# PhishDuck: Capturing User Intention in an Email Client to Combat Phishing

**Shao-Yu Wu**

**December 2009**

**Thesis Committee:**

Jason Hong, chair

Nicolas Christin

M09-Information Networking Thesis

Master of Science in Information Technology – Information Security

Carnegie Mellon University

# Acknowledgements

I would like to express my gratitude to my advisor, Prof. Jason Hong. He has been a truly excellent advisor and taught me everything about doing research. He was available to give advice whenever I needed it and provided me all the resources I wanted In addition, his wide knowledge has been of great value for me.

I owe my sincere gratitude to my reader, Dr. Nicolas Christin for his valuable feedback and always friendly help and support.

I owe a great debt to other members of Prof. Jason Hong's laboratory and the CUPS Laboratory for their feedback and help. Especially, Eiji Hayashi has been a great source for advice throughout graduate school, and specifically the implantation of PhishDuck and user study design. I appreciate his always help and am also glad to have had the opportunity to learn from him. I also want to thank Ponnurangam Kumaraguru for his advice on user studies, Min Kyung Lee for her advice on the interface design and encouragement, Bryan Pendleton for his proof-reading the paper.

I also thank my girlfriend, Yi-Wen Lin (the author of PhishDuck icon) and provided me always support and encouragement. Finally, I am grateful for my family for their love and support. And thank God for everything.

# Contents

# List of Figures

# List of Tables

# Abstract

We present the design and evaluation of PhishDuck, an anti-phishing tool for email clients. Phishduck presents a interfaces to users if they click on suspicious emails, and helps guide them towards making safe decisions. We present two different interfaces, a warning interface and a redundancy interface. In our user study, we found that the Phishduck warning interface was statistically significantly better than the warning in Mozilla Thunderbird, with the participants falling for phish decreasing from 70% to 0%.

# 1. Introduction

## 1.1 Background

Phishing is a form of identity theft where a criminal acquires sensitive information–such as credit card numbers, usernames, or passwords–by impersonating a trusted person or company. Criminals usually carry out a phishing attack by sending convincing emails with links to a phony Web site that asks users to reveal sensitive information [1]. These emails might ask users to take urgent action to avoid a dire consequence such as losing credit card access, renew a password necessarily, or receive a reward.

In terms of security, phishing is a growing and serious problem on the Internet, defrauding many victims every year. In December 2008, the APWG recorded 31,173 sites linked to phishing, a jump of over 800% when compared to figures from January 2008 [2, 3]. An estimated 5 million U.S. consumers lost money to phishing attacks in the 12 months ending in September 2008, a 39.8% increase from a year earlier, according to Gartner, Inc. [4]. Furthermore, phishing attacks are becoming more sophisticated, using browser exploits to install password-stealing malware. As noted by Sheng et al [45], phishing attacks and malware attacks are becoming increasingly blended.

## 1.2 Problems

Most past work in anti-phishing has focused on detecting phishing attacks [7,8,9,10,11] and on evaluating user interfaces of anti-phishing web browser toolbars [12,13,18]. However, there has been little work on preventing users from falling for

phishing email messages, despite the fact that email is the main vector for delivering phishing messages to users [15]. Some email clients (such as Thunderbird) now provide automatic phishing alerts which display warnings for suspicious emails [16]. However, one problem is that the warning may not be always correct. Specifically, the warning may be a false positive that incorrectly labels an email as phish, or worse, a false negative that incorrectly labels a scam email as legitimate. Another problem is that people may not see the warning, believe the warning, or know how to act on the warning. Egelman et al saw evidence of these kinds of problems with respect to anti-phishing warnings in web browsers [18].

## 1.3 Motivations

Our observation here is that today, email clients only capture low-level actions such as mouse clicks and keyboard actions, rather than the intention of the user. Email clients do not know what sites users intend to go to when clicking on links contained in messages. Knowing the users' intended destination could help us block fake emails while allowing legitimate emails.

## 1.4 Overview

In this paper, we describe the design and evaluation of PhishDuck, our anti-phishing tool for email clients. When clicking on links in emails, PhishDuck presents interfaces that try capture the user's intended destination web site, rather than just going directly to the link they click on (which may be faked in scam emails). Two different types of interfaces might be shown to the end-users: (1) if the email is highly suspicious, users will be shown PhishDuck's warning interface to warn them that the email has a strong potential of being phish, educating them about phishing attacks, and getting additional verification before allowing to proceed, while making it possible (but difficult) to get

to the possible phish site; and (2) if we are not sure of the provenance or legitimacy of this email, PhishDuck's redundancy interface will be shown to get extra information from the user to verify that they are going to the site they intend to go to. In designing PhishDuck, we tried to balance several goals: (1) protecting users from phish, (2) educating users about phish, (3) minimal negative effects from habituation, and (4) minimal annoyance to users.

We also present the results of our user study examining the effectiveness of various designs for email clients for protecting people from phishing scams. We found that both of PhishDuck's warning interface and redundancy interface were better than Mozilla Thunderbird's interface in protecting people from phish.

# 2. Related Work

There have been many strategies proposed for protecting people from phishing. We organize related work into three categories: (1) make it invisible, (2) training users, and (3) better user interfaces.

## 2.1 Make It Invisible

Work in this category provides protection without requiring any awareness or action on the part of users. Examples include finding phishing websites and shutting them down (done by law enforcement and ISPs) as well as automatically detecting and deleting phishing emails [6,14,17]. Other examples include Sender Policy Framework (SPF), an open standard to prevent email spoofing [19]; Remote-Harm Detection (RHD), a server-side tool that collects clients' Internet browsing history for identifying phishing websites [20]; and DomainKeys Identified Mail (DKIM), which uses public-key cryptography to let signers electronically sign legitimate emails that the messages can be verified by recipients [21]. Finally, there are also algorithms and heuristics for detecting phishing emails [6,14,17].

## 2.2 Training

There has been past work evaluating the effectiveness of anti-phishing training. For example, researchers have explored the idea of sending simulated phishing emails to test users' vulnerability and evaluate the effectiveness of training delivered through other channels [20,21,22]. Jagatic et al. studied the vulnerability of a university community towards a phishing email that impersonates someone from their own

social network, but did not study the effectiveness of training [23], while Researchers at West Point [24] and at the New York State Office of Cyber Security [25] conducting this type of study in two testing phases. Both studies showed an improvement in the participants' ability to identify phishing emails [26].

Sheng et al. have shown that people can be trained about phishing URLs through an online game called Anti-Phishing Phil, which has been shown to be effective in both a laboratory setting and in the real world [27]. PhishGuru uses simulated phishing emails to teach people about the risks of phishing [15, 26].

PhishDuck is not directly designed to train people about phishing, but is rather an interface designed to help guide people. As such, training is complementary to our goals.

