A Study of Personal Information in Human-chosen Passwords and Its Security Implications

Yue Li*, Haining Wang†, Kun Sun*
*Department of Computer Science, College of William and Mary
{yli,ksun}@cs.wm.edu
†Department of Electrical and Computer Engineering, University of Delaware
hnw@udel.edu

***Abstract*—Though not recommended, Internet users often include parts of personal information in their passwords for easy memorization. However, the use of personal information in passwords and its security implications have not yet been studied systematically in the past. In this paper, we first dissect user passwords from a leaked dataset to investigate how and to what extent user personal information resides in a password. In particular, we extract the most popular password structures expressed by personal information and show the usage of personal information. Then we introduce a new metric called Coverage to quantify the correlation between passwords and personal information. Afterwards, based on our analysis, we extend the Probabilistic Context-Free Grammars (PCFG) method to be semantics-rich and propose Personal-PCFG to crack passwords by generating personalized guesses. Through offline and online attack scenarios, we demonstrate that Personal-PCFG cracks passwords much faster than PCFG and makes online attacks much easier to succeed.**

I. Introduction

Text-based passwords still remain a dominating and irreplaceable authentication method in the foreseeable future. Although people have proposed different authentication mechanisms, no alternative can bring all the benefits of passwords without introducing any extra burden to users [1]. However, passwords have long been criticized as one of the weakest links in authentication. Due to human-memorability requirement, user passwords are usually far from true random strings [2]–[6]. In other words, human users are prone to choosing weak passwords simply because they are easy to remember. As a result, most passwords are chosen within only a small portion of the entire password space, being vulnerable to brute-force and dictionary attacks.

To increase password security, online authentication systems start to enforce stricter password policies. Meanwhile, many websites deploy password strength meters to help users choose secure passwords. However, these meters are proved to be ad-hoc and inconsistent [7], [8]. To better assess the strength of passwords, we need to have a deeper understanding on how users construct their passwords. If an attacker knows exactly how users create their passwords, guessing their passwords will become much easier. Meanwhile, if a user is aware of the potential vulnerability induced by a commonly used password creation method, the user can avoid using the same method for creating passwords.

Toward this end, researchers have made significant efforts to unveil the structures of passwords. Traditional dictionary attacks on passwords have shown that users tend to use simple dictionary words to construct their passwords [9]. Language is also vital since users tend to use their first languages when constructing passwords [2]. Besides, passwords are mostly phonetically memorable [4] even though they are not simple dictionary words. It is also indicated that users may use keyboard and date strings in their passwords [5], [10], [11]. However, most studies discover only superficial password patterns, and the semantic-rich composition of passwords is still mysterious to be fully uncovered. Fortunately, an enlightening work investigates how users generate their passwords by learning the semantic patterns in passwords [12].

In this paper, we study password semantics from a different perspectivethe use of personal information. We utilize a leaked password dataset, which contains personal information, from a Chinese website for this study. We first measure the usage of personal information in password creation and present interesting observations. We are able to obtain the most popular password structures with personal information embedded. We also observe that males and females behave differently when using personal information in password creation. Next, we introduce a new metric called Coverage to accurately quantify the correlation between personal information and user password. Since it considers both the length and continuation of personal information in a password, Coverage is a useful metric to measure the strength of a password. Our quantification results using the Coverage metric confirm our direct measurement results on the dataset, showing the efficacy of Coverage. Moreover, Coverage is easy to be integrated with existing tools, such as password strength meters for creating a more secure password.

To demonstrate the security vulnerability induced by using personal information in passwords, we propose a semantics-rich Probabilistic Context-Free Grammars (PCFG) method called Personal-PCFG, which extends PCFG [13] by considering those symbols linked to personal information in password structures. Personal-PCFG is able to crack passwords much faster than PCFG. It also makes an online attack more feasible by drastically increasing the guess success rate. Finally, we discuss potential solutions to defend against semantics-aware attacks like Personal-PCFG.

Our study is based on a dataset collected from a Chinese website. Although measurement results could be different with other datasets, our observations still shed some light on how personal information is used in passwords. As long as memorability plays an important role in password creation, the correlation between personal information and user password remains, regardless of which language users speak. We believe that our work on personal information quantification, password cracking, and password protection could be applicable to any other text-based password datasets from different websites.

The remainder of this paper is organized as follows. Section II measures how personal information resides in user passwords and shows the gender difference in password creation. Section III introduces the new metric, Coverage, to accurately quantify the correlation between personal information and user password. Section IV details Personal-PCFG and shows cracking results compared with the original PCFG. Section V discusses limitations and potential defenses. Section VI surveys related work, and finally Section VII concludes this paper.

II. Personal Information in Passwords

Intuitively, people tend to create passwords based on their personal information because human beings are limited by their memory capacities and random passwords are much harder to remember. We show that users' personal information plays an important role in human-chosen password generation by dissecting passwords in a mid-sized leaked password dataset. Understanding the usage of personal information in passwords and its security implications can help us to further enhance password security. To start, we introduce the dataset used throughout this study.

A. 12306 Dataset

A number of password datasets have been exposed to the public in recent years, usually containing several thousands to millions of real passwords. As a result, there are several password measurement or password cracking studies based on analyzing those datasets [2], [10]. In this paper, a dataset called 12306 is used to illustrate how personal information is involved in password creation.

