Towards Reliable Storage of 56-bit Secrets in Human Memory

Joseph Bonneau, ***Princeton University;*** **Stuart Schechter,** ***Microsoft Research***

https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/bonneau

This paper is included in the Proceedings of the 23rd USENIX Security Symposium.

August 20–22, 2014 • San Diego, CA

ISBN 978-1-931971-15-7

Open access to the Proceedings of the 23rd USENIX Security Symposium is sponsored by USENIX

Towards reliable storage of 56-bit secrets in human memory

Joseph Bonneau
Princeton University

Stuart Schechter
Microsoft Research

Abstract

Challenging the conventional wisdom that users cannot remember cryptographically-strong secrets, we test the hypothesis that users can learn randomly-assigned 56-bit codes (encoded as either 6 words or 12 characters) through *spaced repetition*. We asked remote research participants to perform a distractor task that required logging into a website 90 times, over up to two weeks, with a password of their choosing. After they entered their chosen password correctly we displayed a short code (4 letters or 2 words, 18.8 bits) that we required them to type. For subsequent logins we added an increasing delay prior to displaying the code, which participants could avoid by typing the code from memory. As participants learned, we added two more codes to comprise a 56.4-bit secret. Overall, 94% of participants eventually typed their entire secret from memory, learning it after a median of 36 logins. The learning component of our system added a median delay of just 6.9 s per login and a total of less than 12 minutes over an average of ten days. 88% were able to recall their codes exactly when asked at least three days later, with only 21% reporting having written their secret down. As one participant wrote with surprise, "the words are branded into my brain."

1 Introduction

Humans are incapable of securely storing high-quality cryptographic keys . . . they are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed. But they are sufficiently pervasive that we must design our protocols around their limitations.

—Kaufman, Perlman and Speciner, 2002 [54]

The dismissal of human memory by the security community reached the point of parody long ago. While assigning random passwords to users was considered standard as recently in the mid-1980s [26], the practice died out in the 90s [4] and NIST guidelines now presume all passwords are user-chosen [32]. Most banks have even given up on expecting customers to memorize random four-digits PINs [22].

We hypothesized that perceived limits on humans' ability to remember secrets are an artifact of today's systems, which provide users with a single brief opportunity during enrolment to permanently imprint a secret password into long-term memory. By contrast, modern theories of the brain posit that it is important to *forget* random information seen once, with no connection to past experience, so as to avoid being overwhelmed by the constant flow of new sensory information [10].

We hypothesized that, if we could relax time constraints under which users are expected to learn, most could memorize a randomly-assigned secret of 56 bits. To allow for this memorization period, we propose using an alternate form of authentication while learning, which may be weaker or less convenient than we would like in the long-term. For example, while learning a strong secret used to protect an enterprise account, users might be allowed to login using a user-chosen password, but only from their assigned computer on the corporate network and only for a probationary period. Or, if learning a master key for their password manager, which maintains a database of all personal credentials, users might only be allowed to upload this database to the network after learning a strong secret used to encrypt it.

By relaxing this time constraint we are able to exploit *spaced repetition*, in which information is learned through exposure separated by significant delay intervals. Spaced repetition was identified in the 19$^{\text{th}}$ century [43] and has been robustly shown to be among the most effective means of memorizing unstructured information [35, 11]. Perhaps the highest praise is its popularity amongst medical students, language learners, and others who are highly motivated to learn a large amount of vocabulary as efficiently as possible [34, 91].

To test our hypothesis, we piggybacked spaced repeti-

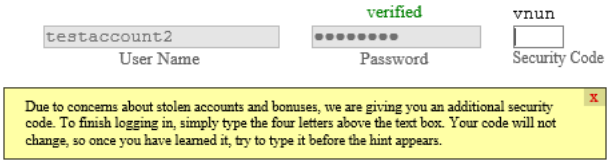

Figure 1: The login form for a user logging in for the first time, learning a code made of *letters*.

tion of a new random secret onto an existing login process utilizing a user-chosen password. Our system can be seen in action in Figure 1. After receiving a user's self-chosen password, we add a new field into which they must type a random security code, which we display directly above this field. With each login we add a $\frac{1}{3}$ second delay (up to a maximum of 10 seconds) before displaying the hint for them to copy, encouraging them to type the code from memory if possible to save time.

We recruited remote research participants to perform a study that required logging into a website 90 times over up to 15 days, which they did at an average rate of nine logins per day. We assigned each participant a random 56-bit 'security code' encoded into three chunks of either four lowercase letters or two words. After participants began to enter the first chunk before it was displayed, we added a second and likewise for the third and final chunk. We did not tell participants that learning the random secret was a goal of the research study; they simply learned it to save time. Participants experienced a median additional delay from using our system of just 6.9 s on each login, or about 11 m 53 s total over the entire study.

Three days after participants completed the initial study and had stopped using their security codes, we asked them to recall their code from memory in a follow-up survey which 88% completed. They returned after a median of 3 days 18 hours (mean 4 days 23 hours). We found that 46 of 56 (82%) assigned *letters* and 52 of 56 (93%) assigned *words* recalled their codes correctly. Only 21% reported writing down or otherwise storing the security codes outside their memory and the recall rate was actually higher amongst those who didn't.

While 56-bit secrets are usually overkill for web authentication, the most common use of passwords today, there are several compelling applications for "high value" passwords such as master passwords for password managers, passwords used to protect private keys, device-unlock passwords, and enterprise login passwords where cryptographically-strong passwords can eliminate an entire class of attack. In debunking the myth that users are inherently incapable of remembering a strong secret, we advocate that using spaced repetition to train users to remember strong secrets should be available in every security engineer's toolbox.

2 Security goals

Evaluating the difficulty of guessing user-chosen passwords is messy [56] and security engineers are left with few hard guarantees beyond empirical estimates of min-entropy, which can be as low as 10 bits or fewer [18]. By contrast, with random passwords we can easily provide strong bounds of the difficulty of *guessing*, if not other attack vectors against passwords [20].

2.1 The cost of brute-force

Random passwords are primarily a defense against an *offline attack* (eq. *brute-force attack*), in which the attacker is capable of trying as many guesses as they can afford to check computationally. We can estimate the cost of brute-force by observing the Bitcoin network [3], which utilizes proof-of-work with SHA-256 to maintain integrity of its transaction ledger and hence provides direct monetary rewards for efficient brute force. While SHA-256 is just one example of a secure hash function, it provides a reasonable benchmark.

In 2013, Bitcoin miners collectively performed $\approx 2^{75}$ SHA-256 hashes in exchange for bitcoin rewards worth $\approx$ US$257M. This provides only a rough estimate as Bitcoin's price has fluctuated and Bitcoin miners may have profited from carrying significant exchange-rate risk or utilizing stolen electricity. Still, this is the only publicly-known operation performing in excess of 2^{64} cryptographic operations and hence provides the best estimate available. Even assuming a centralized effort could be an order of magnitude more efficient, this still leaves us with an estimate of US$1M to perform a 2^{70} SHA-256 evaluations and around US$1B for 2^{80} evaluations.

In most scenarios, we can gain equivalent security with a smaller secret by *key stretching*, deliberately making the verification function computationally expensive for both the attacker and legitimate users [66, 57]. Classically, this takes the form of an iterated hash function, though there are more advanced techniques such as memory-bound hashes like `scrypt` [69] or halting password puzzles which run forever on incorrect guesses and require costly backtracking [25].

With simple iterated password hashing, a modern CPU can compute a hash function like SHA-256 at around 10 MHz [1] (10 million SHA-256 computations per second), meaning that if we slow down legitimate users by ≈ 2 ms we can add 14 bits to the effective strength of a password, and we can add 24 bits at a cost of ≈ 2 s. While brute-forcing speed will increase as hardware improves [38], the same advances enable defenders to continuously increase [72] the amount of stretching in use at constant real-world cost [19], meaning these basic numbers should persist indefinitely.

2.2 Practical attack scenarios

Given the above constraints, we consider a 56-bit random password a reasonable target for most practical scenarios, pushing the attacker cost around US$1M with 14 bits (around 2 ms) of stretching, or US$1B with 24 bits (around 2 s) of stretching. Defending against offline attacks remains useful in several scenarios.

Password managers are a compelling aid to the difficulty of remembering many passwords online, but they reduce security for all of a user's credentials to the strength of a master password used to encrypt them at rest. In at least one instance, a password management service suffered a breach of the systems used to store users' data [63]. Given that password managers only need to decrypt the credentials at startup, several seconds of stretching may be acceptable.

Similarly, when creating a public/private key pair for personal communication, users today typically use a password to encrypt the private key file to guard against theft. Given a sufficiently strong random password, users could use their password and a unique public salt (e.g., an email address) to seed a random number generator and create the keys. The private key could then simply be rederived when needed from the password, preventing the need for storing the private key at all. This application also likely tolerates extensive stretching.

Passwords used to unlock personal devices (e.g. smartphones) are becoming increasingly critical as these devices are often a second factor (or sole factor) in authentication to many other services. Today, most devices use relatively weak secrets and rely on tamper-proof hardware to limit the number of guesses if a device is stolen. Strong passwords could be used to remove trust in device hardware. This is a more challenging application, however. The budget for key-stretching may be 14 bits or fewer, due to the frequency with which users authenticate and the limited CPU and battery resources available. Additionally, entering strong passwords quickly on a small touchscreen may be prohibitive.