## 2.3 Better User Interface

There have also been many anti-phishing interfaces developed. Many of these tools focus on the login process. For example, Passpet [28] is a password manager that simplifies the login process and makes it safer. PassMark is an image based website authentication, which lets users verify that they are logging in securely to the website by identifying their personalized image in the webpage [29]. Dynamic Security Skins individualizes login dialog boxes with randomly generated visual hash [30].

iTrustPage is a tool that uses search engines to help users differentiate between good and bad sites [31]. Web forms are blocked, and users enter keywords that describe the site they intend to be on. Search results are then retrieved and users can examine the results. Our work also uses this notion of user intent to help steer people away from

bad sites, though we do it for email and simplify the process.

Web Wallet is a tool that lets users submit personal information to web forms by clicking on known configurations (e.g. my Paypal login) rather than directly entering in data. Web Wallet can also suggest alternative safe paths to intended sites [13].

As mentioned earlier, we wanted to try to minimize the effects of habituation, where people simply swat away dialog boxes. One solution is to use polymorphic dialogs which change every time, forcing users to slow down [44]. Our approach is to have the most likely decision be the first one, and making it harder (but possible, in case of false positives) to continue to suspicious sites. Our user studies suggest that this is a promising approach.

BayeShield is a conversational user interface for helping users determine the legitimacy of a website [32]. Users are asked a series of questions (e.g., how did you get to this site? Do you recognize the sender?) which helps steer them to making better choices. Our philosophy is aligned with BayeShield, though we aim for using redundancy and try to simplify the process so as to minimize potential annoyance.

# 3. Early Design of PhishDuck

Here, we discuss our first iteration of PhishDuck. We opted to use a pop-up active warning to capture users' attention as well as their intention after they click on a link in an email. We chose this design since past research has shown that active warnings are more effective than passive ones, because they interrupt the user's primary task [18].

We considered several possibilities for capturing user intention. One early idea was to have users type in redundant information like "PayPal" or "Bank of America," which could be used to detect if there is a mismatch between the user's intended destination and the email link's destination. Another idea (of which we had several variants) was to present choices to people and have them simply click on the choice they wanted.

We developed several low- and medium-fidelity prototypes, and based on feedback narrowed our designs from six to one. Our initial design shows a warning interface (Figure 1) to users if they click on either a known or potentially suspicious link, and shows a safe indicator (Figure 2) if they click on a known safe link. We chose not to focus on the design where people typed in redundant information because early feedback indicated that it would be highly annoying having to type information for links, as it would defeat the purpose of clicking on links.

Our first cut for the warning interface (see Figure 1) had the following order of buttons: (1) go to the real web site (in this case PayPal), assuming that heuristics


**Figure 1** Early design of PhishDuck's warning interface. This interface is shown to users after clicking on suspicious links.

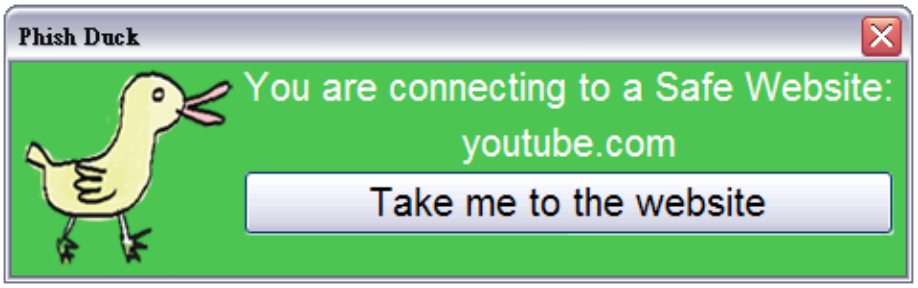


**Figure 2** Safe Indicator. In our early design, this interface is shown to users after they click on a safe link.

could detect basic keywords of highly phished Web sites; (2) go to the Web site that linked in the email; (3) close the warning interface; and (4) type in the web site to go to.

We conducted a pilot evaluation with ten participants. We asked our participants to role play as a specific persona, Pat Jones, to handle the emails in her/his mailbox just like what they will do in their normal life. The persona's mailbox contained seventeen email messages, including six phishing emails. The only difference between our pilot

study and our formal user study (described later below) is that we did not disqualify technically savvy individuals in our pilot studies, so as to get more feedback on our designs.

One piece of feedback we got from our pilot studies was that participants felt showing an indicator after they clicked on a safe link was annoying. Participants also felt that warning message were too long and hard to understand, for example, “suspicious website: 211.167.249.35”.

Moreover, in our pilot studies, we observed that participants were highly inclined to choose the first or second option in our warning interface, even though there was a “NOT Recommended” message in the second option. However, in later pilot tests, we reordered the options, making the second option (“NOT recommended”) the last option, with no participants choosing it subsequently. We concluded that ordering plays an important role when participants needed to make a decision from the options we gave.

We modified our designs to take into account the feedback above. We also applied Egelman’s design patterns for warning interfaces [44] in this second round design (with the patterns based on the C-HIP model for warnings [18]). The revised interfaces are shown in Figures 3 and 4. Finally, we designed our interfaces assuming people would not read the actual text, but would skim the text in buttons.

# 4. Email Classification in PhishDuck

In PhishDuck, emails are classified either as (1) good, (2) not sure, and (3) suspicious. By good, we mean emails that are either known to be legitimate or have an extremely high probability of being so. This verification might be done, for example, through whitelisted domains for links, key continuity management [35], DKIM [34], or some other mechanism. By not sure, we mean emails whose provenance and/or contained links we cannot completely verify as being legitimate. By suspicious, we mean emails that are known to be phish. This verification might be done through blacklists or through conservative heuristics.

The interaction flow for PhishDuck is shown in Figure 3. If an email is good, nothing is shown to users, so as to minimize annoyance. If it is unclear if an email is good or not, PhishDuck's redundancy interface is shown, to get extra information from the user to verify that they are going to the site they intend to go to. Finally, if an email is suspicious, the PhishDuck warning interface is shown. The goal of the warning interface is to warn people that the email has a strong potential of being phish, to educate people about phishing attacks, and to get additional verification from users before allowing them to proceed to a potential phish web site.

