Reeling in Big Phish with a Deep MD5 Net

Brad Wardman[1], **Gary Warner**[1,2], **Heather McCalley**[1],
Sarah Turner[2], **and Anthony Skjellum**[1]
University of Alabama at Birmingham
Affiliations: Computer and Information Sciences[1] and
Justice Sciences[2]
UBOB 402
1530 3rd Ave South
Birmingham, AL 35294-4562
(205) 934-8620
{bwardman, gar, saturner, hcarol, skjellum} @ uab.edu

ABSTRACT

Phishing continues to grow as phishers discover new exploits and attack vectors for hosting malicious content; the traditional response using takedowns and blacklists does not appear to impede phishers significantly. A handful of law enforcement projects — for example the FBI's Digital PhishNet and the Internet Crime and Complaint Center (ic3.gov) — have demonstrated that they can collect phishing data in substantial volumes, but these collections have not yet resulted in a significant decline in criminal phishing activity.

In this paper, a new system is demonstrated for prioritizing investigative resources to help reduce the time and effort expended examining this particular form of online criminal activity. This research presents a means to correlate phishing websites by showing that certain websites are created by the same phishing kit. Such kits contain the content files needed to create the counterfeit website and often contain additional clues to the identity of the creators. A clustering algorithm is presented that uses collected phishing kits to establish clusters of related phishing websites. The ability to correlate websites provides law enforcement or other potential stakeholders with a means for prioritizing the allocation of limited investigative resources by identifying frequently repeating phishing offenders.

Keywords: Phishing, Clustering, Data Mining, Cybercrime Provenance, Phishing Kits

1. INTRODUCTION

Phishing is a form of cybercrime in which a criminal, generally labeled as a phisher, creates a fraudulent website or websites in order to lure victims into providing sensitive information such as usernames and passwords, social security numbers, and/or other information that can lead to identity theft, theft of online resources, or direct theft of assets. The collected information is often used to

withdraw money fraudulently from bank accounts (Li & Schmitz 2009) and may even be sold to other criminals through chat rooms (Jakobsson & Myers 2006). Phishers send spam[1] emails that mimic organizations by presenting recipients with a supposed account problem and a website address, known as a URL (Uniform Resource Locator), where the user can fix the alleged problem (Ludl et al. 2007). That URL leads to the phishing website where the victim is required to enter information to solve the made-up problem, thereby exposing his or her credentials to the cyber criminal. Typically, these counterfeit websites are hosted on web servers compromised through an exploit of software application vulnerabilities or by the use of a stolen userid and password that a webmaster would use to update the website via file transfer protocol (FTP) (Wardman et al. 2009).

Phishing attacks are gaining prevalence as more and more people use e-commerce and Internet banking websites, as discussed by Gartner research, documented evidence shows that more than five million U.S. citizens were phished from September 2007 to September 2008, representing a nearly a 40% increase from the previous year (Litan 2009).

The phished organizations generally take three courses of action in response. The organizations can simply ignore the phishing activity and reimburse financial losses suffered by their customers as a cost of doing business; they can optionally respond defensively by working to prevent users from visiting such malicious websites, or they can alternatively gather intelligence to help investigate, identify, and potentially prosecute the criminals behind the attacks. A review of the on-going practice across industry indicates that the defensive approach is the primary solution implemented by most organizations (Moore & Clayton 2007).

One defensive approach that organizations adopt, known as "takedown," is to identify phishing URLs and then contact the administrators of such websites in order to have the malicious content removed. This organizational response has proven to make a difference, yet this method does not thwart the phishers from future attacks because the corrective action is limited to the elimination of the website (Moore & Clayton 2007).

Blacklists are another defense mechanism for limiting the effectiveness of live phishing websites. A blacklist is a list of website addresses confirmed to be hosting malicious content, ideally through a reliable means that limits the number of websites placed on the blacklist improperly. Such a list can be used to prevent access to URLs in the potential victim's browser (Soldo et al. 2008). Blacklists are regularly improving with the increased reporting of live phishing websites to

[1] Spamming is the practice of sending unsolicited bulk email messages, often for marketing or criminal purposes. In the United States, the CANSPAM Act differentiates between legal and criminal bulk email on points such as whether the sender information is truthful, whether there is a means to "opt out" of future messages, and whether a true mailing address for the sender is provided.

anti-phishing vendor databases. Nevertheless, spam campaigns[2] for newly created phishing websites have been shown to last on average four - six hours; therefore, by the time it takes to blacklist and eventually disable a phishing website, the criminal has likely already moved on to spamming new URLs for the next phishing website (Sheng et al. 2009).

The criminal justice perspective of rational choice theory (Lanier & Henry 2004) indicates that in order to dissuade phishers from future attacks, law enforcement would need to demonstrate that phishing behavior has negative consequences that exceed its rewards. However, because of the complexity and breadth of knowledge required to handle phishing cases, corporate and law enforcement phishing investigators usually require numerous person-hours gathering evidence, analyzing data, and attempting to link smaller cases to a phisher. Law enforcement efforts can be supported through the efforts of organizations such as the Digital Phishnet (DPN), the Anti-Phishing Working Group (APWG), and the Internet Crime and Complaint Center (IC3); organizations that provide law enforcement with evidence assembled through contributions from benevolent private sector entities (Anti-Phishing Working Group 2010; Digital Phishnet 2010; Internet Crime and Complaint Center 2010).

While the collections of phishing websites gathered by blacklist maintainers (McAfee 2010; Netcraft 2010) and recipients of consumer complaints are important, additional analysis can provide two factors critical to pursuing criminal prosecution of the phisher. While blocking the URL may prevent further victimization, the URL does not identify the criminal. However, files on the phishing web server usually contain the email address of the criminal to whom the stolen information is provided. But, because phishing server content is often quickly replaced or deleted by webmasters to prevent further abuse, the file or files containing the criminal email address is often unavailable to investigators once they are able to access the system[3]. If other sites can be shown to be related to the now-terminated phishing website, the missing evidence of identity may be retrieved from a more recent website created by the same or similar phishing kit. This evidence may help investigators in the prosecution of the offender. Law enforcement could then serve legal process against the email account provider to obtain a login history, revealing the criminal's Internet Protocol (IP) address, which could be used to identify their geographic location and Internet Service Provider.

Furthermore, these URL collections do not provide any correlation between a given phishing website and financial losses caused by that site. While a bank may

[2] A spam or phishing campaign refers to the sending of a short, high volume distribution of email messages for a common intent. Such periods of intense transmission of email messages with common intent are denoted a campaign.

