Password Pattern and Vulnerability Analysis for Web and Mobile Applications

Shancang Li, Imed Romdhani, and William Buchanan

School of Computing, Edinburgh Napier University, Edinburgh EH10 5DT, Scotland, UK
{s.li, i.romdhani, b.buchannan}@napier.ac.uk

Abstract. Text-based passwords are heavily used to defense for many web and mobile applications. In this paper, we investigated the patterns and vulnerabilities for both web and mobile applications based on conditions of the Shannon entropy, Guessing entropy and Minimum entropy. We show how to substantially improve upon the strength of passwords based on the analysis of text-password entropies. By analyzing the passwords datasets '*rockyou'* and '*163com'*, we believe strong password can be designed based on good usability, deployability, rememberbility, and security entropies.

Keywords: *Password Strength, Security Entropies, Password Vulnerabilities*

1 Introduction

Although receiving many criticism, the text-based passwords are still heavily used for authenticating web and mobile application users [1,2]. A lot research efforts have been made to protect user's password against attacks [3]. In recent, many password managers have been developed to help people to create/manage secure passwords with enough strength and easy to remember (e.g. *Dashlane*, *Keepass*, *Lastpass*). However, when using a password manager, at least a master password need creating and remembering [4, 5].

A number of websites *have* recently been hacked and millions of user credentials were leaked online [6, 7]. In 2012, six millions of *LinkedIn* users' credentials were leaked. Actually, it was reported that in this case over 117 million user credentials were leaked on the Dark Web [8]. In 2015, the *Sony Pictures* was hacked and many confidential data were leaked, including 173,000 emails and 30,000 separate documents [9]. It was reported that in China 130,000 users' data were leaked via China's train ticketing site *12306* [10] in Dec. 2014. In 2015, a large number of websites (including *163.com, CSDN, TianYa, Duduniu, 7k7k, 178.com, Rockyou,* and *Yahoo*) were hacked and over 100 million of user credentials were leaked online. We believe that investigating the leaked passwords will be helpful in improving the strength of passwords in real data sources.

Most of users have various passwords for different web or mobile application accounts. However, it is difficult to remember so many passwords for a user. Although mobile devices have increasingly been used, it is still difficult to run a password

manager over mobile devices. Besides, unfriendly on-screen keyboards make it more challenging or inconvenient to type passwords with special symbols or mixed-case characters [4]. Many websites and mobile applications (apps) require users to choose complicate passwords (e.g., mixed-case letters, digits, special characters) and the authentication of passwords becomes more complicated. In this case, the text-password input interfaces (e.g. touchscreen virtual keyboards) are applied to protect users' passwords from malwares [4].

In this paper, we will investigate these leaked passwords to comprehensively identify the strength of passwords. Basically, we will focus on four features of the passwords: 1) Length of passwords, 2) Variety of character types in a password, 3) The randomness of passwords, and 4) Uniqueness of passwords. Mathematically, we will analyze the password entropy, guessing entropy, and Minimum entropy for passwords in the leaked passwords lists. A number of password analyzing tools, including *John the Ripper, Hashcat*, and *the password analysis and cracking toolkit (PACT*) will be used to analyze the password lists for password length, password entropies, character types, pattern detection of masks, and other password features.

2 Background and Previous Works

There have been many research efforts have been made for helping users to choose passwords. Measuring the strength of passwords is an important topic. In [1], a method for calculating password entropy was proposed, which is based on the summarization of the distributions of passwords length, placements of character, and number of each character types. Yan et al. [2] filtered weak passwords by improving *dictionary-based checking* with 7 character alphanumeric passwords. The password quality indicator (PQI) was proposed to measure or evaluate the quality of passwords [3, 4].

2.1 Typical Hash-based Login Systems

A typical website or mobile application login system contains following four basic steps: (1) A user registers and uses an assigned password; (2) The password is hashed and stored in the database (The plain-text password should never be written to the database); (3) When logging the system, the hash of the entered password will be checked against the hash saved in the database; (4) If the hashes match, the user will be granted access. The weakness in this system is that the passwords can often be guessed, forgotten, or revealed. To strength this weakness, a stronger password should be created or require additional authentication factors, such as physical token, digital certificates, one-time access code, etc. In many mobile app authentication systems, the mobile devices are increasingly used to enhances login security [17].

