Design and Implementation of E-mail Filtering System

電子郵件過濾系統之設計與實作

Wei-Li Huang(黃威力)　J.Y. Huang(黃貞云)　Wen-Nung Tsai(蔡文能)
Department of Computer Science and Information Engineering,
National Chiao-Tung University
weili@csie.nctu.edu.tw (交通大學資訊工程研究所)
athena@hgiga.com (桓基科技股份有限公司產品經理)
tsaiwn@csie.nctu.edu.tw (交通大學資訊工程學系)

Abstract

Nowadays, email is growing on communications between corporations, educational institutions, government agencies, and all kinds of organizations. Due to this, the importance of their email systems is enhancing within their information infrastructure. However, the significant growth of spam, viruses and other types of email-borne attacks in the past year is causing the system managers to face more challenges in managing and protecting this critical communications asset.

In this paper, we designed and implemented an intelligent e-mail filter, EMF (Electronic Mail Filtering system), which is a general purpose Internet mail filtering tool. It acts as an email gateway to filter inbound and outbound messages by enforcing an organization's email policies. It can be used to block junk mail, prohibit mail relaying, and diffuse mail bomb attacks.

In the EMF system, we also provided a web-based administrative interface for the system administrators to do the system configuration and to set up their filtering policies. The EMF system can also give detail statistics and reports that will give a great help to the administrators for the analysis task.

Keywords: spam, gateway.

摘要

隨著網際網路的發達，電子郵件(Electronic mail, E-mail)也逐漸成為網際網路上重量級的應用軟體，越來越多人利用電子郵件做為主要溝通和傳遞訊息的工具。因此，電子郵件除了成為駭客散播病毒的主要管道之外，從 1978 年開始出現垃圾郵件之後，垃圾郵件數量的快速成長，對於電子郵件的使用者也已經造成莫大的困擾。

電子郵件過濾系統是一套架設於電子郵件閘道口的郵件安全系統，整合防毒引擎、垃圾郵件阻擋工具以及多樣化的反垃圾郵件技術，可以偵測並過濾郵件病毒與垃圾郵件。

電子郵件過濾系統提供友善的 web-based 管理介面，方便管理者進行系統管理，並提供完善的郵件紀錄與統計報表，協助管理者追蹤分析病毒郵件與垃圾郵件。

關鍵詞：垃圾信件，閘道器。

1. Introduction

With the growth of the Internet, email, a natural use of networked communication technology, is developing right along with this evolution. This causes the importance of organizations' email systems to enhance within their information infrastructure. Yet, researches throughout the pass few years have shown significant increase of all types of email-borne attacks, such as spam and viruses. Due to this, system managers now find it more challenging to manage and protect this critical communications asset.

Email is now the most common way that computer viruses are transmitted between computers. The most common mechanism used by most viruses is in the form of an attachment to the message. The attachment facility is normally used for emailing documents, images,

and so on. However, it is also possible for attachments to contain programs which will run automatically when the users try to open the attachments.

MessageLabs [9], the world's leading provider of services for managing email security, released its Labs Intelligence monthly report in July 2005 [10]. This report has highlighted on some disturbing trends in email security threats. MessageLabs found that the ratio of viruses in email was 1 in 32.13, a decrease of 0.44% over June 2005.

The amount of "unwanted mail" which shows up in one's email box has obviously increased during the pass few years. More of us are suffering the annoyance caused by spam. However, spam is more than merely a nuisance. It is also a drain on the economy due to numerous spam problems. This includes lost productivity from the users who must deal with spam messages, and from the computing resources that must be used to handle these messages. The computing resources include additional bandwidth, storage, and e-mail servers.

MessageLabs finds spam by scanning email from its global network of control towers. In July, it found 1 spam in every 1.53 emails (65.16%) from the 500,000 emails per hour to over 1 million. This represents a decrease of 2.09% over May's figures, which peaked at 1 in every 1.48 email (67.25%). According to MessageLabs, the global ratio of spam represents a three-month average of around 67%.

1.1 Objective

In this paper, we designed and implemented an intelligent e-mail filter, EMF (Electronic Mail Filtering system), which is a general purpose Internet mail filtering tool. It acts as an email gateway to filter inbound and outbound messages by enforcing email policies proposed by a company or any organization. It can be used to block junk mail, to prohibit mail relaying, and to diffuse mail bomb attacks. The EMF system works by accepting mail on behalf of your regular mail server. It automatically forwards acceptable mail to your mail server for regular delivery, and flags unacceptable mail with a tag. You may have the EMF system quarantine this flagged mail outright, or redirect it to an administrator for review, or simply mark the subject line.

