PhishZoo: An Automated Web Phishing Detection Approach Based on Profiling and Fuzzy Matching

Sadia Afroz and Rachel Greenstadt
Drexel University
{sa499,greenie}@cs.drexel.edu

ABSTRACT

Phishing is a web-based attack that uses social engineering techniques to exploit Internet users and acquire sensitive data. Most phishing attacks work by creating a fake version of the real site's web interface to gain the user's trust. Despite the fact that these phishing sites look identical or nearly identical to the real sites they imitate, user studies have shown that users ignore browser-based indicators and often use the appearance of a site to judge the authenticity of sites, just as they use the appearance of physical sites to judge their authenticity. This paper proposes a phishing detection approach—PhishZoo—that uses profiles of trusted websites' appearances built with fuzzy hashing techniques to detect phishing. We evaluate our approach on over 600 phishing sites imitating 20 real sites and show that it provides similar accuracy to blacklisting approaches, with the advantage that it can classify new attacks and targeted attacks against smaller sites (such as corporate intranets). PhishZoo has the potential to have a beneficial impact on the phishing "arms race" by reducing the effectiveness of sites that look too much like the real sites and thus giving users a chance to detect sites that "look phishy."

1. INTRODUCTION

Phishing attacks have deceived many users by imitating websites and stealing personal information and/or financial data. According to the Anti-Phishing Working Group (APWG), there are at least 47,324 phishing attacks and a top-ten American bank estimates that at least US$300 is lost for every hour that a phishing site remains up [6]. The main reason for the success of phishing attacks is the failure of human users to detect phishing sites. This paper proposes a phishing detection approach—PhishZoo—that uses profiles of trusted website's appearance built with fuzzy hashing techniques to detect phishing. While similar ideas have been proposed [29, 30], they have not been explored in detail nor rigourously evaluated. A key contribution of this work is the empirical evaluation of this content-matching approach. We show where this type of approach succeeds (and fails) and, in the process, illuminate current trends in phishing attacks.

Why and how phishing works is an interesting question that has been asked by many researchers [19]. Phishing is a kind of social engineering attack that plays with users' psychology to deceive them into revealing their private information. The success of phishing attacks often depends on users' understanding of and knowledge about the Internet. Analyses revealed that over 90% users depend on a website's appearance as an indication of its authenticity [19, 22, 23] and fall for malicious, but well-designed phishing sites that look almost (or exactly) like legitimate sites. Currently used phishing detection tools and browsers give various indications of a site's authenticity and raise flags about questionable materials, however, these flags are often ignored or misunderstood by the users [20, 21]. This human factor has complicated phishing attack prevention. Blacklisting approaches cannot detect new phishing sites right away and most phishing sites are too short-lived (lasting a few hours to a few days [32]) to update and verify the database. Furthermore, these databases cannot provide protection against targeted attacks (for example, against corporate intranets).

We propose that effective phishing detection mechanisms must detect phishing sites from the user's point of view. That is, the detection should be directly related to the look and feel of the site. The majority of users provide sensitive credentials to a small set of sites (fewer than 20). Throughout this paper, we make the assumption that SSL is supported by these sites of interest and secure in both the underlying protocol and the trust model used by the browser. We hope to relax this assumption in future work. Therefore, as these sites support SSL, they can be whitelisted and browsers can automatically verify their authenticity. The problem with whitelisting approaches [7, 14, 25], is that the user must know about and remember to check the interface every time they visit the site, and there is ample evidence that this is beyond most users' capacities. However, warning users when the site they are visiting is not among their sensitive subset is also futile, as the vast majority of sites visited by users are not sensitive and such warnings will be quickly tuned out or turned off. What is needed is for the browser to infer the user's *false belief* that she is visiting one of her sensitive sites and only warn (actively and emphatically) in this case. Our hypothesis is that similar-looking content can be detected by automated methods.

This paper empirically evaluates this hypothesis by pre-

senting a new approach of web phishing detection based on profiles of sensitive sites' appearance and content. Our method—PhishZoo—makes profiles of site that consist of fuzzy hashes of several common content elements (e.g. URL, images, most used texts, HTML codes, script files, etc.), which are related to its structure and appearance. These Profiles are stored in a local database and are matched against all sites at the time of loading. We also make and test against profiles of common phishing pages to increase accuracy.

We evaluated PhishZoo using 636 phishing sites from `www.phishtank.com` and 20 profiles of legitimate sites. Performing these tests and refining our profiling methods illuminated current trends in phishing attacks which are also described. Key findings include:

1. If only HTML code is used in the content profile of a site, our method can detect 49% phishing sites. That means 49% phishing attacks directly copy the real site's html code to make fake sites.

2. When images, scripts, icon files are used along with html code, PhishZoo can detect 66.82% phishing sites using the same weight for each component. But 95.33% phishing sites can be detected if only look alike sites are used.

3. This method only depends on websites' contents to detect corresponding phishing sites. That is why it can detect new phishing sites which are not yet blacklisted.

4. There are many phishing sites which represent legitimate sites but do not look like them. These sites use logos and font colors of real sites to represent them. We discuss ways of detecting these sites, the most effective of which was including a few profiles of previously blacklisted phishing sites. Including these sites brings PhishZoo's accuracy up to 97%.

In section 2, we briefly survey anti-phishing approaches. Section 3 describes the profiling mechanisms used by PhishZoo in detail. Our empirical evaluation techniques and experimental results are discussed in Section 4. Our results show that PhishZoo is effective at detecting phishing attacks with high accuracy (97%). While we believe that attackers will be able to adapt to and defeat PhishZoo's current mechanisms, the overall approach followed leads to an arms race where defenders have the advantage. The shape of this arms race is discussed in Section 5. We conclude by outlining future directions for this line of research and for PhishZoo.

2. RELATED WORKS AND NOVELTY

Most current phishing site detection approaches fall into three categories: (1) Heuristic approaches that use various features to classify unknown sites as authentic or phishing, (2) Blacklisting approaches that collect databases of bad websites and warn users when they visit these sites, and (3) Whitelisting approaches that identify known good sites. These approaches are each discussed, then contrasted with our approach.

Other anti-phishing approaches include detecting phishing emails [31] (rather than sites) and educating users about phishing attacks and human detection methods [24].

