

# Design & Implementation of a Password Strength Meter for Partial Passwords

*Vasileios Gerakaris*



Master of Science  
School of Informatics  
University of Edinburgh

2016



# Abstract

Abstract goes here.

# Acknowledgements

Acknowledgements go here.

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Vasileios Gerakaris*)



# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Thesis contribution . . . . .	2
1.3	Chapter outline . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Passwords . . . . .	3
2.1.1	Partial passwords . . . . .	4
2.2	Attacks on passwords . . . . .	5
2.2.1	Online attacks . . . . .	5
2.2.2	Offline attacks . . . . .	5
2.3	Password strength meters . . . . .	5
2.4	Amazon Mechanical Turk (MTurk) . . . . .	6
<b>3</b>	<b>Design &amp; Implementation</b>	<b>9</b>
3.1	Password strength meter implementations . . . . .	9
3.1.1	Entropy . . . . .	9
3.1.2	Banlists . . . . .	10
3.1.3	zxcvb (by Dropbox) . . . . .	10
3.1.4	Extent of password meter deployment . . . . .	11
3.2	Attacks on partial passwords . . . . .	11
3.2.1	Projection dictionary attack . . . . .	12
3.3	Design choices . . . . .	13
3.3.1	Partial password strength metric . . . . .	13
3.3.2	Result presentation . . . . .	14
3.4	Implementation . . . . .	15
3.4.1	Website . . . . .	18

<b>4</b>	<b>Results</b>	<b>21</b>
4.1	Partial password strength metric correctness . . . . .	22
4.2	Effect on partial password strength . . . . .	23
4.2.1	Methodology . . . . .	23
4.2.2	Results . . . . .	24
4.3	Effect on partial password memorability . . . . .	28
4.3.1	Methodology . . . . .	28
4.3.2	Results . . . . .	29
<b>5</b>	<b>Conclusion</b>	<b>33</b>
5.1	Summary . . . . .	33
5.2	Project Evaluation . . . . .	34
5.3	Limitations . . . . .	34
5.4	Ethical considerations . . . . .	34
5.5	Future work . . . . .	34
5.5.1	Possible improvements on strength meter . . . . .	34
5.5.2	Partial password attack improvements . . . . .	35
5.5.3	Secure partial passwords storage . . . . .	35
<b>Appendix A</b>	<b>Partial password strength meter surveys</b>	<b>37</b>
A.1	Survey about the extent of use of partial password challenges . .	37
A.2	Survey about the useability of the partial password strength meter	39
A.2.1	MTurk HIT . . . . .	39
A.2.2	Questions . . . . .	39
A.3	Return survey about the memorability of partial passwords . . .	42
A.3.1	MTurk HIT . . . . .	42
A.3.2	Questions . . . . .	43
<b>Appendix B</b>	<b>Project Website</b>	<b>45</b>
B.1	Consent Form . . . . .	45
B.2	Home page . . . . .	46
B.3	Introduction . . . . .	46
B.4	Registration page . . . . .	47
B.5	Login page . . . . .	47
B.6	Completion page . . . . .	48







# Chapter 1

## Introduction

This dissertation presents a password strength meter that evaluates passwords' strength against online attacks on partial challenges. We discuss the design choices made behind the implementation and explore its effects on the strength of the selected passwords as well as their memorability.

### 1.1 Motivation

The ubiquity of passwords, as the primary method for user authentication on the internet is undeniable, despite harsh criticism regarding their usability and security [1]. For applications where security is of paramount importance (such as banking services), multiple methods of authentication are commonly used.

*Partial Password* challenges, where users are asked to enter a subset of characters from their selected *memorable information* is such a secondary authentication scheme, commonly used by financial institutions in the UK and, to a lesser degree, in Europe [2]. The topics of cracking user passwords and using password-methods have been thoroughly researched in the past [3, 4], but the research was focused on traditional passwords.

Despite their use in security-critical applications, there is little literature concerning attacks on partial passwords and metrics to evaluate their strength. To our knowledge, there does not exist any implementation of a password strength meter for partial passwords; the attempt to design and build the first of its kind

is presented within this dissertation.

## 1.2 Thesis contribution

The main contributions of this work are the following:

1. Survey of the extent of partial passwords' use as an authentication method.
2. Design & implementation of a novel strength meter for partial passwords.
3. Evaluation of the partial password strength meter's effect on the strength and memorability of the selected passwords.

## 1.3 Chapter outline

In Chapter 2, we present the theoretical background which is considered important for a reader of this work, in order to comprehend the concepts and designs discussed later. Specifically, we present basic concepts about passwords, partial passwords and attacks against them, briefly explain the ideas behind traditional password strength meters and summarise the results of personal research on how extensively strength meters and partial passwords are used in authentication systems.

In Chapter 3, we discuss the attacks on partial passwords in more depth, which guide the reasoning behind the design choices made for the partial password strength metrics and the visual presentation of the strength meter. In the second part of the chapter, we present details of the implementation, as well as the website created to showcase and test it.

In Chapter 4, we describe the methodology used to evaluate the effect of the partial password strength meter in the strength of the selected passwords as well as their memorability. We discuss the results of the surveys and draw conclusions regarding our implementation.

In Chapter 5, we mention the known limitations of this work, planned improvements to the implementation, and list some research questions that were raised during the course of the dissertation, some of which the scientific community may deem interesting and wish to explore further.

# Chapter 2

## Background

### 2.1 Passwords

Passwords are secret strings of characters which have been used as the primary method for both online and offline user authentication in the computing era. They appear in various forms and they are generally classified depending on their allowed character set and length. The term *password* commonly refers to strings of small to average length (6-16 characters), while a *passphrase* is usually a longer string (>20 characters), often a sequence of words. Alphanumeric characters are allowed in passwords and passphrases, and in many cases other ASCII printable characters are included in the available character sets. Passwords created for services often must abide by a *password policy*, a set of requirements enforced by that individual service/organisation, in an effort to increase password strength and security [5, 6], as previous research has shown that in the absence of a password policy, users tend to select generally weaker and guessable passwords [7, 8].

Ideally, passwords must comply with two conflicting requirements: they should be easy to remember and use without writing them down, while at the same time be unique for each service, look random and be hard to guess. This conflict is called the *password problem* and is almost impossible for most users, experts and non-experts alike, who often reuse passwords or write them down to cope with it [9, 10]. Komanduri et al. suggested the use of a password policy that is neither too stringent, but at the same time offers noticeable improvements in

password strength.

### 2.1.1 Partial passwords

*Partial password* was defined by D. Aspinall and M. Just [11]:

A *partial password* is a query of a subset of characters from a full password, posed as a challenge such as “Give me letters 2, 3 and 6 from your password”.

Throughout this dissertation we use a slightly different terminology; the term *partial password challenge* is used to refer to such queries, and we use the term *partial password* to refer to passwords on which *partial password challenges* will be issued. The idea behind partial password challenges is that an adversary that observes a user answering the challenge (e.g. by using a key-logger, a malware, or simply looking over the shoulder of the victim) will not be able to learn the whole password in one go.

#### 2.1.1.1 Extent of partial password deployment

In 2012, Aspinall and Just collected and listed information about the authentication practices of 10 UK banks [2]. In an attempt to refresh this information, we observed the publicly available demos and FAQs of the “big four banks” (a colloquial term, used to refer to the four largest banking groups), along with 3 more banks. All 7 examined financial institutions appeared to employ a partial password challenge as part of their authentication process. Partial passwords use was also encountered in some implementations of the *3-D Secure* protocol for credit/debit card online transactions in Europe, such as the “Verified by Visa” and “MasterCard SecureCode”.

A similar approach, combined with a small-scale user survey on Amazon MTurk, indicated that this authentication scheme is uncommon in the US, as we were unable to detect a single bank that was using it.

## 2.2 Attacks on passwords

Several different methods of attacking passwords exist today (e.g. guessing attacks, dictionary attacks), and they are generally classified into one of the following categories, according to their origin.

### 2.2.1 Online attacks

Online password attacks are performed over the internet, by attackers that attempt to guess a users' passwords. An important characteristic of such attacks is that they are usually rate-limited. Security-aware web services allow only a specific number of attempts of guessing a user's password before restricting or introducing delays for future attempts, as per the National Institute of Standards and Technology's (NIST) digital authentication guidelines [12]. Online attacks can be *targeted*, aiming to get access to a specific user's account, or *trawling*, attacking many different accounts and aiming to crack a portion of them. For instance, a trawling attack with 5% success rate on 1000 bank accounts would manage to break into 50 of them (on average).

### 2.2.2 Offline attacks

Offline attacks on passwords are performed on stolen or leaked database dumps. The main differences between them and online attacks is that there is no limit on the number of attempts for a password; the only limits are the CPU and disk I/O speeds. Brute force searches for matches, cracking dictionaries and rule-based word mutation (mangling) are common techniques in the attempts to crack the passwords contained in the database. *Unbound online attacks*, on web services that do not enforce an attempt limit, can use the aforementioned techniques, but produce results slower, due to the network delay introduced.

## 2.3 Password strength meters

A password strength meter is a Graphical User Interface (GUI) element that is displayed during password creation and offers visual feedback on the strength

of the inputted password. Traditional strength meters have been the focus of previous research, regarding both their correctness and their effectiveness. As there is not yet a strict specification, different password meter implementations use different algorithms to measure a password’s strength or guessability. Following a large-scale analysis of password meters in the wild, Castellucia et al. noted that great inconsistencies can be observed among them and that they often provide “blatantly misleading” strength measurements [13].

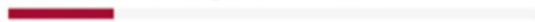
Password strength: Strong



Password strength: Good



Password strength: Too short



(a) Colored bar and text (Google)

<input type="password" value="....."/>		✗ Password is too obvious.
<input type="password" value="....."/>		✓ Password is okay.
<input type="password" value="....."/>		✓ Password is perfect!

