Chapter 4

Case Study of Browser-based Anti-phishing Solutions

This chapter is largely a reproduction of a paper co-authored with Lorrie Cranor, Brad Wardman (University of Alabama), Gary Warner, and Chengshan Zhang, and published at CEAS 2009 [128].

As discussed in Chapter 2, to reduce phishing damage, stakeholders have enacted their own countermeasures. Internet service providers, mail providers, browser vendors, registrars and law enforcement all play important roles. Due to the strategic position of the browser and the concentration of the browser market, web browser vendors play a key role. Web browsers are at a strategic position at which they can warn users directly and effectively. In addition, the browser market is fairly concentrated with two browsers (Internet Explorer and Firefox) accounting for 95% of the total market [101]. Solutions that these two browsers implement provide the majority of users with a defense against phishing. A recent laboratory study shows that when Firefox 2 presented phishing warnings, none of the users entered sensitive information into phishing websites [29]. This study also recommended changes to Internet Explorer's phishing warnings, and Microsoft has already acted on some of them to improve IE 8's warning mechanism.

For browsers to truly realize their potential to protect users, their warnings need to be accurate (low false positives) and timely. Currently, most browsers with integrated phishing protection or anti-phishing browser toolbars rely on blacklists of phish and, sometimes, heuristics to detect phishing websites. Perhaps because toolbar vendors are striving to avoid potential lawsuits from mislabeling websites, blacklists are favored over heuristics due to their low false positives.

In this chapter, we study the effectiveness of phishing blacklists. We used 191 fresh phish that were less than 30 minutes old to conduct two tests on eight phishing toolbars. We found that 63% of the phishing campaigns in our dataset lasted less than two hours. Blacklists were ineffective when protecting users initially, as most of them caught less than 20% of phish at hour zero. We also found that blacklists were updated at different speeds, and varied in coverage, as 47% - 83% of phish appeared on blacklists 12 hours from the initial test. We found that two tools using heuristics to complement blacklists caught significantly more phish initially than those using only blacklists. However, it took a long time for phish detected by heuristics to appear on blacklists. Finally, we tested the toolbars on a set of 13,458 legitimate URLs for false positives, and did not find any instance of mislabeling for either blacklists or heuristics.

To the best of our knowledge, this paper is the first attempt to quantitatively measure the length of phishing campaigns and the update speed and coverage of phishing blacklists. Based on these measurements, we discuss opportunities for defenders, and propose ways that phishing blacklists can be improved.

The remainder of the document is organized as follows: section 2 introduces the background and related work, section 3 discusses the test setup, section 4 presents our results, and section 5 discusses ways in which phishing blacklists and toolbars can be improved.

4.1 Background and Related Work

Efforts to detect and filter phish can be implemented at the phishing e-mail level and at the phishing website level. To prevent phishing emails from reaching potential victims, traditional spam-filter techniques such as bayesian filters, blacklists, and rule based rankings can be applied. Recently, some phishing-specific filters were developed as well [1, 34]. In addition to these efforts, some protocols have been proposed to verify the identities of email senders [24, 124]. Although these efforts are promising, many users remain unprotected. Filtering techniques, are imperfect and many phishing emails still arrive in users' inboxes. Thus, we need to make an effort to detect phishing websites as well.

Generally speaking, research to detect phish at the website level falls into two categories: heuristic approaches, which use HTML or content signatures to identify phish, and blacklist-based methods, which leverage human-verified phishing URLs to reduce false positives. Our research on blacklist measurement contributes to understanding the effectiveness of blacklists to filter phish at the website level.

4.1.1 Anti-Phishing Heuristics

Most of these heuristics for detecting phishing websites use HTML, website content, or URL signatures to identify phish. Machine learning algorithms are usually applied to build classification models over the heuristics to classify new webpages. For example, Garera et al. identified a set of fine-grained heuristics from phishing URLs alone [41]. Ludl et al. discovered a total of 18 properties based on the page structure of phishing webpages [75]. Zhang et al. proposed a content-based method using TF-IDF and six other heuristics to detect phish [148]. Pan et al. proposed a method to compile a list of phishing webpage features by extracting selected DOM properties of the webpage, such as the page title, meta description field, etc [107]. Finally, Xiang and Hong described a hybrid phish detection method with an identity-based detection component and a keyword-retrieval detection component [144]. These methods achieve true positive rates between 85% and 95%, and false positive rates between 0.43% and 12%.

The heuristics approach has pros and cons. Heuristics can detect attacks as soon as they are launched, without the need to wait for blacklists to be updated. However, attackers may be able to design their attacks to avoid heuristic detection. In addition, heuristic approaches may produce false positives, incorrectly labeling a legitimate site as phishing.

Several tools such as Internet Explorer 7 and Symantec's Norton 360 include heuristics in their phishing filters. Our research examines the accuracy of these heuristics in terms of their ability to detect phish and avoid false positives. In addition, we examine how anti-phishing tools use heuristics to complement their blacklists.

4.1.2 Phishing blacklists

Another method web browsers use to identify phish is to check URLs against a blacklist of known phish. Blacklist approaches have long been used in other areas.

Blacklists of known spammers have been one of the predominant spam filtering techniques. There are more than 20 widely used spam blacklists in use today. These blacklists may contain IP addresses or domains used by known spammers, IP addresses of open proxies and relays, country and ISP netblocks that send spam, RFC violators, and virus and exploit attackers [61].

