Global Information Assurance Certification Paper

Case Study: Replacing Sendmail with IronMail for Improved Spam Protection

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

Submitted by: Heather Scott

October 12, 2004

Table of Contents

Abstract/Summary

When I started writing this paper, I discovered that the word spam, as referring to the nasty unwanted email kind, came from a funny Monty Python[i] skit! The internet was a friendlier place back then. Back in its origins in the Usenet Newsgroup days of the early 90's, spam was just an annoyance caused by inconsiderate people who didn't comply with newsgroup etiquette[1]. Today, spam is costing corporations millions of dollars in wasted resources (disk space, processing power, etc). In my company, over forty percent of our mail is spam.

This year, we replaced our tried and true sendmail gateway with two Ciphertrust IronMail appliances[ii] due to the rapidly increasing amount of spam that we were receiving and the limited number of anti-spam options available in sendmail. This paper will discuss the details of the project and will compare the security features of our old sendmail system with those of our new IronMail appliances.

We are very happy with our IronMail appliances – they have stopped most of the spam from getting into our company, they have lots of useful features and they are easier to maintain than our sendmail gateway was.

---

[1] **The official Hacker's Jargon (3.2.0) meaning**

```
:spam: vt.  [from "Monty Python's Flying Circus"] 1. To
   crash a program by overrunning a fixed-size buffer with excessively
   large input data.  See also {buffer overflow}, {overrun
   screw}, {smash the stack}.  2. To cause a newsgroup to be
   flooded with irrelevant or inappropriate messages. You can spam a
   newsgroup with as little as one well- (or ill-) planned message
   (e.g. asking "What do you think of abortion?" on soc.women).
   This is often done with {cross-post}ing (e.g. any message which
   is crossposted to alt.rush-limbaugh and
   alt.politics.homosexuality will almost inevitably spam both
   groups).

   The second definition has become much more prevalent as the
   Internet has opened up to non-techies, and to many Usenetters it is
   probably now (1995) primary. .
```

Before

Background

As a Unix systems administrator of a large Insurance company in 1998, I was assigned as technical team lead for a project to implement our corporate firewall infrastructure that had previously been outsourced to a large ISP. I implemented the Checkpoint firewall on a hardened Solaris 2.6 server. One of my duties was to implement a mail gateway on the firewall, and I chose to implement sendmail (version 8.9.3) using chroot and every available security feature I could find in sendmail at the time. I closely followed Carole Fennelly's three papers on implementing sendmail on a firewall.[iii]

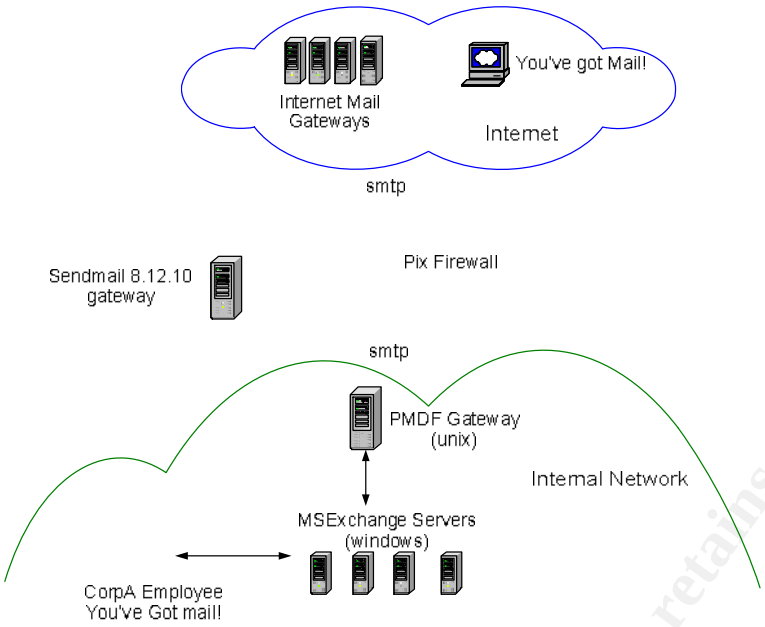
In early 2003, a decision was made to replace the Checkpoint firewalls with PIX firewall appliances. At this time, I was tasked with moving the sendmail gateway off of the firewall onto a separate server in a dmz off the PIX firewalls. I opted to implement the newest version of sendmail, 8.12.10, on a hardened Solaris 2.8 server.

