*Int. J. Communications, Network and System Sciences*, **2012, 5, 416-422**
http://dx.doi.org/10.4236/ijcns.2012.57051 Published Online July 2012 (http://www.SciRP.org/journal/ijcns)

# A Solution for Fighting Spammer's Resources and Minimizing the Impact of Spam*

**Samir A. Elsagheer Mohamed**[1,2]
[1]Electrical Engineering Department, Faculty of Engineering, South Valley University, Aswan, Egypt
[2]College of Computer, Qassim University, Qassim, KSA
Email: samhmd@qu.edu.sa, samirahmed@yahoo.com

Received May 7, 2012; revised May 28, 2012; accepted June 10, 2012

## ABSTRACT

Spam or unsolicited emails constitute a major threat to the Internet, the corporations, and the end-users. Statistics show that about 70% - 80% of the emails are spam. There are several techniques that have been implemented to react to the spam on its arrival. These techniques consist in filtering the emails and placing them in the Junk or Spam folders of the users. Regardless of the accuracy of these techniques, they are all passive. In other words, they are like someone is hitting you and you are trying by all the means to protect yourself from these hits without fighting your opponent. As we know the proverbs "The best defense is a good offense" or "Attack is the best form of defense". Thus, we believe that attacking the spammers is the best way to minimize their impact. Spammers send millions of emails to the users for several reasons and usually they include some links or images that direct the user to some web pages or simply to track the users. The proposed idea of attacking the spammers is by building some software to collect these links from the Spam and Junk folders of the users. Then, the software periodically and actively visit these links and the subsequent redirect links as if a user clicks on these links or as if the user open the email containing the tracking link. If this software is used by millions of users (included in the major email providers), then this will act as a storm of Distributed Denial of Service attack on the spammers servers and there bandwidth will be completely consumed by this act. In this case, no human can visit their sites because they will be unavailable. In this paper, we describe this approach and show its effectiveness. In addition, we present an application we have developed that can be used for this reason.

**Keywords:** Spam Emails; Attacking Spammers; Spam Filtering; Distributed Denial of Service Attacks; Software Development

## 1. Introduction

Email plays a major role in our life. However, it is the main target of the spammers for the commercialization and marketing using the Internet. Statistics show that unsolicited bulk email or spam constitute about 70% - 80% of the total volume of emails. Spam is considered the major threat on the Internet because it affects several components on the Internet. The end users are affected by the annoyance, the time wasting and the affection by the malware spread using the spam. Spam frustrates users by overloading their email boxes with large number of useless and unwanted messages. Scams can cause unwary users to reveal personal information such as credit card numbers or passwords, hence suffering monetary damages as well as losing time and privacy. Corporations looses huge amount of money due to the spam. Service providers loose much money in implementing spam filters to reduce the impact of the spam on the end-users. In addition, major part of their network bandwidth is used for transporting the spam emails. They are also obliged to increase the storage capacity of the email servers to store the received spams. Spam overloads e-mail servers, delaying or preventing the delivery of legitimate e-mail messages. The Internet core is also affected by the spam. Routers and link bandwidths are overloaded transferring the huge volume of spam email.

There exists a wide range of counter measures to deal with the spam problem. Usually, the most common techniques to deal with the spam problem is the spam filtering techniques. Traditional anti-spam techniques include the Bayesian-based filters [1-4], Rule-based Scoring Systems [5-7], DNS MX Record Lookup and Reverse lookup systems [8], DNS Realtime Blackhole List (DNSRBLs) or IP Blacklists [9].

All these techniques are not accurate as they suffer from the false positive and false negative classification errors. These techniques are all objective. However, the most accurate way to identify the spam email is by using

*This work is supported by a grant from King Abdulaziz City for Science and Technology (KACST) via the National Science and Technology Plan, Al-Imam Muhammad bin Saud Islamic University, Saudi Arabia. Project number: 08-INF438-8.

 

a panel of human (which seems to be infeasible solution). Thus, combining the objective techniques and an intelligent subjective technique can give better results [10].

