Waiting Makes the Heart Grow Fonder and the Password Grow Stronger

Experiments in Nudging Users to Create Stronger Passwords

Nathan Malkin

Spring 2013

Despite the pervasiveness of passwords and the importance placed on them as an authentication mechanism, users continue to create weak passwords. Explanations for this behavior range from laziness to lack of education to the rational rejection of security advice. This paper contributes concrete evidence about the effects on password strength of specific conditions, including monetary incentives, priming, strength meters, demographics, and other factors.

In laboratory experiments on Mechanical Turk, we found that the most significant predictor of a strong password was the time spent creating it. Furthermore, we present evidence that, under certain conditions, artificially increasing the amount of time spent on password creation — for example, by asking users to wait on the password creation screen — results in stronger passwords.

In empirical data and survey results, we also found evidence that users make calculated decisions based on the perceived value of creating more or less secure passwords. Additionally, we provide the results of quantifying the costs of better passwords through variables such as recall, willpower, and cognitive effort.

Contents

Acknowledgements

Shriram Krishnamurthi and David Laidlaw have advised me throughout all four of my years at Brown; without their help, this project would not have come to fruition.

I am also grateful to all those whom I inundated with my discussions of the topic, as well as those who (in spite of this) provided me with support, encouragement, feedback, ideas, and passwords.

"I like to use more secure passwords because the world is a scary place."

— survey respondent

1 Introduction

For most of the history of computing, passwords have been the prevailing mechanism of security and authentication. They have faced few challengers and reign supreme on personal computers, networks, and websites.

Despite passwords' ubiquity and long history, the habits associated with their use, exhibited by the general population, are widely regarded as poor. Password reuse is rampant, and users consistently choose weak (i.e., easy to crack) passwords, which is especially dangerous in light of the exponentially faster modern hardware and increasingly sophisticated attack methodologies.

The orthodox approach to this problem has been to treat users as irremediably lazy, if not stupid. This point of view is characterized by the use of password composition policies [3]: rigid requirements that a password must pass before it is accepted by a system (e.g., minimum lengths, the use of capital letters, digits, special characters, etc.) . In one form or another, these can be frequently found in a variety of venues.

An emerging view has taken a more charitable approach. Rather than blaming users' laziness, it has identified lack of education and understanding as the cause of the problem. This perspective is exemplified by password strength meters [4], which are now commonly found on many websites (for example, 70 of the top 100 websites, as ranked by Alexa, use them). These meters do not force any changes to the passwords users create but provide visual feedback about their strength, serving as a way of educating users about their habits.

A more radical view of the situation has been proposed by Herley [2], who argues that users can be seen as engaging in a rational rejection of security advice. Rejecting security advice, like the advice to create a better password, can be a rational decision based on a comparison of the costs of secure behavior to the expected value of its benefits. According to this rational-agent model of security behavior, a person performs a mental calculation each time he or she needs to create a password: does the cost of creating a more secure password exceed the expected return? If so, they will create a less secure password. Under this model, users are sensitive to the pay-offs and probabilities associated with their account being compromised — and, importantly, to the extent to which their behavior can affect these.

This model explains the pervasively poor password habits of the population: for most accounts, the cost of protecting them exceeds their value. It has intuitive appeal, but it has never been empirically tested. Furthermore, it leaves a number of open questions.

- How is the decision to accept or reject security advice made? According to the model, a decision is made on a per-account basis. Is this something that happens consciously or unconsciously?

- How do we assess benefits of having an un-compromised account? This must be a very complex — and likely inaccurate — assessment, as it combines a number of tasks that humans are very bad at: working with low probabilities, personal information of uncertain value, outcomes far into the future, etc.
- What are the costs that we consider? When we are comparing the costs to the benefits, how are we assessing the costs? What enters into the calculation?
- And, of course, is this model actually true? Are we really sensitive to the expected value of the outcome when creating passwords?

These questions are important because they determine how and why we decide to make stronger passwords. Understanding the answers will allow us to architect choices based on these considerations, so as to nudge users into more secure behavior. We therefore begin our study seeking to address these questions.

2 Methodology

This section discusses the methodology of the study, including participant recruitment, experimental conditions, and rationale for the main measures used.

