Web Wallet:
Preventing Phishing Attacks by Revealing User Intentions

Min Wu, Robert C. Miller, Greg Little
MIT Computer Science and Artificial Intelligence Lab
32 Vassar Street, Cambridge, MA 02139
{minwu, rcm, glittle} @ csail.mit.edu

ABSTRACT
We introduce a new anti-phishing solution, the Web Wallet. The Web Wallet is a browser sidebar which users can use to submit their sensitive information online. It detects phishing attacks by determining where users intend to submit their information and suggests an alternative safe path to their intended site if the current site does not match it. It integrates security questions into the user's workflow so that its protection cannot be ignored by the user. We conducted a user study on the Web Wallet prototype and found that the Web Wallet is a promising approach. In the study, it significantly decreased the spoof rate of typical phishing attacks from 63% to 7%, and it effectively prevented all phishing attacks as long as it was used. A majority of the subjects successfully learned to depend on the Web Wallet to submit their login information. However, the study also found that spoofing the Web Wallet interface itself was an effective attack. Moreover, it was not easy to completely stop all subjects from typing sensitive information directly into web forms.

Categories and Subject Descriptors
H.5.2 User Interfaces, H.1.2 User/Machine Systems, D.4.6 Security and Protection.

General Terms
Security, Human Factors, Design, Experimentation.

Keywords
World Wide Web and Hypermedia, E-Commerce, User Interface Design, User Study.

1. INTRODUCTION
Phishing has become a significant threat to Internet users. Phishing attacks typically use legitimate-looking but fake emails and websites to deceive users into disclosing private information to the attacker. Phishing keeps growing: according to the Anti-Phishing Working Group (APWG), 15244 unique phishing attacks and 7197 unique phishing sites were reported in December 2005, with 121 legitimate brands being hijacked. [2]

Most phishing attacks trick users into submitting their personal information using a web form. Even though using a web form to submit sensitive information is common practice on legitimate sites, it has a couple of problems that make phishing attacks effective and hard to prevent.

First, the appearance of a web site and its web forms are easy to spoof. A web site can control what it looks like in a user's browser, so a site's appearance does not reliably reflect the site's true identity. But users tend to decide site identity based on appearance, *e.g.*, "This site looks exactly like the PayPal site that I have been to before. So it must be a PayPal site." [26] As a result, users may be tricked into submitting data to phishing sites.

Second, web forms are used for submitting insensitive data as well as sensitive data. Even though SSL encryption can indicate to the browser that the input data is sensitive, phishing sites do not use SSL and the browser fails to effectively visually differentiate an SSL connection from a non-SSL one. Moreover the semantic meaning of the input data is opaque to the browser. Therefore, the browser fails to give appropriate protection to the sensitive data submission especially under phishing attacks.

Many proposed anti-phishing solutions use toolbars that show different types of security messages to help users to detect phishing sites. Users are also advised to look at the existing browser security indicators, *e.g.*, the URL displayed in the address bar and the lock icon displayed in the status bar when a connection is SSL-protected. However, controlled user studies have shown that these security indicators are ineffective against high-quality phishing attacks for several reasons: [26]

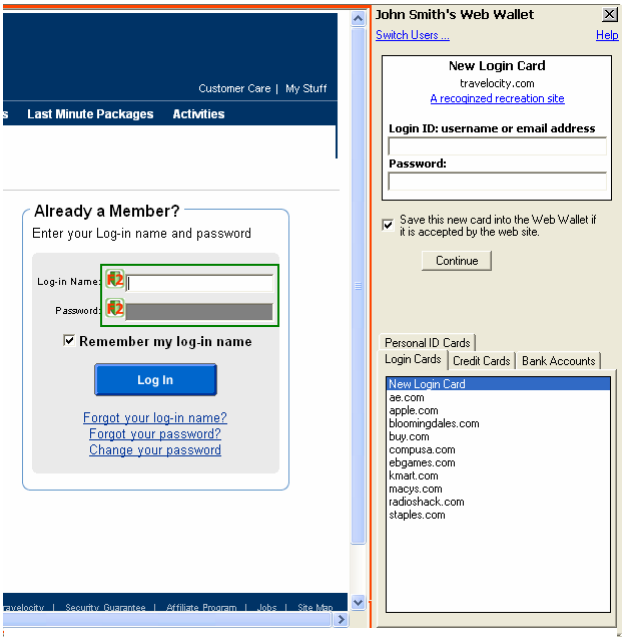


Figure 1. The Web Wallet in Internet Explorer

Symposium On Usable Privacy and Security (SOUPS) 2006, July 12-14, 2006, Pittsburgh, PA, USA.

First, warning indicators located in a peripheral area provide a much weaker signal than the centrally displayed web page and can be easily overwhelmed by convincing web content.

Second, the security-related information shown by the indicators is not really needed for the user's current task. Since security is rarely a user's primary goal, users fail to pay continuous attention to the indicators. Making security a separate task that users are required to remember is not an effective solution.

Third, sloppy but common web practices cause some users to rationalize the violation of the security rules that some indicators use to detect phishing attacks. For example, users are told to examine the hostname displayed in the address bar, to make sure that the hostname is the one they are expecting. But some legitimate websites use IP addresses instead of hostnames (*e.g.*, the Google cache) and some sites use domain names that are totally different from their brand names. Users are also told to find the SSL lock icon before submitting sensitive information. But many legitimate banks still use unprotected login pages. [12] Moreover, some indicators deliver warnings without detailed convincing explanations, which makes users think that the software is buggy and not treat the warning seriously.

Fourth, security indicators tend to show that something is wrong and advise users not to proceed, but they do not suggest good alternatives. This may encourage users to risk submitting their information anyway, since they don't see any other way to accomplish their goal.

1.1 The Web Wallet

To solve the problems that we have observed in controlled studies and in real life, we have designed a new solution, called the Web Wallet, to prevent phishing attacks. The main part of the Web Wallet is a browser sidebar for entering sensitive information (figure 1). When a user sees a web form requesting her sensitive data, she presses a dedicated security key on the keyboard to open the Web Wallet. Using the Web Wallet, she may type her data or retrieve her stored data. The data is then filled into the web form. But before the fill-in, the Web Wallet checks if the current site is good enough to receive the sensitive data. If the current site is not qualified, the Web Wallet requires the user to explicitly indicate where she wants the data to go. If the user's intended site is not the current site (which probably indicates phishing), the Web Wallet shows a warning to the user about this discrepancy, and gives her a safe path to her intended site.

There is one simple rule to correctly use the Web Wallet: "Always use the Web Wallet to submit sensitive information by pressing the security key first." Equivalently, "never submit sensitive information directly through a web form because it is not a secure practice."

We have run a user study to test the Web Wallet interface. The results are promising:

- The Web Wallet significantly decreased the spoof rate of normal phishing attacks from 63% to 7%.
- All the simulated phishing attacks in the study were effectively prevented by the Web Wallet as long as it was used.
- By disabling direct input into web forms and thus making itself the only way to input sensitive information, the Web Wallet successfully trained a majority of the subjects to use it to protect their sensitive information submission.

But there are also negative results which we plan to deal with in future research:

- The subjects totally failed to differentiate the authentic Web Wallet interface from a fake Web Wallet presented by a phishing site. This is a new type of phishing attack. Instead of mimicking a legitimate site's appearance, the attacker fakes the interface of security software that is run by the user.
- It is not easy to completely stop all subjects from typing sensitive information directly into web forms. Users are familiar with web form submission and have a strong tendency to use it.

