International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:2, 2011

An Efficient Spam Mail Detection by Counter Technique

Raheleh Kholghi, Soheil Behnam Roudsari, and Alireza Nemaney Pour

Abstract— Spam mails are unwanted mails sent to large number of users. Spam mails not only consume the network resources, but cause security threats as well. This paper proposes an efficient technique to detect, and to prevent spam mail in the sender side rather than the receiver side. This technique is based on a counter set on the sender server. When a mail is transmitted to the server, the mail server checks the number of the recipients based on its counter policy. The counter policy performed by the mail server is based on some pre-defined criteria. When the number of recipients exceeds the counter policy, the mail server discontinues the rest of the process, and sends a failure mail to sender of the mail; otherwise the mail is transmitted through the network. By using this technique, the usage of network resources such as bandwidth, and memory is preserved. The simulation results in real network show that when the counter is set on the sender side, the time required for spam mail detection is 100 times faster than the time the counter is set on the receiver side, and the network resources are preserved largely compared with other anti-spam mail techniques in the receiver side.

Keywords— Anti-spam, Mail server, Sender side, Spam mail

I. Introduction

Electronic mail, hereinafter “mail”, is one of the most popular and efficient communication methods in the world. Although it is popular and efficient, unwanted and unsolicited mails endanger the existence of the mail system with massive and uncontrollable amount of messages called spam mail [1]. According to reports, the spam mails traffic has increased from 40% in 2002 to 50% in 2008. Every day, millions of people experience receiving spam mails, and use a lot of time to get rid of them. Moreover, spam mails consume the bandwidth of the related network.

The existing techniques to detect and to stop spam mails are classified to reverse lookup [2], black list [3], white list [4], content-based filtering [5], rule based filtering [6], and moderation [7]. However, even with these techniques an effective solution has not been proposed yet to detect and stop spam mails completely. The main purpose of these researches is to improve precision of the existing techniques to stop the spammers; an unidentified user with enough knowledge that is able to use many techniques to send spam mails.

Obviously, all the existing anti-spam techniques have common limitations. First of all, they have been designed to control the spam mail on the receiver side. As a result, the network bandwidth and the receiver’s memory are loaded with unnecessary functions. Second, the cost of the spam mail prevention is high compared to its effective functionality.

This paper proposes an efficient technique to detect and to block spam mails before being transmitted through the network. This technique focuses on sender side spam mail detection rather than receiver side anti-spam technique. As definition, a spam mail is an unwanted mail which is sent to large amount of addresses by a spammer. Therefore, the key point for our proposal is the large number of addresses from the sender side. By applying a counter to the mail server of the sender which can control the number of transmitting mails, it will be possible to stop large amount of unwanted mails from being transmitted through the network.

This paper is organized as follows. Section 2 gives an overview to mechanism of mail protocol and discusses previously proposed techniques. Our proposal is presented in section 4. Section 5 covers the results of our proposal. Finally, Section 6 is the conclusion.

II. Related work and Overview

A. Related Work

Many solutions have been proposed [2]-[11] to detect and stop the spam mails. Most of them have tried to stop the spam mails at the receiver side based on several techniques such as reverse lookup and filtering [2]-[11]. A reverse lookup also known as a reverse DNS (Domain Name System) lookup [2] determines the host associated with a given IP (Internet Protocol) address. This technique is not effective for the mobile users and the users with invalid IP address. In black list filtering [3], the administrator of the mail server sets the IP address to function in certain fields. It means that if corresponding mail server is in black list, the mails from that server may not be delivered. The white list filtering [4] has fewer restrictions regarding this matter. The mails from the white list are in all the times acceptable.

The other methods include other filtering techniques [5]-[11]. The filtering techniques can be classified in two types, content-based filtering, and rule-based filtering. All content based filtering techniques are applied after the receipt of the body of message. The filtering technique is constructed on known keywords. A newer technique regarding filtering is the Naïve Bayesian [8]-[9]. It was proven to be an effective method that consists of two phases, training phase and testing phase. It works by associating of words and Bayesian statistics to compute a probability in order to determine whether the mail is spam or not.

All of the methods mentioned above have some complications. First of all, spammers have enough knowledge against those functions and actions of filters. Therefore, they

Raheleh Kholghi is with Dept. of the IT Engineering, Sharif University of Technology, International Campus, Kish Island, IRAN (e-mail: r.kholghi@kish.sharif.edu).

Soheil Behnam Roudsari is with Dept. of the IT Engineering, Sharif University of Technology, International Campus, Kish Island, IRAN (e-mail: s.behnam@sharif.kish.ac.ir).