Finally, when authenticating users remotely, such as logging into an enterprise network, security requirements may motivate transitioning from user-chosen secrets to strong random ones. Defending against *online guessing*, in which the attacker must verify password guesses using the genuine login server as an oracle, can be done with far smaller random passwords. Even without explicit rate-limiting, attacking a 40-bit secret online would generate significantly more traffic than any practical system routinely handles. 40-bit random passwords would ensure defense-in-depth against failures in rate-limiting.

Alternately, attackers may perform an offline attack if a remote authentication server is breached. In general, we would favor back-end defenses against password database compromises which don't place an additional burden on users—such as hashing passwords with a key kept in special-purpose hardware, dividing information up amongst multiple servers [52] or one limited-bandwidth server [41]. Random passwords would also frustrate brute-force in this scenario, although the opportunity for key-stretching is probably closer to the 2 ms (14 bit) range to limit load on the login server.

3 Design

Given our estimation that a 56-bit secret can provide acceptable security against feasible brute-force attacks given a strong hash function and reasonable key stretching, our goal was to design a simple prototype interface that could train users to learn 56 bits secret with as little burden as possible.

Spaced repetition [43, 70, 62] typically employs delays (spacings) of increasing length between rehearsals of the chunk of information to be memorized. While precisely controlling rehearsal spacing makes sense in applications where education is users' primary goal, we did not want to interrupt users from their work. Instead, we chose to piggyback learning on top of an already-existing interruption in users' work-flow—the login process itself. We allow users to employ a user-chosen password for login, then piggyback learning of our assigned secret at the end of the login step. We split the 56-bit secret up into three equal-sized chunks to be learned sequentially, to enable a gradual presentation and make it as easy as possible for users to start typing from memory.

3.1 Encoding the secret

Although previous studies have found no significant differences in user's ability to memorize a secret encoded as words or letters [77, 64], we implemented both encodings. For *letters*, we used a string of 12 lowercase letters chosen uniformly at random from the English alphabet to encode a $26^{12} \approx 56.4$ bit secret. The three chunks of the secret were 4 letters each (representing ≈ 18.8 bits each).

For *words*, we chose a sequence of 6 short, common English words. To keep security identical to that of the *letters* case, we created our own list of 676 (26^2) possible words such that 6 words chosen uniformly at random would encode a $676^6 = 26^{12} \approx 56.4$ bit secret. We extracted all 3–5 English nouns, verbs and adjectives (which users tend to prefer in passwords [24, 85]) from Wiktionary, excluding those marked as vulgar or slang words and plural nouns. We also manually filtered out potentially insulting or negative words. From these candidate words we then greedily built our dictionary of 676 words by repeatedly choosing the most common remaining word, ranked by frequency in the Google N-gram

web corpus [27]. After choosing each word we then removed all words within an edit distance of two from the remaining set of candidates to potentially allow limited typo correction. We also excluded words which were a complete prefix of any other word, to potentially allow auto-complete. We present the complete list in Table 3.

3.2 Login form and hinting

Unlike typical login forms, we do not present a button to complete sign-in, but rather automatically submit the password for verification via AJAX each time a character is typed. Above the password field we display the word "verifying" while awaiting a response and "not yet correct" while the current text is not the correct password.

After the user's self-chosen password is verified, a text box for entering the first chunk of the user's assigned code appears to the right of the password field, as we show in Figure 1. On the first login, we display the correct value of the chunk immediately above the field into which users must enter it. In the version used for our study, we included a pop-up introducing the security code and its purpose:

> Due to concerns about stolen accounts and bonuses, we are giving you an additional security code. To finish logging in, simply type the [four letters | two words] above the text box. Your code will not change, so once you have learned it, try to type it before the hint appears.

We color each character a user enters into the security code field green if it is correct and red if incorrect. We replace correct characters with a green circle after 250 ms.

With each consecutive login, we delay the appearance of the hint by $\frac{1}{3}$ of a second for each time the user has previously seen the chunk, up to a maximum of 10 seconds. If the user types a character correctly before the delay expires, we start the delay countdown again. We selected these delay values with the goal of imposing the minimal annoyance necessary to nudge users to start typing from memory.

After a user enters a chunk without seeing the hint on three consecutive logins, we add another chunk. In the version used in our study, we show a pop-up which can be dismissed for all future logins:

> Congratulations! You have learned the first [four letters | two words] of your security code. We have added another [four letters | two words]. Just like the first [four letters | two words], once you have learned them, you can type them without waiting for the hint to appear.

When we detect that a user has finished typing the first chunk of their security code, we automatically tab (moved the cursor) to the text field for the second chunk and then start the delay for that chunk's hint. After typing the second chunk correctly from memory three times

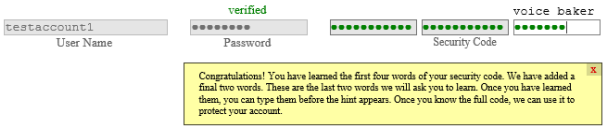

Figure 2: The login form for a user in who has just received the third security code chunk *words*.

in a row, we add the third and final chunk. In the version used in the study, we also displayed one more pop-up:

> Congratulations! You have learned the first [eight letters | four words] of your security code. We have added a final [four letters | two words]. These are the last [four letters | two words] we will ask you to learn. Once you have learned them, you can type them before the hint appears. Once you know the full code, we can use it to protect your account.

We illustrate the login process from our study, using all three chunks, in Figure 2. In a real deployment, once the user is consistently typing the entire security code from memory, entering their self-chosen password would no longer be necessary.

We disable pasting and automatic form-filling for the security code field to encourage users to type from memory. We allow users to type their code in lower or upper case, with all non-letter characters being ignored, including spaces between words as no word is a prefix of any other word. During training we automatically insert a space at the end of any entered code word so users learn that they do not need to type the spaces.

4 Experimental Methodology

We used a remote online study to evaluate our system. To keep participants from realizing the purpose of our study was the security codes and potentially altering their behavior, we presented our study as a psychology study with the security codes a routine part of logging in to participate. We recruited participants using Amazon's Mechanical Turk (MTurk) platform [59] and paid them to participate, which required logging in 90 times in 15 days. For completeness, we provide exact study materials the extended version of this paper [23].

4.1 The distractor task

We intended our distractor task to provide a plausible object of study that would lead us to ask participants to log in to our website repeatedly (distracting participants from the subject of our investigation) and to require a non-trivial mental effort (distracting them from making conscious efforts to memorize their security codes). Yet we also wanted the distractor task to be relatively fast,

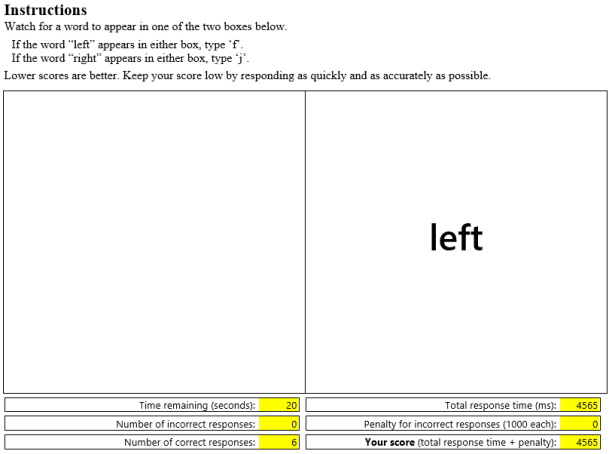

Figure 3: The Attention Game, our distractor task

interesting, and challenging, since we were asking participants to perform a large number of logins.

We designed a game to resemble descendants of the classic psychological study that revealed the Stroop effect [79]. Our game measured participants' ability to ignore where a word appeared (the left or right side of their screen) and respond to the meaning of the word itself. Each 60-second game consisted of 10 trials during which either the word 'left' or 'right' would appear in one of two squares on the screen, as illustrated in Figure 3. The words appeared in a random square after a random delay of 2–4 seconds, after which participants were asked to immediately press the `f` key upon seeing the word 'left' or `j` key upon seeing the word 'right' (corresponding to the left and right sides of a QWERTY keyboard). During the game, participants saw a score based on their reaction time, with penalties for pressing the wrong key.

4.2 Treatments

We randomly assigned participants to three treatments: *letters* (40% of participants), *words* (40%), and *control* (20%). Participants in the *letters* and *words* treatments received security codes consisting of letters and words, respectively, as described in Section 3.1. Participants in the *control* treatment received no security code at all and saw a simple password form for all logins; we included this treatment primarily to gauge whether the additional security codes were causing participants to drop out of the experiment more than traditional authentication would have.

4.3 Recruiting

We recruited participants to our study using Amazon's Mechanical Turk by posting a Human Intelligence Task (HIT) titled "60-Second Attention Study", paying US$0.40, and requiring no login. When participants completed the game, we presented them with an offer to "Earn $19 by being part of our extended study" (a screenshot of the offer is in the extended version of this paper [23]). The offer stated that participants would be required to play the game again 90 times within 15 days, answer two short questions before playing the game, wait 30 minutes after each game before starting a new game session, and that they would have to login for each session. We warned participants that those who joined the extended study but did not complete it would not receive partial payment. Our study prominently listed Microsoft Research as the institution responsible for the study. As we did not want to place an undue burden on workers who were not interested in even reading our offer, we provided a link with a large boldface heading titled "Get paid now for your participation in this short experiment" allowing participants to be paid immediately without accepting, or even reading, our offer.

