Hoodwinking Spam Email Filters

WANLI MA, DAT TRAN, DHARMENDRA SHARMA, SEN LI
School of Information Sciences and Engineering
University of Canberra
ACT 2601
AUSTRALIA

Abstract: - Many spam email filters have been proposed, however spammers regularly find new ways of hoodwinking those filters. Most of those filters are text based and hence spammers try to conceal the text which reveals the spam nature of an email. In order to investigate the ways spammers are using, we consider a large set of spam emails and found that we can classify these emails into 5 categories which are text based, obfuscating, image based, HTML tags, and non-English. We counted the percentage of spam emails in each category and then used a sample spam filter to evaluate the effectiveness of the filter on each of the categories. The TREC Spam Filter Evaluation Toolkit was used in our evaluation.

Key-Words: - Spam email, Text based spam email filter, TREC Spam Filter Evaluation Toolkit.

1 Introduction

Spam emails are one type of the cyber nuisances we have to put up with everyday. The industry and the research community have been investing significant effort in fighting spam emails. There are many spam email filters in operation, ranging from commercial products to open source software. Cormack and Lynam coordinated a comprehensive evaluation on 44 spam email filters, together with 8 open source filters in 2005 [1]. Their conclusion is that “The results presented here indicated that content-based spam filter can be quite effective, but not a panacea”.

On the other hand, there are also many proposed filtering technologies from the research communities. Many papers have been published, for example, Naïve Bayes classifier [2], instance based learning – memory based approach [3], boosted decision tree [4], Maximum Entropy [5], Support Vector Machines [6], LVQ-based neural network [7], and practical entropy coding theory [8]. The results in those publications give us very encouraging pictures.

Yet in our daily life, all of us have been continuously suffering from the frustration of spam emails. Therefore, a logic question is where the problem is. On the one hand, the spam filters have pretty high recognition rates in the evaluations, and most of the results can be repeated. On the other hand, the real life experience does not match the evaluations.

Some may claim that the problem is due to the lack of diligent training to the spam filters. We dispute the claim. Spam is a universal problem, and the training results can be easily shared on the Internet. If the training were the problem, we should not see so many spam emails. A parallel observation can be made by the operation of virus scanning software, where virus signature data can be updated reasonably effectively. With virus scanning software properly installed and also properly configured for updating, one can almost be assured of being free of virus attacks.

The real reason is actually due to the swift adoption of new techniques by the spammers. Their invention to circumvent spam filters outpaces the industry and the research community. Almost all of the spam filters and research proposals are text based, and the evaluations and the research results are also on text based spam. Although the spam corpus used for some evaluations does contain images, HTML tags, and some attachments, the text part of the emails always has some degree of the indication of its spam nature. In the real world, spammers try everything they can to conceal the text which reveals the spam nature of an email. There are several popular ways of hoodwinking the spam filters. The text part of a spam email may not have a trace of its spam nature. Graham-Cumming maintains a comprehensive list of the techniques the spammers use to circumvent spam email filters [9]. Some examples are:

- Using deliberately misspelled words (obfuscating): for example, spell Viagra, a very popular spamming topic, as “v1agra”, “V!@gra” or “VlhAGRA”. The obfuscations are still humanly readable, but they pose serious challenges to a computer program to catch over 6×1020 ways of obfuscations [10].

- Concealing text in images as email attachments: Aradhye et al [15] estimated that 25% of spam emails contain images. C.-T. Wu et al's count is 38% [11]. One of the authors counted spam emails received from his working email address. Among the 256 spam emails received within 15 days, 91 or 36% of emails were image based. Text based email filters are helpless in dealing with image based spamming. Given the fact that image based spam can successfully circumvent spam filters, the situation can only get worse in the future.
- Obscuring keywords by HTML tags: instead of spelling "Viagra" as it is, individual character is wrapped by HTML tags, e.g., `<b>V</b><u></u><b>i</b><span>a<b>g</b>r<i>a</i>`.

The combination of these techniques makes it even harder for a spam filter to correctly judge the nature of an incoming email.

In this paper, we classify spam emails into 5 categories. They are text based, obfuscating, image based, HTML tags, and non-English. We first count the percentage of spam emails in each category and then evaluate the effectiveness of a sample spam filter against spam the emails in each of the categories. We use TREC Spam Filter Evaluation Toolkit developed by Cormack and Lynam [12]. Hopefully, our results will be closer to the real work experience.

The rest of the paper is organized as follows. Section 2 explains the techniques used to circumvent spam filters. Section 3 explains our test environment and the test goals, and Section 4 presents the test results. In Section 5, we conclude the paper with future work.