Although a spam blacklist of known IP addresses or domain names can be used to block the delivery of phishing emails, it is generally inadequate to block a phishing website. One reason is that some phishing websites are hosted on hacked domains. It is therefore not possible to block the whole domain because of a single phish on that domain. So a blacklist of specific URLs is a better solution in the phishing scenario.

Compiling and distributing a blacklist is a multi-step process. First, a blacklist vendor enters into contracts with various data sources for suspicious phishing emails and URLs to be reviewed. These data sources may include emails that are gathered from spam traps or detected by spam filters, user reports (eg. Phishtank or APWG), or verified phish compiled by other parties such as takedown vendors or financial institutions. Depending on the quality of these sources, additional verification steps may be needed. Verification often relies on human reviewers. The reviewers can be a dedicated team of experts or volunteers, as in the case of Phishtank. To further reduce false positives, multiple reviewers may need to agree on a phish before it is added to the blacklist. For example, Phishtank requires votes from four users in order to classify a URL in question as a phish.

Once the phish is confirmed, it is added to the central blacklist. In some instances, the blacklist is downloaded to local computers. For example, in Firefox 3, blacklists of phish are downloaded to browsers every 30 minutes [122]. Doing so provides the advantage of reducing network queries, but performance may suffer between blacklist updates.

A number of these blacklists are used in integrated browser phishing protection [10, 46, 90], and in web browser toolbars [16, 17, 102]. Although blacklists have low false positive rates, they generally require human intervention and verification, which may be slow and prone to human

error. Yet this is the most commonly used method to block phish. Our research investigates the speed of blacklist updates and the accuracy of blacklists?

4.1.3 Related Work

Several authors have studied the effectiveness of phishing toolbars. In Nov 2006, Ludl et. al used 10,000 phishing URLs from Phishtank to test the effectiveness of the blacklists maintained by Google and Microsoft [75]. They found that the Google blacklist contained more than 90% of the live phishing URLs, while Internet Explorer contained only 67% of them. The authors concluded that blacklist-based solutions were "quite effective in protecting users against phishing attempts." One limitation of this study is that the freshness of the data feed was not reported. We overcome this weakness by using a fresh phish feed less than 30 minutes old and by using an automated testbed to visit phishing websites nine times in 48 hours to study the coverage and update speed of blacklists. We arrive at a different conclusion from this chapter.

In a related study, Zhang et al. [147] tested the effectiveness of 10 popular anti-phishing tools in November 2006 using data from Phishtank and APWG. Using 100 URLs from each source and 516 legitimate URLs to test for false positives, they found that only one tool was able to consistently identify more than 90% of phishing URLs correctly, but with false positive rates of 42%. Of the remaining tools, only one correctly identified over 60% of phishing URLs from both sources. This study had a similar weakness to the first study, and it also had a small sample of false positives URLs. We based our study on this setup, but made the following improvements. First, we used a source of fresh phish less than 30 minutes old. Second, we extend the methodology by separately analyzing phish caught by heuristics versus blacklists. Third, we tested phish nine times over 48 hours to study the coverage and update speed of blacklists; Finally, we used a much larger sample to test for false positives.

Other researchers have studied the effectiveness of spam blacklists [58, 61, 111]. For example, Ramachandran et al. measured the effectiveness of eight spam blacklists in real time by analyzing a 17-month trace of spam messages collected at a "spam trap" domain [111]. In their study, whenever a host spammed their domain, they examined whether that host IP was listed in a set of DNSBLs

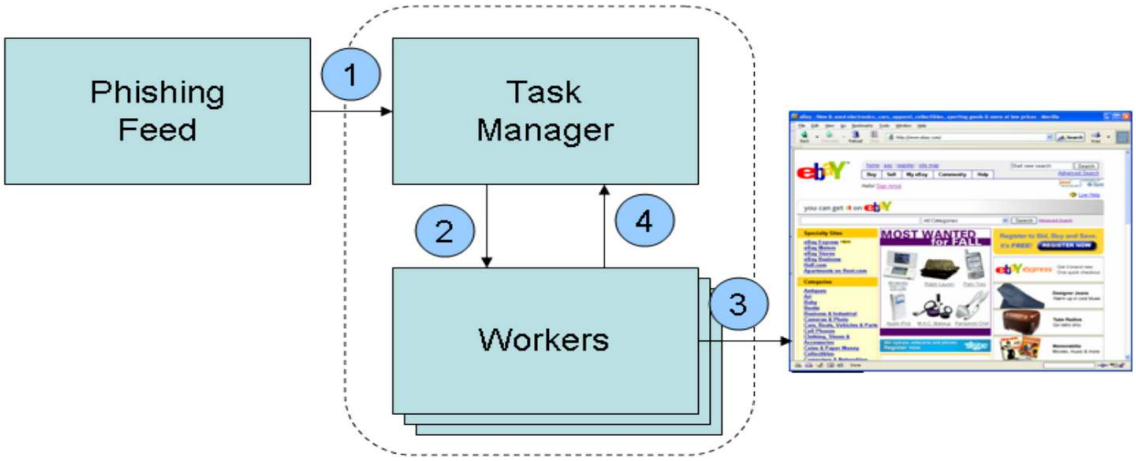

Figure 4.1 High-level system architecture for our anti-phishing evaluation test bed. The Task Manager (1) gets an updated list of URLs from a phishing feed, and then (2) sends that URL to a set of Workers. Each worker (3) retrieves a web page and checks whether the web page was labeled as a phishing 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.

in real time. They found that about 80% of the received spam was listed in at least one of eight blacklists, but even the most aggressive blacklist had a false negative rate of about 50%.

