Global Information Assurance Certification Paper

Implementing a 'Healthy' open source, spam solution.

Tackling the problem of spam in a New Zealand health organisation

Andy Hopkins

GIAC Security Essentials Certification (GSEC) Practical Assignment
Version 1.4c
Option 2

March 19, 2005

Contents

Implementing a 'Healthy' open source, spam solution.

Tackling the problem of spam in a New Zealand health organisation

Abstract

To a non-profit health provider in New Zealand spam is an increasing problem which affects the security and productivity of the organisation and ultimately patient care. The nature of the business is such that a highly accurate solution was required while still minimising the cost; one which fits the organisations strategic direction. The solution would also need to be secure in order to protect information assets. GIAC-Health adopted an open source spam solution and reduced spam by over 90%, saving money and CPU cycles.

Introduction

> *Spam: "An electronic communication containing material or references to material of a commercial, solicitation or illegal nature, directed as part of a bulk distribution to any address where the address-holder has not given explicit prior consent to receive it". (Harris, pg.5)*

GIAC-Health is a service organisation providing IT services to a large, government funded, health provider in New Zealand. To them, like many other organisations globally, spam is a problem. While national administrations attempt to deal with it through legislation and IT vendors look at possible technical solutions, it is the end user in the mean time who continues to face the barrage of this unsolicited, bulk e-mail

The problem had reached the stage that GIAC-Health had to find an effective, cost effective and secure anti-spam solution. This paper discusses the issues facing an organisation such as GIAC-Health and the process and decisions made implementing their chosen anti-spam solution. The role of the author was that of principle technical resource and architect.

The in-depth installation, configuration of the various components, each of which could be subject in its own right, will not be covered here.

The SpamAssassin installation and configuration is as per Alan Schwartz's book 'SpamAssassin' [O'REILLY, 2004] and will not be repeated here other than some references. It is assumed that the reader has an understanding of UNIX, in particular RedHat Linux.

Since SPAM is a registered trademark of the Hormel Corporation here we

will refer to it as spam.

The Problem.

Although for many end users it's simply an annoyance, the effects of spam are many and varied, but whichever way you look at spam, "it comes down to cost shifting; it is the end users that pay, not the spammer" (Harris, Pg.5).

The process of receiving large amounts of unwanted e-mail uses bandwidth; bandwidth that must be paid for in ISP costs and degraded performance. Spam also takes up processor time and disk-space on the e-mail servers. This in turn costs money which GIAC-Health would prefer to spend on healthcare systems and resources which would be better utilised maintaining the performance and availability of health systems.

> One person can generate huge volumes of mail with just a few clicks of a mouse, blanketing millions in a matter of minutes or hours. However, the ability for one individual to generate enough e-mail to take down the systems of a multi-million dollar corporation means that on a daily basis we are faced with situations in which a single person's actions can cause damage and business losses often far in excess of their ability to pay for the trouble they cause.(Everett-Church)

It is clear from inspection of outgoing e-mail gateway logs, and the amount of undeliverable e-mail, users still not only open spam e-mail, but they respond in some way; sometimes to follow-up on an offer, or simply hoping to unsubscribe from it. Not only is there the frustration users experience dealing with so much unwanted e-mail on a daily basis and the distress attached to dealing with the often graphic images attached to pornographic spam, but the untold impact on productivity and the cost that represents to the organisation.

Although the possibility of malicious content or URL's within the spam must be considered a risk, all Internet e-mail will still pass through the anti-virus gateway and virus protection on the e-mail servers and the desktop.

These costs ultimately affected the security of GIAC-Health's information assets. The problem facing GIAC-Health was how to minimise the spam and return the resource usage to where it was meant – dealing with e-mail that's actually wanted.