The key features of our EMF system are as follows:

- Virus Prevention: Utilizing the award-winning anti-virus engine, Sophos Anti-Virus Engine, the EMF system scans all mail at the gateway to protect the entire organization from email viruses, Trojans, and worms.
- Spam Filtering: EMF systems fight spam with numerous techniques, including connection filtering technique and content filtering technique.
- Policy Management: EMF systems' flexible policy framework enforces inbound and outbound message filtering policies to meet an organization's security, communication and regulatory compliance needs.
- Central Administration: A web-based administrative interface provides flexible control over mail filtering. It enables the configuration and management of filtering policies, the management of quarantined messages, statistics and reports.

This paper is organized as follows: Section 1 introduces the motivation of designing the email filtering system. More related works are investigated in Section 2. Section 3 mainly describes the implementation of our email filtering system. Then, we will present the evaluation of our EMF system and compare the system with some similar products in Section 4. Finally, some discussions and conclusions will be given in Section 5.

2. Related Works

In this section, we investigate the related works of anti-virus and anti-spam. In fact, there are already various products of anti-virus; the most popular ones include the anti-virus products of Symentatec and TrendMicro, and the Anti-Virus Engine of Sophos. In section 2.1, we review the techniques that are used to defeat viruses at the gateway. In section 2.2, we introduce the Anti-Virus Engine from Sophos. Section 2.3 gives a brief discussion regarding the SpamAssassin.

2.1. Defeating Viruses at the Gateway

The IDC Research estimated that over 450 new viruses are discovered each month. Gartner Group estimates that more than 80 percent of computer viruses enter the network through email. The early anti-virus products mostly execute on the personal computer of the client. Once we find computer virus embedded in the program to be executed, or in the programs that are attached to emails, the mail system will carry out an action according to the configured rule the user has previously set, such as delete or quarantine. Currently, email with virus is obviously increasing, and we want to find ways to stop these malicious programs before they

infect the users' computer.

Paul Schemel, Supervisor of Support Services in the Technical Customer Support Services Department (TCS), has presented a paper at the Special Interest Group, University and College Computing Services (SIGUCCS) Conference in Portland, Oregon, from October 17-20, 2001. The paper, "Barbarians at the Gateway, Defeating Viruses in EDU" [1], includes a presentation of email server blocking techniques. UTD has already used these techniques to successfully prevent many viruses from entering the network. TCS staff has also employed the techniques to keep viruses out of our network.

Paul Schemel states that viruses are a security problem. Thus, we should implement solutions of normal security problems to solve the virus problem. These include, but are not limited to, establishing written policies to address common security issues, defining appropriate behavior and best practices and publishing them, devising both detection and defense in depth1 strategies, and clearly defining problem identification and cleanup methodologies.

Paul Schemel points out that it is possible to keep a LAN relatively virus free if we construct a fitting anti-virus plan. Universities, any enterprises or organizations must have policies and procedures that establish a unified approach to solving the virus problem. Then it takes a combination of desktop protection, user education, constant OS patching, defense in depth and innovative approaches to implement those policies and make them effective.

One key to a successful security structure is "defense in depth". This means that network administrators do not rely upon one method of protection against a threat. Network administrators should devise multiple layers of protection, so that if one layer fails, another will still protect the campus.

For practice of "defense in depth", we design the EMF system which is an anti-virus and anti-spam protection at your network gateway to remove virus from e-mail before they have a chance to penetrate your network.

2.2. Sophos Anti-Virus Engine

Anti-virus protection is a critical component in protecting a network from email-related threats. An anti-virus engine is required for a mail filtering system to detect and/or remove virus attached to the mail. And a good, updated virus-pattern database is also required for the anti-virus engine to perform accurate scanning task. To maintain an updated virus-pattern database is not an easy task and thus we decided to choose an anti-virus engine available in the Internet.

