

Spring 2015

# Analysis of real-world passwords for social media sites

Mark J. Quinn

*James Madison University*

Follow this and additional works at: <https://commons.lib.jmu.edu/master201019>



Part of the [Information Security Commons](#)

---

## Recommended Citation

Quinn, Mark J., "Analysis of real-world passwords for social media sites" (2015). *Masters Theses*. 32.  
<https://commons.lib.jmu.edu/master201019/32>

This Thesis is brought to you for free and open access by the The Graduate School at JMU Scholarly Commons. It has been accepted for inclusion in Masters Theses by an authorized administrator of JMU Scholarly Commons. For more information, please contact [dc\\_admin@jmu.edu](mailto:dc_admin@jmu.edu).

Analysis of Real-World Passwords for Social Media Sites

Mark J. Quinn

A thesis submitted to the Graduate Faculty of

JAMES MADISON UNIVERSITY

In

Partial Fulfillment of the Requirements

for the degree of

Master of Science

Department of Computer Science

May 2015

## **Dedication**

This work is dedicated to my son, Christian. Your natural curiosity and love of learning inspires me. Never stop asking questions.

Mark

## Acknowledgments

I would like to begin by thanking Dr. Xunhua Wang, my thesis advisor. Your class on Cryptography ignited my interest in this field, and your challenge to create a story that is worth telling made writing this thesis an interesting, memorable, and collaborative experience. I would also like to show my appreciation to Dr. M. Hossain Heydari and Dr. Brett Tjaden for your valuable classes and for agreeing to serve on my thesis committee.

I am indebted to my parents for their love and support. The educational choices which you made for me early in my life have helped me achieve this milestone. Lastly, I would like to thank my wife, Ellen, for everything that you are and do. You will never know how much I appreciate having one of the smartest people I know sharing this journey and many of life's other journeys with me.

## Table of Contents

Dedication.....	ii
Acknowledgments .....	iii
Table of Contents.....	iv
List of Tables .....	vi
List of Figures.....	vii
Abstract.....	viii
Chapter 1 Introduction.....	1
<i>Overview</i> .....	1
Entity Authentication.....	1
Password Authentication .....	2
<i>Problem Statement</i> .....	4
<i>Contributions</i> .....	4
<i>Organization</i> .....	5
Chapter 2 Background Information and Related Work .....	6
<i>History of text based authentication</i> .....	6
<i>Historical evaluation of length and composition research</i> .....	7
<i>Historical evaluation of guessing metrics</i> .....	9
Chapter 3 The LinkedIn Dataset and Password Recovery Methods .....	12
Chapter 4 Analysis of LinkedIn Passwords.....	15

<b>Password Information</b> .....	16
<b>Digits</b> .....	18
<b>Uppercase Characters</b> .....	21
<b>Special Characters</b> .....	25
<b>Analysis summary</b> .....	27
<b>NIST</b> .....	28
<b>Chapter5 Conclusion</b> .....	31
<b>Areas of Further Research</b> .....	32
<b>Appendix A</b> .....	34
<b>Appendix B</b> .....	35
<b>Appendix C</b> .....	37
<b>Appendix C</b> .....	38
<b>Bibliography</b> .....	40

## List of Tables

Table 1: Password Information.....	16
Table 2: Top Ten Digits .....	18
Table 3: How Digits are used in 7+ Character Passwords .....	20
Table 4: Top Ten Case Mangling Rules for 7 characters .....	22
Table 5: Comparison of Lowercase v. Uppercase .....	23
Table 6: Top Ten One Letter Special Characters.....	25
Table 7: Top Ten Structures for Special Characters .....	26
Table 8: LinkedIn Totals based on NIST Guidelines .....	30
Table 9: Password cracking history.....	34
Table 10: Character Set for Entropy Calculations .....	37
Table 11: LinkedIn Entropy Totals.....	38

## List of Figures

Figure 1: Password Information .....	17
Figure 2: How Digits are used in 7+ Character Passwords .....	20
Figure 3: Top Ten Case Mangling Rules for 7 characters .....	22
Figure 4: Comparison of Lowercase v. Uppercase.....	24



## Abstract

Textual passwords have dominated all other entity authentication mechanisms since they were introduced in the early 1960's. Despite an inherent weakness against social engineering, keylogging, shoulder surfing, dictionary, and brute-force attacks, password authentication continues to grow as the Internet expands. Existing research on password authentication proves that dictionary attacks are successful because users make poor choices when creating passwords. To make passwords easier to remember, users select character strings that are shorter in length and contain memorable content, like personal identity information, common words found in a dictionary, backward spellings of common words, recognizable sequences, and easily guessed mnemonic phrases.

A number of these studies identify weaknesses found in passwords on social media sites [1] [2] [3] [4] [5]. However, this body of work fails to explore whether users choose more secure passwords on accounts that protect their professional online identity than they choose on accounts that are used for personal entertainment. In this study, we first cracked passwords from the over 6.4 million unsalted, SHA-1 hashed passwords stolen from the professional, social media site, LinkedIn. Next, we analyzed the length, character set composition, and entropy score of the passwords recovered. Then, we compared our results to the analysis of passwords performed by Weir, et al. on the RockYou! dataset to determine whether professionals protecting their online presence chose wiser passwords than social media site users who play online games.

In our analysis we found that the users of the professional, social media site, LinkedIn, chose more secure passwords than the users of the social media gaming site, RockYou!. LinkedIn passwords contained a greater percentage of numbers, special characters, and uppercase letters than RockYou!. We also found that the LinkedIn

passwords utilized special characters more frequently, but RockYou! passwords applied special character less predictably.

## Chapter 1

### Introduction

#### *Overview*

Entity authentication is the process of confirming the identity of an individual and in modern computing the most common method of performing authentication is through a text-based password. A three month study involving one half of a million users found that the average user owns roughly twenty-five accounts that require typing a password, and the user types about eight passwords a day [6]. Given the popularity of text-based authentication and the number of recent password attacks aimed at government facilities [7], web portals [8], social media networks [3], and gaming sites [5], it is understandable why protecting password-based systems is a major concern in the security industry.

#### **Entity Authentication**

Although password authentication dominates other forms of authentication, it is not the only method of authentication available. Computer applications authenticate a user's identity with three different methods: what the individual knows, what the individual physically possesses, and what physical characteristics make up the individual. Password authentication falls into the first category, what the individual knows. From a security perspective what an individual possesses, a token, and the physical characteristics which make up the individual, biometrics, store longer keys which contain more randomness than human beings can remember. The recommended RSA key size stored on a token is 256 characters, or 2048 bits [9], while an average person only remembers approximately seven characters of random data or 56 bits [10].

However, token-based authentication requires that the user possess the object in order to be recognized. If the token is not in the user's possession, authentication cannot

occur. A threat of being lost or stolen also exists. For this reason many token-based systems also require a pass code, which might be forgotten. Biometrics avoid the problems of tokens. Since physical characteristics travel with a person, biometrics cannot be lost, stolen or left somewhere. However biometric readings may differ from the authentication database due to injury, clothing, background noise, illness, and age. Biometrics, which stores attributes of a person's physical being, also raise privacy concerns for individuals who resist technology, and biometrics cannot be revoked easily.

The greatest drawback of using either token-based or biometric authentication is cost and ease of configuration. Both tokens and biometrics require hardware and software to act as an intermediary between the token reader / biometric reader and the system to which the user is being authenticated. Tokens and biometrics require an initial setup period and troubleshooting is more complex.

The disadvantages of using token based and biometric authentication do not apply to knowledge based authentication. Text-based passwords require no special hardware and travel well. They provide cost effective authentication, which is not susceptible to bad readings or changes in a person's physical characteristics. They are not likely to spawn a debate on privacy issues and can be easily revoked. Password based authentication also enables the users to manage their own accounts without the intervention of a system administrator. For these reasons, it is the preferred choice for access to email, social media, online banking, medical information, student records, credit card data, gaming, and web portal accounts.

### **Password Authentication**

Password-based authentication relies on a challenge-response system of verification. In the most basic form, an authentication server sends the client a request (the challenge) to

provide a password, and the client replies (responds) with the password. The server checks the password against a database of passwords for different users and authenticates the user if the account and passwords match. The inherent problem with storing plaintext passwords in the database file is that anyone with access to the file can read the contents, regardless of whether they have legitimate rights to read the file or not. To correct this flaw, secure authentication servers store passwords in an “encrypted” format.

