Phinding Phish: An Evaluation of Anti-Phishing Toolbars

Lorrie Cranor, Serge Egelman, Jason Hong, and Yue Zhang

November 13, 2006
CMU-CyLab-06-018

CyLab
Carnegie Mellon University
Pittsburgh, PA 15213

Phinding Phish:
An Evaluation of Anti-Phishing Toolbars

Lorrie Cranor, Serge Egelman, Jason Hong, and Yue Zhang
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213

{lorrie, egelman, jasonh}@cs.cmu.edu, zysxqn@andrew.cmu.edu

ABSTRACT
There are currently dozens of freely available tools to help combat phishing and other web-based scams. Many of these tools come in the form of web browser extensions that warn users when they are browsing a suspected phishing site. We used verified phishing URLs and legitimate URLs to test the effectiveness of 10 popular anti-phishing toolbars. Overall, we found that the anti-phishing toolbars that were examined in this study left a lot to be desired. SpoofGuard did a very good job at identifying fraudulent sites, but it also incorrectly identified a large fraction of legitimate sites as fraudulent. EarthLink, Google, Netcraft, Cloudmark, and Internet Explorer 7 identified most fraudulent sites correctly and had few, if any, false positives, but they still missed more than 15% of fraudulent sites. The TrustWatch, eBay, and Netscape 8 toolbars could correctly identify less than half the fraudulent sites, and McAfee SiteAdvisor did not correctly identify any fraudulent sites. Many of the toolbars we tested were vulnerable to some simple exploits as well. In this paper we describe the anti-phishing toolbar test bed we developed, summarize our findings, and offer observations about the usability and overall effectiveness of these toolbars. Finally, we suggest ways to improve anti-phishing toolbars.

1. INTRODUCTION
Over the past few years we have seen an increase in "semantic attacks" — computer security attacks that exploit human vulnerabilities rather than software vulnerabilities. Phishing is a type of semantic attack in which victims are sent emails that deceive them into providing account numbers, passwords, or other personal information to an attacker. Typical phishing emails falsely claim to be from a reputable business where victims might have an account. Victims are directed to a spoofed web site where they enter information such as credit card numbers or Social Security Numbers. There were 9,255 unique phishing sites reported in June of 2006 [1]. Billions of dollars are lost each year due to unsuspecting users entering personal information into fraudulent web sites. To respond to this threat, software vendors and companies with a vested interest in preventing phishing attacks have released a variety of "anti-phishing toolbars." For example, eBay offers a free toolbar that can positively identify the eBay site, and Google offers a free toolbar aimed at identifying any fraudulent site [9, 12]. As of September 2006, the free software download site *download.com*, listed 84 anti-phishing toolbars. Unfortunately, few empirical studies have been performed to examine the effectiveness of these toolbars. Thus, while many anti-phishing toolbars exist, it is not clear how well they actually work.

Previous studies have examined the extent to which users fall for phishing scams and whether users benefit from the information provided by anti-phishing toolbars. These studies have consistently shown that large fractions of users are likely to fall for phishing scams, and that many users ignore warnings provided by anti-phishing toolbars [7, 8, 13, 25]. However, little empirical data is available on the accuracy of toolbars or on the effectiveness of the various approaches to detecting phishing sites. Towards that end, this paper makes three research contributions. First, we describe the design and implementation of a test bed for automatically evaluating anti-phishing toolbars. Second, we describe the results of experiments that assess the accuracy of 10 popular anti-phishing toolbars that use differing techniques to identify phishing sites. Third, we describe techniques we developed for circumventing each of the toolbars tested. Our paper provides the anti-phishing community with insights into the effectiveness of several approaches to combating phishing.

2. OVERVIEW OF ANTI-PHISHING TOOLBARS

There are a variety of methods that can be used to identify a page as a phishing site, including white lists (lists of known safe sites), blacklists (lists of known fraudulent sites), various heuristics to see if a URL is similar to a well-known URL, and community ratings. The toolbars examined in this study employ differing combinations of these methods. We used publicly available information provided on the toolbar download web sites as well as our observations from using each toolbar to get a basic understanding of how each toolbar functions.

2.1 Cloudmark Anti-Fraud Toolbar

The Cloudmark Anti-Fraud Toolbar, shown in Figure 1, relies on user ratings [3]. When visiting a site, users have the option of reporting the site as good or bad. Accordingly, the toolbar will display a colored icon for each site visited. Green icons indicate that the site has been rated as legitimate, red icons indicate that the site has been determined to be fraudulent, and yellow icons indicate that not enough information is known about the site to make a determination. Additionally, the users themselves are rated according to their record of correctly identifying phishing sites. Each site's rating is computed by aggregating all ratings given for that site, with each user's rating of a site weighted according to that user's reputation. The Cloudmark Anti-Fraud Toolbar runs on Microsoft Windows 98/NT/2000/XP with Internet Explorer.

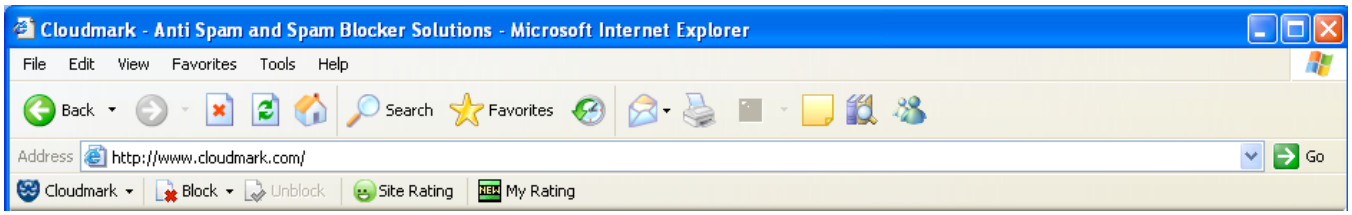

Figure 1: The Cloudmark Anti-Fraud Toolbar indicating a legitimate site.

2.2 EarthLink Toolbar

The EarthLink Toolbar, shown in Figure 2, appears to rely on a combination of heuristics, user ratings, and manual verification. Little information is presented on the EarthLink website; however, we used the toolbar and observed how it functions. The toolbar allows users to report suspected phishing sites to EarthLink. These sites are then verified and added to a blacklist. The toolbar also appears to examine domain registration information such as the owner, age, and country. The toolbar displays a thumb that changes color and position. A green thumbs up represents a verified legitimate site, whereas a gray thumbs up means that the site is not suspicious, but it has not been verified. The red thumbs down means that a site has been verified to be fraudulent, whereas the yellow thumbs down means that the site is "questionable." The EathLink Toolbar runs under Internet Explorer as well as Firefox [10].

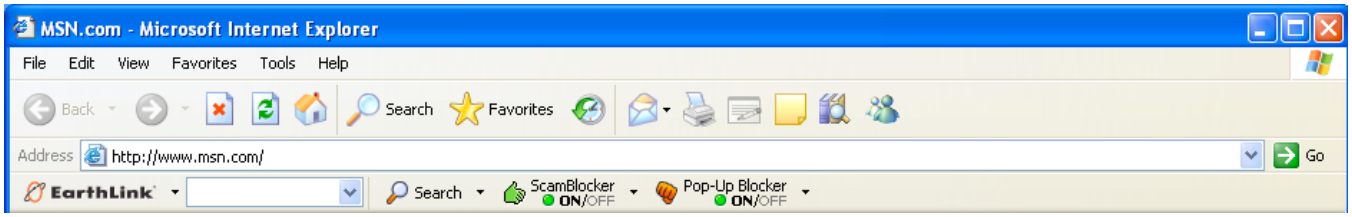

Figure 2: The EarthLink Toolbar indicating a legitimate site.

2.3 eBay Toolbar

The eBay Toolbar, shown in Figure 3, uses a combination of heuristics and blacklists [9]. The Account Guard indicator has three modes: green, red, and gray. The icon is displayed with a green background when the user visits a site known to be operated by eBay (or PayPal). The icon is displayed with a red background when the site is a known phishing site. The icon is displayed with a gray background when the site is not operated by eBay and not known to be a phishing site. The toolbar also gives users the ability to report phishing sites, which will then be

verified before being blacklisted. The eBay Toolbar runs under Microsoft Windows 98/ME/NT/2000/XP with Internet Explorer.

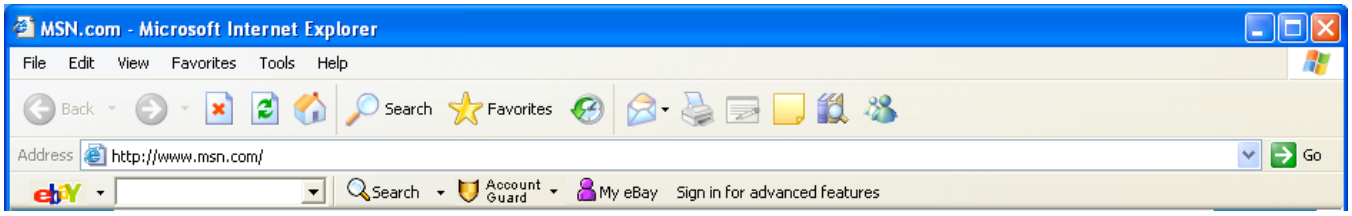

Figure 3: The eBay Toolbar at a site not owned by eBay that is not known to be a phishing site.

