A Collaborative Anti-Spam E-mail Filter

Wei-Li Huang Ying-Sung Lee Wen-Nung Tsai
Department of Computer Science and Information Engineering,
National Chiao-Tung University
{weili, leeyt, tsaiwn}@csie.nctu.edu.tw

ABSTRACT

Email has become one of the most important applications on the Internet. In recent years, system administrators and Internet users tried their best to solve the email spam problem. The problem is that spammers have the ability to send huge amount of junk emails with a low cost and derive a large reward from spamming.

In this paper, we design and implement a collaborative anti-spam e-mail filter (CASEF) system to filter the spam. It acts as an email gateway to filter inbound and outbound messages by enforcing an organization's email policies. Our proposed method exploits the fact that spamming distributes lots of identical spam mails at the same time. CASEF also provides a web-based administrative interface for performing system configuration tasks and for setting up filtering policies. Our system not only blocks the recurring spam mails at the gateway but also achieves higher accuracy than SpamAssassin.

1: Introductions

Today, email has become one of the most important applications on the Internet. Because the email system is free of charge and convenient to use, everyone constantly chooses email as the means for communication. Even though Instant Message and Voice over IP are becoming popular these days, emails do not lose their prevalence around the world. However, despite emails' usefulness, junk mails have become an annoying problem that system administrators and Internet users are desperately trying to solve in recent years.

1.1: Motivation

Every day, millions have trouble finding their needed emails due to unsolicited commercial emails. Companies, governments, non-profit organizations, and individuals receive this type of spam mails, causing them to lose time and to waste network bandwidth. Consequently, many anti-spam filtering systems exist to deal with these problems [1].

Spams are defined by some to be unwanted junk mails sent in bulk from strangers to large mailing lists, usually with some commercial objectives. The problem is that spammers can send junk emails with a very low cost and gain a large reward from spamming. More than two-thirds of the emails in the world are considered spams. MessageLabs [8] finds spams by scanning emails using its global network of control towers. In 2005, 70% of spams in global emails are scanned by Messagelabs [9]. Therefore, the challenge here is not only to save the resources of the network but also to protect the end users.

1.2: Objective

Our goal is to design an anti-spam email filter with the following features. First of all, we would like our system to be efficient. Secondly, we would like it to have a high accuracy at detecting spam mails with low false positives. Furthermore, the system must be flexible enough to allow system administrators to integrate the system to their unique email environment.

This paper is organized as follows: Section 1 introduces the motivation behind the email filtering system. Related works are investigated in section 2. Section 3 mainly describes the implementation of our CASEF system. Then, we will evaluate our CASEF system and compare it with some similar products in section 4. Finally, we provide some discussions and give concluding remarks in section 5.

2: Related Works

In this section, we study several related works on anti-spam filtering techniques and products. Spam is a continually growing problem in the world and many solutions have been proposed. In section 2.1, we will discuss the concept of collaborative anti-spam. In section 2.2, we investigate router-level techniques for spam detection. In section 2.3, we describe SpamAssassin briefly.

2.1: Collaborative Spam Filter

Since spams come in countless varieties and constitute a sizeable volume, spam filters can no longer situate themselves on top of massive repositories in central servers but must base their operations on dynamic knowledge bases located on local servers.

Alan Gray and Mads Haahr have illustrated a collaborative spam filtering in [12] at the Distributed

System Group, Department of Computer Science in Trinity College Dublin, Ireland. Their Collaborative Anti-Spam System Allowing Node-Decentralized Research Algorithms (CASEFANDRA) architecture permits the construction of personalized, collaborative spam filters. Such filters deliver the most relevant spam notices to each user based on the spams identified and reported by network members.

For a personalized, collaborative filter to work, once a new spam is classified, its signature must be computed and transmitted to relevant users that are most likely to receive a similar mail and to also consider it as a spam.

Personalized, collaborative spam filtering has the benefit of being able to track concept drift in spam, and at the same time, minimizing the working set of spam stored by any one user. Similar users will group together on the adaptive P2P network.

Using the collaborative method, emails will not go through just one filter, but will go through many filtration stations. This prevents errors caused by merely focusing on one flaw-reorganization.