The basic e-mail architecture before implementing the anti-spam solution is shown in Figure 1

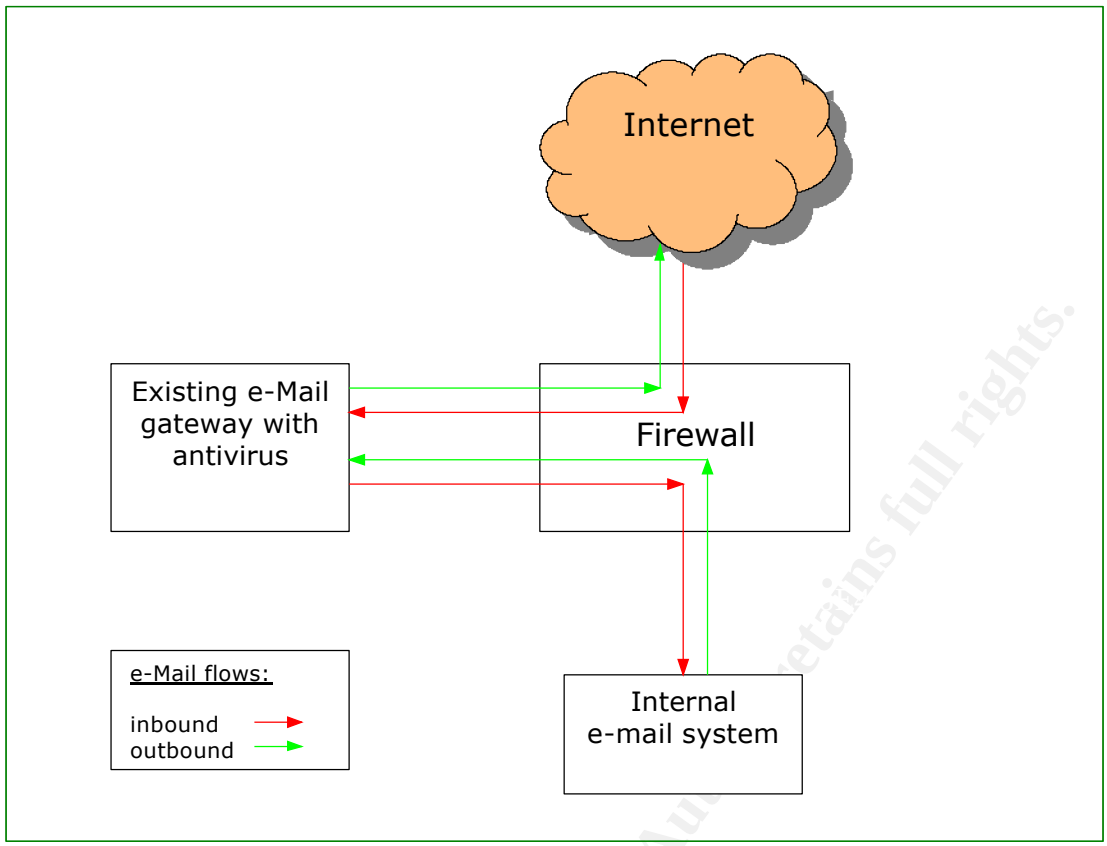

Figure 1

The Solution

Using the Spamhaus DNSBLs you can very safely reject the vast majority of spam at SMTP connect time. The second stage is to scan the remaining mail which gets past first stage IP filtering, looking for URLs (web site addresses) in the message body and testing their host IPs against the SBL.(THE SPAMHAUS PROJECT)

Selecting a solution.

GIAC-Health has stated in their strategic plan to look at open source solutions wherever possible, so the requirements of the solution were defined as:

'Using open source where possible, deny incoming e-mail connections from known spammers based on a respected database, apply smart heuristic filtering to all others and finally embark on a user education campaign to educate users to help themselves'

GIAC-Health opted for a solution based on RedHat Linux running

SpamAssassin in conjunction with Spamhaus' SBL. "This technology offered the best balance of functionality, performance, security and overall cost". (Allen, pg.39).