2.4 GeoTrust TrustWatch Toolbar

GeoTrust's TrustWatch Toolbar, shown in Figure 4, labels sites as green (verified as trusted), yellow (not verified), or red (verified as fraudulent). GeoTrust works with several third-party reputation services and certificate authorities to verify sites as trusted. GeoTrust's web site provides no information about how TrustWatch determines if a site is fraudulent; however, we suspect that the company compiles a blacklist that includes sites reported by users through a button provided on the toolbar. The toolbar also allows the user to store a custom image or bit of text that is constantly displayed so that he or she knows that the toolbar is not being spoofed. TrustWatch runs on Microsoft Windows 98/NT/2000/XP with Internet Explorer [11].

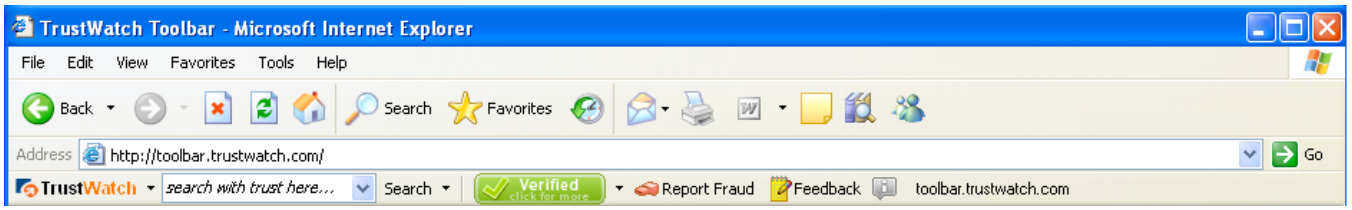

Figure 4: The GeoTrust TrustWatch Toolbar at a verified site.

2.5 Google Safe Browsing

The Google Toolbar, shown in Figure 5, includes a tool called "Google Safe Browsing" designed to identify fraudulent web sites. Google Safe Browsing was originally a Firefox extension, but it has since been integrated into the full toolbar and is expected to be built into Firefox 2.0. Google provides the source code for the Safe Browsing feature and says that it checks URLs against a blacklist. However, it is not known how URLs are added to the blacklist [15]. According to the download site, the tool combines "advanced algorithms with reports about misleading pages from a number of sources [12]." We suspect that this means that the tool uses blacklists as well as heuristics. The toolbar also includes a PageRank feature (which essentially denotes the popularity of a given site) that can be useful in identifying phishing sites, as most phishing sites have a very low PageRank. The toolbar displays a popup if it suspects the visited site to be fraudulent and provides users with a choice of leaving the site or ignoring the warning. The toolbar includes an "Enhanced Protection" option, which the software claims "will provide more up-to-date protection by sending the URLs of sites that you visit and limited information about the site content to Google for evaluation." This option is enabled by default. The toolbar runs on Microsoft Internet Explorer under Windows XP/2000 SP3+, or Firefox on most platforms.

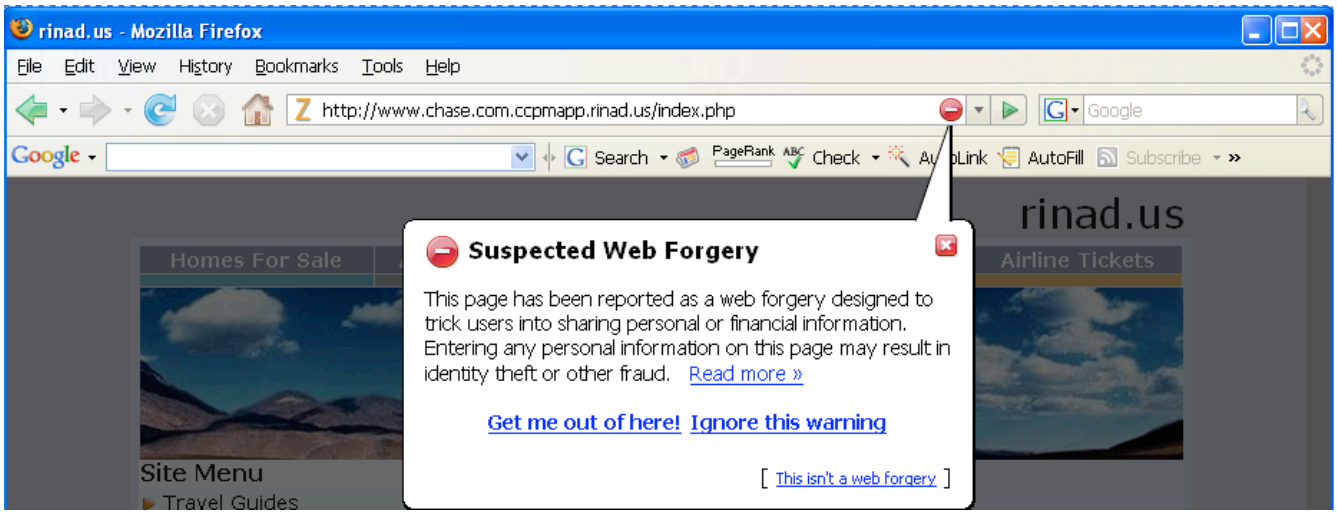

Figure 5: The Google Toolbar at a fraudulent site.

2.6 McAfee SiteAdvisor

McAfee SiteAdvisor, shown in Figure 6, runs on Microsoft Windows, Linux, and Mac OS X with the Firefox web browser, and Internet Explorer under Windows [16]. SiteAdvisor claims to detect not just phishing websites, but any sites that send spam, offer downloads containing spyware, or engage in other similar bad practices. The toolbar follows the traffic light model, similar to some of the other toolbars examined. A green indicator means that the site has been verified to be good. Red indicates that the site has "some serious issues to consider," while yellow indicates that the site may have some minor security issues or had some in the past. However, when SiteAdvisor encounters a site for which it doesn't have any information, it remains gray. The determination is made by a combination of automated heuristics and manual verification. For instance, the toolbar examines the age and country of the domain registration, the number of links to other known-good sites, third-party cookies, and user reviews.

Figure 6: McAfee SiteAdvisor at a verified good site.

2.7 Microsoft Phishing Filter in Windows Internet Explorer 7

The Microsoft Internet Explorer 7 web browser includes a built in phishing filter, shown in Figure 7 [17]. The toolbar largely relies on a blacklist hosted by Microsoft. However, it also uses some heuristics when it encounters a site that isn't in the blacklist. When a suspected phishing website is encountered, the user is redirected to a built in warning message and asked if they would like to continue visiting the site or close the window. Users also have the option of using this feature to report suspected phishing sites or to report that a site has incorrectly been added to the blacklist.

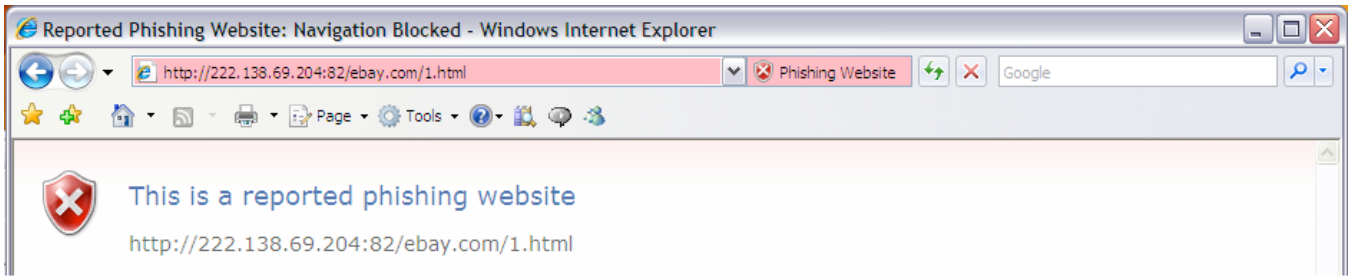

Figure 7: The Microsoft Phishing Filter in Windows Internet Explorer 7 at a fraudulent web site.

2.8 Netcraft Anti-Phishing Toolbar

The Netcraft Anti-Phishing Toolbar, shown in Figure 8, uses several methods to determine the legitimacy of a given web site. The Netcraft web site explains that the toolbar "traps suspicious URLs containing characters which have no common purpose other than to deceive," "enforces display of browser navigation controls (toolbar & address bar) in all windows, to defend against pop up windows which attempt to hide the navigational controls," and "clearly displays sites' hosting location, including country helping you to evaluate fraudulent URLs (e.g. the real Citibank.com or Barclays.co.uk sites are unlikely to be hosted in the former Soviet Union)" [18]. The Netcraft toolbar also uses a blacklist, which consists of fraudulent sites identified by Netcraft as well as sites submitted by users and verified by the company. When a user attempts to access a site that is on the blacklist, a pop-up warning recommends that the access be cancelled, but provides an override option. The toolbar also displays a risk rating between one and ten as well as the hosting location of the site (gleaned from the registration information for the IP address). Users can also use the toolbar to access a more detailed report on a web site. The Netcraft Anti-Phishing Toolbar runs on Firefox 1.0 on most platforms, and on Microsoft Internet Explorer under Windows 2000/XP.

