GoldPhish: Using Images for Content-Based Phishing Analysis

Matthew Dunlop, Stephen Groat, and David Shelly
Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, USA
Email: {dunlop,sgroat,dashelly}@vt.edu

***Abstract*—Phishing attacks continue to plague users as attackers develop new ways to fool users into submitting personal information to fraudulent sites. Many schemes claim to protect against phishing sites. Unfortunately, most do not protect against zero-day phishing sites. Those schemes that do allege to provide zero-day protection, often incorrectly label both phishing and legitimate sites. We propose a scheme that protects against zero-day phishing attacks with high accuracy. Our approach captures an image of a page, uses optical character recognition to convert the image to text, then leverages the Google PageRank algorithm to help render a decision on the validity of the site. After testing our tool on 100 legitimate sites and 100 phishing sites, we accurately reported 100% of legitimate sites and 98% of phishing sites.**

***Index Terms*—Anti-phishing, OCR, Toolbar, Zero-day**

I. Introduction

Attacks that exploit human vulnerabilities have been on the rise in recent years [22]. Some of the most common attacks that fall into this category are phishing attacks. Phishing attacks generally use emails in an attempt to lure unsuspecting users into entering personal information such as credit card numbers or bank account numbers into fake web sites. The fake web sites are designed to look exactly like the authentic web site. Many times even the Uniform Resource Locater (URL) is similar to the authentic site's URL. Studies have shown that even the most computer-savvy users will fall victim to phishing sites [3]. In the first half of 2009 alone, there were 30,131 unique domain names conducting phishing attacks [17]. A report from late 2007 attributed the loss of more than $3 billion to phishing attacks. The same report shows that 3.6 million people lost money to phishing attacks [9].

There have been many tools developed to combat phishing attacks. Most anti-phishing methodologies in use today take advantage of databases that produce a blacklist of known phishing sites [5], [10], [14]. There are a number of disadvantages to this approach. First, this approach relies upon a complete database of all known phishing sites. In that respect, the anti-phishing tool is only as good as the completeness of its database. This is compounded by the fact that the average phishing site is active for only a couple of days, some only for a few hours [11]. Second, this approach does not protect against zero-day phishing attacks; that is, new phishing attacks that the community is unaware of. On average, 82 new phishing sites pop up every day.

We propose an approach that solves the problems that database techniques face by detecting zero-day phishing attacks. Our tool, called GoldPhish, uses a browser plug-in to detect and report phishing sites. We do this by using optical character recognition (OCR) to read the text from an image of the page (specifically from the company logo), grabbing the top ranked domains from a search engine, and comparing them with the current web site. The strength of our tool lies in the user's ability to recognize well-known company logos. A phishing site cannot change a well-known company logo without the phishing target noticing.

The remainder of this paper is organized as follows: Section II briefly surveys other anti-phishing approaches. We describe the detailed design of GoldPhish in Section III. Section IV outlines the conditions under which GoldPhish was tested. We analyze the effectiveness of our tool in Section V and in Section VI we conclude.

II. Related Work

Phishing detection algorithms can be roughly classified into two categories. The first category includes methodologies that use lists to determine phishing sites. The second category uses some sort of heuristic about the site to classify it.

A. List-Based

List-based anti-phishing approaches are widely used today. Their two main strengths are simplicity and speed. Classifying a site as phishing or trusted is a simple database lookup. What these approaches lack is the ability to detect zero-day phishing sites. List-based approaches can be further broken down into blacklist and whitelist.

1) Blacklist: Blacklists are used by most Internet browsers to detect phishing sites. Examples include Internet Explorer [10], and Firefox [5]. The blacklist approach keeps a database of all known phishing sites. Before navigating to a site, the browser checks its database to see if the requested URL is recognized as a phishing site. A drawback of using a blacklist approach is that it depends on the completeness of the blacklist. This has a great deal to do with the source of the list. It also takes time for a phishing site to be added to a blacklist database once it is discovered. Some sources, such as PhishTank [14], provide extensive databases, but most phishing sites become disabled before ever getting into a blacklist database [11]. Another issue is that blacklists cannot detect targeted phishing attacks (spearphishing [7]) since they are aimed at an individual or a small group. The main shortcoming of blacklists is that they cannot detect zero-day

978-0-7695-4023-8/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIMP.2010.24

attacks. Our goal is to build a tool that can work independently, or in conjunction with, blacklists to detect zero-day attacks.