There are many anti-virus engines available on the Internet. These include McAfee Anti-Virus Engine, Sophos Anti-Virus Engine, Computer Associates Vet Anti-virus engine, Norman Virus Control, GossamerHost Anti Virus Scanner, Aladdin Kaspersky Anti-Virus Engine, DrWeb Anti-Virus Engine, Authentium CipherTrust Anti-Virus engine, etc. Sophos is the provider of anti-virus software to over 60% of the FT-100 companies. The Sophos Anti-Virus Engine [4] is a virus protection utility designed for small and medium sized networks. The utility can be directly linked to the MailServer core for extended viral detection and elimination.

One of the problems of maintaining an anti-virus engine is that you need to keep an up-to-date virus signature files. According to Sophos, virus signatures are kept up-to-date and are delivered directly from Sophos's worldwide research labs as part of the automatic updating process. Organizations can choose to update either internally from the organization's intranet, or directly from Sophos. Meanwhile, its products are sold and supported in more than 150 countries. We chose the Sophos Anti-Virus Engine to perform the anti-virus task in our EMF system.

Many universities and corporations provide their users with the comfort of Web-based email. However, web mail uses only the HTTP protocol when sending an email from one internal user to another. In this case, if a user sends email with an infected attachment to another user utilizing a Web mail tool, the message will be delivered without being scanned since traditional anti-virus engines only monitor the POP3 and SMTP ports. We tried to solve this problem by linking Sophos Anti-Virus Engine to our EMF system. In order to integrate the anti-virus engine into our EMF which implemented in PERL, SAVI-Perl is chosen to be the API (APplication Interface). SAVI-Perl is a Perl module interface to the Sophos Anti-Virus Engine.

2.3. SpamAssassin

Spam is the popular term for junk email, also known more formally as unsolicited bulk mail. Unfortunately, it's not as easy to spot and throw away as the junk ads you get in the mail every day at home. Spam mail has grown rapidly within the past few years. Five years ago, a percentage of 10% out of the received mail is spam; now the percentage has vastly increased up to around 85%. There are many anti-spam products, either a commercial one or an open source. SpamAssassin [2] is a popular

open-source software package which can be used to detect spam mail. It is a PERL based spam filter program that utilizes a series of rules to flag mail as Spam. It can be run on the e-mail server and analysis your e-mail message to see if any of them may be SPAM.

SpamAssassin scans the e-mail message looking for key phrases that can be found in most spam messages. Examples are phrases containing: AMAZING or FREE in all capital letters; lots of money; enlarge your penis; SEXY GIRLS; claims you can be removed from the list; claims NOT to be spam; and hundreds of other phrases. SpamAssassin applies a variety of textual and other tests to messages in order to estimate the likelihood that they are spam. This likelihood is represented as a number, the spam score. The spam score is assigned to each message it scans, which can subsequently be used to determine the message's disposition. The spam score assigned to any message is not a certain judgement but is instead an estimate of the likelihood that it's spam. The higher the score, the more likely it is that it's spam, and the lower the score the less likely it is. But it's quite possible for a nonspam to score highly and for a spam to score lowly.

SpamAssassin does NOT filter out or delete any email. It only flags mail that it thinks is spam. The mail will still be delivered to you. You may choose to set up a filter to move all your probable spam mail into a folder, and then go through them when you have time. A quick way to deal the spam mail is to sort them by the sender, and then do a scan for familiar names before trashing the lot.

According to SpamAssassin's documentation, in its most recent test, SpamAssassin differentiated between spam and non-spam mail correctly in 99.94% of cases. SpamAssassin is one of the best anti-spam solutions but it may sometimes label legitimate email as spam (false positive).

SpamAssassin is written in PERL, and can be used as a Spam detection engine. It is quite a robust program, having been used in the Unix world for many years. We chose it as our anti-spam engine and integrated it into our EMF system with several improvements.

3. EMF System

Within the past few years, virus and spam are continuously growing at a rapid rate. This results in increasing security threats. The large amount of spam mail and virus attacks floods the network with useless traffic causing Denial of Service (DoS) attacks. Most organizations agree that they need to protect their networks from virus/worms attacks and spam mail threats by installing an email security product. In this paper, we design and implement an Electronic Mail Filtering (EMF) system to provide consolidated protection – not only against spam, but also against viruses.

