DOI: 10.2507/34th.daaam.proceedings.xxx

ANALYSIS OF PASSWORD SECURITY POLICIES AND THEIR IMPLICATIONS ON REAL-LIFE SECURITY

Jasmin Redzepagic, Vedran Dakic, Josip Stanesic & Matej Basic

This Publication has to be referred as: Redzepagic, J[asmin]; Dakic, V[edran], Stanesic, J[osip] & Basic, M[atej] (2023). Title of Paper, Proceedings of the 34th DAAAM International Symposium, pp.xxxx-xxxx, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-xx-x, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/34th.daaam.proceedings.xxx

Abstract

This paper examines current password security guidelines and best practices, as well as their effectiveness in preventing unauthorized access and data breaches. The research also explores the impact of these policies on user behaviour and the potential trade-offs between security and user experience. The focus is especially on password length and mandatory change requirements since the best practices conflict across different standards. The findings of this study have important implications for organizations in developing and implementing password security policies that effectively balance security and usability. The research will compare different criteria on what defines a “good” password and by applying different password analysis methods establish what objectively should be used in real-world scenarios.

Keywords: password security; password complexity; end-user security; security guidelines; security policy.

1. Introduction

The interconnected world we operate in today is based on the idea that any user can be identified by a standardized process, and we call that process authentication. Sometimes we also use the term identification for the same process, but authentication is more precise. Authentication requires proof of a user's identity while identification just requires identifying the user. The main difference is that by providing a username, the user is identified. By successfully using a password or some other form of authentication, the user is authenticated. There are different ways the user can be authenticated. Usually, we are talking either about something that the user knows, for example, the password or a PIN, something that the user has, for example, a token or a list of one-time passwords, or something that the user is, for example, his voice print fingerprint, or his retina scan.

The biggest problem is that, in most user scenarios that we are encountering today, things that users have or features that users have are unable to be verified because the systems themselves are integrated into the cloud. The ability of the system to provide the user with a safe way of communicating the information needed to verify (for example) a token is either complicated or completely impossible. This applies to physical tokens, a solution to this exists in the form of digital tokens that provide one-time passwords, but they are just slowly gaining traction among users.

This leaves us with the password as the primary source of authentication for the user. Most of the systems today combine passwords with some other forms of identification usually a one-time password or some other generated identification token and we call that two-factor identification but the password as such remains one of the most important ways of identifying a user.

On the other hand, passwords are often considered one of the least secure ways of authenticating users so their complexity and resistance to cracking is a big problem. To make matters worse, password complexity is a heavily debated subject since a standard benchmark is not agreed on [1].

2. Passwords

From a security perspective, this means that creating and maintaining a healthy password policy is something that is one of the most important things when it comes to creating a secure environment. The problem that this poses for the administrators is that the passwords themselves are heavily dependent on the users and their ability to both create working passwords and safely keep them secure.

For a long time, the main strategy for creating a secure environment depended on a clear-cut password policy. This usually involves a couple of things: creating a set of requirements for the password length complexity history and age, creating an environment that will ensure that users are adhering to this policy, and educating users so that the policy is actively enforced.

Before defining the best practices for passwords, we need to define the metrics. When we are talking about passwords, we are usually talking about the following characteristics a password can have.

- Length: A minimum length requirement for passwords, e.g., at least 8 characters. This metric is usually taken as the most important one. A reason for this is that the most common way of guessing a user's password by brute forcing all the combinations is something that is done quickly if a password is short.
- Complexity: A requirement for a mix of distinctive character types, such as upper and lowercase letters, numbers, and special characters. This requirement for a password also comes from the method of brute forcing the passwords. Since we are dealing with permutations and combinations of different characters increasing the character set is by itself a terrific way to slow down the brute forcing process. Also, we must prohibit the user from using known words available in the wordlists.[3][4]
- History: Prohibiting the reuse of previously used passwords. Since users sometimes inadvertently give away or lose their old passwords, it is usually a clever idea to make sure that all the passwords are not able to be reused. Password reuse is extremely common among users, following figure shows percentages of reused passwords from an analysis done by Google in 2018 [2]. There is no reason to believe percentages have changed much since providers can not compare user passwords and warn them of the reuse.

