# CANTINA: A Content-Based Approach to Detecting Phishing Web Sites

Yue Zhang
Dept of Computer Science
University of Pittsburgh
210 South Bouquet Street
Pittsburgh, PA 15260
zysxqn@cs.pitt.edu

Jason Hong
Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
jasonh@cs.cmu.edu

Lorrie Cranor
Institute for Software Research
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
lorrie@cs.cmu.edu

## ABSTRACT
Phishing is a significant problem involving fraudulent email and web sites that trick unsuspecting users into revealing private information. In this paper, we present the design, implementation, and evaluation of CANTINA, a novel, content-based approach to detecting phishing web sites, based on the TF-IDF information retrieval algorithm. We also discuss the design and evaluation of several heuristics we developed to reduce false positives. Our experiments show that CANTINA is good at detecting phishing sites, correctly labeling approximately 95% of phishing sites.

## Categories and Subject Descriptors
C.2.0 [Computer-Communication Networks]: *General – Security and Protection*, H.3.3 [Information Search and Retrieval]: *Retrieval Models*

## General Terms
Algorithms, Measurement, Security, Human Factors

## Keywords
Phishing, Anti-Phishing, TF-IDF, Toolbar, Evaluation

## 1. INTRODUCTION
Recently, there has been a dramatic increase in phishing, a kind of attack in which victims are tricked by spoofed emails and fraudulent web sites into giving up personal information. Phishing is a rapidly growing problem, with 9,255 unique phishing sites reported in June of 2006 alone [2]. It is unknown precisely how much phishing costs each year since impacted industries are reluctant to release figures; estimates range from $1 billion [24] to 2.8 billion [27] per year.

To respond to this threat, software vendors and companies have released a variety of anti-phishing toolbars. For example, eBay offers a free toolbar that can positively identify eBay-owned sites, and Google offers a free toolbar aimed at identifying any fraudulent site [12, 19]. As of September 2006, the free software download site download.com, listed 84 anti-phishing toolbars. However, when we conducted an evaluation of ten anti-phishing tools for a previous study, we found that only one tool could consistently detect more than 60% of phishing web sites without a high rate of false positives [6]. Thus, we argue that there is a strong need for better automated detection algorithms.


*WWW 2007*, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

In this paper, we present the design, implementation, and evaluation of CANTINA,[1] a novel content-based approach for detecting phishing web sites. CANTINA examines the content of a web page to determine whether it is legitimate or not, in contrast to other approaches that look at surface characteristics of a web page, for example the URL and its domain name. CANTINA makes use of the well-known TF-IDF (term frequency/inverse document frequency) algorithm used in information retrieval [35], and more specifically, the Robust Hyperlinks algorithm previously developed by Phelps and Wilensky [32] for overcoming broken hyperlinks. Our results show that CANTINA is quite good at detecting phishing sites, detecting 94-97% of phishing sites. We also show that we can use CANTINA in conjunction with heuristics used by other tools to reduce false positives (incorrectly labeling legitimate web sites as phishing), while lowering phish detection rates only slightly.

We present a summary evaluation, comparing CANTINA to two popular anti-phishing toolbars that are representative of the most effective tools for detecting phishing sites currently available. Our experiments show that CANTINA has comparable or better performance to SpoofGuard (a heuristic-based anti-phishing tool) with far fewer false positives, and does about as well as NetCraft (a blacklist and heuristic-based anti-phishing toolbar). Finally, we show that CANTINA combined with heuristics is effective at detecting phishing URLs in users' actual email, and that its most frequent mistake is labeling spam-related URLs as phishing.

In Section 2, we review related work. We describe our TF-IDF method in more details in Section 3. Section 4 introduces the experiments we conducted to test the effectiveness of our approach. The results are discussed in Section 5. We wrap up in Section 6 with conclusions and future work.

## 2. RELATED WORK
Generally speaking, past work in anti-phishing falls into four categories: studies to understand why people fall for phishing attacks, methods for training people not to fall for phishing attacks, user interfaces for helping people make better decisions about trusting email and websites, and automated tools to detect phishing. Our work on CANTINA contributes a new approach to the development of automated phishing detection tools.

### 2.1 Why People Fall for Phishing Attacks
A number of studies have examined the reasons that people fall for phishing attacks. For example, Downs et al have described the

[1] Carnegie Mellon Anti-phishing and Network Analysis Tool

results of an interview and role-playing study aimed at understanding why people fall for phishing emails and what cues they look for to avoid such attacks [10]. In a different study, Dhamija et al. showed that a large number of people cannot differentiate between legitimate and phishing web sites, even when they are made aware that their ability to identify phishing attacks is being tested [9]. Finally, Wu et al. studied three simulated anti-phishing toolbars to determine how effective they were at preventing users from visiting web sites the toolbars had determined to be fraudulent [37]. They found that many study participants ignored the toolbar security indicators and instead used the site's content to decide whether or not it was a scam.

## 2.2 Educating People about Phishing Attacks

Anti-phishing education has focused on online training materials, testing, and situated learning. *Online training materials* have been published by government organizations [13, 14], non-profits [3] and businesses [11, 28]. These materials explain what phishing is and provide tips to prevent users from falling for phishing attacks.

*Testing* is used to demonstrate how susceptible people are to phishing attacks and educate them on how to avoid them. For example, Mail Frontier [26] has a web site containing screenshots of potential phishing emails. Users are scored based on how well they can identify which emails are legitimate and which are not.

A third approach uses situated learning, where users are sent phishing emails to test users' vulnerability of falling for attacks. At the end of the study, users are given materials that inform them about phishing attacks. This approach has been used in studies conducted by Indiana University in training students [23], West Point in instructing cadets [15, 22] and a New York State Office in educating employees [30]. The New York study showed an improvement in the participants' behavior in identifying phishing over those who were given a pamphlet containing the information on how to combat phishing. In previous work, we developed an email-based approach to train people how to identify and avoid phishing attacks, demonstrating that the existing practice of sending security notices is ineffective, while a story-based approach using a comic strip format was surprisingly effective in teaching people about phishing [25].

## 2.3 Anti-Phishing User Interfaces

Other research has focused on the development of better user interfaces for anti-phishing tools. Some work looks at helping users determine if they are interacting with a trusted site. For example, Ye et al. [39] and Dhamija and Tygar [8] have developed prototype user interfaces showing "trusted paths" that help users verify that their browser has made a secure connection to a trusted site. Herzberg and Gbara have developed TrustBar, a browser add-on that uses logos and warnings to help users distinguish trusted and untrusted web sites [21].

Other work has looked at how to facilitate logins, eliminating the need for end-users to identify whether a site is legitimate or not. For example, PwdHash [36] transparently converts a user's password into a domain-specific password by sending only a one-way hash of the password and domain-name. Thus, even if a user falls for a phishing site, the phishers would not see the correct password. The Lucent Personal Web Assistant [17] and Password Multiplier [20] used similar approaches to protect people.

PassPet [40] is a browser extension that makes it easier to login to known web sites, simply by pressing a single button. PassPet requires people to memorize only one password, and like PwdHash, generates a unique password for each site.

