PhishZoo: Detecting Phishing Websites By Looking at Them

Sadia Afroz
Department of Computer Science
Drexel University
Philadelphia, PA 19104
Email: sa499@drexel.edu

Rachel Greenstadt
Department of Computer Science
Drexel University
Philadelphia, PA 19104
Email: greenie@cs.drexel.edu

Abstract—**Phishing is a security attack that involves obtaining sensitive or otherwise private data by presenting oneself as a trustworthy entity. Phishers often exploit users' trust on the appearance of a site by using webpages that are visually similar to an authentic site. This paper proposes a phishing detection approach—PhishZoo—that uses profiles of trusted websites' appearances to detect phishing. Our approach provides similar accuracy to blacklisting approaches (96%), with the advantage that it can classify zero-day phishing attacks and targeted attacks against smaller sites (such as corporate intranets). A key contribution of this paper is that it includes a performance analysis and a framework for making use of computer vision techniques in a practical way.**

I. 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 were at least $67,677$ phishing attacks in the last six months of 2010 [1]. Their recent reports [2] showed that most phishing attacks are "spear phishing" that target financial and payment sectors. This paper proposes a phishing detection approach—PhishZoo—that uses profiles of trusted websites' appearances to detect targeted phishing attacks. We use URLs and contents of a website to identify imitations. We show where this type of approach succeeds (and fails) and, in the process, illuminate current trends in phishing attacks.

It's a sad, but common story. Alice follows a link to a website purporting to be her bank. She arrives at a webpage that looks reassuringly like her bank, featuring the logo that the bank has invested hefty amounts of cash to graphic designers and advertisers to associate with their brand. The site Alice visits is only a few hours old and unknown to blacklists. Alice delivers her credentials to the attacker.

Perhaps if Alice had known that she should check for the domain name and indicators of a valid SSL connection she would be okay, but most users do not know how to do this. And, even when they do, or when the task is simplified to checking a phishing toolbar, this is exactly the sort of repetitive task that humans are skilled at forgetting. 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 [3], [4]. This human factor has complicated phishing attack prevention.

Perhaps Alice should learn never to click on links. But this too, is unrealistic. The reality is that Alice will give the site a brief glance, and if what she sees does not contradict her expectations, she will log in. Analyses revealed that over 90% users depend on a website's appearance as an indication of its authenticity [5]–[7] and fall for malicious, but well-designed phishing sites that look almost (or exactly) like legitimate sites.

Maybe we should give up on Alice. After all, attackers copy legitimate websites. They look identical to the real sites. Alice will never detect imitation sites by looking at them. However, maybe her browser can help.

The browser can learn which sites Alice has accounts on and detect them via their domain name and SSL certificate. The problem comes when Alice visits other sites. How can the browser distinguish benign novel or untrusted sites from phishing sites? 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.

If the attacker actually copies the real site wholesale (as they do in roughly 50% of the current attacks we have studied) this is trivial. However, if the site is merely designed to look similar, more sophisticated detection methods are needed. In this paper, we discuss how computer vision algorithms can be used to detect phishing attacks that imitate the appearance of legitimate websites. Our approach does not depend on users that vigilantly check for indicators of authenticity and can catch both new, not-yet-blacklisted attacks and targeted attacks (against a corporate intranet, for example) that will not appear in blacklists.

This paper presents and evaluates a new approach for web phishing detection based on profiles of sensitive sites' appearance and content. Our method—PhishZoo—makes profiles of sites consisting of the website contents and images displayed. These profiles are stored in a local database and are either matched against the newly loaded sites at the time of loading or against risky sites (for example, links in email) offline. We also test against profiles of common phishing pages to increase accuracy. Key contributions of PhishZoo approach include:

1) We investigated fast, online detection using URL and HTML content in the profile of a site. Our method can detect 90% of current phishing sites, with only 0.5% false positives. This approach is fast enough to be run in real time, however, there are straightforward ways for attackers to adapt to this approach and make it less effective.
2) We investigated vision techniques to detect phishing sites more robustly. A robust vision solution will ultimately require both matching images, which we explore, and scene analysis (segmenting images into objects). This paper explores the matching problem, which is sufficient to detect current phishing sites. Using the SIFT image-matching algorithm, PhishZoo can detect 96.10% of phishing sites, more slowly, with a false positive rate of 1.4%. This method can be used offline by intermediaries to more quickly detect phishing websites or by users on links in their mail spools.
3) Our approach depends only on websites' content to detect corresponding phishing sites. It can detect new phishing sites which are not yet blacklisted and targeted attacks against small brokerages and corporate intranets.

