Evaluating the Accuracy of Password Strength Meters using Off-The-Shelf Guessing Attacks

David Pereira* and João F. Ferreira†
INESC-ID & Instituto Superior Técnico
University of Lisbon, Lisbon, Portugal
Email: *david.b.pereira@tecnico.ulisboa.pt, †joao@joaoff.com

Alexandra Mendes
University of Beira Interior, Covilhã, Portugal
HASLab, INESC TEC, Porto, Portugal
Email: alexandra@archimendes.com

***Abstract*—In this paper we measure the accuracy of password strength meters (PSMs) using password guessing resistance against off-the-shelf guessing attacks. We consider 13 PSMs, 5 different attack tools, and a random selection of 60,000 passwords extracted from three different datasets of real-world password leaks. Our results show that a significant percentage of passwords classified as strong were cracked, thus suggesting that current password strength estimation methods can be improved.**

***Index Terms*—Password Security, Password-based Authentication, Password Strength Meters, Password Datasets**

I. Introduction

Passwords remain the primary authentication method in today's digital world and will likely prevail in the foreseeable future as a viable, practical and cost-effective method for user authentication [1], [2]. However, weak password selection behaviors [3], [4] combined with the re-utilization of credentials across different services [5], make guessing attacks a serious threat against the integrity of user accounts [6], [7].

Password strength meters (PSMs) are a popular password security mechanism that helps users choose stronger passwords. They rely on the idea of checking password strength through estimation [8], [9], while offering feedback to users. Different PSMs were proposed over the years and many of them have been accepted and widely adopted [10], [11].

Building an accurate PSM is one of the main challenges towards guiding users into better password selection. Depending on the metrics used for measuring password strength as well as how the passwords' estimation strength is computed, meters might misjudge, by over or underestimating, the true strength of passwords, thus failing to capture the passwords' guessing resistance [2], [6], [12]. This means that users trusting in inaccurate meters may actually be misguided into worse password selection. Previous research focused exclusively on evaluating PSMs is scarce [10], [11] but adds crucial information about how to develop better mechanisms and feedback guidance towards better password selection. However, existing work does not relate the output of PSMs with password guessing resistance to off-the-shelf guessing attacks.

In this paper, we study password guessing resistance against off-the-shelf guessing attacks as an accuracy measure of PSMs. We consider 13 PSMs, 5 different attack tools, and a random selection of 60,000 passwords extracted from three different datasets of real-world password leaks (RockYou [13], LinkedIn [14], and 000WebHost [15]). We compare how PSMs rate passwords and analyze the relation between the classifications produced by the PSMs and the passwords guessing resistance to guessing attacks performed using 5 different attack tools. Our findings include: 1) password guessing resistance to off-the-shelf attacks of similarly labelled passwords relate to their password strength estimated by PSMs; 2) a significant percentage of passwords classified as strong were cracked, suggesting that current password strength estimation methods can be improved; 3) the number of cracked passwords classified as medium or below is high, thus suggesting that service providers should only accept passwords classified as strong or very strong.

After presenting the design of the study in Section II, we present in Section III the results obtained. In Section IV, we address the research questions and in Section V we present related work. We conclude the paper in Section VI, where we also discuss future work.

II. Study Design

This section presents the design of our study, including the research questions, the selection of the PSMs and datasets of passwords, and the data collection and analysis methodology.

A. Research Questions

We aim to answer the following research questions:

RQ1: How do PSMs compare regarding strength estimation?

RQ2: Does password guessing resistance to off-the-shelf attacks of similarly labelled passwords relate to their password strength estimated by PSMs?

RQ3: Is it possible to extract new insights from the obtained results in order to build password security mechanisms with better guessing resistance and accuracy?

B. Password Strength Meters

We focus on PSMs that are used by popular web services and easily queryable, i.e. where the setup process together with password feeding and output scraping can be automated. Most of the PSMs considered in this study are from popular websites appearing in the top 100 ranking published by RankRanger[1] according to user online traffic in 2019. In addition, we include

[1]RankRanger Top 100 Websites: https://www.rankranger.com/top-websites

the *Have I Been Pwned?* service[2], which collects database dumps with information about billions of leaked accounts and their respective passwords.

