Statistics on password re-use and adaptive strength for financial accounts

Daniel V. Bailey, Markus Dürmuth, Christof Paar

Horst Görtz Institute for IT-Security, Bochum, Germany
danbailey@sth.rub.de,{markus.duermuth|christof.paar@rub.de}

Abstract. Multiple studies have demonstrated that users select weak passwords. However, the vast majority of studies on password security uses password lists that only have passwords for one site, which means that several important questions cannot be studied. For example, how much stronger are password choices for different categories of sites? We use a dataset which we extracted from a large dump of malware records. It contains multiple accounts (and passwords) per user and thus allows us to study both password re-use and the correlation between the value of an account and the strength of the passwords for those accounts.
The first contribution of our study shows that users in our sample choose (substantially) stronger passwords for financial accounts than for low-value accounts, based on the extracted passwords as well as publicly available lists. This contribution has implications for password research, as some widely-used lists contain passwords much weaker than those used in the real world (for accounts of more than low value). In our second contribution, we measure password re-use taking account values into account. We see that although high-value passwords are stronger, they are re-used more frequently than low-value passwords – valuable passwords are identical to 21% of the remaining passwords of a user. Before our study, little was known about password re-use for different account values.

1 Introduction

Most online services rely on users to choose passwords for authentication. Conventional wisdom holds that users generally do not choose passwords that are difficult to guess. Several alternatives to passwords have been proposed, but none has found widespread use, as passwords are easy to deploy, scale to an Internet-wide user-base, and are easy to understand for the users. Alternative technologies have a number of drawbacks: hardware like *smart cards* and *security tokens* can be expensive to procure and manage for Website operators and can be perceived as an impediment to usability. *Biometric* identification systems also require extra hardware, can raise privacy issues, and many biometrics are not secret (e.g., we leave fingerprints on many surfaces we touch).

Research on password security started as early as 1979 [21], and a number of studies has been published since then. One important aspect is password re-use:

As user accounts proliferate, users are forced to remember more and more passwords that must also remain confidential and hard to guess. In response, users often re-use the same password for multiple logins to keep the number of passwords they have to remember low [12]. When a re-used password leaks, then the security of all accounts using the same password is at risk. Even worse, a rogue service could collect login credentials (typically usernames and corresponding passwords) and test those at other sites, which is hard to detect for the user.

While it is known from leaked password lists that users choose weak passwords on average, there is some hope in the community that users choose stronger passwords for those accounts that are valuable[1] (see, e.g., [14]). However, this belief has never been justified with real-world data. Actually, there is very little data available on high-value passwords at all, which is most likely the reason why so little research has been conducted on the topic. However, this question is of importance, as a number of studies in the literature use low-value passwords as input. Arguably, research on password security is most interesting for high-value passwords, as these are most likely the target of actual attackers.

The lack of available data is one of the main problems in password research as, by their nature, passwords are meant to be confidential. For password re-use, most available studies use data collected in user surveys, where great care has to be taken to ensure ecological validity, see Section 1.1 for more details. Our data show the type of site influences the password strength chosen by a user – at least, for users of malware-infected PCs. As explained later, we feel our data provides insight into the behavior of average users as well. This work is the first, to our knowledge, studying real-world password data collected by malware.

1.1 Related work

As early as 1979 it has been observed [21] that users tend to choose weak passwords that are susceptible to so-called *dictionary attacks*. This problem has been studied extensively since then and led to development of tools such as *John the Ripper* [7] and *HashCat* [13]. More advanced password guessers based on Markov models have been presented recently [22, 30]. To increase the strength of passwords against guessing attacks, various strength measures for passwords have been developed [6]. Strength of passwords generated under different password rules were studied in [16]. However, with very few exceptions in the older literature, relevant research was conducted on passwords for low-value sites, and it is not known if users choose stronger passwords for more valuable sites.

Several studies examine password re-use. Ives et al. give an interesting high-level overview of password re-use [15], including some examples of actual damage done by password re-use. Florencio and Herley [9] present a large-scale user study

[1] The question which accounts have high value is another topic which is out of the scope of this text. We will use financial-related sites as high-value sites, which we believe reflects the intuition of most users. While from a security point of view, email accounts might be at least as valuable, as they are often used as fall-back security mechanism for other sites, it is unknown how many users take this into consideration.

on passwords including password re-use, where they collected their data from browsers running the Windows Live toolbar (from consenting participants). They could only test for exact re-use of passwords and get a moderate bias, both due to the study's design. They find that each user has, on average, 25 accounts and 6.5 passwords, i.e., each password is used for 3.9 accounts.

In a lab study, Gaw and Felten ask participants to conclude when groups of passwords are similar [11]. This approach is adopted to preserve confidentiality of participant passwords, but the resulting similarity measure is vague. They find between 2.2 and 3.2 accounts per password. Komanduri et al. measure the effect of password-creation rules [17]. When asked to create a new eight-character password subject to one of a set of rules, the resulting password had an entropy between 27 and 34 bits. In addition, they report on users' self-reported re-use. As rules become more complicated, the number of re-used passwords increases from 17% up to 33%. Dhamija and Perrig interviewed 30 people and reported that participants used one to seven unique passwords for ten to fifty websites [8]. Sasse et al. report that in a study of 144 employees, an average of 16 passwords was reported, but this was not limited to online activities [25]. Two other studies have based estimations of people's passwords through surveys. Brown et al. surveyed college students, finding an average of 8.18 password uses with 4.45 unique passwords [4]. Riley also used a survey to focus on online accounts, finding students had an average of 8.5 password-protected accounts [24].

