

Password Reuse: The Low Hanging Fruit of Password Protection

Choong-Won Richard Kim

December 14, 2016

COMP116 Security Final Paper

Abstract

The world of computer security is a world of juggling high-stake tradeoffs. Small companies barely have enough resources to duct tape their product together, and larger companies develop more and more avenues for attack as they hire and build out their stack. As more and more priceless data is stored in server racks, attackers have never been more incentivized to find and exploit security vulnerabilities. It's now up to the consumers to assume the worst and enforce best practices amongst their own online behavior. However, as many entrepreneurs know, asking users to change their behavior without real, tangible benefit is virtually impossible. Thus a balance must be found between maximizing security and minimizing change in user behavior. The area in which this is arguably most applicable is password reuse. This paper will explore different tactics and tools available to discourage password reuse and measure them according to security benefit and user burden.

Introduction and To The Community

Password reuse is the low hanging fruit of password protection for four primary reasons. First, It has a huge potential to do catastrophic damage on someone's life. Not only do most people have access to critical banking processes online, but more and more people are depending on software services to make a living. Furthermore, the number of services people sign up for is ever increasing. This is a critically important point to emphasize the importance of eradicating password reuse. Every online account a password-reusing user makes creates a new avenue of attack and increases the value of the network that the password unlocks. Here, "network" is used to describe the cluster of websites the user has logged into, where nodes are websites and edges are reused passwords. In this way, password reuse creates a problem that scales non-linearly.

Secondly, password reuse is overwhelmingly prevalent on the internet. Due to the secret nature of passwords, it's hard to know exactly how many passwords are reused online. However, estimates range from 50%¹ to 70%². According to passwordboss.com, 59% of their users use a single password for all of their internet services³. Given that PasswordBoss is a tool catered towards people concerned with security, there's a good chance the general public is much worse than this. This prevalence makes fixing password reuse a low-hanging fruit because several **billions** of people will benefit from it.

¹ <https://www.entrepreneur.com/article/246902>

² <http://passwordresearch.com/stats/statindex.html>

³ <https://www.passwordboss.com/>

Third, password reuse is actually relatively easy to fix. In fact, it's a virtually free way to defend against one of the most common forms of attacks. Compared to building a secure server-side infrastructure, asking someone to not use the same password is free, instant, and probably a more effective defence mechanism. There are also a number of available resources that make it easier to use unique passwords than to reuse them. This paper will discuss these tools and compare how they affect the user's overall security and behavior.

Finally, preventing password reuse acts as a kind of vaccine for the rest of the online community. Once one database is leaked, employee passwords can be found to access another company's database. For example, if a Google employee's password was found in the Dropbox leak, it wouldn't matter how strong Google's security infrastructure was, the attacker would be able to gain access to confidential information from Google's databases (provided that no other security measures such as two-factor existed). In fact, many of the recent attacks such as Dropbox⁴, Github⁵, and Carbonite⁶ were due to employee credentials being leaked elsewhere.

Thus, password reuse is one of the low hanging fruits of computer security with respect to cost, impact, and prevalence.

⁴ <https://techcrunch.com/2016/08/30/dropbox-employees-password-reuse-led-to-theft-of-60m-user-credentials/>

⁵ <http://thehackernews.com/2016/06/github-password-hack.html>

⁶ <https://www.databreaches.net/carbonite-forces-password-reset-after-password-reuse-attack/>

Defence Options

This paper is going to discuss 3 options for battling password reuse. These options are: memorizing multiple passwords, password managers, and bookmarklet password generators. There are a number of other alternatives, but these four are the most interesting at the time of writing. Each of these options were also suggestions to a post made in an online “hacker” group. Each option will also be scored as follows:

- User behavior (1-5): 5 = doesn't change user behavior at all, 1 = user required to memorize random strings for each website
- Defence Potential (1-5): 5 = even if one website leaked plaintext passwords, no other accounts would be threatened. 1 = all passwords are “password”

Option 1: Manually Memorizing Multiple Passwords

This is simply memorizing a handful of passwords and cycling between them depending on the various attributes of the website. This method is commonly used with two passwords. One is convenient and short, the other is long and complex. The longer password is used for important, high-security websites that contain valuable information. This might include banking, investing, insurance, and employee credentials. The shorter password is used for low-value websites such as Pinterest or reddit. The idea is that the user maximizes convenience, but still protects themselves from a good majority of attacks by splitting the password network in half. One half of the network is high value, but each node in the network has a high level of security,

keeping the overall network strong. The other half of the network is vulnerable to attack, but no node contains anything of value.

The password networks can be further divided with more password combinations. For example, one hyper-secure password can protect critical banking information, while a medium password can protect online-shopping websites, and a low-security password can be used for unimportant websites.

User Behavior: 4 / 5. This method is highly convenient, doesn't require additional software, and requires next to no change in behavior. All a user would have to do to get started would be to change their banking password to something super secure, and continue to use that password on other critical websites.

Defence Potential: 2 / 5. This method essentially concedes that many of the accounts the user creates may get broken into, and prioritizes the protection of a select few. While this is much better than nothing and will likely prevent the likelihood of critical damage by a significant margin, it still allows an attacker to access secure websites from leaks of insecure websites. Furthermore, users may not realize the importance of the data of some of their accounts. For example, a chat app might be categorized as a low-priority app, but the user may have sent confidential information to another user through the app.

