Spencer Baer, Isaiah Freeman, Kyle Galloway, Brad Wilson

CS 491 – Software Security

Dr. Travis Atkison

2016-04-21

Passwords and Multi-Factor Authentication

Originally, countersigns were reserved for military use, or use amongst members of secret societies. The precursor to the modern day password, these countersigns were usually made up of secret symbols or gestures, arbitrary question and answer pairs, or certain pass-phrases given out to members of the group. Their usage was generally reserved for identifying messages from potentially unknown correspondents, and to identify members when more direct means were inadequate or unavailable [1]. This system had several obvious flaws, such as security through obscurity, but generally served it's purpose well.

With the advent of computing hubs and terminals in 1960, researchers at the Massachusetts Institute of Technology needed a way to differentiate users logging in from multiple terminals to access their private files. A text-based pass-code known only to the user seemed like the obvious choice. Other systems could have been used. One method suggested using information only the user would know, such as their mother's maiden name. However, that form of identification would require the storage of several pieces of information exclusively used to identify the user. A small string of text was far easier to store in the limited resources of early computers, and thus the password was born. Unfortunately, the password was not an impenetrable beacon standing steadfast against the darkness. A scant two years after implementing the system, the world's first password breach took place. A researcher at MIT, not receiving enough hours to run his computations to completion, put in a request for a print-off of the passwords file. His request was granted. He shared the blame by passing around the list to his colleagues. Eventually, they were all caught when one of the researchers left mischievous messages in the directors account[2].

More problems began to arise as computing and networking grew in popularity and scale. An idea meant to separate the users' document pool in a research lab, quickly grew to become the defacto means of protecting important data across nearly every computing interface. With such a large growth in use case types, as well as quantity, the one simple password became many complex passwords, used by many forgetful humans. Why such a simple mechanism stands on the front lines of modern computing security can be traced back to three main factors.

First, while computers did indeed grow very quickly, there were no sharp jumps. There was never a time in which it suddenly became clear that computers had evolved completely from their original use cases. For this reason, no one realized that it was time to start removing/changing some of the early features implemented to be just good enough to work. By the time people started questioning security practices, passwords had become too ingrained in computing systems to be switched with an alternative. Passwords, as a whole, are a fairly secure mechanism when implemented correctly. Before the advent of the Internet, passwords were fairly sufficient at keeping people out. Generally, if someone wanted to try figuring out your password, they'd have to obtain physical access to your machine. With the Internet, most password protected data can be accessed via network from anywhere in the world, allowing attackers to throw dictionaries at your password from the comfort of their home [3].

Second, even if there had been a period of time where engineers realized more secure mechanisms were needed, no one knew of a clear alternative to replace passwords. Nothing matches their ease of use, portability, and simplicity in all, or even most, dimensions. While almost all other proposed solutions offer better security, until forced by some authority figure, users are extremely unlikely to switch to the new solution unless it is at least as good as passwords on the three previously mentioned criteria [4].

Lastly, the issue is not that passwords are fundamentally flawed, the issue lies with the flaws in the system in which we’re using passwords (i.e. multiple website logins, computer logins, phone passwords, pin codes, etc.). The necessary security is fundamentally opposed with passwords. Of ease of use, portability, or simplicity, you can have any two, but not all three. This is due to the limits of human beings: we only have a limited capacity to remember passwords, with the number of passwords we can remember going down as the security and complexity of them goes up. Remembering 2 or 3 complex passwords was fine for the first few years of the Internet, when security was not as important, attacks were far less prevalent, and most importantly, their simply were not that many services that required passwords. Now, the average user has over 90 on-line accounts, each with its own password to remember. Having nice, complex passwords for all of these sites is simply a task too complex for a human beings to manage [5]. Again, this was something that no one saw coming. It arose dynamically out of the huge explosion of the web in the 90's. By the time everyone noticed how difficult it was to keep track of so many complex passwords, it was too late to institute any major changes.

