

國立交通大學

電機資訊學院 資訊學程

碩士論文

電子郵件過濾系統設計與實作

Design and Implementation of E-mail Filtering System



研究生：黃貞云

指導教授：蔡文能 教授

中華民國九十四年七月

電子郵件過濾系統設計與實作

Design and Implementation of E-mail Filtering System

研究生：黃貞云

Student : Chen-Yun Huang

指導教授：蔡文能

Advisor : Wen-Nung Tsai

國立交通大學

電機資訊學院 資訊學程

碩士論文

A Thesis

Submitted to Degree Program of Electrical Engineering Computer
Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

July 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年七月

電子郵件過濾系統設計與實作

學生：黃貞云

指導教授：蔡文能 教授

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

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。

Design and Implementation of E-mail Filtering System

Student : Chen-Yun Huang

Advisors : Prof. Wen-Nung Tsai

Degree Program of Electrical Engineering Computer Science

National Chiao Tung University

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 thesis, 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.

誌 謝

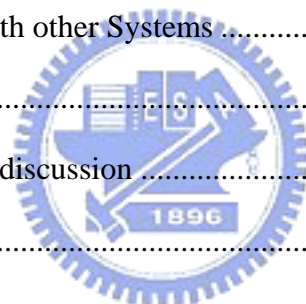
在這三年(2001~2004)的求學過程當中，真的要感謝很多人的幫忙，首先，我要感謝我的主管鼓勵我在職進修，並感謝工作夥伴們在工作上的協助，讓我有時間在職進修；其次，要感謝與我同實驗室的同學們，大家互相切磋與鼓勵，幫助我順利完成學業；最後，要感謝我的指導教授蔡文能老師的指導，在論文內容方面諸多的提點，使得我可以順利完成論文，感謝大家。



Table of Content

Chinese Abstract.....	i
English Abstract.....	ii
Acknowledgment.....	iii
Table of Content	iv
List of Figure	vi
List of Table.....	vii
Chapter 1 Introduction.....	1
1.1 Motivation	1
1.2 Objective.....	2
1.3 Synopsis of the Thesis	3
Chapter 2 Background Knowledge	4
2.1 Internet Email Standards and Fundamentals	4
2.1.1 Simple Mail Transport Protocol (SMTP)	6
2.1.2 Mail Message Format and Encoding	14
2.2 Virus.....	19
2.2.1 Viruses, Worms and Trojans	19
2.2.2 Email virus.....	20
2.2.3 Virus spoofing.....	21
2.3 Spam: Unsolicited Commercial Email	22
2.4 Spamming techniques.....	23
Chapter 3 Related Work	27
3.1 Defeating Viruses at the Gateway	27
3.2 Sophos Anti-Virus Engine	29
3.3 SpamAssassin	30

Chapter 4	EMF System	32
4.1	System Overview.....	32
4.2	System Architecture.....	33
4.2.1	Receiver-SMTP and Sender-SMTP.....	34
4.2.2	Anti-Virus Module.....	38
4.2.3	Anti-Spam Module	39
4.2.4	Policy Filtering Module.....	40
4.2.5	Other Utility Modules.....	40
Chapter 5	Experimental Result.....	41
5.1	Experimental Environment.....	41
5.2	System Performance and Overhead.....	43
5.3	Comparisons with other Systems	44
Chapter 6	Conclusion	47
6.1	Conclusion and discussion	47
6.2	Future Work	50
Reference	51



List of Figure

Figure 1	The Internet Eail System.....	6
Figure 2	The SMTP Model.....	7
Figure 3	Envelope and Message.....	12
Figure 4	Blank HTML example source.....	24
Figure 5	Invisible text examples.....	25
Figure 6	Invalid HTML tags source.....	25
Figure 7	HTML comments source.....	25
Figure 8	Vertical slicing example.....	25
Figure 9	MIME segment example.....	26
Figure 10	Letter spacing example.....	26
Figure 11	URL encoding examples.....	26
Figure 12	JavaScript example.....	26
Figure 13	EMF System overview.....	32
Figure 14	EMF System architecture.....	33
Figure 15	Mail process flow.....	33
Figure 16	Reverse DNS error reply.....	35
Figure 17	Blacklist error relpy.....	36
Figure 18	EHLO/HELO error reply.....	36
Figure 19	Fake sender reply.....	37
Figure 20	Blacklist error reply.....	37
Figure 21	Network environment in a university.....	41
Figure 22	Network environment in a company.....	42
Figure 23	An EMF report in a college.....	43
Figure 24	An EMF Mail processing report.....	44
Figure 25	An EMF spam-Mail Report.....	45
Figure 26	An EMF report in a medium company.....	47
Figure 27	An EMF report in a large company.....	48
Figure 28	Some detail reports in figure 27.....	49

List of Table

Table 1	Internet Standards related to Electronic Mail	4
Table 2	ESMTP common commands	9
Table 3	SMTP Reply Code Format	10
Table 4	Standard Header Fields.....	16
Table 5	The core MIME RFCs	18
Table 6	DNS RBLs which used by EMF system.....	35
Table 7	Experimental environment.....	42
Table 8	Compare EMF to SpamAssassin	46



Chapter 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.

1.1 Motivation

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 [25], the world's leading provider of services for managing email security, released its Labs Intelligence monthly report in June 2003 [26]. This report has highlighted on some disturbing trends in email security threats. MessageLabs found that the ratio of viruses in email was 1 in 125.5, an increase of 13.6% over May 2003.

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 June, it found 1 spam in every 2.6 emails (34.4%) of the 122 million emails scanned. This represents a decrease of 35.3% over May's figures, which peaked at 1 in every 1.8 email (55.1%). According to MessageLabs, spam has continued to increase rapidly, showing a growth of 38.5% in 2003.

1.2 Objective

In this thesis, 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.

1.3 Synopsis of the Thesis

This thesis is organized as follows: Chapter 1 introduces the motivation of designing the email filtering system. In Chapter 2, we study the background knowledge of electronic mail protocols[13], including SMTP and Internet Message Format. Chapter 2 also includes the introduction of email virus and spam technologies. More related works are investigated in Chapter 3. Chapter 4 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 Chapter 5. Finally, some discussions and conclusions will be given in Chapter 6.

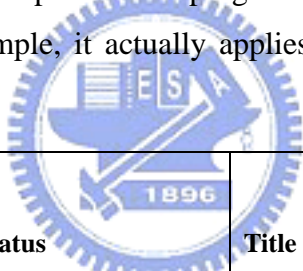


Chapter 2 Background Knowledge

The background knowledge of electronic mail protocols we investigate includes SMTP and Internet Message Format. An introduction to email virus and spam technologies will also be given in this chapter.

2.1 Internet Email Standards and Fundamentals

Email, short for electronic mail, is a method designed for messages to be sent from one computer to another, usually in text-only. The messages can also include other file clips, known as attachments. A computer user composes the email, i.e. the message, at one terminal, assigns one or more address to send it to, and then sends it. The message will be delivered to the recipient's terminal, where the recipient uses a program to receive and read his email. Although the term email seems quite simple, it actually applies to a set of protocols, standards, and conventions.



RFC Number	STD Number	Status	Title
821	10	Recommended	Simple Mail Transport Protocol (SMTP)
1869	10	Recommended	SMTP Server Extensions
1870	10	Recommended	SMTP Server Extension for Message Size
822	11	Recommended	Format of Electronic Mail Messages
1049	11	Recommended	Content-type Header Field

Table 1 Internet Standards related to Electronic Mail

Virtually all Internet protocols are defined in RFC documents, which are published by the IETF (Internet Engineering Task Force). Internet email is not an exceptional (See Table 1). The RFC document of Internet email was initially designed in the early days of the Internet when most users were part of research and development efforts. Nowadays, Internet email is one of the most common Internet applications in use of all people.

The Internet Email system works by running a sequence of events between some certain related elements, including mail message, mail user agent, mail transfer agent, and mail delivery agent. The definitions of these elements are explained in the following paragraphs.

- (1) Mail Message: A mail message includes two parts. One is the message body, which is the text of the message a user wishes to send to another over the Internet. It may also include binary files that are attached to the message. The other part is the administrative data for specifying recipients and transport medium. A message is sent out from a mail user agent (MUA).
- (2) Mail User Agent (MUA): The Mail User Agent (MUA), also know as the Email Client, is a program that is used by a user to receive and send email. The client user uses it to read incoming messages that have been delivered to his or her mailbox, and also uses it to send messages which the MUA will pass to an MTA for sending.
- (3) Mail Transport Agent (MTA): The Mail Transport Agent (MTA), also known as a mail server, or mail exchange server, is a server program or a software agent that acts as a mail router. It receives messages form an MUA or another MTA, decides which delivery method it should use according message header, and then passes the message to the appropriate Mail Delivery Agent for that delivery method.
- (4) Mail Delivery Agent (MDA): The MDA accepts a piece of mail from an MTA and delivers the message to the remote machine or writes it to a local user's mailbox. It would distribute the messages to recipients' individual mailboxes if the destination account is on the local machine, or forwards them back to an SMTP server if the destination is on a remote server.

The most important one among these components is the MTA. All the "intelligent" work of email is done by the MTA. Although it does not actually perform any of the delivery itself, it tells the other parts how to interact and what to do. It is the linkage between all the elements in the emailing process.

The figure below shows how Internet email works by a typical sequence of events between the elements. The scenario of sending mail from the sender to the receiver is given below.

- (1) The sender composes a message, types in the email address of his/her correspondent, and sends it by using an MUA (the sending MUA). The MUA formats the message and uses the Internet's Simple Mail Transport Protocol (SMTP), and MIME extensions if it has binary attachments, to send it to a local MTA, which we call it the sending MTA.
- (2) The sending MTA transfers the messages to its destination address provided in the SMTP protocol. The destination could be one or more MTAs, here known as the receiving MTA.
- (3) The receiving MTA holds an account for the recipient of the message so that it could transfer the message to an MDA after it receives the message. The MDA writes the message to the recipient's mailbox, normally a file on the file system.
- (4) The receiver can get the message from the mailbox by using another MUA (the receiving MUA), which retrieves the message using the Post Office Protocol (POP3) or the Internet Message Access Protocol (IMAP). Since this is not our point, these two protocols are not covered in this thesis.

