The Next Domino to Fall: Empirical Analysis of User Passwords across Online Services

Chun Wang, Steve T.K. Jan, Hang Hu, Douglas Bossart, Gang Wang
Department of Computer Science, Virginia Tech, Blacksburg, VA, 24060
{wchun, tekang, hanghu, bossartd, gangwang}@vt.edu

ABSTRACT

Leaked passwords from data breaches can pose a serious threat if users reuse or slightly modify the passwords for other services. With more services getting breached today, there is still a lack of a quantitative understanding of this risk. In this paper, we perform the first large-scale empirical analysis of password reuse and modification patterns using a ground-truth dataset of 28.8 million users and their 61.5 million passwords in 107 services over 8 years. We find that password reuse and modification is very common (observed on 52% of the users). Sensitive online services such as shopping websites and email services received the most reused and modified passwords. We also observe that users would still reuse the already-leaked passwords for other online services for years after the initial data breach. Finally, to quantify the security risks, we develop a new training-based guessing algorithm. We show that more than 16 million password pairs (including 30% of the modified passwords) can be cracked within just 10 guesses.

CCS CONCEPTS

• **Security and privacy** → **Authentication**; *Systems security*; • **Human-centered computing** → *Human computer interaction (HCI)*; • **Computing methodologies** → Machine learning;

KEYWORDS

Password Reuse; Empirical Measurements; Bayesian Model

1 INTRODUCTION

Today's data breaches (*e.g.*, Equifax, Yahoo, Myspace, Office of Personnel Management, Ashley Madison) are reaching unprecedented scale and coverage. In 2016 alone, there were more than 2000 confirmed breaches causing a leakage of billions of user records [37]. Many of the leaked datasets contain sensitive information such as *user passwords*, which are often made publicly available on the Internet by the attackers [25, 26, 29, 31, 42].

Leaked passwords can pose serious threats to users, particularly if the passwords are reused somewhere else by the users. Reusing the *same* or even slightly *modified* passwords allows attackers to further compromise the user's accounts in other unbreached services [23, 28]. Even worse, if the target user happened to be the administrator of another service, password reuse may lead to new massive data breaches (*e.g.*, Dropbox [4]).

With more and more passwords being leaked [8, 38], there is an urgent need to systematically assess users' password reuse and modification patterns and quantify the security risks. This is not only instrumental to protecting user accounts after data breaches, but can also help to develop more effective tools to manage users' passwords. Due to a lack of large-scale empirical data, most existing works rely on surveys or interviews to study password reuse [7, 14, 30, 32, 34, 40]. The problem is that user studies are often limited in scale (*e.g.*, a few hundred users), and users' self-reported results may contradict their actual behavior in practice [40].

Recently, researchers start to analyze empirical data to understand users' password reuse and modification patterns [5, 6, 24, 40, 43]. However, the scale of existing empirical studies is still very limited. The largest study so far that focuses on both password reuse and modification only covers 6,077 users [5]. The limited scope of the dataset (sample size, service type, user demographics) makes it challenging to examine the generalizability of the observations and quantify the actual security risks.

In this paper, we seek to fill in the gaps by gathering and analyzing a large collection of leaked password datasets across multiple years and various online services[1]. By linking the userID (*i.e.*, email address) in different password datasets, we construct a ground-truth mapping for the same users' passwords and study their reuse and modification patterns. The resulting ground-truth dataset contains 28,836,775 users and their 61,552,446 passwords from 107 online services across 8 years.

Our study has two goals. 1) We seek to empirically understand how users reuse and modify their passwords across online services at a *large-scale*. 2) We want to quantify the security risks introduced by password reuse and modifications after data breaches. To achieve these goals, we have addressed a number of technical challenges. First, while password reuse is easy to determine, password modification is not obvious. To this end, we develop a measurement framework to automatically determine whether two passwords are modified from each other, and extract the transformation rules. This framework enables a deeper analysis of users' password habits and cross-examining our results with the existing small-scale user studies. Second, we develop a new *training-based* password guessing algorithm to guess a target user's password based on her leaked ones. We empirically examine the possibility of password guessing in an *online* fashion. We have a number of key findings:

1. Password reuse and modification are still very common. Among the 28.8 million users, 38% have once reused the same password in two different services and 21% once modified an existing password to sign up a new service (52% collectively). In addition,

[1] Our study has received IRB approval (Protocol #17-393).

CODASPY '18, March 19–21, 2018, Tempe, AZ, USA

ACM ISBN 978-1-4503-5632-9/18/03…$15.00
https://doi.org/10.1145/3176258.3176332

we find that users with more total passwords are more likely to reuse/modify passwords. The reused/modified passwords are statistically shorter but more complex. These results echo and help to confirm early findings of small-scale user studies [24, 40].

2. Sensitive online services have a high ratio of reused and modified passwords. A surprising new finding is that "shopping" services have the highest ratio (>85%) of reused and modified passwords, while "email" services are at the second place (>62%). Shopping services often store users' credit card information and home address, and thus reusing their passwords have key security implications. The problem with email services can be even more serious, given that attackers can use the email address to reset the user's passwords in other accounts (*e.g.*, online banking).

3. Users still reuse the already-leaked passwords for years after the data breach. We find a long delay before users change their already leaked passwords *in other services*. More than 70% of the users are still reusing the already-leaked passwords in other services 1 year after the leakage. 40% of the users are reusing the same passwords leaked more than 3 years ago. This indicates a persistent threat of the leaked passwords from data breaches.

4. Modified passwords are highly predictable. Among a large user population, there is only a small set of rules that users often apply to modify their passwords. Such "low variance" makes the modified passwords highly predictable. Our training-based algorithm can guess 30% of the modified passwords within 10 attempts (46.5% within 100 attempts). If we consider both the reused and modified passwords, we estimate that more than 16 million password pairs in our dataset can be cracked within 10 guesses. Our algorithm achieves a similar performance even if it is trained with only 0.1% of the data.

