

---

# **Aircrack-ng python 'bindings' Documentation**

***Release 0.1.0***

**David Francos Cuartero**

January 20, 2016



<b>1</b>	<b>Aircrack-ng python ‘bindings’</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	Credits . . . . .	3
<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	Types of Contributions . . . . .	9
4.2	Get Started! . . . . .	10
4.3	Pull Request Guidelines . . . . .	10
4.4	Tips . . . . .	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	0.1.0 (2015-01-20) . . . . .	15
<b>7</b>	<b>Indices and tables</b>	<b>17</b>



Contents:



---

## Aircrack-ng python ‘bindings’

---

Simple aircrack-ng implementation in python, giving access to common commands and way to read its output / control its execution

- Free software: ISC license
- Documentation: <https://pyrcrack.readthedocs.org>.

### 1.1 Features

- TODO

### 1.2 Credits

This package was created with [Cookiecutter](#) and the [audreyr/cookiecutter-pypackage](#) project template.





---

# Installation

---

At the command line:

```
$ easy_install pyrcrack
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv pyrcrack  
$ pip install pyrcrack
```



---

### Usage

---

To use Aircrack-ng python ‘bindings’ in a project:

```
import pyrcrack
```



---

## 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:

### 4.1 Types of Contributions

#### 4.1.1 Report Bugs

Report bugs at <https://github.com/XayOn/pyrcrack/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.

#### 4.1.2 Fix Bugs

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

#### 4.1.3 Implement Features

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

#### 4.1.4 Write Documentation

Aircrack-ng python ‘bindings’ could always use more documentation, whether as part of the official Aircrack-ng python ‘bindings’ docs, in docstrings, or even on the web in blog posts, articles, and such.

#### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/XayOn/pyrcrack/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 :)

## 4.2 Get Started!

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

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

```
$ git clone git@github.com:your_name_here/pyrcrack.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 pyrcrack
$ cd pyrcrack/
$ 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 pyrcrack 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.

## 4.3 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, 3.4 and 3.5, and for PyPy. Check [https://travis-ci.org/XayOn/pyrcrack/pull\\_requests](https://travis-ci.org/XayOn/pyrcrack/pull_requests) and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

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





---

**Credits**

---

## 5.1 Development Lead

- David Francos Cuartero <[opensource@davidfrancos.net](mailto:opensource@davidfrancos.net)>

## 5.2 Contributors

None yet. Why not be the first?



---

### History

---

#### 6.1 0.1.0 (2015-01-20)

- Implemented main classes as context managers:
- Aircrack
- Wesside
- Aireplay
- Airodump
- Airmon
- Airdecap



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`