(b) Green bar, text and checkmark (Twitter)

Password: 
 Confirm Password:

The structure of your password is too simple to replace the more complex the password, otherwise unable to register successfully.  
 Password length of 6 to 14, the letters are case-sensitive. [Password is too simple hazards](#)

(c) Text-only (Baidu)

Figure 2.1: Strength meter examples, taken from [14]

## 2.4 Amazon Mechanical Turk (MTurk)

Amazon MTurk is a crowdsourcing platform that allows individuals to coordinate and solicit contributions from large groups of people for a wide range of human intelligence tasks (HITs). The platform is heavily used by academics for large-scale research that involves human subjects and it has been found to be a good source of high-quality data, while also achieving a better diversity in population demographics compared to an on-campus study [15, 16]. Ipeirotis’ analysis of MTurk demographics in 2010 indicated that workers are mainly from



US and India, and, on average, younger and more technically adept than the general population[\[17\]](#).



# Chapter 3

## Design & Implementation

### 3.1 Password strength meter implementations

Accurately measuring *password strength* is a difficult task, because the concept of strength has not been clearly defined; we consider the claim by Egelman et al. that “an ideal strength of a password would be an increasing function of the difficulty it presents to modern cracking tools.” [18] to be an accurate metric for password strength.

Before designing the password strength meter for partial passwords, we explored existing implementations (for regular passwords) and drew inspiration from their design. We found that, despite some differences, most password meters employ specific techniques when estimating password guessability, which we list below:

#### 3.1.1 Entropy

*Password Entropy* is a measure of the randomness or unpredictability of a password (in bits). Most proposed algorithms use different methods of calculating entropy, which leads to the aforementioned discrepancies between meters. In practice, most of these ad-hoc implementations, even those used for websites with large volumes of visitors are inadequate [13, 19, 20]; even the entropy estimation scheme proposed by the NIST [21] was found to be unsuitable for measuring the randomness of human-selected passwords [6]. Generally, pass-

word length, character sets used and known patterns are features considered when calculating entropy. Using the various definitions of entropy as a metric looks appealing to strength meter designers because it can offer estimations in real-time and without the need to download massive dictionaries or tables of password probabilities.

### 3.1.2 Banlists

Many password meters define a list of common passwords and/or dictionary words that are derived from real-world leaked password databases <sup>1</sup>. Passwords are compared against the words compared within these lists and matches have their strength score severely reduced, or are outright prohibited. This happens in order to prevent users from selecting very easily guessable passwords, since attackers also have knowledge of those common passwords and are usually the first ones to be tried in an attack.

### 3.1.3 zxcvb (by Dropbox)

*zxcvb* is an open-source, client-side password strength checker developed by Dropbox and offered to the public, encouraging website administrators to use it and developers to try and improve its checking algorithm [20]. Carnavalet and Mannan, after evaluating 11 different strength meters offered by prominent web service providers, concluded that, while *zxcvb* has some significant shortcomings, it is probably the best and most thorough strength meter from the ones tested; they even endorsed it and urged webmasters to adopt or try to extend it instead of creating yet another ad-hoc solution [13].

Its scoring algorithm divides a password into patterns with possible overlaps, calculates the entropy score for each pattern, and generates the final result by summing the values. It also uses a banlist comprised of five different dictionaries of common passwords, English dictionary words, and names/surnames to penalise patterns that match any words in them; penalties are also applied to specific patterns, such as years, dates, character sequences (e.g. ‘dcba’, ‘12345’)

---

<sup>1</sup>Examples of leaked password databases can be found here: [http://thepasswordproject.com/leaked\\_password\\_lists\\_and\\_dictionaries](http://thepasswordproject.com/leaked_password_lists_and_dictionaries)

and spatial keyboard combinations (e.g. ‘qwerty’, ‘zxcvb’).

### 3.1.4 Extent of password meter deployment

In 2012, Ur et al. discovered that 73% of Alexa’s global top-100 most visited sites that allowed user registration displayed a password strength meter as part of the process [14]. In our attempt to examine how extensively password strength meters are currently used, we followed a similar approach: we examined the top-70 US websites based on Alexa’s ranking [22] while filtering out the duplicate results (e.g. all sites owned by Alphabet/Google use the same registration process and meter) and skipping the few that we did not have access to (banks and software for enterprises).

Our results indicate that there has been a serious decline in the deployment of password meters: from the 40 unique sites we examined, we found that password meters were used in only 10 of them (25%). It is also worth noting that we observed some security/privacy critical websites (e.g. Facebook, Amazon, PayPal) not using password strength meters, while some entertainment websites (Reddit, Tumblr) displayed them during the registration process.

## 3.2 Attacks on partial passwords

Adhering to the definition of password strength presented in Section 3.1, we decided to use the results of the best available attacks against partial passwords as an indicator of strength. As mentioned in Section 2.1.1.1, partial passwords are almost exclusively being used as a method of authentication for financial institutions and credit card transactions; it is therefore a rational decision to consider (bounded) online attacks against them. Another reason that further reinforces that decision is that, due to the characteristics of partial password challenges, partial passwords are likely to be stored non-encrypted in the databases, which render offline attacks unnecessary in case of a database breach. We discuss some possible ways to more securely store partial passwords in Section 5.5.3.

While the problem of finding effective and efficient attacks against passwords has been extensively researched in the past, it revolved regular passwords; devising

attacks against partial passwords is a relatively unexplored field. To the best of our knowledge, the only existing piece of literature on attacks against partial passwords is the work of D. Aspinall and M. Just [11], the findings of whom we used when designing the strength metric.

The partial password challenge is a request for  $m$  distinct password character positions out of the length  $n$ , where  $1 \leq m \leq n$ , therefore the number of different possible challenges is  $\binom{n}{m}$ . The allowed character set size ( $N$ ) of the partial password is often restricted in different implementations, being case-insensitive or prohibiting the use of symbols in a password. All the character set sizes we encountered in different implementations are the following: 36 (a-z, 0-9), 52 (a-z, A-Z), 62 (a-z, A-Z, 0-9), 95 (all printable ASCII characters).

### 3.2.1 Projection dictionary attack

The attack that yielded the best results against partial passwords with  $N = 36$ ,  $n = 8$ ,  $m = 3$  and  $\beta = 10$  max guesses, was the projection dictionary attack. It is a guessing attack that bases its predictions on the fact that many words that share the same projections on the challenged character positions. Intuitive examples include prepositions (“con-”, “dis-”, “pro-”, “ove-”, “pre-”) for the first three character positions or the common ending “-ing” for challenges that request the last three characters. Using this method, some responses become significantly more probable than others and attackers can coalesce dictionary entries to generate the best possible responses for each of the  $\binom{n}{m}$  possible challenges.

This attack is further enhanced if the dictionary it parses to generate the predictions is derived from known password distributions rather than simple word dictionaries, since they reflect passwords that are actually used “in the wild”. The leaked RockYou password database, which is very commonly used by attackers and researchers alike [3, 4, 6, 23], contains 32 million password entries and can be processed to generate password-frequency pairs; using those to generate the projection dictionary has been proven to yield better predictions for the partial password challenges. In the work of Aspinall et al., a projection dictionary using the RockYou dataset achieved a 5.5% success rate, an alarmingly high percentage when considering trawling attacks on thousands of accounts.

RockYou was a gaming website with a very lax password policy, and most of the passwords contained in the database were relatively weak, indicating that users were not overly concerned about the strength/security of their passwords. Using a more recent and important dataset for our research, such as the recently leaked password database from the 2012 LinkedIn breach (containing approx. 117 million entries)[24] would likely result in more accurate predictions; that being said, we decided to use the RockYou dataset both for ethical reasons (described in Section 5.4) and to better align our research with previous work.

## 3.3 Design choices

In this section we explain the reasoning behind the main choices we made, when designing the password strength meter for partial passwords.

### 3.3.1 Partial password strength metric

Schechter et al. claim that the popularity of a password is the most accurate predictor for its weakness [25], a statement we agree with and manage to incorporate into the password meter by generating the projection dictionary from an existing password distribution. The algorithm that calculates the strength is quite straightforward: the resilience of a particular partial password challenge is the inverse of its frequency in the projection dictionary. Different challenges yield different strength scores, for instance, the partial password “*pa;sw<rd*” has good scores for the 36 challenges that include positions 3 and/or 6, but abysmal scores for the 20 challenges that include neither of them. In order to get an accurate result, we compare the characters for each of the  $\binom{n}{3}$  possible challenges with the  $K$  best guesses for those positions and then average the scores to get the *avg\_prob*. A password’s resistance against partial attacks is then:

$$ppass\_str = \frac{1}{avg\_prob}$$

We decided against using a banlist, in order to save both space in the file and computational time; since the projection dictionary was generated by data of existing passwords, common passwords would get low scores anyway. Inspired

by *zxcvb*'s implementation, we decided that the partial password meter should penalise some password patterns. Specifically, we deduct from the strength score whenever 3 or more consecutive same characters appear; this prevents the use of a password with a single, uncommon character, e.g. “§§§§§§§§”. In this first iteration of the strength meter, this is the only pattern checked and penalised, future improvements would include more sophisticated pattern detection, such as sequences or spatial combinations, based on the *zxcvb* codebase.

### 3.3.2 Result presentation

After specifying the algorithm that calculates partial password strength, we need to address the way the results will be presented to the users. The decision delves further in the realm of Human-Computer Interaction (HCI) research than computer security, but it is still very important for this project, since we are interested in affecting users' decisions during the password-creation process.

The effect that different strength feedback indicators have on a selected password's strength has been the subject of research, with the most extensive example being the work of Ur et al., a study involving 2931 people assigned to one of 15 different conditions (password meter implementation) [14]. Their results indicate that password meters indeed have an effect in user security and they conclude that

“The combination of a visual indicator and text outperformed either in isolation. However, the visual indicator's appearance did not appear to have a substantial impact.”

The results from the research described in Section 2.3 seem to agree with that conclusion; 7/10 unique meters we encountered employed both a visual indicator (progress bar) and textual feedback. Based on both the theoretical and practical findings, we decided to create a strength meter that incorporates both a progress bar and a text verdict, with passwords getting a *verdict*  $\in \{Weak, Fair, Good, Strong, Very Strong\}$  depending on their final calculated strength and to change the bar's colour depending on that verdict.