This problem has been one of the most studied problems in computer security, and many solutions have been proposed. Some merely augment passwords, some change the infrastructure significantly, and many do away with passwords altogether. Of these, only the solutions that augment passwords, have seen any kind of mainstream adoption: password managers and two-factor authentication. The first we will discuss later, since its use is relatively low among the mainstream users. The second, however, has become increasingly popular over the last several years as a way to increase security for sites storing sensitive data, such as banks or financial institutions, and sites that could damage a users reputation. Specifically, it adds an additional security layer in the event of a compromised password. An attack would need both the user's password, and the physical authenticator belonging to the user to access the account. This is far from ideal, but much better than relying solely on a password [6].

As previously stated, passwords themselves are relatively secure. When we talk about good passwords, we're usually referring to its level of entropy, or its difficulty to guess in a timely manner given the current technology. Using this metric, a password like 123456abcdef is fairly insecure, while a perfectly random set of eight characters is sufficient [7]. Bumping it up to ten makes it near impossible to crack. A perfectly random 12-character password would take around a one-thousand years to crack with normal hardware; however, a sustained effort by hardware capable of 1,000,000,000,000 guesses per second would take somewhere on the order of 20 days, assuming a pool of seventy-five characters to choose from. Fortunately, the hardware to guess 1,000,000,000,000 passwords per second is extremely expensive, and usually is only brought to bare against extremely high-valuable targets.

The problem lies in the fact that humans do not choose perfectly random passwords. People tend to choose names and dates that are significant to them, they almost always base their password off a dictionary word, they choose easily guessable runs of numbers, the special characters they use are heavily weighted towards a few common ones, as well as a host of other easily predictable behaviors. All of these things add up to make passwords far less secure than their theoretical performance [8]. It brings down the difficulty from one-thousand years for a single password to a few hours for several thousand passwords. That data was not based on an experimental list of passwords, but rather on a list of real leaked passwords that were ran on a real production system. The most popular passwords tended to be things like "12345", "abcde", or "first name + last name + 1" [9]. On top of that, people constantly reuse the same password across multiple sites, increasing the chance that if a password to one site is compromised, then almost all of their accounts on other websites are compromised as well. The problem of people constantly reusing passwords is not caused by a deliberate attempt on their part to be less secure, or by a manifestation of their laziness. It is simply a survival tactic [5].

When people are asked to make passwords that must "be at least 8 characters are longer and contain upper and lower case letters, a number, and a special character" for over ninety sites, something has to give. Users will either make the passwords technically follow the rules, but in a very easily guessed manner, such as "first name + number(s) + !", or they'll make a complex password but reuse it among many of the sites they visit [5]. In fact, as evidence that users are at least somewhat aware of and care about security, many users will have a few passwords. One being for "don't care" sites, as in sites that would cause little to no damage in the case a security breach, another password for more important sites, like social media, and finally a password for financial institutions that has the highest strength [16]. This shows that users do care about security on their sites, but by human nature, are incapable of memorizing the long strings of characters required for adequate security on each site they visit [6]. As a consequence, to help themselves recall the secure passwords, people will write their passwords down.

Despite the security risk inherent in leaving your password lying around for anyone to see, users do tend to have a couple of reasons for the behavior. As mentioned earlier, people simply care less if some sites are breached than other sites. As long as the account is recoverable eventually, little harm comes from the breach. If the password keeps the breach rates acceptably low, then the user is satisfied. The second reason stems from the role of passwords in today’s world. Many users are more concerned with keeping out attackers from the Internet, than people they physically interact with daily. With these reasons in mind, it could very well be an acceptable risk to keep most passwords written down, and keep only the high valuable ones memorized. Even with the increase of risk for local malicious or opportunistic attackers, such as janitorial staff, in many cases the risk is considered acceptable. Should the medium on which the passwords are written be carried on the user's person, then the acceptable local risk is mitigated even further. This very scenario has actually been advised by very high profile security researchers to help alleviate the problem described above of too many passwords; however, this is a more of a stopgap measure at best [16].

