Global Information Assurance Certification Paper

HOWTO: Spam/Virus (Sendmail/SpamAssassin/MIMEDefang/ClamAV) defense perimeter in Solaris

GSEC Practical Assignment, Version 1.4b, Option 1
Ming-Yang Hou
March 29, 2004

Abstract

What is spam? According to CMP TechWeb's definition, it is "E-mail that is not requested."[1] Indeed, everyday, we often receive about 10, 20 or more unsolicited email messages. These messages, so-called "junk mails", seat in our INBOX and spend our time to process one by one. How about viruses? According to CERT's virus analysis report[2], more and more viruses are from email attachments, and most of these emails are spam. Consequences of virus attack not only include loss of important data, but also make hosts affected by virus and become Denial of Service (DoS) to their network resources.

The purpose of this paper is to describe my experience of setting up mail gateway for my friend's small business with spam filtering and anti-virus solutions. I hope that it will help those who would like to deploy anti-spam and anti-virus capabilities with Sendmail under Solaris environment in the future to secure their email systems.

Motivation

Based on "Spam Statistics 2004" [3], it is not difficult to find out how frequent we get spam mails everyday, and according to an article: "Can Spam Act Likely to Increase Record Levels of Spam."[4], the volume of spam is still increasing.

Traditionally, there are two ways to stop spam. First, we can use Mail Transfer Agent (MTA) to block the specified subjects, and/or the specified keywords/signatures in the contents of email messages. Secondly, we can block IP addresses of spammer email sources through Firewalls or Routers. Today, none of the above solutions is capable of slowing down spammers to spam their email messages. The reasons are simple: spammers are constantly changing their email subject lines or contents, and/or they fully utilize dynamic IP address assigned from Dialup, Cable, or DSL connection. Nevertheless, spam content filter software is still a necessary since the software can analyze the contents of a message based on some spam statistical rules to determine if the message is a spam or not.

The implementation of anti-virus solution in the email gateways has applied to most corporate and school environments. The advantage: a virus scanner can guard inbound and/or outbound email traffic to reduce amount of virus passes through either Internet or Intranet, especially "email attack" type of viruses like NetSky, Mimails, Bagel, or Mydoom. Without it, same issues will

[1] CMP
[2] Carpenter, Jeff. "CERT Advisory CA-2004-02 Email-borne Viruses"
[3] SPAM FILTER REVIEW
[4] MessageLabs

be repeated in future virus attacks[5].

Initial Evaluation

The production environment is an Intel machine (Pentium 200Mhz, 256 MB RAM) with Solaris 8 installed. Software support for Solaris x86 arc hitecture is limited, especially virus scan software. Therefore, Open -source code is the only option to solve " Limited support" issue.

After searching spam filter at http://www.google.com /, it shows that SpamAssassin is a popular open -source project[6] for spam filter . This software determines the spam by scoring based on the message header and body, and tags spam report into the message header for people who would like to know how SpamAssassin rates spam message. In addition, SpamAssassin 's Web site also describes key features regarding how to identify spam and how to cooperate with other spam solution modules into SpamAssassin to increase spam detection score. Another successful story using SpamAssassin to intercept spam is in Greg Williamson's SANS GSEC paper[7] for his home business and a large German hospital.

Sendmail is my default MTA. MIMEDefang is one of recommended MTA mail scanners by SpamAssassin that has fully implemented Sendmail Mail Filter (Milter).[8] It provides several basic features to handle spam and virus, such as spam filtering, spam rejection, email message content manipulation, and virus detection by calling the third -party Anti-Virus software[9].

To block spam in MTA level is another advantage by using Sendmail. How does Sendmail determine if an incoming mail message is spam or no t? Sendmail sends a Domain Name System (DNS) query that contains the sender's IP address information to Realtime Block List (RBL) or/and DNS Block List (DNSBL) sites. A RBL or DNSBL DNS server response contains 127.0.0.x reply, it means that the sender 's IP address is recorded as a spam. In addition, Sendmail implemented RBL in 8.9 first, and changed to DNSBL in 8.10.[10] Although SpamAssassin did utilize RBL and DNSBL tech niques, the implementation of RBL and DNSBL in Sendmail make s quicker email rejection to achieve better system performance and save more overhead system resource s. The reason is that Sendmail can detect if the sender 's relay IP address is really from a s pam site without calling SpamAssassin.

Most Anti-Virus (AV) software support both Linux and Windows. Although some commercial AV products do support Solaris OS, the majority of them are developed for SPARC platform. Among all, Clam AV provides source codes download, and it is free. In addition, based on search results from Google.com , Clam AV is one of many AV products supported by MIMEDefang. It has been tested on most UNIX environments includ ing

---

[5] Cole, Eric. P. 309
[6] Railsb ack, Kevin
[7] Wil liamson Greg
[8] www.spamassassin.org . "MTA -Level Integration (Site -Wide Use)"
[9] Roaring Pe nguin
[10] Sendmail.org, "Multiple D NS Blacklists"

Solaris OS Intel version. [11]

Figure 1 shows an email process flow, and overall relationship between each component. Details of each co mponent will be described in "Component Configuration/Installation" section.

