Automating Phishing Website Identification through Deep MD5 Matching

Brad Wardman
Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, Alabama

Gary Warner
Computer Forensics
University of Alabama at Birmingham
Birmingham, Alabama

***Abstract*—The timeliness of Phishing Incident Response is hindered by the need for human verification of whether suspicious URLs are actually phishing sites. This paper presents a method for automating the determination, and demonstrates the effectiveness of this method in reducing the number of suspicious URLs that need human review through a method of comparing new URLs and their associated web content with previously archived content of confirmed phishing sites. The results can be used to automate shutdown requests, to supplement traditional "URL black list" toolbars allowing blocking of previously unreported URLs, or to indicate dominant phishing site patterns which can be used to prioritize limited investigative resources.**

Keywords-Phishing, Detection, Campaigns, Brand-matching

I. INTRODUCTION

Phishing is an area of security which has received a high level of attention because of the potential for financial loss and identity theft. A phishing attack consists of two parts, the creation of a counterfeit website, usually imitating the login page of a financial institution or other online service, and the distribution of spam email messages asking people to visit the site and share their login credentials. Security professionals seek to diminish the impact of phishing by user education, filtering of spammed phishing emails, and the use of anti-phishing toolbars, all designed to prevent users from accessing the phishing page where they will be asked for personally identifiable information which will be transmitted to criminals. Despite their efforts, a large number of phishing sites are created each year. The Anti-Phishing Working Group reports that for each month in 2007, between 23,415 and 38,514 unique phishing sites were reported to them [1].

Another means of preventing the impact of phishing is to terminate the phishing websites themselves. This process is usually undertaken by employees of the offended brands, or by specialty anti-phishing companies and volunteer groups. This process usually involves receiving reports of phishing from customers or specialty brand protection vendors, confirming that the sites are indeed phishing sites, and then identifying appropriate parties, such as webmasters, web hosting companies, or domain registrars, who can assist in removing the offending content from the Internet. One of the greatest delaying factors is that before action can be taken to remove the phishing site, human verification that the site is actually a phishing site must first be asserted.

A. Human Verification

One popular volunteer effort, PhishTank, illustrates the problem that is faced. According to their April 2008 statistics, 22,285 suspected phishing sites were submitted to their system. On the average, it took 10 hours and 14 minutes before each site had been identified as "valid" or "invalid". 985 of the sites were determined to be "Invalid Phishes" [2]. Another volunteer effort, the CastleCops PIRT Team (Phishing Incident Reporting & Termination), automatically fetches each submitted URL, and calculates an MD5 hash of the fetched phishing site's main HTML page. The Digital PhishNet project does the same. Although the fetch of the website is automated at PhishTank, PIRT, and DPN, each of the sites requires that humans identify the victim brand which the phishing site is imitating, and in the case of the first two, asks human volunteers to confirm whether the submitted page is indeed a phishing site. A problem with this approach is the demonstrated inability of humans to tell the difference between phishing and legitimate websites [6].

If an "invalid phish" had been reported for termination, the credibility of the reporter would have been placed at risk, and it is possible that a non-phishing site could have been wrongfully terminated, leading to possible legal liability. In most bank security teams, the time is much less than the 10 hours given above, however, all of the anti-phishing teams interviewed for this research still used human verification, and often had a lengthy queue waiting to be verified.

It has been similarly demonstrated that the effectiveness of anti-phishing toolbars increases with the age of the phishing site, and that performance at "time zero" of new phishing sites is quite poor [4]. Most anti-phishing toolbars still rely on the submission of phishing URLs to a URL blacklist for comparison. The notable exception to this is SpoofGuard,

which uses a complex heuristic analysis to obtain very good results [5]. In the past three months, the authors have submitted more than 800 valid phishing sites per month to a popular phishing toolbar that were not recognized at the time of submission by any major anti-phishing toolbars.

B. *Automatic Classification*

Much work has been done on the automatic classification of malware that has focused on the observed behavioral characteristics [3]. Phishing webpages do not display these behavioral characteristics. Others have focused on natural language processing comparisons, machine learning techniques, and content-based learning that are based on the presence or absence of certain key words [9][10][11][12]. This is the path that has been followed by many spam classification systems, and has been used as part of the heuristic solution in SpoofGuard, above. In many ways our problem is simpler. We wish to know whether the page we are currently reviewing is sufficiently similar to a page that has already been confirmed to be a phish; therefore, we could forego human interaction and label the submitted URL as a phish automatically. This labeling may benefit a spam-filtering program, an anti-phishing toolbar, or an incident response or investigative team.

