Mining web to detect phishing URLs

Conference Paper · December 2012

DOI: 10.1109/ICMLA.2012.104

CITATIONS
15

READS
485

2 authors:

Ram Basnet
Colorado Mesa University
29 PUBLICATIONS **354** CITATIONS

SEE PROFILE

Andrew H. Sung
University of Southern Mississippi
191 PUBLICATIONS **4,565** CITATIONS

SEE PROFILE

Mining Web to Detect Phishing URLs

Ram B. Basnet
Sage Technology Partners, Inc.
Albuquerque, NM, USA
rbasnet@sagetpi.com

Andrew H. Sung
Computer Science and Engineering, New Mexico Tech
ICASA
Socorro, NM, USA
sung@cs.nmt.edu

***Abstract*— Proliferation of phishing attacks in recent years has presented an important cybersecurity research area. Over the years, there has been an increase in the technology, diversity, and sophistication of these attacks in response to increased user awareness and countermeasures. In this paper, we propose a novel scheme to automatically detect phishing URLs by mining and extracting Meta data on URLs from various Web services. Applying the proposed approach on real-world data sets, it is demonstrated that Logistic Regression classifier can achieve an overall accuracy of 97.2-99.8%, false positive rate of 0.1-1% and false negative rate of 0.7-6.5% in detecting phishing and non-phishing URLs.**

Keywords- web mining, phishing detection, phishing URL, anti-phishing, machine learning

I. INTRODUCTION

A Phishing URL is a URL created with malicious intent such as to perform phishing attack, to download malware to the unsuspecting visitors' computer (drive-by downloads [3]), or search engine result manipulation [19], etc. In a typical phishing attack, scammers or "phishers" induce unsuspecting Internet users to click on a link – normally obfuscated – to their phishing websites to trick into revealing their private information, e.g., username, password, bank account, credit card number, etc. Blacklisting is the most common technique used by all major web browsers – Internet Explorer, Firefox, Chrome, Opera, etc. When a user tries to load a URL that is in the browser's blacklist, she is warned about the potential danger of visiting the webpage. Though blacklisting can be very effective in blocking the previously known phishing URL, it can miss the brand new (zero-day) phishing web pages [17].

In reaction to increasing response from service providers and law enforcement, criminals are using increasing technical sophistication to establish more survivable infrastructures that support phishing activities. The key building blocks for these infrastructures are the botnets that are used to send phishing emails and host phishing sites [1]. Also, a recent report by the Anti-phishing Working Group (APWG) indicated more sophisticated schemes seem to have been used in phishing attacks that also exploited an increased number of brands [2].

In this paper, we propose a set of heuristics that can be used in near real-time to evaluate the legitimacy of a URL. Unlike existing works in this area, the proposed heuristics are rooted in the evaluation of Meta data on URLs commonly available from search engines and other popular Web services. By extracting information on a URL from various Web services and using them as features for Logistic Regression classifier, we empirically demonstrate that proposed system is highly effective in detecting phishing URLs with respect to real-world data sets of more than 16,000 phishing and 31,000 non-phishing URLs. Moreover, because of the focus on the URL itself, we believe that the approach can be applied anywhere a URL can be embedded, such as in email, webpages, chat sessions, etc.

The rest of the paper is organized as following. Section II describes in details our approach – how we mine Web to gather Meta data on URLs and what classification model, evaluation criteria, and data sets we use for our experiments – to classify phishing and non-phishing URLs. Section III provides results of our experiments. In section IV, we show the tuning of false positive and false negative rates. In section V, we discuss some limitation of our approach and how adversaries can attack the system if deployed in real-world application. Section VI reviews some related works and section VII provides concluding remarks and some future directions.

II. OUR METHOD

A. Web Mining Based Heuristics

We use search engines to gather information about things that we would like to know more about on a daily basis. Similarly, we employ the top 3 most popular search engines – Google, Yahoo! and Bing – to gather Meta data on URLs. Besides, we also use historical reports and statistics and blacklists published by trustworthy sources to determine whether the URL is phishing or not.