The problems that we’ve been discussing have only come about in modern computing for two reasons: the high volume of accounts users use that need separate passwords and the passwords needed for those accounts becoming more and more complex. The first problem is a recurring issue in security research. In a perfect world, reuse of the same password across multiple sites would not be nearly as high of a security risk. If every website visited used current state of the art encryption algorithms, insured all their equipment was correctly configured and updated, and followed current best practices in security, then reusing passwords would be very acceptable. In fact, using the same password, even after one of the sites its used on is breached, would be acceptable. By current standards, no plain-text, or even encrypted passwords should ever be stored; only their hashes. However, in the real world, most websites do not come anywhere close to storing passwords according to best security practices. Because of this, utilizing a password on more than one means that the security of your password, and therefore all of the sites you use it on, is only as strong as the security of the weakest site [16].

This brings up another issue; the inability to know which sites follow security best practices. Every time you reuse a password, there is a chance *this* will be the site that gets attacked and compromises your password. The issue is further compounded by the ability of attackers to crack passwords increasing every year. Computing power advances along while the ability of the human mind to recall complex passwords remains static. The more complex a password gets, the fewer we can handle [16]. In our current situation, passwords are growing in volume and complexity. And this has just been discussing the vulnerabilities inherent in passwords themselves. So far, we have overlooked things like the ability to reset passwords by knowing the answer to a few “secret” questions, such as mother’s maiden name, or the attacks that passwords are inherently weak too, such as replay attacks or man-in-the-middle attacks. Since something needs be done about the weaknesses of passwords as they currently exists, most sites have begun rolling out two-factor authentication. This appears to be the foreseeable future of sites containing sensitive data with a need for stronger security.

To understand what two-factor authentication(TFA) is, we must first understand the three generally recognized types of factors that can be used to uniquely authenticate you. The types of the methods or things used to identify you are as follows: something you are, used for things like fingerprinting and facial recognition, something you have, such as a cellphone or token authenticator, and something you know, generally passwords, security questions, or security pin numbers. TFA refers to using at least one factor from exactly two of these types. In most usages encountered today, these are passwords plus something you have, like a phone or a device to generate tokens, but in principle they could be something like iris recognition plus the your smart-watch. The important thing about TFA is that it leverages the strength of both of its methods to shore up each other's weaknesses. For example, a user-name, password, and security question are all something you know. Each of those things might be obtained by someone else. By incorporating a physical device you own into the security, an attack would be required to physically take your phone as well as the knowledge of your user-name, password, and security question. A hacker in on the other side of the world can easily steal the password of someone in the United States, but acquiring their mobile phone is something that would be much more difficult. Better yet, even if some malicious hacker did somehow acquire both the mobile phone and password, the user would near immediately be aware of the loss of the former, and could take steps to secure the now vulnerable accounts. This would occur whether the user was doing it specifically to make sure that their user accounts could not be breached or if they were doing it to simply set up a new mobile device. This also protects users from things like phishing and man-in-the-middle attacks, since a TFA ideally cannot be copied and whatever signal is sent usually is only valid for a short period of time. Basically, TFA that is perfectly implemented counters all the vulnerabilities of passwords except for the much rarer cases of when the attackers are agents who are nearby or somehow have the resources to acquire the physical device[12].

Obviously, no TFA implementation is perfect, and even disregarding that, TFAs have inherent downsides. They are much more easily lost than passwords, and unlike passwords, a user must make sure to carry the physical device with them lest they be locked out of their accounts. And there is another problem: what happens if/when something happens to the mobile phone? While it's random or unpredictable, almost everyone either knows someone, or has broken or lost their phone before [6]. By its very nature, the mobile phone TFA cannot be something that can be easily ported to anther phone without the working original at the very least. And since people would be completely unwilling to use a service that made it possible to lose access to their accounts, there must be some method to recover access should the mobile phone be damaged or lost [4]. This, however, introduces a dangerous flaw that can undermine the entire point of TFA. Should the only check to remove TFA be to send a verification email or to answer a few security questions, which many sites do, TFA becomes effectively useless. All an attacker has to do is answer those security questions to break into the account. This task is made even easier thanks to the rise of social media and knowledge of social engineering techniques. This is not even getting into the other vulnerabilities that can undermine TFA, such as incorrect or insecure implementation by the host website or service, or the ability for malware to infect the mobile phone. A compromised mobile device can intercept authentication messages sent to the device, or possibly allow the attack to recreate the authenticator to use at their leisure [6]. So while TFA can indeed improve user security, it is far from an ultimate fix. In its current form, it acts more like a Band-Aid for the problems with passwords. And to be clear, it is only a Band-Aid for some passwords.