The rest of the paper is arranged as follows: in Section 2, we briefly survey anti-phishing approaches and detail the novelty of our approach. Section 3 describes the threat model and assumptions of this work. Section 4 describes the profiling mechanisms used by PhishZoo in detail. In Section 5, data collection method is explained. Our empirical evaluation techniques and experimental results are discussed in Section 6. In Section 7, we discuss possible improvements to PhishZoo's performance, ways in which attackers may be able to adapt to PhishZoo's current mechanisms, as well as countermeasures that can be incorporated into PhishZoo to defeat these more sophisticated attacks. We conclude by outlining future directions for this line of research and for PhishZoo.

II. Related work and novelty

Current phishing detection approaches fall into three main categories: (1) Non-content based approaches that do not use content of the site to classify it as authentic or phishing, (2) Content based approaches that use site contents to catch phishing, and (3) Visual similarity based approaches that identify phishing using their visual similarity with known sites. These approaches are each discussed, then contrasted with our approach.

Other anti-phishing approaches include detecting phishing emails [8] (rather than sites) and educating users about phishing attacks and human detection methods [9].

A. Non-content based approaches:

Non-content based approaches include URL and host information based classification of phishing sites, blacklisting and whitelisting methods.

In URL based schemes, URLs are classified based on both lexical and host features. Lexical features describe lexical patterns of malicious URLs. These include features such as length of the URL, the number of dots, special characters it contains. Host features of the URL include properties of IP address, the owner of the site, DNS properties such as TTL, and geographical location [10]. Using these features, a matrix is built and run through multiple classification algorithms. In real-time processing trials, this approach has success rates between 95-99%. In our approach, we used lexical features of URL along with site contents and image analysis to improve performance and reduce false positive cases.

In Blacklisting approaches, users report or companies seek and detect phishing sites' URLs which are stored in a database. Most commercial toolbars Netcraft [1], Internet explorer 7, CallingID Toolbar, EarthLink Toolbar [2], Cloudmark Anti-Fraud Toolbar [3], GeoTrust TrustWatch Toolbar [4], Netscape Browser 8.1 [5] use this approach. But as most phishing sites are short-lived, last less than 20 hours [11], or change URLs frequently (fast-flux), the URL 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.

Whitelisting approaches seek to detect known good sites [12]–[14], but a user must remember to check the interface every time he visits any 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, for example, Dynamic security skins [15], TrustBar [13], SRD ("Synchronized Random Dynamic Boundaries") [16].

B. Content based approaches:

In content based approach, phishing attacks are detected by examining site contents. Features used in this approach include spelling errors, source of the images, links, password fields, embedded links, etc. along with URL and host based features. SpoofGuard [12] and CANTINA [17] are two such approaches.

Google's anti-phishing filter detects phishing and malware by examining page URL, page rank, WHOIS information and contents of a page including HTML, javascript, images, iframe, etc. [18]. The classifier is regularly re-trained with new phishing sites to pick up new trends in phishing. This classifier has high accuracy but is currently used offline as it takes 76 seconds on average to detect phishing.

Several researchers explored fingerprinting and fuzzy logic based approaches that use a series of (exact) hashes of websites to identify phishing sites [19], [20]. Our experimentation with a fuzzy hashing based approach suggested that this approach

[1] Netcraft: http://toolbar.netcraft.com/

[2] Earthlink toolbar: http://www.earthlink.net/software/free/tool/

[3] Cloudmark: http://www.cloudmark.com/desktop/download/

[4] Geo Trust: http://toolbar.trustwatch.com/support/toolbar/

[5] Netscape: http://browser.netscape.com/ns8/product/security.jsp

can detect current attacks, but can be easily circumvented by restructuring HTML elements without changing the appearance of the site [21].

C. Visual similarity based phishing detection:

Chen et al. used screenshot of webpages to detect phishing sites [22]. They used Contrast Context Histogram (CCH) to describe the images of webpages and k-mean algorithm to cluster nearest keypoints. Finally euclidean distance between two descriptors is used to find matching between two sites. Their approach has 95-99% accuracy with 0.1% false positive. In our experiment we showed that analyzing screenshot is too slow to be used for online phishing detection.