3.1. System Overview

Figure 1 gives an overview of the EMF system that we designed. As shown in Fig. 1, all incoming mail and outgoing mail will be sent to the Receiver-SMTP. The Receiver-SMTP will then forward all the mails to our filtering sub-systems. It will first detect and sort out the viruses within the mails, transfer the un-virused mail to the spam filter, and after filtering, transfer them to the policy-filtering module to enforce the policies configured by the system administrator. Finally, all the remaining mail will be sent to the Sender-SMTP, which is actually the sendmail program, to complete the regular mailing action. Detail process of each module will be illustrated in the following section.

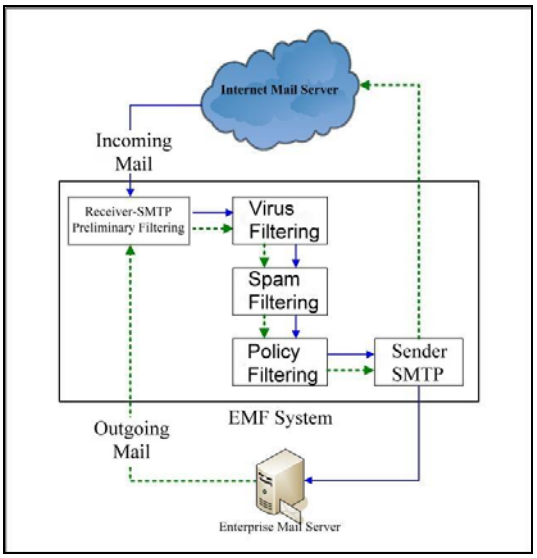

Figure 1 EMF System overview

3.2 System Architecture

EMF system is designed to act as an MTA (Mail Transfer Agent) installed on the Linux system. EMF receives email for your organization, checks to see if it's against filtering rules, and then relays the email to your organization's mail server.

There are four kinds of email filtering in our EMF system include preliminary filtering, virus filtering, spam filtering, and policy filtering. Details about email filtering will be described in following sessions. Figure 2 shows the system architecture of our EMF system. The detail mail process flow is shown in Figure 3.

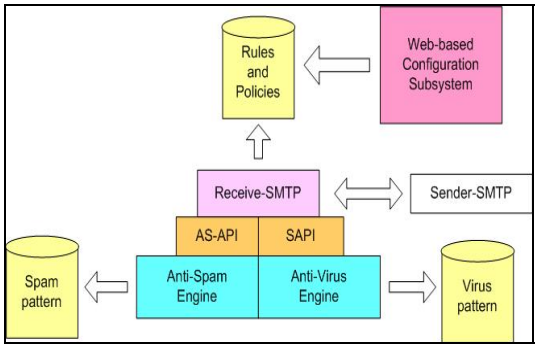

Figure 2 EMF System architecture

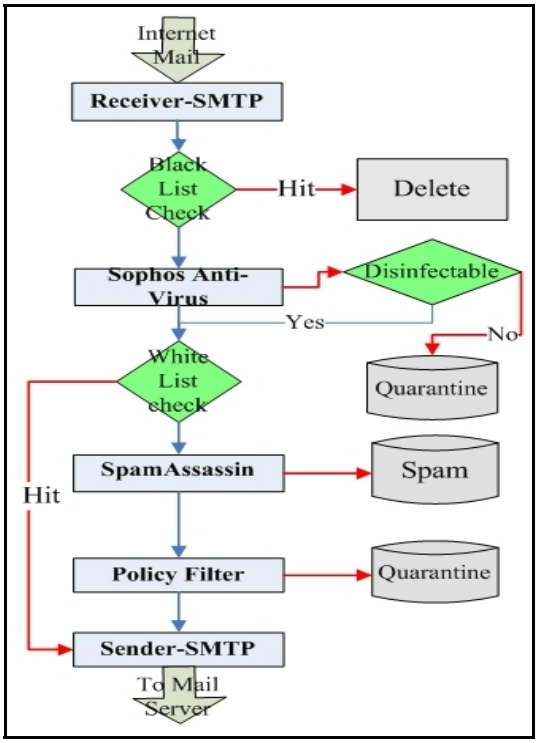

Figure 3 Mail process flow

3.2.1. Receiver-SMTP and Preliminary Filtering

In order to comprehensively control the email filtering procedures, we implement a Receiver-SMTP that acts as an SMTP server to deal with SMTP dialog. And we use sendmail to serve as a Sender-SMTP to deal with mail forwarding. Since the Sender-SMTP, sendmail, is a well-known system program, we will not discuss it here.

