ATLANTIS PRESS

Advances in Computer Science Research, volume 80

International Conference on Computer Science, Electronics and Communication Engineering (CSECE 2018)

A Novel Weak Deduction Password Strength Meter of Chinese Websites

Yu Wu*

Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China

*Corresponding author

***Abstract*—With the increasing situation of passwords datasets disclosure, the security of user's account faces rising risks under various kinds of password attack. To protect the security of user authentication system, password strength meter (PSM), which is used to measure user's password strength is widely accepted by Internet service institution. However, traditional PSMs in use today are rarely consider about user's cross-site vulnerable behaviors. In this paper, we first conduct a large scale of password reuse behaviors from 200 million Chinese websites' passwords. Then we propose our new weak-deduction PSM (WDPSM) that inherits the advantage of traditional probabilistic context free methods. Also, our WDPSM is good at model users' cross-site behaviors. Specifically, we perform a series of experiments to show that our weak-deduction PSM outperforms traditional PSMs in gauging weak passwords.**

Keywords—password strength meter; probabilistic context free grammar; weak deduction; rank correlation

I. INTRODUCTION

The password authentication system is regarded as the important method to protect system security and is widely used in various kinds of online service. In the past few years, a large-scale of password information have been leaked leading to serious threats to user's accounts security. In order to solve this problem, most of the majority of websites provide PSM to evaluate the user's password security. The password selected by the user cannot be received by the system unless it reaches a certain threshold. Hence, enhancing the accuracy becomes an important indicator in designing a PSM.

Burr et al. proposed a rule-based PSM so called NIST in 2006 [1]. After that, in nearly decades, a mass of PSM have been proposed which follow the spirit of NIST such as Google, Microsoft PSM [2]. NIST PSM suggest evaluating password strength in terms of guessing space (e.g., 6 bits entropy added if passwords have both uppercase letter and non-alphabetic characters, 1.5 bits each added for characters 9 through 20). Later, Carnavalet et al. [3] found that most of these PSM are simple and estimate password based on some heuristic rules by studying a large scale of popular websites' password strength meter.

Kelley et al. [4] suggest that the rule-based PSMs are difficult to provide accurate measurement of password strength for reason of lacking analysis on empirical data. To improve the accuracy of the PSM, Kelley et al. presented a new concept, namely: 'guessability' as the password strength metric. 'Guessability' is a metric that characterizes the guess number needed for an efficient password-cracking algorithm to discover a password.

Most research on 'guesssability' is based on Markov [5] and Probabilistic Context Free Grammar (PCFG) [6] cracking algorithm which are proposed by Narayanan et al. and Weir et al. respectively. Houshmand and Aggarwal [7] used probabilistic context free techniques [6] to analyze password strength (passwords probability). They first developed an analyze-modify system to estimate the probability of password. After that, they modifies the password slightly if it is weak enough. Castelluccia et al. [8] first proposed an adaptive PSM based on the Markov chain probabilistic guessing model, and their PSM is much better than NIST, Google and Microsoft PSM.

Das et al. [9] first conduct a user survey to study user's password reuse behaviors, and they developed the first cross-site password-guessing algorithm so called DBCBW algorithm which is better than the standard guessing algorithm presented by John the Ripper [10]. However, their password reuse study does not based on real-world password datasets and their guessing algorithm has not compared with other academic algorithm.

A. Our Contribution

In our work, we first analyze a large scale of password reuse behaviors based on 9 disclosure Chinese password datasets which contains 200 million passwords in total. By analyzing user's cross-site reuse or slightly modify passwords behaviors, we propose our new WDPSM. We first uses a weak password set collecting from leaked password sets to construct a password bk-tree. Then, we learns how to apply mangle rule and transformation rule in generating new password from similar original password. Finally, we extended traditional probabilistic context free method [5] to model user's password reuse behavior and gives rise to WDPSM.

We conduct a series of experiments to compare our new PSM with traditional PSMs, e.g. NIST and PCFG PSM. The result show that our WDPSM performs better in judge weak passwords.

B. Paper Organization

We introduce our password datasets with basic data analysis in section 2. Our WDPSM is proposed in section 3. In section 4 we present the comparison between various existing PSMs with WDPSM and we conclude our work in section 5.