Bonneau [1] used two password lists that both included usernames, allowing re-use measurement between these two sets. Both lists were hashed, so the hashes first needed to be cracked. From those accounts cracked in at least one list, 49% of users used the same password for accounts on both sites, however, this does not take into account those accounts that weren't cracked, and thus we cannot say what the actual re-use rate is. It seems plausible to assume that those passwords that weren't cracked belong to more security-savvy users and that those have a lower rate of password re-use, so 49% most likely constitutes an upper bound. Furthermore, in the same text Bonneau recognizes the need for a study on password re-use based on *account value*.

An industry advisory [28] considers password re-use by utilizing a browser plug-in intended to warn about phishing attempts against banking passwords that also detects re-use. They report that "73% of users share the online banking password with at least one nonfinancial website" [28]. However, not many details are given about the exact setup and distribution of the plug-in. In addition, to compare the results with other work we would require at least the average number of accounts per user they recorded. Forcing users to periodically change their passwords is a common technique to prevent attackers from using leaked passwords. Zhang et al. use a database of 7700 accounts to examine the difficulty in guessing the replacement password given the expired one [31]. They found in this attack model that 41% of replacement passwords could be guessed in a few seconds.

	Abbrev.	Size	Users	PWs/User	Avg. PW Length
Malware-List					
– total	MW	3531	1721	2.05	9.01
– financial	MW-Fin	177	134	1.3	9.1
– rest	MW-Btm	3354	1686	2.09	9.01
Mt. Gox (Bitcoin)	BITC	61,020	61,020	1	–
RockYou	RY	32 M	32 M	1	7.89
Carders.cc	CC	5062	5062	1	7.59

Table 1. Overview of the password lists we used

1.2 Paper outline

In Section 2 we describe our datasets and the preprocessing steps we used. Section 3 studies the relation between password strength and account value. In Section 4 we study password re-use, concluding with some final remarks in Section 5.

2 The datasets

This section describes our dataset along with some limitations.

2.1 The Malware dataset

A username-password combination allows a thief to log into an online-banking account and, depending on further security measures, drain it of funds. Malware such as Trojans specifically target Web browsers and aim to capture the data entered in HTML forms. Many organizations attempt to monitor this situation, working with law enforcement, alerting affected banks, and publishing reports on emerging threats. To do so, they obtain some of this data for forensic purposes. As the malware captures all of the HTTP POST data, IP address, operating system version and so on can prove to be valuable clues on infection rates and locations. One of these organizations allowed the present authors limited access to this data. No additional malware output was collected to enable the present work. The dataset contains thousands of passwords captured by the Zeus Trojan in late 2012. We partition the Malware list (MW) into two (disjoint) subsets according to the perceived value to a user.

– **High-value accounts: Financial passwords (MW-Fin)** The first sample includes passwords for accounts at banks, insurers, brokers, and related financial services. An attacker takeover of one of these accounts has obvious financial consequences and therefore heightened risk perception on the part of the user. We selected the accounts by searching the domain names for financial-services related keywords in a variety of languages, as well as

a number of known banks. In addition, we manually inspected the domain names to ensure accuracy. This yielded a set of 177 passwords from 95 different domains, however, the number of distinct entities/sites/... is smaller as a single bank may service several domains.

– **Lower-value accounts: Remaining passwords (MW-Btm)** This group includes all other passwords. This sample includes well-known email providers and social networks. This yielded a set of 3354 passwords from 1134 different sites; Facebook is the largest subset with 1163 passwords.

Perceived value of accounts The perception of security risk is known to be subjective and based on several factors including dread of consequences [23]. The compromise of a user's financial account obviously carries real financial consequences for a user. Malware-promulgating attackers generally aim to take over an online account and drain it of funds – or perhaps to gather enough sensitive personal information to fraudulently apply for a credit card or loan (often called identity theft). We therefore group these financial-site passwords together (similar to [10]). This classification includes sites likely to directly enable transfer of funds including banks, credit-card issuers, stock brokers, and insurers. In addition, we include those housing sensitive information that would enable identity theft such as payroll processors and tax collectors. In fact, other accounts can be quite valuable to users as well, e.g., email accounts can be used for password recovery. However, for the overwhelming majority of users (except maybe celebrities, bloggers, and corporations) the compromise of a user email or social-networking account leads to practically no direct financial consequences. A common sentiment seems to be that "Nobody wants to read my private email."

A potential objection to this approach is that intuitively, restricting the high-value passwords to financial passwords leaves out other valuable passwords. However, we show in Section 3.2 that the passwords in MW-Fin are significantly stronger than those in MW-Btm. Even if some high-value passwords (not from financial sites) are still contained in MW-Btm, this means that the real difference is even stronger than we measured. So the error incorporated from this rather narrow interpretation would lead us to underestimate the disparity, reinforcing our main point.

Bias in the dataset There are two potential sources of bias in the dataset: First, we have a subset of the total set of passwords collected by the malware only, and second, this bigger set could be biased as it is collected by malware and infections are not necessarily uniform across all users. The sub-sample contains a wide variety of sites in many countries and languages, and represents a snapshot of the actual data available to criminals. Second, only those users infected by malware are included in our dataset. We feel the results will likely hold true for many other users given the widespread nature and infection methods of Zeus. According to industry reports, Zeus variants have been observed in the wild on Windows (IE, Firefox, and Chrome browsers), Android, and Blackberry, including one of every 3000 computers worldwide [27]. Most Zeus infections occur on PCs with up-to-date antivirus software. Zeus spreads through email

attachments as well as "drive-by infection," where a user need only visit a Web site to become infected, thanks to a malicious JavaScript redirection. These properties to a certain extent dispel the misconception that malware afflicts only unsophisticated or careless users. The malware dataset does not include any captures from MacOS or Linux, which induces some amount of bias. However, Windows represents more than 85% of desktops accessing the Internet, so the bias due to operating system choice is expected to be small [26].

