Zeropasswords

Patented

What's wrong with passwords?

Nikos Leoutsarakos

Tiny bio

Nikos has a Physics background and a M.Sc. in Computer science from McGill University in Montreal, Canada, where he lives with his wife and two children. He has been publishing and developing systems in the areas of cryptography, wireless/mobility and digital signatures since 1994.

Nikos can be contacted at NIKOS@ZEROPASSWORDS.COM

Prelude

You do not have to be a data scientist to read this paper. If you use a browser to visit your favorite Websites, meet your social media friends online, and check your email daily, this paper is for you.

Let's start by asking: "what happens after you enter a username and password to login to a Website?"

The login system of the Website receives the two strings you typed in and tries to match them with samples it has previously stored in a database. These samples (originals) in the database were provided by you when you registered the first time. Actually, sparing you the technical details, the login system first searches for the username you typed in and if found, it then fetches the corresponding password from the sample-passwords database and checks whether it matches the password you typed in. You might say, "What's wrong with that? How else will a Website know its registered users and confirm their identity? A Website needs to check whether your secret matches the secret it has for you in store". Well, I found 14 things wrong with that!

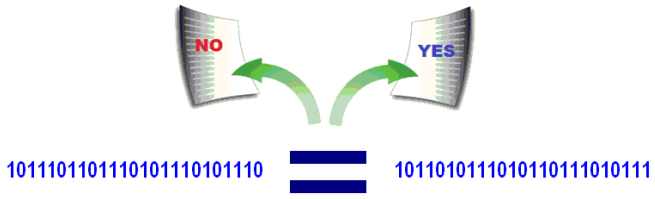

Figure 1

1. **Login systems use the simplistic comparison logical operator to log you in.** This operator takes two input bit-strings (Figure 1) and produces a single Boolean outcome (Yes: the two bit-strings are identical, or No: they are not). In order to *satisfy* this operator, login systems "reduce" users to bit-strings and "reduce" security to the outcome of bit-string comparisons! In practice, login systems turn the username and password you entered into bit-strings in order to compare and verify them against bit-strings they keep in databases. The sad fact is that even if we find a way to do away with usernames and passwords, today's login systems will find something else to compare against stored samples because that's what they do: *to control, they need to compare and verify bit-strings*. By now, 50 years later, we have

ample proof that security that stems from comparisons of shared secrets is miniscule to none.

2. **The Boolean outcome is the most dangerous inherent vulnerability of bit-string verification systems** (Figure 1). Hackers take advantage of this vulnerability and bypass the entire login process altogether compromising only the final Boolean outcome to their favor. Even the most elaborate and intelligent verification system with highly sophisticated input and secure communication channels can be rendered useless by simply compromising the outcome of its last comparison to always result in "Yes". Such a compromised login system would grant access to anyone!
3. **Stored samples are the most infamous inherent vulnerability.** Verification based login systems have no choice but to store and maintain sample passwords and other secrets in databases for future comparison against user input. Hackers have devised ways, and continue to come up with new ones, to gain access to stored secrets such as usernames and passwords. Attempts have been made to salt & hash stored passwords but hackers continue to smash & crab them because they can crack them at their leisure. Once hackers have in their possession a copy of a password database they have all the time in the world to crack the passwords it contains using several parallel ASICs and GPUs running software that implements efficient guessing techniques known as dictionary and rainbow table attacks. Using slow hashing and slow key derivation algorithms to slow hackers down, has resulted in slow Website response to legitimate users in pick traffic time intervals. Increase of the security of the perimeter of the hardware where password databases are stored with the erection of increasingly sophisticated firewalls to control remote access, have not helped either. The now uncontested successes of hackers (CNN and daily media) to get security credentials (passwords) from stored databases demonstrate that for as long as samples are stored somewhere, hackers will continue their efforts to obtain them. It's just good business.
4. **User input is the most exploited inherent vulnerability.** Exploited by hackers that is. Login systems expect "live" input from the user, for every access request. What makes usernames and passwords valuable, i.e. worth stealing and selling to underground world, is the existence of username and password edit boxes on login pages of Websites! Why? Because once known, anyone (users, hackers, even

systems) can enter these character strings, from any place on earth and attempt to login. Online imposters and intruders exist largely because username and password edit boxes exist, and because of the ubiquity and anonymity they offer. If we could eliminate edit boxes from login pages and the need for user input, then stolen secrets, such as usernames and passwords, would become worthless. As it turns out in practice, requesting input from the user does not cause one, but several vulnerabilities to the login process. We list these vulnerabilities below as separate flaws.