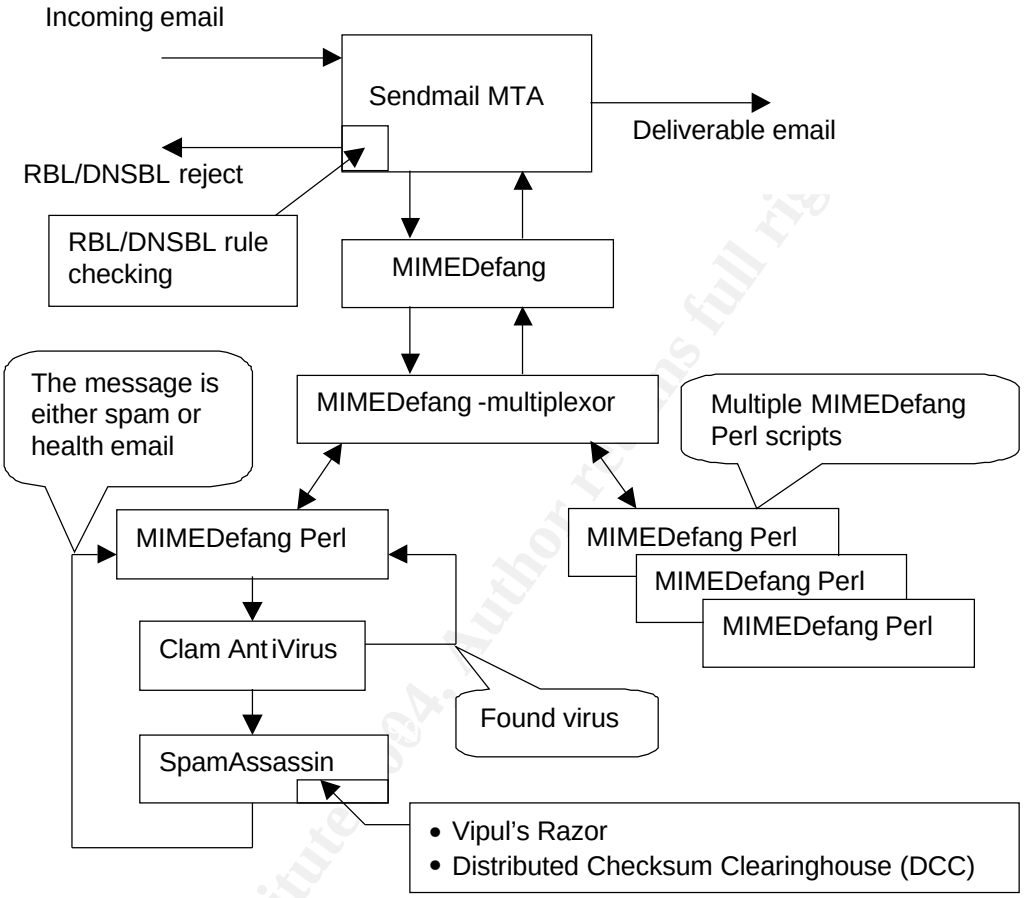

Figure 1: Email Process Flow

Component Conf iguration/Installation

Before performing software configuration and installation, it is recommend ed to apply Solaris OS patches to ensure bug-free in UNIX software (bugs like C header files, or runtime libraries). Some software may be benefited from the new changes in Solaris OS and software configuration. From the secu rity standpoint, keeping Solaris OS up-to-date can reduce vulnerabilities and prevent attacks from threats. [12] SunSolve's "Solaris Patch Clusters" gathers most important patches in one package for Solaris System Administrator's convenience. The package is available at SunSolve Patch Access. [13]

To build Sendmail and Milter packages, C compiler, Perl, and GNU patch are required for their configurations and compilations. By default, C compiler does

[11] Kojm, Tomasz. "Supported p latforms"
[12] Cole, Eric. P. 305
[13] http://sunsolve.sun.com/pub -cgi/show.pl?target=pa tches/patch -access

not ship with Solaris OS software as a standard package. In addition, Perl version in Solaris 8 OS is old and it does not provide up-to-date Perl modules to support Milter packages. Moreover, GNU patch command utility is needed for recommended patches. Sunfreware.com provides most precompiled software packages for Solaris OS users, and the instructions for installing packages are shown in Figure 2:

```
$ /usr/bin/gunzip gcc -2.95.3-sol8-intel-local.gz[1]
$ /usr/bin/gunzip perl -5.8.0-sol8-intel-local.gz[1]
$ /usr/bin/gunzip patch -2.5.4-sol8-intel.local.gz[1]
$ su
# /usr/sbin/pkgadd -d gcc-2.95.3-sol8-intel-local
# /usr/sbin/pkgadd -d perl-5.8.0-sol8-intel-local
# /usr/sbin/pkgadd -d patch-2.5.4 -sol8-intel-local
```

Figure 2: Tool Installation

Sendmail 8.12.11

Sendmail 8.9+sun is the default MTA comes with Solaris 8. Upgrading Sendmail to the latest version is very important due to fixes for bugs and security issues. In particular, Sendmail release notes for fixing a buffer overflow existing in Sendmail 8.12.10. [14] The source package of Sendmail is at ftp://ftp.sendmail.org/pub/sendmail/sendmail.8.12.11 .tar.gz.

There are two preparations before configuring Sendmail package. First, Solaris OS 8 does not provide Berkeley DB Application Programming Interface (API). To install Berkeley DB is useful for Sendmail to read/write its databases by using different formats, such as btree or hash. Sendmail databases are like access, domaintable, or virtualusertable etc.

DB package download site is at http://www.sleepycat.com/update/snapshot/db-4.2.52.tar.gz. Unlike other GNU software, to generate DB package's configuration requires two extra steps (see the command lines in red in Figure 3) to produce a "Makefile" file. After that, the compilation and installation can proceed.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip –dc db-4.2.52.tar.gz | /usr/sbin/tar --tvf -
$ cd db-4.2.52/build_unix
$ ../dist/configure –prefix=/usr
$ make
$ su
# PATH=/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 3: Berkeley DB Library Configuration and Installation

Second, enabling Milter is required. Sendmail provides Milter API in order to inter-exchange SMTP information between Sendmail and third-party spam filters. MIMEDefang is one of the spam filters that implement Sendmail Milter

[14] Sendmail.org, "SENDMAIL RELASE NOTE"

API. To enable Milter API feature in Sendmail , add the following two statements into a file : Sendmail-8.12.10/devtools/Site/site.config.m4:

```
APPENDDEF (`conf_sendmail_ENVDEF', ` -DMILTER')
APPENDDEF (`conf_libmilter_ENVDEF', ` -D_FFR_MILTER_ROOT_UNSAFE')
```

The first line is to set a flag for generating Milter API library during Sendmail compiling time. The next line is to ensure that a spam filter program integrated with Sendmail Milter API will never run as a root privilege based on Sendmail's security recommendation. [15] To build Sendmail package , see Figure 4.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc sendmail-8.12.10.tar.gz | /usr/sbin/tar --xvf -
$ cd sendmail-8.12.10
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ ./Build
$ su
# PATH=/usr/local/bin :/usr/ccs/bin:$PATH; export PATH
# /etc/init.d/sendmail stop
# ./Build install
```

Figure 4: Sendmail Package Configuration and Installation

Two Sendmail configuration files, sendmail.mc and submit.mc (especially sendmail.mc), will determine how MTA to receive, deliver, and send emails. Most important rules used by Sendmail are defined in sendmail.mc, such as stopping itself to relay outside emails, incoming email domain verification, RBL/DNSBL rules, and other features. For submit.mc, it is used only when Sendmail is a local mail client for sending out email from the local machine. Both mc files can be found in Appendix "Sendmail - sendmail.mc" and "Sendmail - submit.mc". Figure 5 shows how to generate sendmail.cf and submit.cf from sendmail.mc and submit.mc and copy them to /etc/mail.

