Validation Report for Threat Analysis with Uptycs

SDK Version	Generation Time	Command Line Arguments Provided
51.0.1.0.695	2024/03/14 15:22:13	`verbose`: True, `cmd`: validate, `package`: .

Results

Severity	Count
Critical Issues:	1
Warnings:	1
Validations Passed:	36

App Details

Attribute	Value
`display_name`	Threat Analysis with Uptycs
`name`	threat_analysis_with_uptycs
`version`	1.0.0
`author`	Uptycs
`author_email`	kadirikumar@uptycs.com
`install_requires`	['resilient-circuits>=51.0.1.0.0']
`description`	Responds to threats everywhere across the cloud, endpoints, containers, and K8s systems
`long_description`	It is a robust extension that seamlessly integrates with the Uptycs platform to retrieve real-time alerts data and analyze potential security threats. This integration empowers security teams to proactively monitor, analyze, and respond to security incidents within their organization's IT infrastructure, ensuring timely detection and mitigation of threats.
`url`	https://www.uptycs.com
`entry_points`	{'resilient.circuits.configsection': 'C:\Users\kadirikumar\Downloads\QRADAR-SOAR-V1\threat_analysis_with_uptycs\threat_analysis_with_uptycs\util\config.py', 'resilient.circuits.customize': 'C:\Users\kadirikumar\Downloads\QRADAR-SOAR-V1\threat_analysis_with_uptycs\threat_analysis_with_uptycs\util\customize.py', 'resilient.circuits.selftest': 'C:\Users\kadirikumar\Downloads\QRADAR-SOAR-V1\threat_analysis_with_uptycs\threat_analysis_with_uptycs\util\selftest.py'}
`python_requires`	>=3.11
`SOAR version`	50.0.9097
`Proxy support`	Proxies supported if running on AppHost>=1.6

`setup.py` file validation

Severity	Name	Description	Solution

Success

Package files validation

LICENSE

INFO: `LICENSE` file is valid

It is recommended to manually validate the license. Suggested formats: MIT, Apache, and BSD

`MANIFEST.in`

Pass

`apikey_permissions.txt`

Pass

`Dockerfile, template match`

Pass

`Dockerfile, base image`

Pass

`config.py`

Pass

`customize.py`

Pass

`SOAR Scripts`

Pass

`README.md`

Pass

`app_logo.png`

Pass

`company_logo.png`

Pass

LICENSE

Pass

Payload samples validation

`payload_samples\uptycs_api`

Pass

`resilient-circuits` selftest

Success

---

tox tests

CRITICAL: 3 tests failed. Details:

```
self = <test_funct_uptycs_api.TestUptycsApi object at 0x00000215666794D0>

    def test_function_definition(self):
        """ Test that the package provides customization_data that defines the function """
        func = get_function_definition(PACKAGE_NAME, FUNCTION_NAME)
```

```
>       assert func is not None
E       assert None is not None

tests\test_funct_uptycs_api.py:54: AssertionError

---

self = <test_funct_uptycs_api.TestUptycsApi object at 0x00000215666E79D0>
circuits_app = <pytest_resilient_circuits.resilient_circuits_fixtures.ResilientCircuits object
at 0x0000021566662710>
mock_inputs = {`uptycs_api_endpoint`: `/`, `uptycs_api_method`: `GET`, `uptycs_api_payload`:
`{}`}

    @pytest.mark.parametrize("mock_inputs", [
        (mock_inputs_1),
        (mock_inputs_2)
    ])
    def test_success(self, circuits_app, mock_inputs):
        """ Test calling with sample values for the parameters """

        results = call_uptycs_api_function(circuits_app, mock_inputs)
>       assert(results)
E       assert None

tests\test_funct_uptycs_api.py:76: AssertionError

---

self = <test_funct_uptycs_api.TestUptycsApi object at 0x00000215666890D0>
circuits_app = <pytest_resilient_circuits.resilient_circuits_fixtures.ResilientCircuits object
at 0x0000021566662710>
mock_inputs = {`uptycs_api_endpoint`: `/assets/count`, `uptycs_api_method`: `GET`,
`uptycs_api_payload`: `{}`}

    @pytest.mark.parametrize("mock_inputs", [
        (mock_inputs_1),
        (mock_inputs_2)
    ])
    def test_success(self, circuits_app, mock_inputs):
        """ Test calling with sample values for the parameters """

        results = call_uptycs_api_function(circuits_app, mock_inputs)
>       assert(results)
E       assert None

tests\test_funct_uptycs_api.py:76: AssertionError

---
```

Run with the `-v` flag to see more information

WARNING: Unsupported tox environment found in envlist in `tox.ini` file

Tests must be configured to run only with tox environments `py36` or greater

---

Pylint Scan

INFO: Pylint scan passed with no errors

Run with `-v` to see the full pylint output

Bandit Scan

INFO: Bandit scan passed with no issues

Run again with `-v` to see the full bandit output
