Spam Filtering in a Small Business Environment, a Case Study

Richard Snow

Spam Filtering in a Small Business Environment, a Case Study

GIAC GSEC Certification Practical
Version 1.4b, Option 2
16 June 2003

Richard L. Snow

Table of Contents

Abstract

This case study describes the process of researching and implementing a filter for email "SPAM" in an organization of modest size, running Microsoft Exchange 5.5 and IMC. At the time of the implementation in Fall 2002 there were few commercial software products available to address this issue in a Microsoft environment. While open source approaches to the problem were fairly mature, the organization does not have expertise with open source software so a commercial solution was desired.

The article outlines the effect of SPAM in our environment, the process we went through in selecting and installing an email filtering system and the resulting situation today. It discusses the network environment in place before the implementation and compares native capabilities in MS Exchange 5.5 against our requirements. Based on needs that were not addressed in Exchange IMS, I discuss why we chose commercial solutions: X-Wall by Data Enter, and SpamAssassin by Deersoft and how they fit in our environment. The philosophy used to make this decision is examined, along with the set up and installation of our system. The results of the final system setup are discussed along with "lessons learned".

Before: Effects of SPAM in our environment

We are a non-profit organization with 35 computer users, running Microsoft Exchange 5.5 and Internet Mail Service (Exchange IMS). Managing email service has been difficult because of various issues – a lack of in-house computer expertise, business failure of a series of Internet Service Providers and the onslaught of email viruses.

One of the first things that the executives requested when I joined the company was to control unsolicited commercial email traffic, or spam. Although the organization came late to the email world, many members of our senior staff use internet email list-serves. These email messages are posted on web archives, and spammers use automated tools to add the email addresses to their address lists.

As of March 2002, one executive was reporting that he received up to 40 spam emails per day. By looking at his email account it became apparent that these were not merely bulk email messages (advertising) – but were clearly unsolicited. The organization's president received many spam messages and assigned his executive assistant to screen his email for him. One department head requested to change his email address to avoid the spam messages, which was effective but inconvenient. Another executive was chagrined when she began receiving spam messages with "adult" content. The organization has a strict acceptable use policy - forbidding messages with pornographic content, so this raised questions about the user's responsibility as a recipient of "unacceptable" messages.

These comments highlighted three significant concerns about email SPAM in our organization: Lost Productivity, the Annoyance Factor, and how spam relates to our Acceptable Use Policy (AUP).

Lost Productivity:

The executives and their staff were forced to spend a significant amount of time deleting junk email from their accounts, clearly a costly waste of staff time.

A recent Ferris Research report determined that for corporate accounts, spam makes up 15 – 20 percent of all email.[1] Research analyst Marten Nelson was quoted as saying: [Email spam costs] "about $10 per user per month." By this metric, spam costs my organization up to $350 a month in lost productivity.

Annoyance Factor:

Using email as a business communications tool is relatively new to our organization. The annoyance caused by excessive and offensive spam messages slows down the acceptance of email communication.

Confusion Regarding our Acceptable Use Policy (AUP):

Users are worried that these unsolicited messages are in violation of our AUP.

During: Addressing the Issues

At the beginning I did some research into how to address the SPAM issue:

- Review the company policy to be sure I could scan email.
- Check that our mail server is not being co-opted to relay spam messages.
- Assure users that we are aware of the problem and aware that they are not the source of the "unacceptable" email messages.
- Learn about the spam filtering options.

Looking into the spam filtering options in detail involved a trial and error process:

- Try out client side filtering software (Outlook and SpamAssassin).
- Find out what we can do with our existing tools (Exchange 5.5, Outlook 2000, NAV for Exchange and the Norton Antivirus Gateway products.

A Managerial Moment: Policies

To get started it was time for a "managerial moment". We needed to have a policy to determine how intensively we would filter our incoming (and potentially our outgoing) mail. It was important to engage with the organization's management and ensure that this project would align the new rules with it's wishes. A spam filter is in essence a filter. Therefore you need to be on firm ground when it comes to implementing a company policy regarding email. How critical is mail delivery and how acceptable is it to lose an occasional message? If no messages may be lost, then the cost of reviewing each message must be considered.

As the administrator, you will require the ability to drill down into the individual emails that your users are sending. It is important that this is clear in your company policy. Does your company policy require filtering on specific terms? Perhaps you have a need to prevent disclosure of information related to a hush-hush project. What would the search terms be if you wanted to filter content? And who can authorize you to filter email with those search terms?

Mail Relaying and Co-opted Servers:

Email spammers may use open mail relays to deliver bulk email. An open mail relay is a mail server that will forward messages from anyone to anyone. Before the popularization of the internet, it was common to configure a mail server to relay messages without authentication. As the network has grown, so too has our responsibility to know to whom we are providing services. I checked that our Exchange IMS was not configured as an open mail relay. This is to prevent spammers from using our mail server to forward messages to others.

To do this, first I needed to upgrade MS Exchange 5.5 to the most recent service pack, Service Pack 4. Because of the common vulnerabilities in email server software, it is critical to maintain software at current revisions. The administrator should also maintain an awareness of new issues related to any software at your site. This can be done by subscribing to email lists such as those provided by Bugtraq, SANS, CERT and your email server software vendor, in this case Microsoft.

Microsoft Knowledge Base Article –199656, "XIMS: How to Stop Spam Mail Messages from Using IMS Relay Agent," is a good place to start in configuring IMS to avoid relaying.[2]

Mail relaying is configured in Exchange 5.5's Internet Mail Service, on the Routing tab. And turning it off is easy on the Internet Mail Service (IMS) Properties screen:

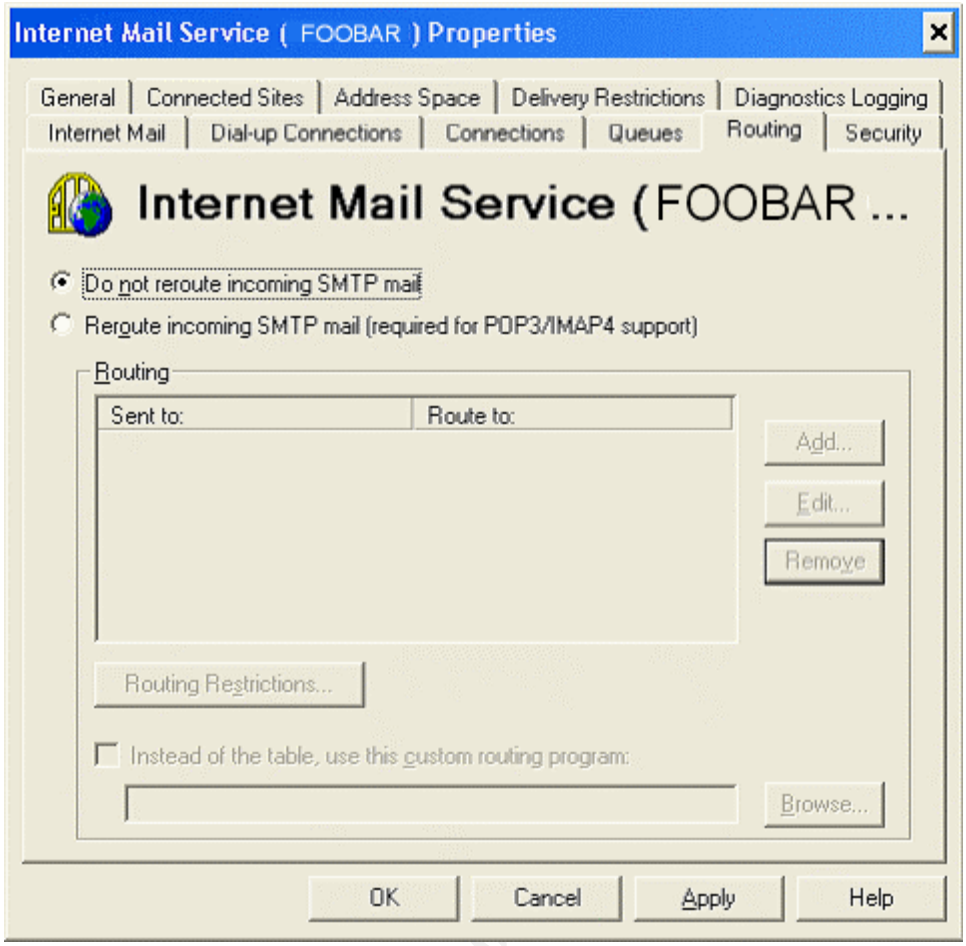

Figure 1: Turning off Mail Relaying

Any time you make changes in the IMS, you need to stop and restart the service in the service's control panel for the changes to take effect. This is similar to restarting the sendmail daemon after reconfiguring sendmail on a unix system.

Often you will need to direct mail from one email domain into another. For example you may have registered more than one domain, so you need to direct the email from username@mydomain.com to username@mydomain.org. In our organization we do this by setting up the Routing tab as follows:

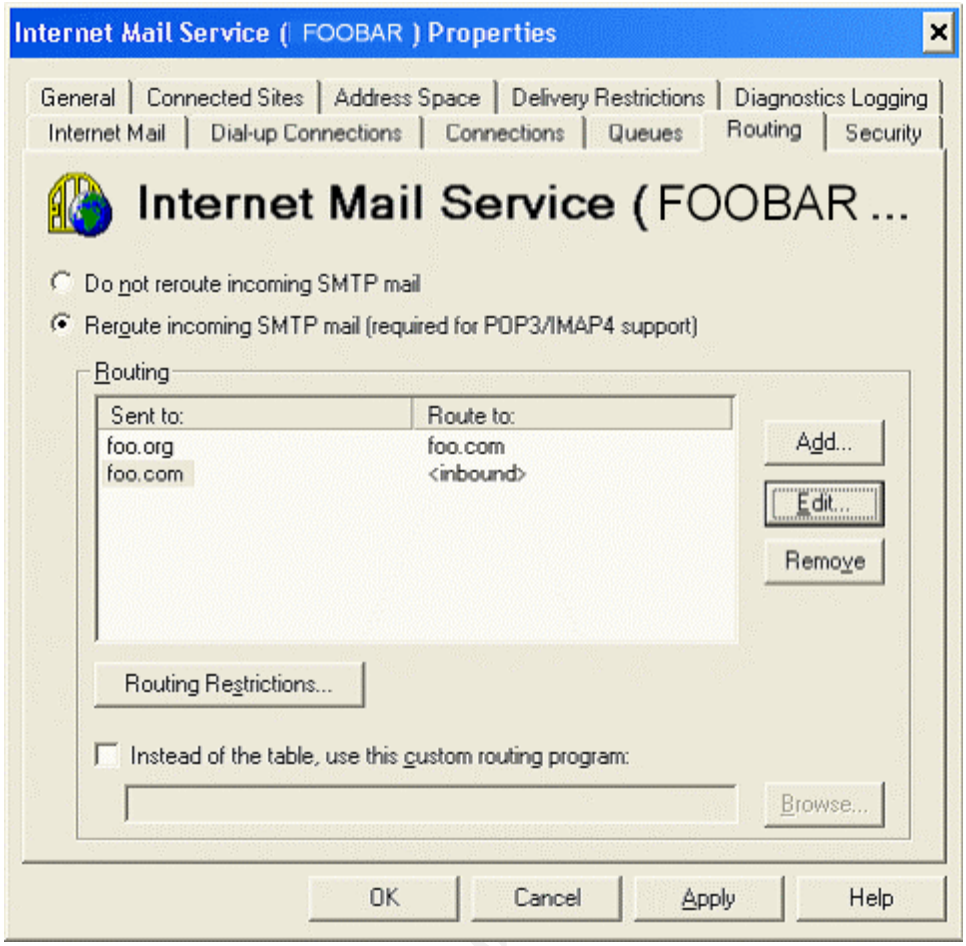

Figure 2: Routing Mail Between Domains

We are both a commercial operation and a non-profit. Our users *receive* mail under both domains. Their *outgoing* mail comes from their account with the '.com' address.