```
# cd ../../cf/cf
# ./Build install-cf
```

Figure 5: sendmail.mc and submit.mc installation

For handling a local mail submission, Sendmail 's start-up script, /etc/init.d/sendmail, is needed to insert additional statement, such as:

```
…
/usr/lib/sendmail $MODE -q$QUEUEINTERVAL $OPTIONS &
/usr/lib/sendmail –Ac -q$QUEUEINTERVAL & (insert this line)
…
```

Due to SMTP security concern, you need to verify if MTA is not an open relay . This is extremely important after upgrading Sendmail. Also, make sure that sendmail.cf rule has been configured properly. Sometimes spammers may

[15] Sendmail.org. "SECURITY HINTS"

scan Internet to find available mail servers. Therefore, the syslog file may record some information, such as “did not issue MAIL/EXPN/VRFY/ETRN during connection to MTA”. It means this mail server with the port 25 is listening requests. After that, spammers will try to use these mail gateways as their spam relays to spam emails all over Internet if these mail gateways are open relays. As a result, these open relay mail gateways may be listed in DNSBL sooner or later after people complains spam messages through proper anti-spam channels. Network Abuse Clearinghouse provides “Mail relay testing” [16] to against MTA setting by using 17 different relay tests.

To setup Sendmail for DNSBL lookup. The syntax in sendmail.mc looks like:

FEATURE(`dnsbl',`_DNSBL_SRV_',`_DNSBL_MSG_')

“_DNSBL_SRV_” is a DNS server where it stores DNSBL cache information based on reversed IP address with DNSBL domain name. “_DNSBL_MSG_” is to provide necessary explanation to the sender when Sendmail rejects the message. For instance:

FEATURE(`dnsbl',`sbl.spamhaus.org',`"550 Mail from " $&{client_addr} " refused by blackhole site sbl.spamhaus.org. Please see http://www.samspade.org/t/rbl?a=" $&{client_addr} " for more information."')

The main purpose for supplying detailed DNSBL response message is to provide useful information for some legitimate senders in order for they can inform their system administrators or Internet Service Provider (ISP) to remove their mail servers out from DNSBL service providers based on Sendmail rejected information.

Where are DNSBL databases available on Internet? Some Web sites maintain DNSBL information in public for who would like to implement DNSBL verification in their mail servers. These Web sites are like http://rbls.org/, http://www.declude.com/junkmail/support/ip4r.htm, or http://moensted.dk/spam/. In general, DNSBL databases have several categories, like Cable, DSL, Dialup, open relays, open proxy servers, formmail scripts, and etc. To increase spam-blocking rate, it is a good idea to utilize different databases from different DNSBL sites in sendmail.mc. Another important thing is to view Sendmail log file frequently because some DNSBL databases are too aggressive to block spam so that legitimate emails may be included as well.

SpamAssassin 2.63

SpamAssassin is sets of Perl programs in order to provide flexible and powerful text analysis. [17] Not only SpamAssassin performs a content filter task, but also it cooperates with other spam packages (like Vipul's Razor and Distributed Checksum Clearinghouse (DCC)), if available.

Vipul’s Razor provides a spam signature database over Internet and allows

[16] Network Abuse Clearinghouse
[17] SpamAssassin Wiki. “SpamAssassin”

the registered users to submit Razor's spam signatures and confidence (cf) values based on the demonstration of the existence of spam messages. [18] An anonymous Razor client can inquire a cf value associated with individual signatures to use against its local message signature. Vipul's Razor is a SourceForge project and its source code package is available at SourceForge site.[19] To compile Razor package (Figure 5), it requires other related Perl modules that are listed in its installation instructions.[20] Instead of downloading individual modules from CPAN, Razor software development kit (SDK) [21] provides an all-in-one package to configure/install these modules to their proper Perl module directories.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc razor-agent-sdk-2.0.3.tar.gz | / usr/sbin/tar -xvf
–
$ cd razor-agent-sdk-2.0.3
$ perl Makefile.PL; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install; exit
$ cd ..
$ /usr/bin/gzip -dc razor-agent-2.0.3.tar.gz | /usr/sbin/tar -xvf –
$ cd razor-agent-2.0.3
$ perl Makefile.PL; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install; exit
```

Figure 5: Vipul's Razor Package Configuration and Installation

Like Razor, DCC is another SPAM filtering network. DCC has introduced "fuzzy" checksum to identify spam without referencing the exact contents of the message. [22] A DCC client reports checksums of the message to a DCC server. The DCC server totals the report of the checksums of messages and feeds back a total number of recipients of the mail. Therefore, this DCC client compares the total count with its threshold to determine the message is spam or not. Figure 6 shows DCC configuration and installation procedures. DCC package is also free for download at Rhyolite.com [23].

