A Survey of Spam Detection in Web Environment

Shweta Prashnani
Master of Technology (CTA)
Gyan Ganga College o Technology

Balram Purshwani
Assistasnt Professor
Gyan Ganga College of Technology

Abstract - Traditional content-based e-mail spam filtering takes into ac-count content of e-mail messages and apply machine learning techniques to infer patterns that discriminate spams from hams. In particular, the use of content-based spam filtering unleashed an unending arms race between spammers and l-tier developers, given the spammers' ability to continuously change spam message content in ways that might circumvent the current filters. In this paper, we perform survey on the horizons of content-based filters by taking into consideration the content of the Web pages linked by e-mail messages. We make survey for extracting pages linked by URLs in spam messages and we characterize the relationship between those pages and the messages. We then use a machine learning technique (a lazy associative classifier) to extract classification rules from the web pages that are relevant to spam detection. We demonstrate that the use of information from linked pages can nicely complement current spam classification techniques, as portrayed by Spa-assassin. Our study shows that the pages linked by spam's are a very promising battleground.

Keywords: URL, Spam.

I. INTRODUCTION

Spam fighting is an \arms race" characterized by an in-crease in the sophistication adopted by both spam filters and spammers [12, 14]. The co-evolution of spammers and anti-spammers is a remarkable aspect of the anti-spam battle and has motivated a variety of works that devise adversarial strategies to treat spam as a moving target [6, 4].

On the spammers' side, the standard counter-attack strategy to face content-based filters is to obfuscate message con-tent in order to deceive filters. In the early years of the spam arms race, obfuscation techniques were as simple as misspelling Viagra as V1agra, but have evolved to complex HTML-based obfuscations and the use of images to prevent action of text-based filters. However, spammers face a trade-o : their final goal is to motivate a recipient to click on their links; too much obfuscation can lead to lower click rates and reduce spammers' gains [3]. No obfuscation at all, on the other hand, will cause the spam to be easily blocked and few mailboxes will be reached. Therefore, in addition to keeping spam detection rates high, content-based filters caused the positive effect (for the anti-spam side) of making each spam message less attractive and less monetizable { even though spammers have tackled that problem by sending larger volumes of spam.

In this paper, we argue that a fundamental component of spam content has been neglected by content-based spam filters: the content of web pages linked by spam messages. We believe web pages can be an useful component added to current spam filtering frameworks for the following reasons:

1. Web page content is an almost unexplored battleground on the spam arms race, in part for the belief that processing those pages would be too expensive in terms of computing resources. Therefore, current spammers may not have major concerns regarding web pages get-ting identified as spam and so do not implement mechanisms to obfuscate web pages, what would represent an extra cost and might cause their pages to become harder to read. Increasing the cost of the spam activity is one efficient strategy to discourage spammers . In addition to that, although spammers send billion of messages daily, the range of products advertised in web sites are not very diverse; a recent report concluded that 70% of spam advertises pharmaceutical products [9]. A recent work has shown that a few banks are responsible for processing transactions of spam product purchases , what is an additional motivation for seeking evidences for spam detection with are closer to the spammer's business and cannot be changed easily.

2. Recently, Thomas et al. have presented Monarch , a real-time system to detect spam content in web pages Published in social network sites and in e-mail messages. Their results show that with the current technology it is feasible to collect and process pages as they show up in web sites posts and e-mail messages.

3. In the underground spam market, there is evidence that the spam sender is not always the same agent as the web page owner, making the adaptation of web page content a separate task, harder to coordinate with the spam messages [1]. This is the case when spammers work as business brokers that connect sellers (the web page owner) and buyers (e-mail users targeted with spam's) .

4. It has been reported that 92% of spam’s in 2010 contained one or more URLs ; previous work reported the presence of URLs in up to 95% of spam campaigns examined . Those numbers indicate that using web page content to improve spam detection is an applicable strategy, as spammers need to use URLs to earn money through advertised web sites.

In this work, we show that web pages can provide precious information about nature of the e-mail message that link to them. Using information from classical spam/ham repositories, we built a new dataset that provides also the information about the web pages mentioned in messages of the set. Using that dataset, we show that by analyzing the web pages pointed by e-mail messages we can complement Spam Assassin regular expressions and blacklist tests providing an improvement of up to 10% in spam classification, without increasing the false positive rate. Our contributions, in summary, are:

We make available this new dataset, which associates web page content and e-mail message content;

We propose and evaluate a methodology for e-mail spam detection that considers the content of web pages pointed by messages;

We show the e effectiveness of this methodology for the e-mail spam detection task.