When workers who had performed the single-game HIT signed up to participate in our 90-game attention study, we presented them with a sign-up page displaying our approved consent form and asking them to choose a username and a password of at least six characters. For the 88 logins following signup (games 2–89), and for login to the final session (in which we did not show the game but instead showed the final survey), we required participants to login using the chosen password and security code (if assigned a non-*control* treatment).

Amazon's policies forbid HITs that require workers to sign up for accounts on websites or to provide their email addresses. These rules prevent a number of abusive uses of Mechanical Turk. They also protect Amazon's business by forbidding requesters from recruiting workers, establishing a means of contact that bypasses Amazon, and then paying hired workers for future tasks without paying Amazon for its role in recruiting the workers. Our HIT was compliant with the letter of these rules because we only required workers to play the attention game, and they were in no way obligated to sign up for the full attention study. We were also compliant with the spirit of the rules, as we were not asking workers to engage in abusive actions and we did not cut Amazon out of their role as market maker—we paid participants for the 90-game attention study by posting a bonus for the HIT they already completed through Amazon.

As in any two-week project, some participants requested extensions to the completion deadline in order to reach 90 completed game. We provided 24-hour extensions to participants who were within 20 games of completing the study at the deadline.

How long has it been since you last slept for at least one hour without interruption?

Please indicate if you have consumed any of the following within the last 60 minutes:

- ☐ Food
- ☐ Beverages
- ☐ Caffeinated substances such as coffee or soft drinks
- ☐ Energy drinks other than caffeine

Figure 4: Participants were asked to fill out this two-question survey before every attention game.

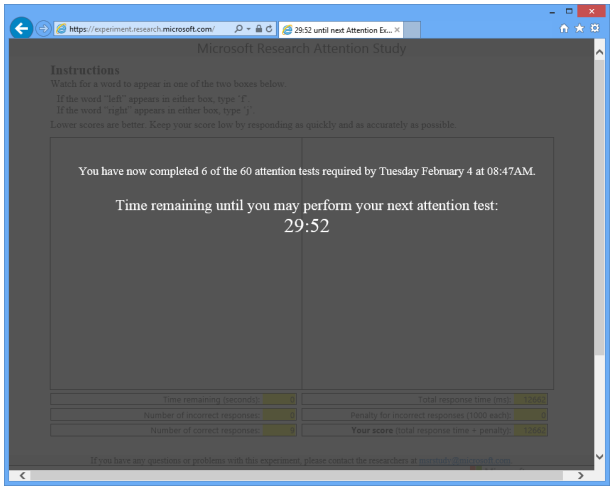

Figure 5: After completing an attention test, participants could not perform another one for 30 minutes.

4.4 Sessions

After each login we presented a very short survey (shown in Figure 4) asking participants about their recent sleep and eating. This was designed solely to support the purported goal of the study and we ignored the responses.

After participants completed the survey we immediately displayed the "Attention Game". When they completed the game, we overlaid a timer on top of the page counting down 30 minutes until they could again fill out the survey and play the game (see Figure 5). The timer also counted down in the title of the page, so that participants would see the countdown when browsing in other tabs and know when they were next allowed to play. If participants tried to log into the website again before the 30-minute waiting period was complete, we displayed the countdown again, starting from the amount of time remaining since they last completed the game.

4.5 Completion survey

When participants logged in for the 90th and final time, we skipped the game and displayed our completion survey. We provide the full text of the survey (with participants' answer counts) in the extended version of this paper [23]. We started the survey with demographic questions and then asked participants if they had written down or stored their passwords or assigned security codes outside of their memory.

We then debriefed participants about the true nature of the study, explaining that the security code was the focus of the study, though we did not reveal that we planned a follow-up study. We could not defer the debriefing to the follow-up study, as participants had not committed to engage with us beyond the end of the study and might not accept invitations for future contact. Indeed, some participants reported discussing the study in forums, but as we had entrusted those who finished the study with the truth, they returned that trust by respecting forum rules against 'spoilers' in all cases we are aware of.

To aid with the follow-up study, we asked participants to provide their email address, stating the question in a manner that we hoped would minimize suspicion that a formal follow-up study was imminent.

> If our analysis raises more questions about your experience during the study, may we contact you and ask you to answer a few questions in return for an additional bonus? If so, provide your email below. (This is optional!)

4.6 Payment

We paid $20 to participants who completed the study, as opposed to the $19 promised, to show extra gratitude for their participation. We informed participants of this only *after* they had completed the 'attention' study and filled out their post-deception ethics questionnaire, so as to not taint their responses about the ethics of the deception. However, this payment came well before the invitation to the follow-up study. Receiving a payment exceeding what we had promised may have increased participants' receptiveness to that invitation.

Despite telling participants they would not be paid unless they completed the study, we did pay $0.20 per login to all participants who logged into the site at least once after signing up. We did so because we couldn't be certain that the extra work of entering a security code didn't cause some participants to drop out. We wanted to ensure that if participants found the security code so arduous as to quit, they would not lose out on payment for the attention tests they did complete. We did not reveal this fact to the participants who completed the study and filled out the ethics survey as we feared they might communicate it to those who had yet to finish.

4.7 Follow-ups

At least 72 hours after a non-control group participant completed the study, we emailed them an invitation to perform an additional HIT for $1 (this email is reproduced in the extended version of this paper [23]). Most

participants provided an email address in the final survey of the attention study; we tried to contact those who didn't via Mechanical Turk. When participants accepted the HIT, we identified them by their Mechanical Turk ID to verify that they'd participated in the main study.[1]

The follow-up study contained only one question:

> Please try to recall and enter the security code that we assigned you during the attention study.
> **If you stored or wrote down your security code, please do not look it up.** We are only interested in knowing what you can recall using *only your memory*. It's OK if you don't remember some or all of it. Just do the best you can.

We presented participants with three text fields for the three chunks of their security code. Unlike the data-entry field used when they logged in for the attention experiment, we used plain text fields without any guidance as to whether the characters typed were correct. We accepted all responses from participants that arrived within two weeks of their completion of the study.

We emailed all participants who completed the first follow-up again 14 days after they completed it with the offer to complete a second identical follow-up for an additional $1 reward.

4.8 Ethics

The experiment was performed by Microsoft Research and was reviewed and approved by the organizations's ethics review process prior to the start of our first pilot.[2]

We employed deception to mask the focus of our research out of concern that participants might work harder to memorize a code if they knew it to be the focus of our study. We took a number of steps to minimize the potential for our deception to cause harm. We provided participants with estimates for the amount of time to complete the study padded to include the unanticipated time to enter the security code. While we told participants they would not be paid if they did not complete the study, we did make partial payments. We monitored how participants responded to the deception, investigating the responses of pilot participants before proceeding with the full study and continued to monitor participants in the full study, using a standard post-deception survey hosted by the Ethical Research Project [82]. We also offered participants the opportunity to withdraw their consent for use data derived from their participants. The vast majority of participants had no objection to the deception and none asked to have their data withdrawn. We provide more detail on participants' ethics responses in the extended version of this paper [23].

[1] We failed to verify that it had been three days since they completed the study, requiring us to disqualify three participants who discovered the follow-up study prematurely (see Section 5.1).

[2] The first author started a position at Princeton after the research was underway. He was not involved in the execution of the study or communications with participants. He did not have access to the email addresses of those participants who volunteered to provide them (the only personally-identifiable information collected).

5 Results

We present overall analysis of the most important results from our study: participant's ability to learn and recall security codes. We present a full accounting of participants' responses to the multiple-choice questions of our final survey and the complete text of that survey in the extended version of this paper [23], including demographics which reflect the typical MTurk population [74].

5.1 Recruitment and completion

We offered our initial attention-game task to roughly 300 workers from February 3–5, 2014. 251 workers accepted the offer to participate in our study by completing the sign-up page and playing the first game. We stopped inviting new participants when we had reached roughly 100 sign-ups for our two experimental groups. Participants' assigned treatment had no effect until they returned after sign-up and correctly entered their username and chosen password into the login page, so we discard the 28 who signed up but never returned. We categorize the 223 participants who did return in Table 1.

5.1.1 Dropouts

Inserting a security-code learning step into the login process creates an added burden for participants. Of participants who completed the study, typing (and waiting for) the security codes added a median delay of 6.9 s per login. To measure the impact of this burden, we tested the hypothesis that participants assigned a security code would be less likely to complete the experiment than those in the *control*. The null hypothesis is that group assignment has no impact on the rate of completion.

Indeed, the study-completion rates in the fourth row of Table 1 are higher for *control* than the experimental groups. We use a two-tailed Fisher's Exact Test to compare the proportion of participants who completed the study between those assigned a security code (the union of the *letters* and *words* treatments, or 133 of 170) to that of the *control* (35 of 41). The probability of this difference occurring by chance under the null hypothesis is $p = 0.2166$. While this is far from the threshold for statistical significance, such a test cannot be used to reject the alternate hypothesis that the observed difference reflects a real percentage of participants who dropped out due to the security code.

