Lexical Feature Based Phishing URL Detection Using Online Learning

Aaron Blum
Department of Computer and Information Sciences
The University of Alabama at Birmingham
Birmingham, Alabama
aaron.blum@gmail.com

Brad Wardman
Department of Computer and Information Sciences
The University of Alabama at Birmingham
Birmingham, Alabama
bwardman@cis.uab.edu

Thamar Solorio
Department of Computer and Information Sciences
The University of Alabama at Birmingham
Birmingham, Alabama
solorio@cis.uab.edu

Gary Warner
Department of Computer and Information Sciences
The University of Alabama at Birmingham
Birmingham, Alabama
gar@cis.uab.edu

ABSTRACT

Phishing is a form of cybercrime where spammed emails and fraudulent websites entice victims to provide sensitive information to the phishers. The acquired sensitive information is subsequently used to steal identities or gain access to money. This paper explores the possibility of utilizing confidence weighted classification combined with content based phishing URL detection to produce a dynamic and extensible system for detection of present and emerging types of phishing domains. Our system is capable of detecting emerging threats as they appear and subsequently can provide increased protection against zero hour threats unlike traditional blacklisting techniques which function reactively.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—Security and protection (e.g., firewalls); I.2.6 [**Artificial Intelligence**]: Learning—Parameter Learning

General Terms

Algorithms, Experimentation, Security

Keywords

phishing detection, online-learning, confidence-weighted classifier

AISec'10, October 8, 2010, Chicago, Illinois, USA.

1. INTRODUCTION

Detection of phishing URLs has become increasingly difficult due to the evolution of phishing campaigns and their efforts to avoid mitigation by blacklists. The current state of cybercrime has made it possible for phishers to host campaigns with shorter lifecycles, diminishing blacklist effectiveness [17]. At the same time, standard supervised learning algorithms are known to generalize well over the specific patterns observed in training data, which makes them a better alternative against phishing campaigns. However, the highly dynamic environment of these campaigns demands updating the models regularly, and this poses new challenges since most of the typical learning algorithms are also computationally expensive to retrain. A more cost-effective solution is to explore online learning approaches, like the ones proposed by Ma et al. for identifying malicious URLs [13, 12]. Unlike traditional learning algorithms, online approaches update their model after processing each test sample.

In this paper, we follow the trend of online learning and explore a real-time approach to phishing URL detection. Our model uses a largely lexical model trained on output from a robust content-inspection based approach [18]. Different from previous work, we do not make use of any host-based features. Our feature set is composed of surface level features derived automatically from the URLs. We demonstrate that with an appropriate training set a confidence weighted approach can achieve a high degree of accuracy classifying previously unseen URLs. This indicates that a model operating solely on data derived by inspection of a URL performs as well as content inspection based systems and can potentially eliminate many of the costs and security risks associated with such systems [16, 8]. We also present a more stringent and realistic evaluation by testing on two different feeds of URLs that provide a broad sampling of phishing URLs. The end result of this work is the conceptual blueprint for a system that could be deployed with minimal committed human resources, yet achieve a high level of labeling accuracy.

The next section reviews recent work on the detection and prevention of malicious URLs. The lexical feature model and a high level description of the online learning algorithm used as the underlying classifier are presented in Section 3. This section also discusses the Deep MD5 Matching algorithm used to generate the gold standard data. Then in Section 4 we present in more detail the two data sets we used to train and evaluate our results. The empirical evaluation setting is discussed in Section 5. Then in Section 6 we present the experimental results followed by a discussion on our findings in Section 7. The paper concludes with a summary of the approach and findings, together with potential avenues for future research in Section 8.

2. RELATED WORK

Researchers have examined a variety of techniques for preventing phishing attacks. Anti-phishing techniques can be placed in three categories: email-based, content-based, and URL-based detection methods. Researchers have employed machine learning techniques on feature sets derived from phishing emails [2, 7]. Other studies investigated classifiers using URL features and search engine results as a means for high detection rates while maintaining low false positive rates [19, 20, 15]. Chandrasekaran et al. [3] attempt to determine phishing emails using structural properties. These anti-phishing techniques are not always applicable as the approaches require phishing emails.