The PSMs that we selected for this study are shown below. Where applicable, we indicate how many bins each PSM uses.

- **zxcvbn** (5 bins) Popular academic PSM created by Daniel Wheeler [9]. We used the Python implementation.[3]
- **haveibeenpwned** This web service returns the frequency of a particular passwords' hash in the available leaked datasets.
- ***From popular websites:***
  - 3 bins: **airbnb**, airbnb.com; **bestbuy**, bestbuy.com; **thehomedepot**, homedepot.com
  - 4 bins: **dropbox**, dropbox.com; **target**, target.com; **facebook**, facebook.com; **microsoftV3**, bit.ly/39LCXT6
  - 5 bins: **cryptowallet**, blockchain.com; **reddit**, reddit.com; **slack**, slack.com; **twitter**, twitter.com

C. Password Datasets

The datasets of leaked passwords that we consider in this study are the following: 1) **RockYou**, compromised in plaintext from the *RockYou* online gaming service of the same name around the year 2009 [13]. The version we obtained contained 32,603,048 passwords. 2) **000webhost**, compromised from a free web space provider for PHP and MySQL applications. The data breach became public in October 2015. The version we obtained contained 15,271,208 passwords. 3) **LinkedIn**, compromised from the professional social networking site *LinkedIn* around the year 2012 [14]. Unsalted password hashes in SHA-1 format were compromised and $\approx 98\%$ of these have subsequently been cracked. These cracked passwords make up the LinkedIn dataset we use in this work. The version we obtained contained 172,428,238 passwords.

Data cleansing and filtering: As recommended in this type of studies [16], each dataset was first filtered according to the password composition policy it is known to have been created under [17]. Passwords containing non-ASCII characters were then removed to avoid encoding issues that might arise due to multi-byte characters being stored as multiple characters, artificially inflating password length. Finally, as shown by Bonneau [18], approximating strength for unlikely passwords is error-prone. As such, each dataset was filtered once again by taking into account its own password frequency distribution, thus resulting in two separate datasets: one with *relaxed conditions* (without taking into account password frequency) and one with *unrelaxed conditions* (that only includes passwords whose frequency is at least 10). After filtering, RockYou, LinkedIn and 000webhost unrelaxed and relaxed datasets ended up with 43.4% and 99.7%, 36.8% and 91.3%, 12.9% and 99.8% passwords of their original dataset, respectively.

D. Attack Tools

To study password guessing resistance we selected two different conceptual approaches widely popular in the password-cracking community and in the academic literature. We locally ran two heuristic cracking tools: JohnTheRipper (JtR, v1.8.0.9-jumbo) and Hashcat (v5.1.0). We also used three probabilistic cracking tools (Probabilistic Context-Free Grammar, Markov Model and Neural Network-based) from the Password Guessability Service (PGS) by CMU.[4] While JtR and Hashcat include wordlists and rule lists samples, they are far smaller than those used in typical attacks and far more ineffective [7]. Therefore we adopted an advanced configuration, by making use of wordlists far more extensive[5] (with near 304,000 and 1,600,000 common password entries and natural-language dictionaries). Moreover, we combined the stock (151), SpiderLabs (5,146) and Megatron (15,329) mangling rules for JtR and the Best64 (77), T0XlC (4,085) and Generated2 (65,117) mangling rules for Hashcat. The PGS probabilistic cracking tools were trained as detailed in the PGS website and we used their recommended configurations.

