People Are Idiots, and It's Our Fault

Adam Plumer

Medford, Massachusetts

Abstract

We are at an unprecedented period in history, where it is possible to have an identity and reach of influence that far surpasses any other point in time. People are members of vast social networks, storing personal information for easy access anywhere in the world. And yet, the security behind these services is beyond reproach. This is not to say that large technology corporations have not invested millions of dollars into shoring up their encryption and security procedures. The Achilles heel is the user and their propensity to pick shoddy passwords like "password", "123456," or worse. This practice is analogous to building someone a top-notch safe to store anything they can imagine, and then having the lock crackable by an AmEx card. We are at an age where there are many additional measures — two-factor authentication to name one — to help users defend themselves, but this does not address the ultimate problem: why do we let users pick passwords at all? In an age where password managers can generate and store passwords of incredible length and complexity, we allow users to select a password, so long as they meet certain criteria, e.g. they include at least one number. Why are we settling for the bare minimum of security? In this paper, I will suggest a method of implementing a user-free password solution, which for the first time eliminates the most potent security threat to the user: the user. This is not a be-all, end-all of password protection, and would not stop the possibility of data breaches, but would be a significant step forward in strengthening password security on a broad scale.

1. Introduction

1.1. Breaking In

In recent years, data breaches have become an almost ubiquitous occurrence, with major corporations seeing breaches multiple times a year. In the

past five years, there have been breaches reported by MySpace [1], Adobe [2], LinkedIn [3], Tumblr [4], Dropbox [5], Ashley Madison [6], and Yahoo [7]. The sum total number of passwords reported stolen ranks close to two billion. The sheer volume of these passwords alone is alarming, but the analysis of these passwords is equally alarming. According to a 2012 survey, 61% of users reuse passwords across multiple websites [8], with half of all users using five or fewer passwords in total. The same survey reveals that this isn't just a problem with the elderly, in fact 76% of 18 to 24 year olds reused passwords, which ranks highest of any age group. Possibly the most frustrating statistic is that out of all users surveyed, 89% of users felt that they were secure in using these passwords [11]. Clearly there is a conceptual gap among the internet-literate.

1.2. It's Not My Fault!

These users are happy to spread the blame, however, with around 60% of users believing that poor security is the fault of the large corporations [9]. Even in 2005, 70% of surveyed users believed that corporations weren't doing enough to protect their information [10]. Given the high volume of breaches, this view is understandable, but it's also because corporations are allowing these password practices in the first place. In a 2015 survey, it was found that 80% of retail websites did not meet minimum password creation standards, and almost a third of those sites accepted the most common weak passwords, like "password" [12]. This problem is well-documented, and is notated as Common Weakness Enumeration (CWE) 521 by MITRE [13]. The reasoning is simple: by enabling users to create weak passwords, they expose themselves to password cracking attempts like brute-force cracking. Brute force cracking is when someone tries every possible combination of characters to crack a password, and this process is usually sped up by using a "dictionary" of commonly used passwords. While users still have the option of following password-making best practices [14], this is not a requirement.

1.3. Alternatives: 2FA

Some in the security community believe that the advent of two-factor authentication, or the use of a separate device to confirm an identity when signing in to a service, can help stem this issue. While as early as 2004, 75% of users were willing to use a two-factor authentication service [15], as recently as 2013, there is still a subset who have concerns over privacy using these services [16]. They fear that giving up their phone number (or other

information) compromises their privacy when registering and using two-factor authentication. The percentage of people with these concerns rose from 2004 to 2013, from 8% to 27% of users. Additionally, as of 2013, almost 75% of users had never used two-factor authentication on a website [16].

1.4. Alternatives: Passing the Buck

There is also the alternative suggestion that instead of creating more layers for users, websites can simply latch on to other services for authentication. One popular method is authenticating against social network logins to gain access. A 2011 study claimed that 77% of internet users in the US are comfortable using such a method [17]. While this solution does increase the likelihood that users create stronger passwords — due to the fact that most major corporations and social networks require strong passwords by default — it does not fully address the problem with the sites that do not adopt this model for various reasons. It also opens up the possibility that if one of those tentpole services is inevitably compromised, e.g. Yahoo, access can then be granted to whatever services may authenticate with Yahoo.