2.1 Heuristic approach:

In this approach, researchers try to understand the anatomy of phished web sites and detect attacks based on several features. Features used in this approach include url, domain name, age of domain, spelling error, source of the images, links, etc. For example, SpoofGuard [14] first checks the current domain name, then the full URL is analyzed to detect obfuscation as well as non-standard port numbers, then Spoofguard analyzes the contents, making note of any password fields, embedded links, and images. In CANTINA [2], the likelihood of a phishing site is calculated from eight heuristics as follows: Age of Domain, Known Images, Suspicious URL, Suspicious Links: checking whether or not a link on the page contains an "at" symbol or a dash, IP Address, Dots in URL, Forms, and TF-IDF textual analysis. In most cases, the results of each of the heuristics are combined and the tool declares the site to be a phishing attack if the result is above a certain threshold. Many heuristic approaches have high false positive rates and can be outwitted by the phishers.

Search engines such as Google that sort search results based on page rank or reputation of sites can be used as heuristics for detect phishing sites. In CANTINA [2], TF-IDF is used to determine the most common terms on the phishing site and they are fed into a search engine (for instance, google). If URL of the site is within five or ten URLs of the result, then it is considered as good site. As the average time that a phishing site stays online was 4.5 days in 2006 [26] and has since been reduced to an average lifetime of 49.2 hours and 0 hour median in 2008 [32], it is likely to be a very low ranked page and thus is not placed within first five search results. AdaBoost-Based detection [3] uses training sets to determine weights for the heuristics used in CANTINA and combines them using AdaBoost algorithm.

2.2 Blacklisting approach:

In this approach, users report or companies seek and detect phishing sites which are stored in a database. Manual verifiers check these sites and update the database which is used by anti-phishing softwares to detect similar phishing sites. Most commercial toolbars Netcraft [5], Internet explorer 7 [12], CallingID Toolbar [4],EarthLink Toolbar [9], Cloudmark Anti-Fraud Toolbar [8], GeoTrust TrustWatch Toolbar [11], Netscape Browser 8.1 [13] use this approach. But as most phishing sites are too short-lived (lasting hours) to update and verify the database, the blacklisting approach fails to detect most phishing attacks. Furthermore, a blacklisting approach will fail to detect an attack that is targeted to a particular user ("spearphishing"), particularly those that target lucrative but not widely used sites such as company intranets, small brokerages, etc.

2.3 Whitelisting approach:

Whitelisting approaches seek to detect known good sites [7, 14, 25], but the user must remember to check the interface every time they visit the site.

In the YURL proposal, a user can assign a "petname" to any site and the user's browser maintains a mapping of a public key hash to petname. When a user visits a page identified by a YURL, the browser displays the petname that the user previously associated with the public key hash [25]. TrustBar [7] allows user to assign logo to specific sites which are shown in the toolbar when the page is loaded. SpoofGuard [14] 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.

Some whitelisting approaches use server side validation to add additional authentication metrics (beyond SSL) to client browsers as a proof of its benign nature. In Dynamic security skins [27],the remote server generates an abstract unique image for each user and each transaction which is used to create a "skin," which customizes the appearance of the server's web page. The browser computes the image that it expects to receive from the server and displays it in the user's trusted window. To authenticate content from the server, the user can visually verify that the images match. TrustBar [7] verifies server using its SSL information and displays SSL information in a simple, concise way, e.g.: 'Gmail identified by VeriSign'. The first identifier (e.g. Gmail) is the name, logo or domain-name of the site; the second identifier is of the entity that actually authenticated it (e.g. VeriSign, which is currently the largest Certificate Authority). In SRD ("Synchronized Random Dynamic Boundaries") a random number generator is used to set a bit that determines whether the browser border is inset or outset. The browser border alternates between inset and outset at a certain frequency in concert with a reference window [28].

2.4 Novelty of PhishZoo

Our approach combines the ability of whitelisting approaches to detect new or targeted phishing attacks with the ability of blacklisting and heuristic approaches to warn users about bad sites. The PhishZoo approach can be combined with other blacklisting, heuristic, or whitelisting approaches to improve accuracy. The importance of a site's appearance in proving its legitimacy has been repeatedly demonstrated [19, 22, 23]. Zdziarski, Yang, and Judge present a fingerprinting approach that uses a series of (exact) hashes to profile websites and identify phishing sites [29]. Aburrous et al. propose a layered approach using fuzzy logic [30]. Neither paper tests their approach against known phishing and genuine sites. PhishZoo is the first project to empirically evaluate the feasibility of using such profiles of site contents and appearance against phishing attacks.

PhishZoo can detect current phishing sites if they look like authentic sites by matching the profile. In order to avoid detection, a phishing site must look significantly different from a real site. Our working assumption is that such different-looking sites have a better chance of catching users' attention about their phishiness. Branding is a problem that is well-studied in the marketing literature, and, with PhishZoo, can be used to provide security as opposed to the current case, when this branding is co-opted by attackers to abuse users' trust.

3. APPROACH

The goal of our work is to use profiles of sensitives sites' appearance and content to detect the *false belief* that phishing sites provoke in their users. This detection could then be used by a browser extension to warn users about suspected phishing sites. The method of warning is beyond the scope of this work, but Cranor's framework for reasoning about the human in the loop [33] provides a useful starting point. The focus of this work is in building a tool that can detect phishing sites with high probability.

The basic approach is to make profiles of the sensitive sites and compare all loaded sites against these stored profiles. A profile of a site is a combination of different metrics that uniquely identifies that site. We noticed three types of phishing sites:

1. Sites that look exactly like the real sites.
2. Sites that represent real sites but do not look like them.
3. Sites that do not represent real sites.

Our approach focuses on catching the sites in category 1, although we discuss methods of catching sites in categories 2 and 3 as well.

To catch phishing sites that look like real sites, profiles of real sites are used. In these profiles, hashes of the SSL certificate and URL are stored for whitelisting and fuzzy hashes of contents (images, HTML code, scripts) as well as fuzzy hashes of the contents of selected known phishing sites that imitate the real site are stored for comparison against potential phishing sites. All sites are compared against these profiles in the background. As sites change their contents (and their SSL certificates), these profiles must be updated regularly. As a result of profile matching, three cases can occur:

1. The site matches with the whitelisted profile; in that case this is the right site.
2. The site does not match with any aspect of the profile; that means we do not have a profile for this site. In the event that the site supports SSL and asks for credentials, the system could query the user if she would like to make a profile.
3. The site partially matches with the profile. This could happen in a number of situations. We will consider two main cases here. These are:
   (a) SSL and address matches, but the content does not match: This could happen if the original site has been changed or attacker inserted phishing materials inside the original page. Our system will check if content has really been changed, if so then profile will be updated. Otherwise the user will be warned about the discrepancy of the content with the stored profile.
   (b) The content matches but SSL does not: Our system will consider this site as a phishing site as it tries to look like a legitimate site but the SSL does not match.