[3] An important goal is to train system administrators on how to preserve phishing and other hacking evidence.

realize it has lost a particular sum of money to phishing, it often cannot associate a given phishing website to a precise volume of financial loss because it does not know which website victimized their account holder. However, by reviewing the email records of the criminal, names and account numbers of victims could be positively linked to the phishing websites where the victimization occurred.

Given these issues and circumstances surrounding phishing, the goal of this research is to provide law enforcement and victim organizations with analyzed data that links evidence together in order to justify an investigation against a phishing campaign. A novel approach is developed here to cluster[4] phishing websites based on the MD5 hash[5] of the main web page and the associated content files used to create the website, such as graphics files, JavaScript files, and cascading style sheet files. In this research, phishing kits are collected, analyzed, and clustered against phishing websites stored in the University of Alabama at Birmingham's Phishing Data Mine (Wardman 2010). The UAB Phishing Data Mine includes phishing-related website content files, phishing website analysis algorithms, phishing kits, and over 75,000 phishing URLs that were either manually confirmed to be phishing sites through visual inspection or automatically confirmed by Deep MD5 Matching[6] (Wardman & Warner 2008) of the associated content files.

The remainder of the paper is structured as follows. Section 2 discusses other work related to the reduction of phishing. Section 3 introduces this research's methodology for collecting and correlating phishing evidence. Section 4 presents the results of the phishing kit analysis and clustering technique. Section 5 discusses the implications of the results for law enforcement investigations and phished institutions. Section 6 presents the conclusion of this work while future work related to this technique is presented in Section 7.

2. RELATED WORK

The following section discusses email filtering, user education, toolbars, and phishing activity aggregation and their use against phishing.

2.1 Email Filtering

One of the most widely used defensive mechanisms against phishing is email filters (Abu-Nimeh et al. 2007; Basnet et al. 2008; Chandrasekaran 2007; Fette et

[4] Clustering is a way of grouping phishing websites that have substantial similarity in one or more respects and are more like each other than other websites. Data mining algorithms are used to create clusters by evaluating similarities and differences; these are in turn denoted "clustering algorithms."

[5] MD5 hash – a value calculated by a standard one-way cryptographic algorithm. If two files have the same MD5 value they are mathematically provable to be identical to within a small probability (Valdes et al. 2003)

[6] Deep MD5 Matching refers to a previous algorithm which compares the MD5 values of many files from the same website with those of another website to determine if the two websites are similar.

al. 2006). The main goal of email filtering is to prevent a phishing email from reaching its intended recipient. Anti-phishing email filters use a variety of methods to recognize that an email is phishing-related, such as its frequency across a network or natural language cues within the email. For instance, Microsoft reports that it's SenderID, embedded in all of its email products and services, stops more than 25 million deceptive messages daily (Microsoft Safety 2010). Microsoft uses an email authentication technology protocol that "validates the origin of e-mail messages by verifying the IP address of the sender against the alleged owner of the sending domain" (Microsoft Safety 2010). Users of Mozilla's Thunderbird 3 open source email client can add this protection with the Sender Verification Extension (Tauberer 2008), and built-in to Thunderbird is the ability to warn users if they click on a link that appears to be directing them somewhere other than what is indicated in the email (Mozilla Messenging 2010).

Early text-based approaches that work against spam were not as effective against phishing as against spam in general (Saberi et al. 2007) since phishing emails are designed to mimic legitimate email and therefore usually include language and keywords similar to those in legitimate email messages. Researchers who are trying to improve anti-phishing email filters have more recently looked at other key features of an email message. Research in this area focuses on identification of a feature set for machine learning algorithms, usually classifier ensembles (Abu-Nimeh et al. 2007; Toolan & Carthy 2009; Yu et al. 2009). L'Huillier *et al.* proposed research into adversarial data mining, classifying by using a game mechanism theory between an adversary and an intelligent and adaptive classifier. (L'Huillier et al. 2009). Fette *et al.* determined whether a communication is deceptive about the sender's identity by analyzing features of the email such as the age of the domain names in the links or non-matching URLs (*i.e.*, differences between the web page link and the text describing the link) (Fette et al. 2007). The limitations of email filtering and browser-based anti-phishing tools have been documented by Zhang (2007).

More recently, social engineering approaches to phishing and attacks on statistical spam filters have thwarted email filters (Stamm et al. 2006; Wittel et al. 2004) . In what is known as spear-phishing, email messages have become highly targeted and contain the recipient's personal information harvested from web pages, such as greeting the recipient by name, or including their job title or company name in the body of the email.

2.2 User Education

As explained in research describing the content-based anti-phishing technique CANTINA (Zhang et al. 2007), even toolbars that employ heuristics and achieve more accurate results can fail in the end result if users do not understand "what the toolbar is trying to communicate". Therefore, researchers have developed training tools, including video games, browser-embedded context-aware warnings, and animated phishing tests designed to teach users how to recognize

phish. These techniques have long-term benefits when combined with other methods, such as the way the Netcraft toolbar incorporates educational content or the AntiPhishing Working Group's redirection page, which encourages webmasters whose sites are compromised to replace the phishing URL with a link to a phishing education page (APWG 2009). Although there are methods for restricting users from completing online forms (Ronda et al. 2008), it is important to avoid visiting phishing pages as there is an increased crossover between phishing and the distribution of malicious software or malware.

2.3 Browser-Based Filters

Phishing victimization can be prevented by detecting and blocking access to identified phishing websites from within a web browser. The Internet Explorer, Firefox, Google Chrome, and Safari browsers (Microsoft Safety 2010; Whittaker et al.) and many turn-key security products now offer anti-phishing capabilities (McAfee 2010; Netcraft Anti-Phishing Toolbar 2010; Symantec 2010). These operate with the use of blacklists, whitelists[7], and heuristics. However, many current blacklists fail to identify phishing URLs in the early hours of a phishing attack because they cannot be updated quickly enough (Ludl et al. 2007). This is important because the majority of phishing campaigns have been shown to last less than two hours (Sheng et al. 2009). Therefore, when a phishing website is positively identified and blacklisted within a browser, the campaign has usually already ended. Nonetheless, blacklists can have an effect that lasts longer than just the length of the campaign itself.