1) Introduction to Dataset: At the end of year 2014, a Chinese dataset is leaked to the public by anonymous attackers. It is reported that the dataset is collected by trying usernames and passwords from other leaked datasets online. We call this dataset 12306 because all passwords are from the website www.12306.cn, which is the official site of the online railway ticket reservation system in China. There is no data available on the exact number of users of the 12306 website; however, we infer at least tens of millions of registered users in the system since it is the only official website for the entire Chinese railway system.

The 12306 dataset contains more than 130,000 Chinese passwords. Having witnessed so many leaked large datasets, the size of the 12306 dataset is considered medium. What makes it special is that together with plaintext passwords, the dataset also includes several types of user personal information, such as a user's name and the government-issued unique ID number (similar to the U.S. Social Security Number). As the website requires a real ID number to register and people must provide real personal information to book a ticket, we consider the information in this dataset to be reliable.

TABLE I: Most Frequent Passwords.

Rank	Password	Amount	Percentage
1	123456	389	0.296%
2	a123456	280	0.213%
3	123456a	165	0.125%
4	5201314	160	0.121%
5	111111	156	0.118%
6	woaini1314	134	0.101%
7	qq123456	98	0.074%
8	123123	97	0.073%
9	000000	96	0.073%
10	1qaz2wsx	92	0.070%

2) Basic Analysis: We first conduct a simple analysis to reveal some general characteristics of the 12306 dataset. For data consistency, we remove users whose ID number is not 18-digit long. These users may have used other IDs (e.g., passport number) to register on the system and count for 0.2% of the whole dataset. The dataset contains 131,389 passwords for analysis after being cleansed. Note that various websites may have different password creation policies. For instance, with a strict password policy, users may apply mangling rules (e.g., $abc -> @bc$ or $abc1$) to their passwords to fulfill the policy requirement [14]. Since the 12306 website has changed its password policy after the password leak, we do not know the exact password policy when the dataset was first compromised. However, from the leaked dataset, we infer that the password policy is quite simple—all passwords cannot be shorter than six symbols. There is no restriction on what type of symbols can be used. Therefore, users are not required to apply any mangling rules to their passwords.

The average length of passwords in the 12306 dataset is 8.44. The most common passwords in the 12306 dataset are listed in Table I. The dominating passwords are trivial passwords (e.g., 123456, a123456, etc.), keyboard passwords (e.g., 1qaz2wsx, 1q2w3e4r, etc.), and "iloveyou" passwords. Both "5201314" and "woaini1314" mean "I love you forever" in Chinese. The most commonly used Chinese passwords are similar to a previous study [10]; however, the 12306 dataset is much more sparse. The most popular password "123456" counts for less than 0.3% of all passwords while the number is 2.17% in [10]. We believe that the password sparsity is due to the importance of the website; users are less prone to use trivial passwords like "123456" and there are fewer sybil accounts because a real ID number is needed for registration.

Similar to [10], we measure the resistance to guessing of the 12306 dataset in terms of various metrics including the worst-case security bit representation (H_∞), the guesswork bit representation ($\tilde{G}$), the α-guesswork bit representations ($\tilde{G}_{0.25}$ and $\tilde{G}_{0.5}$), and the β-success rates (λ_5 and λ_{10}). The result is shown in Table II. We found that users of 12306 avoid using extremely guessable passwords such as "123456" because 12306 has a substantially higher worst-case security and the β-success rate for $\beta = 5$ and 10. We believe users have certain password security concerns when creating passwords for critical service systems like 12306. However, their concern seems to be limited by avoiding only extremely easy passwords. As indicated by values of $alpha$-guesswork, the overall password sparsity of the 12306 dataset is no higher

TABLE II: Resistance to guessing

H_∞	$\tilde{G}$	λ_5	λ_{10}	$\tilde{G}_{0.25}$	$\tilde{G}_{0.5}$
8.4	16.85	0.25%	0.44%	16.65	16.8

TABLE III: Most Frequent Password Structures.

Rank	Structure	Amount	Percentage
1	D_7	10,893	8.290%
2	D_8	9,442	7.186%
3	D_6	9,084	6.913%
4	L_2D_7	5,065	3.854%
5	L_3D_6	4,820	3.668%
6	L_1D_7	4,770	3.630%
7	L_2D_6	4,261	3.243%
8	L_3D_7	3,883	2.955%
9	D_9	3,590	2.732%
10	L_2D_8	3,362	2.558%

"D" represents digits and "L" represents English letters. The number indicates the segment length. For example, L_2D_7 means the password contains 2 letters followed by 7 digits.

than previously studied datasets.

We also study the basic structures of the passwords in 12306. The most popular password structures are shown in Table III. Similar to a previous study [10], our results again show that Chinese users prefer digits in their passwords as opposed to letters like English-speaking users. The top five structures all have a significant portion of digits, and at most 2 or 3 letters are appended in front. The reason behind this may be that Chinese characters are logogram-based, and digits seem to be the best alternative when creating a password.

In summary, the 12306 dataset is a Chinese password dataset that has general Chinese password characteristics. Users have certain security concerns by choosing less trivial passwords. However, the overall sparsity of the 12306 dataset is no higher than previously studied datasets.

B. Personal Information

The 12306 dataset not only contains user passwords but also multiple types of personal information listed in Table IV.

TABLE IV: Personal Information.

Type	Description
Name	User's Chinese name
Email address	User's registered email address
Cell phone	User's registered cell phone number
Account name	The username used to log in the system
ID number	Government issued ID number

Note that the government-issued ID number is a unique 18-digit number, which includes personal information itself. Digits 1-6 represent the birthplace of the owner, digits 7-14 represent the birthdate of the owner, and digit 17 represents the gender of the owner—odd means male and even means female. We take out the 8-digit birthdate and treat it separately since birthdate is very important personal information in password creation. Therefore, we finally have six types of personal information: name, birthdate, email address, cell phone number, account name, and ID number (birthdate excluded).

