Assessing Password Protection Effectiveness Using Markov Processes*

Oleg V. Boychenko[1][0000-0003-3326-1015], Ilya V. Gavrikov[1][0000-0002-7047-9059]

[1] V. I. Vernadsky Crimean Federal University, Simferopol, Russia
bolek61@mail.ru

Abstract. Passwords are an integral part of modern digital life, but their effectiveness has been repeatedly challenged. This paper explores the role of password authentication in modern information systems, the changes it has experienced over time, as well as new alternative approaches to authentication and their interactions with classical password-based systems. Technical and organizational recommendations are formulated based on security research and modern trends in information technology, and an application of Markov processes to assessing password quality is presented, as a quantitative measure of password system security and effectiveness.

Keywords: Security, Authentication, Passwords, Markov Processes.

1 Introduction

Passwords have been a mainstay of information system security ever since the early days of computer use, as a natural security measure that is both easy to understand and to implement. Any system that requires shared use conceivably requires a mechanism to delineate responsibilities, limit access to resources and compartmentalize data between users. The first-ever computer system employing password-based authentication is considered to be the MIT Compatible Time-Sharing System (CTSS), which began service in 1963 [1].

However, advances in computing power and the increasing complexity of modern information systems have caused users, developers, and IT vendors to re-evaluate password authentication and password-based security as a whole. As passwords are simple to implement a measure that provides a sufficient level of security, they became a standard way of authenticating users in computer systems with a high level of usability for both the users and the implementing party. While past computer systems often only required authentication locally, within the scope of a single machine, the advent of the Internet has necessitated the use of passwords to authenticate thousands, even millions of different users.

However, with the wide availability of networked services come risks and dangers: unlike before, modern attackers need not always have physical access to a user's machine or even infect it with computer viruses or other malicious programs. The recent shift to cloud-based services and data storage means attackers can gain access to sensitive data as long as they can gain access to a user's account with a cloud service. Because passwords remain the first (and sometimes only) line of defense in modern authentication mechanisms, attackers can exploit weaknesses in IT systems and the human factor to gain illicit access to passwords and therefore sensitive data.

2 Technical Approaches to Password Authentication

As with many IT systems, a password authentication system must strike a balance between usability and security. A system that is exceedingly simple to use is likely not to be very secure; conversely, an extremely secure system is likely not to be very user and implementer-friendly.

For example, the first password systems, like the CTSS, stored passwords in plaintext; this was also the vector of the first "attack", which could be described in modern terms as a social engineering attack taking advantage of weak data security [1].

With the development of cryptography, it became possible to encrypt passwords and store them as ciphertext; that is, text that is unreadable to humans, but that could be deciphered or otherwise used by the computer. In 1974, Evans, Kantrowitz, and Weiss proposed a password authentication security measure in [2] that remains the de-facto standard approach: instead of storing the passwords plainly, they should be transformed using a one-way function, and the result of that transformation should be used for comparison purposes. Today, hash functions are used as one-way functions to this end – hash functions produce a fixed-length cryptographic digest of a message of arbitrary size. Some of the more well-known hash functions are MD5 (now considered outdated and insecure), the SHA family (SHA-1, SHA-2, etc.), crypt, and others.

While the main attack against hash-supported password authentication is the brute force attack, iterating over probable values, hash functions are still not perfect or immune from other forms of attack. One danger associated with hashing functions is the possibility of collisions, which are instances of two different input messages producing the same digest. Notably, MD5, once a widely-used hash function, was proven insecure and susceptible to collisions along with a set of other hash functions in [3]. Weaknesses in the SHA-1 algorithm were first discovered by Stevens et al., described in [4]; this was later expanded upon by the same authors in [5] and by other researchers. Collisions usually arise due to imperfections in the algorithm itself, and these attacks are generally remediated by switching to a different, more secure hashing algorithm, as was done for MD5 (to SHA-1) and is now being done for SHA-1 (to the SHA-2 and SHA-3 families of algorithms).

Another avenue of attack against hashed password authentication is the use of various dictionaries, tables, and other means of hash digest lookup. Such attacks trade-off storage space for computation time, storing pre-computed digests for commonly used passwords to drastically reduce the time required to find the necessary input data.

Oechslin in [6] first proposed the concept of a so-called “rainbow table”, which uses special chains between source data and hash digests to optimize lookup times. Lookup attacks are rendered ineffective by the use of “salting”, a technique that uses an arbitrary value concatenated with the input data to produce a new hash digest, different from the one that would be generated from the raw input data alone. The salt must be stored with the hash digest for the system to be able to reproduce the results later; however, as the value is chosen is arbitrary, and different for each password, generating a lookup table would be equivalent to a brute force attack in terms of time and resource requirements.