Furthermore, we expect the comparison of the strength of passwords in MW-Fin and MW-Btm (see Section 3.2) to be largely unaffected by these biases, as both lists are sampled with the same bias, and there is no indication that the bias is such that it affects both subsets in a different way.

2.2 More password sets

To relate our findings to previous work, we compare against several other sets.

- **RockYou (RY)** One of the largest lists publicly available is the RockYou list (RY), consisting of 32.6 million passwords that were obtained by an SQL injection attack in 2009. The passwords were leaked in plaintext, but all metadata like username was stripped from the list before it was leaked to the public. This list has two advantages: First, its large size yields precise information also about less-common passwords; second, it was collected via an SQL injection attack therefore affecting all the users of the compromised service, basically removing sample bias. These advantages have made RockYou studies quite popular in the literature, so we use it to compare our findings with previous work.
- **MtGox/Bitcoin (BITC)** Bitcoin is an emerging decentralized currency based on computation; several merchants accept these "coins" for goods and services, and researchers are studying it in terms of cryptography, privacy, and economics. Bitcoins can also be exchanged for other currencies, one of the biggest websites (at the time) providing this service was Mt.Gox. The password file containing over 61 thousand hashed passwords leaked online in 2011 [20].
- **Carders.cc (CC)** Carders.cc is an online forum where hackers would negotiate stolen assets like passwords and credit-card account numbers. In 2010, Carders.cc was itself subject to a hacking attack that exposed its database of 5,062 passwords [18]. Most interesting about this list for our purposes is the user population. Unlike general social-networking sites, this one catered to users who are (on average) both technology-savvy and security aware.

2.3 Ethical considerations

All passwords analyzed in this paper were leaked by attacks in 2012 and collected in support of other efforts to track and remediate malware infections. No additional data was collected specifically to enable the present work. This fact means that practical attackers have already had independent access to our

datasets for more than two years. It is not expected that the present work aids actual attackers.

Nevertheless, special care was taken to avoid our work leading to a new consolidated source of passwords for actual attackers. The Malware passwords themselves were stored in a private enclave away from typical corporate or academic networks. They were only available to researchers through a chain of proxies with a full complement of firewalls, network monitoring, and data-loss prevention tools meant to stop data exfiltration. Then, direct access was eschewed in favor of scripts that returned only statistics to the researchers.

3 Correlation of password strength and account value

One unique aspect of the Malware password list is that it contains passwords for multiple accounts per user, and those are sampled in the same way and with the same bias. A closer inspection reveals that it often contains passwords for accounts that are more valuable than others, which allows us to compare the strength of those passwords. These findings are relevant for several reasons: First, it allows us to test if "users choose more secure passwords for accounts of value", which is often expressed in the literature when weak passwords are discovered. Second, previous password studies are limited to the available data: collections of passwords from social networks or portals like Yahoo! [3]. By contrast, our study includes passwords directly used to protect financial transactions.

3.1 Measures for password strength

At a high level, we can distinguish measures that evaluate *resistance against a specific password cracker* (either by directly attacking them, or by using mathematical models to estimate their effectiveness), and approaches that consider the *distribution of passwords*. While the former are motivated by practice and model common attacks pretty well, they depend on the specific software tool and do not necessarily generalize well. The latter are based on mathematical models and thus have a clearly defined meaning and are (in some sense) optimal, but not necessarily relevant for practice.

Entropy measures A number of different entropy measures have been used to measure the security of passwords. For an overview, as well as more details about the one presented here, see [2, 3]. Guessing entropy [19, 5] measures the average number of guesses that the optimal attack needs in order to find the correct password. However, a practical attacker is generally satisfied with breaking into a certain fraction of accounts, which guessing entropy does not take into account. *Partial guessing entropy* [2] (or *α-guesswork*) takes this into account.

For $0 \leq \alpha \leq 1$ let $\mu_\alpha = \min\{i_0 \mid \sum_{i=1}^{i_0} p_i \geq \alpha\}$ the minimal number so that the guesses cover at least a fraction α of the passwords, and let $\lambda_\alpha = \lambda_{\mu_\alpha} = \sum_{i=1}^{\mu_\alpha} p_i$ the actual sum (which is greater or equal to α). With these, partial

guessing entropy is defined as

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

Intuitively, the first term is contributed by those passwords that weren't guessed in the allotted number of guesses, and the second term is contributed by those password that were. We want to express this in "bits of information" to be able to compare it with other measures more easily. This is done as follows:

$$\tilde{G}_\alpha(X) = \log\left(\frac{2 \cdot G_\alpha(X)}{\lambda_\alpha} - 1\right) + \log\frac{1}{2 - \lambda_\alpha} \tag{2}$$

where the "correction term" $\log \frac{1}{2-\lambda_\alpha}$ is used to make the metric constant for the uniform distribution (see [2] for a more detailed explanation).

We have two reasons to deviate from this approach. First, to approximate the distribution of X (i.e., the probabilities p_i) requires a *large sample set size* which is much larger than the Malware dataset; second, one can be interested in getting a more *comparable metric for a specific attack*. So we are interested in a combination of both, as we define in the following.