A question here is, if an email is suspicious, why not simply filter it? We assume that the majority of phishing emails will be filtered out and never shown to users. However, no filter is 100% accurate, and so users will still have to make decisions. PhishDuck is designed for these situations, to help guide people towards better and

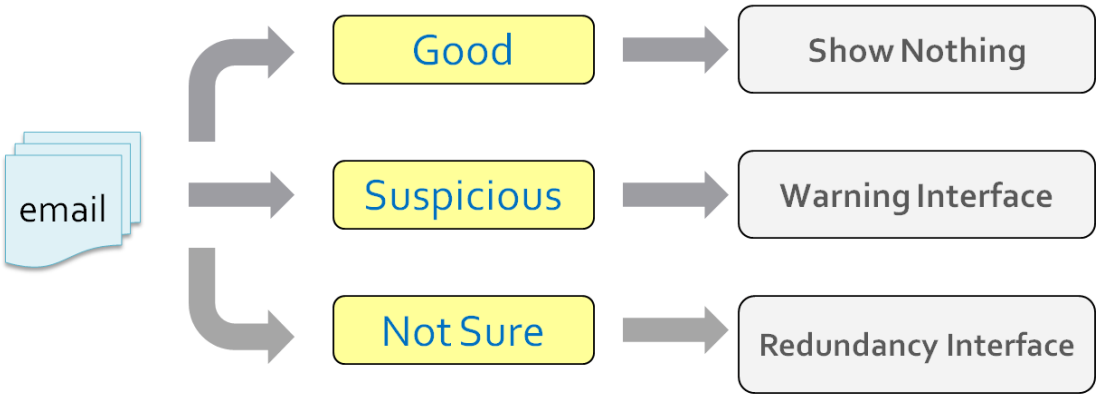


**Figure 3** How different kinds of emails are handled in PhishDuck. Emails are organized into three categories, each of which has a different UI.

safer decisions.

Below, we describe our implementation of PhishDuck. It is important to emphasize that the main contribution here is the evaluation of the interfaces rather than the specific algorithms. Other techniques and algorithms could be used in lieu of the specific ones we used. Furthermore, in our evaluations, we ensured that all of the user interfaces caught the exact same phish, so we can make a direct comparison of the effectiveness of our user interface designs.

# 5. System Design

The overall flow of PhishDuck is shown in Figure 4. PhishDuck starts after a user clicks on a link in an email. The first stage of processing involves filtering using a domain whitelist (Figure 4a), which allows emails from "safe" domain names to be passed through. If the domain name of the link is not in the whitelist, we proceed to the second stage of processing, which involves brand name detection (Figure 4b). The PhishDuck warning interface (Figure 4e) will pop up if a brand name can be found in the content of the message (Figure 4c), otherwise (Figure 4d), PhishDuck's redundancy interface (Figure 4f) will pop up. We describe each part of the system in more detail next.

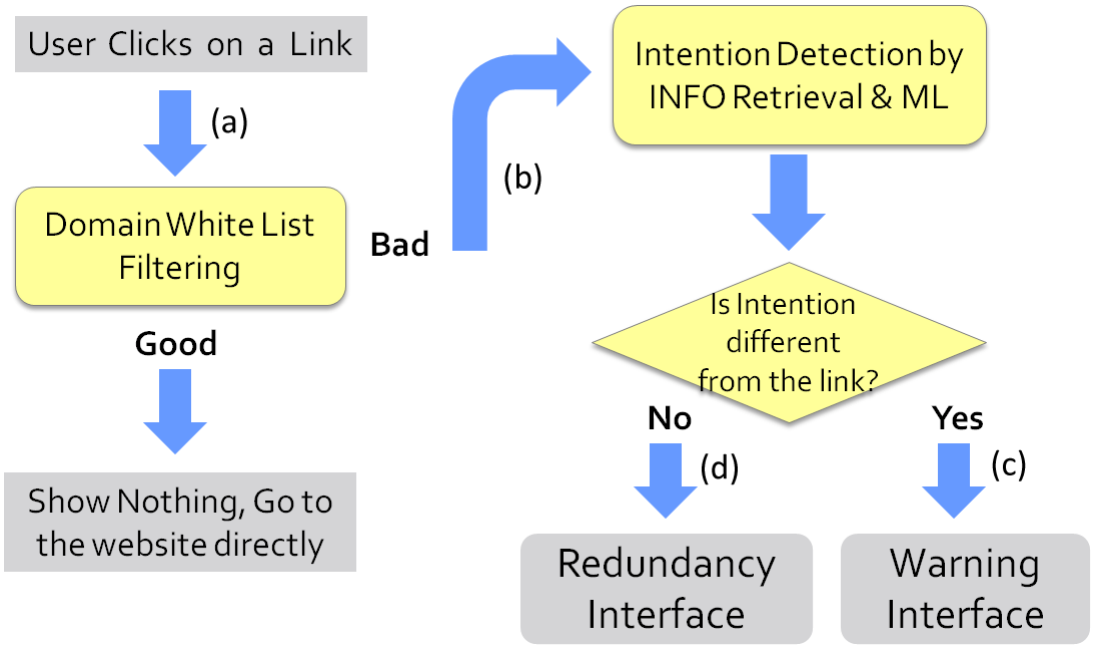


**Figure 4** Interaction flow. Links are filtered with a whitelist (a). If not on the whitelist, we check for user intention in the email (b). If found (c), users are shown a warning interface, otherwise (d) the redundancy interface.

Although we describe our implementation of PhishDuck here, the main contribution of this work is not the system architecture or the algorithms used, but rather the user interface and evaluation. Much of the implementation could be replaced by other similar kinds of techniques.

## 5.1 Domain Whitelist Filtering

To reduce false positives and minimize annoyance to users for legitimate links, we first analyze the domain of the link with our domain whitelist. Our whitelist combines Google Safe Browsing [36] and "millersmiles" [37] (which is a list of 1144 company names which have had phishing attacks before). If the domain of the link is in the whitelist, we show nothing to the user, to reduce annoyance. However, if the domain of the link is not contained in our domain whitelist, it goes to the second stage, described next.

## 5.2 Intention Detection by Information Retrieval

With respect to phishing detection using machine learning techniques, Abu-Nimeh et al. [46] had conducted studies for comparing the predictive accuracy of several machine learning methods including Logistic Regression (LR), Classication and Regression Trees (CART), Bayesian Additive Regression Trees (BART), Sup- port Vector Machines (SVM), Random Forests (RF), and Neural Networks (NNet) for predicting phishing emails. The result showed though RF outperformed all classifiers, it achieved the worst false positive rate of 08.29%. However, unlike predicting spam or phishing, there are only few studies that detect phishing emails based on information retrieval.

Since phishing emails often make use of brand logos and brand names to make them look like legitimate, we created a dictionary of brand names, which is an expanded version of our domain whitelist. If we detect a brand name in the content of the email, the PhishDuck warning interface will pop up. Otherwise, we use the redundancy interface for users to confirm.

However, if there are more than one brand names can be detected, user's intended destination web site needs to be decided through another process before PhishDuck warning interface popped up.

While designing the algorithm for intention detection, we analyzed the location of each detected brand name in the phishing data set [47], and classified the locations into several categories: sender's name, sender's email address, subject, and content. And we found if a brand name can be detected in the sender's name, it should be the user's intended destination web site when clicking on links. Therefore, we gave the brand name in sender's name the highest weight. The second higher weight for deciding user's intended destination was given to the brand name in the email subject, and the lightest weight was given to the brand name in the email content.

## 5.3 Datasets

Two datasets were used to test our implementation: the publicly available phishingcorpus [47] collected between August 7, 2006 and August 7, 2007 (approximately 2279 email messages), and the legitimate portion of the data set contained 2279 email messages were collected from our own mailboxes.

