User Guide

SMTP Trap

Current Version: 1.04l

Table of Contents

License & Copyright Notice

CAREFULLY READ THE FOLLOWING LICENSE AGREEMENT. BY OPENING THE PACKAGE OR CLICKING ON THE "ACCEPT" BUTTON, YOU ARE CONSENTING TO BE BOUND BY AND ARE BECOMING A PARTY TO THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, CLICK THE "DO NOT ACCEPT" BUTTON.

1. LICENSE GRANT. The package (or web site download) contains software ("Software") and related explanatory written materials ("Documentation"). "Software" includes any upgrades, modified versions, updates, additions and copies of the Software. "You" means the person or company who is being licensed to use the Software or Documentation. "We" and "us" means SMTPTrap.com, Inc..

We hereby grant you a nonexclusive, non-transferable, limited license to use one copy of the Software on any single computer, provided the Software is in use on only one computer at any time and only for licensee's internal business operations. The Software is "in use" on a computer when it is loaded into temporary memory (RAM) or installed into the permanent memory of a computer--for example, a hard disk, CD-ROM or other storage device.

If the Software is permanently installed on the hard disk or other storage device of a computer (other than a network server) and one person uses that computer more than 80% of the time, then that person may also use the Software on a portable or home computer subject to the restrictions in this Agreement.

Please contact us at sales@smtptrap.com for information about obtaining a multi-user or network license.

2. TITLE. This license is not a sale. We remain the owner of all right, title and interest in the Software and Documentation.

3. ARCHIVAL OR BACKUP COPIES. You may either:

- -- make one copy of the Software for backup or archival purposes or
- -- transfer the Software to a single hard disk, provided you keep the original solely for backup or archival purposes.

4. THINGS YOU MAY NOT DO. United States copyright laws and international treaties protect the Software and Documentation. You must treat the Software and Documentation like any other copyrighted material--for example a book. You may not:

- -- copy the Documentation,
- -- copy the Software except to make archival or backup copies as provided above,
- -- modify or adapt the Software or merge it into another program,
- -- reverse engineer, disassemble, decompile or make any attempt to discover the source code of this Software,

-- place the Software onto a server so that it is accessible via a public network such as the Internet, or

-- sublicense, rent, lease or lend any portion of the Software or Documentation.

5. LIMITED WARRANTY. We do not warrant the product because we give you 30 days to evaluate the product free of charge.

To the extent permitted by applicable law, THE FOREGOING LIMITED WARRANTY IS IN LIEU OF ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, AND WE DISCLAIM ANY AND ALL IMPLIED WARRANTIES OR CONDITIONS, INCLUDING ANY IMPLIED WARRANTY OF TITLE, NONINFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, regardless of whether we know or had reason to know of your particular needs. No employee, agent, dealer or distributor of ours is authorized to modify this limited warranty, or to make any additional warranties.

SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS, AND YOU MAY ALSO HAVE OTHER RIGHTS WHICH VARY FROM STATE TO STATE.

6. LIMITED REMEDY. Our entire liability and your exclusive remedy shall be:

-- the replacement of any diskette(s) or other media not meeting our Limited Warranty which is returned to us with a copy of your receipt, or

-- If we are unable to deliver a replacement diskette(s) or other media that is free of defects in materials or workmanship, you may terminate this Agreement by returning the Software and documentation and your money will be refunded.

7. DAMAGE LIMITATIONS. IN NO EVENT WILL WE BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS, OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING FROM THE USE OR THE INABILITY TO USE THE SOFTWARE, EVEN IF WE HAVE BEEN ADVISED OF THE POSSIBILITY OF THESE DAMAGES.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION MAY NOT APPLY TO YOU. IF OUR LIMITED WARRANTY AND/OR LIMITED REMEDY SHALL BE HELD INEFFECTIVE OR TO HAVE FAILED THEIR ESSENTIAL PURPOSES, OUR TOTAL LIABILITY FOR DAMAGES, WHETHER IN CONTRACT, TORT OR OTHERWISE, SHALL NOT EXCEED THE LICENSE FEES PAID BY YOU FOR THE SOFTWARE LICENSED HEREUNDER.