2.1 Mechanical Turk

Participants for the experiment were recruited using Mechanical Turk. To avoid priming subjects by focusing their attention on passwords, the task was advertised as "a short survey about yourself and your browsing habits" and its short duration (approximately two minutes) was emphasized. Participants were unable to see the full instructions until beginning the experiment. The payment for participation in the experiment was \$0.05, with additional bonuses offered under various conditions.

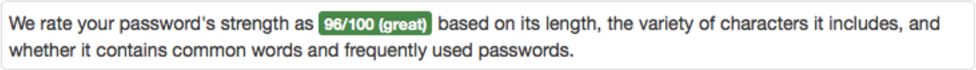
2.2 Common Procedure

All participants in the experiment proceeded through the same sequence of steps.

First, each saw a screen that provided specific instructions for their task, detailed the compensation structure, and asked for consent. The instructions explained that the participants would be taking a two-part survey: the first section would be completed immediately, while the second section, for which they would receive a separate compensation of \$0.10, would take place "in several days." The subjects were told that they would be required to create a password, which they would later use to access the second portion of the survey.

The subsequent screen showed a survey, soliciting the subjects' gender, age, country of residence, and whether their job was related to the field of Information Technology. Participants were able to opt out of any questions they did not wish to answer.

The third screen asked subjects to create a password. To proceed, participants needed to retype the password for verification, mirroring the analogous requirement used by



We rate your password's strength as 96/100 (great) based on its length, the variety of characters it includes, and whether it contains common words and frequently used passwords.

Figure 2.1: Password strength meter used in the study

just about all real-world password creation procedures. The password field could not be left empty, but there were otherwise no requirements (including minimum length) that a password had to meet. A message on the screen reminded participants of the compensation they could expect to receive under their specific condition.

2.3 Strength Meter

An additional feature of the password creation screen was a password strength meter (see Figure 2.1). As users typed in their prospective password, the meter updated to provide feedback about the quality of the password so far. The information was conveyed through three mechanisms:

1. Coarse category. Users were told if their password was “terrible,” “bad,” “average,” “good,” or “great.”

2. Color. Each category was associated with a color, from red for “terrible” to green for ”great.”

3. Numeric score. The meter displayed a number between 0 and 100, representing the quality of the password.

A password’s score is computed using the following method:

- 4 points per character in the password
- 17 points for the first uppercase character, digit, or symbol
- 8 points for a second unique uppercase character, digit, or symbol
- 4 points for the third
- 17 points were deducted if the password contained no lowercase letters
- 17 points were awarded if the password passed a dictionary check

The dictionary check tested if the password could be found in an English dictionary or in the OpenWall list of common passwords.

Note that the total sum of points may be above 100, but the numeric score displayed to the user is capped at 100.

This procedure is based on the `comprehensive8` heuristic described by Ur et al. [4]. Our method differs from the original in two ways:

1. The `comprehensive8` heuristic specifies a minimum password length of 8 characters, which we did not enforce.
2. Ur et al. use a different dictionary: the OpenWall mangled word set.

This heuristic for measuring password strength and its display is representative of the strength meters used by many modern websites.

2.4 Optional Stroop Task

After creating their passwords, participants were asked if they wanted to complete an additional task that took exactly one minute for an extra \$0.05. Before deciding, they were able to see the instructions for the Stroop task, a task commonly used in psychology that has been applied to measuring "ego depletion," a proxy for willpower and cognitive effort.

In the Stroop task, subjects were shown a randomly chosen word from a list of color names (e.g., red, green, blue, etc.). The color in which the word was displayed was also randomly chosen from the same list. To proceed, the participant had to type in the color in which the word was displayed (as opposed to the color identified by the word itself).

When the participant succeeded, a new color-word pair was given to the participant. The task cut off after 60 seconds.

2.5 Return Survey

After at least 48 hours had passed since a participant had completed the initial task, they were sent a message inviting them back for the second half of the task, a two-minute survey for which they would earn \$0.10.

A link from the invitation directed users to a web page, where they could attempt to log in using the password they had created. If they succeeded, they proceeded to a survey that asked several multiple-choice and open-response questions about their password habits.

Participants who did not successfully remember their password were able to click on a button labeled "I forgot my password" and complete the survey anyway.

