

PhorceField: A Phish-Proof Password Ceremony

Abstract

Many widely deployed phishing defense schemes, such as SiteKey, use client-side secrets to help users confirm that they are visiting the correct website before entering their passwords. Unfortunately, studies have demonstrated that up to 92% of users can be convinced to ignore missing client-side secrets and enter their passwords into phishing pages. However, since client-side secrets have already achieved industry acceptance, they are an attractive building block for creating better phishing defenses. We present PhorceField, a phishing resistant password ceremony that combines client-side secrets and graphical passwords in a novel way that provides phishing resistance that neither achieves on its own. PhorceField enables users to login easily, but forces phishers to present victims with a fundamentally unfamiliar and onerous user interface. Victims that try to use the phisher’s interface to enter their password find the task so difficult that they give up without revealing their password. We have evaluated PhorceField’s phishing resistance in a user study in which 21 participants used PhorceField for a week and were then subjected to a simulated phishing attack. On average, participants were only able to reveal 20% of the entropy in their password, and none of them revealed their entire password. This is a substantial improvement over previous research that demonstrated that 92% of users would reveal their entire password to a phisher, even if important security indicators were missing[27].

PhorceField is easy to deploy in sites that already use client-side secrets for phishing defense – it requires no client-side software and can be implemented entirely in javascript. Banks and other high value websites could therefore deploy it as a drop-in replacement for existing defenses, or deploy it on an “opt-in” basis, as Google has done with its phone-based “2-step verification” system.

1 Introduction

High-value web services, such as online banking, have deployed client-side secrets, like those used in SiteKey, as part of their phishing defense systems. These schemes are intended to help users identify a site as valid so they may be confident they are entering their credentials into the proper prompt. Client-side secrets have achieved tremendous industry adoption. For example, Bank of America has deployed its SiteKey system to over 15 million online banking customers.

Phishing is a major form of online fraud. Phishing attacks require little technical skill and are easy to launch. Furthermore, since users often re-use passwords on multiple sites, passwords gained in a phishing attack against one website are often re-usable on other sites. Over 3.6 million people in the U.S. lost money to phishing attacks in 2007, losing over 3 billion USD[15]. Since then, phishing attacks have grown much more sophisticated. For example, Google recently claimed that it uncovered a coordinated set of spear-phishing attacks against high-profile and high-value targets[25]. These attacks use emails that appear to come from their victims’ friends and co-workers and contain plausible details to trick victim’s into lowering their guard. It is unreasonable to expect users to be able to detect such attacks.

Despite the widespread deployment of client side secrets, there is strong evidence that they provide poor phishing protection: 92% of participants in one study ignored missing security indicators,

Metric	SiteKey	PhorceField
Percent users revealing their entire password	92%	0%
Average amount of password revealed	92%	20%

Table 1: Success rate of phishing attacks on PhorceField and SiteKey. The SiteKey statistics are derived from Schechter, et al. [27]. Since SiteKey users reveal all-or-nothing of their password, the average amount of password entropy revealed is the same as the percentage of subjects who revealed their password. In our PhorceField study, all users were assigned passwords with 70 bits of entropy, and they revealed an average of 13.8 bits of information about their password.

including their client-side secret. Current client-side secret systems depend on the user to verify the presence of the secret. A phisher can therefore circumvent the defense mechanism by tricking users into ignoring the missing secret. To solve this problem, we need to force users to verify the presence of the client-side secret before entering their password.

In this paper, we present PhorceField: a novel integration of client-side secrets and graphical passwords that better utilizes existing client-side secrets while alleviating the users of the responsibility for creating, maintaining and supplying passwords. With PhorceField, a legitimate password prompt has access to a secret set of images that enables it to create an easy-to-use password prompt. Phishers do not have access to the secret images and hence must present victims with a fundamentally different and more difficult interface. Users cannot ignore the differences, as with previous schemes such as SiteKey, and hence cannot slip into dangerous click-whirr behaviors[18]. Even if users do attempt to interact with the phisher’s page, though, the phishing page requires so much effort that the victims give up before they can reveal their password.

PhorceField exploits well-known strengths and weaknesses of human memory[3, 9]. During a normal PhorceField login, users must perform an image recognition task, which is relatively easy for humans. During a phishing attack, though, victims must perform an image recall task, which is substantially harder. Furthermore, PhorceField is designed so that victims will experience memory interference during a phishing attack, causing additional frustration and error. To our knowledge, PhorceField is the first password ceremony to take advantage of these properties of human memory.

PhorceField is easy to use. The PhorceField user experience is identical to a cognometric graphical password[10], which have been shown to be usable and to have memorable passwords. Thus, this paper focuses only on PhorceField’s security against phishing attacks.

We have conducted a user study to evaluate PhorceField’s ability to resist phishing attacks. Participants used PhorceField for one week and were then presented with a simulated phishing attack on their PhorceField password. We made special effort to ensure the ecological validity of our study and to avoid participant bias. For example, participants worked on their own computers in their normal environments, and we told participants that the study was focused on “usability” instead of security.

