Phinding Phish: Evaluating Anti-Phishing Tools

Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong
Carnegie Mellon University
zysxqn@andrew.cmu.edu, {egelman, lorrie, jasonh}@cs.cmu.edu

Abstract

There are currently dozens of freely available tools to combat phishing and other web-based scams, many of which are web browser extensions that warn users when they are browsing a suspected phishing site. We developed an automated test bed for testing anti-phishing tools. We used 200 verified phishing URLs from two sources and 516 legitimate URLs to test the effectiveness of 10 popular anti-phishing tools. Only one tool was able to consistently identify more than 90% of phishing URLs correctly; however, it also incorrectly identified 42% of legitimate URLs as phish. The performance of the other tools varied considerably depending on the source of the phishing URLs. Of these remaining tools, only one correctly identified over 60% of phishing URLs from both sources. Performance also changed significantly depending on the freshness of the phishing URLs tested. Thus we demonstrate that the source of phishing URLs and the freshness of the URLs tested can significantly impact the results of anti-phishing tool testing. We also demonstrate that many of the tools we tested were vulnerable to simple exploits. In this paper we describe our anti-phishing tool test bed, summarize our findings, and offer observations about the effectiveness of these tools as well as ways they might be improved.

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 alone [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 tools." For example, eBay offers a free tool that can positively identify the eBay site, and Google offers a free tool aimed at identifying any fraudulent site [9], [12]. As of September 2006, the free software download site *Download.com*, listed 84 anti-phishing tools. Unfortunately, few empirical studies have been performed to examine the effectiveness of these tools. Thus, while many anti-phishing tools 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 tools. These studies have shown that most users are likely to fall for phishing scams, and that many users ignore warnings provided by anti-phishing tools [7], [8], [13], [25]. However, little empirical data is available on the accuracy of these tools 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 tools. Second, we describe the results of experiments that assess the accuracy of 10 popular anti-phishing tools that use differing techniques to identify phishing sites. Third, we describe techniques we developed for circumventing many of the tools tested. Our paper provides the anti-phishing community with insights into the effectiveness of several approaches to combating phishing as well as a methodology for testing anti-phishing tools.

2. Overview of Anti-Phishing Tools

There are a variety of methods that can be used to identify a web page as a phishing site, including whitelists (lists of known safe sites), blacklists (lists of known fraudulent sites), heuristics, and community

ratings. The tools examined in this study employ differing combinations of these methods. We used publicly available information provided on the tool download web sites as well as our observations to get a basic understanding of how each tool functions.

2.1. CallingID Toolbar

The CallingID Toolbar, shown in Figure 1, boasts its use of 54 different verification tests in order to determine the legitimacy of a given site. Like many of the other toolbars, CalingID relies on passive visual indicators. These indicators change from green—to represent a known-good site; to yellow—to represent a site that is "low risk;" to red—to represent a site that is "high risk," and therefore probably a phishing site. Some of the heuristics used include examining the site's country of origin, length of registration, popularity, user reports, and blacklist data. The CallingID Toolbar runs on Microsoft Windows 98/NT/2000/XP with Internet Explorer [2].

2.2. Cloudmark Anti-Fraud Toolbar

The Cloudmark Anti-Fraud Toolbar, shown in Figure 2, relies on user ratings [4]. 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 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. No other heuristics are used in determining a site's rating. Sites determined to be fraudulent are blocked and users are redirected to an information page and given the option of overriding the block. The Cloudmark Anti-Fraud Toolbar runs on Microsoft Windows 98/NT/2000/XP with Internet Explorer. After our study began we learned that Cloudmark is no longer supporting this toolbar. Cloudmark has since removed this toolbar from their web site. They now offer a phishing URL feed for other toolbars and similar applications and a tool called Cloudmark Desktop that works in conjunction with the Microsoft Outlook and Microsoft Outlook Express email clients and labels phishing emails based on millions of reports from users each day. We have not tested Cloudmark Desktop.

2.3. EarthLink Toolbar

The EarthLink Toolbar, shown in Figure 3, 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

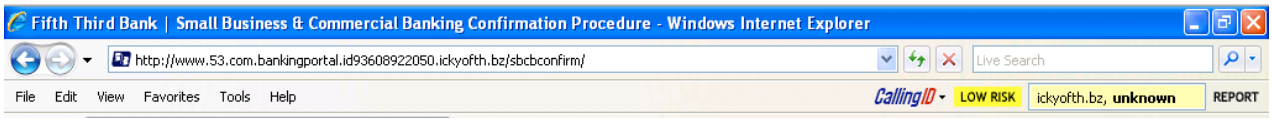

Figure 1: The CallingID Toolbar indicating a low-risk site.

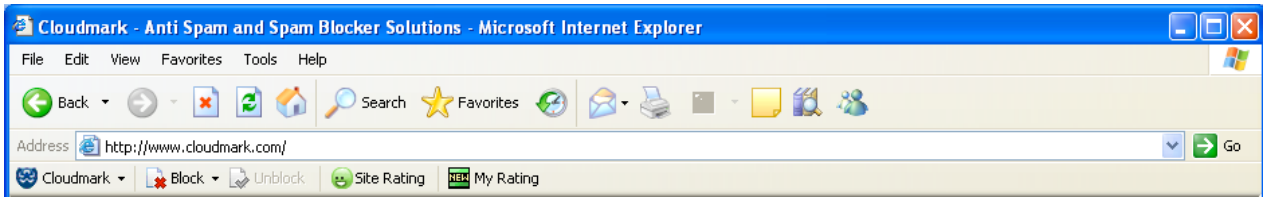

Figure 2: The Cloudmark Anti-Fraud Toolbar indicating a legitimate site.

