Empirical Analysis of User Passwords across Online Services

Chun Wang

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Application

Gang Wang, Chair
Danfeng Yao
David R. Raymond

May 09, 2018
Blacksburg, Virginia

Keywords: Password Reuse, Empirical Measurements, Bayesian Model

Copyright 2018, Chun Wang

Empirical Analysis of User Passwords across Online Services

Chun Wang

(ABSTRACT)

Leaked passwords from data breaches can pose a serious threat if users reuse or slightly modify the passwords for other services. With more and more online services getting breached today, there is still a lack of large-scale quantitative understanding of the risks of password reuse and modification. In this project, 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 a very common behavior (observed on 52% of the users). More surprisingly, 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. Extensive evaluations show that more than 16 million password pairs (30% of the modified passwords and all the reused passwords) can be cracked within just 10 guesses. We argue that more proactive mechanisms are needed to protect user accounts after major data breaches.

Empirical Analysis of User Passwords across Online Services

Chun Wang

(GENERAL AUDIENCE ABSTRACT)

Since most of the internet services use text-based passwords for user authentication, the leaked passwords from data breaches pose a serious threat, especially if users reuse or slightly modify the passwords for other services. The attacker can leverage a known password from one site to guess the same user's passwords at other sites more easily. In this project, we perform the first large-scale study of password usage based on the largest ever leaked password dataset. The dataset consists of 28.8 million users and their 61.5 million passwords from 107 internet services over 8 years. We find that password reuse and modification is a very common behavior (observed on 52% of the users). More surprisingly, we find that sensitive online services such as shopping websites and email services received the most reused and modified passwords. In addition, users would still reuse the already-leaked passwords for other online services for years after the initial data breach. Finally, we develop a cross-site password-guessing algorithm to guess the modified passwords based on one of the user's leaked passwords. Our password guessing experiments show that 30% of the modified passwords can be cracked within only 10 guesses. Therefore, we argue that more proactive mechanisms are needed to protect user accounts after major data breaches.

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Dr. Gang Wang. His great idea and deep insight are invaluable for this project. I am truly grateful to have been his student. I would like to thank Dr. Danfeng Yao and Dr. David Raymond for serving on my committee and providing brilliant comments and suggestions for improvement. I would also like to thank Steve Jan and Hang Hu in Prof. Wang's group for their help and useful discussion for this project. I would also like to thank the Department of Computer Science at Virginia Tech for everything I have learned during my master's study, and also for the support of my master's study with a GTA position. Last but not least, I would also like to express my special appreciation for the love and support from my family.

Contents

List of Figures

List of Tables

Chapter 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 [44]. Many of the leaked datasets contain sensitive information such as *user passwords*, which are often made publicly available on the Internet by the attackers [29, 30, 34, 37, 50].

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 [27, 32]. 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 [5]).

With more and more passwords being leaked [10, 45], 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 [9, 17, 36, 38, 41, 47]. 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 [47].

Recently, researchers start to analyze empirical data to understand users' password reuse and modification patterns [6, 8, 28, 47, 51]. 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 [6]. 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 work, 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.

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

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, 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 [28, 47].

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 4 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 helps to confirm observations from existing small-scale user studies, and 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 will share our dataset with the broad research community. We carefully design a data sharing policy so that the dataset can benefit further research without being misused by malicious parties.

Chapter 2

Review of Literature

Password Reuse and Modification. Text-based password is still the primary authentication method for today's online services. An early study [8] shows that users maintain 25 online accounts on average. 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 [6, 9, 47]. Attackers may leverage the reused passwords to compromise new user accounts, or link user identities by mining the leaked password datasets [26].

Table 2.1 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 [6, 9, 17, 25, 36, 38, 41, 47]. For example, Das et al. [6] have reported that 51% of the users re-use passwords across online services. The results from Notoatmodjo et al. [25] suggest that users with more accounts are more likely to reuse their passwords. More recently, researchers find that users tend to reuse more complex passwords (since they are harder to remember) [47], and passwords that need to be entered frequently [28]. Finally, Stobert and Biddle's interview [38] suggests that password reuse often happens on "less important" services.

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 [47]. To these ends, empirical analysis is needed to understand users' real-world behavior [6, 8, 28, 47, 51]. 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 [8] conducted 10 years ago by Microsoft (500K users). However, researchers of [8] only analyzed password reuse, without covering the danger of password modification across services.

In our work, we seek to fill in the gap by collecting and analyzing a large-scale empirical password dataset. Our dataset contains 28.8 million users and 61.5 million passwords across 107 services. We study 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 [22]. More targeted guessing exploits the fact that users may reuse the same or similar passwords [6, 51]. More recently, target guessing also incorporates users' personal information such as name and birthday [19, 46].

Offline guessing can easily reach trillions of guessing attempts [12, 13, 15, 23, 24, 43, 48]. 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 [20, 24], Mangled Wordlist method [42], Probabilistic Context-Free Grammars Method (PCFGs) [15, 24, 43, 48], and Deep Neural Networks [23].

Table 2.1: Related works on password reuse and modification.

	PW Reuse	PW Modify	Methods	# Users
[25]	✓	×	Survey	26
[38]	✓	×	Survey	27
[47]	✓	×	Empirical+Survey	134
[8]	✓	×	Empirical	544,960
[9]	×	✓	Survey	80
[36]	×	✓	Survey	470
[51]	×	✓	Empirical	7,700
[41]	✓	✓	Survey	49
[28]	✓	✓	Empirical+Survey	154
[17]	✓	✓	Survey	5,000
[6]	✓	✓	Empirical+Survey	6,077
Our	✓	✓	Empirical	28,836,775

Password Meters and Creation Policies. To help users to create strong passwords, online services often use password meters to indicate the strength of the password. Researchers have studied different metrics to quantify password strength. Traditionally, password strength is measured by information entropy [4, 35]. More recently, researchers use the estimated "number of guesses" to quantify the password strength [7, 13, 16, 40, 49]. The resulting password strength may vary for different guessing methods. In addition, online services often have password creation policies [6, 16]. For example, a policy may require a new password to have at least 8 digits with both numbers and upper-case letters. Most policies don't prevent users from including English words or even their own names in their passwords [11, 36, 41], which ultimately weakens the password strength.

Chapter 3

Dataset

Table 3.1: Categories and statistics of the collected 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

To study password usage across online services, we gathered a large number of password datasets and linked the same user's passwords together. Our final dataset contains 28.8 million users and their 61.5 million passwords across 107 services. In the following, we briefly describe our data collection process and perform a preliminary analysis.

