Towards the Integration of Diverse Spam Filtering Techniques

Calton Pu, *Senior Member, IEEE,* Steve Webb, Oleg Kolesnikov, Wenke Lee, and Richard Lipton

***Abstract*— Text-based spam filters (e.g., keyword and statistical learning filters) use tokens, which are found during message content analysis, to separate spam from legitimate messages. The effectiveness of these token-based filters is due to the presence of token *signatures* (i.e., tokens that are invariant for the many variants of spam messages). Unfortunately, it is relatively easy for spammers to hide or erase these signatures through simple techniques such as misspellings (to confuse keyword filters) and camouflage (i.e., combined spam and legitimate content used to confuse statistical filters). Our hypothesis is that spam contains additional signatures which are more difficult to hide. A concrete example of this type of signature is the presence of URLs in spam messages which are used to induce contact from their victims. We believe diverse spam filtering tools should be developed to incorporate these additional signatures. Thus, in this paper, we discuss a new type of URL-based filtering which can be integrated with existing spam filtering techniques to provide a more robust anti-spam solution. Our approach uses the syntactic constraints of URLs to find them in emails, and then, it uses semantic knowledge and tools (e.g., search engines) to refine and sharpen the spam identification process.**

***Index Terms*— Email spam, Internet Applications, Spam filtering.**

I. Introduction

Spam filtering (i.e., distinguishing between spam and legitimate email messages) is a commonly accepted technique for dealing with spam. Current spam filters classify messages based primarily on the tokens found in those messages' text. However, this approach has had mixed results. On the one hand, many spam messages have token signatures that facilitate filtering. These signatures typically consist of tokens that are invariant for the many variants automatically generated by spammers. On the other hand, spammers can use various techniques to defeat filters. For example, keyword filters can be defeated using deliberate misspellings [1], [2], and statistical learning filters can be confused using camouflage (i.e., legitimate content added to spam messages) [2], [3].

To overcome the limitations of token-based filters, we propose a diversified filtering approach that looks at other forms of spam signatures to complement existing text-based techniques. Examples of these other types of signatures include the presence of URLs and the contents of their corresponding websites, the file types and contents of email attachments, and header information such as the sender's identity and the email's routed path. In this paper, we focus our attention on spam messages that contain URLs and provide a novel approach for filtering these messages. The key observation is that most spam messages contain URLs which are "live" since the spammers would not be able to profit without a functioning link to their site. Thus, by checking the URLs found in a message and verifying a user's interest in the websites referenced by those URLs, we are able to add a new dimension to spam filtering.

This paper has two main contributions. First, we describe three techniques for filtering email messages that contain URLs: URL category whitelists, URL regular expression whitelists, and dynamic classification of websites. Second, we describe a prototype implementation that takes advantage of these three techniques to help enhance spam filtering. Our preliminary results suggest that new dimensions in spam filtering (e.g., using URLs) deserve further exploration. However, due to space limitations, we have omitted our experimental results from this paper.

The remainder of the paper is structured as follows. Section II gives an overview of the related work done in this research area. In Section III, we describe our approach, and Section IV discusses the details of our system's implementation. We provide our conclusions in Section V.

II. Related Work

Filtering is currently the standard approach used to stop email spam. However, most of the current approaches and products use content-based filtering. These content-based approaches include whitelisting, blacklisting, keyword-based [4], statistical classification [5], [6], heuristic-based filtering [7], [8], and collaborative filtering [9]. Other classes of filtering approaches include challenge-response [10], MTA/Gateway filtering (Tarproxy [11], greylisting [12], etc.), and micropayments [13], [14].

Some content-based approaches rely exclusively on message headers: automatic and Bayesian whitelisting [15], blacklisting (MAPS, RBL), and others. These techniques have two main disadvantages. Spammers can easily forge message headers, and legitimate domains can easily become blacklisted.