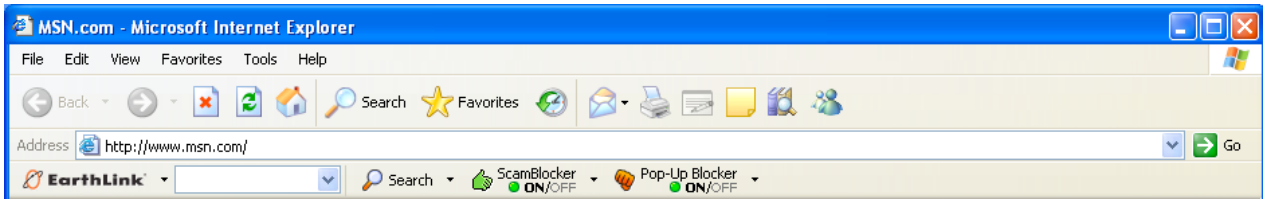

Figure 3: The EarthLink Toolbar indicating a legitimate site.

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." Sites determined to be fraudulent are sometimes blocked, in which case users are redirected to an information page and given the option of overriding the block (and a green thumb is displayed on the information page). The EarthLink Toolbar runs under Internet Explorer as well as Firefox [10].

2.4. eBay Toolbar

The eBay Tool, shown in Figure 4, 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. Known phishing sites are blocked and a pop-up appears, giving users the option to override the block. 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.

2.5. Firefox 2

Firefox 2.0, shown in Figure 5, includes a new feature designed to identify fraudulent web sites. Originally, this functionality was an optional extension for Firefox as part of the Google Safe Browsing Toolbar. URLs are checked against a blacklist, which Firefox downloads periodically [15]. The feature 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. Optionally, the feature can send every URL to Google to determine the likelihood of it being a scam. According to the Google toolbar download site, the toolbar combines "advanced algorithms with reports about misleading pages from a number of sources [12]." We suspect that this means it uses blacklists as well as heuristics. Firefox 2.0 runs on Microsoft Windows, Apple Mac OS X, and Linux. The Google Safe Browsing Toolbar on which this functionality is based runs on Microsoft Internet Explorer under Windows XP/2000 SP3+, or Firefox on most platforms.

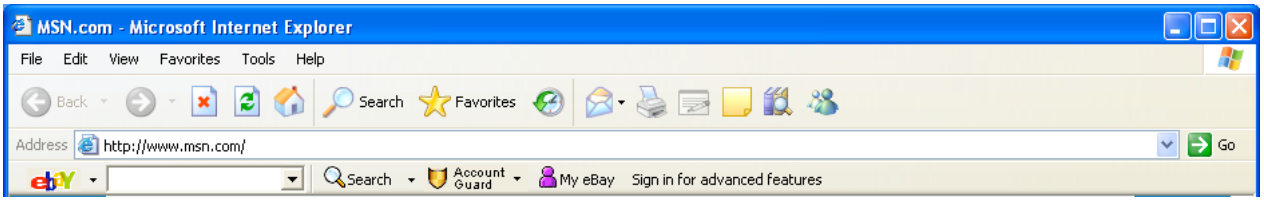

Figure 4: The eBay Toolbar at a site not owned by eBay that is not known to be a phishing site.

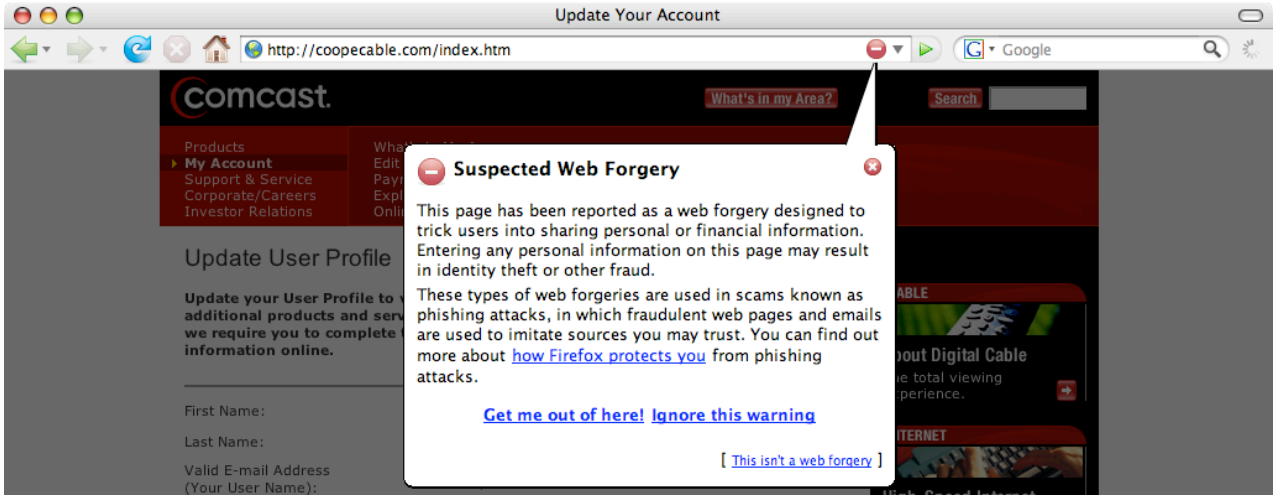

Figure 5: Firefox 2.0 at a suspected fraudulent site.

2.6. GeoTrust TrustWatch Toolbar

GeoTrust's TrustWatch Tool, shown in Figure 6, 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 tool. The toolbar also lets users 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].

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 tool largely relies on a blacklist hosted by Microsoft. However, it also uses some heuristics when it encounters a site that isn't on 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.

2.8. Netcraft Anti-Phishing Toolbar

The Netcraft Anti-Phishing Toolbar, shown in Figure 8, uses several methods to determine the legitimacy of a 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 (tool & 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

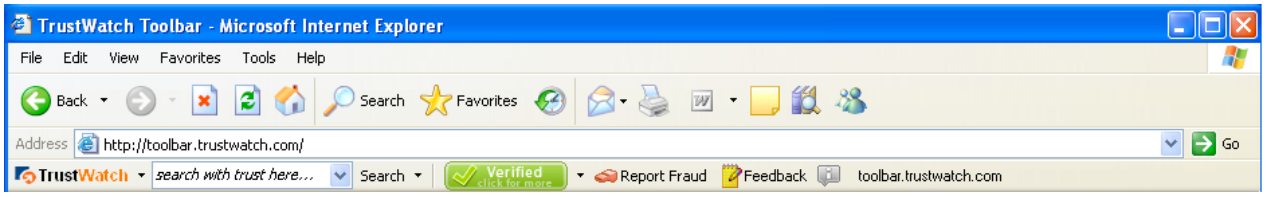

Figure 6: The GeoTrust TrustWatch Toolbar at a verified site.