E. Data Collection and Analysis

In order to answer the proposed research questions, we performed the following experiment: 1) We randomly sampled 10,000 passwords from each previously filtered RockYou, LinkedIn and 000WebHost publicly leaked datasets (with both relaxed and unrelaxed frequency conditions), having a grand total of 60,000 random passwords for this experiment; 2) We then queried the 13 password strength meters considered in this study with those 60,000 passwords. Each meter produced its own password classification distribution (for each dataset sample) according to its own quantization scale; 3) We then set out to attack those passwords by using the attack tools described above; 4) Finally, by relating password guessing resistance with their respective meters' strength classifications, we analyzed each meters' password distribution in light of cracked and uncracked passwords per bin.

Experimental setup: [6] All the experiments were performed in a MacBook Pro laptop, running macOS Catalina (version 10.15.1) with a 2,7 GHz Intel Core i5 dual-core CPU, 8 GB RAM and Intel Iris Graphics 6100 1536 MB GPU.

III. Results

This section presents the results of the experiments carried out in order to analyze the accuracy of PSMs.

A. Password Meter Classification Results

We start with the password classification distributions for each PSM. These are showcased under two different perspectives: first, according to the whole sample of 60,000 random passwords and then according to each dataset sample of 20,000 random passwords. These results are useful because they give us information about the estimation behaviour of each PSM.

Figure 1 shows the password classification distributions for each PSM on the whole sample of 60,000 random passwords. Each percentage corresponds to the number of passwords under each meters' classification quantization bin. Each meter

[2]Have I Been Pwned?: https://haveibeenpwned.com

[3]zxcvbn Python module: https://github.com/dwolfhub/zxcvbn-python

[4]PGS service: https://pgs.ece.cmu.edu

[5]https://github.com/berzerk0/Probable-Wordlists

[6]All the code used is available: https://github.com/davidfbpereira/pws_repo

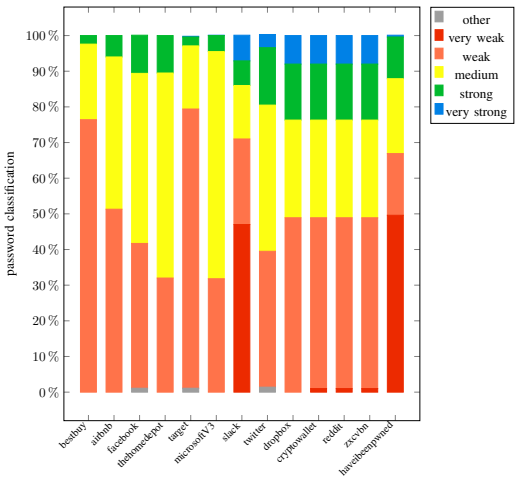

Fig. 1. Password Meter Classification Distributions

has its own quantization scale which is represented by a different colour. The great majority of these bins are represented by a textual or numerical representation, where the lowest bins are commonly called "too short", "weak" or "1 / 4", whereas the highest bins are commonly named as "good", "very strong" or "4 / 4". We clustered the categories "too short", "too long" and "cannot contain ~ or spaces" (from the PSMs **twitter**, **facebook** and **target**) into one single bin dubbed "other". We made this decision because there are few passwords with these assigned classifications and because it simplifies data analysis. Moreover, since **haveibeenpwned** outputs the number of occurrences for each password, we decided to map that number to one of 5 bins: a password not found was deemed "very strong", 1 occurrence was deemed "strong", 2 to 5 occurrences deemed "medium", 6 to 50 occurrences deemed "weak" and over 50 deemed "very weak". As can be seen in Figure 1, the PSMs that have the fewest number of bins are **bestbuy**, **airbnb**, and **thehomedepot** (with 3 bins). The PSMs **facebook**, **target**, **microsoftV3** and **dropbox** have 4 bins, whereas all the others have 5 bins. Before we created the bin "other", **target** was the PSM with more bins (7).

The PSMs **dropbox**, **cryptowallet** and **reddit** produce almost the same password classification distribution results as **zxcvbn**(**dropbox** seems to combine the two weaker bins, but maintains the remaining ones intact). This suggests that these service meters make use of the **zxcvbn** meter internally.