John the Ripper A well-known and wide-spread tool for password cracking is John the Ripper (JtR) [7]. JtR uses a number of heuristics that show good performance in practice. It can be configured in a wide range, but in the standard mode of operation it performs the following steps. (i) *Single crack mode.* In a first step, JtR tries items like username and home-directory name both as-is as well as simple "mangling" modifications like appending digits or reordering letters. (ii) *Wordlist mode.* JtR comes with a dictionary of 3557 common passwords to try, along a set of mangling rules that are applied. (iii) *Incremental mode.* A mode that can try all possible combinations.

We used John the Ripper 1.7.8-jumbo-5, instrumented with an additional patch that logs the number of passwords tried. The Jumbo version supports counting of plaintext guesses as well as hashed passwords. The number of guesses by JtR is often seen as a good approximation for the practical strength of a password. As we are only interested in comparing the strength of different password lists the specific choice does not make a substantial difference. We measured the number of guesses needed for passwords in each of our lists. JtR can run for a very long time generating every possible password of a given length, so for practical considerations, we aborted JtR after a given amount of time.

For the Malware datasets, we ran JtR against every password. As the other lists contained substantially more passwords, we randomly sampled 1024 from each. The BITC list consists of salted, hashed passwords and so required substantially more computation time to check the validity of a guess. The plaintext lists required only the generation of a guess and not the hash. Experimentally, approximately the same number of hashed guesses can be checked in 10 hours of CPU time vs. one minute of CPU time for plaintext.

Experimental Entropies We combine the theoretical entropy measure with real-world password-guessing tools to yield what we will call *experimental guessing entropy*. As discussed before, there are two main reasons why we do not use the above measures directly, namely that entropy measures require substantial knowledge about the distribution and thus a large number of samples to approximate it with sufficient precision, and second that the output of guessing tools is specific to that tool and hard to compare with other results.

To calculate the *experimental partial guessing entropy (EPGE)*, we use JtR to determine the proportion of passwords cracked for a given number of guesses (see, e.g., Figure 1). We then use these probabilities in Equations 1 and 2 instead of the optimal attack considered originally. (i.e., we replace the optimally ordered p_i's with probabilities from a realistic attack with JtR.) Note that the resulting entropy values depend on the guessing tool used, and are in general higher than the true partial guessing entropy, which assumes an optimal guesser. As our main objective is to compare different distributions, the EPGE suffices.

Statistical significance One potential concern about the Malware dataset is that the set is rather small (at least when compared with password lists such as the RockYou list with 32.6 million passwords), which leads to a higher variance of the results. We used an approach similar to that by Bonneau [3] to determine bounds on these effects. We sampled more than 80 uniformly chosen subsets of the RockYou password list of the appropriate size (3354 and 177, respectively), ran password guessing and entropy estimation (for both $\alpha = 0.05$ and $\alpha = 0.2$) just as for the Malware dataset, and measured the confidence interval for the level 95%. We find that the confidence intervals for a sample size of 177 passwords (as in MW-Fin) is ± 0.7 for $\alpha = 0.05$ and ± 4.4 for $\alpha = 0.2$, and for a sample size of 3354 samples (as in MW-Btm) is ± 0.35 for $\alpha = 0.05$ and ± 1.18 for $\alpha = 0.2$.

These (empirical) confidence intervals are determined from another list of passwords that might have different characteristics compared to the Malware list, and thus have to be considered carefully. However, as the differences in entropy that we will encounter later are substantially larger then these confidence intervals, they give us a reasonable level of trust.

3.2 Results: Malware dataset

In the first experiment, we compare the strength of financial password (MW-Fin) to the remaining passwords (MW-Btm).

Running the Experiments We run JtR as described in Section 3.1 against the two Malware sub-lists (see Section 2), i.e., the Malware list filtered for financial passwords (MW-Fin) and the remaining (MW-Btm), the most interesting and directly comparable set of passwords. All passwords in these lists are available in plaintext, so no hash operations need to be performed and running time is no concern. Note that John the Ripper is highly customizable, with the potential for dictionaries and rules tailored for particular lists. This approach clearly gives the best performance in practice. As our purpose here is simply to *compare*

	$\alpha = 5\%$	10%	15%	20%
RockYou	15.1	15.0	17.4	22.2
Malware-Btm	16.2	25.0	28.8	–
Malware-Fin	23.3	28.6	–	–
MtGox	26.1	–	–	–
Carders	14.4	13.6	13.8	14.0

Table 2. Experimental Partial Guessing Entropy for several success probabilities, using John the Ripper as baseline as explained in Section 3.1. A dash means that fewer passwords have been cracked for the respective list, so the respective value cannot be computed from the data at hand.

guessing success among the various lists, the default settings will suffice. Our presented results do not reflect JtR's performance potential in absolute terms.

Results and Discussion Figure 1 shows the resulting graphs, plotting the number of password guesses on the x-axis and the fraction of accounts guessed successfully on the y-axis, Figure 2 shows a more detailed view for fewer guesses. Table 2 gives the experimental guessing entropy for $\alpha \in \{5\%, 10\%, 15\%, 20\%\}$ (along with the entropy values for other password lists we will evaluate in the following). From Figure 1 one can already see quite clearly that the different lists have different strength. This is substantiated by the entropy values in Table 2, where we see that, e.g., for $\alpha = 5\%$ we get entropies of 16.2 and 23.3, respectively. From the measurements in Section 3.1 we conclude that this difference is significant.

While this result is not surprising, prior to the present work limitations in the lists available to researchers served as a hindrance. This is because the differences may be more due to userbase, differing password policies, or other causes than a specific behavior on the part of a user population. With the Malware dataset and the two subsets MW-Fin and MW-Btm, we are finally in the situation to have several passwords sampled under comparable situations. In addition, we believe that the dataset has less bias than lists obtained by phishing. But even though the data is somewhat biased, both sublists MW-Fin and MW-Btm are biased in the same way, so the results for both are comparable.