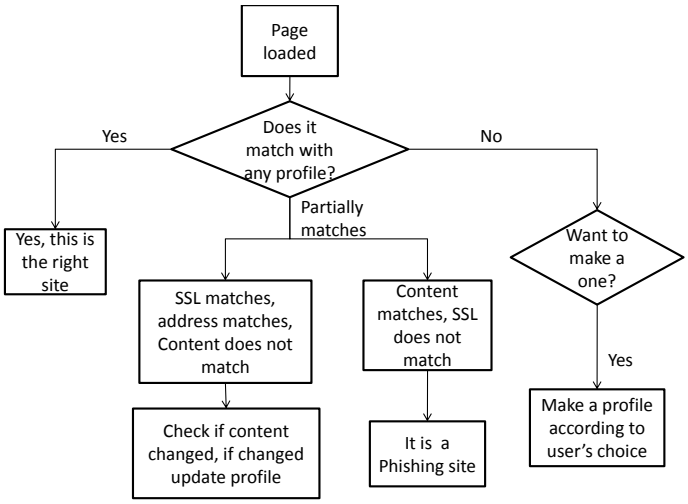

Figure 1: Phishing Detection Approach

PhishZoo's profiling approach can also be used to detect phishing sites that do not look like real sites. In these cases, profiles of previously identified phishing sites are used. URL and contents (images, HTML code, scripts) of phishing sites are stored as profiles. If a site does not match with any real site's profile, it will be matched with phishing profiles. PhishZoo will consider a site as a phishing site if it matches with any phishing profiles.

3.1 Profile making

Profiles of real sites are made at the user's request. Heuristic methods could be used to verify sites contents (validity of SSL, URL and all links). When making phishing profiles, only the phishing sites corresponding to sites in the real profiles are chosen. For example, if a user wants to make a profile of mybank.com, then the hashes of the real site of mybank.com are stored in the real site's profile and verified phishing sites of mybank.com are stored in the corresponding phishing profiles.

Whitelisting profile elements (SSL certificates and URLs) are stored using an exact, secure hashing algorithm. Phishing sites are often similar to real sites but are not identical. Therefore, the appearance content (images, HTML code, scripts) are stored using fuzzy hashing algorithms that can match similar profiles. The fuzzy hashing technique used by PhishZoo is context triggered piecewise hashing [16] as implemented by `ssdeep` [17]. By default, `ssdeep` generates context triggered piecewise hashes, or fuzzy hashes, for each input file. `ssdeep` combines a rolling hash with a traditional hash. An overview of `ssdeep`'s fuzzy hashing algorithm is given below, detailed algorithm is explained in `ssdeep` [17]:

1. A trigger value is computed based on the current file.
2. Rolling hash and traditional hash are computed as each byte of input is processed. A rolling hash is a hash function where the input is hashed in a window that moves through the input. As a traditional hash, Fowler/Noll/Vo (FNV) hash [18] is used.
3. The final hash signature consists of two parts, the trigger value for the second part is twice the trigger value of the first part. When the rolling hash triggers, that is when it is equal to the computed trigger value, a base64 encoded value of the LS6B (the six least significant bits) of the traditional hash is appended to first part of the final signature.
4. When the rolling hash modulo twice the trigger value is equal to twice the trigger value minus one ($h \; mod \; 2t = 2t - 1$), the LS6B of the traditional hash is appended to the second part of the final signature.
5. After processing the whole file, the first part of the signature is checked and, if it is not long enough, the trigger value is halved and the input is processed again.
6. The final hash signature consists of the block size, the two sets of LS6B's and the input's filename in quotes.

The accuracy of PhishZoo can be improved by also storing profiles of previously detected phishing sites that are often used as templates by attackers. These sites often fall into category 2: sites that do not look like the real site but are meant to represent the real site. A common example is a site that has just the logo of the real site and a message about needing to update account information. By storing profiles of a few of these sites (1 or 2 in our experiments), PhishZoo was able to greatly improve its accuracy. In a live system, these sites could be chosen and pushed to the extension in nightly updates (similar to how adblocking and current blacklisting extensions work).

3.2 Profile matching

When a site is fetched, it is first checked to see if it matches any whitelisted profiles. If not, the content of the potential phishing site is compared against the stored profiles of the sensitive set and stored phishing profiles.

1. After finding the signatures of the potential phishing site, their edit distance[1] from the signatures in the stored profiles is found. Signatures (and thus files) match when the ratio of the edit distance to the length is small.
2. Compute the match score of the files. The match score represents a conservative weighted percentage of how much of s1 and s2 are ordered homologous sequences.

$$matchscore = 100 - \frac{100 * S * e(s_1, s_2)}{64(l_1 + l_2)} \qquad (1)$$

where S = trigger value = 64 (by default)
s_i = signature of ith file, i = 1,2
l_i = signature length of ith file, i = 1, 2

[1]Number of insertions, modifications and deletions to turn Signature 1 into Signature 2.

That is, the match score measures of how many of the bits of these two signatures are identical and in the same order. The higher the match score, the more likely the signatures came from a common ancestor and the more likely the source files for those signatures came from a common ancestor. A higher match score indicates a greater probability that the source files have blocks of values in common and in the same order. In our algorithm, we consider any match score greater than zero to be positive.

Since the fuzzy hashing technique is applied separately on the various contents (e.g., images, html codes, scripts) of the site, some method must be used to consider whether the site as a whole is a match—as all sites using the paypal logo are not necessarily phishing paypal. We use a threshold to denote the similarity metric between testing site and stored site calculated as follows:

$$Threshold = m/n \quad (2)$$

where m = number of elements matched with the real site
n = total number of elements of the real site

In our current experiments, all elements in the site are equally weighted, however, we suspect there is room for improvement by prioritizing certain elements (for example, the logo) and deprioritizing other, more transient elements. We investigate the effect of varying the threshold in the next section.

3.3 Running PhishZoo in Bulk

The focus of our analysis imagines PhishZoo as a tool that is used to protect end-users against phishing attacks. However, our approach may ultimately prove useful to intermediaries, such as portals, browsers, ISPs, law enforcement or security companies, who seek to collect phishing sites for the purposes of blacklisting, takedown, or research.

These intermediaries could run a version of PhishZoo that includes many more profiles (of real sites and known phishing sites) on a repository gleaned from links in emails, webcrawling, or ads[2]. This process may enable faster detection than the crowd-sourcing techniques commonly relied upon.

4. EVALUATION AND RESULTS

To evaluate our approach, we used 636 phishing sites from `www.phishtank.com` and 20 profiles. We selected the phishing sites set from the repository of verified phishing sites provided by phishtank. The sites in the repository are submitted by users and then verified by voting. Recent studies have shown this repository to be mostly accurate, though vulnerable to attack [34].