Most modern implementations of password encryption involve cryptographic hashing algorithms. A cryptographic hashing algorithm transforms plain text of variable length to a fixed length hash using a one way function. The computation to create the hash from the plaintext is easy, but retrieving the password from the hash value should be extremely difficult. Effective cryptographic hashing algorithms should also resist collisions which occur when two known plaintext values are hashed to the same hash value. It is important to note that authentication servers never decrypt hashes. The servers simply compare the hashed password sent from the client to the hashed value in the password database.

With the appropriate systematic methods in place the security of password authentication relies on the user’s choices when selecting a password. Stated another way, in secure systems, the length and predictability of the password that a user chooses determines the success or failure of an attack. Many factors determine the choices users make when selecting a password: The ease of typing, value of the asset being protected, memorability, and knowledge of creating secure passwords all determine how the user chooses a password.

Despite the amount of research performed in understanding password choices, there is no definitive answer to the question “Do users select stronger passwords to protect accounts that store valuable information than they select for accounts that provide

entertainment?” In this paper we will focus on the value of the asset being protected by a password. We will attempt to reveal and analyze passwords from the approximately 6.5 million hashes stolen from the professional social media site, LinkedIn. According to reports from Sophos Security, the SHA1 hashed password appeared on a Russian hacking web in June 2012 following the breach [11]. In this analysis we will compare our findings to passwords from the online social gaming site, RockYou! to determine whether the value of the information being protected plays a role in the level of password security.

### ***Problem Statement***

This research attempts to answer the following questions:

1. What are the strengths and weaknesses of passwords which protect professional social media accounts, like LinkedIn, with respect to their length and character composition?
2. Do the accounts of professional social media sites, like LinkedIn, possess more complex passwords than accounts used for personal entertainment, like RockYou!?

### ***Contributions***

The results of this thesis research are two folded

1. We recovered 2,732,643 plaintext passwords from the SHA1 hashed LinkedIn dataset. Our research produced the largest number of passwords studied from a cracked dataset.
2. We analyzed the passwords and found that the passwords used on the professional, social media site LinkedIn possess greater complexity than the RockYou! passwords used to access games on social media sites.

## *Organization*

The remainder of this thesis shall be organized as follows: Chapter 2 gives some background information and related work. Chapter 3 describes how we obtained plaintext passwords from the LinkedIn dataset. Chapter 4 provides the details of our analysis. Concluding remarks appear in Chapter 5.

## Chapter 2

### Background Information and Related Work

#### *History of text based authentication*

Password authentication emerged in modern computing in 1961 on the MIT campus. System administrators needed a method for limiting the computing time granted to each student accessing the Compatible Time-Sharing System (CTSS), and they created the first user accounts protected by text-based passwords [12]. One year later, Allan Scherr, a PhD candidate needing more computing time than the four hours per semester allotted to each student, submitted a print request for a file named “UACCNT.SECRET”. The printout which appeared in Scherr’s mailbox the following day contained a list of usernames and passwords which he exploited to gain more access time and complete his thesis work. Scherr’s activities demonstrate that attempts to undermine password security have existed since the earliest attempt to protect a computer’s assets with passwords. Scherr’s password authentication malfeasance also begins a history of attempts to subvert text based password protection, including not only the unauthorized access of password databases, but also phishing, SQL injection, and dictionary, rainbow table, and brute force attacks. As attacking methods improve, so must the understanding of the forces which drive password choices and the policies which contribute to more secure authentication.

Research on password evaluation falls into two categories: studies of length and composition and studies of guessing. Studies which utilize length and composition metrics analyze a password’s strength by identifying the number of characters and the type of characters chosen. The type of characters may be numeric, alphabetic upper case, alphabetic lower case, special characters, and foreign/other characters. In contrast, studies which



incorporate guessing metrics aim to determine the likelihood of an attacker successfully recovering a password.

### ***Historical evaluation of length and composition research***

In an attempt to plug the largest security hole opened by text based authentication, numerous researchers have analyzed password length and composition characteristics attempting to determine their level of security and to improve their effectiveness. Morris and Thompson's [13] seminal study in 1979 discovered that 71% of the passwords cracked contained either short passwords (less than five characters) or all lowercase or all uppercase for longer passwords (5 or 6 letters). They concluded

“Given free choice, most people will choose their passwords from a restricted character set (e.g. all lower case letters) and will often choose words or names.” [13]

Their findings reveal a fundamental component of human nature; people want easy. By choosing passwords containing easy to type and easy to remember character strings, people gain access quickly and remember their credentials each time they login.

Every study which followed Morris and Thompson's paper echoes their findings regarding easily typed passwords. Spafford's 1991 study which collected passwords on 54 machines at the Department of Computer Sciences and the Computing Center found that the average length of unique passwords was 6.8 characters with 60.6% of the passwords containing all lowercase letters or all numeric characters. Wu, using a dictionary attack of passwords on a Kerberos realm in 1999, found 84.5 % of the passwords contained eight characters or less and 86% of the passwords could be typed without the Shift key [14]. In a more recent study from 2010, Devillers found that, in the RockYou! dataset, most of the passwords fell between six and eight characters. Lowercase only, digit only, and lowercase and digit passwords accounted for 91% of the passwords analyzed [2].

With respect to the factors of memorability, research also matches Morris and Thompson's findings that users will chose words and names rather than random character strings. A study by Riddle et al in 1989 categorized 6226 passwords based on their content, such as names, words, and random strings [15]. Like Riddle, Cazier and Medlin classified passwords based on content in their study of an e-business site with no password policy. Using a five grade scale with obvious names and numbers at the bottom of the scale and unrecognizable strings alphabetic, numeric, and special characters at the top, their research found that the mean of passwords fell below the midpoint of the scale [16]. Words and names are used so frequently in password selection that one psychologist called passwords "a 21st century Rorschach inkblot test" [17], because people choose password based on thoughts just below the subconscious, and these thoughts may possess an emotional component which make them easier to remember.

Other research proves that even when words are not used, memorability drives password choice. Kuo, et al., studied the mnemonic devices used to create passwords from memorable phrases. The team collected phrases based on Google searches of nursery rhymes, advertising slogans, television theme songs, and other memorable quotes. The 400,000 entry word list which they generated, based on a letter, number, or special character representing each word in the phrase, cracked 4% of the phrase based passwords collected [18]. In 2005, Narayanan and Shmatikov proved that the distribution of common passwords is consistent with the distribution of letters in the users' native language and launched a dictionary attack based on this model which recovered 67.6% of the passwords attempted [19].

Although research confirms that users select convenient and memorable passwords, only a few conflicting studies exist that compare password choices and demographics. Most

notably, Medlin and Cazier found that males logging onto an e-commerce site chose more complex passwords than their female counterparts [20]. They suggest that this difference may have resulted from the larger number of males in the workforce, where employers provide secure password training and enforcement. Bonneau's analysis of data on nearly 70 million Yahoo! passwords found the opposite to be true. With regard to demographic factors, like culture, gender, and race, password distributions remained consistent with respect to all of the various subpopulations. [21]

### ***Historical evaluation of guessing metrics***

Claude Shannon's groundbreaking work for Bell Laboratories [22] provides a starting point for another method for evaluating passwords, guessing metrics. Prior to Shannon's work, the definition of information was based largely on the work of Kant who defined information as a subjective reality perceived by the senses and assigned meaning by the mind [23]. Shannon's work focused not on the meaning or significance of information, but on the encoding and transmission of a message through a given channel. This perspective reinterpreted information as objective and quantifiable. "Entropy", the term Shannon gave to the amount of information gained (or the amount of uncertainty reduced), could be measured through a sequence of probabilities involving the symbols (i.e. letters, numbers, and punctuation) used to encode a message. The equation for entropy,  $H$ , can be determined for a discrete, random variable  $x$  using the calculation:

#### **Equation 1: Shannon entropy**

$$H(x) = - \sum_i^n P(x)_i \log_2 P(x)_i$$

where each variable  $(x_1, x_2, x_3, \dots, x_n)$  in the set of  $X$  possesses a probable outcome  $(p_1, p_2, p_3, \dots, p_n)$  in the distribution. As an example, if we wish to determine the information gained

by flipping a coin, we can represent  $x_H$  as landing on heads and  $x_T$  as landing on tails and the entropy calculation becomes  $H(x) = -(P(x)_H \log_2 P(x)_H + P(x)_T \log_2 P(x)_T)$ . If the coin has an even chance of landing on head as tails then  $P(x)_H$  equals 50% and  $P(x)_T$ , equals 50%. By inserting these values into the equation,  $H(x) = -((.5)\log_2(.5) + (.5)\log_2(.5))$ , a single coin flip results in 1 bit of information.