2.2: Controlling Spam Emails at the Routers

If we want to control spam mails, the best defense is to work directly at the recipient level. In 2005, Banit Agrawal, Nintin Kumar, and Mart Molle, three professors from the University of California, described a mechanism for detecting and controlling spam mails at the router level in their paper [13].

When the router sees a SMTP's message, the message will be copied and redirected to another system that does the spam classification. The router can also set a flag normally reserved in TCP header on the packet containing spam's header. This flag is then checked by recipient's MUA when he/she decides to read the mails.

Their system implemented a two phase method for checking spams at the router level. In the first phase, they used a pattern-matching approach to detect spam mails. If the mail is considered a spam in the first phase, it will not go to the second phase. The second phase uses a Bayesian classifier to categorize the remaining mails. If either two phases concludes that the mail is spam, the system will rate limit it at the router level.

We will borrow the router-level concept in our CASEF's design and also incorporate multi-level caches to improve our system's performance.

2.3: SpamAssassin

SpamAssassin [3] is a software package that is designed to detect email messages, to categorize emails into spams or non-spams, and to report the result back to administrators or users alike. The system is rule-based with each rule describing a relationship among words of an email message that assigns a point to be added or deducted from the overall score. If an email message has a score exceeding the preset threshold, the message is reported as spam.

SpamAssassin has become popular despite the numerous available anti-spam systems for several reasons:

1. SpamAssassin derives its spam filtering ability from many different kinds of rules.
2. Tuning the scores associated with each rule or adding new rules based on regular expressions is easy.
3. SpamAssassin can adapt to each system's unique email environment.
4. SpamAssassin can report spams to several spam clearing houses.
5. SpamAssassin is free.

The SpamAssassin system core is composed of a collection of modules written in the Perl programming language. After SpamAssassin decides that an email is spam, that email is neither deleted nor filtered out. On the contrary, the email is marked with a tag on its subject line and is delivered to the recipient.

SpamAssassin is quite robust and has been used in Unix world for many years. Thus, the system is chosen to be the foundation on which we build our anti-spam system.

3: Our CASEF System

CASEF (Collaborative Anti-Spam Email Filtering) is a system designed to filter spam emails at the gateway. The system contains several components such as email policy filtering module, digest matching module and SpamAssassin filtering module. We use several methods to defeat spamming collaboratively. In section 3.1, we give an overview of the CASEF system. In section 3.2, we describe the system architecture and give details on all the modules.

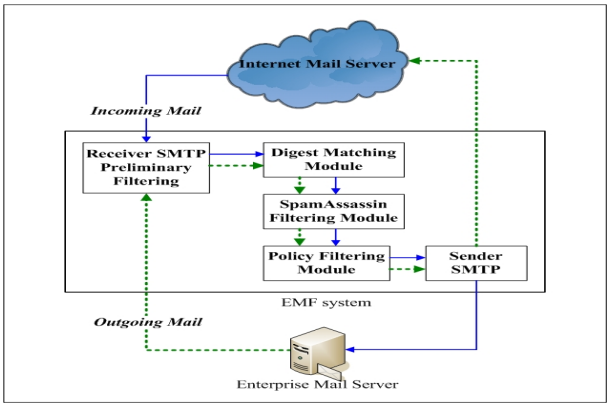

Figure 1 CASEF System overview

3.1: System Overview

Figure 1 gives an overview of the CASEF system. As shown in the figure, incoming and outgoing mails will be sent to the Receiver-SMTP. The Receiver-SMTP will then forward all the mails to our filtering modules. The junk mails will first be detected and sorted out, then

the normal mails are transferred to the policy filtering module to enforce the email policies. Finally, all the remaining mails will be sent to the Sender-SMTP to complete the regular mailing action.

3.2: System Architecture

CASEF system is designed to act as an MTA (Mail Transfer Agent) on a Linux system. CASEF receives emails for your organization, checks them against filtering rules, and then relays the emails to your organization's mail server.

There are four filtering modules in our CASEF system: preliminary filtering, digest matching, SpamAssassian filtering and policy filtering. Figure 2 shows the system architecture of our CASEF system.