2.6 Introducing Incentives

The main theory we are testing suggests that users will not expend effort on security unless there is a tangible benefit they may receive or, equivalently, a distinct cost associated with ignoring it. Thus, in "real-world" situations, passwords protect confidential information or individuals' identities; the compromise of these accounts has concrete costs, often quantifiable ones. When passwords are created in a laboratory setting, however, there are no apparent losses that a user will experience if their account is compromised. The threat model is also less clear to participants: who is the attacker, how will they attack, and — perhaps most importantly — why? Therefore, to test the theory of rational rejection of security advice, we must directly create incentives for creating secure passwords (or sanctions for creating less secure passwords).

Password Strength	Probability of Bonus
Terrible	50%
Bad	55%
Average	60%
Good	65%
Great	70%

Figure 2.2: Pay-off Matrix for Incentivized Condition

We implement incentives through a lottery mechanism. Participants in the incentivized condition are informed that a randomly drawn number will determine if they receive a bonus of $0.05. The "base probability" of this event is 50%, and the probability increases in 5% increments with each category improvement in the password strength. Thus, a move from a "terrible" to a "bad" password will increase the probability to 55%, and a "great" password will increase the participant's chance of earning the bonus to 70% (Figure 2.2).

As the expected value of the lottery increases if participants choose stronger participants, we expect, under the rational model, for participants in this condition to create better passwords.

2.7 Control Condition

As a control for the incentivized condition, we create a "baseline" condition in which there is no incentive for creating stronger passwords. To more closely match the conditions of the incentivized situation, participants in this condition are also entered in a lottery for $0.05. However, the subjects cannot control the likelihood of the outcome: the probability is fixed at 50%. The participants' actions cannot affect this number; the strength of their password, in particular, has no effect on the outcome.

2.8 Probing for Priming

It can be conjectured that any improvements in password strength exhibited in the incentivized condition are due to priming effects. Focusing the attention of participants on passwords, rather than creating incentives for them, could be causing changes in password strength. To account for this possibility, we introduce an additional condition, "primed."

This condition is identical to the control condition in that it includes a lottery for $0.05 with a fixed probability of 50%. It is differentiated by its use of language that evokes the threat model: the probability of the event is described as the likelihood of the subject's password being "cracked."

For consistency, the same language is used in all conditions other than the control.

Wait Length	Probability of Bonus
no wait	50%
5 seconds	55%
10 seconds	60%
15 seconds	65%
20 seconds	70%

Figure 2.3: Pay-off Matrix for Time-Incentivized Condition

2.9 Time/Effort Trade-off

In the incentivized condition, we test whether people will substitute the effort of creating a better password for an increased likelihood of a monetary bonus. But how motivating is this reward? We would like to provide an alternate method for participants to obtain this outcome, to gauge whether subjects seek it out under different circumstances.

In this condition, we again use a lottery of $0.05 with a base probability of 50%. Subjects can increase this probability by 5% by clicking on a button and waiting for five seconds. They can repeat this procedure up to four times, for a final probability of 70% (Figure 2.3). The expected values under this condition are therefore equivalent to those under the incentivized condition; the difference is whether they are obtained through time or effort.

This set-up allows us to examine the trade-off between time and effort in creating passwords. If a greater proportion of participants elect to wait under this condition than choose "great" passwords under the incentivized condition, then we can conjecture that the effort of creating a better password contributes is more salient to users than the time costs.

2.10 Assigning Conditions

To mitigate order effects, participants were sequentially assigned to each of the main conditions as they arrived.

2.11 Password Score as Measure of Strength

How do we measure password strength? In our study, we choose to use a password's `comprehensive8` score, as described above and displayed by the strength meter. It should be noted that this heuristic is not the best measure of a password's "guessability" under real-world attacks. Other techniques will more accurately report the susceptibility of a password to state-of-the-art attacks. However, this is the best measure of the participants' intent. Outside of a user's vague intuition about what makes a secure password, the score displayed by the strength meter is the primary method of feedback a participant gets about the strength of their password. A subject who decides to create a secure password is therefore most likely to follow the feedback of the meter.

In our study, we are examining the conscious decisions of users to create more or less secure passwords. Under these circumstances, it is the intent of a user that matters most, and the password score, as shown to the user by the strength meter, is the best measure of this value.

3 Results and Analysis

This section summarizes the data collected according to the conditions above and presents a statistical analysis thereof. We also discuss several additional conditions implemented and their results.

