User Guide: fn_symantec_dlp_v1.0.0

Table of Contents

- **User Guide:** fn_symantec_dlp_v1.0.0
  - Table of Contents
  - Key Features
  - Poller - Symantec DLP incident Poller
    - Configuration
      - Symantec DLP Report Configuration
  - Function - Symantec DLP: Update incident
  - Custom Fields

Key Features

- Import Symantec DLP incidents using a Configurable Poller Component
- Update a Symantec DLP incident via a Resilient Function

Poller - Symantec DLP Incident Poller

A configurable circuits component which is used to automatically Poll a Symantec DLP instance using a defined interval.

Interfacing with the Symantec DLP incident and Reporting API, the Poller is able to receive all incidents in a saved report and then import each one into the Resilient platform. As a part of the import, certain artifacts are parsed from the incident object and saved as the appropiate Resilient artifact.

Configuration

In order to configure the Poller to run, an app.config value named `sdlp_should_poller_run` needs to be set to `True`.

See the README located at the root of the project for information on the app.config values.

> Note: When the `sdlp_should_search_res` parameter is set to True, the imported incidents from DLP are filtered twice. Once on the resultant list of incidents filtering out those with a non-null `resilient_incident_id` custom attribute. The second filtering occurs if `sdlp_should_search_res` is set to True. In this case, for each incident, a Resilient search API call is performed to ensure no Resilient platform incident exists with the given DLP incident ID set. The search is performed on the `sdlp_incident_id` Resilient Custom Field. This extra filter comes with a performance cost and should be disabled unless needed.

A note on the SOAP client and airgapped envionments.

When setting up the client which will make the SOAP requests to DLP, requests are made to gather the defined namespaces in the WSDL file provided. One of the needed namespaces is www.xmlmime.com/2005/ which

requires a outside network request. In scenarios where you have Resilient Circuits in an airgapped environment, this can be avoided by downloading the xml file located at xmlmime and then saving it in the data directory of the integration. If this is done, the local xml file will be cached to avoid a network request.