Although the entropy equation calculates the resources required to store or transmit hashed passwords, the calculation fails to provide a metric for determining the vulnerability of a system or the ease of guessing a password. A 2006 NIST publication adds ambiguity to the term *entropy* by redefining the word as a metric for measuring password complexity not the amount of information gained or the guessing difficulty. Rather than incorrectly attempt to determine the number of guesses required to by an attacker to uncover a password using Shannon's entropy or the NIST publication's entropy, the metric *guessing entropy* was introduced [24]. To compute the average number of guesses required to determine the value of  $X$  employing an optimal guessing strategy, the equation would be:

**Equation 2: Guessing entropy**

$$E[G(X)] = \sum_{i=1}^n i \cdot p_i$$

Another useful guessing model presented by Botz  [25] simulates a real world attacker's strategy by limiting the number of guesses to  $\beta$  per appears in the equation:

**Equation 3: Botz  guessing success rate**

$$\lambda_{\beta}(X) = \sum_{i=1}^{\beta} p_i$$

Pliam also simulates a real world attacker's approach to uncovering passwords by not trying to guess every account, but instead to crack a predetermined proportion of the accounts,  $\alpha$ , using the equation:

**Equation 4: Pliam guessing proportion**

$$\mu_{\alpha}(X) = \min\{j \mid \sum_{i=1}^j p_i \geq \alpha\}$$

Bonneau combines Botzaş and Pliam's work and shows an attacker limiting the number of guesses and stopping early if a desired number of accounts have been cracked as shown in the calculation:

**Equation 5: Bonneau guessing early termination**

$$G_{\alpha}(X) = (1 - \lambda_{\mu_{\alpha}}) \cdot \mu_{\alpha} + \sum_{i=1}^{\mu_{\alpha}} p_i \cdot i$$

## Chapter 3

### The LinkedIn Dataset and Password Recovery Methods

The LinkedIn dataset will be used in this study, and it contains 6,458,020 SHA1-hashed passwords stored in a Password Verification Data (PVD) file. The PVD file includes 2,936,840 unaltered, SHA1 hashed passwords and 3,521,180 SHA1 hashes in which the leading five characters of the hash have been replaced by zeroes. The SHA1 hashes of many common passwords, like “password”, “linkedin”, “123456” match hashes in the leading zeroes list when the first five characters are replaced with zeros. For this reason, it has been postulated that the zero-leading hash list contains the passwords which have already been cracked by tagging the first five characters with a zero in place of the actual characters.

We began our analysis by separating the PVD file into two groups: the straight SHA1 hashes and the zero-leading SHA1 hashes. We searched each list individually for duplicates and found none, so we started the process of cracking passwords. First we engaged in a rainbow table attack using rcracki\_mt software [26]. Since the zero-leading SHA1 list represents altered hashes, traditional SHA1 cracking tools, like rcracki\_mt, would not work with this list. We subjected only the straight SHA1 password list to the rainbow table attack. The key space for this attack consisted of one to seven characters with all combinations of uppercase letters [A-Z], lowercase letters [a-z], and digits [0-9]. Due to the resource intensive nature of rainbow table cracking, we divided the straight SHA1 hashes into smaller files containing between one thousand and five thousand hashes. These files were cracked distributively on twenty different machines on campus at James Madison University over the course of eight months.

Computer Science classes on-campus utilize lab workstations during the daytime hours, so we conducted our research at night, on weekends, and during breaks, when these

machines were available. Every evening, after classes ended, we ran twenty tasks stored in the Windows Task Scheduler which opened various putty sessions. Each putty session logged into a central on-campus server. The session also ran a remote login command which created an SSH session with a lab computer and ran a script which started the `rcracki_mt` software on a unique set of hashes to crack. Appendix B contains the entire bash script which starts new cracking sessions, resumes existing cracking sessions, and ensures that the cracking session ends at 7:50am, to prevent from interfering with normal operations of the lab.

After completing the rainbow cracking exploit, we removed the hashes of cracked passwords from the straight SHA1 hash list and began cracking passwords using John the Ripper (JtR) [27] on a single desktop computer with Backtrack 5. We installed a patch to JtR called “JtR-Jumbo-5-LinkedIn-SHA1.diff”. This patch adds a format which can be specified at the command line to crack both traditional SHA1 hashes and the leading zeros hashes found in the LinkedIn PVD file. We performed an initial crack of all passwords using the “all.lst” wordlist from openwall.com [27]. A Google search for additional dictionaries found the cracked passwords from the RockYou! security breach, foreign word dictionaries, and medical term wordlists. Then we ran dictionary attacks on each of the wordlists downloaded from the internet. In addition to these attacks, we also applied various default mangling rules found in the `john.config` file. For more mangling options, we downloaded and used rules from Kore Logic Security [28].

In comparing the cracked password results from the two lists, duplicate values began to emerge. Although we performed a duplicates search on the lists individually, a duplicate search between the lists was not performed. For the benefit of future studies and to accurately reflect the percentage of passwords found, we copied the original list of straight

SHA1 hashes, replaced the first five digits with zeros, and searched for duplicates. This search produced 670,781 redundant values between the two lists leaving a total of 5,787,239 hashes between the two lists. Of the passwords cracked we found 54,916 duplicates existed in the two cracked password lists. We removed these passwords from the leading zero passwords cracked list.

Our work cracking straight SHA1 hashes revealed 78,720 passwords through rainbow table cracking and 211,049 passwords through JtR with mangling rules. We uncovered 2,442,874 passwords in the zero leading hash list using JtR with mangling rules. These totals produce a combined sum of 2,732,643 cracked passwords or 47.22% of the entire PVD file with duplicates removed.

It is important to note that, of the significant research in the field of password analysis which we outlined in Chapter 2 and produced in **Table 9: Password cracking history**, our study provides the largest number of cracked passwords analyzed. The passwords in the RockYou! and the Yahoo! datasets contain more passwords, but neither of these datasets require password cracking. The RockYou! dataset originated from an SQL injection exploit which produced the 32 million account names and passwords in clear text. The Yahoo! password dataset of approximately 70 million passwords resulted from a cooperative effort between Bonneau, the author of the study, and Yahoo!, the web portal providing the data.



## Chapter 4

### Analysis of LinkedIn Passwords

No single, commonly agreed upon methodology exists for password analysis. Although many studies rely on length and composition metrics, the methods in which they are applied are as numerous as the studies themselves. Guessing metrics provide a consistent alternative to length and composition analysis, but relatively little password research with real data has been done using any single guessing metric. The NIST metric for measuring password complexity has also been used infrequently, because the definition of the term *entropy* is not consistent with the definition known to most researchers in the field.

To perform analysis on the LinkedIn dataset, we chose to use length and composition metrics. We will also calculate the NIST *entropy* value of the passwords recovered to measure their complexity. We will proceed with the understanding that these metrics measures password complexity, but fail to measure the level of difficulty that an attacker would have breaking into the account. For example a password like Button123! would have the complexity to pass most authentication requirements, but a simple mangling rule would make this password easy prey for a standard dictionary attack. We chose not to use guessing metrics in our analysis because the LinkedIn dataset contains unique hashes. In order to work with guessing metrics, each hash in the dataset must have a probability distribution over the entire set of possibilities.

In order to gauge the level of complexity of LinkedIn passwords, we will compare the passwords revealed through our work with length and patterns in character composition found in Weir's analysis of the RockYou! dataset [5]. Since RockYou! builds social media games and advertising products for sites, like Facebook, MySpace, and Friendster, we expect

to find less complex and less random authentication strings than the password on LinkedIn which presents professionals to a network of other professionals online.