Digging into the data further, we can separate out those participants who abandoned the study after exactly

	Control		*Letters*		*Words*		*Total*	
Signed up for the 'attention' study	41		92		90		223	
Quit after 2 or 3 games	0/41	0%	9/92	10%	12/90	13%	21/223	9%
Otherwise failed to finish	6/41	15%	14/92	15%	12/90	13%	32/223	14%
Completed the 'attention' study	35/41	85%	69/92	75%	66/90	73%	170/223	76%
Received full security code	—		63/68	93%	64/65	98%	127/133	95%
Typed entire code from memory	—		62/63	99%	64/64	100%	126/127	99%
Participated in first follow-up	—		56/63	89%	56/64	88%	112/127	88%
Recalled code correctly	—		46/56	82%	52/56	93%	98/112	88%
Participated in second follow-up	—		52/56	93%	52/56	93%	104/112	93%
Recalled code correctly	—		29/52	56%	32/52	62%	61/104	59%

Table 1: Results summary: participants who signed up for the attention study, the fraction of those participants who completed the study, the fraction of the remaining participants who entered the first two chunks of their security code reliably enough to be shown the full security code (all three chunks), the fraction of those remaining who participated in the follow-up studies (after 3 and 17 days, respectively), and the fraction of those who recalled their security code correctly. The *control* group did not receive security codes and hence are excluded from the latter rows of the table.

two or three games from those who failed to finish later (no participant quit after the fourth or fifth games). While no participant in the *control* quit between two or three games, 9 participants assigned to *letters* and 12 assigned to *words* did. For participants who completed more than three games, the rate of failure to finish the study is remarkably consistent between groups. We do not perform statistical tests as this threshold is data-derived and any hypothesis based on it would be post-hoc. Rather, as our study otherwise presents a overall favorable view of random assigned secrets, we present the data in this way as it illustrates to the reader reason for skepticism regarding user acceptance among unmotivated participants.

5.1.2 Participants who appeared not to learn

Six participants completed the study without receiving all three chunks of their security codes, having failed to demonstrate learning by typing the first chunk (one participant from *letters*) or second chunk (five participants, four from *letters* and one from *words*) before the hint appeared. After the conclusion of the study we offered participants $1 to provide insights into what had happened and all replied. Two in the *letters* group, including the one who only received one chunk, reported genuine difficulty with memory. The other four stated quite explicitly (which we provide in the extended version of this paper [23]) that they purposely avoided revealing that they had learned the second chunk to avoid being assigned more to learn.

5.1.3 Excluded participants

We found it necessary to exclude four participants from some of our analysis. Three participants, two in *words* and one in *letters*, discovered and accepted the follow-up HIT before three days had passed since the end of the study, ignoring the admonition not to accept this HIT without an invitation. Though these participants all completed the 90-game attention study, learned and recalled their entire security code, we count them as having not returned for the follow-up. We corrected this bug prior to the second follow-up. We disqualified one additional 'participant' in the *letters* group which appeared to be using an automated script.

After revealing the deceptive nature of the study we gave participants the option to withdraw their consent for us to use our observations of their behavior, while still receiving full payment. Fortunately, none chose to do so.

5.2 Learning rates

Of non-*control* participants completing the study, 93% eventually learned their full security code well enough to type it from memory three times in a row (91% of *letters* and 96% of *words*). Most participants learned their security codes early in the study, after a median of 36 logins (37 for *letters* and 33 of *words*). We show the cumulative distribution of when participants memorized each chunk of their code in Figure 6.

We consider whether participants first typed their codes from memory in fewer logins with either *letters* or *words*, with the null hypothesis that encoding had no impact on this measure of learning speed. A two-tailed Mann-Whitney U (rank sum) test on the distribution of these two sets of learning speeds estimates a probability of $p = 0.07$ ($U = 1616$) of observing this difference by chance, preventing us from rejecting the null hypothesis.

We had hypothesized that, with each subsequent chunk we asked participants to memorize, their learn-

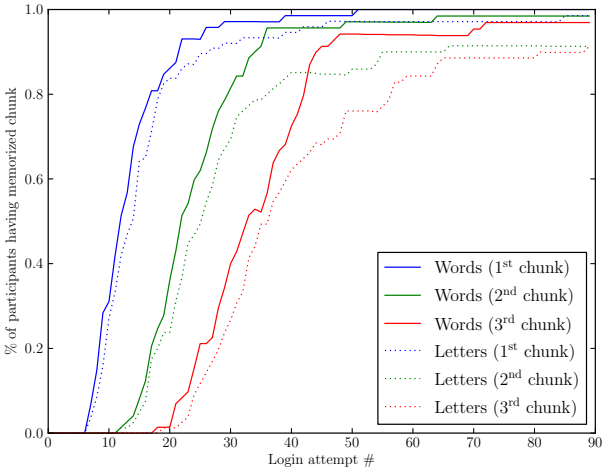

Figure 6: We show the proportion of participants who had memorized each chunk of their security code after a given number of login attempts. We considered a participant to have memorized a chunk after they entered it without a hint in three consecutive logins.

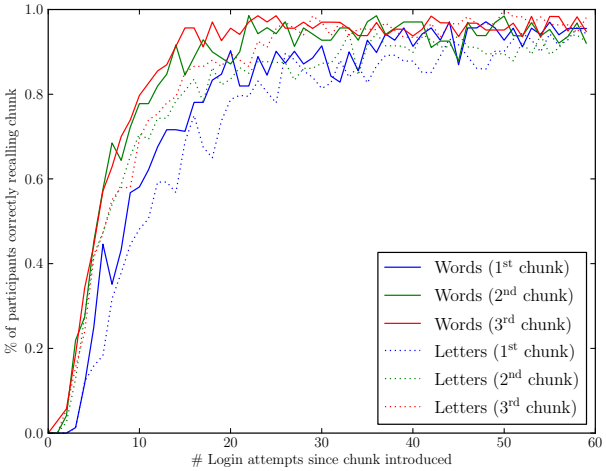

Figure 7: For each of the three chunks in participants' security codes, we show the proportion of participants who entered each chunk without a hint as a function of the number of previous exposures to the chunk (the number of previous logins in which the chunk appeared). On the whole, participants actually memorized their second and third chunks more quickly than the first.

ing speed might decrease due to interference [31] or increase due to familiarity with the system. Learning times decreased. We use a Mann-Whitney U test to compare learning times between the first and final chunks, using times only for participants who learned all three, yielding a significant $p < 0.001$ ($U = 4717$). To remove the impact of the time required to notice the delay and learn that they could enter the code before it appeared, we compare the learning times between the third and second chunks. This difference is much smaller, with a Mann-Whitney U test yielding a probability of $p = 0.39$ ($U = 7646$) of an effect due to chance.

To illustrate the increasing rate of learning we show, in Figure 7, the percent of participants who typed each chunk correctly from memory as a function of the number of previous exposures to that chunk.

5.3 Login speed and errors

Overall, participants in the *words* group took a median time of 7.7 s to enter their security codes, including waiting for any hints to appear that they needed, and participants in the *letters* group took a median time of 6.0 s. Restricting our analysis to those logins in which participants were required to enter all three chunks of the code only increases the median login time to 8.2 s for *words* and 6.1 s for *letters*.[3] The distribution had a relatively long tail, however, with the 95th percentile of logins taking 23.6 s for *words* and 20.5 s for *letters*.

[3]The median login time actually went down for *letters* participants when all three chunks were required, likely because this included more logins typed exclusively from memory with no waiting for a hint.

After computing the median login time for each participant, we compared the set of these values for participants in the two experimental groups using a Mann-Whitney U. We can reject the null hypothesis that the differences between these medians were the result of chance with $p < 0.01$ ($U = 1452$) and conclude that participants in the *letters* group were significantly faster.

Errors in entering security codes (whether typos or genuine memory errors) were relatively rare: over all 90 logins participants in the *words* group made fewer errors (with a median of 5) than participants in the *letters* group (median 7). Using a Mann-Whitney U, we cannot reject the null hypothesis that neither group would make more errors than the other ($p = 0.08$ ($U = 1706$)).

5.4 Recall of security codes in follow-ups

We sent invitations to participants to follow-up studies testing recall of their security codes 3 days after the initial study ended and then 14 more days after they completed the first follow-up. The median time between when participants completed the study and actually took the first follow-up study was 3 days 18 hours (mean 4 days 23 hours). For the second follow-up study the median time was 16 days 0 hours (mean 16 days 13 hours). By comparison, the median time to complete the study itself was 10 days 5 hours (mean 9 days 19 hours).

Overall, 88% of participants recalled their code correctly in the first follow-up and 59% did so in the second. The drop-off at the second follow-up was expected

as memory is believed to decay exponentially with the delay since the information was last recalled [89].

We had hypothesized that participants in the *letters* treatment might be more or less likely to recall their security codes correctly in the follow-ups than participants in the *words* treatment. As seen in Table 1, of participants in the *letters* group 82% recalled their security codes correctly in the first follow-up and 56% did so in the second study, compared to 93% and 62%, respectively, of users in *words*. Using a two-tailed Fisher's Exact Test, we cannot rule out the null hypothesis that participants in either group were equally likely to recall codes correctly, with the observed differences occurring with a $p = 0.15$ chance in the first follow-up and $p = 0.45$ in the second follow-up under the null hypothesis.

