2012 2nd International Conference on Industrial Technology and Management (ICITM 2012)
IPCSIT vol. 49 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V49.55

Throttling Outgoing SPAM for Webmail Services

KONG Ying[1+] and ZHAO Jie [2]

[1] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, Zhejiang Province, China

[2] School of Experimentation and Practice Training Management Center Zhejiang Police Vocational Academy, Hangzhou, Zhejiang Province, China

Abstract. Presented a system that dynamically throttles emails based on the message content at the email server provider (ESP) side. The goal of this system is to reduce the spam generated by the ESP while not introducing long delay to legitimate messages. This goal is achieved by applying spam filters during the email delivery time and by using filter scores to control the throttling effect. The throttling effect is implemented through a computational puzzle system. We present experiments and results that show the effectiveness of this anti-spam system that under state of the art hardware, we can limit the ability of the spammer even though he possesses 1000 times as many CPU resources as the normal sender.

Keywords: Webmail services; spam filter; Emulation Topology

1. Introduction

A large amount of research and industrial efforts have gone into reducing spam messages. The current practice is heavily limited to spam filtering at the receiver side, but this practice neglects a serious fact that all spam traffic consumes unnecessary internet bandwidth. Also analyzing those huge amounts of spam overloads the receiving mail server. Since sending spam from ESP is a substantial source of spam emails, ESPs suffer when their IP addresses get added onto blacklists[1,2]. Also complaints from the recipients not only hurts the ESP's reputation, but handling them consumes expensive human support efforts.

This paper addresses the issue of solving those problems by reducing spam messages generated from the ESPs, mostly in forms of webmail services (e.g. Hotmail, Yahoo, and 163.com). The contributions of this work are: 1) we push the spam filter to the early stage of the email delivery time, and2)we combine the spam filter with the computational cost approach and dynamically assign costs to the senders. By pushing the spam filter to the early stage, we can reduce outgoing spam from ESP side and decrease the bandwidth usage in the Internet. Consequently this alleviates the burden on the receiving mail server.

Our work is different from previous outgoing SPAM controls in that we challenge each message and overcome the problem of challenges with constant cost. Our approach does not rely on the precise knowledge of which email is spam. We use the spam filter to estimate the quality of a message. The email filterhas to be score-based and it would produce a spam likelihood score, not a zero-one decision. We choose to delay emails with computational puzzles, and the puzzle diffculty level is based on the filter score. With this approach, users sending low quality messages would be assigned a high computational cost puzzle.

+ Corresponding author.
E-mail address: Kongying-888@163.com.

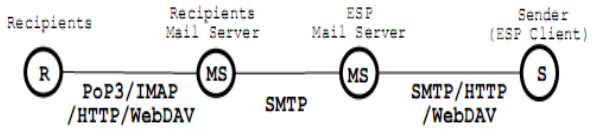

Fig. 1: Protocols for Delivering one Email

Figure 1 shows the popular protocols used today. Because webmail is the most popular form of email interface for most ESPs, we apply this approach to the webmail interface, including mail systems that use HTTP Post and those withWeb-based Distributed Authoring and Versioning (WebDAV) [3].

2. Spam Filters

Most of the current anti-spam research focuses on spam filters. Various forms of filters, such as whitelists, black-lists, and content-based filters are widely used to defend against spam. White-list based filters only accept emails from known addresses. Black-list filters block emails from addresses known to send out spam. Content-based filters make estimations of spam likelihood based on the text of that email message and filter messages based on a pre-selected likelihood threshold. For example, the famous filter t) from Paul Graham assigns a likelihood value to each word or phrase based on its history of use in spam and takes the average as the overall spam-likelihood for the message. Unfortunately all types of spam filters have false positives, with which legitimate messages are misclassified and get lost. Another problem with spam filters is that it can only filter a message after it has already been delivered and stored in the receiver's mail server. The approach presented in this paper also uses spam filters, but for a different purpose–not filtering messages, but estimating their "quality". The quality of the information is then used for selectively delaying messages. Thus a misclassification would only cause a small delay to a message, and the impact of a false positive would be much less severe than the method of dropping messages. This approach is applied at the sender side to reduce outgoing spam, thus it can be used as a complementary technique for the current filtering methods at the recipient side.