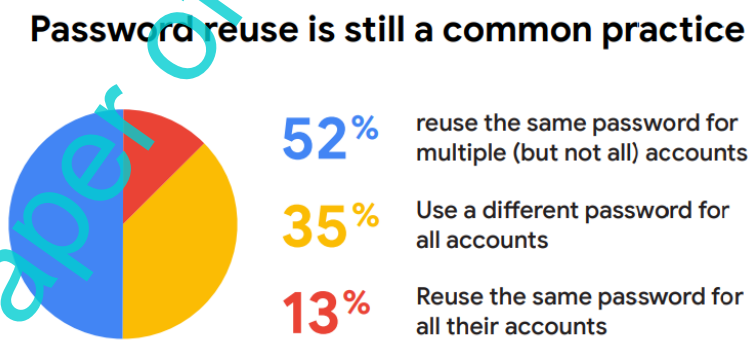

Fig. 1. Password reuse practices in US

- Age: Maximum password age, after which the password must be changed. This requirement is something that is a hot topic these days and will be mentioned a little bit later in this paper.
- Blacklist: A list of prohibited passwords, such as "password" or "123456". By using such a list one can make sure that the users avoid passwords that are easily guessed especially the ones that are available on the Internet on the so-called world lists.[3]
- Two-factor authentication: An additional step of verification, such as a code sent to a phone, in addition to a password. The reason why two-factor identification is important is that with the availability of mobile phones, it is relatively easy to make sure that the password is not the only thing standing between the adversary and the system he is trying to log in to.

Let's use these as they are the most used metrics when defining granular password policies.

2.1. Password retrieval methods

A major point about password storage and retrieval is to take two things into account. First is to never store the password in its plain text form. If a password is stored in plain text form, there is no point in trying to enforce any policy since the only thing that the adversary needs to do is to get to the password database and use the passwords as they are. So, we presume that we are dealing with passwords that have been stored using one of the secure hashing functions which means that they require some time to decipher either by using one of the methods that use brute force functions or by using so-called rainbow tables or words password guessing. [4]

We are not going to deal with password guessing used by dictionary-based attacks because we will presume that the password policy already has dictionary checks implemented so that the passwords that the users can create cannot be in the dictionary. Another thing we are going to disregard is using rainbow tables. Rainbow tables basically mean that for some of the encryption or hashing methods, somebody has already calculated all the possible hashes and that there is an available lookup table for an adversary to directly decipher the password for a given hash. Rainbow tables exist for some of the ciphers and for passwords of limited length. Both these problems can be solved by using a well-known good cipher and limiting the minimum password length.

This leaves us with brute-force attacks, which are a huge field of research. If we need to separate these attacks into different categories, we can talk about using either CPUs or GPUs to decipher passwords or we can deal with dedicated hardware designed specifically to decipher a particular cipher method. Either way, we are dealing with methods that require guessing passwords by creating a hash out of our presumed password and then comparing this hash to a hash of a password that we are trying to guess. This of course requires time and what we are trying to accomplish is to create passwords that take so much time to crack so that the process itself makes no sense to the adversary. [5]

For the last couple of decades, the usual thinking was that passwords should be as complicated as possible because what we are trying to do is to create a large set of characters and in that way a huge initial set of possible passwords so that the password guessing becomes more complex.

2.2. Password policies

So almost by definition, all the different password policies define passwords as having to have all the special characters, upper- and lower-case letters, numbers and all the other possible characters that exist. Usually, the password policy also defines at least 6 to 8 characters as a minimum length for a password, and the user is usually required to change this password regularly. Depending on the requirements usual time interval for this mandatory change is between 30 and 180 days.

Although minimum length and complexity requirements are defined in standards mandatory change is not defined. Mandatory change has been so deeply ingrained in the security policies of companies that even administrators believe that this in fact is part of some standard and enforce it.

Combining these different policy requirements into something that the user can or must do usually means that the user is going to create a short, memorable password. Passwords are then going to be changed slightly whenever mandatory change comes into effect but since the password is going to contain many different characters there is a huge change that the password itself is going to be written down somewhere. This means that users are complying with a policy but at the same time are creating the same password environment that the policy itself was designed to avoid.

2.3. Policy shifts

It is not unusual then that different companies including some of the largest ones like Microsoft have decided to change password policies in a way to stimulate users to create a password that is long and is not complicated in a way we may expect. Instead of relying on complexity, they require the users to create a password that has an exceptionally large length. This makes passwords much more complicated to brute force. But we have another problem.

Almost all the systems come with a predefined maximum length of the password which is now considered a security risk. When we consider the entropy of a password, more precisely the amount of time required to brute force a particular password based on its length it is it is much better to have a long password based on dictionary words than a short password based on special characters.

Also, since we are depending on the password to be extremely difficult to guess it is advisable to eliminate the need for the user to change his passwords from time to time since this almost guarantees that the user is going to create a lot of passwords that are going to be the same or similar.

