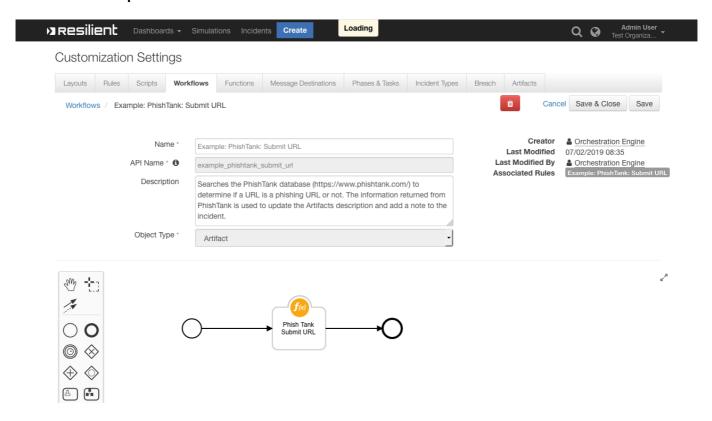
PhishTank Lookup Function for IBM Resilient

- Overview
- Note on Partial URL Support
- Requirements
- Installation
- Uninstall
- Troubleshooting
- Release Notes
- Support

Overview

PhishTank Lookup URL Function for IBM Resilient



Searches the PhishTank database (https://www.phishtank.com/) to determine if a URL is a phishing URL or not. The information returned from PhishTank is used to update the Artifacts description and add a note to the incident

Note on Partial URL Support

Partial URLs are not supported with this integration since the PhishTank API supports full URL matches only, although the PhishTank user interface supports partial matches.

For example:

URL to Check PhishTank UI Result PhishTank API Result

URL to Check	PhishTank UI Result	PhishTank API Result https://safrainternet- br.com/Atualizar.html is a Valid Phish	
https://safrainternet- br.com/Atualizar.html	https://safrainternet- br.com/Atualizar.html is a Valid Phish		
<pre>https://safrainternet- br.com/</pre>	<pre>https://safrainternet- br.com/Atualizar.html is a Valid Phish</pre>	<pre>https://safrainternet- br.com/ not found in Database</pre>	

Requirements

- IBM Resilient >= v31.0.4254
- An Integration Server running resilient_circuits>=30.0.0
 - To setup an Integration Server see: ibm.biz/res-int-server-guide

Installation

- Download the fn_phish_tank.zip
- Copy the zip to your Integration Server and SSH into it.
- Unzip the package:

```
$ unzip fn_phish_tank-x.x.x.zip
```

• Change Directory into the unzipped Directory:

```
$ cd fn_phish_tank-x.x.x
```

• Install the package:

```
$ pip install fn_phish_tank-x.x.x.tar.gz
```

• Import the **configurations** into your app.config file:

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

• Import the fn_phish_tank **customizations** into the Resilient platform:

```
$ resilient-circuits customize -y -l fn-phish-tank
```

• Open the config file, scroll to the bottom and edit your fn_phish_tank configurations:

\$ nano ~/.resilient/app.config

Config	Required	Example	Description
phish_tank_api_url	Yes	http://checkurl.phishtank.com/checkurl/	PhishTank API Access URL
phish_tank_api_key	Yes	-	PhishTank API Key
proxy	No	127.0.0.1	Proxy Server Address. Default is None

- Save and Close the app.config file.
- [Optional]: Run selftest to test the Integration you configured:

```
$ resilient-circuits selftest -l fn-phish-tank
```

• Run resilient-circuits or restart the Service on Windows/Linux:

```
$ resilient-circuits run
```

Uninstall

- SSH into your Integration Server
- Uninstall the package:

```
$ pip uninstall fn-phish-tank
```

- Open the config file, scroll to the [fn_phish_tank] section and remove the section or prefix # to comment out the section.
- Save and Close the app.config file.

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.

Release Notes

v1.0.0

• Initial Release

Support

Name	Version	Author	Support URL
fn_phish_tank	1.0.0	Resilient Labs	http://ibm.biz/resilientcommunity