After downloading sites from phishtank, we manually pruned the data set. We removed phishing sites of foreign language sites (e.g. VakifBank) except the foreign language versions of multinational brand name sites (e.g. French paypal), free offer sites (e.g. Free ipods! Enter information here!), imaginary company sites[3], and adult content sites. Our objective was to find phishing sites of popular brand name companies that users trust and mostly use—those that a user may wish to build a profile of to protect against phishing attacks. Manual analysis of the phishing set revealed the fact that some brand names are more prone to phishing attacks than the others. As the goal of this project is to detect the similarity of phishing sites to the corresponding real sites, we chose sites with many phishing attacks in our profile set. Note that thousands of phishing attacks are happening everyday and phishing trends change quickly, however, within the time frame of our experiment[4] the sites we chose to profile for our experiments had more reported phishing attacks than other sites according to phishtank. For this project we chose the page in a site that asks for users confidential information, e.g., account number, password, PIN number, user ID, etc. We also limited our analysis to sites that supported SSL.

The sites we chose to profile and the number of phishing sites of each profile are shown in Table 1. Overall 18% of the phishing sites had identical hash values and none of them had same url. It is likely that some of these identical sites represent a single attack hosted across multiple domains (as in the Rock Phish attacks described by Moore and Clayton [34]), however, others represent distinct attacks that simply copy sites wholesale from the original page or other phishing attacks. However as the numbers of duplicates we found were significantly lower than the 50% reported in that study, we suspect phishtank has improved their filtering and decided to include these sites in our results.

According to manual analysis, 77.36% of these sites look similar to real sites, 21.07% represent real site but the real site has no such page (e.g. an account confirmation page for paypal where the real paypal has no such page, or claim award page for bank of america), 1.57% of these sites do not represent any real sites. These are free offer sites that ask for bank account numbers or other credentials.

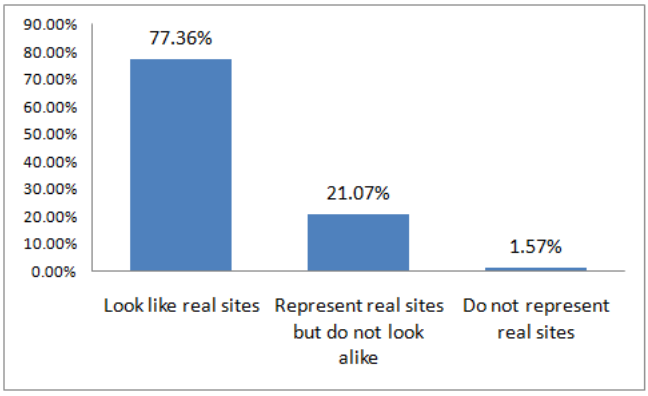

Figure 2: Results of manual analysis of our data set.

Evaluation is done in three phases and results are compared against top-performing toolbars [1]: Netcraft [5] and Firefox version 3 [15]. It is worth pointing out that, as Firefox and netcraft use blacklisting approaches, their accuracy is highly dependent on the freshness of the data. Since all the sites we considered were already verified as phishing sites

[2]Phishing or similar scams have lately been seen in advertisements that slip through screening.
[3]We used google to determine whether the company is real

[4]Timeframe of this experiment was January 2009 to April 2009

Profiles	Number of phishing sites	Number of look alike phishing sites
Abbey bank [38]	60	43
Alliance Leicester [35]	40	40
Anz bank [36]	2	2
Bank of america [37]	41	3
Cahoot [43]	3	3
Capital one bank [46]	2	2
Chase [39]	6	6
Common wealth bank of australia, netbank [40]	3	3
Ebay [41]	195	184
Egg banking [42]	7	7
Frost bank [44]	2	0
Halifax online banking [45]	26	26
HSBC [47]	23	1
Lloyds TSB [49]	7	7
Nationwide Bank [50]	2	0
Orkut [51]	53	52
Paypal [52]	145	95
USAA bank [53]	16	16
Wachovia [48]	2	1
Wells fargo [54]	1	0
Total	636	492

Table 1: Profiled sites examined. Only sites that support SSL were used in subsequent analysis.

by phishtank, this the most favorable possible situation for such tools. Previous studies of Netcraft have shown that its performance varies greatly (between 76% and 96% accuracy) from test to test for this reason [1]. We were unable to evaluate Cantina, one of the best heuristic approaches, on our dataset due to code unavailability. However, the authors report 90% accuracy with 1% false positives on their dataset [2].

We evaluated PhishZoo's profile matching ability under several different parameters. In our first experiment, only the html code of a site was considered in its content profile (apart from the whitelisting profile based on ssl and url). The objective of this phase is to verify how many phishing sites reuse the exact or very similar html code of the real site.

Our results (Figure 3) show that 49% of phishing sites can be detected using only HTML codes of sites as the profile content, whereas firefox and netcraft have a 90% and 96% success rate. The reason that simple html matching fails in 51% of cases is that the fuzzy hashing algorithm used for profile matching divides the text in several blocks and then applies hashing. Thus if the duplicate document was created by adding or deleting some texts from the original document, ssdeep can detect it. However, if the imitating document was structurally rearranged, the hash signature of the file becomes completely different than the original one and ssdeep could not detect similarity. One interesting trend that we observed from performing this experiment was that attacks against sites against which few phishing attacks were performed (such as small banks) tended to be susceptible to this simple version of PhishZoo as the attackers simply copied the site. Attacks against more common targets (paypal, ebay) appeared in both sets.

We also considered only using the logo of a site as its profile content, hypothesizing that this would help catch some of the "please fix your account" phishing sites that do not imitate the appearance of real sites. 54% sites can be detected using only logos. When logo-detection fails, it is because some logos are resized or the design is slightly changed in a way that is unnoticeable to the naked eye. ssdeep cannot detect these types of changes. In future work, we plan to experiment with alternate image-detection algorithms.

In our second experiment, the fuzzy hashing technique was applied to separate content elements (e.g., images, html codes, scripts) of the site. The results from this experiment are shown in Figure 3. Here threshold denotes the similarity metric between testing site and stored site, that is,

$$Threshold = m/n \tag{3}$$

where m = number of elements matched with real site
n = total number of elements of the real site

From the result, it is clear that accuracy of phishing detection depends on the matching threshold, shown in Figure 5. When threshold = 0.2 accuracy is 82.39% with false positive[5] 10%. Threshold = 0.3 gives an accuracy of 66.82% and 1% false positive rate. An example of a site that is caught with accuracy 0.2, but not with accuracy 0.3 can be seen in Figure 4. Accuracy decreased further as the threshold value was increased.

