

Training Johnny to Authenticate (Safely)

Amir Herzberg and Ronen Margulies | Bar-Ilan University

An adaptive authentication mechanism based on image recognition and negative training conditions users to log in safely and increases attack detection rates.

Phishing—password theft via fake websites—is an extremely worrying, widespread phenomenon. With billions of dollars lost and a large increase in the amount of attacks, it's clear that today's defenses aren't adequate.

It's important to understand what makes users so susceptible to phishing and the psychological explanations for this phenomenon. In “Conditioned-Safe Ceremonies and a User Study of an Application to Web Authentication,” Chris Karlof and colleagues described how humans tend to develop *click whirr responses*—automatic responses to repeating situations.¹ Users have developed a click whirr response to login forms, automatically submitting their credentials to a login form on a familiar interface. Most users also follow email links from familiar senders and trust familiar homepages, even if not protected by SSL, and navigate to the site's login page (for instance, by clicking an Enter Your Account button). When following those links and buttons, users might reach a spoofed login page. The three click whirr responses we mention make the Internet a fertile ground for phishing attacks.

To prevent these attacks, we created and tested a site-based login mechanism that trains users to log in safely.

prevent phishing attacks, including passive and interactive indicators (see Figure 1) and bookmark tokens.

Passive Defense Indicators

Early Web browsers include three main indicators to help users identify the websites they visit: the address bar (indicating the site's URL), the https prefix, and the padlock or key image (indicating SSL usage). These indicators are ineffective in providing security because users can successfully log in without checking the indicators and it's easier to skip checking them. Indeed, several experiments showed that users usually enter their passwords without validating these indicators.^{2,3}

Researchers introduced several enhanced indicators—most involving a change to the browser and a few that require only website support. Those indicators display warnings, a custom image or text, and SSL certificate information and emphasize the domain name and protocol in the URL bar. Most indicators developed and deployed so far are passive, requiring no user action. Several experiments measuring users' ability to detect fake sites under different attack scenarios with different passive indicators resulted in mostly disappointing results.^{4–6}

Phishing Defense Mechanisms

Researchers have proposed a variety of methods to

Interactive Custom Indicators

Interactive custom indicators require users to click a

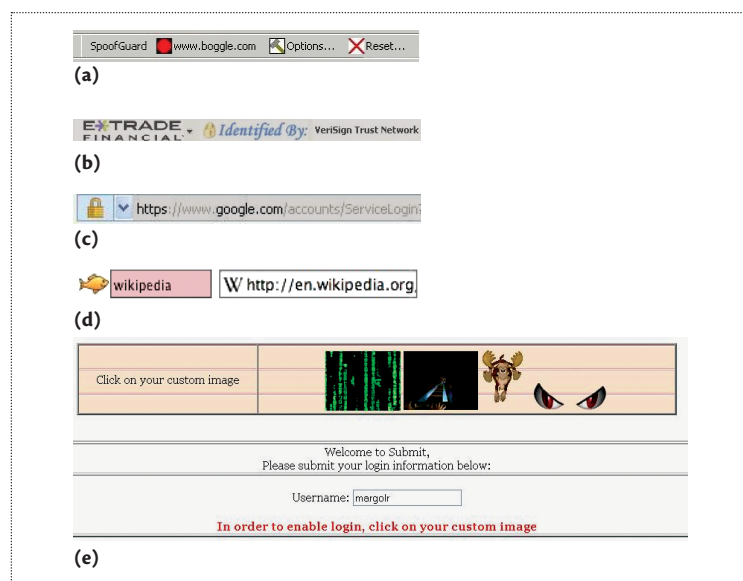


Figure 1. Examples of passive and interactive indicators. (a) The SpoofGuard browser extension displays a warning of a spoofed URL. (b) The Trustbar toolbar displays custom image and certificate authority information from the site's certificate. (c) Internet Explorer 8 emphasizes the domain name and protocol (https). (d) Passpet shows an interactive custom pet image. (e) A site-based interactive login requires users to click the correct image out of four images.

preselected custom image or text to log in. An example is Passpet—a Firefox extension that acts both as a password manager and an interactive custom indicator.⁷ When users click the correct image, Passpet automatically sends their password.