In addition to research work introduced above, a number of industry efforts were used to measure the effectiveness of phishing toolbars as well [59, 82, 88].

4.2 Methodology

In this section we describe our anti-phishing testbed, explain how we collected phishing URLs for testing, and describe our evaluation methodology.

4.2.1 Anti-phishing Testbed

We used the anti-phishing testbed developed by Yue et al. [148]. The testbed has a client-and-server architecture. It includes a task manager and set of workers, each of which is responsible for evaluating a single tool. During the test, the task manager first retrieved a list of potential phishing sites to test against. The task manager then sent each URL to a set of workers, each of which

was running a separate tool. To reduce the number of machines needed, we ran each worker on a virtual machine. Each worker downloaded the specified web page, examined whether its tool had labeled the web page as phishing or not using a simple image-based comparison algorithm, and returned that value back to the task manager. The image-based comparison algorithm works as follows: 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. We capture screenshots of the tools and compare relevant portions of those images to screenshots of the tools in each of their known states. The task manager aggregated all of the results from the workers and tallied overall statistics, including true positives, true negatives, false positives, false negatives, and sites that no longer exist.

4.2.2 Phishing Feed

We obtained the phishing URLs for this study from the University of Alabama (UAB) Phishing Team's data repository. UAB has relationships with several sources who share their spam as part of the UAB Spam Data Mine. One of the largest sources is a spam-filtering company that provides services ranging from small business to the Fortune 500 companies located in more than 80 countries. This company reviews well over one billion emails each day and uses a combination of keyword searching and proprietary heuristics to identify potential phish. They then extract the URLs from these emails and send these URLs to UAB in batches every four minutes.

UAB manually tested the URLs they received from the spam-filtering company to determine if they were phishing URLs. If a URL was a phish and had not been reported to UAB before, it was put on a list to be tested by the testbed. UAB sent this list to the testbed every 20 minutes.[1] The testbed began testing each batch of URLs within 10 minutes of receipt.

Because UAB received phish URLs every four minutes, they were able to label each URL with the four-minute time segments in which it was seen. Thus they could identify the first segment in which a URL was seen and identify subsequent time segments in which the same URL was

[1] Sometimes randomization was introduced to URLs to attempt to defeat exact matching. We do not consider two URLs as unique if their difference is only in the attribute portion of the URLs.

Table 4.1 The top 10 brands that appear in our data set. Total phish: 191

Institutions Victimized	# of phish	Percentage
Abbey	47	24.9%
Paypal	21	11.1%
Lloyds TSB	17	9.0%
Bank of America	14	7.4%
Halifax	13	6.9%
Capital One	11	5.8%
New Egg Bank	11	5.8%
HSBC	7	3.7%
eBay	6	3.2%
Wachovia	6	3.2%
Wellsfargo	6	3.2%

reported. This approach to recording phishing URLs allows us to determine the length of each spam campaign — the time period over which phishers send out emails with the same phishing URL. If the spam campaign lasts for only one day, the effectiveness of anti-phishing tools on subsequent days is not as important as effectiveness on day one. While some users will read phishing emails days after the initial email send time, most users will read phishing emails within a few hours. Thus the most critical time to protect is when emails are still being actively sent by the spammer.

We collected and tested a total of 191 verified phishing URLs during this study. Table 4.1 lists the top 10 brands that appear in our data set.

4.2.3 Evaluation Procedure

Tools tested: We tested eight anti-phishing toolbars that use various blacklists and heuristics. They are Microsoft Internet Explorer version 7 (7.0.5730.11), version 8 (8.0.6001. 18241), Firefox 2 (2.1.0.16), Mozilla Firefox 3 (3.0.1), Google Chrome (0.2.149.30), Netcraft toolbar (1.8.0), McAfee Siteadvisor (2.8.255 free version), and Symantec Norton 360 (13.3.5). Except for Internet Explorer 7 and Symantec, all of these tools use blacklists only. Those two toolbars that use heuristics to complement their blacklists trigger different warnings when a phish is detected by heuristics

versus blacklist. We configured all tools with their default settings, except for Firefox 2, in which case we used the "Ask Google" option to query the central blacklist server every time instead of downloading phishing blacklists every 30 minutes.[2]

Testbed setup: We configured 4 PCs running Intel Core 2 CPU 4300 @ 1.80 GHz. Each PC ran two instances of VMware, each configured with a 720MB RAM and 8GB hard drive. For each toolbar, we ran the task manager and workers on the same machine to avoid network latency. Since some of the toolbars use local blacklists, we left every browser open for six to eight hours before each test to download blacklists, and we left the browser open for 10 minutes between every run during the test. We chose the eight-hour period because the necessary blacklists would download reliably within this time. Thus we are investigating the best case scenario for blacklist effectiveness.