New recommendations, therefore, are for the users to choose not a password as a word that is going to be used but switch to a "passphrase". A phrase means using multiple words that are on the one hand easy to remember for that user but are hard to brute force since the phrase can contain tens if not hundreds of characters. For example, for a given hashing function and eight 8-character passwords taken from a set of 64 different characters is going to be hundreds of orders of magnitude easier to guess than a simple phrase that uses only lowercase letters but is longer than 32 characters.

Number of characters in password	Numbers Only	Lowercase Letters	Lowercase Upper & Letters	Numbers, Upper & Lowercase Letters	Numbers, Upper & Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	4 Seconds
7	Instantly	Instantly	22 Seconds	42 Seconds	6 Minutes
8	Instantly	3 Seconds	19 Minutes	48 Minutes	7 Hours
9	Instantly	1 Minutes	11 Hours	2 Days	2 Weeks
10	Instantly	1 Hours	4 Weeks	6 Months	5 Years
11	Instantly	23 Hours	4 Years	38 Years	356 Years
12	25 Seconds	3 Weeks	289 Years	2K Years	30K Years
13	3 Minutes	11 Months	16K Years	91K Years	2M Years
14	36 Minutes	49 Years	827K Years	9M Years	187M Years
15	5 Hours	890 Years	47M Years	613M Years	14Bn Years
16	2 Days	23K Years	2Bn Years	26Bn Years	1Tn Years
17	3 Weeks	812K Years	539.72M Years	2Tn Years	95Tn Years
18	10 Months	22M Years	7.23Bn Years	96Tn Years	6Qn Years

Table 1. The time it takes to crack passwords using publicly available technology in 2023

This runs a little bit counterintuitive to what a typical administrator right now thinks about password policies since the requirement to have a large set of possible characters is something that a lot of administrators are used to. Having them recommend to their users that they should use a simple but long phrase instead of a shorter but more complicated password is something that a lot of administrators are not inclined to do.

The main reason for this is simply because they don't understand the statistical and mathematical consequences of trying to guess passwords that are longer in length. In Table 1 we are providing a rough idea of how the length of a password correlates to the time it takes to crack the password using publicly available technology [6]. Note that 8-character passwords are vulnerable to brute force attacks in a short time frame but are still recommended as viable password lengths. Also, some of their systems may not be capable of handling longer passwords since a lot of them have been designed with fixed maximum password length and this is something that requires not only a change of policy but also a change in software. That is a huge problem for a lot of legacy systems.

3. Findings, recommendations, and future research

There are more than a few takeaways from the methodology that we're using in our paper. Companies try to enforce password policies to their employees by issuing a blanket policy that "everyone must adhere to", without spending any time considering what that means in practice. Even forgetting the fact that this can lead to employee alienation, it's clear as day that people will always be people and either try to circumvent these rules or try to make their lives easier, in the form of writing down their passwords on post-it notes or in plain-text format in their mobile phones. We can avoid most of these problems by implementing a few recommendations, such as:

- Password length: Any maximum length requirement for the password should be completely avoided. The minimum length for the password should be kept to at least 16 if not more characters to stimulate the user to use a phrase not a single word or a combination of characters. This should be explained to the user.
- Mandatory password changes: Any requirement to periodically change your password should be avoided because it creates an artificial sense that the user is going to change the passwords and make them unique every time. This requirement means that users are just simply reusing old passwords and the risk of this password becoming available due to sheer negligence is almost guaranteed since the user is often going to write the password down.

- User training: Training is something that should be focused on trying to explain to the users that passwords themselves are not something that should be complicated but something that should be long and simple to remember. Users need to understand the reasoning behind this since they spent almost all their working lives being told otherwise.

A good example is to let the users choose a phrase from a book or part of the lyrics of a familiar song. This is going to make for a much better password than a nonsensical set of different characters and if this is presented to a user in such a way that the user understands the idea behind it users are going to be much more inclined to create a good phrase and keep it secret.

These requirements don't even take into consideration the fundamental truth of IT security – managing security in our computing environment is a collection of continuous activities [7]. By prescribing password policies, complexities, and requirements we are just fighting one of the security problems, not thinking about security. This does not only apply to regular users – it applies to everyone, including system engineers and application developers, as it can be difficult having them conform to different security challenges [8].

Our recommendations are based on the premise that the current technologies for guessing or cracking are going to grow linearly and that there are not going to be any breakthroughs in password-guessing technology. The advances in quantum computing unfortunately mean that some of the methods that we are talking about are going to completely change and current research does not point to a solution to this problem. And, even more to the point, companies nowadays face several types of cyber-crime-related attacks [9],[10]. We just do not want to give cyber-criminals more fuel to do even more damage.

