Design of "E-COP": A Distributed Anti-Spam Environment to Prevent Spamming in E-mail

Mubashsharul Islam Shafique, Md. Munir Hossain, Shiekh Sadid-al-Hasan,
Abu Saleh Shah Muhammad Barkat Ullah
Department of Computer Science and Information Technology,
Islamic University of Technology, Board Bazar, Gazipur 1704, Bangladesh.
shafique.cit@gmail.com,munir_iut@yahoo.co.in,sadid_hasan@yahoo.com,barkat@iut-dhaka.edu

Abstract

Unsolicited e-mails pose considerable inconvenience on part of the frequent Internet users. Unfortunately it is not possible to have human moderators monitoring e-mails of other people for possible spammers all the time. It is also a serious issue violating privacy acts, which most mail service providers fear. Individuals as well as e-commerce sites are also getting spammed on regular basis. The reason for this growing trend of spamming is the fact that there is no efficient filtering system fully developed as yet. In this paper, Electronic COP (E-COP) has been proposed. E-COP is a filtering system against known spams as well as newly evolved spam messages by using grid networks incorporated with some protocol level changes. Objective of this paper is to define structure and working principle of E-COP.

Keywords: Electronic Cop (E-COP), Grid architecture, ESEC, Bayesian filtering, MTA, UA.

I. INTRODUCTION

Spam is an abuse of electronic messaging in the form of blind posting of unsolicited e-mail messages to a very large number of recipients. Spam messages are usually sent through bulk-mailers and address lists from www services such as web pages, forums, news groups, and so on. Spammers are capable of generating and transmitting millions of messages in the span of hours thanks to the huge processing power and advent of broadband Internet. A survey made by Radicati Group indicates an organization with 10,000 employees spends an estimated $71.91 per mailbox per year because of spam, and the worldwide cost of spam to businesses is expected to be $30 billion this year, and $113 billion by 2007 [1]. Brightmail Probe Network [2] found that spam alone was responsible for 62% of all Internet e-mail in February 2004. In 2003, the figure was only 24%.

The traditional methods to cope with spamming are legislation, sender verification, filtering methods such as Bayesian-based algorithm [3], Peer-to-peer based algorithm [4], etc. In this paper, we designed a distributed anti-spam environment empowered with grid technology [5]. The proposed architecture incorporates some existing best ideas and introduces new concepts to combat spam in distributed fashion under a common grid. The E-COP incorporates extension of existing SMTP with extra security issues for authentication.

II. PREVIOUS WORK

Motivation for E-COP came from the P2P-based approaches. P2P-based algorithms to design spam filters came as an alternative to popular Bayesian filtering methods. P2P-based approaches uses hash functions to generate the message digest of an e-mail. Then it looks up the digests from a centralized or decentralized storage. The decentralized autonomous nature of P2P-based approaches leads to unpredictable performance and complex routing and searching methods.

III. THE PROPOSED E-COP FRAMEWORK

A. Objectives

The prime objective of our proposed E-COP architecture is to provide convenient environment in E-mail system. For any design, Quality of Service (QoS) is a major issue. In the e-mail system, the better service implicates the efficient filtering of spams. Under this consideration, the essential features included in E-COP architecture are:

- To eliminate all unwanted E-mails and not to eliminate wanted E-mails.
- To require no user input on the part of either the sender or receiver.
- To be compatible with all uses and e-mail infrastructure configurations.
- To be scalable, to remain effective if 90% of Internet users adopt it.
- To resist attempts to evade it.
- To automatically learn new type of spam.
- To create no new problems.

These are the motivations behind the design of the E-COP. The reason behind our proposal of E-COP is that it has more features than essentials and all good existing methods are incorporated together to enhance the best possible fights against spams.

B. The Distributed Architecture

E-COP stands on grid technology. Open Grid Services Architecture [6] has been chosen to define the framework of the proposed E-COP. Grid technology is used because spam messages are delivered globally. To solve this problem, a global infrastructure to collect spam information is required. The servers, clients and e-mail addresses keep changing all the time. This ever-changing phenomenon can be tackled by the dynamic information sharing property of E-COP grid.