From here on, SpamAssassin will be used to refer to the Linux host as well as the spam filtering software.

"Usually, you should position spam filtering at the outermost edge of the e-mail. This "catches" the spam before it travels the internal network and consumes bandwidth and, potentially, storage". (Hallawell and Caplan)

The SpamAssassin would reside in the organisations DMZ, accept incoming e-mail, examine and score messages for their spam likelihood before forwarding the message to the existing e-mail gateway. The decision on if and where any spam would be deleted would be made later. Placement in this manner would allow the host to be easily connected into the existing e-mail architecture with minimum disruption to e-mail services and provide for an equally easy rollout if there were any problems.

Although residing in the DMZ and behind the organisations firewall, the server would still be public facing and a potential target for malicious activity. It would also need to be secured in order to protect it should another server in the DMZ be compromised.

False positives; the bane of all spam-checkers

"False positives are the bane of all spam-checkers" (Schwartz, Pg.31). No anti-spam solution can guarantee to be 100% effective and accurate. One must be mindful that the filter will from time to time mark legitimate messages as spam (false positives) or mark spam as Ok (false negatives). The risks of both scenarios need to be understood; users have an expectation that this will be the end to their spam woes and that they will continue to receive all of their legitimate e-mail

Being a health organisation with over 4000 users, receiving e-mail from literally 100's of sites a day there is a real chance that legitimate e-mail will be received containing terminology which might easily be considered as spam. For example, an e-mail relating to sexual health may be considered to be pornographic, or a discussion on pharmaceuticals may be consider and advertisement for 'Cheap Meds'. The likelihood of false positives was considered to be all too real and simply deleting such e-mail was considered unacceptable.

"SpamAssassin uses a large number of rules and weighs them based on there effectiveness at identifying spam"(Schwartz, Pg.1), any e-mail which exceeds a pre-defined 'required_score' is considered spam. Initially, to overcome the potential for false positives, GIAC-Health decided that SpamAssassin should classify dubious messages at two levels; SPAM-1 would be those messages obtaining a score of at least twice the required

score, SPAM-2 would be those messages which reached the required score but less than that for SPAM-1. As it turned out, SpamAssassin was found to be that accurate that these classifications were soon to be dropped.

The e-mail client software was configured to automatically place such e-mails into the 'Junk e-Mail' folder. This then gives the user the final say as whether to accept the e-mail or simply delete it.

It was envisaged that as the business' end users became more confident in the accuracy of SpamAssassin, then ultimately the decision would be made simply delete all messages marked as SPAM-1, if not SPAM-2 as well.

The new e-mail architecture, post implementation is shown in Figure 2.