C. *Hypothesis*

Our hypothesis is that by considering the MD5 values of additional files that make up the human experience of the webpage, including the .jpg and .gif files, as well as the .css and .js files, we may find large enough similarities to "confirm" a phish even when minor textual changes on the main HTML page would prevent an MD5 match on that page.

II. MOTIVATION

A. *Current Method*

Much has been written about the problems of preventing phishing fraud by informing the user when a page they are attempting to visit is actually a phishing page. Less has been written about the problems being faced in incident response and investigation. Our team receives in excess of 1 million potential phishing URLs each month which must be sorted, de-duplicated, confirmed, labeled, and referred for appropriate action. In more than 10,000 cases the team has reviewed and confirmed a phishing site that was unknown at that time to many of the anti-phishing toolbars being used. Team members have met with representatives from brand protection and incident response teams at many large financial and online institutions. The process is much the same everywhere, potential fraud URLs are reported from customers and vendors. These sets are reduced to unique URLs, sometimes using regular expressions or pattern matching to identify URLs which resolve to the same content. That list is then prepared in a queue, where the incident response staff manually review each site to determine whether it is committing fraud against a brand for which they are responsible. If the site is fraudulent and attacking a brand of interest, additional attributes of the site, such as whois information, the ASN or netblock of the hosting IP address, or the registrar used to register the site are determined. This information is then used to generate communication to parties who are in a position to stop the fraudulent website from resolving. Often this step has been highly automated, but the automated process cannot begin until the URL is retrieved from the work queue and verified. In response, a webmaster or webhosting company staff may "lock" or disable the hosting account, or change permissions to the offending content so that visitors do not experience the content. An ISP may temporarily block internet access for the computer containing the offending content. A registrar may remove name resolution services for the domain name, or may otherwise delete or disable the domain name.

The timeliness of the appropriate response is currently hindered by the delay introduced by the need for human verification of the potential offending content, which is often repeated multiple times by several different players in the process. A customer or vendor learns of the phish via a spam email message, and verifies the phish and reports it to their financial institution. The incident response team verifies the phish and either reports it to shutdown vendors, or begins to request termination from an appropriate webhost/ISP/registrar. Shutdown vendors, if used, also confirm the phish and report it to appropriate webhost/ISP/registrar contacts. The webhost/ISP/registrar staff then confirms the phish, and takes appropriate action to block or terminate the offending content.

It is clear that there would be benefit to having a trustworthy method for confirming phishing sites without the need of human intervention. Given a list of potential phishing URLs, our approach will correctly confirm and label with brand a significant percentage of phishing sites without the need of human intervention. Unknown sites will still be presented for human response, but once confirmed, these sites will be added to the comparison database. Thus allowing sites which meet the new pattern to also be automatically confirmed and labeled. The same technique could be applied earlier in the work queue, for example, to identifying phish from among an all sources spam feed.

III. PROPOSED METHOD

A. *Identifying a Phishing website*

Our experience has shown that very few phishing sites are crafted for a single use. Many phishers re-use the same files,

such as html, css, js, and image files, each time they create a phishing site to attack a particular brand. Particular successful counterfeit websites are packaged together in a "phishing kit", often called a "scam kit" by the criminals. These kits contain all of the necessary files required to easily create a fraudulent website. The kits are usually distributed as a zip or tar.gz archive, and may be used for creating a site on a newly registered fraud domain, or may be uploaded onto a compromised web server belonging to another party. Once the kit has been placed on the destination server, it is moved to the desired directory and the files are extracted from the archive, immediately becoming "live webpages". The phisher then notes the location of his new fraudulent website and begins to advertise this location in spam emails designed to attract victims to reveal their personally identifiable and financial information. Observation of many of these kits indicates that while certain configuration files such as those containing instructions on where to send the compromised credentials are often modified, the majority of the files in the kit may remain untouched across dozens or even thousands of uses. We use this knowledge to our advantage.

Our method is to first retrieve all the files in the phishing URL's local directory, as well as, all the files in the sub-directories. To safely download the files from the URL, we use the GNU Wget software package [7]. Wget uses HTTP, HTTPS, and FTP protocols. The files will be downloaded into the same directory structure as the files on the web server.