## 5.4 PhishDuck Warning Interface

If the domain name of the link that the user clicks on is not in whitelist, and a brand name has been found, the user will see a PhishDuck warning interface (figure 5a) popping up to warn the user that the link is suspicious. As noted earlier, we designed this interface assuming people would not read the actual text, but would skim the text in the buttons.

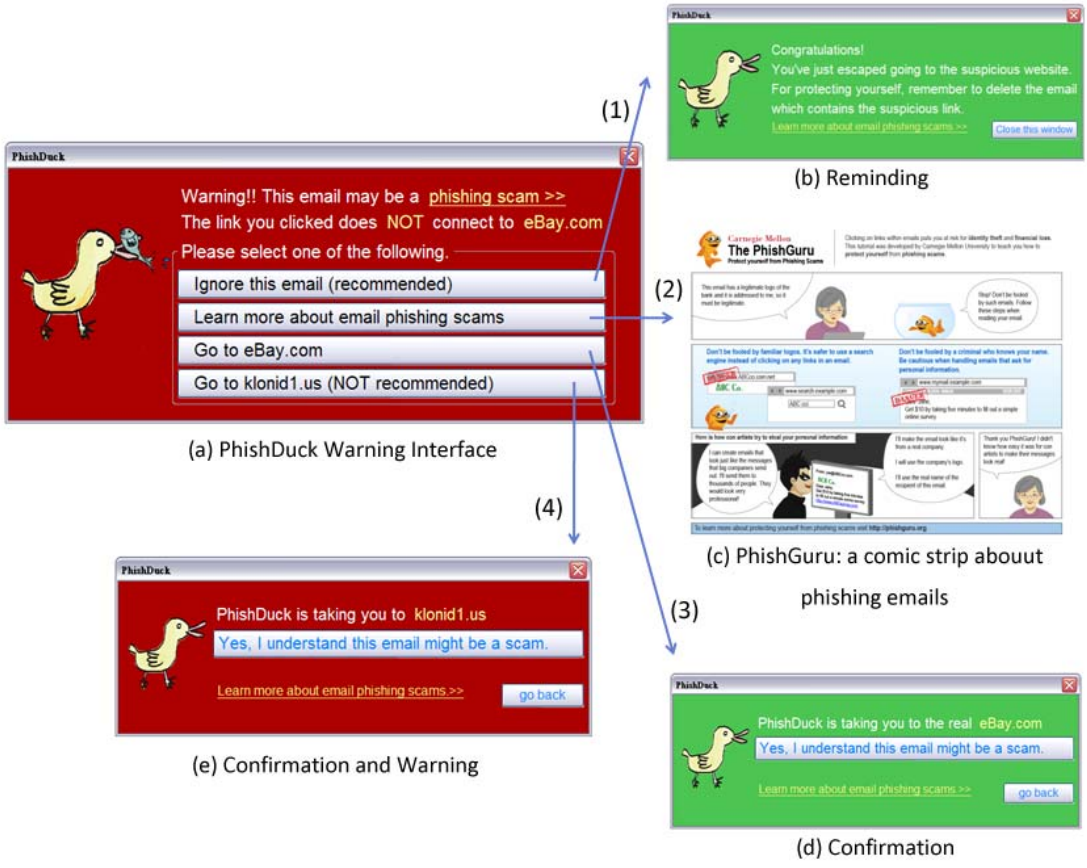


**Figure 5** PhishDuck warning interface. This warning interface is shown to the user if the link s/he clicks on is suspicious. There are four options for the user to choose, each of which has corresponding UI for the further instruction.

To illustrate each part of the interface, we will use a fake eBay phish with link going to “klonid1.us” rather than “ebay.com.” After clicking on the fake link, the user sees the warning interface, with a logo on the left and a short warning on the right. There is also a link that teaches people more about phishing scams, which goes to an education page previously developed for the PhishGuru system [39] (see Figure 5c). Users are also presented with four options:

- *Ignore this email (recommended).* This option closes the PhishDuck dialog and shows a reminder to delete the email with suspicious links (see Figure 5b)
- *Learn more about email phishing scam.* This option opens a web browser showing a PhishGuru education page (see Figure 5c)
- *Go to ebay.com.* This option goes to a confirmation box that says “Yes, I understand this email might be a scam.” We remind users that the email may be a scam because in our pilot studies, we found some users used PhishDuck to safely go to their intended websites but did not understand that they were on a fake site (and so did not see anything about, e.g., their account being closed).

*Go to klonid1.us (NOT recommended).* This option lets people go to the linked site in case of false positives.

## 5.5 PhishDuck Redundancy Interface

If the domain of the clicked link is not in our whitelist, and if no brand name is detected, users are shown the redundancy interface (see Figure 6). In this case, PhishDuck is not certain if the email is phish or not, and requires help from the end-user to disambiguate.

In our example, we take an email we used in our user study for example, which directs people to “pillspad.com.” In this case, “pillspad” is neither in our whitelist nor our brand name dictionary, and so our user is shown the redundancy interface (Figure 6a) if he clicks on the link. The user then has four choices:

- *Learn more about email phishing scams*. Choosing this option is the same as in the warning interface
- *No, I want to go to another site*. This option lets users type in words that would be then re-directed to the Google search engine (see Figure 6c).
- *Search for information about ‘pillspad’*. This option re-directs users to a Google search for “pillspad”

*Take me there (pillspad)*. This option lets users go to a potentially suspicious site, in case of false positives. Users are warned again before going (see Figure 6d). As this option is somewhat risky, we made it the last option.