The rest of the paper is organized as follows. Section 2 surveys other anti-phishing solutions. Section 3 introduces the design principles that drove the Web Wallet design. Section 4 describes the Web Wallet user interface. Section 5 introduces a user study that we ran to evaluate the Web Wallet. Section 6 presents the results from the study and discusses how well the Web Wallet prevents phishing attacks. Section 7 discusses the Web Wallet backend implementation. Finally, Section 8 presents conclusions and the future work.

2. RELATED WORK

In this section, we briefly survey existing anti-phishing solutions. A comprehensive survey of anti-phishing solutions can be found in [6].

One approach is to stop phishing at the email level (*e.g.*, [1]), since most current phishing attacks use broadcast email (spam) to lure victims to a phishing website.

Another approach is to use security toolbars. The phishing filter in IE7 [21] is a toolbar approach with more features such as blocking the user's activity with a detected phishing site. We discussed in the introduction why toolbars fail to effectively prevent high-quality phishing attacks.

A third approach is to visually differentiate the phishing sites from the spoofed legitimate sites. Dynamic Security Skins [5] proposes to use a randomly generated visual hash to customize the browser window or web form elements to indicate the successfully authenticated sites. PassMark [18] includes a personalized image in a web page to indicate that the user has set up an account with the site. This approach places the burden on *users* to notice the visual differences between a good site and a phishing site and then correctly infer that a phishing attack is underway. The Web Wallet, by contrast, detects the discrepancy itself, by comparing the user's intention with what the user is actually doing. The Web Wallet also disables direct input of sensitive information into web forms so that Web Wallet protection is a necessary part of the user's current task.

A fourth approach is two-factor authentication, which ensures that the user not only knows a secret but also presents a security token. [8] However, this approach is a server-side solution. Phishing can still happen at sites that do not support two-factor

authentication. Sensitive information that is not related to a specific site, *e.g.*, credit card information and SSN, cannot be protected by this approach either. The Web Wallet protects users from phishing at the client side. It does not require web sites to change their login mechanisms. It also protects other sensitive information besides login information.

The PRIME project [19] helps users to manage their online identity in a more natural and intuitive way using three UI paradigms. It supports drag-and-drop actions for personal information submission. It does not specifically target the phishing problem but its improved user interface could help users correctly manage their online information. One potential problem with the PRIME interface is its "Just-In-Time-Click-Through Agreements" (JITCTAs) that is used to generate "small agreements [that] are easier for the user to read and process". Users could still ignore the agreements by directly clicking through the "I Agree" button. On the other hand, the Web Wallet integrates security questions into the user's workflow so that users have to explicitly indicate their intended sites when submitting sensitive information.

2.1 Comparison between the Web Wallet and Microsoft InfoCard

Microsoft InfoCard [4][15] is an identity metasystem that allows users to manage their digital identities from various identity providers and employ them in different contexts where they are accepted to access online services. The Web Wallet, although developed independently, shares some similarities with InfoCard. The most important one is that both solutions propose a simple, consistent and predictable user interface for authentication. Users always use a single interface, either the Web Wallet sidebar or the InfoCard identity selector, to provide their identity information to different web sites. On the other hand, there are several fundamental differences between these two solutions.

First, since InfoCard is a new way for users to provide their identity information, web sites have to be modified to accept the InfoCard submission, by adding an HTML <OBJECT> tag that triggers the InfoCard process at the user's browser. Moreover, sites have to add backend functionality to process the credentials generated from different identity providers. The Web Wallet, by contrast, still uses the site's existing authentication mechanism but only makes the submission interface secure at the client side.

Second, since InfoCard is an identity metasystem, it needs support from various identity providers, including banks that issue bank accounts, credit card companies that issue credit cards, and government agencies that issue government IDs. As a result, these identity providers also need to add functionality to process the InfoCard requests.

Third, in order to use InfoCard, users have to contact different identity providers to obtain InfoCards from them, which introduces an out-of-band enrollment process between the users and the identity providers.

Fourth, every time a user selects an InfoCard, she needs to authenticate herself to the identity provider. This authentication either needs system level changes (if hardware tokens or biometrics are used) or is potentially vulnerable to phishing attacks (if username and password are used).

Fifth, when triggered in the user's browser, the InfoCard interface first authenticates sites to the user by displaying site information, including the company's name, location, logo and site certificate. However, the interface is still a generic "are you sure" confirmation with a question of "do you want to send a card?" and two options of "yes, choose a card to send" and "no, return to the website". Users are expected to make the correct decision solely with the site information. But studies have shown that users cannot always make the correct decision in this situation. [26] The Web Wallet tries to get rid of this "are you sure" confirmation by asking users to explicitly indicate their intended site.

3. DESIGN PRINCIPLES

The Web Wallet is based on the following two design principles: (1) structuring the interface so that the user's intention is obvious, and (2) integrating security into the user's task workflow, so that it cannot be ignored.

3.1 Get the User's Intention

Phishing attacks exploit the gap between the way a user perceives a communication and the actual effect of the communication. The computer system and the human user have two different understandings of a web site. The user recognizes a site based on its visual appearance and the semantic meaning of its content. But the browser recognizes a site based on system properties, *e.g.*, whether the site has an SSL certificate, when and where this site registered, *etc*. As a result, neither the computer system nor the human user alone can effectively prevent phishing attacks. On the one hand, it is hard, if not impossible, for the computer to always correctly derive the semantic meaning of the content. On the other hand, ordinary users do not know how to correctly interpret the system properties. The user interface is thus the exact place to bridge the gap between the user's mental model and the system model by letting the human user and the system share what they individually know about the current site.

The Web Wallet helps the users transfer their real intention to the browser, especially when they are doing phishing-critical actions, such as submitting sensitive data to web sites.

When a user is submitting data, her intention includes two parts. The first part is the data type. Is the submitted data sensitive or not? If yes, what kind of sensitive data is it? The second part of her intention is the data recipient: which site does the user intend to submit her data to? When a user uses the Web Wallet — a dedicated interface for sensitive information submission — she implicitly indicates that the submitting data is sensitive. The user further indicates the sensitive data type by using the appropriate card in the Web Wallet (*e.g.,* the login card, the credit card, *etc.*). The Web Wallet then checks to see if the current site is good enough to receive the sensitive data. If the site is good enough, the data is filled from the Web Wallet to the web page.

If the site is suspicious, the Web Wallet lets the user indicate her intended site. If there is a discrepancy between the user's intended site and the current site, in which case the user probably is under a phishing attack, she will be warned effectively by a message like "You may think that this site is [PayPal] but in reality this site has no relationship with [PayPal] (here is why) and thus this site is probably fraudulent." This discrepancy is the fundamental danger of phishing and knowing the user's true intention makes such a warning possible.

Moreover, as long as the Web Wallet knows the user's real intention, the user can be advised not only to stop at the current site but also to continue at her intended site by giving her an alternative safe path. In this way, the user's intention is respected and the user does not need to take risks in order to finish her job.

3.2 Integrate Security into the Workflow

When users are doing tasks online, security is rarely their main concern. Therefore, effective security mechanisms should integrate themselves into the user's current workflow. The Web Wallet does the integration in the following two ways.

First, the Web Wallet does not depend on users remembering to use it. Instead, it *requires* users to use it by disabling the sensitive input fields in the web forms and making itself the only way to input sensitive data.