In many cases, when PhishZoo fails, it is because the phishing site does not look like the site it is imitating. When we restrict the set of phishing sites considered to those that look like the real sites (77.36% of the total dataset), the results improve to 95.33% accuracy with no false positives (as shown in Figure 3). This result shows that PhishZoo is

[5]Number of sites matched with different profile

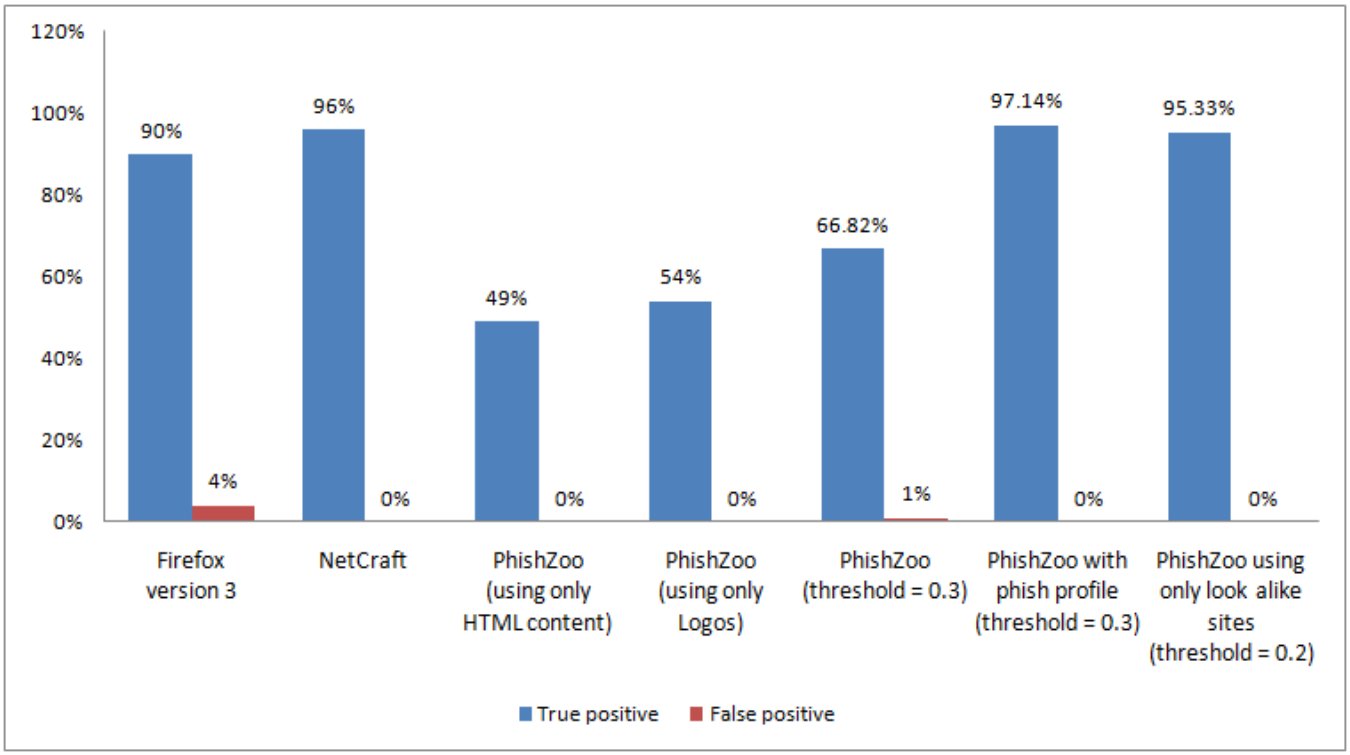

Figure 3: Accuracy of PhishZoo as compared to Firefox 3 and Netcraft. PhishZoo has 97.14% accuracy when used with selected phishing profiles. Without these profiles, Phishzoo is 95.33% accurate in detecting websites that look like the real sites they are imitating.

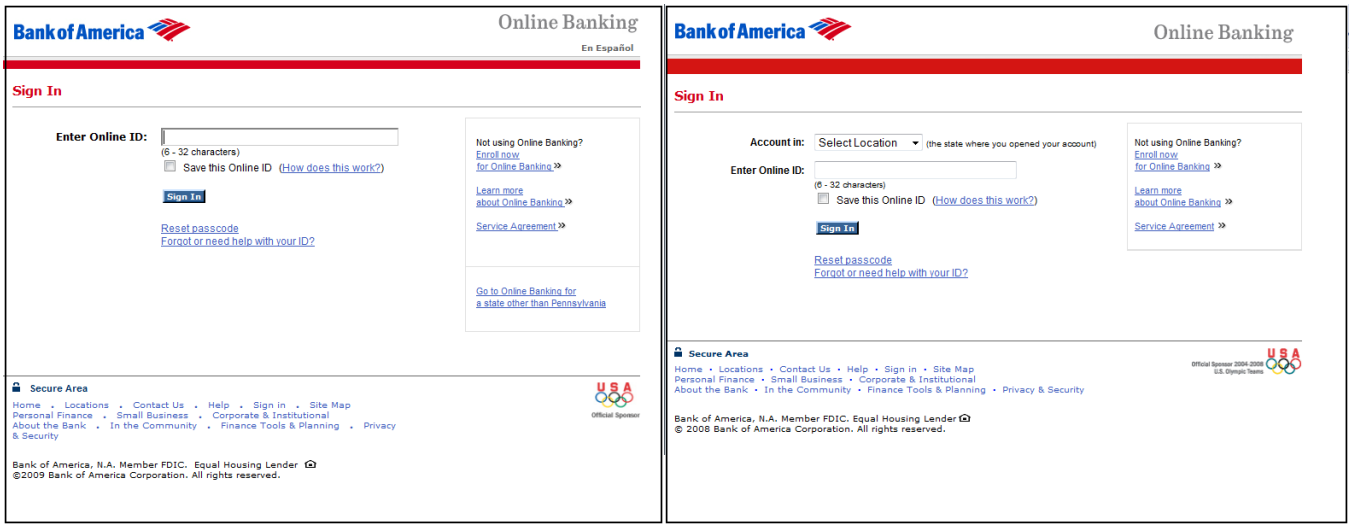

Figure 4: An example of a phishing attack that looks almost, but not entirely like the original site. The image on the left is the Bank of America login site and the image on the right is a phishing site. This site will be caught by PhishZoo if we use a threshold of 0.2 but not with threshold 0.3. As many phishing sites use the template on the right, it is a good candidate to add as an additional phishing profile.