Our survey show that considering web pages for spam detection purposes is a promising strategy that creates a new battleground for spam, which has not yet being exploited by current spam filters.

II. RELATED WORK

Very few works mention the use of web pages for e-mail spam detection purposes. To our knowledge, the first work to suggest the use of web pages for e-mail spam detection is a proposal of a framework which combines different spam filtering techniques, including a web page-based detection scheme, but the authors did not go into details about the strategy .

As previously mentioned, the Monarch system is certainly the one that is more closely related to our proposal.

Their work shows the feasibility of collecting and processing web pages in real time for spam analysis with a cost e effective infra-structure, while identifying a large number of attributes that may be used for that goal. Although their goal is mainly to classify the web content itself, the authors also show how information from a Twitter post containing a URL can be used to improve the classification of the web page pointed by it. However, due to the nature of their data they could not explore the relation between e-mail messages and the web pages they link to, since they did not have the original e-mail messages, but only the list of URLs in them. Our datasets make it possible for us to do exactly that.

Obviously, web pages are the basic unity of analysis for the web spam detection task, which aims to detect arti facially created pages into the web in order to influence the results from search engines . In that direction, Webb has created a dataset containing web pages crawled from spam’s from the Spam Archive dataset [13] comprising messages from 2002 and 2006. However, his dataset did not relate web pages with spam messages.

State-of-the art approaches for e-mail spam detection consider message content features and network features [5]. In terms of content-based filtering, a wide range of machine learning techniques such as Bayesian Filters, SVM Classifier and Decision Trees have being applied over e-mail message content with reasonable success [5, 7, 2]. Although web page content has not being experimented as a spam detection strategy, URLs embedded on e-mail messages are used for phishing detection purposes by considering IP ad-dress, WHOIS and domain properties and geolocalization of URLs . The web pages linked by e-mail messages have also been used by Spam scatter [1] as a means of identifying spam campaigns by the web pages they linked to. However, their technique was applied only to spam messages already identify as such and required creating and comparing snapshots of the rendered web pages.

III. ANALYSIS OF METHODOLOGY

For each e-mail message processed from the message stream, we download the web pages linked by the URLs contained in the message. Then, content analysis techniques are applied to the web page content | basically, the same approach adopted by filters to analyze the content of a spam message. We assign a spami city score to the set of web pages linked by a given message and then combine this score with the result of classical spam filtering techniques and the message receives a final score, that takes into account both the message content and the web page content. Figure 1 summarizes the work flow of our filtering technique, and we discuss the major steps next[12].

3.1 Web Page Crawling

We begin by extracting the URLs from the body of the messages[1] and use simple regular expressions to remove non-HTML URLs, i.e., URLs that provide links to images or executable files. After that, we download and store the web pages[2]. In the case of spam messages containing multiple URLs, all the web pages are downloaded and stored. Many of the URLs considered lead to redirections before reaching their final page; in that case, we follow all redirects and store the content of the final URL.

After extracting messages' URLs and downloading the web pages linked by them, we use lynx , a text-based browser, in order to format the web page's text as users would perceive it (without images). Lynx generates a dump of the web page, already in text mode, removing the non-textual part of the message such as HTML tags and JavaScript code. Textual terms in that dump are then extracted, and the final result is a set of words that are used to determine the spamicity of the web page.