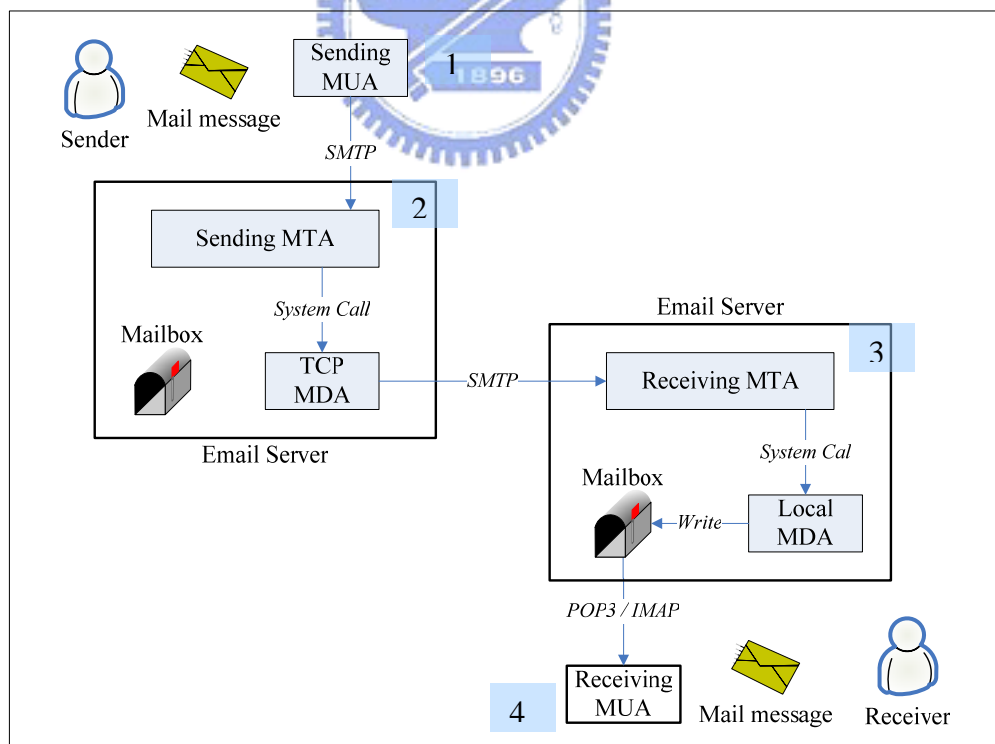


Figure 1 The Internet Eail System

2.1.1 Simple Mail Transport Protocol (SMTP)

SMTP, short for Simple Mail Transport Protocol, is a text-based protocol used for sending email messages between servers. SMTP started becoming widely used in the early 1980s, and the format was defined in RFC 821 [1] and RFC 822 [2] in 1982. As Internet technique developed, the protocol has grown and changed considerably; more protocol extensions [3] were added, and it has become as known as the Extended Simple Mail Transport Protocol (ESMTP). SMTP and ESMTP form the backbone of the Internet Mail System. Here, we use the term “SMTP” to refer to the basic protocol without service extensions; and we use “ESMTP” to refer to SMTP with other service extensions.

As mentioned in the previous section, SMTP is used by MUAs to send mail messages to MTAs, it is also used by MTAs to transfer email to other MTAs. MTAs that provide TCP/IP-base SMTP service, such as *sendmail*, listen to SMTP-port 25. Figure 2 shows the communication model of the SMTP design.

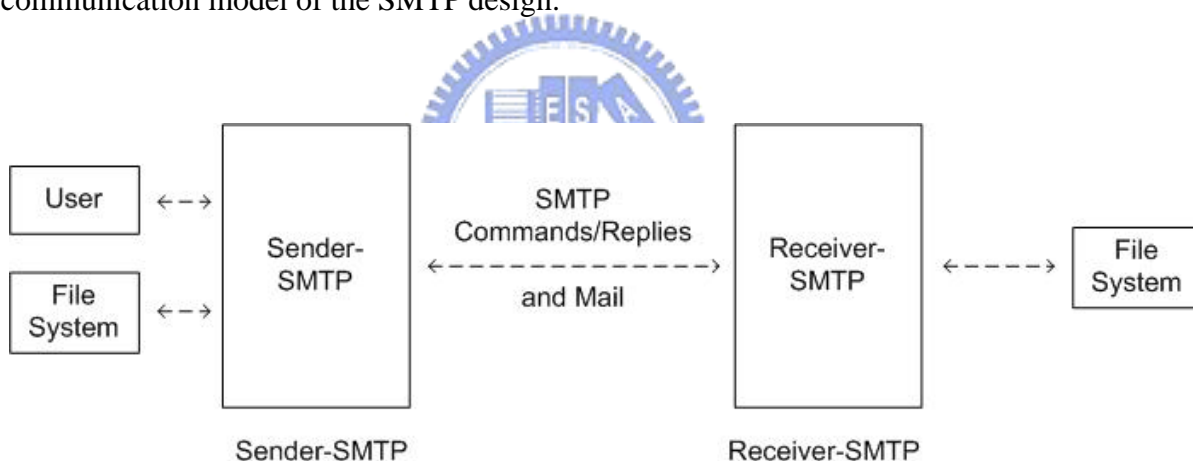


Figure 2 The SMTP Model

As result of a user mail request, the sender-SMTP establishes a two-way transmission channel to a receiver-SMTP. The receiver-SMTP may be either the final destination or an intermediate node. The sender-SMTP generates the SMTP commands and sends them to the receiver-SMTP. To the commands, the receiver-SMTP then sends the SMTP replies to the sender-SMTP. The dialog is purposely locked-step, thus each step could be done only one-at-a-time.

SMTP is a text-oriented, command-based protocol. The client issues a command; the server processes it and sends a reply back to the client. The dialog continues until the session is terminated.

In the following section, we will discuss the SMTP commands and SMTP replies that are used in the SMTP Session. We will also give a brief discussion of message envelope and mail transaction.

SMTP Session

There are several phases in an SMTP session: 1) determining which machine to connect to, 2) establishing a connection, 3) initializing the session, 4) performing various transaction dialogs, and 5) shutting down a session.

The **SMTP** mail session is started by a "HELO" command and ended by a "QUIT" command. To send out an email, we must at least use the three following commands: "MAIL FROM:", "RCPT TO:", and "DATA". Though there are many other commands that can be used in the SMTP dialect, these five commands are used much more than any others.

SMTP Commands

ESMTP consists of a minimal core of required commands and a number of commands associated with the service extensions. All SMTP commands are case insensitive. The core commands are listed in Table 2.

A minimal SMTP server must support commands including HELO, MAIL, RCPT, DATA, RSET, NOOP, and QUIT. Any command that is other than the ones listed above is an extension. The ESMTP server must at least include the EHLO command. The EHLO command enables the server to identify its support for ESMTP commands.

Command	Description
HELO	<p>Format: HELO <SP> <domain> <CRLF></p> <p>Used to identify the client, and stored it in the "Received:" message header.</p>
EHLO	<p>Format: EHLO <SP> <domain> <CRLF></p> <p>ESMTP replacement for HELO.</p> <p>Used to identify the client and request a list of ESMTP service extensions supported by the server.</p>
MAIL	<p>Format: MAIL <SP> FROM:<sender-address> [DSN parameter] <CRLF></p> <p>Used to identify the message sender, and stored in the message envelope.</p> <p>It is not necessary for the information in this field to be the same as that in the "From:" field in the message header. The data in the "From:" field is provided in the message headers, which are a part of the message proper.</p>
RCPT	<p>Format: RCPT <SP> TO:<receiver-address > [DSN parameter] <CRLF></p> <p>Used to specify the recipient of a message, and is also stored in the message envelope.</p> <p>This does not have to be the same information as that in the "To:" or "CC:" fields in the message header. Because the envelope is used by the local delivery agent, the information provided in "RCPT TO:" dictates the destination mailbox for the message, regardless of the information provided in the "To:" or "CC:" fields in the message header.</p>
DATA	<p>Format: DATA<CRLF></p> <p>Used to mark the beginning of the mail message being sent.</p> <p>Any data sent after the "DATA" command is assumed to be the message body. The message body will contain headers created by other mail servers that have processed the mail, and will also contain the original message body and any attachments. The end of the message is marked with a dot (.) surrounded by a pair of carriage-return/line-feeds. Once the end-of-message string has been acknowledged, the SMTP client can either create another mail message or can close the connection using the "QUIT" command.</p>
RSET	<p>Format: RSET<CRLF></p> <p>Used to reset the server and to nullify the entire message transaction.</p>
NOOP	<p>Format: NOOP<CRLF></p> <p>No operation.</p>
QUIT	<p>Format: QUIT <CRLF></p> <p>Used to terminate the SMTP session and closes the mail connection. Rather than the client simply closing the TCP connection, it uses the "QUIT" command to request that the server close the connection.</p>

Table 2 ESMTP common commands

SMTP Replies

The SMTP server sends a reply back to the client after processing a command. Before sending any more commands or data, the SMTP client must wait for a valid response code from the server. This assures the integrity of the message transfer.

To tell whether the information is in success or failure status is the primary purpose of the reply. It is a structured piece of data, designed for both machine and human to understand. Replies may come in as a single-line form or a multi-line form. The format and example for a single-line reply are shown below

```
reply-code<SP>human-readable-text<CRLF>
250 Requested mail action okay, completed
```

Reply Code Format	Meaning	Description
1xx	Positive Preliminary Reply	The command has been accepted and the processing of it is still in progress. Before a new command may be sent, another reply should be responded to the SMTP sender. ※ Note that while this “1xx” type is formally defined in the SMTP specification for completeness, it is not yet used by any of the SMTP commands. This means, there are actually no reply codes between 100 and 199 in SMTP.
2xx	Positive Completion Reply	The command has been successfully processed and completed.
3xx	Positive Intermediate Reply	The command was accepted, but the processing of it has been delayed during receiving additional information. After receiving a DATA command, this reply may often occur. This makes the SMTP sender to send the actual email message that is to be transferred on time.
4xx	Transient Negative Completion Reply	The command was not accepted and no action was taken, but the error is temporary and the command would be tried again. This is used for errors that may be a result of temporary glitches or conditions that may change, such as a resource on the SMTP server being temporarily busy.
5xx	Permanent Negative Completion Reply	The command was not accepted and no action was taken. Trying the same command again is likely to result in another error. An example would be sending an invalid command.