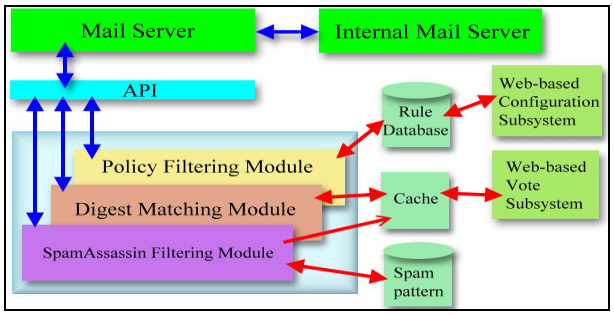

Figure 2 CASEF System Architecture

3.2.1: Receiver-SMTP and Preliminary Filtering. In order to comprehensively control the email filtering procedures, we implement a Receiver-SMTP server to deal with SMTP dialog. And we use ***sendmail*** to serve as a Sender-SMTP to deal with mail forwarding.

Our Receiver-SMTP listens on TCP port 925 so we can use it in conjunction with ***sendmail***. It implements the SMTP command set, including HELO/EHLO, AUTH, MAIL, RCPT, DATA, RSET, and QUIT.

We used the ***iptables*** to redirect SMTP connection to our SMTP-receiver. The ***iptables*** is an IP packet filtering facility that comes with the 2.4.x versions of the Linux kernel. This is done by using the following command.

```
iptables -t nat -D PREROUTING -p tcp --dport
25 -j REDIRECT --to-ports 925
```

The Receiver-SMTP first does preliminary filtering against the information provided in the SMTP dialog, which includes the initial protocol greeting and mail transactions. The preliminary filtering includes validation of the sender IP address, the HELO/EHLO parameters, and the envelope sender. We will briefly explain these tasks below.

Validating the IP address

One of the first things CASEF system filters on is incoming connections. The system will check the address of incoming connection and decides whether to accept it or not. The check items are listed as follows:

- Trusted IP address lists
- Reverse DNS record
- DNS Real-time Black-hole List (RBL)
- Blacklist

Validating the EHLO/HELO parameter

The SMTP HELO and EHLO commands provide one of the first pieces of information available in an SMTP dialog. We verify the EHLO/HELO domain to see if the sending mail server domain exists in DNS. If it does not exist in DNS, CASEF system issues a 554 error reply to the sending machine.

Validating the envelope sender

For SMTP MAIL command, we want to know if the envelope sender is permitted to send email to your mail server. The check items are:

- Trusted List (TL)
- Fake Local Name (FLN)
- Blacklist
- Email Address Validity

After SMTP transaction, the email is received as a MIME-encoded message. Before doing the other filtering, CASEF system must perform message parsing.

Parsing a MIME-encoded email message can be very difficult due to the number of options and different ways of encoding. However, Perl provides a wonderful class (MIME-Tools) that has the ability to understand MIME encapsulation and to return a nice hierarchy of objects representing the message. Our CASEF system thus takes advantage of this Perl module to parse mail messages.

3.2.2: SpamAssassin Filting Module. SpamAssassin filtering module uses SpamAssassin anti-spam engine to detect spam mails. After the mails progress through the digest matching module, they will go through a series of tests conducted by SpamAssassin filtering module.

Event	Action
Spam	The module classifies the mail as spam and gives a tag on the subject.
	The module sends the mail to the internal mail server.
	The module copies the digest of the mail to the First cache.
Non-Spam	The module passes the mail and sends it to the internal mail server.
	The module copies the digest of the mail to the Second cache.

Table 1 Events of Filtering Module

SpamAssassin anti-spam engine scans the entire mail stream and computes scores according to the rules. If the score exceeds the limit set forth by the administrator, the mail's subject field will be tagged. Section 2.3 describes SpamAssassin in detail and we refer interested readers there. In addition to subject tagging, more works are still waiting to be performed. Table 1 shows the events that SpamAssassin filtering module will face and the actions the module will perform for each event.

