
Crackhash Documentation

Release 1.0.1

['5h4d0wb0y']

Oct 27, 2019

Contents:

1	Crackhash	1
1.1	Features	1
1.2	Available Services	2
1.3	Credits	2
1.4	Stargazers over time	2
2	Installation	3
2.1	Stable release	3
2.2	From sources	3
3	Usage	5
4	Contributing	7
4.1	Types of Contributions	7
4.2	Get Started!	8
4.3	Pull Request Guidelines	9
4.4	Tips	9
4.5	Deploying	9
5	History	11
5.1	v1.0.1 (2019-10-27)	11
5.2	1.0.0 (2019-09-15)	11
6	crackhash	13
6.1	crackhash package	13
7	Indices and tables	15
	Python Module Index	17
	Index	19

Crackhash is a tool that try to crack different types of hashes using free online services.

- Free software: GNU General Public License v3
- Documentation: <https://crackhash.readthedocs.io>.

1.1 Features

- Browser supports proxy configuration
- Automatic hash type identification
- Can crack hashes from a file
- **Algorithm supported**
 - MD5 - RFC 1321
 - SHA1 - RFC 3174 (FIPS 180-3)
 - SHA256 - FIPS 180-3
 - SHA384 - FIPS 180-3
 - SHA512 - FIPS 180-3
 - LM - Microsoft Windows hash NTLM - Microsoft Windows hash
 - NTLM Support

- MYSQL - MySQL 3, 4, 5 hash
- CISCO - Cisco IOS type 7 encrypted passwords
- JUNIPER - Juniper Networks \$9\$ encrypted passwords
- RMD128 Support
- RMD160 - RFC 2857
- RMD256 Support
- RMD320 Support
- GOST - RFC 583
- WHIRLPOOL - ISO/IEC 10118-3:2004
- LDAP_MD5 - MD5 Base64 encoded
- LDAP_SHA1 - SHA1 Base64 encoded

1.2 Available Services

It checks the hash in the following services on-line:

- md5.my-addr.com
- nitrxgen.net
- md5decrypt.net
- hashcrack.com
- hashtoolkit.com
- md5hashing.net
- hashkiller.co.uk
- password-decrypt.com
- m00nie.com
- firewallruletest.com
- ifm.net.nz
- ibeast.com
- cmd5.org
- rainbowtables.it64.com

1.3 Credits

This package was developed by @5h4d0wb0y_.

1.4 Stargazers over time

2.1 Stable release

To install Crackhash, run this command in your terminal:

```
$ pip install crackhash
```

This is the preferred method to install Crackhash, as it will always install the most recent stable release.

If you don't have [pip](#) installed, this [Python installation guide](#) can guide you through the process.

2.2 From sources

The sources for Crackhash can be downloaded from the [Github repo](#).

You can either clone the public repository:

```
$ git clone git://github.com/5h4d0wb0y/crackhash
```

Or download the [tarball](#):

```
$ curl -OL https://github.com/5h4d0wb0y/crackhash/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```


CHAPTER 3

Usage

```
usage: crackhash [OPTIONS]
```

```
Crackhash is a tool that try to crack different types of hashes using free
online services.
```

```
Options:
```

```
--use-proxy / --no-proxy      Set a proxy to use
--proxy-host TEXT             Specify the proxy host
--proxy-port INTEGER          Specify the proxy port
--proxy-user TEXT             Specify a proxy user
--proxy-pass TEXT             Specify a proxy user's password
-a, --algo                 Specify hash' algorithm
→ [md5|sha1|sha256|sha384|sha512|rm
→ md5|ldap_sha1]              Specify a hash to crack
                                Specify file containing hashes, formatted
                                this way: hashtype:hashvalue
-h, --hash TEXT               Show this message and exit.
-f, --file FILENAME
--help
```

If the *algo* parameter is not specified, crackhash is able to detect the hash and try to crack it:

```
$ crackhash -h 098f6bcd4621d373cade4e832627b4f6
```

Otherwise try to crack it by specifying the *algo* parameter:

```
$ crackhash -a md5 -h 098f6bcd4621d373cade4e832627b4f6
```

Try to crack it using a proxy:

```
$ crackhash --use-proxy --proxy-host 127.0.0.1 --proxy-port 9050 -h 
→ 098f6bcd4621d373cade4e832627b4f6
```

Crackhash is able to parse a file containing hashes and try to crack them as fast as possible:

```
$ cat ~/crackhash-hashes.txt
md5:5f4dcc3b5aa765d61d8327deb882cf99
sha1:5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
sha256:5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
sha384:a8b64babd0aca91a59bdbb7761b421d4f2bb38280d3a75ba0f21f2bebc45583d446c598660c94ce680c47d19c3078
sha512:b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f132
ripemd128:c9c6d316d6dc4d952a789fd4b8858ed7
ripemd160:2c08e8f5884750a7b99f6f2f342fc638db25ff31
ripemd256:f94cf96c79103c3ccad10d308c02a1db73b986e2c48962e96ecd305e0b80ef1b
ripemd320:c571d82e535de67ff5f87e417b3d53125f2d83ed7598b89d74483e6c0dfe8d86e88b380249fc8fb4
mysql:*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
lm:E52CAC67419A9A224A3B108F3FA6CB6D
ntlm:8846F7EAE8FB117AD06BDD830B7586C
whirlpool:74dfc2b27acfa364da55f93a5caee29ccad3557247eda238831b3e9bd931b01d77fe994e4f12b9d4cfa92a1244
gost:db4d9992897eda89b50f1d3208db607902da7e79c6f3bc6e6933cc5919068564
cisco:021605481811003348
juniper:$9$QHbgz/tu0IcrvBIwgJDmPBIEhSe
ldap_md5:{MD5}NWY0ZGNjM2I1YWE3NjVkJFkODMyN2RlYjg4MmNmOTkK
ldap_sha1:{SHA}NWJhYTYxZTRjOWI5M2YzZjA2ODIyNTBiNmNmODMzMWI3ZWU2OGZkOAo=
$ crackhash -f ~/crackhash-hashes.txt
```

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/5h4d0wb0y/crackhash/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” and “help wanted” is open to whoever wants to implement it.

4.1.3 Implement Features

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

4.1.4 Write Documentation

Crackhash could always use more documentation, whether as part of the official Crackhash 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/5h4d0wb0y/crackhash/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 *crackhash* for local development.

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

```
$ git clone git@github.com:your_name_here/crackhash.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 crackhash
$ cd crackhash/
$ 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 crackhash tests
$ python setup.py test or 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, 3.4, 3.5 and 3.6, and for PyPy. Check https://travis-ci.org/5h4d0wb0y/crackhash/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_crackhash
```

4.5 Deploying

A reminder for the maintainers on how to deploy. Make sure all your changes are committed (including an entry in HISTORY.rst). Then run:

```
$ bumpversion patch # possible: major / minor / patch
$ git push
$ git push --tags
```

Travis will then deploy to PyPI if tests pass.

unreleased v1.0.2

5.1 v1.0.1 (2019-10-27)

- Check if the browser has been started on travis
- Added chromedriver installation on travis
- Removed unused imports and variables and conformed to the PEP 8 style guide
- Added installation of codecov in travis
- Removed setuptools-changelog package, use only bump2version to change history
- Fixed phony and added other commands in makefile
- Changed python to python3 in makefile
- Added chrome browser in travis
- Updated features in README.rst
- Removed unused module terminaltables
- Misc: small fix or general refactoring

5.2 1.0.0 (2019-09-15)

- First release on PyPI.

6.1 crackhash package

6.1.1 Submodules

6.1.2 crackhash.browser module

6.1.3 crackhash.cli module

6.1.4 crackhash.core module

6.1.5 crackhash.helpers module

6.1.6 crackhash.scrapers module

6.1.7 Module contents

Top-level package for Crackhash.

CHAPTER 7

Indices and tables

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

C

crackhash, [13](#)

C

crackhash (*module*), [13](#)