Table 3 SMTP Reply Code Format

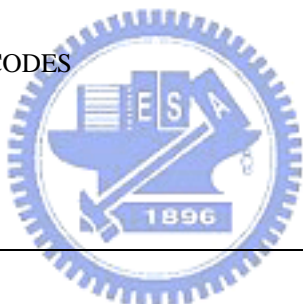
An SMTP reply code consists of three digits that indicate the server's status:

1. The first digit: This digit tells whether the response is good (1,2), intermediate (3), or incomplete (4,5). For more detailed description, see Table 3.
2. The second digit: This digit tells approximately what type of error has occurred.
3. The third digit: This digit gives more specific information about the error.

SMTP replies may also come in a multi-line form. The format is very similar to that of a single-line reply, except that the space character after the three digits is replaced with a '-' on all lines except the last one. The '-'s act as an indicator to the client showing that more reply lines are coming.

The following example shows a client sending an EHLO command and the server replying with a multi-line response:

```
ehlo test.com
250- pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-AUTH LOGIN PLAIN
250 HELP
```



The SMTP client and server can almost guaranteed to exchange mail successfully by using detailed response codes. If a response code could not be understood, the SMTP client would then close the connection to stop the operation, and return an error message back to the original sender.

SMTP relays a mail object containing an envelope and content, i.e. the message itself. Note that SMTP merely deals with how to exchange mail messages between sites; it does not define how the message body, or even the message envelope, should appear. This is why it is called the Simple Mail Transport Protocol.

The concept of the envelope is an interesting aspect to SMTP and ESMTP.

Message Envelope

The message envelope contains two pieces of information: 1) the sender's email address, and 2) the recipient's email address. When an SMTP client sends a message to an SMTP server,

the SMTP server will use the information provided in the "MAIL FROM:" and "RCPT TO:" fields to automatically create the envelope for the message.

After the envelope is constructed, the message is sent from the client to the server. Then the SMTP server will send the envelope and the message to a local delivery agent. The local delivery agent will examine the envelope, and put the message in the right user's mailbox.

The message will also include the addresses of both senders and recipients, but the addresses are not used for sending the message. The information in envelope does not have to be the same as that found in the message's "To:", "From:" and other fields. This loophole enables spammers to hide the identity of those sending bulk and unwanted mail.

Mail Transaction

The main purpose of an SMTP conversation is to exchange the mail message. This is where email messages actually get transmitted from a client to a server. Everything else in the SMTP protocol is there to support this task.

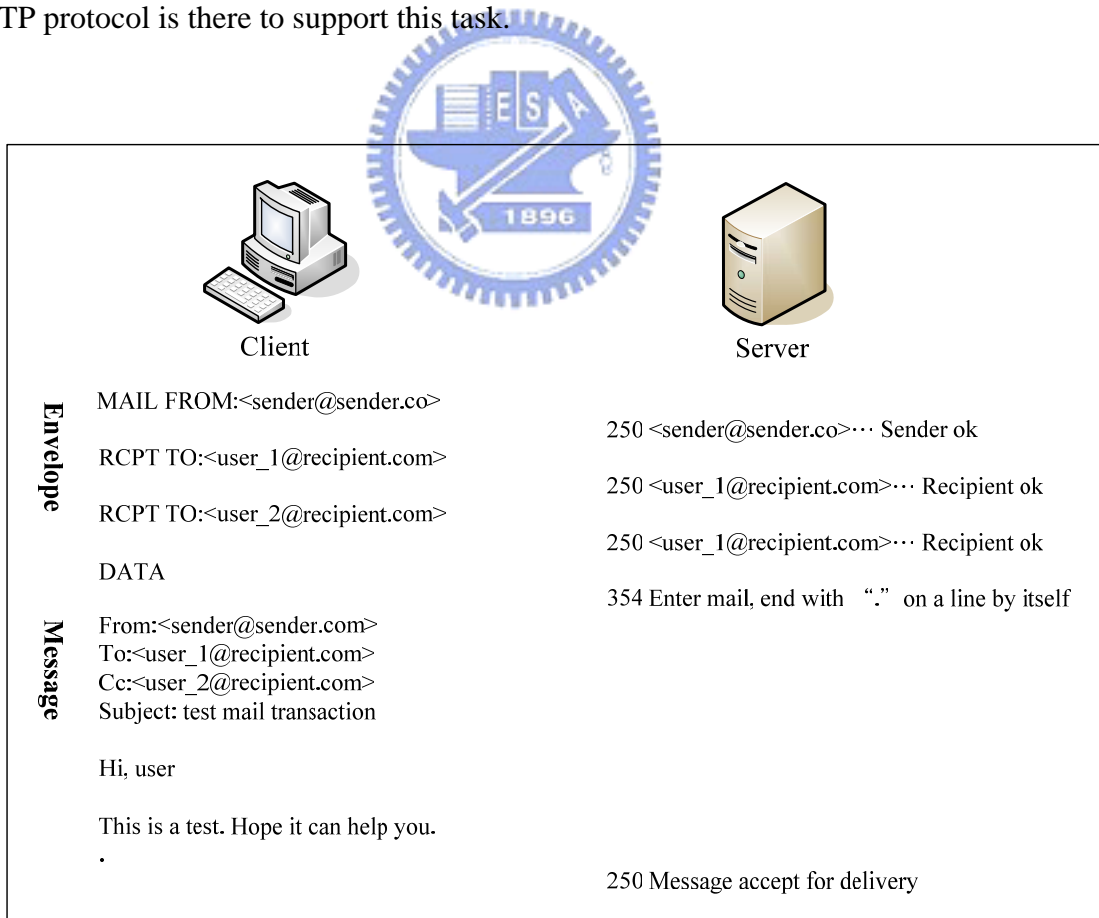


Figure 3 Envelope and Message

Mail messages are transferred one at a time. A new MAIL message will terminate previous sending mail session, if there is one. However, each message sent may go to multiple recipients.

A mail transaction usually begins with a MAIL command, followed by one or more RCPT commands, and a DATA command. The MAIL command and RCPT commands provide the server envelope information. The DATA command provides the mail message. The process is shown in Figure 3.

The REST command resets the server's internal state so that the flow of a mail transaction is disconnected and all of the non-transaction commands may be used. After receiving the REST command, the SMTP server clears its reverse-path, its forwarding-path, and the mail data buffers. The session state will also be reset to be equivalent to that when the HELO command was issued.

For example, supposed we begin sending a message by issuing a MAIL command and then change our mind, perhaps due to an error, we can use the RSET command to terminate the transaction.



2.1.2 Mail Message Format and Encoding

As we mentioned in previous section, the SMTP protocol defined in RFC 821[1] does not define the format of the message body and the message header. However, every aspect of email revolves around the keystone of email, the message. Thus, understanding email requires a solid understanding of how messages are structured. All Internet electronic mail are built upon simple text message format described in RFC 822[2].

RFC 822 defines the basic format that a mail message should be structured in, including how the message headers and body should be presented. RFC 822 has been extended by several later RFCs, but the basic structure remains the same.

The message below shows how a typical Internet email message that has been transmitted between systems may look like:

```
Received: (qmail 32161 invoked from network); 28 Dec 2001 01:22:23 -0000
```

```
Received: from mail.somedomain.com (10.45.124.32) by mail.yourdomain.com with SMTP; 28 Dec 2001 01:22:23 -0000
```

```
Received: (from user@localhost) by mail.somedomain.com (8.11.6/8.11.6) id  
fBS1Mon10019; Thu, 27 Dec 2001 19:22:50 -0600
```

```
Date: Thu, 27 Dec 2001 19:22:50 -0600
```

```
From: sender@sendingdomain.com
```

```
Message-Id: <200112280122.fBS1Mon10019@sendingdomain.com>
```

```
To: you@yourdomain.com
```

```
Subject: Hello
```

```
Hi there, how's it going?
```

An Internet email message contains two parts: one is the message header; the other is the message body. A message begins with several headers, which are formatted lines. A header begins with a header identifier, followed by a colon and a space, and then followed by the contents of the header. Most standard header identifiers are specified in RFC 822 and the other RFCs [4][5][6][7] after the extensions. Any other headers used for non-standard purposes may be created of the form “X-field-name”. After the headers comes an empty line,

followed by the message body. In this case, the message body is the text, “Hi there, how’s it going?”; any other text here are all a part of the message header.

Message Headers

A mail message is very simple: it consists of a series of text lines. Each text line is terminated with a CR (carriage-return) character followed by an LF (line-feed) character. This forms a carriage-return-linefeed combination, CRLF.

The lines of a header are grouped into fields. They provide information about the piece of mail that is to be used by both users and programs.

Each header field includes a field name, an optional whitespace, a colon, an optional command folding-whitespace, and an optional field body. It can also contain leading-whitespace. Usually, there is no whitespace between the field name and the colon.

`field-name<SP>”:”[SP][field-body]CRLF`

The field-name consists of a sequence of any printable US-ASCII characters, but not including the space character or colon. Most field names are a series of alphanumeric characters, often combined with the “-” character.

RFC 822 details the standard headers, which will be used when sending mail across the Internet. Most of these fields are quite common, and are found in most email systems. RFC 822 defines a standard set of fields for mail messages, as shown in Table 4.

The lines in a mail message are not allowed to be longer than 1,000 characters; moreover, long lines can be difficult for humans to read. Due to this, it is possible to split a field into multiple lines for readability. This is called folding. It is allowed in both structured and unstructured fields.

Human readability is also an important concern in creating the mail message standards. Folding is a good example. It was build into standard to keep the length of lines in the range of 72 to 76 characters. This makes email messages easier to read.