Once you turn on routing, restrictions on what hosts can relay messages through the server can be specified in the Routing Restrictions. In the example below the restriction that only hosts that authenticate to the server should be able to relay mail has been enabled.

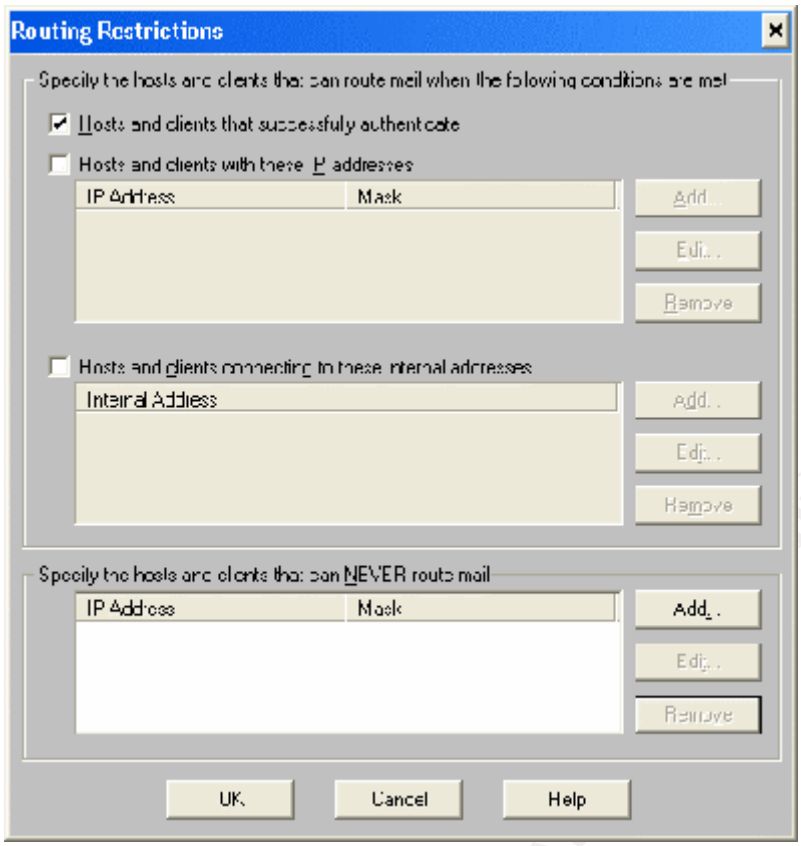

Figure 3: Exchange IMC - Routing Restrictions

Columnists Ben Schorr and Jim McBee of .NET magazine[3] take the relaying configuration one step further. They suggest leaving mail routing enabled on the Routing tab. Instead click Routing Restrictions and leave the entries in this window empty.

By doing this, the list of hosts allowed to relay is empty - so relaying is effectively disabled. What is not immediately obvious is that this configuration will not cause Exchange to create a non-delivery report (NDR). The server is under less of a burden because it does not create the NDR, and it will not create a pile of messages which are then delivered to the email's "From" address. This is particularly useful because the From address is often forged by spammers. In an extreme case we could find ourselves delivering a blizzard of NDRs to an innocent mail server somewhere else!

As you can see in Figure 3, there are other routing restrictions that may be applied, such as:

- Allow hosts with specific IP addresses to relay messages
- Restrict mail relaying to a particular interface on a multi-homed machine
  (Note that you need to turn off IP Forwarding for this to take effect)
- Block specific IP addresses from relaying messages

Anyone using the IMS as a mail exchanger should familiarize themselves with the articles in Microsoft's knowledgebase regarding mail relaying. A current search shows 25 articles related to relaying issues. Some of these articles describe scenarios in which the restrictions above are defeated even when they are set up properly!

Microsoft Knowledge Base Article – 264330, XFOR: Internet Mail Service May Relay Messages Despite Restrictions[4] is a discussion of how routing restrictions can be circumvented when one email domain is routed into another (as shown above). The spammer would need to know beforehand which email domains were routed so perhaps this seems unlikely. An upgrade to Service Pack 4 for Exchange 5.5 patches the vulnerability.

After installing Service Pack 4 and checking the configuration, I was comfortable that we were not providing an open mail relay for spammers to use.

Client Side Filtering:

Heinz Tschabitscher wrote a series of articles on About.com giving steps to use Microsoft Outlook's rules processing to filter spam messages.[5]

Outlook's rules are effective in fighting spam, but these rules run on each client machine. Also, they run only when the user logs in to download email. Our organization has the luxury of having a full time system administrator, and as a result we can manage and install a filter at the server level. Client side filtering has several drawbacks for us, but Client-Side filtering may be the best choice for smaller organizations without an IT Manager on staff.

Drawbacks of Client Side Filtering

- Requires custom rules configuration on each machine – requiring separate setup and monitoring.
- Some of Outlook's rules are 'local' on the client and others run on the server. They will not run the same way when the user logs in from different machines.
- The rules take time to run when the user logs in to Outlook.
- The spam email messages would be using up our internal network bandwidth.
- Client side filtering does not prevent risqué messages from being delivered to the user's mailbox since the filtering actions take place after delivery.

As a pilot project, I decided to install a commercial spam filter for the executive with the most extreme spam problem. I chose SpamAssassin by Deersoft [6]

SpamAssassin works on the individual user's system by filtering his email and putting the spam messages into a special mailbox for review. After some initial training by the end user, the system is very effective at reducing spam emails.

When real messages are captured by mistake, the user can pull them out of his special email box before deleting the spam messages.

The user is very pleased with the results. However, at$40 per user this is an expensive solution to apply to our entire system. It is most useful for heavy email users that are receiving a lot of spam. The filtering is very effective and the solution requires very little intervention by the system administrator. Overall there are still disadvantages of this approach:

Drawbacks of a Client Side SPAM filtering application

- The spam messages are still taking up bandwidth on our internal network.
- The user wastes a small amount of time checking the messages marked for deletion.
- Yet another software package to buy, install and maintain.

In our environment, client side filtering is too expensive and requires too much involvement from the user. However, SpamAssassin is a very good solution, and we will continue to use it for the one user who has an extreme spam problem. In that situation, an alternative could be to change the user's email address. One user requested an address change, which was successful in eliminating his spam problem – for the time being. However as a large scale solution, it would be inefficient to change users' email addresses frequently.

Using Existing Tools (Exchange 5.5 IMS, Norton Antivirus for Gateways):

Back at the server, I investigated native filtering capability in Exchange 5.5's Internet Mail Service. We had upgraded from Exchange 5.5 Service Pack 2 and the filtering capability was limited and did not seem to work as advertised. The filtering is much more sophisticated in Service Pack 4.

SP4 can support a list of blocked IP addresses. I tried implementing a static list of open mail relays from whom I knew I was receiving junk email. To compile the list, I went through the headers of junk email messages to find the mail relay that received the message from the spammer's machine. As you can imagine this was quite time consuming. Also, the user interface for entering these numbers is very cumbersome. There is no way to enter a list of numbers all at once.

In the IMS, under the Connections tab you can select Specify by Host or Message Filtering:

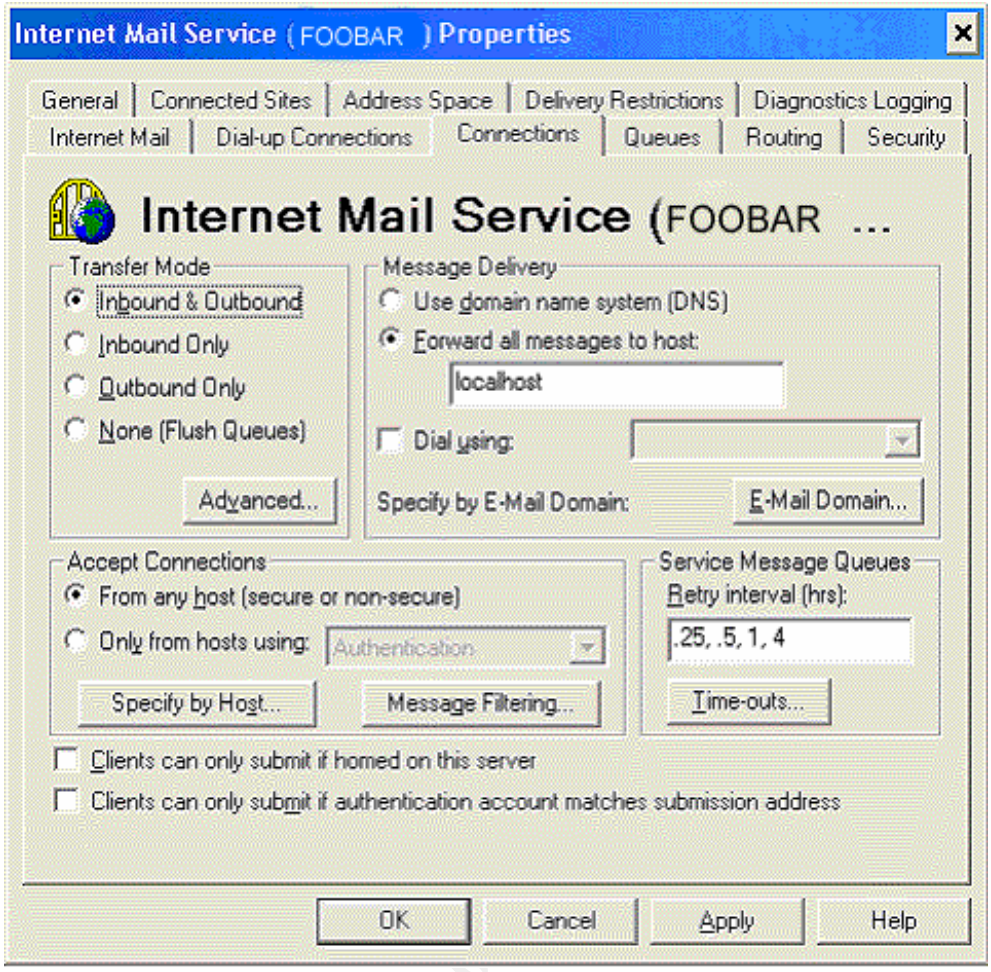

Figure 4: Connections Tab Options

Specify by Host (Figure 5) allows you to deny a connection from a specific host. This applies to a host that is trying to open an SMTP connection to our server to deliver an email message. In many cases this would be coming from a mail relay, but systems could be configured to deliver directly, for example, a mail server in a corporation, a unix workstation or a Linux box.