3.1 Who are the participants?

A total of 146 subjects participated in our study, approximately n=20 per condition. Of these, 49% reported themselves as male, and 51% as female. 69% resided in the United States, while 31% were from India. 31% of subjects also reported themselves as speaking multiple languages, with the most represented non-English languages between Tamil (spoken by 17%) and Hindi (10%). 21% said they worked in a field related to Information Technology, while 75% said they did not. The mean age was 32, and the median age was 29.

Across all conditions, 54% of participants returned to complete the follow-up survey. Of these, 72% were able to recall their password successfully. Out of these, 61% said they committed their password to memory, 29% reported writing it down, and 4% used software to store it. When surveyed about how they usually create passwords, 31% of the returning participants said they created a new password for each site, the same number said they had a base password that they changed slightly every time, 33% reported reusing one of several passwords, and none admitted to using a single password for all of their accounts.

3.2 Testing the null hypothesis: do any of our conditions matter?

The null hypothesis in our study is that the conditions surrounding password creation have no effect on the strength of the resulting passwords. Under this view, people follow a set procedure when asked to create a password, and this procedure does not change, regardless of circumstances or incentives.

To test the null hypothesis, we performed a one-way ANOVA with the password strength score as the dependent variable. We found statistically significant differences between the conditions (F=4.759, $p < 0.001$), allowing us to reject the null hypothesis: password strengths do vary depending on the conditions of creation.

3.3 Does priming matter?

Not all differences between conditions were significant, however. Participants who were primed with language about their password being cracked did not create passwords that were significantly more secure than those in the baseline condition ($t = 10.188$, $p \approx 1$).

The password strength meter could also be considered to be priming users in relation to passwords. To test this, we conducted an additional condition identical to the baseline but without a strength meter. We found no significant difference between password strengths in this and the baseline condition ($t = 9.780$, $p \approx 0.98$).

This finding is consistent with the theory of rational rejection of security advice. Since the expected value of the outcomes is identical in both conditions, users have no incentive to take on extra security measures in one case or another.

3.4 Do incentives result in stronger passwords?

We now examine the incentivized condition. Would people create better passwords if doing so would increase their expected pay-off? Our data showed mixed results.

The median password strength in the incentivized condition was 106, compared to 80.5 in the baseline. A least-significant difference (LSD) pair-wise comparison identified the results of the treatments as being significantly different ($t = 9.560$, $p < 0.05$). However, the Gabriel procedure, which attempts to correct for family-wise error rates, did not find the populations to be significantly different (mean difference $= 21.763, p > 0.05$).

Consequently, we conclude that while our incentives did induce stronger passwords, this effect was weak.

3.5 What are the costs?

The incentives implemented in our conditions appeared to produce only small improvements in password quality. There are two primary ways in which this result can be interpreted.

1. We can use this to reject the rational theory and support the null hypothesis that conditions do not influence password strength.

2. We can presume that actors are behaving rationally, but that, in this particular situation, the costs of secure behavior (nearly) outweighed the expected benefits, for at least some of the subjects.

To analyze the results from this perspective, we must gain a more precise understanding of the costs of better passwords. Several explanatory theories are available.

- *Null hypothesis.* We must entertain the possibility that strong passwords are just as easy for users to come up with as weak passwords.

- *Stronger passwords are harder to remember.* They are more likely to be longer and include numbers, special characters, and capitalizations. Such features do not follow predictable patterns and are more likely to be forgotten, in contrast to simple passwords, like dictionary words.

- *Stronger passwords are harder to come up with.* As with any task, we expend cognitive effort when creating passwords. Security features, such as those described above, may require users to spend more effort, which people may wish to avoid.

We proceed to test each of these theories using the data we have obtained.

3.6 Are better passwords harder to remember?

To test if stronger passwords are harder to remember, we examined whether returning users were able to successfully log in. (Those who could not were able to indicate this by pressing a button, "I forgot my password.") Using a binary regression to fit a logistic function to this data, we found a significant ($p < 0.05$) *positive* effect of password strength on being able to successfully log in. That is, users with stronger passwords were less likely to have been forgotten them.

While this result is counterintuitive, we hypothesize that we are seeing the effect of password storage software. Users of this software do not need to remember their password, as the software stores it for them. Furthermore, many such applications also offer users an option to automatically generate (and store) passwords for them. Being machine-generated, such passwords are likely to be very secure. A similar effect can be attributed to writing down passwords.

3.7 Are better passwords harder to come up with?

To measure the cognitive effort expended by participants in our experiment, we use the Stroop tasks, the mechanism and rational behind which is described above. There are two primary measures available to us:

1. Participation rate. People who have exhausted their willpower after creating their password could choose not to participate in further tasks.