Web Wallet is web browser extension designed to prevent users from sending personal data to the fake page [38]. Web Wallet prevents people from typing personal information directly into a web site, instead requiring them to type a special keystroke to log into Web Wallet and then select their intended web site.

Our work in this paper is orthogonal to this previous work, in that our algorithms could be used in conjunction with better user interfaces to provide a more effective solution. As Wu and Miller demonstrated, an anti-phishing toolbar could identify all fraudulent web sites without any false positives, but if it has usability problems, users might still fall victim to fraud [37].

## 2.4 Automated Detection of Phishing

Anti-phishing services are now provided by Internet service providers, built into mail servers and clients, built into web browsers, and available as web browser toolbars (e.g., [4, 5, 12, 18, 19, 29]). However, these services and tools do not effectively protect against all phishing attacks, as attackers and tool developers are engaged in a continuous arms race [6].

Anti-phishing tools use two major methods for detecting phishing sites. The first is to use heuristics to judge whether a page has phishing characteristics. For example, some heuristics used by the SpoofGuard [4] toolbar include checking the host name, checking the URL for common spoofing techniques, and checking against previously seen images. The second method is to use a blacklist that lists reported phishing URLs. For example, Cloudmark [5] relies on user ratings to maintain their blacklist. Some toolbars, such as Netcraft [29], seem to use a combination of heuristics plus a blacklist with URLs that are verified by paid employees.

Both methods have pros and cons. For example, heuristics can detect phishing attacks as soon as they are launched, without the need to wait for blacklists to be updated. However, attackers may be able to design their attacks to avoid heuristic detection. In addition, heuristic approaches often produce false positives (incorrectly labeling a legitimate site as phishing). Blacklists may have a higher level of accuracy, but generally require human intervention and verification, which may consume a great deal of resources. At a recent Anti-Phishing Working Group meeting, it was reported that phishers are starting to use one-time URLs, which direct someone to a phishing site the first time the URL is used, but direct people to the legitimate site afterwards. This and other new phishing tactics significantly complicate the process of compiling a blacklist, and can reduce blacklists' effectiveness.

Our work with CANTINA focuses on developing and evaluating a new heuristic based on TF-IDF, a popular information retrieval algorithm. CANTINA not only makes use of surface level characteristics (as is done by other toolbars), but also analyzes the text-based content of a page itself. In Section 3.3, we also discuss some additional heuristics we employed to reduce false positives. These heuristics were drawn primarily from SpoofGuard [4] and from PILFER, an algorithm for detecting phishing emails [16].

# 3. A CONTENT-BASED APPROACH FOR DETECTING PHISHING WEB SITES

CANTINA makes use of TF-IDF for detecting phishing sites. TF-IDF is a well-known information retrieval algorithm that can be

used for comparing and classifying documents, as well as retrieving documents from a large corpus. In this section, we first review how TF-IDF works. We then introduce an application of TF-IDF called Robust Hyperlinks. Finally, we describe how we adapted Robust Hyperlinks for detecting phishing web sites.

## 3.1 How TF-IDF Works

TF-IDF is an algorithm often used in information retrieval and text mining. TF-IDF yields a weight that measures how important a word is to a document in a corpus. The importance increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus.

The *term frequency* (TF) is simply the number of times a given term appears in a specific document. This count is usually normalized to prevent a bias towards longer documents (which may have a higher term frequency regardless of the actual importance of that term in the document) to give a measure of the importance of the term within the particular document. The *inverse document frequency* (IDF) is a measure of the general importance of the term. Roughly speaking, the IDF measures how common a term is across an entire collection of documents.

Thus, a term has a high TF-IDF weight by having a high term frequency in a given document (i.e. a word is common in a document) and a low document frequency in the whole collection of documents (i.e. is relatively uncommon in other documents).

## 3.2 Robust Hyperlinks

Phelps and Wilensky developed the idea of Robust Hyperlinks to overcome the problem of broken links [32]. The basic idea is to provide a number of alternative, independent descriptions of networked resources, that is, URLs. Specifically, Phelps and Wilensky proposed adding a small number of well-chosen terms, which they called a lexical signature, to URLs. An example of such a modified signature might be:

```
http://abc.com/page.html?lexical-signature="w1+w2+w3+w4+w5"
```

When locating a web page, one could first try the basic URL. If the resource cannot be found, one could then supply the signature terms to a search engine to locate the document whose signature most closely matches that in the robust hyperlink.

A key issue here is how to create signatures that have appropriate properties. First, signatures should be effective in picking out few documents. Second, subsequent changes to a document should have minimal impact on signature effectiveness. Third, the addition of new documents should have minimal impact on previous signature effectiveness. Finally, the effectiveness of the signature should be largely search-engine-independent.

To meet these requirements, Phelps and Wilensky proposed using TF-IDF to generate lexical signatures. Specifically, they proposed calculating the TF-IDF value for each word in a document, and then selecting the words with highest value. The rationale here is that term frequency provides robustness (repeated words are less likely to all be deleted), while inverse document frequency provides rarity across a set of documents, minimizing the chance that another document will be added with the same term.

Their preliminary empirical results suggest that lexical signatures of about five terms are sufficient to determine a web resource virtually uniquely, out of the more than one billion pages on the web [32]. Their experiments also showed that searching on lexical signatures often yielded a unique document, namely the desired document. In those few cases in which more than one document is returned, the desired document is among the highest ranked.

In the next section, we describe how we applied this idea of Robust Hyperlinks to anti-phishing.

## 3.3 Adapting TF-IDF for Detecting Phishing

We had two observations that led us to believe that Robust Hyperlinks could be effective for detecting phishing scams. The first is that criminals often create phishing sites by copying and then modifying a legitimate site's web pages so that personal information is redirected to the criminals rather than to the legitimate site. For example, Figure 1 shows a phishing page impersonating eBay, which is identical to the real eBay log-in page shown in Figure 2. We reasoned that if a criminal copied a web page and made minimal modifications, then Robust Hyperlinks could be used to find the original log-in page.

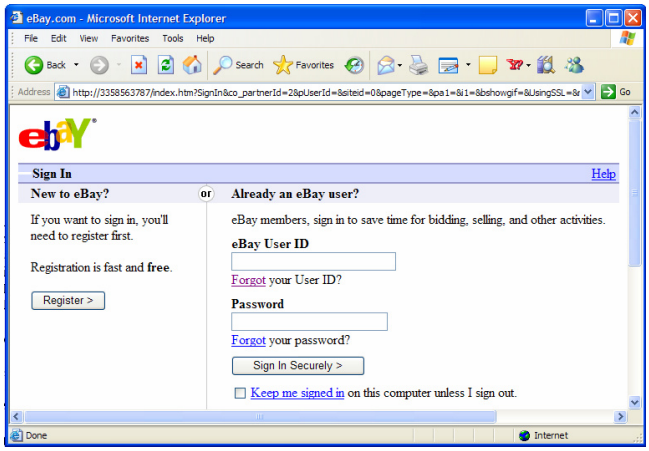


**Figure 1. This phishing site presents an exact copy of eBay's actual login page, except that username and password information is sent to the scam site instead of eBay. The only visual cue that this is not eBay is the URL.**