Search engine has been used in recent papers to detect phishing webpages. Google search engine has been used by Garera et al. [12], Whittaker et al. [10], and Zhang et al. [9]. Whittaker et al. use PageRank from Google proprietary infrastructure. Garera et al. use Google's proprietary technologies such as PageRank, page index, and page quality scores. These are pre-computed during Google's crawl phase and are stored in a table, which they call *Crawl Database*. On the other hand, our search engine based feature gathering technique uses either publicly available APIs or mimics users using search engines to gather information on a URL.

Zhang et al. select the top 5 words with highest TF-IDF (a common technique in information retrieval) value to generate lexical signature of a page. They feed each lexical signature to Google search engine and check if the domain name of the current web page matches the domain name of the top 30 results. If yes, they consider it to be a legitimate website. Though the purpose is similar, we utilize search engines in different ways. Instead of using the query terms to search for the relevant hyper links, we directly search for a hyper link using URL and domain. In other words, we check if a URL and domain exist in the search engines' index by parsing the top 30 search results. If none of the returned link matches the search query we flag the URL as potentially phishing.

If both the URL and the domain do not exist in search engines index, it is a high indication that the domain is a newly created one and thus more likely to be phishing. Hence, we believe that these features also compliment the 'age of domain' feature based on WHOIS used by most of the related works.

Our heuristic is based on the assumptions that the top 3 search engines index the vast majority of legitimate websites, and that legitimate sites usually live longer and hence the search engine crawlers will index them sooner or later. On the other hand, the average time a phishing site stays online is 4.5 days or even less [9]. Moreover, there won't be that many links pointing to the scam site. Because of the low life span and lack of links pointing to the phishing site, we assume that search engines crawlers may not get to the site before they are taken down. We also employ 3 major search engines with the strong reason that at least one of the search engines must have had indexed legitimate website if not all. Moreover, we believe that search engines also try to filter out known malicious web pages from the search results using their proprietary technologies; thus, this heuristic effectively exploits their Web crawling and filtering techniques.

We next describe our reputation-based heuristics gathered from various Web services.

PhishTank.com [11] produces various top 10 statistical reports on phishing websites every month. We downloaded 3 types of statistics: Top 10 Domains, Top 10 IPs, and Top 10 Popular Targets from the first batch of statistics published in October 2006 to October 2010. The idea behind this is to make use of the historical data on top IPs and domains that host phishing websites. If a URL has many other phishing related heuristics and also its host belongs to top IP and/or domain that has historic reputation of hosting most phishing webpages, then we can increase our confidence level to classify the URL at hand as a phishing.

StopBadware.org [4] works with its network of partner organizations such as Google, Sunbelt Software, etc. and individuals to fight back against viruses, spyware, and other badware. It produces top 50 IP address report from number of reported URLs. We check if the IP address of a URL belongs to this top 50 report and flag it as potentially phishing if it does.

We use hpHosts to check if the domain of a URL exists in its database. hpHosts is a community managed and maintained hosts file that allows an additional layer of protection against access to ad, tracking and malicious websites [6].

It is also worth pointing out that, relatively a smaller number (~300 in average) of popular known brands are targeted by phishers. The number of phished brands reached a high of 298 in March of 2010 while the number of brand-domain pairs detected at end of 1st Quarter 2010 was 10,752 [2].

Blacklists are employed by most major modern browsers to keep Internet users from malicious websites. However, centralized blacklist based protection alone is not adequate enough to protect end users from new and emerging zero-day scams that appear in thousands and quickly disappear every day. Nevertheless, we use Google Safe Browsing API [5] to check URLs against Google's constantly updated blacklists of suspected phishing and malware pages and use 3 binary features for membership in blacklists provided by Safe Browsing API. Essentially, these blacklists are also used by Google Chrome and Mozilla Firefox to warn users of potentially malicious websites.

B. Classification Model

Using the heuristics described in section A, we encode each individual URL into a feature vector with 14 dimensions. All of the features are binary indicating whether the corresponding heuristic is present or absent in a URL. We treat the problem of detecting phishing URL as a binary classification problem with phishing URL belong to the positive class and non-phishing URLs belong to the negative class.

We then build a classification model – using Logistic Regression classifier implemented in Weka data mining framework [14] – that attempts to use these features to distinguish phishing and non-phishing URLs.

