

---

# Mantis OpenIOC Importer Documentation

*Release 0.2.0*

**Siemens**

February 27, 2014



<b>1</b>	<b>Mantis OpenIOC Importer</b>	<b>3</b>
1.1	Documentation . . . . .	3
1.2	Quickstart . . . . .	3
1.3	Acknowledgments . . . . .	4
<b>2</b>	<b>Installation</b>	<b>5</b>
<b>3</b>	<b>Usage</b>	<b>7</b>
<b>4</b>	<b>Contributing</b>	<b>9</b>
4.1	The issue tracker for the Django Mantis OpenIOC Importer . . . . .	9
4.2	Types of Contributions . . . . .	9
4.3	Get Started! . . . . .	10
4.4	Pull Request Guidelines . . . . .	11
<b>5</b>	<b>Credits</b>	<b>13</b>
5.1	Development Lead . . . . .	13
5.2	Contributors . . . . .	13
<b>6</b>	<b>History</b>	<b>15</b>
6.1	0.2.0 (2014-01-26) . . . . .	15
6.2	0.1.1 (2013-12-19) . . . . .	15
6.3	0.1.0 (2013-12-19) . . . . .	15
6.4	0.0.9 (2013-12-19) . . . . .	15



Contents:



---

## Mantis OpenIOC Importer

---

A module implementing import of OpenIOC indicator XML files for the Mantis Cyber Threat Intelligence Mgmt. Framework.

### 1.1 Documentation

The full documentation is at <http://django-mantis-openioc-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 OpenIOC indicators into your system as follows:

```
$ python manage.py mantis_openioc_import <xml-file> <xml-file> ... [--settings=<path_to_your_django
```

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

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

```
Imports OpenIOC XML indicator 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 with
-p PLACEHOLDER_FILLERS, --marking_pfill=PLACEHOLDER_FILLERS
    Key-value pairs used to fill in placeholders in marking as described in mark
-n IDENTIFIER_NS_URI, --id_namespace_uri=IDENTIFIER_NS_URI
    URI of namespace used to qualify the identifiers of the created information o

--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.



---

## Installation

---

At the command line:

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

Once this is done, you can include `mantis_openioc_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',  
    ...  
]
```



---

## 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.



---

## Contributing

---

### Contents

- Contributing
  - The issue tracker for the Django Mantis OpenIOC Importer
  - Types of Contributions
    - \* Report Bugs
    - \* Fix Bugs
    - \* Implement Features
    - \* Write Documentation
    - \* Submit Feedback
  - Get Started!
  - Pull Request Guidelines

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 OpenIOC Importer

The further development of the Mantis OpenIOC 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-openioc-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 OpenIOC Importer could always use more documentation, whether as part of the official Mantis OpenIOC 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-openioc-importer* for local development.

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

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





---

**Credits**

---

## 5.1 Development Lead

- Siemens <mantis.cert@siemens.com>

## 5.2 Contributors

None yet. Why not be the first?



---

## History

---

### 6.1 0.2.0 (2014-01-26)

- Fixed bug in import of timestamp.

### 6.2 0.1.1 (2013-12-19)

- Bugfixes: id-namespace was not properly treated.

### 6.3 0.1.0 (2013-12-19)

- Bugfixes; added documentation

### 6.4 0.0.9 (2013-12-19)

- First release on PyPI.