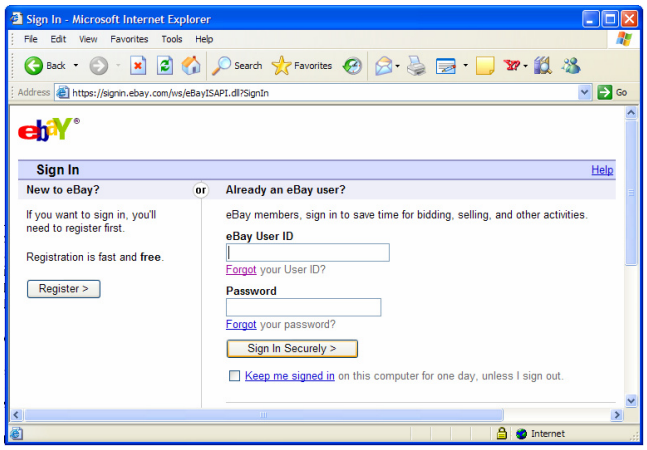


**Figure 2. The real eBay log-in page**

The second observation is that phishing sites often contain brand names and other terms that are common on a given web page but relatively rare across the web, leading us to hypothesize that, again, Robust Hyperlinks could be applied to find the owner of those brands.

Roughly, CANTINA works as follows:

- Given a web page, calculate the TF-IDF scores of each term on that web page.
- Generate a lexical signature by taking the five terms with highest TF-IDF weights.
- Feed this lexical signature to a search engine, which in our case is Google.
- If the domain name of the current web page matches the domain name of the N top search results, we consider it to be a legitimate web site. Otherwise, we consider it a phishing site. (We varied the value of N, as described in the evaluation, to balance false positives with true positives; however, we found that going beyond the top 30 results had little effect.)

Our technique makes the assumption that Google indexes the vast majority of legitimate web sites, and that legitimate sites will be ranked higher than phishing sites. Our experiments (see next section) strongly suggest that both of these assumptions are true.

It is also worth pointing out that, according to the Anti-Phishing Working Group (APWG), the average time that a phishing site stays online is 4.5 days [2]. Our experiences show that sometimes it is on the order of hours [6]. Furthermore, we argue that phishing web pages will have a low Google Page Rank due to a lack of links pointing to the scam. These two factors combined suggest that a phishing scam will rarely, if ever, be highly ranked. At the end of this paper, however, we discuss some ways of possibly subverting CANTINA.

In an earlier implementation, we discovered that TF-IDF alone yields a fair number of false positives, labeling legitimate sites as phishing. To address this problem, we also add the current domain name to the lexical signature. For example, if the page is at http://www.ebay.com/xxxxx, then we add the term "eBay" to the lexical signature (even if it is already there). The rationale here is that if a page is legitimate, the domain name itself usually can best identify itself (e.g., ebay.com, paypal.com, bankofamerica.com). On the other hand, if the suspected page is phishing, no matter what we add onto its content, Google will not return it.

Another design decision was what to do if Google returns zero search results. This sometimes happens because added domain names are sometimes meaningless (for example, "u-s-j.be"). To address this problem, if Google fails to return any result, we now label the suspected site as phishing (initially we labeled it as unknown). We refer to this as the "Zero results Means Phishing" heuristic (ZMP). This heuristic has the potential to increase false positives (incorrectly labeling a legitimate site as phishing), but our early experiments strongly suggest that when combined with adding the domain name to the lexical signature, this approach can reduce false positives while not impacting true positives.

We return to the phishing site shown in Figure 1 to illustrate how our approach works. The top 5 terms used on the page in Figure 1 (as well as eBay's actual log-in page) as calculated by TF-IDF are: *eBay, user, sign, help, forgot*. Figure 3 shows the results of the Google search page. The first result returned by Google has the same domain as the legitimate eBay page shown in Figure 2 (and the fifth result is exactly the page in Figure 2), so the web page in Figure 2 is deemed legitimate. None of the results returned on this search results page match the domain name of the page in Figure 1, so it is deemed to be phishing.

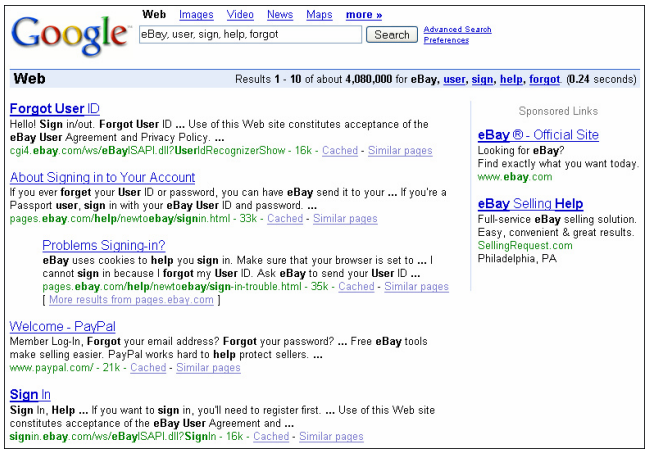


**Figure 3. The lexical signature generated by the web pages shown in Figures 1 and 2 is: *eBay, user, sign, help, forgot*. This screenshot shows search results using Google. The domain name shown of Figure 2 matches the first search result, so it is deemed legitimate. The domain name in Figure 1 does not match any of the top results, so it is deemed phishing.**

We present our evaluation of the effectiveness of TF-IDF and the two heuristics (adding the domain name to the lexical signature and ZMP) in Section 4.1. We discovered that TF-IDF yielded fairly good accuracy (correctly labeling legitimate sites as legitimate and phishing sites as phishing), but also found that it had a fair number of false positives (incorrectly labeling legitimate sites as phishing). To address this problem, we developed a larger set of heuristics and ran an experiment to determine the proper weights to assign to these heuristics, as described in Section 4.2. In Section 4.3 we evaluate the overall effectiveness of TF-IDF plus the heuristics, comparing the results to SpoofGuard and Netcraft. In Section 4.4 we evaluate the effectiveness of CANTINA on phishing URLs gathered from email from four users' inboxes.

We developed our larger set of heuristics based on related work, drawing primarily from SpoofGuard [4] and PILFER [16]. We implemented each heuristic to return either -1 if it looks like a phishing page or +1 otherwise. Section 4.2 describes how we weighted these heuristics. Our heuristics include:

- **Age of Domain** – This heuristic checks the age of the domain name. Many phishing sites have domains that are registered only a few days before phishing emails are sent out. We use a WHOIS search to implement this heuristic. This heuristic measures the number of months from when the domain name was first registered. If the page has been registered longer than 12 months, the heuristic will return +1, deeming it as legitimate, and otherwise returns -1, deeming it as phishing. If the WHOIS server cannot find the domain, the heuristic will simply return -1, deeming it as a phishing page. The Netcraft [29] and SpoofGuard [4] toolbars use a similar heuristc based on the time since a domain name was registered. Note that this heuristic does not account for phishing sites based on existing web sites where criminals have broken into the web server, nor does it account for phishing sites hosted on otherwise legitimate domains, for example in space provided by an ISP for personal homepages.
- **Known Images** – This heuristic checks whether a page contains inconsistent well-known logos. For example, if a

page contains eBay logos but is not on an eBay domain, then this heuristic labels the site as a probable phishing page. Currently we store nine popular logos locally, including eBay, PayPal, Citibank, Bank of America, Fifth Third Bank, Barclays Bank, ANZ Bank, Chase Bank, and WellsFargo Bank. Eight of these nine legitimate sites are included in the PhishTank.com list of Top 10 Identified Targets [34]. A similar heuristic is used by the SpoofGuard toolbar.