Table 3.2: Dataset list.

Social: Myspace, VK.com, LinkedIn, Twitter, XSplit, Crush007, InternetFamous
Adult: Zoosk, Mate1, YouPorn, NaughtyAmerica, MuslimMatch, IChatUSA, AfrikaDating, iChatAsia, IChatLatino
Game: Neopets, 7k7k, Lbsg, MangaTraders, PSN, BCWars, DifferenceGames, Neofriends, CSGameServs, ThePirateGame, SoulSplit, Miniclip, Clan-Amok
Entertain: Lastfm, Swingbrasileiro, LATimes, FijiLive
Internet: 000webhost, Comcast, Yahoo, WtSpy, Aha-Share, MyLLoyD, Keepyourlinks, Sony, TechFactory, SonyPicture, LizardStresser, EmkoElektronik, OVPN, TCS, Wipro, LinuxMint, Aircraftonline, Yelaw
Email: Gmail, Mail.ru, Yandex
Forum: CrackingForum, Abusewith.us, Gawker, leakforums, SkTorrent, LateChef, Global-RS, Ljuska, LizardSquad, EquityDevelopment, HackForums, PalestineWeekly, Tomopop.com, O2C, Migalki, Rskingdom, Surgeryu, Hacker, Lizardstresser, Arena, Tennisarchives, CrackedSpotify, Fyllo, NuteUFSC, 0sec
Shopping: RedBox, 1394store, Myaribags, Mp3mixx, Auction-Warehouse, CardioFitness, TuneSoman, TheatrHall, Autolet, Home.co.li, NoCommonScents, IBuyPower
Unknown: Dataset 1–7
Business: MtGox, Movatiathletic, Hrsupporten, 99Fame, Ilink.ph, Xzonegym, IcePlex, RealExpose, Salalahport

3.1 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 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 [13, 14, 42]. This effort returned 460,874,306 plaintext passwords (93% of all passwords). The remaining

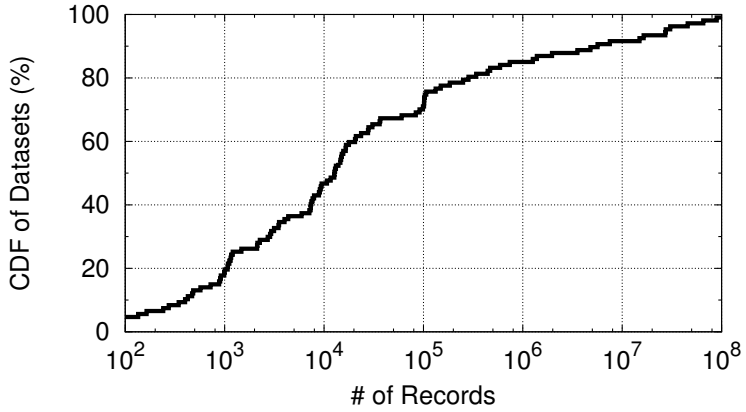


Figure 3.1: # of password records in each dataset.

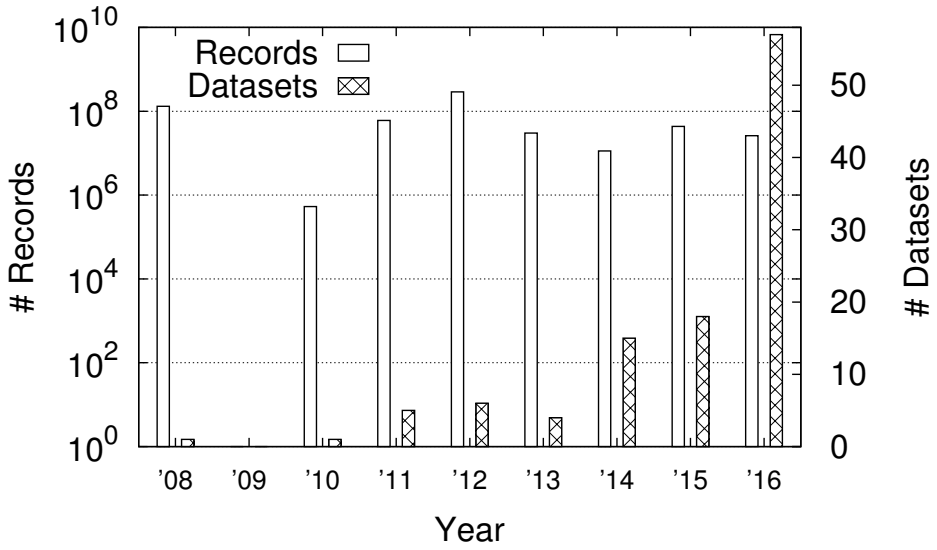


Figure 3.2: # of datasets and total # records per year.

7% are difficult to recover, and we will use them to test our guessing algorithm later in 4.4.2. We have carefully checked each dataset to make sure there are no duplicate records.

Data Statistics. In Table 3.1, we manually classify the 107 online services into 10 categories based on their category information in Alexa[1]. 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 complete

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

list of the 107 datasets with 460 million password records in total is shown in Table 3.2. We use the service name to denote each of the online services.

As shown in Figure 3.1, 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. To this end, our analysis is likely to capture a lower-bound for password reuse and modification across services.

Finally, 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, 10, 45]. Figure 3.2 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.

Note that although we know the year when a dataset was leaked. We don't have the detailed timestamp regarding when a user password was set. These datasets were leaked between 2008–2016. However, we do not think this leak time of the dataset could be used as the timestamp of password usage, since it is only the break time of the dataset and the passwords in the datasets leaked earlier may be newer than from those leaked later. Therefore in this project we will not study the time consideration in password reuse, such as the evolution of habits of users.

3.2 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. This means our study will only estimate a lower-bound of password reuse and modifications.

3.3 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 [6, 7, 13, 19, 42, 43, 46] to develop security mechanisms to protect users in the long run.