In summary, our work makes 3 key contributions.

- We perform the first large-scale empirical analysis on password reuse and modification behavior across online services (28.8 million users, 107 online services). Our analysis provides new insights into how user reuse and modify passwords in practice.
- We develop a new training-based password guessing algorithm to quantify the risk of password modification. Our algorithm can guess a large portion of modified passwords within 10 guesses.
- To facilitate future research, we share the dataset with the research community with carefully designed data sharing policies.

2 RELATED WORK

Password Reuse and Modification. Text-based password is still the primary authentication method for today's online services. Due to the difficulties of memorizing a large number of passwords, users often *reuse* the same passwords or slightly *modify* existing passwords when creating new ones [5, 7, 40]. Attackers may leverage the reused passwords to compromise new user accounts, or link user identities by mining the leaked password datasets [22].

Table 1 lists the key related works on password reuse and modification. On one hand, due to a lack of empirical datasets, most existing works rely on user surveys or interviews to understand password usage [5, 7, 14, 21, 30, 32, 34, 40]. For example, Das et al. [5] have reported that 51% of the users re-use passwords across online services. Stobert and Biddle's interview [32] suggests that password reuse often happens on "less important" services.

	PW Reuse	PW Modify	Methods	# Users
[21]	✓	×	Survey	26
[32]	✓	×	Survey	27
[40]	✓	×	Empirical+Survey	134
[6]	✓	×	Empirical	544,960
[7]	×	✓	Survey	80
[30]	×	✓	Survey	470
[43]	×	✓	Empirical	7,700
[34]	✓	✓	Survey	49
[24]	✓	✓	Empirical+Survey	154
[14]	✓	✓	Survey	5,000
[5]	✓	✓	Empirical+Survey	6,077
Our	✓	✓	Empirical	28,836,775

Table 1: Related works on password reuse and modification.

Inevitably, user studies suffer from key limitations due to the small user population. A recent work also shows that user self-reported results may contradict their real behavior in practice [40]. To these ends, empirical analysis is needed to understand users' real-world behavior [5, 6, 24, 40, 43]. To date, existing empirical studies are still limited in scale, most of which only cover a few hundred (or a few thousand) users. The only exception is a measurement study [6] conducted 10 years ago by Microsoft (500K users), which, however, only analyzed password reuse not password modification across services. In our work, we seek to fill in the gap by collecting and analyzing a large-scale empirical password dataset (61.5 million passwords across 107 services). We focus on both *password reuse* and *modification*, and cross-examine our results with early findings from small-scale studies.

Online & Offline Password Guessing. Another related body of work is password guessing, which can be roughly divided into online guessing and offline guessing. Online guessing has a strict limit on the number of guessing attempts. For example, Trawling based approach simply guesses the most popular passwords chosen by users [18]. More targeted guessing exploits the fact that users may reuse the same or similar passwords [5, 43]. More recently, target guessing also incorporates users' personal information such as name and birthday [15, 39].

Offline guessing can easily reach trillions of guessing attempts [9, 10, 12, 19, 20, 36, 41]. A common scenario is to use offline guessing algorithms to recover plaintext passwords from a hashed password dataset. Over the last decades, a number of guessing methods have been proposed, including Markov Model [16, 20], Mangled Wordlist method [35], Probabilistic Context-Free Grammars Method (PCFGs) [12, 20, 36, 41], and Deep Neural Networks [19]. Offline guessing has also been used to measure password strength [13].

3 DATASET

To study password usage across online services, we gathered a large number of password datasets and linked the same user's passwords.

Data Collection In January 2017, we searched through various online forums and data archives for *public* password datasets. We looked for candidate datasets that meet two criteria. First, the dataset should contain email addresses so that we can link a user's passwords across different services. Second, we exclude datasets

Category	#Plain PWs (#Datasets)	Top 3 Largest Datasets
Social	286M (7)	Myspace, VK.com, LinkedIn
Adult	75.2M (9)	Zoosk, Mate1, YouPorn
Game	40.8M (13)	Neopets, 7k7k, Lbsg
Entertain	30.7M (4)	Lastfm, Swingbrasileiro, LATimes
Internet	16.4M (18)	000webhost, Comcast, Yahoo
Email	9.6M (3)	Gmail, Mail.ru, Yandex
Forum	1.1M (25)	CrackingForum, Abusewith.us, Gawker
Shopping	340K (12)	RedBox, 1394store, Myaribags
Others	210K (7)	Data1, Data2, Data3
Business	10K (9)	Movatiathletic, Hrsupporten, 99Fame
Total	460M (107)	Myspace, VK, LinkedIn

Table 2: Categories and statistics of the collected datasets.

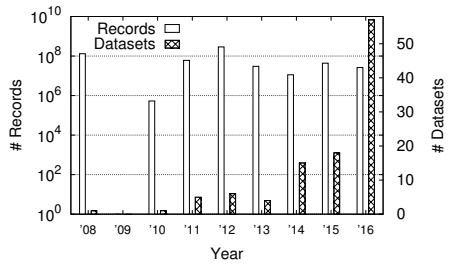


Figure 1: # of datasets and total # records per year.

that only contain *salted hashes* since it is difficult to recover their passwords. In total, we collected 107 datasets leaked between 2008–2016, which contain 497,789,976 passwords and 428,199,842 unique users (email addresses). 14 datasets contain hashed passwords, and we spent a week to recover the plaintext using offline guessing tools [10, 11, 35]. This effort returned 460,874,306 plaintext passwords (93% of all passwords). We have carefully checked each dataset to make sure there are no duplicate records.

