

---

# **Airoscript-ng Documentation**

***Release 0.0.4***

**David Francos Cuartero**

January 08, 2015



<b>1</b>	<b>Airoscript-ng</b>	<b>3</b>
1.1	Features . . . . .	3
1.2	TODO . . . . .	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>
<b>7</b>	<b>0.0.4 (2015-09-01)</b>	<b>17</b>
<b>8</b>	<b>0.0.3 (2015-09-01)</b>	<b>19</b>
<b>9</b>	<b>0.0.2 (2015-08-01)</b>	<b>21</b>
<b>10</b>	<b>0.0.1 (2014-12-26)</b>	<b>23</b>
<b>11</b>	<b>Indices and tables</b>	<b>25</b>



Contents:



---

# Airoscript-ng

---

Airoscript-ng python complete implementation

- Free software: GNU GENERAL PUBLIC LICENSE 2
- Documentation: <http://airoscript-ng.readthedocs.org/en/master>

## 1.1 Features

- Dynamic aircrack-ng API generation (under airoscriptng.aircrack)
- Threaded execution
- Hackability assesment
- Scanning provides a list of best wireless hacking techniques
- Session control (for better process control)
- Wireless monitor interfaces are nicely handled and reused if neccesary
- XMLRPC server implementation

## 1.2 TODO

- Better parameter parsing & format for aircrack-ng parameters file, read them from manpages
- Implement attacks on airoscript-ng class
- Implement cracking (and control of it, once cracked stop all attacks against that network)
- Build a user interface (probably more than one)





---

# Installation

---

At the command line:

```
$ easy_install airoscriptng
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv airoscriptng  
$ pip install airoscriptng
```



---

## Usage

---

To use Airoscript-ng in a project:

```
import airoscriptng
```

From there, you'll have access to session managers and airoscript basic object.

You can also launch main airoscript-ng XMLRPC server just by executing airoscriptng script directly or calling the binary:

```
airoscriptngxmlrpc
```



---

## 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/airoscripntng/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

Airoscrip-ng could always use more documentation, whether as part of the official Airoscrip-ng 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/airoscripntng/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 *airoscriptng* for local development.

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

```
$ git clone git@github.com:your_name_here/airoscriptng.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 airoscriptng
$ cd airoscriptng/
$ 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 airoscriptng 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 2.7. Check [https://travis-ci.org/XayOn/airoscriptng/pull\\_requests](https://travis-ci.org/XayOn/airoscriptng/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_airoscriptng
```





---

**Credits**

---

## 5.1 Development Lead

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

## 5.2 Contributors

None yet. Why not be the first?



---

## History

---



---

**0.0.4 (2015-09-01)**

---

- Bugfix on target handling
- Bugfix on basic cracking
- First attack (deauth) implemented
- Added lots of parameters to parameter json file



---

**0.0.3 (2015-09-01)**

---

- Improved parameter handling
- Implemented basic cracking
- First attack scheduled for v0.4





---

**0.0.2 (2015-08-01)**

---

- First usable thing, still no attacks
- “Hackability” property for aps
- Integrated clients on AP object
- External plugin support
- Reaver support
- XMLRPC Working
- General cleanup



---

**0.0.1 (2014-12-26)**

---

- First release, monitor mode and scanning working



---

## Indices and tables

---

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