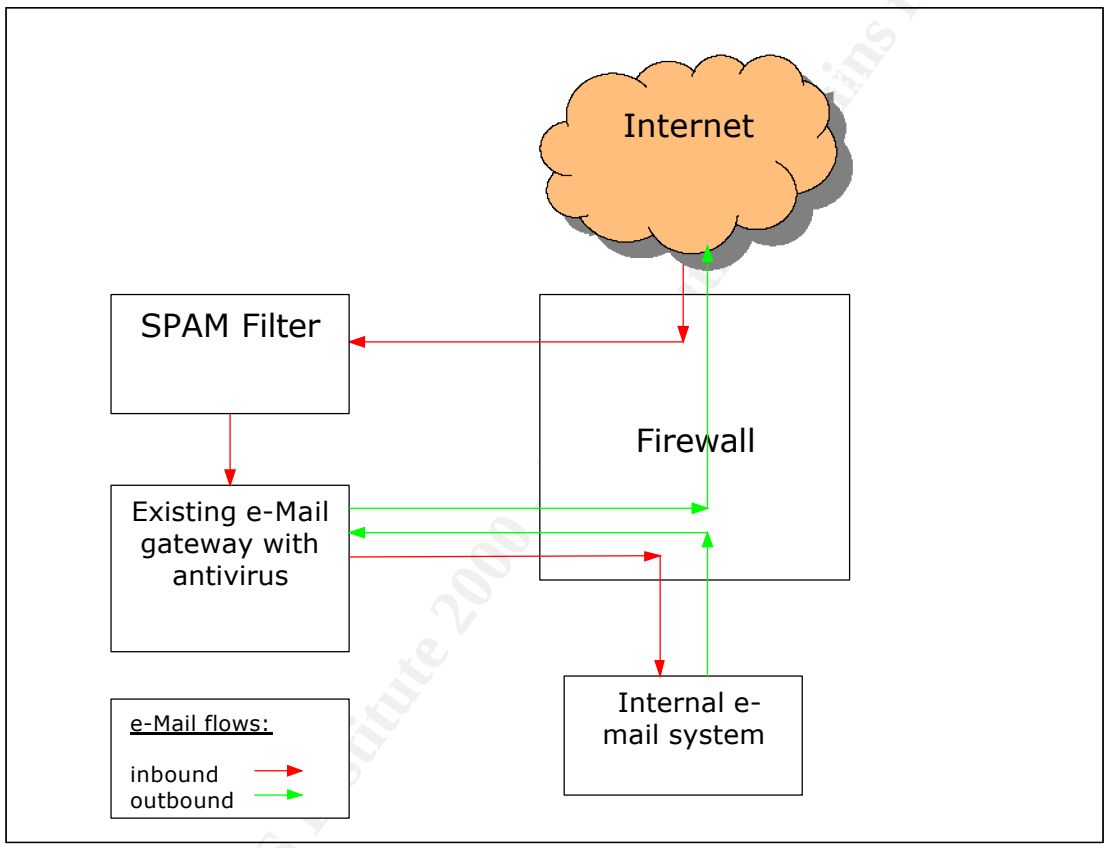

Figure 2

Installation of RedHat

The installation of RedHat Linux was a basic 'Server' installation from original CD. The basic server installation normally installs and enables a number of packages which were not required, such as printing support or X-windows. Only the bare minimum packages were installed, what those packages are will no doubt vary depending upon requirements.

Once installed, applicable patches and updates were down loaded from the RedHat web-site. These patches and versions will vary over time so it's important to check the website rather than rely on a list here which will quickly be out of date.

Securing the installation

Even though only a small set of packages were installed, one should not rely solely on this. There was still some additional work needed to ensure the security of the host.

> The services you enable on a selected host depend on the functions you want the host to provide. Either do not install unnecessary services or turn the services off and remove the corresponding files from the server (Allen Pg 43-45)

Exactly which services and daemons to disable would be decided by the organisation security policy and to a certain degree the hardware configuration in use. For this installation, those services included the PCMCIA (card and socket services), RAID monitoring (md), Network file Shares (nfs), RPC Portmapper. Additionally the xinetd service (which controls the likes of ftp, telnet and the r-services) was disabled and all its configuration files removed – if you don't need them, delete them!

Kernel tuning

"The core of the Linux operating system is the kernel; it has many different parameters which can be tuned, some can affect security" (SANS Institute).

The following kernel network tuning parameters were set (in /etc/sysctl.conf) to reduce the risks of spoofing, Smurf and SYN flood attacks:

```
net.ipv4.conf.eth0.accept_source_route=0
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.conf.all.accept_redirects=0
net.ipv4.conf.all.send_redirects=0
net.ipv4.conf.all.rp_filter=1
net.ipv4.tcp_max_syn_backlog=4096
```

There are other parameters which could be set; many of them, but it's a question of acceptable risk versus the resources available and required performance. Some parameters will take more memory; others might limit the number of concurrent connections and hence slow down mail throughput. GIAC-Health opted to use those recommended by SANS, at least as a starting point.