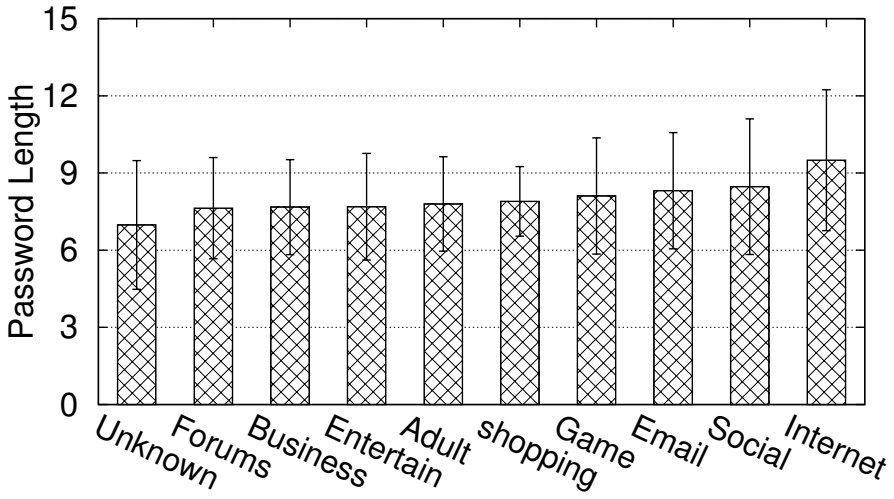


Figure 3.3: Average password length for different online services.

Table 3.3: Password composition. Simple-composition passwords are shown in bold font (40.0%).

#PW Category	# of Passwords	Ratio (%)
Letters only	17,626,490	28.6%
Numbers only	7,188,646	11.7%
Symbols only	96,283	0.2%
Letters+Numbers	34,451,712	56.0%
Letters+Symbols	1,023,441	1.7%
Numbers+Symbols	201,224	0.3%
Letters+Numbers+Symbols	964,650	1.6%
Total	61,552,446	100%

3.4 Preliminary Analysis

We perform an initial analysis on the primary dataset to understand the basic characteristics of the collected passwords. Figure 3.3 shows the average password length for different services. Most passwords are between 6–10 digits. This is expected since most websites require at least 6 or 8 digits for the passwords [16]. We also observe that the "Internet" category has the longest passwords, followed by "social network" and "email" services.

Table 3.3 shows the overall composition of passwords. We find that a large portion (40%) of

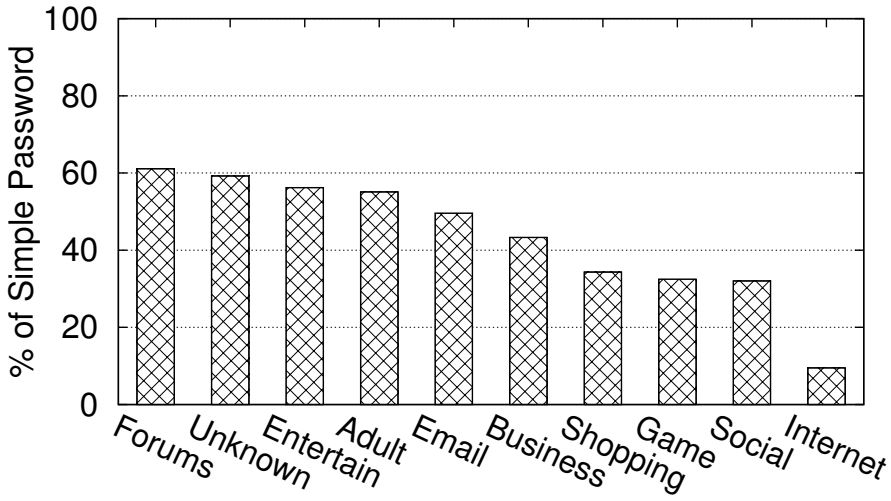


Figure 3.4: % of simple-composition password (passwords that only contain a single type of characters).

the passwords are *simple-composition passwords* which only contain a single type of character (English letters, numbers or symbols). Simple-composition passwords are not necessarily easier to guess, which also depends on the guessing algorithm. However, statistically, simple-compassion passwords help attackers to dramatically reduce the guessing space.

In Figure 3.4, we further examine the ratio of simple composition passwords in different services. Again, "Internet" and "Social" categories have the lowest ratio of simple passwords. On the contrary, the perceived sensitive services such as "Email" and "Adult" are more likely to have simple composition passwords. A possible explanation is users tend to set simpler passwords for services where they need to enter password frequently [47].

Chapter 4

Results

4.1 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 [6, 9, 28, 38, 41, 47] 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 4.2). Finally, we empirically quantify the security risks of password reuse/modifications by performing password guessing experiments (Section 4.3 and Section 4.4), and discuss the implications of our results to the increasingly frequent data breaches (Chapter 5).

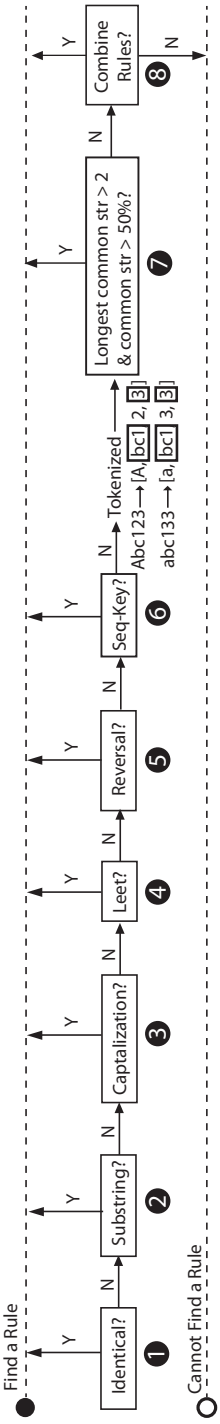


Figure 4.1: The workflow to measure a user's password transformation patterns.

4.1.1 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 [8]. 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 [6].

4.1.2 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, 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 4.1 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 [6, 51, 52]. We test these rules against the password pairs in the *primary dataset*, and the results are shown in Table 4.1.

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.

Table 4.1: Distribution of password transformation rules.

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%

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 4.2, 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%) [31]. 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.

Table 4.2: Substring rule: insertion/deletion patterns.

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%

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%.

Table 4.3: Common substring rule: longest common substring and transformation 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%

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 4.1, 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. Among the 1000 pairs, 44 pairs look like to have accidental overlaps, which projects a false positive rate of 4.4%. 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, we estimate that the common substring rule should match at least 5.4% of all password pairs (lower bound).

Table 4.3 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 4.3 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%).

Table 4.4: Rule combinations (CSS: Common SubString).