1) New Password Representation: To better illustrate how personal information correlates to user passwords, we develop a new representation of a password by adding more semantic symbols besides the conventional "D", "L" and "S" symbols, which stand for digit, letter, and special symbol, respectively. We try to match parts of a user's password to the six types of personal information, and express the password with these personal information. For example, a password "alice1987abc" can be represented as $[Name][Birthdate]L_3$, instead of $L_3D_4L_3$ as in a traditional representation. The matched personal information is denoted by corresponding tags—[Name] and [Birthdate] in this example; for segments that are not matched, we still use "D", "L", and "S" to describe the symbol types.

We believe that representations like $[Name][Birthdate]L_3$ are better than $L_5D_4L_3$ since they more accurately describe the composition of a user password with more detailed semantic information. Using this representation, we apply the following matching method to the entire 12306 dataset to see how these personal information tags appear in password structures.

2) Matching Method: We propose a matching method to locate personal information in a user password. The basic idea is that we first generate all substrings of the password and sort them in descending length order. Then we match these substrings from the longest to the shortest to all types of personal information. If a match is found, the match function is recursively applied over the remaining password segments until no further match is found. We require that a segment should be at least 2-symbol long to be matched. The segments that are not matched to any personal information will then be labeled using the traditional "LDS" tags.

We describe the methods for matching each type of the personal information as follows. For the Chinese names, we convert them into Pinyin form, which is alphabetic representation of Chinese characters. Then we compare password segments to 10 possible permutations of a name, such as lastname+firstname and last_initial+firstname. If the segment is exactly the same as one of the permutations, we consider it a match. For birthdate, we list 17 possible permutations and compare a password segment with these permutations. If the segment is the same as any permutation, we consider it a match. For account name, email address, cell phone number, and ID number, we further constrain the length of a segment to be at least 3 to avoid mismatching by coincidence. Besides, as people tend to memorize a sequence of numbers by dividing into 3-digit groups, we believe that a match of at least 3 is likely to be a real match.

Note that for a password segment, it may match multiple types of personal information. In such cases, all possible matches are counted in the results.

3) Matching Results: After applying the matching method to 12306 dataset, we find that 78,975 out of 131,389 (60.1%) of the passwords contain at least one of the six types of personal

TABLE V: Most Frequent Password Structures.

Rank	Structure	Amount	Percentage
1	[ACCT]	6,820	5.190%
2	D7	6,224	4.737%
3	[NAME][BD]	5,410	4.117%
4	[BD]	4,470	3.402%
5	D6	4,326	3.292%
6	[EMAIL]	3,807	2.897%
7	D8	3,745	2.850%
8	L1D7	2,829	2.153%
9	[NAME]D7	2,504	1.905%
10	[ACCT][BD]	2,191	1.667%

TABLE VI: Personal Information Usage.

Rank	Information Type	Amount	Percentage
1	Birthdate	31,674	24.10%
2	Account Name	31,017	23.60%
3	Name	29,377	22.35%
4	Email	16,642	12.66%
5	ID Number	3,937	2.996%
6	Cell Phone	3,582	2.726%

TABLE VII: Most Frequent Structures in Different Genders.

Rank	Male		Female	
	Structure	Percentage	Structure	Percentage
1	[ACCT]	4.647%	D6	3.909%
2	D7	4.325%	[ACCT]	3.729%
3	[NAME][BD]	3.594%	D7	3.172%
4	[BD]	3.080%	D8	2.453%
5	D6	2.645%	[EMAIL]	2.372%
6	[EMAIL]	2.541%	[NAME][BD]	2.309%
7	D8	2.158%	[BD]	1.968%
8	L1D7	2.088%	L2D6	1.518%
9	[NAME]D7	1.749%	L1D7	1.267%
10	[ACCT][BD]	1.557%	L2D7	1.240%
NA	TOTAL	28.384%	TOTAL	23.937%

TABLE VIII: Most Frequent Personal Information in Different Genders.

Rank	Male		Female	
	Information Type	Percentage	Information Type	Percentage
1	[BD]	24.56%	[ACCT]	22.59%
2	[ACCT]	23.70%	[BD]	20.56%
3	[NAME]	23.31%	[NAME]	12.94%
4	[EMAIL]	12.10%	[EMAIL]	13.62%
5	[ID]	2.698%	[CELL]	2.982%
6	[CELL]	2.506%	[ID]	2.739%

information. Apparently, personal information is frequently used in password creation. We believe that the ratio could be even higher if we know more personal information of users. We present the top 10 password structures in Table V and the usage of personal information in passwords in Table VI. As mentioned above, a password segment may match multiple types of personal information, and we count all of these matches. Therefore, the sum of the percentages is larger than 60.1%. Within 131,389 passwords, we obtain 153,895 password structures. Based on Tables V and VI, we can see that people largely rely on personal information when creating passwords. Among the 6 types of personal information, birthdate, account name, and name are most popular with over 20% occurrence rate. 12.66% users include email in their passwords. However, only few percentage of people include their cellphone and ID number in their passwords (less than 3%).

4) Gender Password Preference: As the user ID number in our dataset actually contains gender information (i.e., the second-to-last digit in the ID number represents gender), we compare the password structures between males and females to see if there is any difference in password preference. Since the dataset is biased in gender with 9,856 females and 121,533 males, we randomly select 9,856 males and compare with females.