2. Iterations completed. There were no incentives attached to completing more iterations of the Stroop task. Subjects experiencing ego depletion could accept the task but not actually perform it; or, they could simply proceed through it at a slower pace.

While we predicted that the participation rate in the Stroop task may be correlated with password strength, we found the participation to be very high: 96% across all conditions.

A linear regression between password strength and the number of Stroop task iterations completed showed a significant ($p < 0.01$) *positive* correlation between the two. Thus, according to our measure, stronger passwords are *not* "harder" to come up with.

3.8 Would we rather wait than create a good password?

An alternate way of testing if cognitive effort contributes to the cost of password creation is substitution. We allow subjects to achieve equal gains in expected value by substituting an alternate cost for cognitive effort; we can then see if participants are motivated to work for the outcome under this condition. As described above, the substituted "good" is time: subjects were given the choice of waiting for five seconds at a time to increase the probability of the bonus by 5%.

Most of our participants were unwilling to make this trade-off. Only 27% of them chose to wait 20 seconds and increase the probability of their bonus to its maximum possible values of 70%. The rest of the subjects, in equal numbers (36% each), chose to wait for only five seconds (for a 5% increase) or not to wait at all. In contrast, 96% of participants in the incentivized condition created a "good" or "great" password.

Based on these results, we conclude that, at this substitution rate, cognitive effort was not more valuable to our participants than their time.

The time-incentivized condition provided us with additional, more surprising results. In this condition, the outcome was fully independent from the strength of the password created. Users have no incentive to make their password better, and we would therefore expect to see password strengths on par with those in the baseline condition. However, the password strengths between these two conditions were significantly different ($p < 0.05$).

3.9 Why are the passwords in the time-incentivized condition stronger?

Despite a lack of incentives, participants in the time-incentivized condition chose passwords significantly stronger than those in the control condition. We offer two explanations of this phenomenon.

1. The participants found the presence of buttons that provided them some control over the outcome to be motivating and chose stronger passwords as a consequence.

2. While they were waiting to accrue the extra probability, participants were able to dedicate more time to creating a stronger password.

We focus on the latter theory, as it is more plausible and easier to test.

3.10 Will idling users create better passwords?

According to our explanation, subjects in the time-incentivized condition created stronger passwords, contrary to their incentives, because they had time to devote to picking a stronger password while they waited for the timer to run out. However, as discussed above, not all participants chose to wait, among other confounding factors.

To test this theory in the absence of confounding factors, we created a new condition. It was identical to the "primed" condition in its language and in the pay-off structure, i.e., a fixed-probability lottery. It was different in that the password creation screen displayed a message asking the participants to wait for 20 seconds. (The countdown began immediately when the screen loaded.)

The results from this condition supported our theory: though again there was no incentive, participants created passwords significantly stronger than passwords in the baseline condition ($p < 0.05$) and statistically indistinguishable from those in the incentivized and time-incentivized conditions.

Thus, users forced to stay on a password creation screen for a fixed amount of time create stronger passwords.

3.11 Does waiting longer help?

The finding that making users wait results in stronger passwords invites a follow-up question: does making them wait longer result in even better password? We answer this question by replicating the previous condition with one modification: rather than waiting 20 seconds, the participants are asked to wait 40 seconds.

Under the new circumstances, we find that users do not create stronger passwords. There were no significant differences between the two conditions.

Thus, we conclude that the returns on making a user wait are diminishing, with password strength likely plateauing at a certain point.

3.12 Forced timeouts: will people cooperate or cheat?

While we have found that our technique improves password strengths, users may be unhappy with being forced to wait for no apparent reason and without an explanation. Indeed, Egelman et al. [1] found that users exhibited limited tolerance for delays in software and were significantly more likely to cheat on a task in the absence of a detailed, security-minded explanation about the reasons for the delay.

To test users' tolerance of the delay, the interface included a loophole that allowed participants to cheat on the task. While the password creation screen asked subjects to wait, clicking on the "Continue" button allowed them to proceed even if the countdown had not yet finished. This option was emphasized by a visual cue that showed the button as enabled for the duration of the countdown.