After downloading all the files, we calculate the MD5 checksums of each file with the executable md5deep [8]. The MD5 checksums are used as a similarity measure to other phishing URLs. While we are not aware of large collections of MD5s for the content pages of phishing sites, we are familiar with the MD5 database provided by the Digital PhishNet (DPN) to its members. Each of these MD5 values represents a 'main page' of a confirmed phishing site which has been labeled by brand. A full content sample of each site is available to DPN members online.

As our project proceeds, we will build our own database of MD5 checksums not just of the 'main pages', but also of each of the associated files of the sites. As we gather and label more data, we store the MD5 checksums in a database for comparisons. If the MD5 checksum of the potential phishing site matches one of the checksums in the DPN database, than we can label the URL as a confirmed phishing website. Each new phishing site we confirm is shared with the Digital PhishNet, the NetCraft toolbar, and the CastleCops PIRT project. Our phishing URLs comprise the majority of the content in the DPN database today.

B. Pattern Matching

Regardless of whether an exact match has been found by "main page" MD5 comparison, we calculate and store the MD5 values of the component pages that make up the phishing website under consideration. In this way a phishing website can be seen as a set of MD5 values, each representing one page of the remotely observable collection of phishing kit files. (Some files in a phishing kit, while present on the server, are not viewable via a remote browser-based connection.) Two websites can then be compared statistically by calculating the number of MD5 values found in common between the site in question and other sites in the database. We also look at statistics, such as the average number of matched files, total count of matched files, percentage of files matched, and number of sites that have at least one same file. These statistics can be used to associate a similarity measure between different websites.

C. Brand Matching

Part of our hypothesis is that the MD5 checksums can be utilized as a method of branding phishing URLs. Since the MD5 checksums in the Digital PhishNet database are labeled with brands, we can use those checksums as a data resource for comparison. When a URL in our list contains a MD5 checksum that is equal to a checksum in the database, we can deduce that the brand of the URL is the same as the file stored in the database. We can now add the URL, its files, their MD5 checksums, and brand to our data resource. When there is no match, someone will have to manually check on the brand of the phish and input it into our data resource. For non-exact matches, a pattern matching of the associated files will be used, and pages that exceed the threshold of similarity will be confirmed and branded. Once a previously unknown site has been manually confirmed and branded, it will become part of the database and will be considered as any other website.

IV. RESULTS

A. Datasets

For our purposes we draw on three data sets. The first is a set of 12,060 unique MD5 values, provided by the Digital PhishNet, which have already been labeled by anti-phishing professionals regarding the phish victim brand to which they belong. This data set consists of the "main HTML page" of the phish and was drawn from a collection of more than 46,000 confirmed phish for 335 separate brands, primarily banks, credit unions, and other online companies. By reviewing the first set, we find that while a large percentage of the phish against any particular brand come from a small set of URLs for that brand, many MD5 values remain unique, with several brands having more than 100 "unique" MD5 values for

their main HTML page, and some having as many as 700 or even 2000 unique URLs in the data. The high number of unique values is one of the challenges that lead to this current research.

The second dataset is our recently fetched phishing site data. We receive phishing reports from many sources, including cooperating anti-spam vendors, our own spam traps, spam submitted from other organizations, and financial institutions who support our research. For purposes of this paper, we consider a set of 1,030 phishing URLs fetched on nine days scattered over a one month period. We will refer to this dataset as the multiple brand dataset. We retrieved, using Wget, the main page which was advertised in a spam message, and all files hosted on that site necessary to display the complete page in a browser. These sites provided 7,156 files, yielding 2,575 unique MD5 values. Graphics hosted on a separate site were not retrieved.

The third dataset consists of fetched phishing site data for 236 URLs spanning a three day phishing period. These URLs were provided for analysis by a single victim brand. These sites provided 1,497 files, yielding 658 unique MD5 values.

B. Defeating Obfuscation

In reviewing unique phishing sites, those which have no MD5 match in the existing dataset of 'main page' URLs, it can be observed that the phishers are using techniques to deliberately obfuscate their data to prevent simple MD5 matching from being successful. As one example, consider a common eBay phishing scheme:

```
marcsevigny.dyndns.org\ebayISAPII.dll\index.php@cmd=Validate
        &54fjcyhaagnfgvebdxgelob3exzt4rl9orvpm1343ingormpl4
marcsevigny.dyndns.org\ebayISAPII.dll\index.php@cmd=Validate
        &819wdcjhaagrtadcitzemrpspcfheaan9gbe0db62nosnu733hr
marcsevigny.dyndns.org\ebayISAPII.dll\index.php@cmd=Validate
        &8n002m3cshja7n6hxensyedqbsanndc5d6yc3you072hb8dqsv
```

Strings have been changed to protect privacy of submitters, which can be decoded to reveal email addresses.