As the user study results summarized in Table 1 show, PhorceField users presented with a phishing attack give up before they are able to reveal their entire password. Participants in our study revealed, on average, only 13.8 bits of information (out of 70 bits of entropy) about their password, and no participant revealed his entire password. This is a substantial improvement over recent results on SiteKey, the current industry standard for preventing password phishing, that show that 92% of SiteKey users will reveal their password to a phisher[27].

PhorceField combines two existing technologies – client-side secrets (e.g. secure HTTP cookies) and graphical passwords – in a novel way to achieve a level of phishing-resistance that neither technology achieves on its own. Our results demonstrate that previous schemes based on client-

side secrets, such as SiteKey[4] and Dynamic Security Skins[11], are not extracting the full benefit of the client-side secret. PhorceField could serve as a drop-in replacement for the login ceremony of either of these schemes, substantially improving their phishing-resistance.

In summary, PhorceField demonstrates three new techniques for designing phishing-resistant password ceremonies:

- It forces phishers to present a fundamentally different interface to their victims. This gives users a better chance to detect the attack. With previous schemes, phishers can create interfaces that differ only superficially from legitimate ones.
- It forces phishing victims to complete a much more difficult task. Even if a victim is fooled by the phisher’s attack, she is unlikely to succeed in communicating her password to the phisher.
- It exploits strengths and weaknesses of human memory to make phishing attacks difficult for their victims. Victims of a phishing attack experience memory interference while looking through hundreds of similar images, causing frustration and error.

We make the following additional contributions:

- We present results from a user study demonstrating that PhorceField successfully protected all participants against a simulated phishing attack.
- We show how PhorceField can easily integrate into existing anti-phishing mechanisms, such as SiteKey and DSS, without introducing any new hardware or software requirements.

2 Background

Phishers steal user credentials by tricking victims into revealing private information, such as passwords. In this section, we review (1) why phishing is not solved by existing technologies, such as SSL or malware defenses, (2) that some proposed phishing solutions offer some protection but do not address the core problem of password disclosure, and (3) that previous solutions targeted specifically at preventing password disclosure are not effective.

Non-solutions. Phishing is a separate problem from malware (such as key-loggers), click-jacking, and other techniques that attackers may use to steal user secrets. Even if a user’s computer is free of malware and has effective defenses against click-jacking, a phisher may still trick the user into divulging his password. Phishing cannot be solved solely with cryptography. SSL and PAKE protocols[5] cannot protect the user’s password if she types it into a phisher’s website. Password managers and one-time passwords do not prevent phishing, either. A phisher can defeat a password manager by convincing victims to disable it and type in their master password. Given the gullibility of users[27] and the usability problems with password managers[6], the ruse is likely to succeed. One-time password generators, whether implemented as a special-purpose token, software on a cell-phone, or delivered to the user’s phone via SMS, all require users to manually copy the one-time password from the device to their computer. Phishers can trick users into entering the password into their page instead of the user’s intended website, as was done in a real phishing attack against Citibank[19].

Therefore, we need separate defenses to help users avoid entering passwords into malicious prompts.

Anti-phishing toolbars[23, 13], spam detectors[26], phishing site blacklists and white-lists[31], and other reactionary mechanisms for fighting phishing offer some protection[7, 35, 1, 2], but

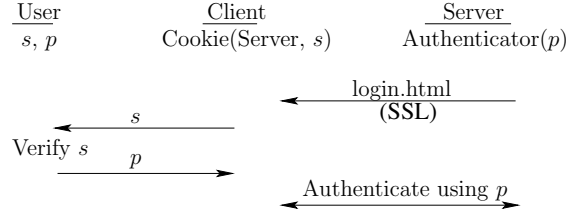


Figure 1: A generalized variant of the SiteKey and Dynamic Security Skins login ceremonies. The user’s secret image, s is stored on her machine and is accessible only to legitimate password prompts, i.e. prompts from “Server” in the case of SiteKey, or prompts presented by the DSS plug-in in the case of Dynamic Security Skins. Users are supposed to verify that s is displayed in the prompt before entering their password. Once the user has entered her password, any password-based authentication protocol can be used to authenticate the user to the remote server.

phishers have developed many techniques for evading these defenses[22], so millions of users fall victim to phishing each year[15].

Two-factor authentication can mitigate the damage of a stolen password, but it does not protect the password itself. PhorceField can complement a second authentication factor, but it can also protect a user’s password even when the password is the sole authentication factor, as in our prototype.

The solutions above fail to prevent phishing because they do not address the core weakness exploited by phishers: Users cannot determine whether a password prompt is legitimate or malicious.

Prior secure password prompt proposals. Researchers have made several attempts to create password schemes that will help users distinguish legitimate prompts from malicious ones.

Secure attention keys allow users to summon a trusted password prompt by pressing a special sequence of keys – typically Ctrl-Alt-Del. Secure attention keys do not prevent phishing since phishers can easily trick users to skip the special key sequence.

Dynamic Security Skins (DSS)[11] and SiteKey[4] are conceptually similar schemes that use a client-side secret to help users recognize legitimate password prompts. In these schemes, a secret image known to the user is stored on the user’s device. This image is presented to the user as part of the standard password prompt. The user is supposed to verify the image is present before entering her password. Figure 1 shows the protocol for logging in using these schemes. SiteKey and DSS differ in the location of the image: SiteKey displays it above the password entry field, DSS displays it behind the field.