2 Circumventing Text Based Spam Email Filters

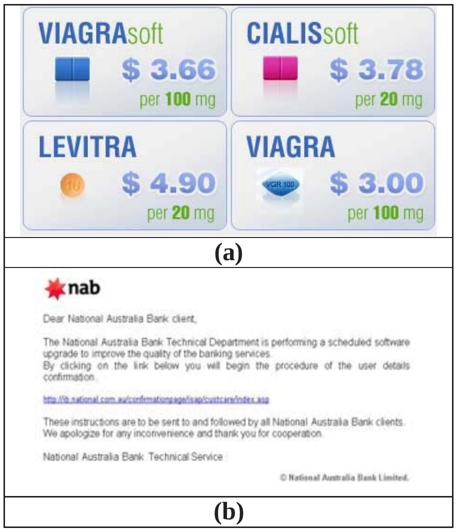
At the very beginning, emails were in ASCII text format only [13]. To be able to convey rich presentation styles, they were extended with multimedia abilities [14]. Image based spam emails take the advantage of using the MIME multipart/alternative directive, which is designed to accommodate multiple displays of an email, such as plaintext format and HTML format. The directive suggests that the enclosed parts are the same in semantics, but with different presentation styles. Only one of them will be chosen to display and a mailer "*must place the body parts in increasing order of preference, that is with the preferred format last*" [14].

Figure 1 is an example of a spam email. The email has three alternative parts: part one is a plain text paragraph cut from a book, part two is a HTML formatted paragraph cut from a book as well, and part three is a JPEG formatted picture as in Figure 2 (a).

```
From: spammer <faked_email address>
To: recepent_email_address
Content-type: multipart/alternative;

--Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
##### Part 1: plain text format #####
Langdon looked again at the fax an ancient myth confirmed in black and white.

--Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
##### Part 2: HTML format  #####
<textarea style="visibility: hidden;">Stan Planton for being my</textarea>

--Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)
##### Part 3: picture format. It has nothing to do with Part 1 or 2 #####
Content-type: image/jpeg; name=image001.jpg
Content-disposition: attachment; filename=image001.jpg

/9j/4AAQSkZJRgABAgAAZABkAAD/7AARRHVja3kAAQAEAAAAHgAA/
+4ADkFkb2JlAGTAAAAAAf/bXFxcXHx4XGhoaGhceHiMlJyUjHi8vMzML
...
--Boundary_(ID_fkG49yFmM6kAJ0sBSY0dzg)--
```

Fig 1. An imaged based spam email sample

A mailer believes that these three parts are semantically identical and will only display one part, Figure 2 (a) in this case. But in this email, the first two parts have nothing to do with the third part. They are purposely included in the email to deceive text based spam filters. Another similar example can be found in Figure 2 (b).



(a)

Dear National Australia Bank client,

The National Australia Bank Technical Department is performing a scheduled software upgrade to improve the quality of the banking services.
By clicking on the link below you will begin the procedure of the user details confirmation.

http://ib.national.com.au/confirmationpage/isap/custcare/index.asp

These instructions are to be sent to and followed by all National Australia Bank clients.
We apologize for any inconvenience and thank you for cooperation.

National Australia Bank Technical Service

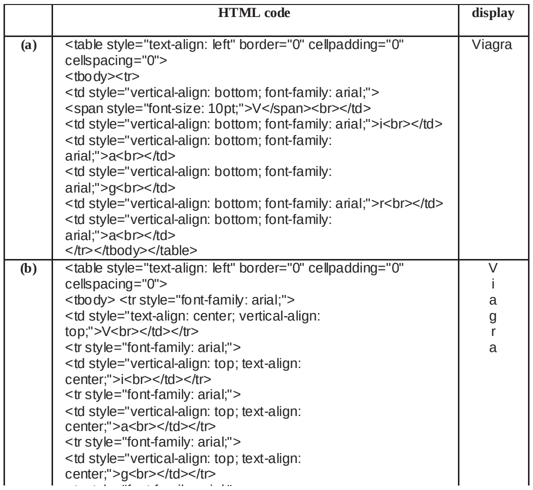
© National Australia Bank Limited.

(b)

Fig 2. Images in spam emails

HTML tags can be used to efficiently obscure the keywords of a spam email. The example given in the Introduction section (`<b>V</b><u></u><b>I </b><span>a<b>g</b>r<i>a</i>`) can be easily dealt with – removing all the HTML tags reveals the underlying keyword. However, it is not so easy to untangle a well crafted HTML tag obscuration.