5.4.1 Types of errors

We observed 14 participants incorrectly entering their code in the first follow-up and 52 in the second. All 13 users who entered incorrectly in the first follow-up and participated in the second entered their code incorrectly again. This sample is too low to draw firm conclusions about the nature of participants' recall errors, but we did see evidence that users retained partial memory, with 75% of users entering at least one component of their code correctly in the second follow-up and 48% missing only one component or entering components in the wrong order. Re-arranging the order of components, which accounted for 10% of errors, could be corrected by accepting components in any order at a loss of only $\log_2(3!) \approx 2.6$ bits of security. Unfortunately, the majority of other errors could not be corrected without significantly downgrading security. Only 3 participants (6%) in the second-followup (and 2 in the first) entered a code within an edit distance of 2 of the correct code. We present further information on the types of errors observed in the extended version of this paper [23].

5.4.2 Storing security codes

A minority of participants reported storing their security code outside of their memory, as presented in Table 2. We were concerned that participants who had stored their security codes might have been tempted to look them up and thereby inflated the recall rate during the follow-up. However, only 82% of participants storing their security code recalled it correctly on follow-up, whereas 89% of participants not storing the security code did. While it's possible that participants who did not rely on a stored code were better able to remember as a result, we had not hypothesized this in advance nor would the differences we observed have been statistically significant.

We had hypothesized that participants might be more likely to write down or otherwise store codes outside their memory if assigned a code composed of letters as opposed to words, or vice versa. The null hypothesis is that treatment has no impact on the choice to store codes. In the completion survey, 18 of the 69 participants in the *letters* treatment reported having stored their security code, as compared to 10 of the 66 in the *words* treatment. We use a two-sided Fisher's Exact Test to estimate that such a difference would occur with probability $p = 0.14$ under the null hypothesis. Thus we can not conclude that either treatment made participants more likely to write their code down.

6 Limitations

Whenever testing a new approach to security, its novelty alone may be enough to reveal to research participants that it is the focus of the study. Despite our best efforts, of the 133 participants in the experimental groups who completed the study (68 in *letters* and 65 in *words*), only 35 (26%, 24 from *letters* and 11 from *words*) reported that they did not suspect that the security code might be the focus of the study. The majority, 70 (53%, 28 from *letters* and 42 from *words*) reported having some suspicion and 28 (21%, 16 from *letters* and 12 from *words*) reported being 'certain' the security code was the focus of the study. Still, to our knowledge no participants revealed any 'spoilers' on public forums. Participants who suspected we were studying their ability to learn the security code may have tried harder to memorize the code than if they had not, though it's not clear how their effort would compare to that of a real-world user relying on a randomly-assigned code to secure something valuable.

7 Background and related work

7.1 Physiological principles of memory

Human memory has been studied extensively by psychologists (as well as neuroscientists and others). The *spacing effect* describes how people are better able to recall information if it is presented for the same duration, but in intervals spaced over a longer period of time. This effect was first described in the 19$^{\text{th}}$ century [43] and is considered one of the most robust memory effects [10]. It has even been demonstrated in animals. The effect is almost always far more powerful than variations in memory between individual people [35].

The cause of the spacing effect is still under debate, but most theories are based around the *multi-store model* of memory [33] in which short-term (or working memory) and long-term memory are distinct neurological processes [8, 9]. One theory of the spacing effect posits

	Did you store any part of the additional security code for the study website, such as by writing it down, emailing it to yourself, or adding it to a password manager? 'Yes'				'No'			
	Letters		*Words*		*Letters*		*Words*	
Completed the study	18/68	26%	10/65	15%	50/68	74%	55/65	85%
Reported storing password	11/18	61%	6/10	60%	2/50	4%	0/55	0%
Received full security code	16/18	89%	9/10	90%	47/50	94%	55/55	100%
Participated in follow-up	14/16	88%	8/9	89%	42/47	89%	48/55	87%
Recalled code correctly	12/14	86%	6/8	75%	34/42	81%	46/48	96%

Table 2: A minority of participants reported storing their security code outside of their memory. Each row corresponds to an identically-named row in Table 1, separated by participants' response to the code storage question in each column. The first row shows the fraction of all participants who completed the study in each group, and each subsequent row as a fraction of the one above, except for the italicized row which identifies participants who reported storing their self-chosen password (which was much more common amongst participants who stored their security code).

that when information is presented which has left short-term memory, a trace of it is recognized from long-term memory [47] and hence stimulated, strengthening the long-term memory through *long-term potentiation* [14] of neural synapses. Thus, massed presentation of information is less effective at forming long-term memories because the information is recognized from working memory as it is presented. In our case, the natural spacing between password logins is almost certainly long enough for the password to have left working memory.

Early work on spaced learning focused on *expanding presentation* in which an exponentially increasing interval between presentations was considered optimal [70, 62]. More recent reviews have suggested that the precise spacing between presentations is not particularly important [11] or that even spacing may actually be superior [53]. This is fortunate for our purposes as password logins are likely to be relatively evenly spaced in time. Other work has focused on dynamically changing spacing using feedback from the learner such as speed and accuracy of recall [68] which could potentially guide artificial rehearsal of passwords.

7.2 Approaches to random passwords

Many proposals have aimed to produce random passwords which are easier for humans to memorize, implicitly invoking several principles of human memory. Early proposals typically focused on *pronounceable* random passwords [46, 90] in which strings were produced randomly but with an English-like distribution of letters or phonemes. This was the basis for NIST's APG standard [2], though that specific scheme was later shown to be weak [45]. The independently-designed `pwgen` command for generating pronounceable passwords is still distributed with many Unix systems [5].

Generating a random set of words from a dictionary, as we did in our *words* treatment, is also a classic approach, now immortalized by the web comic XKCD [67]. This was first proposed by Kurzban [61] with a very small 100 word dictionary, the popular Diceware project [6] offers 4,000 word dictionaries. Little available research exists on what size and composition of dictionaries is optimal.

Finally, a number of proposals have aimed to enhance memorability of a random string by offering a secondary coding such as a set of images [58], a grammatical sentence [7, 50], or a song [65]. Brown's passmaze protocol was recognition-based, with users simply recognizing words in a grid [29]. None of these proposals has received extensive published usability studies.

7.3 Studies on password recall

A number of studies have examined user performance in recalling passwords under various conditions. These studies often involve users choosing or being assigned a new password in an explicitly experimental setting, and testing the percentage of users who can correctly recall their password later. Surprisingly, a large number of studies have failed to find any statistically significant impact on users' ability to recall passwords chosen under a variety of experimental treatments, including varying length and composition requirements [95, 71, 92, 86, 60] or requiring sentence-length passphrases [55].[4] The consistent lack of impact of password structure on recall rates across studies appears to have gone unremarked in any of the individual studies.

However, several studies have found that stricter composition requirements increase the number of users writing their passwords down [71, 60] and users self-report that they believe passwords are harder to remember when created under stricter password policies [60, 92].

[4] Keith et al. [55] did observe far more typos with sentence-length passwords, which needed correcting to isolate the effective recall rates.

At least three studies have concluded that users *are* more likely to remember passwords they use with greater frequency [95, 28, 42]. This suggests that lack of adequate training may in fact be the main bottleneck to password memorization, rather than the inherent complexity of passwords themselves. Brostoff [28] appears to have made the only study of password *automacity* (the ability to enter a password without consciously thinking about it), and estimated that for most users, this property emerges for passwords they type at least once per day.

A few studies have directly compared recall rates of user-generated passwords to assigned passwords. Interestingly, none has been able to conclude that users were less likely to remember assigned passwords. For example, in a 1990 study by Zviran and Haga [94] in which users were asked to generate a password and then recall it 3 months later, recall was below 50% for all unprompted text passwords and no worse for system-assigned random passwords, though the rate of writing increased. A similar lab study by Bunnell et al. found a negligibly smaller difference in recall rate for random passwords [30]. A 2000 study by Yan et al. [92] found that users assigned random passwords for real, frequently-used accounts actually requested fewer password resets than users choosing their own passwords, though those users were also encouraged to write their passwords down "until they had memorized them." Stobert in 2011 [78] found no significant difference in recall between assigned and user-chosen text passwords.

Two studies have exclusively compared user's ability to recall random passwords under different encodings. The results of both were inconclusive, with no significant difference in recall rate between users given random alphanumeric strings, random pronounceable strings or randomly generated passphrases at a comparable security level of 30 [77] or 38 bits [64]. The results appear robust to significant changes in the word dictionary used for passwords or the letters used in random strings. However, users stated that alphanumeric strings seemed harder to memorize than random passphrases [77].

All of these studies except that of Yan et al. face validity concerns as the passwords were explicitly created for a study of password security. A 2013 study by Fahl et al. [44] compared user behavior in such scenarios and found that a non-trivial proportion of users behave significantly differently in explicit password studies by choosing deliberately weak passwords, while a large number of users re-use real passwords in laboratory studies. Both behaviors bias text passwords to appear more memorable, as deliberately weak passwords may be easy to memorize and existing passwords may already be memorized. Also of concern, all of these studies (again excluding Yan et al.) involved a single enrollment process followed by recall test, with no opportunity for learning.