- **Suspicious URL –** This heuristic checks if a page's URL contains an "at" (@) or a dash (-) in the domain name. An @ symbol in a URL causes the string to the left to be disregarded, with the string on the right treated as the actual URL for retrieving the page. Combined with the limited size of the browser address bar, this makes it possible to write URLs that appear legitimate within the address bar, but actually cause the browser to retrieve a different page. This heuristic is used by Mozilla FireFox. Dashes are also rarely used by legitimate sites, so we use this as another heuristic. SpoofGuard checks for both at symbols and dashes in URLs.
- **Suspicious Links –** This heuristic applies the URL check above to all the links on the page. If any link on a page fails this URL check, then the page is labeled as a possible phishing scam. This heuristic is also used by SpoofGuard.
- **IP Address –** This heuristic checks if a page's domain name is an IP address. This heuristic is also used in PILFER [16].
- **Dots in URL –** This heuristic checks the number of dots in a page's URL. We found that phishing pages tend to use many dots in their URLs but legitimate sites usually do not. Currently, this heuristic labels a page as phish if there are 5 or more dots. This heuristic is also used in PILFER [16].
- **Forms –** This heuristic checks if a page contains any HTML text entry forms asking for personal data from people, such as password and credit card number. We scan the HTML for `<input>` tags that accept text and are accompanied by labels such as "credit card" and "password". Most phishing pages contain such forms asking for personal data, otherwise the criminals risk not getting the personal information they want.

## 3.4 Implementation

We have implemented CANTINA as a Microsoft Internet Explorer extension. CANTINA is written in C# using the Microsoft .NET Framework 2003, and is comprised of 800 lines of code as well as four freely available libraries, including the Toolbar extension module [41]; a Google search module [31]; and a TF-IDF component for calculating the score for each term [7]. To calculate inverse document frequencies, we use an already-compiled list of word frequencies based on the British National Corpus. The sample contains 67,962,112 total words, and 9,022 unique words.

In an earlier implementation, we only analyzed the downloaded web page to calculate TF-IDF scores, but discovered that some phishing web sites used JavaScript to dynamically load and modify a web page from a legitimate site. Thus, simply analyzing the downloaded source would not always work. To address this problem, we now analyze the text content in the Document Object Model (DOM), a standard tree-based representation of the web page which represents the current content and state of the web page. Although not a perfect solution (as discussed in section 5.1), it is more reliable than our earlier implementation.

Our extension currently has a simple user interface, displaying a red traffic light in a browser toolbar if a site is deemed a phishing scam. Note that this is a prototype user interface, and we discuss the need for developing a better user interface in Future Work.

# 4. EVALUATION

We conducted four experiments to assess the performance of CANTINA. In the first experiment, we examined the effectiveness of our adaptation of Robust Hyperlinks for detecting phishing sites. In the second experiment, we evaluated our heuristics, to determine the best way of weighting them to reduce false positives. In the third experiment, we evaluated the overall effectiveness of our algorithms and compared them to two existing toolbars. In the fourth experiment we evaluated our algorithm using URLs from actual user emails. We used two metrics to evaluate each approach:

- True positives (correctly labeling a phishing site as phishing, higher is better)
- False positives (incorrectly labeling a legitimate site as phishing, lower is better)

## 4.1 Experiment 1 – Evaluation of TF-IDF

In this experiment, we evaluated how effective our adaptation of Robust Hyperlinks was in detecting phishing sites. Here, we assessed four different conditions:

1. **Basic TF-IDF** – Calculate the lexical signature based on the top 5 terms, submit that to Google, and check if the domain name of the page in question matches any of the top 30 results
2. **Basic TF-IDF+domain** – Same as Basic TF-IDF, except that the domain name of the page in question is added to the lexical signature
3. **Basic TF-IDF+ZMP** – Same as Basic TF-IDF, except that zero search results means that the page in question is labeled as a phishing site (ZMP is "zero means phishing")
4. **Basic TF-IDF+domain+ZMP** – A combination of the two variants above. This combination turned out to have the best results, and is also called **Final-TF-IDF** in later sections.

We tested these variants by visiting 100 phishing URLs and 100 legitimate URLs with each variant, using an automated test bed. Our initial evaluation informed our later work. However, due to a problem with this evaluation, we decided to repeat it later when we conducted Experiment 3. We describe here the methodology and results for the later version of this experiment.

To test these four variants, we chose 100 phishing URLs from PhishTank.com [33] from November 17 to November 18, 2006. All phishing URLs were selected within 6 hours of being reported. We also chose 100 legitimate URLs from a list of 500 used in 3Sharp's study of anti-phishing toolbars [1]. All 200 URLs (100 phishing and 100 legitimate) were English language sites.

We used a test bed we previously developed [6] to gather our results. Our test bed takes a list of URLs, loads each URL into a web browser pre-installed with a given toolbar, and grabs a screen shot of the portion of the web browser where warning indicators are displayed. Since we know the possible states of each toolbar (e.g. showing a red traffic light or other kind of warning), we can compare the image we grabbed to a known image and determine how the toolbar evaluated each URL.

The results are shown in Figure 4. In comparing Basic-TF-IDF to Basic-TF-IDF+domain (conditions 1 and 2), we can see that adding the domain name to the content of the web page can significantly reduce the false positive rate (from 30% to 10%), but this comes at the cost of reduced accuracy (from 94% to 67%). This loss of accuracy is due to meaningless domain names that cause Google to return no results for some phishing sites. In comparing Basic-TF-IDF+domain to Basic-TF-IDF+domain+ZMP (conditions 2 and 4), we can see that the "zero results mean phishing" heuristic increases accuracy (from 67% to 97%) without impacting the false positive rate at all. Thus, the Final-TF-IDF (Basic-TF-IDF+domain+ZMP) seems to be the best.