3 Passwords and the Human Factor

The previous section shows that, provided certain simple recommendations are applied, the only feasible technical attack against hash-based password authentication is a brute force one. However, the human factor is often the leading cause of breaches and a prime vector of attack. The main avenues of human-targeted attacks are being deceived into revealing their passwords, being compelled by force, and relying on personal associations and knowledge to create passwords, as well as relying on external information storage to remember them.

The problem of password creation and memorization is especially relevant to the field of information security, as security guidelines that cannot be followed due to human psychology are largely pointless and unenforceable. Therefore, they must be formulated concerning the average user’s capabilities and limitations.

Uniformly random passwords are the hardest to guess, but also the hardest to remember, especially when there are many to remember. A 2007 study by Florencio and Herley [8] showed that the average user accessed 25 accounts. However, even though password policies dictate that the password for each service should be unique, the average user only had 6.5 unique passwords, indicating a relatively high degree of password reuse. It is therefore likely that the number of accounts for a user has grown in the years since, and likely faster than the number of unique passwords.

4 Practical Applications

In a password security context, quantifying its efficiency means quantifying the quality of a user’s password. Various algorithms attempt to assess the quality of a password. Many are based on the concept of information entropy, which quantifies how “surprising”, or improbable, a random variable – here, the string of bytes – is on the whole.

According to the definition of Shannon information entropy, if pi is the probability of symbol number *i* appearing in the stream of symbols of length n > 0, the entropy for that message for an information unit of base b would be:

$$H = -\sum_{i=1}^{n} p_i \log_b p_i \tag{1}$$

Assuming a perfectly random password is used – i.e. the password is a string of L uniformly random symbols selected from a symbol set of length N – and that bits are used as the information unit, (1) is simplified:

$$II = \log_2 N^L \tag{2}$$

This corollary is equivalent to the Hartley function introduced by Ralph Hartley in 1928. It follows, therefore, that for a truly random password it is preferable to draw from as broad a set of symbols as possible from the start, adjusting the level of security by adjusting the length of the password. However, symbols should be different enough to avoid being present in a common subset, i.e. if the symbol set being used is "alpha-numeric characters and punctuation marks", care should be taken that the uniformly random resulting password does not consist e.g. solely of letters, as that effectively reduces the character set length from over 70 to just 52.

However, (1), and by an extension (2), are unsuitable for assessing entropy of many real-world passwords, as they are generated by people and thus contain patterns, based on language or otherwise. The authors propose Markov processes as a framework for calculating password quality because they are well-suited for quantifying how predictable a password is from the point of view of common patterns.

Let X = {X0, X1, ..., XT } - be a sequence of T random variables (T ≤ 0), V = {V1, V2, ..., VM} – the set of M states in a Markov process. In more familiar terms, X is the password itself, where each of the random variables is a single character, and V is the set of all possible characters in a password.

The conditional probability P(Xt = xt | Xt-1 = xt-1, ..., X1 = x1) describes a situation where the event at time t depends on all of the previous states of X. However, if it depends only on the immediately preceding state, i.e.

$$P(X_t = x_t \mid X_{t-1} = x_{t-1}, ..., X_1 = x_1) = P(X_t = x_t \mid X_{t-1} = x_{t-1}) \tag{3}$$

then it is a first-order Markov process. A zero-order Markov process is a process where every event is independent of every other event, i.e. the events are perfectly random. Entropy for a zero-order Markov process is calculated thus:

$$II_0 = -\sum_{i=1}^{M} P_i \log_2 P_i \tag{4}$$

For a first-order Markov process, the calculation becomes:

$$II_1 = -\sum_{i=1}^{M} P_i \sum_{j=1}^{M} P_{ij} \log_2 P_{ij} \tag{5}$$

And so on for higher-order processes. Markov processes are a useful estimate of how predictable a given password is since many common passwords are words or word combinations and thus well-described by Markov processes. A second-order Markov process would generally be an acceptable compromise between complexity and accuracy.

Markov probabilities may be calculated using publicly available databases of commonly used passwords and dictionaries of most common words in a given language;

for non-English-speaking users, it is generally necessary to assess both English- language and non-English-language probabilities. Additionally, common number-letter substitutions and casing differences could be accounted for on a software level for added accuracy.

The above technical and organizational recommendations have been tried and formulated during the authors' development of online university testing and learning systems. Salted hashing of passwords and the above approach to password quality assessment have been used by the authors in multiple systems, described in [9] and [10]; the systems have been copyrighted in [11-13].

However, it is recommended that passwords not be the only factor in a login system. Given the wide availability of mobile devices equipped with biometrics – mostly fingerprint scanners – it becomes easy to use biometrics as a true second factor, instead of OTPs or other solutions. The authors have explored the possibility of using mobile devices as a biometric authentication platform in a 2016 study [14] and developed a prototype for a mobile biometric authentication system used as an authentication module for web services. The system was designed to be fully passwordless, utilizing a common protocol to authenticate users via biometrics modules built into mobile devices and communicate with the web service requesting authentication. The prototype, copyrighted in [15], included a fingerprint module, which used an Android device's fingerprint sensor to authenticate.