1.5. Moving On

So who's responsible here? We have users who have a bad conception of how to create strong passwords, and corporations having a tough time keeping them secure, keeping users from making bad decisions, or implementing more advanced security techniques like two-factor authentication. Confounding the issue is that while users seem open to the idea of using more secure methods of authentication, corporations are slow on the uptake of these same methods, compromising users' security in the meantime.

2. To the Community

It is impossible to understate the importance of strong password security. It is the number one preventative measure to cracking attempts short of writing everything down on pieces of paper. However, the strong deterrent to adopting a strong password model is a shared impetus between corporations and users. Users are unable or unwilling to create and maintain a list of passwords conforming to the best-practices model for password creation. Corporations are unwilling, but certainly able, to create stronger restrictions on password creation for users. This may arise for a litany of reasons, ranging

from cost concerns to fear of user drop-off. However, these concerns mostly amount to laziness and lack of concern for the users they do have.

If there were a copy of the Ten Commandments that applied to the Internet, two of them would have to be (for website creators), "Thou shalt mandate strong passwords", and (for users), "Thou shalt create strong passwords, never use one more than once, and update them frequently." Unfortunately, we don't live in a world where these commandments exist, and the attitude of users makes it unlikely that they would adopt it by themselves. There is clearly a need for a more elegant solution in this mess, one that abstracts away any possible interference from users and corporations. The ideal solution to this problem thus must meet a series of criteria:

- Be easily adoptable as a standard in Internet use
- Encourage but not require the use of additional security measures
- Little to no involvement from the user
- Little to no added involvement from the corporations
- Be as generalizable as passwords, i.e. can be used seamlessly on mobile
- Ability to create a secure paradigm in line with password best practices
- Any password database breaches would be mostly useless

A good solution would still meet most of the above criteria, but it may be unrealistic to expect all of them to be implemented. I will suggest two approaches to this problem, weighing the benefits of each.

3. Action Items

The motivation for these suggestions is inspired by the age-old adage that no one should implement their own cryptographic algorithm. In the same vein, it is a strong corollary that users should not create their own passwords. The odds are against them in creating and maintaining strong passwords, so the best option is to simply remove users from this process. The two proposed options both take this philosophy to heart, first in the storage of these passwords, and then in the generation of these passwords. These solutions are focused on mitigating the threat that passwords could be

cracked remotely either by brute-force or by analyzing a database of cracked passwords. They are not focused on physical cracking of passwords or theft on the user's machine itself.

3.1. Option 1: Password Managers for All

What may be the simplest, and perhaps cheapest, option for users is the wide-scale adoption of password managers. Currently, only about 8% of Internet users surveyed in the US and UK were using a password management system [18]. And yet, about three-quarters of surveyed users in 2004 claimed that they would be in favor of using a singular authentication service [19]. There are a wide variety of password managers available on the market today, and yet most modern browsers only allow for limited functionality of in-house/first-party options. Allowing for wider adoption of quality and secure offerings, possibly bundled with the browsers themselves, would encourage users to more proactively use these services, especially given the convenience factor.

One of the criticisms of password managers is that they can be compromised by simply cracking the "master password" that is used to secure the manager. One possible solution to this attack would be to rotate the master password over a certain interval, i.e. six months. Since users are abstracted away from remembering 5-10 passwords, remembering one variable password should not be as much of a challenge, especially if they use it constantly.

The rationale for this method is simple and is best illustrated by the following example. Imagine that passwords are money. In many cases, this is an apt analogy since passwords are used to access banking websites. By hoarding the money, alone, there is little guarantee of security, and keeping track of all of the money over time becomes stressful and unruly. Additionally, someone may come around and find out where you keep your money, and steal it. Perhaps you always use the same hiding spot for all of your money. Suddenly, the idea of a central bank emerges, where the upkeep in keeping track of and securing the money is abstracted away to professionals. Sure, the money is all in one place and makes an enticing target for attacks, but the bank is far more capable of providing a defense, since again it is managed by professionals.

