stix2-patterns Documentation

Release 1.1.0

OASIS Open

May 02, 2019

Contents:

The STIX 2 Pattern Validator is a software tool for checking the syntax of the Cyber Threat Intelligence (CTI) STIX Pattern expressions, which are used within STIX to express conditions (prepresented with the Cyber Observable data model) that indicate particular cyber threat activity. The repository contains source code, an ANTLR grammar, and automated tests for the tool. The validator can be used as a command-line tool or as a Python library which can be included in other applications.

CHAPTER 1

Installation

1.1 Requirements

- Python 2.7 or 3.4+
- ANTLR grammar runtime (4.7 or newer):
  - antlr4-python2-runtime (Python 2.7)
  - antlr4-python3-runtime (Python 3)
- six
- typing (Python 3.4)

1.2 Install Package

Using pip is highly recommended:

```
$ pip install stix2-patterns
```

For more information about installing Python packages, see the Python Packaging User Guide.

CHAPTER 2

Usage

The STIX Pattern Validator provides an executable script (`validate-patterns`) in addition to being an importable Python library.

The `validate-patterns` script accepts patterns from either direct user input or a file passed as an option.

2.1 From Python Code

The `run_validator` function can be called on any Python string. It returns a list of errors encountered while parsing the pattern.

```
from stix2patterns.validator import run_validator

pattern = "[file-object:hashes.md5 = '79054025255fb1a26e4bc422aef54eb4']"
errors = run_validator(pattern)
```

2.2 User Input

When prompted, enter a pattern to validate and press enter. The validator will supply whether the pattern has passed or failed. If the pattern fails the test, the validator will supply where the first syntax error occurred. The validator will continue to prompt for patterns until Ctrl-C is pressed. Example:

```
$ validate-patterns

Enter a pattern to validate: [file-object:hashes.md5 =
↪'79054025255fb1a26e4bc422aef54eb4']

PASS: [file-object:hashes.md5 = '79054025255fb1a26e4bc422aef54eb4']
```

2.3 File Input

```
$ validate-patterns -f <path_to_file>
```

Use <path_to_file> to specify the path to a file containing a set of patterns to validate. Each pattern must be on a separate line of the file so that the validator may determine where the pattern begins and ends. The validator will supply the PASS/FAIL result of each pattern.

CHAPTER 3

Developer's Guide

3.1 Updating the Grammar

The ANTLR pattern grammar is maintained in the stix2-json-schemas repository. If the grammar changes, the code in this repository should be updated to match. To do so, use the Java ANTLR package to generate new Python source files. (The .jar file is not needed for normal use of the validator).

1. Download antlr-4.7.1-complete.jar from http://www.antlr.org/download/
2. Clone the stix2-json-schemas repository or download the STIXPattern.g4 file.
3. Change to the directory containing the STIXPattern.g4 file.
4. Run the following command

```
$ java -jar "/path/to/antlr-4.7.1-complete.jar" -Dlanguage=Python2 STIXPattern.g4␣
↪-visitor -o /path/to/cti-pattern-validator/stix2patterns/grammars
```

5. Commit the resulting files to git.

3.2 Testing

The STIX Pattern Validator's test suite can be run with pytest.

You can also test against the examples provided in the supplied example file.

```
$ validate-patterns -f stix2patterns/test/spec_examples.txt
```

CHAPTER 4

CHANGELOG

4.1 1.1.0 - Released 2018-11-20

- Add a visitor to the ANTLR parser
- Add testing for Python 3.7

4.2 1.0.0 - Released 2018-07-18

- #34 - Add documentation on ReadTheDocs: https://stix2-patterns.readthedocs.io/
- #39 - Raise error for unexepected unused character values.
- #41 - Raise error for negative REPEAT values.
- #42 - Improved Timestamp validation.
- #43 - Validate Base64 binary literals.
- #48 - Make pattern qualifier and operator keywords case-sensitive.
- Drop support for Python 2.6 and 3.3.

4.3 0.6.0 - Released 2017-11-13

- #32 - Added a public walk() method to the Pattern class. (@chisholm)
- Make repository structure match other projects. (@emmanvg)

4.4 0.5.0 - Released 2017-07-12

- Separate object and path components in inspector.
- Support "NOT" qualifier on all comparison operators.

4.5 0.4.1 - Released 2017-05-19

- Repackaged to not use a Wheel distribution

4.6 0.4.0 - Released 2017-05-19

- Encapsulated parsed patterns in a new Pattern class

4.7 0.3.0 - Released 2017-05-04

- Update for STIX 2.0 WD02.
- Add "inspector" module to extract features from patterns.
- Improve error messages.
- Update to ANTLR 4.7
- Add testing for Python 2.6 and 3.6

4.8 0.2.2 - Released 2017-03-01

- Update packaging to install correct ANTLR4 runtime depending on Python version.

4.9 0.2.0 - Released 2017-02-24

- Initial public version.

CHAPTER 5

stix2patterns

5.1 stix2patterns package

5.1.1 Subpackages

stix2patterns.grammars package

Submodules

stix2patterns.grammars.STIXPatternLexer module

stix2patterns.grammars.STIXPatternListener module

stix2patterns.grammars.STIXPatternParser module

Module contents

stix2patterns.test package

Submodules

stix2patterns.test.test_inspector module

stix2patterns.test.test_validator module

Module contents

5.1.2 Submodules

5.1.3 stix2patterns.inspector module

5.1.4 stix2patterns.pattern module

5.1.5 stix2patterns.validator module

5.1.6 Module contents

CHAPTER 6

Indices and tables

- genindex
- modindex
- search