8. TERM AND TERMINATION. This license agreement takes effect upon your use of the Software and remains effective until terminated. You may terminate it at any time by destroying all copies of the Software and Documentation in your possession. It will also automatically terminate if you fail to comply with any term or condition of this license agreement. You agree on termination of this license to either return to us or destroy all copies of the Software and Documentation in your possession.

9. CONFIDENTIALITY. The Software contains trade secrets and proprietary know-how that belong to us and it is made available to you in strict confidence. ANY USE OR DISCLOSURE OF THE SOFTWARE, OR OF THE SOFTWARE, OR OF ITS ALGORITHMS, PROTOCOLS OR

INTERFACES, OTHER THAN IN STRICT ACCORDANCE WITH THIS LICENSE AGREEMENT, MAY BE ACTIONABLE AS A VIOLATION OF OUR TRADE SECRET RIGHTS.

10. GENERAL PROVISIONS.

(a) This written license agreement is the exclusive agreement between you and us concerning the Software and Documentation and supersedes any and all prior oral or written agreements, negotiations or other dealings between us concerning the Software.

(b) This license agreement may be modified only by a writing signed by you and us.

(c) In the event of litigation or alternative dispute resolution process between us concerning the Software or Documentation or this Agreement, the prevailing party in the litigation or process will be entitled to recover attorney fees and expenses from the other party.

(d) This license agreement is governed by the laws of the state of New Jersey.

(e) You agree that the Software will not be shipped, transferred or exported into any country or used in any manner prohibited by the United States Export Administration Act or any other export laws, restrictions or regulations.

Introduction

SMTP Trap is a junk/bulk email filter program. This feature rich program acts as a proxy between your mail server and the Internet. It intercepts every incoming message and routes the message to your mail server based upon the rules setup by you. SMTP Trap is a server-based solution and works in conjunction with a mail server. This solution does not delete junk emails from your favorite email client like outlook's inbox after it has been fetched from the mail server. Rather, it eliminates the junk/bulk email before it is even sent to your mail server. In the appendix of this document we have included a junk mail primer. Please refer to the appendix after you are familiar with the features of this program to come up with a good strategy to block unwanted messages.

1.1 Features

SMTP Trap blocks unwanted emails using various customizable techniques. It supports the following features:

- Blocks by MX lookup failure
- Blocks by Reverse lookup failure
- Allows certain IPs/domains based upon a list
- Supports authenticated SMTP (RFC 2554) for your mail server's valid users
- Blocks certain IPs/domains based upon a list
- Supports muli-home mail server
- Blocks based upon a sender
- Blocks based upon receiver
- Blocks based upon regular expression search in subject or body
- Saves what seems like junk email for further action
- Runs as Windows service

These features will be discussed in detail in the configuration section.

Installation

1.2 Requirements

SMTP Trap runs on a Windows 2000 server or professional machine. SMTP trap can be installed on the same machine as your mail server.

1.3 Package Contents

The installation package is supplied in a self-extracting file SMTPTrapInstall.exe. The package contains the following files:

1. allowrelayiplist.txt
2. blockediplist.txt
3. localdomainlist.txt
4. routinglist.txt
5. smtpauthlist.txt
6. smtpflush.exe
7. smtptrap.exe
8. smtptrap.ini
9. trustediplistforlocaldelivery.txt
10. SMTPTrap_UserGuide.pdf (This document)
11. ReverseLookup.pdf
12. readme.txt

1.4 Installation Steps

SMTP Trap installation is real simple. Please follow these simple steps:

- Create a folder for SMTP Trap (c:\smtptrap)
- Extract all the files in the self-extracting file to this folder.
- Install SMTP Trap service, only if you want to run the program as a service. More details are provided later on how to install the program as a service.
- Configure the smtptrap.ini. A separate chapter covers the SMTP Trap configuration.

```
Every SMTP Trap installation requires a valid license key.  SMTPTrap is
now a free product.  This key has been embedded in the SMTPTrap.ini
file.
```

