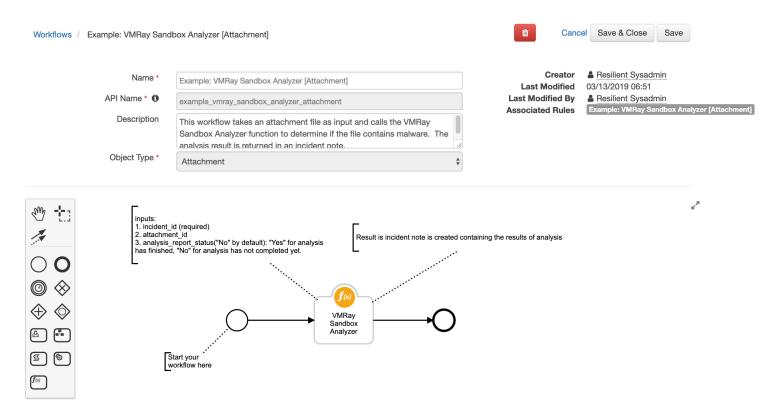
# VMRay Sandbox Analyzer Function for IBM Resilient

#### **Table of Contents**

- app.config settings
- Function Inputs
- Function Output
- Pre-Process Script
- Post-Process Script
- Rules

This package contains a function that executes a VMRay Malware Sandbox Analysis using VMRay Cloud APIs, also included are two example workflows and two example rules that demonstrate how to use this function.



• an attachment or artifact must be a file.

- The report only supports Type of JSON. HTML and PDF are not supported
- Supports a proxy. Just add your proxy details to the proxy section in app.config file.

### app.config settings:

```
# Your VMRay Analyzer API Key

vmray_api_key=

# Your VMRay Server URL, using https://cloud.vmray.com if empty.

vmray_analyzer_url=https://cloud.vmray.com

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

vmray_analyzer_report_request_timeout=60
```

# **Function Inputs:**

Function Name	Туре	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
analyzer_report_status	Boolean	Yes	No	Has the analysis report generated successfully. Options are: Yes or

## **Function Output:**

```
results = {
          "analysis_report_status": analysis_report_status,
          "incident_id": incident_id,
          "artifact_id": artifact_id,
          "attachment_id": attachment_id,
          "sample_final_result": sample_final_result
}
```

#### **Pre-Process Script:**

Example: VMRAY Sandbox Analyzer [Attachment]

```
inputs.incident_id = incident.id
inputs.attachment_id = attatchment.id
```

Example: VMRAY Sandbox Analyzer [Artifact]

```
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

```
def font_color(vti_score,sample_severity):
    color = "green"
    try:
        if sample_severity in ["malicious"] or int(vti_score) >= 75:
            color = "red"
        elif sample_severity in ["blacklisted","suspicious"] or int(vti_score) >= 50:
            color = "yellow"
    except:
        pass
    return color

if not results.analysis_report_status:
    noteText = u"""Successful submit <b>{}</b> to VMRay Cloud Analyzer.However it will t
    ake time to generate an analysis report, please submit it again later. <br/>""".format
```

```
(attachment.name)
else:
noteText = u"""Successful submit <b>{}</b> to VMRay Analyzer.Check the results below
: <br>""".format(attachment.name)
 for sample in results.sample_final_result:
  noteText += u"""-----
____""
  color = font color(sample["sample report"]["sample score"],sample["sample report"]
["sample last reputation severity"])
   noteText += u"""<br>VMRay Sandbox Analysis: <b>{sample_filename}</b> complete.<br>
                  VMRAY Online Attachment: <a href={sample online report}>{sample on
line report}</a><br>
                  VMRay Analyzer result: VTI Score: <b style= "color:{color}">{samp
""".format(sample_filename=sample["sample_report"]["sample_filename"],
                          sample_online_report=sample["sample_report"]["sample_webif
_url"],
                          color = color,
                          sample_vti_score = sample["sample_report"]["sample_score"]
                          sample severity = sample["sample report"]["sample last rep
utation_severity"])
  noteText += u"""<br> | analysis_id | analysis_job_started | analysis_vti_score | an
alysis severity | <br>"""
  for analysis in sample["sample_analysis_report"]:
    color = font_color(analysis["analysis_vti_score"],analysis["analysis_severity"])
    noteText += u""" | <a href={analysis link}> {analysis id} </a> | {analysis job s
tarted} | <b style= "color:{color}"> {analysis vti score}</b> | <b style= "color:{c
olor}">{analysis_severity}</b> |<br>
                """.format(analysis_link=analysis["analysis_webif_url"],
                          analysis_id=analysis["analysis_id"],
                          analysis job started=analysis["analysis job started"],
                          analysis_vti_score=analysis["analysis_vti_score"],
                          analysis_severity=analysis["analysis_severity"],
                          color=color)
  reputations = [str(reputation["reputation lookup severity"]) for reputation in sam
ple["sample_reputation_report"]]
   if "malicious" in reputations:
    color = "red"
```