[18] Vipul's Razor. "Vipul's Razor v2 README"
[19] http://prdownloads.sourceforge.net/razor/razor-agents-2.03.tar.gz?download
[20] Vipul's Razor. "Vipul's Razor v2 Installation Instructions"
[21] http://prdownloads.sourceforge.net/razor/razor-agents-sdk-2.03.tar.gz?download
[22] Distributed Checksum Clearinghouse. "Functions"
[23] http://www.rhyolite.com/anti-spam/dcc/source/dcc-dccproc.tar.Z

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/uncompress -c dcc-dcproc.tar.Z | /usr/sbin/tar -xvf -
$ cd dcc-dccproc-1.2.27
$ ./configure; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 6: DCC Package Configuration and Installation

SpamAssassin installation is quite simple, and its source code can download from Spamassassin.org.[24] Before configuring SpamAssassin, it is suggested by the README file within the package to first patch Razor2 Perl module. Razor2 is not fully taint safe since SpamAssassin 2.60 enables taint mode by default. In Figure 7, it shows how to configure and install SpamAssassin step-by-step. Be aware of the lines in red. It indicates Razor2 patch must be patched before configuring SpamAssassin because SpamAssassin configuration script will prompt for testing Razor2.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc Mail-SpamAssassin-2.63.tar.gz | /usr/sbin/tar -
xvf -
$ cd Mail-SpamAssassin-2.63
$ su
# /usr/local/bin/patch -p0 -d
/usr/local/lib/perl5/site_perl/5.8.2/i86pc-solaris/Razor2 <
Razor2.patch
# exit
$ perl Makefile.PL; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 7: Install Razor2 patch, SpamAssassin Package Configuration and Installation

SpamAssassin package also comes with two spam and non-spam test messages for users who need to verify SpamAssassin has been built and installed properly. There are two steps to test SpamAssassin script:

```
$ /usr/local/bin/spamassassin -t < sample-spam.txt > spam.out
$ /usr/local/bin/spamassassin -t < sample-nonspam.txt > nonspam.out
```

By examining the contents from “spam.out”, it shows that Spamassassin tags spam detection information into the message header by modifying the subject line and adding extra SpamAssassin status lines. In addition, SpamAssassin attaches a detail explanation of spam detection after the email message. DCC and/or Razor report is also a part of the explanation if DCC and/or Razor server is able to identify the email contains spam signature.

There is a spam message header listed in Appendix “MIMEDefang – SMTP Message Header”. “RAZOR2_CF_RANGE_51_100”, “RAZOR2_CECHK”, and “DCC_CHECK” are SpamAssassin rules, and SpamAssassin indicates

[24] http://www.spamassassin.org/released/Mail-SpamAssassin-2.63.tar.gz

spam signatures are found at Razor and DCC database servers.

To set up spam rules and preferences for SpamAssassin , SpamAssassin engine must be done in MIMEDefang runtime configuration file because MIMEDefang proce ss reads its own configuration and passes the configuration to SpamAssassin Engine. Details will be described in MIMEDefang section. In other words, all of SpamAssassin settings/configurations that are stored in /etc/mail/local.cf will not be read by MIMED efang anymore.

Clam AntiVirus 0.68 -1

Like most Anti-Virus software, Clam AV not only provides a console command-line scan program, it is also capable of scan ning contents of email messages.

Certainly, to scan a compressed attachment in email is a part o f feature in Clam AV. Since ZIP and BZIP2 libraries are available in Solaris 8 OS, C lam AV configuration script detects the existences of both libraries and enables a compressed archive feature into its binary execution codes. In addition, Clam AV has ano ther build-in feature to support RAR archive (another format for compressed attachment).

There is only one library that is strongly recommended by Clam AV in order to support Clam virus database digital signatures ; namely, GNU Multiple Precision (MP) arit hmetic library. MP can be found at GNU FTP site [25] and its configuration and installation are listed in Figure 8.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc gmp-4.1.2.tar.gz | /usr/sbin/tar -xvf -
$ cd gmp-4.1.2
$ ./configure; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 8: GNU MP Library Configuration and Installation

In Figure 1, a SMTP message is passed into MIMEDefang from Sendmail, MIMEDefan g will feed the message to Clam AV daemon process, called clamd. Basically, clamd is running in UNIX background and listening all incoming requests, then dispatching the requests to virus scan engine, called clamdscan. The result of virus scan will be pas sed back to clamd then to MIMEDefang.

The communication channel between MIMEDefang and Clam AV can be either UNIX or TCP socket. Since Clam AV daemon does not need to run as the root's privilege to perform virus scan and it supports an option to run as the specified selected user id. Therefore, it is a good practice to reduce any opportunity to be compromised if Clam AV processes are not running as the

[25] http://ftp.gnu.org/gnu/gmp/gmp -4.1.2.tar.gz

root id. For example, the daemon may run over its stack buffer while scanning a large file or the contents of the spam may contain a certain patterns which make the daemon process crash. For another security concern, instead of using TCP socket transmission, UNIX socket is sufficient to handle its local Inter-process Communication (IPC) between MIMEDefang and clamd, and it can prevent TCP attacks from the inside private network users.

Due to UNIX file permission, clamd and MIMEDefang must run as the same user and group ids in order for MIMEDefang accessing clamd's UNIX socket. In Figure 9, it shows how to add a new user and group ids for Clam AV and MIMEDefang.

```
$ su
# /usr/sbin/groupadd -g 8001 defang
# /usr/sbin/useradd -u 8001 -d /var/spool/MIMEDefang -m -g
defang -s /bin/ksh defang
# passwd defang
```

Figure 9: Create Clam AV and MIMEDefang Home Directory

Clam AV becomes a part of SourceForge project recently, and its source code is available for public download [26]. Figure 10 demonstrates Clam AV's configuration and installation.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc clamav-0.67.tar.gz | /usr/sbin/tar -xvf -
$ cd clamav-0.67
$ ./configure --prefix=/usr/local/clamav --with-user=defang --with-
group=defang –with-dbdir=/var/spool/MIMEDefang/
$ make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 10: Clam AV Package Configuration and Installation

Modifying Clam AV's configuration file is required in order to perform virus scan for emails or standalone files. The configuration file is clamav.conf under /usr/local/clamav/etc. In Appendix "Clam AV – clamav.conf", there is an example to illustrate the basic Clam AV settings.

It is highly recommended to retrieve Clam AV's latest virus databases right after Clam AV installation to ensure Clam AV daemon processes can identify latest virus signatures. In Figure 11, it shows how a single command-line can update Clam AV virus database.

