LibreAuth Documentation

Release 0.1.0.dev3

Rodolphe Bréard

Sep 15, 2021

Contents

Python bindings to the LibreAuth library. LibreAuth is a collection of tools for user authentication written in Rust.

CHAPTER 1

Features

This is a work in progress. Some features may not be available.

- Password / passphrase authentication
  - ✓ no character-set limitation
  - ✓ reasonable lenth limit (security vs. DOS)
  - ✓ strong, evolutive and retro-compatible password hashing functions
  - ✓ optional NIST Special Publication 800-63B compatibility
- HOTP - HMAC-based One-time Password Algorithm (OATH - RFC 4226)
  - the key can be passed as bytes, an ASCII string, an hexadicimal string or a base32 string
  - customizable counter
  - customizable hash function (sha1, sha256, sha512)
  - customizable output length
  - customizable output alphabet
- TOTP - Time-based One-time Password Algorithm (OATH - RFC 6238)
  - the key can be passed as bytes, an ASCII string, an hexadicimal string or a base32 string
  - customizable timestamp
  - customizable period
  - customizable initial time (T0)
  - customizable hash function (sha1, sha256, sha512)
  - customizable output length
  - customizable output alphabet
  - customizable positive and negative period tolerance

CHAPTER 2

Reference

2.1 Install

In order to work, you need to install LibreAuth 0.6 or higher.

2.1.1 Installing Rust with rustup

LibreAuth is developed in Rust. If you do not already have the latest stable version of the Rust compiler, you can install it with rustup.

```
curl https://sh.rustup.rs -sSf | sh
rustc --version
cargo --version
```

2.1.2 Building LibreAuth

Now that we have the Rust compiler, let's download and install LibreAuth.

```
wget 'https://github.com/breard-r/libreauth/archive/v0.6.0.tar.gz' -O '/tmp/libreauth.
↪tar.gz'
tar -xvf '/tmp/libreauth.tar.gz'
cd 'libreauth-0.6.0'
make
sudo make install
```

It is not mandatory to install it system-wide. You can also copy the file `target/release/liblibreauth.so` anywhere and specify its path using the `LIBREAUTH_LIB_PATH` environment variable.

2.2 Password module

2.2.1 Hashing a password

```
from libreauth.password import *

password = b'my super secret password'
hashed = password_hash(password)
```

2.2.2 Verifying a password against the hash

```
from libreauth.password import *

password = b'user submited password'
hashed = ''
if is_valid(password, hashed):
    // Successful authentication
    pass
else:
    // Failed authentication
    pass
```

CHAPTER 3

Indices and tables

- genindex