1.4.1 Installation of SMTP Trap service

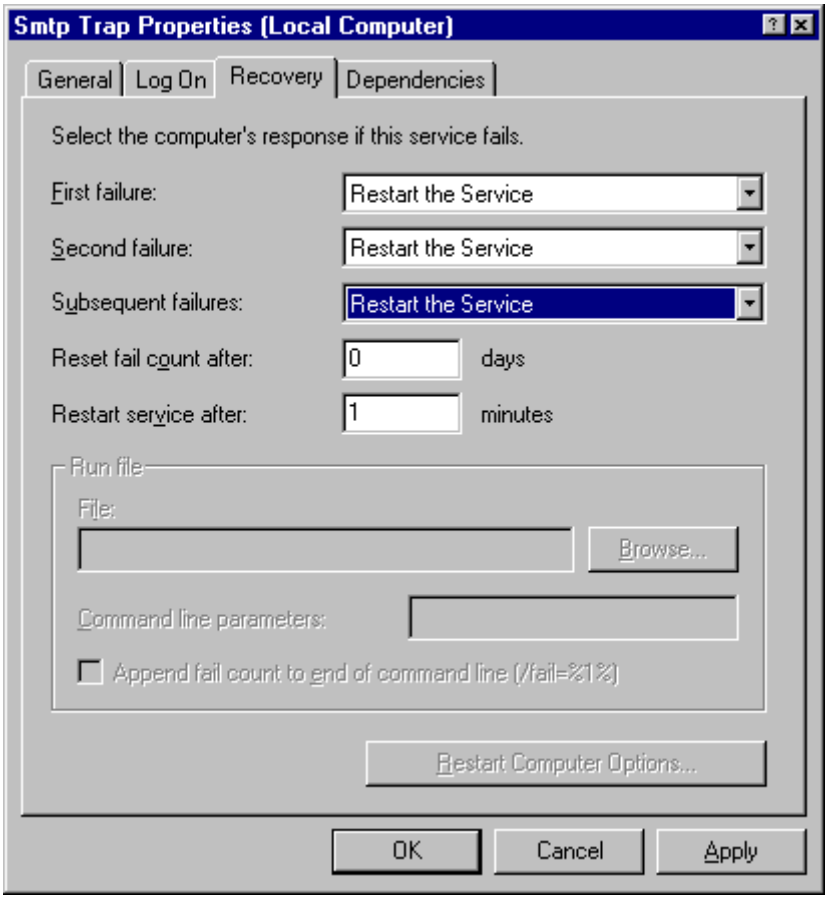
From DOS prompt switch to the SMTP Trap folder and type the following command:

smtptrap /install

SMTP Trap service will be created.

Set the service to start automatically.

To recover from any unexpected crashes, we recommend that you change the Recovery parameter for the service. This can be done from the services control panel. Select the properties for the SMTP Trap service and click on the recovery tab. Change the first failure, second failure, and subsequent failures parameters to 'Restart the Service'.



The service can be removed by running the following command from the DOS prompt:

smtptrap /remove

SMTP Trap Configuration

SMTP Trap works with one initialization file and several other files, which are used to define the rules for blocking unwanted emails. The SMTP Trap configuration is totally customizable and lets the system administrator define as to how tight a system they want to build.

1.5 INI file (SMTPTRAP.INI)

This is the main configuration file, and must be present for SMTP Trap's operation. Please use a text editor like Notepad to edit the file. The following table describes the parameters.