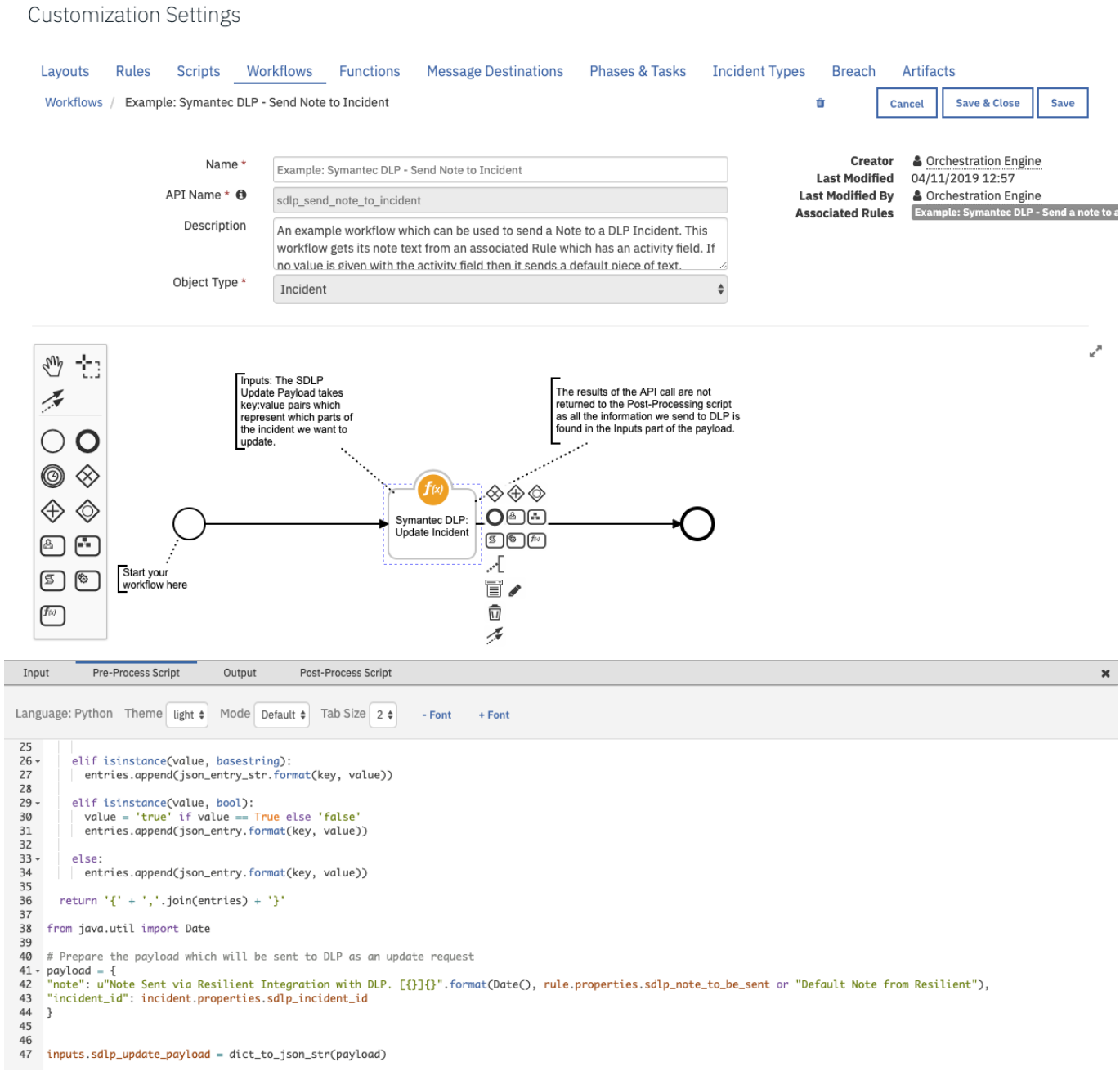
Symantec DLP Report Configuration

The Symantec DLP incident and Reporting SOAP API requires that the polled incidents be a part of a DLP Saved Report and this Saved Report ID is specified in the app.config.

A Saved Report is a way to cluster a number of incidents which match one or more filters. This can be used to pre-configure what sort of incidents should be brought to the Resilient platform. The DLP Filtering tool allows you to prepare a saved report which can simply be for all DLP incidents with a non-null `resilient_incident_id` custom attribute. This will result in an ever shrinking list as the incidents are brought into the Resilient platform. For users targeting a near real-time experience, the above Filtering in addition to a lower polling interval would be recommended. For an example of how to setup a Saved Report, see the README at the root of this project for a step by step instruction.

Function - Symantec DLP: Update Incident

A function which is used to update the details of a Symantec DLP incident. It takes one input which is a dictionary of DLP incident attributes to be changed. To enable updates for multiple custom attributes, provide a list or dictionary of all the attributes to be changed in the format: <attribute_name>: <new_value>



▶ Inputs:

Name	Type	Required	Example	Tooltip
sdlp_update_payload	textarea	Yes	-	A JSON-like object which contains values to be updated on a given Symantec DLP incident

▶ Outputs:

```
results = {
    # TODO: Copy and paste an example of the Function Output within this
code block.
    # To see view the output of a Function, run resilient-circuits in
DEBUG mode and invoke the Function.
    # The Function results will be printed in the logs: "resilient-
```

```
circuits run --loglevel=DEBUG"
}
```

▶ Example Pre-Process Script:

```
########################################
### Define pre-processing functions ###
########################################
def dict_to_json_str(d):
  """Function that converts a dictionary into a JSON stringself.
     Supports basestring, bool and int.
     If the value is None, it sets it to False"""

  json_str = '"{ {0} }"'
  json_entry = '"{0}":{1}'
  json_entry_str = '"{0}":"{1}"'
  json_entry_unicode = u'"{0}":"{1}"'
  entries = []

  for entry in d:
    key = entry
    value = d[entry]


    if value is None:
      value = False

    if isinstance(value, unicode):
      entries.append(json_entry_unicode.format(key, value))

    elif isinstance(value, basestring):
      entries.append(json_entry_str.format(key, value))

    elif isinstance(value, bool):
      value = 'true' if value == True else 'false'
      entries.append(json_entry.format(key, value))

    else:
      entries.append(json_entry.format(key, value))

  return '{' + ','.join(entries) + '}'

from java.util import Date

# Prepare the payload which will be sent to DLP as an update request
payload = {
"note": u"Note Sent via Resilient Integration with DLP. [{}]
{}".format(Date(), rule.properties.sdlp_note_to_be_sent or "Default Note
from Resilient"),
"incident_id": incident.properties.sdlp_incident_id
}
```

```
inputs.sdlp_update_payload = dict_to_json_str(payload)
```

▶ Example Post-Process Script:

```
None
```

Custom Fields

Label	API Access Name	Type	Prefix	Placeholder	Tooltip
Symantec DLP Incident ID	sdlp_incident_id	number	properties	-	ID of a Symantec DLP incident
Symantec DLP Incident URL	sdlp_incident_url	textarea	properties	-	Hyperlink to the Symantec DLP incident

Rules

Rule Name	Object	Workflow Triggered
Example: Symantec DLP - Send a note to a DLP incident	incident	sdlp_send_note_to_incident
Example: Symantec DLP - Update DLP when this incident is closed	incident	sdlp_set_incident_status