User Guide: fn_phish_tank_v1.0.0

Table of Contents

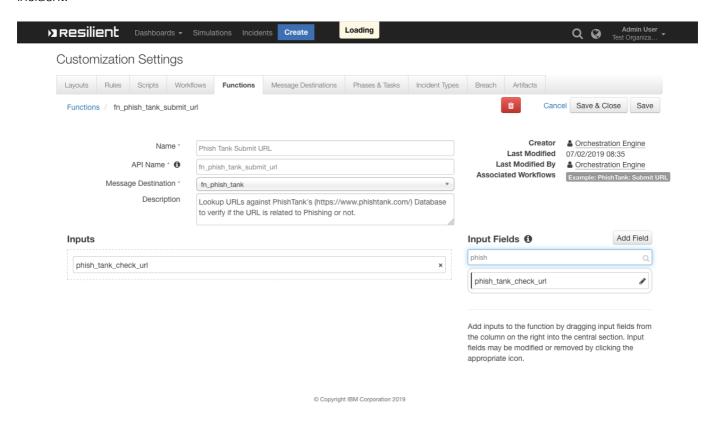
- Key Features
- Function Phish Tank Submit URL
- Rules

Key Features

- Verify if a URL is related to Phishing using PhishTank's Database
- Add a Note to the Incident and append to the Artifact's Description information detailing the results received from PhishTank

Function - Phish Tank Submit URL

Searches the PhishTank database (https://www.phishtank.com/) to determine if a URL is a phishing URL or not. The information returned from PhishTank is used to update the Artifacts description and add a note to the incident.



► Inputs:

Name	Type	Required	Example	Tooltip
phish_tank_check_url	text	Yes	http://www.example.com	URL to lookup in PhishTank's
				Database

► Outputs:

```
results = {
    'inputs': {
        u 'phish_tank_check_url': u 'http://barea-v02.ga/df/'
    },
    'metrics': {
        'package': 'fn-phish-tank',
        'timestamp': '2019-06-25 09:53:11',
        'package version': '1.0.0',
        'host': 'shanes-mbp.galway.ie.ibm.com',
        'version': '1.0',
        'execution time ms': 192
    },
    'success': True,
    'content': {
        u 'meta': {
            u 'status': u 'success',
            u 'timestamp': u '2019-06-25T08:53:11+00:00',
            u 'serverid': u 'df0ef05',
            'timestamp modified': 1561452791000,
            u 'requestid': u '172.31.97.117.5d11e0f7b69ce3.54218678'
        },
        u 'results': {
            u 'verified': True,
            'verified at modified': 1561452674000,
            u 'phish detail page': u
'http://www.phishtank.com/phish_detail.php?phish_id=6093832',
            u 'url': u 'http://barea-v02.ga/df/',
            u 'verified at': u '2019-06-25T08:51:14+00:00',
            u 'phish id': u '6093832',
            u 'valid': True,
            u 'in database': True
        }
    },
    'raw': '{"meta": {"status": "success", "timestamp": "2019-06-
25T08:53:11+00:00", "serverid": "df0ef05", "timestamp_modified":
1561452791000, "requestid": "172.31.97.117.5d11e0f7b69ce3.54218678"},
"results": {"verified": true, "verified_at_modified": 1561452674000,
"phish_detail_page": "http://www.phishtank.com/phish_detail.php?
phish_id=6093832", "url": "http://barea-v02.ga/df/", "verified_at": "2019-
06-25T08:51:14+00:00", "phish_id": "6093832", "valid": true,
"in database": true}}',
    'reason': None,
    'version': '1.0'
}
```

► Example Pre-Process Script:

```
# Get the url from the Artifact's Value
inputs.phish_tank_check_url = artifact.value
```

► Example Post-Process Script:

```
def append artifact description(the artifact, the text):
  """Appends the_text to the_artifact.description safely
  handling unicode"""
  new_description = u""
  if the artifact.description is None:
    current_description = None
  else:
    current description = the artifact.description.get("content", None)
  if current_description is not None:
    new_description = u"{0}<br>>---<br>
{1}".format(unicode(current_description), unicode(the_text))
  else:
    new_description = u"{0}".format(unicode(the_text))
  the_artifact.description = helper.createRichText(new_description)
if results.success:
  # Get the PhishTank Results
  phish tank results = results.content.get("results", {})
  url = phish_tank_results.get("url", u"")
  in_database = phish_tank_results.get("in_database", False)
  is_verified = phish_tank_results.get("verified", False)
  is_valid = phish_tank_results.get("valid", False)
  # Define the comment and msg to be appended to the Artifact's
Description
  comment = u""
  msg = u"""<b>PhishTank Lookup</b> has complete
            <br/><br><bURL:</b> {0}</b>
            <br><b>Found in Database:</b> {1}""".format(url,
unicode(in database))
  if not in_database:
    comment = u"Nothing known about this url"
  else:
    phish_id = phish_tank_results.get("phish_id")
    phish_detail_page_url = phish_tank_results.get("phish_detail_page")
    msg = u'''''\{0\}
```

Rules

Rule Name	Object	workflow Iriggered
Example: PhishTank: Submit URL	artifact	example_phishtank_submit_url