Fu et al. used Earth Mover's Distance (EMD) to compare low resolution screen capture of a webpage [23]. Images of webpages are represented using color of a pixel in the image (alpha, red, green, and blue) and the centroid of its position distribution in the image. They used machine learning to select different threshold suitable for different webpages.

Matthew Dunlop experimented with optical character recognition to convert screenshot of websites to text and then used Google PageRank to identify legitimate and phishing sites.

Other visual similarity based approaches includes Liu et al's visual similarity assessment using layout and style similarity [24] and iTrustPage [25] that uses Google search and user opinion to identify visually similar pages.

D. 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 [5]–[7].

PhishZoo can detect current phishing sites if they look like authentic sites by matching their content against a stored 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, it can be used to improve security as opposed to the current case, when this branding is co-opted by attackers to abuse users' trust.

III. Threat model, Assumptions, and Scope

The goal of this work is to identify phishing attacks through automated means. We define a phishing attack as occurring when an attacker presents a site to the user that uses its visual appearance to appear similar to a legitimate site (look and feel, etc) with the goal of convincing the user to enter credentials (e.g. username and password).

We do not consider sites which exploit browser vulnerabilities to infect machines with malware (drive by downloads [26]). One approach we explored was comparing screenshots of rendered pages to the stored profiles of trusted pages. This approach should be done in an isolated environment to prevent infection. Adding heuristics for the detection of malicious sites (redirects, obfuscated scripts) would be an interesting direction for future work, but is not explored here. We assume throughout the rest of the paper that the machine is not infected and viewing pages will not cause infection.

Our approach depends on users identifying trusted sites for profiling. Like SSH, we assume that when users identify sites the first time, the site is genuine. We further assume that SSL is supported by the sites in question and correctly configured.

The focus of this work is on matching sites: we do not focus on user interface issues. We assume that if false positives are low, sites can be blocked or significant barriers to assess erected. If false positives are high, the approach will likely fail.

IV. Approach

In this section, we explain our phishing detection approach. We start with an overview of the approach, followed by a explanation of site profiling and profile matching. Finally, we explain how PhishZoo can be used for online and offline phishing detection.

We detect phishing sites using content similarity between real sites and malicious sites. Malicious sites tend to use sensitives sites' appearance to provoke *false belief* in users. PhishZoo makes profiles of sensitive sites and compares all loaded sites against these stored profiles. This model has several advantages over non-content based approaches. First, profile matching approach depends only on current contents, so a phishing site can be detected as soon as it is loaded. Second, it can detect phishing attacks in cases where URL-based machine learning approaches fail, for example, targeted attacks on non-popular sites, attacks on compromised sites, phishing sites hosted on reputable hosting services, and URL with benign tokens. Third, as the majority of users provide sensitive credentials to a small set of sites (fewer than 20 [27]), this approach can provide user-customized phishing protection by protecting sites that are important to a particular user. Finally, it can be used to augment current blacklisting approach as it can detect new attacks where other anti-phishing approaches fail, for example targeted and picture-in-picture attacks.

Our approach is illustrated in Figure 1. Whenever a site is loaded it is matched against the stored profiles. If the SSL and URL of the loaded site match with the SSLs and URLs of any of the profiles, then PhishZoo determines the site to be a legitimate site. Otherwise, the site's contents will be matched against our appearance profiles. Matching with profiles consists of a number steps. First, tokens in the hostname, URL and HTML files are extracted. Then, these tokens are searched for specific keywords selected from the protected sites. After this step, all the images of the current site are matched against the

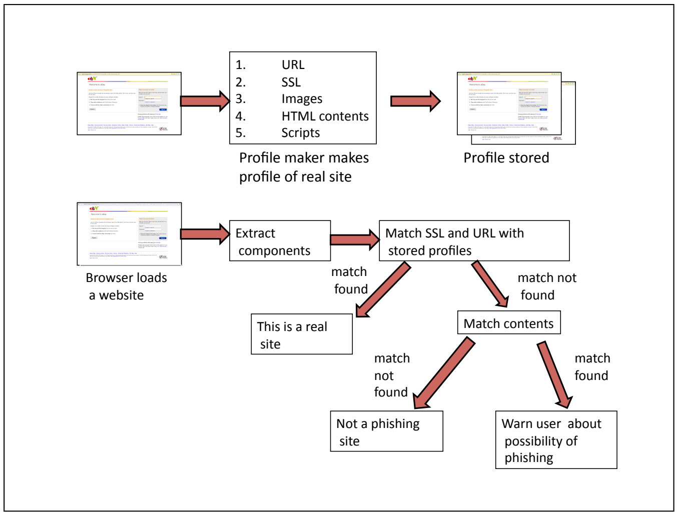

Fig. 1. Phishing Detection Approach. The SSL and URL information is used to detect whitelisted sites. For sites whose SSL and URL do not match, content matching is used to detect phishing attacks (imitations of real sites).