Parameter	Remarks
ListnerPort	Enter the port where SMTP Trap listens for inbound connections. Typically this value should be 25.
ListnerIP	Enter the IP address, which SMTP Trap listens.
OutPort	This port should match with the port, which your mail server is listening.
OutIP	This IP address should match with your mail server's IP address.
MXLookup	0 or 1. If 1, SMTP Trap accepts the connection only from a valid mail server or an unauthenticated client. There are more notes on this later.
ReverseLookup	0 or 1. If 1, SMTP Trap We have supplied you an Adobe document, ReverseLookup.pdf. Please refer to this handy guide to setup the reverse resolution for your mail server(s). You can send this document to other administrators if their configuration requires a reverse resolution.
LogFilePath	Supply the path (ending with a back slash) where the log file should be created.
SMTPFlushFrequency	Time in seconds when SMTP Trap reprocesses the saved messages. A value of 300 is recommended.
CacheReloadFrequency	Time in seconds when SMTP Trap reloads its configuration files. A value of 300 is recommended.
MaxThreads	A number from 1 to 99. Start with a value of 10 to 20. Depending upon how busy your mail system is, you may need to further increase this value.
Logging	normal/verbose/debug/none normal-Logs startup/shutdown and only necessary messages verbose-Logs detailed messages

Parameter	Remarks
	debug-Logs debug messages for problem solving none-No messages are logged
BlockNullSender	0 or 1. If 1, messages are not accepted with no sender information (from address).
BlockNullSubject	0 or 1. If 1, messages are not accepted with blank subject.
WelcomeMessage	This message is displayed when sending program first connects to SMTP Trap. You can use this message to tell the sender about your policies on bulk emails, and contact information in case of disputes. Use \n to break the message into several lines.
ThreadMaxTime	Time in seconds. If the sender does not drop the connection, SMTP Trap waits this long to force-drop the connection. A value of 900 is recommended.
LicenseKey	This key is required to run the program.
RoutingListRecordTerminator	This string identifies the end of a routing record. The default is '[R]'.
MaxMessageSize	Maximum size of the message in Megs, SMTP Trap will accept. (Default 64 Megs)

Note: Any changes made to smtptrap.ini file, require the program to be restarted. Other configuration files get refreshed automatically at desired interval.

1.6 Allowed relay IP list (ALLOWRELAYIPLIST.TXT)

This list contains the IP addresses or names of the machines, which are allowed to use the mail server (relay) unconditionally. Once a connection is authenticated, it goes through only the routing checks, rest of the checks are ignored. You should specify one address/range per line. Use a semicolon as comment.

You should enter the IP addresses or the names for the following:

- The web servers, which will use your mail server to relay.
- The mail servers, which do not have their MX records, and you want to accept messages from them.
- The email clients, who cannot use SMTP Authentication.

Use a star (*) as wild card character while using the names. Note that the names in this context mean the reverse resolved names.

```
Example:
;To grant the whole class C
12.162.5.0
;To grant a subnet
12.162.5.3/255.255.255.0
*mydomain.com
```

1.7 Blocked IP list (BLOCKEDIPLIST.TXT)

Enter the IP addresses or the names of the machines, which you want to block from relaying to your mail server. Technically, this is the best way to prevent unwanted emails. However, this happens to be the least efficient. Most spammers use dynamic IP addresses using dial-up connections, leaving very little behind to track. The syntax for entering the IP addresses is the same as allowrelayiplist.txt file. Use a star (*) as wild card character while using the names. Note that the names in this context mean the reverse resolved names.

1.8 Local domains list (LOCALDOMAINLIST.TXT)

This list contains the local domains, which are being managed by this server. SMTP Trap uses this list to accept a message from an unauthenticated connection only if the recipient(s) belong to one of those domains. You can use a star (*) for wild card. A single star in the file represents open relay.

```
Example:
supportetc.com
smtptrap.com
```

1.9 SMTP authentication list (SMTPAUTHLIST.TXT)

This list contains the user ids and the passwords to validate the users, who want to use authenticated SMTP to relay through your mail server. One entry per line must be specified with the user id followed by the password delimited by a pipe symbol (|).

```
Example:
rick|welcome
sales|password
```

1.10 Trusted IP list for local delivery (TRUSTEDIPLISTFORLOCALDELIVERY.TXT)

This list contains the names of the domains, which can be trusted, and are allowed to send a message to your mail server. An entry this list bypasses the MXLookup check (see smtptrap.ini). This list should contain the names of rather large and well known domains like hotmail.com, yahoo.com etc. Typically, these domains use separate outbound mail servers to send messages. These servers are not listed under the MX records for their domains. You can use a star (*) character for wild card. Use a star (*) as wild card character while using the names. Note that the names in this context mean the reverse resolved names.