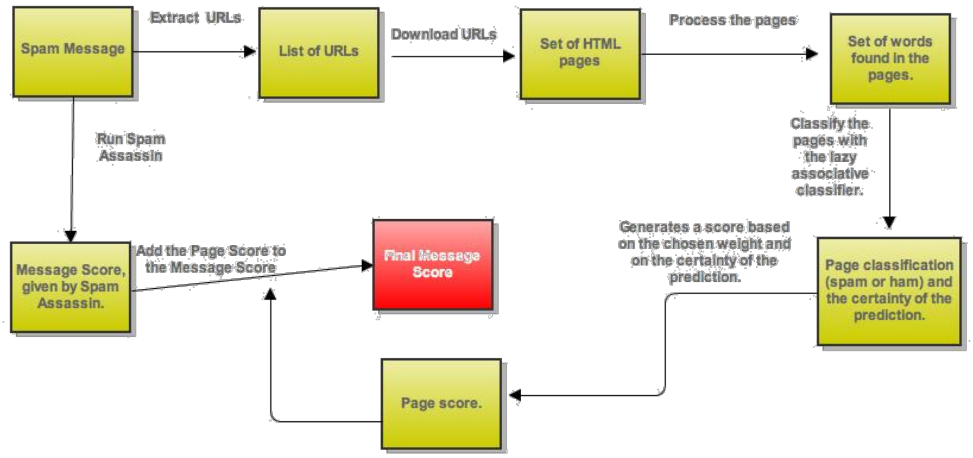

Spam Message
Extract URLs
List of URLs
Download URLs
Set of HTML pages
Process the pages
Set of words found in the pages.
Run Spam Assassin
Classify the pages with the lazy associative classifier.
Generates a score based on the chosen weight and on the certainty of the prediction.
Message Score, given by Spam Assassin.
Add the Page Score to the Message Score
Final Message Score
Page classification (spam or ham) and the certainty of the prediction.
Page score.

Figure 1: Steps of the Web page-based spam filtering approach. Web pages are crawled, URLs are extracted and a spamicity score is assigned to the set of web pages relative to an e-mail message. Then, web page score is combined with conventional spamicity scores to generate a final spamicity assessment.

3.2 Web Page Spam city Score Computation

Given that we have at our disposal textual terms extracted from the set of web pages linked on each spam message, a range of classifiers could be built based on that information. We chose to use LAC, a demand-driven associative classifier . Associative classifiers are systems which integrates association mining with classification, by mining association rules that correlate features with the classes of interest (e.g., spam or ham), and build a classifier which uses relevant association patterns discovered to predict the class of an object . LAC is a demand-driven classifier because it projects/ filters the training data according to the features in the test instance, and extracts rules from this projected data. This ensures that only rules that carry in-formation about the test instance are extracted from the training data, drastically bounding the number of possible rules .

We have chosen to apply a demand-driven lazy classify algorithm for several reasons: (i) it has a good performance for real-time use, (ii) it generates a readable and interpretable model in the form of association rules (which can be easily transformed into a set of regular expressions and incorporated into Spam Assassin), and (iii) it is well calibrated, which means that it provides accurate estimates of class membership properties. Formally, we state that a classifier is well calibrated when the estimated probability p^(cjc) is close to p(cjp^(cjx)), which is the true, empirical probability of x being member of c given that the probability estimated by the classifier is p^(cjx). Therefore, it is possible to know which predictions are more or less accurate and use that in-formation when scoring different pages. For more details on class membership likelihood estimates, please refer to .

The demand-driven associative classifier algorithm generates rules in the form ! c, where is a set of words and c is a class (spam or ham). One typical rule extracted from e-mail messages would be buy; Viagra ! spam. Each one of those rules has a support (how often the rule appears in the training data) and a confidence, which is given by the number of pages in the training data that are classified correctly by the rule divided by the number of pages that contain the set of terms . The final result of each page's classification is a score between 0 and 1 that indicates both the predicted class (spam or ham) and the certainty of the prediction. This score is reliable (as it has been noted that the algorithm is well calibrated), and will be a factor for the final page score.

One of the challenges of spam detection is the asymmetry between the cost associated with classifying a spam message incorrectly and the cost associated with classifying a ham message incorrectly. A false negative might cause slight irritation, as the user sees an undesirable message. A false positive, on the other hand, means that a legitimate message may never reach the user's inbox [10]. Therefore, instead of giving the same importance to false positives and false negatives we build a cost-sensitive classifier [8]. In that scenario, the cost of a class measures the cost of incorrectly classifying a message of that class. As it weighs all the rules obtained for a certain message, the algorithm does a weighted sum of the rules, taking into consideration the confidence of each rule and the cost for each class, in order to give higher importance to rules that point to the class that has the higher cost. That implies that as the cost of the ham class grows, the algorithm needs more certainty in order to classify a page as spam.

3.3 Message Scoring

There are several possible ways to use the resulting score of page classification to classify an e-mail message as spam or ham. Our approach combines the page score S_p with other spamicity scores obtained within the spam message by applying a traditional classifier. The exact formula for S_p will depend on the characteristics of that classifier.