2) Whitelist: Whitelists are less common than blacklists and work off the idea that a site must be explicitly trusted before access can be granted. The basic idea is that the user builds a list of trusted sites that he/she accesses on a regular basis. If the user attempts to navigate to a site that is not in the trusted list, he/she is either blocked from the site (static implementation) or prompted to add the site to the trusted list (dynamic implementation).

The underlying problem with these approaches [1], [2], [6] is that the majority of sites users navigates to are new. In a static implementation, users will be blocked from unvisited sites until they manually add them to their whitelist. It is not difficult to imagine that users will quickly become annoyed and disable this feature. Dynamic implementations will not fair much better. Each time the user navigates to a new site, he/she will be prompted to add the site to the trusted list. Initially, users may carefully consider whether to add the site. However, over time, users will become complacent and automatically add the site in question to the trusted list (if they do not simply disable the tool) [20]. Even if users are diligent about examining sites before they trust them, some users will still add phishing sites anyway. After all, if a phishing site is good enough to convince users to input sensitive information, it is reasonable to expect it can convince them to add it to their trusted lists.

B. Heuristic-Based

Heuristic-based approaches check one or more characteristics of a site to detect phishing rather than look in a list. These characteristics can be the uniform resource locater (URL), the hypertext markup language (HTML) code, or the page content itself. Most approaches will then use machine learning algorithms to make a judgment about the validity of a site. The main strength of heuristic-based approaches is their ability to detect zero-day phishing attacks. GoldPhish falls into this category.

Ludl et al. [8] proposed a heuristics-based approach that used 18 different heuristics to classify a page as safe or phishing. Most of the heuristics were targeted at the HTML source code while two considered the content of the URL. This approach achieved a 16.9% false negative rate and a 0.4% false positive rate.

A technique by Garera et al. [4] uses the composition of URLs to identify phishing sites. The authors combine several different heuristics as well as Google PageRank [12] to determine if a URL is legitimate or phishing. The idea is that phishing sites are new to the web and will not rank very high, while established web sites will have high rankings. Their results demonstrate a 4.2% false negative rate and 1.2% false positive rate.

Two techniques use keyword-retrieval from selected document object model (DOM) properties. Pan et al. [13] propose that a web site's true identity is contained within its DOM properties. By extracting identity information from the DOM properties, such as title, description, copyright, etc., they hypothesize that they can differentiate legitimate sites from phishing sites. Their false positive rate of 29% and false negative rate of 12% indicate this hypothesis might be flawed. Xiang et al. [21] propose a hybrid approach that combines identity-based detection from DOM properties with keyword-retrieval. A key improvement of this scheme over Pan et al. is that they use natural language processing techniques to identify brands. With this approach, they were able to achieve a false negative rate between 6.69% and 9.94% and a false positive rate between 1.95% and 2.26%.

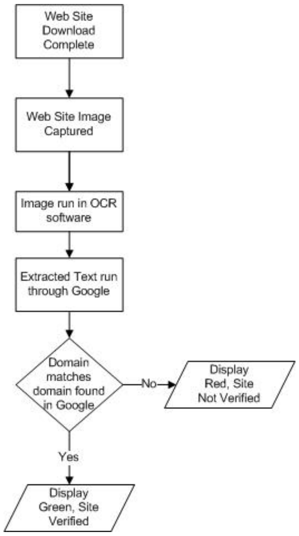

Web Site Download Complete
Web Site Image Captured
Image run in OCR software
Extracted Text run through Google
Domain matches domain found in Google
No
Display Red, Site Not Verified
Yes
Display Green, Site Verified

