keepassx

Release 0.1.0

March 17, 2016

Contents

1 What is this project and why should I care? **3**

2 What's a Password Manager? **5**

3 I've never hard of KeePassx, what is it? **7**

4 Aren't there similar projects already? **9**

5 I'd like to try out this project, what do I do? **11**

6 I just want reference/API docs, where are they? **13**

7 Table of Contents **15**
- 7.1 Getting Started Guide . . . 15
- 7.2 Topics . . . 18
- 7.3 Reference Guide . . . 18

8 Changelog **21**
- 8.1 0.1.0 . . . 21
- 8.2 0.0.3 . . . 21

9 Indices and tables **23**

Python Module Index **25**

KeePassX is a cross platform password manager. However, KeePassX is a desktop GUI application. That's where `python-keepassx` comes in. By using `python-keepassx`, you can access your passwords using a command line interface to KeePassX. You can also use the python library directly in your own python applications.

Python-keepassx: **simple**, command line interface to your passwords.

```
$ kp get github
Password:

title:     GitHub
username:  jamesls
url:       https://github.com/jamesls/
notes:

Password has been copied to clipboard.
```

See the Getting Started Guide to start using `python-keepassx` now.

CHAPTER 1

What is this project and why should I care?

Keepassx is a great password manager. However, if you're like me, you're in the terminal frequently. It would be better if you could access your passwords in the terminal, or even better, in the python code you write. python-keepassx can read and write the database files that keepassx uses, and in doing so allows you to access your passwords.

CHAPTER 2

What's a Password Manager?

We use things that require passwords. From social media such as Facebook, Twitter, and Instagram to things like online banking and tax returns nearly everything we access requires a username and password. It's not uncommon these days to have to deal with over 100 passwords.

If you're following password best practices, you should not be using the same password for more than one site. This makes sense, if you use the same password for your Facebook account and your online banking account, if someone gets your Facebook password, they can now access your online banking account. Also, you shouldn't be using passwords that are easy to guess: no dictionary words, birthdays, family and pet names, addresses, or any personal information. How do we deal with the fact that we need to create and remember 100s of these passwords?

That's where a password manager comes into play. A password manager is an application that you use to enter all of your passwords and usernames. This database of passwords is then secured with some form of a master password and optional key file.

Now whenver you need to log in to a site, you use the password manager, and enter your master password to gain access to the site specific password.

The benefit of this approach is that you can still use a unique (and even randomly generated) password for each password, but at the same time only have to remember a single master password.

CHAPTER 3

I've never hard of KeePassx, what is it?

KeePassX is a password manager. Check out it's homepage. It is based off of the KeePass application, which is only available on windows.

Some of the biggest benefits for using keepassx including:

- Free
- Cross platform
- Open source

The last two options are a really big deal. I use keepassx on windows linux, mac, iPhone, and Ipad. The fact that it's open source makes it easy to port to any platform. It also makes it easy to audit the code and see exactly how it's storing your passwords. The fact that it's open source means you never have to worry about a vendor going away and you being completely out of luck.

CHAPTER 4

Aren't there similar projects already?

Yes. This project is different because it has:

- A simple, straightforward API.
- Full support for key files.
- Both a command line interface and a python API.
- Support for python 2.7, 3.3, and 3.4 (and higher).
- High test coverage
- Thorough documentation.

CHAPTER 5

I'd like to try out this project, what do I do?

Check out the Getting Started Guide for an introduction to using python-keepassx.

CHAPTER 6

I just want reference/API docs, where are they?

Check out the reference docs here: Reference Guide.

CHAPTER 7

Table of Contents

7.1 Getting Started Guide

This tutorial will walk you through how to get up and running with python-keepassx. For the rest of this document, we'll refer to KeePassX GUI as the original KeePassX desktop application, and `keepassx` as the python library.

7.1.1 Installing keepassx

`keepassx` is a python package so you can use pip or your preferred methods to install `keepassx`. In this tutorial we'll use pip:

```
$ pip install keepassx
```

Should should now have a `kp` executable available on the command line:

```
$ kp --version
```

You are now ready to start using `keepassx`.

7.1.2 First Steps

To get started, we're going to use a demo database so can expirement with `kp` features.