Other content-based approaches rely on message tokens and their corresponding statistics. For example, simple Bayesian Machine Learning approaches, introduced by Duda et al. [16] and originally applied to spam filtering by Sahami et al. [5], use the conditional probability of tokens occurring in spam and legitimate messages to distinguish between these two types of messages. The evaluation by Androutsopoulos

This work was partially supported by NSF/CISE IIS and CNS divisions through grants CCR-0121643, IDM-0242397 and ITR-0219902, DARPA/IPTO's Self-Regenerative Systems program, and Hewlett-Packard.

Calton Pu, Steve Webb, Oleg Kolesnikov, Wenke Lee, and Richard Lipton are with the Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: {calton, webb, ok, wenke, rjl}@cc.gatech.edu

et al. [6] showed that these approaches are viable but not without shortcomings. The advantages of these approaches is that they are user-specific and offer low false positive and false negative rates after sufficient training with the current generation of spam. Their disadvantages are that this training process is rather time-consuming, and the resulting training statistics cannot be easily re-used or combined for different users. Additionally, as mentioned above, spammers are able to evade these approaches by using common words [1], [2] and camouflage [3] to augment their spam messages.

Exchange Intelligent Filter [7] and SpamAssassin [8] are two examples of content-based approaches that use heuristics to filter spam. Both tools calculate a score for every message based on manually extracted sets of features (rule bases). The disadvantage of these heuristic-based methods is that they are very ad hoc and require regular updates to ensure accuracy. In fact, these regular updates can often be as complex as the filters themselves [17]. The current version of SpamAssassin attempts to deal with this problem by including a number of plug-ins that support different methods (including Bayesian filtering) to improve the score calculation process. One of these plug-ins is related to our approach: SURBL/SpamCopUri [18]. This plug-in blocks messages by using a blacklist of URLs. The blacklist is created based on the spam submissions received from users. One disadvantage of this method is that it takes time for spam messages to be reported. By the time an update is received, it could already be too late (i.e., other users may have already received the spam messages). Additionally, it is very easy for spammers to change the text of a URL and have it point to the same content (e.g., a redirect).

Our approach is different from SURBL in two ways. First, the definition of what constitutes a spam message in our approach is personalized. Each user has a different set of categories which correspond to that particular user's interests. Additionally, this information can be combined and shared easily among users. Second, in addition to the text of a URL, our approach also uses the contents of the website referenced by that URL. For similar reasons, our approach is also different from the URL module used by Brightmail [19].

III. Description

As spammers become more sophisticated, the token signatures (e.g., tokens found in the message body) used by text-based filters to distinguish between spam and legitimate messages will no longer be valid. Thus, we must focus our attention on the characteristics of spam messages that spammers are unable to successfully obscure. A very clear example of such a characteristic is the presence of URLs in spam messages. Spammers rely on these URLs as a feedback mechanism, and as a result, the URLs must be accessible to the messages' recipients. This required accessibility introduces a new technique for filtering spam messages.

In our new approach, we filter email messages based on the URLs they contain. If the URLs in a particular message point to websites that are of interest to a given user, that message is considered legitimate; otherwise, the message is considered spam. We determine a user's interest in a URL (and its corresponding website) using three techniques: URL category whitelists, URL regular expression whitelists, and dynamic classification of websites. In the following sections, each of these techniques is described in more detail.

A. URL Category Whitelists

Many search engines (e.g., Google, Yahoo!, LookSmart, etc.) maintain directories that contain category information for websites. For example, Google's directory [20] categorizes http://www.google.com as Computers/Internet/Searching/Search Engines/Google. Using these directories, we are able to categorize the URLs a user is interested in, compiling a list of acceptable categories $A_{categories}$. Then, we can use $A_{categories}$ to classify incoming messages as either legitimate or spam based on the URLs they contain. When a new message arrives, the URLs found in that message are categorized. If all of the corresponding categories match categories in $A_{categories}$, the message is classified as legitimate. Otherwise, the message is classified as spam. Unfortunately, not all URLs are listed in the search engines' directories. For the remainder of this paper, we will use the term *uncategorized* URLs to refer to URLs that are not listed in any of the directories, and we will use the term *categorized* URLs to refer to URLs that are listed in at least one of the directories. In the next section, we describe an additional technique utilized to help handle uncategorized URLs.