Logistic Regression (LR): LR [18] is a statistical model for binary classification which is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. The conditional probability that feature vector x has a positive label $y = 1$ is the following:

$$P(y = 1|x) = \sigma(w \cdot x + b) \quad (1)$$

where the weight vector $w \in R^d$ and scalar bias b are parameters to be estimated from training data. The sigmoid function $\sigma(z) = [1+e^{-z}]^{-1}$ gives the probability that feature vector has a positive or negative label, while the variable z represents the exposure of sample x to some set of independent variables. Using a threshold, the right hand side gives the label of the feature vector x.

LR estimates its parameters by optimizing an objection function that closely tracks the error rate. One advantage of LR is that it uses white box model which often has decision rules that are easier to interpret in terms of relevant and irrelevant features.

C. Evaluation Criteria

Classification results were calculated using 10 times 10-fold cross-validation evaluation method, unless stated otherwise. As we formulate the phishing URL detection problem as binary classification problem, each URL falls into one of four possible scenarios: true positive (TP, correctly classified phishing URL), true negative (TN, correctly classified non-phishing URL), false positive (FP, non-phishing URL wrongly classified as phishing), and false negative (FN, phishing URL wrongly classified as non-phishing). Though error rate (fraction of wrongly classified URLs) may be of limited interest in our context where data sets are unbalanced (see next section), we report it anyway to make it easier to compare our results with that from the existing literature. Additionally, we report standard measures such as false positive rate (FPR), false negative rate (FNR), precision, recall, and F-measure which were calculated using the following equations.

$$FPR = \frac{|FP|}{\#\ legitimate\ URLs} \qquad FNR = \frac{|FN|}{\#\ phishing\ URLs}$$

$$precision = \frac{|TP|}{|TP| + |FP|} \qquad recall = \frac{|TP|}{|TP| + |FN|}$$

$$F = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

Note that two types of errors – FPR and FNR – are not of equal importance in detecting phishing. Based on the users' security preference these errors can be tuned using a decision threshold; we discuss this more in section IV.

D. Data Sets

We collected our data from various credible sources. For phishing URLs, use confirmed phishing websites' URLs from PhishTank.com. PhishTank [11] is a collaborative clearing house for data and information about phishing on the Internet. A "phish" once submitted is verified by a number of registered users to confirm it as phishing. We collected first batch of phishing URLs from June 1, 2010 to October 31 of 2010 and call it OldPhishTank data set. We have 11,341 confirmed phishing URLs in this data set. We collected the second batch of phishing URLs from January 1, 2011 to May 3, 2011 and call it NewPhishTank data set. There are 5,456 confirmed phishing URLs in this data set.

In order to address obfuscated URLs with URL shortening services like bit.ly, goo.gl, etc., we used a Python library [13] to expand the shortened URLs to their respective long URLs.

For non-phishing URLs, we use URLs from Yahoo's random page service. A sample page can be generated by visiting *http://random.yahoo.com/bin/ryl*. The link automatically selects a random URL from Yahoo's directory and redirects user to that page. In order to cover wider URL structures, we also made a list of URLs of most commonly phished targets and harvested the hyperlinks from those webpages to also use as non-phishing URLs. We made the assumption, which we think is reasonable, to treat those additional hyperlinks as benign since they were extracted from a legitimate source. We collected 22,213 legitimate URLs from these sources and we call it Yahoo data set. Our second source of legitimate URLs is DMOZ open directory project[1]. There are 9,636 legitimate URLs in this DMOZ data set.

III. Experiments and Results

A. Training with Heuristics

Fig. 1 shows error rate, false positive rate (FPR), and false negative rate (FNR) on OldPhishTank-Yahoo (OY) and NewPhishTank-DMOZ (ND) data sets. On OY data set, LR yields an error rate of 0.43% and false positive rate (FPR) and false negative rate (FNR) of 0.21% and 0.86%, respectively, and on ND data set, it yields comparatively better results of 0.25% error rate, 0.01% FPR, and 0.68% FNR.

LR yields better performance metrics on ND data set. This is not that surprising as OY data set is more representative as it includes URLs harvested from top targets' webpages. FPRs are significantly lower than FNRs on both the data sets. Most of the false positives are due to the non-phishing URLs harvested from top target webpages as they are not in any search engines' indexes; despite their domains being present in the indexes. These URLs are long in nature – perhaps dynamically created with session and other parameters attached to them when the webpages were accessed by our crawlers. Similarly, some phishing URLs – that have been around for a long time; some registered as far back as 2007 – that are in search engines' indexes and also do not have any of the reputation-based heuristics provided false negatives.