Architecture

Our mail system (as in Figure 1) consisted of three main components:

- A MsExchange infrastructure which held everyone's mailboxes.
- An internal mail gateway on a Solaris 2.8 server running a mail messaging product called PMDF by Process Software[iv]. This gateway was originally implemented to route mail to our TAO email on the mainframe before we had internet access. I modified it to redirect internet mail to the sendmail gateway in the dmz.
- A sendmail gateway in the dmz which routed mail to and from the internet.

Figure 1:



Problems Identified

Problem #1: SPAM

The main and most important problem we were encountering was a rapidly increasing amount of spam. It was filling up everyone's mailboxes and was using more and more disk space and processing power. The Information Risk Management department (IRM) along with the MsExchange group had identified that approximately forty percent of our email was spam. The Help Desk was getting many calls from employees who were upset by the spam they received. Spam was causing a loss of productivity.

Problem #2: FAILOVER

Our sendmail server had no failover so if it died, our mail stopped. It did not die very often, mind you, maybe once or twice a year, but when it did, it was a big deal since people rely on email these days for important business communications.

Problem #3: ANTI-VIRUS CHECKING

Our anti-virus software was located on the MsExchange servers, meaning that viruses made it through the firewall, sendmail gateway and the internal PMDF gateway before stopped. It would be nice to stop them earlier.

Problem #4: UPGRADING

When I originally installed sendmail on the firewall, I chrooted it so that if anyone did manage to exploit a bug in sendmail, they would only have access to a separate root directory which only contained the files needed by the sendmail program and not to other important system files. Although the process of chrooting sendmail was fun, it was very time consuming and involved several hours of running truss on the sendmail binary and much testing in order to try to discover a complete list of all the files that sendmail used.

That's why I always chose to patch sendmail rather than upgrade it to the newest versions and why I opted not to chroot when I finally upgraded to 8.12.10. (I didn't feel that security was being jeopardized, though, because the server was no longer on the firewall, the new version had quite a few added security features, and I wasn't running the daemon as suid. But nevertheless, I would have chrooted it anyway had it been easier to maintain.)

Problem #5: PATCHES

Keeping the sendmail server up-to-date with security patches was a bit of a problem too. Since it was a hardened server, it was a Core Solaris install and was stripped right down to the bare bones. There were no compilers installed so the patch procedure involved moving the latest source code using sftp from the sendmail server to a server which did have a compiler, downloading the patch from the sendmail website[v], running pgp on the patch to make sure it had not been tampered with, applying the patch to the source code, recompiling the source code, moving it back to the sendmail server, and applying the changed files to the chrooted filesystem. It was a fairly complicated procedure.

The Solaris OS also had to be kept up to date with the latest security patches, as did the other security software we had installed on the box (Tripwire and Openssh), and a similar patch procedure as above needed to occur.

As we all know, patches need to be applied immediately after they are discovered or the vulnerabilities could easily be exploited. We weren't keeping up with the task.

During

Proposed Solution

The Information Risk Management (IRM) department is in charge of setting security policies, doing Forensic Analysis, and leading Incident Response for my

company. In this case, they were asked to lead a project to solve our spam problem.

A team was formed, headed by IRM, with a resource from the Network Support Department for firewall and dmz work, the MsExchange area of Enterprise Computing for assistance with the spam setup, and myself as the sendmail gateway resource from Enterprise Computing.