In addition to slow updating, another drawback of blacklists is specific to phishing. As many as 78% of phishing websites are hosted on hacked domains (Aaron & Rasmussen 2010), a statistic which indicates that legitimate websites may be left on blacklists long after the offensive content has been removed, potentially causing reputational harm to the legitimate website or organization. Blacklists are important in reducing the overall losses to phishing but are more effective when enhanced with other browser-based components such as heuristics (Sheng et al. 2009).

Whitelists are lists of websites that should never be blocked because they are determined to pertain to legitimate purposes and/or belong to legitimate organizations. These lists can be helpful when combined with other measures, especially in tightly monitored enterprise environments where it is possible to limit internet usage to a small set of websites relevant to business function. They are also employed at the user level, but this strategy often requires a training period that would frustrate most individuals as the browser asks, "Should this webpage be whitelisted?" for each unclassified webpage they visit (Cao et al. 2008).

[7] While blacklists are lists of known-to-be dangerous or criminal websites, whitelists are lists of known-to-be legitimate sites.

Heuristics can be used in combination with blacklists such as (Netcraft Anti-Phishing Toolbar) in browser-based anti-phishing, but automated techniques can lead to high false-positive rates compared to manually updated techniques as automated techniques can falsely identify legitimate websites as phish (Ronda et al. 2008). Techniques to reduce false positives include manual verification before blacklisting a website, whitelisting, and the use of other techniques such as Google's PageRank (Whittaker 2010; Zhang et al. 2007). PhishNet uses a method to enhance existing blacklists by discovering related malicious URLs using known phishing tactics such as changing up the top-level domain only (Prakash et al. 2010).

Google has outlined its blacklist updating process, which uses a scalable machine learning classifier trained on a large, noisy dataset (Whittaker et al. 2010). Features used by the Google blacklist include information extracted about the URL including whether it is on a whitelist, whether the URL uses an IP address as the domain name, and how many hostname components (words separated by a ".") there are before the domain name. The Google process also fetches the page content and extracts features about the host computer and the extent to which pages link to other domains. Combining these features with several others, Google's classifier "learns a robust model for" the millions of pages it crawls (Whittaker et al. 2010). This classifier eliminates false positives by evaluating their proprietary PageRank since, as the researchers conclude, there are no phishing pages with high PageRank.

2.4 Phishing Activity Aggregation

Most methods described above are defensive in nature, but some researchers have focused on the aggregation of information about phishing incidents (Basnet et al. 2008; Irani et al. 2008; Weaver & Collins 2007). Clustering algorithms based on the content of the email messages is found to be ineffective because of the short life of the features that can be extracted from the headers and the duplication that is found in the intended mimicry of the content (Basnet et al. 2008; Irani et al. 2008). Another form of phishing information aggregation is an algorithm that clusters reported phishing scams if they are hosted on the same IP address or network[8] in order to estimate the extent of phishing on those networks (Weaver & Collins 2007). This approach implies that the phishing websites hosted on the same networks are from the same phisher.

3. METHODOLOGY

The research methodology in this work is an information-gathering and analysis process that proactively provides intelligence to phishing investigators or other stakeholders about the phishers who are the most prolific during a certain interval

[8] A network is a collection of IP addresses managed under the same organization. When groups of IP addresses within a network have regularly hosted malicious content, other IP addresses on that network will be considered higher risk by IP reputation systems.

of time. Figure 1 illustrates the overall process used here. The first step in the process is to receive potential phishing URLs from various sources and incorporate them into the UAB Phishing Data Mine. A preprocessing step removes duplicate URLs and those that are already present in the UAB Phishing Data Mine. Duplicate URLs are those that contain the same domain name, path, and filename, ignoring varied parameter values that may follow the filename after a question mark or ampersand appears. The next step is to attempt to automatically confirm the URLs. If a URL is automatically confirmed as a phish through Deep MD5 Matching, then the URL is sent to the automated kit search tool; however, when the URL is not automatically confirmed, it is queued for manual confirmation.

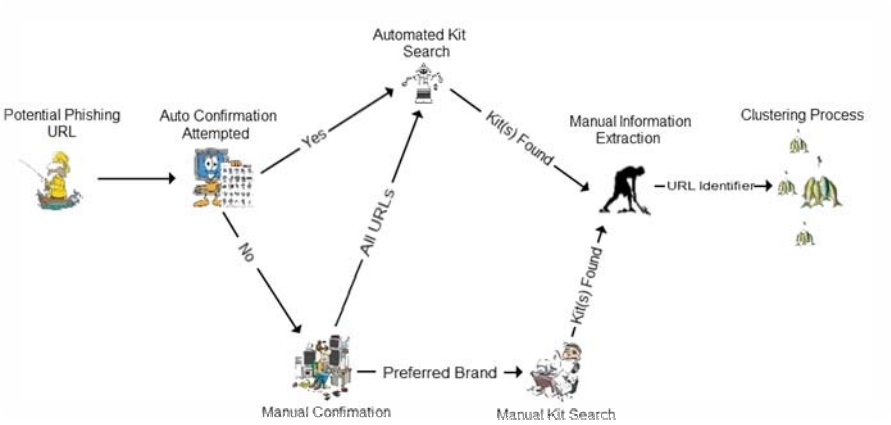

Figure 1. The overall framework for phishing URL confirmation and the phishing kit collection, extraction, and correlation process

When URLs are manually confirmed (*e.g.*, by trained personnel) there are two avenues for collecting phishing kits. For specific UAB-partnered target brands, the person who labeled the URL as a phish traverses the directory tree structure of the URL[9], searching for readable directories that may contain phishing kits. Secondly, all manually confirmed URLs are also sent to the automated kit search tool. Both kit searching methodologies currently require the subsequent manual extraction of phishing kit information. The unique identifier of the URLs where kits were found is finally sent to a clustering algorithm which groups closely-related phishing websites based on the website content files.

[9] "Traversing the directory tree" involves checking each directory included in the path portion of the URL to determine if it may reveal a list of files on the web server in that directory. A secure web server does not allow these file lists to be displayed, but most phishing sites are hosted on web servers with low security; so, open directories are often able to be located.

3.1 Kit Collection

The goal of the UAB phishing operations team is to confirm phishing URLs promptly either through automated techniques or manual inspection. Shorter latencies for detecting and confirming phishing URLs lead to a higher likelihood of researchers being able to collect relevant evidence against phishers, while websites remain up and running. This prompt confirmation is needed because, once notified, system administrators of hacked websites delete evidence that could identify the phisher. Examples of evidence that are often deleted include phishing kits, the phisher's email addresses, and other potential clues as to who created the phishing website.