[26] http://unc.dl.sourceforge.net/sourceforge/clamav/clamav-0.68-1.tar.gz

```
$ su
# /usr/local/clamav/bin/freshclam -u root –
datadir=/var/spool/MIMEDefang/clamav --daemon-notify
```

Figure 11: Update Clam AV Virus Database

UNIX cron command can automate schedule d download for the latest virus database. The following UNIX crontab statement is set to download the virus database every 4 hours, and email administrator for the update result.

```
59 2,6,10,14,18,22,23 * * * /usr/local/clamav/bin/freshclam -u root –
datadir=/var/spool/MIMEDefa ng/clamav --daemon-notify 2>&1 | /usr/bin/mail
administrator@some_domain.com
```

To make Clam AV daemon to be part of OS startup processes, "Clam AV – clamav" in Appendix, shows how to start and stop the daemon process. To take the contents from "Clam AV – clamav", make a file called clamav, and places it under /etc/init.d. To generate Clam AV daemon startup file , make a hard link from /etc/init.d/clamav to /etc/rc2.d/S85clamav, and make another hard link to /etc/rc0.d/K37clamav for shutting down Clam AV daemon.

MIMEDefang 2.41

MIMEDefang is a middleware between Sendmail and other mail filter applications. On one hand, MIMEDefang receives the incoming email message from Sendmail. On the other, it passes the message down to Clam AV daemon and SpamAssassin for virus scan and spam filtering. Based on returned result from either Clam AV daemon or SpamAssassin, MIMEDefang may take a proper action to handle the email message based on its filter rules. For instances, MIMEDefang can discard or quarantine a message that contains a virus with or without notifying the recipient, it can tag extra spam reports into the email message header for future reference.

In order to support the flexibility of Miter rules in MIMEDefang, the kernel of MIMEDefang process, "MIMEDefang Perl" in Figure1, is written in Perl script. Its front-end processor, "MIMEDefang" in Figure1, is , however, developed in C because the design of MIMEDefang takes advantage of Sendmail's Milter API without rewriting an interface to hook up with Sendmail daemon (Milter library configuration and installation are discussed in Sendmail 8.12.11 section). A dispatcher, "MIMEDefang-multiplexor" in Figure 1, is used to distribute all email messages to "MIMEDefang Perl" processes. You can download MIMEDefang package and its required/recommended Perl Modules from Mimedefang.org [27].

There are five Perl modules listed at MIMEDefang download site : MIME::tools patched version, IO::stringy, Mail::Tools, Digest::SHA1, and Unix::Syslog. Basically, some modules come with Perl 5.8.0 package. Few of them are not in Perl default module list, like IO::stringy and MIME::tools. According to MIMEDefang's download web site, MIME::tools patched version is the required module for installation or upgrade because MIMEDefang 2.22 has

[27] http://www.mimedefang.org/node.php?id=1

implemented the new feature based on the patched module. [28] Procedures to patch MIME::tools are shown in Figure 12.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip –dc MIME-tools-5.411a-RP-Patched-02.tar.gz |
/usr/sbin/tar -xvf -
$ cd MIME-tools-5.411a-RP-Patched-02
$ perl Makefile.PL; make
$ su
# PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
# make install
```

Figure 12: Patch MIME::tools Perl Module

The installations for other Perl modules use the same procedures like patching MIME::tools patch. There is another important module, Unix::Syslog, also required for installation, even though MIMEDefang download page says "it is an optional". Without installing Unix::Syslog module, MIMEDefang will generate an error and exit itself during runtime on Solaris OS.

Figure 13 is an example how to configure MIMEDefang software with the specified user id. The user id, defang, is already defined in Clam AV section.

