E-Mail Services

Overview -

E-Mail is a complex issue for administrators. While the concept of getting a message sent from one user to another seems simple, is really a difficult process with many steps. The widespread usage of e-mail has introduced new problems like e-mail viruses and unsolicited commercial e-mail (aka spam) which administrators must be ready to deal with also.

E-Mail Components -

MUA (Message User Agent): This is the application which the user works with. They include command line tools mail, mailx, pine, elm or mutt as well as graphical tools like Outlook, Outlook Express, Eudora, Evolution, KMail, Netscape Communicator and others.

MTA (Message Transfer Agent): This software normally runs in the background and is used to actually transfer messages from one server to another. Generally, the MUA will send outgoing messages to the MTA, which then turns around and talks to the MTA at the receiver's location. The most commonly used MTA is the *sendmail* program. Others include *postfix*, *exim*, and *Microsoft Exchange Server*.

Delivery Agent: When a new message is received by an MTA, it needs to be sent to the user's inbox so they can read the message later. The delivery agent is responsible for accepting the message from the MTA and routing it to the user's mailbox. Mailboxes can be simple text files in a special directory, or perhaps stored in a database. In either case, the delivery agent is the tool that actually updates the mailbox. The most commonly used delivery agents are *procmail*, *smrsh* and *mail.local*.

Access Agents: When the user wishes to read messages from their mailbox, an access agent is normally used to download the messages from the server to their local computer system. The MUA normally must use an access agent for this. Commonly used access agents are *POP3* (Post Office Protocol 3) and *IMAP* (Internet Message Access Protocol).

Message Breakdown -

Envelope: This is used by the MTA to determine where a message is coming from and going to. You will not see the envelope in your

messages, even if you examine the details of the message because it is only used by the MTA.

Headers: The headers are used to store a wide variety of information about the message including the sender (From) and destination (To) lines, the date and time of the message, and a list of all the computers that handled the message. In addition, extra lines can be added to the header by various agents. All of these extra lines should begin with "X-" and will be passed unchanged by the MTA and delivery agents to the user. In general, your MUA will hide most, if not all, of the header lines from the user.

Body: This is the actual text of the message itself. In the past, e-mail was restricted to just plain text, but new mechanisms like MIME (Multipurpose Internet Mail Extensions) allows users to attach documents (and things like viruses) to messages. Generally, MIME attachments are encoded using Base64 encoding, although other encoding schemes like UUencode (Unix-to-Unix encode), BinHex (Binary to Hex) and TNEF (Transport Neutral Encapsulation Format) are also sometimes used.

Sample Message:

```
Return-Path: <kde-devel-bounces-+blueboy=bamafolks.com@mail.kde.org>
Received: from localhost (IDENT:1000@localhost [127.0.0.1])
        by zev.home.bamafolks.com (8.12.4/8.12.4) with ESMTP id h3HF5mie010876
        for <randy@localhost>; Thu, 17 Apr 2003 10:05:48 -0500
Received: from wolf [216.207.245.91]
        by localhost with IMAP (fetchmail-5.9.11)
        for randy@localhost (single-drop); Thu, 17 Apr 2003 10:05:48 -0500 (CDT)
Received: from ktown.kde.org (kde.informatik.uni-kl.de [131.246.103.200])
        by wolf.BamaFolks.com (8.12.9/8.12.9) with SMTP id h3HF0thI011246
        for <blueboy@bamafolks.com>; Thu, 17 Apr 2003 10:00:56 -0500
Received: (qmail 25853 invoked from network); 17 Apr 2003 15:00:55 -0000
Received: from localhost (HELO ktown.kde.org) (127.0.0.1)
  by localhost with SMTP; 17 Apr 2003 15:00:55 -0000
Received: (qmail 25764 invoked by uid 1055); 17 Apr 2003 15:00:45 -0000
Delivered-To: kde.org-kde-devel@kde.org
Received: (qmail 25762 invoked from network); 17 Apr 2003 15:00:44 -0000
Received: from odin.sinectis.com.ar (HELO mail.sinectis.com.ar)
        (root@216.244.192.158)
        by kde.informatik.uni-kl.de with SMTP; 17 Apr 2003 15:00:42 -0000
Received: from carmen.fsl.org.ar (ADSL-200-59-84-208.cordoba.sinectis.com.ar
        [200.59.84.208])
        by mail.sinectis.com.ar  with ESMTP id h3HF0aV27048
        for <kde-devel@kde.org>; Thu, 17 Apr 2003 12:00:36 -0300
X-Originally_To: <kde-devel@kde.org>
Received: from pcmam.fsl.org.ar
        ([192.168.1.8] helo=tempest ident=mail)
        by carmen.fsl.org.ar with esmtp (Exim 3.35 #1 (Debian))
        id 196AsM-0004rF-01
        for <kde-devel@kde.org>; Thu, 17 Apr 2003 12:00:34 -0300
Received: from mdione by tempest with local (masqmail 0.2.19) id
 196AsH-0VP-00 for <kde-devel@kde.org>; Thu, 17 Apr 2003 12:00:29 -0300
Date: Thu, 17 Apr 2003 12:00:29 -0300
From: Marcos Dione <mdione@grulic.org.ar>
To: KDE Development ML <kde-devel@kde.org>
Message-ID: <20030417150029.GA1914@charlie>
```