Unfortunately, 92% of SiteKey users will ignore a missing image and enter their password anyway[27]. We could find no published user study evaluating phishing attacks against Dynamic Security Skins, but it also depends on user vigilance for security and so is likely to offer little protection against phishing.

Like SiteKey and DSS, PhorceField leverages a client-side secret to prevent password phishing. Any system that prevents users from communicating passwords to phishers must use either a client-side secret or some client-side peripheral that is not available to the phisher, since otherwise phishers would be able to emulate the password prompt with enough fidelity to fool many users. Thus, the only question is: how can we extract the most benefit from this client-side secret? PhorceField achieves a far greater level of phishing resistance than previous schemes, such as SiteKey and DSS, without imposing any additional setup or portability costs.

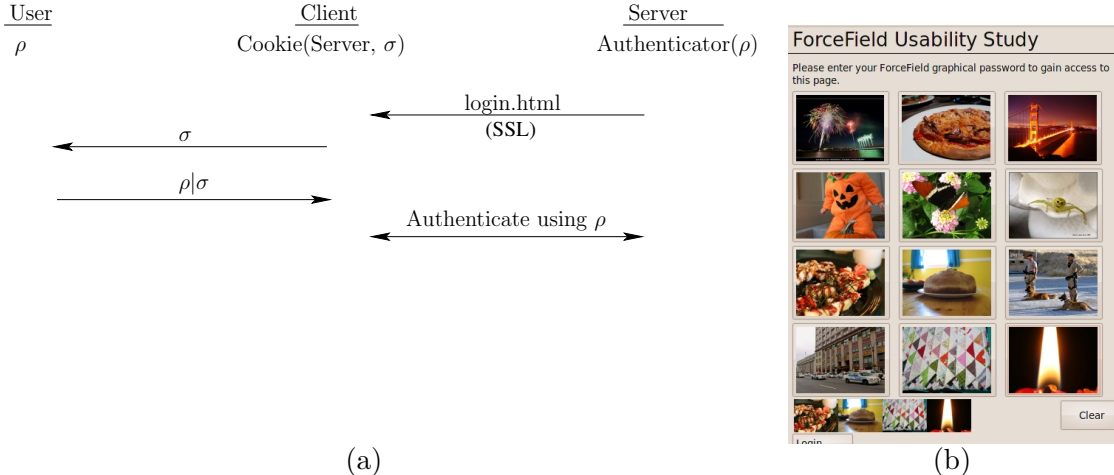


Figure 2: (a) The PhorceField ceremony. The user’s secret image set, σ , is stored on her device and inaccessible to phishers. (b) An example PhorceField password prompt using Creative Commons licensed images from the Flickr photo-sharing site.

3 PhorceField

Our goal is to create a password ceremony such that attackers cannot present a usable password prompt to their victims. Therefore, legitimate prompts must have access to some secret that is not known to attackers. Furthermore, passwords cannot consist of letters, numbers, etc., since phishers can present easy interfaces for inputting those symbols. PhorceField satisfies the above two requirements by using a cognometric graphical password scheme and by storing the graphical password images securely on the user’s device. Figure 2(a) shows the PhorceField login ceremony and Figure 2(b) shows an instance of a PhorceField password prompt.

Cognometric graphical passwords present users with a set, σ , of images and users log in by clicking on the images in a certain sequence or by clicking on a certain subset of images. Prior work by Moncur, et al. has shown that cognometric passwords are usable and memorable[21]. We refer to the user’s password as ρ . In PhorceField, the set σ must be drawn from a much larger set, Σ . Thus, the user’s password is a word in the language Σ^* but, since the prompt already has access to σ , the user only needs to communicate $\rho|\sigma$, which may only constitute 10-20 bits of information. A phisher that does not know σ , though, must trick the user into revealing ρ , which may require the user to communicate hundreds of bits of information, making the task much harder.

For security, Σ should be as large as possible and, for usability, σ should be small. For our prototype implementation, we chose $|\sigma| = 12$, since this makes it easy to enter passwords on cell phones and other mobile devices with the standard phone keys 0-9, “*”, and “#”, and Σ consisted of 188218 creative-commons licensed images collected from the Flickr photo service[34]. There are over 100 million such images on Flickr, so our prototype could be easily scaled up for real-world deployment[30]. We collected images by searching for 193 different concrete nouns, such as “cow”, “flowers”, “sky”, and “tree”. We constructed each participant’s σ by selecting 12 concrete nouns and then selecting an image from each noun’s image set. Passwords in our implementation consist of a sequence of 4 distinct images (order matters). Our system randomly generated passwords for participants. This yields an entropy of 181 bits for σ , 70.0 bits for ρ , and 13.5 bits for $\rho|\sigma$. Our scraper also downloaded the description, tags and titles for each image, which we use later to develop the phishing attack interface in our user study.

Each user’s σ must be stored on her device. Our implementation installed σ on participants’ computers when they enrolled in our user study. We did not provide any mechanism for initializing σ on a second computer, so participants could only log in from the computer they used to register in the study. A full implementation could use a conditioned-safe ceremony[18] to initialize other computers with σ .