The IRM group identified through research and peer groups, that Ciphertrust's IronMail product would be a good one to try. The IronMail appliance is a comprehensive mail tool, as described on Ciphertrust's home page:

> *The company's award-winning IronMail appliance combines the five critical e-mail security components of spam and fraud prevention, virus and worm protection, policy and content compliance, e-mail privacy, and secure e-mail gateway capabilities into a single easy to deploy and manage platform*.

Brief Overview of SPAM Detection in IronMail

IronMail uses several highly configurable anti-spam tools: deny lists, reverse dns, realtime blackhole lists, statistical lookup service, system defined header analysis, user-define header analysis, sender policy framework (SPF), bayesian filtering, and content filtering. I won't go into a description of all of these methods, but will describe a few that were new to me:

Statistical Lookup Service (SLS)

Every company that has an IronMail box, if they choose to enable this service, will send a hash of every email it receives to the Ciphertrust SLS servers. The SLS servers keep track of how many times it has received each email and sends that value back. Administrators decide on a threshold that will determine when an email is considered spam. Apparently, this is the spam-detection method that is the most effective and has the fewest false-positives even when using a small threshold value of 10. Of course, mailing-lists are often labeled as spam this way, so administrators are advised to quarantine rather than reject the mail at first and then add the mailing-list ips to whitelists for the future.

Sender Policy Framework (SPF)

This tool checks the sending domain as found in the envelope against the IP address in the from address. This prevents spammers from forging the from address or using one that doesn't exist at all.

Each email arriving at the IronMail server is checked by these tools, and a configurable action will be taken if identified as spam. IronMail can be configured to either take action as soon as the first tool identifies it as spam, or it can be configured to use confidence-based detection and blocking which is what we do. In this case each email goes through every tool, and a score is calculated based on how many tools identified it as spam. This method is better at reducing false-positives.

Solution Implementation and Testing

It seemed to make sense that we should investigate the spam-fighting options in sendmail before we bought anything else, so I did a bit of research. The only anti-spam feature that I could find that I hadn't already implemented was a realtime blackhole list (rbl) option that would check every sender to one or more specified rbl lists out on the internet, and if found on a list, the mail would be dropped. It was implemented in the m4 sendmail config file:

FEATURE (`dnsbl',`sbl.spamhaus.org',`"550 Email rejected due to Spam"')

I ran our sendmail gateway with this feature turned on for a few minutes one evening and was very impressed with the amount of spam that it caught in such a short amount of time. I proposed we implement it as a solution for the time being, but the main problem was that there was no option to quarantine the mail rather than drop it. So if a sender had erroneously been placed on the rbl list, we would have dropped legitimate mail and that would not be acceptable to our company. I'm sure that with some fancy tweaking of the sendmail configuration file, it could be done, but I already had issues with the maintenance of sendmail and this would not have helped matters much.

It hadn't been identified where in the mail flow the IronMail appliances would go, so I went to the web site and found out that it could replace the sendmail gateway.

I didn't want to replace the sendmail gateway unless I was sure that IronMail had all the same security features as I had implemented on the sendmail server. After reading the documentation and talking to IronMail engineers, I was satisfied that it did. The following are the list of security features we had for the sendmail server and the corresponding one available in the IronMail server:

- Firewall protection
    a. Dmz: The sendmail server is in a protected dmz where only the necessary smtp ports (port 25) were opened on the firewall to allow mail through.

The IronMail servers also sit in the dmz. There are a few more ports opened on the firewalls, which don't compromise security.

1. port 20022 between Ciphertrust server to IronMail servers for troubleshooting during evaluation period.
2. Port 6277 between IronMail servers and Ciphertrust servers for SLS (Statistical Lookup Service):

b. The application inspection "Mailguard"[vi] feature on the Pix firewalls was enabled, which restricts the smtp commands that the mail server can receive to HELO, MAIL, RCPT, DATA, RSET, NOOP and QUIT. This prevents spammers or hackers from trying to gather information about our company by using other commands such as VRFY and EXPN. The Mailguard feature also changes the characters in the SMTP banners to asterisks, also to prevent giving out any information that could be used against us.