A website can also implement interactive custom indicators by hiding the password text field until users click the custom indicator. The interactive nature of these indicators, which conditions users to find and click them, makes users more alert to the indicator's absence on a spoofed login page.

Secure Login Using a Bookmark

A phishing attack's initial stage is to lure users to a spoofed login page. A common scenario is to send a spoofed email containing a link to this page. In addition, most sites' homepages aren't protected by SSL and can be spoofed by a man-in-the-middle attack. Users might put their trust in a spoofed page that looks similar to their target site's homepage and follow its Enter Your Account button leading to the spoofed login page.

A good habit for accessing high-value sites is to create a browser bookmark ("favorite") for a sensitive site's (https) login page—which ensures reaching the correct URL over a secure channel—and always access the site's login page via that

bookmark. Ben Adida presented BeamAuth, a two-factor authentication mechanism based on a login bookmark.⁸ The website provides users a special login bookmark containing a secret token that identifies them to the site. To ensure users always log in via their bookmark, the login page looks for the secret token and prohibits login if a valid token isn't supplied (an error message appears telling users to enter the login page via the bookmark). Users would become accustomed to navigating to the login page via the bookmark only and ideally wouldn't, for example, follow email links.

Alternative Authentication Techniques

A major problem in security is the fact that Internet users have online accounts to many websites, systems, and devices that require them to remember passwords for identification. Because users can only remember a limited number of passwords, many simply forget them. Two strategies attempt to prevent this:

- automatic fallback authentication techniques, for when users forget their password, and
- memorable substitutes for textual passwords that aim to prevent users from forgetting their identifying information.

We review existing mechanisms for fallback authentication, as well as graphical passwords, which were originally suggested as a substitute for textual passwords but are actually more suitable for fallback authentication.

Fallback Authentication Mechanisms

Today's most common mechanisms for automatic password recovery are email-based mechanisms and security questions. Email-based mechanisms require a preconfigured email address for the site to resend the forgotten password or a link to a password reset page. Security questions—the answers to which are established during registration—are used as identifying information and are supposed to be known only to the user.

With email-based password reset mechanisms, attackers breaking into a user's email account can immediately reset the password. Security questions suffer from a few problems. First, they're susceptible to phishing attacks because, like login forms, they stimulate an automatic response, causing users to answer the questions immediately. Karlof and colleagues found a 92.7 percent spoof rate in their user study of security questions.¹ In addition, other studies showed that many answers are forgotten or easily guessed owing to inherently low entropy.⁹

Graphical Passwords

Graphical passwords—a general name for methods in which images are used for authentication—were originally proposed as an alternative to textual passwords owing to increased memorability and ease of use on touchscreen devices. Because of their rather long authentication times (approximately 30 seconds in Rachna Dhamija and Adrian Perrig’s experiment¹⁰), they weren’t widely deployed as a substitute for textual passwords and are better suited for fallback authentication.

Markus Jakobsson and colleagues suggested a fallback authentication technique that requires users to recognize items (images or text phrases) they like or dislike.¹¹ However, this and similar proposals encounter memorability problems because recognizing a large set of items (their security analysis showed that 16 items are necessary) after a long time period without seeing them is difficult.

Owner-to-Mobile Authentication

With the increased popularity of mobile devices and the extent of private information they hold, such as credentials to many sites and applications, authentication to mobile devices is important. Most of today’s mobile devices lock themselves after one or two minutes without user activity; users can choose whether to require authentication to unlock them. The most common mechanisms for owner-to-mobile authentication are PIN codes and lock patterns, which can be forgotten and are rather slow for frequently performed authentication actions.

Long-Term User Study

We conducted an extensive long-term study of real-life Web and email activities, which included different kinds of simulated phishing attacks. We examined the effectiveness of different phishing defense mechanisms, including a login bookmark, interactive custom images, and a combination of the two. We also tested mechanisms that sites can use to prevent users from reaching spoofed login pages by intentionally including nonworking links and buttons on the site’s homepage and email announcements.