Table 1 shows precision, recall, and f-measure evaluation metrics on OldPhishTank-Yahoo and NewPhishTank-DMOZ data sets.

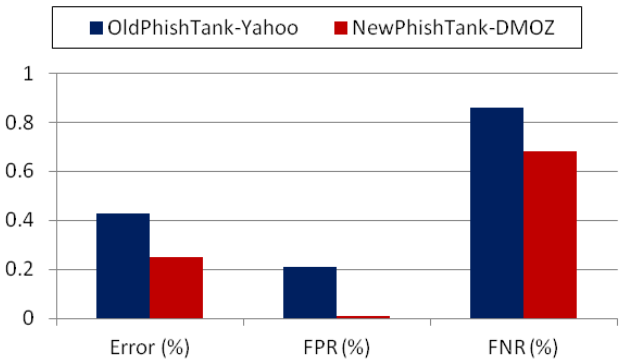

Figure 1. Error rate, false positive rate and false negative rate on OldPhishTank-Yahoo and NewPhishTank-DMOZ data sets using Logistic Regression classifier.

[1] http://www.dmoz.org

TABLE I. PRECISION, RECALL, AND F-MEASURE ON OY AND ND DATA SETS.

Data Set	Precision	Recall	F-measure
OldPhishTank-Yahoo	99.59%	99.14%	99.37%
NewPhishTank-DMOZ	99.98%	99.32%	99.65%

B. *Training with Old Data Set and Testing with New*

As shown in experiments in previous section, training and testing on disjoint subsets of the same data set may yield highly accurate results. Can these results hold when training a classifier with older phishing URLs and testing the model with newer phishing URLs (as in a real-world scenario)? In order to investigate this question, we separated NewPhishTank data set and equal number of randomly selected non-phishing URLs from Yahoo and DMOZ data sets as test data set. We trained Logistic Regression classifier using OldPhishTank and the rest of Yahoo and DMOZ data sets and tested the model with the test data set. Not surprisingly, classifier performed poorly in this scenario yielding 9.33% error rate, 0.15% FPR and a high 18.51% FNR.

In order to address the concept drift, we randomly selected 50% (2,728) of phishing URLs from NewPhishTank data set and added to the training set; and used the rest 2,728 phishing and 5,456 non-phishing URLs as test data set. LR, in this context, achieved much better classification results with 2.81% error rate, 0.99 FPR, and 6.45% FNR. These experiments highlight the importance of judicious selection of training set to achieve better test accuracy and retraining

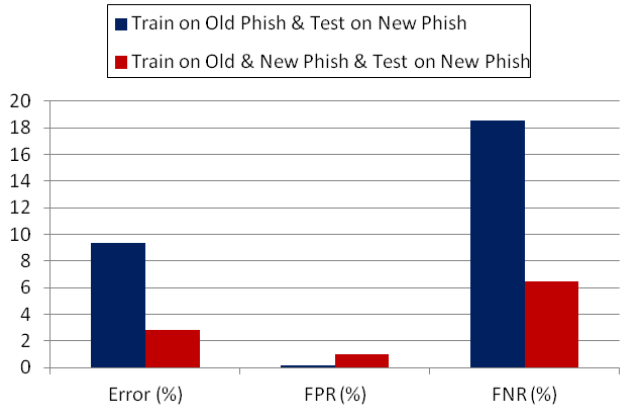

Figure 2. Error rate, false positive rate, false negative rate for training with older phishing URLs and testing with newer phishing URLs using Logistic Regression classifier.

TABLE II. PRECISION, RECALL, AND F-MEASURE WHEN TRAINING ON OLD PHISH AND TESTING ON NEW PHISH USING LOGISTIC REGRESSION CLASSIFIER.

Training and Testing Scenario	Precision	Recall	F-measure
Train on Old Phish & Test on New Phish	99.82%	81.49%	89.73%
Train on Old & New Phish and Test on New Phish	97.93%	93.55%	95.69%

of models with newer and fresh data to overcome the issue of data drift in phishing URL detection.

Other evaluation metrics – precision, recall, and F-measure – for these experiments are displayed in Table II.