Data Statistics. In Table 2, we manually classify the 107 online services into 10 categories based on their category information in Alexa[2]. The "unknown" category contains 7 password datasets with no information about their leakage source. We double checked to make sure the 7 "unknown" datasets did not overlap with any existing ones. The password datasets vary in size. Large datasets from *LinkedIn* and *Myspace* contain hundreds of millions of records, while small datasets such as *InternetFamous* only have a few hundred records. Note that the password dataset may not cover the entire leaked data — attacker might only publish part of the dataset publicly.

We manually label the year *when each dataset was leaked* (excluding "unknown" datasets). We confirm the year of the data breach based on various sources such as reputable news reports and data breach reports [1, 2, 8, 38]. Figure 1 shows the number of datasets and the number of user records in different years. Note that year 2016 covers most of our datasets since it is easier to find datasets that were leaked more recently. For older datasets, they are primarily related to large data breaches.

Ethic Guidelines. Our work involves analyzing leaked datasets that contain sensitive information. We have worked closely with our local IRB and obtained the approval for our research. Our study is motivated by the following considerations. First, we only analyze datasets that are already publicly available. Analyzing such data does not add additional risks other than what already exists. Second, these datasets are also publicly available to potential attackers. Failure to include the data for research may give attackers an advantage over researchers that work on defensive techniques. In the past decades, leaked password datasets have been extensively used in academic research [5, 15, 35, 36, 39] to develop security mechanisms to protect users in the long run.

Primary Dataset (28.8 Million Users) To study cross-site password usage, we focus on users who appear in at least two different services. We construct a *primary* dataset of 28,836,775 users who have at least two plaintext passwords (61,552,446 passwords in total). Note that users outside of the primary dataset are not necessarily risk-free: they might still have accounts in services that we didn't cover. In this study, a *user* is defined by an email address, which helps us to link the same user's passwords together. In practice, it is possible for a person to have multiple email addresses. Our study will only estimate a lower-bound.

4 PASSWORD REUSE & MODIFICATION

Our dataset provides a unique opportunity to study password reuse and modifications across *a large user population* and *a variety of online services*. At the same time, we also seek to cross-compare our results with those from smaller-scale studies [5, 7, 24, 32, 34, 40] to provide a more complete view of this problem.

In the following, we first develop a framework to measure password reuse and modification behavior across online services (current section). Then we use this framework to perform an in-depth analysis to understand how users manage their passwords and generate the statistical patterns of password reuse and modification (Section 5). Finally, we empirically quantify the security risks of password reuse/modifications by performing password guessing experiments (Section 6 and Section 6.2), and discuss the implications of our results to the increasingly frequent data breaches (Section 7).

4.1 Reusing the Same Password

Human brains can only memorize a limited number of passwords, and thus users often reuse their passwords for different online services [6]. To understand the password reuse in practice, we perform a quick measurement on the *primary* dataset. For each user, we cross-examine all the possible password pairs (*e.g.*, if a user has 4 passwords, then we get 6 pairs). In total, we extract 37,301,406 password pairs for the 28.8 million users. We find that 34.3% of the pairs are identical pairs, meaning that the password is reused by the user. At the user level, 38% of the users (10.9 million) have at least one identical pair. This ratio is slightly lower than the self-reported results (51%) from a prior user study [5].

4.2 Classifying Password Modification

In addition to reusing the same password, users may also modify an existing password to sign up for a new service. We refer this type of behavior as *password modification*. Unlike password reuse, password modification is more difficult to measure because users may apply different rules to make the transformation. To this end,

[2]https://www.alexa.com/topsites

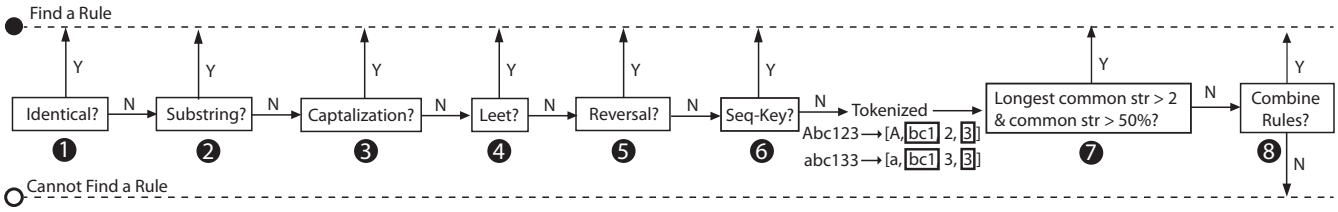


Figure 2: The workflow to measure a user's password transformation patterns.

Rule	# Pairs of Passwords	Ratio (%)
❶. Identical	12,780,722	34.3%
❷. Substring	3,748,258	10.0%
❸. Capitalization	478,233	1.3%
❹. Leet	93,418	0.3%
❺. Reversal	5,938	< 0.1%
❻. Sequential keys	12,118	< 0.1%
❼. Common Substring	2,103,888	5.7%
❽. Combination of Rules	754,393	2.0%
Can Not Find A Rule	17,324,438	46.4%
Total	37,301,406	100%

Table 3: Distribution of password transformation rules.

Insert/Delete Position	Ratio	Inserted/Deleted Length	Ratio
Tail	87.2%	1	48.3%
Head	11.0%	2	28.0%
Both Ends	1.8%	3+	23.7%
Insert/Delete Type	**Ratio**	**Top Inserted/Deleted Str.**	**Ratio**
Digit	74.0%	"1"	24.2%
Letter	17.8%	"2"	4.0%
Combined	4.5 %	"12"	2.1%
Special Char	3.7%	"123"	1.9%

Table 4: Substring rule: insertion/deletion patterns.