As we have discussed previously, the main thing that either blocks the adoption of an otherwise great technology, or allows the adoption of an otherwise sub-par technology, is ease of use. Something that is very convenient to use will almost certainly be much more adopted than something that is more difficult to use, regardless of the relative performance they have on other metrics. And this is exactly the case with TFA. No matter how it is implemented, it makes the usually simple task of logging into a website or service more complicated. The user can no longer simply enter their password. Now they have to enter their password, grab their phone, unlock it, open an app, and transcribe the code therein onto the computer to login [4]. As a comparatively more complex process, it is only being rolled out and utilized on those sights which carry a higher risk should they be compromised, such as email accounts, bank accounts, and social media accounts. Even then, only a small percentage of users have utilized this feature, though there has been an increasing push to get users on these sites to move to utilizing two factor authentication. Usually this is brought about because it costs businesses, especially banks, money when their customers' accounts are compromised, which gives them more incentive to make user accounts harder to break into. One example is the on-line gaming service Steam, where users that do not have TFA activated face additional restrictions.

As we have mentioned, TFAs in their current form are nothing but a Band-Aid, and one for only a limited number of accounts at that. With the ever increasing number of on-line accounts a person uses, as well as the ever increasing difficulty a password requires to be considered secure, a new solution is not just needed, it is required. While the TFAs are, at current, a stopgap, the technology that underlies them may very well represent the future of passwords [4]. Mobile phone TFAs that we have discussed so far work as follows: sign on to an account with the user's name and password and enter a code that is generated on the user's phone [6]. But if your phone contains the keys to the kingdom, why not just enter your user-name into the site and then have your phone, either by communicating to the website through your computer or directly, authenticate you [4]. It itself is already protected by a pass-code, or even more convenient, a thumb-print, so it's a simple matter to authenticate yourself to your phone and let it handle everything else. This immediately solves the problem of having to remember dozens of complex passwords. With your phone handling the authentication, the token that is used to verify your identity can be rendered virtually uncrackable and, better yet, valid only for an extremely short period of time [5]. This protects from a host of vulnerabilities including replay attacks, and key-logging.

Of course, the system is not without its flaws. In fact, this magnifies some of the dangers of mobile phone TFA. At this point, your mobile phone is the lynch-pin to your entire digital life [4]. Should it ever be lost or stolen, you are in serious trouble. It would take a very significant amount of effort to recover the user's accounts. Worse yet, should the user's phone ever be actually compromised, the attacker now has access to every account you own, making impersonating you all the easier, and reclaiming your accounts all the harder. The solution to this problem then, is to reintroduce TFA. But instead of relying on your mobile phone as the TFA vector, or, indeed, anything that is visible to you, the TFA is your behavior. Specifically, researchers have found, that the way we behave is very telling [13]. Take Facebook for example. When logging in, the way a user's mouse moves across the screen, the speed at which they type their email address, the mistakes they make, their actions they tend to take after logging in, the people and types of post they spend the most time on, etc., are all ways to insure that a person is indeed who they say they are. All of this data can be collected during the first few interactions with the service, and then updated continually thereafter as a person slowly changes. This type of security, when using multiple types of a person's habits, is nearly impossible to fool without actually having access to the records stored and algorithms used. The shear amount of variabilities in the ways in which a person can perform several actions, even when allowing for the necessary tolerances, is too huge [13]. Currently, Google, and a few other websites, utilize mouse movements and other user characteristics to verify that a user is human, and therefore bypass the *captcha* checks [14]. Again, this system, too, has its flaws. Chief among them currently is that many of its flaws are unknown, due to the little research and actual testing it has received. One that is obvious is the difficulty many medium and smaller sized websites would have implementing this system, as well as the resources it would require to constantly monitor and assess hundreds of minute actions by all its users.