4 Phishing attacks against PhorceField

We argue in this section that phisher’s only have two possible strategies for extracting ρ from their victims: brute force attacks and search attacks.

Brute force attacks. A phisher that does not know σ may attempt to learn ρ by presenting the user with an invalid password prompt. If the invalid prompt contains some images from the user’s set, σ , then the user may click on them. Note that in many cases, the user may refuse to interact with a prompt unless it contains exactly the images in her σ but, for this analysis, we pessimistically assume that the user will select all the images in her ρ , even for a badly malformed prompt. By performing this attack repeatedly with different images each time, the attacker may eventually recover all of ρ . The attacker can reduce the number of repetitions required to complete the attack by placing more images into each prompt. However, if the number of pictures in the invalid prompt is too large, then users will give up in frustration, and the attacker will learn nothing. If we assume that users will give up if asked to examine more than m pictures in one prompt, then the probability that a phisher can recover σ after repeating the attack r times is $\binom{mr}{|\sigma|} / \binom{|\Sigma|}{|\sigma|}$. For example, our user study described in the next Section used $|\sigma| = 12$ and $|\Sigma| \approx 2 * 10^5$. From the results of our study, we can estimate that $m \leq 3500$ and $r \leq 10$. With these parameters, the probability a phisher can recover σ with this attack is less than 2^{-30} .

Search attacks. Since brute force attacks will not work, a phisher must trick a user into communicating the images in her ρ to the phisher. This is a search problem: the phisher can iteratively make adaptive oracle queries to the victim in order to narrow down the search space for ρ . PhorceField resists search attacks in several ways.

With PhorceField, attackers must present victims with an interface that is substantially different from a legitimate PhorceField password prompt. Victims of phishing attacks therefore cannot slip into a “click-whirr” response mode[18]; they must actively evaluate the situation, giving them a much better chance of detecting the attack.

PhorceField trains users to distinguish their password images from a small set of distractor images – they do not need to remember every detail of their password images. A phisher, however, must coax enough information out of his victims in order to identify their images within a set of thousands or millions of candidate images. In many cases, users will be unable to provide more than the most prominent features of their password images, leaving the phisher with potentially thousands of possible matches. He can try to get the victim to sift through the matches, but our user study shows that victims will often give up before finding their password images.

We constructed the database of images used in our PhorceField prototype so that logging in is easy but cooperating with a phisher is difficult. Recall that Σ contains 188218 images representing 193 different concrete nouns, or about 975 images per noun. Constructing the database in this way causes users to suffer from memory interference while looking through candidate matches for their password images[9, 3]. As the phisher presents the user with more and more candidate images that are similar to the victim’s target image, the victim loses her ability to precisely identify her target image because it becomes blurred together with the candidates. This leads to errors, as victims may accidentally select a wrong image, and frustration within the victims, causing them to quit

cooperating with the phisher before they succeed in finding their password images.

The preceding arguments show that brute force attacks are impractical and that search attacks, no matter how clever, will require significantly more time and mental effort from victims than is required to log in. However, we cannot analytically determine how users will react when presented with a search attack on their password; for that, we need a user study.

5 User Study

The purpose of our user study is to measure the success of phishing attacks against PhorceField passwords. This study does not attempt to measure the usability of the PhorceField prompt or the memorability of graphical passwords – those topics have been explored elsewhere[10]. Our study includes numerous conservative design decisions, so we believe the results of this study represent an upper bound on the success probability a phisher can achieve with this attack.

Participants in our study were told that we were conducting an experiment to evaluate the usability of graphical passwords. After consenting to the study, participants were shown their set σ and password ρ and required to practice entering their password five times. They then downloaded a Firefox plug-in that randomly prompted them to enter their PhorceField password up to four times per day. By prompting participants randomly four times per day, we ensured that participants were familiar with their passwords at the time of our simulated phishing attack.

Participants were told that the study would last two weeks and that they would receive \$20 if they entered their password at least 14 times over the entire study period, \$10 if they entered their password between 7 and 13 times, and no compensation if they entered their password fewer than 7 times. After about one week, they received an email from us indicating that we had lost their password and requesting them to visit the study website to help us recover it. We then measured how much information they were able to divulge about their σ and ρ . Participants were then presented with a debriefing questionnaire.

We were careful to avoid priming the participants about security or phishing in particular. Subjects were told that the study was about usability, not security. All study materials used the name “ForceField” instead of “PhorceField”. The attack email clearly came from us and directed the users to the same website that they visited to sign up for the study. Thus our attack was missing all the cues of a real phishing attack and hence we believe that our results are an upper bound on the success rate a real attacker would experience with this attack.

Figure 3 shows a screen-shot of the password recovery page. Participants could use the search box to enter queries and could click on words in the tag cloud on the right-hand side of the page. As is demonstrated in the screen-shot, the search results were quite good because the Flickr photos were so well tagged. The interface was modeled after the Google image search interface at the time of the study and, like Google, never indicated the number of images that participants had to search through, since we felt that would only discourage them.