Another issue with all these techniques is that they are passive. That means that they take action on the arrival of the spam email. Once the spammers sends a spam campaigns containing thousands of destination email addresses, the email servers and other related servers consume much of their resources to distinguish them from legitimated email messages. In other words, they are like someone is hitting you and you are trying by all the means to protect yourself from these hits without fighting your opponent. As we know the proverbs "The best defense is a good offense" or "Attack is the best form of defense". Thus, we believe that attacking the spammers is the best way to minimize their impact and force them to reduce the volume of the spam they are sending.

In this paper, we present a very good approach to fight the spammers. This is to return the hit of the spammers back on their faces. Instead of being traditional and trying to survive with their attacks (most of the existing spam solutions are passive defenders), why not try the opposite strategy: being attackers. Usually they have limited web server resources. So when we are sure that an email is spam, we collect all the links on it. We build simple applications that visit these links (using HTTP request message). Usually, these links are not direct one, but they redirect you to the site the spammers want you to visit. So, this application should fetch the and download all the content of the original link and the redirected URL and build a database of these links. The application periodically fetch all the spam emails from the spam folders and collect all the links including those of the images. Then, it periodically fetch one of these links and download all the contents of the target web page (promoted by the spammer). By distributing this applications to many Internet users (even the home users), we can have a very hard Distributed Denial of Service (DDoS) Attack to attacking the spammers' servers. In this case, the spammer servers will be very busy answering the applications and their resources (the bandwidth) will be wasted responding to the requests coming from these application. For any casual user who does not use this application, she/he will be protected also (as some of the sent link is trapping links). This is because the spammers' servers will not be able to answer her/his request. In addition, the spammers will not be able to know if a real user is visiting the link or the application. Sure the application must be very clever to not leave any fingerprint that can allow them to identify it.

The rest of this paper is organized as follows: In Section 2 the related works and research efforts are given. The descriptions of the proposed technique and the challenges and analysis of the proposed technique are given in Section 3. We present the application that we have developed in Section 4. Finally, the Conclusions and the future works are given in Section 5.

## 2. Related Works

The most famous approach based on the statistical filtering is the Bayesian filter [2-4], which is based on the Bayes' theorem. This theorem states that the probability that an email is spam, given that it has certain words in it, is equal to the probability of finding those certain words in spam email, times the probability that any email is spam, divided by the probability of finding those words in any email. The drawback of the statistical filtering techniques is the processing time: the time required to process an email and to end up if it is a spam or not. Another problem with this kind of filtering is that it cannot fight against the new tricks of the spammers, like changing vocabulary, introducing the most recognizable terms or adding a relatively high number of random words, miss spell words, adding numbers and symbols in the middle of the word or the phrase, etc.

In [10], a new technique based on the combination of the fast traditional objective anti-spam filtering as well as a smart cooperative subjective spam filtering method is presented to reduce the drawbacks mentioned before.

Another direction to fight the spam problem is the "*machine learning*" techniques. Some of the existing machine learning based techniques are: rule learning, decision trees [11], support vector machines [12,13] or combinations of different learners [14]. The basic and common concept of these approaches is that using a classifier to filter out spam and the classifier is learned from training data rather than constructed by hand. In [15], ant colony optimization (ACO) algorithm is proposed to detect spam in host level. From the machine learning viewpoint, spam filtering based on the textual content of e-mail can be viewed as a special case of text categorization, with the categories being spam or non-spam [5,6,16].

Spam can also be sent embedded in an image [17,18] or in a PDF documents (*i.e*. the text of the message is converted to an image and sent by email to users). Many research efforts concentrate to image spam detection as the ones given in [19,20].

## 3. Actively Attacking the Spammers

In order to defeat your opponents, you have to understand their strategies and the way they behave. Thus, for implementing a good methodology to fight the spammers, we have to understand the way they usually send spam to us. Fortunately, their tricks are known so far. In this Section, we will briefly describe the used spam sending techniques. Then we will present the methodology to attack them.