Longest Comm. Substring	Ratio	Transformation Rules	Ratio
Letter	63.8%	Substitution	84.7%
Digit	22.0%	Insertion/Deletion	32.4%
Combined	13.7%	Capitalization	3.2%
Special Char	0.5%	Switching Order	2.2%

Table 5: Common substring rule: longest common substring and transformation patterns.

we first develop a method to automatically identify and classify modified passwords.

Given a pair of passwords, our goal is to detect if one password is modified from the other password and infer the rule of the transformation. Figure 2 shows the high-level workflow. In total, we construct 8 rules for password transformation based on our manual examinations of 1000 random password pairs and the results from prior studies [5, 43, 44]. We test these rules against the password pairs in the *primary dataset*, and the results are shown in Table 3.

We find that the majority of the password pairs (55.6%) can be explained by one of the transformation rules. To translate the numbers to the user level, 38% of the users have reused the same password at least once, and 21% of the users have once modified an existing password to create a new one. Collectively, these users count for 52%. Below, we discuss each of the rules in detail.

Identical. For completeness, we consider reusing the same password as one of the rules (12 million password pairs, 34.3%).

Substring. This rule indicates that one password is a substring of the other one (*e.g.*, "abc" and "abc12"). This rule matches 3.7 million password pairs (10%), indicating that users have inserted/deleted a string to/from an existing password to make a new one. As shown in Table 4, most insertions/deletions happened at the tail (87.2%). Most inserted/deleted strings are pure digits (74%) and short (1–2 characters), *e.g.*, "1", "2", and "12".

Capitalization. Users may simply capitalize certain letters in a password. Even though the ratio of matched pairs is not high (1.3%), the absolute number is still significant (478,233 pairs). We observe that users commonly capitalize letters at the beginning of the password (73%), particularly the first letter (68.6%).

Leet. 93,418 password pairs match the leet rule (0.3%) [27]. Leet transformation refers to replacing certain characters with other similar-looking ones. Our analysis shows the top 10 most common transformations are: 0↔o, 1↔i, 3↔e, 4↔a, 1↔!, 1↔l, 5↔s, @↔a, 9↔6, and $↔s. These 10 transformations already cover 96.6% of the leet pairs.

Reversal. Reversal rule is rarely used (5938 pairs, <0.1%), which means reversing the order of the characters in a password, *e.g.*, abcd↔dcba. Intuitively, reversed passwords are hard to memorize.

Sequential Keys. Sequential keys include alphabetically-ordered letters (abcd), sequential numbers (1234) and adjacent keys on the keyboard (qwert, asdfg, !@#$%). The matched pairs (*i.e.*, both passwords are sequential keys) are also below 0.1%.

Common Substrings. When a user modifies an existing password to create a new one, we assume the majority of the password remains the same. As shown in Figure 2, we extract the longest common substrings from the two passwords to learn how they transform the rest parts. To avoid accidental character overlaps, we require the longest common string to be >2 characters, and all the common substrings should cover >50% characters of a password (*i.e.*, the majority). This rule matches 2.1 million password pairs (5.7%). To make sure the thresholds make sense, we manually examine a random sample of 1000 matched pairs. For ethical considerations, we use a script to remove the email addresses before manually looking at the passwords. We find only 44 out of 1000 pairs appear to be accidental overlaps. For example, "fighter51" and "nightfall" share a common substring "ight", but do not look like a password modification case. At this point, we can allow false negatives since we have one more rule to check. Based on the false positive rate (4.4%), we estimate that the common substring rule should match at least 5.4% of all password pairs (lower bound).

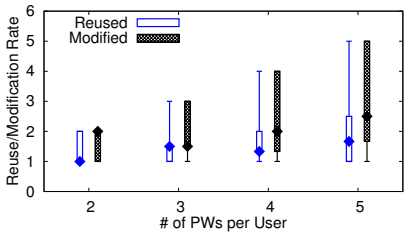


Figure 3: Password reuse/modification rate for users of different # of total passwords. The box plot quantiles are 5%, 25%, 50%, 75%, 95%.

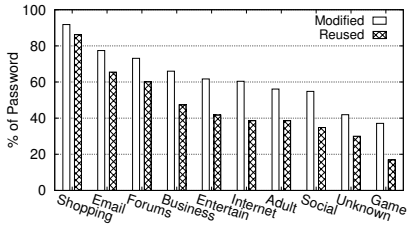


Figure 4: Ratio of reused and modified password under different services. Shopping and email services received the most reused and modified passwords.

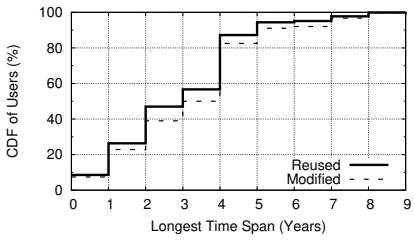


Figure 5: Longest time span between any pair of reused/modified passwords for each user.

Rule Combination	Ratio	Rule Combination	Ratio
Capitalization+Substring	26.2%	Reversal+CSS	6.1%
Leet+CSS	21.8%	Leet+SubString	5.6%
Seqkey+CSS	13.2%	Seqkey+SubString	4.2%
Reversal+Leet+CSS	7.1%	Seqkey+Leet+CSS	2.9%
Capitalization+CSS	6.2%	Others	6.8%

Table 6: Rule combinations (CSS: Common SubString).

Table 5 shows characteristics of the longest common substrings for the matched password pairs. The longest common substring represents the "unmodified" part of the password, most of which are pure letters (63.8%) or pure digits (22%). The majority (56.7%) of the pure-letter substrings are actually English words or English names (based on NLTK corpus [3]). Table 5 shows that the most common transformation is substitution, followed by insertion and deletion. Note that one password pair may have multiple transformations (the accumulated ratio exceeds 100%).