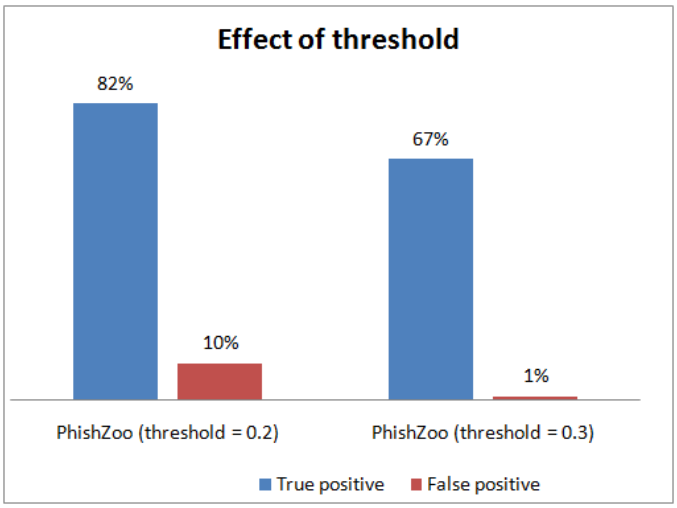

Figure 5: Effect of threshold

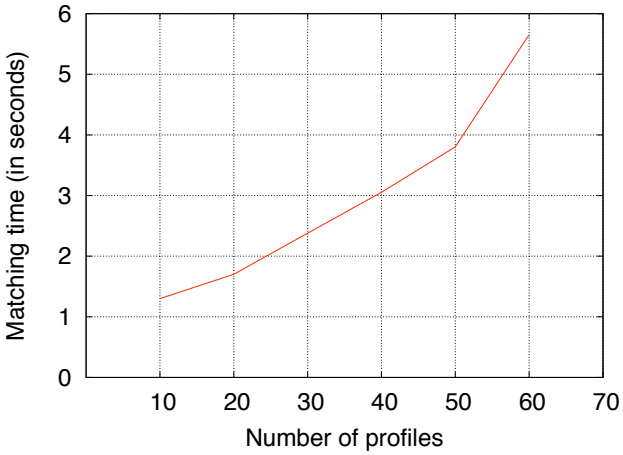

Figure 7: Time for matching profiles

most effective in recognizing those sites that are most likely to fool users.

Unfortunately, many sites "represent" real sites but do not look like them. Fortunately, these sites tend to use a few time tested templates for their cons, such as "Confirm your account" or "Claim your reward." An example of such a phishing attack is shown in Figure 6. As a result, adding a few carefully chosen phishing profiles to the profile of each real site can greatly improve its accuracy. In our third experiment, we used PhishZoo as is recommended, adding profiles of known phishing sites to the real sites' profiles. The number of phishing sites added per profile is shown in Table 2. The phishing sites were chosen from the set of sites that cannot be detected using fuzzy matching. 97% of the phishing sites in our dataset can be detected when we add these phishing profiles.

4.1 Performance Analysis

The time for matching a site against the profile database is a crucial aspect of the profiling approach, as users are unlikely to tolerate high performance penalties. We tested how long it took to match 10-60 profiles on a typical end-user machine (32 bit, 2 GHz processor running Windows Vista with 4 GB Ram). Results are shown in Figure 7; it takes less than 4 seconds to compare a site against 50 profiles. An individual is not likely to have more than 50 important sites that she wants to make profiles of, so matching time will not be a big overhead for the system. Our largest experiment, including profiles of all the real sites and selected phishing sites used 34 profiles (approx 2.5 seconds).

However, our approach may also be useful in "bulk mode" to large organizations that seek to protect users from phishing sites (such as email providers) or find phishing sites for the purpose of improving blacklisting or issuing take-downs. A version of PhishZoo with many more profiles of real sites and phishing sites could run on links gleaned from large email sets to automatically detect phishing sites in close-to-real time. Our approach is sufficiently fast for this application (especially since there will be no impatient human user), but our research application can be optimized much further.

5. DISCUSSION

The high accuracy achieved by the simple fuzzy matching algorithm used in the current version of PhishZoo reveals that most phishing sites are simply copies of real sites. This property of phishing sites has made them difficult for humans to detect, but as we show, easy for computers. However, the attacker community has proved itself able to quickly adapt to anti-phishing measures. In this section, we discuss the limitations of our approach, but also why we believe the approach represented by PhishZoo (if not the actual fuzzy hashing algorithm used) is likely to improve the ability to defend against phishing attacks in the long term.

PhishZoo's approach reduces the ability of attackers to automate their attacks, cutting into their profitability. By using the minimal knowledge base provided by the user-selected profiles, PhishZoo is able to compare potential phishing sites with real sites, making it difficult to do phishing attacks by simple copying. To avoid PhishZoo, phishing sites must look different than the real site or use web components that are different from the original site and *also* different from previous phishing attacks as these are included in the profiles. In the current implementation of PhishZoo, phishing detection could be avoided using resized images and logos, and restructuring html and script files. It is certainly possible to write a program that will automatically permute these elements nondeterministically to produce new phishing attacks that look similar to the original pages but evade PhishZoo's detection, but the parameters of such a program could be incorporated into PhishZoo's detection mechanism, again forcing phishing attacks to be carefully engineered to look similar to real sites using changed web components. Ultimately, this will lead to an "arms race" between attackers and defenders (as in spam detection), but one in which the defenders have an edge.

PhishZoo's approach can be easily extended and made more sophisticated. Matching resized images and restructure files could be easily be accomplished with computer vision algorithms, plagiarism detection tools, statistical classification,