The following resource limits were set (in /etc/security/limits.conf) in order to reduce the potential for resource exhaustion by logged-on users.

```
*          hard    nproc   20
*          hard    core    0
```

Remote Administration

On UNIX systems, connectivity for remote system maintenance could be supported using the r-services, telnet or secure shell (SSH). Due to their inherent vulnerabilities the r-services and telnet have been disabled.

Therefore SSH was selected as the more secure alternative. (Allen, Pg.45) and is only permitted from the internal network

Host firewall

Linux iptables was used as a host based firewall configured to permit connections only for SSH connections from the internal network and SMTP from any source. The following rules were implemented.

```
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Firewall-1-INPUT  all  --  0.0.0.0/0            0.0.0.0/0

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Firewall-1-INPUT (2 references)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0           icmp type 255
ACCEPT     esp  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     ah   --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           state NEW tcp dpt:25
ACCEPT     tcp  --  10.0.0.0/8           0.0.0.0/0           state NEW tcp dpt:22
REJECT     all  --  0.0.0.0/0            0.0.0.0/0           reject-with icmp-host-prohibited
```

'iptables' rules

Securing root access

On a UNIX system the 'root' user has ultimate and complete control of the machine, hence it is very important to restrict as much as possible the potential for misuse.

The root user was given a strong password consisting of upper and lower case characters, numeric and special characters and root access via the network was disabled allowing access from the console only.

This was achieved by adding to /etc/pam.d/login the line

```
auth       required     pam_securetty.so
```

Installation of sendmail

One of the stated requirements was that the anti-spam solution should

'*…deny incoming e-mail connections from known spammers based on a respected database...?*

This first stage filtering was achieved by enabling the sendmail FEATURE option with the dnsbl argument which "turns on rejection of hosts found in an

DNS based rejection list" (sendmail.org)

From /etc/mail/sendmail.mc …

```
dnl # Use the Spamhaus Project SBL Blacklist
FEATURE(`dnsbl', `sbl.spamhaus.org', `', `"451 " $&{client_addr} " in sbl.spamhaus.org"')
dnl #
```

To meet our second requirement to "*apply smart heuristic filtering* " and use SpamAssassin as a filter for sendmail we looked to the sendmail feature called MILTER which can be used for plug-in filters. Unfortunately MILTER support is not compiled in to the version of sendmail that was to be used, so it was necessary to install the necessary development tools (C compiler et al).

Having compiled a version of sendmail with MILTER support, remove the C compiler afterwards, we don't want those kinds of tools hanging around and available to a malicious person who may be find themselves access to the server.

Securing sendmail

"Because of the nature of email, a determined attacker can flood the server with mail fairly easily and cause a denial of service. The effectiveness of such attacks will be limited, by setting limits to directives in `/etc/mail/sendmail.mc` (RedHat)". The following directives were set.

- confCONNECTION_RATE_THROTTLE ; the number of connections the server can receive per second. By default, sendmail does not limit the number of connections. If a limit is set and reached, further connections are delayed. (RedHat)

- confMAX_DAEMON_CHILDREN; the maximum number of child processes that can be spawned by the server. By default, sendmail does not assign a limit to the number of child processes. If a limit is set and reached, further connections are delayed. (RedHat)

Of course we need to be sure that our anti-spam solution can't be used as an open relay for sending spam, so it's configured to allow incoming e-mail for users at applicable domains such as user@GIAC-Health.co.nz

Installation of MIMEDefang and SpamAssassin

The current versions of MIMEDefang and SpamAssassin and any prerequisite packages were downloaded from their respective web-sites. (See Bibliography). Each was checked against the advertised MD5 sums to ensure integrity before being installed and configured in accordance with their respective README's and Alan Schwartz' book (Schwartz)

In order to achieve the two classifications of spam already discussed, a change was required to the filter_end function within the mimedefang-filter script which interfaces sendmail with SpamAssassin

Sub filter_end

….
….

```
        if ($hits >= $req) {
          action_change_header("X-Spam-Score", "$hits ($score) $names");
          md_graphdefang_log('spam', $hits, $RelayAddr);
          if ($hits >= $req * 2) {
            action_change_header("Subject", "[SPAM-1] $Subject");
          } else {
            action_change_header("Subject", "[SPAM-2] $Subject");
          }
          # If you find the SA report useful, add it, I guess...
          action_add_part($entity, "text/plain", "-suggest",
                  "$report\n",
                  "SpamAssassinReport.txt", "inline");
        } else {
```

…
…

User accounts were created for running and managing the MIMEdefang and SpamAssassin components.