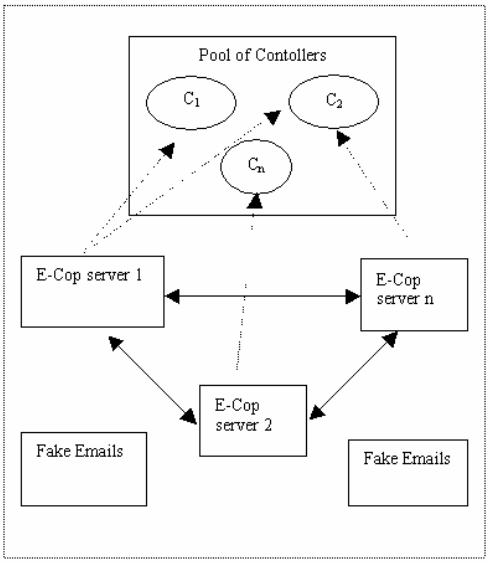

Fig.1 E-COP Architecture

The proposed Grid-framework of E-COP contains E-COP servers. When an E-COP server boots up, it joins the grid of E-COP servers. At that time it exchanges information updates with neighbors. Some controllers control grid operations. These controllers work as information centers and neighbor assigner for the new member within E-COP grid.

E-COP servers will have a list of fake e-mail addresses. Honeypot [7] is used in this architecture to identify genuine spam. The E-COP servers periodically communicate among each other. Hence, is the name E-COP or electronic police officer. In real world, police officers in various stations exchange information of different criminals intercepted at different localities. Similarly in E-COP architecture, the distributed anti-spam servers catch unfriendly spam messages and globally inform the whole grid about them. The distributed architecture of this grid ensures that no single point of failure shuts down E-COP totally. If any particular server goes down, subscribers of that server will be assigned to the then near-most E-COP server via controllers and E-COP will still be functional.

C. Screening at An E-COP Server

To screen any inbound E-mail, the E-COP server needs to ensure the following four things:

1. Legitimate mail not in spam folder.
2. Spam in spam folder.
3. Legitimate mail in inbox.
4. Spam not in inbox.

The functionality of our proposed E-COP inherits all these properties. In the preprocessing phase, the mail is checked whether it has come from a known spammer or not. This checking is performed with the predefined blacklist residing in the E-COP server. If the sender of the mail is found already blacklisted, quickly the mail is marked as a spam and an entry is made in the checked mail table. The mail is also checked against the entries of a white list (the list of good and reliable senders) and if a match is found, it is ranked less probable to be a spam and the mail is sent for further processing.

The header analyzer extracts the subject, timestamp and other necessary information and then the body analyzer checks the body of the mail based on some predefined rules such as keyword matching. The attachment filter works on the attached file of the e-mail (if any) to find any kinds of possibility to mark it as a spam. Our proposal is to implement Bayesian filtering technique as it takes the whole message into account - it recognizes keywords that identify spam, but it also recognizes words that denote valid mail. Bayesian filter is constantly self-adapting, is sensitive to the user, and is multi-lingual and international and difficult to fool, as opposed to a keyword filter. Thus after all the phases, the decision is taken regarding the mail and is stored in the checked mail table. This mail table is the repository of all information regarding all mails arrived at this server and this information is shared with all other servers in the grid. Here is the pictorial view of the functionality of the E-COP just described:

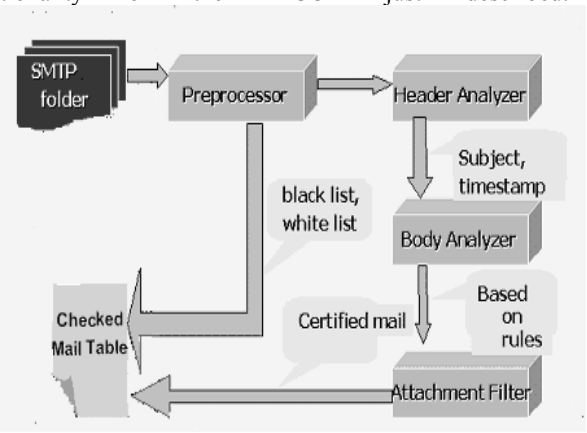

Fig. 2 E-COP Filtering Operation

D. Proposed Algorithm

Module 1: Boot up process For E-Cop servers.