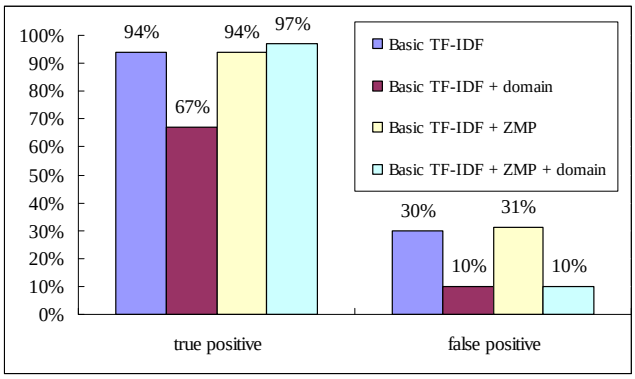


**Figure 4. Comparison of TF-IDF variants**

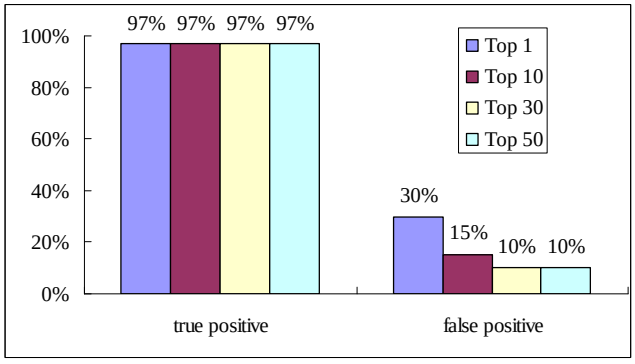


**Figure 5. Comparison of basic+domain+zmp algorithm with varying numbers of search results. The true positive rate remains the same throughout, while increasing the number of search results decreases the false positive rate.**

Using the same set of 100 phishing and 100 legitimate URLs, we also examined whether the number of Google results checked had any meaningful effects. We evaluated the Final-TF-IDF algorithm with only 1 result, 10 results, 30 results (our default), and 50 results. As Figure 5 shows, if we increase the number of Google search results examined, the false positive rate decreases while the true positive rate remains the same. Since we increase the number of domains we examine, the possibility that a legitimate site will match one of them increases. Furthermore, since phishing pages are rarely returned in search results, the true positive rate is not affected. We can also see that comparing against the top 30 results and the top 50 results yields no difference, which suggests if a match is found it should be within the first 30 results.

## 4.2 Experiment 2 – Evaluation of Heuristics

The first experiment suggested that CANTINA could detect phishing sites fairly well, but had a fairly high false positive rate. To reduce the false positive rate, we developed a suite of heuristics and ran another study to determine the best way of combining these heuristics to reduce false positives while not significantly impacting true positives. The heuristics, described in Section 3.3., are summarized in Table 1.

Determining the best weights for these heuristics is a typical classification problem. There are many algorithms for dealing with this kind of classification, including support vector machines and decision trees. For simplicity, we decided to use a simple forward linear model, which has the form:

$$S = f(\sum w_i * h_i) \qquad (1)$$

Where $h_i$ is the result of each heuristic, $w_i$ is the weight of each heuristic, and f is a simple threshold function. Recall in section 3.3 that if a heuristic deems a page as phish, it will return -1; and if a heuristic deems a page as legitimate, it will return +1. For our threshold, we chose a switch function, where:

$$f(x) = 1 \text{ if } x>0, f(x) = -1 \text{ if } x<=0 \qquad (2)$$

Thus, a positive value for the sum of the weighted heuristics means that it is labeled as legitimate, while a negative value or zero means that it is labeled as a phishing site.

**Table 1. Heuristics used to reduce false positives. Note that we added TF-IDF-Final as a "heuristic" to determine the proper weight to assign to it.**

| Heuristic | Suspected Phishing? |
|---|---|
| Age of Domain | <= 12 months |
| Known Images | Page contains any known logos and not on a domain owned by logo owner |
| Suspicious URL | URL contains @ or - |
| Suspicious Links | Link on page contains @ or - |
| IP Address | URL contains IP address |
| Dots in URL | >= 5 dots in URL |
| Forms | Page contains a text entry field |
| TF-IDF-Final | TF-IDF-Final suspects phishing |

The next step is to calculate the weight for each heuristic. Basically, the more effective a heuristic, the higher the weight we should give to it. Ideally, a heuristic should have high accuracy in detecting phishing sites while also having a low false positive rate. To measure the effect of a heuristic, we calculate true positives minus false positives. This is a straightforward approach also used by another report on anti-phishing toolbars [1]. Given the effect $e_i$ of each heuristic, we calculate each weight proportionally, that is:

$$w_i = \frac{e_i}{\sum e_i} \qquad (3)$$

From equations 1, 2, and 3, we can calculate the score S for a URL. If S = 1, we deem the page as legitimate page, and if S = -1, we deem the page as phishing.

To determine the best weights for the heuristics, we used 100 phishing URLs chosen from PhishTank [33] from November 15-16, 2006. We also used the same 100 legitimate URLs as in Experiment 1. All of these 200 URLs are English language sites. We used the same test bed described in Experiment 1.

Table 2 shows the results of Experiment 2. We see that TF-IDF has the highest weight, followed by the Forms heuristic. The sum

of these two heuristics is nearly 0.5, suggesting that these two heuristics are the most effective for finding phishes. It is also worth noting that the Suspicious Links heuristic has more false positives than true positives, suggesting that it is not highly effective.

**Table 2. The results of Experiment 2 showing what weights should be used for the various heuristics. Note that link check generates a negative effect, which means it is nearly useless, so we assign 0 to its effect.**

| Heuristic | True Positive | False Positive | Effect | Weight |
|---|---|---|---|---|
| Age of Domain | 87% | 30% | 57.0 | 0.18 |
| Known Images | 37% | 0% | 37.0 | 0.12 |
| Suspicious URL | 6% | 3% | 3.0 | 0.01 |
| Suspicious Links | 8% | 25% | 0.0 | 0.00 |
| IP Address | 22% | 0% | 22.0 | 0.07 |
| Dots in URL | 45% | 3% | 42.0 | 0.13 |
| Forms | 94% | 27% | 67.0 | 0.21 |
| TF-IDF-Final | 99% | 10% | 89.0 | 0.28 |

## 4.3 Experiment 3 – Evaluation of CANTINA

The third experiment is designed to evaluate the effectiveness of Final-TF-IDF, Final-TD-IDF+heuristics, SpoofGuard, and Netcraft. We wanted to see what effect the heuristics have in impacting true positives and false positives, as well as comparing our overall performance with two popular anti-phishing toolbars, SpoofGuard and Netcraft. We selected SpoofGuard and Netcraft because in our previous study of 10 tools [4], we found that SpoofGuard had the highest true positive rate and that Netcraft was one of the best toolbars overall (when both true positives and false positives were considered). In addition, these two toolbars take somewhat different approaches: SpoofGuard relies entirely on heuristics and Netcraft uses a combination of heuristics and an extensive blacklist. We tested Netcraft 1.7.0, which was the latest version available. We tested Spoofguard with its default setting. In our previous study [4], we found that SpoofGuard will always label a site as legitimate if it is already in Internet Explorer's history, so we deleted the IE history before we tested SpoofGuard.

To test the effectiveness of the weights we calculated in Experiment 2, we used different testing data than we used in Experiment 2. We manually chose 100 phishing URLs with unique domains from PhishTank during November 17-18, 2006 (the same set of URLs used in Experiment 1). We also compiled 100 legitimate URLs using the following strategy:

- Select the login pages of 35 sites that are often attacked by phishers, such as www.citibank.com and www.paypal.com. These pages are very similar to phishing pages so we can see whether our approach can distinguish them.
- Select the 35 top pages from Alexa Web Search [36]. Since these pages are the most popular pages in everyday life, it's very important to label these pages correctly.
- Select 30 random pages from http://random.yahoo.com/fast/ryl, and manually verify that they are legitimate. Since users will visit any kind of site with the toolbar, it's worth testing how well our approach deals with random legitimate URLs.