Second, the Web Wallet incorporates security questions by *helping* users achieve their goals instead of *stopping* them. When users are detected trying to submit sensitive information to a suspicious site, the Web Wallet requests confirmation. But the Web Wallet does not use a generic warning like "are you sure you want to send this information to this potentially fraudulent site?" Such warnings are known to be ineffective because a user tends to say yes — meaning not that the current site is her intended site, but that she definitely wants to continue making progress towards her goal. Instead of a generic yes/no confirmation, the Web Wallet shows a user a list of sites, including the current site, and lets the user explicitly acknowledge and indicate their intended site. Asking users to *choose* a safe mode to finish their tasks has been found to be more dependable and effective than merely *reminding* them to finish their tasks in a safe mode. [25]

4. USER INTERFACE

A user uses the Web Wallet to submit her sensitive information. In this section, we introduce the major features of the Web Wallet by going through a typical process of sensitive information submission — logging in.

4.1 Form Annotation

For every web page that is displayed in the user's browser, the Web Wallet searches for login forms. If a login form is found, the Web Wallet disables the form's password input so that it no longer permits typing. The user has to open the Web Wallet to login.

The username and password are grouped together by the Web Wallet into a single unit called a login card (figure 2). The card metaphor is a natural data submission unit for other sensitive information as well, like credit card information and bank account information.

4.2 Security Key

The Web Wallet does not open automatically; rather, we require the user to press the security key to open it for two reasons. First, many web sites include a login form in their home page and the user may not want to login but simply browse the site. Second, we want to make pressing the security key an essential action so that it becomes habitual. When the user forms this habit, she will always open the real Web Wallet every time she wants to use it.

In our current implementation, we use F2 as the security key. The security key idea has been implemented for a long time by the

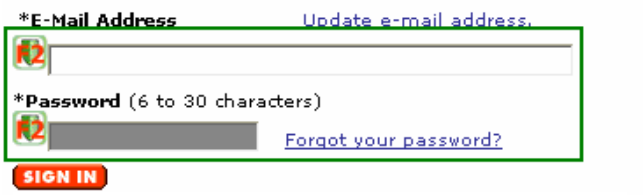


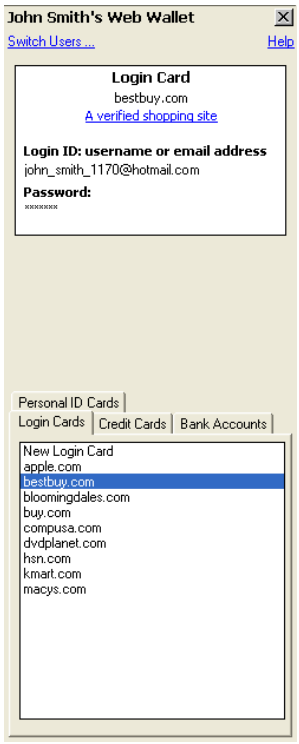
Figure 2. Login form annotation

Windows system to bring the user an authentic login screen, and has also been used as a defense against phishing. [7][20] Pressing a security key to open a trusted interface is more secure than clicking somewhere on the screen because every target on the screen can potentially be spoofed. Of course, to make it secure, the event of pressing the security key should be handled only by trusted code, so that its activation cannot be hijacked by a phishing web page.

4.3 Browser Sidebar

When the user presses F2, the Web Wallet is opened as a browser sidebar (figure 3). The main interface contains two parts: a card presentation area and a card folder. The card folder displays stored login cards. The stored login cards are encrypted using the Web Wallet's master password. Users will not be prompted for the master password until they first interact with the card folder, *i.e.*, when they are retrieving a stored card or saving a new card.

When there is a stored card that matches the request from the web page (this case happens when the user has sent the same card to this site before and has agreed to save that card into the Web Wallet), the stored card is displayed in the card presentation area (figure 3a) and the stored information is automatically filled into the login form. The user then clicks the submit button in the login



(a) Web Wallet with a stored card

(b) New login card

Figure 3. The Web Wallet sidebar

form. This procedure is free from phishing because the user has submitted the same information to this site before.

When there is no stored card matching the request, a new login card is displayed (figure 3b). The Web Wallet does not require users to save their login information. They can only use the Web Wallet interface to do secure submission. The new login card shows the domain name of the site and a description of its trustworthiness. The new login card provides input fields for the user to type in her username and password.

The new login card has a "save card" checkbox, which tells whether or not to save the new login card in the Web Wallet for future use.

Below the checkbox is a submit button. Depending on the site's trustworthiness and the user's history, the button displays different labels and performs different actions. If the site is rated as trusted, the button shows "Fill in" and clicking it fills the typed information into the login form. If the site is not rated as trusted, the button shows "Continue". Note that this does not mean that the site is guaranteed to be phishing; it may simply not meet our criteria in terms of the trustworthiness (described in section 7.4). Pressing the button has two sub cases. In one case, if the user is remembered to have logged into this site using the same login information, the Web Wallet simply fills the information into the web page. In the other case, if the user has not logged into this site before, the Web Wallet needs to confirm with her about her intended site, which will be elaborated in the next section.

4.4 Confirmation Interface

If the site is untrusted and the user has not submitted login information to it before, she will see a confirmation interface (figure 4) asking her to indicate her intended site. The confirmation interface shows a list of domain names generated from the user's Web Wallet history plus the current site's domain name (figure 4a). We use the user's history to generate this list because many effective phishing attacks claim to be legitimate sites that the user has contacted online before.

The user has to go through the list to choose her intended site, which differentiates this confirmation interface from a generic "are you sure?" warning. The user cannot simply ignore this list by clicking a "yes" button. If the user's choice is different from the current site, the user is warned about the discrepancy, and is given an alternative safe link to her intended site (figure 4b). The same warning will show when the user chooses a stored card from the card folder, but the current site does not match the stored card. If the user's choice is the same as the current site, since the current site is not rated as trusted, the user is shown the site report (figure 4c) explaining in detail why this site is not trusted and asked if she wants to continue. The interface also provides a way to do a Google search for the user's intended site. Searching is widely used when a user wants to go to a site but does not know the exact URL.

For some web sites, the Web Wallet may fail to detect the login form and thus cannot fill in the data automatically. A drag-and-drop interface is provided to deal with these pages (figure 5).

The Web Wallet does not greatly complicate a user's interaction with legitimate sites, and can even simplify the interaction. When a user logs in without a stored card, she must perform several extra steps compared with logging in directly: she must press F2,

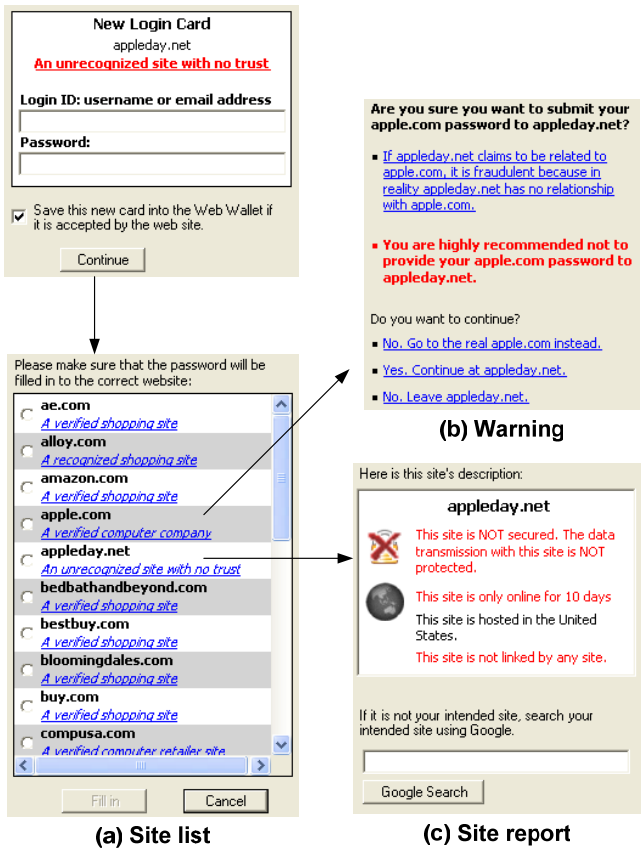


Figure 4. The Web Wallet confirmation interface

Figure 5. The Web Wallet drag-and-drop interface

shift her attention to the Web Wallet, enter her username and password as usual, press the submit button in the Web Wallet, and finally shift her attention back to the web form. However, when a user has a login card stored, she only needs to press F2 to activate the auto-fill feature of the Web Wallet.