Test period: We ran the test for two to three hours on October 2, 8, and 9, 2008 and on December 3, 4, 5, and 15, 2008. During this time, batches of new unique phish were sent to the testbed every 20 minutes. The testbed began testing them 10 minutes after receiving the phish, leaving a total lapse time of approximately 30 minutes. Each worker opened up the desired browser with toolbars for 30 seconds before taking the screenshot. For each URL, we tested the toolbars' performance at hour 0, 1, 2, 3, 4, 5 12, 24 and 48. We cleared the browser cache every hour. We collected and tested 90 URLs in October and 101 URLs in December.

Post verification: After the data was compiled, we manually reviewed every website that toolbars labeled as legitimate. This step was necessary because some host companies did not issue 404 errors when taking down a phish. Instead, they replaced it with their front page. In this case, the toolbar will mark the website as legitimate, but in fact it was the phishing website being taken down.

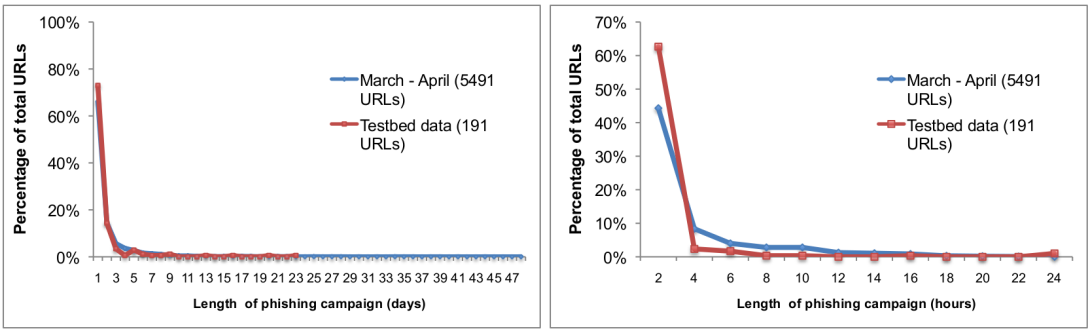

Figure 4.2 Length of phishing campaign, measured as the time between the first and last appearance of the phish in our source report. The graph on the left shows length of phishing campaigns in days. The graph on the right shows length of phishing campaigns in hours for those campaigns that last one day or less.

4.3 Results

4.3.1 Length of Phishing Campaign

We define the length of a phishing campaign (LPC) as the time lapse between the first time a phish appeared in our source report and the last time that phish appeared in our source report. As mentioned in Section 3.2, we received reports from our source every 4 minutes.

Of the 191 phish we used to test phishing blacklists, 127 of them, 66%, had an LPC less than 24 hours, indicating that their corresponding phishing campaign lasted less than 24 hours. A total of 25 URLs had an LPC between 24 and 48 hours, and the remaining URLs had an LPC between 3 and 23 days. Examining the first day's data more closely, we found that 109 URLs were spammed only in a two-hour period, accounting for 63% of the URLs in this dataset.

To validate our finding, we calculated the LPC for 5491 phish provided by the same source and verified by UAB from February 17 through April 13, 2009. Similar to our testbed dataset result, we found that 66% of these phish had an LPC less than 24 hours, 14.5% had an LPC between 24 and 48 hours, and the remaining 19% of URLs had an LPC between 3 and 47 days. We found that 44% of the URLs had an LPC less than two hours. Figure 5.4 shows the LPC combined LPC results for our two datasets.

[2]This feature is no longer available for versions after Firefox 2 update 19.

Table 4.2 Website takedown rate vs. length of phishing campaign (LPC). LPC is measured as the time between the first and last appearance of the phish in our source report. Website takedown rate at each hour is measured by the number of phish taken down at that hour divided by total phish.

Hours	% of website taken down	% Phishing Campaign finished
0	2.1%	0%
2	7.9%	63%
4	17.8%	67%
5	19.9%	70%
12	33.0%	72%
24	57.6%	75%
48	72.3%	90%

It is important to note that the LPC does not necessarily correspond to the time a phishing site is live. In fact, we found that compared to the length of a phishing campaign, the time to take websites down is generally much slower. By hour 2, 63% of phishing campaigns in our dataset were finished, but only 7.9% of those phish were taken down. As shown in Table 4.2, on average, 33% of the websites were taken down within 12 hours, around half were taken down after 24 hours, and 27.7% were still alive after 48 hours.

Our LPC findings demonstrate the freshness of our data and show that current takedown efforts lag behind phishing campaigns. In the test conducted by Ludl et al., 64% of the phish were already down when they conducted their test [75], whereas in our sample, only 2.1% of phish were aleady down in our initial test.

4.3.2 Blacklist Coverage

In this section, we present the results of two tests performed in October and December of 2008 (Figures 4.3 and 4.4). We found that blacklists were ineffective when protecting users initially, as most of them caught less than 20% of phish at hour zero. We also found that blacklists were updated at different speeds, and varied in coverage, as 47% to 83% of phish appeared on blacklists 12 hours from the initial test in October.