Behavioral biometrics represents an extremely rich field for authentication [15]. Amusingly enough, this is analogous to the situation in horror movies when characters claim that another just feels "off". While it's hard to name, humans keep tabs on the normal behaviors of those close to us, and if enough of those start being violated, we think something is wrong. Computers are just now gaining enough power to be able to emulate this.

It seems clear that, for most applications, the entire idea of remembering strings of characters as means of authentication will soon be dead. While people have been preaching the death of passwords for over a decade now, they must die soon, or the protections on our digital accounts might as well be minor annoyances to those who wish to gain entry [4]. It is both the blessing and the curse of computer science that everything advances at an ever increasing rate. What is state of the art today will be routine in a year and outdated in two. Passwords are a relic from the earliest days of our field, and while they were dragged along this far because of their simplicity, it is time for them to quietly fade into obscurity. Luckily we're finally hitting the point of being able to implement systems that mimic, in many ways, how humans identify one another [14]. Not only that, we're finally able to take advantage of the hyper-connectivity in society, and utilize the devices that are always with people as a proxy to their identity. While change will likely begin to happen even more quickly now that passwords are hitting a wall, and are being broken more and more easily, it will take time for whatever system we utilize in the future to come about. In the meantime, we can shore up the defenses of our most precious accounts by utilizing TFA. Of course, this is merely prognosticating for the future, and as history has shown us, nowhere is that more dangerous than with technology. We can be sure that the rule of passwords will end soon, but as to what will replace them, well, we may very well find ourselves surprised. After all, it’s always the things that no one suspected that change the world the most.

1. References

[1] “Countersign”. Merriam-Webster. Meriam-Webster. Web. 16 April 2016.

[2] McMillan, Robert. “The World’s First Computer Password? It Was Useless Too”. Wired. Condé Nast, 27 January 2012. Web. 16 April 2016.

[3] Hiscott, Rebecca. “The Evolution of the Password — And Why It's Still Far From Safe”. Mashable. Mashable Inc., 30 December 2013. Web. 16 April 2016.

[4] Bonneau, Joseph, et al. "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes." Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012. Web. 16 April 2016.

[5] Czeskis, Alexei, et al. "Strengthening user authentication through opportunistic cryptographic identity assertions." Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012. 16 April 2016.

[6] Dmitrienko, Alexandra, et al. "On the (in) security of mobile two-factor authentication." Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014. 365-383. 16 April 2016.

[7] Kelley, Patrick Gage, et al. "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms." Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012. 16 April 2016.

[8] “Unmasked: What 10 million passwords reveal about the people who choose them” WPEngine. WPEnine, Inc., 2015. Web. 16 April 2016.

[9] Goodin, Dan. “Anatomy of a hack: How crackers ransack passwords like ‘qeadzcwrsfxv1331’”. Ars Technica. Condé Nast, 27 May 2013.

[10] Duffy, Jill. “You Have Exactly Three Passwords, Don’t You?”. PCMag. Ziff Davis, 7 June 2011. Web. 16 April 2016.

[11] Florêncio, Dinei, Cormac Herley, and Paul C. Van Oorschot. "Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts." 23rd USENIX Security Symposium (USENIX Security 14). 2014. Web. 16 April 2016.

[12] Aloul, Fadi, Syed Zahidi, and Wassim El-Hajj. "Two factor authentication using mobile phones." (2009): 641-644. Web. 16 April 2016.

[13] Yampolskiy, Roman V., and Venu Govindaraju. "Behavioural biometrics: a survey and classification." International Journal of Biometrics 1.1 (2008): 81-113. Web. 16 April 2016.

[14] Greenberg, Andy. “Google Can Now Tell You’re Not a Robot With Just One Click”. Wired. Condé Nast, 3 December 2014. Web. 16 April 2016.

[15] Gamboa, Hugo, and Ana Fred. "A behavioral biometric system based on human-computer interaction." Defense and Security. International Society for Optics and Photonics, 2004. Web. 16 April 2016.