/* *this module works in the E-COP server of the grid for proper authentication. The controller of the grid monitors it. The module is the first step for connecting to the grid for e/xchanging spam information.*/

a. Mail server and E-COP server boot up and check their time stamp when they were in the grid last.

b. IF it is not the present timestamp THEN

i. FIND the IP address of the controller of the E-COP grid.

ii.IF this fails THEN

1. PRINT error message "Cannot find controller".

2. EXIT

ELSE

1. SEND a request for booting up along with the 1024 bits authentication key to the port CP1 (a port for receiving request for interested servers).

2. WAIT for reply message

3. IF returned message is negative from controller THEN

a1. PRINT error message "Cannot hookup with controller. Access denied!"

a2. EXIT

ELSE

goto Module 2.

ENDIF

ENDIF

ENDIF

Module 2: Updating screening information from the grid.

*/*this module is used in the E-COP server for getting updated spam information. The grid controller coordinates to assign the source of the updated data.*/*

a. OPEN a port P1 for exchanging information between this server and the neighboring assigned E-COP server.

b. SEND request for the latest information since the last update of this server with port address P1

c. WAIT for reply

d. UPDATE information table of this server from the received information.

e. OPEN a port P2 for receiving information from the controller.

f. NOTIFY the controller that all tables have been updated successfully

g. NOTIFY the E-COP server that all the tables have been updated and LISTEN the port P2 in a certain interval

h. IF any request received from the controller THEN

go to module 4.

ENDIF

i. LISTEN the port P1 in a certain interval

j. IF new message found THEN

1. Update the table

2. SEND the new information to the other neighboring servers

ENDIF

Module 3: Screening the mails of the mail server.

/ this module is used in the E-COP server for screening inbound mails from MTA. */*

a. SEND message to own mail server that all tables have been updated successfully.

b. GET the new incoming mails not checked yet and all out going mails.

c. goto module 5

Module 4: Handle request received from the controller.

*/*this module is running when an E-COP server. It listens for any request from the grid controller. */*

a. WAIT for request.

b. FIND how much information is to send to the other requesting E-COP server.

c. SEND the information to the E-COP server.

d. WAIT for receiving a port address for exchanging information now on

e. ADD the port address to the table of neighbors.

Module 5: Screening technique.

*/*this module works in the E-COP server for incorporating the efficient technique for screening the mails taken from the MTA*/*

a. GET the sender address from the header of the mail

b. IF the address is blacklisted THEN

i. make spamranking high(more probable as spam).

ii.goto step d.

ENDIF

c. IF the address is whitelisted THEN

make spamranking low(less probable).

ENDIF

d. GET the subject of the mail.

e. Keyword search and add value in the keyword metric

f. IF the subject is random generated and similar type of subject found earlier THEN

ADD value in the randomcopy metric

ENDIF

g. GET the body of the mail.

h. Keyword search and add value in the keyword metric.

i. Spam-tracking databases search.

j. APPLY Bayesian filtering technique.

k. Hyperlink search parsing HTML and removal of comments.

l. Assign special tokens for HTML tags and URLs.

m. IF mail contains attachment THEN

1. GET the mail attachment.
2. IF it is a text file THEN

a1. goto step i.

a2. IF it is a HTML file and IF there is less token but more HTML tags (like the <IMG> tag) which will be more prevalent in spam over time.

ENDIF

ENDIF

Module 6: Procedure to shut down.

*/*this module is used by the E-COP servers in the grid when they want to quit from the grid*/*

a. NOTIFY the controller that this server is going to shut down.

b. NOTIFY all the servers connected with this server of shutting down.

c. CLEAR the port addresses of the table (who are connection with this server)

c. EXIT.