logos of the stored sites. Image matching step is necessary to reduce false positive rate. Importance of keyword and image matching steps are explained further in the Evaluation section. PhishZoo determines a site as a phishing site if its contents match with any of the protected profiles. Otherwise, the site is considered as a non-phishing site.

A. *Profile making*

A profile of a site is a combination of different metrics that uniquely identifies that site. A user chooses real sites that he wants to protect from phishing to be saved as profiles. Heuristic methods can be used to help verify a site's authenticity at this stage. In a profile, PhishZoo stores SSL certificates, URL and contents related to a site's appearance such as HTML files, extracted features of the logo. In the current version of PhishZoo, logo is selected by the user.

To extract features from a site logo, Scale Invariant Feature Transform (SIFT) algorithm [28] is used. This algorithm transforms an image into a large collection of local feature vectors. Each of these vectors is invariant to image translation, scaling, and rotation, and partially invariant to illumination changes and affine or 3D projection. We noticed that many phishing sites use logos that are scaled or translated version of some original logos. As SIFT features are invariant to these changes, similarity between logos can be detected even in these cases.

B. *Profile matching*

We use the profile contents discussed in previous section to identify targeted phishing attacks. When a site is fetched, PhishZoo checks if its URL matches with any whitelisted URLs. If not, contents of the site is compared against the stored profiles of the sensitive set of sites. Image matching between genuine and phished site improves phishing detection accuracy and reduces false positive rate. In the current version of PhishZoo, we only match logos of the stored sites against all the images of the newly loaded site. We noticed that matching every image of a site with all the images of all the protected sites increases phishing detection accuracy but may slow down the website loading to an unacceptable level. User or site administrator can choose the level of matching depending on the expected level of protection.

Profile matching is performed in several steps. At first, tokens in the hostname (delimited by '.'), in the path URL (strings delimited by '/,' '?,' '.,' ',' '=,'-, and '_) and HTML files are extracted. Then, these tokens are searched for specific keywords selected from the protected sites. TF-IDF technique is used to select keywords from the domain name of the protected URLs and HTML files. A site containing the selected keywords is most likely to masquerade as any of the protected site. The selected keywords are also used to select the most relevant profile whose logo will be matched with the images of the newly loaded site. In second step, logo of the most relevant profile is matched with the images of the current site. SIFT features of the images are used for matching. During the matching, SIFT extracts scale-invariant keypoints of the images on the candidate site and finds matching keypoints that are similar to the keypoints of the logo. Then a match score is computed as follows:

$$Match\ score = \frac{Number\ of\ keypoints\ matched}{Total\ keypoints\ in\ the\ original\ logo} \quad (1)$$

Higher match scores represent greater degrees of similarity. If the match score is greater than a threshold, then the image considered as similar to the logo. In this case, PhishZoo flags the candidate site as a phishing site.

C. *Image matching using SIFT*

SIFT is traditionally used by computer vision applications to recognize objects in cluttered, real world scenes [28]. Detecting logos in a webpage is a similar, but much simpler object recognition task. Scale invariant features are required in this case because many phishers scale, translate, or apply small distortions to an original logo that are hard to notice by humans but can evade simpler image matching approaches. In addition, most current tools fail to detect picture-in-picture phishing attacks where a phishing site uses screenshot and pictures of a real site instead of HTML content. SIFT is used to overcome these obstacles.

Before turning to SIFT, we explored simpler image matching algorithms based on fuzzy hashing or included in packages like ImageMagick. These algorithms are faster than SIFT, but are easy for attackers to circumvent. We also explored OCR algorithms, since many logos contain text. This worked reasonably well in cases where the logo contains only text, for example PayPal logo, but failed when we studied more complex logos such as the logos of EBay and Bank of America. We determined that a more sophisticated, vision-based approach was needed. SIFT image matching is a standard approach that is used in many object recognition and image matching researches [29]. Many subsequent researches used on variants of SIFT to improve matching speed in specific applications [30]. These variants or a customized variant might prove fruitful for anti-phishing research, SIFT is a logical approach for the initial exploration of the space.