4.5 Negative Visual Feedback

We conducted a pilot study of an early Web Wallet prototype. In that study, we introduced a new attack targeting the Web Wallet interface. At its login page, a phishing site displayed a fake Web Wallet with a new login card, thus enticing the user not to bother pressing F2, but rather to type their information into the fake card. To prevent this attack, we used a personalized image to distinguish the real Web Wallet from the fake one that did not include the image. Personalized images that are hard to spoof have been widely proposed for anti-spoofing [5][14][18][27]. But the pilot study found three out of four users still used the fake Web Wallet to login, showing that the absence of the personalized image was not strong enough to raise the user's suspicion. Research (*e.g.*, [22]) has shown that it is harder for humans to detect the absence of something, as opposed to its presence. Therefore, we take another approach. Instead of depending on the absence of positive cues (like the personalized image) to warn users, we present negative visual cues in potentially unsafe situations.

The fundamental difference between the fake Web Wallet and the real one is that the fake one is displayed by a web site while the real one is a local interface. Given that a web site can present anything at a user's computer, a phishing web site may not only spoof other legitimate web sites but also spoof the local security interface as well. [9][28] A way to effectively distinguish a web interface from a local interface may help users to detect the fake Web Wallet.

In order to differentiate the web interface from the local interface, we designed two types of *negative* visual feedback on the user's interaction at web pages. The first type of feedback produces a graphical effect on each character that a user enters at a web page: the typed character quickly zooms out from the typing location to the center of the browser and fades away (figure 6). Since the Web Wallet discourages users from typing sensitive information into web forms, we expect that this feedback will make them uncomfortable, and thus raise their suspicion when they are typing a password or a credit card number at a web page because their sensitive information is shown in plaintext.

To bypass this negative feedback, a phishing site might display an online keyboard as an image map and encourage the user to mouse-click the keys. (Some legitimate sites, *e.g.*, ING Direct, use the same technique to evade keyboard logging attacks. [10]) We therefore added a second type of feedback: whenever a user clicks on a web page, a semi-transparent warning icon will fly up from the clicking position to top of the browsing window (figure 6), where a reminder bar is located and says that "your keyboard typing and mouse clicks are going directly to the web site."

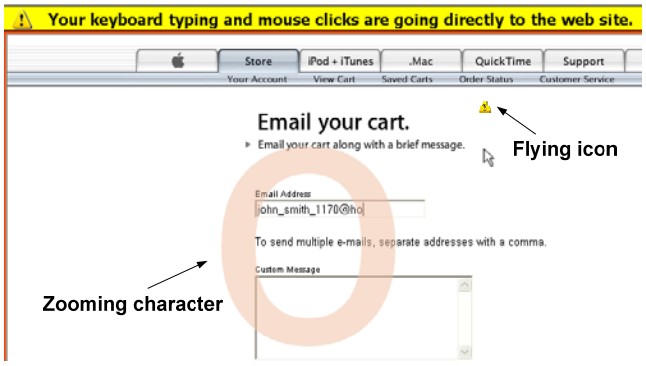


Figure 6. Negative visual feedback

In addition to these reactive visual feedbacks, we also proactively draw a red boundary around a web page to indicate that the bounded area is a web interface and thus potentially unsafe.

5. USER STUDY

Although the Web Wallet is security software, it will be used by the human user. Therefore, before we implement a full-featured Web Wallet, we started with a prototype and ran a controlled user study to test both its usability and its effectiveness at preventing phishing attacks. The tested Web Wallet prototype only supports login information, not credit card or bank information. And the backend is mostly hard-coded for the web sites used in the study.

We used the same scenario from previous studies of anti-phishing toolbars [26]. A subject was told to act as the personal assistant of John Smith. John Smith forwarded 20 emails to the subject and asked her to go to 20 different web sites, log in with his password, and add items to his wish list. We did not include any tutorial about the Web Wallet in the study, since few users in the real world read tutorials. The Web Wallet interface must be self-explanatory.

Five of the 20 forwarded emails were attacks, with links leading the subject to phishing web sites. Phishing attacks were simulated by connecting to the real web site but changing the browser's address bar to display a different hostname (indicating that the web page was from an unusual source). We simulated the ideal phishing attacks whose content is a perfect copy of the actual site.

Previous studies [26] showed that attacks using a similar hostname (*e.g.*, `www.amazon-department.com` to spoof `www.amazon.com`) have the highest spoof rate, compared with attacks using an IP address or a totally different hostname. In this study, all the attacks displayed a URL in the address bar with a hostname similar to the legitimate site. All the attacks did not use an SSL connection. Six out of the 20 tested web sites in the real world do not use SSL to protect their login pages.

5.1 Simulated Phishing Attacks

Among the five attacks, one attack represents a normal phishing attack. In this attack, the phishing site, either copying the legitimate site or acting as a man-in-the-middle between the user and the real site, uses the same HTML login form as the one in the legitimate site. Therefore, the login form can be detected and disabled by the Web Wallet. The user has to open the Web Wallet to login to the phishing site.

Any new security interface should also be tested with new potential attacks. The other four attacks are designed to specifically target the Web Wallet interface.

- *Undetected-form attack:* The Web Wallet uses some heuristic rules to analyze the HTML source code of the current page to detect the login form. It is possible that a phishing site manages to bypass the Web Wallet detection. In this attack, the login form is not disabled. Note, however, that any typing at the undetected form still makes our negative visual cues appear – *e.g.*, the password characters zoom out of the screen.

- *Online-keyboard attack:* This is a modified undetected-form attack. In this attack, the Web Wallet still fails to detect the login form. Furthermore, in order to bypass the zooming-character feedback, the site tells the user that an online keyboard is used as an extra protection to the user's password. A user is required to click the keyboard image to input her password.
- *Fake-wallet attack:* This is another modified undetected-form attack. The Web Wallet fails to detect the login form. Furthermore, at the login page, a fake Web Wallet is displayed, as shown in figure 7. The login form is annotated in the same way as the real Web Wallet will do, but actually by the site itself. The fake Web Wallet displays a new login card and tricks the user into using it. However, interaction with the fake Web Wallet produces negative visual feedback — in particular, typing a password in the fake Web Wallet makes the password characters zoom out of the screen.
- *Fake-suggestion attack:* One of the useful features of the Web Wallet is its ability to suggest a safe path to the intended sites, but this suggestion mechanism can be exploited by attackers. In this attack, the first page pops up a fake warning telling the user that the current site is a known fraudulent site and asking her to choose her intended site from a list, as shown in figure 8. The list, controlled by the phishing site, includes a phishing site that uses a similar name to the user's intended site and is marked as trusted. When the user chooses the phishing site from the list, the leading site performs an undetected-form attack. (The phishing site could perform any of the four attacks listed above, but we simply used the undetected-form attack in this study.)

In this study, the Web Wallet described all the phishing sites in red as "an unrecognized site with no trust" and described all the legitimate sites in blue as verified or recognized sites.