We conducted a pilot study to determine the number of participants needed to estimate the mean number of σ images revealed by a user. During our pilot study, thirteen subjects used our graphical password prompt for seven days and received the phishing attack page under the guise that the experimenters lost their password. The subjects revealed on average 0.35 images in their σ (SD = 0.60). Therefore, we require 20.7 subjects to determine the average number of σ images revealed in a phishing attack with a 99% confidence interval ± 0.34 images[12].

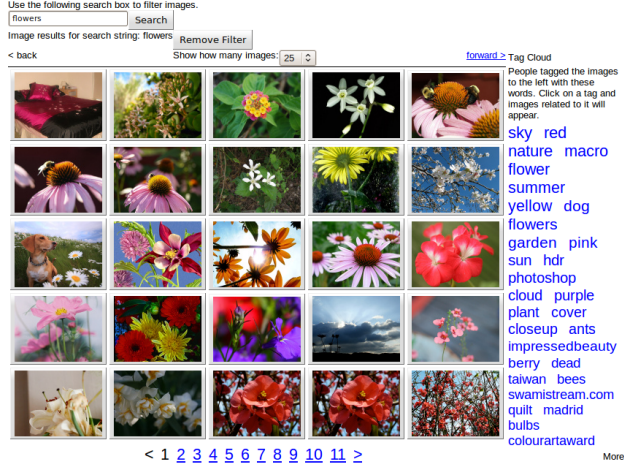


Figure 3: Our phishing attack website. Users could search for images using the search box at the top of the page and could click on tags in the tag cloud on the right-hand side.

6 Results

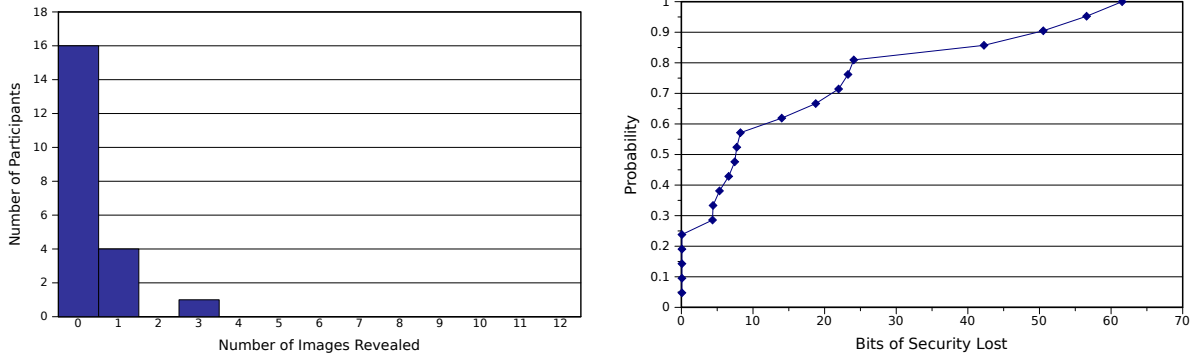
We conducted our user-study over the course of three weeks. We recruited 45 participants from a Craigslist posting. To ensure that subjects had sufficient exposure to their password, we only considered subjects that successfully logged in at least 4 times and at least once in the two days prior to receiving the attack email. 23 subjects met this criteria. This retention rate is comparable to other graphical password user studies[21]. We also eliminated from the analysis two subjects who thought our attack email was a real phishing attack. Participants ranged in age from 18 to 39 years and were varied in race (including Asian, African-American, Caucasian and Hispanic) and profession (including students, actors, IT professionals and HR representatives).

We evaluate both the explicit and implicit password information revealed by participants. Participants explicitly revealed part of their password if they found a password image and clicked on it. They revealed information implicitly by searching for images on the phishing site, even if their search was unsuccessful. For example, if a user spends a long time looking through pictures of dogs on the phishing site, then the phisher can infer that one of the user’s password images is of a dog.

Explicitly-revealed information about σ . Figure 4(a) shows the number of σ images our participants were able to find and click on. On average, participants clicked on 0.3 images of their σ . No participant clicked on an image that was in their σ but not in their ρ . Furthermore, as Figure 4(a) shows, 76% participants did not click on any of their σ images, and the others were only able to find at most three of their σ images, implying that PhorceField offers strong protection for almost everyone. During the attack, several participants clicked on random images out of frustration, but doing so provides no useful information to a phisher.

Implicitly-revealed information about σ . Even if users fail to find an image in their σ , they may still reveal information about that image through their search activities. For example, if a victim searches for the term “flowers” during a phishing attack, then the phisher can reduce the search space for one image from 188218 to 3117 images. Furthermore, if that user looks at 10 pages of results without clicking on any of them, then the phisher can conclude that the user’s image was not on those pages. If the user performs a second search on the term “plant”, the phisher can intersect the two results sets to further narrow the candidate set.

Users may also click on images that are not in their σ but are visually or semantically similar to



(a) The number of σ images clicked by participants in our study. No user clicked on an image in $\sigma \setminus \rho$. (b) The cumulative distribution function of entropy loss for σ from participants in our study.