1.11 Routing list (ROUTINGLIST.TXT)

This file contains the rules for rejecting, accepting, or saving a message based upon its contents. One routing rule may occupy many lines. The end of a rule is terminated by a terminator defined in the INI file (RoutingListRecordTerminator). The default value for this terminator is '[R]'. This parameter is defined in the INI file. When the routing rules are applied, for all matched messages, SMTP Trap can block, accept, or save them for further action. You can set up custom rules by send and/or recipient. The routing rule may contain the following parameters.

Parameter	Remarks
Action	Required. The accepted values are: B-Block A-Accept S-Save When the save action is chosen, the messages are saved in the spool folder under SMTP Trap working folder.
Status	Required. The accepted values are: A-Active (The rule will be applied) I-Inactive
Sender	Optional. You can use the star (*) character as wild card in sender's address.
Recipient	Optional. You can use the star (*) character as wild card in recipient's address.
Pattern	Optional. Supply the pattern in the subject or the body. Simple regular expressions are accepted. Multiple words must be supplied within double quotes. To use double quotes as the search string itself, use two double quotes. And and or conditions can be supplied. See examples below. The parameter, SearchTarget must be supplied with pattern.
SearchTarget	Must be used while supplying the pattern. The parameter identifies the target for the search pattern. B-Body will be searched for the pattern S-Subject will be searched for the pattern
NewSender	Optional. Use this parameter, If you want to accept the message, however, somehow change the sender's address to indicate to the recipient that this is a possible junk email. You can use the mnemonic %S% to replace with original sender. For example, to change the new sender's address prefixed with 'JUNK_', set the NewSender parameter to be 'JUNK_%S%'.

```
Examples:
; The sender for emails from great-steals.com will be prefixed with
'JUNK_'.
Status=A
Action=A
Sender=*great-steals.com
NewSender = JUNK_%S%
[R]
; Email sent to accounts@something.com will be blocked.
Status=A
Action=B
Recipient=accounts@something.com
[R]

; This rule is currently disabled.
Status=I
Action=B
Recipient=nothing@something.com
[R]
; Emails with viagra, lender etc. in the subject line will be blocked.
Status=A
Action=B
Pattern=("*viagra*" or "*lender*" or "*bulk*mail*")
NewSender = JUNK_%S%
SearchTarget=S
[R]
```

1.11.1 Processing of saved messages (SMTPFlush)

If you decide to save the questionable messages using the routinglist.txt file (Action=S), the messages get saved in the spool folder. An accompanied program smtpflush.exe is launched at desired interval. The SMTP flush program reads each saved message and resubmits to SMTP Trap for processing. The idea is that once you decide what to do with those messages, you can change your routing list, and the messages get processed as per the new rules. The INI parameter, CacheReloadFrequency, decides how frequently the configuration files will be reloaded by SMTP Trap in its cache tables. The SMTPFlushFrequency parameter decides how frequently the SMTP Flush program will be launched. Any changes made to the INI parameters require restarting the SMTP Trap program.

1.11.2 Routing rules conflict handling

SMTP Trap scans the routing rules from the top until a match is found. Once a match is found, other rules are ignored. For conflicting rules, only the first one gets processed.

1.11.3 Routing rules syntax errors logging

SMTP Trap logs the syntax error in the log file if it encounters a problem parsing the routing rules (mismatched quotes, parenthesis etc.)

Running the program

1.12 From the DOS prompt or batch file

SMTP Trap can run directly from the DOS prompt or a batch file. Use the following command to run the program from the DOS prompt:

smtptrap /debug

Note that you can run this command even when the SMTP Trap is installed to run as a service. As long as the service is not running, the program can be run from the DOS prompt using the command described above.

1.13 As Windows service

If you installed the program to run as Windows service, use the service control panel to start the service or use the NET START command from the DOS prompt to start the SMTP Trap service.

SMTP Trap logs

SMTP Trap creates a new log file each day. The log file names follow this convention:

logYYYYMMDD.txt

Where YYYYMMDD reflects the date.