Our Receiver-SMTP listens on TCP port 925 so we can use it 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 Receiver-SMTP. The iptables is an IP packet filtering facility that is integrated with the latest 2.4.x versions of the Linux kernel. This work is done by using the following command:

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

First of all, the Receiver-SMTP of our EMF system does preliminary filtering tasks against the information provided in the SMTP dialog. This includes the initial protocol greeting and mail transactions. The preliminary filtering tasks include validating the IP address of the sender, validating the HELO/EHLO parameters, and validating the envelope sender. We will briefly explain these procedures in the follows.

Validating the IP address

One of the first things EMF system filters on is incoming connections, checking the address of incoming connection and deciding whether to accept it. These check items include follows:

- Checking trusted IP address lists

First of all, we check the sender machine's IP address to see if it is listed in your trusted IP address list, so call white lists. If it is listed in white lists, EMF system will skip over all spam check items.

- Checking reverse DNS record

EMF system performs reverse DNS lookup to see if the sender machine has a registered DNS entry. If the sender machine does not have a registered reverse DNS entry, EMF system issues a 554 error to the sending machine, as shown in Figure 4.

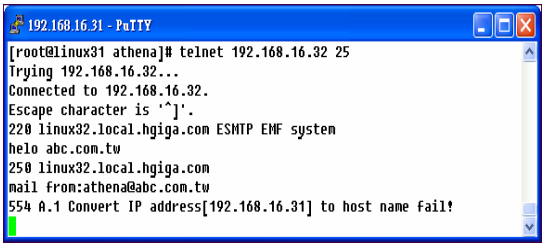

Figure 4 Reverse DNS error reply

- Checking DNS Real-time Black hole List (RBL)

Third step, we validate if the sender machine's IP address is listed in RBL which is a list of IP addresses meeting some criteria such as involvement in Unsolicited Bulk Email. If it is, EMF system issues a 554 error reply to the sending machine, as shown below.

```
554 A.2 Sorry! your IP
address[11.22.123.234] is blocked by
RBL[relays.ordb.org]
```

DNS RBL is an IP-address-listing DNS server. It is an Internet resource that lists IP addresses known to originate spam. Typically these DNS RBLs work by creating a DNS record (for example 11.22.123.234. relays.ordb.org) for each known open relay. It accepts iterative DNS queries from hosts around the Internet asking about various IP addresses. It provides responses showing whether the addresses are on a locally configured list known to originate spam.

- Checking Blacklist

EMF system allows administrators to configure their own blacklist. When EMF system finds the sender machine from the blacklisted IP, it issues a 554 error reply to the sending machine, as shown in Figure 5.

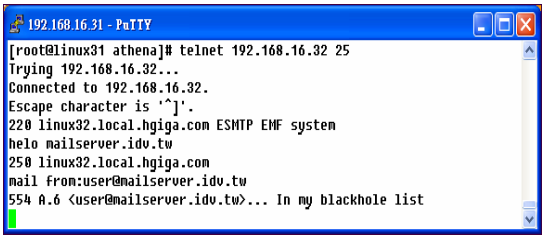

Figure 5 Blacklist error relpy

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 validates that if the sending mail server domain exists in DNS. If the sending mail server domain does not exist in DNS, EMF system issues a 554 error reply to the sending machine, as shown in Figure 6.

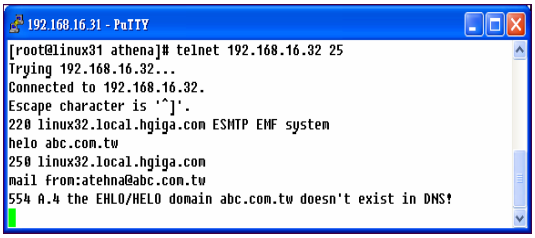

Figure 6 EHLO/HELO error reply

Validating the envelope sender

In SMTP MAIL command we check to see if the envelope sender is permitted to send email to your mail server. There are two check items here.

- Checking Trusted List (TL)

If envelope sender conforms to trusted email or domain, EMF system will skip over all spam check items.

- Checking Fake Local Name (FLN)

When the envelope sender appears to be on the local domain, EMF system checks to see if the sender machine is coming from a relay client or has passed the SMTP authentication. If both are false, the EMF system issues a 554 error reply to the sending machine, as shown in Figure 7.