5. **Input is the security.** In the physical world, for centuries, the identity of a member is determined primarily by his physical presence, accompanied by his knowledge of the secret. When we copied this process to the digital world we ignored the fact that users lack physical presence and adopted only the part of the process that relies on the knowledge of the secret. We chose to cripple the login process and force it to rely exclusively on comparisons of secrets, and without realizing it, our choice made user input the *one and only* factor of access control. He who knows the password cannot be denied access! What a mistake: single factor, single point of failure.
6. **Input must be entered.** Verification based login systems require that a user provide live input for every access request. Hackers know that and they deploy malware to get input strings as they are being typed in, or "fed in" by password management systems. The proliferation of phishing, key-loggers, and other resident malware that exploit iframe vulnerabilities is the proof that if hackers cannot get your passwords from databases at servers, they will continue to try to get them from you, when you input them manually or automatically.
7. **Input must come to the samples.** As if user input was not vulnerable enough, typed in usernames and passwords need to travel to the server. And if the server does not have the samples needed for comparison stored locally, the samples must travel to the server as well. The majority of login systems, Web browsers, mobile apps and the like that accept user input, are required by login systems to transmit usernames and passwords to the server unencrypted! Login systems require unencrypted user input because they need to compare it against stored samples which have been created from unencrypted input provided by you the first time you registered. Can you see how many opportunities a hacker has to get your username and password with malware at the server, malware at the browser and network sniffers on the wire?

8. **Input can be evil.** Did you ever ask yourself: what else can I type in login edit boxes besides a username and password, or what else can I type on the URL line at the top of a browser screen besides a URL? Well, the hackers did ask and the answer they got became one of the most successful ways to hack in. What if you typed a database server command or a command that the Website server recognizes, they asked? To their surprise, or maybe they knew all along, many servers were totally unprepared for this type of input from the user and reacted wrongly, exposing to hackers not only the databases of samples and secrets but their entire filing system in some cases. When these attacks started happening, unsuspected users, companies and government organizations realized the hard way the dilemma that on one hand login systems today cannot function without user input, and one the other hand user input can be dangerous! Perhaps because we needed a fix to the problem quickly or perhaps because we have all accepted our "login fate", we went ahead and patched the servers with filters of user input and kept the edit boxes on login pages!
9. **Login systems are agnostic to the "nature" of the user,** incapable of distinguishing hardware devices or systems from humans. To verification-based login systems, everything and everyone is but a bit-string. While input from a user may come in many forms (PINs, passwords, passphrases) from varying locations (GPS), at different times, from varying computing devices, etc. they all need to be converted to a single bit-string because the comparison logical operator can only examine the equality of two bit-strings. This huge loss of information prevents login systems from recognizing hardware devices or systems from humans. Any human or any system capable of providing the correct bit-string is granted access! Hackers take advantage of this inherent vulnerability and have devised numerous software tools that can provide input automatically clogging servers with login requests. "CAPTCHA" and the policy to lock out users after a number of failed attempts to enter the correct input are two widely used patches that inhibit the functionality of password management systems and impact user convenience.
10. **Login systems do not generate nor maintain usable and binding logs of access transactions.** Yet another consequence of reducing registered users to bit-strings. Login systems do not keep access-transaction logs that bind a user, a device, a desktop, or a server to an access request, to a specific location or time, because

such information cannot be found in two short and meaningless alphanumeric character strings (username and password). Attempts by login systems to create transaction logs using metadata of login transactions, were soon abandoned because access logs of metadata could not stand legal scrutiny. For the same reason (lack of proof) hackers have never felt the need to take serious precautions to hide their identity when they type in stolen usernames and passwords!

