PyIntelOwl

Release 4.0.0

Matteo Lodi

Oct 19, 2021

CONTENTS

Robust Python **SDK** and **Command Line Client** for interacting with IntelOwl API.

CHAPTER

ONE

INSTALLATION

```
$ pip install pyintelowl
```

CHAPTER

TWO

USAGE AS CLI

On successful installation, The `pyintelowl` entryscript should be directly invokable. For example,

```
$ pyintelowl
Usage: pyintelowl [OPTIONS] COMMAND [ARGS]...

Options:
-d, --debug  Set log level to DEBUG
--version    Show the version and exit.
-h, --help   Show this message and exit.

Commands:
analyse                Send new analysis request
analyzer-healthcheck   Send healthcheck request for an analyzer...
config                 Set or view config variables
connector-healthcheck  Send healthcheck request for a connector
get-analyzer-config    Get current state of `analyzer_config.json` from...
get-connector-config   Get current state of `connector_config.json` from...
jobs                   Manage Jobs
tags                   Manage tags
```

Configuration:

You can use `set` to set the config variables and `get` to view them.

Listing 1: View on asciinema

```
$ pyintelowl config set -k 4bf03f20add626e7138f4023e4cf52b8 -u "http://localhost:80"
$ pyintelowl config get
```

Hint: The CLI would is well-documented which will help you navigate various commands easily. Invoke `pyintelowl -h` or `pyintelowl <command> -h` to get help.

CHAPTER

THREE

USAGE AS SDK/LIBRARY

```python
from pyintelowl import IntelOwl, IntelOwlClientException
obj = IntelOwl(
    "4bf03f20add626e7138f4023e4cf52b8",
    "http://localhost:80",
    None,
)
"""
obj = IntelOwl(
    "<your_api_key>",
    "<your_intelowl_instance_url>",
    "optional<path_to_pem_file>"
)
"""

try:
    ans = obj.get_analyzer_configs()
    print(ans)
except IntelOwlClientException as e:
    print("Oh no! Error: ", e)
```

Tip: We very much **recommend** going through the `pyintelowl.pyintelowl.IntelOwl` docs.

CHAPTER

FOUR

INDEX

4.1 Modules

4.1.1 `IntelOwl` class

4.1.2 `IntelOwlClientException` class

4.2 Tests

4.2.1 Configuration

Some tests require file samples, which can be found in the encrypted folder `tests/test_files.zip` (password: "infected"). Unzip the archive in `tests/test_files` folder before running the tests.

Please remember that these are dangerous malware! They come encrypted and locked for a reason! Do NOT run them unless you are absolutely sure of what you are doing! They are to be used only for launching specific tests that require them (`__send_analysis_request`)

- With the following constants in `__init__.py`, you can customize your tests:
  - **MOCKING_CONNECTIONS:** Mock connections to external API to test functions without a real connection or a valid API Key.
- If you prefer to use custom inputs for tests, you can change the following constants:
  - **TEST_JOB_ID**
  - **TEST_HASH**
  - **TEST_URL**
  - **TEST_IP**
  - **TEST_DOMAIN**
  - **TEST_GENERIC**
  - **TEST_FILE**
  - **TEST_FILE_HASH**

4.2.2 Launch Tests

- The test requirements are specified in the `test-requirements.txt` file. Install them using,

```
$ pip3 install -r test-requirements.txt
```

- Launch the tests using `tox`:

```
$ tox
```

CHAPTER

FIVE

INDICES AND TABLES

- genindex
- modindex