Here we are not accepting a connection at all. This is different from denying mail relaying on the Routing Restrictions tab. When we enter a server on the Routing Restrictions tab in the “Specify the hosts and clients that can NEVER route mail” field – we are preventing our server from *forwarding* a message that came in from that IP Address because the message is not directed to recipients on our server, and we disallow routing for a specific IP Address. In the “Specify Hosts” tab – we configure machines that are not even allowed to open a connection!

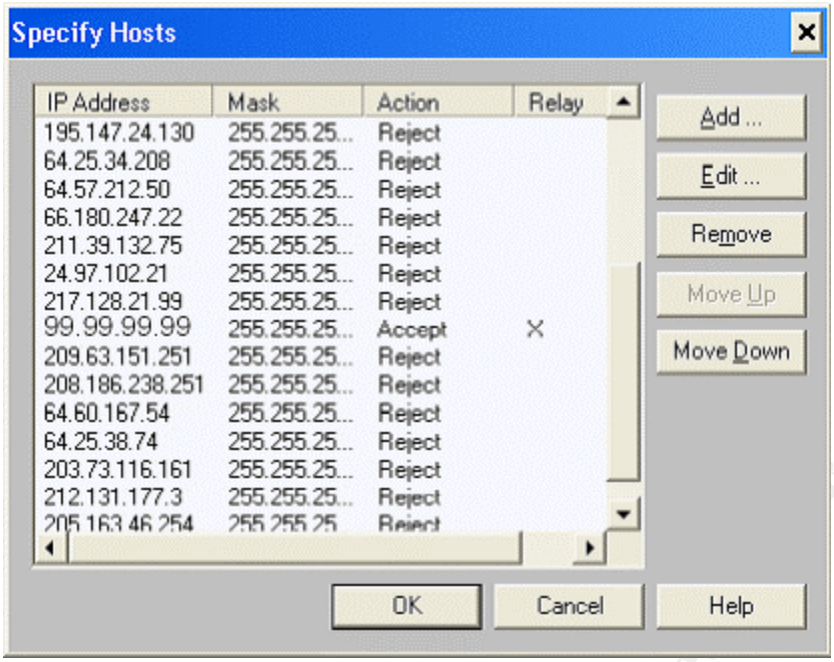

Figure 5: The Specify Hosts List

You can change the action on specific hosts, which is handy. You may have a mail spooling service at your ISP. These are typically configured as the secondary MX record for your domain. Should your primary mail server fail this spooler will queue your incoming mail until your primary server is returned to service. However, a mail spooler is a mail relay.

Obviously you would want to receive your email from this machine. After a power outage or other failure, legitimate email goes to the ISP's system to be queued (or spooled) and is later delivered to your server.

By going through the headers of spam email messages, I discovered that when I deny them access directly – they try the second mail exchanger (MX record) listed in DNS for our domain. This is insidious! Now I am receiving the spam from my ISPs spooling server, but I can't tell who is "knocking at the door"! The secondary MX record configuration defeats filtering by IP addresses.

I found myself putting the mail server "99.99.99.99" on the list of denied addresses and then enabling it once again when I discovered the error.

Pressing the edit button brings up a configuration screen for the selected entry in the Specify Hosts screen (Figure 6).

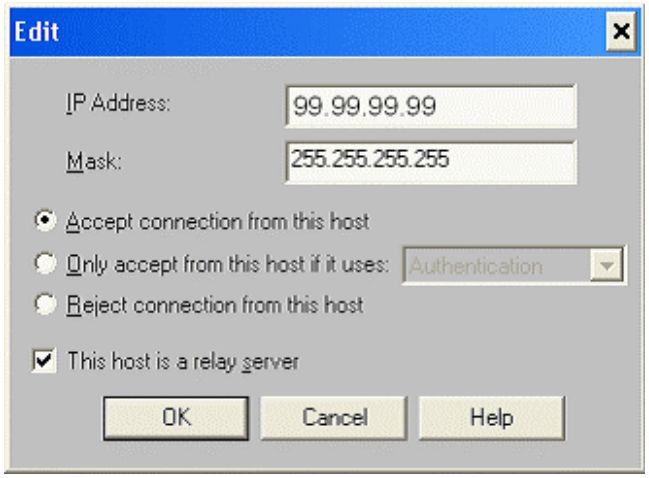

Figure 6: Actions Applied to a Specific Host

Understanding this side effect of the mail spooling server, our ISP's mail spooling service became a liability. I cancelled the ISP's mail spooling service and we were then able to filter effectively based on the IP address of the machine attempting to open an SMTP connection to us.

While the Specify Hosts screen could be useful for a specific configuration issue, in the end I found that it is not a viable option for filtering spam because the list of open mail relays is constantly changing. For that reason various schemes have been developed to deny message transfers to machines based on a dynamic list of mail relays. It just isn't practical to hand configure this list of bad apples.

Real Time Back Hole Lists

Real Time Black Hole Lists or RBL's are systems that use various criteria to maintain a list of sources of email spam. These servers provide a DNS lookup against a list of known sources of spam. When a message transfer is initiated, the server on the receiving end will double check to see if the transmitting server is "blacklisted". The transfer can be denied, or appropriate filtering can be applied to the message if the sending server is listed on the RBL list.

Exchange 5.5 doesn't have the built in ability to work with these Black Hole lists. This is an area of software development where the Open Source movement is years ahead of the commercial market. While Microsoft Exchange 2003 reportedly integrates RBL Lists, efforts in the field are moving beyond Black Hole lists specifically because of their side effect of blocking legitimate email.

The IMS offers a lot of flexibility in dealing with known IP addresses, and these features are useful for filtering in specific circumstances but they are impractical as a SPAM fighting tool. The list of IP addresses for open mail relays, for example, is quite extensive and dynamic.

Another feature of Exchange 5.5 SP4 is a message filtering option. Contrary to what you might expect, this does not filter message content, but filters delivery based on user and/or domain (Figure 7).

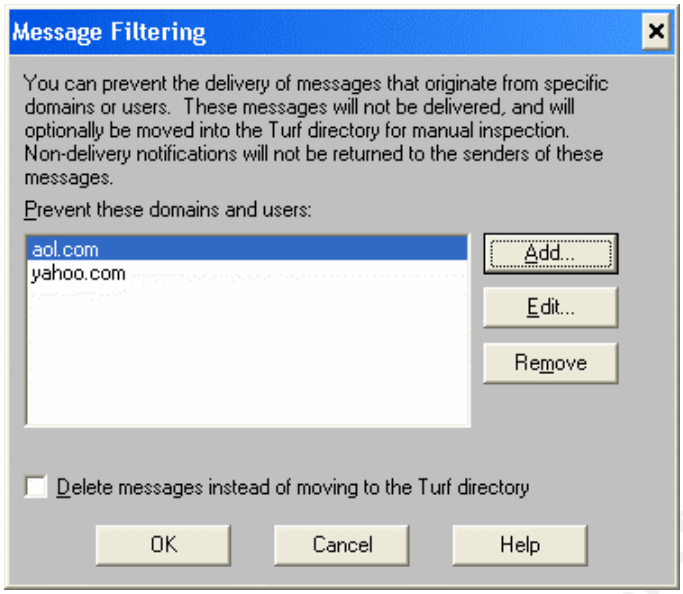

Figure 7: Message Filtering – Based on Address

Again, this is too broad a brush to be useful for spam filtering. Many of the email addresses in spam messages are forged. A sysadmin would have to be clairvoyant to put in the username and address for a spam email sender before it is received. Message filtering should be reserved for special situations.

Using Exchange IMS as a SPAM filter

Exchange IMS has some very useful filtering capability. It would be impractical as a filter on open mail relays or known spam sources because the list of sources of spam is large and constantly changing. It is not easy to input new addresses in IMS and it does not have provisions to automate the process. The message filtering feature is also more useful for dealing with a specific external sender than the universe of spammers.

Filtering Spam at the Server

From an architecture point of view, there are different places that a software application could be installed to do email filtering. One approach is to capture email at the SMTP server – before it is delivered to the Exchange system. Symantec's Norton Antivirus for MS Exchange (NAVMSE) has spam filtering in the gateway virus scanner. Vendors such as SpamCop and others can provide custom gateway machines that filter email for spam.

It works like this: When a sender delivers a message to our system, the firewall receives a message on the standard SMTP port 25 at a published internet IP

address. The firewall is configured to deliver that message to an internal server on port 25. The Exchange IMS is a mail gateway between SMTP and the proprietary Exchange email system. It picks up the call on port 25, and delivers the message to Exchange. To add a spam filter, a "gateway" product would be installed between the firewall and the Exchange IMS. The gateway is configured to pick up messages coming in from the firewall; it does its processing, and then delivers the messages to the IMS. The IMS is itself a gateway, but does not have this advanced filtering capability.

Another way that this could be accomplished would be to have the filtering system hook into API's written into Microsoft Exchange. Microsoft has enhanced the API's (Application Programming Interfaces) available in Exchange 2000, and this is the reason that some commercial products are tied to Exchange 2000. Even greater functionality is promised for Exchange 2003.

Commercial Anti-Spam Products

In the fall of 2002 there were products available for large corporate installations, there were a few products available for Exchange 2000, there were ASP approaches (where my mail would be delivered through an external company or an external company would provide filtering equipment on site), and there was Open Source software.

Commercial software was clearly a year or more behind Open Source software, but we are a Microsoft shop. I had to consider my backup system administrators who would service the installation of a Linux box or a custom solution at those times I am out of the office.

Commercial software would hopefully come with robust support options. Many of the commercial approaches were too expensive for a small organization (ranging from $30 per year – per user to $15,000 appliances) and very few support Exchange 5.5.

Using an ASP

Using an external service provider sounded like a good option because they would take on the brunt of the management. However the recurring cost of $30 (or more) per user per year is daunting on a tight budget. If I direct all my email through an ASP, I would pay $1050 a year for filtering. Then I would need to train my users to look at a web email box periodically to check for false positives. The cost, additional risk of relying on another vendor, and training issues are all drawbacks. Clearly what was needed was a small software application that would run in the Microsoft environment.

Gateway Products

The Norton Antivirus Gateway product is set up as a standalone gateway and does not rely on Exchange APIs. Unfortunately, Symantec's engineers revealed

that the NAV gateway product is resource intensive. It would require its own hardware. Since I do not have the space or the spare hardware to run a separate system I had to look for other options.

Defining Requirements

Knowing that we were unable to use our existing software to accomplish the task, and that we wanted the filtering to occur before the messages are received by MS Exchange, I was ready to draw up a list of requirements:

Requirements for a Spam Filter:

- SPAM filtering using Blacklists, Bayesian filtering, and content filtering
- Low cost (less than $1000)
- Purchase rather than service provider to avoid large recurring costs
- Works reliably with Exchange 5.5
- Filters messages during SMTP transfer (so they don't travel over our internal network)
- Not Resource Intensive

Xwall by DataEnter[7]

Xwall impressed me in a number of ways. The most important is that it supports Exchange 5.5 while products from other vendors such as MacAfee do not. It includes all of the features I could implement using a Linux box, such as black hole lists, Bayesian filters, message content scanning, blocking attachments, and reassembling messages (to prevent hidden attachments). I was able to download the program and try it immediately – while larger manufacturers were not offering a 'try and buy' software download. It is comparatively inexpensive - $298 for a fully functional version. And it runs happily on a busy server – no need for new hardware!