Figure 4: Explicit and implicit information revealed about σ .

images that are in their σ . We used the tags on the images users clicked in order to approximate this information, since the tags assigned by Flickr users cover both visual and semantic aspects of the images. Therefore, we took the tags on each clicked image and added them as search queries during the analysis described below. However, we discovered that many of the tags conveyed contextual information, such as the type of camera used to create the photograph, that would not be apparent to a phishing victim and therefore would not contribute useful information to a phisher. A real phisher could clean the image tags to avoid this problem, but we took a shortcut: For each image a participant clicked on, we computed the set of tags on that image that also occur on an image in the participant’s σ and treat those tags as additional search queries performed by that participant. In reality, a phisher would not know which tags occur in the user’s σ , so this simplification grossly over-estimates the information gained by a phisher.

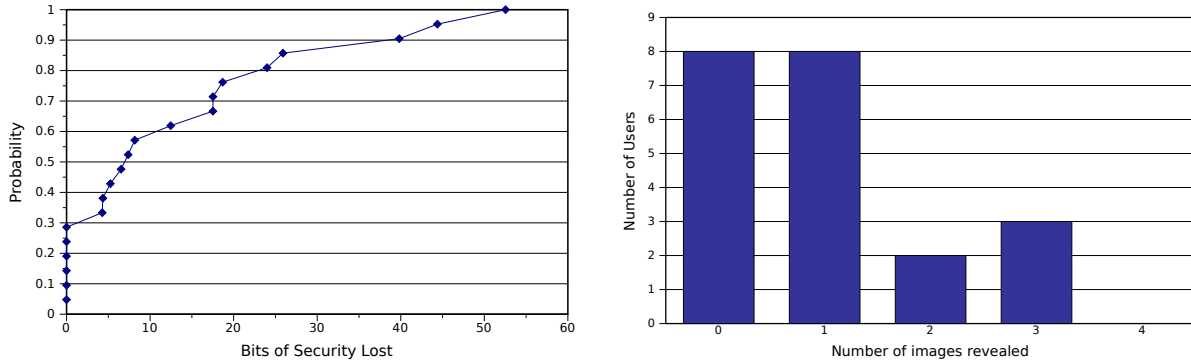
Given the set of search queries performed by a participant, we upper bound the information gained by the phisher as follows. For each search query, q , let S_q be the set of images in the search result set, and let $U_q \subseteq S_q$ be the set of images in S_q that the user never looked at. For search queries derived from tags on clicked images, $U_q = S_q$. For each image $\iota \in \sigma$, let

$$C_\iota = \begin{cases} \{\iota\} & \text{if user clicked on } \iota \\ \Sigma & \text{if } \forall q. \iota \notin S_q \\ \bigcap_{q:\iota \in U_q} U_q \cap \bigcap_{q:\iota \in S_q \setminus U_q} S_q & \text{otherwise} \end{cases}$$

In other words, for each image, we take all the search results that contained that image and intersect them. We also remove any images the user looked at in that search set, unless the user overlooked ι , in which case we take the whole result set. We then used the C_ι values to bound the entropy loss for each user’s σ (details of this computation will be in a companion tech report). Note this estimate is conservative because it assumes the phisher knows which search queries contained each image and whether the user overlooked an image in each results set.

Figure 4(b) shows the cumulative distribution function of entropy loss experienced by participants in our study. No participant revealed more than 62 bits of information about their σ , giving them a residual security of at least 119 bits. Furthermore, more than 80% of participants lost less than 25 bits of σ security.

Implicitly-revealed information about ρ . To estimate the amount of information a phisher can gain about a user’s password, we compute C_ι as above for each $\iota \in \rho$, and compute an upper



(a) The cumulative distribution function of entropy loss for ρ from participants in our study. (b) The number of password images implicitly revealed by participants in our study.

Figure 5: The implicitly-revealed information about ρ . In (b), we assume the attacker steals σ after interacting with the victim. Even in that case, he is not able to learn the victim’s entire password.

bound on the entropy loss as above. Note that this analysis is conservative for the same reasons as before, plus it conservatively assumes the phisher can infer which candidate sets, C_i correspond to images in ρ , and the position of each image in ρ . Figure 5(a) shows the CDF of bits lost on ρ . On average, participants revealed only 13.8 bits of information about their password. No participant revealed more than 52.6 bits of information (out of 70 bits) about her password, and 85% of our participants revealed less than 30 bits of information. PhorceField provided strong protection for all our participants’ passwords.

Implicitly-revealed information about $\rho|\sigma$. Finally, to demonstrate the resiliency of PhorceField, we assume the attacker gains access to σ after conducting the phishing attack. Although a real attacker could attempt to conduct a second phishing attack in this case, the analysis below is intended to show that phishing attacks without σ reveal very little information about user passwords. In this case, we consider an image $\iota \in \rho$ to be completely revealed if the attacker gains any information about it during the phishing attack, i.e. if $C_\iota \neq \Sigma$. Note this assumes the attacker can tell which images are in ρ versus $\sigma \setminus \rho$. Figure 5(b) shows the distribution of the number of password images revealed in this scenario. Over 75% of the participants revealed fewer than 2 of their password images, and no one revealed all 4 of their password images. If we assume the phisher knows the location of each revealed image within the user’s password, then our participants revealed, on average, 3.52 bits of information about $\rho|\sigma$.

