BeanTresor

Project Summary

Kim D. Jeker
kije-dev

January 31, 2015

Contents

1 Introduction

The Problem(s)...

Today, every user has do deal with dozens or even hunderts of passwords daily. This often causes users to reuse the same password on multiple sites.

While this seems very convenient, it's very insecure and can be even dangerous[1], because if someone can retrive the password from one site, he is able to login to other sites where the user also has used this password.

One might argue, nowadays this isn't as much a problem as it was some years ago, since most Online Services started to hash[2] their user's passwords. While this is true, we need to acknowledge that breaking into a database is not the only way to get passwords from users. For example, an attacker can also spy on a user's connection, or he can simply brute-force the password.

Unfortunatelly, the use of a new password for every site (**one-time passwords**) will not prevent attackers from retriving passwords. But it will minimize the damage caused to the user in such a case.

An other problem is the strength of passwords. In order to remember a password, users often use the name of one of their friends, family members, pet, girl- or boyfriend etc...as their password, which isn't very secure, especially if the attacker knows his victim or the victim has published the required information publicly on the internet.

... and the solution...

To make it, nevertheless, easy and convenient for the user to create and use **secure** one-time passwords, we need password managers.

Such a password manager will be **BeanTresor**. It will be focused on a very user-friendly user interface and should be cross-platform and open-source.

[1] e.g. use the same password for `themaliciousshop.com` as for e-Banking
[2] Often also refered as "encrypt", but this is technically not quite correct

2 General Description & Goals

BeanTresor is a secure password manager which helps the user to store and fill in their passwords for e.g. Websites, Servers, Bank-Accounts, Credit-Cards, etc...It also helps the user to create secure (one-time) passwords.

BeanTresor should be developed with the following things in mind:

2.1 Cross-Platform

Since we need our passwords everywhere, it's important that BeanTresor is cross-platform. This does not only include Desktop-Platforms, it also includes mobile platforms such as Android and iOS (and later maybe Windows Mobile, etc...) and maybe even devices like Smart Watches or other wearables (Smart Glasses, etc...).

2.2 User-friendly

It's also very important, that BeanTresor has a user-friendly user interface, because the user has to use the application often, and don't want to spend much time with searching a particular password or function. A good UI also makes it easier for new users to get used to the application. This may also increase the popularity and adoption rate of the application.

2.3 Open-Source

BeanTresor will also use Crypto to prevent unauthorized from having access to the users passwords. This is also important, if the user wants to store the password database on an online storage such as Dropbox, ect...To make the used crypto verifiable by everyone, and because I belive in Open-Source, BeanTresor will be completely opensource and transparent.

Also, to make the

2.4 Documentation

The Project shall be well-documente (UML, Specifications, Source comments), to make it easy for other developers to fork or contribute to the project.

2.5 Tests

To ensure high code quality and simplifying development and testing, BeanTresor should be developed test-driven.

2.6 Standards & Compatibility

BeanTresor should rely as much on known standards as possible. This makes it not only most compatible with other software, it also improves the security of the application.

2.7 Strong Encryption & Privacy

BeanTresor must use very strong crypto. It's important, that only allowed users have access to the informations stored by BeanTresor. It also should do everything possible to protect the users privacy (in every aspect).

3 Legal

BeanTresor (and all it's coresponding Documents [like this Specification, etc...], Assets and Files) are, unless stated otherwise, released under the GNU General Public License Version 3 (**GNU GPLv3**).

License: see seperate LICENCE-File