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%

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 4.4, "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 46.4% of password pairs remain unmatched. To make sure we did not miss any major rules, we randomly sample 1000 unmatched pairs for manual examination. Again, we have removed the email address before manually looking at the passwords. 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".

4.2 Measuring Password Habits

So far we have identified the reused and modified password pairs. 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. *First*, how often do users reuse (modify) the same password for different services? *Second*, what types of passwords are more likely to be reused (modified)? *Third*, what types of online services receive the most reused (modified) passwords? *Fourth*, how demographics affect password reuse and modification behavior? *Finally*, how long do users wait before changing their reused passwords in other services after a data breach?

Some of the questions about *password reuse* (*e.g.*, question 1 and 2) have been studied using small-scale user studies [6, 9, 17, 25, 36, 38, 41, 47]. These studies have provided in-depth results but only for a small user population. Our analysis, on the other hand, covers a

population that is orders of magnitude larger (28 million users), which allows us to cross-examine prior results at a much larger scale. Note that our approach also has a limitation: for each user, we only cover a subset of her online services. The result can only be interpreted as a lower bound.

4.2.1 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 $P_i = [p_{i,1}, p_{i,2}, p_{i,3}, ..., p_{i,K_i}]$. Figure 4.2 shows how the metrics are calculated.

Reuse Rate. 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.

Modification Rate. This metric 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 4.1, 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.

In Figure 4.3, we plot the distribution of RR and MR for users with a different number of total passwords. We seek to 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

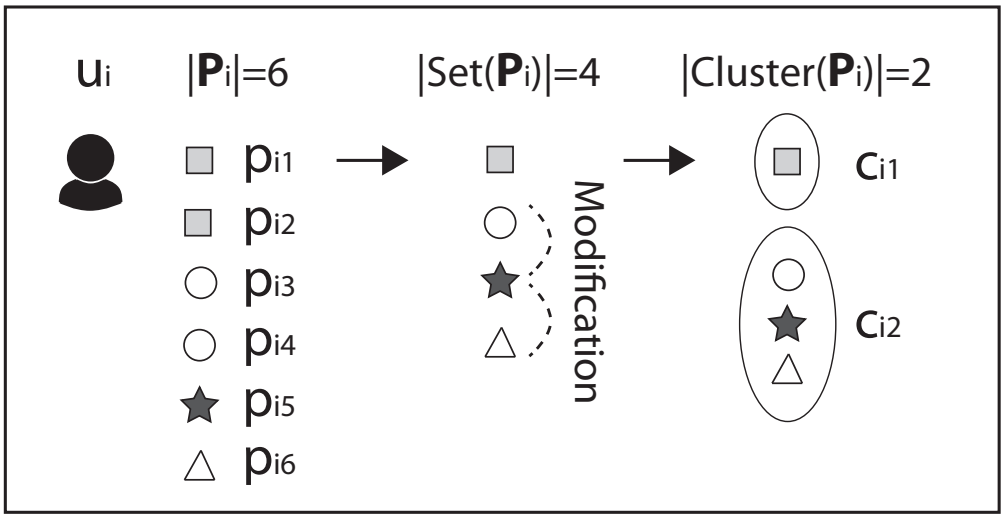


Figure 4.2: Calculating the password reuse rate and modification rate. In this example, the reuse rate (RR) = 6/4 = 1.5; the modification rate (MR) = 6/2 = 3.0.

Table 4.5: Number of passwords for each user in our dataset.

#Passwords per User	# of Users	% of Users
2	25,515,516	88.5%
3	2,877,322	10.0%
4	370,990	1.3%
5	54,258	0.2%
≥ 6	18,726	<0.1%

her passwords are reused.

Our result supports this intuition. As shown in Figure 4.3, both the reuse rate and modification rate are increasing as users have more total passwords. We stop at 5 because the vast majority of users in our dataset have no more than 5 passwords (Table 4.5). Our results agree with prior user studies (100+ users) that examine this intuition on *password reuse* [25, 47].

Figure 4.3 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.

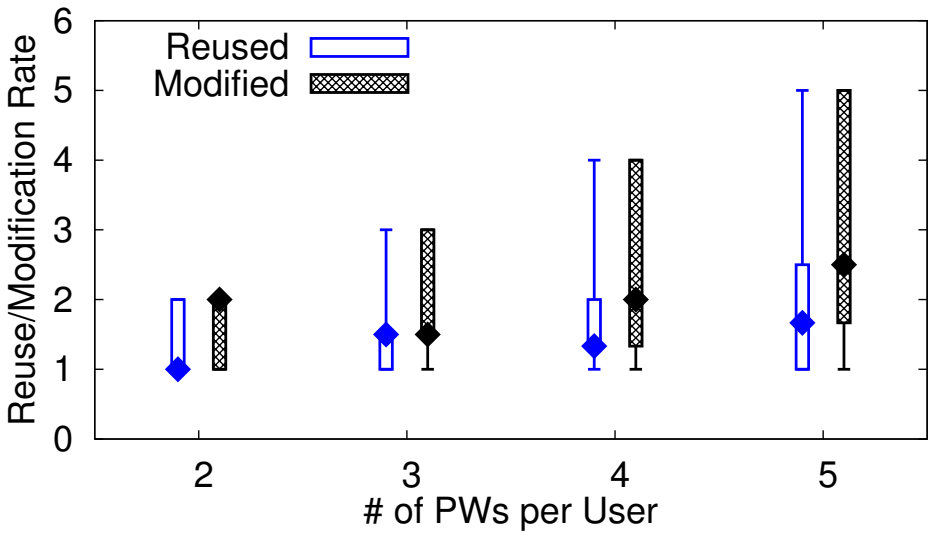


Figure 4.3: Password reuse/modification rate for users of different # of total passwords. The box plot quantiles are 5%, 25%, 50%, 75%, 95%.

4.2.2 Impact of Password Complexity

Next, we examine what types of passwords are more likely to be reused or modified. Our intuition is that users cannot remember a large number of long and complex passwords. The hypothesis is that longer and more complex passwords are more likely to be reused (or modified).

As shown Table 4.6, we divide passwords into different groups. The "reused" and "modified " groups may have overlaps, *i.e.*, a password can involve in both reuse and modification for different services. For passwords that are not involved in reuse or modification, we put them to the "unique" password group. The definition of "unique" password is within the primary dataset.