One explanation for the difference in password strength could be that different password rules were deployed. This is hard to verify, as the passwords are from a wide variety of different accounts, and there is no efficient method to obtain the password rules that were in place at the time a password was changed. However, we are convinced that password rules do not explain the differences for two reasons: First, in general password rules are known to be a bad indicator for password strength [17, 29], so we would not expect such a strong impact on password security. Other studies in the literature [10] find password rules are determined more by a site's need to be usable than the extractable financial value.

3.3 Results: Comparing with other datasets

More interesting insights come from comparing the results for the Malware lists MW-Fin and MW-Btm with other lists of passwords that are publicly available; this also allows us to relate our results to previous research. With the same parameters as in the previous section, we run the experiments again for other password lists: (i) the RockYou (RY) list as examples for a list of weak passwords that is regularly used in the literature that allow us to compare our results with other work, and (ii) the carders.cc (CC) list which represents a list of low-value passwords for a technology-savvy userbase (on average). Again, these lists are available in plaintext, so no hashing is required.

Results and Discussion We see that even the weaker passwords in MW-Btm are significantly more secure than those in the lists RY and CC (and MW-Fin is even more secure). For $\alpha = 0.1$ the entropy of MW-Btm is 25.0, whereas entropies for RY and CC are below 15.0, and similarly for $\alpha = 0.15$. (For $\alpha = 0.5$ the entropy values are still somewhat similar, which means that the weakest passwords are similarly weak in those lists.) This can also be seen in the graphs in Figures 1 and 2. (Our estimates from Section 3.1 suggest that most differences are significant.)

An additional difference between those lists of weak passwords and the MW-Btm list is that the former contain passwords from a single low-value site only, whereas MW-Btm probably contains a mix of low- and medium-value (and potentially even some high-value) sites. Another factor that needs to be taken into account is that the Malware list contains data that was collected in 2012, while the RockYou list leaked in 2009. The enforced password rules as well as user's perception of password security have improved over those years, which explains the difference at least in part.

The list RY is regularly used in the literature both as example for weak passwords and as benchmark for work on password security, which might not be an optimal choice in light of our results.

3.4 Results: Comparing with MtGox

In a third experiment, we compare our results with the only other list of high-value passwords that we are aware of, the MtGox list, which is a representative for a list of high-value passwords for another technology-savvy userbase (on average). This list is, however, not available in plaintext, but in hashed form, which is a likely explanation why it has only rarely been studied in the literature. As only some passwords can be guessed in a reasonable amount of time, this results in a sample bias towards weaker passwords. In fact, this is one of the reasons why we use JtR, as we can directly compare results without additional bias. Running time for these tests is substantial. As we need to compute a hash to check the validity of a guess, it takes about ten hours of CPU time to check the same number of guesses as in one minute of CPU time when passwords are in plaintext. For this reason we only make 345,000,000 guesses per password hash, which limits the resulting graphs.

Results and Discussion We can see that the passwords in the BITC list are substantially more secure than those from any other list we consider. For $\alpha = 0.05$, we estimate an entropy of 26.1 for BITC, which is only moderately harder than the estimate of 23.3 for MW-Fin, but substantially harder than all other estimates which fall in the range from 14.4 bits to 16.2 bits. There are two potential explanations for these difference: First, these passwords (often) protect direct monetary value, so users could be inclined to protect that money and choose strong passwords, and second, the userbase of the Bitcoin system and thus MtGox has a technology-savvy userbase, which are likely to choose stronger passwords. When additionally considering the CC list, which is the least secure one we tested, the following explanation seems likely: Technology-savvy users might differentiate between high-value accounts (BITC, 26.1 Bits for $\alpha = 0.05$) and low-value accounts (CC, 14.4 Bits for $\alpha = 0.05$), whereas the average user differentiates less between high-value (MW-Fin, 23.3 Bits for $\alpha = 0.05$) and low-value accounts and low-value passwords (MW-Btm, 16.2 Bits for $\alpha = 0.05$).

4 Password re-use

Several studies show that users often re-use passwords for several accounts, to decrease the amount of information they need to memorize. However, re-use can be problematic, because single passwords leak quite frequently, which then puts a number of accounts at risk. Even worse, malicious website operators have direct access to a user's login credentials, and misuse will go unnoticed.

However, the studies available so far suffer from two problems: Most work uses *surveys* to answer such questions about re-use, which requires great care to avoid biased data (e.g., caused by the observer-expectancy effect). Moreover, people might not recall every site where they have registered (see Section 1.1 and Table 3). We are aware of two studies not using surveys: one [9] uses data that was collected for another purpose and was available only hashed (i.e., similarity could not be measured). The other [1] used two leaked password lists that both contained usernames, however, both were hashed and thus only those could be compared that were broken by a brute-force attack, which constitutes a bias towards weak passwords, which likely also have higher re-use.

A crucial aspect that has not been considered prior to the present work is that the security implications of re-using a password depend on the value of an account/password. (The only exception being an industry advisory [28] with unclear methodology and little explanation.) Re-using a low-value password at another low-value site can often be seen as a rational choice by the user, as creating a unique password for a large number of low-security sites is practically infeasible. What really constitutes a problem is re-using a password from a high-value site (such as a bank) on a low-value site, as the low-value site is often easier to compromise. We will study this form of re-use in the remainder of this section.