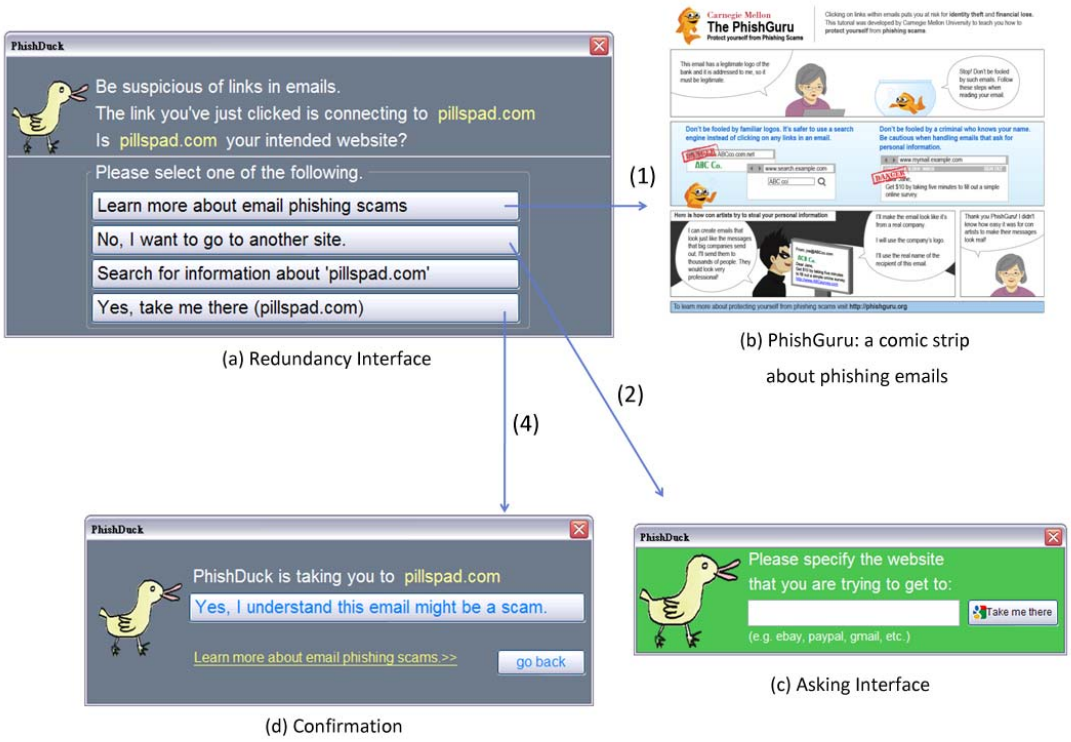


**Figure 6** Redundancy of PhishDuck interface. This interface (a) is shown to the user if we are not sure the legitimacy of the email he gets after he clicks on a link inside. There are four options for the user to choose, each of which has corresponding UI for the further instruction.

# 6. Evaluation

We compared our two warning interfaces with the default Thunderbird Phish Detector interface. We had three conditions, each of which had a different warning interface. There were 10 participants in each condition for a total of 30 participants.

## 6.1 Recruitment

This study was designed as a between-subjects study. We told participants that we were studying email client usage rather than online security, since we did not want to prime people to think about security. We recruited participants from the Pittsburgh metropolitan area. We posted flyers around our university and community bulletin boards. We also posted online to Craigslist and to a university website [41] for recruiting study participants.

These potential participants were then directed to an online screening survey. We used this survey to gather basic demographics from participants. We also screened out both technically savvy individuals (since we did not feel that they were representative of our target users) as well as participants in our previous phishing studies. To gauge technical ability, we asked if users understood the difference between “http” and “https”, and if they could correctly specify the domain name in a simple URL.

Participants were randomly placed in one of three groups: (1) the “PhishDuck with Redundancy group” which used all of the interfaces described (see Figure 4); (2) the “Redundancy group” which only used our redundancy interface (see Figure 6); or (3) the “Default Thunderbird group” which used Thunderbird’s phishing scam detector.

Table 1 shows the demographics of our participants.

| | **Phishduck w/ Redundancy** | **Redundancy** | **Default Thunderbird** |
|---|---|---|---|
| Male | 50% | 40% | 20% |
| Female | 50% | 60% | 80% |
| Average Age | 24.8 | 23.3 | 23 |

**Table 1** Demographics of the participants

## 6.2 Method

The study protocol that we used for this study was essentially the same as previously used by Kumaraguru et al [15]. We highlight the details below.

We used a 1.7GHz IBM laptop running Microsoft Windows XP professional edition, and connecting to a Dell 19-inch Flat Panel Monitor to conduct the user studies. The participants used Mozilla's Thunderbird 2.0 email client for accessing emails and used Firefox 2.0 web browser for accessing the internet.

The user study consisted of a think-aloud session in which participants were given an identity to role play (Pat Jones, an employee at Scolien), with a sheet containing identification, account information, login passwords for Gmail, eBay, PayPal, Facebook, and Bank of America, and a note with the names and email addresses of friends, colleagues, and classmates (one of whom was also a sender of an email in the study). Participants were told that we were studying how people handle emails, and that they should interact with the email the way they would normally.

We also gave each participant a quick tutorial describing how to perform simple actions with Thunderbird (e.g. reply, delete, tag). We also mentioned that we would be able to answer questions about using Thunderbird during the study, but we would not be able to help them make any decisions. We asked participants a few pre-study questions about their use of email to reinforce the idea that this was a study about the usage of email client. We recorded the screen interactions and audio using Camtasia [42].

Each participant was shown a mailbox containing 17 email messages which content were different than what we used in the previous study [15]. These emails were arranged in a predefined order and were assembled into a mailbox for participants to access. Participants were asked to read these emails in chronological order.

In the mailbox, there were six legitimate messages that from co-workers at Scolien, friends, classmates, and school. These emails expected participants to perform simple tasks such as replying, visiting websites, or just reading. The mailbox also contained one spam email, four advertisement emails, and six phishing emails. Table 2 shows the email distribution shown to the users, and Table 3 shows the content of each phishing email.

| | | | |
|---|---|---|---|
| 1. Legitimate | 6. Legitimate | 11. Phishing | 16. Phishing |
| 2. Legitimate | 7. Spam | 12. Phishing | 17. Advertisement |
| 3. Legitimate | 8. Phishing | 13. Advertisement | |
| 4. Phishing | 9. Legitimate | 14. Phishing | |
| 5. Legitimate | 10. Advertisement | 15. Advertisement | |

**Table 2** Email arrangement in the study

| Email | Relevant features of email and sites |
|---|---|
| #4. Facebook | • welcome to Facebook<br>• button of link: www.facebook.com<br>• actual URL: www.facebook.com.login.us |
| #8.Paypal | • warning: Different email address added to this account<br>• link: https://www.paypal.com/us/wf/remove-email.php<br>• actual URL: www.paypal-secure.com |
| #11. eBay | • warning: User ID is Linked to a Suspended User<br>• link: pages.ebay.com/help/policies /rfe-previously-suspended.html<br>• actual URL:signin.ebay.com.klonid1.us |
| #12. Bank of America | • warning : account locked<br>• link: https://www.bankofamerica.com/signin/<br>• actual URL: bankofamerica.updating-database.com |
| #14. Amazon | • warning: Please Update Profile<br>• link: https://amazon.com/account_signin /webscr_/verify/index.html<br>• actual URL: www.amazonaccounts.org |
| #16. Chase Bank | • Reward: do an online survey for earning $20<br>• link: https://www.chase.com/actual URL:chaseonline.chase.com.hentars.net |

**Table 3** Phishing emails in Pat Jones' mailbox

All the phishing, spam, and advertisement emails that we used for this study were based on actual emails we had collected. We created exact copies of the phishing websites on our local machine by running Apache and modifying the host files in Windows so that Firefox would display the URL of the actual phishing websites without showing warnings. All copied phishing websites were completely functional and allowed people to submit information.