Finally, we can observe four distinct meter groups, namely: conservative meters in both the lower and higher bins (**bestbuy** and **target** services); less conservative meters in the lower bins but not the higher ones (**airbnb**, **facebook**, **thehomedepot** and **microsoftV3**); conservative meters in the lower bins but not the higher ones (**slack**); and less conservative meters in both lower and higher bins (**twitter**, **zxcvbn** and its derivatives). The quantization scale used for **haveibeenpwned** shows that nearly half of the randomly sampled passwords appears at least 51 times in their database, while the other half appears less that 50 times. Almost 12% were found only once and less than 0.5% were not found in their password database.

Figure 2 shows the password classification distribution divided into the RockYou, LinkedIn and 000WebHost dataset samples of 20,000 random passwords. In particular, it illustrates the relative classification differences between each individual datasets. In general, the passwords from the 000WebHost dataset sample were classified as being stronger than the ones from the LinkedIn and RockYou dataset samples. The LinkedIn dataset sample also had slightly better ratings when compared to the RockYou dataset sample.

B. Password Guessing Attack Results

The overall cracking results are depicted in Figure 3, where the total percentage of the number of cracked passwords is plotted as a function of the number of attempted guesses tried by each tool. This figure shows that the Probabilistic Context-Free Grammar (PCFG) tool cracked the largest number of passwords (almost 90%), while the other tools success rate ranged from 60% to 70% cracked passwords. Moreover, the PCFG and Markov Model tools needed fewer attempted guesses to reach a higher success rate; the neural network-based probabilistic tool took many orders of magnitude higher in terms of number of guesses, but still had a lower success rate than the former tools. Table I shows the number of passwords cracked under each dataset sample for this current experiment. As expected, the number of cracked passwords under the unrelaxed conditions was higher, with the exception being for the neural network-based tool. A possible reason for this might be because these passwords are more frequent in leaked datasets and are, therefore, used as low-hanging fruits in the wordlists and training data of password guessing tools.

C. Password Classification and Guessing Results Combined

Finally, we relate the PSMs classifications with the percentage of passwords cracked. Due to space limitations, we focus on the best performing tool of each cracking approach: JtR and PCFG. Figure 4 shows the percentage of cracked (dotted bars) and uncracked (clear bars) passwords relative to each PSM classification. When considering the JtR password guessing results, most passwords classified in the lowest bins were cracked. Moreover, a little more than half and a very small part of the passwords classified in the middle and top bins were cracked, respectively. A similar pattern is observed when considering the PCFG tool (Figure 4, right), but the number of cracked passwords is considerably higher. This confirms the expectation that passwords classified in the lower bins (very weak, weak, and medium) are indeed easy to guess, whereas passwords classified in the stronger bins (strong and very strong) are harder to guess. This suggests that services should only accept passwords classified as strong and very strong. Nevertheless, both JtR and the PCFG tools cracked passwords in the stronger bins, suggesting that all meters can (and should) improve their password estimation methods.

When considering each dataset with respect to both cracking tools (Figures 5 and 6), we observe that the passwords from the 000WebHost dataset sample were harder to crack than the ones from the LinkedIn and RockYou. The LinkedIn