C. *Heuristics and Model Analysis*

Table 3 shows the distribution of the heuristics in the combined OYND data set. We find that the search engine-based heuristics are highly prominent in our phishing data sets. Interestingly, a large number (12.35%) of non-phishing URLs were not in Bing's index. Though similar number (~3-4%) of phishing URLs were found in all three search engines' indexes, Google's index had the least number (~5%) of domains indexed from phishing URLs and largest number of domains (~99%) indexed from non-phishing URLs from our data sets. Surprisingly, only 48.65% of the phishing URLs from our data sets were found in Google's phishing blacklist. No non-phishing URL from our data set was found in the phishing blacklist, as expected.

In Logistic Regression because the output of a linear model depends on the weighted sum of the features, the sign and magnitude of the individual parameter vector coefficients can tell us how individual features contribute to a "phishing" or a "non-phishing" prediction. Positive coefficients correspond with phishing features while negative coefficients correspond with legitimate non-phishing features. A zero coefficient means that the corresponding feature will not contribute to the prediction outcome. Interestingly three features – "URL NOT in Top Bing Results", "Domain NOT in Top Yahoo Results", and "URL Contains Top Target" – have negative (non-phishing) coefficients. Not surprisingly, the weights selected for blacklist features are positive and higher indicating that these features are an accurate indicator of phishing.

IV. TUNING FALSE POSITIVES AND FALSE NEGATIVES

In detecting phishing URLs, FPR and FNR may not be of equal importance. Instead of lowering the overall error rate, it may be desirable to tune FPR and FNR. End users may want to tolerate more false positives at the cost of false negatives or vice versa. In case of false positive URLs, users have to be extra vigilant while loading the URL and manually confirm if the webpage is legitimate before submitting any sensitive personal information. Consequently, false negatives may provide false sense of security and users may end up disclosing their personal information to phishers. Large false positive rate may be annoying to the users and large false negative may defeat the purpose of filtering phishing URLs. An ideal system should, therefore, provide low false positive and false negative rates.

Fig. 3 shows this tradeoff between FPR and FNR using a decision threshold *t*. If FPR is set to low 0.1%, the approach yields 10.35% FNR, but if a user can tolerate a little higher FPR of 0.20%, it can achieve a significantly lower FNR of 0.86%.

TABLE III. HEURISTICS AND THEIR STATISTICS AND LOGISITC REGRESSION COEFFICIENTS OBTAINED FROM THE COMPLETE OYND DATA SET.

Heuristic	% Phishing URLs	% Non-phishing URLs	Logistic Coefficient	Odds Ratio
URL NOT in Top Bing Results	96.54	12.35	-3.711	0.0245
Domain NOT in Bing Top Results	88.28	4.56	4.197	66.497
URL NOT in Top Yahoo Results	95.88	6.33	5.699	298.713
Domain NOT in Top Yahoo Results	84.12	4.98	-1.175	0.308
URL NOT in Top Google Results	95.64	1.73	2.849	17.280
Domain NOT in Top Google Results	94.73	0.94	2.141	8.508
URL in Google Phishing Blacklist	48.65	0.00	162.172	2.69334 E70
URL in Google Malware Blacklist	0.34	0.04	1.4162	4.1216
URL in Google RegTest List	23.24	3.86	134.162	1.8437E 6
URL Contains Top Domain	23.24	3.86	1.1573	3.181
URL Contains Top Target	27.39	5.28	-0.7123	0.490
IP in Stopbadware Top 50	1.80	0.75	0.452	1.571
IP in PhishTank Top 10	26.92	1.06	2.6888	14.714
Domain in hpHosts	3.47	0.01	7.0536	1156.995

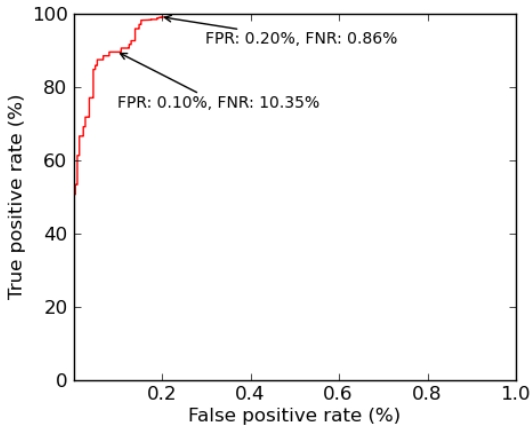

Figure 3. An ROC Curve showing tradeoff between FPR and FNR using LR classier on OldPhishTank-Yahoo data set. Note that FPR ranges between 0 and 1%.