### 3.1. Spam Sending Techniques

The spam sending techniques evolved with time from a

 

very simple email sending strategy to a very sophisticated and hard-to-detect-objectively strategy. In the first used technique, spammers send thousands or millions of e-mail messages from their own e-mail accounts. Detecting this type of spam is relatively easy by checking the number of the identical messages, the subject line and analysis of the emails. As a counter measure to this technique, the service providers implement the blacklists to ban these users. The spammers did not stop here, they invented the next strategy. They used open mail proxies, which are servers that accept connections from any network address, acting as a blind intermediary to virtually any other network address. The received message seems to be originated from these mail proxies. Hence the spammer identity is completely hidden. The counter measure for this technique is similar to the first one, using blacklists to ban the emails coming from these open proxies. The next generation spam sending strategy is the spam zombie [21-23]. Spam zombie consists of infecting unprotected computers with a Trojan horse program that can be controlled remotely by the spammers. The Trojan horse program uses the SMTP of the victim computer to send massive emails. When the spammer wants to send spam campaign, his software agent remotely activate all the victim users' agents and send them the spam message contents as well as a set of emails. Thus, the spam campaign is sent by all infected computers without the knowledge of their users. The large number of attacking machines makes it difficult or impossible either to identify the source of the attack or to take effective corrective action in real time.

### 3.2. How to Know the Spammer's Recourses?

Traditionally, spammers place all the text they want to promote in the body text of the message. However, these method failed by the simple text content filtering techniques. The other alternative is to simply add all the contents on web servers and then send the users the link to the web pages they want the users to visit. A more sophisticated technique which is used massively these days is by using tracking codes that track the users. Spammers use special bulk mailing systems and mail merge. Spammers include HTML image link or Hyperlink links to track the users. They build a large database containing all the emails of the victim users. This database includes special codes as a unique identity of the users, as well as some other codes to distinguish between the spam campaigns. Thus, using the mail merge techniques, customized links containing the tracking codes is sent for every victim user. Once the user opens the email or click on any link or picture of the spam email, on the remote web server, they will know that this specific user is opened the email and/or visited the promoted web page. In addition, they can know the time, from which IP address, etc. Thus, this compromises the privacy of the users and allows the spammers to know much about the users (e.g. their interests, their location).

The war does not stop here. Spam is a real very big business having several players (e.g. email harvesters, message senders). The spammers recently have reengineered their architecture. They have servers (dedicated or even compromised machines that they operate them remotely as if it belong to them). In the new architecture, instead of sending the user the tracking link to the target web page, they added link redirector servers in between. The link redirector server has the database that contains in addition to the tracking URL the target URLs. This is a one-to-many relation. That means the same tracking URL could be redirected to many target URL. Thus, they send to the user a customized URL (HTML link or image link). When the user opens the email and/or click on the target links is chosen and sent back to the browser. In the next list, *ninjaasteroid.com*, *fandragontastic.com*, *click.countrybaby.net* and *engine.gtsmobidistributed.com* are the link redirector servers.

Traditionally, spammers tracks the users using the GET parameters which are included in the URL after the "?" (See Examples 4, 5 and 6 in the next list). However, modern URL rerouting techniques put the form fields in the folder parts of the URL. Examples of such techniques are given in the bellow (Examples 1, 2 and 3 in the next list). From these links that are received in the spam email, we can know much about the spammer resources. We can know the redirector servers as well as almost all the target web servers hosting the promoted web pages or products. It is sufficient to collect the links in the spam emails and using simple code that behave as a browser to get the target server information.