Despite this opportunity, we found the actual occurrence of cheating to be very limited. In the 20-second timeout condition, only one subject cheated, and only by two seconds. While we might expect that subjects who had to wait longer would be more frustrated and more likely to cheat, there was also only one case of cheating in the 40-second timeout condition. Completion times indicate that most subjects waited for the timer to run out and immediately continued with the task (as opposed to switching to a different task on- or offline and returning at some point later).

3.13 How can waiting be used in real-world interfaces?

Proposal 1

Our findings so far suggest that it may be possible to nudge users into creating stronger passwords by making them wait. However, directly translating these conditions into real-world user interfaces is in most cases undesirable from the point of view of user experience, as typically websites wish to minimize the amount of time users spend on tasks like account creation. An additional consideration is that the waiting period penalizes those who create strong passwords quickly: there is no reason to keep them waiting.

In light of this, we propose a mechanism that provides for a graduated waiting period. Users with the best passwords would not need to wait at all, but those with weaker passwords would be required for up to 20 seconds: the weaker the password, the longer the wait (Figure 3.1). The interface would inform users about the waiting periods ahead

Password Strength	Waiting Period
Terrible	20 seconds
Bad	15 seconds
Average	10 seconds
Good	5 seconds
Great	no wait

Figure 3.1: Pay-off Matrix for Graduated Waiting Period

Password Strength	Waiting Period
Terrible	20 seconds
Bad	20 seconds
Average	20 seconds
Good	20 seconds
Great	no wait

Figure 3.2: Pay-off Matrix for Graduated Waiting Period & Higher Penalties

of time. As a result, those inclined to choose weak passwords would be motivated to make them stronger.

Results

We implemented this proposal within our standard experimental framework. Though we expected performance to be on par with the universal waiting time condition, the actual results show that the average password was much weaker: the median password strength was 87, compared with 103 when all participants had to wait.

Proposal 2

What happened? One explanation is that the "penalties" were not strong enough. Users with "average" and "good" passwords were required to wait 10 and 5 seconds, respectively, which is comparable to the amount of time they would need to go back and change their passwords. Therefore, they had little incentive to do so and could be expected to stick with their original passwords instead of making them stronger.

To test if this is the case, we introduce harsher penalties. Users with the best passwords still have no waiting period, but all others will be required to wait for 20 seconds (Figure 3.2). This increases the motivation for those with "average" and "good" passwords to bring them up to be "great."

Results

Contrary to expectations, the harsher penalties did not have a significant effect. The median password strength increased, but only to 91, still considerably lower than the

Variable	Unstandardized β	σ	Standardized β	p
India?	-16.121	5.982	-.220	.008**
Male?	1.855	5.199	.027	.722
Age (years)	-.464	.262	-.137	.080
Time spent (seconds)	.421	.088	.420	.000***
Incentive?	9.746	7.386	.112	.190
Priming?	11.897	7.446	.134	.113
Not memorized?	-11.894	9.269	-.112	.199
New password?	4.152	7.740	.045	.593
Strength meter?	18.752	7.394	.204	.020*

$R^2 = .347,\ constant = 63.442^{***}$

Figure 3.3: Regression Results

results of the universal waiting period.

We conclude that not all forced waiting periods result in stronger passwords. Instrumentation of the condition matters.

3.14 Are we purely rational?

We now return to the question with which we started: do people make rational decisions when deciding to create more secure passwords?

While we cannot disprove this theory, the accumulated evidence provides for a number of other factors (e.g., time). In fact, we can be quite confident in claiming that this is not a purely rational decision. If it were, subjects in the absence of incentives would put the minimum possible effort into creating passwords, resulting in scores approaching zero. However, even in the baseline conditions, the median strength is quite high: 80.5.

3.15 What other factors can affect password strength?

In light of this, we must consider other factors that may have an influence. In addition to time, incentives, and priming, these include demographic data and password creation techniques.

We incorporate all of these into a multiple regression, testing the effect of all of these factors on password strength.

3.16 How does it all fit together?

The table in Figure 3.3 summarizes the results of the regression test. The unstandardized β values report the effect of each variable on the raw password score, while the standardized β values represent each variable's contribution to the model relative to the others.

Expectedly, the gender of the participant had no effect on the passwords chosen. Whether the user created a new password or selected one they already had also made no difference.

The overall contributions of incentives, priming, and whether the participant memorized the password (or used an aid, such as writing it down) were not very significant and about equal to each other.

According to the regression analysis, the presence of a strength meter did significantly enhance password scores.