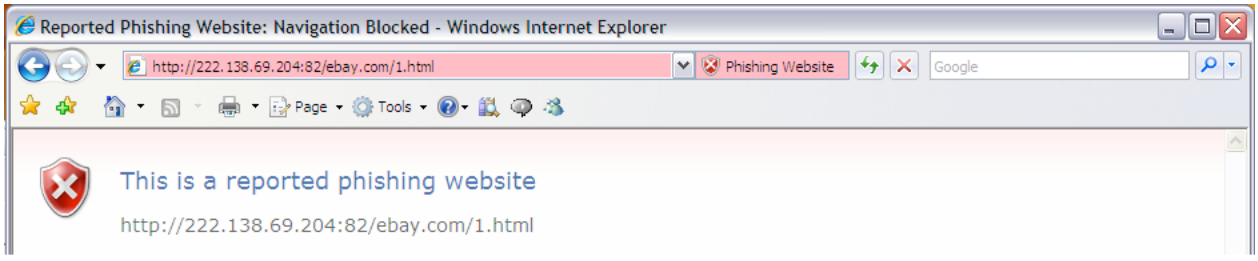

Figure 7: The Microsoft Phishing Filter in Windows Internet Explorer 7 at a fraudulent web site.

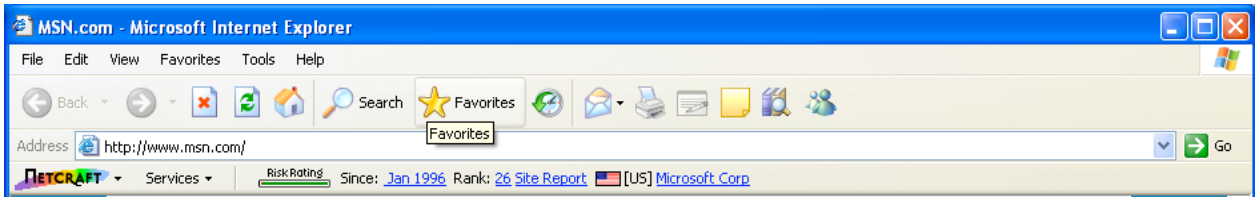

Figure 8: The Netcraft Anti-Phishing Toolbar at a legitimate web site.

also use the toolbar to access a more detailed report on a web site. The Netcraft Anti-Phishing Toolbar runs on Firefox on most platforms, and on Microsoft Internet Explorer under Windows 2000/XP.

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 [5]. 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.

2.10. SpoofGuard

SpoofGuard, shown in Figure 10, is an anti-phishing toolbar developed at Stanford University [2]. Unlike the other tools described here, SpoofGuard does not use whitelists 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 identical 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].

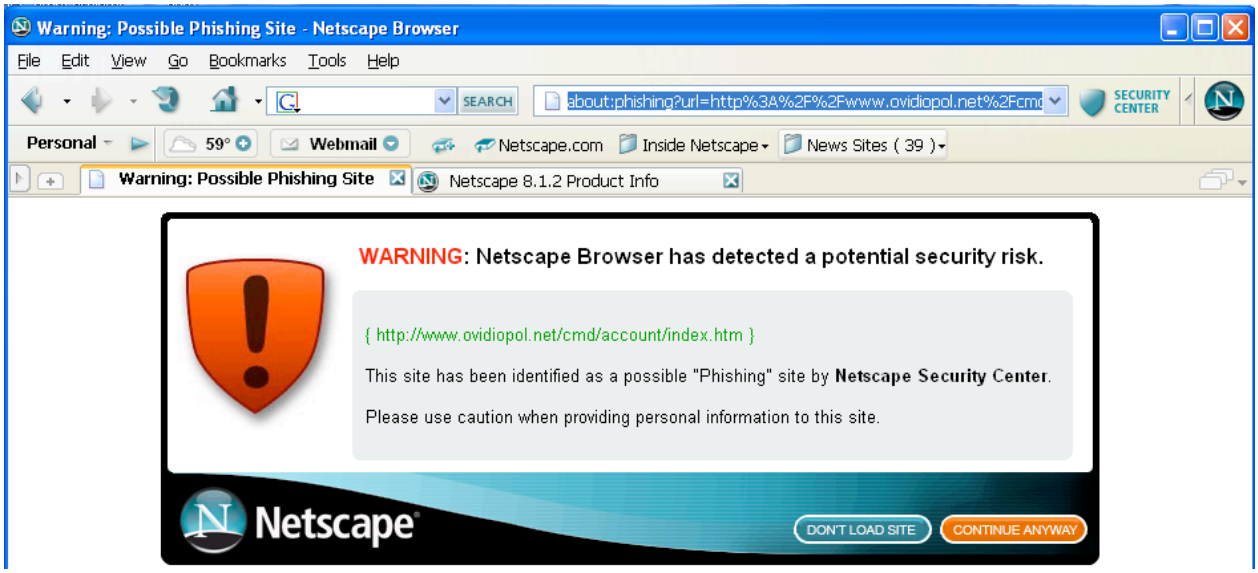

Figure: 9 Netscape 8.1 web browser at a fraudulent web site.

Figure 10: SpoofGuard at a legitimate web site.

3. Anti-Phishing Tool Evaluation

We conducted a series of experiments designed to investigate the accuracy of anti-phishing tools. Our first experiment involved manually evaluating five of the tools described above. This gave us a feel for the behavior and effectiveness of the various tools, but proved labor intensive and posed significant logistical difficulties. As a result, we developed an automated testing system and used it to conduct our subsequent experiments. Using our automated testing system, we were able to test how each of 10 tools responded to a set of URLs multiple times over a 24 hour period, allowing us to observe the effect of blacklist updates and of phishing sites being taken down.