Example 1:
http://ninjaasteroid.com/5264627172690823723 63.UEOVHRA.YTH4D92/197417/141514/3024-006-2-5/b01491b120f69f9534c6ba8ce7106851/euopfedb.5IS6GKTK

Example 2:
http://fandragontastic.com/12081088282704739366.ABKPNZE.CVNOL3L/591323/140557/3472-006-2-5/3b6e615291cb5839bb2928ac038fcbae/luy56abk.D639K5PO

Example 3:
http://click.countrybaby.net/PmgEoCSeXsETlZMEJXUwDzRdWKfrnQEWRijeuRyCDmHRBmlVlrqmdWGihGzV?&n=1585615822&h=a9757f2e2a4885267e6c90e108572ebbf32d26ef

Example 4:
http://engine.gtsmobidistributed.com/www/delivery/ck.php?oaparams=2__bannerid=38413__zoneid=1633__cb=d25aa30b4c__oadest=http%3A%2F%2Fwww.mporn.com%2F%3Futm_source%3Dgtsredirects%26utm_medium%3Dcpm%26utm_campaign%3Dmobile_redirects

Example 5:
http://click.vegasvapor.net/waqCcEoxTjuEeSbiNLIzRGhJlmKEGRzglvpDRYsnRJCPaEcNMdWGEEWGR?&n=1585581499&h=7f5415cf181e3cf08cbcac6837aa1dc534916258

Example 6:
http://www.x3track.com/click.track?CID=209129&AFID=21845&ADID=751949&SID=

 

### 3.3. Proposed Attacking the Spammers' Resources Technique

From Section 3.1, we can see that it is very difficult to stop that spammers using the traditional and even the most modern techniques. In addition, the service providers must dedicate very powerful spam blocking services to filter the spams from the legitimate emails. We know also that much resources of our network are wasted by the spam. From Section 3.2, we can see that knowing the spammers resources (*i.e*. the redirector servers and the target web servers) information is relatively easy and can be automated.

The proposed technique to actively attacking the spammers' resources is as follows:

- Each user having an email box has a spam folder. This folder has all the emails that are classified as spam either using the objective filtering techniques, the subjective techniques or both. This folder must not contain any legitimate emails (an email which is mistakenly classified as spam and hence placed in the spam folder). It is the responsibility of the user to check that periodically.
- A software application must be developed and widely distributed to all the Internet users. This application must be installed on the user's computer (running as client). The user fills in the email accounts' information to this application. Only the credentials and the mail access host/port information (e.g. IMAP/POP3 hostname and port number) are required. The user has to specify the Spam folder(s) for each email account she/he possesses.
- The application periodically reads the new emails in the spam folder(s) of the users and then parses the body text of these emails. It extracts all the links and save them into a local database in the application domain.
- The application periodically selects a link from the database and opens an HTTP request session to this link. It downloads all the content of this link as if the link is opened in a browser by a user. The application parses all the incoming HTTP response messages. If it has any HTTP redirection code, then this link is for a spammer redirector server. Some examples of the HTTP redirection codes are: 301 Moved Permanently; 302 Found; 303 See Other; 307 Temporary Redirect; and 308 Permanent Redirect. In this case, the application does not store the target link, but instead it navigate to it by opening another HTTP request to the returned link from the redirector server. Doing that will place heavy load on both the spammer's redirector servers and their web servers hosting the target pages they want the users to visit.

### 3.4. Advantages of the Proposed Technique over the Passive Techniques

Here is a list of the advantages of using this technique.

- If the application is used by millions of users, all the spammers' resources will be very busy answering the application requests from these users. The resources that will be affected are the redirector servers, the web servers hosting the target web pages; the bandwidth they have (usually limited). This is exactly as if a massive distributed denial of service attack (DDoS) against the spammer resources.
- If any user does not use the application and opens any spam email and click on any link; the redirector servers and the target web servers cannot respond to the user as they are too busy by the massive DDoS attack.
- The spammers have no way to distinguish between the requests coming from a real user and those coming from the application. Thus, they cannot black list it using some footprint techniques.
- The spammers cannot black list all the users. The idea behind that is the large number of users using the application. The success of this methodology depends on the wide spread use of the application. The spammers cannot build black list containing millions of IP addresses that runs on real time.
- One very interesting thing here is that most of the Internet users that would use the application are home users. The access to the Internet with a dynamic IP address given by the ISP. Their addresses changes with time. Thus, if assume that the application is used by only one million persons, after one month the number of different IP addresses that are used by this one million persons could be more than 100 millions. Thus, they cannot ban all the IP addresses.