Alireza Nemaney Pour is with Dept. of the IT Engineering, Sharif University of Technology, International Campus, Kish Island, IRAN (e-mail:, pour@kish.sharif.edu)..

can update their methods to bypass the filters. On the other hand, the filters can only detect the spam mails, and move them to a special box instead of stopping them. These techniques lead users to time consuming issues because users should manually review the mails to decide whether a mail is spam or not. Consequently, these techniques do not prevent spam mails completely.

B. An Overview to Mechanism of Mail Protocol

Before we present our proposal, we present the mechanism of mail protocol briefly. For mail transfer between sender and receiver SMTP (Simple Mail Transfer Protocol) is used. Fig. 1 illustrates the sequence of the SMTP commands between the client and its corresponding server in sender side. First, a TCP (Transmission Control Protocol) connection on port 25 is established. Next, the client is validated by its mail server. After that the client announces the mail addresses of the recipients to the mail server. If the receivers are factual (checked by DNS), the mail server requests the body of the mails. Each time the server replies back to the client by “250 ok” for confirmation. When the client wants to leave the session a “Quit” command is sent to the server. At the end, the SMTP and TCP connections are terminated.

To transmit the mail from the sender side (server) to the receiver side (server), the same procedure shown in Fig. 1 will be done. When the body of the mail is received by the server, spam mail filtering is started. If the mail is recognized as spam, based on the policy of the mail server the mail will be halted. Then, the rejected mail is returned to the originating sender.

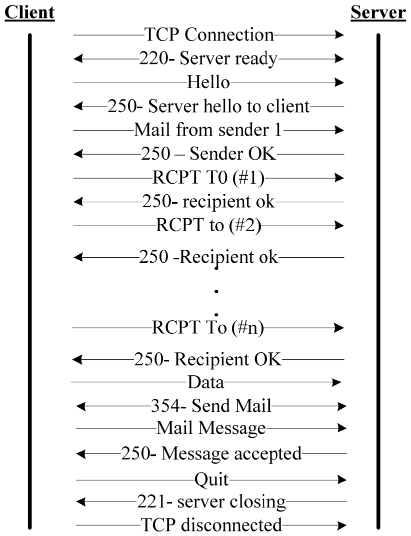

Fig. 1 Sequence of mail transfer between the client and the server

III. DESCRIPTION OF OUR APPROACH

In this section we describe our technique for spam mail prevention at sender side. The key idea of our technique is blocking spam mail in the sender side, based on the number of recipient policy. This technique is based on using SMTP protocol by applying a counter after the session establishment for each recipient. The counter can be set by the administrator of the mail server. To clarify the technique we describe it in the following steps.

1) The client connects to its mail server. We will refer to this step later as TCP connection phase.
2) After the TCP connection, the authentication phase is done by the mail server of the sender. If the client is authenticated, the process is continued to the next step; otherwise the TCP connection is terminated.
3) In the next step, the number of the recipients is verified. This step depends on the administrator of the mail server. We will address this step as the counting phase.
4) Following the counting phase, the SMTP step will be initiated. This phase is the SMTP connection.
5) After the above procedures, the data is transferred to the mail server.

Fig. 2 illustrates the flowchart of our technique based on the above steps. In the TCP connection phase, the client needs to connect to its mail server. For this purpose, the IP address of the mail server is demanded by its ISP (Internet Service Provider). Then, the client sends a request to DNS server to acquire the IP address. The DNS server replies back to the client. This procedure is done via UDP (User Datagram Protocol). After the IP address of the mail server is obtained, TCP connection is established.

First, the client sends the synchronization message to the mail server. Accordingly, the mail server sends the synchronization acknowledge and synchronization agreement. To finalize the procedure, the client sends a TCP acknowledge. On the other side, the mail server of the sender sends “OK” message as a confirmation. The first step is completed, and the TCP connection phase is accomplished.

In the next phase, the authentication phase, the mail address of the client is sent to the mail server by USER command via POP3 (Post Office Protocol). The mail server replies back with an acknowledgement, and requests for the password. The client sends the password to the mail server. The mail server checks the password for its validity. If it is correct, then the mail server sends an “ACK” to the client and authentication phase is ended.

After the authentication phase, we have added a counter to count the number of the recipients. As stated before, this counter can be set by the administrator of the mail server. Based on this counter, if the number of the recipients exceeds the allowed number of the counter, the mail is not accepted by the server. At this time, the mail server replies back to its client with an error message. Notice that for each mail address of the recipients the server checks the validity of the address by DNS. The SMTP commands are continued after the counting phase as shown in Fig. 1.