System logging

> Collecting data generated by system, network, application and user activities is essential for analysing the security of your information assets and detecting signs of suspicious and unexpected behaviour (Allen Pg 198)

It is also important to be able to monitor the performance of the spam filters and the performance of the host. This helps identify any mail delivery bottle necks or excessive resource usage on the host which could point to malicious activity,

Syslog was configured to replicate all the host logs to the central logging server on the internal network from where existing processes would be used to analyse the logs.

The risk of malicious activity deliberately causing excessive logging to the central server resulting in a denial-of-service was evaluated. Although the potential impact would be significant, to that server, there would little or no

direct impact on health systems; this risk was considered acceptable. However, resource exhaustion on the anti-spam server was a concern as it would effectively prevent all incoming mail delivery; therefore logging of mail logs on the host was turned off in favour logging to the central server only. Not an ideal solution and one which needs to be revisited a better solution would be to log to a separate disk and remotely.

To facilitate easy gathering of host performance information (cpu, load, memory and swap), syslog was used to send resource usage data to the central logging server every 10 minutes.

Backups

There are a number of backup strategies available for backing up servers.

> A typical approach used for public servers such as DNS where the content changes at a predictable rate is to maintain an authoritative version of the information content of the server on a secure server. If the server was compromised, the information can be reloaded from that secure server. (Allen Pg. 60)

Since the only regular changes to this server would be changes to the SpamAssassin rule-base, the strategy chosen for this scenario was to backup the operating system to DVD and maintain an up to date copy of the rule set on a separate, secure server on the internal network.

Finally, with the installation complete, the configuration checked and backup taken and following the necessary internal change control process the server was rebooted connected to the DMZ network.

Firewall changes

Of course none of this would work without the requisite changes to the firewall;

- NAT rule changed to direct incoming SMTP connections to the SpamAssassin host instead of the anti-virus host.
- Allow NTP (network time protocol) from SpamAssassin to the time source on the internal network in order to keep its clock synchronised with the rest of the GIAC-Health network.
- Allow SSH from the internal network to SpamAssassin for remote administration.
- Allow SYSLOG from SpamAssassin to the internal, central logging server.

The Results

Prior to implementing the SpamAssassin GIAC-Health had no clear data on exactly how much of a problem spam was. In order to prove the ultimate effectiveness of the solution a number of key users were identified who received between 80 and 160 spam e-mails a day. It is from their feedback that the success of this system would be judged.

Within a matter of days of implementation it was noted among the sample users a 95% positive identification of delivered spam with 0% false positives which equates to a drop in spam from 80 to 4 items of spam a day as seen by the user.

Analysis of the mail logs show a 3% rated of SMTP connections being rejected because the delivering host appeared on Spamhaus' SBL. Together with a 17% rate of mail identified as spam by SpamAssassin, gives an overall spam 'hit-rate' of 20%. This is shown in Figure 3.