Other researchers have attempted to use the content of websites as a means for identifying phishing websites. Wardman and Warner [18] use a technique that computes the similarity between the content files from potential and known phishing websites. Ludl et al. [11] classify phishing websites using features extracted from the main phishing webpage. Medvet et al. [14] compare the visual similarity of phishing websites to legitimate websites. Content-based approaches, however, require access to the phishing website.

Several studies have explored applications of online learning in URL-based classification. Most relevant to this paper is Ma et al.'s work exploring different types of online learning algorithms in this context [13, 12]. Though some use of lexical features was made, there was no apparent effort to separate the lexical (URL based features) and host based, such as IP, connection speed, or registrar information. We intend to open for review the possibility of relying on features derived purely from URLs for classification.

PhishNet was recently proposed as an attempt to extend the usability of blacklists [10]. By means of heuristics PhishNet generates new URLs based on pre-existing URLs in blacklists. The newly generated URLs are filtered by selecting only those that can be resolved by a DNS lookup. In their experiments, only 14% out of 1.5 million URLs had a DNS entry. PhishNet labels all the remaining URLs as phish as long as there is at least a 90% of similarity content between the newly generated URLs and the URLs in the blacklists. The second part to PhishNet performs a soft match of URLs with those in the blacklist by breaking down the URLs into components: domain, top level domain, directory, filename, and query string. Different components are assigned different weights and a threshold function determines which URLs are labeled as phish. The success rates reported for PhishNet are promising. However, it is not clear how much more coverage this approach has than blacklisting as the newly generated URLs depend heavily on the initial seed set (the original blacklists) and an exhaustive generation of new URLs from here would be a time consuming approach.

Felegyhazi et al. explored a proactive approach for domain blacklisting [6]. They begin with a seed list of blacklisted domains and then expand the list by using information on the DNS zone file data and "WHOIS" domain registration data. The intuition behind this approach is that cybercriminals will need to register a large number of domains to keep their activities going. In addition, they exploit name server features, such as the freshness of the name server registration and self-resolution. The evaluation results show this approach can greatly decrease the timespan to blacklisting for a large percentage of domains (60% - 75%). One drawback from this approach though is that it depends on the availability of name server information in the zone file and WHOIS database. The former might not be always available and access to the WHOIS database can turn into a bottleneck. Moreover, according to [1], as many as 78% of phishing websites are hosted on hacked domains and therefore were registered by legitimate registrants, which implies that this approach will fail for 78% of phishing websites.

3. APPROACH

This work focuses on the exploration of surface-level features from URLs to train a confidence-weighted learning algorithm. The idea is to restrict the source of possible features to the character string of the URL and avoid having the vulnerability of extracting host-based information. Each URL is represented as a vector of binary features that are fed to the online algorithm, namely the confidence-weighted approach developed by Dredze et al. [5], during training. At time of testing, previously unseen URLs are then mapped to this binary feature vector. The learner processes this new vector and outputs the final class (phish, non-phish).

Features are recorded using a bag of words approach. To facilitate feature extraction, each URL was split into three sections: protocol, domain, and path. All subsequent feature extraction was performed on these sub-regions. The feature types (or bags) are noted in Table 1. The offsets are noted in the table using curly braces and refer to the zero based token distance from the right most end of a region. Tokens are formed by splitting a region on all of the following values, "/", "?", ".", "=", and "_". As an example, the domain{2} of "`mail.google.com`" refers to the token "`mail`". Double offsets indicate a bigram pair. For example, path{1}{0} in the path "`/app/index.html`" refers to the "`index-html`" bigram.

These feature groups capture some of the critical elements in malicious URLs. Domain tokens in this context behave similarly to a fuzzy blacklist as tokens that show up often in malicious URLs will make the algorithm more likely to classify a URL as malicious. Benign tokens will have the opposite effect confirming that a URL is safe. This alone is not adequate as many phishing domains included legitimate domain tokens as illustrated in Table 2. However, these tokens will not appear at the root level as they would in a legitimate domain. Subsequently, we also record position sensitive tokens and bigrams for the first several levels of a domain which provides for token context sensitivity in our model.