Spaced repetition for passwords was recently suggested by Blocki et al. [16], who proposed designing password schemes which insert a minimal number of artificial rehearsals to maintain security. After our study, Blocki published results from a preliminary study on mnemonic passwords with formal rehearsals [15]. Compared to our study, participants performed a much lower number of rehearsals spaced (about 10) spaced over a longer period (up to 64 days), prompted by the system at specific times rather than at the participant's convenience. Unlike our study participants were aware that memorization was the explicit goal of the study. Blocki also incorporated additional mnemonic techniques (images and stories). This study provides evidence that spaced repetition and other techniques can be applied more aggressively for motivated users, whereas as our study demonstrates the practicality with few system changes and unmotivated users.

7.4 Alternative authentication schemes

Several approaches have been explored for exploiting properties of human memory in authentication systems. One approach is to query already-held memories using *personal knowledge question* schemes such as "what is your mother's maiden name?" though more sophisticated schemes have been proposed [93, 48] While these schemes typically enable better recall than passwords, they are vulnerable to attacks by close social relations [76], many people's answers are available in online search engines or social networks [73], and many questions are vulnerable to statistical guessing [21, 76]. An advantage of personal knowledge questions is that they represent *cued recall* with the question acting as a cue, which generally increases memory performance over *free recall*.

Graphical passwords aim to utilize humans' strong abilities to recognize visual data [13]. Some schemes employ cued recall only by asking users to recognize a secret image from a set [40, 87, 80]. Others use uncued memory by asking users to draw a secret pattern [49, 81, 12] or click a set of secret points in an image [88, 37]. These schemes are often still vulnerable to guessing attacks due to predictable user choices [39, 83, 84]. The Persuasive Cued Click-Points scheme [36] attempts to address this by forcing users to choose points within a system-assigned region, which was not found to significantly reduce recall. Still, it remains unclear exactly what level of security is provided by most graphical schemes and they generally take longer to authentication than typing a text password. They have found an important niche on mobile devices with touch screens, with current versions of Android and Windows 8 deploying graphical schemes for screen unlock.

Bojinov et al. [17] proposed the use of implicit memory for authentication, training users to type a random key sequence in rapid order using a game similar to one used in psychological research to study implicit memory formation [75]. After a 30–45 minute training period, users were tested 1–2 weeks later on the same game with their trained sequences and random sequences, with about half performing significantly better on trained sequences. Such a scheme offers the unique property that users are unaware of their secret and thus incapable of leaking it to an attacker who doesn't know the correct secret challenge to test on, providing a measure of resistance against "rubber-hose" attacks (physical torture). Without dramatic improvements however this scheme is impractical for normal personal or corporate logins due to the very long enrollment and login times and the low rate of successful authentication.

8 Open questions and future work

As this was our first exploration of spaced repetition for learning random secrets, many of our design choices were best guesses worthy of further exploration. The character set used when encoding secrets as letters, namely 26 lowercase letters, might be inferior to an expanded set such as base-32 with digits included [51]. Our choice of a dictionary of 676 words is almost surely not optimal, since we deliberately chose it for equivalence to the size of our character set. Splitting the secret into three equal-sized chunks was also simply a design heuristic, performance might be better with more or fewer chunks.

We expect spaced repetition to be a powerful enough tool for users to memorize secrets under a variety represenation formats, though the precise details may have important implications. We observed letters to be slightly faster to type and words slightly faster to learn. We also observed double the rate of forgotten codes after three days in the *letters* group and, though this difference was not statistically significant given our sample sizes and the low absolute difference, this is worthy of further study as this difference could be important in practice.

Our system can likely be improved by exploiting additional memory effects, such as dual-coding secrets by showing pictures next to each word or requiring greater depth of processing during each rehearsal. Cued recall could also be utilized by showing users random prompts (images or text) in addition to a random password.

On the downside, interference effects may be a major hindrance if users were asked to memorize multiple random passwords using a system like ours. This is worthy of further study, but suggests that random passwords should only be used for critical accounts.

Changing the login frequency may decrease or increase performance. We aimed to approximate the number of daily logins required in an enterprise environment in which users lock their screen whenever leaving their desks. In this context, the trade-offs appear reasonable if newly-enrolled users can learn a strong password after two weeks of reduced security (to the level of a user-chosen password) with about 10 minutes of aggregate time spent learning during the training period.

In contexts with far fewer logins, such as password managers or private keys which might be used once per day or less, learning might require a larger number of total logins. If a higher total number of logins are needed and they occur at a slower rate, this may lead to an unacceptable period of reduced security. In this case, security-conscious users could use rehearsals outside of authentication events. Further, if codes are used extremely infrequently after being memorized, artificial rehearsals may be desirable even after learning the secret. These are important cases to study, in particular as these are cases in which there is no good alternative defense against offline brute-force attacks.

While the learning rates of our participants did not slow down as the number of chunks they memorized increased, they might have more have trouble as the number of chunks grows further or as they have to associate different codes with different accounts. Fortunately, most users only have a small number of accounts valuable enough to require a strong random secret.

9 Conclusion

For those discouraged by the ample literature detailing the problems that can result when users and security mechanisms collide, we see hope for the human race. Most users *can* memorize strong cryptographic secrets when, using systems freed from the constraints of traditional one-time enrollment interfaces, they have the opportunity to learn over time. Our prototype system and evaluation demonstrate the brain's remarkable ability to learn and later recall random strings—a fact that surprised even participants at the conclusion of our study.

10 Acknowledgments

We thank Janice Tsai for assistance and suggestions on running an ethical experiment, Serge Egelman and David Molnar for help running our experiment on Mechanical Turk, and Arvind Narayanan, Cormac Herley, Paul van Oorschot, Bill Bolosky, Ross Anderson, Cristian Bravo-Lilo, Craig Agricola and our anonymous peer reviewers for many helpful suggestions in presenting our results.

References

[1] HashCat project. `http://hashcat.net/hashcat/`.

[2] "Automated Password Generator (APG)". *NIST Federal Information Processing Standards Publication* (1993).

[3] Bitcoin currency statistics. `blockchain.info/stats`, 2014.

[4] ADAMS, A., SASSE, M. A., AND LUNT, P. Making passwords secure and usable. In *People and Computers XII*. Springer London, 1997, pp. 1–19.

[5] ALLBERY, B. pwgen—random but pronounceable password generator. *USENET posting in comp.sources.misc* (1988).

[6] ARNOLD, R. G. The Diceware Passphrase Home Page. `world.std.com/~reinhold/diceware.html`, 2014.

[7] ATALLAH, M. J., MCDONOUGH, C. J., RASKIN, V., AND NIRENBURG, S. Natural language processing for information assurance and security: an overview and implementations. In *Proceedings of the 2000 New Security Paradigms Workshop* (2001), ACM, pp. 51–65.

[8] ATKINSON, R. C., AND SHIFFRIN, R. M. Human memory: A proposed system and its control processes. *The Psychology of Learning and Motivation 2* (1968), 89–195.

[9] BADDELEY, A. Working memory. *Science 255*, 5044 (1992), 556–559.

[10] BADDELEY, A. D. *Human memory: Theory and practice*. Psychology Press, 1997.

[11] BALOTA, D. A., DUCHEK, J. M., AND LOGAN, J. M. Is expanded retrieval practice a superior form of spaced retrieval? A critical review of the extant literature. *The foundations of remembering: Essays in honor of Henry L. Roediger, III* (2007), 83–105.

[12] BICAKCI, K., AND VAN OORSCHOT, P. C. A multi-word password proposal (gridWord) and exploring questions about science in security research and usable security evaluation. In *Proceedings of the 2011 New Security Paradigms Workshop* (2011), ACM, pp. 25–36.

[13] BIDDLE, R., CHIASSON, S., AND VAN OORSCHOT, P. C. Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys (CSUR) 44*, 4 (2012), 19.

[14] BLISS, T. V., AND LØMO, T. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *The Journal of Physiology 232*, 2 (1973), 331–356.

[15] BLOCKI, J. *Usable Human Authentication: A Quantitative Treatment*. PhD thesis, Carnegie Mellon University, June 2014.

[16] BLOCKI, J., BLUM, M., AND DATTA, A. Naturally rehearsing passwords. In *Advances in Cryptology-ASIACRYPT 2013*. Springer, 2013, pp. 361–380.

[17] BOJINOV, H., SANCHEZ, D., REBER, P., BONEH, D., AND LINCOLN, P. Neuroscience meets cryptography: designing crypto primitives secure against rubber hose attacks. In *Proceedings of the 21st USENIX Security Symposium* (2012).

[18] BONNEAU, J. *Guessing human-chosen secrets*. PhD thesis, University of Cambridge, May 2012.

[19] BONNEAU, J. Moore's Law won't kill passwords. Light Blue Touchpaper, January 2013.

[20] BONNEAU, J., HERLEY, C., VAN OORSCHOT, P. C., AND STAJANO, F. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *2012 IEEE Symposium on Security and Privacy* (May 2012).

[21] BONNEAU, J., JUST, M., AND MATTHEWS, G. What's in a Name? Evaluating Statistical Attacks on Personal Knowledge Questions. In *FC '10: Proceedings of the the 14th International Conference on Financial Cryptography* (January 2010).