Most previous work in the field consisted of short-term lab studies (lasting a few hours)^{4–6} in which users might be more or less cautious than in real life.¹² To the best of our knowledge, there was no prior long-term user study examining users’ responses to emails or the detection rates of spoofed pages.

Antiphishing Mechanisms

Our study used training functions to help users perform safer logins. Karlof and colleagues introduced

the notion of a *conditioned-safe ceremony*, a process that forces users to take safe actions.¹ *Forcing functions* are behavior-shaping constraints that could help in achieving a conditioned-safe ceremony by preventing a task’s progress until users perform a specific action whose omission results in failure. Because users must take this action during every instance of the task, they’re trained to always perform this action, and soon it becomes a click whirr response.

To defeat conditioned-safe ceremonies, attackers try to make users perform a dangerous action instead of the forcing function, thus bypassing their protection (for example, convincing users to follow a link to a spoofed login page instead of clicking the login bookmark). Because such actions are indeed possible, we introduce *negative training functions*. Unlike forcing functions, negative training functions aren’t part of the login ceremony and constantly train users to avoid dangerous actions by making them experience failure when performing those actions. Negative training functions can work with forcing functions to better train users on what they should and shouldn’t do.

Because humans tend to make routine actions mindlessly and, in particular, skip any voluntary actions during a login ceremony, we shouldn’t fight this tendency. A conditioned-safe login ceremony should consist of several forcing and negative training functions—at least one function against each click whirr response that puts users in danger. In particular, there should be forcing and negative training functions against automatic following of links, automatic submission of credentials, and automatic entrance to a site’s login page by clicking an Enter Your Account button on the homepage.

The forcing function mechanisms we used were

- a login bookmark that forces users to log in via their bookmark only (other attempts result in failure), and
- an interactive custom image that forces users to find and click their custom image before submitting their password.

The login bookmark defends against reaching a spoofed login page, and the interactive custom image defends against automatic submission of credentials. Combining the two mechanisms provides both prevention and detection of spoofed login pages, thus achieving defense in depth and guaranteeing that omitting one of them won’t be enough for a successful phishing attack. Therefore, we also tested their combination.

In the combined method, the site initially recognizes users with the secret token sent after the bookmark click and then displays their custom image. The login ceremony is simpler and faster than typical login

ceremonies, as it requires only two mouse clicks (for the bookmark and interactive image) and typing only the password. (By using a fragment identifier to store both the secret token and the username in the bookmark link, as Adida suggested,⁸ a JavaScript can read the contents of the fragment identifier and automatically fill in the username in the appropriate text box.) This saves the need for typing or searching for the site's homepage URL, navigating to its login page, and typing both the username and password. The combined method's login ceremony is faster than the typical ceremony even with browser autocompletion.

A variation of interactive custom images is to choose the custom image from a small set of random images on the login page. This makes users even more aware of their custom image and improves detection rates. With only one image displayed, users' click whirr response to click an image could cause users to immediately click any displayed image, even a fake one. Making them choose the correct image reduces the chance for an immediate, mindless click.

We also tested two negative training functions to prevent users from reaching spoofed login pages:

- We intentionally included nonworking links to the login page in the site's email announcements. When users clicked it, they reached the login page, which displayed an error telling them to log in only via their bookmark. This experience trained users to never follow links.
- We intentionally included a nonworking account-entrance button on the site's homepage. When users clicked it, they reached the login page, which displayed a similar error message. This experience trained users to never enter the site's homepage when wanting to log in.

These negative training mechanisms might cause undesired user interpretation (such as thinking the site is untrustworthy or contains bugs) and should be considered in contexts other than security and usability (for instance, marketing and perception). However, we believe that by providing an instructive error message that explains that the site's purpose is to protect users from taking hazardous actions, the mechanisms will be accepted by most users and might even gain their trust and respect. In addition, the mechanisms were shown to be effective and reduced the amount of links being clicked, thus reducing possible user misinterpretation.