The anti-spam module in our CASEF system can also defuse other mail bombs effectively. Mail bomb can quickly overrun a mail server, and even completely disable it. Before defusing a mail bomb, we have to know what type of bomb has hit and where: inbound or outbound email. In CASEF anti-spam module, we try to stop email bombs at the SMTP dialog phase.

Dictionary Harvest Attack, also known as DHA, is a technique exploited by spammers to flood mail servers by sending hundreds or thousands of messages to random addresses, hoping that some of these addresses are valid. This attack can slow down email systems to the point that companies have to increase spending on extra server space and bandwidth. Standard approaches to spam filtering or IP address blocking are useless against DHAs. Our CASEF detects the DHA attack at the SMTP layer in the gateway; thus, it can effectively prevent a DHA before any of the bombing traffic affects the real mail-server.

3.2.3: Digest Matching Module. Digest matching module has two caches holding digests of mails released from SpamAssassin filtering module. The first cache implements the LRU policy and stores the digests of mails flagged as spam. The second cache replaces its content in FIFO order and stores the digests of mails judged to be normal.

When preliminary filtering forwards the mails to the digest matching module, the digests of the mails are computed and matched against the digests stored in the two caches. In our CASEF system, Nilsimsa digest technique is chosen to be the digest algorithm. Nilsimsa is an open source digest-based technique for spam detection [14]. A Nilsimsa digest is 32-byte long. The digest computed using MD5 or SHA1 has the property that a small change in the content will produce a large difference in the digest. Contrarily, Nilsimsa digests will differ only by about 0%~10% when the two respective contents have a 0%~ 10% difference in their file size. Due to this nice property, Nilsimsa is superior for spam detection even if spam messages have little differences.

When the computed digest reaches the digest matching module, the CASEF system will compare the digest computed with the digests stored in the first and the second cache in this order. The digest matching scenario is given in Table 2.

Event	Action
Hit the First	The module classifies the mail as spam and gives a tag on the subject.
cache	The module sends the mail to the internal mail server.
	The g module updates the digest that is hit in the First cache in LRU order.
Hit the Second cache	The module sends it to the spamassassin filtering module.
	The module moves the digest that is hit in the Second cache to the Third cache.
No Hit	The module forwards it to the spamassassin filtering module.

Table 2 Events of Digest Matching Module

3.2.4: Policy Filtering Module. Policy filtering module enables all users to filter incoming and outgoing emails according to the nature of email contents and/or email headers, as well as the filenames of e-mail attachments. System administrators can configure policy rules according to keywords in the mail header or the mail body. Policy filtering module will enforce these policy rules via one of the following actions specified by the administrator:

- Quarantine the suspect mail
- Delay delivery of the mail
- Forward the mail blindly
- Remove the attachment

Professional users can edit the configuration file directly in the Unix system. Novice system administrators, on the other hand, can utilize the web-based interface in configuring the policy file according to users' filtering policy.

3.2.5: Other Utility Modules.

Mail Archiving Module

Besides the modules mentioned above, the CASEF system also includes an archiving module to archive specified emails for later examination. MySQL is chosen as the database system for mail storage. In case of a confidentiality breach, a company can refer to the archive as a way to trace back the source of the leak.

Mail Statistics Module

The CASEF system also provides some statistical reports to facilitate management through the Statistics Module. These reports reveal each user's email activities, and the flow rate of spam mails, etc.

System Maintenance Module

A user-friendly web-based interface is provided for system administrators to configure and/or maintain the CASEF system. System administrators are able to perform the following administrative tasks through this module: mail routing configuration, database maintenance, alerting method configuration, filtering policy set up, and so on.

4: Experimental Result

When email goes through the CASEF system, our SMTP-Receiver does the receiving. In this section, our experiments will show the increased overhead of using our method with respect to using merely ***sendmail*** and SpamAssassin. The difference of the filtering capabilities between the two methods is also shown in the results.

4.1: Experimental Environment

We cannot simulate the various behaviors of spammers by merely running the simulation in a lab. Thus, to be more realistic, we test our system on real networks. We install the CASEF system on the mail server of a university (having around 20 thousand email accounts) and also of a medium enterprise (having 683 email accounts). The mail servers is based on PC compatible machines with a Intel Xeon™ CPU at 3.06 GHz, a 4 GB memory and a SCI 60 GB hard disk. The configuration is changed to fit the organization's environment. We then collect one month worth of statistical data to do further analysis. The network topology of the university is shown in **Figure 3**.