The average password lengths for males and females are 8.41 and 8.51 characters, respectively, which shows that gender does not greatly affect the length of passwords. We then apply the matching method to each gender. We observe that 61.0% of male passwords contain personal information while only 54.1% of female passwords contain personal information. We list the top 10 structures for each gender in Table VII and personal information usage in Table VIII. These results demonstrate that male users are more likely to include personal information in their passwords than female users. Additionally, we have two other interesting observations. First, the total number of password structures for females is 1,756, which is 10.3% more than that of males. Besides, 28.38% of males' passwords fall into the top 10 structures while only 23.94% of females' passwords fall into the top 10 structures. Thus, passwords created by males are denser and more predictable. Second, males and females vary significantly in the use of name information. 23.32% passwords of males contain their names. By contrast, only 12.94% of females' passwords contain their names. We notice that name is the main difference in personal information usage between males and females.

In summary, passwords of males are generally composed of more personal information, especially the name of a user. In addition, the password diversity for males is lower. Our analysis indicates that the passwords of males are more vulnerable to cracking than those of females. At least from the perspective of personal-information-related attacks, our observations are different from the conclusion drawn in [15] that males have slightly stronger passwords than females.

III. Correlation Quantification

While the statistical numbers above show the correlation between each type of personal information and passwords, they cannot accurately measure the degree of personal information involvement in an individual password. Thus, we introduce a novel metric—Coverage—to quantify the involvement of personal information in the creation of an individual password in an accurate and systematic fashion.

A. Coverage

The value of Coverage ranges from 0 to 1. A larger Coverage implies a stronger correlation, and Coverage "0" means no personal information is included in a password and Coverage "1" means the entire password is perfectly matched with one type of personal information. While Coverage is mainly used for measuring an individual password, the average Coverage also reflects the degree of correlation in a set of passwords. In the following, we describe the algorithm to compute Coverage and elaborate the key features of Coverage.

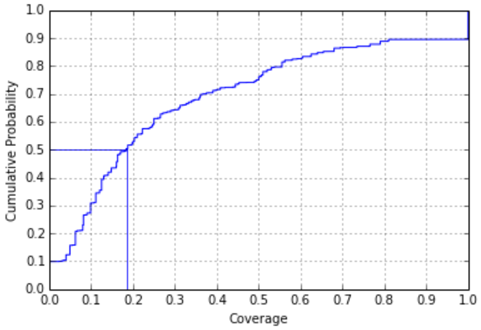
To compute Coverage, we take password and personal information in terms of strings as input and use a sliding window approach to conducting the computation. We maintain a dynamic-sized window sliding from the beginning to the end of the password. The initial size of the window is 2. If the segment covered by the window matches to a certain type of personal information, we enlarge the window size by 1. Then we try again to match the segment in the larger window to personal information. If a match is found, we further enlarge the window size until a mismatch happens. At this point, we reset the window size to the initial value 2 and slide the window to the password symbol that causes the mismatch in the previous window. Meanwhile, we maintain an array called *tag array* with the same length as the password to record the length of each matched password segment. After we slide the window through the entire password string, the tag array is used to compute the value of Coverage—the sum of squares of matched password segment length divided by the square of password length. Mathematically we have

$$CVG = \sum_{i=1}^{n}(\frac{l_i^2}{L^2}), \tag{1}$$

where n denotes the number of matched password segments, l_i denotes the length of the corresponding matched password segment, and L is the length of the password. Note that a match is found if at least a 2-symbol-long password segment matches to a substring of certain personal information. We then show an example to compute Coverage for a user password. Alice, who was born on August 16, 1988, has a password "alice816!!". We apply the coverage computing algorithm on Alice. After sliding the window thoroughly, the tag array is [5,5,5,5,5,3,3,3,0,0]. The first five elements in the array, i.e., {5,5,5,5,5}, indicate that the first 5 password symbols match certain type of personal information (name in this case). The following three elements in the array, i.e., {3,3,3}, indicate that the 3 symbols match certain type of personal information (birthdate in this case). The last two elements in the array, i.e., {0,0}, indicate that the last 2 symbols have no match. Based on Equation 1, the coverage is computed as $CVG = \sum_{i=1}^{2} \frac{l_i^2}{L^2} = \frac{5^2+3^2}{10^2} = 0.34$.

Coverage is independent of password datasets. As long as we can build a complete string list of personal information, Coverage can accurately quantify the correlation between a user's password and its personal information. For personal information segments with the same length, Coverage stresses the continuation of matching. A continuous match is stronger than fragmented matches. That is to say, for a given password of length L, a matched segment of length l ($l \leq L$) has a stronger correlation to personal information than two matched segments of length l_1 and l_2 with $l = l_1 + l_2$. For example, a matched segment of length 6 is expected to have a stronger correlation than 2 matched segments of length 3. This feature of Coverage is desirable because multiple shorter segments (i.e., originated from different types of personal information) are usually harder to guess and may involve a wrong match due to coincidence. Since it is difficult to differentiate a real match from a coincidental match, we would like to minimize the effect of wrong matches by taking squares of the matched segments to compute Coverage in favor of a continuous match.

Fig. 1: Coverage distribution - 12306.



B. *Coverage Results on 12306*

We compute the Coverage value for each user in the 12306 dataset and show the result as a cumulative distribution function in Figure 1. To easily understand the value of Coverage, we discuss a few examples to illustrate the implication of a roughly 0.2 Coverage. Suppose we have a 10-symbol-long password. One matched segment with length 5 will yield 0.25 Coverage. Two matched segments with length 3 (i.e., in total 6 symbols are matched to personal information) yield 0.18 Coverage. Moreover, 5 matched segments with length 2 (i.e., all symbols are matched but in a fragmented fashion) yield 0.2 Coverage. Apparently, Coverage of 0.2 indicates a fairly high correlation between personal information and a password.