2.2 Recent Trends

In the past few decades, many password attacking methods have been developed, and most of them fall into two different types: *online* and *offline*. *Online* password attacks on websites or mobile apps are not easy for hackers since most login systems limit the times of trying passwords for a login. In *offline* password attacks, a hacker may take a password hash, copy it, and take it home to work on it.

A lot of concerns have been raised on the security of web and mobile application log in systems, however, we should bring awareness to the inherent weakness of password cracking by considering the following situations:

- Most authentication of websites and mobile applications have not moved to stronger hash type. Many weak hash types, such as Unix DES, NTLM and single-round salted SHA1 are still in using.
- Existing password policies have led to exploitable predictability.
- Authentication systems with design flaws are vulnerable to pass-the-hash attacks, for example the website like *163com*, *CSDN*, *Tianya* (before 2014).
- Power graphics processing unit (GPU) can significantly speed up brute force attacks to weak hashes, or even for long password length.
- Authentication enhanced by mobile device. In some applications, the username and password combination are inadequate for strong authentication, to further enhance login security addition factors (such as mobile device) can be used to make authentication stronger.

2.3 Password Attacks

A hacker can break passwords with many ways, out of which the following attacks are widely used:

1) Dictionary Attack

A hacker may use a password cracking dictionary (such as wordlist, dictionary, and password database leak) to find a password. The password dictionary is a very large text files that includes millions of generic passwords. The hacker may use higher performance computer or game graphics cards to try each of these passwords until find the right one [13].

2) Brute-force Attack

The brute-force attack, or exhaustive password attack, is still one of the most popular password cracking methods. It tries every possible combination until it gets the password. In practice, the password space for all possible combinations might be huge, which makes the brute force attack very difficult to carry out.

3) Man-in-the-Middle Attack

The man-in-the-middle (MITM) attacks may be used for regular web/mobile apps logins. It is possible for an attacker to find out authentication requests/responses from the recorded network traffic, and then capture candidates for password related contents. Passwords attack dictionary can be built by choosing login information over highly popular websites.

Traditionally, strong passwords are created by the following methods:

- *Proper length of passwords*. The length of passwords should be properly selected by balancing the user convenience and security. An eight-character password with a mix of numbers, symbols, and uppercase and lowercase can take at most months or years to crack. There is no minimum password length everyone agrees on, but in generally a password is required to be a minimum of 8 to 14 characters in length. A longer password would be even better [16].
- *Mix of numbers, symbols, uppercase and lowercase*. It is very difficult to crack such password; the only techniques are to try huge number of combinations until find the right one. For an eight-character password there are 83^8 possible combinations and need 10 days and 2 hours to crack [12, 16].
- *Salt password and avoid using password listed in password cracking dictionary*. The dictionary, wordlist, and password database are widely used in password cracking. In [13], a password cracking dictionary with a size of 15 GB has been released, which was used to successfully crack 49.98% of a password list with 373,000 passwords. To create a safer password, a better salting scheme is needed.
- *One-size-fits-all password*. Most websites or mobile apps apply one-size-fits-all approach to ensure that users choose strong passwords.
- *Outsource the security*. For most websites or mobile apps, outsourcing the security can provide massive value and it is a trend that looks like it will continue.

In summary, it is not very difficult to create a strong password with proper length, and a mix of many different types of characters. It is hard to guess such passwords due to its randomness. However, memorizing such a strong password is a problem. It is very difficult for most users to memorize a strong password created with a random password generator of websites and mobile applications. In creating a strong and memorable password, we need to think about how to avoid using something obvious with dictionary characters. For example, we can create a strong password based on a simple sentence like “I live in 20 Colinton Road at Edinburgh. The rent is $500 each month.” We can easily turn this simple sentence into a strong password by using the first letter or digit of each word, as “Ili20CR.Edi$5em”, which is a memorable and strong password with mix of numbers, characters, symbols, and uppercase letter and lowercase. It may be hacked in at least *420805123888006* years [12].

3 Password Strength Metrics and Evaluation

Password strength measurements can help to warn users away from highly vulnerable passwords [14]. Many authentication systems of websites and mobile applications require passwords must be able to resist eavesdroppers and off-line analysis of authentication protocols run. In general, the security of passwords can be measured with password strength. Password strength is defined in terms of probability of a determined attacker discovering a selected users' password by an inline attack. The password strength is also a function of both the entropy of the password and the way unsuccessful trials are limited. Entropy is believed as a standard measure of security [5].