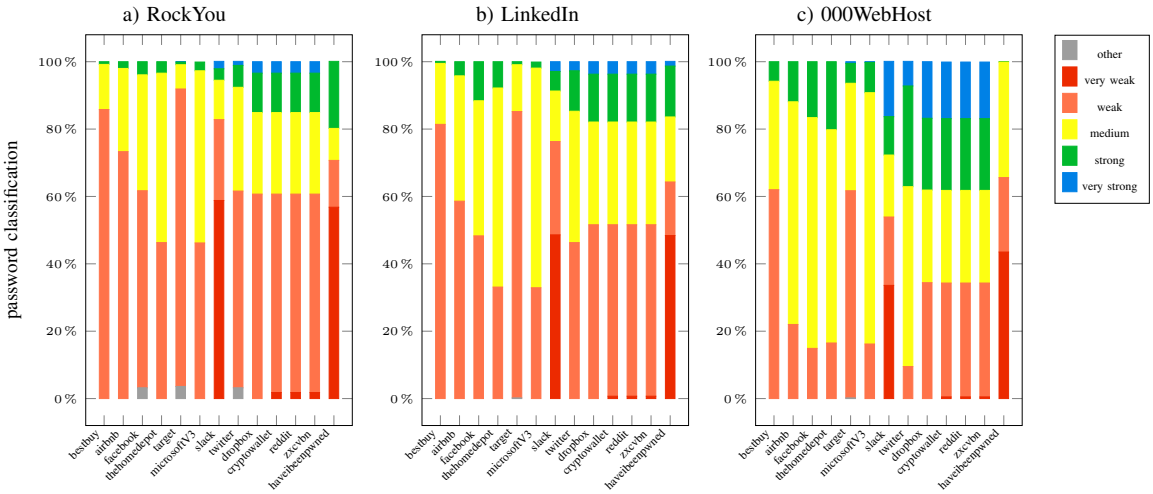

Fig. 2. Password Meter Classification Distributions of All Datasets

TABLE I
PASSWORD GUESSING RESISTANCE BY DATASET SAMPLE

	relaxed conditions			*unrelaxed conditions*			
Dataset Sample	**RockYou**	**LinkedIn**	**000WebHost**	**RockYou**	**LinkedIn**	**000WebHost**	**Total (out of 60k passwords)**
JohnTheRipper	5.5k	4.8k	2.4k	9.6k	9.2k	7.4k	38.9k (65%)
Hashcat	4.7k	3.7k	2.1k	9.6k	9.1k	7.2k	36.4k (61%)
Markov Model	9.9k	3.7k	1.9k	10k	9.5k	7k	42k (70%)
PCFG	9.7k	8.4k	6.3k	10k	9.8k	9.3k	53.5k (89%)
Neural Network	6.7k	7.7k	8.9k	4.1k	5.8k	7.8k	41k (68%)

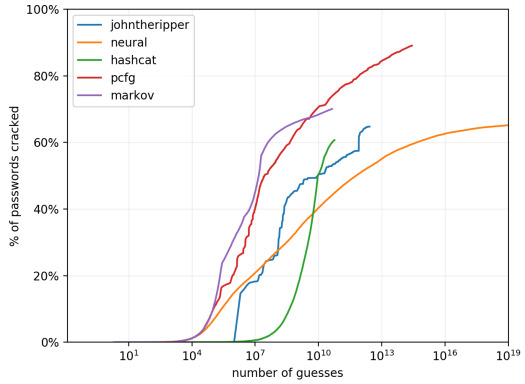

Fig. 3. Password Guessing Resistance Results

password dataset sample was also slightly harder to crack when compared to the RockYou dataset sample.

IV. DISCUSSION

This section addresses the proposed research questions.

RQ1: *How do PSMs compare regarding strength estimation?*

Our results show that some PSMs are considerably more conservative than others (Figure 1). We found that the most conservative are **bestbuy** and **target**, with more than 96% of passwords classified with a maximum strength of medium and around 80% classified as weak. The less conservative **twitter** and **zxcvbn**, with around 20% of passwords classified as strong or very strong.

When considering the dataset samples individually (Figure 2), all PSMs, except **haveibeenpwned**, consider the 000WebHost passwords stronger than those in the other two datasets. Moreover, the LinkedIn password samples were classified as being stronger when compared to RockYou. This is likely due to the use of more stringent password composition policies under which the passwords contained in 000WebHost (lowercase and digits required and length$\geq$6) and LinkedIn (length$\geq$6) were created [10], [17]. The fact that **haveibeenpwned** does not consider the 000WebHost passwords stronger than those in the other two datasets suggests that passwords from 000WebHost are more frequent in the service. In fact, in our sample there were no passwords considered very strong by **haveibeenpwned** and only 0.1% were considered strong.