3.1. Manual Evaluation of Anti-Phishing Tools

Our first experiment was conducted using five laptops to simultaneously test five anti-phishing tools. One experimenter was assigned to each laptop. The experimenters manually entered URLs to be tested into web browsers running on each laptop, and then observed and recorded the results. This was a slow and labor-intensive process.

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 4.5 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. We also tried to find a source that was not used by any of the tools we were testing for updating their blacklist, although it was difficult to determine conclusively whether any tools were using the phishing feeds we tried. 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 tool was tested with 50 confirmed phishing URLs identified within the previous 36 hours. Because we generally did not receive more than 20 new phishing URLs each day, we conducted the study during three separate sessions over a two-week period.

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 tools. 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 tool. 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 freely available .NET components, including Compare Images [23], which checks if two images are identical.

Step 1 – Retrieve Potential Phishing Sites. First, the task manager obtains a set of phishing URLs to test

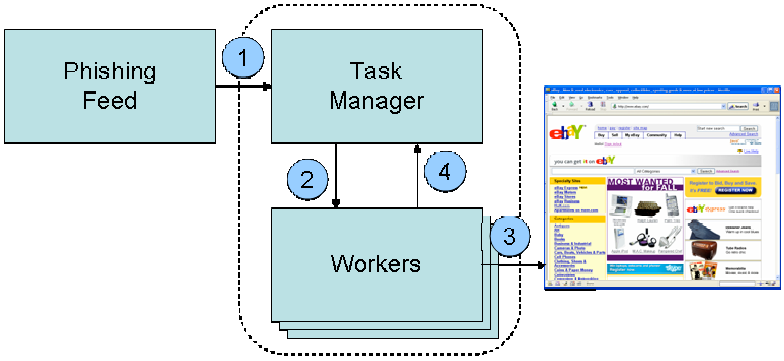

Figure 11: High-level system architecture for our anti-phishing evaluation test bed. The Task Manager (1) gets an updated list of URLs from a phishing feed, and then (2) sends that URL to a set of Workers. Each worker (3) retrieves a web page and checks whether the web page was labeled as a phishing 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.

against. We experimented with automating this process by extracting URLs from a feed of validated phishing URLs or by using our own heuristics to select phishing URLs from a feed of unvalidated phishing email messages.[1] We found that by the time phishing URLs are validated and distributed on a phishing feed, they tend not to be very fresh and many of the sites have been taken down. Furthermore, some phishing tools update their blacklists using data from validated phishing feeds. Unvalidated phishing URLs or email messages are a better source for fresh phishing URLs; however, use of these sources requires that phishing URLs be manually selected and validated. This process can be partially automated. However, if heuristics are used to select valid phishing URLs, phishing sites that cannot be identified using those heuristics may be excluded from the test and thus the results may be biased in favor of tools that use heuristics similar to the selection heuristics. In order to get large numbers of very fresh phishing URLs without bias we decided to manually select and validate phishing URLs from a phishing feed and repository, using automated tools only to extract URLs from suspected phishing messages and remove those we had already seen. 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 to Workers. In the second step, the task manager sends each URL to a set of workers, each of which is running a separate tool. 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 tools that work with the same web browser, as the tests should be run with only one tool installed in the web browser at a time. Virtual machines can reduce the number of test machines needed.

Step 3 – Worker Evaluates Potential Phishing Site. In the third step, each worker downloads the specified web page, examines whether its tool 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 tool. Each tool 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 tool can be set up to be in a known location in the web browser. Thus, we simply capture screenshots of the tools beforehand and compare relevant portions of those images to screenshots of the current state of the tool. The primary advantage of this image-based approach is that it works for all tools regardless of the programming language in which the tool was written, whether or not the tool 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.

3.3. Evaluation of Anti-Phishing Tools

We used our automated anti-phishing test bed to evaluate 10 anti-phishing tools. We tested the built in phishing filters in Microsoft Internet Explorer 7.0.5700.6, Netscape Navigator 8.1.2., and Firefox 2.0. We used Internet Explorer 6 to test the following tools: CallingID 1.5.0.150, Cloudmark 1.0, EarthLink 3.3.44.0, eBay 2.3.2.0, Netcraft 1.7.0, TrustWatch 3.0.4.0.1.2, and SpoofGuard.

We began our experiment using FireFox 1.5.0.6 to test Google Toolbar 2.1. However when Firefox 2.0 was released it included the Safe Browsing feature from the Google Toolbar, and we found that when the Google Toolbar was configured with its default settings it produced the same results as FireFox 2.0 configured with the "Ask Google" option. Thus, we decided to continue our experiment using Firefox 2.0 instead of the Google Toolbar. We also tested McAfee SiteAdvisor 1.7.0.53 in the early part of our experiment, but removed it from the experiment when it became apparent that it does not actually detect phishing URLs.[2] After removing McAfee SiteAdvisor from our experiment, we added CallingID. Thus we did not test CallingID on the complete set of phishing URLs.

[1] Phishing email messages often contain multiple links, some of which lead to fraudulent sites and some of which lead to legitimate sites.

[2] The McAfee web site claimed that SiteAdvisor protects against "online scams," however; a company representative explained that claim refers to protection against long-running online scams such as greencard lotteries, and not the more transient phishing attacks. McAfee has a premium product, SiteAdvisor Plus, that does provide phishing protection. However, we have not yet tested this product.

We configured all tools with their default settings. However, we tested Firefox 2.0 with the default setting (which uses a blacklist, downloaded approximately every 30 minutes) and with the "Ask Google" option (which sends every URL visited to Google for testing) and report these results separately. 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.

On November 4-5, 2006 we tested 100 phishing URLs extracted from the list of unvalidated phishing reports on phishtank.com. 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.[3] We extracted 100 confirmed, active phishing URLs and examined each URL within six hours of its being posted on phishtank.com.

On November 21 and 27, 2006 we tested 100 phishing URLs extracted from the APWG feed of reported phishing emails. We downloaded new messages from the feed every two hours and manually identified and verified active phishing URLs from these messages. We extracted 100 confirmed, active phishing URLs and examined each URL within two hours of its being received on our APWG feed.