2.1. Cost-based Approach

A cost-based approach is the most promising general solution for resisting network abuse, such as spam and network DoS attacks. Cost takes many forms, such as monetary payments, "hashcash", and computational puzzles. By requiring the remote peer to consume some computational resources before granting the service, the protected side can reduce the risk of network abuse.

2.2. Previous work on outgoing spam

Reverse turing test is one well-known cost approach that has been widely adopted by many ESPs to reduce spam. In this approach, users are required to pass a simple test before getting an account. Some ESPs even move a step further and require a reverse turing test before sending any email messages.

3. Our Approach

This section presents our approach of adaptively. Reducing outgoing spam on the ESP side. We first review the current email relaying practice of ESPs, and then we explain how to build the cost mechanism into the ESP message relaying process. Finally we present our adaptive cost assigning system that selectively adds cost to users.

3.1. ESP email delivery protocol

To our knowledge, the current practice is limited to the following protocols: (1)SMTP, (2)HTTP, and (3)HTTP with WebDAV. The latter two are more popular for web based ESP service (used by Hotmail, yahoo, mail.sina, etc) because they provide identifications to the ESP. When HTTP is used, messages are delivered to the server with the HTTP Post command. WebDAV is an extension of HTTP that is designed to enable multiple users to manage and modify the files in a remote system. With WebDAV enabled clients, users can view, open, edit, and save files directly into the file system of the website as if it was a local system.

Since email data are still delivered through HTTP Post command, we present the mail client with WebDAV in the same way as the client purely using HTTP.

3.2. Cost mechanism

The goal of this work is to integrate the cost approaches into these systems and show that putting spam filters at an early stage of email delivery can help reduce the spam from the sender. With the cost mechanism, the server would be able to assign a computational task to the client with a controllable difficulty. The server would then verify the computational results before accepting the messages for forwarding. The cost mechanism has to be robust, tamper resistant, and efficient. Many existing studies have addressed these issues and designed algorithms for this task. We are not going to repeat this task. Instead, we focus on how to combine them with spam filters. We picked a simple computational puzzle algorithm for our system. In this algorithm, when a sender makes a connection and delivers a message to the ESP server, the server randomly generates a string for this connection, calculates and saves the MD5 hash output, and sends the hash output back to the sender The email sender is asked to search for a string that has the same hash output and send back the string as the answer. The server controls the puzzle complexity by controlling the string length and the search space size.

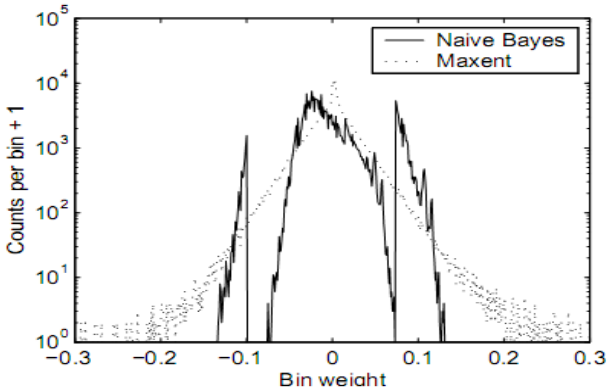

Fig. 2: Histograms of normalized feature weights fornaive Bayes and maxent filters

Process for the two protocols, SMTP and HTTP, respectively. When the client uses SMTP to forward messages to the ESP server, the client's SMTP agent has to be modified. When SMTP is used, and ESP has no control over which SMTP client a user adopts, then adding this mechanism would be considerably harder than the HTTP or HTTP with WebDAV cases. For the later case, the client side software is embedded in the web interface, which can be easily modified by the ESP server to add this cost mechanism, by using a client side script.

3.3. Selective cost assignment

With the knowledge of where in the delivery process we assign the cost, this subsection describes the algorithm of assigning puzzle difficulty. We chose two guidelines for the difficulty assignment. First, we would like to assign no computational cost to every connection if the spam messages are very rare overall. Second, we would like to assign no or negligible computational cost to good email messages even when the overall spam volume is high. To achieve this goal, we design a two level adaptation system in which an email connection's cost is assigned based on a product $C(m) = Q \times q(m)$, in which C(m) is the cost level for a message m, and Q is the overall average message quality level measured over a recent history, and its value is between 0 (low spam ratio) and 1 (high spam ratio), and q(m) is the quality measurement for this individual message with a value also ranging from 0 to 1.