First we need to download the demo database. You can download the demo databases here:

```
$ wget https://github.com/jamesls/python-keepassx/raw/master/misc/demo.kdb
```

The first thing we can do is list the contents of the kdb file. The password for the demo kdb file is "password".

```
$ kp -d demo.kdb list
Password:
Entries:

+---------+----------------------------------+-----------+
| Title   |               Uuid               | GroupName |
+---------+----------------------------------+-----------+
| Github  | 477ee351ada4883c7b018a0535ab1a5d | Internet  |
| Gmail   | 297ee351218556022ef663376783dabd | eMail     |
| mytitle | c4d301502050cd695e353b16094be4a7 | Internet  |
+---------+----------------------------------+-----------+
```

From the output above, we can see that there are three entries available, with the titles "Github", "Gmail", and "mytitle". We can get the username and password of an account in a number of ways. First, we can refer to the Title of the entry:

```
$ kp -d demo.kdb get Github
Password:


title:     Github
username:  githubuser
url:       github.com/githubuser
notes:


Password has been copied to clipboard.
```

We can see that the username is "githubuser" and that our password has been copied to the clipboard. We can confirm this by pasting the contents. On a mac we can run:

```
$ echo $(pbpaste)
mypassword
```

Efficient Retrieval

We can improve how we retrive passwords. First, we can set an env var so we don't have to repeatedly type '-d demo.kdb'

```
$ export KP_DB_FILE=./demo.kdb
$ kp list
 Password:
 Entries:

 +---------+----------------------------------+-----------+
 | Title   |               Uuid               | GroupName |
 +---------+----------------------------------+-----------+
 | Github  | 477ee351ada4883c7b018a0535ab1a5d | Internet  |
 | Gmail   | 297ee351218556022ef663376783dabd | eMail     |
 | mytitle | c4d301502050cd695e353b16094be4a7 | Internet  |
 +---------+----------------------------------+-----------+
```

Secondly, we have a few options when specifying the entry we're looking for. In the example above we used the exact Title of the entry to retrive the entry details. However, we can also do the following:

```
# Get by uuid
$ kp get 477ee351ada4883c7b018a0535ab1a5d

# Case insensitive matching.
$ kp get github

# Prefix matching
$ kp get git

# Fuzzy matching
$ kp get githbu
```

In the case of fuzzy matching, it's possible that multiple results can be matched. When this happens, the most relevant entry will be displayed.

Controlling Output

You can also control which fields are displayed by specifying the fields you want after a get command. For example:

```
$ kp get github title username
Password:


title:     Github
username:  githubuser


Password has been copied to clipboard.
```

In the example above, we are only showing the title and username. The available fields are:

Name	Description
uuid	A unique identifier associated with the entry.
group	The group associated with this entry (one Group can have many entries).
imageid	The id of the image associated with theis entry.
title	The title of the entry.
url	A url for the entry. This can be the login URL for a website.
username	The username of the entry.
notes	Any misc. notes associated with the entry.
creation_time	The time the entry was created.
last_mod_time	The time the entry was last modified.
last_acc_time	The time the entry was last accessed.
expiration_time	The time the entry expires.

Scripting

`python-keepassx` is a CLI and is written such that it is possible to use in a scripting environment. Here's a few tips for working with `kp` in scripts.

First, you can prevent copying to the clipboard by using the `-n/--no-clipboard-copy` option. For example, if you wanted to get the username for your github account you could run:

```
$ username=$(kp get -n github username | awk '{print $2}')
Password:
$ echo "Your username is $username"
```

7.1.3 Next steps

This tutorial covered using an existing kdb file to list and get passwords. The next steps would be to create your own kdb files. Currently, `python-keepassx` does not support creating kdb files (though this is a planned feature). For now you will have to download keepassx and create your own kdb files.

Another powerful feature of keepassx worth investivating is using keyfiles. `python-keepassx` supports keyfiles via the *-k* argument or the `KP_KEY_FILE` environment variable. Check out the Topics Guide for more info on setting this up..

7.2 Topics

7.2.1 Key File Support

keepassx fully supports key files. You can provide a key file through the `-k/--key-file` argument or the `KP_KEY_FILE` environment variable. A key file can be used either as an alternative to a password or in combination with a password. By using both a key file and a password to secure your password database, you will require both something you know (your password) and something you have (your keyfile) in order to access the entries in the password database. For more information on key files, see the keepass docs.