As a mail gateway operating at the SMTP level, I needed to change the configuration in Exchange IMS so that I could have XWall pick up the messages coming in through the firewall. One common way to do this is to set up the internal mail exchanger (IMS) to listen on port 24 instead of port 25. My new gateway will pick up the messages on port 25 and deliver them to Exchange IMS on port 24. (In the unix world, this is similar to implementing the program SMAP from the TIS firewalls toolkit).

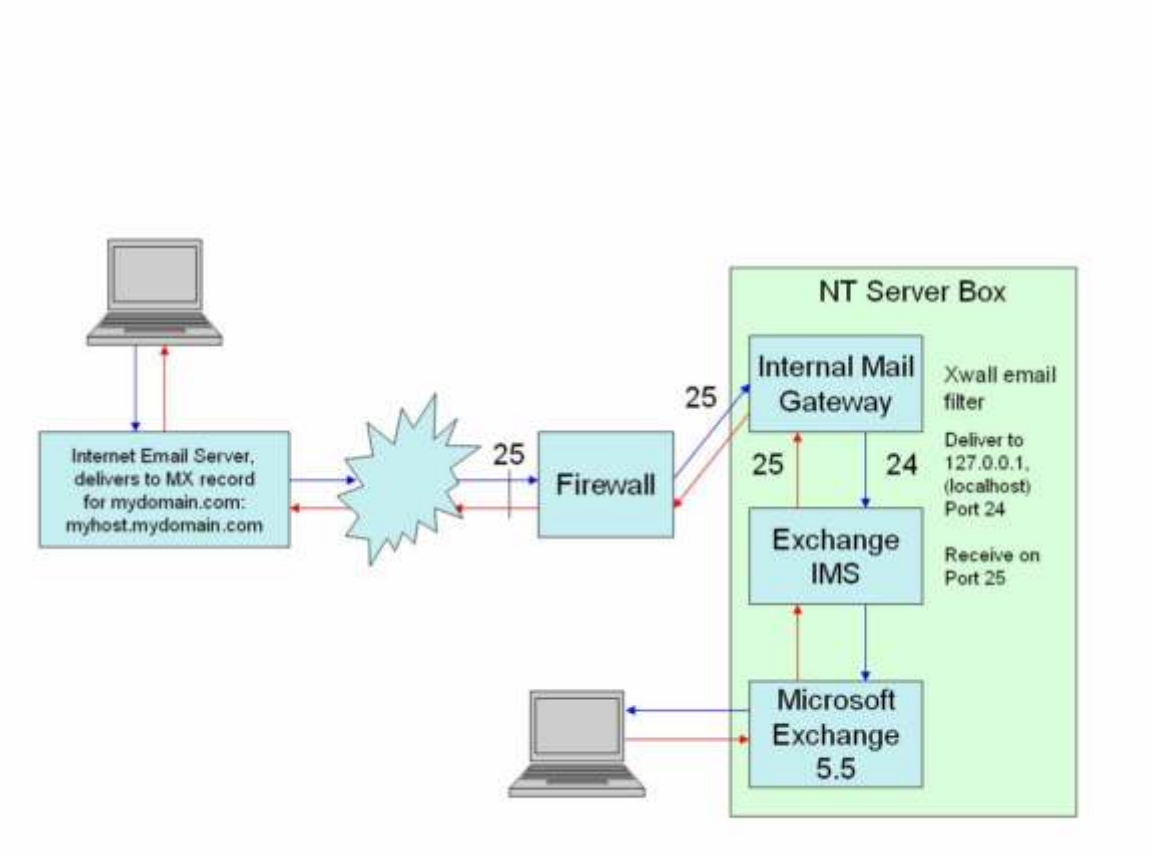

Figure 8: SMTP Configuration for Email Filtering Gateway (Note: Outbound filtering is optional)

The installation procedure for X-Wall on Exchange 5.5 is shown on DataEnter's website.[8]

> open the file services, usually located in C:\WINNT\system32\drivers\etc\SERVICES with Notepad or any other text editor. Locate the line smtp 25/tcp mail and change 25 to the port of your choice (use 24 if you are not sure which one you should use) and save the file.
> Restart the IMS (Internet Mail Service) of the Exchange server to bring the new settings into affect.

This process will shut down the mail gateway on port 25 during the installation, so for the duration – email will be queued by the sender.

Having changed the services file, and restarted IMS it's a good idea to verify that IMS is receiving email or port 24 by opening a telnet session to the Exchange server on port 24:

telnet someserver 24

And we see the server respond, so port 24 is ready:

220 someserver.somewhere.com ESMTP Server (Microsoft Exchange Internet Mail Service 5.5.2653.13) ready

After downloading and extracting the ZIP archive we can install the XWall gateway.

There are two options, either the gateway is installed as a Service under Windows NT type systems, or it may be run as a program. The program is called a Console. You run the Console from the Start menu, and XWall will begin to deliver messages. This is a handy way to get the system set up because you do not need to pore over log files to see error messages. Once the system is set up you can then make the server run as an NT service. At that point the service will start every time you boot NT, which is ideal for a mail gateway.

After extracting the program, launch setup and configure the basic options for the gateway, as described on Data Enter's website:
http://www.dataenter.co.at/doc/xwall.htm#inst

The setup program will prompt you for:

- The Postmaster's address
- The name or IP address of the Exchange server.
- The port that you have configured the Exchange IMS to receive SMTP messages on.
- The email domain that Exchange handles messages for.

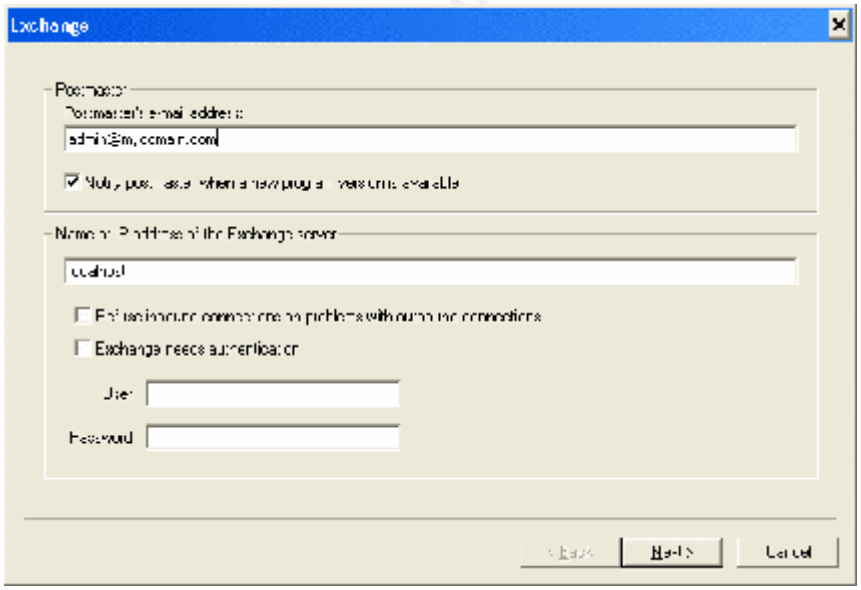

Figure 9: XWall Setup for Exchange

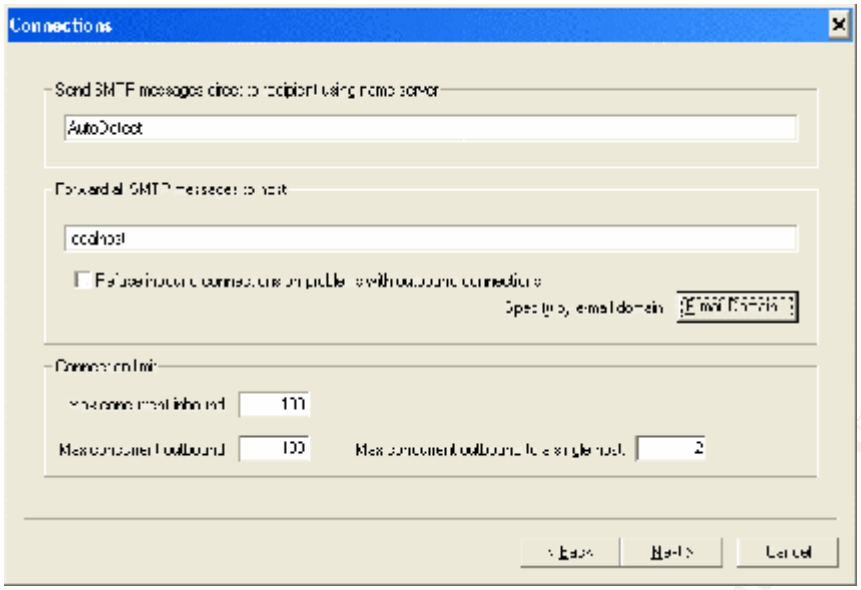

Figure 10: XWall Connections Setup

Note that you can configure different hosts to receive email for different domains.

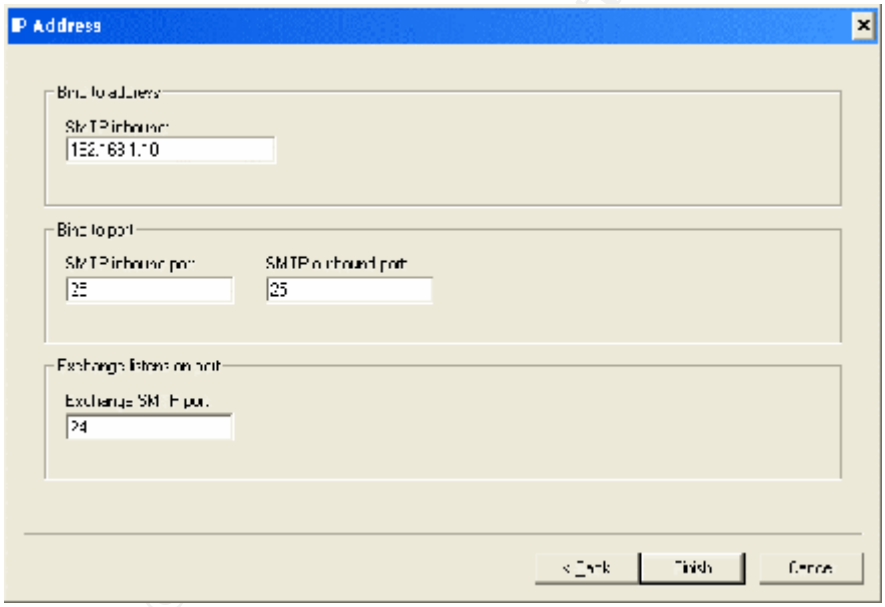

Figure 11: XWall IP Settings

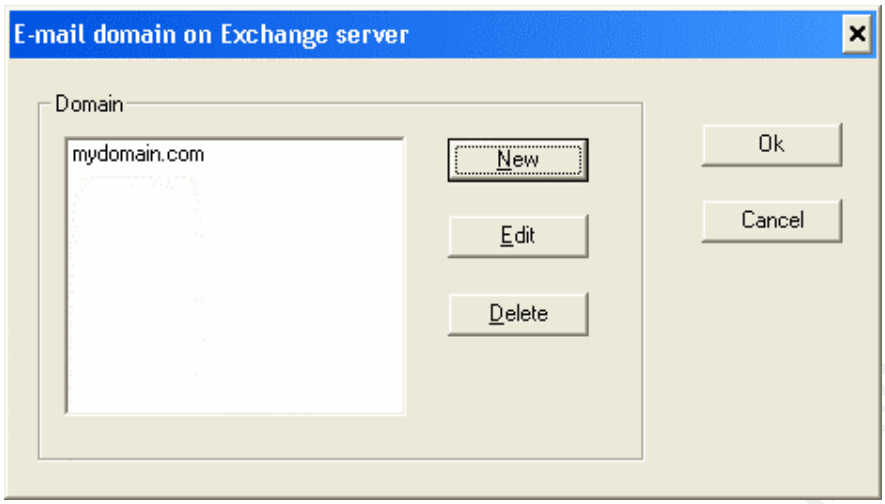

Figure 12: XWall Domain Settings

To test the program you can start up the Console program from the entry that was installed in your Start Menu.