This example illustrates the attractiveness of integrating password managers more directly into browsers. However, another example is needed to motivate the rationale for making password manager use mandatory. In some states in the United States, like New York, in order to legally drive, you must

have valid car insurance. The rationale for this is simple: if you get in an accident, you're (mostly) protected from any fees associated with the damage from the accident. Using passwords on the Internet should be very similar: in order to use sites that require passwords, you should be forced to use a password manager to safeguard against future infractions.

So how can this be implemented on the client-side? Here are some suggestions:

- Anytime an `input=password` field is detected in an HTML form, the browser restricts entry to registered password managers only
- All passwords entered via Signup forms are automatically entered into the password manager — this functionality is already present in most password managers, e.g. 1password, but prompts the user before saving — only requiring the master password to encrypt them
- All passwords are randomly generated to the maximum security capability of the site in question
- If a site does not support minimum security standards, a severe warning is issued by the browser, possibly with blacklisting the site
- Password managers can attempt to refresh passwords over a certain interval if the site in question allows for it — this would again require the master password for encryption, but can be scheduled for when user is active

How can this be implemented on the server-side? Here are some suggestions:

- Create a framework to allow for automated password creation and refresh commensurate with client-side protocol
- Actually enforce minimum security requirements
- Require passwords be refreshed after a certain period

There are, of course, disadvantages to this system. If one type of password manager is cracked, then all users who rely on that manager are compromised, with potentially all of their passwords out in the open. One potential solution

to this would be to spread out passwords across multiple password protection systems, though this may not be ideal due to the necessity (or suggestion) that each manager have its own variable master password. This is akin to having multiple bank accounts to spread out liability in the eventuality that one bank unexpectedly closes, but may not be directly analogous to passwords, since you can use one singular identity at each bank.

3.2. Option 2: 2FA Takes Over

Using password managers for everything is certainly a step above the status quo, but having passwords still as a settable option, exposed to the user, still seems disconcerting. This option suggests marrying a cornerstone of most two-factor authentication systems with password managers. 2FA systems generally rely on what is known as a **Time-based One-time Password Algorithm**, where a secret key generated once and shared by the client and the server is used in conjunction with the current time (or time interval, e.g. 30 seconds) to generate the passcode. Since 2FA is by definition an augmented method for authentication, most passcodes are generally kept short and strictly numeric, e.g. *654980*. However, there is a great potential to expand this principle to encompass passwords in general. To explain this in more detail, here is an example:

1. Server generates the "secret" based on the username alone and shares it with the client
2. Possibility that server also generates a more static representation of the password for recovery purposes, stored separately on server
3. Second possibility that 2FA is also used in conjunction with this method
4. Client stores the secret in a password manager
5. At any point where authentication is required, the password manager sends the correct passphrase*
6. At a certain interval, the secret is refreshed between the server and the client

* the passphrase in this case is more complex than in 2FA, i.e. it can be longer and use more varied characters

The advantage of using this method is that the possibility of remote attacks are again reduced. It also takes the user out of the equation of entering a password from the get-go. Passwords are not revealed because

they don't have to be, and doing so is mostly useless since the codes would change frequently.

This does not mitigate the possibility of data breach from the server or the client, in which case the secrets are stolen, but the above possible implementations, e.g. also using 2FA, having a recovery password in the event of catastrophic breach, would reduce the possibility that stolen secrets would do that much damage.

The other concern is that this method is insufficient for use cases where users are not using their own devices or computers. This is a valid concern, and the following solution is suggested. The user in question would most likely have their smart device on their person. They could send an authorization request to their phone, where they would sign in (authenticate) and approve the remote entry request. This limits the disclosing of secrets on the remote machine, since all authentication is handled remotely. The concern is still valid that the machine could be corrupted and perform malicious acts in the user's name, but that issue is out of the scope of this paper.