In Spam Assassin, for example, a message is usually considered spam if it reaches 5 or more score points, assigned from a Bays Filter, regular expressions and blacklists [5]. One way of incorporating our technique to Spam Assassin is to simply assign S_p spamicity score points to a message based on the web content of its linked pages, considering whether our classifier says it is spam (Is = 1) or ham (Is = 1), weighting the classifier certainty in that pre-diction (c):

$$S_p = Is \quad W_p \quad c \qquad (1)$$

Note that if the classifier judges the web page to be ham, S_p will be negative and it will contribute to reduce the overall spam city of a message[14]. In this way, web pages that are more \spammy"will result in higher scores for their messages; web pages that look more like \ham" will result in lower (negative) scores. This is the strategy we use in this paper. The choice of W_p will influence how much impact the web page classification will have on the final score; we evaluate that for Spam Assassin in Section 4.

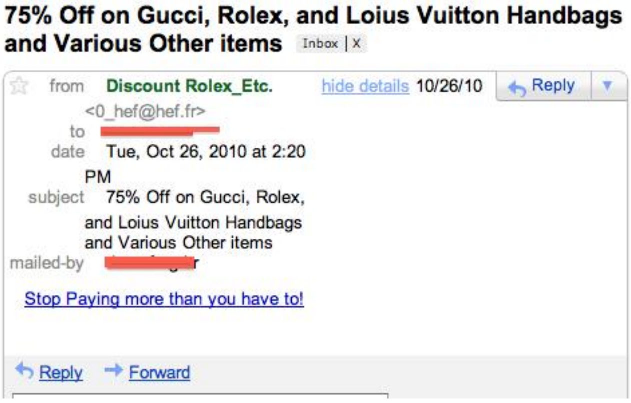
Another alternative would to completely eliminate the use of blacklists, and substitute them for our technique, when appropriate. This could be an interesting approach when the network resources are scarce, since both blacklists and our technique demand that a request be made to another server.

Messages that do not have URLs cannot be filtered by our technique, for obvious reasons. Those messages are filtered by conventional spam filtering methods, and their spamicity assessments are not affected by our strategy.

3.4 Analysis of real time Examples

In this section, we present a step-by-step example of the application of our Web page-content based technique. We picked a spam message identified in October of 2010, from the Spam Archive dataset.

Figure 2 shows the body of the message; it can be noted that the message is very concise, exhibiting very small textual content. Spam Assassin (considering queries to black-lists) yields the rules shown in Table 1.



75% Off on Gucci, Rolex, and Loius Vuitton Handbags and Various Other items

from Discount Rolex_Etc. <0_hef@hef.fr>
to [redacted]
date Tue, Oct 26, 2010 at 2:20 PM
subject 75% Off on Gucci, Rolex, and Loius Vuitton Handbags and Various Other items
mailed-by [redacted]

Stop Paying more than you have to!

Figure 2: Spam Message extracted from Spam Archive. Small textual content poses challenges for analysis of spamicity based on message content.

Table 1: Spam Assassin Rules extracted for Spam Message from Figure 2. Spam Assassin regular expressions and queries to blacklists are not sufficient to classify the message as spam.

Rule	description	score
HTML MESSAGE	HTML included in mes sage	0.001
RCVD_ I _ BRBL- LASTEXT	DNS Blacklist BRBL	1.644
URIBL- BLACK	Contains an URL listed in the URIBL blacklist	1.775

The resulting score considering just spam content is only 0.001. Taking blacklist information into account, the resulting score is 3.4 {still not enough to classify the message as spam. An excerpt from the web page pointed by the URL is shown in Figure 3.

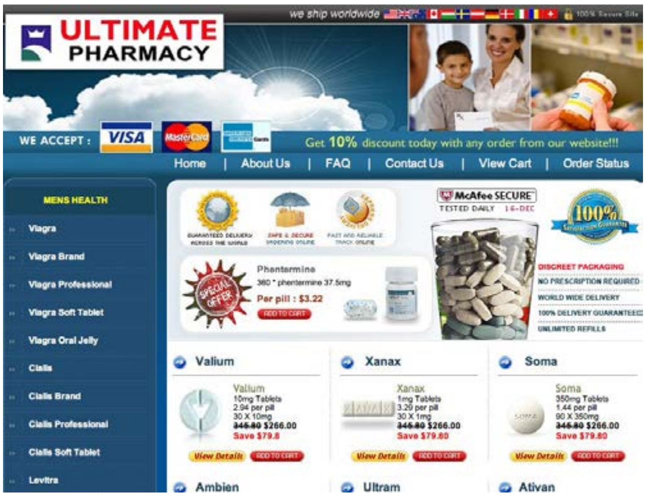


Figure 3: Web page linked by URL present of message from Figure 2. As current filters do not consider web page con-tent, spammers do not need to obfuscate and sacrifice read-ability.