3.1 Password Entropy

Shannon entropy is a popular method to evaluate the security strength of a password, which is also used as password entropy. Assuming a finite variable X corresponds to n passwords set (p_1, p_2 ,..., p_n), the password entropy can be modeled with Shannon entropy as $H(X)$

$$H(X) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i) \quad (1)$$

where p_i denotes the occurrence probability of ith possible outcome. A password using lowercase characters can be represented as $log_2(26) \approx 4.7$ bits of entropy per character. For a password "iliveinedinburgh" would have an entropy value of about $4.7 * 16 \approx 75$ bits.

The Shannon entropy is commonly used to measure the passwords. In recent, some variants of entropy have been proposed to measure other features of passwords such as guessing entropy, Minimum Entropy, relative entropy, etc.

3.2 Guessing Entropy

The ability of passwords that resist against complete off-line attacks can be measured with Guessing entropy. Guessing entropy is a measure of the difficulty to guess the passwords in a log in system [**6**]. If the values of $Y = sort_d(X)$ are sorted with decreasing probability， the guessing entropy of Y can be defined as

$$G(Y) = \sum_{i=1}^{n} i \cdot p_i \quad (2)$$

The guessing entropy is closely related to the average size of passwords. If a password has n bits guessing entropy, an attacker has as much difficult guessing the average password as in guessing an n bits random quantity [15].

3.3 Minmum Entropy

Since in some cases, the password strength cannot warn users away from reusing the same password because they are usually based on heuristics (e.g., numbers, password length, upper/lowercase, symbols). Minimum entropy is a way to estimate the strength of a password, the Min Entropy, H_{Min} is defined by

$$H_{min}(X) = -\min log_2(p_i) \quad (3)$$

For example, a low strength password p_b has low minimum entropy ($H_{min}(p_b) = 1$). High minimum entropy ($H_{min}(X) = \alpha$) guarantees that with high probability the adversary will always need to use around 2^{α} guesses to recover the users' passwords. The Minimum entropy shows the resistance of offline password cracking attacks with high probability.

3.4 Password over Mobile applications

Many mobile applications (apps) require password input and the authentication task over mobile platforms is more complicated due to using full-size key-board. In some mobile applications, the inconveniences caused by the unfriendly can affect users to create/use strong passwords. An example is that more than 80% of mobile device users are using digit-only passwords [6]. In recent, a number of password generation methods have been developed for mobile applications. For example, the object-based password (ObPwd) has been implemented over Android platform for generating password from a user-selected object (e.g., pictures) [7].

4 Analysis of the passwords

In this section, we investigated the way people create their passwords from five aspects: length, character types, randomness, complexity, and uniqueness. We analyzed over 100 million leaked and publicly available passwords from several popular websites (*Rockyou, CSDN, TianYa, 163com*).

4.1 Password length

Most of the passwords have length 6 to 10 characters as shown in Table.1 and Table.2, which specifies the percentage of total passwords was analyzed.

From both Table.1 and Table.4.1, we can find that 85% of passwords being between 8 to 10 characters long which is pretty predictable. Around 50% of the passwords both in *rockyou* and 163com lists are less than eight characters. Few passwords have a length greater than 13, the main reason is that most websites and mobile apps require a maximum length of 8 and the rememberability of the passwords.

Table 1. Password Length Analysis (*Rockyou.txt*).

Length	Percentage (%)	Number of Items
8	20	2966037
7	17	2506271
9	15	2191039
10	14	2013695
6	13	1947798

Table 2. Password Length Analysis (*163com.txt*).

Length	Percentage (%)	Number of Items
8	23	1159984
7	17	973951
6	17	870857
9	16	822077
10	11	572185
11	7	399021
12	2	141935
13	1	69776
14	1	58601

4.2 Character types

The length of passwords makes significant contributions to the entropy of passwords. Similarly, the diversity of the character types in passwords can be categorized into following sets: number, uppercase, lowercase, special-case.

The character-set both in *Rockyou.txt* and *163com.txt* are shown in Table.3 and Table.4.

Table 3. Password Character-set Analysis (*Rockyou.txt*).

Character-set	Percentage (%)	Number of Items
loweralphanum	88	4720183
upperalphanum	06	325942
mixedalphanum	05	293432

The character-set types analysis help us understand the usability and security and it is good to consider having three or more character types. In *rockyou* passwords, more than 80% passwords had only one-character type (lowercase). In *163com*, more than 88% of the passwords had only use numeric passwords.

Table 4. Password Character-set Analysis (*163com.txt*).