# 7. Result

We consider someone to have fallen for a phishing scam if they click on a link in one of our simulated phishing emails and then provide personal information on the corresponding simulated phishing website. We do not consider someone to have fallen for a phishing scams if they go to the phishing websites without providing personal information, since we found some participants go to the phishing websites just want to compare their appearance and addresses with the legitimates ones. However, in practice, even visiting a fake site is risky due to the chance of “drive-by” malware that installs itself using an exploit in the web browser.

## 7.1 Results of Condition #1: PhishDuck with Redundancy

In this group, participants used Thunderbird with the PhishDuck extension installed. Once participants click on a link in an email, they would see PhishDuck’s warning popped up if the link is suspicious (see figure 5).

Table 4 shows that none of the ten participants in this condition went to any of the phishing websites or entered personal information after they saw our PhishDuck warning. Moreover, using Fisher’s exact test, we found that both the number of the participants using PhishDuck extension visited the simulated phishing websites or provided personal information on those websites significantly less than the number of those using default Thunderbird phish detector ($p < 0.01$).

| Condition Name | Clicked | Visited | Phished |
|---|---|---|---|
| PhishDuck with Redundancy | 10(100%) | 0(0%) | 0(0%) |
| Redundancy | 10(100%) | 2(20%) | 1(10%) |
| Thunderbird Phish Detector | 10(100%) | 7(70%)* | 7(70%)* |

**Table 4** Number of participants (by condition) who clicked at least one phishing URL, visited at least one phishing site, and provided personal information on at least one phishing site. For example, all 10 participants in the Thunderbird Phish Detector group clicked at least one phishing URL. Of these, seven provided personal information on at least one of the phishing websites. The asterisk signifies a significant difference between this condition and its linked condition.

In detail, eight of the ten participants chose the first option at least once - “ignore this email (recommended)”, six chose the third option - “go to legitimate website”, and two chose the second option - “learn more about email phishing scams” and then read it.

In the post study session, we asked participants if they understood what this interface meant, and all of the participants (10) did answer correctly. For example:

- “The link given in the mail is fake and it is an incident of phishing.”
- “The link I clicked may not be taking me to the site it said it was.”
- “Email might be fraudulent.”

We also asked participants for general feedback, which was quite positive overall. One participant also suggested adding sound to provide redundant cues of the warning. Other comments included “I like the idea of this PhishDuck software. And I

do like the color and learn more pages. I think I will keep using it!!:) And probably recommend it to my family and friends!" Another person said that "PhishDuck is a great program for people who aren't particularly familiar with phishing. I would definitely install this on my parents' computer as they don't really understand web safety. I doubt I would use this as my spam filter removes this garbage from my inbox."

## 7.2 Results of Condition #2: Redundancy

To test the effectiveness and usability of simply using redundancy, participants in this condition used Thunderbird with PhishDuck only having its redundancy feature turned on (see Figure 6). Once participants click on a link in an email, they would see the redundancy dialog popped up if the link was deemed potentially suspicious.

Table 4 shows that after seeing the PhishDuck Redundancy interface, two of the ten participants went to at least one phishing website, and only one entered personal information. In this case, the participant fell for "paypal-secure.com" (which would have been caught using the brand name detection described earlier).

Moreover, using Fisher's exact test, we found that the number of participants here that provided personal information was significantly less than in the Thunderbird phish detector condition ($p < 0.05$).

In detail, five participants (50%) chose the first option at least once - "learn more about email phishing scams." and then read it. Four chose the second option - "No, I want to go to another site". Six of the ten participants chose the third option - "search

for information about ‘domain name of the link’”. Finally, five of the ten participants closed the redundancy dialog and then deleted the email after they saw the interface.

In the post study session, we asked participants if they understand what this interface meant, and all of the participants did answer correctly (similar to the above condition). General feedback was also positive. One person captured the essence of PhishDuck well: “I think that it worked very well. Instead of just telling me that the link was about to take me to another website, it gave me the option of getting more information about the site that the link was trying to take me to, and helped me to get to the site I actually wanted to go to. I think that PhishDuck seems like a very helpful program.”

## 7.3 Results of Condition #3: Default Thunderbird

Participants in this group used Thunderbird with its default phish warning (see Figure 7). This feature works by detecting a mismatch between the visible link and the underlying URL [43]. While this feature alone yields false positives and false negatives in practice, we made sure that this feature would only activate on all six phishing emails we used in our study. Again, our goal here is to evaluate the effectiveness of the user interface rather than the underlying detection algorithm.

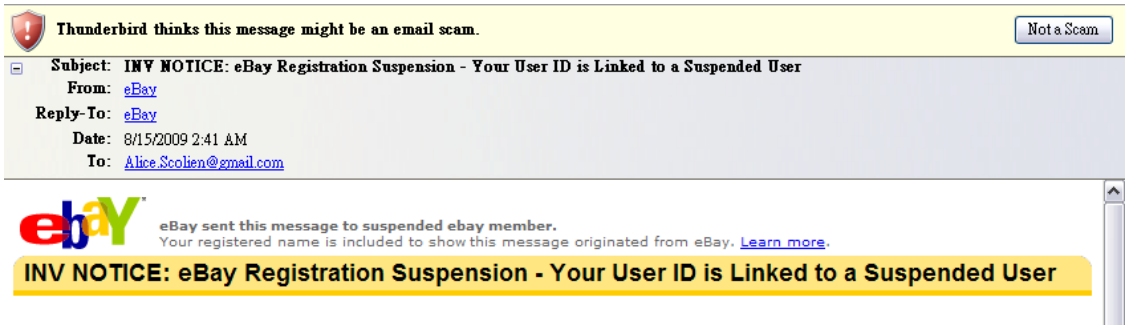


**Figure 7** First warning interface of Thunderbird Phish Detector. It shows a banner above the message: “Thunderbird thinks this message might be an email scam” if it finds a mismatch between the visible link and the underlying URL.

Thunderbird's phish detector uses two mechanisms to warn users. The first is to show the warning "Thunderbird thinks this message might be an email scam" under the subject of the suspicious email (see Figure 7).

In our post-study questionnaire, three of the ten participants did not recall seeing the warning, and another three saw the warning but did not understand it. One person commented, "Not sure, I thought it was a bug of Thunderbird, so I just ignored it. Whatever, I opened this e-mail anyway." Only four participants in this condition saw this warning and said they understood its meaning (Figure 7).

The second warning of Thunderbird's phish detector is popping up an email scam alert when clicking on an email (see Figure 8), though this was not effective in our studies. Every participant who followed a link from a simulated phishing message also provided personal information to the simulated phishing websites (Table 4).


**Figure 8** The second warning interface of Thunderbird Phish Detector. It pops up a warning dialog if the user clicks on the link in the possible scam email (in Figure 7).

# 8. Comparison

During the post-study session, we also asked the participants several questions probing their attitudes.