This feature is still in place with our IronMail servers.

- Anti-Virus software on our MsExchange Servers, updated daily

  The IronMail server has virus-checking software built in (you can chose between Mcafee or Sophos). It has an automatic update feature where you can specify (in hours) how often to check for and download new virus signatures.

- Sendmail is installed on a hardened Solaris server

  The IronMail appliance runs a hardened version of Unix. From their website:

  > We have created an operating system that is custom hardened for our environment. It contains only the services and components needed by IronMail. We integrated our custom application software, written for extremely high security, with this operating system. We continually test for vulnerabilities and exploits, regularly attacking the appliance. We do this with internal resources, and well as outsiders, from leading security organizations such as VeriSign. Only such a rigorous process can provide enterprise-class security.

- Secure shell (openssh) access for administration

IronMail has https access to their web administration tool, where most of the work is done. It also has ssh access to their command line interface (CLI) on the appliance.

- Tripwire run daily (host-based intrusion detection)

  IronMail has host based intrusion detection which is run nightly. It also has a built in network-based IDS, which constantly monitors for attacks. There is also a vulnerability assessment feature that can be run at any time to scan the IronMail server for vulnerabilities.

- Sendmail specific security features:
  a. Anti-relay by default
     Configurable as an anti-spam tool in IronMail.
  b. Only accepts mail from known hosts (reverse dns lookup)
     Configurable as an anti-spam tool in IronMail.
  c. Will not run unless permissions on sendmail files are set correctly
     IronMail's host-based IDS checks for changes in permission.
  d. M4 sendmail config file options:
     i. `define(`confPRIVACY_FLAGS',`goaway,authwarnings,noetrn')`

        Disables most SMTP commands such as VRFY and EXPN which could be used by someone outside corpA to try and discover corpA user information

        This is done on the firewall as well, so we are still covered.

     ii. `define(`confMAX_MESSAGE_SIZE',`15000000')`

        Sets the maximum allowable message size to 15 mb. This helps to prevent denial of service by receiving mail messages larger than the available space in the /var/spool filesystem.

        Configured in the SMPTI service on IronMail. Can specify 2 different values, one for users outside our domain and one for those inside.

     iii.
```
MASQUERADE_AS(`corpA.com')
MASQUERADE_DOMAIN(`corpA.com')
MASQUERADE_DOMAIN(`corpA.net')
FEATURE(`masquerade_entire_domain')
```

        The Masquerade commands force all email from corpA to look like it is coming from user@corpA.com. Without this, if

it was coming from a unix server, say kluane.corpA.net, it would look like it came from user@kluane.corpA.net, revealing information from the internal network that we don't want to give out.

IronMail has an Address Masquerade feature.

iv. `FEATURE(`smrsh')`

Enables smrsh, the sendmail restricted shell program. Disallows mailing to programs (i.e. .forward, calendar,etc) rather than users. Only the programs contained within the smrsh directory would be allowed, and we didn't allow any.

I'm assuming this is covered in IronMail since it is a hardened Unix server.

v. `FEATURE(`access_db')`

Use the access database to reject or accept mail from specific sites.

```
corpA.net          RELAY
corpA.com          RELAY
paypal.com         DISCARD
Connect:localhost  RELAY
```

It is allowing mail from our corpA domains and is rejecting mail from the paypal domain. The access database is a powerful feature and can be used for rejecting mail to or from specific users, sites, networks, etc. It is also possible to reject mail with a text message attached

The relay servers are specified in the mail configuration panel of IronMail. IronMail has a "local deny list" where you can specify senders you want to reject.

The project team received a day of training from an IronMail engineer who flew onsite.