## 3.4 Implementation

The final result of our design process was a client-side strength meter calculator written in JavaScript, using features of jQuery to update the elements in the HTML rendering of the page. The decision to follow a client-side architecture was made mainly for security reasons, so parts of the password would not continuously get transmitted back-and-forth between the client and the server. The source code is also made public, therefore concerned users can check their partial passwords locally by opening an offline resource (an HTML file and the accompanying JavaScript file) without the need to launch a web server or connecting to any web page. At the same time, a client-side approach is also good for scalability reasons: since each user's browser handles the computation itself, multiple concurrent connections could be handled without fear of overloading the server. The code that calculates the password strength (based on the algorithm described Section 3.3.1) can be found in Listing 1, while the referenced function `getScoreAfterPenalties()` is shown in Listing 2.

---

```

1  function calculateStr(password) {
2      //[...]
3      // Projection Dictionary attack
4      var ppass_str = 0.001;
5      combos = Util.get_combinations(options.challenges, password.length);
6      for (var i = combos.length - 1; i >= 0; i--) {
7          nums = combos[i].split(' ');
8          ppass = '';
9          for (var pos = 0; pos < options.challenges; pos++) {
10             ppass += password[nums[pos] - 1];
11         }
12         for (var j = guesses['proj_dict'][combos[i]].length - 1; j >= 0; j--) {
13             if (ppass === guesses['proj_dict'][combos[i]][j][0]) {
14                 ppass_str += guesses['proj_dict'][combos[i]][j][1];
15                 break;
16             }
17         }
18     }
19     // Apply penalties (currently only for consecutive same charactes)
20     ppass_str = getScoreAfterPenalties(ppass_str, password);
21     // Average out on all possible challenges
22     ppass_str /= combos.length;
23     // Use the inverse as password strength
24     ppass_str = 1 / ppass_str;
25     // [...]
26 }

```

---

Listing 1: Calculation of password strength

The partial password strength meter was developed in a way that it could cover partial passwords of any size or type, and website administrators could alter specific parameters (such as the password min/max size, the challenge size and

---

```

1 function letterCount(pw){
2     var s = pw.match(/(.)\1{2,}/g) || [];
3     return s.map(function(itm){
4         return [itm.charAt(0), itm.length];
5     });
6 }
7
8 function getScoreAfterPenalties(ppassStr, pw) {
9     // Penalties
10    // Consecutive letters
11    var consecutives = letterCount(pw);
12    for (var i = consecutives.length - 1; i >= 0; i--) {
13        ppassStr += consecutives[i][1] * options.consec_penalty;
14    }
15    return ppassStr;
16 }

```

---

Listing 2: Penalising consecutive same characters

the weight of the penalties imposed) to fit their underlying password policies. In our study, we used passwords with  $n = 6 - 15$  characters,  $N = 95$  (all printable ASCII) and  $m = 3$ . A modified version of the scripts used by Aspinall et al. [11] generated the projection dictionary from the RockYou dataset, which was then stored as a JSON file. There was an important decision to be made in the number  $K$  of best guesses for each of the  $\binom{15}{3} = 455$  challenges to be stored in the dictionary, as a fine balance was needed between the dictionary file size and coverage of passwords checked. We found that  $K = 1000$  was a good value, covering an average of 34.3% of all passwords, and generating a file that was 7.2 MB in size, reduced to 1.57 MB if sent after performing gzip HTTP compression. While the size is not negligible, it is small enough to load within seconds with a modern internet connection speed. The file was sent asynchronously using AJAX to clients during registration (snippet shown in Listing 3), and an earlier attempt was made to send it during the instructions page, so the download would complete while users were reading and it would be cached when required in the registration page, resulting in a better overall user experience. A Python program with a command-line interface (CLI) that implemented the same algorithm was also developed, to allow for fast and efficient offline (and batch) testing of password strengths.

A rather unique feature of the partial password strength meter is that while a user types more characters, the strength score is not guaranteed to increase, unlike most existing password strength meter implementations, where extra characters usually add to the entropy of the password (except when they complete a dictionary word). Due to the way the score is calculated in our algo-

```

1 var crackDict = 'static/js/dicts/rockyou.json';
2 var guesses;
3 // [...]
4 $.getJSON(crackDict, function (data) {
5   guesses = data;
6   if ($("#ppassword").val() !== '')
7     $("#ppassword").change();
8 });

```

Listing 3: AJAX loading of projection dictionary

rithm, entering a character that, combined with the previous characters has high probabilities in the possible challenges, can lead to a decrease of the password strength. These frequent increases and decreases as users enter their passwords proved to be confusing in a small-scale survey performed on a group of students in the School of Informatics of the University of Edinburgh. Adapting to these findings, we altered the visual indicator of the partial password strength meter to fill up to one of 5 discrete levels (down from 50); the final result, including all the possible states is displayed in Figure 3.1. The threshold values for each level were:

- Weak: [0-15)
- Fair: [15-25)
- Good: [25-50)
- Strong: [50-75)
- Very Strong: 75+ <sup>2</sup>

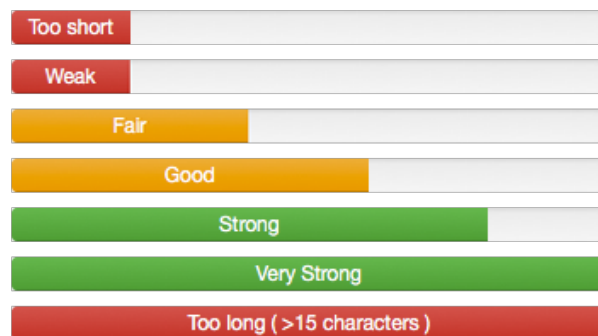


Figure 3.1: Password strength meter result display

<sup>2</sup>Due to the way partial password strength is calculated, values  $> 100$  are possible

### 3.4.1 Website

In order to showcase the password meter and test its effectiveness, we created a website where users could complete a mock registration process. We used Python with Flask [26] as the underlying web framework for the back end of our application and Bootstrap [27] as the front-end (CSS) framework, as a way to improve the UI/UX of the website and be compatible with all kinds of devices. We decided to follow the object-oriented paradigm during development of the application and used SQLAlchemy [28] as an object-relational mapper (ORM) [29] to decouple the model logic with the underlying database schema and implementation. The schema generated from the specified models can be found in Figure 3.2 (the surveys' questions columns were omitted for brevity). It should be mentioned here that the passwords were stored in cleartext (non-encrypted) in the database (for the reasons mentioned in Section 3.2), a fact that the workers were informed about both before and during the task.

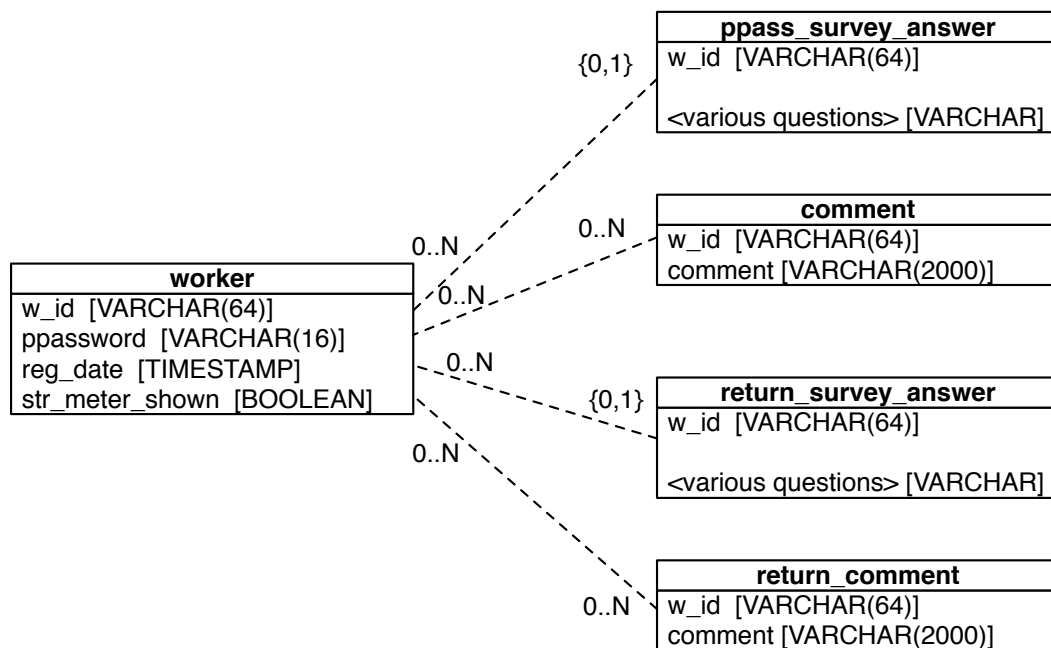


Figure 3.2: Database Schema

The created website used a simple control flow to emulate a registration process, as seen in Figure 3.3. Images of each webpage can be found in Appendix B. On the landing page (B.1), MTurk workers were shown a consent form offering information about the researchers and a security notice, to which they had

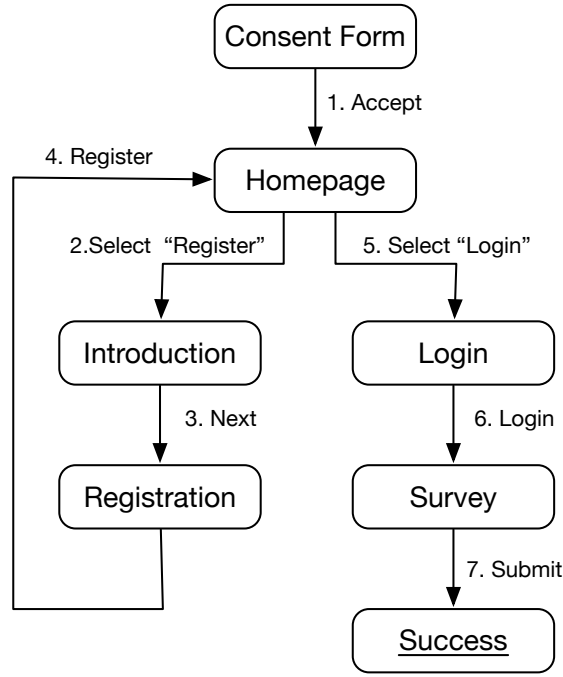


Figure 3.3: Control flow diagram of website

to agree before they could proceed further. On the *homepage* (B.2), they were presented with two buttons, “Register” and “Login”; selecting the former on their first visit would lead them to the *Introduction* page (B.3). This page explained the task and asked participants to imagine that they were creating a password for their banking account, a notice that prior work has proven to lead to stronger password creation [30], and also informed them that they would be asked to return to this website, so they should use their usual methods for remembering and protecting an important password.