Field name	Description
From	The creator of the message
Sender	The sender of the message
Reply-To	The address to send replies to
To	Primary recipients of the message
Cc	Secondary recipients of the message
Bcc	Blind Carbon Copy recipients of the message
Message-ID	The message's unique identifier
In-Reply-To	The message being replied to
References	All messages ancestors
Date	The date the message was created
Received	MTA footprint
Return-Path	The address of the originator
Subject	The subject of the message
Comments	Miscellaneous comments regarding the message
Keywords	Topical keywords related to the message
Encrypted	Encryption information (obsolete)
Resent-*	Fields created when redistributing
X-*	Extension fields

Table 4 Standard Header Fields

A CRLF followed by some amount of whitespace can break long lines into shorter lines, improving the readability. Note that whitespace is necessary because followed by only a CRLF for a header line will not work, since the mail system interprets that as the beginning of a new header. An example of a folded header looks like as follows:

```
Content-Type: multipart/related; type="multipart/alternative";
        boundary="-----_NextPart_XQbgIHkK8cEhPs1ZmYjUkfroG"
```

Whitespace can consist of either ASCII space or TAB characters.

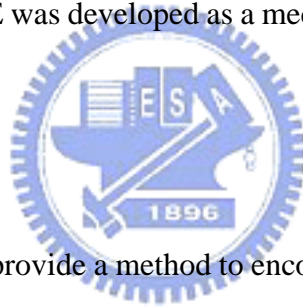
Message Body

The message body comes after the message headers. An empty line separates the two components. Everything before the empty line is considered to be a header, and everything after it is considered to be the message body.

The "format" of the message body is not necessarily specified in RFC 822; however, it applies some standards. Although all types of content can be sent over SMTP, messages that do not conform to these standards will not make it across the Internet's SMTP network. For example, messages that contain 32-bit binary objects will not be able to cross a multiple-hop mail route, since some systems only support 7-bit ASCII characters.

To convert extended data into 7-bit ASCII, there are two popular encoding techniques in use today: Uuencode and MIME. Uuencode is based on free technology; it has been around for a long time. However, Uuencode has several limitations which prevent its use in many situations. As a solution, MIME was developed as a mechanism for providing rich content over an email transport.

Uuencode and Uudecode



Uuencode and Uudecode provide a method to encode and decode extended data into/from 7-bit ASCII form. However, this method is not yet defined as an RFC.

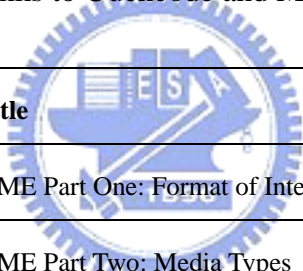
Uuencode and Uudecode only support a single flat namespace on the filesystem with poor flexibility. They do not work well with Macintosh, AS/400, HP-3000, or other systems that use a multi-node file system.

Even though Uuencoded file attachments can be infinitely attached in messages, the messages may not be always cleanly integrated into the message body by the receiving mail system. Also, because international language support for Asian and European keyboards needs 8-bit ASCII characters in the message body, an attachment-centric approach isn't an apt solution. MIME is better suited for this particular problem.

Multipurpose Internet Mail Extensions (MIME)

Unlike Uuencode and Uudecode, the Multipurpose Internet Mail Extensions (MIME) is actually an Internet Standard for the format of email. The Internet mail system can use MIME to attach files to mail messages. MIME is a protocol for Internet email that enables the transmission of binary data (non-text data) such as graphics, audio, video and other binary types of files. The main MIME RFCs are draft standards (See Table 5). The combination of the SMTP protocol and the MIME specification is the basis of the modern Internet mail system.

Besides converting data and attaching binary files, Uuencode and MIME can also be used to relay extended characters such as those found in Asian and European dialects. For example, a sample message may be shown in English, and it does not use any extended ASCII characters found in *non*-English languages. Yet, many nations use extended characters that are not found in the 7-bit ASCII character set, and these messages require encapsulation in order to be sent across the Internet reliably. Thanks to Uuencode and MIME to make this possible.



RFC Number	Title
RFC 2045	MIME Part One: Format of Internet Message Bodies
RFC 2046	MIME Part Two: Media Types
RFC 2047	MIME Part Three: Message Header Extensions for Non-ASCII Text
RFC 2048	MIME Part Four: Registration Procedures
RFC 2049	MIME Part Five: Conformance Criteria and Examples

Table 5 The core MIME RFCs

2.2 Virus

As the Internet provides vast benefits, its security is now being threatened by viruses and spam. What exactly is a computer virus? Computer virus is a program or a piece of computer code that executes itself onto any computer devices to invade computers and networks. So how was the first computer virus started? In the mid-1980s two programmers in Pakistan, Basit and Amjad, found that people were plagiarizing their software. So they wrote a program that would put a copy of itself and a copyright message on any floppy disk copies their customers made. Consequently, an entire virus counter-culture has emerged. Nowadays, new viruses are globally spreading every moment, corrupting data, slowing networks down, and even causing some serious damage.

2.2.1 Viruses, Worms and Trojans

As mentioned above, a virus or a worm is a computer program that can spread across computers and networks by automatically making copies of itself. Most of the time, the user will not notice. Viruses can display irritating messages, steal data, or even enable other users to control over your computer, causing harmful effects.

To infect your computer, a virus program would have to be executed first. The virus will make sure that this happens. They sometimes attach themselves to other programs or hide in code that runs automatically when you open certain types of file. The loopholes of security flaws in your computer's operating system also enables the viruses to automatically execute and spread themselves.

An infected file might come in an email attachment, a download from the Internet, or on a disk. The virus code automatically starts to run as soon as the file is opened, copying itself to other files or disks, altering and damaging your computer.

Worms are similar to viruses. Yet, they can create exact copies of themselves and spread via the communications between computers, not needing to be carried through any programs or documents. Many viruses, such as MyDoom or Bagle, behave like worms and use email to forward themselves.

Trojan horses pretend to be legitimate software, but are actually programs with harmful functions that could do similar damage as viruses. They cannot spread as fast as viruses because they do not make copies of themselves. However, they now often work hand-in-hand with viruses: Viruses may download Trojans that record keystrokes or steal information, and Trojans are used to infect a computer with a virus.

As an example of Trojans, DLoader-L may arrive in an email attachment as an urgent update from Microsoft for Windows XP. If you run it, it will download a program that uses your computer to connect to certain websites, and then try to overload them. This is called a denial of service attack.

2.2.2 Email virus

Email is now the most common way that viruses are transmitted between computers. Email system provides the attachment facility which is normally used for emailing documents, images, and so on. However, it is also possible for attachments to contain malicious programs. Sometimes, the email virus embedded in attachments will automatically execute when the user reads, or, in certain cases, previews the email. This is different from email-borne viruses, which will cause infection only if the user opens the attachment.

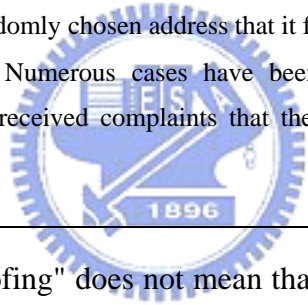
In the pass few years, email viruses have aimed for Microsoft Outlook and Outlook Express users, though other mail clients may pose a risk. In any occasion, if an attachment is involved, the user might open it and become infected.

Nimda Worm [19] is a well know email virus. It is a multi-vector virus, which causes the user to be infected while visiting an infected website, seeking out vulnerable servers on the Internet and uploading its files to it, via email, or through a network. These kind of viruses are also known as Internet worms. While most worms just make carbon-copies of themselves all over the hard drive or through email to others, Nimda embeds its code into executable files (.EXE file) found on the local drives. It is the first worm that actually infects other files; this makes it unique. If a user who is not very web-literate receives the email virus, the user might suppose that the mail is sent from a trusted source and open the attachment. The viruses will then infect the mail software and use the content of the host's address book to send themselves to other mailboxes.

2.2.3 Virus spoofing

Some Email-distributed viruses used spoofing technique, the well known examples are the Klez [20] and Sobig [21] virus. They randomly takes a name from the infected person's hard disk and mail themselves out as if they were from that randomly chosen address. This way the email cannot be traced back to the originator who is sending the virus. It may cause the users of uninfected computers to be wrongly informed. Recipients of these viruses are therefore misled as to the address from which they were sent, and may end up sending to an address that does not exist, or alerting the wrong person. Email spoofing also attempts to trick the user into making a damaging statement or releasing sensitive information. More information regarding the spoofing technique can be found in many web sites of those anti-virus products providers.

The following paragraph is extracted from Symantec's web site [20]:



This worm often uses a technique known as "spoofing." When it performs its email routine it can use a randomly chosen address that it finds on an infected computer as the "From:" address. Numerous cases have been reported in which users of uninfected computers received complaints that they sent an infected message to someone else.

Be sure to note that "Spoofing" does not mean that your computer is infected by "worm, virus, etc." and sending out messages from your address book. Instead, it may imply that "someone's" or "your friend's" machine, which has your email address in their address book, is infected or hijacked and sending out messages with your email address as well as other email addresses that are randomly taken from the address book of infected machine.

2.3 Spam: Unsolicited Commercial Email

We all hate “Junk mail” because they wasted our time. Spam, also known as Unsolicited Commercial Email (UCE), is the Internet version of “Junk mail.” It is an attempt to deliver a message, over the Internet, to someone who would not otherwise choose to receive it. Almost all spam is commercial advertising, such as “want a sex girl,” “get-rich-quick schemes,” “home-working jobs,” “loans or pornographic website,” etc. Potential target lists can be gathered in many ways. We will discuss this in section 2.3.1.

Email spamming refers to sending email to thousands and thousands of users - similar to a chain letter. Spamming is often done deliberately to use network resources. Email spamming may be combined with email spoofing, so that the recipients cannot trace the original source. Some email systems, including Microsoft Exchange, have the ability to block incoming mail from a specific address. However, it is difficult to prevent some spam from reaching your email inbox because these individuals change their email address frequently.

The word "spam", as it is used to denote junk e-mail or junk Netnews postings, is derived from a Monty Python sketch set in a movie/tv studio cafeteria. In the sketch, the word "spam" takes over each item offered on the menu until the entire dialogue consists of nothing but "spam," such as such as "Spam, egg and Spam, bacon and Spam." This does so closely resembles what happens when mass unsolicited mail and posts take over mailing lists and Netnews groups that the term has been pushed into common usage in the Internet community.