When the program starts up, it will check for a virus scanner in the TEMP directory and in the C:\XWall directory by writing the EICAR test string to the disk.[9] If virus scanning is active in these directories XWall displays an error message on the console screen. Before using the program we need to set an exclusion in our antivirus program so that the Virus Scanner will not scan Xwall's temp directory, or the XWall installation directory: C:\Xwall.

At your site, you will need to understand how this interacts with your virus protection software. Does leaving this directory un-scanned create a new vulnerability for you? In our case the email is scanned within the Exchange data store, and all the workstations and the server have active virus protection.

For those who do not have a tailor made email antivirus program it is also possible to configure XWall to run a standard anti-virus program against received email. This would catch viruses coming in via SMTP, but it would not catch viruses within the Exchange message store, such as viruses sent within the company.

After setting up the exclusions in NAV the XWall console starts up smoothly (Figure 13).

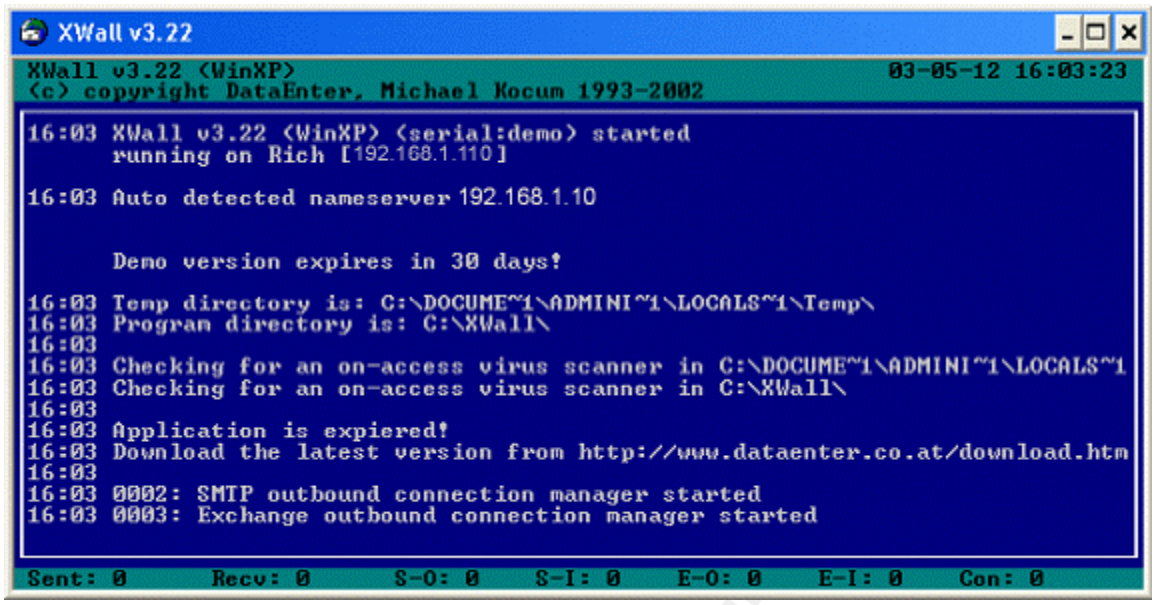

Figure 13: XWall Console

While the console is open, XWall will receive messages and forward them to the Exchange server.

Opening a telnet to port 25 on the server will test that Xwall is listening on port 25:

C:\>telnet someserver 25

220 someserver.somedomain.com ESMTP XWall v3.26

And there is our server 'someserver' talking back to us… So the server is now open for business. The next step in the process is to try sending email from an internet email account to a local Exchange account. Again, you can watch the console screen to debug the process, or go back through XWall's logs and the minimal logs provided by Exchange.

Configuring anti-spam features

Once the gateway is set up you need to have another 'managerial moment' to translate your policy on lost email into filters configured in the program. What is the sensitivity to lost email in your organization? I face a typical situation at my site: we don't want to lose any email, but we want to prevent SPAM.

To assist with this, there are different actions we can apply to a message that has fallen awry of the spam filter. Initially I recommend that you set XWall to mark the message subject and forward the message to the administrator. All the 'spam' messages that are received will be forwarded to the administrator. This allows the admin to monitor the messages being captured and build up a list of

exclusions over time. But we need to be careful because the side effect is that a greater burden of reviewing junk email now rests on the Administrator. Remember, part of the goal is to reduce staff time spent cleaning up mailboxes! It is very tempting to just delete all those junk emails, but with my configuration it seems that as many as 4 out of 100 messages marked as spam may in fact be desired by the end user. Obviously I can't just throw them away!

XWall is not purely an anti spam product. It handles a number of functions in one package, and includes options such as virus scanning, message rewriting, content scanning, using blacklists to prevent delivery, rewriting messages to prevent various hacks in attachments, checking for non-standard features such as Microsoft's TNEF formatting, and adding a disclaimer to all outbound emails.

It may be most practical to think of X-Wall as a policy gateway for email. It gives you more flexibility to implement a corporate email policy in an MS Exchange or Microsoft Mail environment.

X-Wall Settings

General Tab:

Set up logging and make sure to write a statistics file and an SMTP blocking file. It is critical that you keep logs to be able to debug message delivery problems. They may also allow you to justify the project to your management and determine your return on investment. We have chosen to rotate the logs every 60 days to conserve disk space. If I had the option to do this again, I would have chosen a larger time frame, like a year. The log files at my site are not so large.

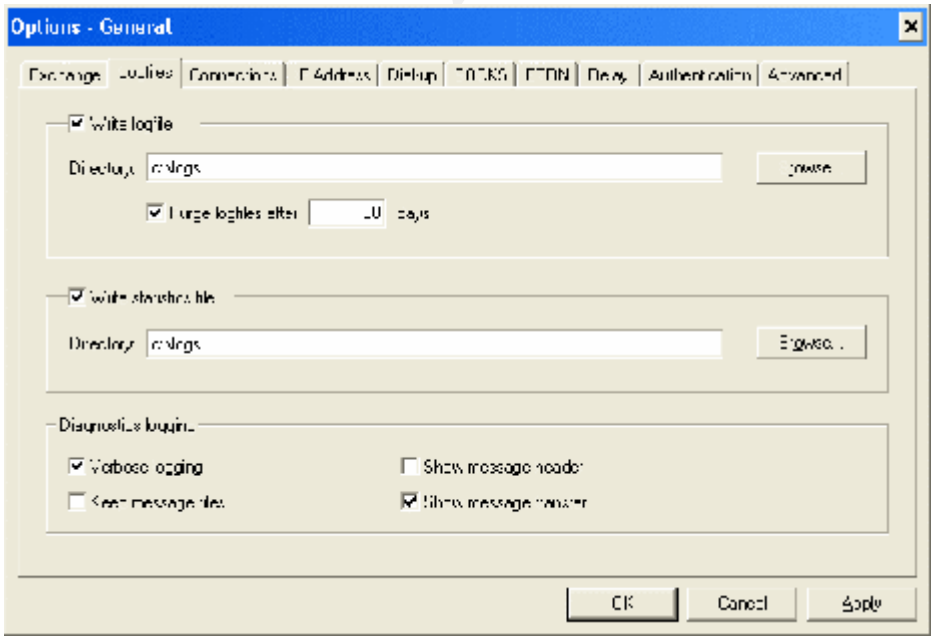

Figure 14: General Tab

Blocking vs. Filtering

When fighting email spam it is important to understand the difference between Blocking, and Filtering.

Blocking prevents messages from being received via SMTP. A lookup in a Blackhole list, rules regarding the mail exchanger, or other criteria may be used to determine whether to allow the message transfer to occur at all.

Filtering is applied after the message is received, and is often based on content.

Blocking:

Each of the tabs in the Blocking section can take some time to configure. I chose to Block attachments with a dot at the end of the filename, attachments with a double extension (ex. blah.jpg.scr) and attachments with a CLSID extension. A CLSID extension could be manipulated so that the system does not see the full extension of an executable attachment and as a result fool the system or the end user into launching an attachment containing malicious code. In my case, I am not blocking external attachments because I am concerned this could have the effect of breaking images in HTML emails.

I also Block a specific list of executeable attachments. In our environment there is no use for attachments that contain scripts, com and exe files. If legitimately needed, such attachments could still be zipped for sending. A nice feature of XWall is the “Add Unsafe” button (see Figure 16). This will put a list of common executeable extensions into the blocked attachments window. You should check over the list to ensure that you are not blocking required functionality! Taking time to understand each of these options is necessary to setting up your configuration.

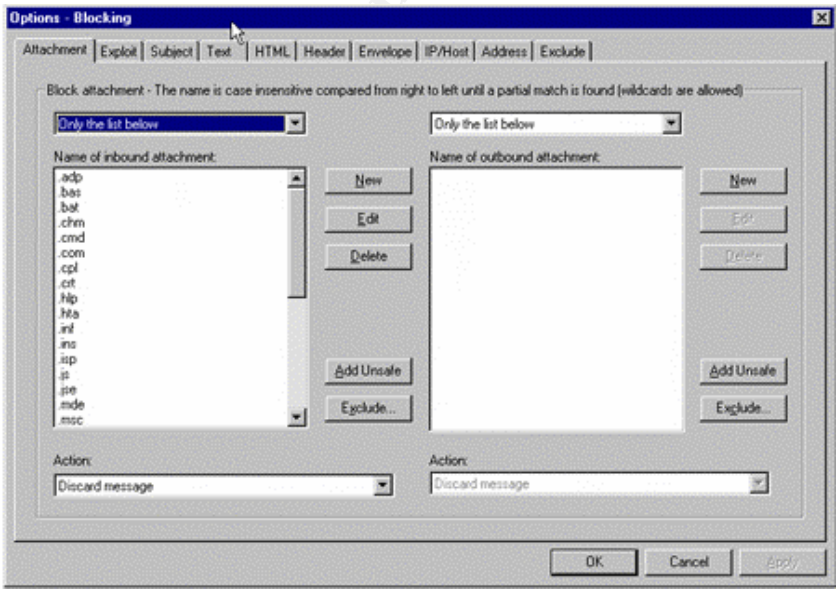

Figure 15: Configuring blocked attachment types

XWall can optionally be configured to filter outgoing mail as well as incoming mail. You might choose to filter content on outgoing mail, for example you could review outgoing messages that have specific text in the subject lines or message bodies. Perhaps you want to prevent an engineer from emailing technical information about a new product outside of the company. A list of terms could be entered to prevent email containing certain topics from leaving the company.