The median value for a user's Coverage is 0.186, which implies that a significant portion of user passwords have relatively high correlation to personal information. Furthermore, Around 10.5% of users have Coverage of 1, which means that 10.5% of passwords are perfectly matched to exactly one type of personal information. On the other hand, around 9.9% of users have zero Coverage, implying no use of personal information in their passwords.

The average Coverage for the entire 12306 dataset is 0.309. We also compute the average Coverages for male and female groups, since we observe that male users are more likely to include personal information in their passwords in Section II-B4. The average Coverage for the male group is 0.314, and the average Coverage for the female group is 0.269. It complies with our previous observation and indicates that the correlation for male users is higher than that of female users. Conversely, it also shows that Coverage works very well to quantify the correlation between passwords and personal information.

C. *Coverage Usage*

Coverage could be very useful for constructing password strength meters, which have been reported as mostly ad-hoc [7]. Most meters give scores based on password structure and length or blacklist commonly used passwords (e.g., the notorious "password"). There are also meters that perform simple social profile analysis, such as rejecting a password when it contains the user's name or the account name. However, these simple analysis mechanisms can be easily mangled, while the password remains weak. Using the metric of Coverage, password strength meters can be improved to more accurately measure the strength of a password. Moreover, it is straightforward to implement Coverage as a part of the strength measurement (only a few lines of Javascript should do). More importantly, since users cannot easily defeat the Coverage measurement through simple mangling methods, they are forced to select more secure passwords.

Coverage can also be integrated into existing tools to enhance their capabilities. There are several Markov model based tools that predict the next symbol when a user creates a password [14], [16]. These tools rank the probability of the next symbol based on the Markov model learned from dictionaries or leaked datasets, and then show the most probable predictions. Since most users would be surprised to find that the next symbol in their mind matches the tool's output exactly, they may switch to choose a more unpredictable symbol. Coverage helps to determine whether personal information prediction ranks high enough in probability to remind a user of avoiding the use of personal information in password creation.

IV. Personal-PCFG

After investigating the correlation between personal information and user passwords through measurement and quantification, we further study their potential usage to crack passwords from an attacker's point of view. Based on the PCFG approach [13], we develop Personal-PCFG as an individual-oriented password cracker that can generate personalized guesses towards a targeted user by exploiting the already known personal information.

A. Attack Scenarios

We assume that the attacker knows a certain amount of personal information about the targets. The attacker can be an evil neighbor, a curious friend, a jealous husband, a blackmailer, or even a company that buys personal information from other companies. Under these conditions, targeted personal information is rather easy to obtain by knowing the victim personally or searching online, especially on social networking sites (SNS) [17], [18]. Personal-PCFG can be used in both offline and online attacks.

In traditional offline password attacks, attackers usually steal hashed passwords from victim systems, and then try to find out the unhashed values of these passwords. As a secure hash function cannot be simply reversed, the most popular attacking strategy is to guess and verify passwords by brute force. Each guess is verified by hashing a password (salt needs to be added) from a password dictionary and comparing the result to the hashed values in the leaked password database. High-probability password guesses can usually match many hashed values in the password database and thus are expected to be tried first. For offline attacks, Personal-PCFG is much faster in guessing the correct password than conventional methods, since it can generate high-probability personalized passwords and verify them first.

For an online attack, since the attacker does not even have a hashed password database, he or she instead tries to log in directly to the real systems by guessing the passwords. It is more difficult to succeed in online attacks than offline attacks because online service systems usually have restrictions on login attempts for a given period of time. If the attempt quota has been reached without inputting a correct password, the account may be locked for some time or even permanently unless certain actions are taken (e.g., call the service provider). Therefore, online attacks require accurate guesses, which can be achieved by integrating personal information. Personal-PCFG is able to crack around 1 out of 20 passwords within only 5 guesses.

B. A Revisit of PCFG

Personal-PCFG is based on the basic idea of PCFG [13] and provides an extension to further improve its efficiency. Before we introduce Personal-PCFG, we briefly revisit principles of PCFG. PCFG pre-processes passwords and generates base password structures such as "$L_5D_3S_1$" for each of the passwords. Starting from high-probability structures, the PCFG method substitutes the "D" and "S" segments using segments of the same length learned from the training set. These substitute segments are ranked by probability of occurrence learned from the training set. Therefore, high probability segments will be tried first. One base structure may have a number of substitutions, for example, "$L_5D_3S_1$" can have "$L_5$123!" and "$L_5$691!" as its substitutions. These new representations are called pre-terminal structures. No "L" segment is currently substituted since the space of alpha strings is too large to learn from the training set. Next, these pre-terminals are ranked from high probability to low probability. Finally "L" segments are substituted using a dictionary to generate actual guesses. Since PCFG can generate statistically high probability passwords first, it can significantly reduce the guessing number of traditional dictionary attacks.

C. Personal-PCFG

Personal-PCFG leverages the basic idea of PCFG. Besides "L", "D", and "S" symbols in PCFG, we add more semantic symbols including "B" for birthdate, "N" for name, "E" for email address, "A" for account name, "C" for cell phone number, and "I" for ID number. Richer semantics makes Personal-PCFG more accurate in guessing passwords. To make Personal-PCFG work, an additional personal information matching phase and an adaptive-substitution phase are added to the original PCFG method. Therefore, Personal-PCFG has 4 phases in total and the output of each phase will be fed to the next phase as input. The output of the last phase is the actual guesses for trying. We now describe each phase in detail along with simple examples.