II. PASSWORD DATASET ANALYSIS

In this section, we analyze nine representative Chinese password databases in two patterns: basic password statistics analysis and password reuse analysis. We collected nine famous Chinese password datasets which contains 203 million valid password in total. Table I shows their basic informationMaintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

TABLE I. NINE CHINESE PASSWORD DATASETS

Datasets	Total PWs	Unique PWs	When Leaked
CSDN	6,414,425	4,026,595	2011.1
163	117,602,494	21,730,096	2011, 2014
Dodonew	16,020,739	9,989,007	2011
Tianya	29,010,375	12,817,411	2011.12
Renren	4,681,142	2,800,426	2014
Zhenai	5,236,113	3,503,765	2014
17173	9,956,882	3,629,363	2011.12
7k7k	9,435,506	4,887,257	2011
Weibo	4,942,426	2,825,096	2011.12
Total	203,300,102	66,326,824	

A. Basic Password Statistics Analysis

1) The most popular passwords: We analyze the nine datasets of Chinese websites and find that the most popular password is '123456' in most of the datasets. In addition, the most popular password is '123456789' in CSDN.

2) Length distribution: We analyze the nine datasets of Chinese websites and find that the most common used password length are among 6 ~ 11. Dodonew, weibo and renren have some password length less than 6.

3) Structure distribution: To understand the constitution of password in various website. We separate the character consist of passwords into four classes: L represent lower case letter, U represent upper case letter, D represent digits, S represent special symbol. Specially, for example, LD means a password with letter and digits. We analyze the nine datasets of Chinese websites and find that the top three most popular password structure is D-only, LD and L-only. Surprisingly, about half of users around most website use D-only password.

B. Password Reuse Analysis

In this section, we start to learn user's password reuse. We define the behavior that one user use identical password or similar password from different websites as password reuse. In our work, we perform the first systematic assessment on password reuse based on a large-quantity of Chinese passwords datasets.

Before we describe the password reuse analysis, we split password reuse into 7 rules as follow. **Identical rule:** this rule means one password is same with another. **Substring rule:** this rule means one password is a substring of another one (e.g., wuyu and wuyu123). **Longest Common Substring (LCS) rule:** LCS rule implies the majority of two passwords are same, but one is not another one's substring. (e.g., wuyu123 and wuyu@123). **Uppercase rules:** this rules means the pair of two password is not same but upper case is. **Leet rule:** this rule uses some alphabetic character to replace the similar-looking character. (e.g., 0↔o, a↔@,1↔I, e↔3,5↔s, s↔$, password ↔p@ssword). **Reverse rule:** this rule means one password is the reverse form of another password. **Other rule:** the rest of passwords are classified as the other rule.

We analyze the password reuse by following steps:

- Removing invalid password from our datasets to avoid disturbing.
- Intersecting a dataset with another dataset by matching account name or email. This step produces a new datasets with two brother passwords for each account or user.
- Using above mentioned 7 rules to learn how do user use one password of brother pair to produce another one.

We use our password reuse analyze method to analyze the nine datasets and the result of the analysis is demonstrated in Table II.

In total, about 62% password pair is same with each other, this result show highly rate than the investigation result Das et al. [8] have done. About 11% password pair is similar to each other. Only 27% password is unmatched.

TABLE II. PASSWORD REUSE ANALYSIS

Transformatin rule	Ratio (%)
Uppercase	0.14%
Identical	62.09%
LCS	2.66%
Leet	0.00%
Substring	7.39%
Reversal	0.00%
Other	27.72%

III. OUR WEAK-DEDUCTION PSM

In this section, we propose our WDPSM which outperforms traditional PSMs in gauging weak passwords. First we give the formal definition of PSM and ideal PSM.

Definition 1 (PSM). A PSM is a function *F(.)* that take a password as an input over an alphabet Σ and outputs a probability *p* (as a number) in range of [0,1]. The probability *p* of a password is inversely proportional to the weakness of it.

Definition 2 (Ideal PSM). An ideal PSM is a function *F(.)* that *F(pw) = P(pw)* where *pw* is a password and *P(pw)* is a real probability of pw in an authentication system.

Through the password reuse analysis of previous section, we find that users have a very high probability of reusing password. This fact is in vast contrast with PCFG-based PSM assumes that users construct new passwords from scratch. To model user's password reuse behavior, we propose our WDPSM to measure passwords strength. The whole process can divide into three phases: **Training**, **Measuring** and **Updating**.

First we uses a set of leaked weak passwords as "Weak set" *W* to construct a weak password parsing bk-tree [11]; while using another target website or target website similarity website leaked passwords set as training set *T*. Then we parse password in T and automatically derive password-mangling rules.