Combination of Rules. As a final step, we combine possible rules to find a match. Note that rule3 – rule6 modify the characters (or the sequence of characters) in a password, while rule2 and rule7 operate on substrings. Our approach is to use a combination of rule3 – rule6 to modify the password first, and then test if rule2 or rule7 can declare a match. In this way, we further matched another 754,000+ pairs (2.0%). As shown in Table 6, "Capitalization" and "Substring" are the most common combination (26.2%), followed by the combination of "Leet" and "Common substring" rules.

Unmatched Password Pairs. After testing all the above rules, there are still 46.4% password pairs remain unmatched. To make sure we did not miss any major rules, we randomly sample 1000 unmatched pairs for manual examination. Through our manual analysis, we did not find any of the 1000 password pairs exhibiting a meaningful transformation. Some example password pairs are: (`samsungi5700`, `nokiae61`), (`phone80720`, `computer7`), (`iloveyou12`, `12081999`), and (`sleepwalker`, `123456`). We regard the remaining 46.4% of password pairs as the result of users "making new passwords from scratch".

5 MEASURING PASSWORD HABITS

Next, we leverage the labeled data to answer key questions about users' password habits. We focus on the *individual users* to explore a series of key questions. *Firs*, how often do users reuse (modify) the same password for different services? *Second*, what types of online services receive the most reused (modified) passwords?*Finally*, how long do users wait before changing their reused passwords in other services after a data breach?

User-level Reuse and Modification Rate. To measure password reuse and modification at the *per-user* level, we calculate a *reuse rate* and a *modification rate* for each user. Given a user u_i, we define her online services as $S_i = [s_{i,1}, s_{i,2}, s_{i,3}, ..., s_{i,K_i}]$, where K_i is user u_i's total number of services. The corresponding passwords are $P_i = [p_{i,1}, p_{i,2}, p_{i,3}, ..., p_{i,K_i}]$. First, *reuse rate* describes how many times a user's password is reused in different services on average. $RR(i) = \frac{|S_i|}{|Set(P_i)|}$. RR=1, if the user sets a unique password for each service. A higher value of RR indicates a more severe password reuse. Second, *modification rate* describes how many times a user's password is reused or modified for different services. $MR(i) = \frac{|S_i|}{|Cluster(P_i)|}$, where $Cluster()$ groups the user's passwords based on whether one password is modified from the other. Based on the transformation rules in Section 4, we group passwords into the same cluster if they match with one of the transformation rules. The resulting clusters don't overlap, representing independent password groups. A higher value of MR indicates more frequent password reuse and modifications.

Figure 3 shows the distribution of RR and MR for users with different numbers of total passwords. We examine whether users with more passwords (*i.e.* online accounts) are more likely to reuse their passwords. The intuition is that a user can only memorize a limited number of passwords. The more online services she has, the more likely her passwords are reused. Our result supports this intuition. As shown in Figure 3, both the reuse rate and modification rate are increasing as users have more total passwords. Our results agree with prior user studies (100+ users) that examine this intuition on *password reuse* [21, 40]. Figure 3 also shows that the black bars (modification) are consistently higher than the blue bars (reuse). This indicates that password modification is broadly applied by users. Analysis that does not consider password modification will under-estimate the security risks.

Impact of Online Services. In Figure 4, we examine what types of online services have received the most reused or modified passwords. We find that "shopping" services have the most reused/-modified passwords with a ratio over 85%. Shopping services usually store users' credit information and home address information.

Reusing passwords of shopping services have key security implications. A possible explanation is that users may have too many accounts for various online stores, making it difficult to memorize a unique password for each one.

More surprisingly, we find that "email services" contain the second-most reused and modified passwords. This result has more serious security implications. First and foremost, an email account can be used to reset the password for other online services (*e.g.*, banking accounts). Many of the online accounts will be in danger if the user's email account is compromised. The ratio of reused email passwords is over 62% and the ratio of modified email passwords is an even higher 78%. Noticeably, our observation contradicts with the results from a prior user study (154 users) [24], which shows that "email" is among categories with the least password reuse.

Delay of Changing Passwords. Finally, we examine the password reuse and modification *across time*. More specifically, we examine how long it takes before users change their reused passwords in other services after data breaches. For example, suppose service A was breached in year t_A and service B was breached later in year t_B. If a user has the same password for both A and B in our dataset, it means this user did not bother to change the reused password for $t_\delta = t_B - t_A$ years. Another interpretation is that the user still signs-up new services using the same password leaked t_δ years ago. For users who have reused/modified passwords, we calculate the largest time-span between her reused/modified password pairs. The result is shown in Figure 5.

Surprisingly, our results indicate that after a service was breached, a large number of users did not reset their reused passwords in *other services* for a long time. More than 70% of the users with reused passwords are still reusing the leaked passwords 1 year after the initial leakage. 40% of users are still reusing the same passwords leaked 3 years ago. Not too surprisingly, slightly modified passwords are continuously used for a longer time than the reused passwords. Our result indicates a persistent threat from reused/modified passwords after data breaches. Attackers may still use the leaked passwords to compromise user accounts in other services after a long time.

6 PASSWORD GUESSING EXPERIMENT

So far, the measurement results suggest that password reuse and modification have potential security risks. Next, we seek to *quantify* the security risks by performing password guessing experiments. In this section, we develop a new *training-based* password guessing algorithm and answer the following key questions. First, how quickly can attackers guess a modified password based on a known one? Second, can attackers use a small training data (*e.g.*, 0.1%) to achieve an effective guessing?