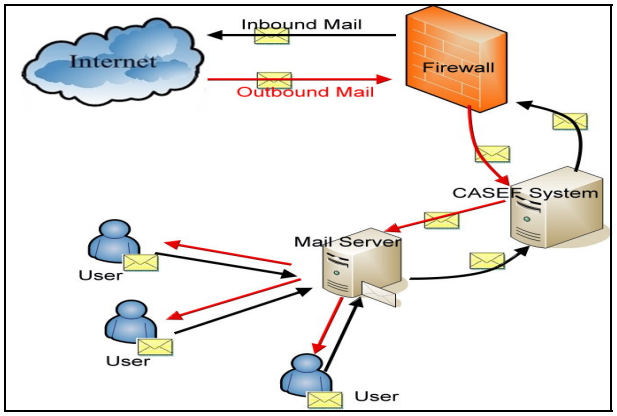

Figure 3 Network environment in a university

In order to determine the overhead incurred by the CASEF system, we collect statistics from real network environment. The same hardware specification of the university we base our experiments on previously is employed again in this test. This experiment starts first without installing the CASEF system on the mail server. After the CASEF system is installed, the experiment is conducted again. The results are then compared to compute the additional overhead incurred by the CASEF system.

4.2: System Performance and Overhead

As mentioned in the previous section, we have installed the CASEF system on the mail system of an university and collected about one month worth of data.

Figure 4 below shows the Mailing Statistical Report from July 4th to July 28th the CASEF in that university has produced.

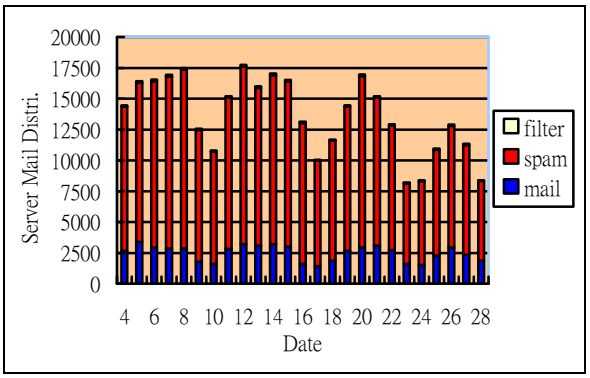

Figure 4 A CASEF statistical report

From this report, we see that 2759 spam mails out of 307992 are falsely accepted as legitimate mails. This shows that the CASEF system has a false positive rate of 0.9% (2795/307922). The other spam mails are successfully detected and processed according to the policies or filtering rules. This report also shows that the CASEF system has a spam-blocking rate of around 78% ((307992 - 2759) / 390610).

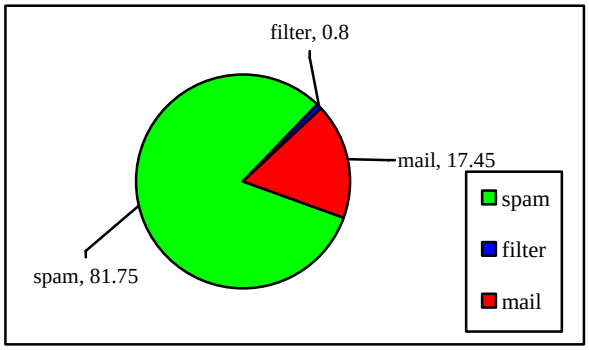

Figure 5 A CASEF report in a university

As shown in **Figure 5** above, the CASEF system, by using SpamAssassin engine along with white/black lists, and the Collaborative voting subsystem, successfully protects multilingual message streams, safely removing up to 76.21% of spam at the gateway.

The following table shows additional functions implemented in our CASEF system relative to SpamAssassin.