The key point in our proposing technique is the counting phase. Since the validity of each mail address is checked by DNS, a spammer can consume the network resource in the sender side by setting a huge amount of addresses. But, by setting a counter in the sender side, the activity of the spammers can be limited.

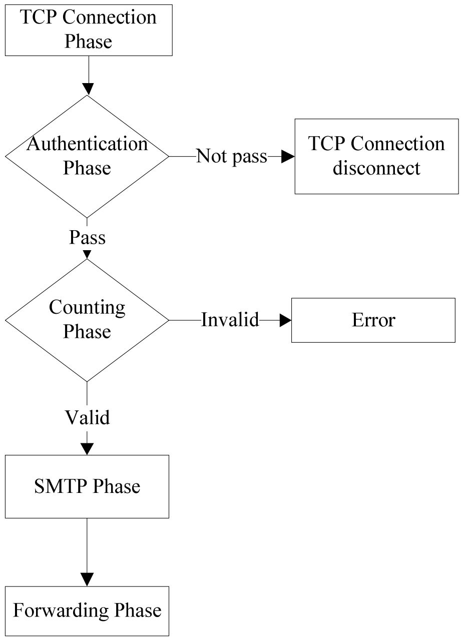

Fig. 2 Flowchart for our counter technique

IV. Results

This section describes the results of our proposal. Fig. 3 illustrates the network structure of our proposal. We implemented the real network by 20 clients and 4 mail servers. The DNS server is set on windows server 2003. In addition, we set a domain for each mail server such as “a.com” for the sender mail server and “b.com for the receiver mail server. For better understanding, we have shown the results of two mail servers as the sender and receiver however, this simulation can be extended to several servers.

In this network, we assumed that client1 is going to send many mails to other clients. As mentioned before, the client should obtain its mail server IP address. For this purpose, the reply from DNS server takes 2ms. After receiving the IP address, TCP connection and authentication is established. The time needed for the authentication of the client is utmost 4ms. As an administrator of the mail server we set our counter to 20 recipients per message. This implies that the sender cannot exceed 20 recipients for each mail.

For comparing our results between the sender side and the receiver side, we set the counter in different positions, once in the sender side and once in the receiver side. In the former position, we set the counter to 20. As soon as the mail server receives the mail address of the first recipient, the counter indicates 1 recipient, and increases with the next recipient and so forth. This counter stops at 20. If there are no more receivers, the next steps are continued by the mail server otherwise the mail server sends a failure message, and notifies the client that the number of the recipients exceeds the limit. The time needed to process this transaction is negligible.

Fig. 4 shows the result of the counter set in the sender side. In this simulation when the counter is set to 20, a client with more than 20 recipients in its mail (n > 20) receives error. The response time for this error is roughly 6ms. This time is required for the DNS request and authentication of the client. On the other hand, if n < 20, the client’s mail is transferred successfully. The time required for this mail step (n < 20) is utmost 630ms. The fluctuation occurs because part of the network connection is connected by wireless.