Mime-Version: 1.0
Content-Type: text/plain;
  charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.5.4i
Subject: kde head, xfree 4.3 and keyb
X-BeenThere: kde-devel@mail.kde.org
X-Mailman-Version: 2.1.1
Precedence: list
Reply-To: kde-devel@kde.org
List-Id: For discussion of all KDE-related development issues
	<kde-devel.mail.kde.org>
List-Unsubscribe: <http://mail.kde.org/mailman/listinfo/kde-devel>,
	<mailto:kde-devel-request@mail.kde.org?subject=unsubscribe>
List-Archive: <http://mail.kde.org/mailman/private/kde-devel>
List-Post: <mailto:kde-devel@mail.kde.org>
List-Help: <mailto:kde-devel-request@mail.kde.org?subject=help>
List-Subscribe: <http://mail.kde.org/mailman/listinfo/kde-devel>,
	<mailto:kde-devel-request@mail.kde.org?subject=subscribe>
Sender: kde-devel-bounces-+blueboy=bamafolks.com@mail.kde.org
Errors-To: kde-devel-bounces-+blueboy=bamafolks.com@mail.kde.org
X-Virus-Scanned: by AMaViS 0.3.12
X-Spam-Status: No, hits=-16.7 required=3.5
	tests=AWL,USER_AGENT_MUTT
	autolearn=ham version=2.52
X-Spam-Level:
X-Spam-Checker-Version: SpamAssassin 2.52 (1.174.2.8-2003-03-24-exp)
Status: R
X-Status: N
X-KMail-EncryptionState:
X-KMail-SignatureState:

   I just uinstalled x-4.3 and now the kxkb doesn't work, the win keys (which I
set up for anything related to kwin) don't work, etc. I realize that xkb has
changed (once again), so is there any support for this change?

--
La gelatina hecha con vodka pega mas que 'La Gotita'
	--oyente anonimo del programa de radio "Dos tipos audaces"

>> Visit http://mail.kde.org/mailman/listinfo/kde-devel#unsub to unsubscribe <<

E-Mail Problems

Viruses: In general, Unix and Linux users don't need to worry too much about viruses, but if you have any Windows users, then this can be a real problem for them. And remember, even if the viruses does not infect Unix and Linux systems, they can still steal bandwidth and network resources. I personally use the Amavis virus filter in conjunction with McAfee's Unix Virus Scanner. Amavis is available from http://www.amavis.org and the McAfee Unix Virus Scanner is available at http://www.nai.com/naicommon/buy-try/try/products-evals.asp or http://www.mcafeeb2b.com/products/virusscan-cl/default-virusscan-cl.asp. A recent newcomer that you may want to also investigate is the ClamAV product, which is the first true open-source, free anti-virus system. You can find ClamAV at http://clamav.sourceforge.net.

SPAM: The longer you have the same e-mail address, the more likely that you will begin receiving unwanted junk mail. Researchers claim that spam now accounts for almost 50% of all the e-mail on the Internet. There have been many attempts to control spam, but some methods work better than others. I highly recommend (and use) the SpamAssassin filter available from http://www.spamassassin.org. Other solutions include using one of the many RBL (Remote Blacklists), SPF (Sender Policy Framework), C/R (challenge/response) systems or the Yahoo-initiated DomainKeys system.