Training: we measure the frequencies of certain patterns associated to the training passwords in T. First we assume weak password set $W = \{w_1, w_2, ..., w_N\}$, a password of length *n* will be labeled as W_n. *n* is no longer than the maximum length accepted by target system. Each password in the training set *T* is parsed by the bk-tree using a new algorithm so called optimal longest common subsequence matching algorithm (OLCS). We will describe OLCS algorithm in **Algorithm 1.** If there exists some parts of *pw* which is unmatchable, we match the unmatchable part in the original PCFG method.

Algorithm 1 Optimal longest common subsequence matching algorithm (OLCS)
Input: *pw*, *W*, *D* (the distance threshold), *S* (the similarity threshold).
Output: *omv* (the optimal match value).
1) *optimal* = {}
2) *substring*(*pw*) = {all substring of *pw*}
3) for *w* in *W*
4) for *sub* in *substring*(*pw*)
5) *lcs* = *LCS*(*sub*,*pw*)
6) *dist* = *distance*(*w*, *pw*)
7) *smil* = *smilarity*(*w*,*pw*)
8) *optimal.add*((*w*,*lcs*) if *length* <= *D* and *score* < *S*)
9) *optimal* = *optimal.select*(w if *min*(*dist*))
10) *optimal* = *optimal.select*(w if *max*(*smil*))
11) *omv* = *optimal.select_one*(w if max(*P*[*w*]))
12) return *omv*

Step 1 to 7 will get calculate all possible string pair with their distance score, similarity score and longest common subsequence. Step 8 will get a minimum distance score string pair set. Step 9 will get all string pair with maximum similarity score. Step 10 to 11 return the optimal match with high probability. Specifically, we choose select Levenshtein-distance as distance function, choose Levenshitein-smilarity as similarity function in **Algorithm 1**.

Our context free grammar is defined as $G=(V, \Sigma, S, P)$ where $V=\{S,L,D,S, insert, delete, replace, W_1, W_2,...,W_n\}$ is a finite set of variables, *S* is the start variable, Σ={95printable ASCII characters}is a finite set of terminals and *P* is finite set of productions of the form $\alpha \rightarrow \beta$,with $\alpha \in V$ and $\beta \in V \cup \Sigma$.

For example, 123.456abc $\in T$, 123456 $\in W$, OLCS would return (123.456,123456). The result tell us 123.456 is mostly similar with Weak password 123456, so 123.456 would be parsed into W_6 and the remain part 'abc' is matched by original PCFG match. As a result, 'abc' will be parsed into L_3. Then the whole password pattern is represented as W_6L_3. In total, the whole parsing phase is similar with PCFG approach.

From our training set *T*, we derive a set of productions that generate the base structures, base segments (including weak passwords and other segments) with their associated probabilities of occurrence. Note that, *G* is a probabilistic context-free grammar because for a specific *LHS* variable all the associated productions add up to 1.

TABLE III. EXAMPLE OF EXTENDED GRAMMAR

LHS	RHS	probability
S	W6	1
W6	no	0.7
W6	insert	0.3
insert	D1	0.8
insert	S1	0.2
D1	0	1
S1	.	0.75
S1	+	0.25

TABLE IV. EXAMPLE OF PROBABILISTIC CONTEXT FREE GRAMMAR

LHS	RHS	probability
S	W6	1
W6	123456	0.7
W6	000000	0.3

Measuring: The Grammar *G* would be used to measure passwords probability in measuring phase. An example grammar is show in Table III~IV. The P("123.456")=$P(W_6)$* P(W_6->*insert*) * P(*insert*->S_1) * P(S_1->.) = 0.7 * 0.3 * 0.2 * 0.75 = 0.0315.

Updating: The updating phase can dynamic modifies the grammar with the time go by. For example, while 123.456abc is accepted by the system, all probabilities that related to the base structure W_6 and L_3, transformation rule W_6->*insert and insert*->S_1, and terminals 123456, abc shall be updated. This process is similar with The PSM in [7]. While the probability measured by meter of a valid password is higher than the threshold probability, that password will be add to weak password set. We named the above process that an original weak password deduce a new weak password as Weak

Deduction. Our WDPSM can associated one password with the similar password through weak deduction, which is in vast contrast with the PCFG-based PSM. So we claim that our context free grammar is a weak deduction one.