### Password Information

To begin our analysis, we will identify the length and character type composition of passwords from the two datasets. Weir generated a table which captures passwords greater than various lengths beginning with seven characters. His findings appear in **Table 1:**

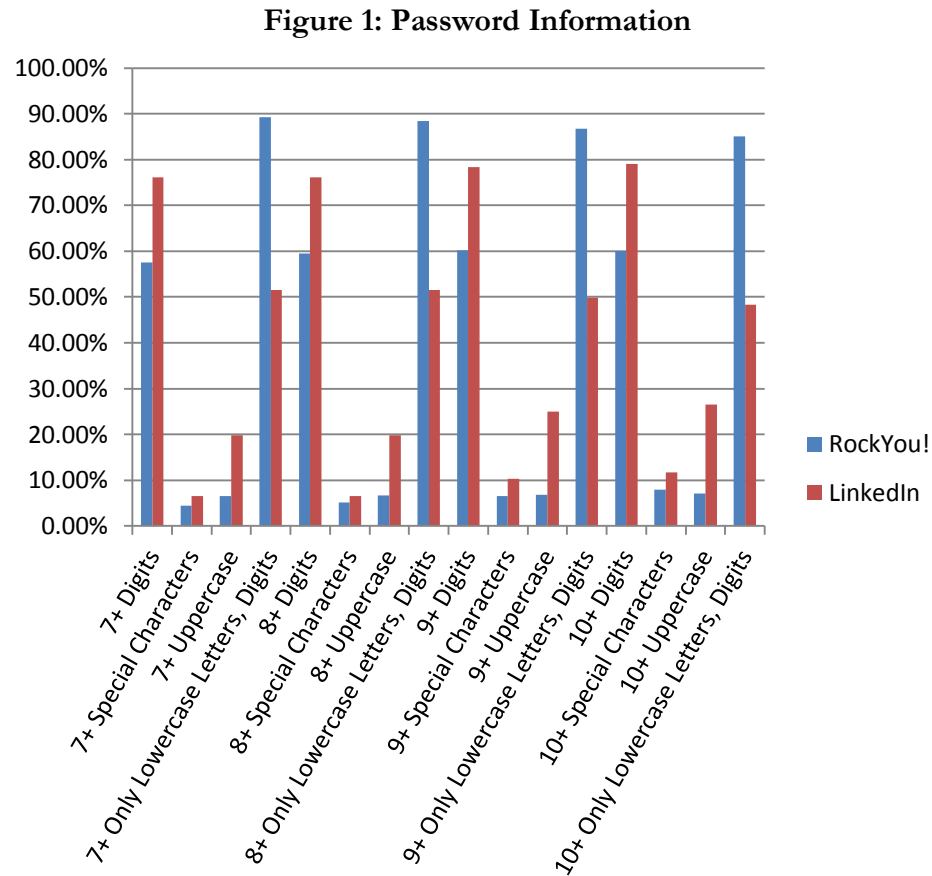
**Password Information** with our results added for comparison.

**Table 1: Password Information**

Character Set Contains	7+ Chars		8+ Chars		9+ Chars		10+ Chars	
	RY!	LI	RY!	LI	RY!	LI	RY!	LI
Digits	57.5%	74.2%	59.5%	76.1%	60.2%	78.3%	60.0%	79.0%
Special Characters	4.4%	5.6%	5.1%	6.6%	6.6%	10.3%	8.0%	11.7%
Uppercase	6.5%	19.1%	6.7%	19.8%	6.9%	25.0%	7.1%	26.5%
Only Lowercase Letters, Digits	89.2%	50.9%	88.4%	51.5%	86.7%	49.8%	85.1%	48.3%

Weir found that as passwords grew in length the percentage of digits, special characters, and uppercase letters also increased. The LinkedIn dataset echoes this trend. The percentage of digits, special characters and uppercase letters in the LinkedIn passwords

not only grew, but also surpassed the RockYou! dataset for each password length as shown in Figure 1: Password Information.



The only category where the RockYou! dataset surpasses LinkedIn is the Lowercase Letters and Digits. Studies have shown that the most common passwords are also the easiest to type and contain only lowercase letters [1] [2] [13]. In some studies, easy to type passwords, which include only lowercase letters and numbers, account for over 50% of the passwords [1] [29]. Other studies show this number much higher; exceeding 80% [2] [13] [14]. The RockYou! dataset not only exceeds the LinkedIn dataset in these easier to type passwords at every length, but also exceeds the LinkedIn dataset by a significant amount (36.80-38.30%).

## Digits

Users frequently add numeric characters to passwords. This practice increases the complexity of the password by expanding the key space and strengthening the security of the authentication system. A password which contains only lowercase characters of the alphabet possesses a key space of  $26^L$ , where  $L$  is the length of the password. When upper case characters are added, the key space increases to  $52^L$ . The key space grows to  $62^L$ , if digits are introduced to the uppercase and lowercase letters.

Of the 2,160,956 passwords cracked with seven or more characters, our study produced 1,603,813 passwords with digits (74.21%). This is much higher than the 26% found by Wu and 31.7% found by Spafford [29], but not quite as high as the 81% found by Schneier [4]. Weir does not provide a percentage of passwords which contain digits. Instead his study approaches the use of digits by analyzing the most frequently used number strings and the placement of numbers within a password.

**Table 2: Top Ten Digits** shows the detailed results of Weir's study of frequently used number strings. We added our findings on a random sample of 100,000 passwords with digits.

**Table 2: Top Ten Digits**

Rank	RY! Digit	RY! Percentage	LI Digit	LI Percentage
1	1	10.98%	1	10.07%
2	2	2.79%	2	3.01%
3	123	2.29%	3	2.60%
4	4	2.10%	0	2.27%
5	3	2.02%	4	2.03%
6	123456	1.74%	123	1.99%

Rank	RY! Digit	RY! Percentage	LI Digit	LI Percentage
7	12	1.49%	01	1.54%
8	7	1.20%	12	1.49%
9	13	1.07%	7	1.28%
10	5	1.04%	5	1.26%

The results for this analysis demonstrate that users rarely choose digits randomly. In the two top ten lists, eight of the ten values are shared: 1, 2, 3, 4, 5, 7, 12, and 123. Weir discovered that 26.72% of the digits used appear in the top ten list. Our study closely mirrors the RockYou! dataset with 27.54% of the total digits used appearing in the top ten list.

Studies by Schneier [4], Devillers [2], and Wu [14] recognize the popularity of the number one. Both our study and Weir's analysis place the number one in the top position in number rankings. The number one not only holds the top position in the Weir study, but five of the items in Weir's top ten list also contain a one. Digits containing a one comprise 20% of the total passwords with digits. Our top ten list also places one in the top position with 10.07% of the total which is slightly lower than Weir found. The number one appears as one of the digits in three other values in our top ten list, 123, 01, and 12. In total, the number one appears in 15.09% of the top ten list for the LinkedIn sample. With respect to the most popular digits found, numbers in the passwords from the LinkedIn dataset are less predictable than the numbers found in the RockYou! dataset.

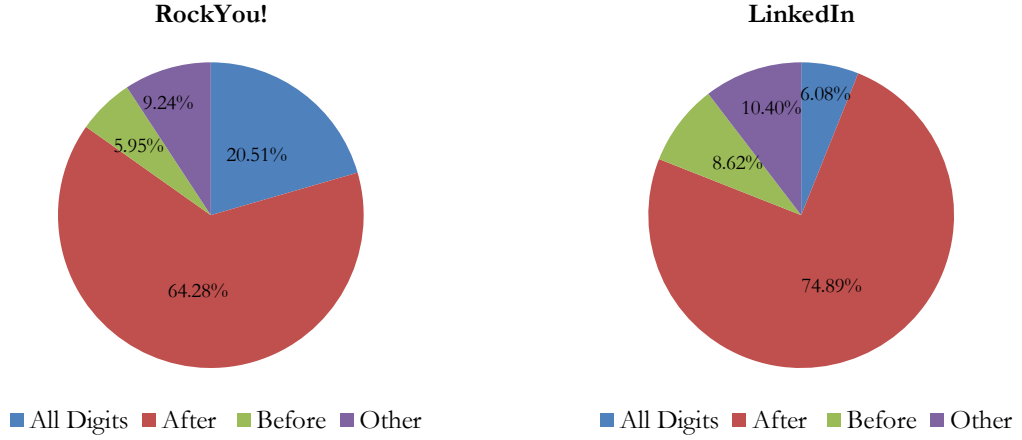
To fully understand the use of digits in a password, consideration must also be given to where the digits in a password appear. Common patterns in digit placement indicate a lower level of security than random placement. A comparison of passwords in Weir's

RockYou! dataset and the LinkedIn dataset appear in **Table 3: How Digits are used in 7+ Character Passwords** and **Figure 2: How Digits are used in 7+ Character Passwords**.

**Table 3: How Digits are used in 7+ Character Passwords**

Location	Example	RockYou!	LinkedIn
All Digits	1234567	20.51%	6.08%
After	password123	64.28%	74.89%
Before	123password	5.95%	8.62%
Other	passw0rd, pass123word, p1a2ssword, ...	9.24%	10.40%

**Figure 2: How Digits are used in 7+ Character Passwords**



As described previously, passwords which contain only digits use a smaller key space and provide less security than passwords of the same length with both numbers and letters. The RockYou! dataset triples LinkedIn with respect to digit only passwords. Several studies [1], [2], [4], [14] [19] mention another common pattern of simply prepending or appending digits to an alphabetic string of characters. In this regard, the LinkedIn password set has a greater percentage than RockYou!. When LinkedIn users type numeric digits, 83.51% of

these passwords prepend and append digits to alphabetic characters as opposed to the 70.23% by RockYou! users.

