Spamhaus Lookup for IBM Resilient

Release Notes

v1.0.1

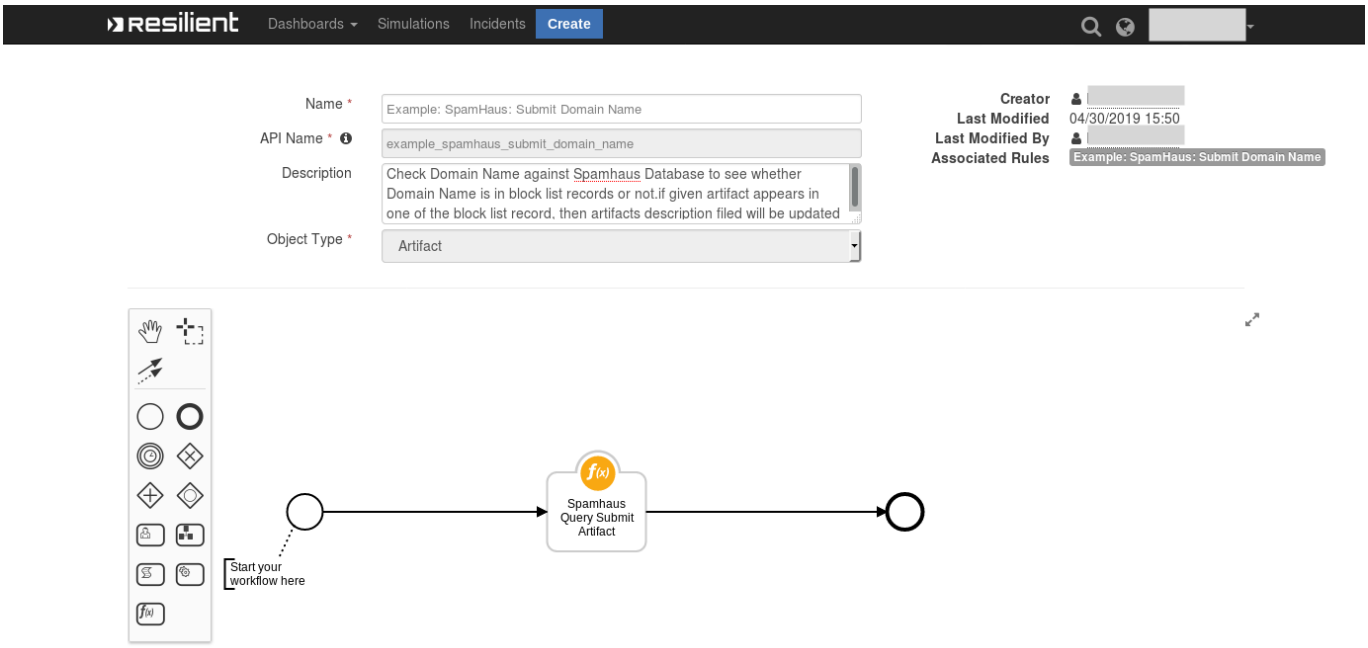
- Added support for App Host

v1.0.0

- Initial Release

Overview

Look up IP Addresses + Domain Name in Spamhaus Datasets



If a given artifact appears in one of Spamhaus Datasets, then the artifact's description is updated with additional enrichment information.

Requirements

- Resilient platform >= `v35.0.0`
- An Integration Server running `resilient_circuits>=33.0.0`
    - To set up an Integration Server see: ibm.biz/res-int-server-guide
    - If using API Keys, minimum required permissions are:
        - Org Data: Read, Edit
        - Function: Read

Installation (App Host)

With App Host, all the run-time components are pre-built. Perform the following steps to install and configure:

- Download the `app-fn_spamhaus_query-x.x.x.zip`.
- In Resilient navigate to **Adiminstrator Settings > Apps**
- Click the Install button and select the downloaded `app-fn_spamhaus_query-x.x.x.zip`. This will install the associated customizations.
- Once installed, navigate to the app's Configuration tab and edit the app.config file updating the `[resilient]` section as necessary and updating the `[fn_spamhaus_query]` section as necessary.