Phishing websites typically collect data from the victim in an HTML form. Each form has an "action" that calls a program telling the website what to do with the stolen information. Usually the action calls a program on the web server that sends an email message to the criminal. These destination email addresses are called "drop addresses." Occasionally, the action statement will contain the drop address, but usually it is hidden in the program, which can be found in the phishing kit. Inside a phishing kit, the drop addresses may be found in cleartext or encoded with techniques such as ASCII-to-hex and/or Base64 encoding, making the drop address less obvious to a human investigator (Cova et al. 2008).

The UAB phishing operations team analyzed and documented 470 phishing kits between November 2008 and March 2010. The phishing kit retrieval process has evolved since the team began analyzing kits as some important lessons were acquired through the collection process. First, evidence needs to be saved, not just documented. Early in the process, the existence of a phishing kit was documented, but the kit itself was not preserved. Secondly, the URL that the kit was acquired from does not always correlate to the URL distributed through email, as many URLs contain a command that automatically redirects the victim to an additional website where the phishing content would likely be found. Because of such potential redirection, some of the kits discovered through a manual directory traversal are not located on the same server as the URL that was sent as the email link. An automated approach was developed in order to resolve these issues.

Algorithms and prototype software were devised to search for phishing kits in domains of phishing websites when phishing URLs are confirmed[10]. The tool produced from this effort searches for commonly used phishing filenames (*i.e.*, paypal.zip, eBay.zip, or chase.zip) by traversing the directory structure of the phishing URL. In this study, the tool was used to search for 130 common phishing kit names and to download the kit using GNU Wget 1.11.4 Red Hat modified. After download, the phishing kits are manually analyzed, and

[10] "Confirming a URL as a phish" means to verify that a website is a phishing website and not benign (at least with regard to phishing).

evidence, such as the email addresses and aliases of phishing kit creators and editors, is extracted and stored for use in future investigations.

3.2 Deep MD5 Matching

Comparing the MD5 hash of the web page advertised to the victim, hereafter referred to as the main index page, with those of confirmed phishing websites is a method for automatically verifying phishing websites (Provos 2009). When the MD5 values of the main index pages match, the potential phishing URL can be confirmed to be a phish and labeled with the brand that the website is imitating. This technique is often used by organizations and takedown companies to automatically confirm phish. However, this technique has significant limitations as discussed in (Wardman & Warner 2008). Simplistic obfuscation can defeat this technique by causing MD5s of the main index pages to be different while still using all the same files from a kit. Examples of obfuscation methodologies are the use of website scripts to include the recipients' email address passed from the URL or to include the current timestamp in the main index file each time the website is visited. Given the examples below, including this dynamic information causes the MD5 of the main webpage to vary even though the content is identical. In the below case, the recipients' email addresses are passed through the parameter 'login_email.'

*http://www.paypal.com.ufiyr4gscz.125tcb5cbquts9howt09.com/cgi-bin/webscr/?login-dispatch&login_email=**victim1@mailaddress.com**&ref=pp&login-processing=ok*

*http://www.paypal.com.e20jqm91gysjhz7yt.125ci3qk5uipl4wo3hr3.com/cgi-bin/webscr/?login-dispatch&login_email=**victim2@domain.com**&ref=pp&login-processing=ok*

*http://www.paypal.com.0o4589zq8stnemcjy.125kpszbkwapqkvzhkp3.com/cgi-bin/webscr/?login-dispatch&login_email=**victim3@mailserver.net**&ref=pp&login-processing=ok*

Deep MD5 Matching is a patent-pending technique for overcoming obfuscation by determining the similarity between two sets of associated files downloaded from potential and known phishing websites. Deep MD5 Matching uses Wget to download the content files associated with a potential phishing website. Content files are typically images, scripts, and style sheets such as gif, jpg, js, php, and css files. The set of content files from the potential phishing website is compared to sets of files of previously confirmed phishing websites using the value of their Kulczynski 2 coefficient (Kulczynski 1927). The Kulczynski 2 coefficient is expressed in Equation 1 where a is the number of matching file MD5s between the sets 1 and 2, b is the number of elements in set 1 that do not have MD5s matching a file in set 2, and c is the number of elements in set 2 that do not have MD5s matching a file in set 1.

$$Kulczynski\ 2 = \frac{1}{2}\left[\frac{a}{a+b} + \frac{a}{a+c}\right] \qquad \textit{Eq. 1}$$

The value provided be evaluating Equation 1 measures the similarity between two file sets by taking the average of the proportion of matching files in the two file sets. The Kulczynski 2 similarity coefficient was selected because the percentage of matching files in one file set should have equal weight to the percentage of matching files in the other file set, so as not to discriminate against the file set of either URL. Other matching criteria are also possible.

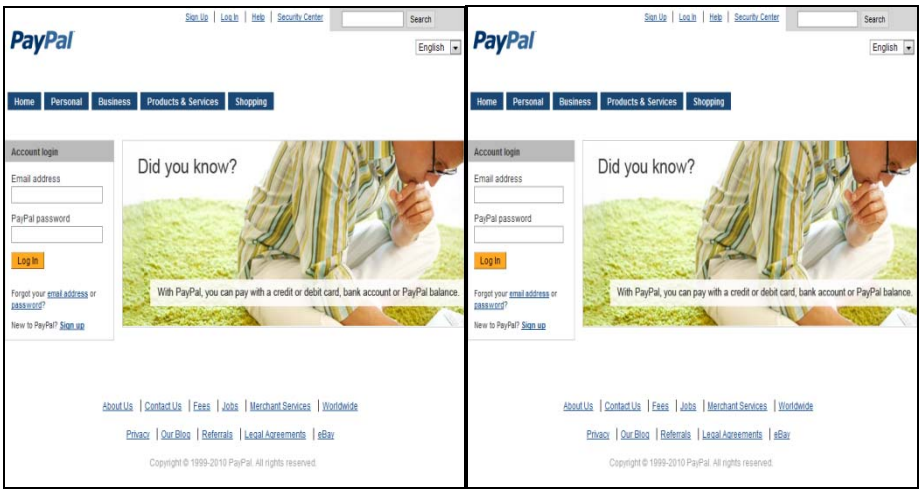

Figure 2. index.php (left), index.html (right)

Figure 2 helps to illustrate how Deep MD5 Matching operates. The two nearly identical PayPal phishing websites depicted in Figure 2 were hosted on two different domains. The web page depicted on the left is a confirmed phishing website, while the web page on the right represents a potential phishing website. The main index page on the left is named index.php and has six associated content files that comprise the web page's appearance and functionality. The main index page on the right is named index.html and also has six content files. Both websites contain the same total number of files and have identical, associated content files as verified through their MD5 values. The only difference between the two file sets are the MD5s of the main index pages. Equation 2 illustrates how the Kulczynski 2 coefficient is applied to the two PayPal websites.

$$Kulczynski\ 2 = \frac{1}{2}\left[\frac{6}{6+1} + \frac{6}{6+1}\right] = 0.857 \qquad \textit{Eq. 2}$$

The result of Equation 2 produces a similarity score greater than our threshold (which is chosen at .85 currently); therefore, Deep MD5 Matching identifies and brands the potential phishing URL as a PayPal phishing website.