The URLs we gathered allow us to perform a comparative evaluation; they are not intended to be representative and thus cannot be used to compute absolute accuracy measures.

All 200 URLs are English language sites. We used the same test bed as in Experiments 1 and 2. Figure 6 shows our results. All 4 toolbars tested had high true positive rates, and our Final-TF-IDF and Netcraft did the best with 97% true positives. SpoofGuard had a fairly high false positive rate of 48%, and Final-TF-IDF had a false positive of 6%. The Final-TF-IDF+heuristics and Netcraft had 1% and 0% false positive rates respectively.

The difference in true positives between Final-TF-IDF and SpoofGuard are not statistically significant (p=0.0740), nor is the difference between Final-TF-IDF+Heuristics and SpoofGuard (p=0.6374). However, the difference in true positives between Final-TF-IDF and Final-TF-IDF+Heuristics is statistically significant (p=0.0266).

For false positives, the difference between SpoofGuard and all the others are statistically significant (p<0.0001). The difference between Final-TF-IDF and NetCraft is also statistically significant (p=0.0129). However, there are no other significant differences. Final-TF-IDF and Final-TF-IDF+Heuristics had p=0.3161, and Final-TF-IDF+Heuristics and NetCraft had p=0.0544.

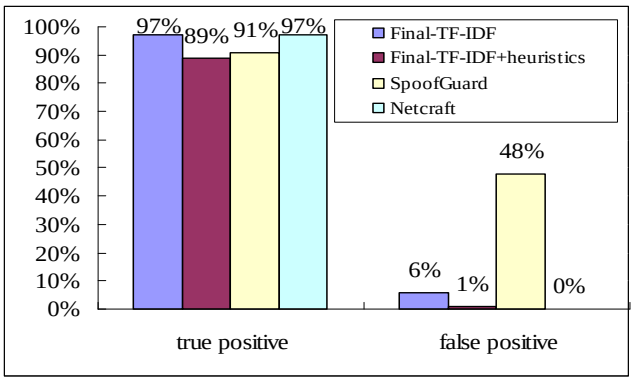


**Figure 6. Comparison of Final-TF-IDF, Final-TF-IDF+heuristic, SpoofGuard, and Netcraft**

The results suggest that the pure heuristic based approach has a tradeoff between true positives and false positives. Our Final-TF-IDF did very well in catching phish, but has fairly high false positives. By combining it with some simple heuristics we reduced false positives from 6% to 1% (although this result is not statistically significant), but at the same time we reduced the true positives from 97% to 89%. In Section 5.1, we discuss some ideas for improving the true positive rate of Final-TF-IDF+heuristics while not affecting the false positive.

SpoofGuard, a heuristic-only toolbar, did very well in catching phish, but incorrectly labels nearly half of the legitimate sites as phishing sites. Netcraft did well in both true positive and false positives. In our previous studies Netcraft's performance varied from test to test, ranging from 75% to 96% of phishing sites detected [6]. This is likely due to the fact that Netcraft relies on an extensive blacklist and thus its performance depends heavily on how quickly phishing sites are added to that blacklist and how fresh the sites are when Netcraft is tested.

In summary, in this experiment, our Final-TF-IDF algorithm performed roughly the same as SpoofGuard in terms of true

positives and better in terms of false positives, and was comparable to Netcraft in true positives. Final-TF-IDF+heuristics outperformed SpoofGuard in false positives and has nearly the same true positives, but does not do as well as Netcraft. However, because our approach does not rely on blacklists, it will detect very fresh phishing sites that Netcraft may not be able to detect.

## 4.4 Experiment 4 – Evaluation of CANTINA Using URLs Gathered from Email

The fourth experiment was designed to evaluate CANTINA using URLs gathered from users' actual email inboxes rather than URLs from a phishing feed (which might not reflect the phishing attacks a person might normally encounter). From February 6 to 12, 2007, four members of our research group sent all of their email through a proxy that extracted every URL that appeared in these messages. We gathered 3038 unique URLs, of which only 2519 were active, from the 3385 email messages sent through the proxy. Every evening we used Final-TF-IDF, Final-TF-IDF+heuristics, and Netcraft to check each unique, active URL gathered during the previous day. We manually labeled the active URLs as "phishing," "spam," or "legitimate." We defined phishing URLs as pages that impersonate known brands and ask for personal data, the same definition used in the previous experiments. We defined spam URLs as those selling unsolicited products or services (mostly pharmaceuticals, sex-related products, counterfeit products, and credit). All other URLs were deemed legitimate. Of the 2519 active URLs, 19 of them were identified as phishing, 388 were identified as spam, 2100 were identified as legitimate, and 12 were unknown (mostly because they are Asian language sites that we could not read). We measured true positives and false positives, as before. We also measured *false spams*, where a spam site was incorrectly labeled as phishing (lower is better).

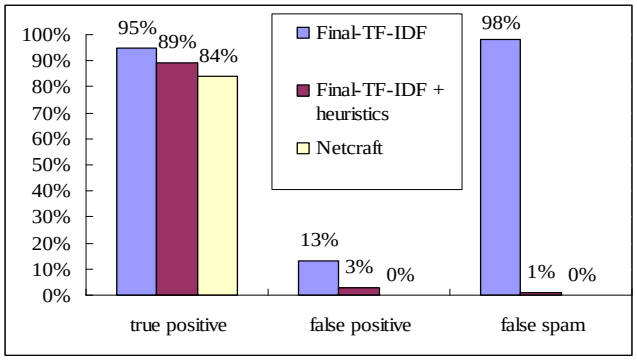


**Figure 7. Comparison of Final-TF-IDF, Final-TF-IDF+heuristics, and Netcraft in a more actual environment**

As shown in Figure 7, the true positives and false positives are similar to what we observed in Experiment 3, demonstrating that CANTINA performs well when used on actual users' email. All three approaches were able to catch 80% or more phishing URLs. Final-TF-IDF had the highest true positive rate, but all the differences were not statistically significant ($p>0.2$). Although the results may be less representative due to the small sample size (only 19 phishing URLs out of 2519 active URLs), they are consistent with the previous experiments. The Final-TF-IDF has 13% false positives, and Final-TF-IDF has 3% false positives. The difference is significant ($p<0.001$). Again Netcraft has no false positives, which is significantly better than the two TF-IDF approaches ($p<0.001$).

We tested spam URLs separately from legitimate URLs due to the fact that spam and phishing URLs have some similar characteristics. In addition, it may not matter if a phishing filter identifies a spam message as phishing if the phishing filter is being used in conjunction with a spam filter to filter out both phishing and spam messages. Because our Final-TF-IDF approach relies solely on whether Google will return the suspected URL, it marks most spam URLs—which tend not to be indexed in Google—as phishing. However, the heuristics we added to Final-TF-IDF+heuristics are all phishing specific. For example, the non-matching images are a typical feature of phishing, not spam. Therefore, Final-TF-IDF+heuristics is able to distinguish phishing URLs and spam, falsely identifying only 1% of spam URLs as phishing URLs. Netcraft did not identify any spam URLs as phishing. The difference between Final-TF-IDF and the other approaches is significant ($p<0.001$), and the difference between Final-TF-IDF+heuristics and Netcraft is not significant.