The exact line between spam and legitimate email, or spam and free bulk email is not as obvious as it may seem. To some, any and all email that does not come from an approved source is spam. According to Mail Abuse Prevention System (MAPS) [14]:

An electronic message is "spam" IF: 1) the recipient's personal identity and context are irrelevant because the message is equally applicable to many other potential recipients; AND 2) the recipient has not verifiably granted deliberate, explicit, and still revocable permission for it to be sent; AND 3) the transmission and reception of the message appears to the recipient to give a disproportionate benefit to the sender.

The IETF describe why spam is a problem in documents such as RFC 2635[8]. And the IETF document RFC 2505 [9] gives more information to mail administrators on how to deal with spam. Geoff Mulligan wrote a book discussing how to remove spam in email processing in March 1999[12].

The low cost of e-mail spamming engines offered for sale with millions of e-mail addresses, coupled with the fact that the sender does not pay extra to send e-mail, has resulted in the current explosive growth of "Junk email." Currently, there is no legal remedy to use to stop the e-mail spammers unless the spammer offers to sell illegal items.

USA Congress is currently considering legislation to require the marking of unsolicited commercial email (UCE). However, the legislation is not yet complete.

2.4 Spamming techniques

Before implementing a good mail filter system to defuse malicious mail, we must understand those tricks used in spam mail. Spam mail has grown rapidly within the past few years. Spammers are using increasingly sophisticated techniques to undermine legitimate businesses, making it more difficult to prevent their attacks.

In March 2004, Allister Cournane and Ray Hunt present a paper entitled "An analysis of the tools used for the generation and prevention of spam" [11] .

In that paper, it examines the problems caused by the spamming of e-mail and newsgroup users. Spamming is now considered to be a serious threat to the Internet and is posing a serious threat to both users and ISP resources. In particular, this paper examines the motivation of, and the tools used to generate, spam. Methods of protection and prevention are then discussed.

The techniques used by most spammers are briefly illustrated as follows.

(1) Acquired E-mail lists

The methods used to obtain emailing lists are: Common sources, Renting lists, Purchasing lists, and Email harvesters. Harvester is an application which scans HTML pages for e-mail addresses and associated names. By scanning HTML source for mailto: tags, e-mail addresses can be found.

(2) Spam obfuscation

As new techniques for detecting spam e-mail are created, new counter-techniques are introduced into spam e-mail generation software. Tricks that Spammers may used to obfuscate the mail filter includes:

(a) Blank HTML: sending e-mail messages which contain no plain text, Figure 4

illustrates an example of using Blank HTML.

- (b) Invisible text: attempting to hide legitimate text inside a message, Figure 5 is the example.
- (c) Invalid HTML tags: hiding legitimate text within a spam message with poorly formatted HTML form, an example is given in Figure 6.
- (d) HTML comments and blank tags: using HTML comments to split possible key words and prevent a filter from successfully identifying them, Figure 7 is an example.
- (e) Vertical slicing: slicing the spam message text into vertical strips and placing them within a HTML table, as shown in Figure 8.
- (f) MIME segment: placing a legitimate message in the plain text segment (which is often never displayed) and the spam message in the HTML segment, an example is shown in Figure 9.
- (g) Letter spacing: separating each character by a space character, Figure 10 is the example.
- (h) URL encoding: encoding URL in hexadecimal form with escape characters to prevent anti-spam filters from identifying, an example can be found in Figure 11.
- (i) Malicious JavaScript: placing spam message in a JavaScript variable which will be written to the page at runtime, the example is shown in Figure 12.

Spam's increasing use of complex HTML coding means that spam filters must be able to read and understand HTML and identify these types of new tricks, in order to be effective for an organization.

```
<html><div>  
<a href="http://www.your-info-station.com/Sla/eb.php?x=52c">  
</a></html>
```

Figure 4 Blank HTML example source

-
- (a) 445992001 448223992 unanimously repetition 002348821 799340773 435893022
modem (noun) a device or program that enables a computer to transmit
data 344000232
 - (b) X-Mime-Key: search words: interface external telephone modem copper
modulate demodulate Internet e-mail 56k bandwidth binary signal voice
bps bits per second compression fax CCITT V.34 protocol
 - (c) search words: interface external telephone
modem copper modulate demodulate Internet e-mail 56k
-

Figure 5 Invisible text examples

< Despite statements last week from chief U.N. inspector Hans Blix that full cooperation was expected from Iraq, Iraqi Foreign Minister Najji Sabri lashed out at the United Nations in a 19-page letter to Secretary-General Kofi Annan written in Arabic. In it, Sabri repeated previous claims that Iraq has no weapons of mass destruction and that the inspections are just a false pretense for the United States and Britain to attack his country. >

Figure 6 Invalid HTML tags source

-
- (a) <HTML><BODY>milli<!-- xe64 -->onaire</BODY></HTML>
 - (b) <HTML><BODY>millionaire</BODY></HTML>
-

Figure 7 HTML comments source

-
- (a) <table cellpadding=0 cellspacing=0 border=0>
<tr>
<td>
<table cellspacing=0 cellpadding=0 border=0><tr><td><font
face="Courier New, Courier, mono" size=2> H
 T
 S
</td></tr></table>
</td>
<td>
<table cellspacing=0 cellpadding=0 border=0><tr><td><font
face="Courier New, Courier, mono" size=2> E
 H
 P
</td></tr></table>
</td>
...
 - (b)

H	E	L	L	O						
T	H	I	S	I	S	M	Y			
S	P	A	M	M	E	S	S	A	G	E
-

Figure 8 Vertical slicing example

```

-----_NextPart_001_2D3DF_01C29D73.26716240
Content-Type: text/plain;
Short for modulator-demodulator. A modem is a device or program that enables a
computer to transmit data over, for example, telephone or cable lines. Computer
information is stored digitally, whereas information transmitted over telephone
lines is transmitted in the form of analog waves. A modem converts between
these two forms.
-----_NextPart_001_2D3DF_01C29D73.26716240
Content-Type: text/html;
<p><b><font face=Arial>Earn thousands now from home, NO RISK INVOLVED! 1000%
return GUARANTEED!</font></b></td>

```

Figure 9 MIME segment example.

```

(a)  U N I V E R S I T Y   D I P L O M A S
(b)  F * R * E * E   L - O - A - N - S   O ^ N ^ L ^ I ^ N ^ E

```

Figure 10 Letter spacing example

```

(a)  http://7763631671/obscure.htm
(b)  http://0xCeBF9e37/obscure.htm
(c)  http://0316.0277.0236.067/obscure.htm
(d)  http://3468664375@3468664375/o%62s%63ur%65%2e%68t%6D

```

Figure 11 URL encoding examples

```

<HTML><HEAD><SCRIPT LANGUAGE="Javascript"><!-- var Words = "
%3CHTML%3E%3CBODY%3E%3CH1%3E%20you%20w%62nt%20to%20e%62rn%20million%24%3F.%3C
/H1%3E%3CH2%3E%3Click%20%3C%62%20href%3D%27http%3A//www.mywe%63sit%66.%64om%27%3E
here%3C/a%3E%20to%20find%20out%20more%21%3C/H2%3E%3C%63r%3E%3Cimg%20src%3D%27ht
tp%3A//www.myim%62gesour%64%66.%64om/im%62g%66.jpg%27%3E%3C/BODY%3E%3C/HTML%3E"
function SetNewWords() { var NewWords; NewWords = unescape(Words);
document.write(NewWords); } SetNewWords(); // --> </SCRIPT> </HEAD> <BODY>
</BODY> </HTML>

```

Figure 12 JavaScript example

Chapter 3 Related Work

In this chapter, 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 Symantec and TrendMicro, and the Anti-Virus Engine of Sophos. In section 3.1, we review the techniques that are used to defeat viruses at the gateway. In section 3.2, we introduce the Anti-Virus Engine from Sophos. Section 3.3 gives a brief discussion regarding the SpamAssassin.

3.1 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" [10], 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 depth 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.



3.2 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 [17] 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.

3.3 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 [15] 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 judgments 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 non-spam 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. The PERL module [Mail::SpamAssassin classes](#) can be found here:

<http://spamassassin.apache.org/>



Chapter 4 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 thesis, we design and implement an Electronic Mail Filtering (EMF) system to provide consolidated protection – not only against spam, but also against viruses. Figure 13 gives an overview of the EMF system that we have implemented.

4.1 System Overview

As shown in Fig. 13, 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.

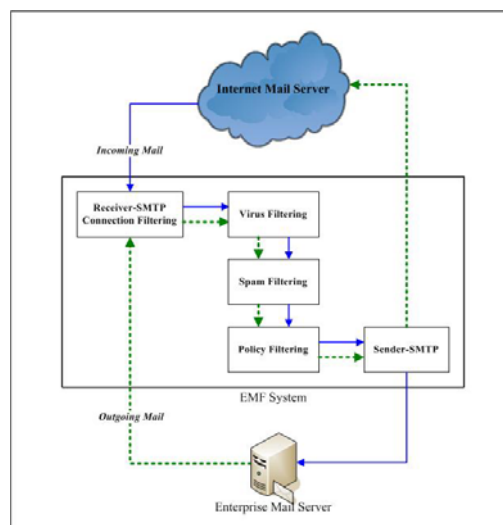


Figure 13 EMF System overview

4.2 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.

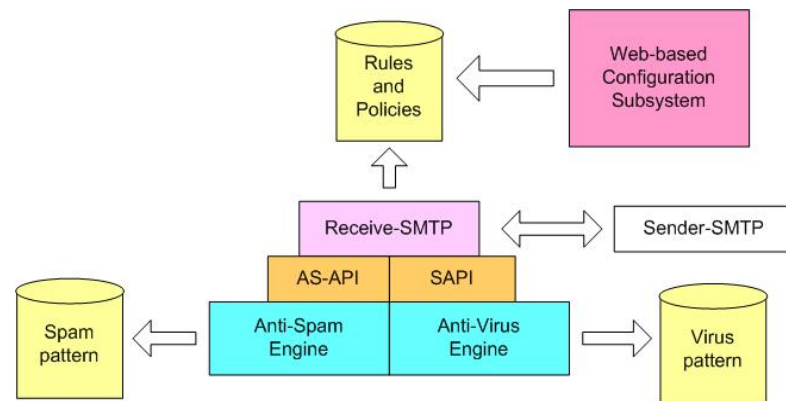


Figure 14 EMF System architecture

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 14 shows the system architecture of our EMF system. The detail mail process flow is shown in Figure 15.