Source	#accts	#pwds	$\frac{\#\text{accts}}{\#\text{pwds}}$	re-use rate (RR)
Previous work				
Florencio/Herley [9]	25	6.5	3.9	12%
Gaw/Felten [11]	–	–	2.3–3.2	–
Komanduri [17]	–	–	–	27% to 52%
Dhamija/Perrig [8]	10–50	1–7	–	–
Brown et al. [4]	8.18	4.45	1.84	12%
Trusteer Inc. [28]	–	–	–	(73%)[2]
Our work				
RR_0^{all}	–	–	–	14%
$RR_{0.2}^{\text{all}}$	–	–	–	19%
RR_0^{fin}	–	–	–	21%
$RR_{0.2}^{\text{fin}}$	–	–	–	26%

Table 3. Comparing our results on password re-use with previous work. (A dash means that the values are not given/cannot be computed from the data.)

4.1 Measuring re-use from random samples

Previous work on password re-use often gives results as *average number of passwords per user* and *average number of accounts per password*. This is less than ideal, as it does not differentiate between the case where each cluster has the same size, or where the size of clusters is heavily skewed, which can make a big difference in practice. In addition, to make such a statement one needs complete knowledge of the user's accounts and passwords. This is problematic because depending on how much you press a user he will remember a different number of accounts he has, making the measure rather fragile, and when working with randomly sampled data there is no way to compare the results.

Therefore, we introduce a new measure for password re-use that we call *re-use rate*. The re-use rate gives the following probability: Choosing a user at random, and choosing two of his accounts at random, what is the probability that the two passwords for the two accounts are identical? As one would expect, a re-use rate of 0 means that no passwords are re-used, and a re-use rate of 1 means that for each user, all passwords are identical. Note that this measure can handle very well the situation when one has access to a subset of a user's passwords, provided that this sample is randomly chosen: Choosing a random password from *all* passwords or from a randomly sampled subset does not make a difference. Hence the re-use rate is a suitable measure for our dataset, where only a random sample of passwords for each user is available.

We are not only interested in exact re-use of passwords, but also in re-use of similar passwords. In practice, tools like JtR implement established concepts

[2] This number is not directly comparable to the other numbers, as they only measured in *any* other password matched, which yields (much) higher percentages than the re-use rate.

like *(normalized) edit distance*. The edit distance of two strings s_1 and s_2 is the minimal number of weighted edit operations required to transform s_1 to s_2. Typical edit operations are *delete/insert/substitute character* (weight 1); we add *prepend/append character* (weight 0.5) to approximate JtR's mangling rules. We normalize the resulting value by dividing by the length of the longer string.

To compare our results with previous work, we convert the numbers from previous work to re-use rate. Here, we have to make assumptions on the sizes of the clusters, which we assume to be of the same size. Writing A for the number of accounts and B for the number of accounts per password, the probability that we get the same password is $\mathrm{RR} = \frac{B-1}{A-1}$. The results are shown in Table 3.

4.2 Results

- First, we measured *re-use across all passwords* of a user, regardless their assignment to MW-Fin or MW-Btm. These measurements allow us to compare the results with previous work.
- Second, we measured *re-use of financial passwords on other sites*, i.e., the re-use rate when, for a fixed user, selecting one password randomly from MW-Fin and one from MW-Btm. Such results have never been obtained before and are enabled by the specifics of our dataset.

For both scenarios, we considered both exact re-use as well as approximate re-use (such as "password" and "password1" for instance). For exact re-use across all passwords we got 14%, for a (normalized) edit distance of 0.2 we have 19%, and for re-use of financial passwords we got 21% and 26% percent, respectively. The results are summarized in Table 3, which also gives figures from previous work for comparison. The detailed graphs are given in Figure 3, where we plot the normalized edit distance on the x-axis, and the fraction of password pairs with normalized edit distance up to that bound on the y-axis.

4.3 Discussion

We can see that the re-use rates only increase slightly between the distance 0 and 50%, which is already larger than what is usually considered "similar". For example, the strings "*password*" and "*password-123*" have edit distance 20% while the strings "*use*" and "*re-use*" have edit distance 50%. This means that among people that re-use their password, most re-use it in the exactly same form. (Re-using with even small modifications would be a much wiser choice than exact re-use, as this would already prohibit simple forms of attack.)

Surprisingly, we find that re-use is more common for financial passwords than for all passwords, 21% vs. 14% for exact re-use and 26% vs. 19% for approximate re-use. We speculate that financial passwords are re-used more frequently because their increased strength represents a cognitive burden on the user, and this is something of a maladaptive coping strategy.

When we compare these results with the work of Florencio and Herley [9], we see that our results are very similar; because they determined a re-use rate

of 12% compared with our 14%, we feel confident that these results are correct. Comparison with the study by Trusteer Inc. [28] is not easy, as they do not describe their methodology. They state that "73% of users share the online banking password with at least one nonfinancial site." How this relates to our results depends on the number of accounts they observed per user, and it is not clear how they handle the case where one user has multiple banking passwords.

5 Conclusion

In this work we studied two important aspects of password security that have received little attention previously. We used a dataset obtained by malware, which has passwords for multiple accounts for most users. This allowed us to compute meaningful statistics on two aspects of password security: first if users choose stronger passwords for accounts that are more valuable, and second on the re-use of passwords from high-value accounts on low-value accounts.

We found that password strength indeed *does* correlate with account value, a result we also were able to confirm with other lists of leaked passwords. This means that high-value real-life passwords are stronger than widely suspected, even though more work is required to see if they are actually strong enough. We were also able to show that users *do* re-use their high-value password on low-value accounts, a practice which must be considered unsafe, and we were able to confirm previous results on password re-use.