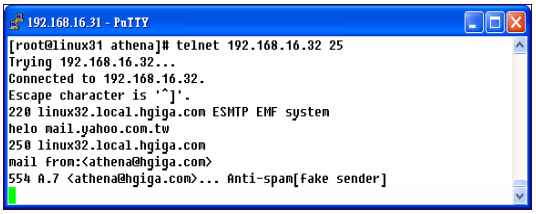

Figure 7 Fake sender reply

- Checking blacklist

If envelope sender conforms to blacklisted email or domain, EMF system issues a 554 error reply to the sending machine, as shown in Figure 8.

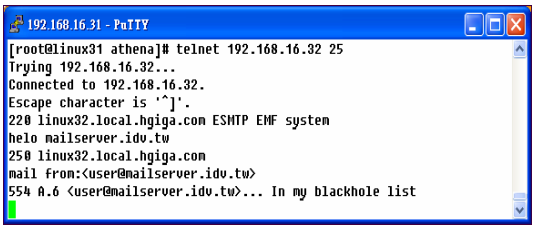

Figure 8 Blacklist error reply

- Checking email addresses for validity

EMF system makes several checks on envelope sender. First, it checks to see if there are any MX records or A records in DNS for the domain part of the email address. Second, it tries to connect to an email server directly via SMTP to check if the mailbox is valid. It uses a combination of MAIL and RCPT commands which simulates sending an email. This can detect bad mailboxes in many cases. If any invalid sender email address is found, EMF system issues a 554 error reply to the sending machine, as below:

> 554 A.5 <user@abc.com.tw>... Sender address does not exist!

After SMTP transaction, we get the mail message, a MIME-encoded message. Before doing the other filtering, EMF system must perform message parsing.

Understanding an email message encoded with MIME can be very difficult. It can get frustrating due to the number of options and different ways to do the actual encoding. Thanks to Perl's extensive bag of tricks, CPAN, it has a wonderful class (MIME-Tools) that provides the ability to understand this MIME encapsulation, returning a nice hierarchy of objects that represent the message. Our EMF system utilizes this PERL module to parse mail messages.

3.2.2. Anti-Virus Module

An effective anti-virus solution for mail system should use an intelligent heuristic for finding viruses within Multipurpose Internet Mail Extensions (MIME) attachments. Most users agree that the most important factor in the successful protection of the network against viruses is how fast you are notified that the new virus signature files when a new virus emerges. Email allows viruses to be spread at lightning speed in a matter of hours, and a single email virus is enough to crash the whole network. To maintain the virus signature database is an intricate task. Therefore, we do not maintain these data in our EMF system. In stead, we used the anti-virus engine provide by a famous company, Sohpos.

Sophos is a leading provider of network security, content security and messaging software. In EMF, we used SAVI-Perl as the API (APlication Interface) to Sophos Anti-Virus Engine. SAVI-Perl is a Perl module interface to the Sophos Anti-Virus Engine. It allows you to scan files for viruses directly from Perl.

Perl is one of the best languages for writing email filters. The reason that we chose Sophos Anti-Virus Engine is that Sophos Anti-Virus Engine adopts a so-called Genotype technology which provides proactive protection from new variants of virus families, even before specific, signature-based protection becomes available. It offers an easily updated, flexible business solution for managing the complexity of networks from small local area networks to large multi-server, multi-platform WAN's.

3.2.3. Anti-Spam Module

Spam or unsolicited commercial email has become a serious network problem due to the large quantities of email taking up too much space and network resources. In section 3 we have already illustrated the related techniques of anti-spam. In our EMF system, we used an anti-spam engine, SpamAssassin, to block/flag spam.

SpamAssassin anti-spam engine contains a number of new technologies designed to protect against the changing techniques used by spammers. These features include checking for web links of known spam advertisers, a modular plug-in architecture, improved SQL database support for storing user data in server installations, and improved email classification.

The anti-spam module in our EMF 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. Preventing a bomb is always better than recovering from one. In EMF anti-spam module, we try to stop email bombs at the SMTP dialog phase.

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

3.2.4. Policy Filtering Module

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

- Quarantine the suspect mail
- Delay delivery of the mail
- Blind forward the mail
- Remove the attachment

Professional users can edit the configuration file directly in the Unix system. As for novice system administrators, the web-based interface facilitates configuring the policy file according to the users' filtering policy.