B. URL Regular Expression Whitelists

Using search engine directories to categorize and classify URLs is a novel solution, but it is not always successful. As previously mentioned, uncategorized URLs are not listed in these directories. Additionally, in some cases, users have an interest in websites that cannot be expressed easily with categories. For example, a user might want to register an interest in all websites under a given top-level domain (e.g., *.edu, *.mil, etc.). For these special cases, an additional technique is needed to classify the URLs. One possible technique is the construction of a URL regular expression whitelist which contains a list of acceptable regular expressions A_{regex}. When a new message arrives, the URLs found in that message are compared to the regular expressions in A_{regex}. If all of the URLs match at least one of those regular expressions, the message is classified as legitimate. Otherwise, the system obtains the categories for the URLs that did not match any of the regular expressions and compares those categories to $A_{categories}$ (as explained in the previous section). Unfortunately, this process might still result in uncategorized URLs that do not match any regular expressions in A_{regex}. Thus, in the next section, we explain another technique used to deal with the remaining uncategorized URLs.

C. Website Classification

In addition to creating $A_{categories}$ and A_{regex}, our approach also retrieves the contents of the websites referenced by the URLs used to create those whitelists. These website contents are used to train a learning spam filter (e.g., Naïve Bayes, Support Vector Machines, LogitBoost, etc.) which is used to classify uncategorized URLs that do not match any of the regular expressions in A_{regex}. When a new message arrives containing these uncategorized URLs, the contents of the websites referenced by those URLs are retrieved. Then, the learning spam filter is used to classify each of the websites. If the filter classifies one of those websites as spam, the corresponding message is classified as spam. Otherwise, if all of the websites are classified as legitimate, the corresponding message is classified as legitimate.

IV. Implementation

Our system can be implemented as either server-based or client-based. In the server-based implementation, all users' mail is filtered by a central server, and that server keeps track of each user's profile. In the client-based implementation, each client runs a separate copy of our system. The main advantage of the server-based implementation is the improved performance obtained by maintaining a global cache of URL information (see Section IV-A for a description of this cache). The main advantage of the client-based implementation is the improved privacy protection it provides each user.

For our prototype implementation, we chose a server-based approach. It uses a RedHat Linux v9 machine which runs an Apache web server with `mod_ssl`. Our system consists of two parts: a web-based configuration interface and a mail classifier. The mail classifier is implemented using Perl and procmail, and it also includes a learning spam filter based on POPFile [21]. The system's operation is broken into two phases: training and configuration. These two phases are described in detail in the following sections.

A. Training

Since each user has unique website interests, a separate profile is maintained for every user in the system. Each of these profiles consists of three main parts:

- A list of acceptable categories $A_{categories}$ (as explained above in Section III-A).
- A list of regular expressions A_{regex} for acceptable URLs (as explained above in Section III-B).
- A trained learning spam filter (as explained above in Section III-C).

A user's profile is created during the system's training phase. By default, this profile is generated automatically by the system, but the user also has the option of creating the profile manually. Additionally, our system provides three predefined profiles for academic, business, and home users. These profiles can be used without modification; they can serve as a template for users, or they can be disregarded completely. A partial example of the Academic profile is given in the Appendix.

The automatic profile generation process occurs as follows. First, training URLs are extracted from the user's existing legitimate email messages. Additional URLs can also be obtained from the user's Bookmarks/Favorites list which is maintained by the user's favorite web browser. Once the training URLs are obtained, they are categorized using the directories of multiple search engines (e.g., Google, Yahoo!, LookSmart, etc.), and the corresponding categories are stored in $A_{categories}$.