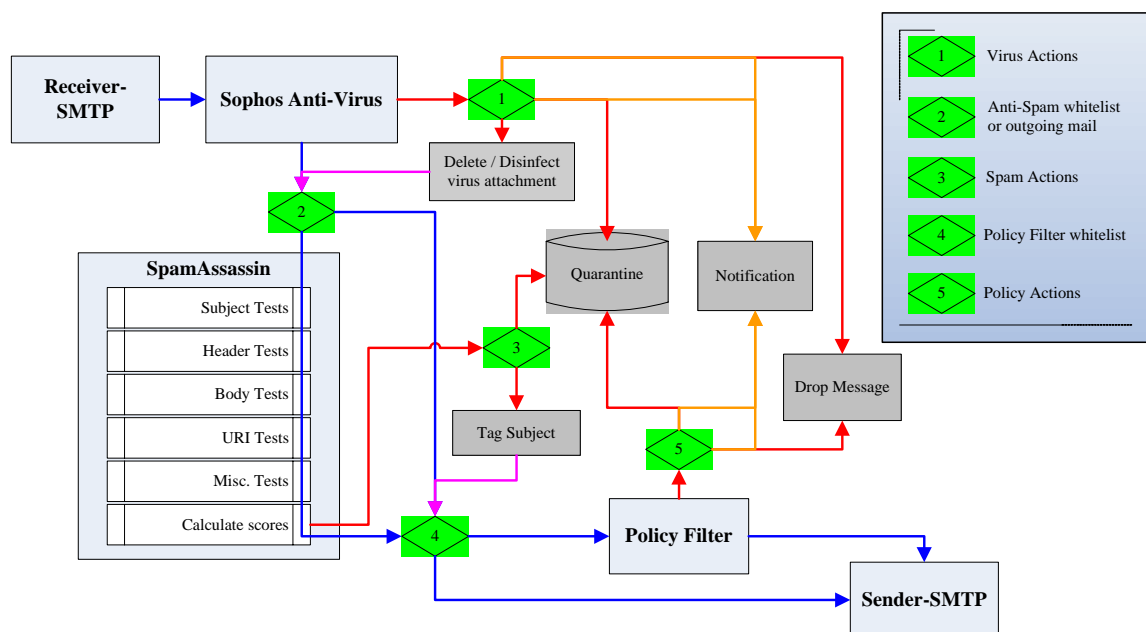


Figure 15 Mail process flow

4.2.1 Receiver-SMTP and Sender-SMTP

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 SMTP-receiver. 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 16.

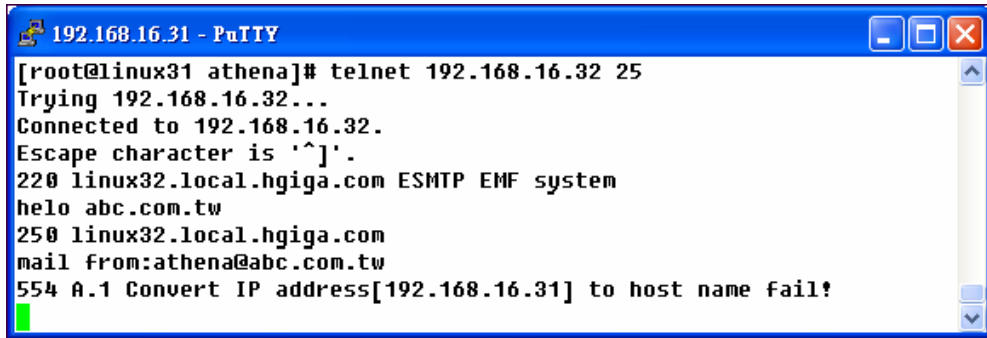


Figure 16 Reverse DNS error reply

- 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]
```

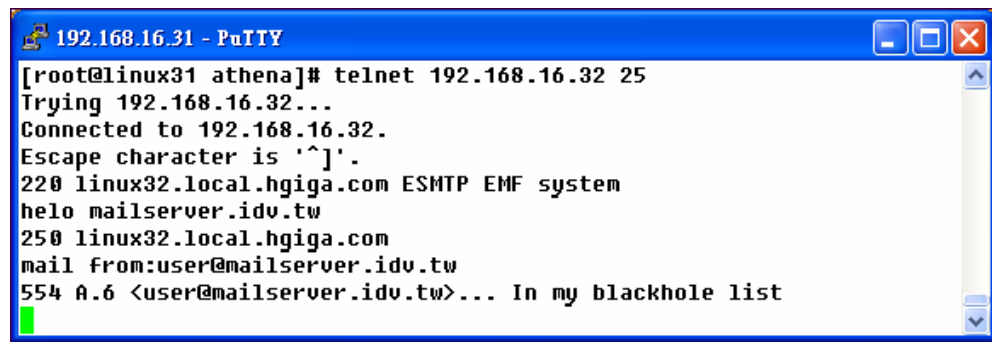
RBL	Description	Supporter
blackholes.mail-abuse.org	Rejected	http://www.mail-abuse.org/rbl/
dialups.mail-abuse.org	Dialup	http://www.mail-abuse.org/dul/
relays.mail-abuse.org	Open spam relay	http://work-rss.mail-abuse.org/rss/
spam.dnsrbl.net	This is a list of known/proven SPAM sites.	http://www.dnsrbl.com/
sbl.spamhaus.org	SPAM sites	http://www.spamhaus.org/SBL
relays.ordb.org	Open spam relay	http://www.ordb.org
list.dsbl.org	Open spam relay	http://dsbl.org

Table 6 DNS RBLs which used by EMF system

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. There are 7 DNS RBLs used in EMF system, as shown in Table 6.

- 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 17.

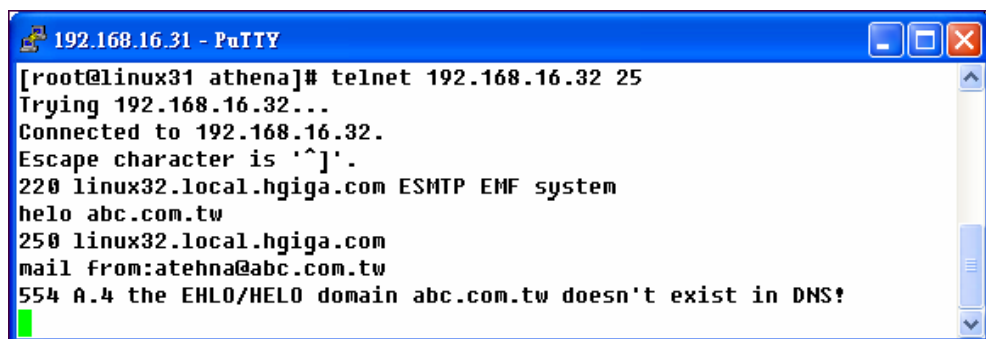


```
192.168.16.31 - PuTTY
[root@linux31 athena]# telnet 192.168.16.32 25
Trying 192.168.16.32...
Connected to 192.168.16.32.
Escape character is '^]'.
220 linux32.local.hgiga.com ESMTPEMF system
helo mailserver.idv.tw
250 linux32.local.hgiga.com
mail from:user@mailserver.idv.tw
554 A.6 <user@mailserver.idv.tw>... In my blackhole list
```

Figure 17 Blacklist error reply

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 18.



```
192.168.16.31 - PuTTY
[root@linux31 athena]# telnet 192.168.16.32 25
Trying 192.168.16.32...
Connected to 192.168.16.32.
Escape character is '^]'.
220 linux32.local.hgiga.com ESMTPEMF system
helo abc.com.tw
250 linux32.local.hgiga.com
mail from:atehna@abc.com.tw
554 A.4 the EHLO/HELO domain abc.com.tw doesn't exist in DNS!
```

Figure 18 EHLO/HELO error reply

Validating the envelope sender

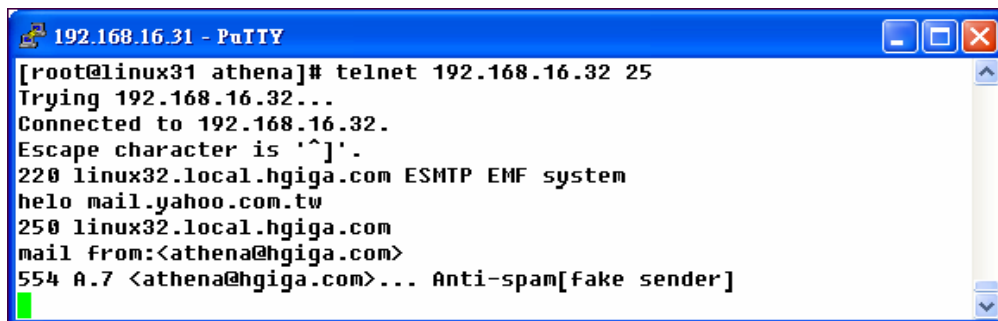
In SMTP MAIL command we check to see if the envelope sender is permitted to send email to your mail server. There are two check items here.

- 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 19.

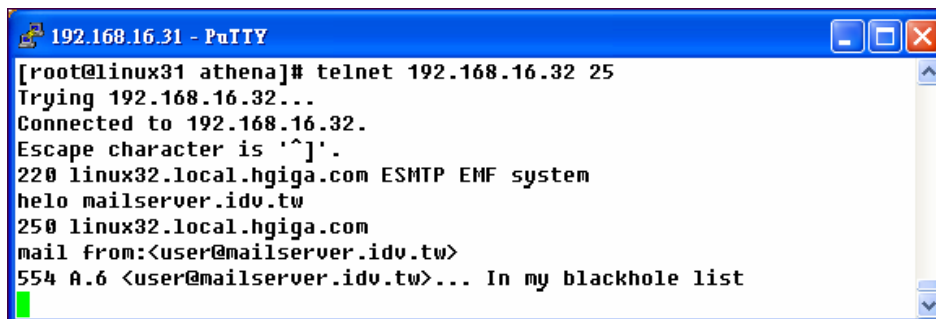


```
192.168.16.31 - PuTTY
[root@linux31 athena]# telnet 192.168.16.32 25
Trying 192.168.16.32...
Connected to 192.168.16.32.
Escape character is '^]'.
220 linux32.local.hgiga.com ESMTPEMF system
helo mail.yahoo.com.tw
250 linux32.local.hgiga.com
mail from:<athena@hgiga.com>
554 A.7 <athena@hgiga.com>... Anti-spam[fake sender]
```

Figure 19 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 20.