On loading the *Instructions* page, the server would randomly assign them to either the control group (no strength meter) or the experiment group (strength meter shown), and try to asynchronously load the projection dictionary for the latter group, in preparation for the next page. The *Registration* page (B.4) differed between the conditions, as shown in Figure B.2 (n.b. the password fields were of type “text” instead of “password”, so users could see their passwords while typing), with the experiment group getting visual feedback on their passwords, as well as the option to get suggestions of strong passwords generated from a dictionary.

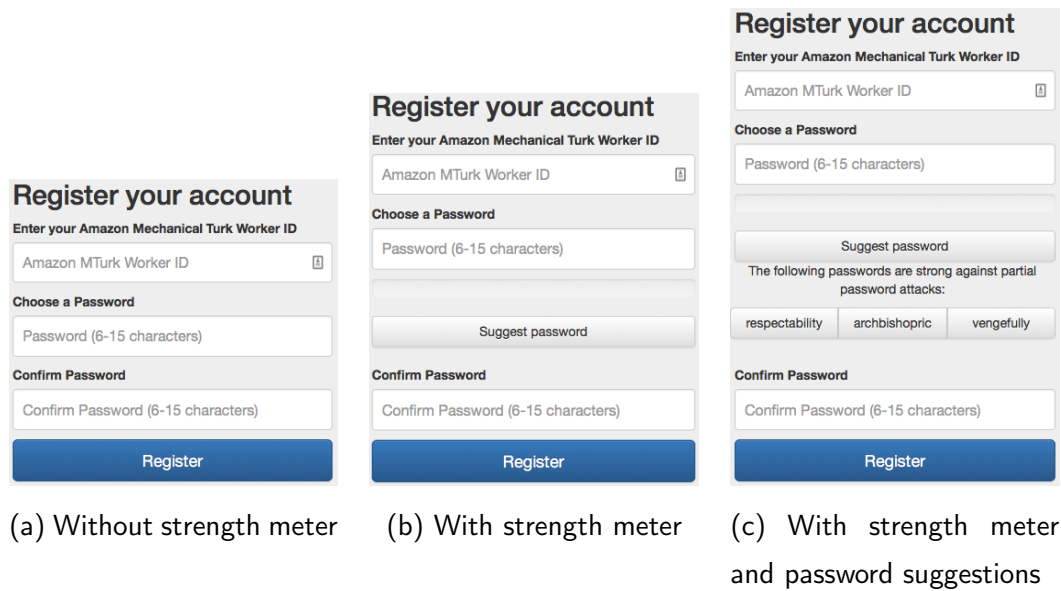


Figure B.2: Registration page (repeated from page 47)

After a successful registration, the workers were redirected to the homepage and selected “Login”, whereupon they were presented with a partial password challenge in the *Login* page (B.5). Following a successful login, the users were asked to complete a survey about their experience and demographics, presented in detail in section ??, and finally received a completion code, so they could submit their HIT on MTurk.

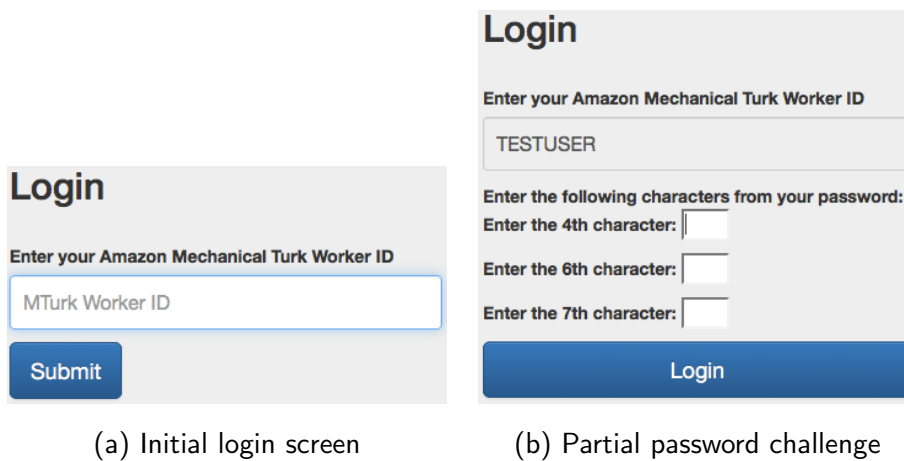


Figure B.3: Login page (repeated from page 47)

The website that we developed was deployed and hosted on the free tier of the OpenShift Online Platform-as-a-Service (PaaS) [31] offered by RedHat, using PostgreSQL [32] as the database management system (DBMS).

# Chapter 4

## Results

To the extent of our knowledge and research, no prior work exists that examines strength meters in a partial password setting. The results of our research are therefore the first ones published for this setting and can serve as a baseline for future research on the subject. At the same time, we can indirectly compare our findings with the ones from research on traditional passwords, and examine if some hypotheses hold true in both settings.

The primary goal of this research was to examine the effect of the partial password meter on the strength of the created partial passwords, and secondly, its effect on their memorability. In order to draw conclusions, we formulated and tested the following null hypotheses in the field:

$H_{0a}$ : Partial passwords are not stronger when the partial password meter is present during creation.

$H_{0b}$ : Partial passwords are not more memorable when the partial password meter is present during creation.

Following the same methodology as Ur et al., we conducted an online study consisting of two parts. Ideally, the study would be conducted with participants that were using partial passwords in their online routine - possibly limiting the demographic to UK residents. An option to gather the necessary data would be to run the experiment in an on-campus laboratory environment, recruiting students and other researchers to participate in the study. We deemed that doing so would restrict us to a very specific demographic, with a distinct age distri-

bution, education level and experience in research studies and could therefore skew the results. In order to align this research as best as possible with previous work, a crowdsourcing platform that uses a mainly UK demographic and on which custom Human Intelligence Tasks can be specified for execution would be an ideal solution. Unfortunately, after examining the available options, none was found that fit the criteria, while also being cost-efficient. Ultimately, we decided recruiting subjects for our study (also referred to as *workers*) from Amazon’s MTurk platform, while providing them with a brief introduction and explanation of partial passwords.

Participants were required to be at least 18 years old and use a web browser with enabled JavaScript capabilities. The details for each task are described in the following sections. While not directly mentioning the purpose of the study to the subjects, we did not employ subterfuge to hide it, unlike what Egelman et al. did in their study [18]. By saying that “We are conducting an academic survey about partial challenges on passwords and we are testing a new partial password system.”, participants were made aware of the general focus of this research. Since we were particularly interested in accurately pinpointing the effect the password meter had in the strength of the created passwords, we decided to use a significance level of  $\alpha = 0.01$  for that statistical test, in order to minimise the probability of a type I error. For all other statistical tests in our analysis, the common  $\alpha = 0.05$  value was used.

## 4.1 Partial password strength metric correctness

Before setting out to test the hypotheses, the correctness of the implemented partial password meter needed to be verified. As mentioned in Section 3.1, the term “password strength” is ill-defined, which renders a theoretical proof of correctness impossible. In accordance with the definition of password strength we adopted, a practical approach was followed when testing the correctness of the developed meter.

A Python script containing the partial password strength calculation algorithm described in Section 3.3.1 combined with a projection dictionary generated by the RockYou dataset was used to evaluate the password strength of different



password lists. The password lists tested were the password database leaks from the RockYou website and the PHPBB forums, as well as the top 10 million passwords dictionary released in 2015 [33], a revision of the top-500 [34] and top-10000 [35] password lists released in 2005 and 2011, respectively. For each set, we filtered out the passwords that did not meet the specified length (6-15 characters) and discovered the most common password that was ranked as “Strong” or “Very Strong” from our algorithm. The Python CLI version of the strength meter was used for faster and more flexible offline evaluation of the aforementioned dictionaries - the algorithm and scores are consistent between the Python and JS implementations.

The findings of the aforementioned analysis are displayed in Table 4.1. We observe that the algorithm is adequately strict in the ranking of partial passwords, even when testing other datasets. It is worth noting, that while these passwords seem and are weak in a traditional setting with offline dictionary attacks, the chance of successfully guessing a partial password challenge of them within  $\beta = 10$  max guesses in an online attack scenario is minuscule. We conclude therefore that the strength metric’s performance is acceptable and can be used to test the selected hypotheses.

Dataset	RockYou		PHPBB		top 10M	
Verdict	Strong	Very Strong	Strong	Very Strong	Strong	Very Strong
# of passwords	31063584		230905		9061038	
Most common	50cent		qazwsx		696969	1qaz2wsx
Frequency	3294		92		3050	2531
% frequency	0.011%		0.040%		0.034%	0.028%

Table 4.1: Most common password ranked “Strong” or “Very Strong” per dataset

## 4.2 Effect on partial password strength

### 4.2.1 Methodology

We recruited 200 participants from Amazon MTurk to participate in this research. For the first part of the study, subjects were paid 75 cents and were

asked to “Answer a survey about partial passwords after completing a mock registration process”. In order to qualify for the task, they had to complete a small screener test that ensured they were able to correctly answer partial password challenges. This qualification test also served to prime workers for the task and inform them about the concept of partial passwords, since most of them were expected to not have experience with partial password login systems, as our prior survey indicated (described in Section 2.1.1.1). Before the task was published, a small-scale (20 subjects, both MTurk workers and university students) pilot was carried out, in order to discover possible bugs or other problems with the website. Since the feedback received from this pilot run motivated changes in the website and the strength meter, we discarded their results to ensure consistency within the study samples. The participants in our study were heavily based in the US (183/200, 91.5%), and had a 60/40 Men/Women ratio. Ages varied from 18-69, having a mean age of 32.77 with 10.07 standard deviation and with 46.5% of the participants being in the 26-35 age range.