Module 7: Procedure for the E-COP Controller.

*/*this module is used by the controller controlling the grid. This is the main procedure of the controller*/*

(Assuming that all the authentication identification numbers (1024 bits) of the E-COP servers known to the controller and it is saved in the ID table and a port CP1 is opened for receiving the E-COP servers request)

a. WAIT for an E-COP server request.

b. IF request found from a new server wanting to join THEN

i. GET the authentication number of the server

ii. COMPARE it with the stored numbers in the ID table

ENDIF

c. IF match found THEN

i. GET the location of the E-COP server.

ii. FIND the best E-COP server with whom the requesting server may hookup.

iii. SEND a message to the server that is selected in the previous testing of the selection and SEND the IP address of the requesting server to the selected server.

iv. SEND the IP address of the selected server to whom the requesting server will hookup.

ELSE

SEND message “Authentication failed. Access denied!”

ENDIF

d. IF request found from an existing server that wants to quit THEN GET the table (who are connection with this server) of the server

ENDIF

e. WHILE there is at least a server that is assigned to a new one DO

i.FIND the best E-COP server with whom the requesting server may hookup.

ii.SEND a message to the server that is selected in the previous testing of the selection and SEND the IP address of the requesting server to the selected server.

iii.SEND the IP address of the selected server to whom the server (i) will hookup.

END WHILE

IV. NETWORK PROTOCOL BINDINGS

The Grid framework of E-COP can be instantiated on a variety of different protocol bindings. SOAP+HTTP with TLS for security is one example, but others can and have been defined. Here we discuss some issues that arise in this context:

Reliable transport. To share the up-to-date spam information, E-COP require support for reliable service invocation. One way to address this requirement is to incorporate appropriate support within the network protocol binding, as for example in HTTP-R [8].

Authentication and delegation. Illegal manipulation and attack from hackers may occur during E-COP operation. So we require support for communication of proxy credentials to remote sites. One way to address this requirement is to incorporate appropriate support within the network protocol binding, as for example in TLS extended with proxy credential support [9].

Ubiquity. The Grid goal of enabling the dynamic formation of anti-spam E-COP servers from geographically distributed locations means that, in principle, it must be possible for any arbitrary pair of servers to interact.

GSR format. The Grid Service Reference (GSR) can take a binding-specific format. One possible GSR format is a WSDL document; CORBA IOR is another.

V. PERFORMANCE ANALYSIS

Anti-Spam Grid is based on two presuppositions. The No.1 presupposition is that an e-mail is called a spam only if it is sent to too many recipients. Thus we assign a Metric value to each distinct e-mail, which is the number of copies, people receive the e-mail. We can now tell whether a new e-mail is a spam or not based on its Metric value. The No.1 presupposition can be corroborated by the following facts: Spam has low response rates (on the order of 15 per million), so spammers make up for it with high volumes. Each spammer sends out between 10 to 50 million of spam every day, to address lists scraped from all over the net, obtained from other spammers, copied from spam CDROMs, etc. Over 90% of all the spam received in North America and Europe originates from only about 200 senders [10]. In fact, Brightmail [12] has a similar idea on this, which maintains a network of fake e-mail addresses. Any e-mail sent to these addresses must be spam and can be filtered out when users receive the same e-mail. The system will calculate “signature” for each e-mail and it would be unlikely that a different e-mail would have exactly the same signature. Thus the signature represents the e-mail and can be used to compare whether two e-mails are the same. Unfortunately, spammers can attack the signature-based filter method by adding random stuff to each copy of a spam to give it a distinct signature. As a result, it catches only 50-70% of spam [11]. As we’ve said that we will use the best algorithms for detecting spam in the grid so, to avoid this problem, we can use fuzzy Metric values instead of accurate ones. The overall Metric value of an e-mail will be calculated out from a set of neighboring Metric values. Though the length of the checksum is large enough to minimize the possibility of different e-mails have the same checksum, it is still possible when the number of e-mails increase. So, we should aging the checksums. By these means, there's a fairly high chance of overcoming the tricks used by spammers to add random stuff to e-mails. The No.2 presupposition is that *the information gathered by many computers will be more accurate and complete than that from only one computer*. Thus we will apply a distributed Bayesian filtering algorithm, which will carry out Bayesian studying process among hundreds of thousands of client computers and then collect and spread the up-to-the-minute information to all active clients. Also, servers can make a great contribution to this process by setting up numerous fake e-mail accounts to attract spam and then screen it. We will use methods used by rule-based filters those can also be integrated into our Bayesian algorithm, such as whether the user has ever received legitimate e-mails from the sender before, whether the pictures or attachments in e-mails are the same, etc. Incidentally, some viruses that spread by e-mails often has attachments or links of the same type, so they are likely to be fenced out by the filter similarly. The fuzzy Metric value and distributed Bayesian algorithm will have many advantages. First, new users needn’t to train