Both the overall and the individual message quality measurements are made by a spam filter. Although spam filters can't judge the spamminess of a message with a 100 percent accuracy, the average score over many messages gives a good indication of the spam and-non-spam ratio. We choose a Bayesian based spam filter called QSF for the message quality estimation at the ESP mail server side. QSF is a lightweight statistical spam filter written in C. In QSF, an overall score is calculated to find out whether the email should

be considered spam or not. An evaluation of QSF by a third party shows its filtering precision is 99.1% with a 0.27% false negative and a 0.02% false positive rate[5].

With this approach, there is still a large design space for choosing an adaptation algorithm. A few issues need to be addressed including over how long a period should the average quality measurements be made, how responsive should the system be towards message quality changes, and how much we adjust the cost each time we sense the quality changes. We ended up choosing one proportional control algorithm, in which the cost level is assigned with the following equation:

$$if \overline{S} - S_m > 0, Q = P * (\overline{S} - S_m)^i$$
$$if \overline{S} - S_m <= 0, Q = 0 \quad (1)$$

Q is the cost we want to calculate, S is the average email score over recent period of time, we update the S periodically, Sm is the mean score value of the good emails from QSF training set. We calculate the distance between the S and Sm, and we raise this distance to the power of i, so that when the quality of emails are low, the score will be most likely high, and sender will get more punishment for those low quality emails. P is a multiplier used to map the value into the puzzle generator's input range. Even with this algorithm, there are several challenges towards achieving this goal. We need to make the false positive impact as tiny as possible when the spam filter makes a low quality estimation for a good mail. We also need to avoid high processing overhead, so that the ESP server can still support a large number of accounts.

To address this limit, we set an upper bound to the cost level, so that even the maximum cost level would not cause a connection to delay more than 5 minutes. This number is estimated assuming the same level of computational power as our experiment machine at the ESP client side. People have concerns that applying cost proportionally to the amount of messages might affect legitimate bulk email senders, such as Amazon and eBay. However, legitimate bulk email senders have motivations to identify themselves with the ESP so that they can be put on the white-list to avoid these computational cost. Furthermore, the cost is not only related to the message volume, but also the message quality. The aggregate cost for a large volume of good quality messages is still low.

4. Evaluation

This section presents an evaluation of our adaptive throttling system at the ESP. We first present the experiment methodology, including the experiment setup and the metrics we used to evaluate the system. Then we present the empirical results for both with and without the adaptive throttling system[6].

Figure 3 illustrates the topology used in our experiments.

All the machines in the systems are 2.6GHz Dell PCs, running Linux 2.4.23, and they are connected through a 100Mbps switches. The mail server is supposed to forward messages to the Internet. Since we only care about the quality of the outgoing email messages, we forward all the messages to /dev/null. We use two machines to emulate normal senders and spammers. Both machines connect to the email server through a NIST Net router that emulates the network between the clients and the ESP mail server.

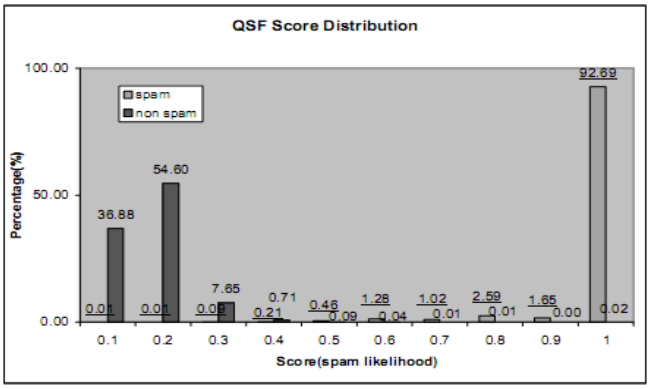

Fig. 3: Email Score Distribution

Because we are emulating many senders using a single machine, in our emulation, the cost of a computational puzzle given to the sender is reduced proportionally. A similar arrangement happens at the spammer side. The resource ratio is controlled with a parameter, and results of different ratios are presented in the next subsections. The cost mechanism we can see in the fig 3.