	SpamAssassin	CASEF system
Detect fake routing path	No	Yes
Detect DHA attacks	No	Yes
False positive	Medium	Low
Spam blocking rate	Good	Better than SpamAssassin
Flexibility	Medium	High
Defend the repeated spam mails	No	Yes

Detail reports	Poor	good

Table 3 CASEF vs SpamAssassin

Installing CASEF consequently increases the system overhead. To find out how much overhead is expended by CASEF, we use a mail generator to generate 10,000 random-length mails that include 75% spam mails. We measure the time that is required to completely receive the 10,000 mails, and compare the timing of both before and after installing the CASEF system. We repeat this scenario 10 times with different random seeds, and calculate the average. The experimental result indicates that the overhead is around 8.95%, which is at an acceptable range.

5: Conclusion and Future Work

We have designed a CASEF system that integrate the collaborative spam filtering technique with the SpamAssassin anti-spam engine, and have chosen the Perl language for implementation.

We reduce the waste on network resources by blocking spams at the gateway or Mail Transfer Agent (MTA). These resources include Internet bandwidth, mail server processing cycles, and storage capacity. As shown in Figure 5, more than 80% of the emails are useless. Thus, users can eliminate a lot of wasted time by not reading garbage mails.

In addition, corporate communication policies are easily managed using CASEF system's flexible web-based policy manager. System administrators thus gain complete and precise control over mail filtering.

No single technology can consistently eliminate spam in the long run. Providing multiple defenses is the best way for approaching a complete spam protection solution. Some additional protections can include:

- Local mailbox existence check
- Checking blank email, black/white list, and fake routing
- Defuse DHA
- Detecting spam more precisely
- Defend the repeated spam mails

Our CASEF system is based on the anti-spam filtering system, SpamAssassin, and improves on its spam detection accuracy. However, there are still rooms for improvement. We list some works that can be done in the future below.

1. We will create several dummy email accounts to trap spammers. The spams induced by these dummy accounts enable us to analyze the patterns that are characteristic to spams and help us increase the accuracy of their detection.
2. Some spammers may find a way to create spams that confuse our CASEF system. We can hire some mail janitors to manually classify such spam mails.
3. We can share the digests of the two caches with other mail filtering systems.

REFERENCES

[1] Wei-Li Huang, J.Y. Huang, Wen-Nung Tsai, "Design and Implementation of E-mail Filtering System," National Computer Symposium (NCS 2005), Tainan, Taiwan, Dec. 15-16, 2005.

[2] Paul Schmehl, "Barbarians at the Gateway: Defeating Viruses in EDU", in Proceedings of the 29th annual ACM SIGUCCS conference on User services, Pages 177 - 180 , Portland, Oregon, USA, 2001.

[3] The Apache SpamAssassin Project, http://spamassassin.apache.org/

[4] Anti-Spam Research Group (ASRG) of the Internet Research Task Force (IRTF), http://asrg.sp.am/

[5] Gillmor, D., "Data Privacy Protection Must Start with IT", Computerworld, Vol. 32, No. 45, November 9, 1998.

[6] Hartman L.P., "The Rights and Wrongs of Workplace Snooping", Journal of Business Strategy, Vol. 19, No. 3, May/June 1998, 16-19.

[7] Miller-Seumas and John Weckert, "Privacy, the workplace and the Internet", Journal of Business Ethics, Dec 2000, Vol.8, No.3, pp.255-265.

[8] MessageLabs, http://www.messagelabs.com.

[9] MessageLabs Intelligence July 2005 Monthly report, http://www.messagelabs.com/intelligence.

[10] Email Bombing and Spamming, http://www.cert.org/tech_tips/email_bombing_spamming.html

[11] Tom Merritt, "What is Email Spoofing?", May 09, 2000. http://www.techtv.com/screensavers/answerstips/story/0,24330,2566233,00.html

[12] Alan Gray and Mads Haahr, "Personalised, Collaborative Spam Filtering", Technical Report TCD-CS-2004-36, Trinity College Dublin, Ireland, 2004.

[13] Banit Agrawal, Nitin Kumar, and Mart Molle, "Controlling Spam E-mail at the Routers," In IEEE International Conference on Communications (ICC 05), Seoul Korea, 2005.

[14] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati, "An open digest-based technique for spam detection," In Proceedings of the 4th IEEE international conference on peer-to-peer computing, 2004.