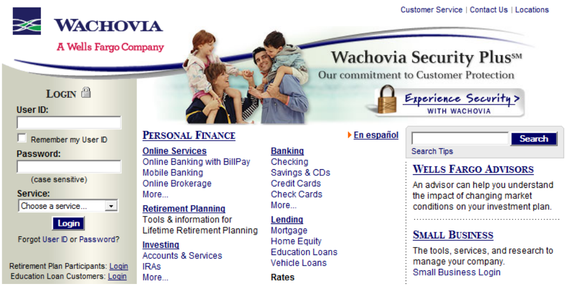
Fig. 1. Flowchart of GoldPhish Design

Zhang et al. proposed a heuristics-based approach called *Cantina* [23]. Cantina primarily uses a lexical signature heuristic based off of the TF-IDF (term frequency/inverse document frequency) algorithm [18] to detect phishing sites. The authors combine this heuristic with seven other heuristics, such as domain age and suspicious links, to achieve more accurate results. During testing, Cantina achieved a 89% true positive rate (11% false negative) and a 1% false positive rate.

All of the aforementioned approaches are primarily text-based. The problem with text-based approaches is that they can easily be gamed. For example a phishing site can be created using images instead of text. Alternately, a phishing site may contain text matching the background color of the site (invisible text). GoldPhish is robust against these weaknesses in that it examines web sites as images as discussed in Section III.

III. Design

In order to design a tool that protects against zero-day phishing attacks, a dynamic and adaptive approach needs to be utilized. Our approach utilizes the fact that trusted organizations have easily recognizable logos on their websites. These logos are brand name images that are capable of being matched by an Internet search engine. Our approach succeeds where other



Customer Service | Contact Us | Locations

WACHOVIA
A Wells Fargo Company

Wachovia Security Plus℠
Our commitment to Customer Protection

Experience Security >
WITH WACHOVIA

LOGIN
User ID:
□ Remember my User ID
Password:
(case sensitive)
Service: Choose a service...
Login
Forgot User ID or Password?

Retirement Plan Participants: Login
Education Loan Customers: Login

PERSONAL FINANCE
En español

Online Services
Online Banking with BillPay
Mobile Banking
Online Brokerage
More...

Retirement Planning
Tools & information for Lifetime Retirement Planning

Investing
Accounts & Services
IRAs
More...

Banking
Checking
Savings & CDs
Credit Cards
Check Cards
More...

Lending
Mortgage
Home Equity
Education Loans
Vehicle Loans
Rates

Search
Search Tips

WELLS FARGO ADVISORS
An advisor can help you understand the impact of changing market conditions on your investment plan.

SMALL BUSINESS
The tools, services, and research to manage your company.
Small Business Login

Fig. 2. Sample screenshot captured by GoldPhish

text-based approaches fail for two reasons. First, it accounts for the case where a web page may be composed entirely of images that can not serve as input to a search engine. Second, it accounts for sites that use invisible text designed to fool search engines. GoldPhish utilizes Google as the search engine because of its highly accurate PageRank [12] mechanism. This mechanism gives higher rank to well-established web sites. Since most phishing sites are usually only active for less than a few days, it is unlikely they will achieve a very high rank. Our current implementation is only capable of rendering websites written in English since we are using an English-based OCR tool and English-based Internet search engine. Our tool was designed using the Simple Plug-in Creator for Internet Explorer and is compatible with Microsoft Windows and Internet Explorer.

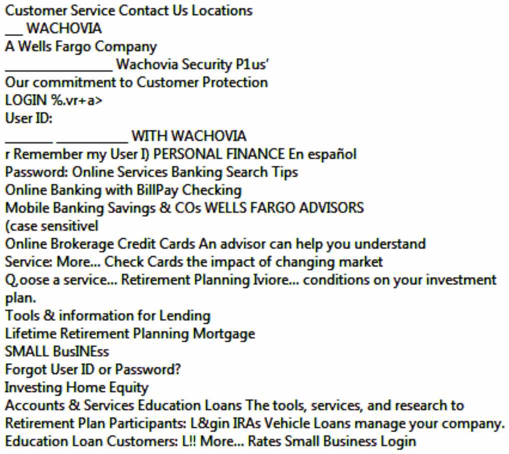
Our design approach can be broken down into three major steps. The first step is to capture an image of the current website in the user's web browser. The second step is to use optical character recognition techniques to convert the captured image into computer readable text. The third step is to input the converted text into a search engine to retrieve results. Fig. 1 gives a general overview of the design of GoldPhish.