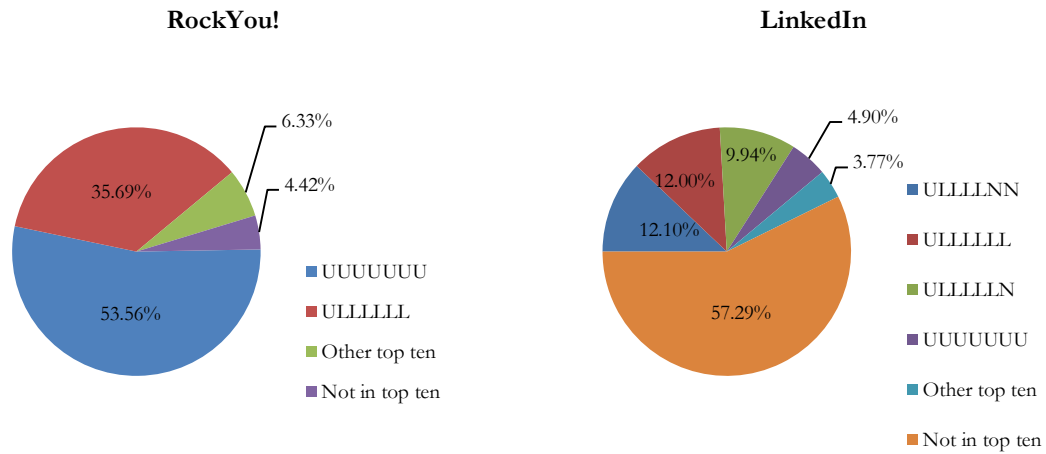
The most uncommon location where digits will be found is in the middle of a password. Only a slight 1.16% difference exists between the LinkedIn and RockYou! datasets. For passwords which contain digits, LinkedIn passwords possess numbers in the middle of the password 10.40% of the time, while RockYou! passwords contain numbers in the middle 9.24% of the time.

### **Uppercase Characters**

As with adding digits to a password, the addition of uppercase characters increases the size of the key space which, in turn, makes a password more secure. Like with digits, password security depends not only on the existence of uppercase characters to expand the key space, but also on placement of these characters in unpredictable locations within passwords. Users frequently incorporate capital letters in passwords by typing all capital letters or by typing an initial capital letter followed by all lower case letters [1] [19]. Weir noticed that these two capitalization patterns account for almost 90% of all passwords in the RockYou! dataset which contain an uppercase character. The composition of 7 character passwords which include at least one uppercase character appear in **Table 4: Top Ten Case Mangling Rules for 7 characters** and **Figure 3: Top Ten Case Mangling Rules for 7 characters**. The LinkedIn dataset appears to the right of the totals from the Weir study for comparison.

**Table 4: Top Ten Case Mangling Rules for 7 characters**

Rank	RockYou! String:	RockYou! Probability	LinkedIn String	LinkedIn Probability
1	UUUUUUU	53.56%	ULLLLNN	12.10%
2	ULLLLLL	35.69%	ULLLLLL	12.00%
3	ULLLULL	1.05%	ULLLLLN	9.94%
4	LLLLLLL	1.03%	UUUUUUU	4.90%
5	ULLLLLLU	0.90%	ULLNNNN	3.03%
6	ULLULLL	0.85%	UUUNNNN	2.48%
7	ULULULU	0.68%	ULLLNNN	2.43%
8	LLLLLLU	0.62%	UUUUUNN	1.57%
9	UULLLLL	0.61%	ULLNLLL	1.13%
10	UUULLLL	0.59%	UUUUUUN	1.07%

**Figure 3: Top Ten Case Mangling Rules for 7 characters**

In the RockYou! dataset, all uppercase passwords account for 53.56% of the total number of passwords while the single uppercase character followed by all lowercase characters comprise another 35.69%. For an attacker aiming to exploit the easiest targets,



applying mangling rules with only these two variations would crack almost nine out of ten vulnerable passwords with capital letters.

The LinkedIn dataset produced 477,945 passwords with at least one uppercase character (17.49%). In comparison, to Weir's results, the LinkedIn passwords provided a greater degree of security. The top ten mangling patterns for LinkedIn account for only about 50% of all passwords. The two most common patterns, all uppercase and first letter uppercase follow by lowercase letters, rank fourth and second respectively in the LinkedIn list and contribute only 16.90% to the number of passwords with a capital letter. To approach the 90% that the first two mangling rules cover in the RockYou! dataset, an attacker would need to apply 359 different mangling rules with uppercase characters to the LinkedIn dataset.

The Weir study determined that passwords which contain at least one uppercase letter had a higher probability of having at least one digit or special character. Our analysis of the LinkedIn dataset produced similar results as shown in **Table 5: Comparison of Lowercase v. Uppercase** and

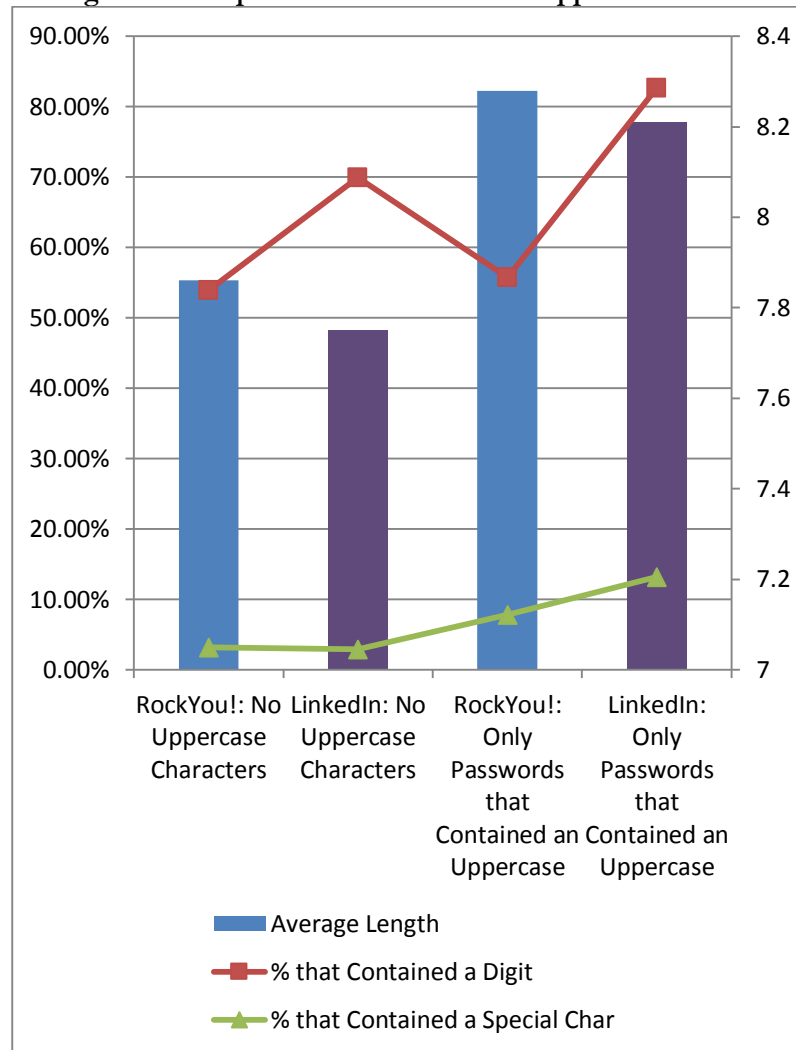
**Figure 4: Comparison of Lowercase v. Uppercase.**

**Table 5: Comparison of Lowercase v. Uppercase**

Metric	RockYou!: No Uppercase Characters	LinkedIn: No Uppercase Characters	RockYou!: Only Passwords that Contained an Uppercase	LinkedIn: Only Passwords that Contained an Uppercase
Average Length	7.86 characters	7.75 characters	8.28 characters	8.21 characters
% that Contained a Digit	53.93%	69.94%	55.74%	82.68%
% that Contained	3.15%	2.96%	7.87%	13.18%

a Special Char				
----------------	--	--	--	--

**Figure 4: Comparison of Lowercase v. Uppercase**



A significant difference exists between the two datasets in the digit and special character category. In the RockYou! dataset, Weir observed only a 1.81% difference between passwords with at least a single uppercase character and passwords without an uppercase character. The difference observed in the LinkedIn dataset was 12.74%. In the category of at least one uppercase character and at least one special character, Weir's dataset started slightly higher (3.15% ) for passwords without uppercase characters than the LinkedIn dataset (2.96%). However, the RockYou! dataset only grew to 7.87% in this

category, while the LinkedIn dataset more than quadrupled in size (13.18%) to surpass RockYou!.