Framework System and Users

For our study, we used an online exercise submission system called Submit (submit.cs.biu.ac.il), which is

used in most Bar-Ilan University computer science department courses. Students use the system to submit exercises and receive emails announcing new grades. Most logged in to the system dozens to hundreds of times throughout the study.

We simulated several phishing attacks and collected the attacks' results. We used low attack frequencies to prevent users from being overly suspicious and to make them act as in real life. Even though they were aware of the study's purpose, we didn't expect their motivation to detect attacks to be higher than usual owing to their constant use of the system, unlike in short-term lab studies.

We used the system for four semesters, with a population of approximately 400 students (the entire computer science department). On the basis of the two surveys we conducted, approximately one-third of the students had no understanding of phishing and Web security or didn't understand the meaning of their defense mechanisms despite the instructions we provided, and many (one-fifth) had little understanding. (For further details, see "Usable and Phishing-Resistant Authentication Mechanisms."¹³)

Users' Login Methods

During registration, we randomly assigned each user one of five login methods:

- image only—an interactive custom image only;
- bookmark only—a login bookmark only;
- bookmark + image—a login bookmark combined with an interactive custom image;
- bookmark + four images—same as the bookmark + image, but the login page displays three additional noncustom images; and
- none—no site-based indicator assigned, used as a control group.

Users could only log in with the method assigned to them during registration. In normal system use, nonbookmark users (none and image only) reached the https login page via the system's http homepage by providing their username. Bookmark users (bookmark only, bookmark + image, and bookmark + four images) who entered the login page the same way received an error message telling them to log in via their bookmark (except when attacked)—our second negative training function.

Email Methods

Users were also assigned an email method, which determined how they got their grade announcement emails from the system. We used three types of emails:

Table 1. Detection rates and overall resistance rates for a classic phishing attack.*

Mechanisms	Detection rate (%)	Overall resistance rate (%)
None	19.61 ± 4.95	40.22 ± 10.24
Bookmark only	42.56 ± 5.61	49.84 ± 14.77
Bookmark + nonworking links		81.08 ± 12.88
Image only	59.84 ± 6.24	76.12 ± 8.44
Bookmark + image	72.71 ± 6.31	85.19 ± 6.93
Bookmark + four images	81.94 ± 5.17	
Bookmark + image(s)	77.55 ± 4.06	80 ± 10.03
Bookmark + image(s) + nonworking links		93.24 ± 7.8

*We merged results when it didn't make sense to split them (for instance, nonworking links don't affect the detection rates, only the prevention rates) or when we needed higher confidence level. Bookmark + image(s) combines bookmark + image with bookmark + four images.

- link—emails that contained a link to the system's login page telling users they have to log in to view their grade and submission details;
- no link—emails that contained the grade and submission details in the mail body and contained no link; and
- warning—emails like the latter containing no link and also containing a warning saying that the system never includes links in its emails because clicking email links is dangerous.

Users always got the same type of email except when the system sent a spoofed email. Bookmark users from the link group received nonworking links regularly (except when attacked), which directed them to the system's login page where they were shown an error message—our first negative training function.

Experiment Details

We introduced our study by displaying an instruction page that briefly described what phishing is and the extent of phishing attacks, who we are and what our goals were, and the experiment details. We asked the students to cooperate and promised our gratitude. To provide an incentive to cooperate, we announced a bonus of up to five points in one of Herzberg's courses for correctly detecting attacks.

We told the students that they might encounter a fake login page and asked them not to provide their credentials when they believed they reached a spoofed page. We asked them to click a Report Phishing Page button at the top right corner of the login page; the same button was also displayed on all the spoofed login pages. Though such a button doesn't exist on everyday sites, the long-term use of the system and the fact that the button is passive and rarely used makes it likely to be ignored most of the time. Therefore, we believe

this button doesn't affect users' behavior in long-term use of the system. We considered an attack successful when users provided credentials to a spoofed login page, and unsuccessful when they clicked the report button or disconnected without trying to authenticate. Clicking the report button on the genuine login page was considered a false positive. Users who submitted their credentials to a spoofed page were automatically logged in and didn't receive any indication of the attack.