A. Image Capturing

Image Capturing is the first step in our design approach. When the user visits a new web page, GoldPhish takes a screen capture of the page. GoldPhish utilizes an internal web browser for the screen shot. This method is favorable because it is independent of the resolution and display settings of the user's browser. This eliminates problems that could arise in the OCR software due to resolution that is either too high or too low. The screen capture is converted from a Bitmap image into a TIFF image and is saved into a temporary folder for OCR processing. The size of the page that is captured is variable. We chose a screen shot of 1200×400 pixels because this resolution was sufficient to achieve high accuracy while minimizing the time it took to run OCR software on a page. Our observation was that the most brand descriptive content of a page occurred in the top portion of the page. Fig. 2 shows an example screenshot of a typical web page.

B. Optical Character Recognition

Optical Character Recognition software processes the saved screen capture image in the second step of GoldPhish. The software that we use is Microsoft Office Document Imaging (MODI), which comes as part of Microsoft Office. Other commercially available OCR software reports more accurate OCR results and could potentially increase GoldPhish accuracy. It is reasonable to expect that as OCR tools improve, GoldPhish performance and accuracy will also improve.



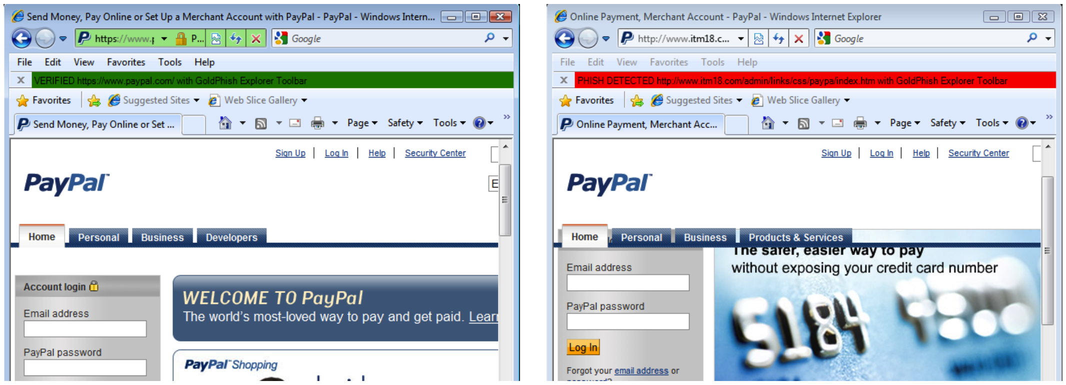
```
Customer Service Contact Us Locations
___ WACHOVIA
A Wells Fargo Company
________________ Wachovia Security P1us'
Our commitment to Customer Protection
LOGIN %.vr+a>
User ID:
_______ ___________ WITH WACHOVIA
r Remember my User I) PERSONAL FINANCE En español
Password: Online Services Banking Search Tips
Online Banking with BillPay Checking
Mobile Banking Savings & COs WELLS FARGO ADVISORS
(case sensitivel
Online Brokerage Credit Cards An advisor can help you understand
Service: More... Check Cards the impact of changing market
Q,oose a service... Retirement Planning Iviore... conditions on your investment
plan.
Tools & information for Lending
Lifetime Retirement Planning Mortgage
SMALL BusINEss
Forgot User ID or Password?
Investing Home Equity
Accounts & Services Education Loans The tools, services, and research to
Retirement Plan Participants: L&gin IRAs Vehicle Loans manage your company.
Education Loan Customers: L!! More... Rates Small Business Login
```

Fig. 3. Sample text extracted from Fig. 2 using the OCR tool

In addition to reading the text on a web page, the OCR software used in GoldPhish is capable of reading the text off of logos that exist on web pages. This is really where the strength of GoldPhish lies. Most common brands have well-known logos that include text unique to the company. These logos cannot easily be modified without alerting the phishing target. Additionally, these logos are typically located at the top of the page, which is why we only need to capture the top portion of the page. GoldPhish also works for pages without logos because, as mentioned, GoldPhish will capture and extract descriptive text at the top of the page. As the OCR software converts the image to text, a list of text entries is produced for submission to the chosen search engine (Google, in this case). A sample of text obtained from Fig. 2 using the OCR software can be seen in Fig. 3.