```
$ PATH=/usr/local/bin:/usr/ccs/bin:$PATH; export PATH
$ /usr/bin/gzip -dc mimedefang-2.41.tar.gz | /usr/bin/tar -xvf -
$ cd mimedefang-2.41
$ ./configure --prefix=/usr/local/mimedefang –with-user=defang
$ make
$ su
# make install
```

Figure 13:MIMEDefang Package Configuration and Installation

It is inevitable to modify MIMEDefang Perl script and its configuration files. For example, to enable Razor spam checking in SpamAssassin requires to modify /usr/local/mimedefang/bin/mimedefang.pl at the line 90 and set "$SALocalTestOnly" to "0". Once "$SALocalTestOnly" is disabled, SpamAssassin will perform Razor spam checking and RBL/DNSBL verification. Since Sendmail is already doing RBL/DNSBL lookup, there is no need for SpamAssassin to do the extra work. To stop RBL checking in SpamAssassin engine, add or edit the option "skip_rbl_checks" to "1" in /etc/mail/spamassassin/sa-mimedefang.cf (see Appendix "MIMEDefang – sa-mimedefang.cf" for details).

SpamAssassin no longer reads its own configuration file. Instead, MIMEDefang process reads sa-mimedefang.cf from /etc/mail/spamassassin, and passes SpamAssassin parameters into SpamAssassin engine during runtime. In other words, MIMEDefang takes all of controls over SpamAssassin. For example, DCC process path and other DCC options need to be defined in sa-mimedefang.cf in order for MIMEDefang to pass information to SpamAssassin engine through DCC.

[28] Skoll, David F. "MIMEDefang 2.22-BETA-4 is available"

One of MIMEDefang features is to tag spam information into email message header. In Appendix "MIMEDefang – mimedefang-filter", it is a Perl subroutine to handle the result from Spam Assassin engine and it is at /etc/mail/mimedefang-filter. There are few changes in the subroutine: 1. Three lines (in red) need to be added, 2. Four lines (in blue) need to be commented out. The result of this modification will make MIMEDefang capable of adding extra spam detection information into each email message header, such as a detailed spam analysis information by tally, and tagging "***SPAM***" in front of the email message subject line. Appendix "MIMEDefang – SMTP Message header" shows an example of the email message that is tagged by MIMEDefang process. The commented out portions tell MIMEDefang process not to append spam analysis information into email message body, and to keep the current spam score within its email message header regardless it is spam or not. Most users do not want to know how SpamAssassin determines if a message is spam or not. Therefore, it is a good idea to tag spam analysis information into SMTP header to provide useful information for future studies regarding why some legitimate messages are marked as spam, or vice versa.

Users also write their own filter rules for virus and spam to satisfy different needs and plug into MIMEDefang Perl script. For example, MIMEDefang 2.41 has been changed to discard all incoming email message that contain virus. However, some users may want to quarantine these virus email messages, then they can apply back MIMEDefang 2.39's quarantine message rule to 2.41 in order to quarantine messages.

In order to start or stop MIMEDefang daemon processes when Solaris OS starts up or shuts down, like Clam AV, Appendix "MIMEDefang – mimedefang" script demonstrates start/stop tasks. The script needs to be copied to /etc/init.d/mimedefang first, then make two hard links to /etc/rc2.d/S86mimedefang for MIME Defang startup script and /etc/rc0.d/K37mimdefang for its shutdown script.

Hence, Clam AV daemon process is started up before MIMEDefang processes. And, MIMEDefang processes are ready running before Sendmail starts up.

Integration Test

An integration test is to ensure all components are working properly, like the entire email process flow shows in Figure 1. Several OS command-line utilities are helpful to monitor process runtime log information, or examine process run states. These two commands are:

ps - a command reports a process status. For instance, "ps -ef | grep defang | grep –v grep" will list all processes that contain the text, "defang". The result should return four MIMEDefang processes and one Clam AV daemon process (Figure 14), and it means all MIMEDefang processes are running normal. There is another daemon process needed to be verified, namely Sendmail. By using the ps command, you can tell how many Sendmail daemon

processes are running (Figure 14).

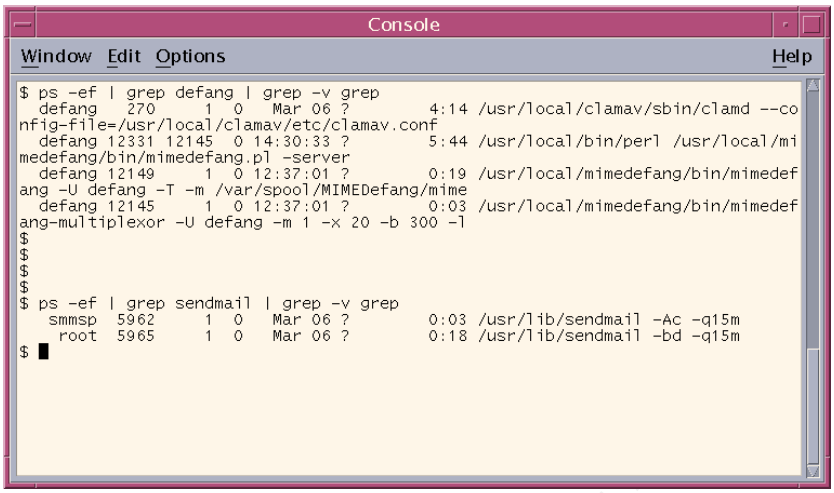

Figure 14: MIMEDefang and Sendmail process list

tail - a utility to view the last portion from a file at real-time. By default with Solaris OS, all email information related to MIMEDefang and Sendmail will be logged into /var/log/syslog. In the syslog file, it can trace a SMTP connection traffic from Sendmail to MIMEDefang, from MIMEDefang to Clam AV and SpamAssassin. Of course, the syslog may also log any failure information while processing a message.

To test if the current email gateway is able to tag spam message subject line, send or forward some junk mails to it. There is an example in Figure 15. These spam message subject lines are appended with "***SPAM***" in front of the original subject text.

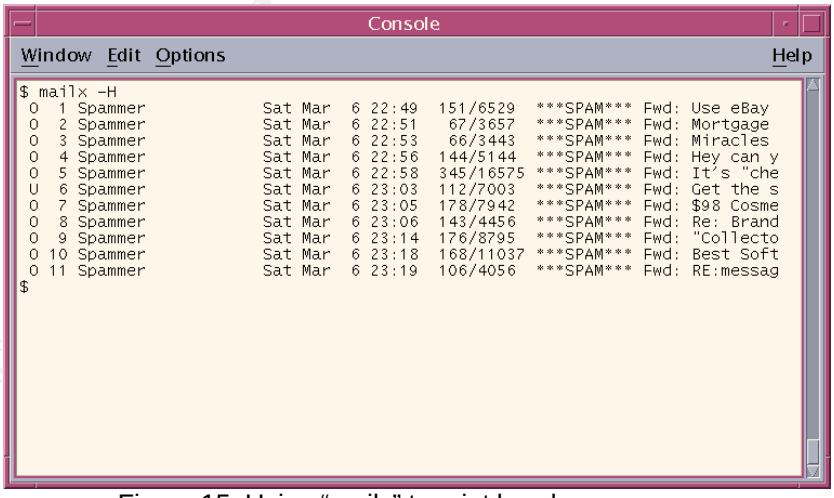

Figure 15: Using "mailx" to print header summary

Users can easily distinguish regular or spam messages with tagging subject lines. Most popular email clients, like Netscape and Outlook, have an ability to filter and move an incoming message with the specified keyword in the message header to a different folder instead of staying in the user's INBOX

folder. Therefore, users can easily set up their mail clients to filter spam message based on a subject line that contains "***SPAM***" text. As a result, users can process their important emails faster and worry about spam later.

OS and Softwa re Maintenance

There is an issue while using SunSolve "Solaris Patch Clusters" to upgrade Solaris OS. Since the default Sendmail 8.9+sun is replaced by the new Sendmail 8.12.11, patching "Solaris Patch Clusters" may override Sendmail 8.12.11 if patches con tains a new security fix for Sun Sendmail. In order to avoid Sendmail 8.12.11 to be changed by "Solaris Path Clusters" script , look for the sendmail patch ID number from a file, called CLUSTER_README that comes within the patch package and remove it . The following line may show in the cluster readme file:

```
110616-10 SunOS 5.8_x86: sendmail patch
```

The first column is Sun Sendmail patch ID number. This ID number must be removed from another file, called patch_order. After that, run "install_cluster" to per form Solaris OS patch installation.

Just like the reason for patching Solaris OS , it is very important to upgrade Sendmail, MIMEDefang, and Clam AV as well. It is a good ideal to subscribe each package's mailing list in order to be informed by email once there is a new version available. Sendmail, MIMEDefang, and Clam AV upgrade procedures are same as their new configurations and installations that are described in Component Configuration and Installation section. It is a good practice to backup the curr ent software before performing software updates in case the newer software does not work properly.

Conclusion

This implementation is successful for my friend's small business email gateway. After using MIMEDefang/Clam AV/SpamAssassin solution , 85 to 90 percent of spam messages are filtered out per day, and on the average, 1 to 4 seconds were used to process the entire message header and body for each email message. I also apply same implementation on Solaris Sparc platform, and it works the same way lik e Solaris Intel platform. For companies with heavy email traffic, they can use this implement ation on a faster Sun servers with multiple CPUs and large size memory to perform spam filtering and virus scan.

Appendix

Sendmail - sendmail.mc

```
divert(0)dnl
VERSIONID(`$Id: generic  -solaris.mc,v 8.13 2001/06/27 21:46:30 gshapiro Exp $')
OSTYPE(solaris2)dnl
DOMAIN(generic)dnl
INPUT_MAIL_FILTER(`mimedefang',   `S=unix:/var/spo ol/MIMEDefang/mimedefang.sock,
F=T, T=S:360s;R:360s;E:15m')
define(`confLOG_LEVEL',   `14' )
FEATURE(`always_add_domain')
FEATURE(`access_db')
FEATURE(`blacklist_recipients')
MAILER(`local')dnl
MAILER(`smtp')dnl
```

Sendmail - submit.mc

```
divert(0)dnl
VERSIONID(`$Id: submit.mc,v 8.6.2.4 2002/12/29 03:54:34 ca Exp $')
define(`confCF_VERSION', `Su  bmit')dnl
define(`__OSTYPE__',`') dnl dirty hack to keep pro    to.m4 from complaining
define(`confTIME_ZONE', `USE_TZ')dnl
define(`confDONT_INIT_GROUPS', `True')dnl
define(`confLOG_LEVEL', `14' )
```

Sendmail – database makefile

```
MAKEMAP=/usr/sbin/makemap
all: ac cess.db
%.db : %
      $(MAKEMAP) hash $@ < $<
clean:
      rm   -f access.db
```

Clam AV - clamav.conf

```
LogFile / var/log/clamd.log
LogTime
PidFile /var/run/clamd.pid
DataDirectory  /var/spool/MIMED efang/clamav
LocalSocket  /var/spool/MIMEDefang/ clamd.sock
FixStaleSocket
MaxDirectoryRecursion 15
User defang
AllowSupplementaryGroups
ScanMail
ScanRAR
ArchiveMaxFileSize 10M
ArchiveMaxRecursion 5
ArchiveMaxFiles 1000
```

Clam AV startup/stop script - cla mav

```
#!/sbin/sh
# Clam AV daemon start/stop script

remove_sock ()
{
      /bin/rm   -f /var/spool/MINEDefan g/clamd.sock > /dev/null 2>&1
```

```
}

case "$1" in
'start')
        /usr/local/clamav/sbin/clamd
        ;;

'stop')
        /usr/bin/pkill  -x -u 8001 clamd
        remove_sock
        ;;

*)
        echo "Usage: $0 {   start | stop }"
        exit 1
        ;;
esac
exit 0
```

MIMEDefang startup/stop script - mimedefang

```
#!/sbin/sh
#
# mimedefang
#

MULTIPLEX_SOCK="/var/spool/MIMEDefang/mimedefang    -multiplexor.sock"
MULTIPLEX_LOG="/var/log/mimedefang/multiplexor.log"
MIMEDEF ANG_SOCK="/var/spool/MIMEDefan  g/mimedefang.sock"

remove_sock()
{
      /usr/bin/rm   –f $MULTIPLEX_SOCK
      /usr/bin/rm   –f $MIMEDEFANG_SOCK
}

case "$1" in
'start')
        if [  -f /usr/local/mimedefang/bin/mi  medefang ] ; then
              # startup m  imedefang multiplexor
               OPTIONS1=" -U defang  -m 1  -x 5  -b 300  -l -d -s $MULTIPLEX_SOCK   -t
MULTIPLEX_LOG   -T"
                /usr/local/mimedefang/bin/mimedefang    -multiplexor $OPTIONS1

              # startup mimedefang
              OPTI ONS2=" -U defang  -T -m $MULTIPLEX_SOCK   -p $MIMEDEFANG_SOCK"
               /usr/local/mimedefang/bin/mim  edefang $O  PTIONS2
        fi
        ;;

'stop')
        /usr/bin/pkill  -x -u 8001 '(mimedefang  -mult|mimedefang|mimedefang.pl)'
        remove_sock
        ;;

*)
        echo "Usage: $0 { start | stop }"
        exit 1
        ;;
```

```
esac
exit 0
```

MIMEDefang – sa-mimedefang.cf

```
required_hits               4.0
ok_locales                  all
ok_languages                all
rewrite_subject             1
report_header               1
use_terse_report            1
defang_mime  0
skip_rbl_che cks            1
auto_whitelist_factor       0.5
auto_whitelist_path         /var/spool/spamassassin/auto  -whitelist
auto_whitelist_file_mode    0666
use_dcc                     1
dcc_timeout                 10
dcc_path                    /usr/local/dcc/bin/dccproc
bayes_path                  /var/spool/MIMEDefang/.spamassassin/bayes
```

MIMEDefang – mi medefang-filter

```
# If SpamAssassin found SPAM, append report.  We do it as a separate
# attachment of type text/plain
sub filter_end ($) {
   …
   # Spam checks if SpamAssassin is installed
   if ($Features{"SpamAssassin"}) {
         …
         #  MUA filters can easily be written to trigger on a
         # minimum number of asterisks...

          action_change_header("X  -Spam-Score", "$hits ($score) $names");

          action_add_header( "X  -Spam-Report", "$report" );

         if ($hits  >= $req) {
            md_graphdefang_log('spam', $hits, $RelayAddr);

             action_change_header( "Subject", "***SPAM*** $Subject" );