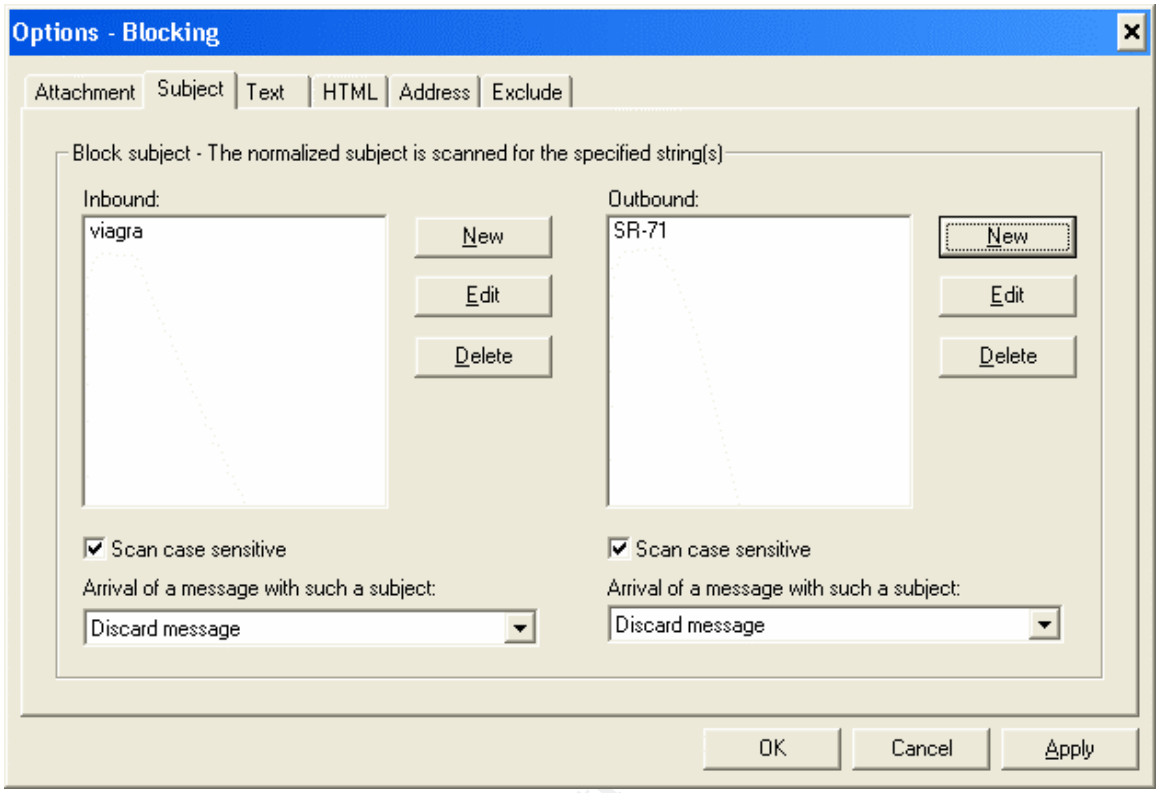

Figure 16: The Subject Tab

The Subject Tab:

The Subject Tab is used to configure inbound and outbound message blocking based on the message subject. Be aware that the text used will recognize regular expressions! I found this out the hard way. The Subject Tab is a very sharp scalpel – so use it sparingly. Don't set the action to "discard message" unless you have tested the configuration, set the program to forward the message to the administrator for review instead!

The Envelope Tab:

This tab allows you to check to see if the message was sent using blind carbon copy (BCC). You can also check if the message has a forged Mail From: address, and you can require that the sending host belongs to the sending domain. I have set these up to mark the message subject, because much legitimate commercial falls into these categories. As a result, we tell everyone "hey, this could be spam".

The Exclude Tab:

Make the Exclude tab your friend... My experience is that there are a number of legitimate commercial messages and mailing lists that are automatically marked as spam. This could be because of the inaccuracy of black hole lists, or the fact that the MX server did not have a reverse DNS listing among other reasons. However the recipient really wants to receive messages from these services.

Over time, I built up a list of the common subscriptions that I want to pass through the system. These go in the Exclude tab. There are different places to configure the Exclusions in place for Blocking and Filtering, and they operate in a similar fashion.

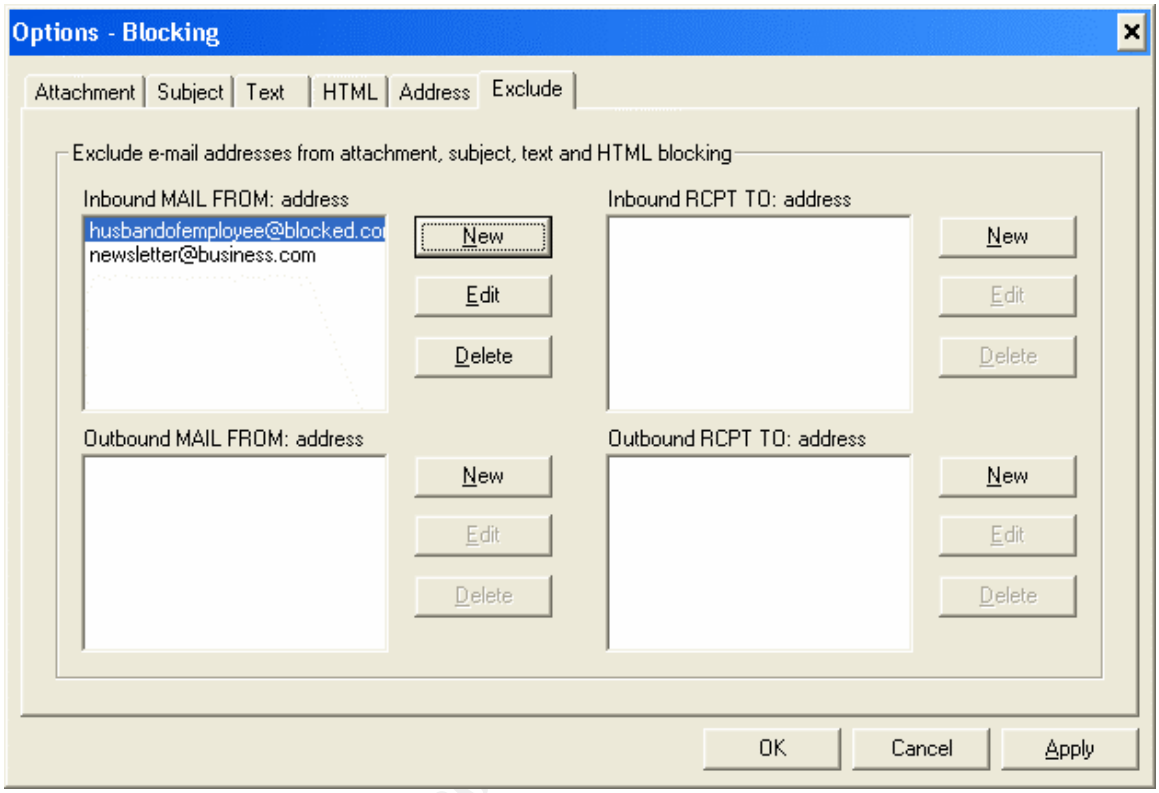

Figure 17: The Exclude Tab

The Spam Options Tab:

The SPAM Options tab allows you to look up the IP Address of connecting hosts in black hole lists.

After some experimenting I have found the following lists to be reasonable and not contain a lot of false positives: orbs.dorkslayers.com, relays.ordb.org, bl.spamcop.net and spam.dnsrbl.net. These lists are free to the public. Fee based subscription services are also available and may have higher accuracy.

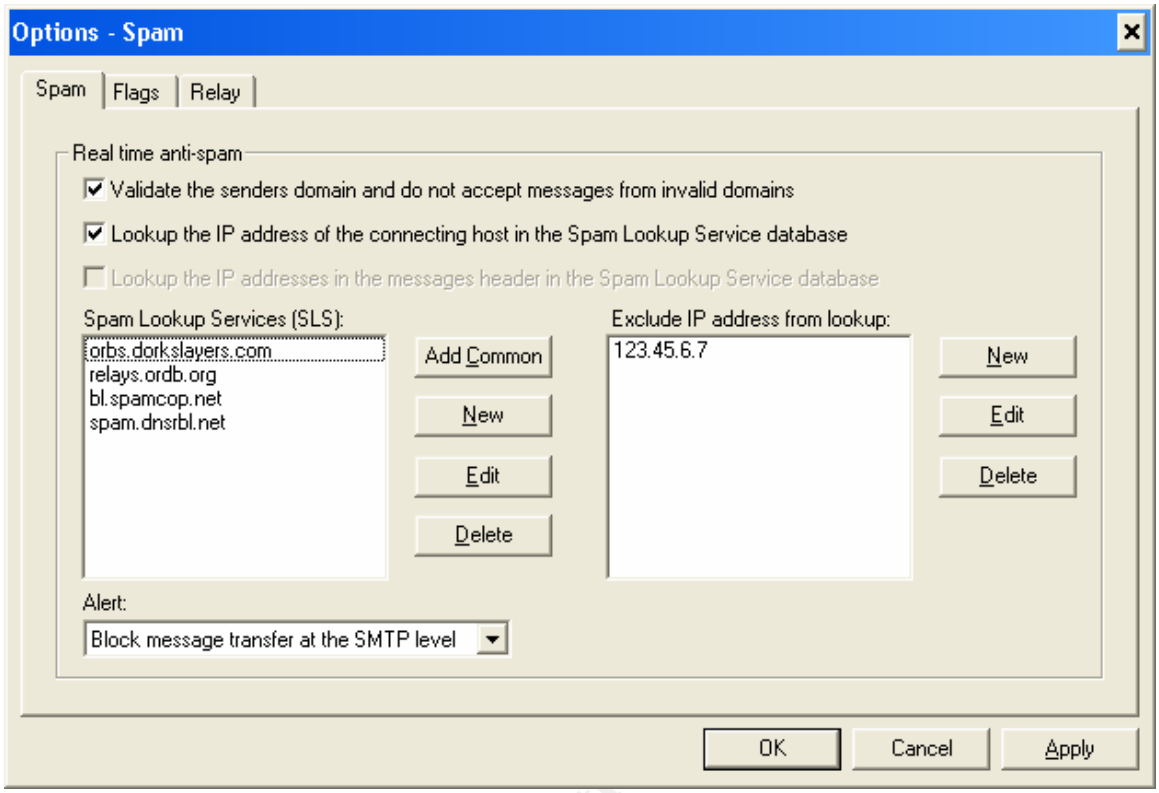

Figure 18: Configuring Black Hole Lists on the SPAM Options Tab

Each of the RBL lists rely on different criteria to create their list. It is important to understand what the criteria are because it will affect how exacting your filters are.