Open Relays: One of the ways in which junk e-mail is spread is through the use of unprotected open relay servers. An open relay is a incorrectly configured MTA that will gladly accept messages from any user at any computer and transmit messages to another other user at any other computer. Why should I allow users in France to ask my mail server to send messages to another user located in Argentina? You should make sure your server only processes the messages coming from or going to your users.

Other E-mail Features

Aliases: An alias allows you redirect messages sent to a e-mail address to a different, or perhaps multiple, mailboxes. For example, I have an alias on my system that sends all messages sent to rlp@bamafolks.com to the mailbox for the user named "randy". Therefore the addresses rlp@bamafolks.com and randy@bamafolks.com are equivelant. I have another alias called home@bamafolks.com that sends messages to both my account and my wife's account. Edit the file **/etc/mail/aliases** to add or delete aliased names. After editing, you must run the **newaliases** command to inform sendmail of the changes.

Virtual Hosts: This feature allows you to setup a special kind of alias that forwards all messages sent to a specific domain name to a single mailbox. For example, until recently I was hosting the domain name "BamaBeef.com" and wanted all messages sent to that domain to be delivered to a single mailbox. Sendmail can do this via the **virtuser** option of its configuration file.

Access Control: An access control list allows the administrator to specifically grant or deny specific computers access to the MTA. This can be used as a primitive way to reduce spam, but is not very effective since the list of computers that send spam is changing on an almost daily basis. It can be used to grant access to remote computers if needed, but only works correctly if the computers have a static (fixed) IP address. For users that travel and use laptops, a

better method is to require authentication.

Authentication: Generally, users must supply their user account and password in order to read or download messages from the server. With the rise in spam, it is also a very good idea to force users to login to send messages. This feature is not generally turned on by default, so in order to add this, you must reinstall sendmail yourself.

Vacation: It is nice if users can setup an auto responder that notifies people that you are on vacation or out of the office and unable to reply to their message. You will find the **vacation** program useful for setting this up.

Web-Based Mail: As more and more of your users travel, you may find that using POP3 and IMAP for them becomes awkward. This is especially true if users do not have their own laptops that they carry with them. In this case, a web interface to the user's mailboxes is very convenient. I personally use the SquirrelMail system which is available from http://www.squirrelmail.org. It is easy to install and has many extras such as plugin support that can be used to add new features to the system.

Configuration Files and Directories

Most of sendmail's configuration files are stored in the **/etc/mail** directory. The installation process copies several files here that will generally work to send and receive e-mail for the local computer. The two most important files are named **sendmail.cf** and **submit.cf**. The first contains options and settings is used by the MTA to handle incoming messages received from remote servers, while the second controls options used when transmitting messages out of your server to remote sites.

These files are very cryptic and hard to edit by hand. Most Linux distributions will also copy additional versions of these files that may have all the features you desire enabled. In that case, you can just copy the sample file and restart sendmail. Finally, you can use a set of programming tools to generate new versions of this files using macros. It is much easier to edit/create a macro file than it is to edit/create a .cf file. Slackware installs these tools under the /usr/share/sendmail directory. Of course, you could also download your own copy of sendmail from http://www.sendmail.org which includes all the source code, not just the configuration file tools.

Steps to create a new configuration file -

- 1: As root, cd to the /usr/share/sendmail/cf/cf directory.
- 2: After reviewing the existing samples, copy one of the ".mc" files and rename to "sendmail.mc".
- 3: Customize the settings as needed. (See http://www.sendmail.org for details.)
- 4: Issue this command: sh Build sendmail.cf
- 5: Install using this command: sh Build install-cf

A Typical Macro File -

```
divert(0)dnl
VERSIONID(`$Id: generic-linux.mc,v 8.1 1999/09/24 22:48:05 gshapiro Exp $')
OSTYPE(linux)dnl
define(confMAX_MESSAGE_SIZE,10000000)dnl
dnl Require authentication via SASL
define(`confAUTH_OPTIONS', `A')dnl
define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
TRUST_AUTH_MECH(`LOGIN PLAIN')dnl
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
FEATURE(nouucp,`reject')dnl
FEATURE(always_add_domain)dnl
FEATURE(access_db)dnl
FEATURE(virtusertable)dnl
FEATURE(local_procmail)dnl
FEATURE(blacklist_recipients)dnl
dnl
dnl Change Mlocal to use AMaViS-Perl
define(`LOCAL_MAILER_PATH', `/usr/sbin/amavis')dnl
define(`LOCAL_MAILER_ARGS', CONCAT(`amavis $f $u /usr/bin/',
LOCAL_MAILER_ARGS))dnl
dnl please set the path to your procmail accordingly!
dnl the following works only with sendmail 8.10.x or above
MODIFY_MAILER_FLAGS(`LOCAL', `-m-f-r')dnl
MAILER(procmail)dnl
MAILER(smtp)dnl
```