5 Conclusion

Password authentication has changed and evolved over the years, but the user experience has largely remained the same. However, their effectiveness is now challenged by the ever-growing number of services that require them, and the ever-stricter requirements for their complexity as the number of attacks on Internet users grows.

Passwords are easy to implement for developers of authentication systems and integrators who connect them to services. However, modern passwords should be reasonably long and mostly random, which presents challenges for users – such passwords are hard to remember, so weaker passwords are created, and later reused.

The risk of "cracking" a password in a system has mostly been eliminated, but the risk of correctly guessing a password, even if it is through brute force iteration, remains significant. Some solutions have been proposed: password management software, or different approaches to generating passwords that are easier to remember. However, the problem remains.

Today, issues with password security have largely been secured against using multi-factor or multi-step authentication: the use of other means of authenticating a user in addition to their password – from simple ones like temporary codes delivered by text message or temporary time-limited passwords, to more complex solutions like biometrics or hardware token authentication.

As such, passwords will likely remain in some form as an authentication factor for a long time yet. However, the advent of widespread biometrics and an overall higher level

of network security is making it possible to use passwordless and multi-factor authentication more widely than ever before. On the whole, perfect must not be the enemy of the good, but there must be a basic level of security for any system – and passwords alone are no longer sufficient.

References

1. McMillan, R.: The world's First Computer Password? It was Useless too. WIRED. 2012. https://www.wired.com/2012/01/computer-password/ (2012)
2. Evans, A., Kantrowitz, W., Weis,s E.: A User Authentication Scheme not Requiring Secrecy in the Computer. Commun. ACM17(8). 1974. P. 437–442. https://doi.org/10.1145/361082.361087 (1974)
3. Wang, X., Feng, D., Lai, X., Yu, H.: Collisions for Hash Functions MD4, MD5, HAVAL-128, and RIPEMD. Cryptology ePrint Archive. Report 2004/199. 2004. https://eprint.iacr.org/2004/199 (2004)
4. Stevens, M., Karpman, P., Peyrin, T.: Freestart Collision for Full SHA-1. Cryptology ePrint Archive. Report 2015/967. 2015. https://eprint.iacr.org/ 2015/967 (2015)
5. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., Markov, Y.: The First Collision for Full SHA1. Annual International Cryptology Conference. 2017. P. 570—596. (2017)
6. Oechslin, P.: Making a Faster Cryptanalytic Time-Memory Trade-Off. Annual International Cryptology Conference. 2003. P. 617–630. (2003)
7. Ma, W., Campbell, J., Tran, D., Kleeman, D.: Password Entropy and Password Quality. 2010 Fourth International Conference on Network and System security. PP. 583–587. (2010).
8. Florencio, D., Herley, C.: A Large-Scale Study of Web Password Habits. Proceedings of the 16th international conference on World Wide Web. P. 657–666. (2007).
9. Boychenko, O., Gavrikov, I.: Using Two-Factor Authentication for Securing User Accounts in an Online Testing and Education System. International Scientific Review of the Problems and Prospects of Modern Science and Education: Proceedings of the XVI International Scientific and Practical Conference. V. 9 (19). P. 15–16. (2016).
10. Gavrikov, I.: Using Two-Factor Authentication for Securing User Accounts in an Online Testing and Education System. European Research: Innovation in Science, Education, and Technology: Proceedings of the XVII International Scientific and Practical Conference. N. 6 (17). P. 32–33. (2016).
11. Apatova, N., Boychenko, O., Gavrikov, I.: BI Virtual Education System. Certificate 2016660742 (in Russian). 2017. https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber=2016660742&TypeFile=html (2017)
12. Apatova, N., Boychenko, O., Georgiadi, A., Gavrikov, I.: UWT Online Testing System. Certificate 2017611276 https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM&DocNumber= 2017611276&TypeFile=html (in Russian). (2017).

13. Apatova, N., Boychenko, O., Kapustina, E., Gavrikov, I.: CALS: Computer Assisted Learning System. Certificate 2017615183 https://www1.fips.ru/registers-doc-view/fips_servlet?DB=EVM& DocNumber=2017615183&TypeFile=html (in Russian). (2017).
14. Boychenko, O., Gavrikov, I.: Implementing a Biometric Security Solution Using Mobile Devices. Regional information science and security: Proceedings of the XV International conference. P. 73–75. (in Russian). (2016).
15. Boychenko, O., Gavrikov, I.: FPAS: Mobile Biometric Authentication System. Certificate 2017619639 https://www1.fips.ru/ registers-doc-view/fips_servlet?DB=EVM&DocNumber=2017619639& TypeFile=html (in Russian). (2017).