Each URL was tested against each tool within one hour of extraction. In addition, each URL was tested against all tools except SpoofGuard 1, 2, 12, and 24 hours later. By testing each URL multiple times we were able to observe blacklist updates as well as how long it took for phishing sites to be taken down.

During preliminary testing, 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. However, as SpoofGuard's determination is based only on heuristics and not on blacklists, SpoofGuard's assessment does not change over time. Thus we decided to test SpooGuard only once on each URL. The apparent change in accuracy of SpoofGuard over time in our reported results is due entirely to some of the phishing web sites being taken down.

[3] Phishtank.com also provides a feed of phishing URLs that have been verified by users. We manually selected URLs from those submitted rather than using the feed in order to get fresher URLs and to reduce the chance of using a feed that was being used by one of the tools being tested.

We compiled a list of 516 legitimate URLs to test for false positives. The URLs were compiled from the following sources:

- 416 URLs were taken from the list of 500 legitimate URLs compiled by 3Sharp and published in a September 2006 report [22] (the remaining 84 URLs tested by 3Sharp were no longer active).

- 35 URLs were compiled by selecting the log-in pages of sites that are often attacked by phishers, such as www.citibank.com and www.paypal.com. These pages were selected to see whether tools can distinguish phishing site from the legitimate sites they commonly spoof.

- 35 URLs were compiled by selecting the most popular web pages reported by Alexa Web Search. These pages were selected to see whether tools label frequently-visited pages correctly.

- 30 pages were compiled by selecting random pages from http://random.yahoo.com/fast/ryl, and manually verifying that they are legitimate. These pages were selected to see whether tools label random legitimate URLs correctly.

3.3.1. Catch Rate. The most important function of an anti-phishing tool is to accurately and conspicuously identify phishing web sites that users visit. Since not all of the tools provide the same types of indicators, we had to come up with a standard way of measuring accuracy. Some of the tools provide binary indicators (i.e. either that site is phishing or it is not), while some tools use a ternary system (i.e. a site can be phishing, not phishing, or unknown). We count only positive identification of phishing as a "catch." Most of the tools we tested have only one form of positive identification. However, IE7 and EarthLink can either warn or block when they identify phish, so we count either as a catch. We do not count "unvalidated" ratings or other uncertain ratings as a catch.[4] We define "catch rate" (or true positive rate) as the number of phishing sites positively identified by a tool out of the total number of active phishing sites visited, with sites that had been taken down at the time of testing removed from the denominator. The rationale here is that it makes no difference whether a tool identifies a taken-down site as a phishing site, since a site that has been taken down does no harm to the user.

[4] One previous study counted other warning indicators, such as TrustWatch's yellow "not verified" indicator as catches [22].

Table 1, Figure 12, and Figure 13 show the percentage of phishing sites correctly identified over time using the phishtank.com and APWG URLs. After 24 hours, 70 of the phishtank.com URLs and 67 of the APWG URLs remained active. The performance of the tools varies considerably depending on the source of the URLs used for testing. Some tools performed significantly better with URLs from one source or the other, but none of the tools we tested performed well across the board. SpoofGuard had a consistently high catch rate of over 90%, but also had a 42% false positive rate. Of the other tools we tested, only IE7 had a catch rate better than 60% with both sources, but it still missed 25% of the APWG phishing URLs and 32% of the phishtank.com phishing URLs.

When we tested the tools with phishtank.com URLs, SpoofGuard, EarthLink, and Netcraft performed best at identifying phishing sites initially. A chi-square test ($p=0.01$) demonstrated that SpoofGuard performed significantly better than Netcraft. 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%.

When we tested the tools with the APWG URLs, SpoofGuard was able to identify 96% of the phishing sites. The next best tool, IE7, identified only 75% of the phishing sites initially, and 85% after 24 hours had passed. Netcraft, Firefox/Google, and EarthLink, were able to identify 50-60% of phishing sites initially and 70-75% of phishing sites after 24 hours. Firefox, TrustWatch, Netscape, CallingID, and CloudMark all identified less than 50% of phishing sites initially. TrustWatch and Firefox improved to over 65% after 24 hours, while Netscape, CallingID, and CloudMark improved but still remained under 50% after 24 hours. eBay identified 52% of phishing sites initially, and did not improve over the 24-hour testing period.

Our results indicate that the source of phishing URLs can have a major impact on test results. Neither of the sources we used lend themselves to use as completely automated feeds, and none of the tools we tested was able to correctly identify all of the phishing URLs from either of the feeds. Therefore we do not believe that any of the tools were updating their blacklists automatically from either of these sources directly. However, the APWG feed includes data from multiple sources, and some of the tools might update their blacklists using data from some of them. In addition, phishtank.com has a validated phishing URL feed that includes a subset of the URLs that we validated ourselves. It is likely that some of the tested tools use this feed to update their blacklists. We also observed that some types of phishing attacks appeared more frequently in one source than the other. For example, spoofs of eBay-owned brands appeared more often in the APWG feed. Finally, we checked the APWG feed more frequently than we checked phishtank.com, and thus we believe the APWG URLs to be fresher than the phishtank.com URLs.