The five attacks were randomly assigned to five fixed positions: the 5th, 8th, 12th, 16th, and 19th forwarded email. Under the phishing attacks, the user may have the stored login card in the Web Wallet. Therefore, in this study, for each subject, we randomly chose two to three attacks to include the saved login cards for their spoofed legitimate sites. A total of ten login cards were initially saved and each subject had to use the new login card for the other ten sites.

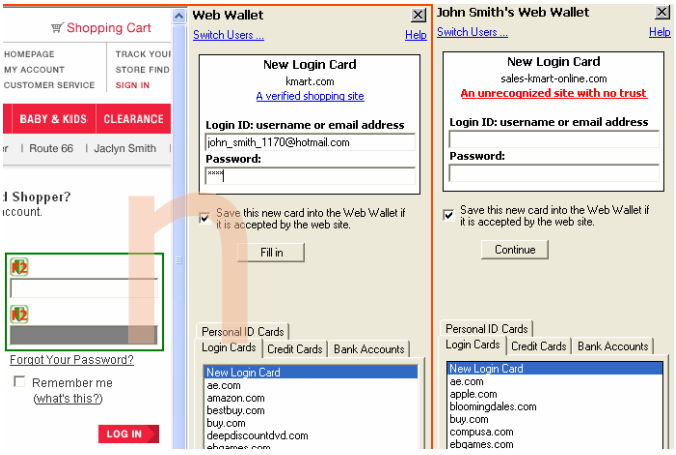


Figure 7. Fake-wallet attack (after the security key is pressed, the fake wallet and the real one are side-by-side.)

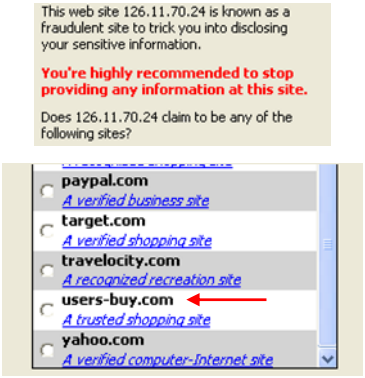


Figure 8. Fake-suggestion attack

To evaluate the Web Wallet, we used a control group with subjects who used Internet Explorer 6.0 (IE) without the Web Wallet in it. The only security indicators for this group are the browser's address bar and the status bar. The control group saw five normal attacks, presenting perfect web content but a changed URL. None of these attacks used SSL.

6. RESULTS AND DISCUSSION

A total of 21 subjects with previous experience in online shopping, 11 females and 10 males, were recruited at a college campus. Thirteen subjects (62%) were college students from 11 different majors. All subjects had at least a college education. The average age was 24 (the range, 19 to 34). Fourteen subjects were randomly assigned to use the Web Wallet and the other 7 were in the control group.

To gauge subjects' experience with online shopping, we asked them which of our 20 selected e-commerce sites they had visited. All 21 subjects had used Amazon and Yahoo, and 15 or more had used Apple, Target, Travelocity and Bestbuy. On average, each subject had used 9 sites in our study.

Before the study, subjects were briefed about the scenario and their role as John Smith's assistant. The subjects were told to be careful with John Smith's account information during the study.

We personally observed the subjects' browsing behaviors during the study. We did not interrupt the study except when subjects clicked the "report fraud" button, at which point we asked them to explain why they reported fraud and told them to stop the task at the current phishing site. At the end of the study, we interviewed the subjects by going over the unrecognized attacks to find out why they did not recognize them.

We define the *spoof rate* as the fraction of simulated attacks that successfully obtain John Smith's username and password without raising the subject's suspicion. Figure 9 shows the spoof rate of the normal attack with and without Web Wallet protection, and the spoof rate of all the attacks in the Web Wallet study. The Web Wallet protection significantly lowers the spoof rate of the normal attack from 63% to 7% (one-tail $t(42) = 5.09$, $p < 1e\text{-}05$).

Of the seven subjects in the control group, two of them reported all the phishing attacks based on (1) the odd URLs and (2) the fact that the login page is not SSL-protected. Note that one subject believed that four good sites were attacks because of the lack of SSL. The other five subjects were tricked by at least three attacks,

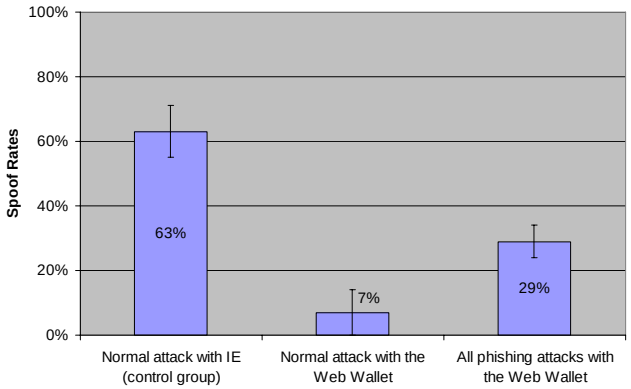


Figure 9. Spoof rates with and without the Web Wallet protection (including the standard errors)

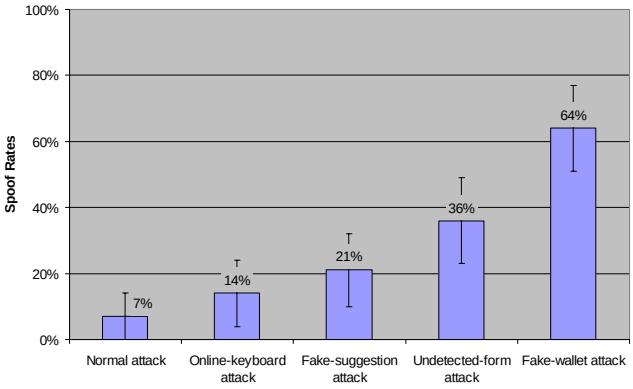


Figure 10. Spoof rates of the five attacks in the Web Wallet test (each type had a total of 14 attacks)

including three subjects being tricked by all of them, by either not looking at the URLs or explaining away the odd URLs ("*the url [signin.travelocity.com.zaga-zaga.us] starts with signin and I do want to sign in and it ends with 'us' and I know that Travelocity is in US.*").

For the rest of this section, we will focus on the Web Wallet test. Figure 10 shows the spoof rates of all the five attacks with the Web Wallet test.

Among the 14 subjects who used the Web Wallet, the first eight subjects and the last six ones used two different interfaces. We will first introduce the results of the first interface, then explain why we changed the interface in the middle of the study, and finally introduce the results of the modified interface. As we will show later, a majority of the successful attacks happened without the Web Wallet being open. Therefore, we combine the results together in figure 9 and 10 because changing interface does not affect the totally spoof rate much. (But changing interface did improve the security when the Web Wallet was in use.)

6.1 Results of the First Interface

There were 40 attacks experienced by the first eight subjects (figure 11). Twelve attacks were reported without the Web Wallet being open (figure 11a). Six of them were detected because of the odd URLs. The other six were correctly reported as fraudulent, but we should also note that the subjects were still tricked by the fake interfaces under the attacks. In particular, five attacks were fake-suggestion attacks and were reported because "*A window popped letting me know that the website was using fraudulent methods to conceal its identity from me.*" But this warning window was itself fraudulent. The other one attack was the fake-wallet attack and was reported: "*Not sure how to resolve the disagreement between the Web Wallet UI which reports 'radioshack.com' and the Address Bar of IE which reports 'radioshack.no-ip.info'. This seems suspicious.*" Again, this subject did not seem to suspect that the Web Wallet interface itself was fake.