7.2.2 Config File

Instead of specifying the `-d/--db-file` and the `-k/--keyfile` everytime you can instead put these values in a config file. The config file is located at `~/.kpconfig`. and is a yaml file with the following format:

```
db_file: /path/to/dbfile.kdb
key_file: /path/to/keyfile.key
```

Below is a summary of the options you have available for providing your db/key file:

Config Option	Command Line Argument	Environment Variable	Config File Name
DB File	`-d/--db-file foo.kdb`	`KP_DB_FILE=foo.kdb`	`db_file: foo.kdb`
Key File	`-k/--key-file foo.key`	`KP_KEY_FILE=foo.key`	`key_file: foo.key`

In the table above, the precedence is from left to right. So, for example, the `-d` option will trump the `KP_DB_FILE` option, and the `KP_DB_FILE` option will trump the `db_file:` value in the `~/.kpconfig` config file.

7.3 Reference Guide

7.3.1 DB Object

class `keepassx.db.`**`Database`**(*contents*, *password=None*, *key_file_contents=None*)

Database representing a KDB file.

`find_by_title`(*title*)

Find an entry by exact title.

Raise EntryNotFoundError

`find_by_uuid`(*uuid*)

Find an entry by uuid.

Raise EntryNotFoundError

`fuzzy_search_by_title`(*title*, *ignore_groups=None*)

Find an entry by by fuzzy match.

This will check things such as:

- case insensitive matching
- typo checks
- prefix matches

If the `ignore_groups` argument is provided, then any matching entries in the `ignore_groups` list will not be returned. This argument can be used to filter out groups you are not interested in.

Returns a list of matches (an empty list is returned if no matches are found).

class `keepassx.db.`**`Entry`**
A password entry in a KDB file.

class `keepassx.db.`**`Group`**
The group associated with an entry.

class `keepassx.db.`**`Header`**(*contents*)
Header information for the keepass database.

From the KeePass doc:

```
Database header: [DBHDR]

[ 4 bytes] DWORD    dwSignature1  = 0x9AA2D903
[ 4 bytes] DWORD    dwSignature2  = 0xB54BFB65
[ 4 bytes] DWORD    dwFlags
[ 4 bytes] DWORD    dwVersion       { Ve.Ve.Mj.Mj:Mn.Mn.Bl.Bl }
[16 bytes] BYTE{16} aMasterSeed
[16 bytes] BYTE{16} aEncryptionIV
[ 4 bytes] DWORD    dwGroups        Number of groups in database
[ 4 bytes] DWORD    dwEntries       Number of entries in database
[32 bytes] BYTE{32} aContentsHash   SHA-256 of the plain contents
[32 bytes] BYTE{32} aMasterSeed2    Used for the dwKeyEncRounds AES
                                    master key transformations
[ 4 bytes] DWORD    dwKeyEncRounds  See above; number of transformations

Notes:

- dwFlags is a bitmap, which can include:
  * PWM_FLAG_SHA2     (1) for SHA-2.
  * PWM_FLAG_RIJNDAEL (2) for AES (Rijndael).
  * PWM_FLAG_ARCFOUR  (4) for ARC4.
  * PWM_FLAG_TWOFISH  (8) for Twofish.
- aMasterSeed is a salt that gets hashed with the transformed user master
  key to form the final database data encryption/decryption key.
  * FinalKey = SHA-256(aMasterSeed, TransformedUserMasterKey)
- aEncryptionIV is the initialization vector used by AES/Twofish for
  encrypting/decrypting the database data.
- aContentsHash: "plain contents" refers to the database file, minus the
  database header, decrypted by FinalKey.
  * PlainContents = Decrypt_with_FinalKey(DatabaseFile - DatabaseHeader)
```

CHAPTER 8

Changelog

8.1 0.1.0

- [feature] Add `--stdin` option to read the master password from stdin.
- [bugfix] Fix issue with entries that contained non-ascii characters in their passwords (issue 4).
- [bugfix] Fix issue with master password containing non-ascii characters (issue 3).
- [bugfix] Don't print traceback on invalid passwords.

8.2 0.0.3

- [bugfix] Support key file only kdb files (pull request)
- [docs] Add section on key files.
- [docs] Add section on config files.
- [docs] Switch to guzzle sphinx theme.
- [feature] Support linux clipboard copy via xclip.

CHAPTER 9

Indices and tables

- genindex
- modindex
- search

k

keepassx, 18
keepassx.db, 18

D

Database (class in keepassx.db), 18

E

Entry (class in keepassx.db), 19

F

find_by_title() (keepassx.db.Database method), 18
find_by_uuid() (keepassx.db.Database method), 18
fuzzy_search_by_title() (keepassx.db.Database method), 18

G

Group (class in keepassx.db), 19

H

Header (class in keepassx.db), 19

K

keepassx (module), 18
keepassx.db (module), 18