The same is true of the tokens in the path as they use auto-exploit phishing kits that attach themselves to vulner-

Table 1: Lexical feature groups

Feature Group	Number of Features
Length	10
IP vs Domain	1
Protocol	6
IP (First Octet)	114
IP A.B (First and Second Octet)	704
TLD (top level domain)	220
Domain{1}	16,572
Domain{2}	7,933
Domain{3}	1,511
Domain{all others}	5,407
Domain{1}{0}	16,934
Domain{2}{1}	16,116
Domain{3}{2}	4,747
Domain Tokens (all)	27,576
Path Tokens (all)	49,359
Path Bigrams (all)	100,401
LPT (last path token)	17,508
Path{1}	11,465
Path{2}	8,503
Path{3}	8,456
Path{4}	4,695
Path{1}{0}	20,510
Path{2}{1}	19,889
Path{3}{2}	17,587
Path{4}{3}	13,361
Total	369,585

Table 2: Phish vs benign URLs

Benign	Phish
`www.paypal.com`	`www.paypal.fr.j-ksa.com`
	`paypal.cactus-mall.com`

able modules on an existing site. This leads to a URL with a completely legitimate domain and seemingly benign path. The final path tokens, however (following the exploited module), will be the base of the phishing kit. In anticipation of such URLs, we record location groups for path tokens and bigrams as well.

3.1 The Confidence Weighted Algorithm

Our confidence-weighted model for URL classification is based on that described by Ma et. al. [13]. The mathematical operation of the model is effectively unchanged; however, our feature set does not include any host-based features and has a significantly more extensive lexical feature set.

This approach originally introduced by Dredze and Crammer [5] is a linear classification utilizing individual confidence-weighted parameters to improve overall accuracy and flexibility in classification. The association of an individual confidence factor for each parameter allows a model to automatically react and correct for parameters which are highly indicative of a sample's class and those that are not. Further, this allows the model to automatically adjust when parameters change in significance, as often occurs throughout the course of a phishing campaign [17].

This model maintains a mean μ and standard deviation σ representative of the class and confidence for each feature. The class of a new data member represented by a feature vector x is determined by computing the sign of $w \cdot x$, where $w \sim N(\mu, \Sigma)$ and Σ represents the covariance matrix with a diagonal of σ, and zero for all off-diagonal elements. We use the improved form of this algorithm covered in detail by Crammer et al. [4].

To extract reference labels from the UAB Phishing Data Mine we use the Deep MD5 Matching algorithm developed by [18]. The next subsection describes this approach in more detail.

3.2 Deep MD5 Matching

In addition to simple URL and rule based blacklists, financial institutions and anti-phishing organizations commonly compare the MD5 hash of the main index pages of previously confirmed phishing websites to potential phishing URLs. The URLs are confirmed and branded when the MD5s of the main index pages match. However, simple obfuscation techniques can bypass this methodology by varying the content (and subsequently MD5s) of the main index pages while still using the same primary content files in a phishing kit. This is often achieved through techniques such as including the current time on the web page or passing the recipient's email as a parameter and presenting it on the main index page. Table 3 shows an example of the latter, where the recipients' email addresses are passed through the parameter "`login_email`".

In order to overcome this, a technique called Deep MD5 Matching was developed to compare the similarity between the content files from potential and known phishing websites. This approach downloads all of the content files associated with a potential phishing website using GNU's wget. Content often includes images, scripts, and style sheets that are used in association with the main index page. This set of content files is compared to sets of files from previously confirmed phishing websites. The similarity between two websites' content files is determined by the value of their Kulczynski 2 coefficient [9]. If the result of a potential and known phishing websites' Kulczynski 2 coefficient is greater than 0.75, then the potential website is confirmed as a phishing URL. 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 any file in set 1.

$$Kulczynski2 = 0.5 \times \left[\frac{a}{(a+b)} + \frac{a}{(a+c)} \right] \quad (1)$$

The result of Equation 1 measures the similarity between two file sets by taking the average of the proportion of matching files. The Kulczynski 2 similarity coefficient was ideal for this task since it gives equal weight to each set percentage irrespective of the number of files it contains.

