# 

# Prototype Extension Criteria

The password manager prototype created by our group needs to fulfill the following criteria:

1. The extension should be built for Google Chrome. This came from an informal review of the extension ecosystem for the major web browsers (Firefox, Chrome, Edge, Safari). In addition to being the most widely used browser[[1]](#footnote-0), Chrome has, in our view, the following advantageous features:
   1. More extensive documentation, with detailed implementation of interaction with most elements of the browser
   2. More versatile JavaScript API, with support for browser bar addons as well as page content focus
   3. More simplistic addition of extensions, with an easy local upload, and non-automated removal of developer plugins
   4. Similar file structure to Firefox extensions (allowing for an easy port), while being more descriptive, and having explicit permissions

While deploying the extension to all of the major platforms would be an important usability criteria in general, only a single browser is initially necessary for our prototyping.

1. The extension should allow the user to set a master password for the saved passwords because it mitigates attackers who seek to steal unguarded information. We know this from the work by Zhao and Yue [1], who found that, in 2014, Google Chrome, Internet Explorer, and Safari did not allow users to set a master password for their browser-based password managers. Without setting a master password, the collection of saved passwords can be vulnerable to malware executing on the system.
2. The master password scheme should be 2words16, as suggested by Shay et al [2]. This scheme requires the password to contain:
   1. At least 16 characters
   2. Two alpha-numeric words
   3. At least one non alpha-numeric character separating each word

It was chosen because the researchers found it to be easily remembered, relative to the other tested schemes, while providing strong protection against password attacks. The downside is that password creation can be difficult, but as this password will be the key to all other passwords, a convenience trade-off is necessary to ensure security.

1. The extension should allow the user to generate randomized, secure, passwords. At the user’s discretion, this should include:
   1. Upper and lower-case letters
   2. Digits
   3. Special characters
   4. Using numbers and special characters in place of letters

All of these points are from Riley and Chaparro [3]. In doing so, the generated passwords will be secure and difficult to crack through dictionary and brute-force attacks.

1. The extension should warn the user if they try to use a password that is insecure, or reused by the user. This might be accomplished through a dictionary of commonly cracked passwords. These collections are available online, especially offered alongside paid password crackers such as John the Ripper[[2]](#footnote-1). This is mentioned as a feature of some managers by Gallagher [4], and has the potential to thwart a data breach before it happens by advising against poorly chosen passwords.
2. Notwithstanding the previous two points, the extension should allow the user to easily dismiss the password generation, and ignore insecure warnings. The reason for this is flexibility when dealing with variable password requirements that many websites specify, as discussed by Komanduri et al [5]. In this way, the user is not restricted from using certain websites, regardless of their potentially inferior scheme.
3. The extension should allow for the user to easily update their existing passwords. This is especially important in the case of a data breach, but also relevant in day-to-day usage. For example, many organizations enforce an expiration date of user passwords, as noted by Farcasin and Chan-tin [6]. Keeping all passwords up-to-date is essential for continued use of the extension.

By building a prototype that adheres to these guidelines, we believe that the extension will be highly usable and exhibit the expected functionality of password managers. While there are other ways to extend the prototype in terms of security (e.g., two-factor authentication, biometric authentication), we consider the stated goals as realistic relative to the timeframe for phase 1.

## References

[1] Rui Zhao and Chuan Yue. 2014. Toward a secure and usable cloud-based password manager for web browsers. *Computers & Security* 46 (2014), 32–47.

[2] Richard Shay, Saranga Komanduri, Adam L. Durity, Phillip (Seyoung) Huh, Michelle L. Mazurek, Sean M. Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2014. Can Long Passwords Be Secure and Usable?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’14)*. ACM, New York, NY, USA, 2927–2936. https://doi.org/10.1145/2556288.2557377

[3] Shannon Riley and Barbara S. Chaparro. 2006. User Password Generation Practices and Strong Password Guideline Compliance. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50. SAGE Publications Sage CA: Los Angeles, CA, 1812–1816.

[4] Elizabeth A. Gallagher. 2019. Choosing the Right Password Manager. *Serials Review* 45, 1-2 (2019), 84–87. https://doi.org/10.1080/00987913.2019.1611310

[5] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. 2011. Of Passwords and People: Measuring the Effect of Password-composition Policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’11)*. ACM, New York, NY, USA, 2595–2604. https://doi . org/10 . 1145/1978942 . 1979321

[6] Michael Farcasin and Eric Chan-tin. 2015. Why We Hate IT: Two Surveys on Pre-generated and Expiring Passwords in an Academic Setting. *Sec. and Commun. Netw*. 8, 13 (Sept. 2015), 2361–2373. https://doi.org/10.1002/sec.1184

1. https://www.w3schools.com/browsers/default.asp [↑](#footnote-ref-0)
2. https://www.openwall.com/wordlists/ [↑](#footnote-ref-1)