6.1 Guessing Algorithm

We build a new password guessing algorithm to quantify the security risks of password reuse and modification. The algorithm seeks to guess a target user's password by transforming a known password of the same user. The high-level idea is to test different password transformation rules (*e.g.*, rules in Table 3) on the known password. This idea is similar to DBCBW [5], a popular algorithm for targeted password guessing. DBCBW's focuses on *simplicity* which, however, has to make a few compromises. First, due to the lack of training data, the DBCBW uses hand-crafted transformation rules. Second, it tests these rules in a *fixed order*, which may not be optimal for individual passwords. For example, given "`l0ve`", the most probable rule should be leet (`0`→`o`);

18 Features Extracted from a Password
PW (password) length, # Lowercase letters, # Uppercase letters, # Digits, # Special chars, Letter-only pw?, Digit-only pw?, # Repeated chars, Max # consec. letters, Max # consec. digits, Max # sequential keys, Englishword-only pw?, # Consec. digits (head), # Consec. digits (tail), # Consec. letters (head), # Consec. letters (tail), # Consec. special-chars (head), # Consec. special-chars (tail)

Table 7: Feature list of the Bayesian model.

Our algorithm overcomes these drawbacks by introducing a *training phase*. Using ground-truth password pairs, we learn two things: (1) the transformation procedure for each rule, and (2) a model to customize the ordering of the rules for each password.

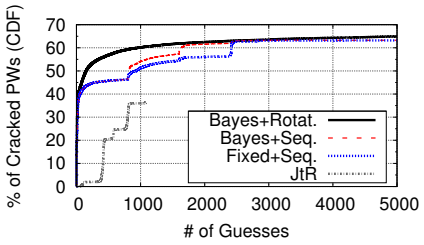
Training 1: Transformation Procedures. A transformation procedure describes how to transform a password to a new one. For each rule in Table 3, we seek to learn a list of possible transformations during the training phase. For each rule R_i, the learned transformation is $T_i = [t_{i1}, t_{i2}, ...t_{iN_i}]$, which is sorted by the frequency of each transformation in the training dataset. For the "substring rule", t is characterized by *<insert/delete><position><string>*. For the "capitalization rule", t is characterized by *<position><#chars>*. In a similar way, we learn the lists of transformations for "leet", "sequential keys" and "reversal". For the "identical" rule, no transformation is needed, and we simply test if the password is reused.

There are special designs for the "common substring" rule and the "combination" rule. For the common substring rule, we can learn and sort the transformations (*e.g.*, insert, delete, replace, substitute, switch orders) based on the training data. However, when applying the transformation to a given password, we need to split the password to detect potential common substrings. In our design, we test 3 types of candidates: (1) substrings of pure digits/letters/special characters, (2) English words/names, and (3) popular common substrings in the *training data*. For the "combined rule", T is a sorted list of rule-combinations where each rule-combination has a sorted list of transformations to be tested.

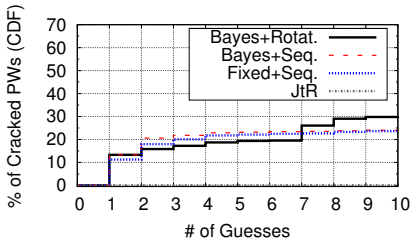
Training 2: Rule Ordering. For a given password, we also learn which rule should be applied first. We treat this as a multiple-class classification problem. Given a password, we train a model to estimate the likelihood that the password can be transformed by each rule. To achieve a quick training, we choose the Naive Bayes classifier (multinomial model) [17], which produces the *probability* that a data point (password) belong to a class (rule). Based on the probability, we customize the ordering of the rules for this password. Table 7 shows the 18 features used in the Bayesian model.

Password Guessing Method. For a password pair (pw_1, pw_2), we seek to test how many attempts are needed to guess pw_2 by transforming a known pw_1. We first use the Bayesian model to generate a customized order of rules for pw_1. Following the ordered rule list, we have two options for guessing:

- **Sequential**: testing one rule at a time. After testing all the transformations under a rule, we move to the next rule. Since certain rules have a significantly longer list than others, we set a



(a) 5000 Guesses



(b) 10 Guesses

Figure 6: Password guessing with 50% of the data for training.

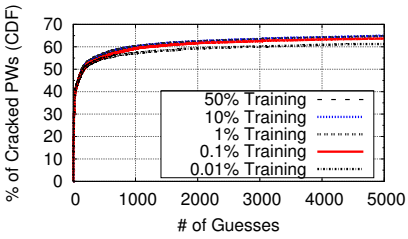


Figure 7: Password guessing with different training data sizes.

threshold M as the maximum number of guesses under each rule ($M = 800$ for our experiment).

- **Rotational:** testing one rule and one transformation at a time. After testing one transformation under a rule, we move to the next rule to test another transformation. We rotate to test each rule for just one guess.

Note that sequential guessing requires a higher accuracy of the predicted order. If the predicted order is wrong, it will waste many guesses on the wrong rule before moving on.

Baselines. When choosing baselines, we ruled out algorithms that don't fit our threat model. First, we rule out non-targeted guessing algorithms [19, 35], since non-targeted algorithms are primarily for offline guessing (*e.g.*, 10^{12} guesses needed). Second, we also rule out targeted guessing algorithms that require the user's PII information (*e.g.*, real name, date of birth) [15, 39]. Such PII information is not available in our datasets.

For our experiment, we run two baseline algorithms for comparison purposes. First, instead of customizing the order of rules for each password, we apply these rules with *a fixed order* for "sequential guessing". The fixed order is based on the overall rule popularity in the training data. Our second baseline is a widely used password cracking tool John the Ripper (JtR) [11]. We use the "single" mode and follow the default setting. Given a password, JtR applies a list of mangling rules to transform the password. It stops when all the mangling rules have been exhaustively tested.