C. Google Search

The final step in the GoldPhish design approach is to submit the text that has been processed by the OCR tool to Google. The current implementation of GoldPhish only enters English-readable text into the Google Search API. This text is submitted line-by-line to preserve the layout of the page while also reducing terms lost due to Google's 50 query term limitation. We implemented the Google Search API to return only the first four results. The first four results are sufficient in our design because a legitimate website will generally come up within the first four results in a Google search due to its high PageRank. Meanwhile, a phishing site that has only been up for a short amount of time will have a low PageRank and will not be indexed in the top four search results. Even if the logo is not processed correctly by the OCR software or does not include text, the other text on the web page will be able


Send Money, Pay Online or Set Up a Merchant Account with PayPal - PayPal - Windows Intern...

VERIFIED https://www.paypal.com/ with GoldPhish Explorer Toolbar

Online Payment, Merchant Account - PayPal - Windows Internet Explorer

PHISH DETECTED http://www.itm18.com/admin/links/css/paypa/index.htm with GoldPhish Explorer Toolbar

(a) Sample of a legitimate site verified by GoldPhish

(b) Sample of a phishing site detected by GoldPhish

Fig. 4. Screenshots of the GoldPhish toolbar implementation

to be properly searched in Google.

The top-level and second-level domains [16] of the URL that the user is navigating to are parsed by GoldPhish and compared to the top-level and second-level domains of the first four results obtained in the Google Search. If no match is found within the first four Google results, the next line in the list of text entries is searched. Once a match is found, GoldPhish can verify the identity of the site and inform the user via the GoldPhish toolbar. A successful match can be seen in Fig. 4(a). If no Google matches are found after all text entries have been checked, GoldPhish states that it can not verify the identity of the site, and informs the user via the GoldPhish toolbar. An example of this can be seen in Fig. 4(b).

IV. Test Approach

In order to test our application's phishing detection rate, several known phishing sites and their corresponding legitimate sites were obtained. We used known phishing sites because it is difficult to test our application against zero-day phishing sites since zero-day phishing sites are new, unreported phishing sites. Our test approach is equivalent to testing against zero-day phishing sites because our design verifies the identity of all sites, regardless of whether or not the site is a previously known phishing site or a brand new phishing site. The web site PhishTank [14] was used to supply a list of known phishing sites. PhishTank is continuously updated by community users with the names of phishing sites and their links. Since these sites are reported as known phishing sites, their links are usually only active for a limited time before being disabled.

We found 100 active phishing sites to use for our initial test of GoldPhish. The group of phishing sites imitate a total of 18 legitimate sites that are well known e-commerce, banking, and social-networking sites. We also included 100 legitimate sites in our test. The 100 sites were a combination of the Internet's most popular websites [19], random web sites [15], and commonly phished websites [14]. The specific breakdown is 50 popular websites, 30 random web sites, and 20 commonly phished websites. When testing the known phishing sites, false negatives would exist if the phishing site is verified as a legitimate site, when in reality it is not. False positives do not exist for phishing sites because that would just result in a positive confirmation of a phishing site, which is proper operation. When testing the legitimate, verified sites, false positives would exist if the site is labeled as a phishing site, when it is actually legitimate. False negatives do not exist for verified sites because that would result in GoldPhish not marking a legitimate site as a phishing site, which is proper operation as well. Our test of GoldPhish only had two false negatives, resulting in an overall phishing detection rate of 98% for known phishing sites. A summarization of these results can be seen in Table I(a). GoldPhish had zero false positives, resulting in an overall verified detection rate of 100%. A summarization of these results can be seen in Table I(b).