IV. EXPERIMENT ANALYSIS

A. Experiment Prepare

To model user's password reuse behavior, we use Houshmand and Aggarwal [7] method to collect weak passwords from the weakest dataset as weak password. Tianya datasets is the weakest one [12], so we use it in our experiments. Then we randomly split each dataset into equally four parts, the training set uses part-1 and testing set uses part-2, as [8,13] has done. Table V show how we choose training and testing set for experiment.

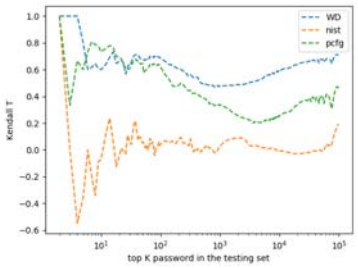
To compare with different password strength meter, we use Kendall coefficient. We will use it in our experimental evaluation.

TABLE V. TRAINING SET AND TESTING SET

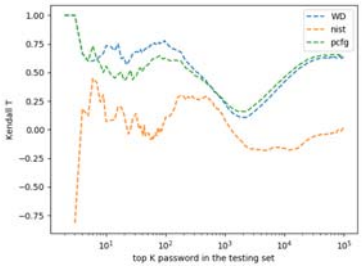
Week password set	Traning set	Testing set
tianya	1/4 weibo	1/4 weibo
tianya	1/4 7k7k	1/4 7k7k
tianya	1/4 tianya	1/4 tianya

B. Result Evaluation.

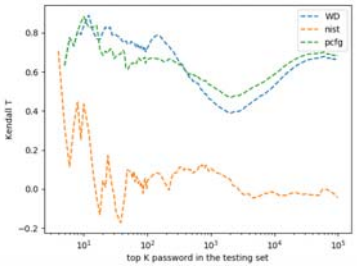
In this section, we compare the performance of our WDPSM with NIST PSM [1] and PCFG-based PSM [7]. The result is show in Figure I. All the tested password meters are measured by computing the Kendall rank correlation scores against with ideal meter. In most cases, DBPSM outperforms other PSMs.



(a) 1/4 weibo training with 1/4 weibo testing



(b) 1/4 7k7k training with 1/4 7k7k testing



(c) 1/4 tianya traning with 1/4 tianya testing

FIGURE I.EXPERIMENT RESULTS OF PSMS ACCURACY ANALYSIS

CONCLUSION

In this paper, we have analyzed 200 millions of passwords and millions of password pairs of Chinese password data. A high percentage (72.29%) of passwords are reused or slightly modified by user for different services. To defense the cross-site password attacking, we proposed a new password strength meter which can characterize user's password creation policy. Experiments show that our WDPSM performs higher accuracy than existing wildly used password strength meter. Although the data used in experiment is from Chinese passwords dataset, the measure method can be used to other language environment either.

REFERENCES

[1] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic authentication guideline: NIST special publication 800-63, 2006.

[2] D. Wang and P. Wang, “The emperor’s new password creation policies,” in Proc. ESORICS 2015, pp. 456–477.

[3] X. Carnavalet and M. Mannan, “A large-scale evaluation of high-impact password strength meters,” ACM Trans. Inform. Syst. Secur., vol. 18, no. 1, pp. 1–32, 2015.

[4] 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 Proc. IEEE S&P 2012, pp. 523–537.

[5] A. Narayanan and V. Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In Proc. ACM CCS 2005, pp. 364–372.

[6] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek. Password cracking using probabilistic context-free grammars. In Proc. IEEE S&P 2009, pp. 391–405.

[7] S. Houshmand and S. Aggarwal, “Building better passwords using probabilistic techniques,” in Proc. ACSAC 2012, pp. 109–118.

[8] C. Castelluccia, M. D¨urmuth, and D. Perito, “Adaptive password strength meters from markov models,” in Proc. NDSS 2012, pp. 1–15.

[9] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang. The tangled web of password reuse. In Proc. NDSS 2014, pp. 1–15.

[10] John the ripper password cracking toolkit. http://www.openwall.com/john/.

[11] BK-tree - Wikipedia. https://en.wikipedia.org/wiki/BK-tree.

[12] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in Proc. IEEE S&P 2014, pp. 689–704.

[13] M. Weir, S. Aggarwal, M. Collins, and H. Stern, “Testing metrics for password creation policies by attacking large sets of revealed passwords,” in Proc. ACM CCS 2010, pp. 162–175.