RQ2*: Does password guessing resistance to off-the-shelf attacks of similarly labelled passwords relate to their password strength estimated by PSMs?*

Overall, we observe that passwords classified in the lower bins are more easily cracked than passwords classified in the upper bins (Figure 4). This suggests that password guessing resistance to off-the-shelf attacks of similarly labelled passwords relate to their password strength estimated by PSMs. Looking at Figure 4, **haveibeenpwned** appears to be an exception with the ratio of cracked/uncracked passwords being greater in the strong bin than on the medium bin. However, this

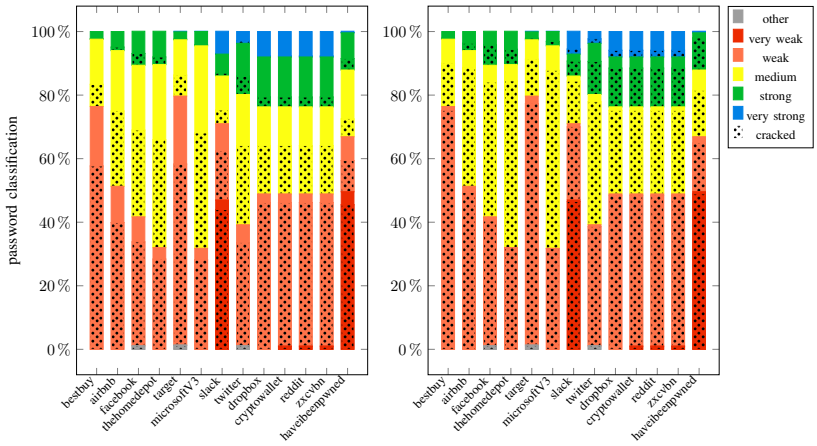

Fig. 4. Guessed Password Meter Classification Distributions with JtR (left) and PCFG (right) tools

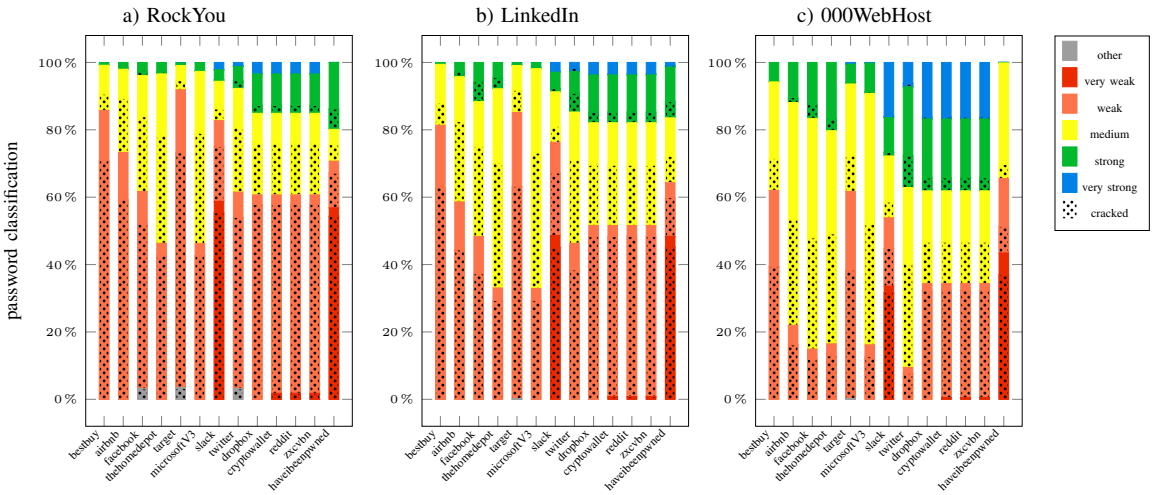

Fig. 5. Password Meter Classification Distributions of All Datasets According to JtR