Participants were randomly assigned to either the control or the experiment condition and were asked to complete a registration process on our website, followed by a successful login using a partial challenge on their passwords, as shown in the control flow in Figure 3.3. After successfully logging in, they were presented with a survey containing questions about their opinions concerning the strength meter (this group of questions differed between conditions), questions about the partial password and their method of answering the challenge, as well as some basic demographic questions. The multiple-choice survey questions were presented in a equidistant (neutral) manner and designed to offer five-level Likert-type scale responses [36], in order to minimise user bias. An *attention-check* question (a question with a clear, predefined answer) was also inserted in the survey, to reduce the likelihood of a worker randomly or automatically (with a script) completing the questionnaire. The full survey is included in Appendix A.2.

## 4.2.2 Results

Due to the way partial password strength is calculated, scores  $> 100$  are possible, so during the processing of the results we normalised all password strength

scores over that threshold to 100. We categorised the created partial passwords depending on their strength rating, using the same thresholds specified for the strength meter, and counted the occurrences for each on both conditions. This process was used to get a better overview and visualisation of the meter’s effect, the results are presented in Table 4.2 and Figure 4.1.

Verdict	Meter	No Meter
Very Strong	70	57
Strong	8	5
Good	11	10
Fair	7	16
Weak	4	12
Total	100	100

Table 4.2: Verdict frequencies

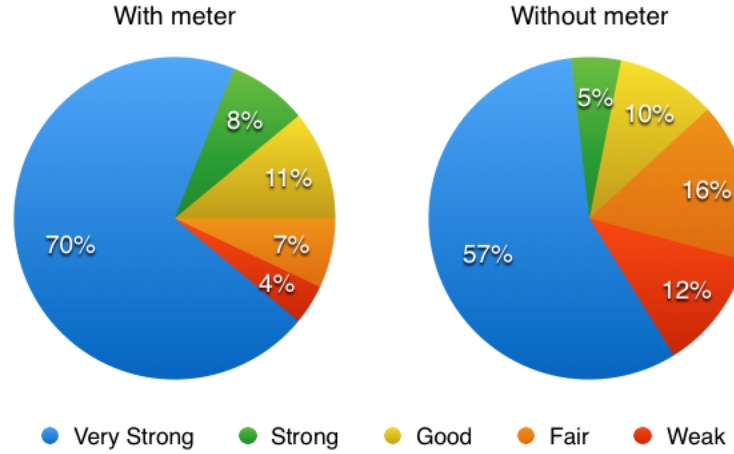


Figure 4.1: Pie charts of verdict frequencies

From the generated overview, a noticeable effect on the partial password strength can be observed. In order to test the null hypothesis  $H_{0a}$ , we performed a t-test on the normalised partial password strengths. The samples from the two groups of participants are considered independent and identically distributed, so we use the unpaired, two-sample, unequal variance test (Welch’s t-test) with two tails. As we previously mentioned, for this test we had decided (*a priori*) to use a significance level of  $\alpha = 0.01$ .

Judging from the results of this process, which is presented in Table 4.3, we observe that the t-statistic of the experimental data is over the critical value. We therefore **reject the null hypothesis  $H_{0a}$  (with 99% confidence) and claim that the partial password strength meter has a measurable positive effect in the strength of the created partial passwords.**

Apart from the main hypothesis that was tested, the answers in the study offer insight in various other aspects of partial passwords. Testing for differences between demographics indicated that there was no significant variation in strength between men and women and neither age seemed to affect partial password strength. Participants aged 55+ had a 12% higher than average score,

	Meter	No Meter
Mean ( $\bar{X}$ )	86.2767	75.3703
Variance ( $\sigma^2$ )	577.7259	1061.4647
# of Observations	100	100
Hypothesised mean difference	0	
Degrees of freedom ( $df$ )	182	
t-statistic	2.6938	
$P(T \leq t)$ two-tail	0.0077	
t critical two-tail	2.6031	

Table 4.3: t-Test: Two-Sample - Unequal Variances,  $\alpha = 0.01$ 

but the population sample is small enough (8) that it can likely be statistical noise. Similarly, workers from India had a lower than average password strength (69.34), but the small sample (9) does not allow us to draw any conclusions.

The experiment and control group displayed a similar distribution in their perceptions of their own password's resilience against partial attacks, both being in general more pessimistic than what the meter suggested (or would have suggested, in the case of the control group). This similar distribution in perceived strength, combined with the significant differences in the actual password strength leads to the control group displaying a slightly positive correlation (+0.225 factor) between perceived and actual strength, while the experiment group having practically no correlation at all (-0.076 factor). This comes as a surprise, considering that 60% of the experiment group indicated that they did not believe the strength meter gave an incorrect score to their password, and only 19% disagreed with the displayed score.<sup>1</sup>

A question in the survey concerned the methods subjects used to count the characters, when answering the partial password challenge; this is of particular interest, since it a unique feature that applies only to partial passwords. We discovered that the subjects' methods of counting the characters did not differ significantly between conditions, as shown in Figure 4.2. Mental counting appears as the main method of counting with 42% of the total subjects reporting

<sup>1</sup>When values do not sum to 100%, the difference exists in the *Neutral* response of the scale.

to use it, counting with fingers came second with 32%, while counting while seeing the password came at a close third, with 26% of participants.

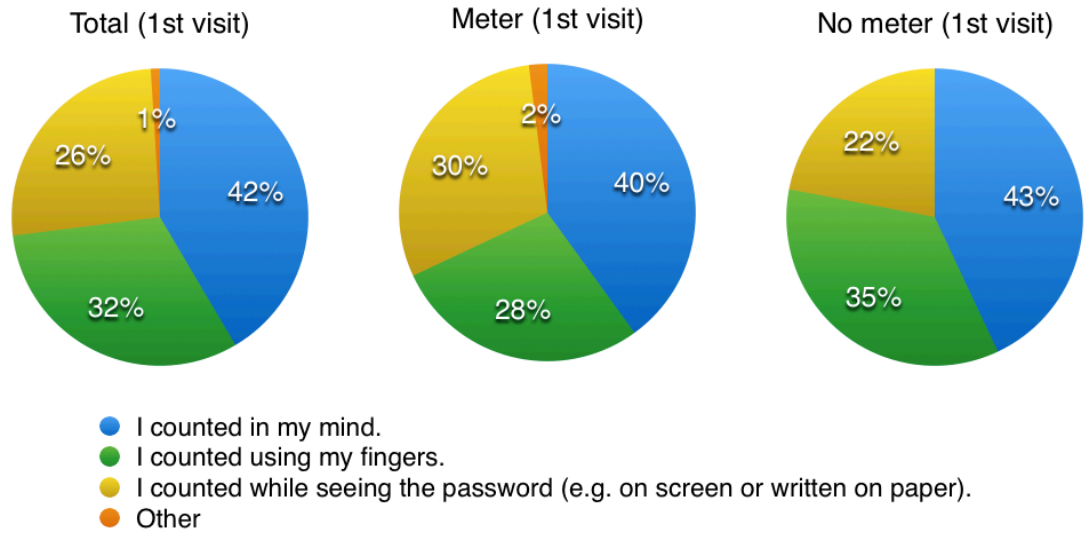


Figure 4.2: Counting methods for the partial password challenge

As this project lies at the intersection of computer security with HCI, it is important to examine the strength meter from a usability perspective, and the first questions in this survey were designed to help do so. Participants in the experiment group were asked to indicate their level of agreement on statements concerning the helpfulness of the password meter on creating stronger passwords, the importance of getting a high rating, the annoyance caused by it and their judgement of its correctness. Subjects in the control group were asked similar questions, but for strength meters in general; the exact wording of the statements can be found in Appendix A.2, and the results are presented in Table 4.4.

Level of Agreement	Helpfulness		Importance		Annoyance		Incorrect results	
	Meter	No meter	Meter	No meter	Meter	No meter	Meter	No meter
Strongly Agree	17	13	25	19	2	15	6	5
Agree	46	52	44	31	5	20	13	22
Neither Agree or Disagree	20	14	16	24	11	7	21	27
Disagree	13	18	9	21	40	41	42	40
Strongly Disagree	4	3	6	5	42	17	18	6

Table 4.4: Participants' beliefs about password meter

We can observe that the subjects in the *Meter* condition had a generally positive

experience with the partial password strength meter. 63% of them reported that it helped them select a stronger password, with only 17% claiming it did not help them, while the vast majority 80% did not consider it annoying. Running omnibus ( $\chi^2$  goodness of fit) tests between the groups, we discovered that a borderline significant difference ( $p = 0.048$ ) difference exists on how important a high score on the displayed strength meters is. Accounting for the bias introduced by the fact that object of the research was clear to the participants, we suggest not accepting this result as significant, as to avoid type I errors. A certain difference between the groups was observed in the annoyance caused by this vs password meters in general, with a probability  $p = 0.0000040 \ll \alpha = 0.05$ . The margin is large enough for us to safely conclude that there is a significant difference in that aspect, but we need to keep in mind that possible confusions between “*password meter*” and “*password policy*” might have influenced the results of the control group.

## 4.3 Effect on partial password memorability

In this section, we detail the methodology followed while testing the  $H_{0b}$  null hypothesis, and report our results, as well as other findings that emerged from the analysis of the collected data.

### 4.3.1 Methodology

Four days after each participant’s completion of the task, they were sent an email informing them about a new, follow-up survey that they could return and complete. For the final part of the study, workers were compensated with 10 cents for attempting to login, while those who were able to remember their password and successfully login were granted a bonus of 22 cents, as advertised. This reward allocation method was used in order to give MTurk workers a monetary incentive to try and recall their passwords, as Ipeirotis discovered that one of their primary motivations is the financial compensation [17]. At the same time this allocation was fair, since workers who forgot their passwords were not presented with a survey and consequently spent less time on the HIT.