Using an invisible HTML table is one of the examples. The keyword of a spam email, say Viagra, is dissolved into the cells of the table, one character each cell. The visual appearance is still the same, but the HTML text does not have the keyword as a single word any more, Figure 3 (a).



	HTML code	display
(a)	<table style="text-align: left" border="0" cellpadding="0" cellspacing="0"> <tbody><tr> <td style="vertical-align: bottom; font-family: arial;"> <span style="font-size: 10pt;">V</span> </td> <td style="vertical-align: bottom; font-family: arial;">i </td> <td style="vertical-align: bottom; font-family: arial;">a </td> <td style="vertical-align: bottom; font-family: arial;">g </td> <td style="vertical-align: bottom; font-family: arial;">r </td> <td style="vertical-align: bottom; font-family: arial;">a </td> </tr></tbody></table>	Viagra
(b)	<table style="text-align: left" border="0" cellpadding="0" cellspacing="0"> <tbody> <tr style="font-family: arial;"> <td style="text-align: center; vertical-align: top;">V </td></tr> <tr style="font-family: arial;"> <td style="vertical-align: top; text-align: center;">i </td></tr> <tr style="font-family: arial;"> <td style="vertical-align: top; text-align: center;">a </td></tr> <tr style="font-family: arial;"> <td style="vertical-align: top; text-align: center;">g </td></tr> <tr style="font-family: arial;">	V i a g r a

Fig 3. HTML code and visual display

It is not a trivial task to merge the contents of different table cells together, let alone using different alignments of the keyword, e.g., vertical spelling in Figure 2 (b). Using non-uniform table cell structure can further complicate the situation.

Deliberated misspelling (obfuscating) is also hard to detect. It is not so hard to detect misspells of Viagra as V1agra, \/iagra (V as \ and /) and Vi@gra etc. However, it is not so easy for a program to determine that VIhAGRA is actually Viagra. Given so many ways of obfuscating a keyword, e.g., 6×1020 ways of obfuscating Viagra as listed in [10], it is not an easy task for a spam filter to recognize all possible obfuscations, yet makes no mistakes on other words of the email.

3 The Evaluation Environment

We used TREC Spam Filter Evaluation Toolkit (spamfilterjig-full-0.2) [12] to evaluate the effectiveness of a spam filter against the emails in each of the categories. For our purpose, only ham misclassification percentage (hm%) and spam misclassification percentage (sm%) [1] were used. The spam filter we used was the default sample filter coming with the toolkit, bogofilter [15].

The spam corpus was from SpamArchive [16]. There were 4629 spam emails in these files. Among them, there were 4590 emails for text based, obfuscating, image based and HTML tags, and 39 miscellaneous emails. We used the ham emails coming with the toolkit, which were originated from SpamAssassin corpus.

Spam emails from SpamArchive	**Ham emails from the evaluation toolkit**	
text based	easy_ham	2055
image based	easy_ham_2	1223
obfuscating	Hard_ham	613
HTML tags		67
non English language		632
miscellaneous		39
Total		**4629**

Table 1. Spam email categories

For each category, we generated the index file by randomly mixing its spam emails with the same amount of ham emails, up to the total number of all ham emails. We first reset the filter directory (example-filter) and then run the filter against the mixed emails as instructed by the toolkit.

The experiment was conducted on a Fodera Core 5 Linux operating system platform, with 2.4 GHz Pentium 4 CPU and 512 MB memory

4 Experimental Results

The results for filtering emails in different categories are listed in Table 2

Category	hm%	sm%
text based	0.00%	9.32%
image based	0.17%	22.63%
obfuscating	0.00%	1.47%
HTML tags	6.00%	7.46%
non English language	0.00%	10.99%
Total	**0.13%**	**11.92%**

Table 2: Ham and spam misclassification percentage

5 Conclusion and Future Work

Text based spam email filters are effective in dealing with text based spam emails; however, they are helpless in dealing with obscured spam emails, such as imaged based spam emails, using HTML tags to conceal spam email keywords, and spam email keyword obfuscating etc. This is the fundamental reason responsible for many successful penetrations of spam email filters. Given the fact that text based spam email filters can be easily circumvented by concealing the text, spammers keep finding and adapting more creative ways to hoodwink the filters.

By text based spam filters alone, we are far away from deterring spam emails. However, on the other hand, in essence, the purpose of spam emails is to deliver some kind of information. Ultimately, text as least, humanly readable text has to be displayed on the screen. This is, in our opinion, the key to fight spam. As text based spam email filters can be effective in dealing with text, and the ultimate goal of spam emails is to deliver text to the screen for humans to read, effective spam filtering will rely on the building of tools which can convert the obscured formats of spam text into clear text. Based on the clear text, a text based spam email filter can easily recognize the nature of an email, being a spam or not.