### 3.5. Challenges and Analysis of the Proposed Technique

Several issues have to be highlighted regarding to the proposed technique:

- *What about the privacy of the end-users*? This issue is studied thoroughly during the design of the proposed technique. As said before, the spammers usually send a tracking code in the link. This allows them to know for a given email if this email is active or not and from where the user is accessing the email (the location or country of the user). Therefore, allowing the application to open the link on behalf of the user allows the spammers to know the mentioned information about the user. This is true! A possible solution to this is to ask any user to create an email account (usually the case for most of the user) that is used as a trap for the spammer. This trapping email account can be spread widely in the Internet forums, form regis-

 

tration, etc. However, if millions of users are using this application, the obtained information cannot harm the user, because the spammers cannot know if this email address (an active one) is really used by the user or just a trapping one. Another issue here is that the user credentials for accessing the email accounts cannot be compromised because the application will run in the user computer not on a public server.

- *What if the link redirectors or the target web servers are compromised* (*infected by Trojan horse*) *without the knowledge of their owners*? It may happen that the spammers are using resources that are not belonging to them. For example, they can hack any web server and install Trojan horse on it making these compromised machines to function as link redirectors and target web server. In this case, the proposed solution can deny the service of these victim servers. Again, this is true! However, this can be used as an alarm signal to the servers administrator who find their servers being under attack to clean these servers from the infection in order to not annoy the others. An aging variable can be used with each link in order to decrease the probability of using this link with time. This can make any server that is being attacked by the application to be removed from the attack after certain time.

## 4. Software Package for Attacking the Spammers Resources

In this Section, we will present a software application that we have developed to validate the approach and to be used by the users to attack the spammers' resources. The application implement the same approach presented in the Previous Section. It is developed using Ms Dot Net Framework v4.0.

The user can add unlimited number of email accounts to the system (see **Figure 1**). The only restriction is that IMAP protocol is allowed for these email accounts. POP3 is not supported in the current version because it does not allow us to access the list of the email folders. The user, in the setting page, enters the IMAP hostname/ port number for the email server. In addition, the user must enter the username and the password for that account. After clicking on the "Fetch Email Folder" button, the application fetches all the email folders for that email account. Then the user has to select which folder contains the spam emails (the junk mail folder). Then clicking on the "Save Settings" button will add the information of this email account to the application database.

After entering all the email accounts of the user, in a separate thread, the application periodically check all the emails found all specified spam folders. It downloads the body contents of all the emails. To avoid downloading an already seen email, the Message ID (MsgID) of the last received email is stored. The body contents of each downloaded email is parsed to extract all the hyperlinks (the *href* attribute of the *anchor* tag) and all the links for the images (the *src* attribute of the *img* tag). The application does not store the body contents, but instead it saves all the extracted unique links in the database. This thread repeats the action every one hour.

In another thread, the application periodically fetches one of the stored links. The application has a hidden web browser (the same engine as MS Internet explorer). This hidden web browser navigate to the fetched link exactly the same way as if a real user clicked on that link in a

