

Password Security: KeePass Password Safe

There are a number of recommendations that should be followed in order to use passwords securely. First of all a password needs to be sufficiently long and complex (i.e. drawn from a large set of characters) to make a brute force attack infeasible. Especially in the absence of rate-limiting or account blocking after failed attempts; such as when using poorly developed applications or if there is a possibility of an offline attack [1]. Secondly, dictionary words, common or previously compromised passwords, context related information (e.g. the website name), or easily obtained personal information (e.g. birthdays, phone numbers) should not be used to create a password. Otherwise the password would be vulnerable to dictionary attacks (a type of brute force attack where only likely possibilities are attempted). Finally the password should be kept private, never stored in a plain text file or written down [2].

These requirements are further complicated by the importance of using unique passwords for different services. If one of these services were attacked and the password exposed, the attacker could gain access to any other services using the same password. There are many reasons why this could be possible, some examples include: An application that fails to properly implement password hashing; perhaps using a weak algorithm, or by not using salt, making the passwords vulnerable to rainbow table attacks (a table of precomputed hash chains) [3]. Or misconfigured server logging may capture and store passwords as plain text [4, 5].

The more services that a person uses it becomes increasingly difficult for them to remember good unique passwords for each one. Research by Experian plc found that 'people have on average up to 26 online accounts protected by only five different passwords' [6]. In my personal experience I found that I was using the same password for countless different websites, all using the same email address. A possible solution to this problem was to use a password manager [7], this is a utility that removes the need to remember numerous passwords by storing them in an encrypted database accessible using a single master password¹.

I elected to use KeePass Password Safe [8], it has the advantage of being free and open source, although compared to premium services (such as LastPass) it does require the user to be entirely responsible for storing and synchronising the database file. Fortunately in my situation this was not a problem as I was already self hosting a Seafile file server. This setup adds a further layer of security because not only is the KeePass database file encrypted itself with its own password, no external parties can gain access to the file because it is only stored on my own encrypted disks and it is synchronised between them over HTTPS.

When using KeePass on a desktop environment² I have installed KeePassHttp, a plugin that exposes the database through a HTTP server, making entries accessible to browser plugins such as chromeIPass³. Not only does browser integration improve usability it potentially adds protection against phishing: The browser plugin will only extract entries from KeePass when the present URL matches the database value, therefore the chance of being fooled by a phishing site is reduced because the user would notice that the password fields are not being populated [7].

There is one argument against using a password manager some risks of using a password manager

¹Password managers may also combine the password with other forms of authentication such as a key file

²If you use a clipboard manager (I use Ditto), an exclusion should be added to prevent capturing KeePass entries, which would otherwise get stored as plaintext

³The server is bound exclusively to the localhost / loopback interface, and a preshared AES key stored in the database is used to encrypt the traffic between the two applications.

References

- [1] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y.-Y. Choong, K. K. Greene, and M. F. Theofanos, “Digital identity guidelines: Authentication and lifecycle management,” Tech. Rep., Jun. 2017, Section 5.1.1.2, Appendix A. DOI: 10.6028/nist.sp.800-63b. [Online]. Available: <https://doi.org/10.6028/nist.sp.800-63b>.
- [2] *Password recommendations*. [Online]. Available: <https://security.web.cern.ch/security/recommendations/en/passwords.shtml>.
- [3] P.-H. Kamp, “LinkedIn password leak: Salt their hide,” *Queue*, vol. 10, no. 6, p. 20, Jun. 2012. DOI: 10.1145/2246036.2254400. [Online]. Available: <https://doi.org/10.1145/2246036.2254400>.
- [4] C. Cimpanu, *Github accidentally recorded some plaintext passwords in its internal logs*, May 2018. [Online]. Available: <https://www.bleepingcomputer.com/news/security/github-accidentally-recorded-some-plaintext-passwords-in-its-internal-logs/>.
- [5] *Keeping your account secure*, May 2018. [Online]. Available: https://blog.twitter.com/official/en_us/topics/company/2018/keeping-your-account-secure.html.
- [6] *Experian reveals the five key factors that make people & businesses more vulnerable to cyber fraud*, Dec. 2016. [Online]. Available: <http://www.experian.com/blogs/news/2016/05/19/experian-reveals-five-key-factors-make-people-businesses-vulnerable-cyber-fraud/>.
- [7] *What does the ncsc think of password managers?* Jan. 2017. [Online]. Available: <https://www.ncsc.gov.uk/blog-post/what-does-ncsc-think-password-managers>.
- [8] D. Reichl, *Keepass password safe*. [Online]. Available: <https://keepass.info/>.

Using LDAPS for centralised authentication

The design o