6.2 Password Guessing Results

We use the proposed algorithm to evaluate the risks of *modified passwords*. For this experiment, we exclude identical password pairs (34.3%) since they only take one guess, and 46.4% of the pairs that don't match a rule. This leaves us 7,196,242 password pairs that represent password modifications (*exp dataset*). Our experiment contains two parts. First, we split the *exp dataset* randomly to use 50% for training and the other 50% for testing. Second, we use a much smaller training dataset to train the guessing algorithm. During the password guessing phase of our experiment, we test both directions for each password pair ($pw_1 \rightarrow pw_2$ and $pw_2 \rightarrow pw_1$), which doubles the size of the testing data.

Training on 50% of the Data. As shown in Figure 6, our best algorithm guessed 46.5% of the passwords within just 100 attempts. Figure 6(b) shows that 10 guesses already cracked 30% of the passwords. In comparison, the JtR baseline almost got nothing in the first 10 attempts and exhausted all the mangling rules after 1081 guesses. Since we evaluate an online-guessing scenario, we stopped our algorithm after 5000 guesses for each password.[3]

Comparing different algorithms, we show that the Bayesian model outperforms the fixed ordering method. This confirms the benefits of prioritizing the more likely rules for each password. In addition, we show that rotational guessing is better than sequential guessing. Sequential guessing has a clear stair-step increase of the hit rate after switching to a new rule. This indicates that the first few transformations under each rule are the most effective ones. Rotational guessing has an overall better performance due to switching the rules more frequently.

We argue that Bayesian-based sequential guessing still has its value, especially for *online guessing* attacks. As shown in Figure 6(b), sequential guessing's advantage is in the first 7 guesses — if the Bayesian prediction is correct, sticking to the right rule helped to guess the password more quickly. Within the first 7 guesses, Bayesian-based sequential guessing can guess 3% more passwords than rotational guessing. Given the large number of passwords being tested (3.6 million pairs, 7.2 million passwords), 3% still involve a large number of passwords (216K).

Using Smaller Training Data. Next, we try to use smaller datasets to train our algorithm (Bayesian+rotational). We vary the size of the training data from 0.01% to 10% of the *exp dataset*. To be consistent, we use the same 50% as the testing data (training and testing data has no overlap). As shown in Figure 7, the 0.1%-training curve is still overlapped with the 50%-curve, suggesting that extremely small training data can achieve a comparable performance. The result suggests that users are following a small number of consistent rules to modify their passwords. This is likely to make the modified passwords more predictable.

To measure the number of vulnerable password pairs, we use the 0.1%-trained model to guess the rest 99.9% of the password pairs. Since we guess both directions, the testing data essentially has 14 million passwords. Within 10 attempts, we guessed 30% (4.2 million passwords) — 3.8 million *password pairs* are cracked for at least one direction. Together with the identical password pairs (12.8 million), over 16.6 million pairs can be cracked within 10 attempts.

7 DISCUSSION & CONCLUSION

In this paper, we perform a large-scale empirical analysis on leaked password datasets over 8 years. We find that a majority of users have reused the same password or slightly modified an existing password for multiple services. Particularly, "shopping" and "email"

[3] Our experiment shows that 50,000 guesses can crack 70%.

services received the most reused and modified passwords (contradicting with existing results [24, 32]). In addition, users are still reusing their leaked passwords in other online services for years after the initial data breach, which introduces a persistent threat. More importantly, we find that the password modification patterns are highly consistent across various user populations, allowing attackers to quickly guess a large number of passwords with minimal training. Moving forward, we believe more proactive steps should be taken to protect user accounts after data breaches. Major online services such as Google have made an initial progress along this direction to proactively detect and stop malicious login attempts using the leaked passwords [33].

Limitations. Our study has a few limitations. First, our dataset is by no means complete, even for the 107 online services. For a given user, there are likely more reused or modified passwords in other services outside of our dataset. Our results can only be interpreted as a lower bound. Second, we treat each email address as a "user", but in practice, a user may have multiple email addresses. Again, the estimated password reuse and modification rates may be only a lower bound. Finally, our password guessing algorithms requires training data. We argue that such training data is relatively easy to obtain, and only a small training dataset is needed.

Data Sharing. To facilitate future research, we share our password dataset with the research community[4]. Although these datasets are already public on the Internet, it will still take a significant effort to search, collect, and clean the datasets. Sharing the dataset will benefit the research community as a whole. At the same time, we believe careful steps are needed to make sure the dataset is not misused by malicious parties. To this end, we follow a conservative data sharing policy that is commonly used by password researchers [10, 35]. First, we remove the email address from all the datasets, and use a hashed string as the identifier. Second, we remove the service name of each dataset. Finally, we will carefully verify the data requester's identity (*e.g.*, based on his/her institutional email address) before sharing the dataset.

ACKNOWLEDGMENTS

This project was supported by NSF grant CNS-1717028. Any opinions, conclusions or recommendations expressed in this material do not necessarily reflect the views of the funding agency.