We implemented the IronMail appliances in a two-phased approach. During the first phase we inserted it into the dmz alongside the sendmail server and redirected all the mail going to a high-volume spam receiver to the IronMail server. I did this using the VIRTUSERTABLE feature of sendmail. I also had configured the test PMDF server and we had a test Msexhange server in the flow, so we were not impacting any of the production servers.

When we were satisfied that it would work, we implemented Phase II which was to replace the sendmail server with the IronMail box. The implementation went fairly smoothly. We upgraded to a new version of IronMail a few weeks later and at the same time implemented a failover IronMail server (the failover was accomplished using MX records on our external dns provider).

I recently installed a patch to our IronMail servers and it was as easy as clicking a few buttons on the web gui to download the patch and install it. It did reboot the system so there was a short downtime but since we had failover servers the mail didn't stop.

Another problem we encountered is that IronMail has a hard-limit of 500 for mailing lists. If a list has more than that it simply drops the extras. I implemented a feature on the PMDF messageing server that breaks up large mailing lists into chunks of 100, and this fixed the problem.

The IronMail engineers provided a best practice guideline on how to configure the spam filters and that is what IRM implemented originally but only in "logging" mode—i.e. the spam was identified but still getting through. By doing this, IRM could carefully decide on their plan of action and watch for any "false-positives" that might occur. One of the first things they noticed was that the statistical lookup service tends to label mail coming from large mailing lists (eg the Sans mailing lists) as spam. So they needed to whitelist a lot of the mailing lists that people in our company subscribe to. They also implemented a process where the subject line for spam email is rewritten indicating that it has been detected as spam and including the accumulative spam score that IronMail has computed for it (which is an indication of how offensive it is). If a user doesn't agree that the email was spam, they can report it and IRM will decide whether to whitelist it (only business related emails will be whitelisted). Recently, we have started to drop and quarantine the spam that has the highest scores.

Conclusion

Although I'm a big sendmail fan and appreciate how huge a contribution it has made to the email world, I'm really glad we switched to IronMail. For all I know, the IronMail system may be running sendmail as its MTA. On the Ciphertrust web page they say "Integration of a high-performance mail transfer agent (MTA) allows the IronMail secure platform to effectively manage and route millions of e-mail messages a day".

The only issue I have with switching to an appliance is trust. It's a black box. How do we know for sure that it's been hardened correctly or if they supply patches promptly for vulnerabilities in their servers? The thing with doing it all yourself, like I did with the sendmail server, is that you know exactly what OS and application hardening procedures have been applied and you have total control

over the security of the box. Going to an appliance feels a little bit like passing the buck, but most people just don't have the time nowadays to spend babysitting the security of a custom installed server.

The spam features, the reporting, the IDS, the easy maintenance, the virus checking, the easy configuration, and the many other features that we haven't even explored yet, make IronMail a great tool to have.

The following statistics show the amount of spam we are now catching that used to get through to mailboxes. We are still getting about forty percent of spam, but every month, as we continue tuning the IronMail spam filters, we are stopping more and more from getting in.

Action Taken	July	August	September
Viruses Quarantined	238,397	73,935	45,976
SPAM Quarantined	24,698	149,783	152,867
SPAM Dropped	122	10,475	107,537
Total SPAM Stopped	24,820	160,258	260,404

Spam is not getting in, we have failover now, the virus checking is being done "at the door" now rather than far inside our network, and patching and upgrading is much easier. Problem solved.