```
192.168.16.31 - PuTTY
[root@linux31 athena]# telnet 192.168.16.32 25
Trying 192.168.16.32...
Connected to 192.168.16.32.
Escape character is '^]'.
220 linux32.local.hgiga.com ESMTPEMF system
helo mailserver.idv.tw
250 linux32.local.hgiga.com
mail from:<user@mailserver.idv.tw>
554 A.6 <user@mailserver.idv.tw>... In my blackhole list
```

Figure 20 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. Detail information regarding this PERL module can be found in the following URL:

<http://search.cpan.org/~dskoll/MIME-tools-5.417/lib/MIME/Tools.pm>

4.2.2 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 (APplication 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. Information about these products can be found at the following URL:

<http://www.sophos.com/products/software/>

The latest version of SAVI-Perl is available on CPAN, or via the URL:

<http://www.csupomona.edu/~henson/www/projects/SAVI-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.

4.2.3 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 chapter 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 bandwidth. 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.

4.2.4 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.

4.2.5 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.

Chapter 5 Experimental Result

When email goes through the EMF system, it is our SMTP-Receiver which does the receiving process. In this chapter, 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.

5.1 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 29116 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.

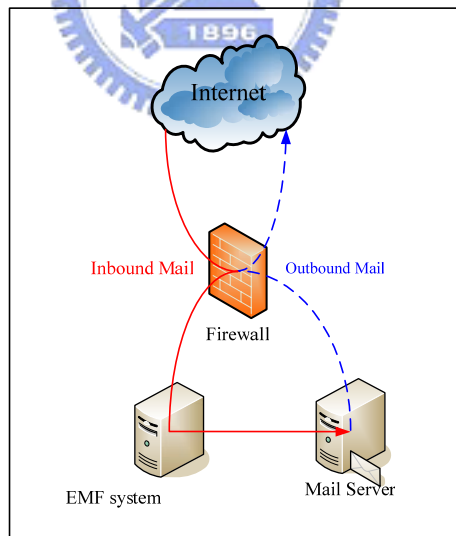


Figure 21 Network environment in a university

The network topologies of the university and of the enterprise are shown in figure 21 and figure 22, respectively.

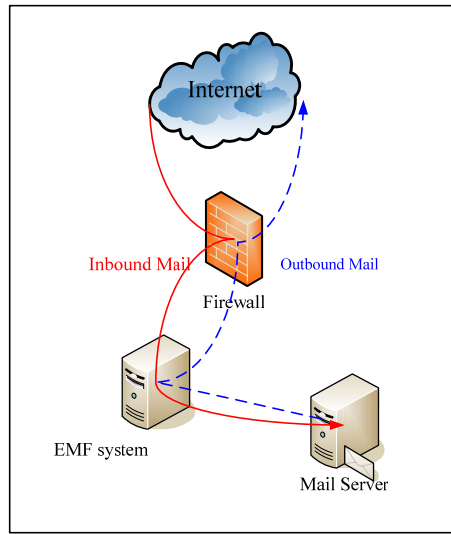


Figure 22 Network environment in a company

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. Table 7 shows the experimental environments.

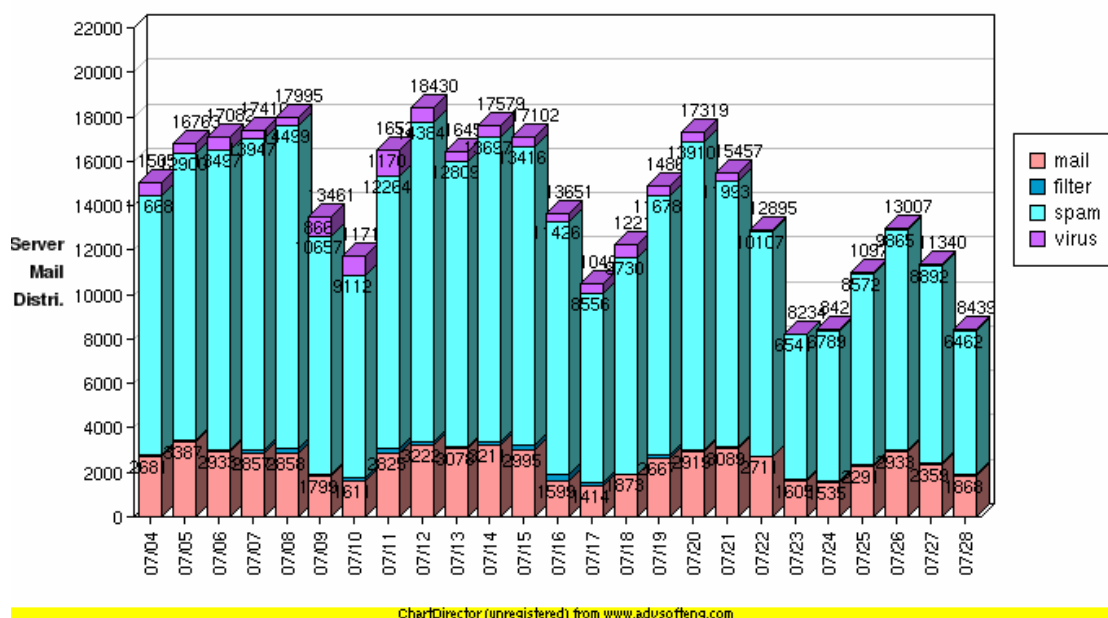
	Hardware in the university	Hardware in the company
CPU	2 * Intel(R) Xeon(TM) CPU 3.06GHz	4 * Intel(R) Xeon(TM) CPU 3.06GHz
Memory	4 GB	5 GB
NIC	Intel(R) PRO/1000 Ethernet	Tigon3 [partno(NA) rev 1002 PHY(5703)] (PCI-X:100MHz:64-bit) 10/100/1000BaseT Ethernet
Hard Disk	SCSI 60 GB	SCSI 60 GB

Table 7 Experimental environment

5.2 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 23 below shows the Mailing Statistical Report from July 4th to July 28th the EMF in that university has produced.

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% (2759/307922). 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).



信件類別	信件數	比例	子類別		
MAIL	68158	17.45%	正常信件	68158	100.00%
FILTER	3115	0.80%	過濾信	3115	100.00%
SPAM	307992	78.85%	放行垃圾信	2759	0.90%
			刪除的垃圾信	587	0.19%
			貼標籤垃圾信	69936	22.71%
			隔離的垃圾信	234710	76.21%
VIRUS	11345	2.90%	隔離的病毒信	11345	100.00%
總計	390610	100%			

Figure 23 An EMF report in a college

Installing EMF will consequently increase expenditure on overhead. Yet how much of an amount is increased? To find out, we use 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.

5.3 Comparisons with other Systems

Though we use the SpamAssassin as our anti-spam engine, our EMF system has also implemented some additional techniques to improve the system accuracy in determining legitimate mail. Different from the mail relay server of a general SpamAssassin, EMF has the capability to check whether the local mailbox exists or not, and block the mails which the receivers' mailbox doesn't exist. Therefore, the loading of the mail server will be reduced. The following report (Figure 24) shows a high rate of 97.45% of blocked spam mails due to blocking the mails of non-existing local mailboxes.

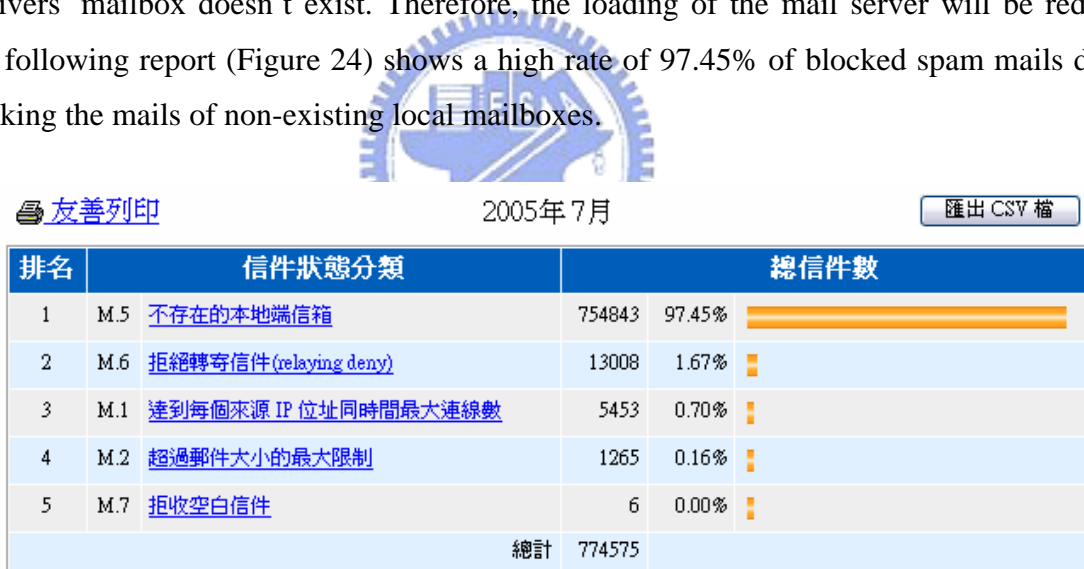


Figure 24 An EMF Mail processing report

From the figure 25 below, we see that the added design of EMF enables the mail system to block an extra amount of spam to those that the SpamAssassin has already blocked. For example:

- (1) A.11 in the figure 25: Used for checking the spam with a “MAIL FROM: <aaglm.aaglm@edwinedwin.com>” during the SMTP transaction.
- (2) A.7 in the figure 25: Used for checking the spam which source is not from a relay

client and has not passed the SMTP Auth, but shows to be from a local mailbox during the SMTP transaction.

- (3) A.12 in the figure 25: Used for checking the sender's IP and comparing it with the domain name of "MAIL FROM". If it is not a corresponding IP-domain name, then it is declared as a spoofed sender address. This function is effective when blocking mail with a "MAIL FROM: <sombody@yahoo.com.tw>" but is actually not from a Yahoo or a Hotmail server.

友善列印 2005年 7月 匯出 CSV 檔

排名	信件狀態分類		總信件數		
1	A.10	Anti-Spam 引擎評定為垃圾信件	450291	67.64%	<div style="width: 67.64%; height: 10px; background-color: orange;"></div>
2	A.11	Anti-Spam 垃圾信寄件者	79854	11.99%	<div style="width: 11.99%; height: 10px; background-color: orange;"></div>
3	A.7	假冒本地端寄件者	43147	6.48%	<div style="width: 6.48%; height: 10px; background-color: orange;"></div>
4	A.3.1	IP 位址被列入暫時性黑名單	40691	6.11%	<div style="width: 6.11%; height: 10px; background-color: orange;"></div>
5	A.5	寄件者 Email 信箱不存在	32949	4.94%	<div style="width: 4.94%; height: 10px; background-color: orange;"></div>
6	A.12	虛假路由	10376	1.55%	<div style="width: 1.55%; height: 10px; background-color: orange;"></div>
7	A.6	Email 被列入黑名單	8338	1.25%	<div style="width: 1.25%; height: 10px; background-color: orange;"></div>
總計			665646		

Figure 25 An EMF spam-Mail Report

A spammer might bombard a domain with thousands of generated email addresses, attempting to collect valid email addresses from an organization. This is known as Directory Harvest Attack (DHA). However, SpamAssassin cannot detect DHA.

The EFM system has combined the techniques of black list/white list, and thus, can prevent DHA from attacking. This is easily done: Once it is discovered that half of the user accounts (in reply to RCPT TO command) are unknown or incorrect, the system would take this as a DHA attack. It would then consign the source IP to the temporary black list. This not only slows down spammers from collecting local mailboxes, but also prevents DHA from causing DoS.