Table 4.6 shows that reused and modified passwords are slightly shorter (8.07 and 8.17 characters) compared to unique passwords (8.21 characters). We examine the statistical significance using the standard ANOVA test (given that the data is continuous). The resulting

Table 4.6: Composition and length of passwords in different groups. The differences are statistically significant based on ANOVA tests and Chi-square tests ($p < 0.0001$).

Password Group	Simple	Double	Triple	Avg. Length
Reused	40.2%	58.6%	1.2%	8.07
Modified	38.6%	60.0%	1.4%	8.17
Unique	42.9%	55.3%	1.7%	8.27
Overall	40.5%	58.0%	1.6%	8.21

$p < 0.0001$, confirming the statistical significance.

In addition, Table 4.6 shows that reused/modified passwords are indeed more complex. Comparing to unique passwords, the reused and modified passwords are less likely to be single-composition passwords. We examine the statistical significance of the results using Chi-square tests (given that the data is categorical). The resulting $p < 0.0001$, confirming the statistical significance and our hypothesis is supported.

Overall, the above results partially support our hypothesis: reused and modified passwords are indeed more complex but shorter than average. Our results echo the finding of a prior user study (N=134) on *password reuse* [47], which shows users tend to reuse more complex passwords (measured by entropy). Another user study (N=154), however, provides a different perspective showing that reused passwords are "weaker" than unique passwords (based password guessing metrics) [28].

4.2.3 Impact of Online Services

In Figure 4.4, we examine what types of online services have received the most reused or modified passwords. Considering that the total number of passwords of each service type is different, we compare the *ratio* instead of the absolute number.

We find that "shopping" services have the most reused/modified passwords with a ratio over

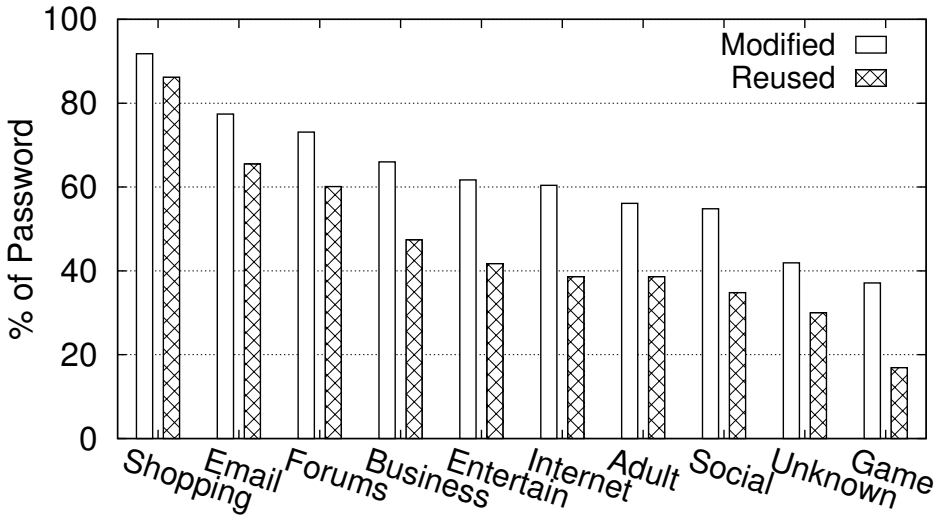


Figure 4.4: Ratio of reused and modified password under different services. Shopping and email services received the most reused and modified passwords.

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) [28], which shows that "email" is among categories with the least password reuse.

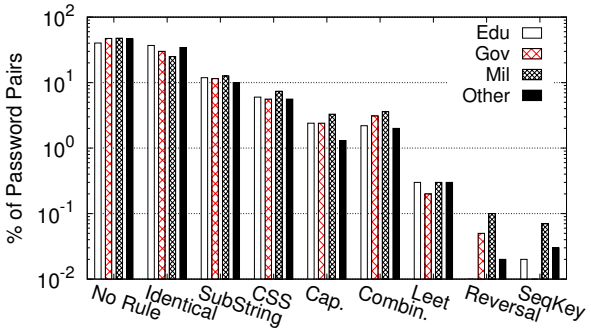
4.2.4 Impact of User Demographics

Next, we analyze how much user demographics affect their password reuse and modification patterns. We focus on the profession and country of users, which can be inferred from users' email addresses. We do not consider other demographics, such as the gender of the user since this information is only available for a small number of datasets.

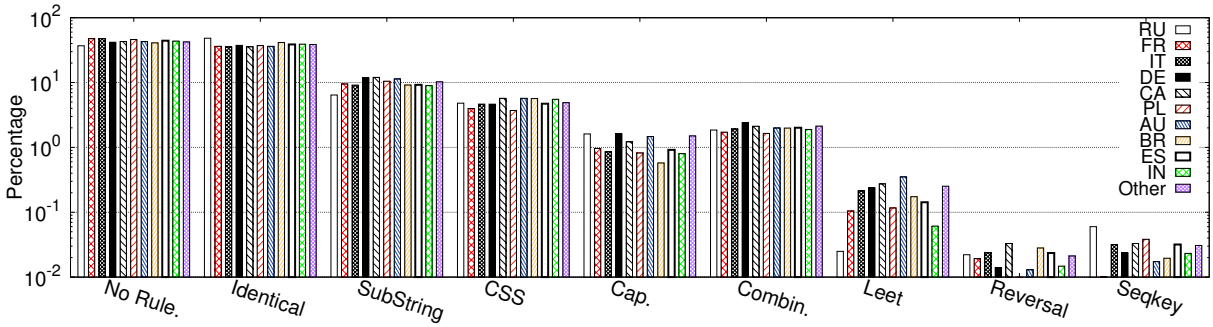
Profession. Certain email domains are exclusive to people of special organizations. Based on users' email addresses, we identify 128,036 users from higher educational institutions ("`.edu`"), 7,376 users from military ("`.mil` "), and 3,384 users from government agencies ("`.gov`"). As shown in Figure 4.5(a), military users have a slightly lower reuse rate than other groups. However, the overall patterns are surprisingly consistent: about 30% password pairs are identical, followed by those transformed by the substring rule (10%), the common substring rule (5%) and the capitalization rule (3%). Rules such as leet and reversal are consistently below 1%. We also run Chi-square tests and the resulting $p > 0.1$. This confirms the insignificant differences between the different user groups.