4. DATA

We make use of two distinct and extensive phishing datasets. Both provide a broad sampling of phishing URLs encountered in corporate environments. Our primary training data comes from the UAB Phishing Data Mine, an extensive database of URLs collected over two and a half years of anti-phishing research. The primary testing data was pro-

Table 3: Example of URLs including the recipient's email address

```
http://www.paypal.com.ufiyr4gscz.125tcb5cbquts9howt09.com/cgi-bin/webscr/
?login-dispatch&login_email=UABtest1@uab.edu&ref=pp&login-processing=ok

http://www.paypal.com.e20jqm91gysjhz7yt.125ci3qk5uipl4wo3hr3.com/cgi-bin/
webscr/?login-dispatch&login_email=UABtest21@uab.edu&ref=pp&login-processing=ok

http://www.paypal.com.0o4589zq8stnemcjy.125kpszbkwapqkvzhkp3.com/cgi-bin/
webscr/?login-dispatch&login_email=UABtest3@uab.edu&ref=pp&login-processing=ok

http://www.paypal.com.uuyfzt55pc2od2l0z5l.125yruv5rynnjj4jbt52.com/cgi-bin/
webscr/?login-dispatch&login_email=UABtest4@uab.edu&ref=pp&login-processing=ok
```

Table 4: Examples of benign and URLs in the UAB Phishing Data Mine

Benign URLs	Malicious
http://link.charterone.com/r/VMYSIE/WLUJ1/ 5TWNJ/YBLYV/7MP1B/A7/t	http://www.gtmweb.com.br/1/2/CAM10-jsessionid= 000026MQ7KnXUxsKmiYKszFUkGJ12c58ti63.htm
http://factips.com/satellitetvtopc/script/net/ BANKAWAY.html	http://www.clube3.com.br/galeria/bradesco/log/ site/perfil/
http://aigcorpebus.com/a/ tBJLYIyAJwDPhB7Xgat$Loz2iJE/mastercard	http://dekoracjestolu.pl/images/1sex1.php
http://purepolicy.com/cs/www.genworth.com/ index.html	http://www.munilince.gob.pe/bri/index.htm
http://chilp.it/503fca	http://www.portalbancodobrasil.com/portalbb/ aapf/login/index.bb

vided by a trusted research partner, Cyveillance. Both feeds have a wide variety of URLs and portray many of the nuanced facets of modern URLs. The UAB Phishing Data Mine presents an especially interesting data set as all URLs appearing in the feed were marked as phish by some algorithm or user prior to their processing at UAB. This means that even the "benign" URLs from this feed may appear "phishy" lexically as illustrated in Table 4. When compared to actual phishing URLs, these URLs exhibit many of the same characteristics. The feed from Cyveillance also has lexically similar phishing URLs as shown in Table 5.

All feeds are fully de-duplicated, meaning that malicious or benign URLs only occur once regardless of how many times they are reported or encountered. To avoid the possibility of testing on data that we had already trained on we prepared a subset of the Cyveillance feed (the "Abridged" feed) with all overlapping UAB URLs removed. We present accuracy for both the full and abridged feeds in our results section.

In an effort to sample the level of diversity in our training and testing feeds, we examined the number of unique domains represented by both data sets. The Cyveillance feed appears to cover a wider range of domain names than UAB's Phishing Data Mine. Cyveillance had 18,990 unique domains from a total of 34,234 URLs (all malicious). UAB however, had 9,506 unique domains in its 25,203 URLs (6,114 malicious). Furthermore, as shown in Figure 1, the 100 most common domains make up greater than 50% of the UAB data while the same number of domains only represents about 11% of the Cyveillance feed. The two malicious data sets consisted of URLs targeting a diverse number of phished organizations. The UAB data contained URLs targeting 105 different organizations with the brands having a URL count mean of 58.2 and median of 6. The number of organizations phished in the Cyveillance feed is not fully known by these researchers; however, when considering the 8,500 overlapping URLs between the UAB and Cyveillance data sets, 72 organizations were targeted.

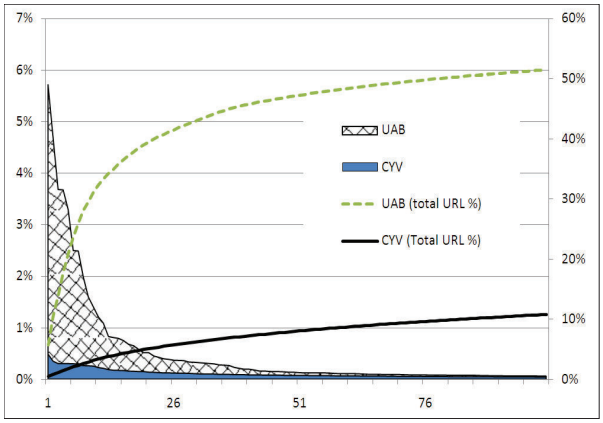

Figure 1: Percentage of total URLs vs. Individual Domains

Table 5: Examples of malicious URLs in the Cyveillance feed

http://kwii.or.kr/data/board/ing/hsbc
http://www.trans-aubonline.com/AUB-Online.html
http://nuo-la.com/elements/index.html
http://samcoline.com/images/stories/promotions/2010/Q2/dat/hsbc/
http://h4oplusplus.org/nwolb%27/default/index_1.html