# 5. DISCUSSION

In this section we discuss some of the limitations of our approach and some possible ways to address them.

## 5.1 True Positives and False Positives

Final-TF-IDF had a 97% true positive rate in Experiment 3, incorrectly labeling three phishing sites as legitimate. These sites used a JavaScript technique we had not previously encountered to "hide" the real content of their page. Our content-based approach using the DOM works well for most web pages, but needs to be modified to address this problem. It is possible that CANTINA reads from the DOM too early, before the JavaScript has finished modifying the DOM. We plan to study the phishers' obfuscation techniques further to find a way to combat them.

Final-TF-IDF had a 6% false positive rate in Experiment 3, which incorrectly labeling six legitimate sites as phishing sites. We found that our parser will sometimes return the wrong text, for example, the text "bank | log-in" is parsed as "banklog-in" in the text structure of DOM. These mistakes can negatively impact the accuracy of our lexical signatures. In addition, we found that some legitimate sites are composed mostly of images with little text, preventing TF-IDF from returning a useful lexical signature.

Our heuristics in Final-TF-IDF+heuristics reduced the false positive rate from 6% to 1%. The one remaining failure came from a legitimate page that had many phishing characteristics, such as containing forms requiring personal data, a relatively new domain name, and many dots in the URL. However, Final-TF-IDF+heuristics reduced the true positive rate from 97% to 89%, which means there are eight phishing sites that Final-TF-IDF labeled as phish but the weighted heuristics labeled as legitimate. These phishing sites were all on domains that have been registered more than 12 months, and were attacking sites whose logos we did not store. We can easily fix this problem by adding these logos to our logo heuristic; however, there will always be more logos and legitimate companies than we can easily include. One possible solution here is to develop an algorithm that does an image search and compares what web pages that logo is seen on to the current page.

## 5.2 Limitations of CANTINA and TF-IDF

Our current implementation of CANTINA has several limitations. The first is that it does not include a dictionary for languages

other than English. We also discovered in informal evaluations that TF-IDF does not work well with East Asian languages. These languages are harder to parse, since a word can be composed of several different characters, and since there is not the equivalent of a space between characters to demarcate where one word ends and another begins. However, we believe that this is a parsing problem rather than a fundamental issue with TF-IDF itself.

CANTINA suffers from performance problems due to the time lag involved in querying Google. There are several ways of addressing this problem. The first is to only send queries if the page passes simple heuristics that we have already implemented, e.g., having a text entry input field. The second is to do the query in the background, but then stop people if they try to submit information. This approach would give CANTINA several extra seconds to determine if a web page is legitimate while having a minimal impact on the user experience. A third approach is to cache URLs that have already been checked. A fourth approach is to implement these algorithms as part of a server-based email filter rather than a browser toolbar. CANTINA might also be useful in applications where real-time identification of phishing sites is not required. For example, it could be used to identify phishing URLs in web server referer logs, or it could be used by web hosting providers to check for phishing sites on their servers.

A third issue is the potential for attackers to circumvent CANTINA. Phishing is an arms race, with criminals continually devising new ways of tricking people. We have identified three direct approaches by which CANTINA might be attacked.

The first approach is to attack the TF-IDF algorithm by changing the web page. One technique would be to use images instead of words. A second technique would be to add “invisible” text, text that is tiny or matches the background color of the page. While our current implementation does not address these problems, one could imagine additional checks to see if these measures are being used (for example, simple computer vision or comparing the color of text to the background color). A third technique would be to change enough words on the phishing page to confuse TF-IDF. However, we argue that this is difficult to do in practice. Term frequency is fairly robust on a given page, and would require a fair amount of editing to change. Inverse document frequency is also something criminals have no control over. As a result, a scammer would have to make significant enough changes to their copy of a web page, small enough to convince potential victims that the page is still the right one but large enough to lead TF-IDF to the fake domain name rather than the original. We argue that this is an unlikely proposition given the relatively short lifespan of phishing attacks and given that phishing web pages often have a low Google PageRank due to lack of links pointing to the scam.

This leads to the second approach for subverting CANTINA, which is to game Google’s PageRank algorithm, as has been done by search engine optimization companies and by pranksters setting “Google bombs." If CANTINA becomes widely adopted, scammers may try to subvert it by trying to break into sites that already have high PageRank or by waiting until a phishing site is indexed by Google before using the URL in phishing emails.

The third approach to subverting CANTINA is to attempt a distributed denial of service attack on Google. However, such an attack could be countered easily if CANTINA used multiple search engines instead of relying on a single one.

# 6. CONCLUSIONS AND FUTURE WORK

In this paper, we presented the design and evaluation of CANTINA, a novel content-based approach for detecting phishing web sites. CANTINA takes Robust Hyperlinks, an idea for overcoming page not found problems using the well-known Term Frequency / Inverse Document Frequency (TF-IDF) algorithm, and applies it to anti-phishing. We described our implementation of CANTINA, and discussed some simple heuristics that can be applied to reduce false positives. We also presented an evaluation of CANTINA, showing that the pure TF-IDF approach can catch about 97% phishing sites with about 6% false positives, and after combining some simple heuristics we are able to catch about 90% of phishing sites with only 1% false positives.

In future work, we plan on refining CANTINA in preparation for wider-scale deployment and evaluation. We also plan on developing and evaluating better user interfaces. As previous research has shown [37], even if an anti-phishing toolbar is highly accurate, users might still fall victim to fraud if users do not understand what the toolbar is trying to communicate.

# 7. ACKNOWLEDGMENTS

Thanks to the members of the Supporting Trust Decisions project for their feedback, to Tom Phelps for providing source code for Robust Hyperlinks, and to Ian Fette for setting up the email proxy used in Experiment 4. This work was supported in part by National Science Foundation grant CCF-0524189 (“Supporting Trust Decisions”) and by Army Research Office grant DAAD19-02-1-0389 (“Perpetually Available and Secure Information Systems”). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation or the U.S. government.

# 8. REFERENCES

[1] 3Sharp, 3Sharp Study finds Internet Explorer 7 Edges Out Netcraft As Most Accurate for Anti-Phishing Protection. 2006. http://www.3sharp.com/projects/antiphishing/

[2] Anti-Phishing Working Group, Phishing Activity Trends Report. 2006. http://www.antiphishing.org/reports/apwg_report_june_06.pdf

[3] Anti-Phishing Working Group (APWG). Visited: Nov 20, 2006. http://www.antiphishing.org/

[4] Chou, N., R. Ledesma, Y. Teraguchi, D. Boneh, and J.C. Mitchell. Client-Side Defense against Web-Based Identity Theft. In *Proceedings of The 11th Annual Network and Distributed System Security Symposium (NDSS '04)*. http://crypto.stanford.edu/SpoofGuard/webspoof.pdf

[5] Cloudmark Inc. Visited: Nov 20, 2006. http://www.cloudmark.com/desktop/download/

[6] Cranor, L., S. Egelman, J. Hong, and Y. Zhang. Phinding Phish: Evaluating Anti-Phishing Tools. In *Proceedings of The 14th Annual Network and Distributed System Security Symposium (NDSS '07)*. February 28- March 2, 2007.