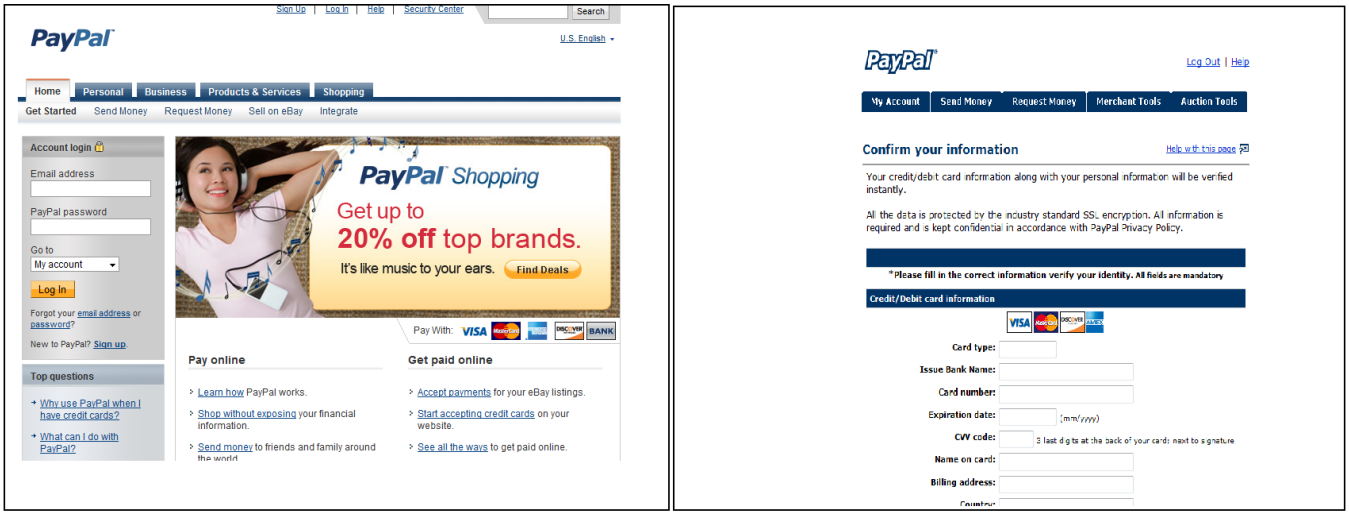

Figure 6: An example of a phishing attack that "represents" the real site, but does not look like it. The image on the left is the paypal login page, on the right is one of the ubiquitious "confirm your account" phishing scam pages.

Profiles	Number of phishing sites in profile
Abbey bank [38]	1
Alliance Leicester [35]	1
Bank of America [37]	2
Ebay [41]	2
Frost Bank [44]	1
HSBC [47]	1
Nationwide Bank [50]	1
Orkut [51]	1
Paypal [52]	3
Wachovia [48]	1
Total	14

Table 2: Number of phishing sites in used profile.

and/or other fuzzy matching techniques. The goal of the attacker will be to make websites that are different to computer algorithms, but (close to) identical to human eyes. If the attackers still prove successful in defeating PhishZoo, they will have contributed to our understanding of the vision problem in much the same way that spammers have improved statistical machine learning and bots that defeat captchas have improved optical character recognition algorithms. If these techniques succeed, but reduce the efficiency of PhishZoo, it can be run offline (on email links) or by intermediaries.

Ultimately, phishers will be forced to manually handcraft sites that defeat PhishZoo (once) and/or design phishing attacks that can be distinguished from real sites based on their basic appearance.

6. CONCLUSIONS AND FUTURE WORK

This project is the first step to a new approach of web-phishing detection which is much simpler and potentially promising. We believe there are many ways possible to improve current result. Topics of future research include:

1. **Matching images of sites:** Sometimes phishers use screen image of the original site instead of html codes. The system should have a provision for matching sites with their screen images.

2. **Consider other similarity matching algorithms:** Fuzzy hashing (ssdeep) is one way to match similar content. Other algorithms or a combination of different similarity matching algorithms may produce better results.

3. **Number of pages to consider:** Phishers could imitate any page of a legitimate site. As the most important page of a site is the page that asks for user's account information, we have only considered that page while making a profile. Other pages also should be considered for complete security. But the matching time increases as more pages added to the system. This tradeoff should be investigated further and included as a security parameter of PhishZoo.

4. **Sites without SSL:** System should be extended to identify phishing attack of sites without SSL.

5. **SSL spoofing:** Validating site's authenticity only based on SSL is not always reliable. SSL spoofing cases should also be considered.

Today a large portion of phishing detection relies upon human users to report and verify phishing sites. In this work, we investigate a new approach for phishing detection based on profiling the content of phished websites to determine when a user is being deceived by a false belief. We provide an empirical evaluation showing that this method works well (97% accuracy) against current phishing attacks and will identify new and targeted phishing sites where blacklisting-based toolbars fail. This method is also most accurate against sites that look most like the real sites (those hardest for end users to detect). Further research on this approach will help to create a robust system for phishing detection with minimal human intervention. Future incarnations of PhishZoo can make use of advances in visual pattern matching and computer vision and has the potential to drive advances in this field as well.

7. REFERENCES

[1] Phinding Phish: Evaluating Anti-Phishing Tools, Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong, In Proceedings of the 14th Annual Network & Distributed System Security Symposium (NDSS 2007), San Diego, CA, 28th February - 2nd March, 2007.

[2] CANTINA: A Content based approach to detecting phishing web sites, Yue Zhang, Jason Hong, Lorrie Cranor, In Proceedings of the 16th International conference on World Wide Web, Banff, Alberta, Canada, May 8-12, 2007.

[3] A Proposal of the AdaBoost-Based Detection of Phishing Sites, Daisuke Miyamoto, Hiroaki Hazeyama, and Youki Kadobayashi (Internet Engineering Laboratory, Graduate School of Information Science, Nara Institute of Science and Technology, Japan),In JWIS, August 2007.

[4] CallingID, Ltd. Accessed: November 03, 2008. `http://www.callingid.com/DesktopSolutions/CallingIDToolbar.aspx`

[5] Netcraft. Netcraft Anti-Phishing Tool. Accessed: November 03, 2008.`http://toolbar.netcraft.com/`.

[6] Global Phishing Survey: Domain Name Use and Trends in 1H2008, `http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey1H2008.pdf`

[7] TrustBar: Protecting (even Naive) Web Users from Spoofing and Phishing Attacks. Herzberg, A. and A. Gbara, 2004, Cryptology ePrint Archive: Report 2004/155.

[8] Cloudmark, Inc. Accessed: November 4, 2008. `http://www.cloudmark.com/desktop/download/`

[9] EarthLink, Inc. EarthLink Tool. Accessed: November 4, 2008. `http://www.earthlink.net/software/free/tool/`

[10] eBay, Inc. Using eBay Tool's Account Guard. Accessed: November 4, 2008. `http://pages.ebay.com/ebay_toolbar/tours/tour5.html`

[11] GeoTrust, Inc. TrustWatch Tool. Accessed: November 4, 2008. `http://toolbar.trustwatch.com/support/toolbar/`

[12] Microsoft Corporation. Internet Explorer 7. Accessed: November 4, 2008. `http://www.microsoft.com/protect/products/yourself/phishingfilter.mspx`

[13] Netscape Communications Corp. "Security Center." Accessed: November 4, 2008. `http://browser.netscape.com/ns8/product/security.jsp`

[14] Client-Side Defense against Web-Based Identity Theft,Chou, Neil, Robert Ledesma, Yuka Teraguchi, Dan Boneh and John C. Mitchell, in Proceedings of The 11th Annual Network and Distributed System Security Symposium (NDSS '04), San Diego, CA February, 2004.

[15] Google Safe Browsing Service in Mozilla Firefox Version 3. Accessed: December 1, 2008. `http://code.google.com/apis/safebrowsing/firefox3_privacy.html`

[16] Identifying almost identical files using context triggered piecewise hashing, Jesse Kornblum, Digital Investigation,volume-3,2006, pages 91-97.