5. EXPERIMENTAL SETTING

Our primary training and testing was conducted on daily batches of URLs from each feed. Training was initially conducted on UAB Phishing Data Mine data. We believe that the UAB dataset, generated by Wardman and Warner's Deep MD5 algorithm, provides a reasonable mixture of URLs and contains a sufficient volume to support meaningful testing. Detection rates were then computed for the Cyveillance abridged, Cyveillance full, and UAB feeds.

In order to emulate a deployed system and ease data-processing, primary testing was performed in daily "slices" simulating a model updated by a daily URL blacklist/whitelist feed. In the ideal case, the model will be updated constantly to reflect emerging threats from the moment that they are detected. The rapidly changing nature of phishing domains dictates the prudence of updating detection as rapidly as possible [17]. To further reinforce the value of continuous updating, we have also tested using a continuous interval (training after processing each URL).

False positive and false negative error rates were computed based on how many times the algorithm misclassified a URL relative to the total number of URLs classified. In the continuous training/testing runs, the model is trained on each URL after it is used to test for classification accuracy.

6. RESULTS

In our first round of experiments, Figure 2 shows that our model reaches strong detection rates for homogenous (within UAB) data and relatively consistent detection rates on both the abridged and full Cyveillance data sets. The disparity between the accuracy of the two feeds likely stems from differences in major determining features. As we show later, classification accuracy greatly improves when the model is allowed to train on members from both feeds. Both Cyveillance and UAB feeds appear to have similar URL volume densities, potentially suggesting similar coverage of global phishing URL volume.

To illustrate the value of broadening the range of testing data, we prepared a fourth URL set from the combination of the UAB and Cyveillance data. This was tested in the same way as the original homogenous tests performed on the UAB data set and consistently reaches better accuracy than tests run on purely UAB data (Figure 3). This strongly suggests that more diverse training data will improve detection rates as features from additional feeds may enhance detection rates over a singular data source.

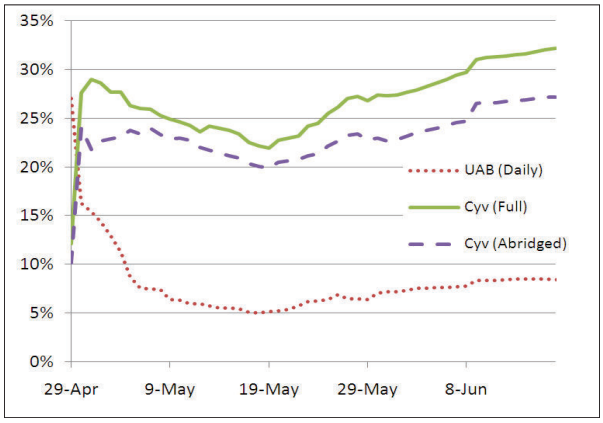

Figure 2: Cumulative daily percentage error rates for UAB, Cyveillance, and Cyveillance abridged

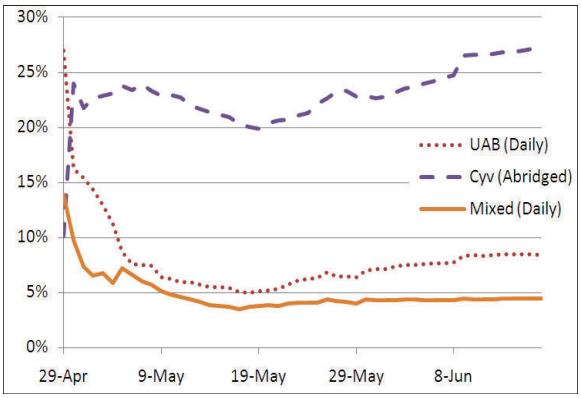

Figure 3: Cumulative daily percentage error rates for UAB plus Cyveillance (Mixed), UAB daily, and Cyveillance abridged

Finally, as a further illustration of the value gained by updating the model continuously versus daily, we tested the mixed data set in both continuous and daily modes (Figure 4). The results are not surprising, the accuracy of the model is higher when its parameters are tweaked after each URL has been predicted. Thus changes in the patterns of the phishing campaigns can be addressed earlier in the process resulting in more accurate predictions.