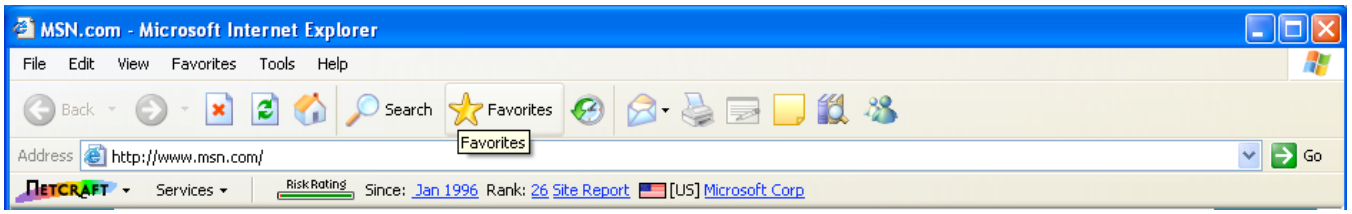

Figure 8: The Netcraft Anti-Phishing Toolbar at a legitimate web site.

2.9 Netscape Browser 8.1

The Netscape Navigator 8.1 web browser includes a built in phishing filter, shown in Figure 9 [19]. From our testing, as well as third party reviews, it appears that this functionality relies solely on a blacklist, which is maintained by AOL and updated frequently [4]. When a suspected phishing site is encountered, the user is redirected to a built-in warning page. Users are shown the original URL and are asked whether or not they would like to proceed. The Netscape Browser runs under Microsoft Windows, Linux, and Mac OS X.

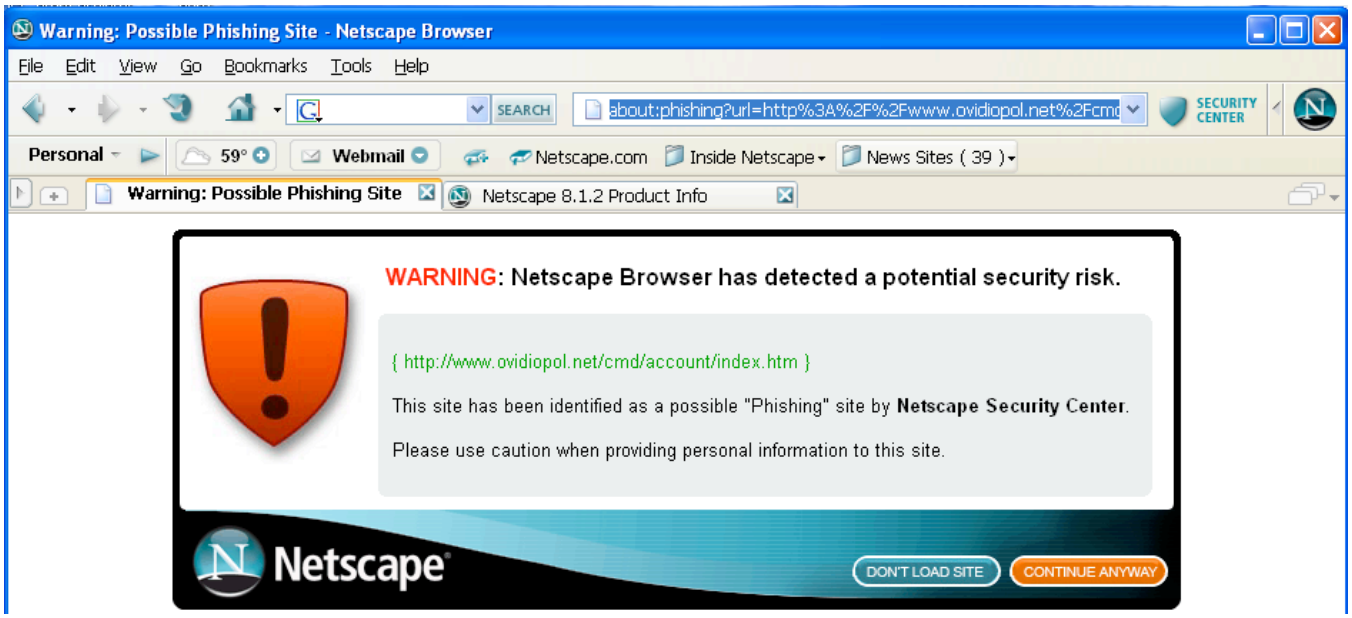

Figure 9: Netscape 8.1 web browser at a fraudulent web site.

2.10 SpoofGuard

SpoofGuard, shown in Figure 10, is an anti-phishing toolbar developed at Stanford University. Unlike the other toolbars described here, SpoofGuard does not use white lists or blacklists. Instead, the toolbar employs a series of heuristics to identify phishing pages. The toolbar first checks the current domain name and compares it with sites that have been recently visited by the user to catch fraudulent web sites that have a similar-looking domain name. Next, the full URL is analyzed to detect obfuscation as well as non-standard port numbers. Afterwards, the contents of the page are analyzed, making note of any password fields, embedded links, and images. Following this, SpoofGuard analyzes links in the web page itself using the heuristics described above. Finally, it examines images on the web page by hashing them to see if it has found similar images on other sites the user has visited. If two identical images are spotted on different web sites, there is a chance that a fraudulent site has copied the images from the legitimate site.

SpoofGuard computes a score for each web page in the form of a weighted sum of the results of each set of heuristics. Users can change the weights for each set of heuristics in an options menu. If the score surpasses a certain threshold, the toolbar displays a red icon, warning users that the site is a positively identified phishing site. If some of the heuristics are triggered but not enough to exceed the threshold, the icon turns yellow to indicate that it cannot make a determination about the site. If none of the heuristics are triggered, the icon turns green to indicate a safe site. SpoofGuard runs on Microsoft Windows 98/NT/2000XP with Internet Explorer [2].

Figure 10: SpoofGuard at a legitimate web site.

3. ANTI-PHISHING TOOLBAR EVALUATION

Our anti-phishing toolbar evaluation included three experiments designed to investigate the accuracy of anti-phishing toolbars. Our first experiment involved manually evaluating five of the toolbars described above. This gave us a feel for the behavior and effectiveness of the various toolbars, but proved labor intensive and posed

significant logistical difficulties. As a result, we developed an automated testing system and used it to conduct our second experiment. In our second experiment we were able to test how each toolbar responded to a set of URLs multiple times over a 24 hour period, allowing us to observe the effect of black list updates and of phishing sites being taken down. In our third experiment we expanded the set of toolbars tested to all 10 of the toolbars listed above and we changed our method for identifying phishing URLs with which to test the toolbars.

3.1 Experiment #1 – Manual Evaluation of Anti-Phishing Toolbars

We designed our first experiment to test the ability of each toolbar to accurately detect "fresh" phishing web sites identified within 36 hours of the experiment. The testing environment consisted of one laptop running Mac OS X 10.4 with Firefox installed for the Netcraft Toolbar, and four laptops running Microsoft Windows XP, each running a different toolbar and (depending on the requirements of the toolbars) Microsoft Internet Explorer or Firefox. By using five laptops we were able to simultaneously test five toolbars.

Once phishing web sites are identified, they are often taken down quickly. According to the Anti-Phishing Working Group (APWG), the average time that a phishing site stays online is 4.8 days [1], though our experience suggests that many are taken down within hours. Therefore, it was critical to find a source of freshly reported phishing sites to test in our experiment. After experimenting with feeds that consisted of mostly phishing sites that had already been taken down, we obtained access to a feed of phishing URLs provided by an email filtering vendor. Each of the anti-phishing toolbars described in Section 2 were tested with 50 confirmed phishing URLs identified within the previous 36 hours. We manually navigated each web browser to each URL and recorded the information provided by the anti-phishing toolbar. Because we generally did not receive more than 20 new phishing URLs each day, we conducted the study during three separate sessions between April 11th and April 26th, 2006.

As shown in Table 1, the Netcraft Toolbar was the most effective, identifying 48 out of 50 phishing sites (96%). The next best toolbar, TrustWatch, was only able to identify 34 (68%). SpoofGuard was similar in performance, positively identifying 31 phishing sites (62%). Neither the Google toolbar nor the Cloudmark toolbar were able to positively identify any of the phishing sites. However, the Google toolbar caused Firefox to crash when visiting almost every phishing site. While this could serve as a reasonable deterrent to naïve users trying to navigate to such sites, upon further examination, we concluded that we were experiencing compatibility problems and decided to eliminate the Google toolbar from our study. The Cloudmark icon remained yellow for every page visited, indicating that it did not have enough data to make a determination about any of the sites. Since Cloudmark relies on user ratings, it appears that at the time of this first study, they were unable to collect a sufficient number of ratings quickly enough to be effective against phishing.