```
reputation lookup severity = "malicious"
  elif "suspicious" in reputations:
    color = "yellow"
    reputation_lookup_severity = "suspicious"
  elif "blacklisted" in reputations:
    color = "yellow"
    reputation lookup severity = "blacklisted"
  elif "not_suspicious" in reputations:
    color = "green"
    reputation lookup severity = "not suspicious"
  elif "whitelisted" in reputations:
    color = "green"
    reputation lookup severity = "whitelisted"
   else:
    color = "green"
    reputation_lookup_severity = "unknown"
  noteText += u"""Reputation lookup result: <b style= "color:{color}">{reputation l
ookup severity} </b> <br/>""".format(color=color, reputation lookup severity=reputatio
n_lookup_severity)
incident.addNote(helper.createRichText(noteText))
```

#### Example of adding a incident note from post-processing scripts:

Resilient Sysadmin added a note to the Incident 01/29/2019 07:50

Successful submit 0655d58db2798ad8336f92dd580f988312f37f3e52b405c9c71d3afd2bd2c290 to VMRay Analyzer.Check the results below:

WMRay Sandbox Analysis: 0655d58db2798ad8336f92dd580f988312f37f3e52b405c9c71d3afd2bd2c290.rtf complete.

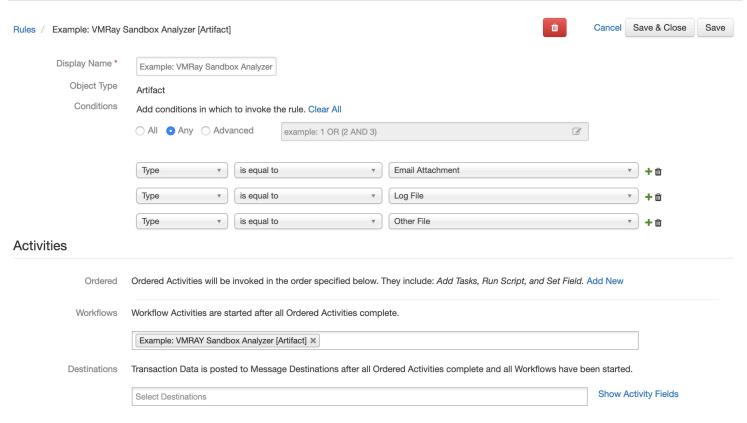
VMRay Online Attachment: https://cloud.vmray.com/user/sample/view?id=2996559

VMRay Analyzer result: VTI Score: 100, Severity: blacklisted

analysis\_id	analysis\_job\_started	analysis\_vti\_score	analysis\_severity
2668927	2019-01-30T15:41:48	100	malicious
2668886	2019-01-30T15:32:43	100	malicious
2668886	2019-01-30T15:28:45	100	malicious
2668884	2019-01-30T15:24:45	100	malicious
26688848	2019-01-30T15:24:45	100	malicious
26688848	2019-01-30T15:24:45	100	malicious
26688848	2019-01-30T15:24:45	100	malicious
26688848	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious
26688481	2019-01-30T15:24:45	100	malicious

#### Rules

Rule Name	Object Type	Workflow Triggered
Example: VMRay Sandbox Analysis [Artifact]	Artifact	Example: VMRay Sandbox Analyzer [Artifact]



Rule Name	Object Type	Workflow Triggered		
Example: VMRay Sandbox Analyzer [Attachment]	Attachment	Example: VMRay Sandbox Analyzer [Attachment]		