1) Personal Information Matching: Given a password string, we first match the entire password or a substring of the password to its personal information. The matching algorithm is similar to that in Section II-B2. However, this time we also record the length of the matching segment. We replace the matched segments in the password with corresponding symbols and mark the symbols with length. Unmatched segments remain unchanged. For instance, we assume Alice was born in August 16, 1988 and her password is "helloalice816!". The matching phase will replace "alice" with "N_5" and "816" with

"B_3". The leftover "hello" is kept unchanged. Therefore the outcome of this phase is "$helloN_5B_3$!".

2) Password Pre-processing: This phase is similar to the pre-processing routine of the original PCFG; however, based on the output of the personal information matching phase, the segments already matched to personal information will not be processed. For instance, the sample structure "$helloN_5B_3$!" will be updated to "$L_5N_5B_3S_1$" in this phase. Now the password is fully described by semantic symbols of Personal-PCFG, and the output in this phase provides base structures for Personal-PCFG.

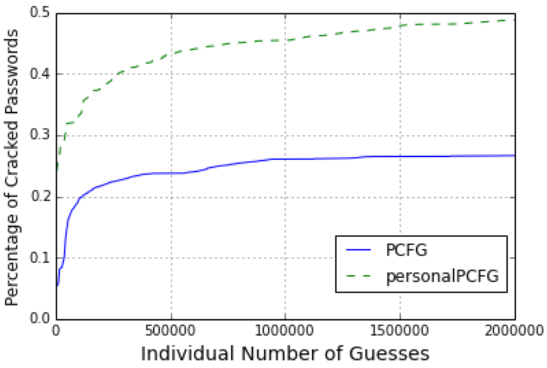
3) Guess Generation: Similar to the original PCFG, we replace "D" and "S" symbols with actual strings learned from the training set in descending probability order. "L" symbols are replaced with words from a dictionary. Similar to PCFG [13], we output the results on the fly so we do not need to wait for all the possible guesses being calculated and sorted. Note that we have not replaced any symbols for personal information so the guesses are still not actual guesses. We do not handle personal information in this step, since personal information for each user is different and personal information symbols can only be substituted until the target is specific. Therefore, in this phase our base structures only generate pre-terminals, which are partial guesses that contain part of actual guesses and part of Personal-PCFG semantic symbols. For instance, the example "$L_5N_5B_3S_1$" is instantiated to "$helloN_5B_3$!" if "hello" is the first 5-symbol-long string in the input dictionary and "!" has the highest probability of occurrence among 1 symbol special character in the training set. Note that for "L" segments, each word of the same length has the same probability. The probability of "hello" is simply $\frac{1}{N}$, in which N is the total number of words of length 5 in the input dictionary.

4) Adaptive Substitution: In the original PCFG, the output of guess generation can be applied to any target user. However, in Personal-PCFG, the guesses will be further instantiated with personal information, which are specific to only one target user. Each personal information symbol is replaced by corresponding personal information of the same length. If there are multiple candidates of the same length, all of them will be included for trial. In our example "$helloN_5B_3$!", "N_5" will be directly replaced by "alice". However, since "B_3" has many candidate segments and any length 3 substring of "19880816" may be a candidate, the guesses include all substrings, such as "helloalice198!", "helloalice988!", . . ., "helloalice816!". We then try these candidate guesses one by one until we find out that one candidate matches exactly the password of Alice. Note that instead of having multiple candidates, not all personal information segments can be replaced because same length segments may not always be available. For instance, a pre-terminal structure "$helloN_6B_3$!" is not suitable for Alice since her name is at most 5 symbols long. In this case, no guesses from this structure should be generated for Alice.

D. Cracking Results

We compare the performance of Personal-PCFG and the original PCFG using the 12306 dataset, which has 131,389 users. We use half of the dataset as the training set, and the other half as the testing set. For the "L" segments, both methods need to use a dictionary, which is critical for password cracking. To eliminate the effect of an unfair dictionary selection, we use "perfect" dictionaries in both methods. Perfect dictionaries are dictionaries we collected directly from the testing set, so that any string in the dictionary is useful and any letter segments in the passwords must appear in the dictionary. Thus, a perfect dictionary is a guarantee to find correct alpha strings efficiently. In our study, both PCFG perfect dictionary and Personal-PCFG perfect dictionary contain 15,000 to 17,000 entries.

Fig. 2: PCFG vs. Personal-PCFG (Offline).

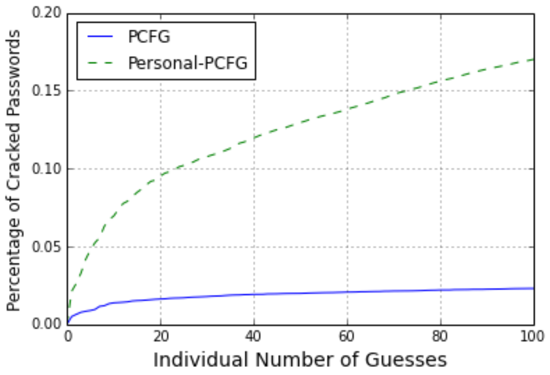


We use *individual number of guesses* to measure the effectiveness of Personal-PCFG and compare with PCFG. The individual number of guesses is defined as the number of password guesses generated for cracking each individual account, e.g., 10 guess trials for each individual account, which is independent of the password dataset size. In Personal-PCFG, the aggregated individual number of guesses (i.e., the total number of guesses) is linearly dependent on the password dataset size. By contrast, in conventional cracking strategy like PCFG, each guess is applied to the entire user base and thus the individual number of guesses equals the total number of guesses. Regardless of such discrepancy between Personal-PCFG and conventional cracking methods, the performance bottleneck of password cracking lies in the large number of hash operations. Due to the salt mechanism, the total number of hashes is bounded by $G \cdot N$ for both Personal-PCFG and other password crackers, where G is the individual number of guesses and N is the size of the dataset.