The log file gets created at the location, specified by LogFilePath, the INI parameter.

A typical log file looks like the following:

```
00:00:07 Connection from 218.98.64.62!
00:00:07 Oldest thread found: 488 (age: 501)
00:00:07 Connection accepted! (thread count: 3)
00:00:11 Connection from 69.140.129.3!
00:00:11 Oldest thread found: 488 (age: 505)
00:00:11 Connection accepted! (thread count: 4)
00:00:11 client ip was reverse resolved to:
pcp04423446pcs.nrockv01.md.comcast.net
00:00:12 IP: 69.140.129.3 (pcp04423446pcs.nrockv01.md.comcast.net) Sender:
xyz@flashcom.net MX:1 Rev:1 LDom:1 ACCEPTED
00:00:12 Routing check for recipient rffyli@jagat.com: save_bit=0 relay_bit=0
block_bit=0
00:00:16 client ip was not reverse resolved!
00:00:18 IP: 218.98.64.62 Sender: tr1772@ahoo.com MX:1 Rev:0 LDom:1 REJECTED
00:02:53 SMTPFlush launched!
00:02:59 Cache reloaded!
```

Accepted and rejected messages are logged in the following format:

00:00:18 IP: 218.98.64.62 Sender: tr1772@ahoo.com MX:1 Rev:0 LDom:1 REJECTED

MX:1 Represents that the MX lookup check passed for the sender.

Rev: 0 Represents that the reverse lookup check failed.

LDom: 1 Represents that the receiver of this email belonged to the local domain list.

Typically these three checks will either allow or reject an incoming message. Routing check related log messages are saved separately.

Appendix A. Junk Mail Primer

Junk mail, unsolicited e-mail, or Spam, are words every Internet user knows. There are legal interpretations of what and what may not be constituted as junk mail. Should the e-mail sent by your friend without permission, be considered, a junk e-mail? All the unsolicited e-mails are not necessarily bad. For example, I did purchase a foreign language learning CD after I read about it through an unsolicited e-mail. On the same token, breast enlargement e-mail will definitely go into my trash bin, for many reasons (One of them being that I am a male). These types of questions make the distinctions very hard. Yet, somehow, we have a look at the subject line and we know what it is. No one can argue that the junk e-mailers have every right to send what they send. There is this issue of responsibility, which fades away in the virtual world of e-mail. If a business or a person sends the junk mail using U.S. mail, it incurs some expense. The expense makes them responsible. They target the segment only what they think will work. However, the cost of sending e-mails is nothing more than your monthly charges to your ISP. This has made some of these folks very irresponsible. As a result, yours and my mailboxes are flooded with messages we don't want to wish to see. Folks are so scared with the junk e-mails that they are changing e-mail addresses every so often. I consider my email address like my phone number and I don't think that I have to change my phone number every month just because I am getting too many unwanted phone calls. You have as much right to stop the Spam as the spammers do to send them.

How do they do it?

Spam has become a big business. There is technology behind it. A spammer has to make sure the following:

1. Hide their foot prints
   This usually means use a phony return address or sometimes a valid return address, belonging to someone else.
2. Don't make use of your ISP's mail server, because most ISPs, like normal people don't want to be bothered with irate recipients.

The Internet e-mail uses a protocol called SMTP. The SMTP stands for Simple Mail Transfer Protocol. As the name implies, the SMTP protocol is really simple and straight forward. The designers of the protocol were looking for functionality at that time. An SMTP server is much like a U.S. Post office mailbox. Consider the e-mail, a letter, which anyone can drop in the mailbox, and U.S. Mail will deliver the letter to where it is intended to go. In the e-mail world, you don't have to put stamps on the letter. Like the mailbox analogy, it does not matter to the SMTP server, who you are and where the message is going. In the earlier days of the Internet, the spammers use any mail server, which will accept their bulk messages. Soon, the ISPs and the mail administrators realized this conspiracy, and took actions. Their actions were analogues to moving the post office mailbox inside the building. This way, only the authorized people are allowed to drop the letter in the mailbox. Any mail server, which is left out in the open these days, is termed as "open relay", and is black listed by many watchdog agencies. Eventually, the spammers felt the squeeze, and went back to the drawing boards. The mechanism they came up, is still used by most of the bulk e-mail programs.