Character-set	Percentage (%)	Number of Items
Numeric	58	(2931867)
loweralphanum	30	(1527719)
loweralpha	08	(450746)
loweralphaspecialnum	00	(38913)
mixedalphanum	00	(26097)
upperalphanum	00	(23905)
specialnum	00	(15614)
loweralphaspecial	00	(4830)
mixedalpha	00	(4353)
upperalpha	00	(3142)
All	00	(2172)
upperalphaspecialnum	00	(1722)
mixedalphaspecial	00	(550)
Special	00	(164)
upperalphaspecial	00	(133)

4.3 Randomness

In our investigation, we found that many of the usual culprits are used such as "*password*", "*123456*", "*abc123*", cities name. We also found that many passwords are related to the fact that this passwords lists were apparently related to a competition: name of city, part of the user names, country names, etc. A few of these are very specific but there may have been context to this in the sign up process.

In Table.5 and 6 we analyzed the mask of passwords in both Rockyou.txt and 163com.txt. In Table., only greater of 1% of all passwords shown patterns matching the advanced masks, specifies the majority of "string-digit" passwords consist of a string with two or four digits following it.

4.4 Uniqueness

This uniqueness is about password sharing for different accounts. According to the analysis on many Chinese websites (*12306, 163.com, 126.com, Tianya,* CSDN, etc.), we found that many users are sharing the same passwords between their accounts in these websites.

Table 5. Password Advanced Masks Analysis (*163com.txt*).

Advanced Masks	Percentage (%)	Number of Items
?l?l?l?l?l?l?d?d	07	(420318)
?l?l?l?l?l?d?d	05	(292306)
?l?l?l?l?l?l?l?d?d	05	(273624)
?l?l?l?l?d?d?d?d	04	(235360)
?l?l?l?l?d?d	04	(215074)

Table 6. Password Advanced Masks Analysis (*163com.txt*).

Advanced Masks	Percentage (%)	Number of Items
?d?d?d?d?d?d	14	(727942)
?d?d?d?d?d?d?d?d	13	(701557)
?d?d?d?d?d?d?d	13	(692425)
?d?d?d?d?d?d?d?d?d	06	(348786)
?d?d?d?d?d?d?d?d?d?d	04	(244521)
?d?d?d?d?d?d?d?d?d?d	03	(162921)
?l?l?l?l?l?l?l?l	02	(117516)
?l?l?l?d?d?d?d?d?d	01	(90281)
?l?l?l?l?l?l?l?l?l	01	(78441)
?l?l?l?l?l?l	01	(74678)
?l?l?d?d?d?d?d?d	01	(71890)
?l?l?l?l?l?l?l	01	(67221)
?l?l?l?l?l?l?l?l?l?l	01	(66316)
?l?l?d?d?d?d?d?d?d	01	(65743)
?l?l?l?d?d?d?d?d?d?d	01	(61386)
?l?d?d?d?d?d?d?d	01	(53065)

It is believed that the leakage of *12306, Tianya*, etc. are caused by the *hit the library attack*, in which users privacy data leakage is more like a hacker hit the library behavior. The hit the library attack is compromised by hackers to collect internet user and password information, after generating the corresponding dictionary table, try another batch landing sites, users can visit to obtain a series of previously attacked *Jingdong* also knocked library. By this way, the hacker can deal with almost any website login system, users use the same username and password when you log on different sites, which is equivalent to their own with a handful of 'master key' to facilitate its own, but also convenient for hackers.

In *Sony leakage* case, 92% of passwords were reused across both in 'Beauty' and 'Delboca' login systems. Only 8% of identical passwords are used. In internet web and mobile applications, for simplicity many users are using the same email as their login username which increases the risks of password sharing. In the Chinese passwords leakages, a statistically good chance that the majority of users will work with other websites, and the users are misleading because anyone can grab these off the net right now.

5. Conclusion

In this paper, we analyzed the strength of passwords and investigate the password leakages cases from four viewpoints: length, character types, randomness, complexity, and uniqueness, which is expected to warn users away from highly vulnerable passwords.

References

1 BURR, W. E., DODSON, D. F. POLK, W. T. (2006) Electronic Authentication Guideline: Recommendations of the National Institute of Standards and Technology. 1.0.2 ed., NIST, USA. http://csrc.nist.gov/publications/nistpubs/800-63/SP80063V1.pdf

2 CAMPBELL, J., KLEEMAN, D. MA, W. (2006) Password Composition Policy: Does Enforcement Lead to Better Password Choices? 17th Australasian Conference on Information Systems (ACIS 2006). Adelaide, Australia.