It is clear that these URLs each go to the same destination website, however the resulting webpage for each of these yields a unique MD5 value. In this case the page being fetched is not static html, but rather a PHP program. Part of the program creates random strings throughout the resulting web page source code so that each instance of the website generates a unique MD5 value. For instance, the href tag for a forgotten userid:

```
<a href="?cmd=Validate&v4bdujgudetxohc9gdet50xdm9303qwqbq2hppjr4l1atw">I forgot my user ID</a>
<a href="?cmd=Validate&myjkafe7lyu5v1zoxkfh549evfwfkofmtrp8enbwptexk9i">I forgot my user ID</a>
<a href="?cmd=Validate&y8tj4jssjl8ov2en968o7ly9iary9b4sxadvynqe7er4wyj7z">I forgot my user ID</a>
```

In the example above we show that three "calls" to the same site on the same page yielded three different content pages, generating three separate MD5 values. This problem is magnified when multiple sites are taken into consideration. By considering the other files on this site, we are able to show a high correlation with other phishing sites. In this example:

spacer.gif	119 matches
s.gif	87 matches
imgCrnrO3.gif	44 matches
imgCrnrO4.gif	44 matches
imgpanelurgrey.gif	44 matches
imgpanelllgrey.gif	41 matches
imgpanellrgrey.gif	41 matches
imgpanelulgrey.gif	41 matches
logoEbay_x45.gif	34 matches
logoNewVeriSign_100x65.gif	30 matches
areaTitleDeployment_SSL_e5391us.css	3 matches
ebay-ns_e5391us.css	3 matches
signin_base_e5392us.js	3 matches
signinyukon_SSL_e5391us.css	3 matches

Table 1. Content files from defeating obfuscation example

In this case, "spacer.gif" and "s.gif", was an insignificant match which appeared on both related and unrelated phishing sites. The "imgpanel" set of graphics, representing the upper right, lower left, lower right, and upper left corners of a box, was found on 41 sites that were all phish for the same brand. The latter are exact matches for the graphics currently found on the live "ebay.com.au" website found on 30 additional sites, as is the VeriSign logo,  :

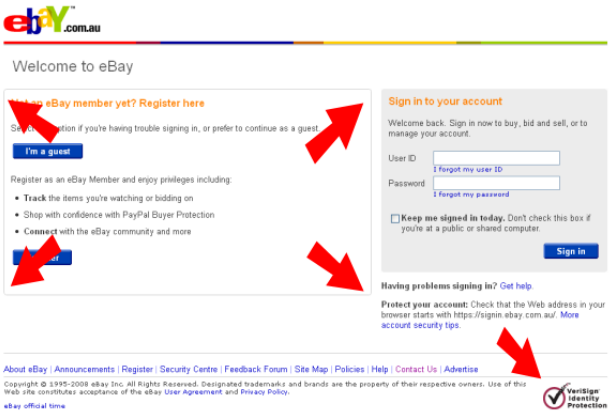

Figure 1. Image of the 4.2 example phishing website

In this example, 15 files were considered. The "main page" MD5 did not match any pattern because of the obfuscation by the phisher, but the other 14 MD5 values all provided matches.

The two most common files provided little value, would be ruled insignificant, the next 8 considered as a set provided strong evidence that this was indeed an eBay phish. The final four files, found on three sites each, are evidence of the exact same kit being used to create the three separate sites.

C. Analysis of organization's URLs

We found that out of the 236 single victim brand dataset URLs, 120 phishing websites contain at least a single file that matches other phishing website file sets, this means that 116 websites do not have any files matching other websites. 29 out of the 236 URLs were offline when the files were being downloaded. The results show that 66 of the URLs had 40 or more files that matched files of other websites. 79 of the URLs had files that matched at least 11 other URLs. The analysis showed there are three sites consisting of 64 files each, which are perfectly identical. Some of these 64 files are also matched with 29 other phishing sites to produce a total file count of 411 matched files. These 411 files are matched against 31 different websites. These three sites are the top sites with total matched files. Our results show that 70 out of the 236 URLs have html files with MD5 values that match the MD5 values stored in the Digital PhishNet's database. This means that 29.7% of phishing URLs can automatically be labeled and branded as a phish. The results illustrates that set matching has the potential of confirming 50 more URLs.