3.3. Putting It All Together

It is the suggestion of this paper that these methods all be combined into one singular password management philosophy. Namely, that passwords are taken out of the responsibility of the user. For all sites that do not adopt standardized password policies, the password manager would be responsible for creating as secure a password as possible and prompting refreshes for them more strenuously. With the hope that most users begin to use password managers and thus adopt stronger, more variant passwords, the responsibility of corporations can be put into adopting smarter, more secure systems like 2FA or the above password-manager 2FA hybrid.

It is not the view of this paper that this is the perfect or ideal solution to the password problem. The claim is that these steps are viable in reducing the exposure of user credentials and accounts to malicious attacks. This is based on the fact that the status quo is such that most users have few, insecure passwords for multiple sites, which leads to the higher likelihood of data being easily compromised. The suggestions given here highlight the feasibility, fundability, and enforceability of strong security practices for all Internet users. The suggestions are feasible because password managers exist and can be easily procured, fundable because large corporations could invest in providing these managers for free instead of settling data breach lawsuits at enormous cost later [20]. Finally, these suggestions are enforceable if the

makers of the popular web browsers commit to providing and mandating the use of these tools over time. While this may not be realistically implementable within the next five years, it remains in strong consideration to be an effective standard to address the problem of data breaches and poor password habits by users.

4. Conclusion

Creating and maintaining secure passwords has been an issue in information security for decades. The problem stems from multiple pain points, involving both users and corporations. The users present a serious liability in password security, primarily because they predominantly choose few and unimaginative passwords for use on a broad range of websites. Reusing weak passwords is one method password crackers count on when they attempt to break into a user's account remotely. The corporations are not blameless either, as they have shown a repeated ineptitude at securing passwords correctly. Countless breaches over the course of one calendar year result in hundreds of thousands of passwords being exposed and put up for sale. Corporations also allow for users to create insecure passwords, and to not require them to refresh them over a certain interval.

The greatest issue, however, is the unwillingness to affect change. Corporations have the ability to adjust their password policies and to take database security more seriously. Users have the commensurate ability to adopt stronger passwords and to rotate them frequently. However, in acknowledgement of the unwillingness to perform these actions, this paper puts forth several suggestions for future password policy.

First, password managers should be ubiquitous in all modern web browsers. This would allow the user to be abstracted out of the password problem, leaving the responsibility to security professionals. This would also prove to be no burden on the corporations, while still providing enormous security gains to the end-user. This is because the password manager could create complex, unique passwords for every website, so that in the event of a breach, only one user account would be compromised. This is superior to the model of using one social network identifier as a login, because in the event that that login is compromised, all other accounts are compromised. The corollary from this point is that the master password for the password manager could similarly be compromised, but for the scope of this paper, the assumption is that the master password is stored locally, making it tougher to crack remotely.

To further expand on this suggestion, and to mitigate the possibility that passwords not refreshed frequently enough could also be cracked, it is suggested that the current method for 2FA, or the Time-based One-time password algorithm could be used to replace traditional passwords. Instead of the short, numeric-based passcodes currently generated for 2FA, these passwords could be complex and use multiple character sets.

These suggestions would address a demonstrable problem in computer security. They would provide only a few, addressable inconveniences to the user in terms of adoption of new technology. However, it's been shown that users are willing to accept using a singular mode of authentication for web services. These suggestions also allow for the highest compatibility with existing systems, since password managers work just as effectively on less secure sites as they do on more secure sites (in the sense that they can generate unique passwords for both).

Finally, these suggestions are not complete, and also not believed to be ideal, but they offer a realistic solution to addressing an endemic issue in information security.

5. Supporting Material

Supporting material is present in the `2fa-example` repository on my GitHub account, username `CaerusKaru`. The supporting material demonstrates an example of creating a time-based one-time password using Python and available libraries. The code is not production-ready but demonstrates the feasibility of implementing such a system.

6. Acknowledgements

Thanks to Prof. Ming Chow, Department of Computer Science, Tufts University, for his mentorship and feedback on this paper.