Installation (Integration Server)

- Download the `app-fn_spamhaus_query.zip`.
- Copy the `.zip` to your Integration Server and SSH into it.
- **Unzip** the package:

```
$ unzip app-fn_spamhaus_query-x.x.x.zip
```

- **Change Directory** into the unzipped directory:

```
$ cd fn_spamhaus_query-x.x.x
```

- **Install** the package:

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

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

```
$ resilient-circuits config -u -l fn-spamhaus-query
```

- Import the fn_spamhaus_query **customizations** into the Resilient platform:

```
$ resilient-circuits customize -y -l fn-spamhaus-query
```

- Open the config file, scroll to the bottom and edit your fn_spamhaus_query configurations:

```
$ nano ~/.resilient/app.config
```

Config	Required	Example	Description
spamhaus_wqs_url	Yes	`https://apibl.spamhaus.net/lookup/v1/`	The endpoint for Spamhaus API
spamhaus_dqs_key	Yes	``	The API Key
http_proxy	Yes	``	A HTTP proxy
https_proxy	Yes	``	A HTTPS Proxy

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

```
$ resilient-circuits selftest -l fn-spamhaus-query
```

- **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-spamhaus-query
```

- Open the config file, scroll to the [fn_spamhaus_query] 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.

Support

Name	Version	Author	Support URL
fn_spamhaus_query	1.0.1	Resilient Labs	https://ibm.biz/resilientcommunity

User Guide: fn_spamhaus_query_v1.0.1

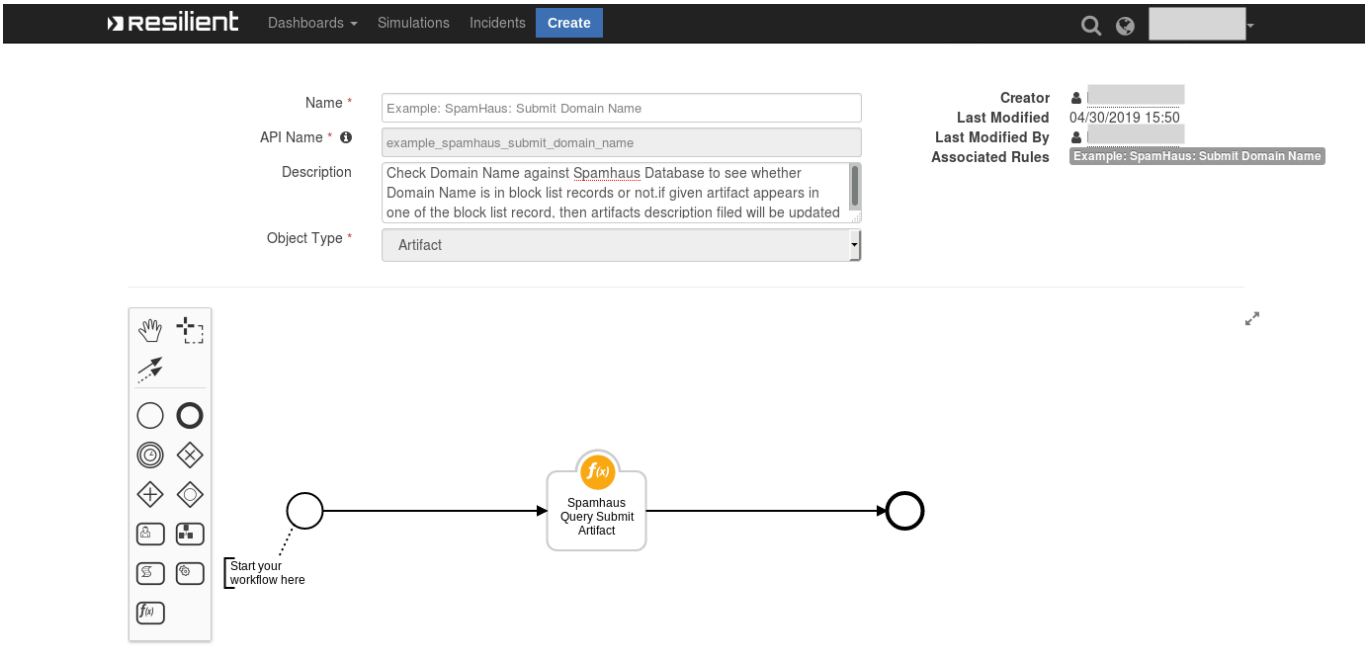
Table of Contents

Key Features

- Look up IP Addresses in Spamhaus Datasets
- Look up Domain Names in Spamhaus Datasets
- Add the enrichment data to the Resilient artifact

Function - Spamhaus Query Submit Artifact

Function to check IP Addresses & Domain Names against Spamhaus Dataset to see whether it appears in the Dataset or not



▶ Inputs:

Name	Type	Required	Example	Tooltip
`spamhaus_query_string`	`text`	Yes	`IP Address/Domain Name`	An IP Address or a Domain Name

Name	Type	Required	Example	Tooltip
spamhaus_search_resource	text	Yes	SBL,XBL,PBL,SBL-XBL,ZEN,MSR,AUTHBL,ZRD,DBL	The name of the Dataset to seach (e.g. SBL)

▶ Outputs:

```
results = {
    'inputs': {
        'spamhaus_search_resource': 'SBL',
        'spamhaus_query_string': '127.0.0.2'
    },
    'metrics': {
        'package': 'fn-spamhaus-query',
        'timestamp': '2020-08-20 15:27:44',
        'package_version': '1.0.1',
        'host': 'example',
        'version': '1.0',
        'execution_time_ms': 412
    },
    'success': True,
    'content': {
        'status': 200,
        'resp': [1002],
        'is_in_blocklist': True,
        1002: {
            'URL': 'https://www.spamhaus.org/sbl/',
            'explanation': 'IP addresses are listed on the SBL because
they appear to Spamhaus to be under the control of, used by, or made
available for use by spammers and abusers in unsolicited bulk email or
other types of Internet-based abuse that threatens networks or users.',
            'dataset': 'SBL'
        }
    },
    'raw': '{"status": 200, "resp": [1002], "is_in_blocklist": true,
"1002": {"URL": "https://www.spamhaus.org/sbl/", "explanation": "IP
addresses are listed on the SBL because they appear to Spamhaus to be
under the control of, used by, or made available for use by spammers and
abusers in unsolicited bulk email or other types of Internet-based abuse
that threatens networks or users.", "dataset": "SBL"}}',
    'reason': None,
    'version': '1.0'
}
```

▶ Workflows

▶ Example Pre-Process Script:

```
inputs.spamhaus_query_string = artifact.value
inputs.spamhaus_search_resource =
rule.properties.spamhaus_domain_name_resource
```

▶ Example Post-Process Script:

```
results_data = results.get('content')
tmp_text = ""
tmp_desc = artifact.description
if results_data.get('is_in_blocklist'):
   tmp_text = "<br><br><b>This artifact checked against Spamhaus and it is
in block list.</b>"
   resp_list = results_data.get('resp')
   for code in resp_list:
        code = str(code)
        tmp_text += "<br><b>code :</b> {}</br>".format(code)
        tmp_text += "<br><b>dataset :</b> {}
</br>".format(results_data.get(code).get('dataset'))
        tmp_text += "<br><b>explanation :</b> {}
</br>".format(results_data.get(code).get('explanation'))
        tmp_text += "<br><b>URL :</b> </br>{}
</br>".format(results_data.get(code).get('URL'))
else:
   tmp_text = "<br><br><b>This artifact checked against Spamhaus Dataset:
{} and it is not in block list.</b></br>
</br>".format(results.get('inputs').get('spamhaus_search_resource'))
if tmp_desc:
   tmp_desc = tmp_desc.get('content')
else:
   tmp_desc = ""
complete_tmp_text = tmp_desc+tmp_text
rich_text = helper.createRichText(complete_tmp_text)
artifact.description = rich_text
```

Rules

Rule Name	Object	Workflow Triggered
Example: SpamHaus: Submit IP Address	artifact	example_spamhaus_submit_ip_address
Example: SpamHaus: Submit Domain Name	artifact	example_spamhaus_submit_domain_name