3.3 Clustering

This study employs manual and automatic kit collection and Deep MD5 Matching for gathering and correlating evidence for law enforcement. The collection process consists of identification, download, and analysis of the phishing kits, while the correlation process uses an agglomerative[11] clustering algorithm based on Deep MD5 Matching. In order to improve computational speed, the clustering algorithm is performed in four phases as depicted in Figure 3.

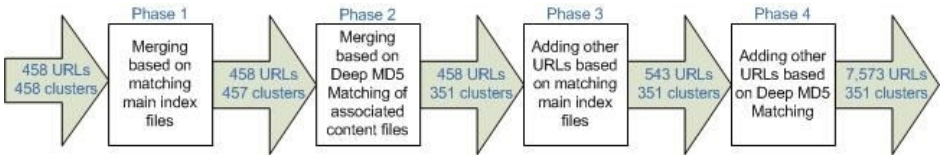

Figure 3. The four phases and results of the clustering process.

The Deep MD5 clustering algorithm proceeds as follows. Phases 1 and 2 initially merge clusters only within the set of URLs of collected kits. Each URL that results in the discovery and acquisition of a phishing kit represents an initial cluster of size one. As clusters merge and items are subsequently added to clusters, there remains only one representative URL for each cluster. All future items compared to the cluster only use the representative URL to indicate similarity. In Phase 1, clusters merge if the MD5 of the main index pages are equal. Phase 2 merges clusters whose representative URLs have content files with a Kulczynski 2 similarity coefficient greater than or equal to 0.85. The average number of files per phishing website that contained at least two files was computed over a three month data set of collected phishing websites. The resulting average number of files retrieved per website was 8.73. If the main index page of a website under consideration matches exactly, it would be clustered under Phase 1 or 3. Therefore, Deep MD5 matching (Phases 2 and 4) assumes at least one file has a non-matching MD5. The calculation of 7.73 files of 8.73 would give a similarity of 0.89. (While the threshold 0.85 was chosen for this experiment future work will test the false positives and false negatives generated by other threshold values, with particular attention to exploring lower threshold values to ensure reduction in false negatives, but with the concomitant risk of added false positives.)

[11] Agglomerative clustering is an algorithm that places individual elements into their own clusters and merges these clusters based on particular conditions such as similarity coefficients or distance metrics (Han & Kambler 2001).

Phases 3 and 4 enhance the clusters created in phases 1 and 2 by measuring the similarity between those clusters and all other URLs in the UAB Phishing Data Mine. Phase 3 compares the MD5 of the main index page for each representative URL against the MD5s of the main index page all unclustered URLs. If the MD5s match, the URL is added to the cluster. In Phase 4 those URLs that are not yet clustered are considered in the same manner as Phase 2, and joined to a cluster if the threshold is exceeded.

4. EXPERIMENTAL RESULTS

This section describes the results of the manual and automatic phishing kit collection and phishing website clustering algorithm.

4.1 Phishing Kits

In this study a total of 460 phishing kits were collected through a combination of both the manual and automated tree traversal methodologies discussed in Section 3.1. The manual approach collected 323 phishing kits between November 2008 and March 2010 that were associated with phishing URLs in the UAB Phishing Data Mine. The automated technique used in this study had a duration of two weeks in September 2009 during which 137 valid phishing kits were retrieved.

The manual analysis of the 137 phishing kits retrieved automatically discovered 181 unique email addresses and 81 unique aliases of kit creators or the criminals who customized a particular kit in order to include their own email address. The manual processing of the kits includes following the action parameter from the main phishing page to the filenames in the kit. A kit will normally contain at least one drop email address, usually in cleartext[12], but the kit often contains other information that may help to identify its author or distributor, such as an alias, a comment, or other artifact. As mentioned above, kit authors sometimes create a covert channel to receive stolen information by encoding their drop email addresses in their kit so that unaware phishers will do the work of creating the websites while the kit author still receives the victim information via the secret drop addresses embedded in the kit.

4.2 Clustering of Phishing Websites

A clustering algorithm was created to identify phishing websites that may prove to be of interest for further investigation and link them to the aliases and email addresses found in the above methodology. The combination of the manual and automated phishing kit collection methodologies gathered 460 phishing kits. These 460 phishing kits yielded 458 unique URLs as two of the URLs had a kit retrieved both manually and automatically. The clustering algorithm consists of four phases as illustrated in Figure 3.

[12] Text with no obfuscation or encryption applied to it.

The first two phases of the presented clustering algorithm merge clusters consisting of 458 phishing websites. The latter two phases add additional phishing sites to these clusters by comparing them to other phishing sites found in the UAB Phishing Data Mine that did not have phishing kits associated. The results of performing Phase 1, which merges the initial clusters by main index file MD5 matching, only merged two clusters. Therefore, 457 clusters were the input to Phase 2, which merges clusters based on Deep MD5 Matching. After Phase 2, 106 clusters were merged, leaving 351 clusters. The largest cluster after Phase 1 consisted of two phishing sites, while the largest cluster after Phase 2 contained 24 phishing sites.

Phases 3 and 4 perform a similar function as Phases 1 and 2, respectively, except Phases 3 and 4 increase the size of existing clusters instead of merging clusters. Using the existing 351 clusters, Phase 3 added 85 phishing sites from the UAB Phishing Data Mine to the clusters, while Phase 4 added 7,030 phishing sites. The largest cluster in Phase 3 contained 67 phishing sites, and the largest cluster in Phase 4 contained 865.