V. Analysis of Results

We used the design described in Section III to test the accuracy of the GoldPhish tool. In this section, we describe how GoldPhish performed, as well as explain why some sites resulted in false negatives. We also discuss some of the limitations of GoldPhish and how to overcome them.

A. Performance

Accuracy. GoldPhish performed exceptionally well without the use of any complex machine learning or regression algorithms. Many other heuristics-based tools [4], [8], [13], [21], [23] rely on both techniques to make phishing decisions. GoldPhish does a simple screen capture, text translation, and Google lookup to determine a site's legitimacy. Our 0% false positive (FP) rate and our 2% false negative (FN) rate outperformed all other heuristic-based schemes (See Table II).

Speed. The time it takes an anti-phishing tool to make a decision about the legitimacy of a site is critical for usability. The majority of users will be unwilling to wait an excessive amount of time for the tool to render a decision. With that in mind, we tested the time it takes GoldPhish to return a decision on a page. We realize that processing time varies greatly depending on many factors. However, our goal is to provide users with some benchmark measurements.

TABLE I
NUMBER OF FALSE POSITIVES (FP) AND FALSE NEGATIVES (FN) FOR GOLDPHISH WHEN TESTED AGAINST LEGITIMATE AND KNOWN PHISHING SITES.

(a) Known Phishing Sites

Web Site	Phishing Site Tested	FN	Phishing Detection Rate
Amazon	9	1	89%
Bank West	2	0	100%
Barclays	2	0	100%
Bank of America	9	0	100%
CapitalOne	10	0	100%
CareerBuilder	4	0	100%
Citibank	7	0	100%
eBay	6	0	100%
Facebook	10	0	100%
Google	3	0	100%
HSBC	7	0	100%
MySpace	9	1	89%
RBC	2	0	100%
USBank	7	0	100%
Wachovia	2	0	100%
WalMart	2	0	100%
WaMu	2	0	100%
Wells Fargo	7	0	100%
Total:	**100**	**2**	**Avg: 98%**

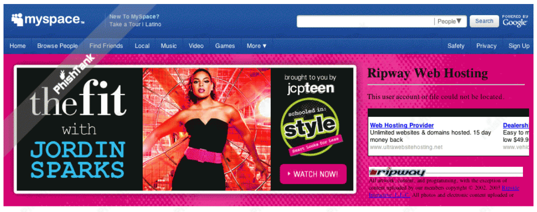
(b) Legitimate Sites

Web Site	Verified Site Tested	FP	Verified Detection Rate
Commonly Targeted	20	0	100%
Most Popular [19]	50	0	100%
Randomly Chosen	30	0	100%
Total:	**100**	**0**	**Avg: 100%**

TABLE II
COMPARISON OF FP/FN RATES FOR SELECTED HEURISTIC-BASED ANTI-PHISHING TOOLS

Anti-phishing Tool	FP rate	FN rate
GoldPhish	0%	2%
Ludl et al. [8]	0.4%	16.9%
Garera et al. [4]	1.2%	4.2%
Pan et al. [13]	29%	12%
Xiang et al. [21]	1.95-2.26%	6.69-9.94%
Cantina [23]	1%	11%

To test the time it takes GoldPhish verify a web page, we subtracted the time it takes to load a web page without GoldPhish from the time it takes to render a decision using GoldPhish. All of our testing was done using a Single Core 2.26 GHz, 1 GB RAM laptop running Windows Vista SP2. To ensure that network performance would not affect results, each page was locally cached. We ran 25 iterations using different web pages and found the mean, $\bar{x} = 4.31 seconds$, and a standard deviation, $\sigma = 0.22$. It was not possible for us to determine if our results were acceptable as compared with other heuristic-based algorithms because we were unable to find any other such timing results to use for comparison.



Ripway Web Hosting

This user account or file could not be located.

