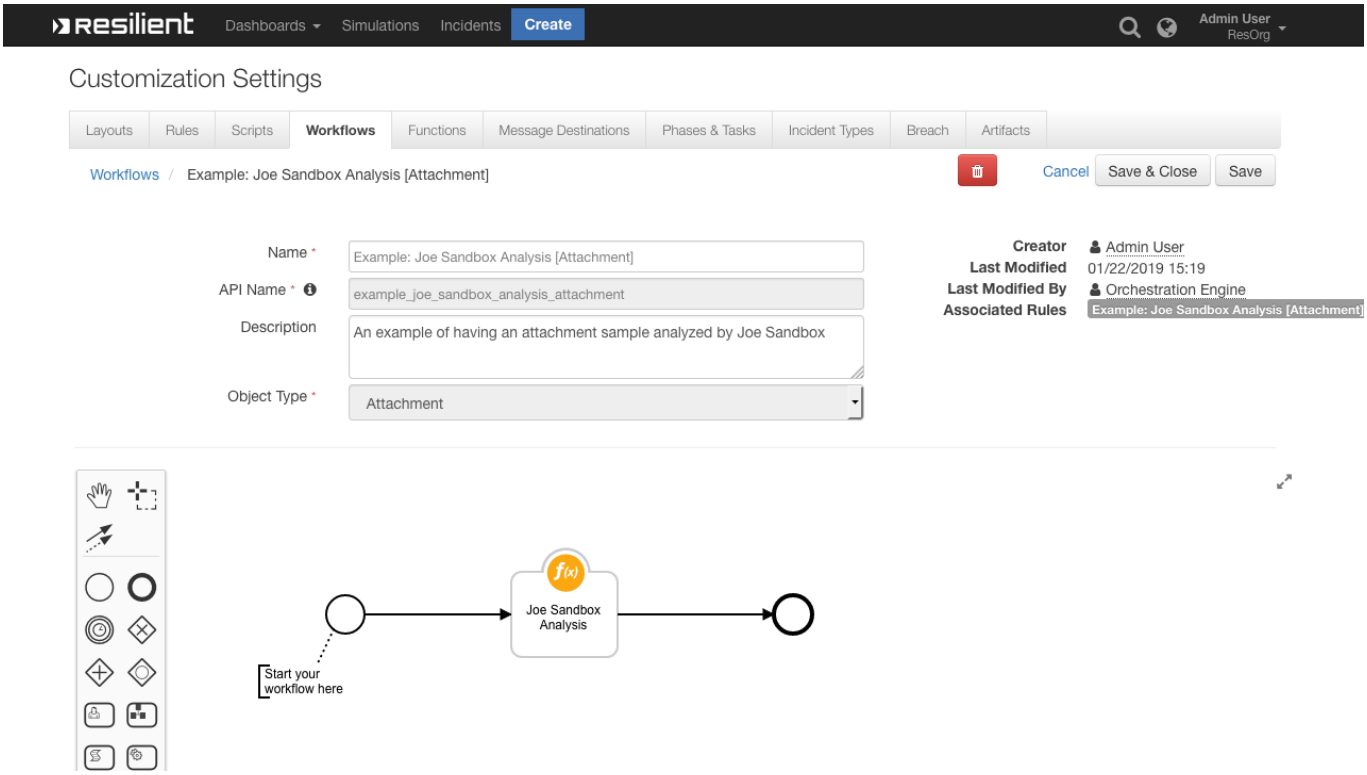
Joe Sandbox Analysis Function for IBM Resilient

Table of Contents

This package contains a function that executes a Joe Sandbox Analysis of an Attachment or Artifact and returns the Analysis Report to IBM Resilient.



- Supports an attachment or artifact that is a file, or where the artifact's value contains a URL.
- Allows users to select the type of report, PDF, HTML, or JSON, which is returned from Joe Sandbox.
- Supports a proxy. Just add your proxy details to the `app.config` file.
- The function depends on **Joe Security's python module, jbxapi**.See here for more details

---

app.config settings:

```
[fn_joe_sandbox_analysis]
# Accept Terms & Conditions
jsb_accept_tac=True

# Your JoeSandbox API Key
```

```
jsb_api_key=

# The analysis URL
jsb_analysis_url=https://jbxcloud.joesecurity.org/analysis

# Amount of time in seconds to wait until checking if the report is ready
again
jsb_analysis_report_ping_delay=120

# This is the max time in seconds the function will wait for the report to
be generated
jsb_analysis_report_request_timeout=1800

# Set if you need to use a proxy to access JSB
#jsb_http_proxy=http://user:pass@10.10.1.10:3128
#jsb_https_proxy=http://user:pass@10.10.1.10:1080
```

Function Inputs:

Function Name	Type	Required	Example	Info
`incident_id`	`Number`	Yes	`1001`	The ID of the current Incident
`attachment_id`	`Number`	No	`5`	The ID of the Attachment to be analyzed
`artifact_id`	`Number`	No	`6`	The ID of the Artifact to be analyzed
`jsb_report_type`	`Select`	Yes	`"json"`	The type of report to be returned from Joe Sandbox. Options are: `html`, `pdf`, or `json`

Function Output:

```
results = {
    "analysis_report_name": "My Malicious Scan Report",
    "analysis_report_id": 123,
    "analysis_report_url":
"https://jbxcloud.joesecurity.org/analysis/123/456",
    "analysis_status": "clean"
}
```

Pre-Process Script:

This example just sets the function inputs.

```
inputs.incident_id = incident.id
inputs.artifact_id = artifact.id
```

Post-Process Script:

This example adds a Note to the Incident and color codes the `analysis_status` depending if it was **malicious** or **clean**

```
color = "#45bc27"

if (results.analysis_status != "clean"):
  color = "#ff402b"

noteText = """<br>Joe Sandbox analysis <b>{0}</b> complete
              <b>Artifact:</b> '{1}'
              <b>Report URL:</b> <a href='{2}'>{2}</a>
              <b>Detection Status:</b> <b style="color: {3}">{4}
</b>""".format(results.analysis_report_name, artifact.value,
results.analysis_report_url, color, results.analysis_status)

incident.addNote(helper.createRichText(noteText))
```

Rules

Rule Name	Object Type	Workflow Triggered
Example: Joe Sandbox Analysis [Artifact]	`Artifact`	`Example: Joe Sandbox Analysis [Artifact]`
Example: Joe Sandbox Analysis [Attachment]	`Attachment`	`Example: Joe Sandbox Analysis [Attachment]`