At the same time, this is the single most important limitation in this paper. Focusing on passwords is important, but trying to predict new technologies that may make password guessing even more efficient is almost impossible. Further improvement in security should be first switching to passphrases but in the long run, only technologies that completely avoid using user-generated passwords will enable us to maintain secure authentication in the longer run.

Our future research could include topics like SSO (Single Sign-On) and security from the context perspective. For example, it's wrong to consider senior IT engineers who deal with cloud services every day and a novice IT administrator who only has limited experience with security policies and security as the same. Context plays a role, as does company culture, working environment, and systems/applications used. Single Sign-On systems fundamentally change the way that we approach security. Having in mind that external authentication systems like Azure Active Directory and a host of other SAML/OAuth-based systems have support for MFA (Multi-Factor Authentication), there will be fundamental changes in the way we approach user authentication in the future. Some of these systems will make authentication easier, but the question remains – at what cost?

4. Conclusion

Passwords are still the primary way of protecting most of the low and medium-security data and due to their importance for the average user, some changes must happen when it comes to the way passwords are generated and used. Password policies should be changed to reflect the realities of the brute force hash guessing methods. As methods keep getting quicker and more sophisticated passwords must be exchanged for passphrases, reflecting the need to get them as long as possible. Users need to be educated on why it is important to have long passwords and keep them safe. The requirement to change passwords regularly should be avoided, using long passphrases will mean that users will unavoidably write the phrases down and this will void all the ideas from this paper. As a result, we wholeheartedly agree that the strength and frequency of passwords (changes) are the wrong focus. We can be quite a bit more user-friendly with long passwords based on user's preferences, which will make other security issues smaller – for example, fewer passwords will leak via paper or digital trail.

Users' education should also involve incentives to use MFA and other authentication technologies that will enable them to move away from using only passwords or if possible from using passwords at all.

5. References

[1] Carnavalet, X. & Mannan, M. (2015). A Large-Scale Evaluation of High-Impact Password Strength Meters, ACM Transition on Information and System Security, Vol. 18, Issue 1, Article No. 1, ISSN 1094-9224, DOI: 10.1145/2739044

[2] https://services.google.com/fh/files/blogs/google_security_infographic.pdf, (2019). Google Online security survey, Accessed on: 20-09-2023

[3] Houshmand, S.; Aggarwal, S. & Flood, R. (2015). Next Gen PCFG Password Cracking, IEEE Transactions on Information Forensics and Security, Vol. 10, No. 8, pp. 1776-1791, ISSN 1556-6021, DOI: 10.1109/TIFS.2015.2428671

[4] Tatli, E. I. (2015). Cracking more password hashes with patterns, IEEE Transactions on Information Forensics and Security, Vol. 10, No. 8, pp. 1656-1665, ISSN 1556-6021, DOI: 10.1109/TIFS.2015.2422259

[5] Binnie, C. (2016). Password Cracking with Hashcat, In: Linux Server Security: Hack and Defend, Shimonski, R., Tulton, A.O. (Ed), 99-111, Wiley, 978-1-119-27765-1, ISBN: 978-1-119-28309-6, United States of America
[6] https://www.homesecurityheroes.com/ai-password-cracking/?password=ufskE34%24%240, (2023). An AI Just Cracked Your Password, Accessed on: 20-09-2023
[7] Dakic, V.; Jakobovic, K. & Zgrablic, L. (2022). Linux security in physical, virtual and cloud environments, Proceedings of the 33rd DAAAM International Symposium, ISSN 1726-9679, ISSN 1726-9679, ISBN 978-3-902734-36-5, DOI: 10.2507/33rd.daaam.proceedings.021
[8] Dakic, V.; Redzepagic, J. & Basic, M. (2022). CI/CD toolset security, Proceedings of the 33rd DAAAM International Symposium, ISSN 1726-9679, ISBN 978-3-902734-36-5, DOI: 10.2507/33rd.daaam.proceedings.022
[9] Moric, Z.; Redzepagic, J. & Gatti, F. (2021). Enterprise tools for data forensics, Proceedings of the 32nd DAAAM International Symposium, ISSN 1726-9679, ISBN 978-3-902734-36-5, DOI: 10.2507/32nd.daaam.proceedings.014
[10] Inglesant, P. & Sasse, M.A. (2010). The True Cost of Unusable Password Policies: Password Use in the Wild, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Computing Systems, pp. 383–392, ISBN: 978-1-60558-929-9, DOI: 10.1145/1753326.1753384