Country. Similar results are observed in Figure 4.5(b), where we divide users based on their country code. More specifically, we select email domains that contain a country code (*e.g.*, "`.ru`" stands for Russia), which returns 233 country codes and 5,892,528 users. Figure 4.5(b) shows the rule distribution for top 10 countries (counting for 90.5% of the users with a country code). Again, the transformation patterns are very similar for users of different country codes. Chi-square tests again return a $p > 0.1$, indicating the insignificant differences between user groups.

Our result demonstrates a high-level of consistency (low variance) for password modification patterns across different user populations. This, however, could make the attacker's job easier — it is possible for the attacker to learn the basic transformation patterns from a



(a) Profession



(b) Country

Figure 4.5: Distribution of password transformation rules for users of different professions and countries.

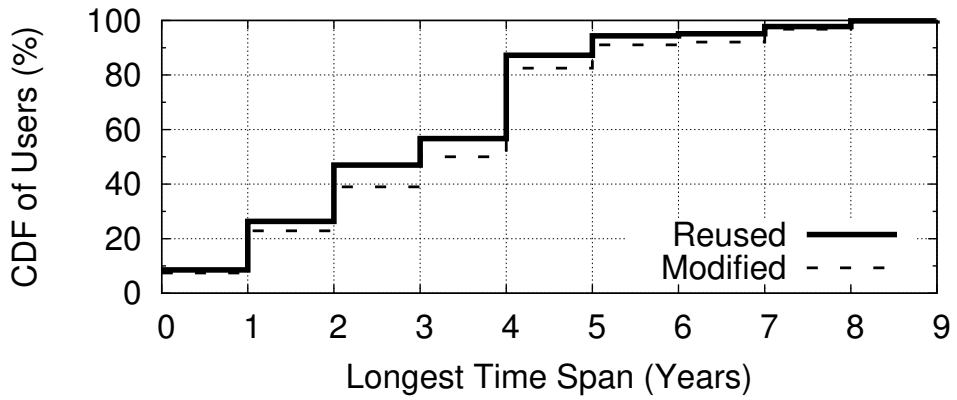


Figure 4.6: Longest time span between any pair of reused/modified passwords for each user.

small dataset that be generalized to a broader user population. Later in section 4.3, we develop a *training-based* algorithm based on this intuition.

4.2.5 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 4.6.

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.

4.3 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, given the low variance of password transformation patterns (section 4.2), can attackers use a small training data (*e.g.*, 0.1%) to achieve an effective guessing?

4.3.1 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 4.1) on the known password. This idea is similar to DBCBW [6], 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`); For "`1234!`", we may try to remove "!" first.

Table 4.7: Feature list of the Bayesian model.

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)

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 4.1, 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) [21], 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 4.7 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 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.

4.3.2 Baselines

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. This baseline mimics the same design of DBCBW but still takes advantage of the training data to obtain the fixed order of rules. Our second baseline is a widely used password cracking tool John the Ripper (JtR) [14]. 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.

When choosing baselines, we have ruled out existing algorithms that don't fit our threat model. First, we rule out non-targeted guessing algorithms. Non-targeted guessing algorithm is not optimized to guess a user's password based on a known password. Even the-state-of-the art algorithms [23, 42] will take 10^{12} guesses to hit 50% of the passwords in one of our datasets (000webhost). Second, we also rule out targeted guessing algorithms that require the user's PII information (*e.g.*, real name, date of birth) [19, 46]. Such PII information is not available in our datasets.

4.4 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 (*i.e.*, new passwords created from scratch). This leaves us 7,196,242 password pairs that represent password modifications (*exp dataset*).

4.4.1 Guessing Modified Passwords

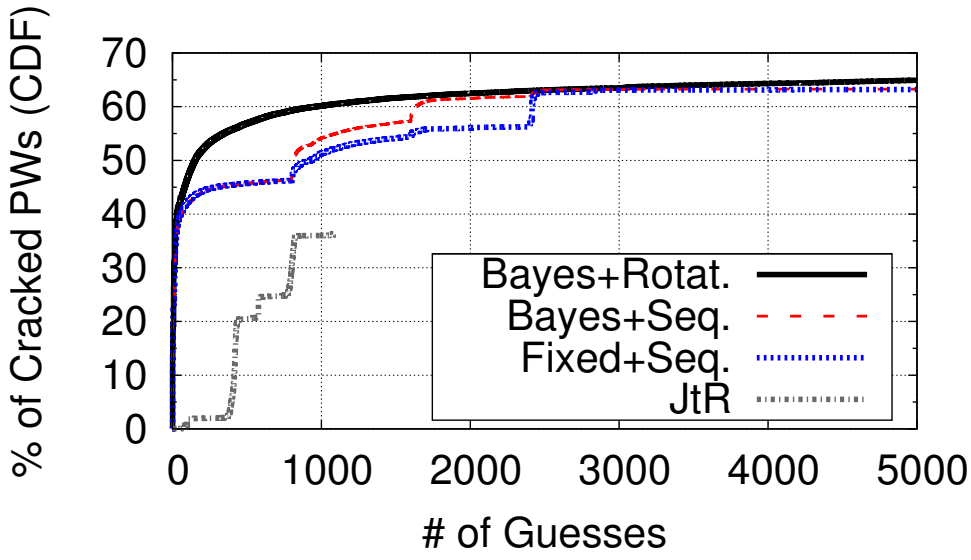
Our experiment contains two parts. First, we split the *exp dataset* randomly to use 50% for training and the other 50% for testing. Second, we use a much smaller training dataset to train the guessing algorithm. During the password guessing phase of our experiment, we test both directions for each password 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 4.7, our best algorithm guessed 46.5% of the passwords within just 100 attempts. Figure 4.7(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.[1]