### Special Characters

We will conclude our analysis of character sets by identifying habits and patterns of users when they construct passwords which contain special characters. Of the 2,732,643 LinkedIn passwords cracked, 129,744 passwords (4.75%) contain at least one special character. As we discovered with numbers, users rarely choose special characters randomly. Our study revealed that of the thirty-three special characters, 90.72% of the passwords which contain a single letter special characters appear in the top ten ranking. Weir discovered that a similarly high number (85.34%) of single letter special characters in the RockYou! dataset appear in the top ten ranking. The results of both studies appear in **Table 6: Top Ten One Letter Special Characters**.

**Table 6: Top Ten One Letter Special Characters**

Rank	RockYou!: Special Character	RockYou!: Probability	LinkedIn: Special Character	LinkedIn: Probability
1	.	17.81%	!	24.08%
2	_	14.72%	@	17.93%
3	!	11.34%	#	10.46%
4	-	10.25%	\$	8.18%
5	<space>	8.72%	.	7.98%
6	@	7.19%	*	7.43%
7	*	6.54%	_	6.35%

Rank	RockYou! Special Character	RockYou! Probability	LinkedIn: Special Character	LinkedIn: Probability
8	#	3.92%	-	5.16%
9	/	3.01%	%	1.60%
10	&	1.84%	&	1.55%

As with the study of digits, these two datasets share eight of the top ten special characters: exclamation point (!), underscore ( \_), period (.), commercial at(@), hyphen(-), asterisk (\*), pound (#), and ampersand (&). These eight characters represent 73.61% of the one letter special character passwords in the RockYou! dataset which Weir studied and 80.94% of the LinkedIn passwords which we revealed. In terms of key space, the attacker would be able to reduce the amount of work required in a brute force attack by searching only one quarter of the special characters for a return of approximately 75% of the passwords with one special character.

To better understand how a user constructs passwords which contain special characters, **Table 7: Top Ten Structures for Special Characters** identifies the placement of special characters in seven character passwords.

**Table 7: Top Ten Structures for Special Characters**  
A=Alpha, D=Digit, S=Special

Rank	RockYou! Structure	RockYou! Probability	LinkedIn: Structure	LinkedIn: Probability
1	AAAAAAS	28.50%	AAAAASD	13.60%
2	AAASAAA	7.87%	AAAASDD	13.22%
3	AAAASDD	6.32%	AAAAAAS	11.95%
4	AAAAASD	6.18%	AAAADDS	6.25%

Rank	RockYou! Structure	RockYou! Probability	LinkedIn: Structure	LinkedIn: Probability
5	AASAAAA	3.43%	AAAAADS	6.12%
6	AAAASAA	2.76%	AAASDDD	5.38%
7	AAAAASA	2.64%	AAADDDS	4.81%
8	SAAAAAS	2.50%	AAASAAA	3.86%
9	ASAAAAA	2.38%	AAAASAA	2.13%
10	AAAAASS	2.17%	AADDDDS	1.61%

Again we observe overlap in the lists. Five character patterns appear in both the RockYou! and the LinkedIn dataset: AAAAAAS, AAASAAA, AAAASDD, AAAASD, and AAAASAA. The pattern of appending a single special character to the end of an alphabetic string holds the top ranking in the RockYou! dataset with 28.50% of the passwords with special character passwords using that pattern. At 13.60%, the top ranking item in the LinkedIn list is less than half of the top ranking item for RockYou!. For passwords with special characters, the top ten list comprises 64.75% of all patterns for the RockYou! dataset and 68.93% of all patterns for the LinkedIn dataset. As shown in **Table 1: Password Information**, the LinkedIn dataset contains a higher percentage of passwords with special characters than the RockYou! dataset. However, users generally chose the same characters and add them in a predictable manner making their use less effective from a security perspective than if they had been applied in a less conventional manner.

### Analysis summary

Since numbers, uppercase characters, and special characters expand the key space and add to the complexity of a password, we studied each of these character types to better

understand whether RockYou! users or LinkedIn users made more secure password choices. In both the RockYou! and LinkedIn password lists, we found that as the length of the password increases the percentage of digits, uppercase characters, and special characters also increase. We noticed that the use of common numbers was consistent between the two password lists with eight of the top ten digits shared between the lists. When digits appeared, the RockYou! dataset used them more predictably and less securely with three times the number of all digit passwords than the LinkedIn dataset. We discovered that the use of uppercase characters was more prevalent in the LinkedIn dataset and more secure. When uppercase characters were used in the RockYou! dataset, almost 90% of the passwords containing uppercase characters utilized only two patterns: all uppercase letters and a first character uppercase followed by all lowercase characters. In comparison, the top two patterns for LinkedIn account for only 24.10% of the total number of passwords with capital letters. Special characters appeared more frequently in the LinkedIn dataset also. The LinkedIn users chose more predictable special characters and used them in common patterns more frequently than RockYou! users.

## **NIST**

Next we calculated the password entropy as defined in the NIST 800-63-1 Electronic Authentication Guidelines [30]. NIST entropy provides a measurement system for rating a password's complexity based on the attributes of length and character composition. It should be made clear that we will not use Shannon's definition of *entropy* which is the default definition of the term in the field of information theory. Verhuel [31] and Massey [32] question the effectiveness of applying Shannon's entropy calculations to evaluate password strength, and Weir [5] proves that entropy fails to measure the effectiveness of password creation rules. We will also not use the NIST entropy score to gauge the password's

resistance to an attack. Instead, we calculated entropy in this study to measure the complexity of the two datasets and to compare the results.

Up to this point, we have included passwords with foreign letters and other characters in our analysis because they also contain characters from the uppercase, lowercase, number and special character sets found in **Table 10: Character Set for Entropy Calculations**. For a more concise calculation of entropy on the LinkedIn dataset, the 328 passwords with a character outside of the 95 letter character set appearing in will be omitted.

The NIST standard calculates password entropy using the following criteria:

1. Assign the first character 4 bits.
2. Assign characters two through eight 2 bits each.
3. Assign characters nine through twenty 1.5 bits each.
4. Assign characters greater than twenty 1 bit.
5. Add 6 bits for both upper case and non-alphabetic characters.
6. Add 6 bits for an extensive dictionary check.

The LinkedIn dataset contains no passwords with less than six characters. For this reason, the minimum entropy of the dataset begins at 14, and 522,186 passwords possess this value. The maximum entropy value of 48 belongs to two passwords: “Thequickbrownfox666.” and “Supercal1frag111st1c”. We calculated the average number of entropy bits in the LinkedIn passwords cracked to be 18.95. In comparison Weir observed passwords with entropy values in the range of 4 to 32. Weir does not provide an average entropy calculation on the RockYou! dataset.

**Table 8: LinkedIn Totals based on NIST Guidelines** displays the total number of passwords which meet each category for calculating entropy. **Table 11: LinkedIn Entropy Totals** in the Appendix C provides calculations for arriving at the average.

**Table 8: LinkedIn Totals based on NIST Guidelines**

Character length	No uppercase or dictionary bonus	Number of passwords which contain uppercase and either a special character or number	Number of passwords which receive dictionary bonus only	Number of passwords which contain uppercase and either a special character or number and receive dictionary bonus
6	522186	26277	7916	15308
7	460506	38655	28019	34812
8	722772	97138	70982	35354
9	241420	52442	21280	18268
10	140298	31936	15440	13198
11	50858	13804	7461	7096
12	22545	6191	4432	3771
13	7827	2170	1682	1776
14	3154	779	885	737
15	1027	206	302	264
16	485	75	302	149
17	30	4	9	17
18	9	1	13	8
19	6	1	3	2
20	4		3	2
21			2	
22			2	
23			1	



## Chapter5

### Conclusion

Text based passwords dominate all other forms of authentication. Their popularity stems from the cost effective means that they provide for controlling access to a system. Despite this widespread use, the choices that human beings must make to create memorable passwords limit the level of security that password authentication can possess. Our study echoes the findings of previous research which proves that users frequently select easy to remember and type passwords which provide more security than the password creation rules require, but fail to provide enough security to deter an attacker. In this thesis we first recovered 2,732,643 plaintext passwords from the SHA1 hashed LinkedIn Password Verification Data file. Using open source password cracking tools, we uncovered 47% of the passwords in the LinkedIn dataset.