Option 2: Password Managers

The primary password manager discussed will be 1password.com. Alternatives exist, but 1Password is widely regarded as the best password manager available. A

password manager stores all the passwords you create on various websites through the use of chrome extensions and native applications. Password managers can also inject user credentials into password fields through the use of shortcuts. Thus, a user can use a password manager as either a reference for all the passwords they created, or can use the password-generator feature offered by most products. This feature will randomly generate a secure password and remember it for future use.

The downside to using a product like this is that a master password is used to unlock the password manager. This effectively makes that password incredibly valuable. However, if this password were never used elsewhere and otherwise secure, this becomes a very effective method of preventing password reuse. 1Password also adds an extra layer of protection by giving their users an additional secret key that is immediately erased from the database. This secret key is used to decrypt the store of passwords for the purpose of syncing 1Password to another device. Since it is immediately removed from the database, the company effectively protects the user against employee errors and company security vulnerabilities.

User Behavior: 3 / 5. In the perfect scenario, this tool can actually be easier than reusing a single password everywhere because the native shortcut will inject the correct passwords into the right fields for the user. However, there are a few edge cases that reduce the chances of changing user behavior. First is price. 1Password is \$70 for the application and \$2/month for the web service. Second is the account secret. Since the company removes this from their database, there is absolutely no way to recover this secret if it is lost. If the user depended on 1Password for randomly generated

passwords, losing this secret would mean losing every single account they've ever made. Finally, without a device with 1Password on it, all the passwords are useless. This means that in the event the user doesn't have access to their phone or computer, they're completely incapable of accessing the network of passwords protected by 1Password.

Defence Potential: 4.5 / 5. The fact that 1Password protects the user from the company's security vulnerabilities is absolutely crucial. This means that anybody who attacks 1Password cannot access any of the information they acquired. Furthermore, 1Password encourages randomly generated passwords, which means that even leaked plain text passwords wouldn't have an effect on the overall network. The problem with 1Password is that 1 master password unlocks everything, and there is nothing forcing the user to use a unique password, so the same password might still be used everywhere.

Option 3: Bookmarklet Password Generators

SuperGenPass⁷ is the best example of a bookmarklet password generator. The idea is that the user provides a master password, and the bookmarklet salts the password with the website's URL. For example, if someone's password is "password", then a bookmarklet will hash "password.facebook.com" and paste it into the correct fields. By doing so, the user never actually knows what the password is, and each password is completely unique. Furthermore, no password is ever stored anywhere, so

⁷ <https://chriszarate.github.io/supergenpass/>

the tool can be completely open source (SuperGenPass is) and it wouldn't affect the security of any of its users. Furthermore, a shortcut can be used to trigger the pasting mechanism, which makes the tool even more convenient.

User Behavior: 5 / 5. The user doesn't ever have to remember more than 1 secure password (the main password). They are required to download a browser extension, but are not required to create an account and SuperGenPass is 100% free.

Defence Potential: 4.5 / 5. Since it is salted by the website url, a breach may still cause issues. If many people started using this tool, then it would be as if nobody was using the tool. An attacker would simply run all of the potential combinations through the same open-sourced hashing algorithm, salted with the website url. For example, if an attacker was doing a dictionary attack on facebook.com and knows that 50% of the users use SuperGenPass, then the attacker would just have to also run every dictionary combination through SuperGenPass with an appended ".facebook". However, for all intents and purposes (i.e. living in a realistic world where <1% of the population will use a tool like this), this is a perfect solution to resolve password reuse.

Conclusions

Password reuse is an unbelievably stupid problem, and yet it is likely the result of billions of dollars in damage and hundreds of millions of user credentials leaked online. However, combatting password reuse is mostly a matter of informing the public of the available options. Fixing this critical problem can be easy and hugely beneficial. Thus, to conquer password reuse is to capture the low hanging fruit of password protection.

References

Authentication Statistic Index. (2016). Retrieved December 14, 2016, from

<http://passwordresearch.com/stats/statindex.html>

Conger, K., & Lynley, M. (2016, August 30). Dropbox employee's password reuse led to theft of 60M user credentials. Retrieved December 14, 2016, from

<https://techcrunch.com/2016/08/30/dropbox-employees-password-reuse-led-to-theft-of-60m-user-credentials/>

D. (2016, June 21). Carbonite forces password reset after password reuse attack.

Retrieved December 14, 2016, from

<https://www.databreaches.net/carbonite-forces-password-reset-after-password-reuse-attack/>

Okyle, C. (2015, June 3). Password Statistics: The Bad, the Worse and the Ugly

(Infographic). Retrieved December 14, 2016, from

<https://www.entrepreneur.com/article/246902>

Password Manager | Password Boss Free Download. (n.d.). Retrieved December 14,

2016, from <https://www.passwordboss.com/>

SuperGenPass: A Free Bookmarklet Password Generator. (n.d.). Retrieved December

14, 2016, from <https://chriszarate.github.io/supergenpass/>

Khandelwal, S. (2016, June 16). Github accounts Hacked in 'Password reuse attack'

Retrieved December 14, 2016, from

<http://thehackernews.com/2016/06/github-password-hack.html>