The final results after all four phases of clustering left 351 clusters containing a total of 7,573 phishing sites. During the manual labeling method practiced by the UAB phishing operations team, websites that are not displaying phishing content at the time of manual review are marked as "unknown" or "not a phish." However, the automated process downloads website content when the URL is first reported, and through the matching of content files, the clustering algorithm was able to establish that 1,467 websites that had been labeled as being either an unknown or as not a phish could now be identified as phish. Twenty-four phished institutions were represented by these clusters. Approximately 18% of the sites in the clusters had files that were exactly the same, as measured by a similarity coefficient of 1.0. Therefore, sites were merged or added to clusters 82% of the time because of Deep MD5 clustering where the similarity measure is greater than or equal to 0.85. This means that if the clustering algorithm were limited to matching only the index pages or exact matches of all content files, then 6,245 phishing sites would not have been included in clusters and would therefore be considered singleton instances of a phishing attack.

Approximately 34% of the clusters contained at least one other phishing site in the cluster, while the other 64% were singletons. There were 57 clusters that contained ten or more phishing sites, and the 24 largest clusters each contained more than 100 phishing websites. Table 1 shows the cluster sizes for the top ten clusters in this study.

Rank	Cluster Size	Brands
1	**865**	**Bank A**
2	**591**	**Bank B**
3	**549**	**Bank A**
4	**445**	**Bank A**
5	**433**	**Bank A**
6	**357**	**Bank C**
7	**332**	**Bank D**
8	**228**	**Bank D**
9	**203**	**Bank C**
10	**181**	**Bank A**

Table 1. The top 10 cluster sizes with associated brands

The results contained 190 clusters where the seed website contained only one file, the main index page. Of these 190 clusters, only 25 had more than one website within the cluster, demonstrating that dynamic content often bypasses main index matching. The largest of these one file clusters consisted of 135 websites and is the 19th largest cluster. Finally, phishing kits whose MD5 values matched had URLs found in the same clusters, an indication that the hypothesis, that websites created by the same phishing kit will cluster, holds.

5. DISCUSSION

This section discusses the results of the Deep MD5 clustering algorithm. First, the results of the clustering phases are described. Next, a representative cluster is described in detail. Lastly, the relevance of the technique to law enforcement and phished organizations is discussed.

5.1 Clustering Phase Analysis

The results of the clustering phases demonstrated the ability for Deep MD5 clustering to enhance the grouping of similar websites when compared to main index page matching. In both the merging and adding phases described in Section 4.2, Deep MD5 clustering was able to merge and add clusters at a much greater rate than compared to main index clustering. Although this experiment started with only 458 phishing websites, the final results established relationships among 7,573 websites. Furthermore, the algorithm showed the ability to confirm phishing websites that were previously labeled as unknown or not a phish by manual review.

Phase 1 – Main index clustering	**2 seconds**
Phase 2 – Deep MD5 clustering	**3 minutes 11 seconds**
Phase 3 – Main index clustering	**11 minutes 12 seconds**
Phase 4 – Deep MD5 clustering	**8 hours 23 minutes 1 second**

Table 2. Run times of clustering phases

As expected, main index clustering performed poorly because of the dynamic content in the main index page. Table 2 demonstrates that main index matching has a considerably faster run time than Deep MD5 clustering. On larger data sets, the difference between the two clustering algorithms' run times will have a greater impact on total run time. This suggests that main index clustering should be used to help reduce the number of websites to be clustered by other more time-intensive techniques.

5.2 Cluster Analysis

Although clusters can be viewed as collections of phishing URLs and their associated content files, each cluster has its own distinguishing characteristics such as the composition of the set of files and variations found in the distinct files used to create the clusters. Different versions of phishing kits contain a number of similar files, but, over time, creators modify the kit design, dispersing them through a variety of distribution avenues, such as websites where they can be downloaded for "free". Even though kits by the same creator typically have many similar files, the number of files in the kit will vary slightly, and only a handful of files will be distinct.

In order to gain a deeper understanding of the composition of a cluster, the third largest cluster containing 549 members was evaluated. This cluster was chosen for further description since it contained the largest collection of phishing kits found through clustered URLs. In this particular cluster, there were 38 URLs that had an associated phishing kit downloaded from them. This cluster has URLs with file counts ranging from 26 files to 46 files. For 94% of the URLs, the files numbered in the range of 30 – 35. There is apparently a strong relationship between the number of files downloaded from the URL and the number of files found in the associated kit. Each of the URLs in this cluster has at least an 85% similarity to the seed URL. When slight changes in file counts are found in closely-related kits, they are considered by the authors' approach to be related to

"versioning" of the kit. For example, there might be a single new graphic added or a new set of questions requiring an additional JavaScript file, but generally the preponderance of the kit remains constant.

Table 3 shows the number of downloaded files with the number of kit files across the 38 URLs where kits were obtained. In most cases, there are 26 more files extracted from the kit than are downloaded from the URL. The significant file difference is because of the limitations of the fetching technique used to download the URL files. The phishing site is designed to present the victim with a series of forms to be completed. Wget only obtains the files associated with the main index page as downloading the additional web pages would require user input. This does not impact the validity of the findings as trends are still evident in the results.

Number of Kits	Number of Files from URL	Number of Files from the Kit
3	**30**	**56**
1	**30**	**62**
1	**32**	**60**
5	**33**	**58**
23	**33**	**59**
2	**33**	**60**
1	**33**	**61**
1	**34**	**60**
1	**34**	**61**

Table 3. Illustrates the similarities and mutations of phishing kits

Some phishing kits produce websites that present the victim with multiple pages for user-provided information, but the pages after the first are not processed unless realistic answers are provided at each step. All of the content files not downloaded via Wget are part of these subsequent user-input pages. Table 4 contains a comparison of the types of files downloaded from the URLs of the analyzed cluster to the files extracted from the associated phishing kits. These particular file lists are associated with the URLs that have 33 associated website files and 59 extracted kit files.

Analysis of the mean number of kit files yielded additional drop email obfuscation methods. For example, one of the kits that contained 57 files was determined to have a fake image file that contained a hidden email address. Additionally, kits that were missing the file named check_fields.js instead used a hexadecimal-to-ASCII obfuscation in the main index page.

Files downloaded from URL	Files extracted from kit
1 – PHP file	**1 – PHP file**
8 – Cascading Style Sheets	**4 – HTML files**
14 – GIF images	**8 – Cascading Style Sheets**
10 – JavaScript files	**24 – GIF images**
	21 – JavaScript files
	1 – ASPX

Table 4. A comparison of files between the URL file set and the kit file set.