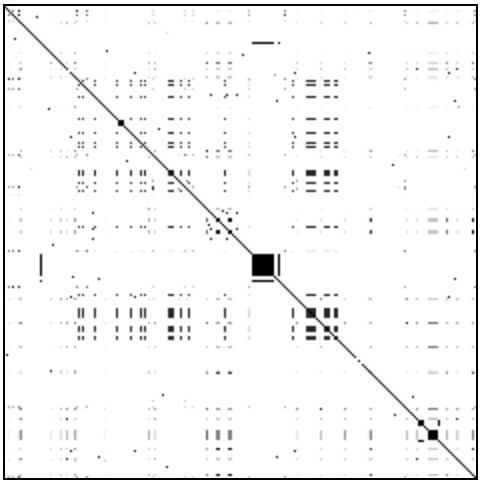

Figure 2. This is the single victim brand dataset's similarity matrix with pixel intensities representing percentage of similar files between phishing websites.

Figure 2 shows the similarity matrix for the single victim brand set of phishing URLs. The x and y axis consist of the phishing websites, in alphabetical order, making the matrix symmetric. The intensity of the pixel shading is determined by the percentage of files that are the same. Hence, the dark diagonal line is when x = y, which means both x and y is the same phishing URL and contain the same files. Observing Figure 2, we see a number of a few repeated patterns that create sets.

As an example, there are three sets of websites that all contain the exact same total amount of files matched as well as the exact same number of files downloaded from the URL. One set contains 11 URLs, another with 5 URLs, and the largest set contains 24 URLs. The set of 24 URLs, all contain 6 files in the website's directory. These URLs can be seen in Figure 2 as the scattered set of dark pixel streaks. The small black square residing on the diagonal represents the set of 11 URLs. All 11 URLs, which are similarly named and that is why they are next to each other in the matrix, contain one html file. Each html file is a direct match each of the other 10 phishing sites' html file. These three sets might be from different phishers using the same kit, or may be part of a phishing campaign. More analysis needs to be done in order to correctly label a group of websites as a phishing campaign.

D. Analysis of random list of URLs

Out of the list of 1030 random URLs, there are 628 URLs that matched a single file to at least one other website's file set. 296 of the websites contained a file that matches at least 26 other website file sets. The majority of the 296 websites matched more than one file from the other websites. Our results showed that there are seven phishing websites that all contain the largest total number of matched files. These seven sites have 1829 files that match 139 of the tested websites.

Of the 1030 URLs from the multiple brand dataset, 180 of the phishing websites were offline when downloading occurred. Our results show that 351 out of 1030 URLs have html files with MD5s that match the MD5 in the Digital PhishNet's database. That means 34.1% of the multiple brands dataset's URLs could automatically be labeled and branded as a phish. 302 of the 351 DPN matching websites matched files of other websites. Our results show evidence that there is a possibility of doubling the automatic labeling and branding of URLs through set matching.

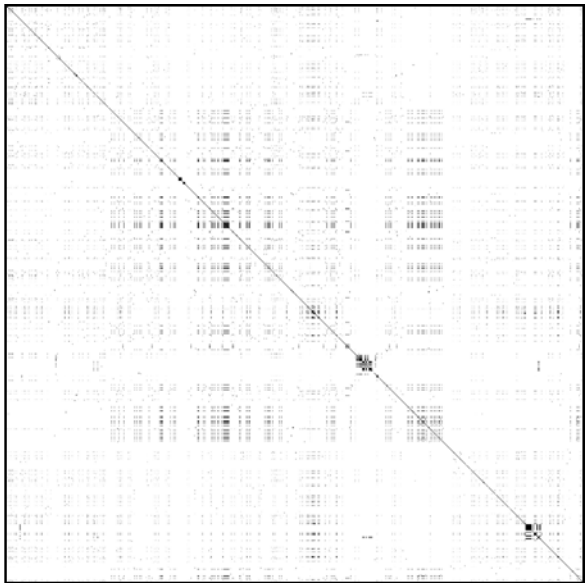

Figure 3. This is the multiple brand dataset's similarity matrix with pixel intensities representing percentage of similar files between phishing websites.

For this paper, we will take a look at two distinct sets of URLs that can be observed in Figure 3. One set containing 93 sites, matched files from exactly 139 or 140 other websites. When looking at Figure 3, you can see these phishing websites as the large amount of scattered pixels at the beginning and end of the rows and columns. This set also has some elements that are present in the middle of the similarity matrix. The pixel intensities are lightly shaded due to the fact that only a small percentage of the files in the phishing website's directory match the files in other phishing websites. This leads us to believe that there are probably very few similar files that link all of these sites together. This motivates our hypothesis.