When users tried to enter the system's login page via the homepage or via their bookmark, there was a low probability of them being randomly directed to one of the system's spoofed login pages. Again, we used low attack probabilities to prevent increased awareness due to frequent attacks. Spoofed emails were also sent to users with rather low probabilities; these emails contained similar content to the system's genuine emails, apart from the fact that the links contained a spoofed page's URL. For image users, the spoofed pages contained no images, broken images, or fake images. (For further details, see "Usable and Phishing-Resistant Authentication Mechanisms."¹³)

Study Results and Conclusions

We found significant differences between the different mechanisms in both detection rates and overall resistance rates (which measure the percentage of unsuccessful attacks, whether they were prevented or detected; see Table 1). By combining a login bookmark with nonworking email links and an interactive custom image and displaying multiple images on the login page, we achieved detection rates and overall resistance rates higher than any other mechanism previously tested in real life (82 percent and 93 percent, respectively, compared to 20 percent and 40 percent for the typical login ceremony).

Table 2. Link following rates for bookmark and nonbookmark users.

Method	Email method	Followed	Sent	Following rate (%)
Nonbookmark	No link	50	69	72.46 ± 8.84
Nonbookmark	Warning	68	100	68 ± 7.67
Nonbookmark	Working link	95	139	68.34 ± 6.49
Bookmark	No link	53	81	65.43 ± 8.69
Bookmark	Warning	74	97	76.29 ± 7.1
Bookmark	Nonworking link	36	116	31.03 ± 7.06

Detection Rates

Regarding spoofed pages that were entered, not including emails with spoofed links that weren't clicked, there was a significant gap in the detection rates of the different methods ranging from 20 percent for none users, 42 percent for bookmark only, 60 percent for image only, 73 percent for bookmark + image, and 82 percent for bookmark + four images.

In particular, image users (image only, bookmark + image, and bookmark + four images) detected more than twice as many attacks as nonimage users (none and bookmark). Therefore, the interactive custom image is a highly effective forcing function against automatic submission of credentials.

Users' Response to Emails

Table 2 shows the spoofed link following rates for bookmark and nonbookmark users with regard to the different email methods. First, we can see that there's no significant difference between no link and warning users. Warnings against following links in legitimate emails didn't prevent users from following links in spoofed emails. Our results for email warnings correlate with the results of Karlof and colleagues' study.¹

Now, let's focus on nonbookmark users. The legitimate emails of nonbookmark users who normally received no links (their grades appeared in the email body) looked entirely different from the spoofed emails (which didn't contain grades and asked users to follow a link to retrieve them). Despite the difference, the percentage of links followed by those users was as high as nonbookmark link users, whose legitimate and spoofed emails looked similar (both contained working links). From this, we conclude that users don't distinguish between spoofed and nonspoofed emails, even if the email's structure is unfamiliar. This can be explained by the fact that clicking a link is a click whirr response. Users might also put too much trust in an email's From header, which can be spoofed easily, or think that the system has changed its email announcements' structure.

Finally, and most important, the results show that bookmark users who normally received no links followed a similar percentage of spoofed links as nonbookmark users. Only bookmark users who normally received nonworking links followed less than half the spoofed links followed by nonbookmark users. We found a similar significant difference for nonspoofed emails. Therefore, the login bookmark was only effective against automatic following of links when users received nonworking links in genuine emails and experienced failure in reaching the login page by following a link. Sites should consider intentionally including nonworking links in their email announcements to constantly train their users not to follow email links. (Even if the site doesn't normally send email announcements, we advise sending some from time to time just for training.)