Other observations. As mentioned earlier, we asked participants at the end of the study whether they suspected our email was part of a phishing attack, and we removed two subjects who refused to visit our password recovery page due to phishing suspicion. We also tested users to confirm that they still remembered their passwords – only one subject failed this test.

7 Discussion

The results of our user study imply that PhorceField will prevent most password phishing attacks from succeeding. 76% of our participants failed to find even a single image in their password, and no participant found his entire password. Even assuming the phisher later gained access to σ , no participant entirely compromised his password. This compares quite favorably to passive phishing defenses such as SiteKey[4]. Previous studies have shown that 92% of users will reveal their entire text password to a phisher, even if their SiteKey is missing. In our study, 0% of participants

revealed their entire password.

Our results suggest reasonable security parameters for a real-world PhorceField deployment. Our choice of $|\Sigma| \approx 2 \times 10^5$ was sufficiently large to make searching for images difficult. Likewise, selecting about a thousand images for each concrete noun made examining search results a tedious task, improving phishing-resistance. Future deployments can increase security without harming usability by choosing a larger Σ . For example, there are over 100 million Creative Commons licensed photos on Flickr[30], so it would be straightforward to construct a Σ with over 100 million images representing over 10000 concrete nouns. Given that some participants revealed 3 of their password images, $|\rho|$ should be at least 6 and possibly higher. Passwords should not allow repeated images, since this forces users to find the maximum possible number of images during a phishing attack.

Most users gave up after trying to find 1 or 2 images, so their search queries only revealed information about 1.33 images in their σ on average. Phishers could design attacks to avoid this bottleneck, but the success rate may or may not improve. For example, a phisher could present victims with 12 text boxes and ask the victim to describe each of the images in their σ . By forcing victims to describe their entire σ instead of allowing them to focus on one image at a time, the attacker may gain information about more images in σ . However, the information gained about each image would probably be less specific, e.g. the victim might simply enter “bird” instead of performing several searches such as “bird”, “flying”, and “flock”, which together reveal much more information about the given image. We could render this attack ineffective by using images, such as random art images, that admit no easy description.

8 Related Work

We discussed related anti-phishing technologies in Section 2, so we focus only on related graphical password research here.

Suo, et al., provide an extensive review of graphical password systems and group them into three categories[29]. Cognometric graphical password systems are the most prevalent. The Déjà Vu system[10] asks users to select a subset of images from a collection of random art images. Weinshall, et al., have an interface that shows users 100-200 sets of image where the user password consists of selecting a predetermined image from each of the sets[32]. Systems like Passfaces take advantage of users’ innate ability to recall faces to improve recall[8]. Some systems use thumbnails generated from a single image rather than utilizing image sets[16]. Shoulder resistant schemes for graphical password systems have been developed as well. One such interface asks users to click within a convex hull formed by the objects consisting of their passwords[28]. Another shoulder-surfing resistant strategy includes using an eye tracker to determine the sequence of images the user looked upon[20]. Drawmetric graphical password systems require users to reproduce a shared secret to authenticate. One such technique has the user draw a secret on a 2-D grid and reproduce it for authentication. Other similar techniques include presenting a signature provided by either a stylus[14] or mouse[17]. Locimetric graphical passwords have users repeat a sequence of actions. These systems have users click on a series of interesting and meaningful points in an image in a predetermined sequence[24, 33]. Builders of these systems argue that a large password space can be constructed from a single image since images can contain hundreds of memorable points.

9 Conclusion

In this paper, we presented PhorceField, a password ceremony designed to depend on human laziness rather than vigilance. PhorceField uses a client-side secret and graphical password scheme to make it effectively impossible for a user to provide her password to a phisher. As long as the phisher does not know the client-side secret, he can only present the user with a non-standard and difficult-to-use password interface. We also designed PhorceField to exploit weaknesses of human memory to make phishing attacks even less likely to succeed.

We conducted a user study to verify that users will be unable to comply with a phisher's requests. No participant in our study successfully entered his entire password into our phishing web page, and the vast majority of participants revealed less than half of their password.