In the following table, we compare the difference between SpamAssassin and our EMF system.

	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
Detail reports	Poor	Good

Table 8 Compare EMF to SpamAssassin



Chapter 6 Conclusion

As email is becoming more and more an important element in business, email servers are now suffering from more and more risks and threats such as mail bomb attacks and virus attacks. To prevent your organization's server from becoming the next victim, it is necessary to implement a good email-filtering tool.

There are several researches and products of anti-virus and anti-spam as described in related works. As shown in the previous chapters, 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 chapter, we summarize the contribution of this thesis, and present some ideas that can be used for future research.

6.1 Conclusion and discussion

EMF is an enterprise level 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.

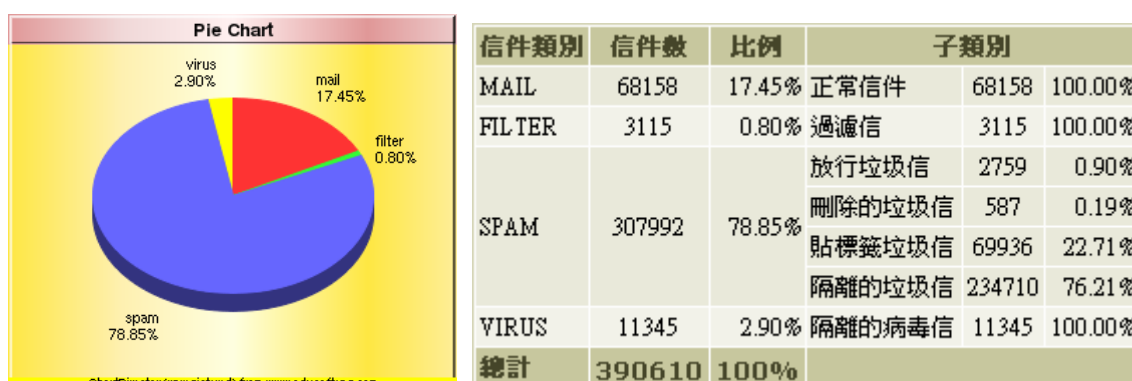


Figure 26 An EMF report in a medium company

As shown in the figure 26, 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.

友善列印 2005年 7月 [匯出 CSV 檔](#)

排名	信件狀態分類	總信件數		
1	M 郵件伺服器限制條件	774540	32.50%	
2	A Anti-Spam	665572	27.92%	
3	S 正常信件	400897	16.82%	
4	T 連線逾時	199403	8.36%	
5	I 不完整的 SMTP 程序	115643	4.85%	
6	D 攻擊行為	99168	4.16%	
7	C SMTP 命令錯誤	94995	3.98%	
8	V Anti-Virus	18303	0.76%	
9	X 伺服器狀態	14506	0.60%	
總計		2383027		

Figure 27 An EMF report in a large company

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.

The figure 27 shows a report generated by an EMF installed in a large enterprise. From the report, we can observe noticeable effects that the EMF system's additional functions have brought.

These additional features include:

- (1) Local mailbox existence check (M.5 in the figure 28)
- (2) checking blank email (M.7 in the figure 28)
- (3) checking black/white lists (A.3.1 and A.6 in the figure 28)
- (4) checking fake routing (A.12 in the figure 28)
- (5) Defuse DHA (D.2 in the figure 28)

排名	信件狀態分類	總信件數
1	M.5 不存在的本地端信箱	754843 97.45%
2	M.6 拒絕轉寄信件(relaying deny)	13008 1.67%
3	M.1 達到每個來源 IP 位址同時間最大連線數	5453 0.70%
4	M.2 超過郵件大小的最大限制	1265 0.16%
5	M.7 拒收空白信件	6 0.00%
總計		774575



排名	信件狀態分類	總信件數
1	A.10 Anti-Spam 引擎評定為垃圾信件	450291 67.64%
2	A.11 Anti-Spam 垃圾信寄件者	79854 11.99%
3	A.7 假冒本地端寄件者	43147 6.48%
4	A.3.1 IP 位址被列入暫時性黑名單	40691 6.11%
5	A.5 寄件者 Email 信箱不存在	32949 4.94%
6	A.12 虛假路由	10376 1.55%
7	A.6 Email 被列入黑名單	8338 1.25%
總計		665646

排名	信件狀態分類	總信件數
1	D.2 字典攻擊	98741 99.56%
2	D.1 RSET 命令 DoS 攻擊	433 0.43%
總計		99174

Figure 28 Some detail reports in figure 27

6.2 Future Work

In the near future, we may focus on helping Email Service Providers (ESPs) prevent their users from sending spam. The ESPs here include most commercial ISPs (e.g., Hinet and Seednet), free email account providers (e.g., Hotmail and Yahoo), universities, and so on.

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.” [11] In the paper, it examines some of the current (2003) 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] Jonathan B. Postel, “RFC 821 - Simple Mail Transfer Protocol”, August 1982.
<http://www.ietf.org/rfc/rfc0821.txt>
- [2] David H. Crocke, “RFC 822 – Standard for the FORMAT of ARPA Internet Text messages,” August 1982, <http://www.ietf.org/rfc/rfc0822.txt>
- [3] J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker, “RFC 1869 - SMTP Service Extensions”, November 1995. <http://www.ietf.org/rfc/rfc1869.txt>
- [4] J. Klensin, Editor AT&T Laboratories, “RFC 2821 - Simple Mail Transfer Protocol”, April 2001. <http://www.ietf.org/rfc/rfc2821.txt>
- [5] P. Resnick, Editor QUALCOMM Incorporated, “RFC 2822-Internet Message Format”, April 2001. <http://www.ietf.org/rfc/rfc2822.txt>
- [6] N. Borenstein, N. Freed, “RFC 1521 - MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies”, September 1993. <http://www.ietf.org/rfc/rfc1521.txt>
- [7] K. Moore,” MIME (Multipurpose Internet Mail Extensions) Part Two: Message Header Extensions for Non-ASCII Text”, September 1993.
<http://www.ietf.org/rfc/rfc1522.txt>
- [8] S. Hambridge, A. Lunde, “RFC 2635 - DON'T SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam*)”, June 1999.
<http://www.ietf.org/rfc/rfc2635.txt>
- [9] G. Lindberg , “RFC 2505 - Anti-Spam Recommendations for SMTP MTAs”, February 1999. <http://www.ietf.org/rfc/rfc2505.txt>
- [10] 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.
- [11] 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.
- [12] Geoff Mulligan, “Removing the Spam: Email Processing and Filtering”,

- Addison-Wesley, March 16, 1999, **ISBN:** 0201379570.
- [13] Kevin Johnson, "Internet Email Protocols: A Developer's Guide", Addison-Wesley, January 15, 2000.
- [14] Mail Abuse Prevention System (MAPS), <http://www.mail-abuse.org>
- [15] The Apache SpamAssassin Project, <http://spamassassin.apache.org/>
- [16] Anti-Spam Research Group (ASRG) of the Internet Research Task Force (IRTF), <http://asrg.sp.am/>
- [17] The Sophos Anti-Virus engine, <http://www.sophos.com/products/>
- [18] Sophos virus analysis: W32/Mimail-L, <http://www.sophos.com/virusinfo/analyses/w32mimail.html>
- [19] Cert Advisory CA-2001-26 Nimda Worm, <http://www.cert.org/advisories/CA-2001-26.html>
- [20] The W32.Klez.H@mm worm , <http://securityresponse.symantec.com/avcenter/venc/data/w32.klez.h@mm.html>
- [21] Virus information: W32/Sobig-F, <http://www.sophos.com/virusinfo/analyses/w32sobigf.html>
- [22] Gillmor, D., "Data Privacy Protection Must Start with IT", Computerworld, Vol. 32, No. 45, November 9, 1998.
- [23] Hartman L.P., "The Rights and Wrongs of Workplace Snooping", Journal of Business Strategy, Vol. 19, No. 3, May/June 1998, 16-19.
- [24] Miller-Seumas and John Weckert, "Privacy, the workplace and the Internet", Journal of Business Ethics, Dec 2000, Vol.8, No.3, pp.255-265.
- [25] MessageLabs, <http://www.messagelabs.com>.
- [26] MessageLabs Intelligence June 2003 Monthly report, <http://www.messagelabs.com/intelligence>.
- [27] Email Bombing and Spamming, http://www.cert.org/tech_tips/email_bombing_spamming.html
- [28] Tom Merritt, "What is Email Spoofing?", May 09, 2000. <http://www.techtv.com/screensavers/answerstips/story/0,24330,2566233,00.html>