[22] BONNEAU, J., PREIBUSCH, S., AND ANDERSON, R. A birthday present every eleven wallets? The security of customer-chosen banking PINs. In *FC '12: Proceedings of the the 16th International Conference on Financial Cryptography* (March 2012).

[23] BONNEAU, J., AND SCHECHTER, S. Towards reliable storage of 56-bit secrets in human memory (extended version). Tech. rep., Microsoft Research.

[24] BONNEAU, J., AND SHUTOVA, E. Linguistic properties of multi-word passphrases. In *USEC '12: Workshop on Usable Security* (March 2012).

[25] BOYEN, X. Halting password puzzles. In *USENIX Security Symposium* (2007).

[26] BRAND, S. Department of Defense Password Management Guideline.

[27] BRANTZ, T., AND FRANZ, A. The Google Web 1T 5-gram corpus. Tech. Rep. LDC2006T13, Linguistic Data Consortium, 2006.

[28] BROSTOFF, A. *Improving password system effectiveness*. PhD thesis, University College London, 2004.

[29] BROWN, D. R. Prompted User Retrieval of Secret Entropy: The Passmaze Protocol. *IACR Cryptology ePrint Archive 2005* (2005), 434.

[30] BUNNELL, J., PODD, J., HENDERSON, R., NAPIER, R., AND KENNEDY-MOFFAT, J. Cognitive, associative and conventional passwords: Recall and guessing rates. *Computers & Security 16*, 7 (1997), 629–641.

[31] BUNTING, M. Proactive interference and item similarity in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition 32*, 2 (2006), 183.

[32] BURR, W. E., DODSON, D. F., AND POLK, W. T. Electronic Authentication Guideline. *NIST Special Publication 800-63* (2006).

[33] CAMERON, K. A., HAARMANN, H. J., GRAFMAN, J., AND RUCHKIN, D. S. Long-term memory is the representational basis for semantic verbal short-term memory. *Psychophysiology 42*, 6 (2005), 643–653.

[34] CAPLE, C. *The Effects of Spaced Practice and Spaced Review on Recall and Retention Using Computer Assisted Instruction.* PhD thesis, North Carolina State University, 1996.

[35] CEPEDA, N. J., PASHLER, H., VUL, E., WIXTED, J. T., AND ROHRER, D. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin 132*, 3 (2006), 354.

[36] CHIASSON, S., FORGET, A., BIDDLE, R., AND VAN OORSCHOT, P. C. Influencing users towards better passwords: persuasive cued click-points. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1* (2008), British Computer Society, pp. 121–130.

[37] CHIASSON, S., VAN OORSCHOT, P. C., AND BIDDLE, R. Graphical password authentication using cued click points. In *Computer Security–ESORICS 2007*. Springer, 2007, pp. 359–374.

[38] CLAIR, L. S., JOHANSEN, L., ENCK, W., PIRRETTI, M., TRAYNOR, P., MCDANIEL, P., AND JAEGER, T. Password exhaustion: Predicting the end of password usefulness. In *Information Systems Security*. Springer, 2006, pp. 37–55.

[39] DAVIS, D., MONROSE, F., AND REITER, M. K. On User Choice in Graphical Password Schemes. In *USENIX Security Symposium* (2004), vol. 13, pp. 11–11.

[40] DHAMIJA, R., AND PERRIG, A. Déjà Vu: A User Study Using Images for Authentication. In *Proceedings of the 9th Conference on USENIX Security Symposium - Volume 9* (Berkeley, CA, USA, 2000), SSYM'00, USENIX Association, pp. 4–4.

[41] DI CRESCENZO, G., LIPTON, R., AND WALFISH, S. Perfectly secure password protocols in the bounded retrieval model. In *Theory of Cryptography*. Springer, 2006, pp. 225–244.

[42] DUGGAN, G. B., JOHNSON, H., AND GRAWEMEYER, B. Rational security: Modelling everyday password use. *International Journal of Human-Computer Studies 70*, 6 (2012), 415–431.

[43] EBBINGHAUS, H. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot, 1885.

[44] FAHL, S., HARBACH, M., ACAR, Y., AND SMITH, M. On the ecological validity of a password study. In *Proceedings of the Ninth Symposium on Usable Privacy and Security* (2013), ACM, p. 13.

[45] GANESAN, R., DAVIES, C., AND ATLANTIC, B. A new attack on random pronounceable password generators. In *Proceedings of the 17th {NIST}-{NCSC} National Computer Security Conference* (1994).

[46] GASSER, M. A random word generator for pronounceable passwords. Tech. rep., DTIC Document, 1975.

[47] GREENE, R. L. Spacing effects in memory: Evidence for a two-process account. *Journal of Experimental Psychology: Learning, Memory, and Cognition 15*, 3 (1989), 371.

[48] JAKOBSSON, M., YANG, L., AND WETZEL, S. Quantifying the security of preference-based authentication. In *Proceedings of the 4th ACM Workshop on Digital Identity Management* (2008), ACM, pp. 61–70.

[49] JERMYN, I., MAYER, A., MONROSE, F., REITER, M. K., RUBIN, A. D., ET AL. The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium* (1999), vol. 8, Washington DC, pp. 1–1.

[50] JEYARAMAN, S., AND TOPKARA, U. Have the cake and eat it too—Infusing usability into text-password based authentication systems. In *Computer Security Applications Conference, 21st Annual* (2005), IEEE.

[51] JOSEFSSON, S. The Base16, Base32, and Base64 Data Encodings. RFC 4648 (Proposed Standard), Oct. 2006.

[52] JUELS, A., AND RIVEST, R. L. Honeywords: Making Password-cracking Detectable. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (New York, NY, USA, 2013), CCS '13, ACM, pp. 145–160.

[53] KARPICKE, J. D., AND ROEDIGER III, H. L. Expanding retrieval practice promotes short-term retention, but equally spaced retrieval enhances long-term retention. *Journal of Experimental Psychology: Learning, Memory, and Cognition 33*, 4 (2007), 704.

[54] KAUFMAN, C., PERLMAN, R., AND SPECINER, M. *Network security: Private communication in a public world*. Prentice Hall Press, 2002.

[55] KEITH, M., SHAO, B., AND STEINBART, P. J. The usability of passphrases for authentication: An empirical field study. *International Journal of Human-Computer Studies 65*, 1 (2007), 17–28.

[56] KELLEY, P. G., KOMANDURI, S., MAZUREK, M. L., SHAY, R., VIDAS, T., BAUER, L., CHRISTIN, N., CRANOR, L. F., AND LOPEZ, J. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *2012 IEEE Symposium on Security and Privacy* (2012), IEEE, pp. 523–537.

[57] KELSEY, J., SCHNEIER, B., HALL, C., AND WAGNER, D. Secure applications of low-entropy keys. In *Information Security*. Springer, 1998, pp. 121–134.

[58] KING, M. Rebus passwords. In *Proceedings of the Seventh Annual Computer Security Applications Conference, 1991* (Dec 1991), pp. 239–243.

[59] KITTUR, A., CHI, E. H., AND SUH, B. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), CHI '08, ACM, pp. 453–456.

[60] KOMANDURI, S., SHAY, R., KELLEY, P. G., MAZUREK, M. L., BAUER, L., CHRISTIN, N., CRANOR, L. F., AND EGELMAN, S. Of passwords and people: measuring the effect of password-composition policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 2595–2604.

[61] KURZBAN, S. A. Easily Remembered Passphrases: A Better Approach. *SIGSAC Rev. 3*, 2-4 (Sept. 1985), 10–21.

[62] LANDAUER, T., AND BJORK, R. Optimum rehearsal patterns and name learning. InM. M. Gruneberg, PE Morris, & RN Sykes (Eds.), Practical aspects of memory (pp. 625-632), 1978.

[63] LASTPASS. LastPass Security Notification. http://blog.lastpass.com/2011/05/lastpass-security-notification.html.

[64] LEONHARD, M. D., AND VENKATAKRISHNAN, V. A comparative study of three random password generators. In *IEEE EIT* (2007).

[65] MEUNIER, P. C. Sing-a-Password: Quality Random Password Generation with Mnemonics. 1998.

[66] MORRIS, R., AND THOMPSON, K. Password Security: A Case History. *Communications of the ACM 22*, 11 (1979), 594–597.

[67] MUNROE, R. Password Strength. `https://www.xkcd.com/936/`, 2012.

[68] PAVLIK, P. I., AND ANDERSON, J. R. Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied 14*, 2 (2008), 101.

[69] PERCIVAL, C. Stronger key derivation via sequential memory-hard functions. 2009.

[70] PIMSLEUR, P. A memory schedule. *Modern Language Journal* (1967), 73–75.

[71] PROCTOR, R. W., LIEN, M.-C., VU, K.-P. L., SCHULTZ, E. E., AND SALVENDY, G. Improving computer security for authentication of users: Influence of proactive password restrictions. *Behavior Research Methods, Instruments, & Computers 34*, 2 (2002), 163–169.

[72] PROVOS, N., AND MAZIERES, D. A Future-Adaptable Password Scheme. In *USENIX Annual Technical Conference, FREENIX Track* (1999), pp. 81–91.

[73] RABKIN, A. Personal knowledge questions for fallback authentication: Security questions in the era of Facebook. In *Proceedings of the 4th Symposium on Usable Privacy and Security* (2008), ACM, pp. 13–23.