The second set contains 86 URLs that match 88 different websites. Each URL in that set contains 6 files, except two which has 7 files. In Figure 3, we can distinctly find this set by looking at the compact streaks of dark pixels, which makeup the inner square. The fact that the pixels are dark means that a high percentage of files in both phishing sites match each other. We can infer that these sites are almost directly related by exact files and that these phishing websites were either posted by the same phisher or produced by the same phishing kit.

V. Future Work

A goal of ours is to use the presence of similar files within phishing websites to allow us to correctly label and brand other URLs as a phish. We did not fully prove this in our results, but did show progress towards that goal. Future work will include producing a similarity measure, through various statistics, that would allow us the ability to properly label and brand a phishing URL.

Our original goal was to help prioritize limited investigative and law enforcement resources by identifying groups of phishing sites so similar that it would be reasonable to assume a relationship between the criminal entities behind the group of sites. We think that we can identify these phishing campaigns by looking at the percentage of files that are exactly the same, as well as using other methodologies, such as matching spam emails which may further refine our knowledge about whether this is the "same" campaign or not. Details of hosting and registration of domains may also be helpful in properly clustering this data.

VI. Conclusion

We have shown that we are currently able to use MD5 matching to automatically label and brand around 30% of phishing websites using only the main HTML MD5. When the MD5 comparisons of other files are included, we demonstrate clear patterns among the sites represented in our datasets. We can distinctly see what URLs contain files that match each other. We can also see, through the pixel intensities, to what percent of files matched. We will next refine our similarity scores for frequently occurring sets of confirmed phishing files, such as .jpg, .gif, .js, and .css. This will allow a manually set label to propagate to all members of the set, greatly increasing our percentage of sites which can be automatically labeled and branded.

Acknowledgment

The authors are grateful to those organizations whose data helps us on a daily basis, including the Digital PhishNet, CastleCops, and NetCraft. We are also grateful to the un-named organizations who entrust their phishing URLs to our care. Special thanks to data visualization expert David O'Gwynn, for his contributions of Figures 2 and 3, and their underlying algorithms

References

[1] Anti-Phishing W orking G roup. P hishing A ctivity Trends, 2007. http://www.antiphishing.org/reports/apwg_report_dec_2007.pdf

[2] PhishTank. P hishTank A pril '08 stats. L earn to p rotect y ourself, your company. h ttp://www.phishtank.com/blog/2008/05/phishtank-april-08-stats-are-live/

[3] Bailey M., Oberheide J., Andersen J., Mao M., Jahanian F., and Nazario J. Automated Classification and Analysis of Internet Malware. Technical Report CS E-TR-530-07, D epartment of E lectrical E ngineering a nd Computer Science, University of Michigan, April 2007

[4] Cranor L., Egelman S., H ong J ., a nd Z hang Y . P hinding P hish: A n Evaluation o f A nti-Phishing T oolbars. C yLab T echnical R eport CMU-CyLab-06-018. November 2006.

[5] Chou N ., L edesma R ., T eraguchi Y., a nd Mitchell J .C. Client-side defense a gainst we b-based i dentity t heft. I n t he P roceedings o f t he Network and Distributed System Security Symposium, 2004.

[6] Dhamija R., Tygar J. D., and Hearst M. Why Phishing Works. In the Proceedings of the Conference on Human Factors in Computing Systems, 2006.

[7] Wget. http://www.gnu.org/software/wget/

[8] md5deep. http://md5deep.sourceforge.net/

[9] Pan Y., and Ding X. Anomaly Based Web Phishing Page Detection. In the Proceedings of the 22nd Annual Computer Security Applications Conference (ACSAC'06), 2006.

[10] Abu-Nimeh S., Nappa D., Nair S. A Comparison of Machine Learning Techniques for Phishing Detection. In the Proceedings of the APWG eCrime Researchers Summit, 2007.

[11] Zhang Y., Hong J., and Cranor L. CANTINA: A Content-Based Approach to Detecting Phishing Web Sites. In the Proceedings of the International World Wide Web Conference Committee (IW3C2), 2007.

[12] Fette I., Sadeh N., and Tomasic A. Learning to Detect Phishing Emails. ISRI Technical Report. CMU-ISRI-06-112, 2006. http://reportsarchive.adm.cs.cmu.edu/anon/isri2006/abstracts/06-112.html