Participants were asked to return to our website and answer a short survey, which was presented to them after a successful login using a partial password challenge. It was made clear to them (in the HIT instructions) that they would be able to complete the task regardless of their ability to recall their passwords, by clicking on a “Forgot Password” link, forfeiting the bonus. This was done in order to reassure them and encourage the participation of the subjects that had since forgotten their passwords<sup>2</sup>. It is imperative to note here that during the first part of the study subjects **were informed** that they would need to remember their passwords for future use and asked to employ their usual methods for remembering and protecting an important password.

From the 200 participants invited, 144 returned to complete the second part of the study. Those that succeeded in recalling their password were asked questions about the difficulty in recalling it, the method they used to remember it and (again) the method of counting the characters for the partial challenge. They were also asked how many unsuccessful attempts they made before logging in; this is something that can be measured server-side but it was not implemented and the work by Ur et al. indicated that workers were generally honest when answering those questions, with only 1.6% found lying in their responses [14].

### 4.3.2 Results

The analysis began similarly with the first stage, by counting and grouping the results in order to get an overview of the meter’s effect; the values are presented in Table 4.5. We observe that participants from both conditions were equally likely to accept the return HIT, which is in agreement with our expectations and that there is a very slight increase in memorability for subjects in the *Meter* condition.

In order to examine the null hypothesis  $H_{0b}$ , we performed an omnibus ( $\chi^2$  goodness of fit) test on the actual and expected values. The calculated probability was  $p = 0.51$ , indicating that there is a high probability that the differences were simply statistical noise; we therefore claim that we did not observe a significant effect of the partial password strength meter on the memorability of

---

<sup>2</sup>N.B. the HIT approval rate % is an important metric in the MTurk platform, so workers generally avoid HITs they fear failing

	Meter	No meter	Total
<b>Remembered</b>	52	47	99
<b>Forgot</b>	24	21	45
<b>Returned</b>	73	71	144
<b>Remembered %</b>	71.23	66.20	68.75

Table 4.5: Memorability of partial passwords

partial passwords and **accept**  $H_{0b}$ .

After examining the collected data, there are other interesting observations to be made. As shown in Table 4.6<sup>3</sup> (and the accompanying Figure 4.3), the main method of remembering was memorising the password (67%), with writing them down being the second most frequent way (21%). Ur et al.’s findings were that 38.0% of the participants that returned had stored their passwords, either electronically or on paper, a value that does not significantly differ from our data (32.7%), reinforcing our belief in the validity of the collected data. The results of  $\chi^2$  tests on our data determined that the proportion of participants using each method did not differ between conditions ( $p = 0.741$ ), an outcome that also is in agreement with their findings. Two users who selected the “Other” option reported sending an email containing the password to themselves and taking a photo of the password using their mobile phones. It is regrettable to notice that only a very small portion of subjects in our sample employ password management software (e.g. KeePass, LastPass, 1Password etc), which was one of the most common suggestions security experts offered with regard to best online security practices [37].

Because of the partial password setting of the study, we chose to examine changes in the ways participants used to count the characters between the first and second stage. As seen in Table 4.7, subjects who counted the characters in their minds or while seeing them did not significantly change their behaviour; 78.1% and 80.6% of them, respectively, reported using the same method between the two stages of the study. On the other hand, only 64.7% of the subjects that counted using their fingers indicated that they used the same method the sec-

<sup>3</sup>Sum of values is larger than the number of subjects that remembered their passwords, as they were asked to “select all that apply”



Method	Count
Memorised	74
Wrote down	23
Stored in file	5
Password manager	5
Other	3

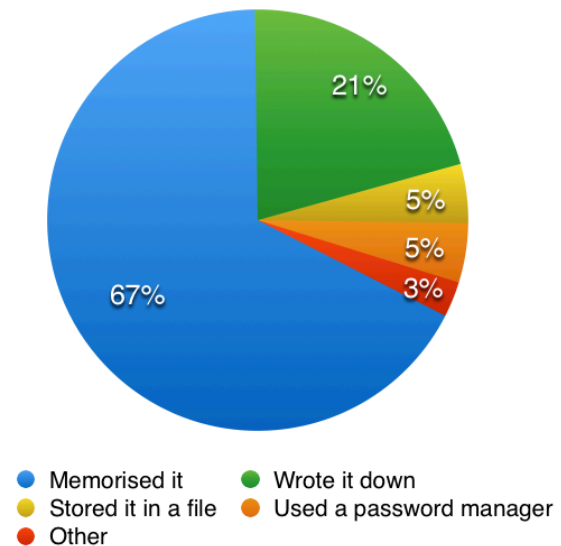


Table 4.6: Remembering methods      Figure 4.3: Pie chart of remembering methods

ond time they logged in. From the workers that changed their way of counting, we observed that most switched to counting while seeing the password, with the “changed” users constituting the 34.2% of the total number of subjects found using the method; the corresponding percentages were 21.9% and 18.5% for mental and finger counting. This behaviour can possibly be attributed to the subjects storing or writing the passwords down as a means to remember it - the coinciding numbers of those that somehow stored their password (36) with those that reported counting while seeing the password (38) further strengthen this assumption.

From/To	Mental	Fingers	Seeing pass
Mental	25	2	5
Fingers	4	22	8
Seeing pass	3	3	25

Table 4.7: Changes in counting methods for partial password challenges



# Chapter 5

## Conclusion

### 5.1 Summary

In this dissertation, we presented the first password strength meter for partial passwords and examined its effect on various security and usability parameters. Preparing for the project, we conducted a survey to gather information about the extent of partial password deployment in the UK and the US and studied existing password strength meter implementations. We assessed the design decisions and their effects, as they were analysed by academic researchers in the past, and drew inspiration for our own design. The main part of the project required devising an accurate partial password strength metric based on the most sophisticated attacks available; the projection dictionary attack found in the work of D. Aspinall and M. Just [11] was used for this purpose.

We implemented the partial password strength meter in JavaScript and in Python, and generated a projection dictionary containing the 1000 best guesses for each possible partial challenge from the RockYou dataset. Validating the correctness of the strength verdicts produced by the scoring algorithm by evaluating known leaked password databases marked the beginning of the second part of the project. A particularly interesting research question was whether this meter would have a significant effect on the created passwords' strength and memorability. In order to test the hypotheses, a website emulating a bank registration process was developed from scratch using the Python-Flask framework and surveys with questions pertaining to the examined topics were designed.

The two-part study involved a registration-login-survey process as the first stage, with a returning login and second survey taking place three days after each subjects' registration date. This study was conducted with 200 participants recruited through the Amazon MTurk crowdsourcing service, equally divided between the *experiment* (were shown a meter during registration) and *control* (no meter displayed) conditions. Statistical analysis of the results indicated that the strength meter we developed had a significant impact on the strength of the selected passwords, while also being received positively by the subjects from a usability perspective.

## 5.2 Project Evaluation

## 5.3 Limitations

self-selection bias from HIT description, MTurk workers not familiar with ppass, run on UK demographic, meter effect during pass creation (not only final result), also measure time spent selecting password, edit distance etc. Dictionary needs to be constantly updated.

## 5.4 Ethical considerations

Everyone has rockyou.phpbb, top 10M so no problem! we did not use the linkedin as it was recent and we did not want to attract further attention to it.

## 5.5 Future work

### 5.5.1 Possible improvements on strength meter

Experimental studies to find a better set of rules.

### 5.5.2 Partial password attack improvements

Continue work on the paper by D. Aspinall and M. Just [11].

Take into account password policies when generating dictionary.

- Case where  $N=62$ ,  $n>X$ . (maximum letter position asked is useful meta-data!)
- Using letters outside of challenge to make better guesses (e.g. knowing  $\#1='1'$ ,  $\#2='2'$  and  $\#5='5'$ , I can make a good guess for 3,4,6 even at w0.)

### 5.5.3 Secure partial passwords storage

Common passwords: salted hash (bcrypt). Because of their unique characteristic in the form partial challenges, they are in cleartext - or possibly encrypting all 3-letter challenges - trivially brute forceable. This is VERY bad in case of db leak.

Reversible encryption as a solution?



# **Appendix A**

## **Partial password strength meter surveys**

### **A.1 Survey about the extent of use of partial password challenges**

## Instructions

We are conducting an academic survey regarding the use of *partial password challenges* in online banking services.

You will **NOT** be asked to provide any identifying information or credentials.

Note that only one submission per worker will be accepted.

"*Partial Password Challenges*" refers to challenges of the form "Please enter characters in positions 3, 4 and 6 of your password/PIN".

You must use online banking services, and your bank must present a partial password challenge during the login process in order to get your submission approved.

**If you are ineligible for this HIT, click 'Return HIT' to avoid any impact on your approval rating.**

**1. Do you use online banking services with a bank that presents a Partial Password Challenge during the login process?**  
**If not, click 'Return HIT' to avoid any impact on your approval rating.**

☐ Yes

☐ No

**2. What is the name of that bank?**

**3. In which country is that bank based?**

**4. What is the type of the password used for the Partial Password Challenge?**

☐ Memorable Word (e.g. "bananas")

☐ Memorable Phrase (e.g. "I love apples")

☐ PIN (e.g. "356982")

☐ Other (Please specify)

**5. Did a strength meter like the one shown in the image below appear (during the registration process), when selecting the word/phrase/PIN to be used for Partial Password Challenges?**

Choose a password:  Password strength: **Weak**  
Minimum of 8 characters in length.

Choose a password:  Password strength: **Fair**  
Minimum of 8 characters in length.

Choose a password:  Password strength: **Strong**  
Minimum of 8 characters in length.

From google.com

☐ Yes

☐ No

☐ Cannot remember

**6. Imagine your password is "recipe73". Answer the following question carefully:**  
**"Enter the following characters from your password"**

Enter the 3rd character:

Enter the 6th character:

Enter the 7th character:

Submit



## A.2 Survey about the useability of the partial password strength meter

### A.2.1 MTurk HIT

#### A.2.1.1 Description

Answer a survey about partial passwords after completing a mock registration process [ 5 minutes] (the registration is used exclusively for this research and only asks for Worker ID - no identifiable information is required).