A more unexpected result is that participants from India had significantly lower password scores. Possible explanations include a smaller degree of penetration by the web into daily life (and, consequently, less private information online to protect), language, and general cultural differences.

Finally, we see that, in harmony with our previous findings, the time spent creating a password had the greatest relative contribution to password strength as well as demonstrating the most significant effect.

4 Discussion

This section summarizes the major findings, provides possible explanations, and discusses their limitations.

4.1 Expectations

While none of the results subverted to common sense, some of our findings ran contrary to the expectations we had. We adopted the model of rational rejection of security advice and predicted that users would only expend effort on more secure passwords if their payoff depended on it. We identified cognitive effort as a potential cost of creating a more secure password and predicted that we could measure its effect through ego depletion. We also hypothesized that participants would value cognitive effort over time and would therefore be more likely to wait to gain an expected reward, rather than putting the effort to create a more secure password.

Instead, our results often times showed the opposite of what we predicted. The effects of incentives were weak, and participants consistently created strong passwords even when they had zero incentives for doing so. Most subjects opted not to trade off their time for an outcome with a higher expected value. The search for ego depletion yielded exactly the opposite of what we expected: people who ostensibly expended more effort on creating stronger passwords completed more work as part of the Stroop task.

However, these results can be reconciled by way of the following explanations.

4.2 Explanations

4.2.1 Rationality

In our results, participants responded weakly to incentives for creating stronger passwords. This may suggest that, when making security decisions, people do not take into account the expected value of the outcome. However, responses collected from participants in the survey portion of the task indicate otherwise.

Respondents repeatedly report that they choose more secure passwords for higher-value accounts: “I try to use a more secure password with more important sites (such as banking, financial).” In contrast, many mentioned that they chose simpler passwords for less valuable accounts and were more likely to reuse passwords for these accounts. Several respondents spoke directly to the trade-off:

> “The more valuable the account (bank accounts, investing, etc) the more complex my password will be. For random forum accounts, I usually use the same simple password because I am not concerned with security.”
>
> “Anything having to do with money, bank, credit cards, etc gets a more secure password. Things like accounts to read a newspaper or log on to pinterest get something easier to remember and less secure and more likely to be repeated across those kinds of less critical accounts.”

Another user made similar choices and explained their thought process in detail:

> “Some of my passwords are more secure because I feel there’s a need for extra security on them. For example, my bank log in information is completely different from anything I’ve ever used anywhere else. Same deal for my loan log in. I would never use the same passwords for these examples as I would for something like YouTube or a junk email account. If those were compromised because someone figured out my password it would not devastate me whereas my bank and loan, which has all my personal info, would devastate me if someone got a hold of it.”

This is precisely the kind of analysis predicted by the rational model. Why did it not take place in the experiment?

The most likely explanation is that the incentives offered, the stakes at play, were simply not high enough. Users are used to dealing with passwords that protect high-value information, like credit card numbers, private correspondence, and other confidential materials. When what is at stake is a fraction of a penny, this does not enter into calculation.

While users do take into account outcomes and make decisions based on the value of what is at stake, this happens at a ‘macro’ level; they do not display minute sensitivity to expected values, contrary to the predictions of the rational model.

4.2.2 Time

Our findings about the role of time come in two parts. First, we saw unwillingness by subjects to trade time for a higher probability of a bonus. This, like the weak response to incentives, is best explained by the low value of the proposition. With an expected earning rate of \$0.05 per minute, the opportunity cost of idling for 20 seconds is \$0.017, while the expected value of waiting is \$0.01. Furthermore, this calculation does not take into account the well-studied phenomenon of risk aversion, which pushes down the utility of our lottery even further.

Our results also show a very significant correlation between time spent creating a password and the resulting strength. While this is perhaps expected, somewhat surprising is the finding that this relation can be causal: artificially inflating the time spent on creation increases the strength of the result (up to a point). The most plausible explanation, supported by the data in our extra conditions, suggests that the additional time allows users to reflect on their password and choose a stronger one, if they wish.

However, other explanations may be possible, and this phenomenon merits further investigation.

4.2.3 Stroop task

Rather than displaying evidence of ego depletion, the results of the Stroop task suggest that participants with stronger passwords had more rather than less willpower. We offer three theories to explain this observation.