Table 1: Number of phishing sites out of 50 correctly and incorrectly identified by anti-phishing toolbars.

	Sites correctly identified as phishing	Sites for which toolbar made no determination	Sites incorrectly identified as safe
Cloudmark	0 (0%)	50 (100%)	0 (0%)
Netcraft	48 (96%)	2 (4%)	0 (0%)
TrustWatch	34 (68%)	16 (32%)	0 (0%)
SpoofGuard	31 (62%)	12 (24%)	7 (14%)

SpoofGuard incorrectly marked seven phishing sites as legitimate. SpoofGuard made this determination using its image hashing heuristic, because it had not previously seen any of the images on these sites. It stands to reason that when visiting a phished site after SpoofGuard has already seen its legitimate counterpart, this problem should not arise. We also observed that when a phishing site is visited prior to the corresponding legitimate site, SpoofGuard mistakenly identifies the legitimate site as a phishing site.

3.2 Design and Implementation of an Automated Anti-Phishing Test Bed

Our first experiment was very labor intensive, making this method infeasible for evaluating larger data sets across longer periods of time. Therefore, we developed an automated test bed for evaluating the effectiveness of anti-phishing toolbars. This test bed will facilitate the evaluation of new approaches to phish detection and the examination of long-term phishing trends, giving the anti-phishing community a clearer picture of how much progress we are making towards automatically detecting phishing sites. Figure 11 shows the high-level system architecture. Our system includes a Task Manager and a set of Workers, each of which is responsible for evaluating a single toolbar.

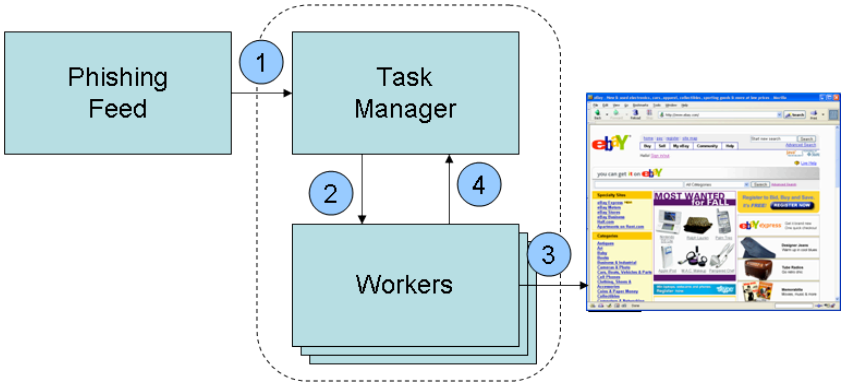

Figure 11: High-level system architecture for our anti-phishing evaluation test bed. The Task Manager (1) gets an updated list of URLs from a phishing feed, and then (2) sends that URL to a set of Workers. Each worker (3) retrieves a web page and checks whether the web page was labeled as a phishing scam or not, and (4) sends the result back to the Task Manager, which aggregates all of the results. The Task Manager and Workers are grouped together because they can be run on the same machine or on separate machines.

Step 1 – Get Potential Phishing Sites from a Phishing Feed

First, the Task Manager downloads a set of URLs from a phishing feed, providing a set of potential phishing sites to test against. For our second experiment we used a phishing feed provided by the Anti-Phishing Working Group (APWG), which includes reported phishing emails forwarded by individuals and corporations. To obtain a list of potential phishing sites, we first extract all URLs from the reported phishing emails. Because phishing emails typically contain links to both legitimate and fraudulent web sites, we use two heuristics to filter out sites unlikely to be phishing scams. We then manually confirm that the remaining URLs are scams.

The first heuristic removes URLs that point to images (as opposed to HTML pages) and sites that do not have text entry fields that request sensitive data such as passwords and credit card numbers. The rationale here is that if there is no way for people to give up personal information, then the site is probably not a phishing scam.

The second heuristic uses a variant of Robust Hyperlinks [20] to remove sites that are likely legitimate. The basic idea is to create a lexical signature by choosing words that are common on the current web page (term frequency), but rare on other web pages (inverse document frequency). The words comprising the lexical signature are then fed into a search engine, which in our case is Google. If the domain name of the current web page matches the domain name of the top search results (presently the top 30 search results), then we consider it to be a legitimate web site. The rationale here is that phishing operators often copy web pages directly from the sites they are impersonating, modifying only the minimal amount of HTML necessary. Consequently, most of the web page's original content is left intact, meaning that the lexical signature generated will often lead us to the original and legitimate web page rather than the phishing scam. It is also worth noting that this approach is robust, in that it uses both term frequency and inverse document frequency. To subvert this technique, a scammer would have to make significant 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 our heuristic 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.

We developed our heuristics for the purpose of removing URLs unlikely to be phishing scams from the list of URLs contained in reported phishing messages. However, as these heuristics appear to work quite well in practice, they might be incorporated into a future anti-phishing toolbar.

Once the Task Manager has extracted a list of URLs from the APWG feed, it removes duplicate URLs as well as URLs from domains we have recently checked. A human inspects the web sites referenced by each of the remaining URLs to verify that they are phishing sites. For our experiments, we labeled a site as a phishing scam only if it impersonates a known brand. This means, for example, that we did not include e-commerce sites that might rip you off, or web sites for fictitious companies that conduct identity theft by tricking prospective employees into submitting their resumes.

Step 2 – Send URL of Potential Phishing Site to Workers
In the second step, the Task Manager sends each URL to a set of Workers, each of which is running a separate toolbar. The Workers can be run on the same machine as the Task Manager or on separate machines. However, running workers on the same machine can be problematic when testing multiple toolbars that work with the same web browser, as the tests should be run with only one toolbar installed in the web browser at a time. We conducted some of our tests using multiple physical machines and some of our tests using virtual machines on the same physical machine.

Step 3 – Worker Evaluates Potential Phishing Site
In the third step, each Worker downloads the specified web page, examines whether its toolbar has labeled the web page as phishing or not, and returns that value back to the Task Manager. Workers retrieve web pages using the Tor anonymity network [24], thus making it harder for phishing operators to observe that we are evaluating their sites. We have developed a simple image-based approach for Workers to check a given toolbar. Each toolbar has several known states (e.g., a red icon if it has detected a phishing site and a green icon if it has not), and each toolbar can be set up to be in a known location in the web browser. Thus, we simply capture screenshots of the toolbars beforehand and compare relevant portions of those images to screenshots of the current state of the toolbar. The primary advantage of this image-based approach is that it works for all toolbars regardless of the programming language in which the toolbar was written, whether or not the toolbar provides an explicit API, and what web browser is being used.

Step 4 – Task Manager Aggregates Results
In the fourth step, the Task Manager aggregates all of the results from the Workers and tallies overall statistics, including true positives, true negatives, false positives, false negatives, and sites that no longer exist.

Our automated anti-phishing test bed is currently implemented in C# and is comprised of 2000 lines of code. Our implementation also makes use of three freely available .NET components, which include Compare Images [23], which checks if two images are identical; and a TF-IDF component [6].

3.3 Experiment #2 – Automated Evaluation of Five Anti-Phishing Toolbars

Using our test bed, we evaluated the same five toolbars as in our preliminary study. Using Microsoft's Internet Explorer v6.0, we tested Netcraft v1.6.2, TrustWatch v3.0.4.0.1.2, SpoofGuard, and Cloudmark v1.0. Using FireFox v1.5.0.6, we tested Google Toolbar v2.1. We configured all toolbars with their default settings. The Task Manager was run on a 1.6GHZ Toshiba Portege M200 Notebook and the Workers were run on an IBM 1.6GHz ThinkPad T42 Notebook, with only one Worker loaded at any given time. We tested the five toolbars against a set of known phishing URLs as well as a set of known legitimate URLs.

We examined 1,084 messages from the APWG feed during September 4-6, 2006, and used the procedure described in section 3.2 to extract 100 confirmed, active phishing URLs spanning 100 unique domains.[1] We examined each message within one hour of its arrival. Each URL was tested against each toolbar immediately upon extraction as well as 1, 2, 12, and 24 hours later. By testing each URL multiple times we were able to observe black list updates as well as how long it took for phishing sites to be taken down.

We examined 99 additional messages from the APWG feed on September 6, 2006 and used a procedure similar to the one described in section 3.2 to extract 60 unique legitimate URLs, which we manually confirmed. We then choose another 40 legitimate URLs, mostly from well-known bank websites such as www.citizensbank.com. Thus we created a list of 100 legitimate URLs, which we used to test the five toolbars for false positives.

During preliminary trials with our test bed, we observed that SpoofGuard treats all re-visited URLs as legitimate, even if it initially identified them as phishing. Thus, SpoofGuard failed to identify any URLs as phishing URLs on the second and later visits. We discovered that if we cleared the web browser history before testing each URL this problem goes away. Therefore we decided to clear the web browser history before each test during this experiment. However, as SpoofGuard's determination of whether a site is phishing or not is based only on heuristics and not on black lists, SpoofGuard's assessment does not change over time. Thus the apparent change in accuracy of SpoofGuard over time is due entirely to some of the phishing web sites being taken down. It appears that it is only necessary to test SpoofGuard on each URL once.