Many of these lists allow anyone to submit reports. A vendor could blacklist a competitor, disrupting their email while the listing is vetted. Many of the lists publish the whistleblower's email address to prevent spurious complaints. The biggest drawback is that users can, and do submit otherwise legitimate commercial email to black hole lists – out of frustration. Obviously this reduces the utility of the RBL List to others who wish to receive messages from these commercial sources.

The quality of the individual RBL lists varies greatly, and the use of these lists is controversial because they deny delivery to other people's servers! You must understand and keep an eye on how these lists operate. Fortunately there is a good reference for selecting your lists on the Declude website[10]: There are commercial RBL lists which may be used for a subscription fee, as well. These commercial lists may be more selective and may allow you to increase the accuracy of your filtering by comparison to using free RBL services.

The Spam Options Flags Tab:

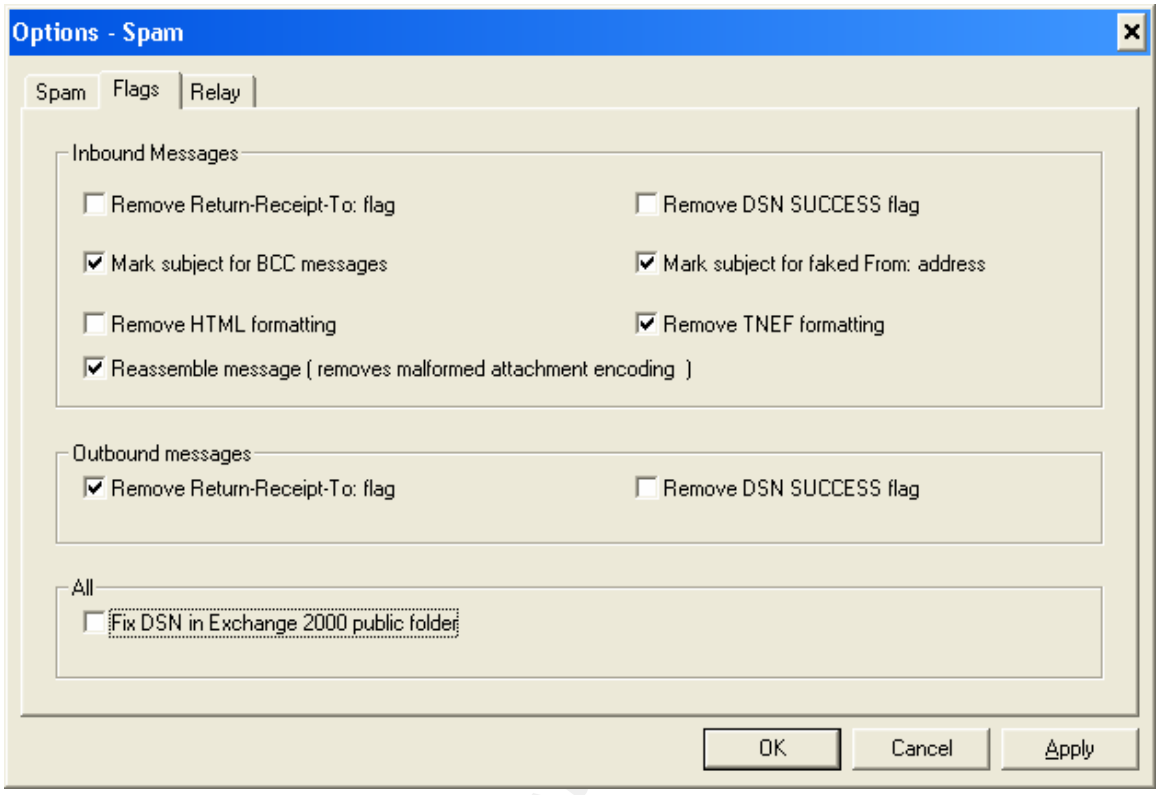

Figure 19: The Spam Options Flags Tab

The Spam Options Flags tab is where you set up filters to remove undesirable message formatting, and to reassemble the message to remove malformed attachment encoding.

The Spam Options Relay Tab:

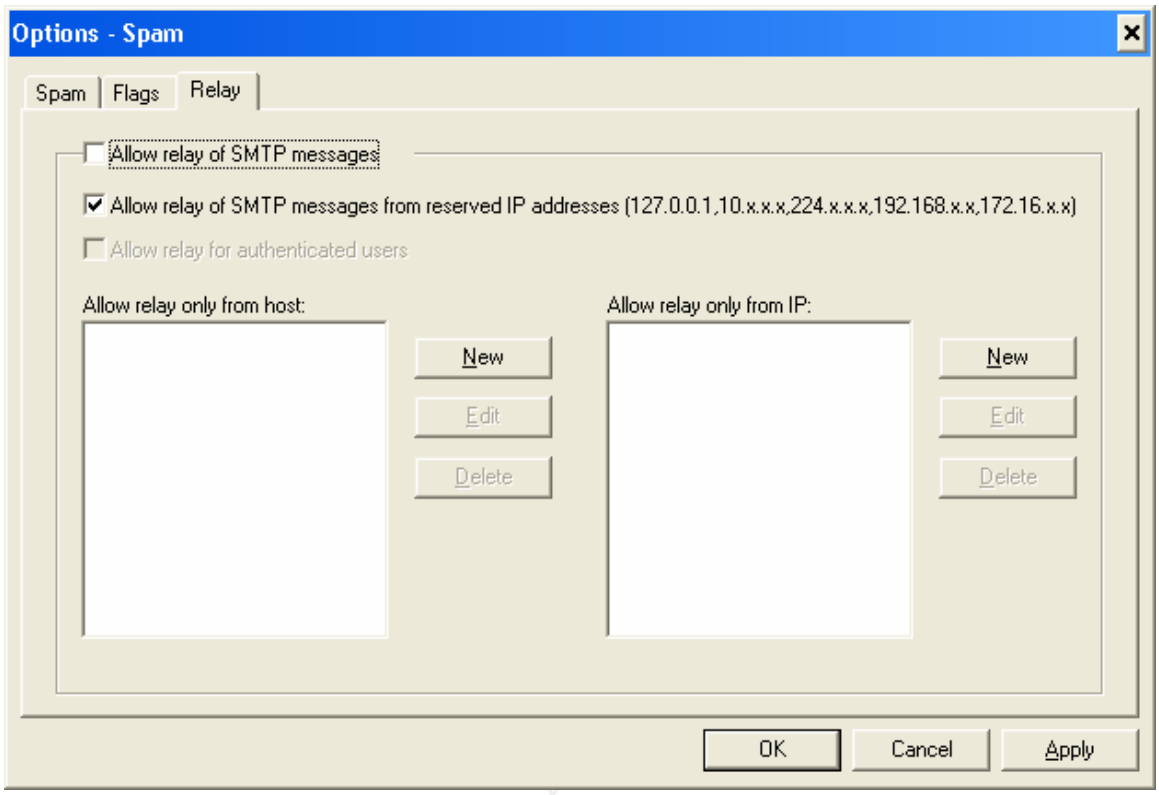

Figure 20: The Spam Options Relay Tab

The Spam Options Relay Tab is where restrictions are set on who may transfer mail through our server. In our case the email is coming from the Exchange IMS itself. This tab is configured to allow the loop-back address 127.0.0.1 to relay messages from Exchange IMS. In this case it would also allow messages from any machine on our local IP network.

Spam Filtering

The main options to understand when setting up a spam filter are: Black Hole Lists, Content Filters, Bayesian Filtering; and when to apply an Exclusion to a rule.

In XWall, Content Filtering occurs before the message is relayed to the Exchange server. In a content filter, a message is scanned looking for a specific string. If the string is found the message subject may be marked, the message transfer blocked, or forwarded as specified by your policy.

A Bayesian Filter is a specialized Content Filter. These filters build up a list of terms that are commonly used in email spam. The messages are scanned for these terms and are weighted based on the result. XWall will then mark the subject line, delete, or forward the messages based on the result. Items identified

as spam by such methods as RBL lists, will automatically be classified using the Bayes algorithm.

If the system is configured to use the XWall gateway while sending as well as receiving messages, then you may specify an email address to use as a manual submission point for spam. The instructions suggest using spam@bayes.spam for example. When Exchange IMS goes to send a message to that address, the IMS delivers all messages to localhost. The XWall gateway will relay outbound messages from Exchange. At that point the gateway can siphon off messages to 'spam@bayes.com' to feed the Bayes classification engine.

I have found the RBL lists to be more useful to me than Bayesian filtering. I think more could (and will) be done in this area over time. One thing you can do to improve on Bayesian filtering is to increase the amount of storage that the classification process uses. Another is to fine-tune the methods that feed the classification engine.

On a practical level, all of the Blocking and Filtering methods result in some inaccuracies. It takes substantial tweaking to find the best combination of RBL servers and filters for each organization.

Testing

During the testing phase you should start XWall as a program. This allows you to see the console statistics during processing. Once you have gotten the initial bugs out of your configuration, then you can install XWall as a service, so that it will automatically load at boot time.

After installing XWall to run from an NT service, setting the configuration, and then starting the service you are up and running in production.

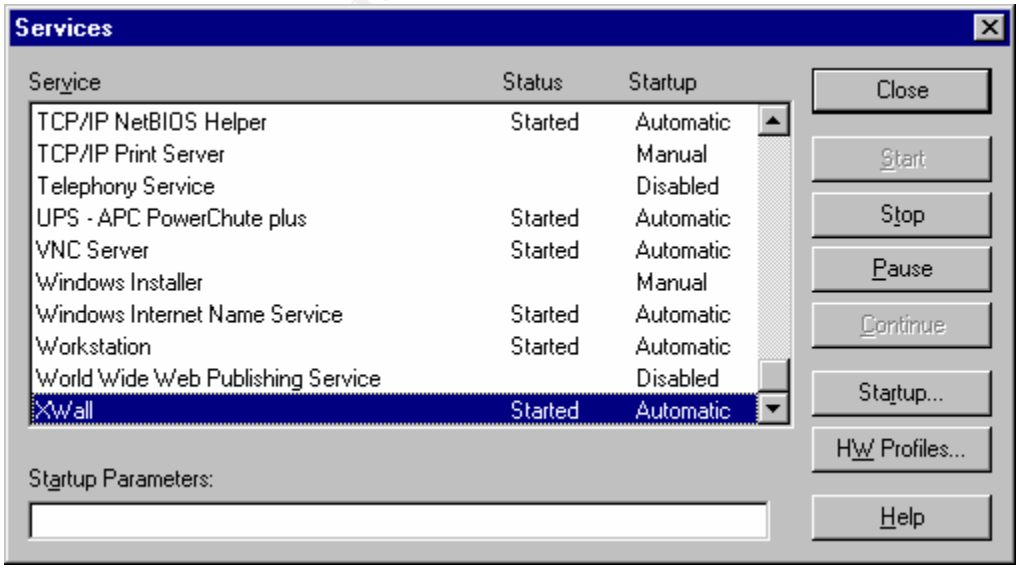

Figure 21: Starting the XWall Service

After: Current Status and Lessons Learned

X-Wall is a good solution in our environment. Its low cost and frequent updates make it a welcome addition to our defensive arsenal. While our initial task was to find a spam filter, we were more than glad to add other defensive capability to our system. For example the reassemble message feature of XWall is almost an application level proxy for email.