References

- [1] Saeed Abu-Nimeh, Dario Nappa, Xinlei Wang, and Suku Nair. A comparison of machine learning techniques for phishing detection. In *eCrime '07: Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*, pages 60–69, New York, NY, USA, 2007. ACM.
- [2] M. Aburrous, M.A. Hossain, F. Thabatah, and K. Dahal. Intelligent phishing website detection system using fuzzy techniques. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on*, pages 1–6, April 2008.
- [3] M. C. Anderson and J. H. Neely. *Memory. Handbook of Perception and Cognition*, chapter Interference and inhibition in memory retrieval. Academic Press, 2nd edition, 237–313 1996.
- [4] Bank of America. SiteKey: Online Banking Security. <http://www.bankofamerica.com/privacy/sitekey/>.
- [5] Mihir Bellare, David Pointcheval, and Phillip Rogaway. *Authenticated Key Exchange Secure against Dictionary Attacks*, pages 139–155. Springer, 2000.
- [6] Sonia Chiasson, P. C. van Oorschot, and Robert Biddle. A usability study and critique of two password managers. In *USENIX Security '06: Proceedings of the 15th conference on USENIX Security Symposium*, Berkeley, CA, USA, 2006. USENIX Association.
- [7] Neil Chou, Robert Ledesma, Yuka Teraguchi, Dan Boneh, and John C. Mitchell. Client-side defense against web-based identity theft. In *NDSS '04: Proceedings of the 11th Annual Network and Distributed System Security Symposium*, February 2004.
- [8] Real User Corporation. The Science Behind Passfaces. Technical report, Real User Corporation, June 2004.
- [9] Kenneth A. Deffenbacher, Thomas H. Carr, and John R. Leu. Memory for words, pictures, and faces: Retroactive interference, forgetting, and reminiscence. *Journal of Experimental Psychology: Human Learning and Memory*, 7(4):299–305, 1981.
- [10] Rachna Dhamija and Adrian Perrig. Déjà vu: a user study using images for authentication. In *SSYM'00: Proceedings of the 9th conference on USENIX Security Symposium*, pages 4–4, Berkeley, CA, USA, 2000. USENIX Association.
- [11] Rachna Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM.
- [12] John Eng. Sample size estimation: how many individuals should be studied? *Radiology*, 227(3):309–313, 2003.
- [13] GeoTrust Inc. <http://www.geotrust.com/comcasttoolbar/>.

- [14] Joseph Goldberg, Jennifer Hagman, and Vibha Sazawal. Doodling our way to better authentication. In *CHI '02: CHI '02 extended abstracts on Human factors in computing systems*, pages 868–869, New York, NY, USA, 2002. ACM.
- [15] Rick Hodgins. Phishing cost the u.s. \$3.2 billion in 2007. <http://www.tomshardware.com/news/phishing-cost-u-s-3-2-billion-2007,4576.html>, December 2007.
- [16] Wayne Jansen. Authenticating mobile device users through image selection, May 2004.
- [17] Ian Jermyn, Alain Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin. The design and analysis of graphical passwords. In *SSYM'99: Proceedings of the 8th conference on USENIX Security Symposium*, pages 1–1, Berkeley, CA, USA, 1999. USENIX Association.
- [18] Chris Karlof, J.D. Tygar, and David Wagner. Conditioned-safe Ceremonies and a User Study of an Application to Web Authentication. In *Sixteenth Annual Network and Distributed Systems Security Symposium (NDSS 2009)*, February 2009.
- [19] Brian Krebs. Citibank phish spoofs 2-factor authentication. http://blog.washingtonpost.com/securityfix/2006/07/citibank_phish_spoofs_2factor_1.html, July 2006.
- [20] Manu Kumar, Tal Garfinkel, Dan Boneh, and Terry Winograd. Reducing shoulder-surfing by using gaze-based password entry. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 13–19, New York, NY, USA, 2007. ACM.
- [21] Wendy Moncur and Grégory Leplâtre. Pictures at the atm: exploring the usability of multiple graphical passwords. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 887–894, New York, NY, USA, 2007. ACM.
- [22] Tyler Moore and Richard Clayton. An empirical analysis of the current state of phishing attack and defence. In *In Proceedings of the 2007 Workshop on the Economics of Information Security (WEIS)*, 2007.
- [23] Netcraft. Anti-phishing toolbar. <http://toolbar.netcraft.com/>.
- [24] L.D. Paulson. Taking a graphical approach to the password. *Computer*, 35(7):19–19, Jul 2002.
- [25] Matt Richtel and Verne G. Kopytoff. E-mail fraud hides behind friendly face. *The New York Times*, 2011.
- [26] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization: Papers from the 1998 Workshop*, Madison, Wisconsin, 1998. AAAI Technical Report WS-98-05.
- [27] Stuart E. Schechter, Rachna Dhamija, Andy Ozment, and Ian Fischer. The emperor's new security indicators. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 51–65, Washington, DC, USA, 2007. IEEE Computer Society.
- [28] Leonardo Sobrado and Jean-Camille Birget. Graphical passwords. In *The Rutgers Scholar, An Electronic Bulletin of Undergraduate Research*, volume 4, 2002.
- [29] Xiaoyuan Suo, Ying Zhu, and G. Scott. Owen. Graphical passwords: A survey. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 463–472, Washington, DC, USA, 2005. IEEE Computer Society.
- [30] Michelle Thorn. Analysis of 100m cc-licensed images on flickr. <http://creativecommons.org/weblog/entry/13588>, March 2009.
- [31] Yue Wang, R. Agrawal, and Baek-Young Choi. Light weight anti-phishing with user whitelisting in a web browser. In *Proceedings of the 2008 IEEE Region 5 Conference*, pages 1–4, April 2008.
- [32] Daphna Weinshall and Scott Kirkpatrick. Passwords you'll never forget, but can't recall. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, pages 1399–1402, New York, NY, USA, 2004. ACM.

- [33] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir Memon. Authentication using graphical passwords: Basic results. In *Human-Computer Interaction International (HCII 2005)*, New York, NY, USA, 2005. Springer.
- [34] Yahoo! Welcome to flickr. <http://flickr.com/>.
- [35] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong. Phinding phish: Evaluating anti-phishing tools. In *NDSS '07: Proceedings of the 14th Annual Network and Distributed System Security Symposium*, February 2007.