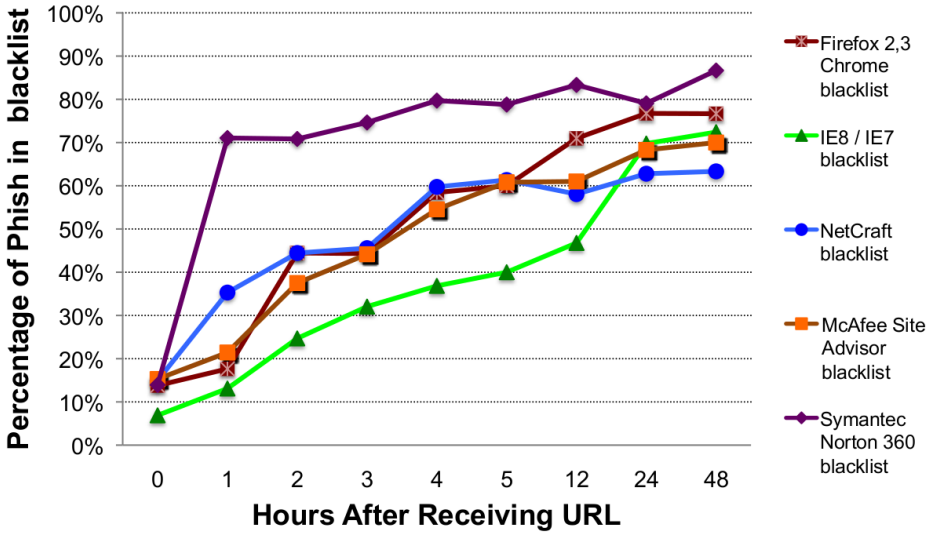

Figure 4.3 Percentage of phish caught by various blacklists in October 2008 data. This percentage is defined as the total number of phish on the blacklist divided by the total phish that were alive. URLs that were taken down at each hour were excluded in the calculation. Total phish at hour 0 was 90.

At any given hour, we define the coverage of the blacklist as:

$$\frac{No.\ of\ phish\ appearing\ on\ blacklist}{Total\ phish - phish\ that\ were\ taken\ down}$$

We found that coverage rates of some of the blacklists were highly correlated. Firefox 2, 3 and Google Chrome appear to use the same blacklists. Internet Explorer 7 and 8 also share a blacklist. In our analysis, we combined the results for those tools that use the same blacklists.

In our October test, all of the blacklists contained less than 20% of the phish initially. New phish appeared on the blacklists every hour, suggesting that the blacklists were updated at least once every hour.

One notable improvement is the Symantec blacklist. In hour 0, their blacklist caught as much phish as the others, but in hour 1 it caught 73% of the phish, 2 to 3 times more than the rest of the toolbars. This difference is also statistically significant until 12 hours from the initial test.[3] One possible explanation is that Symantec uses results from their heuristics to facilitate rapid blacklist updates [5].

[3] ANOVA, $p < 0.05$

We observed that the coverage of the Firefox and Netcraft blacklist is consistently highly correlated. Five hours after our initial test in October, 91% of the URLs that appeared in the Netcraft blacklist also appeared in the Firefox blacklist, and 95% of the URLs that appeared in the Firefox blacklist also appeared in Netcraft. The two blacklists are consistently highly correlated every hour except for our initial test in December. This suggests that the two blacklists have overlap in some of their data sources or have data sources with similar characteristics. Others were less correlated, phish on Internet Explorer only appear 45% of time on Firefox blacklist and 73% vice versa, suggesting they use different feeds with not much overlap.

We found that the Firefox blacklist was more comprehensive than the IE blacklist up to the first 5 hours, and the Symantec blacklists performed significantly better than the rest of the toolbars from hour 2 to 12. After 12 hours, the differences were no longer statistically significant. Figure 4.3 shows this result in detail.

In our December dataset, we observed similar trends in terms of coverage for some toolbars. However, Firefox and Netcraft performed much better here than in October. The Firefox blacklist contained 40% of phish initially and by hour 2, 97% of phish were already on the blacklist. One reason for this difference could be that during this period, the two tools acquired new sources that were similar to our feed. Finally we did not observe statistically significant improvement in other toolbars.

Finally, we examined phish that the IE 8 blacklist and Firefox blacklist missed five hours after our initial test in October. We observed that at hour 5 the IE 8 blacklist missed 74 phish, of which 73% targeted foreign financial institutions. The Firefox blacklist missed 28 phish, of which 64% targeted foreign financial institutions. However, given our limited sample size, we did not observe a statistically significant difference in the speed at which phish targeting US institutions and foreign institutions were added to the blacklist. There were some notable differences between the phish missed by the IE8 blacklist and Firefox. For example, IE8 missed 21 Abbey Bank phish while Firefox missed only 4 Abbey Bank phish.