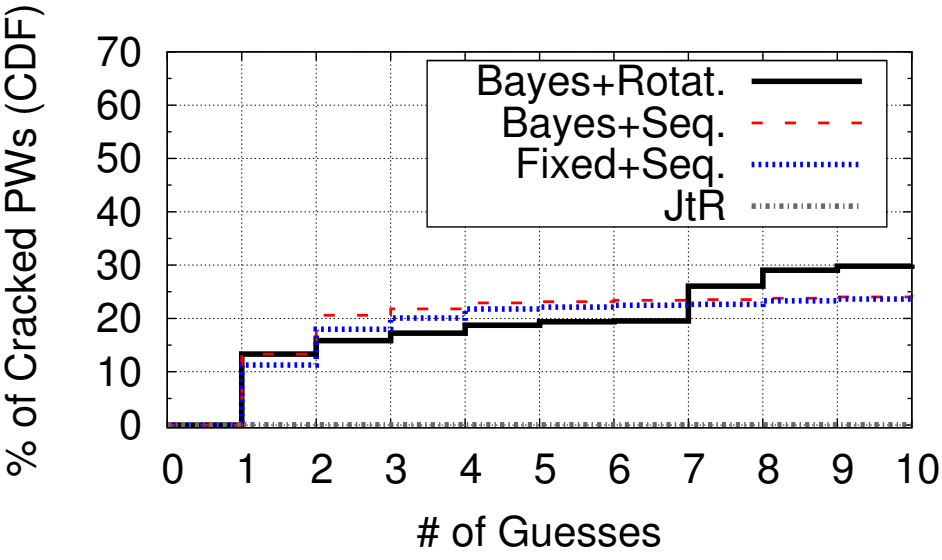
Comparing different algorithms, we show that the Bayesian model outperforms the fixed ordering method. This confirms the benefits of prioritizing the more likely rules for each password. In addition, we show that rotational guessing is better than sequential guessing. Sequential guessing has a clear 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 4.7(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

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



(a) 5000 Guesses



(b) 10 Guesses

Figure 4.7: Password guessing with 50% of the data for training.

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 4.8, 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.

4.4.2 Cracking the Remaining Hashes

Finally, we perform a quick experiment on the uncracked hashes. In Section 3, when pre-processing our datasets, we run a number of password guessing tools to recover plaintext passwords from their hashes. After a whole week of non-stop computation, there were still uncracked hashes remained. Here, we test our algorithm on the uncracked hashes. In total, we identify 6,218,778 password pairs where one password is an uncracked hash and the other one has plaintext. Within 5000 attempts, our algorithm successfully recovered 939,400

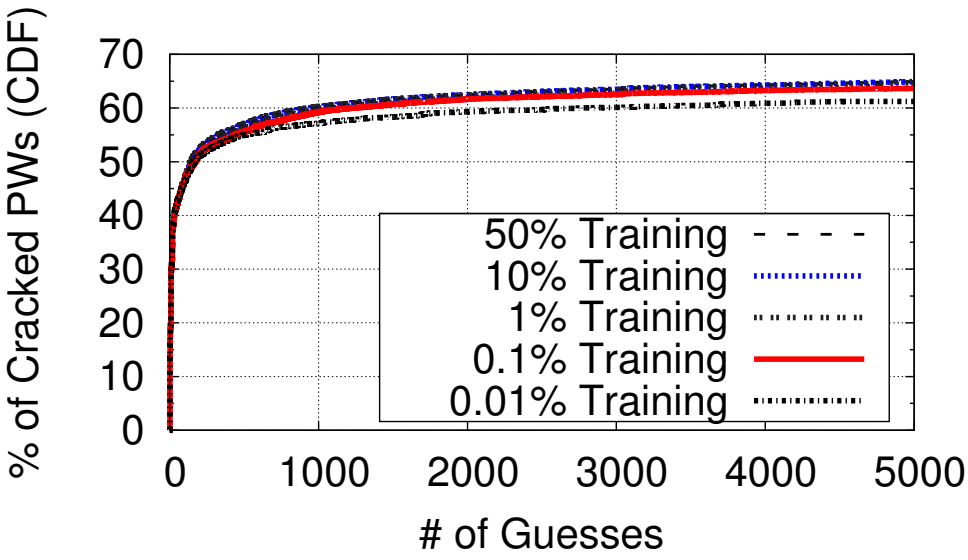


Figure 4.8: Password guessing with different training data sizes.

(15.1%) of the remaining hashes. This demonstrates the value of our algorithm over existing offline guessing tools. As a future work, we plan to further test our algorithm on other leaked datasets with *salted* password hashes.

Chapter 5

Discussion

Our analysis demonstrates the security threat of password reuse and modification. This threat is still escalating due to the increasingly more frequent breaches of online services. Below, we briefly discuss the key implications of our results, the possible defense approaches moving forward, our limitations and the data sharing plan.

Key Implications. Our study provides new insights into users' password reuse and modification patterns by analyzing large-scale empirical data. Comparing with existing user studies, the advantage of our method is the large *scale*. Our results confirm some of the observations in small-scale studies. For example, users who have more passwords are more likely to reuse passwords, and reused passwords are typically more complex [47]. On the other hand, some of our findings contradict with small-scale user studies. For example, unlike [28, 38], we show that "email" is one of the top online service categories with the most reused passwords. We believe our method is complementary to existing studies, together providing a more comprehensive understanding of users' password usage.

Defense. Moving forward, a bigger challenge is how to effectively mitigate the threat from password data breaches. Given the high reuse rate of passwords, it is necessary to immediately notify users to reset their passwords after the breach, not only for the breached service but also other services with a similar password. Unfortunately, in reality, not all the breached services would immediately disclose the incident and/or contact users [18, 33]. In addition, during password reset, it is important to make sure users don't simply modify the already-leaked password to create the new one. A better practice is to use password managers (*e.g.*, 1Password) to set a unique and strong password for each service without the need to memorize them. The problem is, users who adopted password managers rarely take advantage of this feature. A recent study shows that password manager users still have most of their passwords reused [47]. Finally, for online services, basic defense should be in place to detect and stop malicious login attempts using the leaked passwords. Major online services such as Google have made an initial progress along this direction [39].

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 plan to share our password dataset with the research community to facilitate future research. Although these datasets are already public on the Internet, it will still take a significant effort to search, collect, and clean the datasets. Therefore, 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 [13, 42]. 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. More details about our data sharing policy can be found from our project website[1].

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

Chapter 6

Conclusions

In this project, we perform a large-scale empirical analysis on leaked password datasets over 8 years. By analyzing 28.8 million users' 61.5 million passwords across 107 services, we reveal new insights into users' password reuse and modification patterns. 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" services received the most reused and modified passwords. 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. This not only applies to the breached services, but also other services that share a reused or modified password of the user.

Bibliography

[1] Abusewith. Abusewith. http://abusewith.us/, 2017.

[2] Breach Alarm. Breach alarm. https://breachalarm.com/all-sources/, 2017.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[4] Joseph Bonneau, Cormac Herley, Paul C. van Oorschot, and Frank Stajano. Passwords and the evolution of imperfect authentication. *Commun. ACM*, 58(7):78–87, 2015.