7. DISCUSSION

Our selected features appear to be highly successful achieving cumulative error rates as low as 3%. Even lower rates (around 2%) were achieved when previous data was mixed with new data from the Cyveillance feed. Homogeneous testing on the UAB data feed shows a 4-5% decrease in accuracy when shifting from continuous to daily training (see Figure 2). Nevertheless, on the mixed set this loss was only about 2%, indicating that a sufficiently rich training set can help

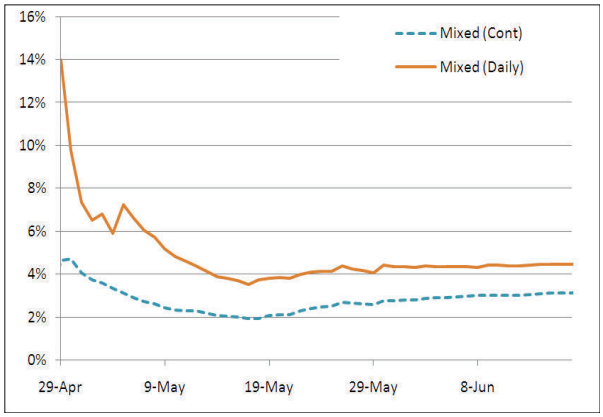

Figure 4: Comparison on cumulative error percentage rates for Mixed daily vs Mixed continuous. The Mixed data is the combination of the UAB Phishing Data Mine and Cyveillance feeds.

offset any loss of accuracy incurred by a delay in training data. These error rates were especially impressive as they are comparable to previous work [13] that depended on a combination of lexical and host based features.

The erratic rise and fall of the Cyveillance testing results likely arise from the fact that there is little overlap on the actual URLs and phishing campaigns represented by the two feeds. This indicates that the classification depends entirely on common features (like the terminal path string or subdomain format). Full understanding of this relationship would require more knowledge of how the Cyveillance and UAB feeds are generated, something that we are not currently privy to.

The disparity between the full Cyveillance feed and the feed with overlapping UAB URLs seems to indicate that the overlapping URLs from the UAB feed may be more difficult to classify lexically than the remaining Cyveillance ones. It seems like a strong conjecture that the Cyveillance feed is generated (at least in part) by a lexically sensitive algorithm. Due to the largely independent features used for classification, this approach is resilient to common practices employed by phishers such as modification of domain names or modification of domain substrings. This extends to all features, including path tokens, as any feature that begins to provide less accurate classification information will have its weight immediately adjusted by the model. It might be argued that phishing campaigns that simply replace a legitimate site index page with their own would potentially be mislabeled by this system. While this is true, compromised top level domains tend to be discovered and repaired more rapidly than those hidden in module paths (like those in Table 5. They also fail to provide the URL with the lexical information meant to mislead the user (e.g `legitmate.domain.com/modules/ebay.security/sign-in/login.php`). These two restrictions potentially decrease the value of an individual phish and increase the overall work required to run a campaign.

8. CONCLUSION

In this paper we explored the possibility of using a confidence weighted model trained on features derived exclusively from URLs for classification. This approach differs from that of previous work we are aware of in that it does not have a dependency on any host based features. Training on a single, labeled feed and testing on another completely separate feed show that this lexical feature based classification approach is robust and can certainly be integrated into current phishing detection systems. Results of daily training and testing indicate that the model is highly capable of identifying both new (not previously seen URLs) and URLs from another distinct feed. Furthermore, when provided with quality diverse training data our approach can reach higher than 97% classifying accuracy on emerging phishing URLs when trained continuously. This is of particular interest because it rivals the level of accuracy achieved by conventional content inspection based approaches without the delay, security risks, and overhead associated with examining content. While we illustrate our classification approach with training data from the UAB Phishing Data Mine it should be noted that training data can be drawn from any (or multiple) malicious URL feeds. Any sufficiently dynamic (and accurate) malicious URL feed can be coupled with this classification approach to yield a real-time classification system with a high degree of accuracy.