Next we analyzed the passwords and compared our results with the RockYou! passwords that Weir studied to discover that the passwords used to authenticate users on the professional social media site LinkedIn provides slightly better security than the passwords which authenticate users of social media RockYou! games. We found that LinkedIn passwords contained a greater percentage of numbers, special characters, and uppercase letters than RockYou!. We discovered fewer all digit passwords which have a relatively small key space. The LinkedIn dataset also possessed greater complexity when uppercase letters appeared in passwords. Although special characters appeared more frequently in the LinkedIn dataset, the RockYou! dataset demonstrated a higher level of complexity than LinkedIn with respect to one letter special character usage. Lastly we found that the entropy based on the NIST 800-63-1 Electronic Authentication Guidelines started and ended higher

for LinkedIn passwords than RockYou!. The lack of data with respect to average entropy in the Weir study prevents a direct comparison of this metric.

### **Areas of Further Research**

The hashes cracked and analyzed in our study possess two significant differences with the dataset used in the Weir study of RockYou! passwords. The password lists were obtained in completely different manners, and the password lists contain differences with respect to the existence of duplicate values.

An attacker exploited the RockYou! authentication system using a SQL injection attack. This type of exploit allowed the Weir study to randomly shuffle and analyze groups of plaintext passwords from the entire RockYou! list. The LinkedIn dataset appeared on a Russian hacking website hashed. Through our efforts we obtained only about half of the passwords, and these passwords represent the easiest to crack. Although we have proven that the LinkedIn list contains greater complexity with respect to length, character composition, and NIST entropy average, we cannot prove the true difference between the datasets, because the most complex, hashed, LinkedIn passwords cannot be used in our analysis.

The RockYou! dataset studied by Weir contains over 32 million plaintext passwords. In this list we identified 14,344,386 unique passwords. Of this list 2,459,759 passwords appeared more than once. Although duplicates appeared in the LinkedIn list, the list of zero-leading hashes is unique and the list of straight SHA1 hashes is unique. The duplication between lists occurred when we replaced the first five characters of the straight SHA1 hashes with zeroes and searched for duplicates in the zero-leading hashes. Since each duplicates only appeared one time, it seems that duplication resulted from copying hashes between lists and not moving hashes between lists.

For future research, we would recommend cracking a greater percentage of LinkedIn hashes to better understand the password choices made on the professional social media site. We would also recommend performing an independent study of unique RockYou! passwords, rather than using results which contain duplicates from the Weir study. As an alternative if a dataset from a personal social media site, like Facebook or Google+, became available, analyzing and comparing unique passwords would provide a better understanding of the choices that users make when protecting different types of social media accounts.

## Appendix A

**Table 9: Password cracking history**

Name	Year	Passwords	Number (%) Cracked	Source	Obtained	Cracking Method
Morris, Thompson	1978	3289	2381 (86%)	Unknown	“gathered from many users over a long period of time”	Dictionary
Klein	1990	13797	3340 (24.2%)	Survey requesting /etc/passwd file on Unix machines	Voluntary solicitation of friends	Dictionary, Mangling
Spafford	1992	19100	13787 (72.18%)	54 machines in the Department of Computer Sciences and Computing Center	collection software installed on machines	Dictionary
Wu	1999	slightly over 25,000	2045 (approximately 8.18%)	authentication server of a large Kerberos realm, serving over 25,000 users	collected from authentication server	Dictionary and simple mangling
Kuo	2004	144	4%	Survey Participants to Craigslist ad	Voluntary solicitation from craigslist and student-hosted bulletin board	Dictionary, Mangling, BruteForce
Narayanan, Shmatikov	2005	142	67.6	Passware	provided by Passware	Dictionary – modified version of Markovian filter Rainbow
Schneier	2006	34000	N/A	MySpace	Phishing attack	N/A
Dell Amico, Michiardi, Roudier	2010	Italian 9317 Finish 15,812 My Space 33,671		three: Italian IM server Finnish web Forum MySpace 2006	Italian? Finnish – publicly disclosed, My space - phishing	Dictionary, mangling, Markov chains
Weir, Aggarwal, Collins Stern	2010	32 M	N/A	Rock You!	SQL injection attack (unencrypted)	N/A
Devillers	2010	32M	N/A	RockYou!	SQL injection attack	N/A
Bonneau	2012	70M	N/A	Yahoo!	Cooperation of web portal	N/A

## Appendix B

```
#!/bin/bash

#bash script to create an ssh connection and
#run rcracki_mt between 4:00pm and 7:50am
#if a previous session exists for the crack
#it will resume. Otherwise a new session will
#be started.

#k Gives hour M gives minute 10:30 is 1030
time="date +%k%M"

InFile="$HOME/HashInputFiles/Hashesupto915000x5k.txt"
SesFile=Hashesupto915k
SessionFile="$HOME/Hashesupto915k.session"
OutFile="$HOME/HashOutputFiles/passListupto0915000x5k.out"

#create ssh connection and run rainbow crack
#if a previous session does not exist start new session otherwise
resume previous session
if [ ! -f ${SessionFile} ]; then
    echo "142 File not found..."
    if [ ! -f ${OutFile} ]; then
        echo "142 New Session!"
        ./rcracki_mt -l ${InFile} -s ${SesFile} -t 8 -o ${SessionFile}
~/RT_Files &
    else
        echo "142 Execution Complete!" >> runCracks.out
        sleep 10h
        exit
    fi
else
    echo "142 File found - resume existing sessions!"
    ./rcracki_mt -l ${InFile} -r -s ${SesFile} -t 8 -o
${SessionFile} ~/RT_Files &
fi

#get the pid of the ssh connection which was just created
pid=$(pgrep -u quinnmj ssh -n)
echo "PID = $pid"
```

```

#If the time is between 4pm and 11:59pm or 12:00am and 7:50am
continue running script otherwise exit script
intimerange="1"
while [ $intimerange -eq 1 ]; do
    if ([[ $(eval "$time") -ge 1600 ]] && [[ $(eval "$time") -le 2359
]]) || ([[ $(eval "$time") -ge 000 ]] && [[ $(eval "$time") -le 750
]]);then
        echo "Running at `date +%D%t%T`."
        sleep 10m # Use sleep 10m
    else
        intimerange=0
    fi
done

#kill ssh connection
kill -9 $pid

echo "Exiting at `date +%D%t%T`."

```

## Appendix C

**Table 10: Character Set for Entropy Calculations**

A	N	a	n	0	@	\	>
B	O	b	o	1	#		,
C	P	c	p	2	\$	]	<
D	Q	d	q	3	%	}	Space
E	R	e	r	4	^	[	
F	S	f	s	5	&	{	
G	T	g	t	6	*	‘	
H	U	h	u	7	(	“	
I	V	i	v	8	)	;	
J	W	j	w	9	-	:	
K	X	k	x	~	_	/	
L	Y	l	y	`	=	?	
M	Z	m	z	!	+	.	

## Appendix C

**Table 11: LinkedIn Entropy Totals**

Character length	No uppercase or dictionary bonus	Total bits no uppercase or dictionary bonus	Uppercase but no dictionary bonus	Total bits Uppercase no dictionary bonus	No uppercase but dictionary bonus	Total bits uppercase but dictionary bonus	Both uppercase and dictionary bonus	Total bits Both uppercase and dictionary bonus
6	522186	7310604	26277	525540	7916	158320	15308	398008
7	460506	7368096	38655	850410	28019	616418	34812	974736
8	722772	13009896	97138	2331312	70982	1703568	35354	1060620
9	241420	4707690	52442	1337271	21280	542640	18268	575442
10	140298	2946258	31936	862272	15440	416880	13198	435534
11	50858	1144305	13804	393414	7461	212638.5	7096	244812
12	22545	541080	6191	185730	4432	132960	3771	135756
13	7827	199588.5	2170	68355	1682	52983	1776	66600
14	3154	85158	779	25707	885	29205	737	28743
15	1027	29269.5	206	7107	302	10419	264	10692



Character length	No uppercase or dictionary bonus	Total bits no uppercase or dictionary bonus	Uppercase but no dictionary bonus	Total bits Uppercase no dictionary bonus	No uppercase but dictionary bonus	Total bits uppercase but dictionary bonus	Both uppercase and dictionary bonus	Total bits Both uppercase and dictionary bonus
16	485	14550	75	2700	302	10872	149	6258
17	30	945	4	150	9	337.5	17	739.5
18	9	297	1	39	13	507	8	360
19	6	207	1	40.5	3	121.5	2	93
20	4	144			3	126	2	96
21					2	86		
22					2	88		
23					1	45		

Total bits                    51774839.5  
 Total passwords            2732302  
 Average bits                18.94916429