Since it may take several seconds to query the search engines for each URL, our system also maintains a system-wide cache for query results to improve performance. This cache contains category and other information retrieved from search engines, and its entries are periodically expired to ensure the information remains current. Thus, when a user needs to query the search engines, the cache is consulted first. If the necessary information is not found there, the query is forwarded to the search engines.

After the categories are stored in $A_{categories}$ and the system-wide cache, regular expressions are created to match each of the user's unique, uncategorized training URLs. These regular expressions are then stored in A_{regex}. Next, the system retrieves the contents of the websites referenced by the URLs that were used to create $A_{categories}$ and A_{regex}. The system also retrieves the contents of the websites referenced by the URLs present in the user's existing spam messages. Once the system has these websites' contents, those contents are used to train the system's learning spam filter.

An immediate problem that arises when obtaining a website's contents is redirection. Spammers can easily use multiple redirects to hide their real website. Thus, any attempt to obtain website content must handle redirects correctly. In our system, we resolve this issue by relying on the cached copies of websites which are stored by search engines. When the search engines index websites, their crawlers automatically follow the redirects. Thus, the cached copies stored by the search engines are the end-sites rather than the redirecting pages. Our system's learning spam filter uses these cached copies during its training and classification phases.

After the profile generation process is complete, users may edit and verify their $A_{categories}$ and A_{regex} at any time to ensure their interests are properly reflected. They are also able to provide the learning spam filter with additional training data or correct any misclassifications made by the filter.

B. Classification

Once the training phase is complete, the system uses the user's profile to classify incoming email messages. This classification process works as follows. When a new message arrives, it is scanned for URLs. If the message does not contain any URLs, it is passed to another filtering subsystem which is beyond the scope of this paper. Otherwise, the system checks every URL in the message according to the following process.

First, each URL is compared to the regular expressions in A_{regex}. If all of the URLs match at least one of those regular expressions, the message is classified as legitimate. Otherwise, the category information is obtained for the URLs that did not match any of the regular expressions. To improve performance and reduce the load placed on the search engines, the system initially consults the system-wide cache for each URL's category. If the cache does not contain the necessary information, the search engines are queried, and the results are placed in the cache. Once the system has the categories for the URLs, those categories are compared to the categories in $A_{categories}$. For every URL with a category found in $A_{categories}$, a new regular expression is created and added to A_{regex}. The purpose of this new regular expression is to optimize the system's performance when this URL is encountered again, and we refer to this optimization process as *incremental learning*. If all of the URLs have categories found in $A_{categories}$, the message is classified as legitimate. Otherwise, if one of the URLs has a category not found in $A_{categories}$, the message is classified as spam. However, if some of the URLs are uncategorized, the learning spam filter is used to classify the contents of the websites referenced by those URLs. If all of those websites are classified as legitimate, the corresponding message is classified as legitimate. Otherwise, if one of the websites is classified as spam, the message is classified as spam.

V. Conclusions

In this paper, we discussed the need for a diversified approach to spam filtering. Concretely, we presented a new method of filtering spam which focuses on the presence and significance of URLs as a spam signature. URLs are very reliable indicators of spam since they need to be "live" in order for spammers to profit from potential contact with their victims. First, we parse the messages and identify the URLs they contain. Then, we use URL category information already maintained by search engines to check the validity and content classification of those URLs. Based on this information, we are able to determine if the messages are spam.

Our new filtering method complements the current generation of token-based spam filters which are vulnerable to spammers' manipulation of spam message content. It is also a good example of a diverse and independent technique that can be integrated with other techniques to create a spam filtering system which is more robust and effective than each of the comprising techniques.

References

[1] G. L. Wittel and S. F. Wu, "On attacking statistical spam filters," in *Proc. 1st Conference on Email and Anti-Spam (CEAS 2004)*, 2004.

[2] D. Lowd and C. Meek, "Good word attacks on statistical spam filters," in *Proc. 2nd Conference on Email and Anti-Spam (CEAS 2005)*, 2005.