D. Running PhishZoo in Bulk

Our analysis envisions PhishZoo as a tool that will be used to protect end-users against phishing attacks. However, our approach may ultimately prove more 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 advertisements [6].This process may enable faster detection than the crowd-sourcing techniques commonly relied upon.

E. Online and offline profile matching

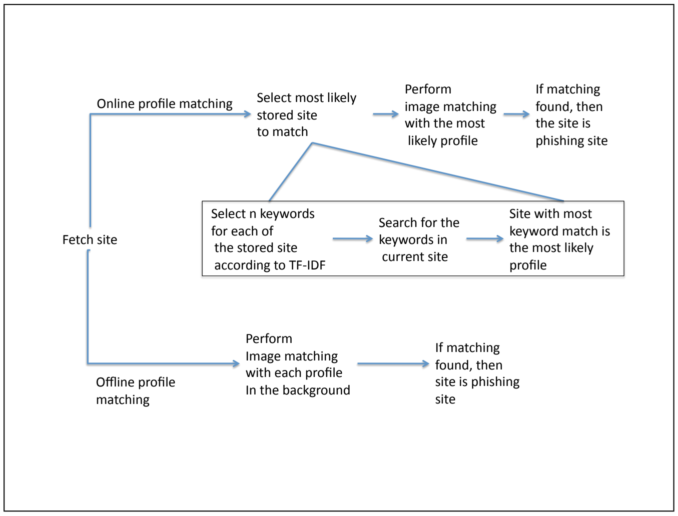
Online profile matching
Select most likely stored site to match
Perform image matching with the most likely profile
If matching found, then the site is phishing site
Fetch site
Select n keywords for each of the stored site according to TF-IDF
Search for the keywords in current site
Site with most keyword match is the most likely profile
Perform Image matching with each profile In the background
If matching found, then site is phishing site
Offline profile matching

Fig. 2. How PhishZoo should be used

V. Data Selection

This section describes the data set we used to evaluate our approach. We selected 1000 verified phishing sites from Phishtank. These sites are reported by users and verified as phishing by voting. For false positive testing, we used 200 most popular sites accessed by Internet users (taken from http://www.alexa.com/topsites).

Our objective was to find phishing sites of popular brand name companies that users trust and mostly use—those that a user might want to build a profile of to protect against phishing attacks. Manual analysis of the phishing set revealed that some brand names are more prone to phishing attacks than the others. In our profile set we chose sites with many phishing attacks so that we have sufficient data to evaluate our approach. Note that thousands of phishing attacks are happening everyday and phishing trends change quickly, however, according to phishtank within the time frame of our experiment [7] the sites we chose to profile had more reported phishing attacks than other sites. Within any site, we made profile of the page that asks for confidential information, for example account number, password, PIN number, user ID. We also limited our analysis to sites that support SSL.

In our dataset, 18% of the phishing sites had identical hash values. 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 [31]), however, others represent distinct attacks that simply copy sites wholesale from the original page or other phishing attacks. 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 our manual analysis, 77.36% of the phishing sites in our dataset look similar to some real sites. 21.07% of the sites represent some real sites but the real site has no such page, for example an account confirmation page for Paypal but the real Paypal site has no such page, or a fake “claim your award” page for Bank of America. 1.57% of the phishing sites do not represent any real sites. These are free offer sites that ask for bank account numbers or other credentials.

VI. Evaluation

In this section, we show effectiveness of our approach in phishing detection and discuss performance issues and error cases. In particular, we show the effect of profile contents and threshold values in phishing detection, robustness of SIFT and discuss the common trends in the phishing sites where our approach fails.

A. Profile Content Analysis:

We evaluated PhishZoo’s effectiveness under several different parameters, results are shown in Table I and Figure VII. According to our results, 90.2% of the phishing sites were detected with keyword and image matching. When only keywords from profiles were used, PhishZoo detected 97.6% of the phishing sites, but with high false positive.

Our results also indicated that 21.5% of the phishing sites directly reused real sites’ elements and can be detected by HTML code matching. More phishing (70.3%) was detected by considering only visible texts [8] of the site instead to the whole HTML. One interesting trend observed was that attacks against sites against which few phishing attacks were found (such as small banks) can be detected using this simple version of PhishZoo as the attackers copied original sites in these cases. Attacks against more common targets (such as Paypal and Ebay) appeared in both sets.