## Bibliography

- [1] M. Dell'Amico, P. Michiardi and Y. Roudier, "Password Strength: An Empirical Analysis," in *INFOCOM'10: Proceedings of the 29th Conference on Information Communications*, 2010.
- [2] M. M. Devillers, "Analyzing Password Strength," *Radboud University Nijmegen*, pp. 1-10, July 2010.
- [3] B. Schneier, "MySpace Passwords Aren't So Dumb," 2006. [Online]. Available: <http://www.wired.com/politics/security/commentary/securitymatters/2006/12/72300>. [Accessed 21 February 2014].
- [4] B. Schneier, "Real-World Passwords," 14 12 2006. [Online]. Available: [https://www.schneier.com/blog/archives/2006/12/realworld\\_passw.html](https://www.schneier.com/blog/archives/2006/12/realworld_passw.html). [Accessed 21 2 2014].
- [5] M. Weir, S. Aggarwal, M. Collins and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS'10)*, 2010.
- [6] D. Forencio and C. Herley, "A Large-Scale Study of Web Password Habits," *WWW '07 Proceedings of the 16th International Conference on World Wide Web*, pp. 657-666, 2007.
- [7] L. Didio, "Cyberattack Prompts DoD to Boost Security," *ComputerWorld*, vol. 32, no. 9, p. 14, 1998.
- [8] J. Lyne, "Yahoo Hacked And How To Protect Your Passwords," *Forbes*, 31 January 2014. [Online]. Available: <http://www.forbes.com/sites/jameslyne/2014/01/31/yahoo-hacked-and-how-to-protect-your-passwords/>. [Accessed 14 April 2014].
- [9] EMC, "4.1.2.1 What key size should be used?," [Online]. Available: <http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/key-size.htm>. [Accessed 25 October 2014].
- [10] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," *The Psychological Review*, pp. 81-97, 1956.
- [11] J. Finkle and J. Saba, "LinkedIn suffers data breach," 6 June 2012. [Online]. Available: <http://www.reuters.com/article/2012/06/06/net-us-linkedin-breach-idUSBRE85511820120606>. [Accessed 9 September 2014].
- [12] M. V. Wilkes, *Time-sharing computer systems*, New York: Elsevier, 1968.

- [13] R. Morris and K. Thompson, "Password security: a case history," *Unix Programmer's Supplementary Documentation*, November 1979.
- [14] T. Wu, "A Real-World Analysis of Kerberos Password Security," *Network and Distributed System Security Symposium*, February 1999.
- [15] B. L. Riddle, M. S. Miron and J. A. Semo, "Passwords in use in a university time sharing environment," *Computers and Security*, vol. 8, no. 7, pp. 569-578, 1989.
- [16] J. A. Cazier and A. D. Medlin, "Password Security: An Imperical Investigation into E-Commerce Passwords and Their Crack Times," *Information Systems Security*, vol. 15, no. 6, pp. 45-55, 2006.
- [17] L. W. Andrews, "Passwords Reveal Your Personality," *Psychology Today*, vol. 35, no. 1, p. 16, 2004.
- [18] C. Kuo, S. Romanosky and L. F. Cranor, "Human Selection of Mnemonic Phrase-based Passwords," *SOUPS '06 Proceedings of the Second Symposium on Usable Privacy and Security*, pp. 67-78, 2006.
- [19] A. Narayanan and V. Shmatikov, "Fast Dictionary Attacks on Passowrds Using Time-Space Tradeoff," in *CCS'05: Proceedings of the 12th ACM Conference on Computer and Communications Security*, 2005.
- [20] B. D. Medlin and J. A. Cazier, "An Investigative Study: Consumers Password Choices on an E-Commerce Site," *Journal of Information Privacy & Security*, vol. 1, no. 4, pp. 33-52, 2005.
- [21] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," *2012 IEEE Symposium on Security and Privacy*, pp. 538-552, 2012.
- [22] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379-423, 1948.
- [23] I. Kant, "The Project Gutenberg EBook of The Critique of Pure Reason," 4 February 2013. [Online]. Available: <http://www.gutenberg.org/files/4280/4280-h/4280-h.htm>. [Accessed 10 February 2015].
- [24] C. Cachin, *Entropy Measures and Unconditional Security in Cryptography*, Zurich: Ph.D Dissertation, 1997.
- [25] S. Botzas, "Entropies, Guessing, and Cryptography," Department of Mathematics, Royal Melbourne Institute of Technology, Melbourne, 1999.

- [26] FreeRainbowTables.com, "Free Rainbow Tables," 10 April 2014. [Online]. Available: <https://www.freerainbowtables.com/>. [Accessed 10 April 2014].
- [27] Openwall.com, "Openwall: Bringing Security into Open Environments," [Online]. Available: [www.openwall.com](http://www.openwall.com). [Accessed May 2014].
- [28] Kore Logic Security, "'Crack Me If You Can' - DEFCON 2010," 2012. [Online]. Available: <http://contest-2010.korelogic.com/rules.html>. [Accessed 22 9 2014].
- [29] E. H. Spafford, "Observations on reusable password choices," Perdue University, West Lafayette, July 1992.
- [30] "NIST Special Publication 800-63-1: Electronic Authentication Guideline," National Institute of Standards and Technology, Gaithersburg, MD, 2011.
- [31] E. Verhuel, "Selecting secure passwords," in *Topics in Cryptology – CT-RSA 2007*, Berlin, Springer Berlin Heidelberg, 2007, pp. 49-66.
- [32] J. L. Massey, "Guessing and Entropy," *Proceedings of 2012 IEEE international symposium on information theory*, p. 204, 1994.
- [33] N. van Heijningen, "A State-of-the-Art Password Strength Analysis Demonstrator," Rotterdam University, Rotterdam, Netherlands, 2013.
- [34] R. E. Smith, *Authentication From Passwords to Public Keys*, Addison-Wesley Professional, 2001.
- [35] A. Scherr and T. Van Vleck, "Compatible Time-Sharing System (1961-1973): Fiftieth Anniversary Commemorative Overview," IEEE Computer Society, Washington, DC, 2011.
- [36] D. V. Klein, "'Foiling the Crackers': A Survey of and Improvements to, Password Security," *Proceedings of the Second USENIX Security Workshop*, pp. 5-14, 1990.
- [37] C. Herley, P. van Oorschot and A. S. Patrick, "Passwords: If We're So Smart, Why Are We Still Using Them?".
- [38] M. E. Hellman, "A Cryptanalytic Time - Memory Trade-off," *IEEE Transactions on Information Theory*, Vols. IT-26, no. 4, pp. 401-406, July 1980.
- [39] D. C. Feldmeier and P. R. Karn, "UNIX Password Security - Ten Years Later," in *Proc. CRYPTO '89*, 1989.

- [40] J. Borst, B. Preneel and J. Vandewalle, "On the Time-Memory Tradeoff Between Exhaustive Key Search and Table Precomputation," in *19th Symp. on Information Theory in the Benelux*, Veldhoven, Netherlands, 1998.
- [41] LinkedIn, "About LinkedIn," 2014. [Online]. Available: <http://press.linkedin.com/about>. [Accessed 29 March 2014].
- [42] P.-H. Kamp, "LinkedIn Password Leak: Salt Their Hide," *Queue*, vol. 10, no. 6, p. 20, 7 June 2012.
- [43] J. Fontana, "Breach clean-up cost LinkedIn nearly \$1 million, another \$2-3 million in upgrades," 3 August 2012. [Online]. Available: <http://www.zdnet.com/breach-clean-up-cost-linkedin-nearly-1-million-another-2-3-million-in-upgrades-7000002115/>.
- [44] M. Zviran and W. J. Haga, "Password Security: An Empirical Study," *Journal of Management Information Systems*, vol. 15, no. 4, p. 161, 1999.
- [45] R. Lewand, Cryptological Mathematics, Washington, D.C.: The Mathematical Association of America, 2000.
- [46] "Openwall," 11 April 2014. [Online]. Available: <http://www.openwall.com/>. [Accessed 11 April 2014].
- [47] Microsoft, "<https://www.microsoft.com/security/pc-security/password-checker.aspx>," 30 March 2014. [Online]. Available: <https://www.microsoft.com/security/pc-security/password-checker.aspx>. [Accessed 30 March 2014].
- [48] D. E. Robling Denning, *Cryptography and Data Security*, Addison-Wesley, 1982, p. 100.
- [49] P. Oechslin, "Making a Faster Cryptanalytic Time-Memory Tradeoff," *Proc. CRYPTO '03*, vol. 2729 of LNCS, pp. 617-630, 2003.
- [50] B. D. Medlin and J. A. Cazier, "An Investigative Study: Consumers Password Choices on an E-Commerce Site," *Journal of Information Privacy and Security*, vol. 1, no. 4, pp. 33-52, 2005.