent' loading

…

2018-04-07 12:38:05,435 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_host\_risk.FunctionComponent' function 'get\_host\_risk ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,436 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_ip\_risk.FunctionComponent' function 'get\_ip\_risk ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,437 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_user\_risk.FunctionComponent' function 'get\_user\_risk ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,438 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_risk\_model\_instances.FunctionComponent' function 'get\_risk\_model\_instances ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,439 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_risk\_model\_instance\_details.FunctionComponent' function 'get\_risk\_model\_instance\_details ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,440 INFO [actions\_component] 'fn\_risk\_fabric.components.get\_action\_plans.FunctionComponent' function 'get\_action\_plans ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,441 INFO [actions\_component] 'fn\_risk\_fabric.components.set\_event\_classifications.FunctionComponent' function 'set\_event\_classifications ' registered to 'risk\_fabric\_integration\_functions'

2018-04-07 12:38:05,442 INFO [actions\_component] 'fn\_risk\_fabric.components.set\_event\_mitigations.FunctionComponent' function 'set\_event\_mitigations ' registered to 'risk\_fabric\_integration\_functions'

…

2018-04-07 12:38:05,729 INFO [actions\_component] Subscribe to message destination 'risk\_fabric\_integration\_functions'

…

2018-04-07 12:38:05,731 INFO [stomp\_component] Subscribe to message destination actions.<org id>.risk\_fabric\_integration\_functions

…

Resilient Platform Configuration

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

Message Destination

* Risk Fabric Integration Functions – Default Message Destination for the Risk Fabric Integration Functions

Integration Functions

| Function | Description | Inputs | Outputs |
| --- | --- | --- | --- |
| RF Get Host Risk | Query the Risk Rating Information for a hostname. | rf\_hostname: Hostname for a computer endpoint | Risk Score for a computer endpoint |
| RF Get IP Risk | Query the Risk Rating information for an IP address. | rf\_ipaddress: IP Address such as 123.123.123.123 | Risk Score for an IP Address |
| RF Get User Risk | Query the Risk Rating information for a username | rf\_username: Username for a user account. | Risk Score for a user |
| RF Get Action Plans | Query the set of action plans for an account | None | List of Action Plans, including the rf\_actionplanguid for performing other actions like adding comments or updating event classifications and mitigations |
| RF Get Risk Model Instances | Query the set of Risk Model Instances | rf\_limit: For limited how many risk model instances to pull | List of Risk Model Instances, including the rf\_riskmodelinstanceid for performing other actions like classifications and mitigations. |
| RF Get Risk Model Instance | Get the set of Event Scenarios for a Risk Model Instance | rf\_riskmodelinstanceid: ID for the Risk Model Instance being requested | Additional details for a Risk Fabric instance, including Event Scenarios and Entity Collections with their rf\_cardinstanceid and rf\_focusentityid for performing other actions such as classifications and mitigations. |
| RF Set Classifications | Update Event Classifications | * rf\_riskmodelinstanceid: ID for the Risk Model Instance being classified. * rf\_cardinstanceid: ID for the Card Instance being classified. * rf\_focusentityid: ID for the Focus Entity being classified. * rf\_actionplanguid: ID for the action plan being classified. | None |
| RF Set Mitigations | Update Mitigation statues | * rf\_riskmodelinstanceid: ID for the Risk Model Instance being classified. * rf\_cardinstanceid: ID for the Card Instance being classified. * rf\_focusentityid: ID for the Focus Entity being classified. * rf\_actionplanguid: ID for the action plan being classified. | None |

Example Workflows

* RF Example: Get IP Risk  
  Example workflow for getting an IP address risk score. Workflow expects an IP address artifact, and updates the artifact description based on the artifact value with a risk score. Used by the example rule with the same name to automatically assign risk scores to IP address artifacts at creation.
* RF Example: Get Host Risk  
  Example workflow for getting a host risk score. Workflow expects a system name artifact, and updates the artifact description based on the artifact value with a risk score.
* RF Example: Get User Risk  
  Example workflow for getting a user risk score. Workflow expects a user account artifact, and updates the artifact description based on the artifact value with a risk score.
* RF Example: Mitigate Persistent Insider Threats  
  Example workflow for classifying and mitigating persistent insider threats. Add other integration functions such as disabling users in LDAP and notifying managers to create a fully-automated mitigation process.

Example Rules

* RF Example: Get IP Risk  
  Example rule for automatically updating an IP address artifact description field with the risk score associated with IP address. This rule calls the Get IP Risk Workflow which uses the RF Get IP Risk Integration Function.

Example Scripts

* create\_incidents\_action\_plans.py  
  Example script to create Incidents from Risk Fabric action plans. Requires creating and configuring an Incident Type, such as Action Plan.
* create\_incidents\_risk\_models.py  
  Example script to create Incidents from Risk Fabric risk models. Requires creating and configuring an Incident Type, such as Risk Model.

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 status 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 log file 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 creates progress information. Failures appear as errors and may contain Python trace statements.

Support

For additional support, contact [support@baydynamics.com](mailto:support@baydynamics.com).

Include relevant information from the log files to help us resolve your issue.