#### A.2.1.2 Instructions

We are conducting an academic survey about partial challenges on passwords and we are testing a new partial password system.

***“Partial Password Challenges”* refers to challenges of the form “Please enter characters in positions 3, 4 and 6 of your password/PIN”.**

You will **NOT** be asked to provide any identifiable information or credentials.

Please visit the link below (note that JavaScript needs to be enabled), register with the website and then login to complete a short survey. At the end of the survey, you will receive a code to paste into the box below to receive credit for taking our survey.

**Make sure to leave this window open as you complete the survey.** When you are finished, return to this page to paste the code into the box.

Note that only one submission per worker will be accepted.

### A.2.2 Questions

**Q1. To what extent do you agree or disagree with each of the following statements?**

	Strongly Disagree	Disagree	Neither Agree or Disagree	Agree	Strongly Agree
The password strength meter helped me select a stronger password.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think the password strength meter gave an incorrect score of my password's strength.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is important to me that the password strength meter gives my password a high score.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Please select "Agree".	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
The password strength meter was annoying.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Table A.1: Experiment group (with strength meter)

	Strongly Disagree	Disagree	Neither Agree or Disagree	Agree	Strongly Agree
A password strength meter would have helped me select a stronger password.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I think that password strength meters generally give an incorrect score of my password's strength.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
It is important to me that password strength meters give my password a high score.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Please select "Agree".	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
I generally consider password strength meters to be annoying.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Table A.2: Control group (without strength meter)

**Q2. How difficult do you believe it would be for a malicious person to correctly guess the three requested letters from your password?**

- ☐ Very Difficult
- ☐ Dificult
- ☐ Neutral
- ☐ Easy
- ☐ Very Easy

**Q3. Do you use the partial password you selected for any accounts on other websites?**

- ☐ Yes
- ☐ No

**Q4. How did you count the characters to answer the partial password challenge?**

- ☐ I counted in my mind.
- ☐ I counted using my fingers.
- ☐ I counted while seeing the password (e.g. on screen or written on paper).
- ☐ Other (please specify):

**Q5. Thinking back over the past year of using your partial passwords, how many times did you need to reset them (use the “forgot password” feature)?**

- ☐ N/A
- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3
- ☐ More than 3

**Q6. What is your age?**

**Q7. What is your gender?**

- ☐ Male
- ☐ Female
- ☐ Prefer not to answer

**Q8. What is your nationality?**

**Q9. What is your country of residence?**

**Q10. How often do you login to online banking services that require a reply to a partial password challenge?**

- ☐ Daily
- ☐ Weekly
- ☐ Monthly
- ☐ Less frequently
- ☐ Never

## A.3 Return survey about the memorability of partial passwords

### Invitation email

Greetings,

You are invited to take part in another survey regarding partial passwords. You have been awarded the required qualification and you should be able to view and accept the HIT in the following link, provided your HIT approval rate remains above 97%

<https://www.mturk.com/mturk/searchbar?requesterId=A6N1BQ2VBW7RS>

You can complete the HIT even if you cannot remember your password, but there is a significant bonus for those who can, so we suggest you try to recall it.

Thank you for your contributions in our research.

### A.3.1 MTurk HIT

#### A.3.1.1 Description

Return to the website where you created a password a few days ago and answer a small survey about remembering passwords. \$0.22 bonus for workers that remember their password. [1-2 minutes]

#### A.3.1.2 Instructions

We are conducting an academic survey about the memorability of passwords used for partial password challenges.

***“Partial Password Challenges”* refers to challenges of the form “Please enter characters in positions 3, 4 and 6 of your password”.**

Please visit the link below (note that JavaScript needs to be enabled), login using the password you created a few days ago, and complete a short survey (5 questions). At the end, you will receive a code to paste into the box below to receive credit for taking our survey.

If you cannot remember your password, click at the “*Forgot password?*” link at the login page (after you enter your Worker ID) to get the survey code, forfeiting the \$0.22 bonus.

**Make sure to leave this window open as you complete the survey.** When you are finished, return to this page to paste the code into the box.

Note that only one submission per worker will be accepted.

### A.3.2 Questions

#### Q1. How difficult was it to recall the password?

- ☐ Very Difficult
- ☐ Difficult
- ☐ Neutral
- ☐ Easy
- ☐ Very Easy

#### Q2. How did you remember your password (select all that apply)?

- ☐ I memorized it.
- ☐ I wrote it down.
- ☐ I stored it in a file (on my computer / USB stick / other).
- ☐ I saved it in a Password Manager (e.g. browser’s password manager, LastPass, KeePass, 1Password, etc.).
- ☐ Other (please specify):

#### Q3. How did you count the characters to answer the partial password challenge?

- ☐ I counted in my mind.
- ☐ I counted using my fingers.
- ☐ I counted while seeing the password (e.g. on screen or written on paper).
- ☐ Other (please specify):

**Q4. How many (failed) login attempts did you make, before successfully logging in?**

- ☐ 0
- ☐ 1
- ☐ 2
- ☐ 3
- ☐ More than 3

**Q5. Do you use this password for any accounts on other websites?**

- ☐ Yes
- ☐ No

# Appendix B

## Project Website

### B.1 Consent Form

You are being invited to participate in this academic study. This study is being done by Vasilis Gerakaris as part of a MSc Project under the supervision of professor [David Aspinall](#) for the [University of Edinburgh](#).

If you agree to take part in this study, you will be asked to complete an online survey/questionnaire after completing a mock registration process.

The selected partial password (memorable information) will be stored in cleartext (not encrypted) in our database. In simple terms, this means that we will be able to see the password you have selected.

While we do not retain any identifying information other than your Worker ID, we strongly suggest that you do **not** select a password you already use on other accounts.

We believe there are no known risks associated with this research study; however, as with any online related activity the risk of a breach of confidentiality is always possible. To the best of our ability your answers in this study will remain confidential. After the experiment has concluded and statistical results have been generated, all stored information will be permanently deleted.

If you have questions about this project or if you have a research-related problem, you may contact the researcher, Vasilis Gerakaris at the following email

address: s1568962@sms.ed.ac.uk

By clicking “Accept” below you are indicating that you are at least 18 years old, have read and understood this security notice and agree to participate in this research study. Feel free to print a copy of this page for your records.

## B.2 Home page

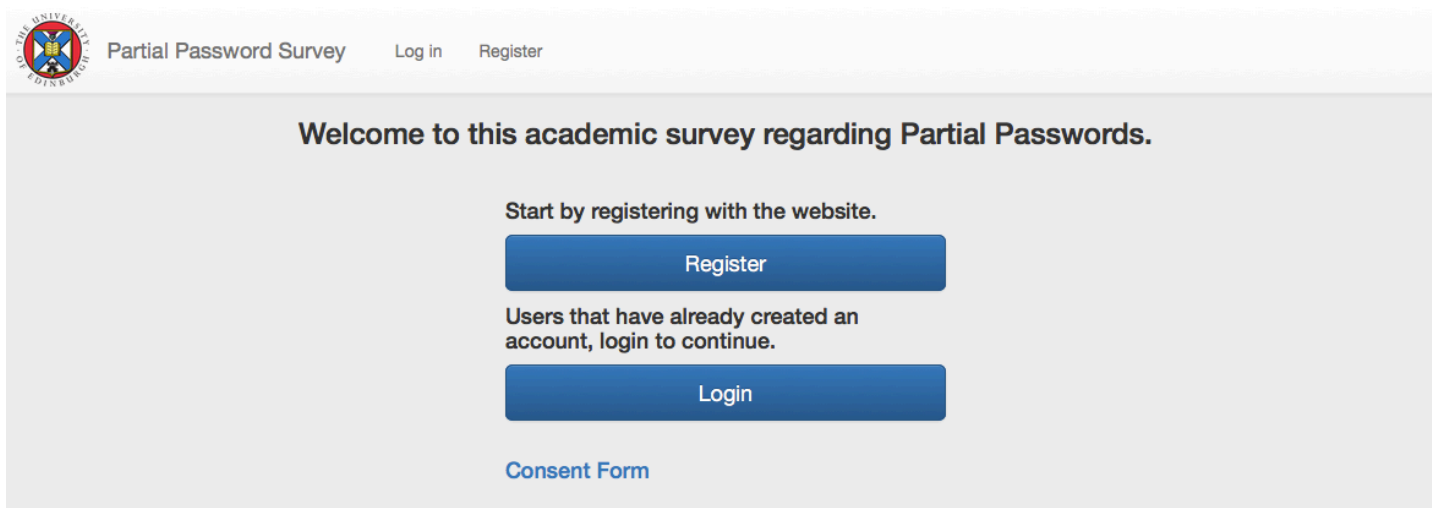


Figure B.1: Homepage

## B.3 Introduction

Imagine that the following is part of the registration process for a bank. The memorable information you select will be used to access online banking services. You will be asked to use this memorable information in a few days to log in again, so it is important that you remember it. Please take the steps you would normally take to remember and protect this information as you would normally protect the ones for your bank account. Behave as you would if this were your real banking credentials.

Do not select a password you already use on other accounts, as I will be able to see them (they are stored in cleartext, not encrypted).



## B.4 Registration page

The figure displays three variations of the 'Register your account' page for Amazon Mechanical Turk:

- (a) Without strength meter:** This version includes fields for 'Amazon MTurk Worker ID', 'Choose a Password' (6-15 characters), and 'Confirm Password' (6-15 characters), followed by a 'Register' button.
- (b) With strength meter:** This version adds a 'Suggest password' button below the password field. The 'Confirm Password' field remains.
- (c) With strength meter and password suggestions:** This version includes a 'Suggest password' button and a list of suggested passwords: 'respectability', 'archbishopric', and 'vengefully'. It also includes a 'Confirm Password' field.