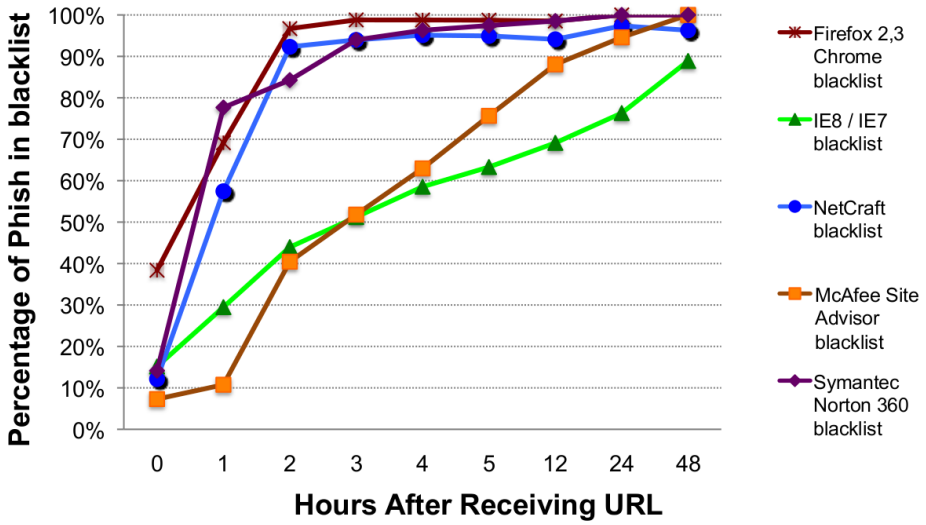

Figure 4.4 Percentage of phish caught by various blacklists in December 2008 data. This percentage is defined as the total number of phish on the blacklist divided by the total phish that were alive. URLs that were taken down at each hour were excluded in the calculation. Total phish at hour 0 was 101.

4.3.3 False Positives

We compiled a list of 13,458 legitimate URLs to test for false positives. The URLs were compiled from three sources, detailed below.

A total of 2,464 URLs were compiled by selecting the login pages of sites using google's inurl function. Specifically, we used Google to search for pages where one of the following login-related strings appears in the URL: login, logon, signin, signon, login.asp. A script was used to visit each URL to determine if it was running and also whether it included a submission form. These pages were selected to see whether tools can distinguish phishing sites from the legitimate sites they commonly spoof. Ludl et al. also used this technique to gather their samples [147].

A total of 994 URLs were compiled by extracting 1000 emails reported to APWG on August 20, 2008. Out of the 1000 emails we scanned, we removed URLs that were down at the time of testing or URLs used in spam campaigns through a spam URL blacklist service `uribl.com`. This left us with 1076 URLs, which comprised a host of phish, malware, some spam and legitimate sites. We manually checked each of these URLs and removed phishing URLs, leaving 994 verified

non-phishing URLs. We ran the test for false positives within 24 hours after retrieval. The list was selected because it represented a source of phishing feeds that many blacklist vendors use, and thus we would expect to have more false positives than other sources. While spam messages may be unwanted by users, the URLs in these messages should not be classified as phishing URLs.

Similarly, we compiled 10,000 URLs by extracting non-phishing URLS from the list of spam, phish, and malware URLS sent to UAB's spam data mine on December 1-15, 2008. We tested these URLs within one week of retrieval. Again, this represents a source of phishing feeds that blacklist vendors would likely receive, and thus we would expect this source to have more false positives than other sources.

We did not find a single instance of mislabeling legitimate login sites with phish. Among the 1,012 URLs from APWG, there was one instance where a malware website was labeled as a phish by the Firefox blacklist. Finally we did not find any false positives in the 10,000 URLs from the UAB spam data mine.

Compared with previous studies [147], our study tested an order of magnitude more legitimate URLs for false positives, yet our findings on false positives are the same: phishing blacklists have close to zero false positives.

Our results differ from a 2007 HP research study [88] in which the author obtained the Google blacklist and checked each entry to see if it was a false positive. This study reports that the Google blacklist contains 2.62% false positives. However, the methodology for verifying false positives is not fully explained and the list of false positives is not included in the report. In our test of false positives, we manually verified each URL labelled as phish and double-checked it with one of the known repositories of phish on the Internet.

It is also possible that Google changed their techniques or sources for phishing URLs since 2007. For future work, we would like to verify the Google blacklist using the same method used in the HP study [88]. However, Google's blacklist is no longer publicly available.

Table 4.3 Accuracy and false positives of heuristics

	Detected by blacklist at hour 0	Detected by heuristics	false positives
IE7 - Oct 08	23%	41%	0.00%
Symantec - Oct 08	21%	73 %	0.00%
IE7 - Dec 08	15%	25%	0.00%
Symantec - Dec 08	14%	80%	0.00%

4.3.4 Accuracy of Heuristics

Heuristics are used in Symantec's Norton 360 toolbar and Internet Explorer 7. In this section, we report on their performance.

We found that tools that use heuristics were able to detect significantly more phish than those that use only blacklists. At hour 0, Symantec's heuristics detected 70% of phish, while Internet explorer 7's heuristics caught 41% of phish. This is two to three times the amount of phish caught by the blacklists in that period. Furthermore, the heuristics triggered no false positives for the 13,458 URLs we tested. Table 4.3 summarizes these results.

We also found that IE 7 and Symantec use heuristics somewhat differently. Both tools display a transient and less severe warning for possible phish detected by heuristics. However, Symantec's toolbar introduced a feedback loop. When a user visits a possible phish which is detected by heuristics and is not on the blacklist then the URL is sent to Symantec for human review [5]. In our test, 95% of the phish detected by Symantec heuristics appeared on the Symantec blacklist at hour 1, while none of the phish detected by IE7 heuristics appeared on the IE blacklist at hour 1.

This feedback loop is important at the user interface level. If a phish is detected by heuristics, toolbars display less severe, passive warnings to avoid potential liability. However, once the phish is verified as a phishing site by human, toolbars can block the content of the web page completely (active warnings). A recent laboratory study [29] showed that users only heed active phishing warnings and ignore passive warnings.