[3] S. Webb, S. Chitti, and C. Pu, "An experimental evaluation of spam filter performance and robustness against attack," 2005, proc. 1st Int'l Conf. on Collaborative Computing (CollaborateCom 2005).

[4] W. W. Cohen, "Learning rules that classify e-mail," in *Proc. 1996 AAAI Spring Symposium on Machine Learning and Information Access*, Palo Alto, CA, 1996.

[5] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A bayesian approach to filtering junk e-mail," in *Proc. AAAI Workshop on Learning for Text Categorization*, Madison, Wisconsin, July 1998, pp. 55–62.

[6] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam filtering with personal e-mail messages," in *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, Athens, Greece, July 2000, pp. 160–167.

[7] M. Corporation. (2003) Exchange intelligent message filter. [Online]. Available: http://www.microsoft.com/exchange/downloads/2003/imf/default.mspx

[8] S. D. Team. (2004) The apache spamassassin project. [Online]. Available: http://spamassassin.apache.org/

[9] V. V. Prakash. (2004) Vipul's razor. [Online]. Available: http://razor.sourceforge.net/

[10] M. Iwanaga, T. Tabata, and K. Sakurai, "Evaluation of anti-spam methods combining bayesian filtering and strong challenge and response," in *Proc. IASTED International Conference on Communication, Network, and Information Security (CNIS 2003)*, 2003, pp. 214–219.

[11] M. Lamb, "Tarproxy: Lessons learned and what's ahead," MIT Spam Conference, 2004.

[12] E. Harris. (2003) The next step in the spam control war: Greylisting. [Online]. Available: http://projects.puremagic.com/greylisting/

[13] R. E. Kraut, S. Sunder, J. Morris, R. Telang, D. Filer, and M. Cronin, "Markets for attention: Will postage for email help?" in *Proc. 2002 ACM conference on Computer supported Cooperative Work*, New Orleans, LA, 2002, pp. 206–215.

[14] B. Templeton. (2003) E-stamps. [Online]. Available: http://www.templetons.com/brad/spam/estamps.html

[15] E. Kidd. (2002) Bayesian whitelisting: Finding the good mail among the spam. [Online]. Available: http://www.randomhacks.net/stories/bayesian-whitelisting.html

[16] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, New York: Wiley, 1973, ch. Bayes Decision Theory, pp. 10–43.

[17] J. Goodman, "Spam filtering: From the lab to the real world," MIT Spam Conference, 2003.

[18] J. Chan. (2004) Surbl - spam uri realtime blocklists. [Online]. Available: http://www.surbl.org/

[19] K. Schneider, "Brightmail url filtering," MIT Spam Conference, 2004.

[20] Google. (2005) Google directory. [Online]. Available: http://dir.google.com/

[21] J. Graham-Cumming. (2004) Popfile - automatic email classification. [Online]. Available: http://popfile.sourceforge.net/

Appendix

Academic Profile: Category Whitelist

```
Science/Conferences
Science/News
Science/Publications
Computers/Computer_Science/Organizations
Computers/Computer_Science/Research_Institutes
Computers/Computer_Science/Academic_Departments/North_America/United_States/Georgia
Science/Technology/Academia
News/Colleges_and_Universities/
Computers/Internet/Policy
Computers/Internet/Searching/Search_Engines/Google
Computers/Supercomputing
Computers/Systems/
News/By_Subject/Information_Technology
News/By_Subject/Information_Technology/Computers
News/By_Subject/Information_Technology/Internet/Headlines_and_Snippets
...
```

Academic Profile: Regular Expression Whitelist

```
*.edu
*.gov
*.org
*.mil
*.yahoo.com
*.google.com
www.cnn.com
www.nytimes.com
portal.acm.org
ieeexplore.ieee.org
citeseer.ist.psu.edu
www.research.ibm.com
www.research.att.com
www.research.microsoft.com
...
```