Mantis STIX Importer Documentation

Release 0.2.0

Siemens

February 27, 2014

Contents

Contents:

CHAPTER 1

Mantis STIX Importer

A module implementing import of STIX and CybOX XML files for the Mantis Cyber Threat Intelligence Mgmt. Framework.

1.1 Documentation

The full documentation is at http://django-mantis-stix-importer.readthedocs.org.

1.2 Quickstart

Please refer to the quickstart information of MANTIS, available at http://django-mantis.readthedocs.org.

Once you are set up with MANTIS, you can use the Django `manage.py` to import STIX indicators into your system as follows:

```
$ python manage.py mantis_stix_import <xml-file>  <xml-file> ... [--settings=<path_to_your_django_set
```

Here is the output of `--help` for `mantis_stix_import`:

```
Usage: manage.py mantis_stix_import [options] xml-file xml-file ... (you can use wildcards)

Imports stix XML files of specified paths into DINGO

Options:
  -v VERBOSITY, --verbosity=VERBOSITY
                        Verbosity level; 0=minimal output, 1=normal output, 2=verbose output, 3=very
  --settings=SETTINGS   The Python path to a settings module, e.g. "myproject.settings.main".
                        If this isn't provided, the DJANGO_SETTINGS_MODULE environment variable will
  --pythonpath=PYTHONPATH
                        A directory to add to the Python path, e.g. "/home/djangoprojects/myproject"
  --traceback           Print traceback on exception
  -m MARKING_JSON, --marking_json=MARKING_JSON
                        File with json representation of information of marking to be associated wit
  -p PLACEHOLDER_FILLERS, --marking_pfill=PLACEHOLDER_FILLERS
                        Key-value pairs used to fill in placeholders in marking as described in marki
  --version             show program's version number and exit
  -h, --help            show this help message and exit
```

1.3 Acknowledgments

The basic layout for this Django app with out-of-the-box configuration of `setup.py` for easy build, submission to PyPi, etc., and Sphinx documentation tree was generated with Audrey Roy's excellent Cookiecutter and Daniel Greenfield's cookiecutter-djangopackage template.

CHAPTER 2

Installation

At the command line:

```
$ pip install django-mantis-stix-importer
```

Once this is done, you can include mantis_stix_importer as app in your Django settings, together with the apps dingos and mantis_core on which mantis_stix_importer depends:

```
INSTALLED_APPS_list = [
                       ...,
                       'dingos',
                       'mantis_core',
                       'mantis_openioc_importer',
                       'mantis_stix_importer',
                       ]
```

CHAPTER 3

Usage

Run the Django `help` command to view the commands this app makes available via the command line. Doing `--help` on any of these commands provides you with additional information about each command.

CHAPTER 4

Contributing

Contents

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways.

4.1 The issue tracker for the Django Mantis STIX Importer

The further development of the Mantis STIX importer will occur within the further development of the Django Mantis Cyber-Threat Intelligence Management Framework. So, please use https://github.com/siemens/django-mantis/issues as issue tracker for bugs, feature requests and other feedback regarding django-mantis-stix-importer.

4.2 Types of Contributions

4.2.1 Report Bugs

Report bugs at https://github.com/siemens/django-mantis/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.2.2 Fix Bugs

Look through the GitHub issues (https://github.com/siemens/django-mantis/issues) for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

4.2.3 Implement Features

Look through the GitHub issues (https://github.com/siemens/django-mantis/issues) for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.2.4 Write Documentation

Mantis STIX Importer could always use more documentation, whether as part of the official Mantis STIX Importer docs, in docstrings, or even on the web in blog posts, articles, and such.

4.2.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/siemens/django-mantis/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.3 Get Started!

Ready to contribute? Here's how to set up *django-mantis-stix-importer* for local development.

1. Fork the *django-mantis-stix-importer* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/django-mantis-stix-importer.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv django-mantis-stix-importer
   $ cd django-mantis-stix-importer/
   $ python setup.py develop
   ```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

Now you can make your changes locally.

5. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

6. Submit a pull request through the GitHub website.

4.4 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7.

CHAPTER 5

Credits

5.1 Development Lead

- Siemens <mantis.cert@siemens.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.2.0 (2014-02-26)

- Added ability to generate identifier for top-level element (usually a STIX_Package) if an identifier for that element is missing: if a default namespace has been defined, then an identifier is generated by taking the MD5-hash of the xml file.
- Markings present in STIX_Package are read out and attached to all InfoObjects generated from the STIX_Package.

  Note: Mantis does currently not interpret the XPATH expression that specifies the scope of the marking (which is not much of an issue, since it seems that the feature to restrict the scope of a marking is not much used at the moment).
- Timestamp present in *STIX_Header/Information_Source/Time/Produced_Time* is read.
- Added a command-line argument to add a default-timestamp to the STIX import command.
- Bug fixes:
  - Attributes other than *id* and *idref* that contained a namespace were not handled correctly. The handler function *attr_with_namespace_handler* fixes this.
  - In *0.1.0*, the *xsi:type* attribute was not recorded, because in most cases, its information is used for determining the data type of elements and InfoObjects. But there are cases, e.g., in Markings, where this is not the case. For these cases, the *xsi:type* attribute is kept in the InfoObject.
  - Family revision info was not recorded; this has been fixed.

6.2 0.1.0 (2013-12-19)

- Bugfixes; added documentation

6.3 0.0.9 (2013-12-18)

- First release on PyPI.