7. References

[1] Perez, S. (2016). Recently confirmed Myspace hack could be the largest yet — TechCrunch. Retrieved December 14, 2016, from https://techcrunch.com/2016/05/31/recently-confirmed-myspace-hack-could-be-the-largest-yet/

[2] Krebs, B. (2013). Adobe Breach Impacted At Least 38 Million Users ? Krebs on Security. Retrieved December 14, 2016, from https://krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/

[3] Hackett, R. (2016). LinkedIn Data Breach: 117 Million Emails and Passwords Leaked. Retrieved December 14, 2016, from http://fortune.com/2016/05/18/linkedin-data-breach-email-password/

[4] FRANCESCHI-BICCHIERAI, L. (2016). Hackers Stole 65 Million Passwords From Tumblr, New Analysis Reveals — Motherboard. Retrieved December 14, 2016, from https://motherboard.vice.com/read/hackers-stole-68-million-passwords-from-tumblr-new-analysis-reveals

[5] Hunt, T. (2016). Troy Hunt: The Dropbox hack is real. Retrieved December 14, 2016, from https://www.troyhunt.com/the-dropbox-hack-is-real/

[6] Hackett, R. (2015). Ashley Madison Hack: Everything to Know. Retrieved December 14, 2016, from http://fortune.com/2015/08/26/ashley-madison-hack/

[7] Goel, V. (2016). Yahoo Says 1 Billion User Accounts Were Hacked - The New York Times. Retrieved December 14, 2016, from http://www.nytimes.com/2016/12/14/technology/yahoo-hack.html

[8] CONSUMER SURVEY: PASSWORD HABITS www.csid.com. (2012).

[9] DESPITE LACK OF TRUST, INTERNET USERS? SECURITY BEHAVIORS FAR FROM IDEAL, ROBOFORM STUDY FINDS. (2012).

[10] RSA Security Consumer Study Reveals Major Concerns Over Online Security and Identity Protection. (2005). Retrieved December 14, 2016, from http://www.prnewswire.com/news-releases/rsa-security-consumer-study-reveals-major-concerns-over-online-security-and-identity-protection-54044377.html

[11] Larsen, T., Norsis, O. C. (n.d.). Password^12 Password survey in Norway.

[12] Dashlane Password Manager - Ecommerce Security Roundup. (2015). Retrieved December 14, 2016, from https://www.dashlane.com/internet-security-roundup/ecommerce-2015

[13] CWE - CWE-521: Weak Password Requirements (2.9). (2014). Retrieved December 14, 2016, from https://cwe.mitre.org/data/definitions/521.html

[14] Krebs, B. (2016). Password Do?s and Don?ts ? Krebs on Security. Retrieved December 14, 2016, from https://krebsonsecurity.com/password-dos-and-donts/

[15] Survey Finds Identity Theft Negatively Impacting Consumer Use of the Internet. (2004). Retrieved December 14, 2016, from http://www.prnewswire.com/news-releases/survey-finds-identity-theft-negatively-impacting-consumer-use-of-the-internet-74390692.html

[16] Impermium Study Unearths Consumer Attitudes Toward Internet Security — Business Wire. (2013). Retrieved December 14, 2016, from http://www.businesswire.com/news/home/20130627005473/en/Impermium-Study-Unearths-Consumer-Attitudes-Internet-Security

[17] Goings, K., Abel, P. (2011). Consumer Perceptions of Online Registration and Social Login.

[18] US and UK Password Practices Leave Users Vulnerable, According to Survey Sponsored by Siber Systems. (2015).

[19] Survey Finds Identity Theft Negatively Impacting Consumer Use of the Internet. (2004). Retrieved December 14, 2016, from http://www.prnewswire.com/news-releases/survey-finds-identity-theft-negatively-impacting-consumer-use-of-the-internet-74390692.html

[20] Reuters. (2016). Ashley Madison Owner Reaches $1.6 Million Settlement - The New York Times. Retrieved December 14, 2016, from http://www.nytimes.com/2016/12/14/business/ashley-madison-settlement.html