3.2.5. Other Utility Modules

Mail Archiving Module

Besides the modules mentioned above, the EMF system also includes the Archiving module which provides a function to archive specified email for later examination. MySQL is chose as the database system for storing the mail. If a company find it's secret files exposed, checking the archived email may trace back and find out who has let out this secrecy by mail.

Mail Statistics Module

The EMF system also provides some statistical reports to facilitate management. This is done by the Statistics Module. These reports reveal the amount of users' incoming and outgoing mail, the rate of spam mail, the rate of virus-infected mail, etc.

System Maintenance Module

A user-friendly web-based interface is provided for the system administrators to configure and/or maintain the EMF system. The system administrators can do the following administrative tasks through this module: mail routing configuration, database maintenance, configuring alerting method, setting filtering policies, and so on.

4. Experimental Result

When email goes through the EMF system, it is our Receiver-SMTP which does the receiving process. In this section, the experimental results will show the increased amount by using this method against 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 EMF system on the mail server of a university (including around 20 thousand email accounts) and on that of a medium enterprise (including 683 email accounts). Certain configuration changes are done in order to fit the organizations' environment. We then use the report of the statistical data within a month to do further analysis. The network topology of the university is shown in Figure 9, respectively.

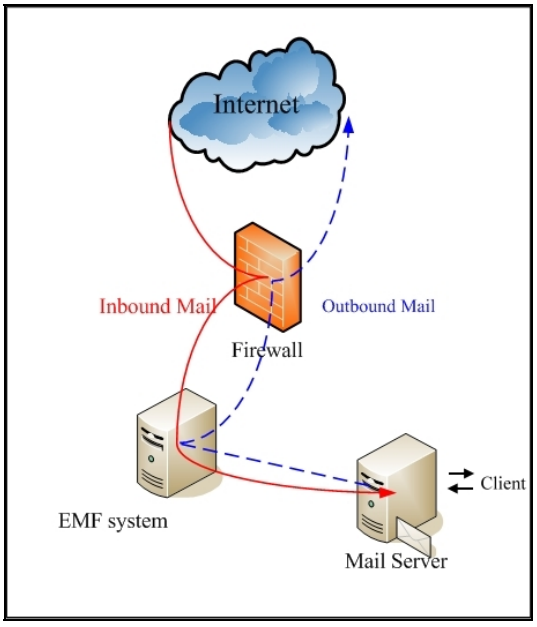

Figure 9 Network environment in a university

In order to evaluate the overhead incurred by the EMF system; we use the statistics data from the real environment and the spam mail intercepted by our EMF system as a reference data for experimenting on our system. We use the same hardware specification as those of the university we've previously experimented with our EMF system. We first run our experiment before installing the EMF system on the mail server. After installing the EMF system, we run the experiment again and compare the results to see how much overhead were incurred by the EMF system.

4.2 System Performance and Overhead

As mentioned in previous section, we install an EMF on the mail system of a college and collect the data report of a month. The Figure 10 below shows the Mailing Statistical Report from July 4th to July 28th the EMF in that university has produced.

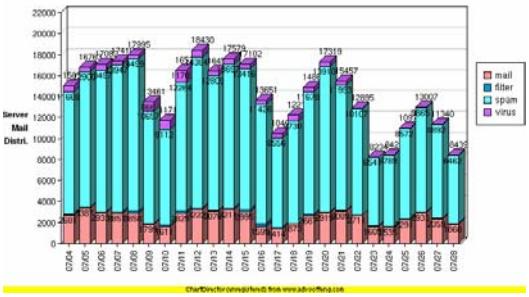

Figure 10 An EMF statistical report

The college has done certain configuration changes on the EMF system to fit its mail system in order to generate the month report shown above. From this report, we see that 2759 spam mails out of 307992 are mistakenly accepted as legitimate mail. This shows that the EMF system has a false positive rate of 0.9% (2795/307922). The other spam mails are successfully detected and taken into various actions according to the policies or filtering rules. The EMF's spam-blocking rate is around 78% ((307992 - 2759) / 390610).

Installing EMF will consequently increase expenditure on overhead. Yet how much of an amount is increased? To find out, we write a mail generator to generate 10,000 random-length mails that include 75% of spam mails. We measure the time that is required to complete receiving the 10,000 mails, and compare the results with the timing of that before installing the EMF system. We repeat this test 10 times with different random seeds, and calculate the average. The experimental result shows that the

rate of increased overhead is around 8.95%, which is in an acceptable range.