[6] Phishing or similar scams have been seen in advertisements that slip through screening. http://www.bizreport.com/2007/05/google_pulls_phishing_ads.html

[7] Timeframe of this experiment was August 2010 to September 2010

[8] Visible text is the portion of text in a HTML that is visible to the user in a webpage.

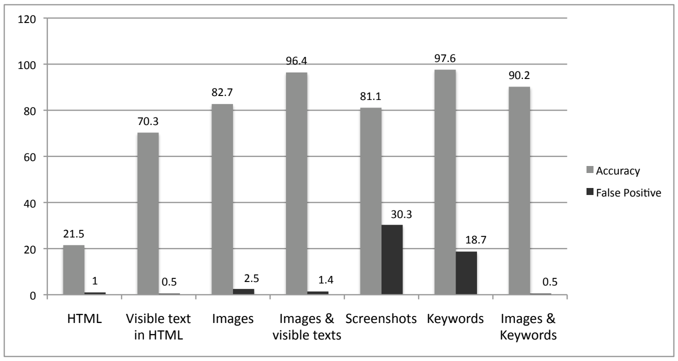

Fig. 3. Comparison of PhishZoo using different profile contents. The grey bars denote accuracy in phishing detection and the black bars denote false positive rate. PhishZoo performs best when both images and keywords are used as profile content.

PhishZoo approach: profile content	Accuracy	False positive
HTML	21.5%	1%
Visible text in HTML	70.3%	0.5%
Images	82.7%	2.5%
Image and visible texts	96.4%	1.4%
Screenshots	81.1%	30.3%
Keywords	97.6%	18.7%
Image and Keywords	90.2%	0.5%

TABLE I
PHISHZOO PHISHING DETECTION PERFORMANCE WITH DIFFERENT PROFILE CONTENTS

We also considered using only logo of a site as its profile content, hypothesizing that this would help catching some of the "please fix your account" phishing sites that reuse logo but do not imitate the whole appearance of a legitimate site. Using SIFT algorithm, we can detect 82.73% of the phishing sites in this case. Logos that SIFT cannot detect contained either more elements than the actual logo or consisted of part of the actual logo. These types of logos can be detected by decreasing matching threshold or considering screenshots instead of individual images, however, this will increase the false positive rate. To detect such logos and screenshots successfully while keeping false positives low, SIFT will need to be paired with an image segmentation algorithm or bag of words model. We discuss such approaches in Section 6.

B. SIFT Robustness Evaluation:

To verify the robustness of SIFT we used the Stirmark benchmark [32], [33] to modify logos of sites by applying rotation, noise, convolution, and affine translation. Stirmark was developed as a benchmark to study the robustness of image watermarking methods. We found that with a 40% matching threshold, SIFT can detect 84% logos even after applying various affine transformations, Gaussian noise, rotation (up to 30 degrees), and scaling. As an image is converted to grayscale (.pgm) format before applying SIFT, discoloration of an image has no effect on matching performance. However, with this threshold, SIFT fails to detect images that are rotated more than 30 degrees.

C. Performance Analysis

Computation time is a crucial aspect for any online anti-phishing approach, as users are unlikely to tolerate high performance penalties. Profile matching requires two sets of operations: keyword matching and image matching. During keyword matching, each site's HTML documents is searched for all the selected keywords. For keyword searching we used Knuth-Morris-Pratt string search algorithm which is linear to the length of the HTML document and keyword. Image matching with SIFT includes two tasks: creation of keypoints descriptors from an image and matching of the keypoints. SIFT keypoint generation is fast, as claimed in Lowe's work [28] that 1000 SIFT keys generation requires less than 1 second of computation time. In our case, number of SIFT keypoints for an image is between 39 to 127. Image matching using the SIFT keys is calculating Euclidean distance between the keypoints, which is linear to the number of keypoints in the images.

Using image matching along with keyword matching gives high accuracy. But when keyword matching fails, a candidate site is compared against all the profiled logos. This recommended approach takes about 7 to 17 seconds on average to compare a site against all the profiles. We experimented with SURF (Speeded Up Robust Features) [34], but it did not improve the time performance as the main performance bottleneck depends on the number of stored profiles in the system.

However, our approach would be useful in "bulk mode" to large organizations that seek to protect users from phishing sites (such as email providers or system administrators) or find phishing sites for the purpose of improving blacklisting or issuing takedowns. 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 there is reason to believe our research implementation can be optimized much further and better performance numbers obtained.