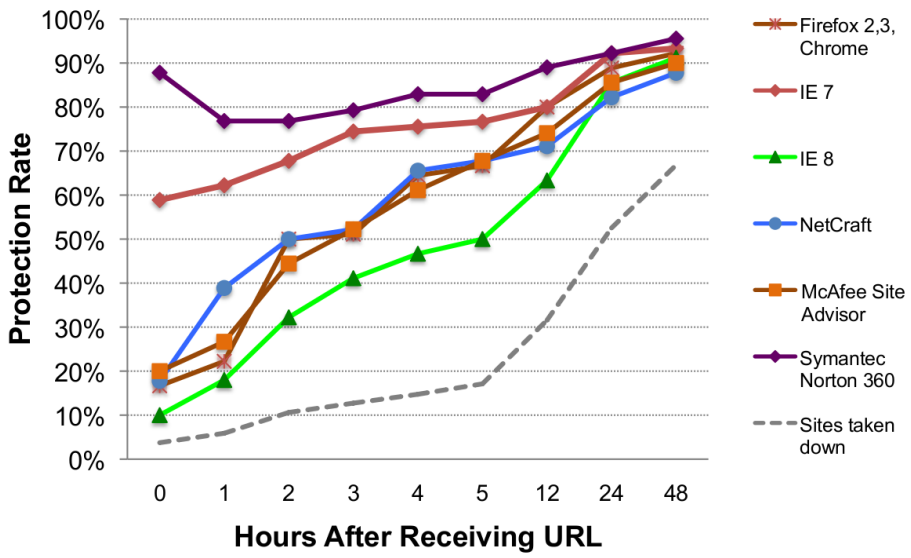

Figure 4.5 Protection rate for the October run of 91 phishing URLs. Protection rate is defined as total number of phish caught by blacklist or heuristic plus phish taken down divided by the total number of phish.

4.3.5 Total Protection

Finally, we consider protection offered to users by phishing toolbars. We define protection rate as:

$$\frac{phish\ on\ blacklist\ +\ detected\ by\ heuristics +\ taken\ down}{Total\ phish}$$

Figures 4.5 and 4.6 present our findings. We found that at hour 0, tools that use heuristics to complement blacklists offered much better protection than tools that use only blacklists. By hour 48 a large fraction of phishing sites are taken down, and the tools we tested detected most of thelive phishing sites. In the December test we found that by hour 48 most tools offered near-perfect protection.

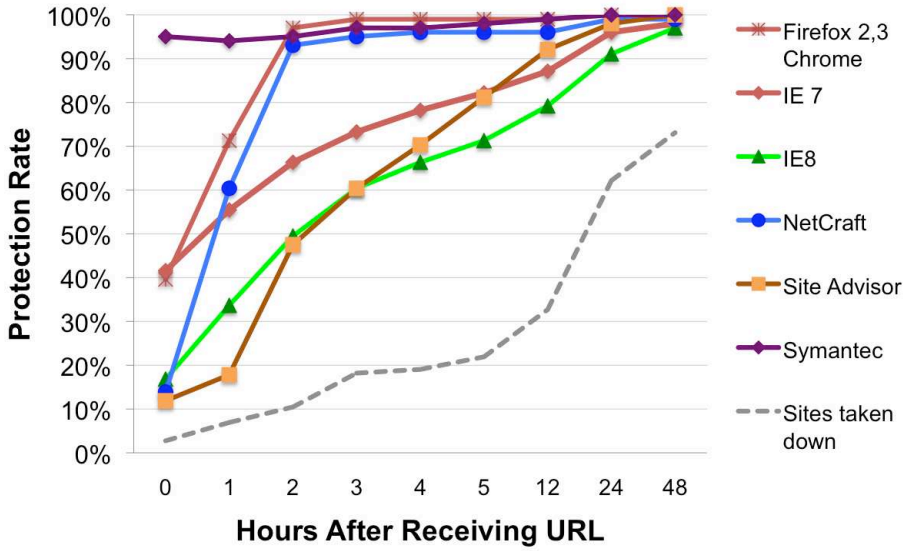

Figure 4.6 Protection Rate for the December run of 101 phishing URLs. Protection rate is defined as total number of phish caught by blacklist or heuristic plus phish taken down divided by the total number of phish.

4.4 Discussion

4.4.1 Limitations

There are a few limitations to our study. First, all of our URLs came from a single anti-spam vendor, therefore the URLs received may not be representative of all phish. Second, all the URLs were detected by a spam vendor and presumably never reached users protected by that vendor. However, as not all users are protected by commercial spam filters, it is important that browsers also detect these phishing URLs. Second, these URLs were extracted only from email and did not include other attack vectors such as Internet messenger phishing.

4.4.2 Opportunities for Defenders

The window of opportunity for defenders can be defined as the length of the phishing campaign plus the time lapse between the time a user receives a phishing email and the time the user opens the email. Users are protected if they either do not receive any phish or if, by the time they click on a phish, the website is blocked by browsers or taken down.