Web Hosting Provider
Unlimited websites & domains hosted. 15 day money back
www.ultrawebsitehosting.net

Dealersh
Easy to m
low $49.9
www.vehic

All artwork, content, and programming, with the exception of content uploaded by our members copyright © 2002, 2003 Ripside Interactive, L.L.C. All photos and electronic content uploaded or

Fig. 5. Screenshot of a false negative website. The ad claims to be from a web hosting company called Ripway Web Hosting. The OCR tool mistakenly classifies this site as valid because the Google query for Ripway matches the domain of its URL (http://h1.ripway.com/ThizzKidd/login.php).

B. False Positives and False Negatives

Although GoldPhish did not result in any false positives, it is possible for a false positive to occur. There is a limitation of the internal web browser used in GoldPhish. Two different methods of screenshots were analyzed when creating GoldPhish: using the web browser as seen by the user or internally rendering the webpage. Since different screen resolutions could cause variations in performance on different systems, the internal rendering method was chosen for consistency. The internal web browser, as implemented in C#, has a limitation of not running Javascript. Without the use of Javascript, some webpages do not render properly. Therefore, it is possible for there to be insufficient text in the screenshot to search.

The two FN that occurred during testing were due to free hosting ads placed on phishing websites. In order to generate revenue, many legitimate, free hosting companies automatically place ads of hosted webpages. When placed close to the top of pages, these ads are captured in the screenshot. Once the image is run through the OCR engine, the text from the ad is sent to Google for comparison. If the ad points to the same domain as the phishing site, a false negative occurs. While hosting companies attempt to remove phishing sites from using their ads, detection may still take some time. Banking and other sites not generating revenue from ads are not as susceptible to these attacks since they do not generally use ads on their pages. Other sites, such as MySpace, frequently use ads (see Fig. 5).

C. Limitations of GoldPhish

Due to the processing requirements of the OCR procedure combined with the Google lookup, GoldPhish delays the rendering of a webpage. While delay will be unacceptable for some users, others wanting zero-day protection will be willing to wait the few extra seconds for webpages to render. Regardless, processor speed will continue to increase and OCR software will improve; making delays negligible.

Similar to Cantina [23], GoldPhish is also vulnerable to attacks on Google's PageRank algorithm and Google's search service. An attack on Google's PageRank algorithm could improperly advance a phishing site in Google's search results and possibly provide the site as a valid option to GoldPhish. This is unlikely though since elevating a site's PageRank takes time and most phishing sites have short lifespans. Denial of service (DoS) attacks could also prevent Google's search engine from returning results. However, a DoS attack is unlikely due to Google's size and currently implemented defenses against such attacks. In order to make GoldPhish less reliant on Google, multiple search engines could be implemented to prevented these attacks from being successful.

The textual content available on a page and the fonts used may limit the effectiveness of GoldPhish. Due to its reliance on the OCR image of the top of the page, GoldPhish is limited by the amount and style of text, logos, and images caught in the OCR image. Problems can arise if a web page does not include sufficient data such as text, logos, or images to verify the domain of a web site. Pages that do not include sufficient data, however, are usually entrance pages to the main page of a website. The main page would generally have sufficient data to verify the legitimacy of the web site.

VI. Conclusion & Future Work

Although GoldPhish achieved better accuracy than the other heuristic-based anti-phishing techniques mentioned previously in this paper, improvements can still be made. Future work for GoldPhish includes the integration of other anti-phishing techniques into our design. For example, phishing site blacklists can be incorporated into GoldPhish. Checking against blacklists saves time and could be the first line of defense. Further analysis using images could be conducted as a second measure to protect against zero-day phishing sites. Similarly, whitelists could be used to quickly verify well-known legitimate sites. We could also incorporate machine learning algorithms to remove ads from images prior to running the OCR tool. By doing this, we can reduce the number of false negatives that occur due to phishing site advertisements as illustrated in Fig. 5.

As previously mentioned in Section III-B, some commercially available OCR software is available that would produce faster, more accurate results. As this technology becomes freely available, it will be able to be integrated in GoldPhish and improve its performance. The incorporation of additional compatible languages, web browsers, and operating systems as noted in Section V-C, are areas of future work as well. Also, future implementations of GoldPhish will include verification using more than one search engine and compatibility with Flash and Javascript.

