pcf Decrypt Documentation

Release 0.1.1

Joachim Brandon LeBlanc

October 26, 2014

Contents

Contents:

CHAPTER 1

Cisco pcf Decrypt

Decrypt encoded passwords in Cisco VPN pcf files.

- Free software: MIT license
- Documentation: https://pcf_decrypt.readthedocs.org.

1.1 Features

- TODO

CHAPTER 2

Installation

At the command line:

```
$ easy_install pcf_decrypt
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv pcf_decrypt
$ pip install pcf_decrypt
```

CHAPTER 3

Usage

To use Cisco PCF Decrypt in a project:

```
import pcf_decrypt
```

The command line script:

```
usage: pcf_decrypt [-h] file_or_hash [file_or_hash ...]

Decrypt encryped passwords in Cisco pcf files

positional arguments:
  file_or_hash  File to parse or hash to decode

optional arguments:
  -h, --help    show this help message and exit

Arguments can either be a pcf file with at least one enc_FIELD field within to
decrypt or an encrypted hash to decrypt. If any hashes are present (determined
if the operating system say it's not a path to a file and can be successfully
decoded from a hex string to a byte array), they will be outputted in order
presented first, followed by any files in the form: filename:FIELD:plaintext
```

CHAPTER 4

pcf_decrypt package

4.1 Module contents

pcf_decrypt.**decrypt**(*hex_or_bin*)

exception pcf_decrypt.**PcfDecryptionError**
Bases: exceptions.Exception

pcf_decrypt.**DecodeError**
alias of TypeError

CHAPTER 5

Contributing

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

You can contribute in many ways:

5.1 Types of Contributions

5.1.1 Report Bugs

Report bugs at https://github.com/demosdemon/pcf_decrypt/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.

5.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

5.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

5.1.4 Write Documentation

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

5.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/demosdemon/pcf_decrypt/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 :)

5.2 Get Started!

Ready to contribute? Here's how to set up *pcf_decrypt* for local development.

1. Fork the *pcf_decrypt* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/pcf_decrypt.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 pcf_decrypt
   $ cd pcf_decrypt/
   $ 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. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

   ```
   $ flake8 pcf_decrypt tests
   $ python setup.py test
   $ tox
   ```

   To get flake8 and tox, just pip install them into your virtualenv.

6. 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
   ```

7. Submit a pull request through the GitHub website.

5.3 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 2.6, 2.7, 3.3, and 3.4, and for PyPy. Check https://travis-ci.org/demosdemon/pcf_decrypt/pull_requests and make sure that the tests pass for all supported Python versions.

5.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_pcf_decrypt
```

CHAPTER 6

Credits

6.1 Development Lead

- Joachim Brandon LeBlanc <demosdemon@gmail.com>

6.2 Contributors

None yet. Why not be the first?

CHAPTER 7

History

CHAPTER 8

0.1.0 (2014-10-26)

- First release on PyPI.

CHAPTER 9

0.1.1 (2014-10-26)

- Fix some README stuff

CHAPTER 10

Indices and tables

- *genindex*
- *modindex*
- *search*

p

D

P