Given different individual number of guesses, we compute the percentage of those cracked passwords in the entire password trial set. Figure 2 shows the comparison result of the original PCFG and Personal-PCFG in an offline attack. Both methods have a quick start because they always try high probability guesses first. Figure 2 clearly indicates that Personal-PCFG can crack passwords much faster than PCFG does. For example, with a moderate size of 500,000 guesses, Personal-PCFG achieves a similar success rate that can be reached with more than 200 million guesses by the original PCFG. Moreover, Personal-PCFG is able to cover a larger password space than PCFG because personal information provides rich personalized strings that may not appear in the dictionaries or training set.

Personal-PCFG not only improves the cracking efficiency in offline attacks, but also increases the guessing success rate in online attacks. Online attacks are only able to try a small number of guesses in a certain time period due to the system

Fig. 3: PCFG vs. Personal-PCFG (Online).



constraints on the login attempts. Thus, we limit the number of guesses to be at most 100 for each target account. We present the results in Figure 3, illustrating that Personal-PCFG is able to crack 309% to 634% more passwords than the original PCFG. We then show several representative guessing numbers in Figure 4. For a typical system that allows 5 attempts to input the correct passwords, Personal-PCFG is able to crack 4.8% of passwords within only 5 guesses. Meanwhile, the percentage is just 0.9% for the original PCFG, and it takes around 2,000 more guesses for PCFG to reach a success rate of 4.8%. Thus, Personal-PCFG is more efficient to crack the passwords within a small number of guesses.

Therefore, Personal-PCFG substantially outperforms PCFG in both online and offline attacks, due to the integration of personal information into password guessing. The extra requirement of Personal-PCFG on personal information can be satisfied by knowing the victim personally or searching on social networking sites (SNS).

V. Discussion

A. Limitations

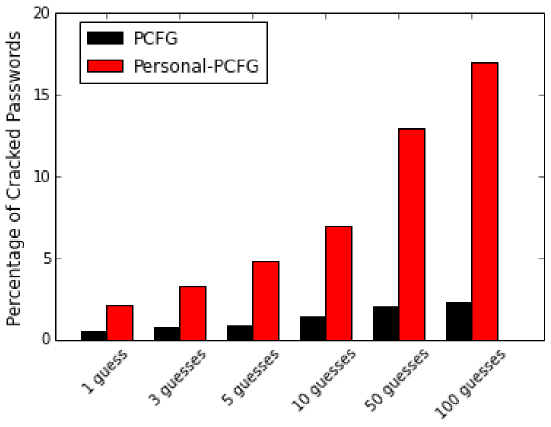
Only a single dataset is used in this study. Most users of the 12306 website are Chinese, and the numbers of males and females are not balanced. Thus, there might be cultural, language, and gender biases on the analytical results. Moreover, the effectiveness of the Coverage metric and Personal-PCFG is merely validated on a single website. However, the publicly available password datasets leaked with personal information are very rare. To extend this work, we plan to derive personal information from multiple leaked password datasets in the future.

B. Potential Defenses

Using a password manager can mitigate this problem as users do not need to remember individual site passwords. In the semi-automatic creation of those site passwords, much more randomness is introduced and much less personal information is involved. However, the master password of a user still remains vulnerable to Personal-PCFG.

One easy way to mitigate personal information correlation in password creation is to mentally distort a password by users themselves. Applying simple distortion on existing passwords can easily break personal information integrity and continuation. Such distortion can be selected by users as simple as adding an extra symbol (e.g., letter, number, or special character) between each pair of existing password letters/numbers. We have observed that this simple distortion is able to significantly reduce the value of Coverage (i.e., the personal information correlation in a password), and Personal-PCFG becomes ineffective. Even if an attacker knows that users distort their passwords, it is still hard to successfully crack a password due to the diverse ways of distortion and the increasing difficulty of learning a personal information pattern in passwords. There are also other solutions to mitigate personal information in user passwords, such as personal-information-aware password meters mentioned in Section III-C. However, the efficacy of these defense methods need to be fully validated through rigorous security analysis and a user study, which is left as our future work.

Fig. 4: Representative Points (Online).



VI. Related Work

Researchers have done brilliant work on measuring real-life passwords. In one of the earliest works [9], Morris and Thompson found that passwords are quite simple and thus are vulnerable to dictionary attacks. Malone et al. [3] studied the distribution of passwords on several large leaked datasets and found that user passwords fit Zipf distribution well. Gaw and Felton [19] showed how users manage their passwords. Mazurek et al. [15] measured 25,000 passwords from a university and revealed correlation between demographic or other factors, such as gender and field of study. Bonneau [2] studied language effect on user passwords from over 70 million passwords. Through measuring the guessability of 4-digit PINs on over 1,100 banking customers [20], Bonneau et al. found that birthdate appears extensively in 4-digit PINs. Li et al. [10] conducted a large-scale measurement study on Chinese passwords, in which over 100 million real-life passwords are studied and differences between passwords in Chinese and other languages are presented.