*"On a scale of 1 to 7, where 1 is not at all confident and 7 is most confident, how confident will you be while clicking on links in emails with the helping of this warning interface?"*

In the PhishDuck with Redundancy group the values ranged from 3 to 7 (average = 6.0, s.d. = 1.2), in the Redundancy group values ranged from 4 to 7 (average = 6.0, s.d. = 1.15), and in the default Thunderbird group values ranged from 2 to 7 (average = 4.1, s.d. = 1.37)

There is a significant difference of the confidence between default Thunderbird group and either PhishDuck with Redundancy group or the Redundancy group (both $p < 0.01$), while there is no significant difference the two PhishDuck groups.

*"On a scale of 1 to 7, where 1 is strongly disagree and 7 is strongly agree, how much do you agree or disagree with the following statement: The warning interface can protect me from connecting to a Phishing (fraudulent) website."*

In the PhishDuck with Redundancy group the values ranged from 5 to 7 (average = 6.4, s.d. = 0.70), in the Redundancy group values ranged from 4 to 7 (average = 6.4, s.d. = 1.07), and in the default Thunderbird group values ranged from 1 to 6 (average

= 3.8, s.d. = 1.69)

There is a significant difference of the result between the default Thunderbird group and either PhishDuck with Redundancy group or the Redundancy group (both $p < 0.01$), while there is no significant difference the two PhishDuck groups.

# 9. Discussion

In this section we discuss some of the implications and limitations of this current work.

## 9.1 Interrupting the primary task

As has been noted before for web toolbars [12] and browser warnings [18], phishing warnings for email clients need to interrupt the user's task. We found that the passive indicator used in Thunderbird, which did not interrupt the user's task, was not effective in protecting people.

## 9.2 How phishers could respond

Although our goal in this paper is evaluating the effectiveness of the user interface rather than the underlying detection algorithm, phishers are still likely to respond should tools like PhishDuck be more widely deployed. As noted earlier, our technique for detecting brand names is relatively simple and could be evaded by any competent criminal. To a large extent, this is why we also included the redundancy interface.

It is also possible that phishers would try to include interfaces like PhishDuck inside their emails (the so-called picture-in-picture attack), which would likely confuse many users. Here, there needs to be a clearer delineation between what is associated with the email client and what is the content associated with the sender of an email.

## 9.3 Habituation

It is difficult to evaluate habituation in a single study, so we do not make any strong claims about the effectiveness of our interface in this regards. It is possible that the

positive results were due to novelty effect as well, which will need to be examined in future work.

## 9.4 Findings

However, one positive outcome was that none of our participants in the two PhishDuck conditions explicitly complained about being annoyed. Furthermore, in the warning and redundancy condition, none of our participants fell for the phish, which may suggest that habituation went in the correct direction, towards safety. To a large extent, this follows the design pattern suggested by Egelman [44], that the default choice should be safe, with our first and second options being safe ones.

Our findings also suggest two intriguing research directions for the design of security warnings in general. The first is ordering, namely, can our informal findings about ordering of choices generalize to other security warnings? If so, this could be a simple yet highly effective way of nudging people towards making safer decisions.

The second is labeling of options. We intentionally designed PhishDuck assuming that few people would read the formal warning, and instead placed choices in the buttons assuming that more people would read those. Is this also a generalizable design pattern for security warnings? If so, this would be a simple change to most user interfaces yet could also lead to more effective designs for warnings.

## 9.5 False positives and false negatives

In our user study, we did not have any false positives (real email labeled as phish) or false negatives (phish email labeled as legitimate). We assumed the algorithm would be 100% accurate since our focus was on designing and evaluating user interfaces. As

such, in the user study, our warning interface only showed when the email was actually a phishing email, though our redundancy interface did show for some legitimate emails. This is a limitation of the current study, and will be the subject of some of our future work.

# 10. Conclusion and Future Work

In this paper, we presented the design and evaluation of PhishDuck, an anti-phishing tool for email clients. PhishDuck has two different types of interfaces. The first is a warning interface, which is shown when for suspicious emails. This warning interface warns users that the email has a strong potential of being phish, educates them about phishing attacks, and tries to steer users away from potential phish. The second is a redundancy interface, which gets extra information from users to verify that they are going to the site they intend to go to.

The Phishduck warning interface was significantly better than the interface for Thunderbird, with the rate of falling for phish falling from 70% to 0%. Our redundancy interface alone was also significantly better than Thunderbird, with phish rates decreasing from 70% to 10%.

In future work, we plan on refining the machine learning algorithm of PhishDuck and preparing for a wider-scale deployment and evaluation.

# 11. Learning Experience

## 11.1 How to do research

I learned an invaluable experience from this practicum. First, I have developed my research skills throughout the project. I started the project by writing the proposal on January till presented my work on November. During this period of time, I did much literature survey, exchanged experiences with other Ph.D. students, weekly met with Prof. Hong, while also implementing and evaluating PhishDuck. Especially, submitting a paper to CHI was an invaluable experience for me.

Doing practical research in the United States also helped me to improve my language skills including reading, writing, speaking, and listening. Also, I had many chances to talk to researchers, Ph.D students, and faculties in different fields at CMU. I learned more about researcher, Ph.D. student, and academic life. This experience also inspired me to continue my research life.

## 11.2 User Interface Design

There were several rounds for the user interface design for PhishDuck. To balance the tradeoff between effectiveness, usability, and comprehensibility of the interfaces for end-users, I modified the PhishDuck interface several times based on Egelman's design patterns for warning interfaces and feedbacks from Prof. Hong and many pilot studies. Through this experience, I learned the lessons of C-HIP model for warnings and the importance of pilot studies.

## 11.3 Design and Conduct User Studies

In this project, I evaluated the effectiveness and usability of PhishDuck through pilot studies and formal user studies. Through this experience, I learned the way of conducting user studies, including writing IRB documents, recruitment, designing pre-study and post-study surveys, and conducting between-subjects user studies. Also, I learned many concepts of HCI methods, such as paper prototyping and think-aloud protocol.

Conducting user studies was also a wonderful experience. I experienced around 45 participants from the Pittsburgh metropolitan area for the pilot studies and the formal user studies. Learning how people think was interesting, especially, many participants had different opinions. I learned how to combine their opinions and feedback to improve the design of PhishDuck.

## 11.4 Implantation Skills

Current PhishDuck is an extension for Mozilla Thunderbird, and it was developed using JavaScript, XUL, and CSS. It was the first time for me to develop applications in the Mozilla development platform. It was valuable to gain experience in development in the combination of languages.

