Safe Documentation

Release 0.4

Hsiaoming Yang

January 21, 2016

Contents

Is your password safe? **Safe** will check the password strength for you.

CHAPTER 1

How it works

Safe will check if the password has a simple pattern, for instance:

1. password is in the order on your QWERT keyboards.
2. password is simple alphabet step by step, such as: abcd, 1357

Safe will check if the password is a common used password. Many thanks to Mark Burnett for the great work on 10000 Top Passwords.

Safe will check if the password has mixed number, alphabet, marks.

CHAPTER 2

Installation

Install Safe with pip:

```
$ pip install Safe
```

If pip is not available, try easy_install:

```
$ easy_install Safe
```

CHAPTER 3

Usage

It's very simple to check the strength of a password:

```
>>> import safe
>>> safe.check(1)
terrible
>>> safe.check('password')
simpile
>>> safe.check('is.safe.password')
medium
>>> safe.check('x*V-92Ba')
strong
>>> strength = safe.check('x*V-92Ba')
>>> bool(strength)
True
>>> repr(strength)
'strong'
>>> str(strength)
'password is perfect'
>>> strength.valid
True
>>> strength.strength
'strong'
>>> strength.message
'password is perfect'
```

CHAPTER 4

Environ Variables

1. **PYTHON_SAFE_WORDS_CACHE**: cache words in this file, default is a tempfile
2. **PYTHON_SAFE_WORDS_FILE**: words vocabulary file, default is the 10k top passwords

CHAPTER 5

Other Implementations

1. **JavaScript**: lepture/safe.js

CHAPTER 6

Developer Guide

Here is the API reference for safe.

safe.**check**(*raw*, *length=8*, *freq=0*, *min_types=3*, *level=3*)
Check the safety level of the password.

Parameters

- **raw** – raw text password.
- **length** – minimal length of the password.
- **freq** – minimum frequency.
- **min_types** – minimum character family.
- **level** – minimum level to validate a password.

class safe.**Strength**(*valid*, *strength*, *message*)
Measure the strength of a password.

Here are some common usages of strength:

```
>>> strength = Strength(True, 'strong', 'password is perfect')
>>> bool(strength)
True
>>> repr(strength)
'strong'
>>> str(strength)
'password is perfect'
```

Parameters

- **valid** – if the password is valid to use
- **strength** – the strength level of the password
- **message** – a message related to the password

safe.**is_asdf**(*raw*)
If the password is in the order on keyboard.

safe.**is_by_step**(*raw*)
If the password is alphabet step by step.

safe.**is_common_password**(*raw*, *freq=0*)
If the password is common used.

10k top passwords: https://xato.net/passwords/more-top-worst-passwords/

CHAPTER 7

Changelog

Here is the full history of safe.

7.1 Version 0.3

Released on Jul 28, 2014

1. API changes. Use `check()` instead of safety.
2. Cache file with a version number.

7.2 Version 0.2

Released on Jul 24, 2014

1. Typo and bugfix
2. API changes in `Strength`
3. Remove threading
4. Add environment variable

7.3 Version 0.1

First preview release.

S

C

I

S