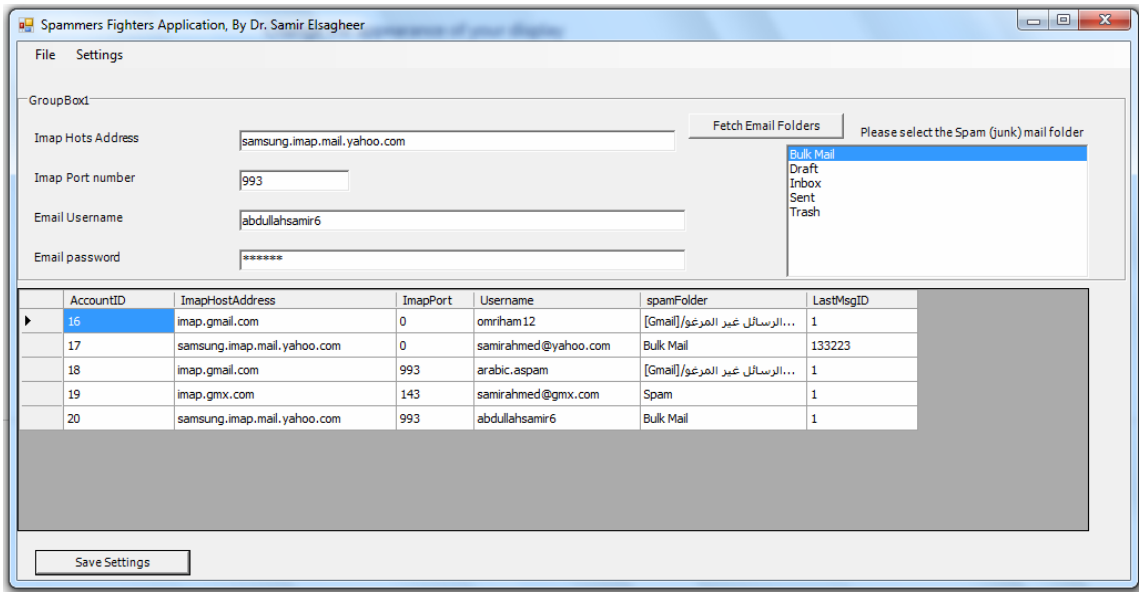


**Figure 1. A Screenshot of the developed application that can be used to attack the spammers' resources.**

 

real browser. In this case, all the HTML and the referenced media files (images, pictures, documents, etc.) are downloaded. The application handles all the cases of HTTP redirections and navigates through these links until it reaches the target web page and downloads all the contents of this page. To make it impossible to detect the application by the spammers, the hidden browser uses the same signature as the Internet Explore.

In this case, this application actively attacks the spammer resources (the servers and their Internet access bandwidth). If this application is used by millions of users, this for sure can constitute a massive DDoS on all the spammers' resources.

## 5. Conclusions and Future Directions

In this paper, an active spammers' resources attacking technique is proposed. The technique consists of using a special software application that reads all the email in the Spam folder of the user. The application grabs all the links that are sent from the spammers and actively open HTTP connection to download the contents of these links from the spammers' servers. HTTP redirections are to be processed recursively. The application has to be used by large number of users. In this way all the spammers' resources, namely the servers and the network bandwidth will be consumed by the application. This constitutes massive DDoS attack against these resources. The advantages of the proposed technique over the passive techniques is presented. To show the effectiveness of the proposed technique to fight the spammers, we have developed an application that can be used for this purpose.

One problem to be tackled in the future research is about the malicious use of the system to launch a DDoS attack on legitimated web server. In order to deny the service of any web server, a spammer can simply send a spam campaign having any valid link hosted on that server. This problem is not solved in the current paper, and left for future work.

## REFERENCES


[1] Z. Li and H. Y. Shen. "SOAP: A Social Network Aided Personalized and Effective Spam Filter to Clean Your E-Mail Box," *Proceedings IEEE INFOCOM*, Shanghai, 10-15 April 2011, pp. 1835-1843.

[2] J. A. Zdziarski, "Ending Spam—Bayesian Content Filtering and the Art of Statistical Language Classification," 5th edition, No Starch Press, San Francisco, 2005.

[3] L. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos and P. Stamatopoulos, "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach," *Proceedings of the Workshop on Machine Learning and Textual Information Access*, 2000, pp. 1-13.

[4] M. Sahami, S. Dumais, D. Heckerman and E. Horvitz, "A Bayesian Approach to Filtering Junk Email," Learning for Text Categorization—Papers from the AAAI Workshop, 1998, pp. 55-62.