Spoofed Homepage Attacks

Here, we focus on the spoofed homepage attack, which led users to a spoofed login page. For all methods, we noticed that detection rates were lower when users entered spoofed login pages from the system's homepage compared to email links or bookmark clicks (17 percent versus 26 percent for none users, 18 percent versus 46 percent for bookmark only, 58 percent versus 63 percent for image only, 57 percent versus 74 percent for image + bookmark, and 74 percent versus 83 percent for bookmark + four images). A possible explanation is that users put high trust in the homepage of a familiar-looking site, even if it doesn't provide an SSL certificate.

With reduced detection rates in this attack, prevention is even more important. The vast majority of all bookmark users tried to enter the site's login page via its homepage, despite their bookmark. To prevent them from doing so, the site can choose not to include an account-entrance button on its homepage, or to include a nonworking button that leads the user to an error page. We only tested the second option; a closer look at the attack log showed that, despite the rather

high attack probability we used, only two bookmark users were attacked more than once. Thus, combining the login bookmark with a nonworking account-entrance button in the site's homepage achieved effective prevention. Sites should consider intentionally including a nonworking account-entrance button or link on their homepage.

Both options let users experience failure when trying to navigate to the login page via the homepage and train users not to enter the homepage in the first place when wanting to log in. Yet, when users are triggered to enter a spoofed page that looks similar to the target site's homepage and includes an account-entrance button, they might be tempted to click it because they didn't experience failure in this specific action. The vast majority of bookmark users did click the account-entrance button in our study, and by doing so, they experienced failure.

Usability Survey

Because users are forced to set the bookmark (or, alternatively, an authentication cookie) on each computer or browser they use, this process should be secure and usable. We used a registration email containing the bookmark link, which ensures a secondary channel and allows installation on multiple computers by keeping the registration email. Though bookmark setup isn't as fast as the login ceremony, we believe most users usually access high-value sites from only a few computers.

We tested this assumption and users' overall impression in a survey at the end of our study. The survey's results confirmed our assumption—users enter high-value sites from only 1.75 computers on average and enter medium-values sites, such as social networks or webmail, from 3.26 computers on average. Most users (72 percent) mentioned that they would want to use a login bookmark to access high-value sites, and half the users (51 percent) would also use it on medium-value sites. We can conclude that when considering their security pros and usability cons, most users are willing to use login bookmarks on sensitive sites.

Learn-by-Use Graphical Password

Our phishing experiment showed that interactive custom images are very effective at training, so we looked for additional authentication uses for interactive custom images. Our learn-by-use mechanism applies to several Web and mobile authentication scenarios—in particular, fallback authentication. The mechanism relies on memorization of one custom image on each primary login and recognition of several images for fallback authentication or when suspecting impersonation attacks.

Mechanism Description

During registration, users choose (or are assigned) a small number of custom images (k) from a large collection of images.

In single-stage (weak) authentication, the system displays a small set of L images, one of which is one of the user's k custom images, and the user must click his or her custom image to log in. After each successful login, a new set of L images is chosen randomly for the next login. Because users' custom images are displayed and clicked on each login, the primary login ceremony constantly refreshes users' memory of their custom images, so they're better remembered when fallback authentication is needed. Single-stage authentication provides weak authentication ($1/L$ guessing probability) and will often be used in conjunction with other identifiers (passwords, PIN codes, possession of mobile device). In addition, our mechanism increases its authentication difficulty adaptively when suspecting impersonation attacks, thus the weak authentication can rapidly become strong.

In multiple-stage authentication, which is used for fallback authentication or upon suspecting attacks, users are shown a series of k stages. Each stage displays n images, only one of which is the custom image, and users must correctly click most or all of their custom images. If users choose one or more wrong images, the ceremony continues until the k th click and only then announces an unsuccessful authentication attempt.

Detecting and Handling Impersonation Attacks

Because one of the user's custom images is displayed on the primary login page, an attacker can find that image simply by accessing the primary login page (for both mobile and Web) and providing an initial identifier (Web only). After each successful login, the system changes the custom image used. Therefore, attackers could constantly monitor the login page, checking if any new images are displayed until they find most or all of the user's custom images.