Analyzing the files that have remained unchanged revealed that one of the two email addresses *s33th3rs@yahoo.co.uk* and *seether@safe-mail.net* was hidden with Base64 encoding in JavaScript files. This relation was found in 36 out of the 38 kits, providing confidence that the other 511 URLs in the cluster have a 95% likelihood of being associated with the same drop email addresses.

5.3 Law Enforcement Investigations

This research indicates the effect that clustering of phishing website content files have on identifying malicious actors. The website clusters help to present law enforcement and financial institutions as to which cybercriminals most likely warrant investigation based on the volume of websites these criminals are distributing and maintaining. Because there are many competing priorities for investigative attention, the application of clustering techniques to phishing data will help to ensure that habitual offenders are investigated with a higher priority than first-time offenders who are likely to have claimed fewer victims at this point in time. Reports generated using the techniques above are currently being shared with law enforcement. Investigators within financial institutions are also using these reports to assist in their internal investigations and have expressed that this information greatly enhances their investigative capacity.

6. CONCLUSIONS

Phishing is an important cybercrime that needs to be hindered, and phishing criminals need to be identified as a prerequisite to pursuing them legally. In response, phishing web site clustering is a new, effective methodology for identifying significant high-volume offenders in the area of phishing, as compared to others, with direct impact on the ability to prioritize investigations. In particular, by clustering phishing URLs, evidence can be provided to law enforcement that distinguishes clusters of criminal activity indicating a potential high value target for investigation. The largest URL clusters can be further investigated to identify associated kits that reveal the drop email addresses of the most significant suspects. These known phishing email addresses and aliases, identified in the kits and linked to the clusters, can serve as the starting points of a criminal investigation. Law enforcement can then use the process of subpoena

and search warrant to identify the IP address of the alleged criminal who checks that email and the identities of the victims who can now be tied to this particular criminal's activities. The URL clusters generated by this research provide two missing factors for law enforcement – clues to the identity of the phisher and a means to measure the phishing damages caused by this particular criminal.

7. FUTURE WORK

In the next phase of this research, the authors plan to work with a law enforcement partner agency to focus law enforcement attention on identified "top kits" and to measure the likelihood of successful identification, arrest, and prosecution compared to traditional methods. To further automate this process, software will be developed to extract pertinent information such as email addresses and aliases from the phishing kits.

A further analysis of the clustering algorithm developed here will be applied to over 75,000 phishing URLs in the UAB Phishing Data Mine and used to identify and document the key trends that have evolved over time. Modifications to the clustering algorithm will allow for generation of clusters given a large random data set, rather than beginning with URLs corresponding to phishing kits.

Adjustments to the thresholds used in this work and measurements of false positives and false negatives associated with varying this parameter will be studied, as will multiple approaches to match files and sets of files beyond use of MD5 hashes. Additionally, filters will be applied to ensure that many common small files do not cause false positives, and sampling methods will be explored to reduce the total volume of files that must be downloaded from certain sites with huge numbers of files and aggregate file size.

8. ACKNOWLEDGMENTS

This research was made possible by the support of the UAB Department of Computer & Information Sciences, the UAB Department of Justice Sciences, and funding received from the Bureau of Justice Assistance (#2008-DD-BX-0407) and the Office of Community Oriented Policing Services (#2006-CKWX-0582). Points of view and opinions expressed are those of the authors and are not necessarily those of the United States Department of Justice, the Bureau of Justice Assistance, or the Office of Community Oriented Policing Services. The authors also express appreciation to the staff of the UAB Phishing Operations Team in the UAB Computer Forensics Research Laboratory.

REFERENCES

Aaron, G. and Rasmussen, R. (2010). 'Global Phishing Survey 2H/2009'. Counter eCrime Operations Summit IV. May 11-13, 2010. Säo Paulo, Brazil.

Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S. (2007). 'A Comparison of Machine Learning Techniques for Phishing Detection'. APWG eCrime Researchers Summit, October 4-5, 2007. Pittsburgh, PA.

Anti-Phishing Working Group (2009). 'APWG/CMU CUPS Phishing Education Landing Page Project: Optimizing Counter-eCrime Consumer Education Through Just-in-Time Delivery of Computer Safety Instruction', APWG Public Education Initiative, Lexington, MA.

Anti-Phishing Working Group (2010), 'APWG', http://www.antiphishing.org/, July 17, 2010.

Basnet, R., Mukkamala, S., and Sung, A. (2008), "Detection of Phishing Attacks: A Machine Learning Approach," Studies in Fuzziness and Soft Computing, 226: 373-383.

Cao, Y., Han, W., and Le, Y. (2008). 'Anti-phishing Based on Automated Individual White-list'. ACM Workshop on Digital Identity Management. October 31, 2008. Alexandria, VA.

Chandrasekaran M., Narayanan, K., and Upadhyaya, S. (2006). 'Phishing E-mail Detection Based on Structural Properties'. New York State Cybersecurity Conference Symposium on Information Assurance: Intrusion Detection and Prevention. July 14-15, 2006. Albany, NY.

Cova, M., Kruegel, C., Vigna, G. (2008). **'**There is No Free Phish: An Analysis of "Free" and Live Phishing Kits'. USENIX Workshop on Offensive Technologies. July 28, 2008. San Jose, CA.

Digital PhishNet (2010), 'Digital PhishNet', https://www.digitalphishnet.org/, July 17, 2010.

Fette, I., Sadeh, N., and Tomasic, A. (2007). 'Learning to detect phishing emails'. International Conference on World Wide Web. May 8-12, 2007. Banff, Alberta, Canada.

Han, J., and Kamber, M. (2001), Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA.

Internet Crime Complaint Center (2010), 'Internet Crime Complaint Center', http://www.ic3.gov/, July 17, 2010.

Irani, D., Webb, S. Giffin, J., Pu, C. (2008) 'Evolutionary Study of Phishing'. eCrime Researchers Summit. October 14-16, 2008. Atlanta, GA.

Jakobsson, M. and Myers, S., Eds. (2006), Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft, Wiley-Interscience, Hoboken, NJ.

Kulczynski, S. (1927), "Die Pflanzenassoziationen der Pienienen," Intern. Acad. Pol. Sci. Lett. Cl. Sci. Math. Nat, Ser. B, 1927(2): 180.

Lanier, M.M. and Henry, S. (2004), Essential Criminology, Westview Press, Boulder, CO.