To understand their method we must first understand how the e-mails get routed on the Internet. Everything at the Internet level works with IP addresses. Similar to human analogy, an IP address is a unique address for a computer. Without an IP address a message could not be delivered. Since the IP addresses are just numbers and we humans are not very good at remembering large numbers, we have delegated this task of converting the names to IP

addresses to computers. These computers are called DNS (Domain Name Servers). A domain is a name we can relate to, like yahoo.com etc. The domains are registered by many agencies. These agencies (also known as registrars) maintain the owners' information as well as the IP addresses of the DNS', which contain information about the domain. Whenever a request originates to contact a server under a domain (www.yahoo.com for example), the request first goes to its registrar. The registrar says, I do not where this server is, however, I could tell you who has the information about this server. So, the registrar points the request to the DNS server(s) for the domain. The DNS for the domain converts www into an IP address and informs the sender of this address. This mechanism is called name resolution (Conversion of name into an IP address). Apart from name resolution, the DNS servers contain a special entry in their tables for the mail servers responsible for the domain. This special entry or record is called Mail Exchange (MX) record. When queried for an MX record, a DNS server usually returns the IP address(s) of the mail server(s) responsible for a domain.

Sophisticated bulk e-mail programs actually query the DNS for every recipient's domain and deliver the message to the mail server responsible for the recipient's domain. Since the mail message is designated for a recipient within it's domain, the mail server gladly accepts the message. Note that the same mechanism is used when you use your ISP's mail server to send a mail to anyone outside your domain. In other words, the recipient's mail server does not really differentiate whether a valid mail server, or a bulk e-mail program is sending a message. The spammers, all around the world use this little loophole and exploit it.

Junk Mail Filters

It is actually not hard to eliminate the junk e-mails. The problem is that as soon as you implement the rules, the good guys get eliminated too. As a result most administrators do not bother implementing them or keep them rather loose.

The following section covers different mechanism you can adopt and their pit falls:

1. Block the sender by IP address. This is a very ineffective way of suppressing junk email. The good spam artist tries to hide his footprints. Most of them use dial-up connections with dynamic Ips. Blocking those Ips permanently is not fair to the people who may get those addresses subsequently. I have also seen when overzealous system administrators block the whole class C of addresses, when the spam originates from fixed IP collocated servers. This is also unfair to other companies who may be collocating at the same location and sharing the same class C. There are websites who claim to be watch dog for spams and maintain list of blocked IP addresses. If this technique is used, it should be used with caution and with fairness to everyone.

2. Do not accept any email, if it does not come from a valid mail server.
   This is easy. Check the name of the sending server. From the name, derive the domain. Determine the IP addresses of all the mail servers for that domain, if this server's IP address is one of those IP addresses then allow, else reject the message. Most of the small domains will be able to come in using this technique. However, big domains like hotmail, yahoo etc. use outbound servers, which are not listed as valid mail servers under their domains. Being large domains, they have so many of them.

3. Do not allow anyone who cannot be reverse resolved.
   The mechanism of reverse resolution is just the opposite of name resolution. A reverse resolution is conversion of an IP address to a name. Many spammers disguise themselves as someone else by putting a bogus name for the mail server. Many mail servers will look at the name and allow them in because they think they are valid (based upon a name they trust). However, performing a reverse resolution will reveal that the sender is not who it claims to be. The pit fall for this check is that many administrators

are either too lazy to do it or are simply ignorant about it. As a result, **good guys** get eliminated again.

4. Eliminate by sender, subject, and the contents.
   This is probably the most used, yet most ineffective way to eliminate the spam. Most spammers use changing return addresses. Most likely, they will never use the same identity to send the same spam. Eliminating spam by subject and contents has the risk of eliminating e-mails, which are valid and genuine.

A good strategy to fight the Spam is to utilize a combination of the techniques above and find the one, which works for you.