VII. DISCUSSION

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 image or text matching algorithm used) is likely to improve the ability to defend against phishing attacks in the long run.

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. In the current implementation of PhishZoo, phishing detection cannot be avoided using resized images and logos, and restructuring HTML files. If attackers find ways of evading PhishZoo's detection mechanisms, their attacks can be incorporated into PhishZoo's detection model. Different variation of logos used by the phishers can be added to the system to improve SIFT's matching accuracy.

PhishZoo's current image matching algorithm fails to find matching in cases where the logos are rotated (more than 30 degrees), combined with other site elements or cropped. These changes can be detected by decreasing the matching threshold, but with an unacceptable increase in the false positive rate. An attacker can manipulate the combination of image and HTML files in attack pages to produce such failures. The only way to prevent such attacks is to match against screenshots of rendered pages. However, as Figure shows, using SIFT directly on screenshots reduces accuracy and dramatically increases false positives.

However, PhishZoo's approach can be extended and made more sophisticated. To actually "detect sites by looking at them" (their screenshots, that is) will require scene analysis or image segmentation—as mentioned in the introduction—and matching—as studied in this work. Both the performance and accuracy of PhishZoo's matching algorithm can likely be improved by using an image segmentation algorithm to preprocess images to be matched into regions [35]. We also plan to investigate using a bag-of-words model that uses probabilitistic latent semantic analysis (pLSA) [36] and latent dirichlet allocation (LDA) [37] to extract coherent regions within images. These methods have been been applied in visual domain to solve the scene analysis problem (distinguishing objects within a scene) successfully [38], [39]. Our results showing SIFT's success in detecting logos that are correctly cropped (even when scaled or distorted) lead us to believe that applying these techniques to the images will result in matching algorithms that are difficult for attackers to fool.

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. This understanding can be used to improve vision algorithms and scene analysis for a wide variety of applications.

If these techniques succeed, but reduce the efficiency of PhishZoo, it can be run offline (on email links) or by intermediaries.

VIII. Conclusions and Future Work

This paper presents the first step in a new approach of web-phishing detection using vision techniques. We presents results that defeat the vast majority of current attacks, but we believe there are many ways possible to improve these results. Our image matching approach fails on images whose contents do not match up well with the compared logo image. Preprocessing the images or screenshots using a bag-of-words model or image segmentation algorithm will lead to higher accuracy and protection against more sophisticated attacks.

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 (96.10% accuracy) against current phishing attacks and will identify new and targeted phishing sites where blacklisting-based toolbars fail. We also present a faster method based on HTML parsing that can be used to detect unsophisticated attacks in real time. 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.

References

[1] A.-P. W. Group, "Global phishing survey: Domain name use and trends in 2h2010," http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_2H2010.pdf.

[2] A. P. W. Group, "Phishing activity trends report," 2009, http://www.antiphishing.org/reports/apwg_report_Q4_2009.pdf.

[3] S. Egelman, L. Cranor, and J. Hong, "You've been warned: An empirical study of the effectiveness of web browser phishing warnings." in *CHI '08: Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 2008.

[4] S. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "The emperor's new security indicators," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2007.

[5] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *CHI '06: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2006.

[6] J. Downs, M. Holbrook, and L. Cranor, "Decision strategies and susceptibility to phishing," in *Proceedings of the 2006 Symposium On Usable Privacy and Security*, 2006.

[7] M. J. et al., "What instills trust? a qualitative study of phishing," in *Proceeding of first Int'l Workshop on Usable Security, Springer-Verlag*, 2007.

[8] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *WWW2007: Proceedings of the 16th International World Wide Web Conference*, 2007.

[9] P. Kumaraguru, Y. Rhee, A. Acquisti, L. F. Cranor, J. Hong, and E. Nunge, "Protecting people from phishing: The design and evaluation of an embedded training email system," in *CHI2007: Proceedings of the Conference on Human Factors in Computing Systems*, 2007.

[10] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Identifying suspicious urls: An application of large-scale online learning," in *ICML '09: Proceedings of the International Conference on Machine Learning*, 2009, pp. 681–688.

[11] T. Moore and R. Clayton, "Examining the impact of website take-down on phishing," in *Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit*. ACM, 2007, pp. 1–13.

[12] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Client-side defense against web-based identity theft," in *NDSS '04: Proceedings of the 11th Annual Network and Distributed System Security Symposium, San Diego*, 2005.