3.3.1 Experimental Results – Anti-Phishing Toolbars Catch Rate

When using an anti-phishing toolbar, the most important detail is that it accurately and conspicuously identifies phishing web sites when a user visits one. Since not all of the toolbars function the same way or provide the same types of indicators, we first had to create a quantitative measurement of accuracy. Some of the toolbars provide binary notifications (i.e. either that site is phishing or it is not), while some toolbars use a ternary system (i.e. a site can be phishing, not phishing, or unknown). Thus, we define catch rate as the number of phishing sites positively identified by a toolbar out of the total number of active phishing sites visited, with sites which had been taken down removed from the denominator. The rationale here is that it makes no difference whether a toolbar identifies a taken-down site as a phishing site, since a site that has been taken down does no harm to the user.

Figure 12 and Table 2 show the catch rate of each toolbar over time. As can be seen, SpoofGuard was able to identify the most phishing web sites, whereas Cloudmark was able to identify very few. The results for the first time period are over the complete set of 100 phishing URLs. After 1, 2, 12, and 24 hours 98, 88, 73, and 40 URLs remained active respectively.

[1] We manually examined 108 URLs extracted using this procedure and found that 8 of them were not actually phishing URLs. We eliminated these 8 false positives to arrive at our list of 100 confirmed phishing URLs.

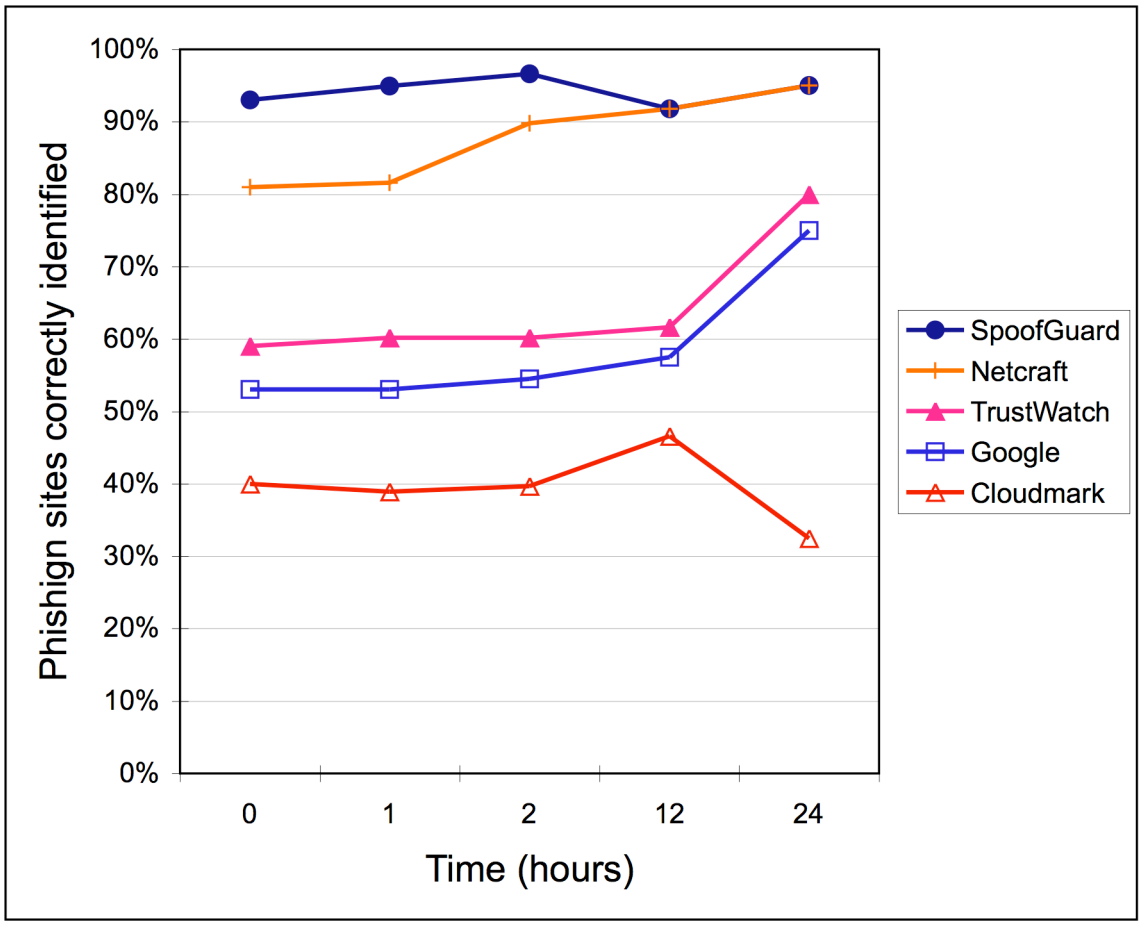

Figure 12: Experiment 2 catch rate of each toolbar over time.

The performance difference between SpoofGuard and the second most accurate toolbar, Netcraft, was statistically significant using a chi-square test at the initial time period ($p=0.025$) and the 1-hour time period, but became insignificant after two hours. Throughout the experiment, the differences between Google and TrustWatch were also insignificant. While Cloudmark performed worse after twelve hours, at that time it did not perform significantly different from Google or TrustWatch. The reason here is at 24 hours, 60 out of 100 phishing sites had been taken down, and more of the sites that it properly identified in the first 12 hours were removed than the sites that it did not properly identify. When examined using an ANOVA to determine differences across all of the time periods, we found that there was a statistically significant difference between TrustWatch, Google, and Cloudmark ($F=13.1$, $p=0.001$).

Table 2: Number of phishing sites correctly identified and legitimate sites falsely identified as phishing sites by anti-phishing toolbars in Experiment 2.

	Time since phishing site URLs were extracted					Legitimate sites	
	0 hours	**1 hour**	**2 hours**	**12 hours**	**24 hours**	**Falsely identified as phishing**	**Unsure**
Netcraft	81 (81%)	80 (81.6%)	79 (89.8%)	67 (91.8%)	38 (95%)	0 (0%)	0 (0%)
TrustWatch	59 (59%)	59 (60.2%)	53 (60.2%)	45 (61.6%)	32 (80%)	0 (0%)	8 (8%)
SpoofGuard	93 (93%)	93 (94.9%)	85 (96.6%)	67 (91.8%)	38 (95%)	37 (37%)	55 (55%)
Cloudmark	40 (40%)	38 (38.9%)	35 (39.7%)	34 (46.6%)	13 (32.5%)	0 (0%)	84 (84%)
Google	53 (53%)	52 (53%)	48 (54.5%)	42 (57.5%)	30 (75%)	0 (0%)	0 (0%)
Active URLs	100	98	88	73	40	100	100

As time progressed, Google, TrustWatch, and Netcraft improved in accuracy. However, we found that these were not statistically significant improvements. We also examined the phishing URLs that a toolbar initially missed and looked to see how many of them the toolbar was able to correctly identify before they were taken down. Of the 19 phishing URLs initially missed by Netcraft, it was able to correctly identify 11 of them before they were taken down. The correct identifications were all made within two hours of the initial test, probably due to Netcraft's large and frequently updated blacklist. The other toolbars were unable to correct as many of their initially incorrect identifications, and most of their corrections occurred after 12 hours. TrustWatch later correctly identified 7 of the 41 phishing URLs it originally missed, Google later correctly identified 13 of the 47 phishing URLs it originally missed, and Cloudmark later correctly identified only 3 of the 60 URLs it originally missed.

3.3.2 Experimental Results – False Positive Rates

While the catch rate for real phishing sites is the paramount concern, caution needs to be taken with regard to false positives. False positives pose a major usability problem for any security software. If a user is continually alerted to a pending a danger (in this case phishing) even when the user knows no such danger exists, he or she is most likely to disable or ignore the tool that is creating the alerts. Thus, while a phishing toolbar must identify phishing sites, it should also be careful to not identify legitimate web pages as phishing.

Each toolbar was tested against 100 legitimate URLs (as described at the beginning of section 3.3). As shown in Table 2, SpoofGuard erroneously denoted 37 of the URLs as phishing and reported that it was unsure about 55 of the URLs. None of the other toolbars had any false positives, although TrustWatch was unsure about a few of the legitimate sites and Cloudmark was unsure about most of them. SpoofGuard's poor performance on recognizing legitimate sites is likely related to the fact that it was the only toolbar to not use blacklists. Instead, SpoofGuard relies on heuristics. In this case, the heuristics are sometimes overly aggressive. Since the toolbar lets end-users customize the scoring for each heuristic, it is possible to tweak these settings so as to minimize false positives.