As shown in Section 4.1, 44% of phishing campaigns lasted less than 2 hours. Recent research shows that, for a non-negligible portion of the Internet population, the time between when a user receives and opens a phishing email is less than two hours. For example, Kumaraguru et al. sent simulated phishing emails to students and staff at a U.S. University and educated them once they clicked on the link in the email. They found that 2 hours after the phishing emails were sent, at least half the people who would eventually click on the phishing link had already done so; after 8 hours, nearly everyone (90%) who would click had already done so [66]. Their study also found that people with technical skills were equally likely to fall for phish than their non-technical counterparts. In a recent national survey, AOL asked 4,000 email users aged 13 and older about their email usage. The survey found that 20% of respondents check their email more than 10 times a day, and 51% check their email four or more times a day (up from 45% in 2007) [9]. Assuming that those who check their emails do so at a uniform rate, 20% of people check their emails once

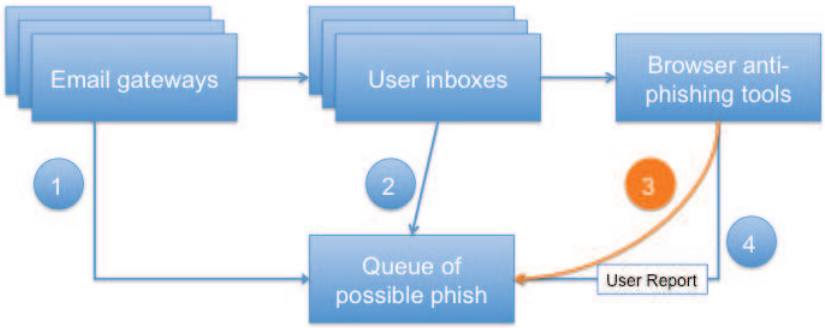

Figure 4.7 High-level view of sources of URLs for phishing blacklists. Potential phishing URLs can be collected from (1) URLs extracted from spam and phishing filters at mail exchange gateways, (2) URLs extracted from user reports of phishing email, (3) phishing websites identified by heuristics, and finally (4) user reports of phishing websites.

every hour and half, and 51% check their email once every four hours[4]. These findings suggest that the critical window of opportunity for defense is between the start of a phishing campaign and 2 to 4 hours later.

Our findings have several implications for phishing countermeasures. First, anti-phishing efforts should be more focussed on upstream protections such as blocking phish at the email gateway level. At the browser level, this effort should be focused on updating the blacklist more quickly or making better use of heuristic detection. Secondly, more research and industry development efforts to effectively educate users (eg. [68, 127]) and to design trusted user interfaces (eg. [22, 116, 145, 146]) are needed to overcome the initial limited blacklist coverage problem.

[4] Assuming eight hour sleep time.

4.4.3 Improving blacklists

The first step to improving blacklists is earlier detection of more phishing URLs. As shown in Figure4.7, potential phishing URLs can be gathered from URLs extracted from spam and phishing filters at e-mail gateways, URLs extracted from users' reports of phishing emails or websites, and phishing websites identified by toolbar heuristics (Figure4.7). Each of these sources have different coverage. We first discuss ways to improve each source.

E-mail gateway filters are the first point of contact with phishing emails. Given the limited window of opportunity for defenders, as discussed in section 4.1, vendors should focus their gathering efforts here. However, regular spam filters are not sufficient as they contain a lot of spam that would require much human effort to filter. To improve detection of phish at this level, we recommend using spam filters as the first line of defense, and then applying heuristics developed to detect phishing websites as a second layer. Once a suspicious URL is marked by both sources, it should be submitted for human review. As residential email accounts and business email accounts receive a different distribution of emails, to get the widest coverage vendors should collect URLs from a variety of sources.

User reports of phishing emails and websites are likely to contain phish that spam filters missed. Therefore user reports should be use to complement email gateway spam filter data. However, users may lack incentives to report and verify phish. User incentives (e.g. points, prizes) may help overcome this problem.

Finally, we recommend browser anti-phishing tools use heuristics to improve their blacklists. This method is analogous to early warning systems for disease outbreaks. When a user visits a possible phish that is detected by heuristics and is not on the blacklist, the tool can send the URL for human review and adds the URL to the blacklist once verified. This system would be likely to succeed based on the fact that some users check their email much more frequent than others [9].

4.4.4 Use of heuristics

As shown in Section 4.4 and 4.5, the two tools using heuristics to complement blacklists caught significantly more phish initially than those using only blacklists. Given the short length of phishing campaigns, there is great value in using heuristics. However, vendors may be concerned about the greater possibility of false positives when using heuristics and potential liability for mislabeling websites.

In a court case in 2005, Associated Bank-Corp sued Earthlink after the Earthlink anti-phishing software ScamBlocker blocked the bank's legitimate page [11]. Earthlink was able to fend off the suit on the basis that it was using a blacklist of phish provided by a third party, thus it cannot be held liable as a publisher when that information is erroneous under a provision in the Communication Decency Act. However, if a toolbar uses heuristics to detect and block a phish that turns out to be a false positive, the toolbar vendor may be regarded as "a publisher" under CDA, and thus not immunized.

In our testing, we did not detect any false positives triggered by either the blacklists or heuristics. However, it is the potential of false positives that worries vendors. To overcome this liability issue, we recommend vendors first use heuristics to detect phish and then have experts verify them. We also encourage more discussion about the liability associated with providing phishing blacklists and heuristics. So far, there has been no test case on this matter. Lack of clarity on these matters could further reduce vendors' incentives to apply heuristics. Major vendors such as Microsoft or Firefox, which offer protection to the majority of users, do not lose money directly from phishing. However, if they implement heuristics and get sued, they could potentially lose millions of dollars in restitution and legal fees.