V. LIMITATIONS AND POTENTIAL ADVERSARIAL ATTACKS

Though the overall classification results given by the approach is impressive, we can spot some performance limitations. Analysis of heuristics (section III.A) shows that search engine-based features are highly discriminative features in determining phishing URLs. The same feature set also contributes to a major performance bottleneck due to the time lag involved in querying Google and other search engines.

To render search engine-based features inaccessible, attackers may try to DDoS search engines; but it is not very likely to happen on all three of them simultaneously.

Though index size of a search engine plays a big role in our approach, we can always look for an alternative search engine while there are plenty to choose from. Using Blackhat search engine optimization (SEO) techniques [19], adversaries can get their phishing websites crawled and indexed in a short period of time. Our technique may provide a large number of false negatives if phishing links are found in search engines' indexes. However, the attackers may be discouraged to manipulate indexes of all major search engines in a short period of time. Consequently, manipulating search engine results and altering the PageRank of a phishing page require significant investment, which reduces the potential profit from the phishing campaign.

Conversely, our method may flag new legitimate web pages as phishing in particularly those that have not yet been crawled and indexed by search engines.

To make reputation-based features less suspicious, phishers may try to host their webpages in domains and IPs that do not have any historic reputation of hosting malicious or phishing websites. Sites with good reputation, however, are either too difficult to exploit or their administrators typically remove malicious pages under their control promptly, thus limiting the potential audience and profitability of phishing campaign hosted in their web servers.

Attackers may leverage well-known infrastructure such as hosting phishing page on a legitimate popular domain such as free webhosting services or by breaking into legitimate web sites or by exploiting common Cross-site Scripting (XSS) vulnerabilities [20]. We can overcome this drawback by looking into the contents of the web pages. This drawback, however, is not particular to our approach, but to all the approaches that rely only the URL metadata and structures to detect potential maliciousness.

VI. RELATED WORK

The work by Garera et al. [12] is the most closely related to our work. They use logistic regression over 18 hand-selected features to classify phishing URLs. The features include the presence of certain red flag key words in the URL, features based on Google's Page Rank and Google's web page quality guidelines. They achieve a classification accuracy of 97.3% over a set of 2,500 URLs. Though similar in motivation, our approach differs significantly in both methodology (considering only web mining-based features and scale (considering an order-of-magnitude more training examples).

Ma et al. [8] propose a method to classify malicious URLs using lexical and host-based properties of the URLs. Using these features, they compare the accuracy of four

classifiers: Naïve Bayes, SVMs with an RBF kernel, and a linear kernel, and ℓ_1-regularized logistic regression.

Zhang et al. [9] present CANTINA, content-based approach to detecting phishing websites, based on the TF-IDF information retrieval algorithm. By using a weighted sum of 8 features (4 content-related, 3 lexical, and 1 WHOIS-related) they show that CANTINA can correctly detect approximately 95% of phishing sites. The goal of our approach is to avoid downloading the actual web pages and thus reduce the potential risk of analyzing the malicious content on user's system. In order to achieve this goal, we evaluate only Meta data on URLs.

Ludl et al. [17] propose 18 hand-selected features by studying page structures of phishing webpages. Using C4.5 classifier on 5,751 phishing webpages and 4,335 legitimate webpages, their approach achieves true positive rate of 83% and false positive rate of 0.4%.

In [10], Whittaker et al. use a proprietary classifier to analyze millions of pages a day, examining the URL and the contents of a page to determine whether or not a page is phishing. Their system classifies web pages submitted by end users and URLs collected from Gmail's spam filters.

Traditional rule-based approach has been applied in detecting phishing webpages [16]. In related context, heuristics and email contents have been used to detect phishing emails [7], [15].

VII. Conclusions and Future Work

We have proposed a novel approach for classifying phishing URLs or non-phishing using supervised learning across features extracted from various Web services. Applying the Web mining-based heuristics on Logistic Regression classifier, we experimentally demonstrated that phishing URLs can be detected with an accuracy of more than 99% and false positive and false negative rates of less than 1% with respect to real-world data sets.