Despite some limitations, GoldPhish is a powerful tool that is capable of detecting zero-day phishing sites. GoldPhish does this by utilizing a browser plug-in to detect and report phishing sites. Our plug-in is based on a dynamic and adaptive approach which relies on image capturing, optical character recognition, and Internet search engine results. We tested our application against 100 legitimate web sites and 100 known phishing web sites. Our testing resulted in an overall phishing detection rate of 98%, and an overall verified detection rate of 100%. Further analysis of our results allows us to identify some areas of future work that could improve performance and usability. After comparing our design with similar methodologies, we were able to conclude that GoldPhish is one of the most accurate tool bars available to defeat zero-day phishing attacks.

References

[1] BitDefender Anti-Phishing. Available at: http://content-down.bitdefender.com/windows/desktop/antiphishing/final/en/BitDefender_APFE_2009_Userguide_en.pdf accessed on 4 Nov 2009.

[2] 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*, pages 51–60, New York, NY, USA, 2008. ACM.

[3] 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*, pages 581–590, New York, NY, USA, 2006. ACM.

[4] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A framework for detection and measurement of phishing attacks. In *WORM '07: Proceedings of the 2007 ACM workshop on Recurring malcode*, pages 1–8, New York, NY, USA, 2007. ACM.

[5] Google safe browsing v2.2 protocol. Available at: http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_1H2009.pdf accessed on 26 Oct 2009, 2009.

[6] 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/.

[7] B. Kesler, H. Drinan, and N. Fontaine. News briefs. *IEEE Security and Privacy*, 4(2):8–13, 2006.

[8] C. Ludl, S. Mcallister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. In *DIMVA '07: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 20–39, Berlin, Heidelberg, 2007. Springer-Verlag.

[9] T. McCall. Gartner survey shows phishing attacks escalated in 2007; more than $3 billion lost to these attacks. Available at: http://www.gartner.com/it/page.jsp?id=565125, 2007.

[10] Microsoft's approach to anti-phishing. Available at: http://www.microsoft.com/mscorp/safety/technologies/antiphishing/vision.mspx accessed on 26 Oct 2009, 2007.

[11] T. Moore and R. Clayton. The impact of incentives on notice and take-down. In *Proceedings of the Seventh Workshop on Economics of Information Security (WEIS 2008)*, June 2008.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[13] Y. Pan and X. Ding. Anomaly based web phishing page detection. In *ACSAC '06: Proceedings of the 22nd Annual Computer Security Applications Conference*, pages 381–392, Washington, DC, USA, 2006. IEEE Computer Society.

[14] Phishtank. Available at: http://www.phishtank.com/ accessed on 26 Oct 2009.

[15] randomwebsite.com. Available at: http://www.randomwebsite.com/ accessed from 10-28 Oct 2009.

[16] J. Postel and J. Reynolds. Domain requirements. RFC 920, Oct. 1984.

[17] R. Rasmussen and G. Aaron. Global phishing survey: Trends and domain name use in 1h2009. Available at: http://www.antiphishing.org/reports/APWG_GlobalPhishingSurvey_1H2009.pdf, October 2009.

[18] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

[19] Web100.com. Available at: http://www.web100.com/web-100 accessed on 5 Nov 2009.

[20] M. Wu, R. C. Miller, and S. L. Garfinkel. Do security toolbars actually prevent phishing attacks? In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 601–610, New York, NY, USA, 2006. ACM.

[21] G. Xiang and J. I. Hong. A hybrid phish detection approach by identity discovery and keywords retrieval. In *WWW '09: Proceedings of the 18th international conference on World wide web*, pages 571–580, New York, NY, USA, 2009. ACM.

[22] Y. Zhang, S. Egelman, L. Cranor, and J. Hong. Phinding Phish: Evaluating anti-phishing tools. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium*, Feb. 2007.

[23] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing web sites. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 639–648, New York, NY, USA, 2007. ACM.