It can be noted that, in this case, the content of the message and the content of the page are totally different { one seems to be selling watches and bags, the other is selling medicine. The content of the page is then extracted into a set of words (with lynx), and is then delivered as input for the already-trained associative classifier (it was trained with other pages from Spam Archive and from ham pages from Spam Assassin's dataset). The associative classifier a list of rules, some of which are listed on Table 2.

After weighting all the rules, the associative classifier yields a result: the page is a spam page, with 90% of certainty (i.e., c = 0.9). If we set W_p = 4.0, then S_p = 3.6. This web page, therefore, would have a 3.6 score. Adding this score to the score obtained by Spam Assassin (Table 1), this message would have a 7.0 score | more than enough to be classified as spam.

Table 2: Rules extracted for Web Page linked by message from Figure 3, unveiling high spamicity terms.

Rule	Support	Confidence
Viagra ! Spam	36.70%	99.84%
levitra ! Spam	34.01%	99.90%
rather ! Ham	2.97%	67.30%

IV. CONCLUSIONS

Web pages linked by spam messages may be a reliable source of evidence of spamicity of e-mail messages. In this work, we propose and evaluate a novel spam detection approach which assigns a spamicity score to web pages linked in e-mail messages. Our approach is suitable to work as a complement to traditional spamicity assessments { such as message content and blacklists.

Our motivation for such analysis is the observation that, so far, web pages linked in e-mail messages are not explored by current spam filters, and, despite that, they offer {currently { clear and unobfuscated content for spamicity assessments. Since spam filters currently do not examine web page content, spammers usually do not obfuscate their advertised sites. Even if spammers begin to obfuscate their web pages, the effort required would serve as an additional disincentive for spam activity.

We believe that this analysis explores a new frontier for spam filtering strategies, introducing a new aspect that has not yet been explored in the literature. In other words, the web pages that are linked by spam messages are a new battle-ground, one that spammers are not currently worried about, and one that may be explored through many different approaches and algorithms.

REFERENCES

[1] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker. Spamscatter: Characterizing Internet Scam Hosting Infrastructure. In Proceedings of the 16th IEEE Security Symposium, pages 135{148, 2007.

[2] I. Androutsopoulos, J. Koutsias, K. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An evaluation of naive bayesian anti-spam filtering. CoRR, cs.CL/0006013, 2000.

[3] B. Biggio, G. Fumera, and F. Roli. Evade hard multiple classifier systems. In O. Okun and G. Valentini, editors, Supervised and Unsupervised Ensemble Methods and Their Applications, volume 245, pages 15{38. Springer Berlin / Heidelberg, 2008.

[4] D. Chinavle, P. Kolari, T. Oates, and T. Finin. Ensembles in adversarial classificationfor spam. In CIKM '09: Proceeding of the 18th ACM conference on Information and knowledge management, pages 2015{2018, New York, NY, USA, 2009. ACM.

[5] G. V. Cormack. Email spam filtering: A systematic review. Found. Trends Inf. Retr., 1:335{455, April 2008.

[6] N. Dalvi, P. Domingos, Mausam, S. Sanghai, and D. Verma. Adversarial classification. In KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 99{108, New York, NY, USA, 2004. ACM.

[7] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. IEEE TRANSACTIONS ON NEURAL NETWORKS, 10(5):1048{1054, 1999.

[8] C. Elkan. The foundations of cost-sensitive learning. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, pages 973{978, 2001.

[9] eSoft. Pharma-fraud continues to dominate spam. www.esoft.com/network-security-threat-blog/ pharma-fraud-continues-to-dominate-spam/, 2010.

[10] T. Fawcett. "in vivo" spam filtering: a challenge problem for kdd. SIGKDD Explor. Newsl., 5:140{148, December 2003.

[11] I. Fette, N. Sadeh, and A. Tomasic. Learning to detect

phishing emails. In Proceedings of the 16th international conference on World Wide Web, WWW '07, pages 649{656, New York, NY, USA, 2007. ACM.

[12] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. Commun. ACM, 50(2):24{33, 2007.

[13] B. Guenter. Spam Archive, 2011. http://untroubled.org/spam/.

[14] P. H. C. Guerra, D. Guedes, J. Wagner Meira, C. Hoepers, M. H. P. C. Chaves, and

K. Steding-Jessen. Exploring the spam arms race to characterize spam evolution. In Proceedings of the 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS), Redmond, WA, 2010.