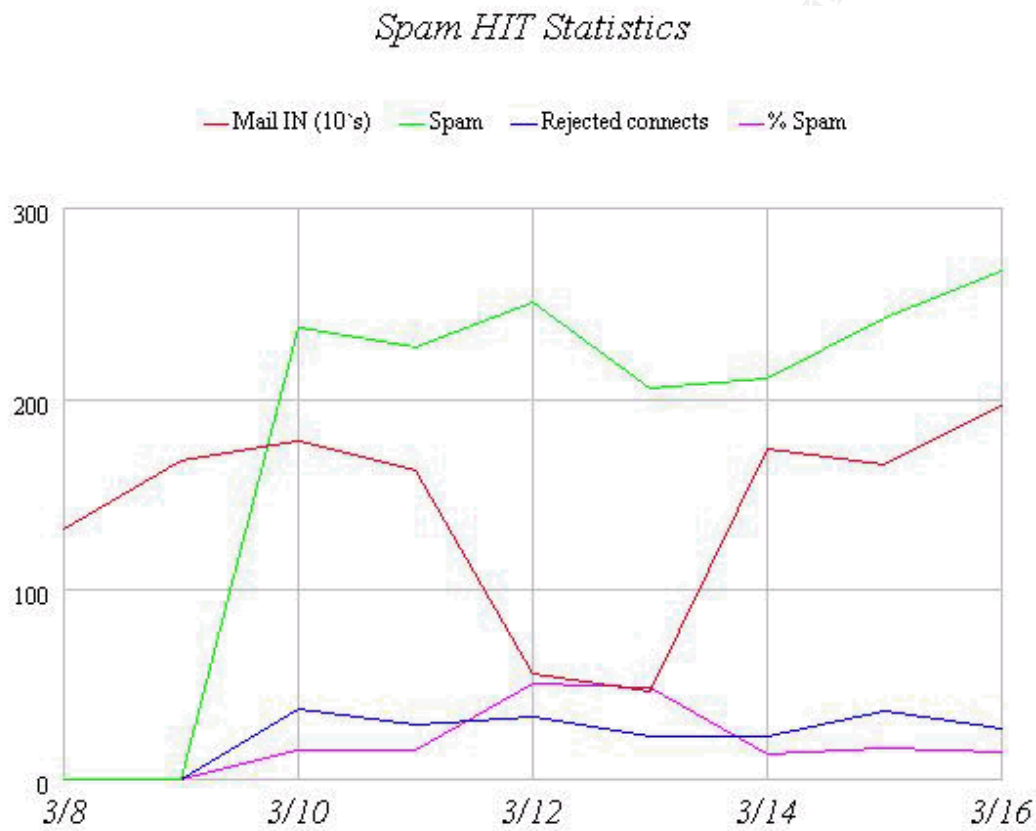

Figure 3.

These results were so pleasing that the decision was quickly made to not only delete SPAM-1 at the gateway, but also SPAM-2

As for host performance; the statistics showed the server to be generally under a light load. Typical performance information, taken at 10 minute intervals can be seen in Figure 4.