REFERENCES

[1] Abusewith. 2017. Abusewith. http://abusewith.us/. (2017).
[2] Breach Alarm. 2017. Breach Alarm. https://breachalarm.com/all-sources/. (2017).
[3] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python* (1st ed.). O'Reilly Media, Inc.
[4] Kate Conger. 2016. Dropbox employee's password reuse led to theft of 60M+ user credentials. TechCrunch. (2016).
[5] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The Tangled Web of Password Reuse. In *Proc. of NDSS'14*.
[6] Dinei Florencio and Cormac Herley. 2006. A Large Scale Study of Web Password Habits. In *Proc. of WWW'06*.
[7] S.M. Taiabul Haque, Matthew Wright, and Shannon Scielzo. 2013. A Study of User Password Strategy for Multiple Accounts. In *Procs. of CODASPY'13*.
[8] HIBP. 2017. Have I Been Pwned. https://haveibeenpwned.com/. (2017).
[9] Shouling Ji, Shukun Yang, Xin Hu, Weili Han, Zhigong Li, and Beyah. 2015. Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords. *IEEE TDSC* PP, 99 (2015), 1–1.
[10] Shouling Ji, Shukun Yang, Ting Wang, Changchang Liu, Wei-Han Lee, and Raheem Beyah. 2015. PARS: A Uniform and Open-source Password Analysis and Research System. In *Proc. of ACSAC'15*.
[11] JtR. 2017. John the Ripper. http://www.openwall.com/john/. (2017).
[12] Patrick Gage Kelley et al. 2012. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *Proc. of IEEE S&P'12*.
[13] Johannes Kiesel, Benno Stein, and Stefan Lucks. 2017. A Large-scale Analysis of the Mnemonic Password Advice. In *Proc. of NDSS'17*.
[14] Saranga Komanduri et al. 2011. Of Passwords and People: Measuring the Effect of Password-composition Policies. In *Proc. of CHI'11*.
[15] Yue Li, Haining Wang, and Kun Sun. 2016. A Study of Personal Information in Human-chosen Passwords and Its Security Implications. In *Proc. of INFOCOM'16*.
[16] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. 2014. A Study of Probabilistic Password Models. In *Proc. of IEEE S&P'14*.
[17] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.
[18] Michelle L. Mazurek et al. 2013. Measuring Password Guessability for an Entire University. In *Proc. of CCS'13*.
[19] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2016. Fast, Lean, and Accurate: Modeling Password Guessability Using Neural Networks. In *Proc. of USENIX Security'16*.
[20] Arvind Narayanan and Vitaly Shmatikov. 2005. Fast Dictionary Attacks on Passwords Using Time-space Tradeoff. In *Proc. of CCS'05*.
[21] Gilbert Notoatmodjo and Clark Thomborson. 2009. Passwords and Perceptions. In *Proc. of AISC'09*.
[22] Heen Olivier and Neumann Christoph. 2017. On the Privacy Impacts of Publicly Leaked Password Databases. In *Proc. of DIMVA'17*.
[23] Pierluigi Paganini. 2016. Reuse of login credentials put more than 20M Alibaba accounts at risk. Security Affairs. (2016).
[24] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, , and Alain Forget. 2017. Let's go in for a closer look: Observing passwords in their natural habitat. In *Proc. of CCS'17*.
[25] Roi Perez. 2017. Hacker publicly releases 900GB of data stolen from Cellebrite. SC Magazine. (2017).
[26] Sarah Perez. 2016. 117 million LinkedIn emails and passwords from a 2012 hack just got posted online. TechCrunch. (2016).
[27] QNTM. 2017. Leet Transformation. https://qntm.org/l33t. (2017).
[28] Steve Ragan. 2015. Mozilla's bug tracking portal compromised, reused passwords to blame. CSO. (2015).
[29] TOBIAS SALINGER. 2015. Hackers post personal data stolen from adultery website Ashley Madison to dark web: reports. NY DailyNews. (2015).
[30] Richard Shay et al. 2010. Encountering Stronger Password Requirements: User Attitudes and Behaviors. In *Proc. of SOUPS'10*.
[31] OLIVIA SOLON. 2014. NHS patient data made publicly available online. Wired. (2014). http://www.wired.co.uk/article/care-data-leaks.
[32] Elizabeth Stobert and Robert Biddle. 2014. The Password Life Cycle: User Behaviour in Managing Passwords. In *Procs. of SOUPS'14*.
[33] Kurt Thomas, Frank Li, Ali Zand, Jake Barrett, Juri Ranieri, Eric Severance, Luca Invernizzi, Yarik Markov, Oxana Comanescu, Vijay Eranti, Angelika Moscicki, Dan Margolis, Vern Paxson, and Elie Bursztein. 2017. Data Breaches, Phishing, or Malware? Understanding the Risks of Stolen Credentials. In *Proc. of CCS'17*.
[34] Blase Ur et al. 2015. "I Added '!' at the End to Make It Secure": Observing Password Creation in the Lab. In *Proc. of SOUPS'15*.
[35] Blase Ur et al. 2015. Measuring Real-world Accuracies and Biases in Modeling Password Guessability. In *Proc. of USENIX Security'15*.
[36] Rafael Veras, Christopher Collins, and Julie Thorpe. 2014. On Semantic Patterns of Passwords and their Security Impact. In *Proc. of NDSS'14*.
[37] Verizon. 2017. Verizon's Data Breach Investigations Report. http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/. (2017).
[38] Vigilante. 2017. Vigilante. https://www.vigilante.pw/. (2017).
[39] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. 2016. Targeted Online Password Guessing: An Underestimated Threat. In *Proc. of CCS'16*.
[40] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. 2016. Understanding Password Choices: How Frequently Entered Passwords Are Re-used across Websites. In *Proc. of SOUPS'16*.
[41] Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. 2009. Password Cracking Using Probabilistic Context-Free Grammars. In *Proc. of IEEE S&P'09*.
[42] Joon Ian Wong. 2016. Stolen Dropbox passwords are circulating online. Here's how to check if your account's compromised. Quartz. (2016).
[43] Yinqian Zhang, Fabian Monrose, and Michael K. Reiter. 2010. The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis. In *Proc. of CCS'10*.
[44] Leah Zhang-Kennedy, Sonia Chiasson, and Paul van Oorschot. 2016. Revisiting password rules: facilitating human management of passwords. In *Proc. of eCrime'16*.

[4]Project website: https://people.cs.vt.edu/gangwang/pass