11. **Login systems are "agnostic" to the data they protect.** Password-based login systems are digital locks on digital doors, and very much like physical locks, they are unaware of what's behind the door. A digital door may open to reveal vacation photos, or a digital door may open to reveal government classified documents. Verification-based login systems stand physically and logistically separate from the data they protect, and sadly in most implementations the protected data is stored unencrypted. Defenders of today's login systems point out correctly that introducing a digital door for every file, document, song, movie, or software application is a utopia, because it is impractical and humanly impossible to memorize hundreds if not thousands of usernames and passwords. So, we are stuck with agnostic digital doors, making hackers happy, encouraging them to continue to pick their digital locks because if they succeed, the payoff is more often than not, total access to a plethora of digital valuables. The embarrassing aspect of protecting digital doors with digital locks is when hackers do not even use them and come in through "the windows" using planted malware, email-injected malware, or inside jobs.
12. **Password fatigue.** Live input requires us to recall a username and password every time. On average, when we surf the Web, we are asked 10-40 times per day to recall and type in strings with 4-8 alphanumeric characters each from a group of 2-5 strings we have memorized. And if you are a professional working at a bank for example these statistics are much worse. Non-surprisingly, in order to mitigate our memory fatigue we all take dangerous shortcuts, such as using the same password with different services or servers, or choosing weak passwords that are easier to remember, or even worse writing passwords down somewhere. Without a doubt we have too many passwords today and we will need many more in the future as more and more services become available online. However, there is a limit to our capacity to remember and recall words, numbers, images, etc. and most of us are getting

dangerously close to it. Web and mobile password management systems are our current solution to password fatigue problem, but only a partial solution as username and password injection to all CAPTCHA, several banking, and many Government Websites cannot be automated. In addition, password management servers add yet another place on the net where our passwords are stored, to the delight of hackers. Password fatigue is real and a serious limitation of verification-based login systems.

13. **Server has both your data and your security data.** Websites, online servers and clouds today have the data protected by your username and password and at the same time they also have your username and password. This makes you irrelevant. What stops a server administrator from accessing your data? Mostly, ethics, honesty, personal integrity and in some cases the law, but clearly not the login system of the server. The irony of the situation is that the username-password pair they gave you and you try so hard to keep secret, and even harder to remember, is available to them (easier than it is available to hackers). Technologically speaking, banks, clouds, governments and online services do not ask you in order to execute a transaction or access your digital valuables because they do not need to ask you! Passwords cannot protect you from your own partner! (Website or cloud) who may, willingly or after it has been compromised, decides to access your money, files, photos or anything else you have uploaded, just because he can. Corporations know that passwords are powerless when it comes to inside jobs and use "legal language" to protect themselves and small print to transfer responsibility back onto you.
14. **7.3 billion hackers.** Passwords have the power to turn anyone of 7.3 billion people on earth and anyone of 11 billion computing devices online into a hacker. A hacker can be anywhere on earth, even outside the earth!, and be able to access your digital valuables once he knows (or have bought) your username and password. While ubiquitous access to our digital valuables is desirable, ubiquitous access control is dangerous. Access control must be local and always involve you. The fact that password-based verification systems allow cloaked access transactions from anywhere, anytime and by anyone, and without users/owners' consent, is what makes passwords valuable and worth stealing and selling.

Conclusion

Clearly, the problem is not whether passwords are short or long, whether they contain numbers, capital letters, or symbols. Password strength is not the problem. What's wrong with passwords is that they are shared secrets stored in two places. They are stored in your head and they are also stored in password databases at Websites. The same databases hackers take.

But that's not passwords' fault. The blame goes to verification-based access control systems which need to compare A against B in order to function.

For as long as login systems rely on verification-by-comparison to provide user authentication and access control, security concerns will remain. Comparisons are indeed the "plague of the Internet" that has infected our login and our security processes since the beginning (early 1960s) and have caused psychological, mental, physical and monetary damages to us, to machines, societies and institutions. Today, with security breaches on the news every so often, we know with certainty that processes that rely on verification of shared secrets are not secure because one of the partners in a group that shares a secret will leak it eventually.

One cannot but wonder how is it that we have come to rely on this simplistic comparison operator to provide security on the Internet? How did we ever come to trust trillions of dollars, our identities, our personal and medical info, our work and inventions and all our digital valuables to the outcome of a comparison of two bit-strings?

The question is not how to fix verification of shared secrets, but what will replace it?