[5] D. Karthika-Renuka, *et al*., "Spam Classification Based on Supervised Learning Using Machine Learning Techniques," *International Conference on Process Automation, Control and Computing*, Coimbatore, 20-22 July 2011, pp. 1-7. doi:10.1109/PACC.2011.5979035

[6] H. Drucker, D. Wu and V. N. Vapnik, "Support Vector Machines for Spam Categorization," *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, 1999, pp. 1048-1054. doi:10.1109/72.788645

[7] C.-Y. Tseng and M.-S. Chen, "Incremental SVM Model for Spam Detection on Dynamic Email Social Networks," *International Conference on Computational Science and Engineering*, Vol. 4, 2009, pp. 128-135.

[8] S. Suwa, *et al*., "DNS Resource Record Analysis of URLs in E-Mail Messages for Improving Spam Filtering," *IEEE/IPSJ* 11*th International Symposium on Applications and the Internet* (*SAINT*), Munich, 18-21 July 2011, pp. 439-444.

[9] A. Khanal, *et al*., "Improving the Efficiency of Spam Filtering through Cache Architecture," 15*th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Istanbul, 24-26 October 2007, pp. 303-309. doi:10.1109/MASCOTS.2007.27

[10] S. A. Elsagheer-Mohamed, "Combining Objective and Subjective Smart Technqiye for a Better Spam Filtering Solution," Under Submission.

[11] X. Carreras and L. Márquez, "Boosting Trees for Antispam Email Filtering," *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, 2001, pp. 58-64.

[12] I. M. Rafiqul, *et al*., "Dynamic Feature Selection for Spam Filtering Using Support Vector Machine," 6*th IEEE/ACIS International Conference on Computer and Information Science*, Melbourne, 11-13 July 2007, pp. 757-762.

[13] W. W. Cohen, "Learning Rules That Classify E-Mail," *Proceedings of AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, 25-27 March 1996, pp. 18-25.

[14] X.-L. Pang, *et al*., "The Compensation Strategy of Unseen Feature Words in Naïve Bayes Text Classification," *Journal of Harbin Institute of Technology*, No. 6, 2007, p. 26.

[15] A. Taweesiriwate, *et al*., "Web Spam Detection Using Link-Based Ant Colony Optimization," *IEEE* 26*th International Conference on Advanced Information Networking and Applications* (*AINA*), Fukuoka, 26-29 March 2012, pp. 868-873.

[16] S. Mohamed, *et al*., "A New Technique for Automatic Text Categorization for Arabic Documents," *Proceeding of* 5*th International Conference on Internet and Information Technology in Modern Organizations*, Cario, 13-15 December 2005.

[17] H. Q. Zuo, *et al*., "Image Spam Filtering Using Fou-


 

rier-Mellin Invariant Features," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Taibei, 19-24 April 2009, pp. 849-852.

[18] J.-H. Hsia, *et al*., "Language-Model-Based Detection Cascade for Efficient Classification of Image-Based Spam E-Mail," *IEEE International Conference on Multimedia and Expo*, New York, 28 June-3 July 2009, pp. 1182-1185.

[19] B. Battista, *et al*., "Image Spam Filtering Using Visual Information," 14*th International Conference on Image Analysis and Processing*, Modena, 10-14 September 2007, pp. 105-110.

[20] H. B. Aradhye, G. K. Myers and J. A. Herson, "Image Analysis for Categorization of Image-Based Spam E-Mail," *Proceeding of the* 8*th International Conference on Document Analysis and Recognition*, Korea, 29 August-1 September 2005, pp. 914-918.

[21] Zhen. H. Duan, *et al*., "Detecting Spam Zombies by Monitoring Outgoing Messages," *IEEE INFOCOM*, Rio de Janeiro, 19-25 April 2009, pp. 1764-1772.

[22] Zhen. H. Duan, *et al*., "Detecting Spam Zombies by Monitoring Outgoing Messages," *IEEE Transactions on Dependable and Secure Computing*, Vol. 9 , No. 2, 2012, pp. 198-210.

[23] K. Saraubon, *et al*., "Fast Effective Botnet Spam Detection," *Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, 24-26 November 2009, pp. 1066-1070. doi:10.1109/ICCIT.2009.128

 