5. Experiment Results

As a reference to measure the effectiveness of this outgoing spam control, we first measure the overall spam ratio, the goodput, and the delay behaviors of normal emails. We run a server with both emulated normal users and spammers. We further assume that the server supports 100,000 users, and each user in average sends five emails a day. This number was obtained from a recent study on a British ESP[7] .Our own measurement over a nation wide ESP shows a similar rate. We control the normal email rate according to this average, and we emulate a spammer that sends emails in a best-effort way. In our measurements, we found the CPU is the bottleneck for email senders, rather than memory or bandwidth. With the best-effort strategy, the spammer automatically accept the whatever cost assigned from the ESP. We vary the spammer's CPU resources to show its impact to the spam ratio and the goodput and delay for normal messages. This result is presented in table 1. In this result, the spammer's CPU resources are represented by its ratio to the normal users' average computational power. Typically the ratio is around 1, meaning that spammer uses a similar powerful machine as a normal user does. We consider the typical ratio range is between 0.1 and 10. The ratio is increased when spammer has a top-of-the-line system, or compromises a good number of zombie machines for sending emails. So we also consider some larger ratio to represent this scenario. The result indicates that when the spam volume increase proportionally with the spammer's resources the spam volume can bypass that of normal email messages (which is the goodput). The spam volume stops increasing once the email rate is high enough to hit the server's maximum throughput. Under this situation, the majority of messages are spam. Arguably, the Internet email system is getting close to this situation, given reports that more than 50 percent of Internet messages are spam.

TABLE I. Delay for the normal messages with best effort spammers (SD: Standard Deviation)

Resource Ratio	***Average Delay(sec)***	***SD***
0.1	0.56	1.44
1	0.85	3.53
10	2.43	6.60
100	6.9	10.60
1000	28.33	17.28

To quantify this impact, we present the measured normal email delay result in Table 1, including both average delay and standard deviation of the average delay. The result is that most of the messages have very low delay when the spammer's resources are low or comparable to a normal user's resources. The delay increases when the spammer's resources get higher. However, the average delay is still within tens of seconds. The worst delay to legitimate emails are controlled below the maximum cost level which is 5 minutes. This time interval has been commonly used in the email delivery for timeout value, such as the SMTP commands.

6. Conclusion

A web-based email service is the most popular interface used by the existing email service providers. In this paper, we presented an anti-spam system for ESPs in order to reduce spam messages originating from them. The system dynamically assigns costs based on the estimated quality of the messages, and the quality is derived from scores produced by a spam filter. Experiments show that by dynamically assigning the cost based on the message quality, the system slows down spammers but assigns zero (or little cost) to the normal messages because they tend to have high quality. The majority of normal messages belong to this category. Misclassified messages (false positives) would not be dropped but only incur slight delay.

7. References

[1] Philip Jacob. The Spam Problem: Moving Beyond RBSs, 2003. http://theory.whirlycott.com/rbl.

[2] Michelle Delio. Not All Asian E-Mail is Spam. In Wired News, Feb 19 2002.

[3] Paul Festa. Microsoft anti-spam campaign hypocritical. available at http://news.zdnet.co.uk/business/.

[4] Prince, M., Dahl, B., Holloway, L., Kellera, A., & Langheinrich, E. (2005). Understanding how spammers steal your e-mail address: An analysis of the first six months of data from project honey pot. Proceedings of the Conference on Email and Anti-Spam.

[5] Tresp, V., & Yu, K. (2004). An introduction to nonparametric hierarchical Bayesian modelling with a focus on multi-agent learning. In Switching and learning in feedback systems, vol. 3355 of Lecture Notes in Computer Science, 290–312. Springer.

[6] Richard Clayton. Stopping Spam by Extrusion Detection. In Proceedings of the First Email and SPAM conference, July 2004.

[7] Goldstein, J and Sabin, R. Using Speech Acts to Categorize Email and Identify Email Genres. In Hawaii International Conference on Systems Sciences, HICSS, January, 2006.