The Web Wallet helped to detect 17 attacks (figure 11b). Fourteen attacks were reported because of the domain name and the site description displayed in the new login card (like figure 3b) with a typical report as "*The Web Wallet says it is 'unrecognized site with no trust' and the server name is different from* `compusa.com`." The other three attacks were detected when the subjects chose a stored card for login. The Web Wallet detected the discrepancy between the subject's intended site and the current site and then warned the subjects (like figure 4b).

Eleven out of 40 attacks succeeded (figure 11c). In six attacks, the subjects failed to open the Web Wallet. Four of them were the fake-wallet attacks when the subjects logged in using the fake

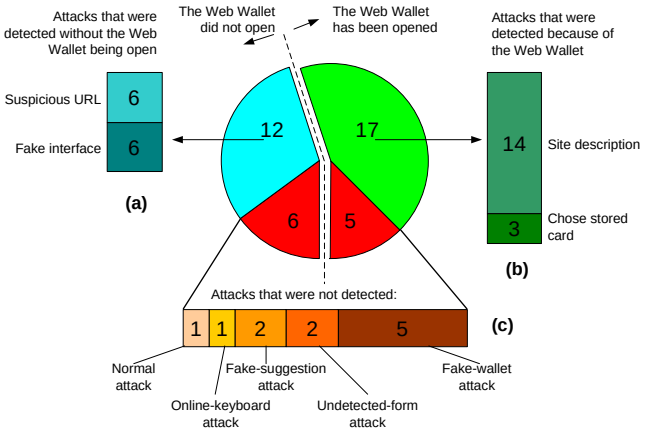


Figure 11. Forty attacks with the first interface

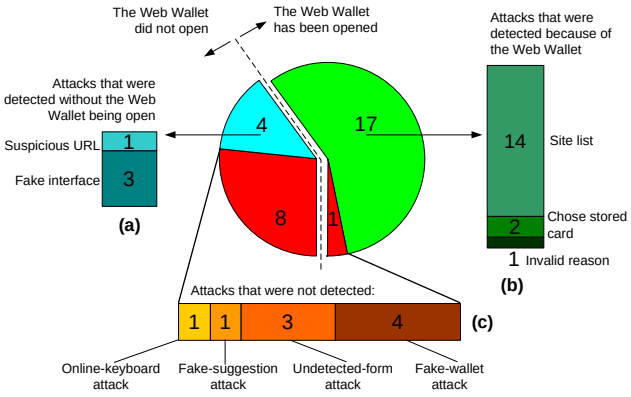


Figure 12. Thirty attacks with the modified interface

Web Wallet. We also had one subject who was tricked by all five attacks. In the interview, she said that she did not trust the Web Wallet and she tried to ignore everything (including the real Web Wallet warnings) except the web page content because she just wanted to get the job done. "*If something happens, it is John's fault because it is he who has forwarded these emails to me.*"

Five successful attacks happened even when the Web Wallet had been opened during the attacks. We analyzed these five attacks and found that two of them happened because we originally implemented the Web Wallet confirmation interface in a wrong way! The site list did not include the current site and acted as a generic "are you sure?" warning. The subject who got tricked bypassed the site list simply by answering no to the question "does this site claim to be any of the following sites that you have had an account with?" without even looking at the list.

Two other successful attacks reflected another usability problem. The Web Wallet originally did not support typing when it did not detect any login form. The subjects had to open the new login card to type. Both attacks were the undetected-form attacks, one of which was redirected by a fake-suggestion attack. Both subjects opened the Web Wallet and saw the site description in red, but they did not believe it: "*The red label made me a little nervous. But everything else looked good. So I thought that it is simply the browser did not like this site. Or maybe John Smith has never done this site before.*" And they wanted to continue. But only the web form provided a way for direct typing. So they tried the web form ("*Let's see if it works.*") and got tricked. We expected that always displaying a card, especially the new login card under phishing attacks that supports typing, would prevent users from using the web forms.

Even though there were problems in the first interface, the results are still valuable and we have learned the following lessons:

- By blocking the web forms from the legitimate sites, the Web Wallet could train the subjects to depend on it to login. As a result, the subjects would try the Web Wallet before they tried other newly proposed login schemes. This explained why the online-keyboard attack had the second lowest spoof rate. Under this type of attack, the subjects tended to open the Web Wallet to login and saw the site description in red.
- The Web Wallet successfully draws the subjects' attention to the site description. The description was a much stronger signal than the traditional browser's indicators and sometimes even stronger than the convincing web content.

Based on the problems we had found in the first interface, we modified the Web Wallet. First, we added the current site to the site list so that the user had to examine the list to find her intended site in order to continue.

Second, no matter whether there is a detected login form or not, the Web Wallet, when it is open, always displays a login card. This improvement would prevent the subjects from using the undetected login form while the Web Wallet is open.

Third, because the majority of the attacks were detected by the site description, the card did not display it in the modified interface so that we could test other features of the Web Wallet. Note that this change is different from the first two changes. It was done not because of a problem in the user interface but because we wanted to test more features. The site description had shown to be a useful feature, since it was responsible for detecting 14 out of 22 attacks.

6.2 Results of the Modified Interface

There were 30 attacks experienced by the last six subjects with the modified interface (figure 12). Four attacks were detected without the Web Wallet being opened (figure 12a), including one being reported because of the odd URL and the other three being reported because of the fake warning window under the fake-suggestion attacks.

In 18 attacks the Web Wallet had been opened and it helped to detect 17 of them (figure 12b). The modified interface dramatically decreased the Web Wallet failure rate from 23% (5 out of 22 successful attacks) to 6% (1 out of 18 successful attacks).

In 14 out of the 17 detected attacks, the subjects saw the site list including both the phishing site and the spoofed legitimate site. In eight of these instances, the subjects chose the legitimate site, which is their intended site, and then saw the warning informing them that the current site was fraudulent (like figure 4b). In five instances, the subjects clicked the phishing site's description to open the site report (like figure 4c) and reported fraud. In the final instance, the subject immediately reported fraud when he found that both the phishing site and the spoofed site were in the list.

Two attacks were detected when the subjects chose a stored card of their intended site and were warned that the current site was not their intended site.

Nine out of 30 attacks (spoof rate of 30%) successfully tricked the subjects (figure 12c). But only one attack happened when the Web Wallet was open. It was an undetected-form attack to spoof `amazon.com`. The subject opened the Web Wallet at the login page and chose the stored Amazon card. She saw the warning claiming that the current site `amazon-department.com` was spoofing `amazon.com`. Instead of dealing with the warning, she then interacted with the web form. In the interview, the subject claimed that she did notice the warning but failed to pay attention to it because she felt so comfortable with the Amazon site. We believe this problem can be easily solved: whenever a warning is displayed, the Web Wallet already knows that the user's intended site is not the current site, so interaction with the current site in main browsing window should be totally blocked until the user has acknowledged the warning.

We learned the following lessons with the modified interface:

- Always providing affordance for typing was effective at preventing the undetected-form attack with the Web Wallet being open. Six times the subjects saw an undetected form and opened the Web Wallet. When facing two login forms that support typing, the subjects always to type into the Web Wallet because they had learned to depend on it to login and their attention was already switched to it. However, we acknowledge that it is easy to provide the right card in this study because the new login card is the only choice. When the Web Wallet includes other sensitive information, how to design an appropriate default input interface is a problem for future work.
- Including both the phishing site and the spoofed legitimate site in the site list and letting the user choose her intended site was effective at preventing phishing attacks.

- The warning based on the discrepancy between the subject's intended site and the current site was effective at preventing phishing attacks because it is the fundamental danger of phishing. Combined with the results from the first interface, the subjects saw this warning 14 times. Only once did the subject fail to pay attention to it and we already discussed how to deal with it. Four times the subjects followed the suggested safe path. In the rest of the times, the subjects reported fraud based on the warning with a typical reasoning like "*users-buy is not related to* `buy.com`."

