PA193 Project Documentation

Review of KeePassX application

Lenka Bačinská Ondřej Koutský Lukáš Němec

12.12. 2013

Contents

1 Introduction of KeePassX

KeePassX is open-source, cross-platform and desktop-based application aiming at providing users with possibilities for managing their passwords. KeePassX offers options for storing not only passwords but also different information such as user names, URLs, attachments and comments in one single database. The complete database is always encrypted either with AES or Twofish encryption algorithm using a 256 bit key. KeePassX also includes a little utility for secure password generation.

KeePassX project is hosted by GitHub (https://github.com/keepassx/keepassx.git) and is published under the terms of GNU General Public Licence.

2 Static analysis results

In order to perform automated static analysis of project's code, Cppcheck tool was applied. However, even thought the source code consist of nearly 50,000 lines of code, the tool was not able to discover any serious, important or other worth considering problem.

3 Evaluation of application functionality

Despite the fact that there were no serious problems discovered by static analysis tools, deeper manual review of application's source resulted in few observations. In following sections evaluation of KeePassX's functionalities is presented.

3.1 Password database

Application provides a user with possibility to store whole set of strong passwords in database encrypted by 256 bit AES algorithm. The database is protected by master password which serves as encryption key (after hashing by SHA256 algorithm). User is therefore no more required to remember any password but the master one. This settings, however, is weakened by the fact, that strength of master password is fully left to the user to be defined. Security of whole system then depends on the strength of master password which might be (and in most cases also will be) weaker then the strength of passwords stored in database. User defined passwords will also easily tend to be vulnerable against dictionary attacks.

Application also offers option to protect database with external key file. However, this file has to be accessible for the application, therefore located on HDD or external storage. This might turn to be even weaker solution then the one with user defined master password.

Generally, there should be some limitation regarding strength of master password. Other possibility would be to implement support for smartcards. Database could then be encrypted by the smartcard itself.

3.2 Password generator

Password generator is feature of KeePassX offering user to generate random passwords. User is allowed to specify length of password and also what set of characters should be used for generating.

As a drawback of password generating utility, used source of randomness could be mentioned. Application uses libgcrypt library with GCRY_STRONG_RANDOM setting, which is usually recommended for session keys etc. User is not allowed to set strength of randomness for longer term passwords specifically. Even thought this settings is sufficient for majority of cases, some switch to enable better level of randomness could be sometimes useful (flag GCRY_VERY_STRONG_RANDOM for example).

4 Summary

Despite the problems described in forgoing sections, KeePassX can be considered as useful tool for managing passwords. In the case when strong master password is chosen for encrypting the database, application allows users to use much more different and much stronger passwords then they would be able to remember without using it. Possibilities to set expiration dates for passwords, cross-platform support, well-arranged GUI and easy installation are also worth mentioning.