An impersonation attack includes both a monitoring attack and guessing attacks of users' passwords or custom images. To detect such attacks, the system can notice when a user reaches the login page and doesn't try to authenticate at all or incorrectly guesses the user's image (out of L), especially if this happened several times in a predetermined time period. Our mechanism can also detect guessing attacks by monitoring failed authentication attempts, in which the user submits a wrong password or clicks incorrect images on the fallback authentication page.

When the system suspects an attack, it increases the authentication difficulty adaptively to prevent attackers from gaining more of the user's custom images

or applying guessing attacks. First, the system can increase the number of images displayed on the login page (for example, double them) after reaching some certainty threshold. Second, the system can apply the multiple-stage scheme for increased difficulty. Third, it can increase the amount of stages to $m > k$ stages, each containing a “none of my images are here” choice, and only k of the stages display a user-custom image. As the difficulty increases, attackers are likely to make more incorrect clicks or disconnect without trying to authenticate, which increases the attack’s certainty.

After a preconfigured level of certainty is reached, the system can ask users (after they successfully log in) to replace their secret token and, by logging the images that the suspected attacker correctly clicked, can also ask users to replace those images. “Usable and Phishing-Resistant Authentication Mechanisms”¹³ provides an algorithm for detecting and handling impersonation attacks, with varying levels of certainty, by using all the previously mentioned measures.

Applications

Our learn-by-use mechanism’s applications can be tailored to high- and medium-value sites and mobile devices.

Authentication to high-value sites. This application uses our learn-by-use authentication mechanism in conjunction with the forcing and negative training functions we discussed earlier. Users have k custom images instead of just one, and after clicking the bookmark and reaching the login page (and providing the secret token), L images are displayed, one of which is one of the custom images. In addition to the site identification, the image click provides users with the necessary memorization. After clicking the custom image, users can choose to submit their password or go to the fallback k -stages authentication page.

In a 10-week study, we found that 84 percent of users who entered the system more than 20 times successfully identified all their k custom images (k was 4 or 5), and 95 percent had one or no mistake.¹³ In addition, there was no degradation in the detection rates of spoofed pages due to the use of several custom images. Authentication times were similar to Dhamija and Perrig’s experiment—approximately 30 seconds.¹⁰

Authentication to medium-value sites. Most of today’s sites don’t require two-factor authentication—many require a password or a previously installed cookie (containing an authentication token). Our mechanism can also be used on such sites:

- When an authentication cookie is already installed,

the site can display one custom image, together with several noncustom images, which provides the necessary memorization and further security (when passwords aren’t required). The mechanism can be set to allow no mistakes, double the amount of images immediately after the first mistaken click, and rapidly increase the difficulty on further mistakes. If passwords are also required, the mechanism provides phishing protection.

- When an authentication cookie isn’t installed, the site should ask for the password and only then display the images. This doesn’t provide phishing protection but does provide the necessary memorization and weak (and increasing) security.
- If users forget their password, the site can let them authenticate using the k -stages authentication by providing their username only.

When used on medium-value sites, our mechanism provides the necessary memorization, increases authentication strength, and can provide best-effort phishing protection.

Authentication to mobile devices. For many mobile device users, an initial weak authentication is enough, as the mere possession of the device provides additional identification. In addition, the frequent use obligates the authentication ceremony to be quick. A mobile device can initially display several (L) images, double this amount upon a single mistaken click, and rapidly increase the difficulty upon more mistaken clicks.

For users who want stronger security, the initial difficulty can be higher (for example, two stages of L images), and the mechanism can also be used in conjunction with a PIN code, a lock pattern, or biometrics (such as automatic voice or gait recognition¹⁴), thus providing complementary authentication and memorization. If users forget their PIN or pattern or biometrics fails, k -stages authentication can be used.

We implemented all the mechanisms we investigated, including the fallback Web authentication, in our server-side solution called WAPP (Web Application Phishing Protection). A demo written in PHP is available at <http://submit2.cs.biu.ac.il/WAPP>. ■

Acknowledgments

We thank Ben Adida for his feedback and helpful suggestions. This work was supported by Israeli Science Foundation grant ISF1014/07.