References

[i] Garcia, Dan. "Dan Garcia's Spam Homepage".
URL: http://www.cs.berkeley.edu/~ddgarcia/spam.html#jargon

[ii] Ciphertrust, Inc. URL: http://www.ciphertrust.com/

[iii] Fennely, Carole. "Setting up Sendmail on a Firewall- Part 1". April 1999.
URL: http://www.wkeys.com/articles/swol/Apr_99.html
Fennely, Carole. "Setting up Sendmail on a Firewall- Part 2". May 1999.
URL: http://www.wkeys.com/articles/swol/May_99.html
Fennely, Carole. "Setting up Sendmail on a Firewall- Part 2". June 1999.
URL: http://www.wkeys.com/articles/swol/Jun_99.html

[iv] Process Software. "Process Software – PMDF".
URL: http://www.process.com/tcpip/pmdf.html

[v] The Sendmail Consortium. "Sendmail Home Page". URL:
http://www.sendmail.org

[vi] Cisco Systems, Inc. "Configuring Application Inspection (Fixup)" URL:
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_sw/v_63/config/fixup.htm#wp1103488

Last Updated: May 27th, 2017

Upcoming Training

SANS Stockholm 2017	Stockholm, Sweden	May 29, 2017 - Jun 03, 2017	Live Event
Security Operations Center Summit & Training	Washington, DC	Jun 05, 2017 - Jun 12, 2017	Live Event
SANS Houston 2017	Houston, TX	Jun 05, 2017 - Jun 10, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Jun 05, 2017 - Jun 10, 2017	Community SANS
SANS San Francisco Summer 2017	San Francisco, CA	Jun 05, 2017 - Jun 10, 2017	Live Event
SANS Charlotte 2017	Charlotte, NC	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Rocky Mountain 2017 - SEC401: Security Essentials Bootcamp Style	Denver, CO	Jun 12, 2017 - Jun 17, 2017	vLive
SANS Secure Europe 2017	Amsterdam, Netherlands	Jun 12, 2017 - Jun 20, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Jun 12, 2017 - Jun 17, 2017	Community SANS
SANS Rocky Mountain 2017	Denver, CO	Jun 12, 2017 - Jun 17, 2017	Live Event
SANS Minneapolis 2017	Minneapolis, MN	Jun 19, 2017 - Jun 24, 2017	Live Event
SANS Columbia, MD 2017	Columbia, MD	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS Cyber Defence Canberra 2017	Canberra, Australia	Jun 26, 2017 - Jul 08, 2017	Live Event
SANS Paris 2017	Paris, France	Jun 26, 2017 - Jul 01, 2017	Live Event
SANS London July 2017	London, United Kingdom	Jul 03, 2017 - Jul 08, 2017	Live Event
Cyber Defence Japan 2017	Tokyo, Japan	Jul 05, 2017 - Jul 15, 2017	Live Event
SANS Cyber Defence Singapore 2017	Singapore, Singapore	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Minneapolis SEC401	Minneapolis, MN	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Los Angeles - Long Beach 2017	Long Beach, CA	Jul 10, 2017 - Jul 15, 2017	Live Event
Community SANS Phoenix SEC401	Phoenix, AZ	Jul 10, 2017 - Jul 15, 2017	Community SANS
SANS Munich Summer 2017	Munich, Germany	Jul 10, 2017 - Jul 15, 2017	Live Event
Mentor Session - SEC401	Ventura, CA	Jul 12, 2017 - Sep 13, 2017	Mentor
Mentor Session - SEC401	Macon, GA	Jul 12, 2017 - Aug 23, 2017	Mentor
Community SANS Colorado Springs SEC401	Colorado Springs, CO	Jul 17, 2017 - Jul 22, 2017	Community SANS
Community SANS Atlanta SEC401	Atlanta, GA	Jul 17, 2017 - Jul 22, 2017	Community SANS
SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANSFIRE 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Jul 24, 2017 - Jul 29, 2017	vLive
Community SANS Charleston SEC401	Charleston, SC	Jul 24, 2017 - Jul 29, 2017	Community SANS
Community SANS Fort Lauderdale SEC401	Fort Lauderdale, FL	Jul 31, 2017 - Aug 05, 2017	Community SANS
SANS San Antonio 2017	San Antonio, TX	Aug 06, 2017 - Aug 11, 2017	Live Event
SANS Prague 2017	Prague, Czech Republic	Aug 07, 2017 - Aug 12, 2017	Live Event