Though there may be some performance bottlenecks – if the system is deployed in real-world phishing detection application – we show that mining the Meta information on a URL across the Web can, once trained, effectively detect a potentially dangerous URL and thus help Internet users from avoiding those sites.

As future work, we plan to integrate these heuristics with keyword, lexical, host, and content-based features to improve the state-of-the-art in detecting phishing webpages. Including the context in which the URLs are distributed, perhaps, could also help improve the accuracy of the system. Because of its light-weight approach (using Meta data only on URLs) and its high classification accuracies, the system has a great potential to be used as a real-time phishing URL classification system. It, therefore, would be interesting to see the time taken – including time taken to gather heuristics – to classify a URL by the system.

Acknowledgment

Support from Institute for Complex Additive Systems Analysis (ICASA), a research division of New Mexico Tech, is gratefully acknowledged.

References

[1] J. Milletary, "Trends in Phishing Attacks," http://www.us-cert.gov/reading_room/phishing_trends0511.pdf

[2] APWG Phishing Activity Trends Report- 2nd Half 2010, http://apwg.org/reports/apwg_report_h2_2010.pdf (2011)

[3] M. Cova, C. Kruegel, G. Vigna, "Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code," Proc. World Wide Conference, Releigh, North Carolina, 2010, pp. 281-290.

[4] StopBadware - IP Address Report – Top 50 by Number of Reported URLs, http://stopbadware.org/reports/ip

[5] Google Safe Browsing API. http://code.google.com/apis/safebrowsing/

[6] hpHosts Online - Simple, Searchable & FREE!, hpHosts: http://hosts-file.net/

[7] R. B. Basnet, S. Mukkamala, A. H. Sung, "Detection of phishing attacks: A machine learning approach," Prasad, B. (ed.) Studies in Fuzziness and Soft Computing, vol. 226, Springer, Heidelberg, 2008, pp. 373-383.

[8] J. Ma, L. K. Saul, S. Safage, G. M. Voelker, "Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs," Proc. ACM SIGKDD, Paris, France, 2009, pp. 1245-1253.

[9] Y. Zhang, J. Hong, L. Cranor, "CANTINA: A Content-Based Approach to Detecting Phishing Web Sites," Proc. WWW 2007, Banff, Alberta, Canada, ACM Press. 2007.

[10] C. Whittaker, B. Ryner, M. Nazif, "Large-Scale Automatic Classification of Phishing Pages," Proc. 17th Annual Network and Distributed System Security Symposium, California, USA, 2010.

[11] PhishTank - Out of the Net, into the Tank, http://www.phishtank.com/developer_info.php

[12] S. Garera, N. Provos, M. Chew, A. D. Rubin, "A Framework for Detection and Measurement of Phishing Attacks," Proc. 5th ACM Workshop on Recurring Malcode (WORM '07), ACM Press, New York, 2007, pp. 1-8.

[13] PyLongURL - Python Library for LongURL.org, http://code.google.com/p/pylongurl/

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, "The WEKA Data Mining Software: An Update. ACM SIGKDD Explorations" 11, 2009, pp. 1-8.

[15] R. B. Basnet, A. H. Sung, "Classifying Phishing Emails Using Confidence-Weighted Linear Classifiers," Proc. International Conference on Information Security and Artificial Intelligence, Chengdu, China, 2010, pp. 108-112.

[16] R. B. Basnet, A. H. Sung, Q. Liu, "Rule-Based Phishing Attack Detection" Proc. International Conference on Security and Management (SAM'11), Las Vegas, NV, 2011, pp. 624-630.

[17] C. Ludl, S. McAllister, E. Kirda, C. Kruegel, "On the Effectiveness of Techniques to Detect Phishing Sites," Proc. International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '07), Springer-Verlag, 2007, pp. 20-39.

[18] S. le Cessie, J. C. van Houwelingen, "Ridge Estimators in Logistic Regression," Applied Statistics. 41, 1992, pp. 191-201.

[19] Anvil, Search Engine Optimization Whitepaper, http://www.anvilmediainc.com/wp-content/uploads/ami_seo_whitepaper_1104.pdf

[20] Cross-site Scripting (XSS), http://www.owasp.org/index.php/Cross-site_Scripting_(XSS)