There are several works investigating specific aspects of passwords. Yan et al. [6] and Kuo et al. [21] investigated mnemonic-based passwords. Veras et al. [5] showed the importance of date in passwords. Das et al. [22] studied how users mangle one password for different sites. Schweitzer et al. [11] studied the keyboard pattern in passwords. Besides the password itself, research has been done on human habits and psychology towards password security [23].

It has been shown that NIST entropy cannot accurately describe the security of passwords [24]. The α-guesswork and the β-success rate used by Bonneau et al [2], [20] are considered to be more accurate metrics to measure password database strength. These metrics are also used by other researchers [10].

Password cracking has been studied for more than three decades. Attackers usually attempt to recover passwords from a hashed password database. Though reversing hash function is infeasible, early works found that passwords are vulnerable to dictionary attacks [9]. However, in recent years as password policies become more strict, simple dictionary passwords are less common. Narayanan and Shmatikov [4] used the Markov model to generate guesses based on the fact that passwords need to be phonetically similar to users' native languages. In 2009, Weir et al. [13] leveraged Probabilistic Context-Free Grammars (PCFG) to crack passwords. Veras et al. [12] tried to use semantic patterns in passwords. OMEN+ [25] improves the Markov model [4] to crack passwords. It even includes experiments to prove usefulness of personal information in password cracking. However, their experiments are in a much smaller scope based on the Markov model, and the improvement is limited.

There has been research on protecting passwords by enforcing users to select more secure passwords, among which password strength meters seem to be one effective method. Castelluccia et al [26] proposed to use the Markov model as in [4] to measure the security of user passwords. Meanwhile, commercial password meters adopted by popular websites have proved inconsistent [7]. There are works focusing on providing feedback to users using trained leaked passwords or dictionaries [14], [16].

VII. Conclusion

In this paper, we conduct a comprehensive quantitative study on how user personal information resides in human-chosen passwords. To the best of our knowledge, we are the first to systematically analyze personal information in passwords. We have some interesting and quantitative discovery, such as that 3.42% of users in the 12306 dataset use birthdate in passwords, and male users are more likely to include their name in passwords than female users. We then introduce a new metric, Coverage, to accurately quantify the correlation between personal information and a password. Our Coverage-based quantification results further confirm our disclosure on the serious involvement of personal information in password creation, which makes a user password more vulnerable to a targeted password cracking. We develop Personal-PCFG based on PCFG but consider more semantic symbols for cracking a password. Personal-PCFG generates personalized password guesses by integrating user personal information into the guesses. Our experimental results demonstrate that Personal-PCFG is significantly faster than PCFG in password cracking and eases the feasibility of mounting online attacks. Finally, we discuss the limitation of this work and solutions to prevent weak passwords that include personal information.

Acknowledgments

This work is partially supported by U.S. ARO grant W911NF-15-1-0287, and ONR grants N00014-15-1-2396 and N00014-15-1-2012.

References

[1] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *IEEE Security & Privacy*, 2012.

[2] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *IEEE Security & Privacy*, 2012.

[3] D. Malone and K. Maher, "Investigating the distribution of password choices," in *ACM WWW*, 2012.

[4] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *ACM CCS*, 2005.

[5] R. Veras, J. Thorpe, and C. Collins, "Visualizing semantics in passwords: The role of dates," in *IEEE VizSec*, 2012.

[6] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: Empirical results," *IEEE Security & Privacy Magazine*, 2004.

[7] X. de Carné de Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *NDSS*, 2014.

[8] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, "Does my password go up to eleven?: the impact of password meters on password selection," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013.

[9] R. Morris and K. Thompson, "Password security: A case history," *Communications of the ACM*, 1979.

[10] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis of chinese web passwords," in *Proc. USENIX Security*, 2014.

[11] D. Schweitzer, J. Boleng, C. Hughes, and L. Murphy, "Visualizing keyboard pattern passwords," in *IEEE VizSec*, 2009.

[12] R. Veras, C. Collins, and J. Thorpe, "On the semantic patterns of passwords and their security impact," in *NDSS*, 2014.

[13] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *IEEE Security & Privacy*, 2009.

[14] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *ACM CCS*, 2010.

[15] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur, "Measuring password guessability for an entire university," in *ACM CCS*, 2013.

[16] S. Komanduri, R. Shay, L. F. Cranor, C. Herley, and S. Schechter, "Telepathwords: Preventing weak passwords by reading users' minds," in *USENIX Security*, 2014.

[17] R. Gross and A. Acquisti, "Information revelation and privacy in online social networks," in *ACM WPES*, 2005.

[18] B. Krishnamurthy and C. E. Wills, "On the leakage of personally identifiable information via online social networks," in *ACM COSN*, 2009.

[19] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *ACM SOUPS*, 2006.

[20] J. Bonneau, S. Preibusch, and R. Anderson, "A birthday present every eleven wallets? the security of customer-chosen banking pins," in *Financial Cryptography and Data Security*. Springer, 2012.

[21] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *ACM SOUPS*, 2006.

[22] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in *NDSS*, 2014.

[23] D. Florencio and C. Herley, "A large-scale study of web password habits," in *ACM WWW*, 2007.

[24] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. Lopez, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *IEEE Security & Privacy*, 2012.

[25] C. Castelluccia, A. Chaabane, M. Dürmuth, and D. Perito, "When privacy meets security: Leveraging personal information for password cracking," *arXiv preprint arXiv:1304.6584*, 2013.

[26] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from markov models." in *NDSS*, 2012.