6.3 How Well Does the Web Wallet Work?

The Web Wallet effectively prevented the normal phishing attack. Subjects had to use the Web Wallet in this attack, and there were many opportunities for them to detect the attack, including the site description, the stored card, the site list, and the warning based on the discrepancy between their intended site and the current site.

The Web Wallet effectively prevented the online-keyboard attack. As long as the subjects depended on the Web Wallet to login, they used the Web Wallet first before trying other proposed login schemes.

The fake-suggestion attack is risky for the attacker. Many subjects (eight out of 14) stopped at the warning window. In the real world, the fake warning may also make users reconsider the requesting phishing email or try to type in their expected URL directly. However, if the users are successfully redirected, they tend to trust the phishing site that they are redirected to and are likely to be spoofed. Even though the real Web Wallet warns the users at the redirected phishing site, they would question the Web Wallet: "you have redirected me here but why are you warning me again?"

The Web Wallet failed to effectively prevent the undetected-form attack. Adding support for typing did decrease the spoof rate as long as the Web Wallet was open. But because users are used to web form submission, they have a strong tendency to use it. Changing this habit is not easy. We may be able to help address this problem by explaining the benefits of the Web Wallet to users in order to encourage them to break their habit of using web forms directly, but it needs further testing.

The fake-wallet attack is a very successful attack. Nine out of 14 subjects used the fake Web Wallet to login and the other five subjects reported fraud for reasons that were all unrelated to the real Web Wallet protection. Only two subjects pressed F2 to open the real Web Wallet. When he saw two Web Wallets side-by-side, one subject thought that another identical Web Wallet was open because he hit F2 by accident and thus he closed the real one and used the fake one.

The high spoof rates of the undetected-form attack and the fake-wallet attack indicate that negative visual feedback fails. Moreover, many subjects disliked the negative visual feedback. Nine of them said it was annoying or distracting. Three subjects noticed that sometimes the password was reflected in plaintext but they never thought about it seriously. Only two subjects found the visual feedback to be valuable. As the observers in the study, we also felt that the mouse-click feedback was annoying. Fortunately, the online-keyboard attack did not seem effective anyway, so we may drop the mouse-click feedback. We want to test more on the character-type feedback — perhaps try other methods besides zooming. We should also add a detailed explanation to the user at an appropriate time about the purpose of this feedback. We still think this feedback can be a useful feature if it succeeds because it can prevent both the undetected-form and the fake-wallet attacks.

The red boundary around the web interface also failed to differentiate the web interface from the local interface. No subjects knew why it was put there.

7. IMPLEMENTATION

The Web Wallet is implemented in C# because we want to integrate it into Microsoft Internet Explorer, the most popular web browser, so it can protect the majority of the Internet users. In this section, we will discuss how we implement the major backend functionalities in the real world.

7.1 Form Detection

It is very important for the Web Wallet to successfully detect web forms that ask for sensitive information. We use Naïve Bayesian classifier and Hidden Markov Model to detect sensitive inputs based on the HTML source code. We have trained the algorithm using a set of web pages. A preliminary test showed that this algorithm can accurately detect the sensitive inputs at most tested sites. [24]

To further increase the accuracy of form detection, web sites can add a new attribute to input elements, identifying them as sensitive fields that should be protected by the Web Wallet.

Even when the algorithm fails at some sites, the Web Wallet has a fallback. It provides a drag-and-drop mechanism for those undetected forms. Moreover, as long as the user deals with those forms once, the Web Wallet can locally remember those forms. If the user agrees, she can also report those undetected forms to improve the machine learning algorithm.

We admit that the phishing pages can be structured to bypass the form detection algorithm. But here is our claim: as long as the Web Wallet correctly detects and disables the sensitive forms at most legitimate sites, users are expected to form the habit to always use the Web Wallet when submitting sensitive information even at sites with undetected sensitive forms.

7.2 Sensitive Information Definition

Initially, we designed the Web Wallet to protect the submission of four types of sensitive information: login password, credit card information, bank account information, and personal IDs, including SSN, state ID and driver's license number. These types of information are most often targeted by current phishing attacks. As a future plan, we want to integrate P3P [23] into the Web Wallet so that the Web Wallet can protect all the defined data types in P3P. We choose P3P not only because it already has a set of terminology for personal data, but also because P3P-enabled sites can easily add the Web Wallet attribute to their sensitive input fields.

7.3 Site Definition

For most e-commerce sites, the domain name is a natural key to associate with the login card for that site. However, domain is not a perfect criterion, for two reasons.

First, an organization with a single domain name may ask for different passwords to access different sub-domains. For example,

users need a Yahoo password to access their Yahoo account. But they need a Yahoo security key (another password) in order to access the sub-domain of `https://secure.yahoo.com/`, where the Yahoo Wallet is located. When the Web Wallet finds out that there are multiple passwords under a single domain, it should further differentiate these passwords using the sub-domains or the hostnames.

Second, given that outsourcing is a common practice, different domains may be related to a single organization. For example, the Cambridge Trust Company (`www.cambridgetrust.com`) uses an Internet banking service (`cib.ibanking-services.com`) to authenticate users. The Web Wallet will work best if it can reliably figure out the outsourcing information and group the corresponding sites together. It is helpful that legitimate sites can provide their outsourcing information. Behera and Agarwal have proposed a mechanism where servers provide security relevant metadata (including the outsourcing information) to the web browser via a standard protocol. [3] Dealing with outsourcing is a problem of future work.

7.4 Web Site Trust Analysis

The Web Wallet uses site trust analysis to improve both usability and security. The trust analysis assigns a trust rating to every web site. The information provided to the Web Wallet can be directly filled into web forms if the current site's trust rating is above a threshold. Moreover, the trust analysis generates a site report with more details, to help users to decide if they really want to continue providing information if the site is not rated as trusted.

The Web Wallet uses the TrustWatch service [11] developed and maintained by GeoTrust. It depends on the trust rating assigned by the TrustWatch service to each site and displays the following factors in the site report to help users better understand the current site.

SSL certificate. An SSL certificate from a well-known Certificate Authority (CA) is a good trust indicator. Most legitimate web sites use SSL to protect sensitive information submission. On the other hand, few phishing sites use SSL because obtaining an SSL certificate from a well-known CA requires site identity information that can be traced. But the SSL certificate should not be the only factor used in the trust analysis because some phishing attacks do use SSL, thanks to sloppy practices by some CAs. [16]

Trusted third-party certificates. The TrustWatch service indicates whether a web site is verified by trusted third-parties, including TRUSTe, ScanAlert, BizRate and CNET.

Site popularity. Site popularity is measured by how many other sites link to this site. Legitimate sites want to be popular in order to attract more users, but phishing sites do not want to be popular in order to avoid public inspection. Therefore, the site popularity is a useful indicator to include in the site report. It may help users detect a phishing site if they notice that a site claims to be a well-known site but actually there are no other sites that link to it. TrustWatch service uses Alexa (`download.alexa.com`) to indicate how many sites link to the current site.

Site registration information. Alexa also provides when and where the domain name was registered. Given the fact that most phishing sites are short-lived and many of them are hosted overseas to spoof US sites, the site's registration information is a useful indicator to include in the site report. It may help users detect a phishing site if they notice that a site claims to be a well-known US site but it was actually registered overseas just a few days ago.