Our work also hints at further interesting research topics. First, it is interesting to find other meaningful sources for passwords that have multiple passwords for the same user, that are either larger or have a different/less bias than our present dataset. Evaluating these datasets would further increase the trust and the understanding of our results. Second, understanding the exact motives that lead to the observable differences both in password strength and password re-use is important. A reasonable method seems to be user interviews, which also might inform efforts to influence users towards better behavior, i.e., choosing strong passwords for those accounts that have high value, and to re-use only those passwords that have low value or are sufficiently protected on the server.

References

1. Joseph Bonneau. Measuring password re-use empirically, February 2011. `http://www.lightbluetouchpaper.org/2011/02/09/measuring-password-re-use-empirically/`.
2. Joseph Bonneau. *Guessing human-chosen secrets*. PhD thesis, University of Cambridge, May 2012.
3. Joseph Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *2012 IEEE Symposium on Security and Privacy*, 2012.
4. A. S. Brown, E. Bracken, S. Zoccoli, and K. Douglas. Generating and remembering passwords. *Applied Cognitive Psychology*, 18(6):641–651, 2004.
5. Christian Cachin. *Entropy Measures and Unconditional Security in Cryptography*. PhD thesis, ETH Zürich, 1997.

6. Claude Castelluccia, Markus Dürmuth, and Daniele Perito. Adaptive password-strength meters from Markov models. In *Proc. Network and Distributed Systems Security Symposium (NDSS)*. The Internet Society, 2012.
7. Solar Designer. John the ripper. Online at `www.openwall.com/john`.
8. R. Dhamija and A. Perrig. Deja vu: A user study using images for authentication. In *Proc. 9th USENIX Security Symposium*, 2000.
9. D. Florencio and C. Herley. A large-scale study of web password habits. In *Proc. 16th international conference on World Wide Web (WWW 07)*, pages 657–666. ACM, 2007.
10. D. Florencio and C. Herley. Where do security policies come from? In *Symposium on Usable Privacy and Security (SOUPS)*, 2010.
11. Shirley Gaw and Edward W. Felten. Password management strategies for online accounts. In *Proc. Symposium On Usable Privacy and Security (SOUPS)*, 2006.
12. S. M. Taiabul Haque, Matthew Wright, and Shannon Scielzo. A study of user password strategy for multiple accounts. In *Proc. 3rd ACM Conference on Data and Application Security and Privacy (CODASPY)*, pages 173–176, 2013.
13. HashCat. Online at `hashcat.net/hashcat`.
14. C. Herley, P. van Oorschot, and A. S. Patrick. Passwords: If we're so smart, why are we still using them? In *Proc. 13th International Conference on Financial Cryptography and Data Security (FC 2009)*, 2009.
15. Blake Ives, Kenneth R. Walsh, and Helmut Schneider. The domino effect of password reuse. *Communications of the ACM*, 47(4):75, April 2004.
16. P.G. Kelley, S. Komanduri, M.L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L.F. Cranor, and J. Lopez. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *2012 IEEE Symposium on Security and Privacy*, 2012.
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. Conference on Human Factors in Computing Systems (CHI 2011)*, 2011.
18. Brian Krebs. Fraud Bazaar Carders.cc Hacked, May 2010. `http://krebsonsecurity.com/2010/05/fraud-bazaar-carders-cc-hacked/`.
19. J.L. Massey. Guessing and entropy. In *IEEE International Symposium on Information Theory*, page 204, 1994.
20. Jason Mick. Inside the Mega-Hack of Bitcoin: the Full Story, June 2011. `http://www.dailytech.com/Inside+the+MegaHack+of+Bitcoin+the+Full+Story/article21942.htm`.
21. Robert Morris and Ken Thompson. Password security: a case history. *Commun. ACM*, 22(11):594–597, 1979.
22. Arvind Narayanan and Vitaly Shmatikov. Fast dictionary attacks on passwords using time-space tradeoff. In *Proc. 12th ACM conference on Computer and communications security (CCS)*, pages 364–372. ACM, 2005.
23. Jason RC Nurse, Sadie Creese, Michael Goldsmith, and Koen Lamberts. Trustworthy and effective communication of cybersecurity risks: A review. In *Proc. Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 60–68. IEEE, 2011.
24. Shannon Riley. Password security: What users know and what they actually do. *Usability News*, 8(1), 2006.
25. M. A. Sasse, S. Brostoff, and D. Weirich. Transforming the 'weakest link' a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3):122–132, 2001.

26. Stat Owl. Microsoft market dominance. Online at http://www.statowl.com/custom_microsoft_dominance.php, 2013.
27. Trusteer, Inc. Detects rapid spread of new polymorphic version of zeus online banking trojan. Security Advisory, online at http://www.trusteer.com/news/press-release/trusteer-detects-rapid-spread-new-polymorphic-version-zeus-online-banking-trojan, 2010.
28. Trusteer, Inc. Reused login credentials. Security Advisory, online at http://landing2.trusteer.com/sites/default/files/cross-logins-advisory.pdf, 2010.
29. Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proc. 17th ACM conference on Computer and communications security (CCS 2010)*, pages 162–175. ACM, 2010.
30. Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. Password cracking using probabilistic context-free grammars. In *Proc. IEEE Symposium on Security and Privacy*, pages 391–405. IEEE Computer Society, 2009.
31. Yinqian Zhang, Fabian Monrose, and Michael K. Reiter. The security of modern password expiration: an algorithmic framework and empirical analysis. In *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 176–186, 2010.