The following table shows the additional functions in our EMF compare to SpamAssassin.

	SpamAssassin	EMF 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
Detial reports	Poor	good

Table 1 Compare EMF to SpamAssassin only

5. Conclusion

As described in the previous sections, we've designed an EMF system integrated with anti-virus engine and anti-spam techniques, and used the PERL language to implement the EMF system. In this section, we summarize the contribution of this paper, and present some ideas that can be used for future work.

5.1 Conclusion and discussion

EMF is a solid email security solution that provides Virus Protection, Spam Filtering and Email Security in one complete package. Because EMF performs the filtering task before the mail enters your network, you do not need to worry about high volumes of spam from threatening your network or overloading your bandwidth.

Blocking spam at the gateway, or Message Transfer Agent (MTA), reduces network resource wastage. These resources include Internet bandwidth, mail server processing cycles, and storage capacity. As shown in the following figure, more than 80% of the emails are useless.

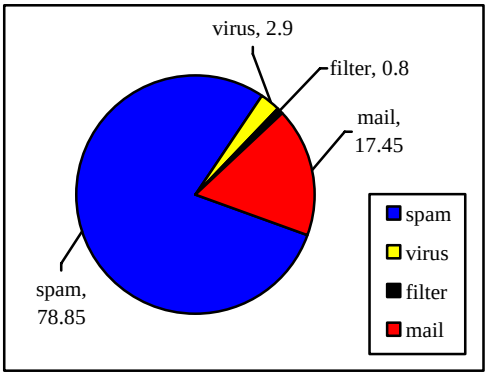

Figure 11 An EMF report in a university

As shown in the Figure 11 above, using the SpamAssassin anti-spam techniques along with checking white/black lists, the EMF system protects multilingual message streams, safely removing up to 76.21% of spam at the gateway. Thus, users can eliminate a lot of time reading garbage mail.

Either known or unknown viruses could both be scanned by the anti-virus engine used in our EMF system; this protects the network against wreckful codes. In addition, with the web-based friendly user interface, corporate communications policies can be easily managed using EMF system's flexible policy manager to gain complete and precise control over mail filtering.

No single technology can consistently eliminate spam over the long term. Providing multiple defenses is the best way to approach complete spam protection. Besides integrating Sophos' Anti-Virus Engine and SpamAssassin anti-spam techniques, we have designed some additional functions on our EMF system such as checking black/white lists. These additional features include:

- Local mailbox existence check
- checking blank email
- checking black/white lists
- checking fake routing
- Defuse DHA

These additional features enhance the accuracy of the EMF system.

5.2 Future Work

In March 2004, Allister Cournane and Ray Hunt presented a paper entitled "An analysis of the tools used for the generation and prevention of spam." [13] In the paper, it examines some of the current spam obfuscation techniques such as

HTML comments or messages that are composed entirely of URLs, etc. Further strategies should be investigated and operated in order to stop new malicious mail.

Our EMF system utilizing the Sophos Anti-Virus Engine to detect virus attached to the email. However, there are many other virus scanners. Each virus scanner has its own strength. We believe that no single anti-virus engine can fully protect against all possible threats. Therefore, to integrate multiple anti-virus engines into our EMF system should be taken in consideration when the server is power enough to do so.

Spam and viruses are now flooding through the entire network. Although the constantly improving technology has enabled us to come up with a great deal of solutions to fight them, spammers are also rapidly inventing more and more new tricks to get by the filters and anti-spam systems. Therefore, we need to improve the EMF by aiming at spammers, breaking any trick whenever they come up with one.

Reference

[1] 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.

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

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

[4] The Sophos Anti-Virus engine, http://www.sophos.com/products/

[5] Sophos virus analysis: W32/Mimail-L, http://www.sophos.com/virusinfo/analyses/w32mimaill.html

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

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

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

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

[10] MessageLabs Intelligence June 2003 Monthly report, http://www.messagelabs.com/intelligence.

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

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

[13] Allister Cournane, Ray Hunt, “ An analysis of the tools used for the generation and prevention of spam”, Computers & Security, Volume 23, Issue 2, Pages 154-166, March 2004.