[17] ssdeep-2.0,`http://ssdeep.sourceforge.net/usage.html`

[18] FNV hash:`http://www.isthe.com/chongo/tech/comp/fnv/#FNV-1`

[19] Why Phishing Works, Rachna Dhamija, J. D. Tygar and Marti Hearst. In the Proceedings of the Conference on Human Factors in Computing Systems (CHI2006), 2006.

[20] The Emperor's New Security Indicators, Stuart Schechter, Rachna Dhamija, Andy Ozment and Ian Fischer. In Proceedings of the IEEE Symposium on Security and Privacy, May 2007.

[21] You've Been Warned: An Empirical Study of the Effectiveness of Web Browser Phishing Warnings. S. Egelman, L. Cranor, and J. Hong. In Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI2008), 2008.

[22] Decision Strategies and Susceptibility to Phishing. J. Downs, M. Holbrook, and L. Cranor. In Proceedings of the 2006 Symposium On Usable Privacy and Security, 12-14 July 2006, Pittsburgh, PA.

[23] What Instills Trust? A Qualitative Study of Phishing,M. Jakobsson et al., Proceeding of first Int'l Workshop on Usable Security, Springer-Verlag, 2007;

[24] Protecting People from Phishing: The Design and Evaluation of an Embedded Training Email System, Ponnurangam Kumaraguru, Yong Rhee, Alessandro Acquisti, Lorrie Faith Cranor, Jason Hong, Elizabeth Nunge, In Proceedings of the Conference on Human Factors in Computing Systems (CHI2007), 2007.

[25] Waterken Inc., Waterken YURL Trust Management for Humans, `http://www.waterken.com/dev/YURL/Name/`

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

[27] The Battle Against Phishing:Dynamic Security Skins,Rachna Dhamija, J.D. Tygar, In SOUPS 2005: Proceedings of the 2005 ACM Symposium on Usable Security and Privacy, ACM International Conference Proceedings Series, ACM Press, July 2005, pp. 77-88

[28] Trusted Paths for Browsers, Zishuang Ye, Sean Smith,

In Proceedings of the 11th Usenix Security Symposium,2002.

[29] Intelligent Phishing Website Detection System using Fuzzy Techniques, M. Aburrous and M.A. Hossain and F. Thabatah, and K. Dahal, In ICTTA 2008, Proceedings of Information and Communication Technologies: From Theory to Applications, 2008.

[30] Approaches to Phishing identification Using Match and Probabilistic Digital Fingerprinting Techniques, J. Zdziarski, W. Yang, and P. Judge, Spam Conference, 2006.

[31] Learning to Detect Phishing Emails, Ian Fette , Norman Sadeh and Anthony Tomasic, In WWW2007, Proceedings of the 16th International World Wide Web Conference, May 2007.

[32] The Impact of Incentives on Notice and Take-down, Tyler Moore and Richard Clayton, In Seventh Workshop on the Economics of Information Security. June 26-28, 2008.

[33] A Framework for Reasoning about the Human in the Loop, Lorrie Faith Cranor, In Workshop on Usability, Psychology, and Security. 2008.

[34] Evaluating the Wisdom of Crowds in Assessing Phishing Websites, Tyler Moore and Richard Clayton, In Financial Cryptography and Data Security. 2008.

[35] Alliance Leicester. Accessed: February 4, 2009. https://www.mybank.alliance-leicester.co.uk/index.asp

[36] ANZ Bank. Accessed: February 4, 2009. https://www.anz.com/INETBANK/bankmain.asp

[37] Bank of America. Accessed: February 4, 2009. https://www.bankofamerica.com/index.jsp

[38] Abbey bank. Accessed: February 4, 2009. https://myonlineaccounts2.abbeynational.co.uk/CentralLogonWeb/Logon?action=prepare

[39] Chase online. Accessed: February 4, 2009. https://www.chase.com/

[40] Common wealth bank of australia, netbank. Accessed: February 4, 2009. https://www3.netbank.commbank.com.au/netbank/bankmain

[41] Ebay. Accessed: November 8, 2008. https://signin.ebay.com/ws/eBayISAPI.dll?SignIn&ru=http%3A%2F%2Fwww.ebay.com

[42] Egg Online Bank. Accessed: February 4, 2009. https://new.egg.com/security/customer/logon?URI=https://new.egg.com/customer/youraccounts

[43] Cahoot, a division of Abbey National plc. Accessed: February 4, 2009. https://ibank.cahoot.com/servlet/com.aquarius.security.authentication.servlet.LoginEntryServlet

[44] Frost bank. Accessed: February 4, 2009. https://www.frostbank.com/cgi-bin/ecomm/portal/signin/loginhelpenter.do

[45] Halifax Online Banking. Accessed: February 4, 2009. https://www.halifax-online.co.uk/_mem_bin/formslogin.asp?source=halifaxcouk&page=login&simigvis=NC4yMC4yNjc0OTU5OTY1MzI3Ny4xMjM5OTI3MzcONDYx*

[46] Capital One Online Banking. Accessed: February 4, 2009. https://onlinebanking.capitalone.com/capitalone/

[47] HSBC Online Banking. Accessed: February 4, 2009. https://www.us.hsbc.com/1/2/3/personal/online-services/personal-internet-banking/log-on

[48] Wachovia. Accessed: February 4, 2009. https://onlineservices.wachovia.com/auth/AuthService?action=presentLogin&url=https%3a//onlineservices.wachovia.com/NASApp/NavApp/Titanium%3faction%3dreturnHome112

[49] Lloyds TSB. Accessed: February 4, 2009. https://online.lloydstsb.co.uk/customer.ibc?WT.svl=ibcplogon&WT.ac=hpIBlogon

[50] Nationwide bank. Accessed: February 4, 2009. https://bankonline.nationwidebank.com/bankonline/login.do

[51] Orkut. Accessed: February 4, 2009. https://www.google.com/accounts/ServiceLogin?service=orkut&hl=en-US&rm=false&continue=http%3A%2F%2Fwww.orkut.com%2FRedirLogin.aspx%3Fmsg%3D0%26page%3Dhttp%253A%252F%252Fwww.orkut.com%252FSignup.aspx&cd=US&passive=true&skipvpage=true&sendvemail=false

[52] Paypal. Accessed: February 4, 2009. https://www.paypal.com/

[53] USAA bank. Accessed: February 4, 2009. https://www.usaa.com/inet/ent_proof/proofingEvent?action=Init&event=forgotOnlineId

[54] Wells fargo. Accessed: February 4, 2009. https://www.wellsfargo.com/