3.4 Experiment #3 – Automated Evaluation of Ten Anti-Phishing Toolbars

Having refined our methodology in the first two experiments, we undertook a third experiment to test 10 anti-phishing toolbars. This included the latest available versions of the five toolbars previously tested and five additional toolbars. We tested the built in phishing filters in Microsoft Internet Explorer v7.0.5700.6 and Netscape Navigator v8.1.2. We used Internet Explorer 6 to test the following toolbars: EarthLink v3.3.44.0, eBay v 2.3.2.0, McAfee v1.7.0.53 Netcraft v1.7.0, TrustWatch v3.0.4.0.1.2, SpoofGuard, and Cloudmark v1.0. Using FireFox v1.5.0.6, we tested Google Toolbar v2.1. We configured all toolbars with their default settings. The Task Manager was run on a 1.6GHZ Toshiba Portege M200 Notebook. The Workers were run on an IBM 1.6GHz ThinkPad T42 Notebook and a 1.7 GHz Compaq Presario v2000 Notebook.

For our third experiment we used the list of reported phishing sites published on phishtank.com as our source of phishing URLs. These are suspected phishing sites submitted by Internet users. During the period November 4-5, 2006 we visited phishtank.com every six hours and retrieved all new suspected phishing URLs that had been submitted within the previous six hours. We manually verified that they were phishing sites and that the sites were still online. We extracted 100 confirmed, active phishing URLs and examined each URL within six hours of its being posted on phishtank.com. Each URL was tested against each toolbar immediately upon extraction. In addition, each URL was tested against all toolbars except SpoofGuard 1, 2, 12, and 24 hours later. Based on our previous observations about SpoofGuard's behavior, we assume that SpoofGuard will not change its assessment of a given URL over time.

We used the same list of 60 legitimate URLs that we compiled from the APWG feed for our second experiment to test for false positives. In addition, we used a list of 500 legitimate URLs compiled by 3Sharp and published in a September 2006 report [22]. Of these 560 legitimate URLs, 50 were no longer available at the time of our testing, so we ended up testing only 510 legitimate URLs.

3.4.1 Experimental Results – Anti-Phishing Toolbars Catch Rate

We used the same procedure for calculating toolbar catch rate as was used in Experiment 2. Figure 13 shows the percentage of phishing sites correctly identified over time. After 1, 2, 12, and 24 hours 100, 98, 93, and 70 URLs remained active respectively. SpoofGuard, EarthLink, and Netcraft performed best at identifying phishing sites initially. SpoofGuard performed significantly better than Netcraft upon performing a chi-square test (p=0.01). However, we did not find a statistically significant difference between EarthLink and Netcraft, or SpoofGuard and EarthLink at the initial time period. Google, Cloudmark, and IE7 also did well. TrustWatch was able to only identify about half the phishing sites tested, while eBay identified 28% and Netscape identified 8%. McAfee failed to identify any phishing sites.

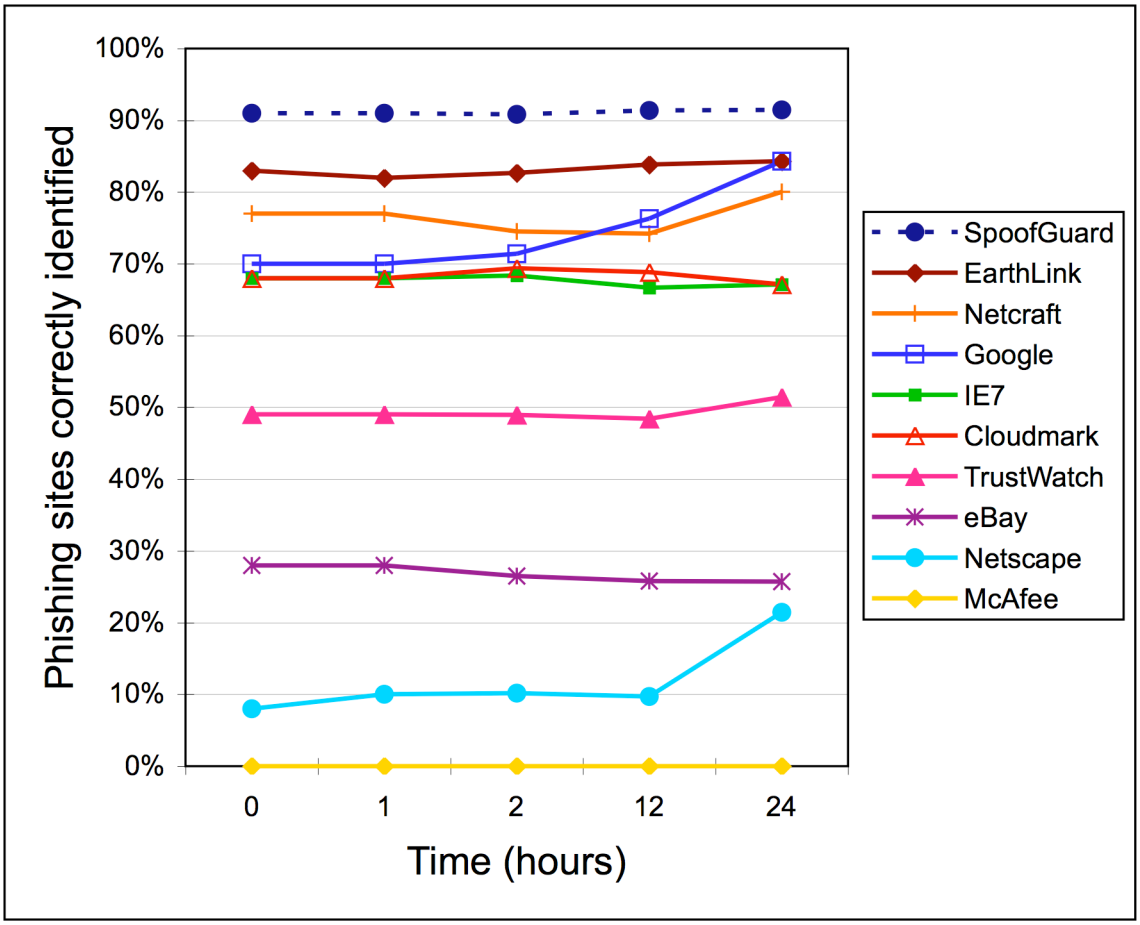

Figure 13: Experiment 3 catch rate of each toolbar over time. Note, SpoofGuard's catch rate is estimated after time 0.

As Table 3 shows, only five of the toolbars were able to correctly identify phishing sites in later tests that they incorrectly identified initially. The changes in accuracy observed for the other toolbars are due entirely to some of the phishing sites being taken down. Google and Netscape showed the largest performance improvements over time, suggesting that they are not updating their black lists as quickly as some of the other toolbars. Interestingly, we also saw that some of the toolbars initially made a correct identification of a phishing site and later reversed themselves. Netcraft made an incorrect reversal 6 times, Cloudmark made an incorrect reversal twice, and IE7 made an incorrect reversal once. After 24 hours, Google's performance edged out Netcraft and was as good as the second ranked EarthLink. Upon using an ANOVA to examine the differences over time between Netcraft, Google, and Cloudmark, we found there to be a significant difference (F=6.93; p=0.01).

Table 3: Number of phishing sites initially identified incorrectly that were later identified correctly by anti-phishing toolbars.

	Time since phishing site URLs were extracted			
	1 hour	**2 hours**	**12 hours**	**24 hours**
Cloudmark	0	1	0	0
EarthLink	0	0	0	0
eBay	0	0	0	0
IE7	0	1	0	0
Google	0	1	4	5
McAfee	0	0	0	0
Netcraft	0	1	0	4
Netscape	2	0	0	7
SpoofGuard	0	0	0	0
TrustWatch	0	0	0	0
Active URLs	100	98	93	70

3.4.2 Experimental Results – False Positive Rates

Each toolbar was tested against 510 legitimate URLs. SpoofGuard erroneously labeled 38% of these URLs as phishing. In addition it reported that it was unsure about an additional 45% of these URLs. The only other toolbars to falsely identify any URLs as phishing sites were EarthLink and Cloudmark, which each misidentified 1% of the legitimate sites. Cloudmark, EarthLink, McAfee, and TrustWatch were also unsure of a large number of sites.

Table 4: Number of phishing sites correctly identified and legitimate sites falsely identified as phishing sites by anti-phishing toolbars in Experiment 3.

	Time since phishing site URLs were extracted					Legitimate sites	
	0 hours	**1 hour**	**2 hours**	**12 hours**	**24 hours**	**Falsely identified as phishing**	**Unsure**
Cloudmark	68 (68%)	68 (68%)	68 (69%)	64 (67%)	47 (67%)	5 (1%)	489 (96%)
EarthLink	83 (83%)	83 (83%)	81 (82%)	78 (84%)	59 (84%)	4 (1%)	464 (91%)
eBay	28 (28%)	28 (28%)	26 (27%)	24 (26%)	18 (26%)	0 (0%)	0 (0%)
IE7	68 (68%)	68 (68%)	67 (68%)	62 (67%)	47 (67%)	0 (0%)	0 (0%)
Google	70 (70%)	70 (70%)	70 (71%)	71 (76%)	59 (84%)	0 (0%)	0 (0%)
McAfee	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	121 (24%)
Netcraft	77 (77%)	77 (77%)	73 (74%)	69 (74%)	56 (80%)	0 (0%)	0 (0%)
Netscape	8 (8%)	10 (10%)	10 (10%)	9 (10%)	15 (21%)	0 (0%)	0 (0%)
SpoofGuard	91 (91%)	91 (91%)*	89 (91%)*	85 (91%)*	64 (91%)*	195 (38%)	231 (45%)
TrustWatch	49 (49%)	49 (49%)	48 (49%)	45 (48%)	36 (51%)	0 (0%)	246 (48%)
Active URLs	100	100	98	93	70	510	510

4. Toolbar Exploits

As we tested the five toolbars in this study, we got a feel for how they identified fraudulent sites and developed some ideas for exploiting them. We describe two of these potential exploits here, as well as ways the vulnerable toolbars could be modified to protect against them.