An anti-virus system is a reactive method of dealing with email born threats to our system. By contrast, capabilities such as rewriting email messages and eliminating attachments with multiple extensions (for example 'porn.jpg.scr') create greater depth as we attempt to defend our internal systems. These rules for mail processing may prevent viruses and attacks that are not yet identified in virus updates.

Spam filtering is not a panacea. The disadvantages of using spam filtering are not limited to Xwall. Filtering requires substantial customization to fit the infrastructure and policies of any company. There is great depth in the technical details of the configuration, and it requires tuning. In many cases someone will need to review the tagged emails to confirm that there are no false positives. Filtering is also controversial, both at a local level and within the technical community. The installation has shifted the burden of reviewing the tagged messages to the system administrator from the end user. In our environment this takes about 10 minutes a day. At the moment, this is acceptable, but not ideal. If email traffic increases in the future this could become a problem. Improving the accuracy of the filter would reduce this burden.

As a practical matter each spam filtering technique may be tested by tagging the subject line and forwarding the tagged message to the administrator. The administrator can run through the forwarded messages to check the accuracy of the filter, delivering those that were inadvertently trapped. Once the technique is vetted, the action could be changed to "delete" the message. Unfortunately deleting a message is irreversible.

As of May 2003, we typically receive around 110 messages that are identified as spam on a daily basis. Of these, perhaps one or two messages every other day are false positives. Once a 'false positive' is identified it may easily be configured as an 'exclusion' in the XWall administrative interface. Over time this reduces the number of false positives. But still there is a lot of room for improvement in accuracy. I feel that I have reached the greatest accuracy available by choosing from free DNS RBL lists. Subscription based RBL lists could improve your accuracy. We have not had the opportunity to explore these services in detail.

Statistics

X-Wall writes log files in CSV format. You can manually draw up statistics based on the log files to get a sense of how the filter is working and how to tune it. The following statistics are from June 11th, 2003 – a "heavy" spam day.

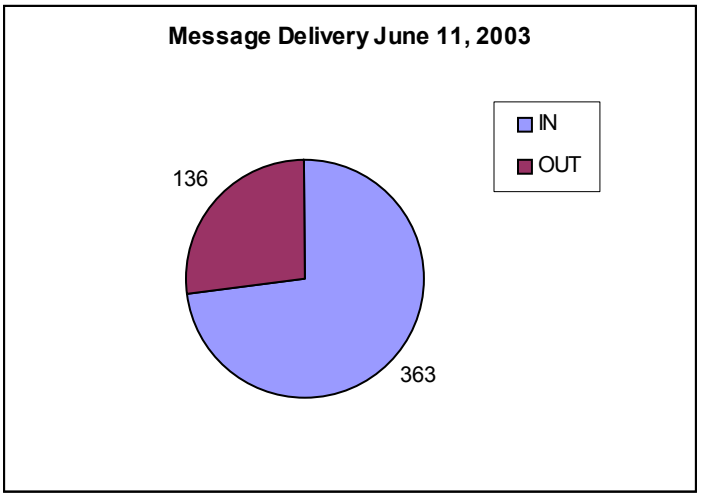

Figure 22: Messages that were Delivered (Were not Blocked)

A total of 449 messages were delivered on June 11th, 2003. Of these, 363 were inbound (from the internet) and 136 were outbound.

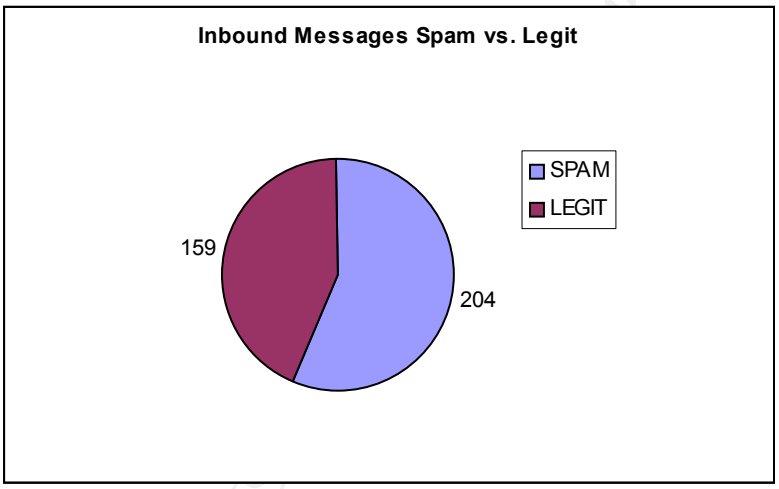

Figure 23: Spam vs. Legitimate Inbound Messages

Of these 363 inbound messages, 204 were verified as spam. There were 67 false positives and the spam filter tagged 271 messages total. There were 159 "legitimate" inbound messages. Roughly 25% of the tagged messages on this day were false positives.

We have determined that for our environment, a message is not "spam" unless it is both unsolicited by the recipient and the recipient in fact does not want to receive the message. For example, false positives were received from Roving software – a commercial email house. These were acceptable opt-in email bulletins – by our standards, and were not spam. Mailing lists are frequently tagged as spam sources for a number of reasons. Often the sender's email

domain will not match the domain of the mail exchanger, for example. In practice it seems that every desired mailing list should be configured as an exclusion to the blocking and filtering rules.

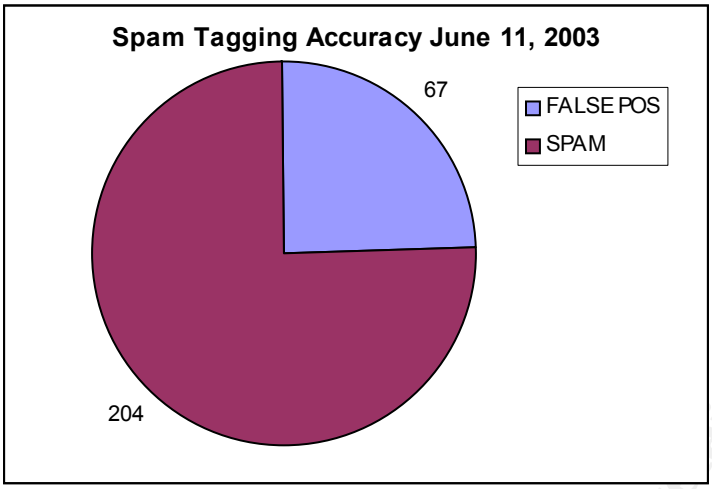

Figure 24: Accuracy of Message Tagging

Looking into the log file for message transfers, on this day fifteen messages were denied delivery because the sender domain could not be resolved. In many companies a more aggressive policy would further reduce messages accepted at the SMTP level.

X-Wall is compatible with the public domain MRTG monitoring package, providing a very sophisticated "dashboard" for the mail exchanger. To get the most out of X-Wall you will need to have quick access to statistics. Otherwise it is very hard to see how well you are doing in combating spam. Installing the MRTG package is the next item on my to-do list, because I feel that I can greatly improve my filter's accuracy with a faster way to track progress!

Careful with that Scalpel, Doctor!

It is a well known adage that the majority of system outages are caused by system administrators. Content filtering is, as I suggest above, a very sharp scalpel. It is important to schedule your configuration and testing so you are sure exactly what effects your filtering is having.

One Friday during the testing I became fed up with spam purporting to improve on the male anatomy. "I can fix that!" I thought. I decided to filter out any messages with related terms in the subject line. And nobody in our organization would be sending legitimate email with that subject anyway. Entering that string was so satisfying, that I decided to get rid of messages containing the subject, asterisk asterisk ("**"). Not only did I set this up, but I left the default action of delete enabled. Everything seemed just great and I went home for the weekend.

Monday morning, I discovered that XWall's content filtering uses regular expressions to match the search string to the subject line of the message. A search string of "**" will match every message! All email traffic over the weekend was deleted! The best I could do for my users was to inform them who had sent email to them over the weekend and ask them to request that it be re-sent.

From this experience, I draw two lessons:

1) Test everything
2) Make no significant changes before leaving the office for the weekend.

These are things that any good Sys-Admin already knows, but it is very tempting to drop in a few changes at the last minute. Don't do it!

Table of References

[1] Ferris Research. “Spam Control: Problems and Opportunities.” Ferris Research Report. Jan. 2003. URL: http://www.ferris.com/url/spam.html. (15 June 2003).

[2] Microsoft Corporation. "XIMS: How to Stop Spam Mail Messages from Using IMS Relay Agent." Microsoft Knowledge Base Article – 199656. 22 May 2003.
URL: http://support.microsoft.com/?kbid=199656. (15 June 2003).

[3] Schorr, Ben & McBee, Jim. “Eliminate Open-Relay Annoyances.” .NET Magazine. Sept. 2002.
URL: http://www.fawcette.com/dotnetmag/2002_09/magazine/columns/askthepros/. (15 June 2003).

[4] Microsoft Corporation. "XFOR: Internet Mail Service May Relay Messages Despite Restrictions." Microsoft Knowledge Base Article – 264330. 22 May 2003.
URL: http://support.microsoft.com/?kbid=264330. (15 June 2003).

[5] Tschabitscher, Heinz. “How to filter out spam from your mailbox.” About.com. 2002.
URL: http://email.about.com/cs/spamfiltering/index.htm. (15 June 2003).

[6] Network Associates. “Network Associates Acquires Deersoft, Inc. Anti-Spam Technology.” Network Associates Website. 6 Jan. 2003.
URL: http://www.networkassociates.com/us/about/press/corporate/2003/20030106.htm. (15 June 2003).

[7] DataEnter. **“**XWall for Microsoft Exchange.**”** DataEnter Website. 2002.
URL: http://www.dataenter.co.at. (15 June 2003).

[8]DataEnter. **“**XWall for Microsoft Exchange - Documentation.**”** DataEnter Website. 2002.
URL: http://www.dataenter.co.at/doc/xwall.htm#inst. (15 June 2003).

[9] Symantec. “What is the Eicar Test String?.” Symantec Security Response Website. 14 May 2002.
URL: http://securityresponse.symantec.com/avcenter/venc/data/what.is.the.eicar.test.string.html. (16 June 2003).

[10] Declude. “List of all Known DNS-based Spam Databases (blacklists).” Declude Website. 16 June 2003. URL: http://www.declude.com/junkmail/support/ip4r.htm. (16 June 2003)