L'Huillier, G., Weber, R., and Figueroa, N. (2009). 'Online Phishing

Classification Using Adversarial Data mining and Signaling Games'. ACM SIGKDD Workshop on Cybersecurity and Intelligence Informatics. June 28, 2009. Paris, France.

Li, S. and Schmitz, R. (2009). 'A Novel Anti-Phishing Framework Based on Honeypots'. APWG eCrime Researchers Summit. October 20-21, 2009. Tacoma, WA.

Litan, A. (2009), 'The War on Phishing Is Far From Over'. Gartner, Inc., Research ID Number G00166605, April 2, 2009.

Ludl, C., McAllister, S., Kirda, E., and Kruegel, C. (2007). 'On the Effectiveness of Techniques to Detect Phishing Sites'. International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. July 12-13, 2007. Lucerne, Switzerland.

McAfee (2010), 'McAfee SiteAdvisor Software', http://www.siteadvisor.com/, July 17, 2010.

Microsoft Safety (2010), 'Anti-phishing Technologies', http://www.microsoft.com/mscorp/safety/technologies/antiphishing/, April 8, 2010.

Microsoft Safety (2010), 'Sender ID',

http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx, April 8, 2010.

Moore, T. and Clayton, R. (2007). 'Examining the Impact of Website Take-down on Phishing'. APWG eCrimes Researchers Summit. October 4-5, 2007. Pittsburgh, PA.

Mozilla Messenging (2010), 'Thunderbird 3 Features', http://www.mozillamessaging.com/en-US/thunderbird/features/, April 8, 2010.

Netcraft (2010), 'Anti-Phishing Toolbar', http://toolbar.netcraft.com/, June 30, 2010.

Prakash, P., Kumar, M., Kompella, R.R., and Gupta, M. (2010). 'PhishNet: Predictive Blacklisting to Detect Phishing Attacks'. IEEE Conference on Computer Communications. March 15-19, 2010. San Diego, CA.

Provos, N. (2009). 'Google: Malware Sites on the Upswing'. Eweek, August 27, 2009.
http://securitywatch.eweek.com/online_malware/google_malware_sites_on_the_upswing.html

Ronda, T., Saroiu, S., and Wolman, A. (2008). 'Itrustpage: a User-assisted Anti-phishing Tool'. ACM Sigops/Eurosys European Conference on Computer Systems. April 1-4, 2008. Glasgow, Scotland.

Saberi, A., Vahidi, M., and Bidgoli, B.M. (2007). 'Learn to Detect Phishing Scams Using Learning and Ensemble Methods'. Web Intelligence and Intelligent Agent Technology Workshops, IEEE/WIC/ACM International Conferences. November 2-5, 2007. Silicon Valley, CA.

Sheng, S., Wardman, B., Warner, G., Cranor, L.F., Hong, J., and Zhang, C. (2009). 'An Empirical Analysis of Phishing Blacklists'. CEAS 2009 – Sixth Conference on Email and Anti-Spam. July 16-17, 2009. Mountain View, CA.

Soldo, F., El Defrawy, K., Markopoulou, A., Krishnamurthy, B., van der Merwe, J. (2008). 'Filtering Sources of Unwanted Traffic'. Information Theory and Applications Workshop. Jan 27, 2008 – Feb 1, 2008. San Diego, CA.

Stamm, S., Ramzan, Z., and Jakobsson, M. (2006). 'Drive-by Pharming'. Information and Communication Security 2007. December 12-15, 2010. Zhengzhou, China.

Symantec (2010), 'Firewall – Anti Virus – Phishing Protection | Norton 360', http://www.symantec.com/norton/360, July 17, 2010.

Tauberer, J. (2008), 'Add-ons for Thunderbird: Sender Verification Antiphishing Extension 0.9.0.2.', https://addons.mozilla.org/en-US/thunderbird/addon/345, April 8, 2010.

Toolan, F. and Carthy, J. (2009). 'Phishing Detection Using Classifier Ensembles'. APWG eCrime Researchers Summit. October 20-21, 2009. Tacoma, WA.

Valdes, A., Almgren, M., Cheung, S., Deswarte, Y., Dutertre, B., Levy, J., Saïdi, H., Stavridou, V., and Uribe, T. (2003). "An Architecture for an Adaptive Intrusion-Tolerant Server," Security Protocols: Lecture Notes in Computer Science, 2845: 569-574.

Wardman, B. (2010). 'UAB Phishing Data Mine'. University of Alabama at Birmingham Computer and Information Sciences Department Technical Report Number : UABCIS-TR-2010-111710-1. November 17, 2010. http://www.cis.uab.edu/forensics/TechReports/PhishingDataMine.pdf.

Wardman, B., Shukla, G., and Warner, G. (2009). 'Identifying Vulnerable Websites by Analysis of Common Strings in Phishing URLs'. APWG eCrime Researchers Summit. October 20-21, 2009. Tacoma, WA.

Wardman, B. and Warner, G. (2008). 'Automating Phishing Website Identification Through Deep MD5 Matching'. eCrime Researchers Summit. October 15-16, 2008. Atlanta, GA.

Weaver, R. and Collins, M. (2007). 'Fishing for Phishes: Applying Capture-Recapture Methods to Estimate Phishing Populations'. APWG eCrime Researchers Summit. October 4-5, 2007. Pittsburgh, PA.

Whittaker, C. Ryner, B., and Nazif, M. (2010). 'Large-Scale Automatic Classification of Phishing Pages'. Network and Distributed Systems Security Symposium. February 28-March 3, 2010. San Diego, CA.

Wittel, G., and Wu, S. (2004), 'On Attacking Statistical Spam Filters', CEAS 2004 – First Conference on Email and Anti-Spam. July, 2004. Mountain View, CA.

Yu, W.D., Nargundkar, S., and Tiruthani, N. (2009). 'PhishCatch - A Phishing Detection Tool'. IEEE International Computer Software and Applications Conference. July 20-24, 2009. Seattle, WA.

Zhang, Y., Egelman, S., Cranor, L. and Hong, J. (2006), 'Phinding Phish: Evaluating Anti-Phishing Tools', CyLab Technical Report, CMU-CyLab-06-01, November 13, 2006.

Zhang, Y., Hong, J.I., and Cranor, L.F. (2007). 'Cantina: A Content-based Approach to Detecting Phishing Web Sites'. International Conference on World Wide Web. May 8-12, 2007. Banff, Alberta, Canada.