3 WCAMPBELL, J., KLEEMAN, D. MA, W. (2007) The Good and Not So Good of Enforcing Password Composition Rules. Information Systems Security, 16, 2-8.

4 CISNEROS, R., BLISS, D. GARCIA, M. (2006) Password auditing applications. Journal of Computing in Colleges, 21, 196-202.

5 Meng-Hui Lim, and Pong C. Yuen, Entropy Measurement for Biometric Verification Systems. EEE TRANSACTIONS ON CYBERNETICS, VOL. 46, NO. 5, MAY 2016.

6 M. Jakobsson, E. Shi, P. Golle, and R. Chow, Implicit Authentication for Mobile Devices, USENIX Workshop on Hot Topics in Security (HotSec09), Montreal, Canada, Aug. 2009.

7 MOHAMMAD MANNAN AND P.C. VAN OORSCHOT, "Passwords for Both Mobile and Desktop Computers ObPwd for Firefox and Android", Available from: https://www.usenix.org/system/files/login/articles/mannan.pdf

8 MakeUseOf, "What You Need To Know About the Massive LinkedIn Accounts Leak", [Accessed 11th May 2016] Available from:http://www.makeuseof.com/tag/need- know-massive-linkedin-accounts-leak

9 Sean Fitz-Gerald, "Everything Thats Happened in the Sony Leak Scandal", [Accessed 11th May 2016] Available from: http://www.makeuseof.com/tag/need-knowmassive-linkedin-accounts-leak

10 Eileen Yu, "130K users' data leaked via China's train ticketing site", [Accessed 11th May 2016] Available from: http://www.zdnet.com/article/130k-users-data-leakedvia-chinas-train-ticketing-site

11 Troy Hunt, "A brief Sony password analysis", [Accessed 11th May 2016] Available from: https://www.troyhunt.com/brief-sony-password-analysis

12 Random-ize, "How Long to Hack My Password", [Accessed 11th May 2016] Available from: http://random-ize.com/how-long-to-hack-pass

13Eric Escobar, "How Long to Hack My Password", [Accessed 11th May 2016] http://www.quickanddirtytips.com/tech/computers/how-to-crack-a-password-likea-hacker?page=1

14Jeremiah Blocki, "Password Strength Meters", [Accessed 11th May 2016] http://www.cs.cmu.edu/ jblocki/entropyAndMinimumEntropy.htm

15NIST, "Electronic Authentication Guideline (NIST Special Publication 800-63) (Apr. 2006)", [Accessed 11th May 2016] http://itlaw.wikia.com/wiki/NIST-Special-Publication-800-63

16 Bill Buchanan, [Accessed 11th May 2016]. http://asecuritysite.com/encryption/passes

17 Matt Sarrel, Authentication Via Mobile Phone Enhances Login Security, [Accessed 11th May 2016], http://www.informationweek.com/applications/authentication-via-mobile-phone-enhances-login-security/d/d-id/1103017?

Biographies

Dr Shancang Li (s.li@napier.ac.uk) (BEng, MEng, PhD) is a lecturer in Network Forensics in School of Computing at Edinburgh Napier University. Over the last few years, he has been working on a few research projects funded by EU, EPSRC, A4B

(Academic expertise for Business), TSB (Technology Strategy Board), and industry. Based on these research projects, dozens of papers have been published. His current research interests include network forensics, security, wireless sensor networks, Internet of Things, and lightweight cryptography over IoT.

Dr Imed Romdhani (i.romdhani@napier.ac.uk) is an Associate Professor in computer networking at Edinburgh Napier University. He was awarded his PhD from the University of Technology of Compiegne (UTC), France in May 2005 and an engineering and a Master degree in networking obtained respectively in 1998 and 2001 from the National School of Computing (ENSI, Tunisia) and Louis Pasteur University of Strasbourg (ULP, France). He worked extensively with Motorola Research Labs in Paris and authored 4 patents in the field of IPv6, Multicast Mobility and the Internet of Things.

Professor William Buchanan (w.buchanan@napier.ac.uk) is a Professor in the School of Computing at Edinburgh Napier University, and a Fellow of the BCS and the IET. He currently leads the Centre for Distributed Computing, Networks, and Security and The Cyber Academy, and works in the areas of security, Cloud Security, Web-based infrastructures, e-Crime, cryptography, triage, intrusion detection systems, digital forensics, mobile computing, agent-based systems, and security risk.