the filter before use, since the statistics information got from the other computers and servers is enough for a good start. Second, the network will be very sensitive to any new-style spam that goes beyond the statistics of Bayesian model, since their sharply increasing fuzzy Metric value will alert all clients. Third, the whole system will be evolving all the time, studying on every new e-mail. Finally, the scheme can prevent the filters to be "false positives", since it will never regard an e-mail with low Metric value as a spam, even though it is full of *bad* words such as "Guaranteed" and "Free", etc.

Other issues related to grid technology:

1. E-COP relies on grid technology. So total failure like a standalone system will not occur.

2. The spam information is shared among all E-COP servers. The work done per E-COP server is $^1/_N$ where, N is the total number of E-COP server.

3. The E-COP architecture includes latest connection-rate control mechanisms to block incoming spam in any of the distributed servers. The availability of updated information is easier.

4. The infrastructure of E-COP is flexible since further controllers and E-COP servers can be added with an increase of subscribers or users.

VI. CONCLUSION

No perfect spam control solution has been found so far. Filtering approaches are compatible with a broad range of E-mail uses and infrastructure but no filter perfectly identifies even a fraction of unwanted E-mails without eliminating at least some wanted E-mails. Furthermore, the more widely a filter is used the greater the incentive becomes for the spam senders to test against it to ensure that their spam gets through. In this research work, we have tried to present the design issues of E-COP. From analysis, it should be more effective than traditional spam filtering techniques.

However, there is a scope of improving E-COP through exploring issues such as synchronization of the anti-spam servers. In practice, this may not be the case since there may be a lot of factors involved such as traffic, bandwidth constraints, etc. The servers may not be fully coherent but near coherent.

REFERENCES

[1] Jon Panker, Spam is a pricey pest, http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci902228,00.html

[2] www.spamwall.hiwaay.net/about.html

[3] Sahami, M. Learning limited dependence Bayesian classifier. In: Proceeding of the Second International Conference on knowledge Discovery and Data Mining, 335-338, 1996

[4] Feng Zhou, Li Zhuang, Ben Y. Zhao, Ling Huang, Anthony D. Joseph and John Kubiatowics, Approximate Object Location and Spam Filtering on Peer-to-peer Systems, In: Proceeding of ACM/IFIP/USENIX InternationalMiddleware Conference, 2003

[5] Ian Foster, Carl Kesselman, ed. The Grid 2: Blueprint for a New Computing Infrastructure, 2nd edition, Morgan Kaufmann, Nov. 2003, ISBN: 1558609334

[6] www.birds-eye.net/article_archive/grid_computing_toolkits_ides.htm

[7] www.newscientist.com/channel/info-tech/mg18524856.900

[8] www.hyperwrite.com/features/httpr.htm

[9] www.rfc3820.x42.com

[10] The Spamhaus Project, http://www.spamhaus.org

[11] Different Methods of Stopping Spam, http://www.secinf.net/anti_spam/Stopping_Spam.html

[12] Brightmail Probe Network, http://www.brightmail.com