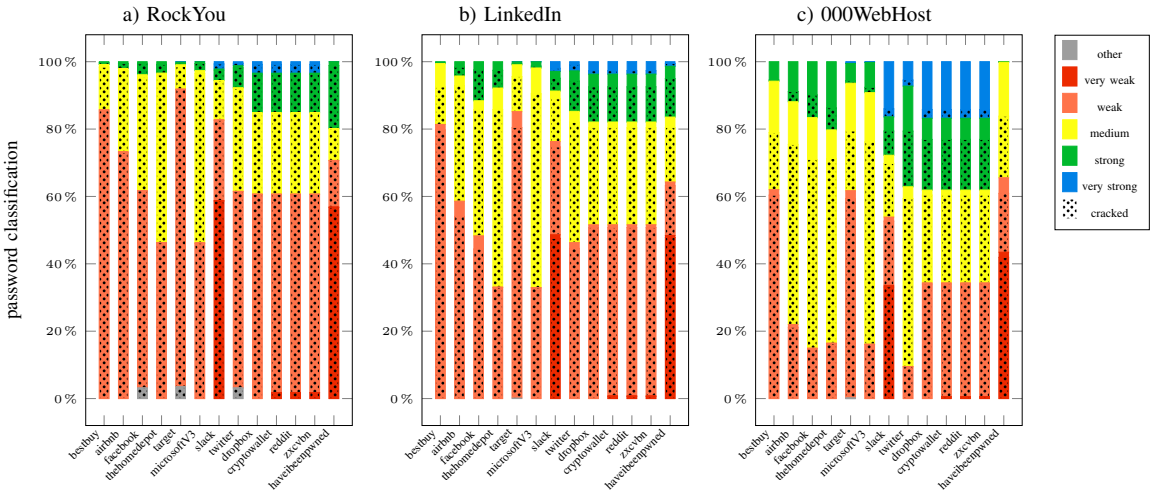

Fig. 6. Password Meter Classification Distributions of All Datasets According to PCFG

is justified by the fact that **haveibeenpwned** only considers 0.1% of 000WebHost passwords as strong and a substantial number of passwords classified as medium were not cracked, likely due to the more stringent password composition policy (Figures 5 and 6). This also suggests that combining samples from different password datasets that use different password composition policies might lead to spurious results. It is thus important to undertake separate analysis on each sample.

Finally, the results show that only a small percentage of passwords classified as strong/very strong by **zxcvbn** have been cracked. This suggests that **zxcvbn** might be the best PSM in terms of accuracy and security. Nevertheless, a significant percentage of passwords classified as strong were cracked, suggesting that password strength estimation can be improved.

RQ3: *Is it possible to extract new insights from the obtained results in order to build password security mechanisms with better guessing resistance and accuracy?*

Based on the results obtained, we highlight the following points: 1) We advise that service providers should only accept passwords classified as strong or very strong, since the number of cracked passwords classified as medium or below is high; 2) The number of 000WebHost passwords cracked is much smaller than in other datasets (Figure 6c), suggesting that the use of more stringent password composition policies is advised; 3) Although our samples are built from leaked password datasets, some of the passwords contained in these do not appear as leaked in the service *Have I Been Pwned?*. Examples include the passwords `westsidetavern` (LinkedIn), `ozzyismybabe&ilovehimtodeath` (RockYou), and `asdfghjklqwertyuiopzxcvbnm1234` (000WebHost). Services that use *Have I Been Pwned?* in their password security checks should take this into consideration.

V. Related Work

Research focused exclusively on evaluating PSMs is scarce. de Carné de Carnavalet and Mannan [11] analyzed 11 PSMs deployed in popular websites by measuring the strength labels assigned to common passwords from several password dictionaries. They found evidence that the commonly used meters are highly inconsistent and fail to provide coherent feedback. Recently, Golla and Gürmuth [10] formulated a methodology for measuring the accuracy of a PSM. However, unlike the study presented here, none of these two approaches attempt to relate the output of PSMs with password guessing resistance to easily available, off-the-shelf guessing attacks.