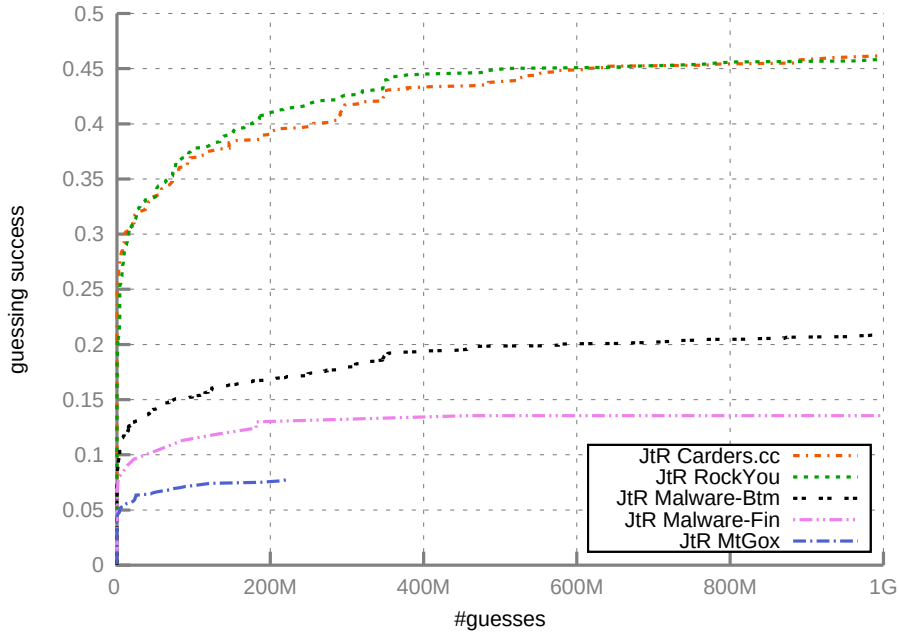

Fig. 1. Fraction of passwords successfully guessed when running JtR against various password lists.

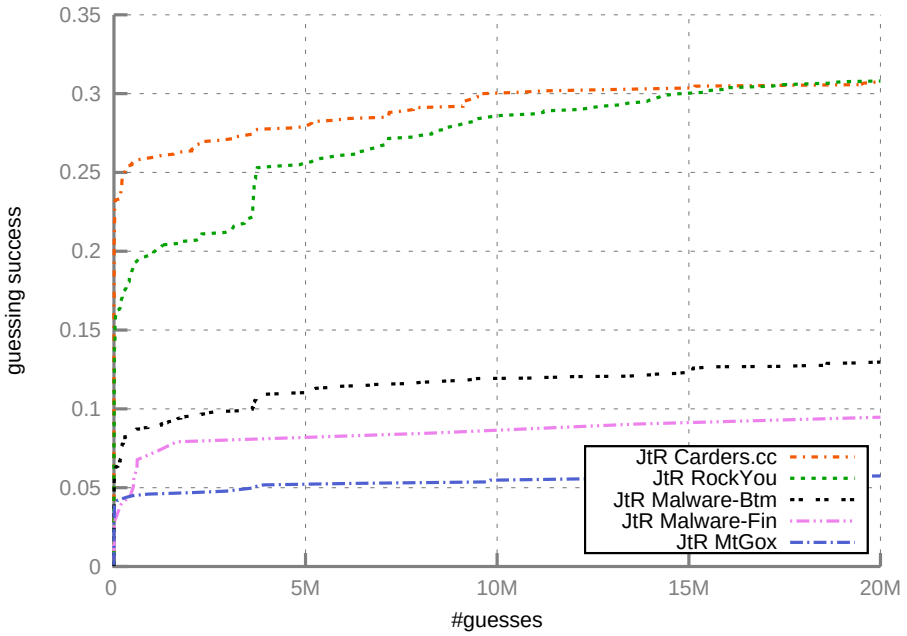

Fig. 2. Fraction of passwords successfully guessed when running JtR against various password lists (zoomed in).

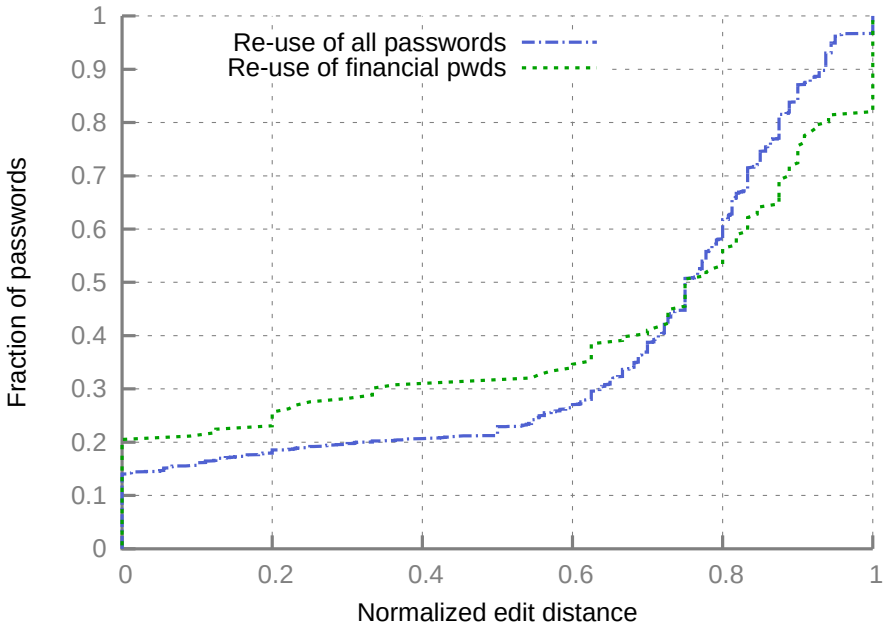

Fig. 3. Measuring the re-use of passwords for variable levels of similarity, given by their edit distance.