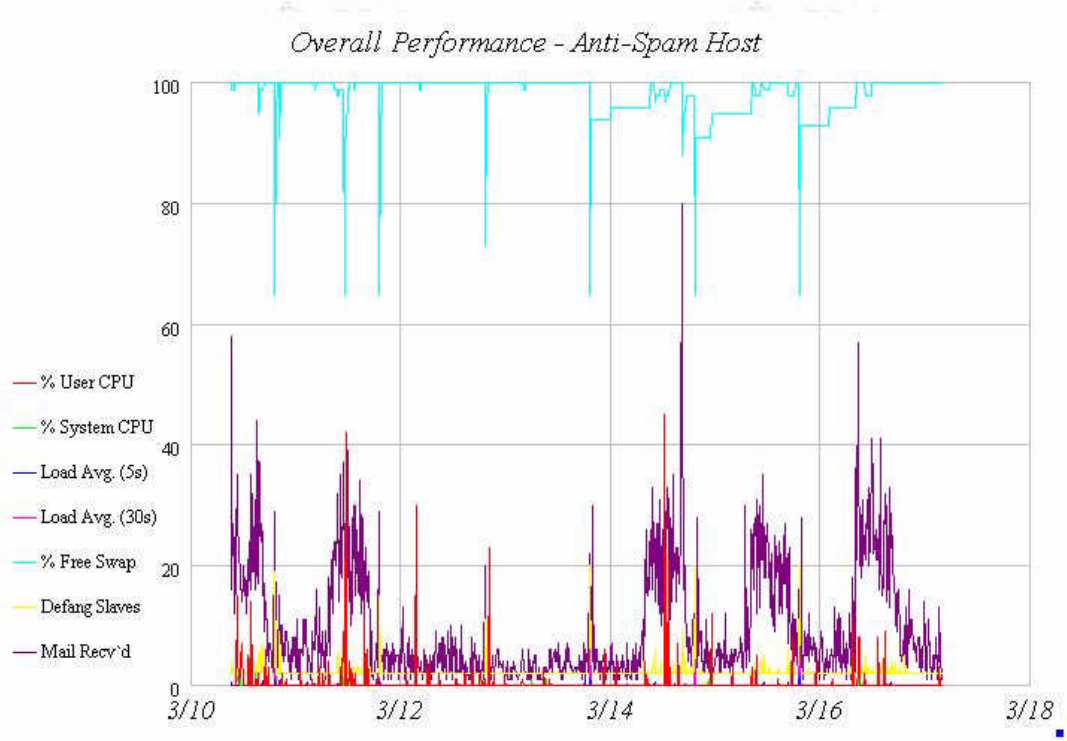

Figure 4

Where to from here?

Where to from here? Or, what else could we have done?

The anti-spam solution is in and working, but the story doesn't stop there. Continual monitoring of server, filter performance and system logs is required to ensure optimum performance, availability and accuracy.

Ongoing monitoring of security advisories and vendor sites is necessary to ensure bugs are quickly dealt with.

The SpamAssassin solution currently is implemented on a single server and as such represents a single point of potential failure. Redundancy and load balancing will be implemented through a second server.

There's more that can be done to secure sendmail, e.g. running the daemon with less privileges.

References

Harris, David. 'Drowning in Sewage', Legal & Regulatory Affairs Committee Reports, InternetNZ 3 September 2003 <http://www.internetnz.net.nz/public/committee-reports/ctte-legal-and-regulatory-affairs/larac030903spam-white-paper.pdf>

[Schwartz, Alan. SpamAssassin, Sebastopol: O'Reilly Media Inc., 2004.

SANS Institute. Track 1 SANS Security Essentials and the CISSP 10 Domains Vol.1.6. SANS Press, January 2004

Allen, Julia H. The CERT Guide to System and Network Security Practices. Addison-Wesley, May 2001

THE SPAMHAUS PROJECT, Effective Spam Filtering. SPAMHAUS, March 2005 < http://www.spamhaus.org/effective_filtering.html>

Sendmail.org, Sendmail Configuration Files, July 2004 < http://www.sendmail.org/m4/readme.html>

Lindberg,G., RFC2505, Anti-Spam Recommendations for SMTP MTAs, February 1999 <http://www.rfc.net/rfc2505.html>

Everett-Church, Ray. Statement before the United States House of Representatives Subcommittee on Telecommunications, Trade & Consumer Protection. November 3, 1999 <http://www.everett.org/testimony/house/>

Hallawell,Arabella and Caplan Grey, Maurene. "How to select spam-filtering products and services" TechRepublic, October 31, 2003 <http://techrepublic.com.com/5100-6270-5093725.html>

RedHat, Red Hat Linux Reference Guide, RedHat Inc. 2003, <http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/>

RedHat, Red Hat Linux x86 Installation Guide, RedHat Inc, 2003 <

http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/>

RedHat, Red Hat Linux Security Guide, 2003 <http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-server.html>

CAUCE : Coalition Against Unsolicited Commercial e-Mail Website, 2005 <http://www.cauce.org/>

The Apache SpamAssassin Project Website, 2005 <http://spamassassin.apache.org>

MIMEDefang Website, 2005 <http://www.mimedefang.org>