
nisteag Documentation

Release 0.4.0

Diogo Baeder

September 27, 2016

1	Overview	3
2	Installation	5
3	Examples	7
3.1	Library	7
3.2	Command-line	7
4	Development	9
5	Indices and tables	11

An implementation, in Python, of recommendations from the NIST Electronic Authentication Guideline. The starting implementation will be for Special Publication 800-63-2: <http://dx.doi.org/10.6028/NIST.SP.800-63-2>

The full documentation is here: <http://nisteag.readthedocs.io/>

Contents:

Overview

This Python package was built with the intention of implementing most of the recommendations in NIST Special Publication 800-63-2, titled “Electronic Authentication Guideline”. This is the link for the document: <http://dx.doi.org/10.6028/NIST.SP.800-63-2> (if the link doesn’t work, you can download the document from [here](#))

The main reason behind this implementation is to cover the need to check if passwords or pass-phrases meet minimum requirements in the system that uses it; Since “strong password” is mostly used in a subjective manner, I felt the need of a more research-based way of determining how strong or weak a password is. And this publication by NIST seemed to be the best resource available for this.

The first published version will contain checkers for levels 1 and 2 for Memorized Secret Tokens, but the intention is to organically grow the package and include implementation for other recommendations, and not only token verification.

Installation

To install *nisteag*, just run:

```
$ pip install nisteag
```

You can also use `easy_install`:

```
$ easy_install nisteag
```

The package also installs a command-line script; see [Command-line](#)

Examples

3.1 Library

Checking that a password meets the minimum requirements:

```
from nisteag.token.requirements.memorized import BaseThrottler, Level1Checker

class MyThrottler(BaseThrottler):
    def check(self, username, token):
        """Verify that the token hasn't failed too many times and too frequently."""

checker = Level1Checker(MyThrottler())

checker.check('This Is a blg and r3l3v4nt passwrod!')
checker.check('this') # will fail, however.

# also fails, since the token matches the provided word dictionary
checker.check('known one', dictionary=['known one', 'something else'])

# also fails, since it's an anagram of the username
checker.check('silent', username='listen')
```

You can also just calculate the entropy directly, if you want:

```
from nisteag.entropy import EntropyCalculator

calculator = EntropyCalculator()
calculator.calculate('abcd') # returns 10.0
```

3.2 Command-line

You can also check the entropy of a password from the command-line:

```
$ check-entropy abcd
10.0

$ check-entropy # will pick the password via user input
```

Development

This is the repository for the project: <https://github.com/yougov/nisteag>

Indices and tables

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