[7] Dao, T., Term frequency-Inverse document frequency implementation in C#, The Code Project - C# Programming. Visited: Nov 20, 2006. http://www.codeproject.com/csharp/tfidf.asp

[8] Dhamija, R. and J.D. Tygar. The battle against phishing: Dynamic Security Skins. In *Proceedings of the First Symposium on Usable Privacy and Security (SOUPS 2005)*. pp. 77-88 2005.

[9] Dhamija, R., J.D. Tygar, and M. Hearst. Why Phishing Works. In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI2006)*, pp. 581-590, April 2006.

[10] Downs, J.S., M.B. Holbrook, and L.F. Cranor. Decision strategies and susceptibility to phishing. In *Proceedings of the Second Symposium on Usable Privacy and Security (SOUPS 2006)*. pp. 79-90 2006.

[11] eBay Inc., Spoof Email Tutorial. Visited: Nov 20, 2006. http://pages.ebay.com/education/spooftutorial/

[12] eBay Inc., Using eBay Toolbar's Account Guard. Visited: Nov 20, 2006. http://pages.ebay.com/help/confidence/account-guard.html

[13] Federal Trade Commission, An E-Card for You game. Visited: Nov 20, 2006. http://www.ftc.gov/bcp/conline/ecards/phishing/index.html

[14] Federal Trade Commission, Federal Trade Commission. Phishing Alerts. Visited: Nov 20, 2006. http://www.ftc.gov/bcp/conline/pubs/alerts/phishingalrt.htm

[15] Ferguson, A.J., Fostering E-Mail Security Awareness: The West Point Carronade, *EDUCASE Quarterly*, 2005. http://www.educause.edu/ir/library/pdf/eqm0517.pdf

[16] Fette, I., N. Sadeh, and A. Tomasic. Learning to Detect Phishing Emails. ISRI Technical Report. CMU-ISRI-06-112, 2006. http://reports-archive.adm.cs.cmu.edu/anon/isri2006/abstracts/06-112.html

[17] Gabber, E., P.B. Gibbons, Y. Matias, and A.J. Mayer. How to make personalized web browsing simple, secure, and anonymous. In *Proceedings of Financial Cryptography*. pp. 17–32 1997.

[18] GeoTrust Inc., TrustWatch Toolbar. Visited: Nov 20, 2006. http://toolbar.trustwatch.com/tour/v3ie/toolbar-v3ie-tour-overview.html

[19] Google Inc., Google Safe Browsing for Firefox. Visited: Nov 20, 2006. http://www.google.com/tools/firefox/safebrowsing/

[20] Halderman, J.A., B. Waters, and E.W. Felten. A Convenient Method for Securely Managing Passwords. In *Proceedings of 14th International World Wide Web Conference*, 2005.

[21] Herzberg, A. and A. Gbara, TrustBar: Protecting (even Naive) Web Users from Spoofing and Phishing Attacks. 2004, Cryptology ePrint Archive: Report 2004/155. http://www.cs.biu.ac.il/~herzbea/Papers/ecommerce/spoofing.htm

[22] Jackson, J.W., A.J. Ferguson, and M.J. Cobb. Building a University-wide Automated Information Assurance Awareness Exercise: The West Point Carronade. In *Proceedings of 35th ASEE/IEEE Frontiers in Education Conference* 2005. http://fie.engrng.pitt.edu/fie2005/papers/1694.pdf

[23] Jagatic, T., N. Johnson, M. Jakobsson, and F. Menczer, Social Phishing, 2006, http://www.indiana.edu/~phishing/social-network-experiment/phishing-preprint.pdf

[24] Keizer, G., Phishing Costs Nearly $1 Billion, *TechWeb Technology News*. Visited: Nov 20, 2006. http://www.techweb.com/wire/security/164902671

[25] Kumaraguru, P., Y.W. Rhee, A. Acquisti, L. Cranor, and J. Hong. Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System. In *Proceedings of CHI2007*.

[26] Mail Frontier, Phishing IQ. Visited: Nov 20, 2006. http://survey.mailfrontier.com/survey/quiztest.html

[27] McMillan, R., Gartner: Consumers to lose $2.8 billion to phishers in 2006, *NetworkWorld*, 2006. http://www.networkworld.com/news/2006/110906-gartner-consumers-to-lose-28b.html

[28] Microsoft, Consumer Awareness Page on Phishing. Visited: Nov 20, 2006. http://www.microsoft.com/athome/security/email/phishing.mspx

[29] Netcraft, Netcraft Anti-Phishing Toolbar. Visited: Nov 20, 2006. http://toolbar.netcraft.com/

[30] New York State Office of Cyber Security & Critical Infrastructure Coordination. 2005. Gone Phishing… A Briefing on the Anti-Phishing Exercise Initiative for New York State Government. Aggregate Exercise Results for public release.

[31] Panahy, A., Google Parser, The Code Project - C# Programming. Visited: Nov 20, 2006. http://www.codeproject.com/csharp/googleparser.asp

[32] Phelps, T.A. and R. Wilensky, Robust Hyperlinks and Locations, *D-Lib Magazine*, vol. 6(7/8), 2000. http://www.dlib.org/dlib/july00/wilensky/07wilensky.html

[33] PhishTank. Visited: Nov 20, 2006. http://www.phishtank.com/

[34] PhishTank, Statistics about Phishing Activity and PhishTank Usage. Visited: Nov 20, 2006. http://www.phishtank.com/stats/2006/10/

[35] Salton, G. and M.J. McGill, *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill, 1986.

[36] Stanford Applied Crypto Group, PwdHash. Visited: Nov 20, 2006. http://crypto.stanford.edu/PwdHash

[37] Wu, M., R. Miller, and S. Garfinkel. Do Security Toolbars Actually Prevent Phishing Attacks? In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI2006), CHI Letters 8(1)*. Quebec, Canada: ACM Press. pp. 601-610, April 2006.

[38] Wu, M., R.C. Miller, and G. Little. Web Wallet: Preventing Phishing Attacks by Revealing User Intentions. In *Proceedings of The Second Symposium on Usable Privacy and Security (SOUPS 2006)*. pp. 102-113 2006.

[39] Ye, Z., S. Smith, and D. Anthony, Trusted paths for browsers. *ACM Transactions on Information and System Security* 2005. **8**(2): p. 153-186.

[40] Yee, K.-P. and K. Sitaker. Passpet: Convenient Password Management and Phishing Protection. In *Proceedings of The Second Symposium on Usable Privacy and Security (SOUPS 2006)*. pp. 32-43 2006.

[41] Zolnikov, P., Extending Explorer with Band Objects using.NET and Windows Forms, The Code Project - C# Programming. Visited: Nov 20, 2006. http://www.codeproject.com/csharp/dotnetbandobjects.asp