4.1.1 Content Distribution Networks

Nine of the ten toolbars we examined appear to rely on blacklists. Some of the toolbars take the entire URL into account, possible by using a hash or pattern matching. Other toolbars seem to make their decision based on

information that is known about the domain name or IP block where the site is hosted. Thus, by obfuscating the URL or forcing it to be routed through another domain name, an attacker might be able to convince the toolbar that a blacklisted site is really a non-blacklisted site. We were interested in whether visiting a web site through a content distribution network (CDN) would provide sufficient obfuscation.

We tested the CDN attack using the Coral Project CDN [5]. The Coral Project is a content distribution network that runs on top of PlanetLab, which dynamically routes HTTP traffic through any of 260 servers located around the world [21]. These servers primarily reside at academic or research institutions. To use Coral, one simply appends "*.nyud.net:8090*" to a given URL's domain name portion. Thus, all URLs passed through Coral appear to be on the .nyud.net domain. We re-examined some of the URLs that had been identified by a toolbar as fraudulent, this time passing them through Coral. Some of the toolbars failed to properly identify any of the URLs as fraudulent when they were passed through Coral. Figure 14 shows a comparison of the TrustWatch Toolbar visiting a phishing site with and without the use of Coral. As can be seen, the original phishing URL causes the toolbar to display a red warning. When the URL is run through Coral, TrustWatch says that the site is now unverified. This exploit works on Cloudmark, Google, McAfee, TrustWatch, Netcraft, and Netscape.

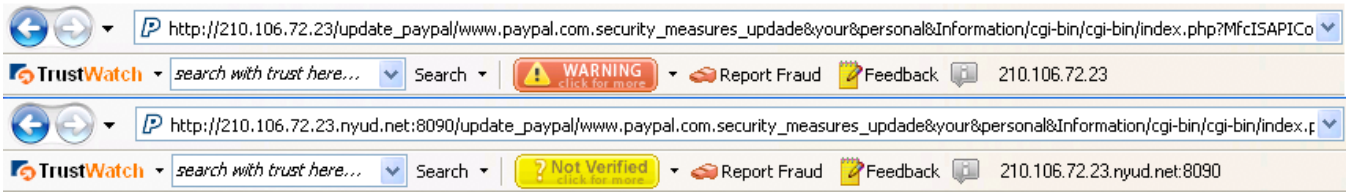

Figure 14: Demonstration of the CDN attack on the TrustWatch toolbar. The first screenshot shows TrustWatch correctly labeling a site as a phishing scam. The second screenshot shows how redirecting that same scam through the Coral CDN causes the same site to be labeled incorrectly.

As would be expected, this exploit did not work on the SpoofGuard toolbar. Since SpoofGuard does not use a blacklist, nothing can be gained by causing the URL to hash to a different value or appear to come from a different domain name. In fact, this particular exploit caused SpoofGuard to perform better. One of the heuristics that SpoofGuard checks is whether the destination web site is running on a non-standard port. For this particular exploit to work with Coral, the destination web site must be running on port 80 (the standard HTTP port). However, after running the URL through Coral, the URL will now point to port 8090 on a PlanetLab server, thus triggering this heuristic from within SpoofGuard.

While this vulnerability is worrisome, it should be fairly easy to address either by blacklisting CDNs or, preferably, by checking for blacklisted URLs that appear as sub-strings of the URL being checked.

4.1.2 Page Load Attack

While SpoofGuard was not susceptible to the CDN attack mentioned in Section 4.1.1, we were able to discover an exploit to which it was vulnerable. SpoofGuard examines the content on a web site when making a determination about whether or not the site is fraudulent. It must therefore wait for the entire web page to load before it can make a decision. This was confirmed during our tests when we noticed that while a page was loading, SpoofGuard would display a yellow icon (indicating that it cannot determine whether or not the site is fraudulent). After all content on the web page had loaded, only then might the icon change to either red or green. We hypothesized that if a page took an extremely long time to load, the indicator would remain yellow for a dangerously long period of time; a minute or two are more than adequate for a user to enter authentication information on a given phishing page.

To test this, we constructed a simple PHP script and mirrored a phishing site that SpoofGuard had previously identified. This PHP script consisted of five lines that created a GIF header. Upon sending the GIF header, the script would then enter an infinite loop, transmitting one byte per second. We placed this image on the phishing

site and visited it using SpoofGuard. We found that since the web page would take an infinite amount of time to load, SpoofGuard would never display anything other than the yellow icon (which it displays on most non-phishing sites anyway). From the user's perspective, the entire web page would appear to be rendered. Only savvy users would be able to tell that the page is still loading, but it is unclear if even they would find this suspicious. Thus, any phishing page can be easily altered to prevent SpoofGuard from warning users.

eBay was the only other toolbar on which we were able to demonstrate this attack (by hosting our own spoofed PayPal site). However, without the ability to add our script to sites identified as phishing by the other toolbars we were unable to test their vulnerability to this attack.

This vulnerability is quite simple to fix. A default timeout needs to be added to vulnerable toolbars so that they will stop loading a web page once the timeout occurs. They should then evaluate the portion of the page that has been received to determine the risk of it being a phishing site. This timeout needs to be short enough that users will be unable to submit information to the web page before the toolbar can evaluate it. One way of ensuring this is by not displaying the web page until the toolbar has had a chance to make a determination. Additionally, if the timeout has expired and some of the content on the page has failed to load, the toolbar could fill these spaces with warnings (e.g. replacing incomplete images with images of warnings).

5. DISCUSSION

5.1 Toolbar Performance

Overall, we found that the anti-phishing toolbars that were examined in this study left a lot to be desired. SpoofGuard did a very good job at identifying fraudulent sites, but it also incorrectly identified a large fraction of legitimate sites as fraudulent. EarthLink, Google, Netcraft, Cloudmark, and IE7 identified most fraudulent sites correctly and had few, if any, false positives, but they still missed more than 15% of fraudulent sites. The other four toolbars we tested could correctly identify less than half the fraudulent sites, and one did not correctly identify any fraudulent sites. Many of the toolbars we tested were vulnerable to some simple exploits as well.

Our experiments also suggest that there is no single technique that will always outperform others for identifying phishing web sites. Most of the tools we tested used blacklists, but only half of them were able to identify the majority of phishing web sites. We don't know the size of the blacklists used by each toolbar, nor do we know what heuristics are used by any of the toolbars other than SpoofGuard. We suspect that the toolbars that performed best use larger and more frequently updated black lists. They may also use heuristics that allow them to detect phishing sites that haven't yet been put on the blacklist. The only toolbar we tested that is known to make no use of blacklists was SpoofGuard. While it was able to identify the majority of phishing sites using only heuristics, it still missed some phishing sites and it had a very high false positive rate. SpoofGuard could potentially be improved through the use of a whitelist, which would prevent the problems that occurred when phishing sites were visited before their corresponding legitimate sites. The whitelist would not necessarily need to be extremely large or updated frequently to be effective. The success of a blacklist relies on massive amounts of data being collected at frequent intervals. Relying solely on heuristics requires that the software is designed with the foresight to prevent circumvention. In this study we were able to exploit both techniques, which leads us to believe that a combination of techniques is necessary.

5.2 Testing Methodology

Testing anti-phishing toolbars is a time consuming and difficult process. In order for results to be comparable, multiple toolbars need to be tested on the same set of URLs within a short time frame, and URLs are only useful for testing purposes while they are fresh. Although we were able to automate much of the testing process, we still found the process of identifying phishing URLs to test to be problematic. Ideally toolbars should be tested with URLs extracted from phishing messages immediately after those messages arrive in users' mailboxes. However, it takes time for phishing messages to be identified and propagated through phish feeds. We were able to collect URLs fresh enough that the sites had not yet been taken down, but we were unable to determine how fresh the

URLs we tested actually were. However, given the small number of improvements we saw in toolbar performance over the 24 hour period after we began testing each URL, we suspect that most of the URLs we tested were at least several hours old, and thus had already made their way onto many of the black lists. In order to test the speed at which toolbars are able to identify phishing sites and add them to their black lists we would need a fresher source of phishing URLs.