# Bibliography


1. Study: 'The war on phishing is far from over'. http://www.creditcards.com/credit-card-news/phishing-attacks-increase-research-1282.php.
2. APWG releases Phishing report for second half of 2008. http://www.creditcards.com/credit-card-news/phishing-attacks-increase-research-1282.php.
3. Phishing Activity Trends Report. 2nd Half / 2008. http://www.antiphishing.org/reports/apwg_report_H2_2008.pdf
4. Gartner Says Number of Phishing Attacks on U.S. Consumers Increased 40 Percent in 2008. http://www.smartbrief.com/news/aaaa/industryBW-detail.jsp?id=C87DC2AF-1BA9-4A16-9717-A07ACBB7BCDF
5. Brustoloni, J C, Villamarin-Salomon. Improving Security Decisions with Polymorphic and Audited Dialogs. SOUPS 2007, ACM Press (2007), 77-87
6. Fette, I., N. Sadeh and A. Tomasic. Learning to Detect Phishing Emails. Sep 2009. ISRI Technical report, CMU-ISRI-06-112. http://reports-archive.adm.cs.cmu.edu/anon/isri2006/CMU-ISRI-06-112.pdf.
7. Ludl, C., McAllister, S., Kirda, E., Kruegel, C. On the Effectiveness of Techniques to Detect Phishing Sites. Detection of Intrusions and Malware, Spring (2007), 20-39
8. Pan, Y. and Ding, X. Anomaly based web phishing page detection. In Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06), pages 381–392, 2006.

9. Zhang, Y., Hong, J., Cranor, L. Cantina: A Content-based Approach to Detecting Phishing Websites. Proc. WWW07, (2007), 639-64

10. Doshi, S., Provos, N., Chew, M., Rubin, A.D. A framework for detection and measurement of phishing attacks. In Proceedings of the 2007 ACM Workshop on Recurring Malcode, pages 1–8, 2007.

11. Xiang, G. and Hong, J. A hybrid phish detection approach by identity discovery and keywords retrieval. In Proceedings of the 18th International Conference on World Wide Web (WWW'09), 2009.

12. Wu, M., Miller, R., Garfinkel, S. Do Security Toolbars Actually Prevent Phishing Attacks? Proc. CHI 2006, ACM Press (2006), 601-610.

13. Wu, M., Miller R., Little, G. Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. Proc. SOUPS 2006, ACM Press (2006), 102-113.

14. SpamAssassin. Retrieved September 13, 2009, http://spamassassin.apache.org.

15. Kumaraguru, P., Rhee, Y., Acquisti, A., Cranor, L., Hong, J., and Nunge, E. Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. In *Proc. CHI 2007*, ACM Press (2007), 905-14

16. Messages asking for personal information http://mail.google.com/support/bin/answer.py?hl=en&ctx=mail&answer=8253

17. PhishPatrol. http://www.wombatsecurity.com/phishpatrol

18. Egelman, S., Cranor, L., and Hong, J. You've Been Warned: An Empirical Study on the Effectiveness of Web Browser Phishing Warnings. In *Proc. CHI 2008*, ACM Press (2008)

19. Sender Policy Framework. http://www.openspf.org/

20.Ferguson, A.J. Fostering E-Mail Security Awareness: The West Point Carronade. EDUCASE Quarterly, (1), 2005.

21.Jagatic, T., Johnson, N., Jakobsson, M. and Menczer, F. Social phishing. Communications of the ACM, 50(10):94–100, October 2007.

22.Morin, R.A. and Fernandez Suarez, A. Risk aversion revisited. Journal of Finance, 38(4):1201–16, September 1983.

23.Jagatic, T., Johnson, N., Jakobsson, M. and Menczer, F. Social phishing. Communications of the ACM, 50(10):94–100, October 2007.

24.Ferguson, A.J. Fostering E-Mail Security Awareness: The West Point Carronade. EDUCASE Quarterly, (1), 2005.

25.New York State Office of Cyber Security & Critical Infrastructure Coordination. Gone phishing… a briefing on the anti-phishing exercise initiative for new york state government. Aggregate Exercise Results for public release, 2005.

26.Kumaraguru, P., Cranshaw, J., Acquisti, A., Cranor, L., Hong, J., Blair, M.A. and Pham, T. School of Phish: A Real-Word Evaluation of Anti-Phishing Training. SOUPS 2009.

27.Sheng, S., Magnien, B., Kumaraguru, P., Acquisti, A., Cranor, L.F., Hong, J. and Nunge, E. Anti-phishing phil: The design and evaluation of a game that teaches people not to fall for phish. In SOUPS ’07: Proceedings of the 3rd symposium on Usable privacy and security, pages 88–99, 2007.

28.Yee, K.P., Sitaker K. Passpet: Convenient Password Management and Phishing Protection. Proc. SOUPS 2006, ACM Press (2006), 32-43.

29.PassMark Security. http://www.rsa.com/

30.Dhamija, R, Tygar J D. The Battle Against Phishing: Dynamic Security Skins. Proc. SOUPS 2005, ACM Press (2005), 77-88

31.Ronda, T., Saroiu, S., and Wolman, A. iTrustPage: A User-Assisted Anti-Phishing Tool. Proc. EuroSys'08.

32.Likarish, P., Dunbar, D., Hourcade, J.P., Jung, E. BayeShield: conversational anti-phishing user interface. Proc. SOUPS 2009, 26.

33.Jakobsson, M., Juels, A., and Ratkiewicz, J. Remote harm-diagnostics. Retrieved, Sep, 2009, http://www.ravenwhite.com/files/rhd.pdf.

34.DomainKeys Identified Mail (DKIM) http://www.dkim.org/

35.Garfinkel, S.L., Miller, R.C. Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. In *Proc. SOUPS 2005*

36.Google safe browsing http://sb.google.com/safebrowsing/update?client=navclient-auto-ffox&appver=2.0.0.11&version=goog-white-domain:1:23

37.millersmiles. http://www.millersmiles.co.uk/scams.php

38.PhishTank. http://www.phishtank.com/

39.PhishGuru. http://phishguru.org/

40.Google, I'm Feeling Lucky http://www.google.com/search?hl=en&source=hp&q=&btnI=I%27m+Feeling+Lucky&aq=f&oq=&aqi=

41.Center for Behavioral Decision Research at Carnegie Mellon http://www.cbdr.cmu.edu/

42.Camtasia Studio, TechSmith's Screen Recording Software http://www.techsmith.com/camtasia.asp

43.Mozilla, mail and news settings http://kb.mozillazine.org/Mail_and_news_settings

44.Egelman, S. Trust Me: Design Patterns for Constructing. Trustworthy Trust Indicators.

CMU-ISR-09-110 http://reports-archive.adm.cs.cmu.edu/anon/isr2009/CMU-ISR-09-110.pdf

45.Sheng, S., P. Kumaraguru, A. Acquisti, L. Cranor, J. Hong. Improving Phishing Countermeasures: An Analysis of Expert Interviews. APWG eCrime Researcher's Summit (eCrime 2009). To Appear.

46.Abu-Nimeh, S., Nappa, D., Wang, X. and Nair, Suku. A comparison of machine learning techniques for phishing detection. Proc. APWG eCrime Researcher's Summit (eCrime 2007).

47.J. Nazario. Phishing corpus. http://monkey.org/~jose/phishing/phishing3.mbox.