References


1. C. Karlof, J.D. Tygar, and D. Wagner, "Conditioned-Safe Ceremonies and a User Study of an Application to Web Authentication," *Proc. 5th Symp. Usable Privacy and Security (SOUPS 09)*, ACM, 2009.
2. R. Dhamija, J.D. Tygar, and M. Hearst, "Why Phishing Works," *Proc. SIGCHI Conf. Human Factors in Computing Systems*, ACM, 2006, pp. 581–590.
3. A. Herzberg and A. Jbara, "Security and Identification Indicators for Browsers against Spoofing and Phishing Attacks," *ACM Trans. Internet Technology*, vol. 8, no. 4, art. 16, 2008; <http://doi.acm.org/10.1145/1391949.1391950>.
4. M. Wu, R.C. Miller, and S.L. Garfinkel, "Do Security Toolbars Actually Prevent Phishing Attacks?," *Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 06)*, ACM, 2006, pp. 601–610.
5. A. Herzberg, "Why Johnny Can't Surf (Safely)? Attacks and Defenses for Web Users," *Computers & Security*, vol. 28, nos. 1–2, 2009, pp. 63–71.
6. S. Schechter et al., "The Emperor's New Security Indicators," *Proc. 2007 IEEE Symp. Security and Privacy (SP 07)*, IEEE CS, 2007, pp. 51–65.
7. K.P. Yee and K. Sitaker, "Passpet: Convenient Password Management and Phishing Protection," *Proc. 2nd Symp. Usable Privacy and Security*, ACM, 2006, pp. 32–43.
8. B. Adida, "BeamAuth: Two-Factor Web Authentication with a Bookmark," *Proc. 14th ACM Conf. Computer and Comm. Security (CSS 07)*, ACM, 2007, pp. 48–57.
9. S. Schechter, A.J.B. Brush, and S. Egelman, "It's No Secret. Measuring the Security and Reliability of Authentication via 'Secret' Questions," *Proc. 2009 IEEE Symp. Security and Privacy (SP 09)*, IEEE CS, 2009, pp. 375–390.
10. R. Dhamija and A. Perrig, "Déjà Vu: A User Study Using Images for Authentication," *Proc. 9th Conf. Usenix Security Symp.*, vol. 9, Usenix, 2000, p. 4.
11. M. Jakobsson, L. Yang, and S. Wetzel, "Quantifying the Security of Preference-Based Authentication," *Proc. 4th ACM Workshop on Digital Identity Management (DIM 08)*, ACM, 2008, pp. 61–70.
12. A. Sotirakopoulos, K. Hawkey, and K. Beznosov, "I Did It Because I Trusted You: Challenges with the Study Environment Biasing Participant Behaviours," *SOUPS User Workshop*, ACM, 2010; http://cups.cs.cmu.edu/soups/2010/user_papers/Sotirakopoulos_environment_biasing_participants_USER2010.pdf.
13. R. Margulies, "Usable and Phishing-Resistant Authentication Mechanisms," master's thesis, Computer Science Dept., Bar-Ilan Univ., 2011.
14. M. Boyle et al., "Toward Gait-Based Smartphone User Identification," *Proc. 9th Ann. Int'l Conf. Mobile Systems, Applications, and Services*, ACM, 2011, pp. 395–396.

Amir Herzberg is a professor in Bar-Ilan University's computer science department. His research interests include security, reliability, and efficiency of networking and computing. Herzberg has a DSc in computer science from the Technion, Israel. Contact him at herzbea@cs.biu.ac.il.

Ronen Margulies is a student at Bar-Ilan University. His research interests include authentication techniques, antiphishing, content filtering, protocols security, and secure system architectures. Margulies has an MSc in computer science from Bar-Ilan University. Contact him at maronen1@gmail.com.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



LISTEN TO GRADY BOOCH
"On Architecture" Podcast
www.computer.org/onarchitecture