Site category information. Alexa indicates the purpose of a web site by providing the site's category information from the Open Directory Project (`www.dmoz.org`), the largest, most comprehensive human-edited directory of the Web. Legitimate organizations want their web sites to be categorized. But phishing sites do not want to because their web contents would have to be examined first. It may help users detect a phishing site if they notice that a site claims to be a well-known shopping site, but the site is not categorized at all.

7.5 User's History

The Web Wallet uses the user's history to improve both usability and security. On the one hand, users can save their input data for later use. The data is encrypted using the Web Wallet's master password. On the other hand, if users do not want the Web Wallet to save their data, their submission history is remembered by storing the hash value of the data indexed by the current site's domain name or by the hostname if multiple different submissions are detected. The original data cannot be recovered from the hash value. The hash value is used for the Web Wallet to check whether a user has submitted the same data into the same site before, so that the next time when she input the same data again, the Web Wallet directly fills in the data no matter whether the site is rated as trusted or not. The user's submission history is also used to generate the site list in the Web Wallet confirmation interface.

8. CONCLUSIONS AND FUTURE WORK

We introduced a new anti-phishing solution called the Web Wallet and ran a user study to test its effectiveness in preventing phishing attacks.

The Web Wallet significantly decreased the spoof rate of normal phishing attacks from 63% to 7%. The Web Wallet also effectively prevented other kinds of phishing attacks as long as it was used.

- By blocking web forms for sensitive input, the Web Wallet successfully makes itself an integrated part of the user's workflow.
- Because the Web Wallet is an integrated part of the user's workflow, it successfully draws the user's attention to its display. The warning from the Web Wallet is no longer a weak signal that can be easily ignored.
- The site list of the Web Wallet encourages the user to choose her intended site, which is very useful for the Web Wallet to detect the discrepancy between the user's intended site and the current site.
- The warning based on the above discrepancy effectively stops the user from providing information to the phishing attacks because it points out the fundamental danger of phishing.

We see many ways to improve the design of the Web Wallet. For example, the Web Wallet should not only support login using an existing password, but also other password-related activities, such as registering a new account and changing the password of an

existing account. The full-featured Web Wallet should by default also protect credit card information, bank account information and personal identity information. Eventually, the Web Wallet should be able to protect any personal data defined by P3P.

Clear and detailed explanations should be added to the Web Wallet interface in order to help users better understand the purpose of the Web Wallet and to correctly use it, *i.e.*, open it whenever necessary.

We also have to find a solution to prevent the Web Wallet itself from being spoofed. We plan to use image recognition techniques to detect the presence of a fake Web Wallet. Image recognition techniques have been proposed to detect phishing attacks by measuring the suspicious pages' visual similarity to the protected pages. [17] However, recognizing web pages has a scalability problem since there are millions of web pages that the user can access. Web sites are also constantly changing their appearance.

But recognizing the Web Wallet interface does not have these problems since it is the only fixed interface that the system needs to recognize. The system knows exactly what the Web Wallet looks like and in which part of the screen the Web Wallet is located. Moreover, since the Web Wallet is a working prototype, its interface is free to redesign, so we can use image recognition techniques to design the Web Wallet interface so that it is easy to recognize.

The Web Wallet recognition should also be guided by user feedback. We need to minimize the possibility that a distorted fake Web Wallet interface is accepted by the user, but fails to be recognized by the system.

9. ACKNOWLEDGEMENTS

We gratefully acknowledge help and suggestions from Ruth Rosenholtz and Anthony Fu. We thank Mike Rowan from GeoTrust to give us access to the TrustWatch service. This work was supported in part by the National Science Foundation (award number IIS-0447800). Any opinions, findings, conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the National Science Foundation.

10. REFERENCES

[1] Adida, B., Hohenberger, S., Rivest, R. Lightweight Encryption for Email. USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), 2005.

[2] Anti-Phishing Working Group. Phishing Activity Trends Report, December 2005. http://antiphishing.org/reports/apwg_report_DEC2005_FINAL.pdf

[3] Behera, P., Agarwal, N. A confidence model for web browsing. Toward a More Secure Web - W3C Workshop on Transparency and Usability of Web Authentication, 2006.

[4] Cameron, K., Johns, M. Design Rationale behind the Identity Metasystem Architecture.2006. http://www.identityblog.com/wp-content/resources/design_rationale.pdf

[5] Dhamija, R., Tygar, J.D. The Battle Against Phishing: Dynamic Security Skins. SOUPS 2005.

[6] Emigh, A. Online Identity Theft: Phishing Technology, Chokepoints and Countermeasures. ITTC Report on Online Identity Theft Technology and Countermeasures. October 3, 2005.

[7] Emigh, A. Trusted Path in Heterogeneous Environments. 1st TIPPI Workshop, 2005.

[8] FDIC. Putting an End to Account-Hijacking Identity Theft. 2004. http://www.fdic.gov/consumers/consumer/idtheftstudy/identity_theft.pdf

[9] Felten, E.W., Balfanz, D., Dean, D., Wallach, D.S. Web Spoofing: An Internet Con Game. Proceedings of the 20th National Information Systems Security Conference, 1997

[10] Fight Identity Theft Blog. ING Direct Fights Keystroke Loggers. December 11, 2005. http://fightidentitytheft.com/blog/?p=23

[11] GeoTrust. TrustWatch Tools. http://www.trustwatch.com/

[12] Herzberg, A. The 'Unprotected Login' Inter-Net Fraud League (I-NFL) Hall of Shame.2005. http://www.cs.biu.ac.il/~herzbea//shame/

[13] Herzberg, A. TrustBar: Re-establishing Trust in the Web. 2006. http://www.cs.biu.ac.il/~herzbea/TrustBar/

[14] Jakobsson, M., Myers, S. Stealth Attacks and Delayed Password Disclosure. http://www.informatics.indiana.edu/markus/stealth-attacks.htm

[15] Johns, M. A Guide to Supporting InfoCard v1.0 Within Web Applications and Browsers. March, 2006. http://www.identityblog.com/?page_id=412#infocardg_topic5a

[16] Krebs, B. The New Face of Phishing. The Washington Post. Feb 2006.

[17] Liu, W., Deng, X., Huang. G., Fu. A.Y. An Antiphishing Strategy Based on Visual Similarity Assessment. IEEE Internet Computing, Vol. 10, No. 2, pp. 58-65, March/April, 2006.

[18] PassMark Security. Two-Factor Two-Way Authentication. http://www.passmarksecurity.com/

[19] Pettersson, J. et al. Making PRIME Usable. SOUPS, 2005.

[20] Ross, B., Jackson, C., Miyake, N., Boneh, D., Mitchell, J. Stronger Password Authentication Using Browser Extensions. Proceedings of the 14th Usenix Security Symposium, 2005.

[21] Sharif, T. Phishing Filter in IE7, September 9, 2006. http://blogs.msdn.com/ie/archive/2005/09/09/463204.aspx

[22] Treisman, A., Gormican, S. Feature analysis in early vision: Evidence from search asymmetries. Psychological Review, 95, 15-48. 1988.

[23] W3C. Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/

[24] Wu, M. Fighting Phishing at the User Interface. PhD Thesis. MIT. 2006.

[25] Wu, M., Garfinkel, S., Miller, R. Secure Web Authentication with Mobile Phones. DIMACS Workshop on Usable Privacy and Security Software, 2004.

[26] Wu, M., Miller, R., Garfinkel, S. Do Security Toolbars Actually Prevent Phishing Attacks? CHI 2006.

[27] Ye, E., Smith, S. Trusted Paths for Browsers. Proceedings of the 11th USENIX Security Symposium, 2002.

[28] Ye, E., Yuan Y., Smith, S. Web Spoofing Revisited: SSL and Beyond. Technical Report TR2002-417, 2002.