             # If you find the SA report useful, add it, I guess...
            #action_add_part($e   ntity, "text/plain", " -suggest",
            #              "$report   \n",
             #             "SpamAssassinReport.txt", "inlin   e");
         } else {
            # Delete any existin g X-Spam-Score header?
            #action_dele  te_header("X -Spam-Score");
         }
      }
   }
   ….
}
```

MIMEDefang – SMTP Message Header

```
X-Spam-Score: 8.039 (********) CLICK_BELOW,DCC_CHECK, HTML_30_40,
```

HTML_LINK_CLICK_HERE, HTML_MESSAGE, HTML_WEB_BUGS,
MSGID_FROM_MTA_HEADER, RATWARE_STORM_U RI, RAZO R2_CF_RANGE_51_100,
RAZOR2_CHECK
X-Spam-Report: Spam detection software, Status: "Yes, hits=8
.0 required=4.0 tests=CLICK_BELOW, DCC_CHECK, HTML_30_40,
HTML_LINK_CLICK_HERE, HTML_MESSAGE, HTML_WEB_BUGS,
MSG_ID_FROM_MTA_HEADER, RATWARE_STORM_URI, RA ZOR2_CF_RANGE_51_100,
RAZOR2_CHECK autolearn=no version=2.63"
pts rule name description
---- ---------- ----------
0.1 HTML_LINK_CLICK_HERE BODY: HTML link text says "click here"
0.8 HTML_30_40 BODY: Message is 30% to 40% HTML
0.6 HTML_WEB_BUGS BODY: Image tag intended to identify you
0.0 HTML_MESSAGE BODY: HTML included in message
1.6 RAZOR2_CF_RANGE_51_100 BODY: Razor2 gives confidence between 51 and 100
[cf: 100]
1.5 RATWARE_STORM_URI URI: Bulk email fingerprint (StormPost) found
0.9 RAZOR2_CHECK Listed in Razor2 (http://razor.sf.net/)
1.8 DCC_CHECK Listed in DCC (http://rhyolite.com/a nti-spam/dcc/)
0.0 CLICK_BELOW Asks you to click below
0.8 MSGID_FROM_MTA_HEADER Message -Id was added by a relay
X-Scanned-By: MIMEDefang 2.39

Reference

Carpenter, Jeff, Dougherty, Chad, Havrilla, Jeff , Householder, Allen, King, Brian, Lindner, Marty, Manion, Art, Morda, Damon, Murawski, Rob. "CERT Advisory CA-2004-02 Email-borne Viruses." CERT. 24 January 2004. URL: http://www.cert.org/advisories/CA-2004-02.html (28 March 2004).

CMP. "spam." TechEncyclopedia URL: http://www.techweb.com/encyclopedia/defineterm?term=spam (28 March 2004).

SPAM FILTER REVIEW. "Spam Statistics 2004." URL: http://www.spamfilterreview.com/spam-statistics.html (28 March 2004).

MessageLabs. "Can Spam Act Likely to Increase Record levels of Spam." 30 November 2003. URL: http://www.messagelabs.com/news/pressreleases/detail/default.asp?contentItemId=614&region= (28 March 2004).

Cole, Eric, Fossen, Jason, Northcutt Stephen, and Pomeranz, Hal. SANS Security Essentials with CISSP CBK, Volume 1. SANS PRESS, 2003. 309

Railsback, Kevin. "SpamAssassin takes aim at e-mail." Canning Spam. July 18 2003. URL: http://www.infoworld.com/article/03/07/18/28FEspamassassin_1.html (24 March 2004).

Williamson, Greg. "Spam Filtering: Large Site Principles At A Small Site." SANS GSEC Paper. October 10, 2003. URL: http://www.giac.org/practical/GSEC/Greg_Williamson_GSEC.pdf (28 March 2004).

www.spamassassin.org. "MTA-Level Integration (Site-Wide Use)." URL: http://www.spamassassin.org/where.html. (28 March 2004).

Roaring Penguin. URL: http://www.roaringpenguin.com/products/mimedefang/ (28 March 2004).

Sendmail.org. "Multiple DNS Blacklists." New check_* rulesets/patches for sendmail 8.9. 22 February 2003. URL: http://www.sendmail.org/~ca/email/chk-89n.html (28 March 2004).

Kojm, Tomasz. "Supported platforms." Clam AntiVirus 0.68 User Manual. 2 February 2004. URL: http://www.clamav.net/doc/0.68/html/node8.html (28 March 2004).

Cole, Eric, Fossen, Jason, Northcutt Stephen, and Pomeranz, Hal. SANS Security Essentials with CISSP CBK, Volume 1. SANS PRESS, 2003. 305

Sendmail.org. “SENDMAIL RELEASE NOTES.” 16 September 2003.
URL: http://www.sendmail.org/ftp/RELEASE_NOTES (28 March 2004).

Sendmail.org. “SECURITY HINTS” 24 September 2003.
URL: http://www.technoids.org/milter -README.txt (28 March 2004).

Network Abuse Clearinghouse. “Mail relay testing.”
URL: http://www.abuse.net/relay.html (31 December 2003).

SpamAssassin Wiki. “SpamAssassin.”
URL: http://wiki.apache.org/spamassassin/SpamAssassin (28 March 2004).

Prakash, Vipul Ved. “Vipul's Razor v2 README.” 13 June 2002.
URL: http://razor.sourceforge.net/docs/whatsnew.php (28 March 2004).

Prakash, Vipul Ved. “ Vipul's Razor v2 Installation Instructions.” 17 September 2002.
URL: http://razor.sourceforge.net/docs/doc.php?type=text&name=INSTALL (28 March 2004).

Skoll, David F. “MIMEDefang 2.22 -BETA-4 is available.” 7 October 2002.
URL: http://lists.roaringpenguin.com/pipermail/mimedefang/2002 -October/011508.html (28 March 2004).

Distributed Checksum Clearinghouse. “Functions.” 01 January 2004.
URL: http://www.rhyolite.com/anti -spam/dcc/ (28 March 2004).