[5] Kate Conger. Dropbox employee's password reuse led to theft of 60m+ user credentials. TechCrunch, 2016.

[6] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. The tangled web of password reuse. In *Proc. of NDSS'14*, 2014.

[7] Xavier de Carné de Carnavalet and Mohammad Mannan. From very weak to very strong: Analyzing password-strength meters. In *Proc. of NDSS'14*, 2014.

[8] Dinei Florencio and Cormac Herley. A large scale study of web password habits. In *Proc. of WWW'06*, 2006.

[9] S.M. Taiabul Haque, Matthew Wright, and Shannon Scielzo. A study of user password strategy for multiple accounts. In *Procs. of CODASPY'13*, 2013.

[10] HIBP. Have i been pwned. https://haveibeenpwned.com/, 2017.

[11] Philip G. Inglesant and M. Angela Sasse. The true cost of unusable password policies: Password use in the wild. In *Proc. of CHI'10*, 2010.

[12] Shouling Ji, Shukun Yang, Xin Hu, Weili Han, Zhigong Li, and Beyah. Zero-sum password cracking game: A large-scale empirical study on the crackability, correlation, and security of passwords. *IEEE TDSC*, PP(99):1–1, 2015.

[13] Shouling Ji, Shukun Yang, Ting Wang, Changchang Liu, Wei-Han Lee, and Raheem Beyah. Pars: A uniform and open-source password analysis and research system. In *Proc. of ACSAC'15*, 2015.

[14] JtR. John the ripper. http://www.openwall.com/john/, 2017.

[15] Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proc. of IEEE S&P'12*, 2012.

[16] Johannes Kiesel, Benno Stein, and Stefan Lucks. A Large-scale Analysis of the Mnemonic Password Advice. In *Proc. of NDSS'17*, 2017.

[17] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Serge Egelman. Of passwords and people: Measuring the effect of password-composition policies. In *Proc. of CHI'11*, 2011.

[18] Kif Leswing. Here's why linkedin wants you to reset your password. BusinessInsider, 2016.

[19] Yue Li, Haining Wang, and Kun Sun. A study of personal information in human-chosen passwords and its security implications. In *Proc. of INFOCOM'16*, 2016.

[20] Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. A study of probabilistic password models. In *Proc. of IEEE S&P'14*, 2014.

[21] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[22] Michelle L. Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring password guessability for an entire university. In *Proc. of CCS'13*, 2013.

[23] William Melicher, Blase Ur, Sean M. Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, lean, and accurate: Modeling password guessability using neural networks. In *Proc. of USENIX Security'16*, 2016.

[24] Arvind Narayanan and Vitaly Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In *Proc. of CCS'05*, 2005.

[25] Gilbert Notoatmodjo and Clark Thomborson. Passwords and perceptions. In *Proc. of AISC'09*, 2009.

[26] Heen Olivier and Neumann Christoph. On the privacy impacts of publicly leaked password databases. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017*, 2017.

[27] Pierluigi Paganini. Reuse of login credentials put more than 20m alibaba accounts at risk. Security Affairs, 2016.

[28] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, , and Alain Forget. Let's go in for a closer look: Observing passwords in their natural habitat. In *Proc. of CCS'17*, 2017.

[29] Roi Perez. Hacker publicly releases 900gb of data stolen from cellebrite. SC Magazine, 2017.

[30] Sarah Perez. 117 million linkedin emails and passwords from a 2012 hack just got posted online. TechCrunch, 2016.

[31] QNTM. Leet transformation. https://qntm.org/l33t, 2017.

[32] Steve Ragan. Mozilla's bug tracking portal compromised, reused passwords to blame. CSO, 2015.

[33] Jeff John Roberts. Yahoo could face legal trouble over delay in disclosing hack. Fortune, 2016.

[34] TOBIAS SALINGER. Hackers post personal data stolen from adultery website ashley madison to dark web: reports. NY DailyNews, 2015.

[35] Claude Elwood Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, 2001.

[36] Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L. Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering stronger password requirements: User attitudes and behaviors. In *Proc. of SOUPS'10*, 2010.

[37] OLIVIA SOLON. Nhs patient data made publicly available online. Wired, 2014. http://www.wired.co.uk/article/care-data-leaks.

[38] Elizabeth Stobert and Robert Biddle. The password life cycle: User behaviour in managing passwords. In *Procs. of SOUPS'14*, 2014.

[39] 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. Data breaches, phishing, or malware? understanding the risks of stolen credentials. In *Proc. of CCS'17*, 2017.

[40] Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L. Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. How does your password measure up? the effect of strength meters on password creation. In *Proc. of USENIX Security'12*, 2012.

[41] Blase Ur, Fumiko Noma, Jonathan Bees, Sean M. Segreti, Richard Shay, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. "i added '!' at the end to make it secure": Observing password creation in the lab. In *Proc. of SOUPS'15*, 2015.

[42] Blase Ur, Sean M. Segreti, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, Darya Kurilova, Michelle L. Mazurek, William Melicher, and Richard Shay. Measuring real-world accuracies and biases in modeling password guessability. In *Proc. of USENIX Security'15*, 2015.

[43] Rafael Veras, Christopher Collins, and Julie Thorpe. On semantic patterns of passwords and their security impact. In *Proc. of NDSS'14*, 2014.

[44] Verizon. Verizon's data breach investigations report. http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/, 2017.

[45] Vigilante. Vigilante. https://www.vigilante.pw/, 2017.

[46] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proc. of CCS'16*, 2016.

[47] Rick Wash, Emilee Rader, Ruthie Berman, and Zac Wellmer. Understanding password choices: How frequently entered passwords are re-used across websites. In *Proc. of SOUPS'16*, 2016.

[48] Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *Proc. of IEEE S&P'09*, 2009.

[49] Daniel Lowe Wheeler. zxcvbn: Low-budget password strength estimation. In *Proc. of USENIX Security'16*, 2016.

[50] Joon Ian Wong. Stolen dropbox passwords are circulating online. here's how to check if your account's compromised. Quartz, 2016.

[51] Yinqian Zhang, Fabian Monrose, and Michael K. Reiter. The security of modern password expiration: An algorithmic framework and empirical analysis. In *Proc. of CCS'10*, 2010.

[52] Leah Zhang-Kennedy, Sonia Chiasson, and Paul van Oorschot. Revisiting password rules: facilitating human management of passwords. In *Proc. of eCrime'16*, 2016.