	PhishTank					**APWG**				
Time since URL extraction	**0 hours**	**1 hour**	**2 hours**	**12 hours**	**24 hours**	**0 hours**	**1 hour**	**2 hours**	**12 hours**	**24 hours**
CallingID	NA	NA	NA	NA	NA	23 (23%)	26 (26%)	25 (27%)	30 (38%)	24 (36%)
Cloudmark	68 (68%)	68 (68%)	68 (69%)	64 (67%)	47 (67%)	22 (22%)	24 (24%)	21 (22%)	25 (31%)	25 (37%)
EarthLink	83 (83%)	83 (83%)	81 (82%)	78 (84%)	59 (84%)	54 (54%)	53 (54%)	51 (54%)	51 (64%)	47 (70%)
eBay	28 (28%)	28 (28%)	26 (27%)	24 (26%)	18 (26%)	52 (52%)	52 (53%)	51 (54%)	43 (54%)	35 (52%)
IE7	68 (68%)	68 (68%)	67 (68%)	62 (67%)	47 (67%)	75 (75%)	74 (75%)	72 (77%)	67 (84%)	58 (87%)
Firefox	NA	NA	NA	NA	NA	28 (28%)	50 (50%)	51 (54%)	47 (59%)	44 (66%)
Firefox/ Google	70 (70%)	70 (70%)	70 (71%)	71 (76%)	59 (84%)	53 (53%)	54 (55%)	56 (60%)	56 (70%)	49 (73%)
Netcraft	77 (77%)	77 (77%)	73 (74%)	69 (74%)	56 (80%)	60 (60%)	59 (60%)	57 (61%)	62 (78%)	49 (73%)
Netscape	8 (8%)	10 (10%)	10 (10%)	9 (10%)	15 (21%)	31 (31%)	31 (31%)	32 (34%)	37 (46%)	30 (45%)
SpoofGuard	91 (91%)	91 (91%)	89 (91%)	85 (91%)	64 (91%)	96 (96%)	95 (96%)	90 (96%)	78 (98%)	65 (97%)
TrustWatch	49 (49%)	49 (49%)	48 (49%)	45 (48%)	36 (51%)	44 (44%)	43 (43%)	44 (47%)	45 (56%)	45 (67%)
ActiveURLs	100	100	98	93	70	100	99	94	80	67

Table 1: Number of phishing sites correctly identified by anti-phishing tools. Note, SpoofGuard's catch rate is estimated after time 0.

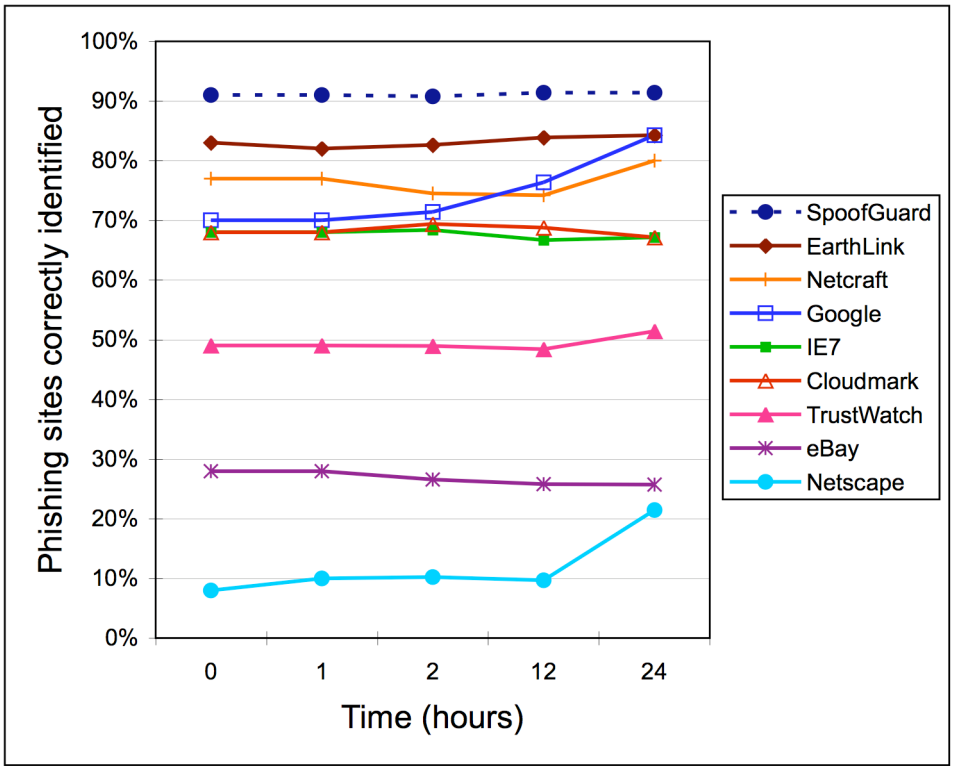

Figure 12: Catch rate of each tool over time using phishtank.com URLs. Note that SpoofGuard's catch rate is estimated after time 0.

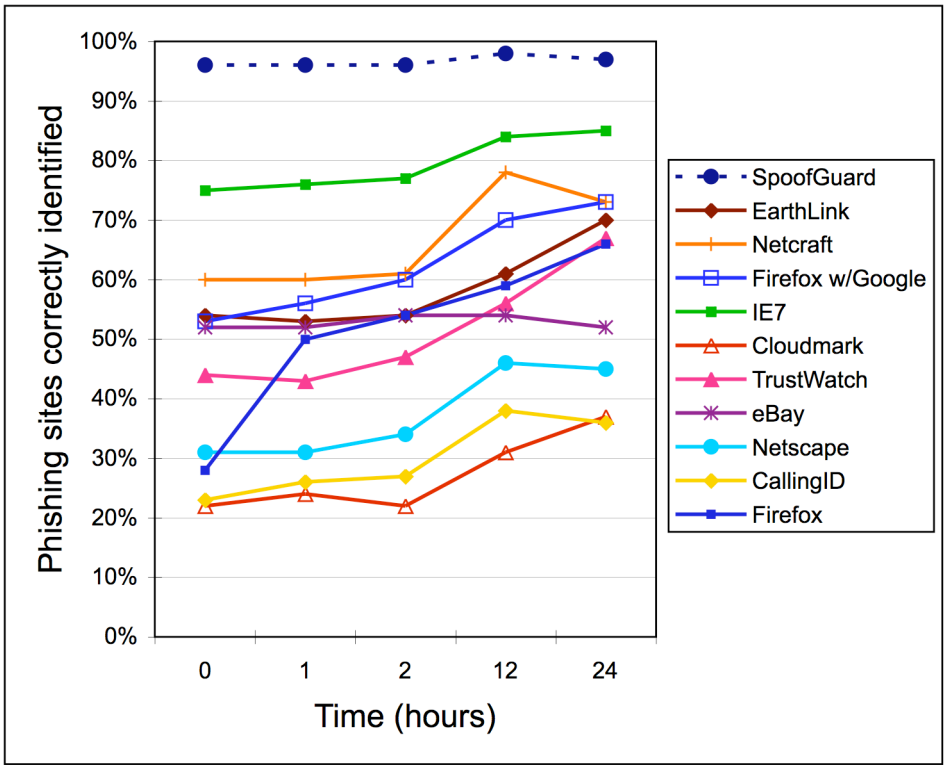

Figure 13: Catch rate of each tool over time using APWG URLs. Note that SpoofGuard's catch rate is estimated after time 0.

