

Resilient Integration with WHOIS

This package contains one function which provides enrichment information about a domains WHOIS information. Takes in an input of an IP address or URL and then queries for WHOIS info related to this input. Results are saved in a rich text note.

Table of Contents

- [app.config settings:](#)
- [Function Inputs:](#)
- [Function Output:](#)
- [Pre-Process Script:](#)
- [Post-Process Script:](#)
- [Rules](#)

The screenshot displays the Resilient platform's 'Customization Settings' page, specifically the 'Workflows' section. The workflow being configured is 'Example: Whois Query Against Artifact'. The configuration fields include:

- Name:** Example: Whois Query Against Artifact
- API Name:** example_whois_query_against_artifact
- Description:** An example workflow which is ran on an artifact. Takes in the artifacts value is an input and submits a WHOIS query against this input. Results are returned in the form of a note.
- Object Type:** Artifact

Metadata on the right indicates the workflow was created by 'Integrations Server A' on 17/12/2018 at 16:16, last modified by the same user, and is associated with the rule 'Run Whois Query Against Artifact'.

Below the configuration fields is a visual workflow diagram. It starts with a 'Start your workflow here' node, followed by a 'Whois: Query' function node (highlighted with a dashed blue border), and ends with a 'Note' output node. A toolbar on the left provides various icons for creating and editing the workflow.

At the bottom, the 'Post-Process Script' tab is active, showing a Python script that formats the WHOIS query results into a rich text note. The script includes comments and uses the `format` method to insert the query input and results.

```
1
2
3- if results["success"]:
4     # We have results
5     noteText = u"""Whois Query ran against input <b>{0}</b><br> Results found: <br>""".format(results.inputs["whois_query"])
6     """
7     for key, val in results.domain_details.items():
8         noteText += u"""<b> {0} : {1}</b>""".format(key, val)
9     """
10-    for keyval in zip(results.domain_details_keys, results.domain_details_values):
11        noteText += u"""<br><b> {0}</b> : {1} """.format(keyval[0].capitalize(), keyval[1])
12- else:
13     noteText = u"""Whois Query ran against input <b>{0}</b><br> No results found""".format(results.inputs["whois_query"])
14     incident.addNote(helper.createRichText(noteText))
15
```

app.config settings:

If you wish to use the integration with a proxy, this will need to be set in the app.config, otherwise no config values are needed.

```
whois_https_proxy=http://0.0.0.0:3128/
```

Function Inputs:

Function Name	Type	Required	Example
whois_query	String	Yes	'https://www.ibm.com'

Function Output:

```
results = {
  'inputs': {'whois_query': 'https://www.ibm.com'},
  'success': True,
  'domain_details': {'creation_date': '03/19/1986',
                     'expiration_date': '03/20/2019',
                     'last_updated': '09/18/2018',
                     'name': 'ibm.com',
                     'name_servers': ['ns1-206.akam.net',
                                      'usc3.akam.net',
                                      'eur5.akam.net',
                                      'asia3.akam.net',
                                      'usc2.akam.net',
                                      'usw2.akam.net',
                                      'ns1-99.akam.net',
                                      'eur2.akam.net'],
                     'registrar': 'CSC Corporate Domains, Inc.'},
  'domain_details_keys': ['name',
                          'registrar',
                          'creation_date',
                          'expiration_date',
                          'last_updated',
                          'name_servers'],
  'domain_details_values': ['ibm.com',
                           'CSC Corporate Domains, Inc.',
                           '03/19/1986',
                           '03/20/2019',
                           '09/18/2018',
                           ['ns1-206.akam.net',
                           'usc3.akam.net',
                           'eur5.akam.net',
                           'asia3.akam.net',
                           'usc2.akam.net',
                           'usw2.akam.net',
                           'ns1-99.akam.net',
                           'eur2.akam.net']]
}
```

```
}
```

Pre-Process Script:

```
inputs.whois_query = artifact.value
```

Post-Process Script:

This example **adds a Note to the Incident**.

```
if results["success"]:
    # We have results
    noteText = u"""Whois Query ran against input <b>{0}</b><br> Results
found: <br>""".format(results.inputs["whois_query"])
    """
    for key, val in results.domain_details.items():
        noteText += u'''<b> {0} : {1}'''.format(key,val)
    """
    for keyval in
zip(results.domain_details_keys,results.domain_details_values):
    noteText += u"""<br><b> {0}</b> : {1}
""".format(keyval[0].capitalize(),keyval[1])
else:
    noteText = u"""Whois Query ran against input <b>{0}</b><br> No results
found""".format(results.inputs["whois_query"])
incident.addNote(helper.createRichText(noteText))
```

Rules

Rule Name	Object Type	Workflow Triggered
Run Whois Query Against Artifact	Artifact	Example: Whois Query Against Artifact