[13] A. Herzberg and A. Gbara, "Security and identification indicators for browsers against spoofing and phishing attacks," Cryptology ePrint Archive, Report 2004/155, 2004, http://eprint.iacr.org/.

[14] W. Inc., "Waterken yurl trust management for humans," http://www.waterken.com/dev/YURL/Name/.

[15] R. Dhamija and J. Tygar, "Protecting people from phishing: The design and evaluation of an embedded training email system," in *SOUPS 2005: Proceedings of the 2005 ACM Symposium on Usable Security and Privacy*, 2005.

[16] Z. Ye and S. Smith, "Trusted paths for browsers," in *Proceedings of the 11th Usenix Security Symposium*, 2002.

[17] Y. Zhang, J. Hong, and L. Cranor, "Cantina: A content based approach to detecting phishing web sites," in *Proceedings of the 16th International conference on World Wide Web*, 2007.

[18] C. Whittaker, B. Ryner, and M. Nazif, "Large-scale automatic classification of phishing pages," in *NDSS '10*, 2010.

[19] M. Aburrous, M. Hossain, F. Thabatah, and K. Dahal, "Intelligent phishing website detection system using fuzzy techniques," in *ICTTA 2008: Proceedings of Information and Communication Technologies: From Theory to Applications*, 2008.

[20] J. Zdziarski, W. Yang, and P. Judge, "Approaches to phishing identification using match and probabilistic digital fingerprinting techniques," in *Spam Conference*, 2006.

[21] S. Afroz and R. Greenstadt, "Phishzoo: An automated web phishing detection approach based on profiling and fuzzy matching," Technical Report DU-CS-09-03, Drexel University, Tech. Rep., 2009.

[22] K.-T. Chen, J.-Y. Chen, C.-R. Huang, and C.-S. Chen, "Fighting phishing with discriminative keypoint features," *IEEE Internet Computing*, vol. 13, no. 3, pp. 56–63, 2009.

[23] A. Y. Fu, L. Wenyin, and X. Deng, "Detecting phishing web pages with visual similarity assessment based on earth mover's distance (emd)," *IEEE Trans. Dependable Secur. Comput.*, vol. 3, no. 4, pp. 301–311, 2006.

[24] W. Liu, X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," *IEEE Internet Computing*, vol. 10, no. 2, pp. 58–65, 2006.

[25] T. Ronda, S. Saroiu, and A. Wolman, "Itrustpage: a user-assisted anti-phishing tool," in *Eurosys '08: Proceedings of the 3rd ACM SIGOPS/EuroSys European Conference on Computer Systems 2008*. New York, NY, USA: ACM, 2008, pp. 261–272.

[26] N. Provos, D. McNamee, P. Mavrommatis, K. Wang, and N. Modadugu, "The ghost in the browser analysis of web-based malware," in *HotBots'07: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*. USENIX Association, 2007, pp. 4–4.

[27] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in *DIM '08: Proceedings of the 4th ACM workshop on Digital identity management*. New York, NY, USA: ACM, 2008, pp. 51–60.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*. Washington, DC, USA: IEEE Computer Society, 1999, p. 1150.

[29] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615—1630, 2005.

[30] P. Fogg, G. L. Peterson, and M. Veth, "Image background matching for identifying suspects," in *IFIP Int. Conf. Digital Forensics*, 2008, pp. 307–321.

[31] T. Moore and R. Clayton, "The impact of incentives on notice and takedown," in *Proceedings of the Seventh Workshop on the Economics of Information Security*, 2008.

[32] M. G. K. Fabien A. P. Petitcolas, Ross J. Anderson, "Attacks on copyright marking systems," in *David Aucsmith (Ed), Information Hiding, Second International Workshop, IH98, Portland, Oregon, U.S.A.*, 2009.

[33] F. A. P. Petitcolas, "Watermarking schemes evaluation," in *I.E.E.E. Signal Processing, vol. 17, no. 5, pp. 5864*, 2000.

[34] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Computer Vision and Image Understanding (CVIU),Vol. 110, No. 3*, 2008, pp. 346–359.

[35] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *ACCV*, 2009.

[36] T. Hofmann, "Probabalistic latent semantic analysis," in *UAI*, 1999.

[37] D. Blei and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993—1022, January 2003.

[38] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR*, 2005.

[39] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering object categories in image collections," in *Int'l Conf. Computer Vision*, 2005.