We also observed some variability in results depending on which phish feed we used. When we used the APWG feed, a much larger fraction of the sites we tested were taken down within 24 hours than when we used phishtank.com. Of the toolbars we tested using both feeds, some performed better on sites from one feed and some performed better on sites from the other feed. It is likely that some of the toolbars use these feeds as input into their blacklists; however, as none of the tools properly identified all of the phishing URLs, they apparently don't put all the URLs in these feeds into their blacklists automatically. When we compared our results with those of a similar study conducted by 3Sharp, we also found the results to be comparable for some toolbars but not for others [19]. Because the 3Sharp study and our experiments 2 and 3 were each conducted during different time periods and with different toolbar versions, it is unclear how much of the differences we observed were due to the feeds and how much were due to changes in the toolbars themselves or in their procedures for maintaining their blacklists. We would like to repeat experiment 3 with other feeds to gain additional insights into this.

We conducted all of our tests using toolbars' default configuration options. It would also be interesting to test toolbars with multiple configuration options to observe the impact these options have on toolbar accuracy and false positives.

5.3 User Interfaces

Prior research has focused on user studies of new anti-phishing solutions, not on solutions that are in widespread use. Literature on the usability of popular anti-phishing solutions is scarce at best. Since our study only measured the technical accuracy of five popular anti-phishing toolbars, we have only anecdotal evidence of their usability.

Eight of the ten toolbars examined employed indicators based on red and green color schemes. Green represents a legitimate site, and red represents a positively identified phishing site. Seven of the ten toolbars also use a yellow or gray indicator to indicate that nothing conclusive is known about the site. Given the predominance of red/green color blindness, this may be a poor choice unless the colored indicator includes other readily noticeable cues.

Besides colored indicators, IE7, Netscape, eBay, Netcraft, Google, and SpoofGuard use popup dialog boxes to warn when a site has been identified as fraudulent. While the Netscape, eBay, and Netcraft dialog boxes offer to not even display the phishing site, the SpoofGuard dialog box contains "yes" and "no" buttons with which to dismiss it. Regardless of which button is pressed, the web page remains open. Previous studies have shown that when presented with dialog boxes containing buttons with which to dismiss them, most users will simply dismiss the boxes without reading them [14]. When Google's toolbar encounters a site that it has positively identified as phishing, it darkens the page to draw attention to a dialog box with similar options as the Netcraft toolbar. IE7 does not even display the page, instead showing a different page where the user is given the choice of displaying the suspected phishing site or closing the window. eBay's toolbar puts a red warning box at the top of the suspected phishing site, but does not interact with the user in any meaningful way. The TrustWatch and Cloudmark toolbars do not present the user with any indications beyond the red/green/yellow icons. Since we were not able to get the McAfee toolbar to detect any phishing sites, we are unsure if it does anything beyond turning red.

Based on our cursory review, all of the toolbars examined appear to have some usability problems. We believe that it is important for these problems to be resolved if these toolbars are to be effective. 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.

In future anti-phishing toolbar studies, we also plan to conduct usability testing. As stated previously, a technically sound toolbar is of little use if users are unsure of what it is trying to communicate to them. Previous research has examined the relative successes of using different techniques for informing users about phishing. While such research is needed for testing new interfaces, this says little about existing solutions which are already in popular use. Thus, we believe it would be beneficial to have users test existing popular anti-phishing toolbars to observe how they react to the warnings. Key things to look for would be whether the toolbars adequately convey threats and how users react to these threats as well as how they react to any false positives.

6. CONCLUSIONS

We conducted two experiments assessing the effectiveness of five anti-phishing toolbars. To facilitate evaluation of larger data sets across longer periods of time, we developed an automated test bed for assessing the effectiveness of anti-phishing toolbars. We found that three of the 10 toolbars, SpoofGuard, EarthLink and Netcraft, were able to identify over 75% of the phishing sites tested. However, four of the toolbars were not able to identify even half the phishing sites tested. At the same time, SpoofGuard incorrectly identified 38% of the legitimate URLs as phishing URLs. It would seem that such inaccuracies might nullify the benefits SpoofGuard offers in identifying phishing sites. The 10 toolbars that we examined used a variety of methods for identifying fraudulent sites; however, we were able to exploit vulnerabilities in most of them. Thus, much more work needs to be done in this area from a technical standpoint. Yet even if it is possible to create a technically sound anti-phishing toolbar, it is still unclear as to whether or not this would be beneficial to users. Usability problems plague all varieties of software, security software in particular. When using an anti-phishing toolbar, poor usability could mean the difference between correctly steering someone away from a phishing site and having them ignore the warnings only to become a victim of identity theft. Thus, we plan to further examine both the technical aspects of this domain as well as the human factors.

7. ACKNOWLEDGEMENTS

Thanks to Joseph Schwartz for his assistance conducting our preliminary studies and to the other members of the Supporting Trust Decisions project for their feedback. Thanks to Tom Phelps for providing us code for Robust Hyperlinks. This work was supported in part by National Science Foundation under grant CCF-0524189, and by the Army Research Office grant number DAAD19-02-1-0389. 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. Anti-Phishing Working Group. Phishing Activity Trends Report. June, 2006. http://www.antiphishing.org/reports/apwg_report_june_06.pdf.
2. Chou, Neil, Robert Ledesma, Yuka Teraguchi, Dan Boneh and John C. Mitchell, “Client-Side Defense against Web-Based Identity Theft,” in Proceedings of The 11th Annual Network and Distributed System Security Symposium (NDSS '04) Held in San Diego, CA February, 2004. http://crypto.stanford.edu/SpoofGuard/webspoof.pdf.
3. Cloudmark, Inc. Accessed: September 5, 2006. http://www.cloudmark.com/desktop/download/.
4. Computer Crime Research Center. “Netscape: Anti-Phishing Bundled.” February 2, 2005. Accessed: November 9, 2006. http://www.crime-research.org/news/02.02.2005/938/.
5. The Coral Content Distribution Network. Accessed: June 13, 2006. http://www.coralcdn.org/.
6. Dao, Thanh. Term Frequency/Inverse Document Frequency Implementation in C#. Accessed: September 9, 2006. http://www.codeproject.com/csharp/tfidf.asp.
7. Dhamija, R., Tygar, J.D., and Hearst, M. 2006. Why phishing works. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal, Quebec, Canada, April 22 - 28, 2006). CHI '06.

8. Downs, Julie S., Mandy Holbrook, and Lorrie Cranor, "Decision Strategies and Susceptibility to Phishing," in Proceedings of The 2006 Symposium on Usable Privacy and Security Held in Pittsburgh, PA 12-14 July 2006.
9. eBay, Inc. Using eBay Toolbar's Account Guard. Accessed: June 13, 2006. http://pages.eBay.com/help/confidence/account-guard.html.
10. EarthLink, Inc. EarthLink Toolbar. Accessed: November 9, 2006. http://www.earthlink.net/software/free/toolbar/.
11. GeoTrust, Inc. TrustWatch Toolbar. Accessed: June 13, 2006. http://toolbar.trustwatch.com/tour/v3ie/toolbar-v3ie-tour-overview.html.
12. Google, Inc. Google Safe Browsing for Firefox. Accessed: June 13, 2006. http://www.google.com/tools/firefox/safebrowsing/.
13. Jagatic, T., Johnson, N., Jakobsson, M., Menczer, F. Social Phishing. *Commun. ACM. To appear.* http://www.indiana.edu/phishing/social-network-experiment/phishing-preprint.pdf
14. Jendricke, U, D. Gerd tom Markotten, "Usability Meets Security - The Identity-Manager As Your Personal Security Assistant for The Internet," in Proceedings of The 16th Annual Computer Security Applications Conference (ACSAC'00), 2000.
15. Kerner, Sean Michael. 2006. Firefox 2.0 Bakes in Anti-Phish Antidote. InternetNews. http://www.internetnews.com/dev-news/article.php/3609816.
16. McAfee, Inc. McAfee SiteAdvisor. Accessed: November 9, 2006. http://www.siteadvisor.com/.
17. Microsoft Corporation. Internet Explorer 7. Accessed: November 9, 2006. http://www.microsoft.com/windows/ie/default.mspx.
18. Netcraft. Netcraft Anti-Phishing Toolbar. Accessed: June 13, 2006. http://toolbar.netcraft.com/.
19. Netscape Communications Corp. "Security Center." Accessed: November 9, 2006. http://browser.netscape.com/ns8/product/security.jsp.
20. Phelps, Thomas A., and Robert Wilensky. "Robust Hyperlinks and Locations," in D-Lib Magazine, July/August 2000.
21. The PlanetLab Consortium. PlanetLab: Home. Accessed: June 13, 2006. http://www.planet-lab.org/.
22. Robichaux, Paul. 2006. Gone Phishing: Evaluating Anti-Phishing Tools for Windows. 3Sharp Technical Report. http://www.3sharp.com/projects/antiphishing/gone-phishing.pdf
23. Rouse, Mark. Comparing Images using GDI+. Accessed: September 9, 2006. http://www.codeproject.com/dotnet/comparingimages.asp.
24. Tor: An Anonymous Internet Communication System. Accessed: September 7, 2006. http://tor.eff.org/.
25. Wu, Min, Robert C. Miller, and Simson L. Garfinkel, "Do Security Toolbars Actually Prevent Phishing Attacks?" in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Held in Montreal, QC 22-28 April 2006, 601-610. New York: ACM Press, 2006.