The problem of maximizing password guessing resistance has been extensively researched [5], [6], [7]. A greater emphasis on studying password strength, on how to define and quantify it, has been carried out progressively [2], [6], [12], [10]. Moreover, new approaches with the aim of assisting and protecting users against modern password guessing attacks, through the development of effective security mechanisms [9], [19], have been introduced while trying to maintain the usability of passwords at the same time.

VI. Conclusion

We show that password guessing resistance to off-the-shelf attacks of similarly labelled passwords relate to their password strength estimated by PSMs. As future work, we plan to extend the analysis to include more attacks and password samples that satisfy specific composition policies. Based on password patterns identified during our experiments, we have started the implementation of an extension of zxcvbn that aims to be more accurate. We will investigate whether it can be integrated in the verified Linux PAM modules that we created [20].

Acknowledgment: This work was supported by Fundação para a Ciência e a Tecnologia under project UIDB/50021/2020.

References

[1] C. Herley and P. C. van Oorschot, "A research agenda acknowledging the persistence of passwords," in *Published in IEEE Security and Privacy Magazine, Volume 10 Issue 1, Jan.-Feb.* IEEE, 2012, pp. 28–36.

[2] D. Florêncio, C. Herley, and P. C. van Oorschot, "An administrator's guide to internet password research," in *Proc. LISA*, 2014.

[3] D. Malone and K. Maher, "Investigating the distribution of password choices," in *Proc. WWW*, 2012.

[4] A. Vance, "If your password is 123456, just make it hackme," *The New York Times*, 2010. [Online]. Available: https://nyti.ms/3guN6WH

[5] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman, "Of passwords and people: measuring the effect of password-composition policies," in *Proc. CHI*, 2011.

[6] P. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. López, "Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms," in *Proc. IEEE Symp. Security & Privacy*, 2012.

[7] B. Ur, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, S. Komanduri, D. Kurilova, M. L. Mazurek, W. Melicher, and R. Shay, "Measuring real-world accuracies and biases in modeling password guessability," in *Proc. USENIX Security*, 2015.

[8] M. Bishop and D. V. Klein, "Improving system security via proactive password checking," in *Computers and Security, Volume 14, Issue 3*. IFIP, 1995, pp. 233–249.

[9] D. L. Wheeler, "zxcvbn: Low-budget password strength estimation," in *Proc. USENIX Security*, 2016.

[10] M. Golla and M. Dürmuth, "On the accuracy of password strength meters," in *Proc. CCS*, 2018.

[11] X. de Carné de Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *Proc. NDSS*, 2014.

[12] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. CCS*, 2010.

[13] N. Cubrilovich, "Rockyou hack: From bad to worse," https://tcrn.ch/2PoXZNW, Dec 2009, (Accessed on 02/08/2020).

[14] M. Burgess, "Check if your LinkedIn account was hacked," https://bit.ly/33qwps0, May 2016, (Accessed on 02/08/2020).

[15] T. Brewster, "13 million passwords appear to have leaked from this free web host," *Forbes*, 2015, (Accessed on 02/08/2020). [Online]. Available: https://bit.ly/33saroy

[16] S. Johnson, J. F. Ferreira, A. Mendes, and J. Cordry, "Skeptic: Automatic, justified and privacy-preserving password composition policy selection," in *Proc. AsiaCCS*, 2020.

[17] S. Johnson, J. F. Ferreira, A. Mendes, and J. Cordry, "Lost in disclosure: On the inference of password composition policies," in *Proc. Symposium on Software Reliability Engineering Workshops (ISSREW)*, 2019.

[18] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Security & Privacy*, 2012.

[19] W. Melicher, B. Ur, S. M. Segreti, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *Proc. USENIX Security*, 2016.

[20] J. F. Ferreira, S. A. Johnson, A. Mendes, and P. J. Brooke, "Certified password quality - A case study using Coq and Linux pluggable authentication modules," in *Proc. Integrated Formal Methods*, 2017.