If we take the results at face value, we can conclude that creating a stronger is not actually a depleting task — i.e., it does not use up more willpower. Given that a password can be strengthened simply by adding a few characters, which takes little time and effort, this is plausible. However, this is not an entirely sufficient explanation, as it does not explain why there is a positive correlation between strength and Stroop task completion, rather than no correlation.

To account for this, we posit that the Stroop task's measure of ego depletion due to password creation is being confounded by another variable: overall effort. It is a well-documented phenomenon among studies on Mechanical Turk that some workers will complete their tasks earnestly, while others will attempt to get by with the minimum amount of effort. In our study, these two types of workers would manifest as follows: the earnest worker will make a good faith effort to create a strong password and engage completely in the Stroop task, trying to complete as many iterations as possible. On the other hand, a disinterested worker will put little effort into creating a stronger password and may not complete any Stroop tasks at all, since there is no incentive to do so (they will still get paid if they sit idly during the task). These differences in behavior would be more pronounced than any possible ego depletion, making this a confounding variable.

To correct the assumptions that lead to this, we suggest a modified procedure: Stroop tasks would be administered both before password creation and afterwards, allowing the experimenter to capture the initial effort levels and effectively measure depletion.

Finally, we entertain the possibility that the Stroop task is an ineffective measure of ego depletion and/or a bad proxy for cognitive effort.

4.3 Limitations

Several limitations potentially affect the outcomes of our study. Some have already been discussed; for example, the incentive levels may have been too low to motivate our subjects. To account for this, future work could replicate the conditions with larger incentives.

Other limitations stem from the nature of the study as an online experiment. We do not know if our records of time spent on the task, and its subcomponents, are accurate, because participants may have temporarily switched away to another activity in the middle. However, clusters of data do suggest that most stayed focused on the task.

A significant concern with this experiment is ecological validity: how well do its findings translate into the real world? The demographic pool is more diverse than the typical one drawn from college students; however, it is still not representative of the world (or even U.S. and India's) population at large. Real-world passwords are usually created to protect personal information of some sort, while passwords in this study had nothing of value associated with them. Finally, workers on Mechanical Turk are compensated based on the number of tasks they complete and therefore try to complete as many as possible back-to-back. They may therefore be more sensitive to time costs than regular users.

4.4 Applications

Several lessons can be taken away from our findings and applied to security in real-world applications.

Consistent with previous findings, a number of respondents were either unaware of the prevalent threat model or seemed more worried about people they know or those who are familiar with them. One subject confessed, "some of my passwords could probably be guessed by people who know me well," and another noted that they "used some numbers that were significant to me, but that were not birthdays or other dates that might be guessed from my personal information." Therefore, websites already committed to educating users (e.g., through strength meters) may do well do emphasize the dominant threat models of dictionary and brute-force attacks.

Survey respondents also overwhelmingly spoke to using stronger passwords with accounts they considered to be high-value, primarily those with financial information in them but also, in some cases, email accounts. Given that people do seem to take such factors into consideration, providers of such accounts may be advised to emphasize the value that they provide and the costs if they are compromised, in order to influence the calculation at the time of creation.

What do you do if you're not a bank? Since users are not naturally willing to create more secure passwords for accounts they do not consider critical, providers need to design the user experience in a way that encourages the creation of stronger passwords. For example, our findings suggest that both strength meters and forced waiting periods on the creation page can be used to that effect.

5 References

[1]: Serge Egelman, David Molnar, Nicolas Christin, Alessandro Acquisti, Cormac Herley, Shriram Krishnamurthi. 2010. Please Continue to Hold: An

Empirical Study of User Tolerance of Security Delays. In *Proceedings of the 2010 Workshop on the Economics of Information Security.*

[2]: Cormac Herley. 2009. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proceedings of the 2009 workshop on New security paradigms workshop (NSPW '09).* ACM, New York, NY, USA, 133-144.

[3] Patrick Gage Kelley, Saranga Komanduri, Michelle L. Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP '12).* IEEE Computer Society, Washington, DC, USA, 523-537.

[4]: Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle L. Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. 2012. How does your password measure up? the effect of strength meters on password creation. In *Proceedings of the 21st USENIX conference on Security symposium (Security'12).* USENIX Association, Berkeley, CA, USA, 5-5.

[5]: Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. 2009. Password Cracking Using Probabilistic Context-Free Grammars. In *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy (SP '09).* IEEE Computer Society, Washington, DC, USA, 391-405.