	phishtank.com URLs				APWG URLs			
Time since URL extraction	**1 hour**	**2 hours**	**12 hours**	**24 hours**	**1 hour**	**2 hours**	**12 hours**	**24 hours**
CallingID	N/A	N/A	N/A	N/A	3	0	6	0
Cloudmark	0	1	0	0	2	0	5	2
EarthLink	0	0	0	0	0	0	3	1
eBay	0	0	0	0	1	0	2	0
Firefox	N/A	N/A	N/A	N/A	23	3	1	5
Firefox/Google	0	1	4	5	1	5	3	3
IE7	0	1	0	0	0	0	2	1
Netcraft	0	1	0	4	0	4	10	0
Netscape	2	0	0	7	0	1	10	0
SpoofGuard	0	0	0	0	0	0	0	0
TrustWatch	0	0	0	0	0	1	5	9
Active URLs	100	98	93	70	99	94	80	67

Table 2: Number of phishing sites initially identified incorrectly that were later identified correctly by anti-phishing tools.

As Table 2 shows, most phishing sites are detected quickly by the tools we tested, but some are detected after several hours or even a day or more after they appear in phishing emails. Some tools improved more than others as our experiment progressed. When using phishtank.com URLs, only five of the tools were able to correctly identify phishing sites in later tests that they incorrectly identified initially. The changes in accuracy observed for the other tools are due entirely to some of the phishing sites being taken down. When using APWG URLs, all tools except SpoofGuard were able to correctly identify phishing sites in later tests that they incorrectly identified initially. The larger changes over time observed when using the APWG URLs are likely due to the APWG URLs being fresher than the phishtank.com URLs. The biggest improvement was seen with Firefox, which correctly identified 23 APWG phishing sites after 1 hour that it had missed initially. As we were testing Firefox we observed that it initially missed a large number of sites until it automatically downloaded the latest version of its blacklist. This suggests that Firefox test results (without the "ask Google" option) are likely to vary depending on how recently the blacklist has been downloaded.

Interestingly, we also saw that some of the tools initially made a correct identification of a phishing site and later reversed themselves. We observed this only once or twice with most of the tools, but we observed Netcraft make an incorrect reversal 11 times.

In general, the differing approaches taken by the tools resulted in their catching different sets of phish. All but one URL from each data set was caught by at least one tool at time 0. For 85% of the APWG phishing URLs we tested, at least three tools identified them correctly when they were first tested, and at least five tools identified them correctly after 24 hours. For 85% of the phishtank.com URLs, at least four tools identified them correctly when they were first tested, and at least five tools identified them correctly after 24 hours. It was rare, even after 24 hours for eight or more tools to identify a URL correctly. Some phishing URLs were missed by one of the better tools but caught by another, or even caught by one of the tools that did not perform well overall.

3.3.2. 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 tool must identify phishing sites, it should also be careful to not identify legitimate web pages as phishing.

Each tool was tested against 516 legitimate URLs. SpoofGuard erroneously labeled 42% of these URLs as phishing. In addition it reported that it was unsure about an additional 50% of these URLs. The only other tools to falsely identify any URLs as phishing sites were EarthLink and Cloudmark (which misidentified 1% of the legitimate sites), and CallingID (which misidentified 2% of the legitimate sites). CallingID, Cloudmark, EarthLink, and TrustWatch were also

unsure of a large number of sites. The false positives results are summarized in Table 3. Overall, false positives do not appear to be a major problem for most of the tools we tested.

	Falsely identified as phishing		Unsure	
CallingID	10	(2%)	177	(34%)
Cloudmark	5	(1%)	497	(96%)
EarthLink	5	(1%)	493	(96%)
eBay	0	(0%)	0	(0%)
Firefox	0	(0%)	0	(0%)
Firefox/Google	0	(0%)	0	(0%)
IE7	0	(0%)	0	(0%)
Netcraft	0	(0%)	0	(0%)
Netscape	0	(0%)	0	(0%)
SpoofGuard	218	(42%)	256	(50%)
TrustWatch	0	(0%)	256	(50%)

Table 3: Number of legitimate sites (out of 516 tested) falsely identified as phishing sites by anti-phishing tools

4. Tool Exploits

As we tested the five tools 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 tools could be modified to protect against them.

4.1. Content Distribution Networks

Nine of the ten tools we examined appear to rely on blacklists. Some of the tools take the entire URL into account, possibly by using a hash or pattern matching. Other tools 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 tool 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 [6]. 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 tool as fraudulent, this time passing them through Coral. Some of the tools failed to properly identify any of the URLs as fraudulent when they were passed through Coral. Figure 14 shows a comparison of the TrustWatch Tool visiting a phishing site with and without the use of Coral. As can be seen, the original phishing URL causes the tool 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, TrustWatch, Netcraft, and Netscape.

As would be expected, this exploit did not work on the SpoofGuard tool. 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.

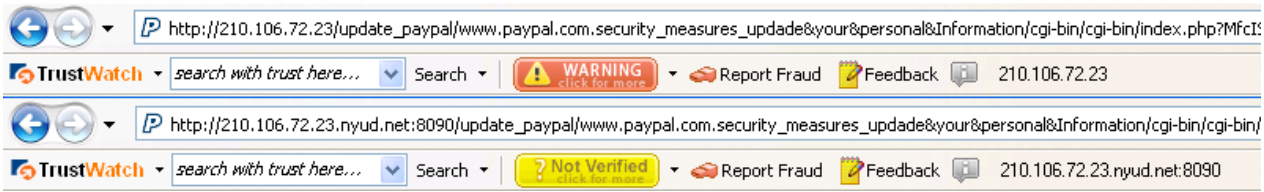

Figure 14: Demonstration of the CDN attack on the TrustWatch tool. The top screenshot shows TrustWatch correctly labeling a site as a phishing scam. The bottom screenshot shows how redirecting that same scam through the Coral CDN causes the same site to be labeled incorrectly.