Our approach uses a relatively simple feature set extracted through a simple parser. There are more features that could be experimented with that would potentially add value to the model and improve accuracy. Additionally, expansion of the scale of experiments to include additional real-world data sets may provide better detection rates. Study of training data sets and their dynamics might also provide insight as to what qualities an ideal training set or sets might exhibit.

One relevant direction for future work involves the combination of active learning techniques in this problem of phishing URLs detection. While the strength of the online learning algorithms, such as the confidence-weighted algorithm used here, lies in the reduced cost for updating the model, this updating requires the availability of the reference label for each new instance, which can turn into a bottleneck. However, by carefully selecting which URLs could be more useful in decreasing the uncertainty of the model, the cost of acquiring the reference labels could be reduced dramatically.

9. ACKNOWLEDGEMENTS

We would like to graciously thank our research partner Cyveillance for the use of their data and the Department of Computer and Information Sciences at the University of Alabama at Birmingham for their support. We also thank the four anonymous reviewers for their valuable comments.

10. REFERENCES

[1] G. Aaron and R. Rasmussen. Global phishing survey 2h/2009. In *Counter eCrime Operations Summit IV*, Säo Paulo, Brazil, May 2010.

[2] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nai. A comparison of machine learning techniques for phishing detection. In *APWG eCrime Researchers Summit (eCRS)*, Pittsburgh, PA, October 2007.

[3] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya. Phishing e-mail detection based on structural

properties. In *Academic Track of the 9th Annual NYS Cybersecurity Conference*, Albany, NY, June 2006.

[4] K. Crammer, M. Dredze, and F. Pereira. Exact convex confidence-weighted learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[5] M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 264–271, Helsinki Finland, 2008.

[6] M. Felegyhazi, C. Kreibich, and V. Paxson. On the potential of proactive domain blacklisting. In *3rd USENIX Workshop on Large-Scale Exploits and Emerging Threats*, San Jose, CA, April 2010.

[7] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect phishing emails. In *International World Wide Web Conference (WWW)*, pages 649–656, Banf, Alberta, Canada, May 2007.

[8] Technical Info. http://www.technicalinfo.net/papers/xmorphic.html. Online, 2010 (last accessed).

[9] S. Kulczynski. Die pflanzenassoziationen der pieninen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, pages 57–203, 1927.

[10] M. Kumar, P. Prakash, M. Gupta, and R. R. Kompella. Phishnet: Predictive black-listing to detect phishing attacks. In *Proceedings of IEEE Infocom (Mini-Conference),(Infocom)*, pages 1–5, San Diego, CA, March 2010.

[11] C. Ludl, S. McAllister, E. Kirda, and C. Kruegel. On the effectiveness of techniques to detect phishing sites. In *Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA)*, Switzerland, July 2007.

[12] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learning to detect malicious web sites from suspicious urls. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1245–1253, Paris, France, June–July 2009.

[13] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Identifying suspicious urls: An application of large-scale online learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML-2009)*, pages 681–688, Montreal, Quebec, Canada, 2009.

[14] E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In *IEEE International Conference on Security and Privcay in Communication Networks*, Istanbul, Turkey, September 2008. IEEE Computer Society Press.

[15] D. Miyamoto, H. Hazeyama, and Y. Kadobayashi. An evaluation of machine learning-based methods for detection of phishing sites. In *15th International Conference on Neural Information Processing of the Asia-Pacific Neural Network Assembly, ICONIP 2008*, pages 539–546, Auckland, New Zealand, November 2008.

[16] Finjan Vital Security. http://www.finjan.com/mcrcblog.aspx?entryid=2278. Online, June 2009.

[17] S. Sheng, B. Wardman, G. Warner, L. Cranor, and C. Zhang. An empirical analysis of phishing blacklists. In *Sixth Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, July 2009.

[18] B. Wardman and G. Warner. Automating phishing website identification through deep MD5 matching. In *eCrime Researchers Summit 2008*, pages 1–7, October 2009.

[19] C. Whittaker, B. Ryner, and M. Nazzif. Large-scale automatic classification of phishing pages. In *The 17th Annual Network and Distributed Security Symposium (NDSS)*, 2010.

[20] Y. Zhang, J. Hong, and L. Cranor. Learning to detect phishing emails. In *CANTINA: A Content-Based Approach to Detecting Phishing Web Sites*, pages 639–648, Banf, Alberta, Canada, May 2007.