Notes -

define(confMAX_MESSAGE_SIZE,10000000)dnl

This line restricts sendmail to only accept messages smaller than 10,000,000 bytes in size.

define(`confAUTH_OPTIONS', `A')dnl
define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
TRUST_AUTH_MECH(`LOGIN PLAIN')dnl

These lines force users to authenticate (using a PLAIN LOGIN) before attempting to send relay e-mail to other servers. These options required other changes and will not work unless you build sendmail yourself. I use this to allow laptop users to send e-mail from any IP

address, provided they login prior to attempting to send the e-mail.

FEATURE(redirect)dnl

When e-mail is sent to *user.REDIRECT*, sends a message back to the sender that the person they are trying to contact has moved and to try the new e-mail address. The best way to do this is by adding an alias so incoming messages are redirected to a *user.REDIRECT@new-domain*.

FEATURE(use_cw_file)dnl

Tells sendmail that the file /etc/mail/local-host-names contains a list of valid names for which this computer will accept incoming mail. This is needed if you are handling e-mail for several different domain names on a single server. I support both **bamafolks.com** and **pearsonconsulting.com** this way for example.

FEATURE(use_ct_file)dnl

This tells sendmail to read the file /etc/mail/trusted-users for a list of users that can send e-mail, but use a different From: line without generating a warning. Since I have several scripts that do this, I have added one or two users to this file to stop warning messages from appearing in my logs.

FEATURE(uucp,`reject')dnl

This prevents sendmail from relaying uucp style messages that contain an ! (exclamation mark) in the e-mail address. This is an old way of handling e-mail through dial-up lines that is mostly used by folks that send spam today.

FEATURE(always_add_domain)dnl

When sending e-mail from one user to another on the same server, generally only the user names are used in the To: and From: lines. This option adds the fully qualified domain name even if the message is just being delivered to another mailbox on the same server.

FEATURE(access_db)dnl

This option tells sendmail to consult the /etc/mail/access.db file for a list of IP address/computer names that are specifically granted or denied access to the MTA. The access.db file must be created from a plain text file using a command similar to the following:

makemap hash /etc/mail/access < /etc/mail/access

HINT: Create a script.

FEATURE(virtusertable)dnl

This option tells sendmail to read the /etc/mail/virtusertable.db file for a list of special domain redirections. This feature allow me to send all messages sent to *.BamaBeef.com to a single mailbox. The virtusertable.db file must be created from a plain text file using a command similar to the following:

makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable

HINT: Create a script.

FEATURE(local_procmail)dnl

Tells sendmail to use /usr/bin/procmail as the delivery agent to move received messages into the user's mailbox. This is highly recommended and the default for most versions of Linux.

FEATURE(blacklist_recipients)dnl

Allows you to add specific e-mail addresses to the /etc/mail/access file to blacklist those accounts. Works in conjunction with the access_db feature.

define(`LOCAL_MAILER_PATH',`/usr/sbin/amavis')dnl
define(`LOCAL_MAILER_ARGS', CONCAT(`amavis $f $u /usr/bin/', LOCAL_MAILER_ARGS))dnl
MODIFY_MAILER_FLAGS(`LOCAL','-m-f-r')dnl

These lines override the usage of procmail with the program /usr/sbin/amavis as the locally defined delivery agent. In this case, the Amavis program will scan the message for viruses and if no virus is found, forwards the message to procmail for final delivery. If a virus is found, Amavis will normally send a warning message to both the original sender and an alias account named virusalert. The Amavis program must already be installed in order to use these lines.

MAILER(procmail)dnl
MAILER(smtp)dnl

These lines allow sendmail to use procmail or standard SMTP

connections to deliver mail. The first is used for local deliveries while the second is used for deliver to remote computers.

POP3 and IMAP

Slackware has already installed both a POP3 and IMAP services for you. If you did not install all software, then the installation CD has the packages.