Figure B.2: Registration page

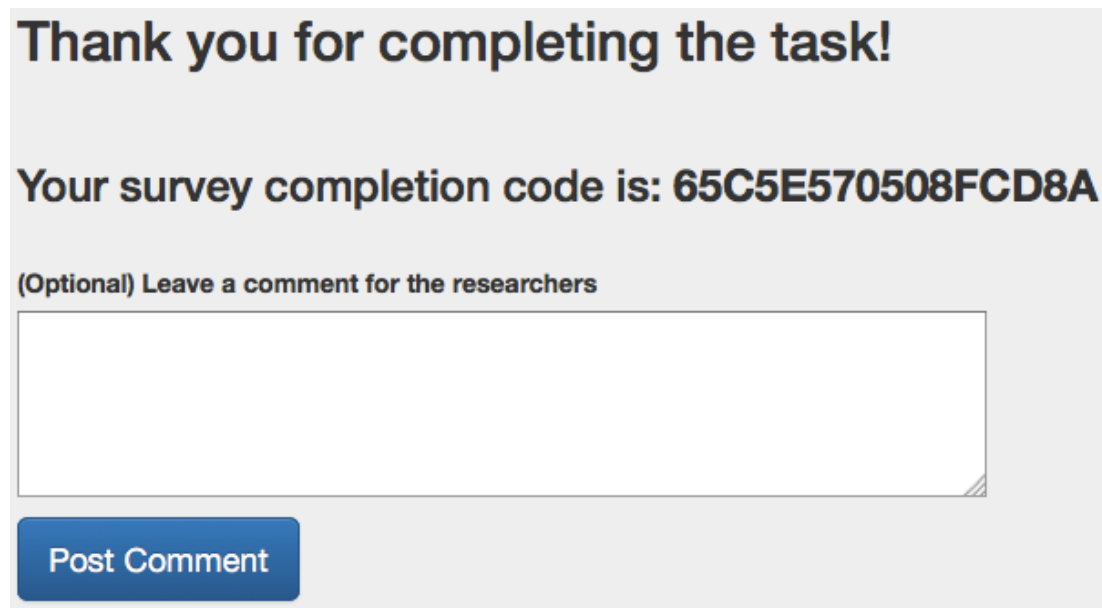
## B.5 Login page

The figure displays three variations of the 'Login' page for Amazon Mechanical Turk:

- (a) Initial login screen:** This version includes a field for 'Enter your Amazon Mechanical Turk Worker ID' (containing 'MTurk Worker ID') and a 'Submit' button.
- (b) Partial password challenge:** This version includes a field for 'Enter your Amazon Mechanical Turk Worker ID' (containing 'TESTUSER') and three character challenge fields: 'Enter the 4th character:', 'Enter the 6th character:', and 'Enter the 7th character:'. A 'Login' button is at the bottom.
- (c) Partial password challenge with "Forgot password" option:** This version includes a field for 'Enter your Amazon Mechanical Turk Worker ID' (containing 'TESTUSER') and three character challenge fields: 'Enter the 4th character:', 'Enter the 6th character:', and 'Enter the 7th character:'. A 'Login' button is at the bottom, and a link for 'Forgot password? (forfeit bonus)' is at the bottom right.

Figure B.3: Login page

## B.6 Completion page



Thank you for completing the task!

Your survey completion code is: **65C5E570508FCD8A**

(Optional) Leave a comment for the researchers

Post Comment

The image shows a survey completion page with a light gray background. At the top, it says "Thank you for completing the task!" in bold black text. Below that, it displays the survey completion code "65C5E570508FCD8A" in bold black text. Underneath the code, there is a label "(Optional) Leave a comment for the researchers" in a smaller font. Below the label is a large, empty rectangular text input field. At the bottom left of the form, there is a blue button with the text "Post Comment" in white.

Figure B.4: Completion page

# Bibliography

- [1] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes,” in *Security and Privacy (SP), 2012 IEEE Symposium on*, pp. 553–567, May 2012.
- [2] M. Just and D. Aspinall, “On the security and usability of dual credential authentication in uk online banking,” in *Internet Technology And Secured Transactions, 2012 International Conference for*, pp. 259–264, Dec 2012.
- [3] M. Dell’Amico, P. Michiardi, and Y. Roudier, “Password strength: An empirical analysis,” in *INFOCOM*, vol. 10, pp. 983–991, 2010.
- [4] 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 *Security and Privacy (S & P), 2012 IEEE Symposium on*, pp. 523–537, May 2012.
- [5] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, “Encountering stronger password requirements: User attitudes and behaviors,” in *Proceedings of the Sixth Symposium on Usable Privacy and Security*, SOUPS ’10, (New York, NY, USA), pp. 2:1–2:20, ACM, 2010.
- [6] 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, (New York, NY, USA), pp. 162–175, ACM, 2010.

- [7] D. Florencio and C. Herley, “A large-scale study of web password habits,” in *Proceedings of the 16th international conference on World Wide Web*, pp. 657–666, ACM, 2007.
- [8] W. C. Summers and E. Bosworth, “Password policy: The good, the bad, and the ugly,” in *Proceedings of the Winter International Symposium on Information and Communication Technologies*, WISICT ’04, pp. 1–6, Trinity College Dublin, 2004.
- [9] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon, “Pass-points: Design and longitudinal evaluation of a graphical password system,” *Int. J. Hum.-Comput. Stud.*, vol. 63, pp. 102–127, July 2005.
- [10] E. Stobert and R. Biddle, “Expert password management,” in *International Conference on Passwords*, pp. 3–20, Springer, 2015.
- [11] D. Aspinall and M. Just, *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, ch. “Give Me Letters 2, 3 and 6!”: Partial Password Implementations and Attacks, pp. 126–143. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [12] P. A. Grassi and J. L. Fenton, “SP 800-63-3. Digital Authentication Guideline (Draft),” tech. rep., National Institute of Standards & Technology, Gaithersburg, MD, United States, August 2016.
- [13] X. de Carné de Carnavalet and M. Mannan, “From very weak to very strong: Analyzing password-strength meters,” in *Network and Distributed System Security Symposium (NDSS 2014)*, Internet Society, 2014.
- [14] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, “How does your password measure up? the effect of strength meters on password creation,” in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, (Bellevue, WA), pp. 65–80, USENIX, 2012.
- [15] M. Buhrmester, T. Kwang, and S. D. Gosling, “Amazon’s mechanical turk: A new source of inexpensive, yet high-quality, data?,” *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

- [16] A. Kittur, E. H. Chi, and B. Suh, “Crowdsourcing user studies with mechanical turk,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, (New York, NY, USA), pp. 453–456, ACM, 2008.
- [17] P. G. Ipeirotis, “Demographics of mechanical turk,” Tech. Rep. ;CeDER-10-01, New York University, March 2010.
- [18] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, “Does my password go up to eleven?: The impact of password meters on password selection,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, (New York, NY, USA), pp. 2379–2388, ACM, 2013.
- [19] M. M. Devillers, “Analyzing password strength,” *Radboud University Nijmegen, Tech. Rep*, vol. 2, 2010.
- [20] D. Wheeler, “zxcvbn: realistic password strength estimation.” <https://blogs.dropbox.com/tech/2012/04/zxcvbn-realistic-password-strength-estimation/>, 2012. Online; accessed 08/2016.
- [21] W. E. Burr, D. F. Dodson, E. M. Newton, R. A. Perlner, W. T. Polk, S. Gupta, and E. A. Nabbus, “Sp 800-63-1. electronic authentication guideline,” tech. rep., Gaithersburg, MD, United States, 2011.
- [22] Alexa Internet, Inc, “The top 500 sites on the web (US).” <http://www.alexa.com/topsites/countries/US>, 2016. Online; accessed 08/2016.
- [23] J. Bonneau, M. Just, and G. Matthews, *What’s in a Name?*, pp. 98–113. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- [24] C. Scott, “Protecting our members.” <https://blog.linkedin.com/2016/05/18/protecting-our-members>, May 2016. Online; accessed 08/2016.
- [25] S. Schechter, C. Herley, and M. Mitzenmacher, “Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks,” in *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, HotSec’10, (Berkeley, CA, USA), pp. 1–8, USENIX Association, 2010.

- [26] A. Ronacher, “Flask (A Python Microframework).” <http://flask.pocoo.org/>. Online; accessed 08/2016.
- [27] M. Otto and J. Thornton, “Bootstrap · The world’s most popular mobile-first and responsive front-end framework.” <https://getbootstrap.com/>. Online; accessed 08/2016.
- [28] M. Baye, “The Python SQL Toolkit and Object Relational Mapper.” <http://www.sqlalchemy.org/>. Online; accessed 08/2016.
- [29] E. J. O’Neil, “Object/relational mapping 2008: Hibernate and the entity data model (edm),” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD ’08, (New York, NY, USA), pp. 1351–1356, ACM, 2008.
- [30] 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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, (New York, NY, USA), pp. 2595–2604, ACM, 2011.
- [31] A. Lomov, “Openshift and cloud foundry paas: High-level overview of features and architectures,” *white paper*, Altoros, 2014.
- [32] “PostgreSQL: The world’s most advanced open source database.” <https://www.postgresql.org/>. Online; accessed 08/2016.
- [33] M. Burnett, “Today i am releasing ten million passwords.” <https://xato.net/today-i-am-releasing-ten-million-passwords-b6278bbe7495>, 2015. Online; accessed 08/2016.
- [34] M. Burnett, *Perfect Password: Selection, Protection, Authentication*. Elsevier Science, 2006.
- [35] M. Burnett, “10,000 top passwords.” <https://xato.net/10-000-top-passwords-6d6380716fe0>, 2011. Online; accessed 08/2016.
- [36] W. M. Vagias, “Likert-type scale response anchors,” *Clemson International Institute for Tourism & Research Development, Department of Parks, Recreation and Tourism Management. Clemson University*, 2006.

- [37] I. Ion, R. Reeder, and S. Consolvo, ““...no one can hack my mind”: Comparing expert and non-expert security practices,” in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, (Ottawa), pp. 327–346, USENIX Association, Jul 2015.