We called these tools as normalizers, and the process of converting obscured formats into clear text normalization. The evaluation described in this paper is part of our proposal of developing specialized normalizers as the preprocessors for text based spam email filters. The results reinforce our beliefs of the need for multiple normalizers. So far, we have been concentrating on imaged based spam email normalizers and obfuscating normalizers [17-21]. The preliminary experiment is very encouraging, and a large scale evaluation of the effectiveness of the normalizers is on the way. We anticipate better results in the near future.

Acknowledgments

This research work is supported by the Divisional grants from the Division of Business, Law and Information Sciences, University of Canberra, Australia, and the University grants from University of Canberra, Australia.

References:

[1] Cormack, G. and T. Lynam. TREC 2005 Spam Track Overview, in the Fourteenth TExt Retrieval Conference (TREC 2005). 2005. Gaithersburg, MD, USA.

[2] Sahami, M., S. Dumais, et al. A Bayesian Approach to Filtering Junk E-mail, in AAAI-98 Workshop on Learning for Text Categorization. 1998.

[3] Sakkis, G., I. Androutsopoulos, et al., A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists. INFORMATION RETRIEVAL, 2003. 6(1): p. 49-73.

[4] Carreras, X. and L. Marquez. Boosting Trees for Anti-Spam Email Filtering, in 4th International Conference on Recent Advances in Natural Language Processing (RANLP-2001). 2001.

[5] ZHANG, L. and T.-s. YAO. Filtering Junk Mail with A Maximum Entropy Model. in 20th International Conference on Computer Processing of Oriental Languages (ICCPOL03). 2003.

[6] Drucker, H., D. Wu, and V.N. Vapnik, Support vector machines for spam categorization. IEEE Transactions on Neural Networks, 1999. 10(5): p. 1048-1054.

[7] Chuan, Z., L. Xianliang, et al., A LVQ-based neural network anti-spam email approach. ACM SIGOPS Operating Systems Review, 2005. 39(1): p. 34 - 39.

[8] Zhou, Y., M.S. Mulekar, and P. Nerellapalli. Adaptive Spam Filtering Using Dynamic Feature Space, in the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05). 2005.

[9] Graham-Cumming, J. The Spammers' Compendium. 2006 15 May 2006 [cited 2006 May]; Available from: http://www.jgc.org/tsc/.

[10] Cockeyed. There are 600,426,974,379,824,381,952 ways to spell Viagra. 2006 [cited 2006 October 2006]; Available from: http://cockeyed.com/lessons/viagra/viagra.html.

[11] Wu, C.-T., K.-T. Cheng, et al. Using visual features for anti-spam filtering, in IEEE International Conference on Image Processing, 2005 (ICIP 2005). 2005.

[12] Cormack, G. and T. Lynam. SPAM Track Guidelines - TREC 2005 and 2006. [cited 2006 July 2006]; Available from: http://plg.uwaterloo.ca/~gvcormac/spam/.

[13] Postel, J.B. Simple Mail Transfer Protocol. 1982 [cited 2006 May]; Available from: http://www.ietf.org/rfc/rfc0821.txt.

[14] Freed, N. and N. Borenstein. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. 1996 [cited 2006 May]; Available from: http://www.ietf.org/rfc/rfc2046.txt.

[15] Raymond, E.S., D. Relson, et al. Bogofilter. [cited 2006 July 2006]; Available from: http://bogofilter.sourceforge.net/.

[16] SpamArchive. Spam Archive. [cited 2006 June 2006]; Available from: http://www.spamarchive.org/.

[17] Ma, W., D. Tran, et al. Detecting Spam Email by Extracting Keywords from Image Attachments, to be published in VIP2006. 2006.

[18] Tran, D., W. Ma, and D. Sharma. Fuzzy Normalization for Spam Email Detection, in Proceedings of SCIS & ISIS 2006 Conf., pp. 1505-1509.

[19] Tran, D., W. Ma, and D. Sharma. A Noise Tolerant Spam Email Detection Engine, to be published in WITSP'06. 2006.

[20] Ma, W., D. Tran, and D. Sharma. Detecting Image Based Spam Email by Using OCR and Trigram Method, to be published in SIT2006. 2006.

[21] Tran, D., W. Ma, et al. A Proposed Statistical Model for Spam Email Detection, in Proceedings of the First International Conference on Theories and Applications of Computer Science (ICTAC 2006). 2006.