4.2. Page Load Attack

While SpoofGuard was not susceptible to the CDN attack mentioned in Section 4.1, 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 tool 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 tools 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 tools 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 tool can evaluate it. One way of ensuring this is by not displaying the web page until the tool 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 tool could fill these spaces with warnings (e.g. replacing incomplete images with images of warnings).

5. Conclusions and Future Work

We conclude with our observations on tool performance, testing methodology, and user interfaces, as well as some directions for future work.

5.1. Tool Performance

Overall, we found that the anti-phishing tools 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. The performance of the other tools varied considerably depending on the source of the phishing URLs. Of these other tools, only IE7 was able to correctly identify over 60% of phishing URLs from both sources, but it still missed 25% of the APWG phishing URLs and 32% of the phishtank.com phishing URLs. Half the tools we tested could correctly identify less than half the phishing sites. Many of the tools 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 do not know the size of the blacklists used by each tool, nor do we know what heuristics are used by any of the tools other than SpoofGuard. We suspect that the tools that performed best use larger and more frequently updated blacklists. They may also use heuristics that allow them to detect phishing sites that have not yet been put on their blacklist.

The only tool 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.

Some of the tools that rely on blacklists send the URLs requested by a user to a central blacklist server, which may raise privacy concerns and could potentially impact browser performance (however, we did not observe perceptible performance impacts in our testing). We observed that Firefox performed poorly when the blacklist has not been downloaded recently. On the other hand, even after the blacklist was updated, when we configured Firefox to send every URL to Google it was able to identify an additional 6% of phishing sites.

5.2. Testing Methodology

Testing anti-phishing tools is a time consuming and difficult process. In order for results to be comparable, multiple tools 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 tools 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 tool 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 blacklists. In order to test the speed at which tools are able to identify phishing sites and add them to their blacklists we would need a fresher source of phishing URLs.

Ideally, all tools would be tested in parallel. However, this would require a separate computer for each tool to be tested. We did not have the resources to do this, so we ran multiple "worker" processes on each test computer and did some manual loading and unloading of tools.[5] As a result, there was a difference of as much as 1 hour between the testing of a particular URL on the first tool and the last tool. If we were to use a fresher source of phishing URLs and attempt to more precisely monitor the speed of blacklist updates, it would be important to test all tools simultaneously on separate computers. Close to simultaneous testing could also be achieved using virtual machines.

We conducted all of our tests using the tools' default configuration options, except for Firefox, which we also tested using the "Ask Google" option. It would also be interesting to test tools with multiple configuration options to observe the impact these options have on tool accuracy and false positives. However, each additional option requires an additional "worker" in the automated test bed.

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 ten popular anti-phishing tools, we have only anecdotal evidence of their usability.

Eight of the ten tools 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 tools 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, several tools use popup dialog boxes to warn when a site has been identified as fraudulent. While the IE7, eBay, Firefox, Netscape, and Netcraft dialog boxes block the phishing site unless the user overrides the block, 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 Firefox encounters a site that it has positively identified as phishing, it darkens the page to draw attention to a dialog box. IE7, Netscape and Cloudmark do not even display the page, instead they show a different page where the user is given the choice of displaying the suspected phishing site or closing the window. eBay's tool 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, SpoofGuard, and CallingID tools do not present the user with any indications beyond the

[5] Initially we had planned to use virtual machines to alleviate the need to load and unload tools and allow our testing of each tool to proceed almost simultaneously. However, this ended up being unworkable due to the limitations of our test bed computers and network.

red/green/yellow icons. User testing is needed to better understand how users react to each style of warning.

Based on our cursory review, all of the tools examined appear to have some usability problems. We believe that it is important for these problems to be resolved if these tools are to be effective. An anti-phishing tool could identify all fraudulent web sites without any false positives, but if it has usability problems, users might still fall victim to fraud.

Future anti-phishing tool studies should also include usability testing. A technically sound tool is of little use if users are unsure of what it is trying to communicate to them. Previous research has examined the effectiveness of several techniques for informing users about phishing [25]. However, it did not evaluate the effectiveness of pop-up warnings, or the difference in user reaction upon seeing a warning versus having a web site blocked.

Usability problems plague all varieties of software—security software in particular. For an anti-phishing tool, 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.

6. Acknowledgments

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. 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.

7. References

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

[2] CallingID, Ltd. Accessed: December 1, 2006. http://www.callingid.com/DesktopSolutions/CallingIDToolbar.aspx.

[3] 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),* San Diego, CA February, 2004. http://crypto.stanford.edu/SpoofGuard/webspoof.pdf.

[4] Cloudmark, Inc. Accessed: September 5, 2006. http://www.cloudmark.com/desktop/download/.

[5] 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/.

[6] The Coral Content Distribution Network. Accessed: June 13, 2006. http://www.coralcdn.org/.

[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). New York: ACM Press, 2006.

[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,* Pittsburgh, PA 12-14 July 2006.

[9] eBay, Inc. Using eBay Tool's Account Guard. Accessed: June 13, 2006. http://pages.eBay.com/help/confidence/account-guard.html.

[10] EarthLink, Inc. EarthLink Tool. Accessed: November 9, 2006. http://www.earthlink.net/software/free/tool/.

[11] GeoTrust, Inc. TrustWatch Tool. Accessed: June 13, 2006. http://tool.trustwatch.com/tour/v3ie/tool-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 Tool. Accessed: June 13, 2006. http://tool.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," *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 Tools Actually Prevent Phishing Attacks?" *in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* (Montreal, Quebec, Canada, April 22 - 28, 2006), 601-610. New York: ACM Press, 2006.