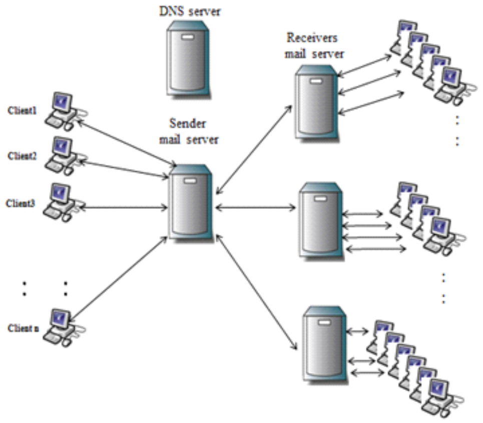

Fig. 3 Network structure of simulation

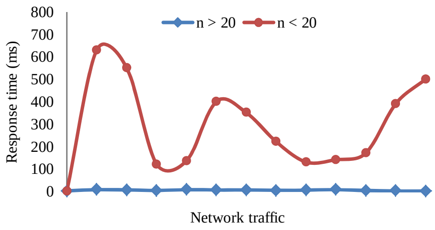

Fig. 4 Mail server traffic in sender side

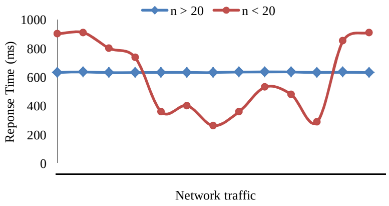

Fig. 5 Mail server traffic in receiver side

From another perspective, we set the counter on the receiver side. When the mail server of the sender receives mail, it looks up the IP address of the receiver mail server. Immediately it checks the DNS server and obtains the IP address. Next, the mail server of the receiver sends a "WELCOME" message to the sender mail server. After authentication, the receiver's mail server starts the process as demonstrated in Fig. 1. The moment the mail server receives the first mail address the counter is increased. This procedure is repeated according to the number of mail addresses. Again, if the numbers of the recipients is more than 20 (n>20), the mail server sends a failure message to the sender mail server, and the process is discontinued. The time for this process is nearly 600ms. At this time, the body of the mail is not checked. If the number of recipients is less than 20 (n<20), then the mail server will accept the mails and requests for the body of the mail. The maximum time for this step is about 930ms. Fig. 5 illustrates the results of the maximum and minimum time needed for receiving the mail.

Based on the simulation in real network, when the counter is set on the receiver side, the time required to achieve a full cycle of the above criteria is nearly 100 times more than the time the counter is set on the sender side. This is clearly shown in Figs. 4 and 5. When the counter is set on the sender side, the response time takes 6ms to determine whether the mail is spam or not while when the counter is set on the receiver side this time reaches up to 630ms.

V.Conclusion

In this paper, we introduced an efficient technique to detect, and to prevent spam mails on the sender side. In our technique, the mail server of the sender checks the number of the receivers. If the sum of the mail addresses is more than the allowed limit, the filter will block the mails, and the sender receives a failure message from the mail server, and the transaction is terminated. In this technique, we proved that our idea is efficient because just the resources in the sender side are accessed. This implies that if a mail is identified as spam, the receiver's bandwidth and memory is preserved which will assure a better performance.

References

[1] C. Dhinakaran, Jae Kwang Lee, and D. Nagamalai, "An Empirical Study of Spam and Spam Vulnerable email Accounts," *in IEEE Conf.* of *Future Generation Communication and Networking (FGCN 2007), Jeju, Korea,* 2007, pp. 403-413.

[2] B. Agrawal, N. Kumar, and M. Molle, "Controlling spam emails at the routers," *in Proc. of the 2005 IEEE International Conf. on Communications (ICC 2005),* Seoul, Korea, 2005, pp. 1588-1592.

[3] A. Ramachandran, D. Dagon, and N. Feamster, "Can dns-based blacklists keep up with bots?," T*he Third Conference on Email and Anti-Spam*, July 27-28, 2006, California, USA, 2006.

[4] A.Ramachandran, and N. Feamster, "Understanding the network-level behavior of spammers," *In Proc. of SIGCOMM*, Pisa, Italy, 2006, pp. 291-302.

[5] A. Ciltik, and T. Gungor, "Time-efficient spam e-mail filtering using n-gram models," *Pattern Recognition Letters*, vol. 29, no. 1, pp. 19–33, Jan. 2008.

[6] G. Cormack, and A. Bratko, "Batch and online spam filter comparison," *In Proc. of CEAS*, California, USA, 2006.

[7] I. Androutsopoulos, G. Paliouras, V.Karkaletsis, G. Sakkis, C.D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A Comparison of a Naïve Bayesian and a Memory-Based Approach," *4 European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000),* Lyon, France , 2000, pp.1-13..

[8] M. Saiful Islam, S.M. Khaled, K. Farhan, A. Rahman, and M. Rahman, "Modeling Spammer Behavior: Naïve Bayes vs. Artificial Neural Networks," *IEEE International Conference on Information and Multimedia Technology (ICIMT 2009),* Jeju, Korea, 2009, pp.52-55.

[9] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz, "A baysian approach to filtering junk e-mail," *AAAI Workshop on Learning for Text Categorization WS-98-05,* Madison, Wisconsin, 1998, pp. 55–62.

[10] P. Roy, A. Roy, Amrit, and V. Thirani, "A New Approach Towards Text Filtering," *2nd International Conference on Machine Vision (ICMV '09),* Dubai, UAE, 2009, pp. 282-285.

[11] S. Jungsuk, D. Inque, M. Eto, Kim C. Hyung, and K. Nakao, "An Empirical Study of Spam: Analyzing Spam Sending Systems and Malicious Web Servers," *IEEE/IPSJ 10th Annual International Symposium on Applications and the Internet,* Seoul, Korea, 2010, pp.257-260.