[74] ROSS, J., IRANI, L., SILBERMAN, M. S., ZALDIVAR, A., AND TOMLINSON, B. Who Are the Crowdworkers?: Shifting Demographics in Mechanical Turk. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2010), CHI EA '10, ACM, pp. 2863–2872.

[75] SANCHEZ, D. J., GOBEL, E. W., AND REBER, P. J. Performing the unexplainable: Implicit task performance reveals individually reliable sequence learning without explicit knowledge. *Psychonomic Bulletin & Review 17*, 6 (2010), 790–796.

[76] SCHECHTER, S., BRUSH, A. B., AND EGELMAN, S. It's No Secret. Measuring the Security and Reliability of Authentication via "Secret" Questions. In *Security and Privacy, 2009 30th IEEE Symposium on* (2009), IEEE, pp. 375–390.

[77] SHAY, R., KELLEY, P. G., KOMANDURI, S., MAZUREK, M. L., UR, B., VIDAS, T., BAUER, L., CHRISTIN, N., AND CRANOR, L. F. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (2012), ACM, p. 7.

[78] STOBERT, E. A. Memorability of Assigned Random Graphical Passwords. Master's thesis, Carleton University, 2011.

[79] STROOP, J. R. Studies of Interference in Serial Verbal Reactions. *Journal of Experimental Psychology 18*, 6 (Dec. 1935), 643–662.

[80] STUBBLEFIELD, A., AND SIMON, D. Inkblot authentication. *Microsoft Research* (2004).

[81] TAO, H., AND ADAMS, C. Pass-Go: A Proposal to Improve the Usability of Graphical Passwords. *IJ Network Security 7*, 2 (2008), 273–292.

[82] THE ETHICAL RESEARCH PROJECT. Post-experiment survey for deception studies. `https://www.ethicalresearch.org/`.

[83] VAN OORSCHOT, P. C., AND THORPE, J. On predictive models and user-drawn graphical passwords. *ACM Transactions on Information and System Security (TISSEC) 10*, 4 (2008), 5.

[84] VAN OORSCHOT, P. C., AND THORPE, J. Exploiting predictability in click-based graphical passwords. *Journal of Computer Security 19*, 4 (2011), 669–702.

[85] VERAS, R., COLLINS, C., AND THORPE, J. On the semantic patterns of passwords and their security impact. In *Network and Distributed System Security Symposium (NDSS'14)* (2014).

[86] VU, K.-P. L., PROCTOR, R. W., BHARGAV-SPANTZEL, A., TAI, B.-L. B., COOK, J., AND EUGENE SCHULTZ, E. Improving password security and memorability to protect personal and organizational information. *International Journal of Human-Computer Studies 65*, 8 (2007), 744–757.

[87] WEINSHALL, D., AND KIRKPATRICK, S. Passwords You'll Never Forget, but Can't Recall. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2004), CHI EA '04, ACM, pp. 1399–1402.

[88] WIEDENBECK, S., WATERS, J., BIRGET, J.-C., BRODSKIY, A., AND MEMON, N. PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies 63*, 1 (2005), 102–127.

[89] WIXTED, J. T. The psychology and neuroscience of forgetting. *Annual Psychology Review 55* (2004), 235–269.

[90] WOOD, H. M. *The use of passwords for controlled access to computer resources*, vol. 500. US Department of Commerce, National Bureau of Standards, 1977.

[91] WOZNIAK, P. SuperMemo 2004. *TESL EJ 10*, 4 (2007).

[92] YAN, J. J., BLACKWELL, A. F., ANDERSON, R. J., AND GRANT, A. Password Memorability and Security: Empirical Results. *IEEE Security & privacy 2*, 5 (2004), 25–31.

[93] ZVIRAN, M., AND HAGA, W. User authentication by cognitive passwords: an empirical assessment. In *Proceedings of the 5th Jerusalem Conference on Information Technology* (Oct 1990), pp. 137–144.

[94] ZVIRAN, M., AND HAGA, W. J. Passwords Security: An Exploratory Study. Tech. rep., Naval Postgraduate School, 1990.

[95] ZVIRAN, M., AND HAGA, W. J. Password security: an empirical study. *Journal of Management Information Systems 15* (1999), 161–186.

able	abuse	acid	acorn	acre	actor	add	adobe	adult	aft	age	agile	agony
air	alarm	album	alert	alive	ally	amber	ample	angle	anvil	apply	apron	arbor
area	army	aroma	arrow	arson	ask	aspen	asset	atlas	atom	attic	audit	aunt
aura	auto	aware	awful	axis	baby	back	bad	baker	bare	basis	baton	beam
beer	begin	belly	bench	best	bias	big	birth	bison	bite	blame	blind	bloom
blue	board	body	bogus	bolt	bones	book	born	bound	bowl	box	brain	break
brief	broth	brute	buddy	buff	bugle	build	bulk	burst	butt	buy	buzz	cabin
cadet	call	camp	can	cargo	case	cedar	cello	cent	chair	check	child	chose
chute	cider	cigar	city	civil	class	clear	climb	clock	club	coal	cobra	code
cog	color	comic	copy	cord	cost	court	cover	craft	crew	crime	crown	cruel
cups	curve	cut	cycle	daily	dance	dark	dash	data	death	debt	decoy	delay
depot	desk	diary	diet	dim	ditto	dizzy	dose	doubt	downy	dozen	drawn	dream
drive	drop	drug	dry	due	dust	duty	dwarf	eager	early	easy	eaten	ebb
echo	edge	edit	egg	elbow	elder	elite	elm	empty	end	enemy	entry	envy
equal	era	error	essay	ether	event	exact	exile	extra	eye	fact	faith	false
fancy	far	fatal	fault	favor	feast	feet	fence	ferry	fetch	feud	fever	fiber
field	fifty	film	find	first	fit	flat	flesh	flint	flow	fluid	fly	focus
foe	folk	foot	form	four	foyer	frame	free	front	fruit	full	fume	funny
fused	fuzzy	gala	gang	gas	gauge	gaze	gel	ghost	giant	gift	give	glad
gleam	glory	glut	goat	good	gorge	gourd	grace	great	grid	group	grub	guard
guess	guide	gulf	gym	habit	half	hand	happy	harsh	hasty	haul	haven	hawk
hazy	head	heel	help	hem	here	high	hike	hint	hoax	holy	home	honor
hoop	hot	house	huge	human	hurt	husk	hyper	ice	idea	idle	idol	ill
image	inch	index	inner	input	iris	iron	issue	item	ivory	ivy	jade	jazz
jewel	job	join	joke	jolly	judge	juice	junk	jury	karma	keep	key	kid
king	kiss	knee	knife	known	labor	lady	laid	lamb	lane	lapse	large	last
laugh	lava	law	layer	leaf	left	legal	lemon	lens	level	lies	life	lily
limit	link	lion	lip	liter	loan	lobby	local	lodge	logic	long	loose	loss
loud	love	lowly	luck	lunch	lynx	lyric	madam	magic	main	major	mango	maple
march	mason	may	meat	media	melon	memo	menu	mercy	mess	metal	milk	minor
mixed	model	moist	mole	mom	money	moral	motor	mouth	moved	mud	music	mute
myth	nap	navy	neck	need	neon	new	nine	noble	nod	noise	nomad	north
note	noun	novel	numb	nurse	nylon	oak	oats	ocean	offer	oil	old	one
open	optic	orbit	order	organ	ounce	outer	oval	owner	pale	panic	paper	part
pass	path	pause	pawn	pearl	pedal	peg	penny	peril	petty	phase	phone	piano
piece	pipe	pitch	pivot	place	plea	plot	plug	poet	point	polo	pond	poor
poppy	porch	posse	power	press	price	proof	pub	pulse	pump	pupil	pure	quart
queen	quite	radio	ram	range	rapid	rate	razor	real	rebel	red	reef	relic
rents	reply	resin	rhyme	rib	rich	ridge	right	riot	rise	river	road	robot
rock	roll	room	rope	rough	row	royal	ruby	rule	rumor	run	rural	rush
saga	salt	same	satin	sauce	scale	scene	scope	scrap	sedan	sense	serve	set
seven	sewer	share	she	ship	show	shrub	sick	side	siege	sign	silly	siren
six	skew	skin	skull	sky	slack	sleep	slice	sloth	slump	small	smear	smile
snake	sneer	snout	snug	soap	soda	solid	sonic	soon	sort	soul	space	speak
spine	split	spoke	spur	squad	state	step	stiff	story	straw	study	style	sugar
suit	sum	super	surf	sway	sweet	swift	sword	syrup	taboo	tail	take	talk
taste	tax	teak	tempo	ten	term	text	thank	theft	thing	thorn	three	thumb
tiara	tidal	tiger	tilt	time	title	toast	today	token	tomb	tons	tooth	top
torso	total	touch	town	trade	trend	trial	trout	true	tube	tuft	tug	tulip
tuna	turn	tutor	twist	two	type	ultra	uncle	union	upper	urban	urge	user
usual	value	vapor	vat	vein	verse	veto	video	view	vigor	vinyl	viper	virus
visit	vital	vivid	vogue	voice	voter	vowel	wafer	wagon	wait	waltz	warm	wasp

Table 3: The 676 (26^2) words used by the *words* treatment