Even though they are installed, they are not active by default. In order to turn them on, follow these steps:

- 1: Edit the **/etc/inetd.conf** file
- 2: Locate the "pop3" and "imap2" lines and remove the comment (#).
- 3: Save the changes.
- 4: Tell the inetd program to reread its configuration using this command: **killall -HUP inetd**

Amavis Installation

Amavis is a very nice tool that can scan email messages for viruses before they are delivered to the user's mailbox. You can download it from http://www.amavis.org. That website contains full instructions, but here is the quick and dirty instructions. The reason so many extra software packages are needed is because viruses can be hidden inside many different types of files. All these tools are needed so Amavis can uncompress and decode every type of file and scan them for viruses.

- 1: Install (or verify) the following software is installed first -
  - An anti-virus program (I personally use McAfee Virus Scan for Unix)
  - Perl 5.6 or higher (usually already installed)
  - An up to date **file** command (Slackware has this)
  - ARC version 5.21e or higher (ftp://ftp.uu.net/pub/archiving/)
  - bunzip2 (Slackware has this)
  - LHA 1.14j or higher (Slackware has this)
  - UNARJ (Slackware has this)
  - UNCOMPRESS (Slackware has this)
  - UNRAR (ftp://sunsite.unc.edu/pub/Linux/utils/compress)
  - UNFREEZE (http://metalab.unc.edu/pub/Linux/utils/compress)
  - TNEF (http://world.std.com/~damned/software.html)

- ZOO (Slackware has this)

- 2: Install (or verify) the following Perl modules are installed -

  - IO-stringy
  - Unix-Syslog
  - MailTools
  - MIME-Base64
  - MIME-tools version 5.313 or higher
  - Convert-UUlib version 0.111 or 0.201 or higher
  - Convert-TNEF 0.06 or higher
  - Compress-Zlib 1.14 or higher
  - Archive-Tar
  - Archive-Zip
  - libnet

  - You can install/update these Perl modules as follows:

    - **perl -MCPAN -e shell**
    - **install Unix::Syslog**
    - **install Convert::UUlib**
    - **install Convert::TNEF**
    - **install Compress::Zlib**
    - **install Archive::Tar**
    - **install Archive::Zip**
    - **install G/GB/GBARR/MailTools-1.15.tar.gz**
    - **install MIME::Tools**
    - **install libnet**
    - **quit**

- 3: Extract, build and install Amavis itself like this:

  - **tar xzvf amavis-*version*.tgz**
  - **cd amavis-*version***
  - **./configure**
  - **make**
  - **make check**
  - **make install**

- 4: Configure sendmail to use Amavis as follows:

  - Add the following lines to your sendmail.mc file

define(`LOCAL_MAILER_PATH',`/usr/sbin/amavis')dnl

define(`LOCAL_MAILER_ARGS', CONCAT(`amavis $f $u /usr/bin/', LOCAL_MAILER_ARGS))dnl
MODIFY_MAILER_FLAGS(`LOCAL','-m-f-r')dnl

- Rebuild your sendmail.cf file by running: **sh Build sendmail.cf**
- Install your new sendmail.cf file by running: **sh Build install-cf**

Now sendmail should filter all messages through Amavis when they are received.

SpamAssassin Installation

Luckily installing the powerful SpamAssassin software is much easier to install than Amavis. It too was written in Perl, so that must be installed first of course. Here are the quick and easy instructions on installing and configuring the program:

- 1: Download the latest version from http://www.spamassassin.org
- 2: Extract: **tar xzvf Mail-SpamAssassin-*version*.tgz**
- 3: Change to the SpamAssassin directory: **cd Mail-SpamAssassin-*version***
- 4: Prepare the package: **perl Makefile.PL**
- 5: Build the software: **make**
- 6: Install the software: **make install**
- 7: Add the following to your **/etc/rc.d/rc.local** file:

  - if [-x /usr/bin/spamd]; then
  - echo "Starting SpamAssassin daemon..."
  - /usr/bin/spamd -d -c -a
  - fi

- 8: Create/edit **/etc/procmailrc** to include these lines:

  - DROPPRIVS=yes
  -
  - :0fw
  - * < 512000
  - | spamc

By adding the rule above to the **/etc/procmailrc** file, you are asking the procmail program to send all messages less than 512,000 bytes through the "spamc" program. That program uses the "spamd" service to check messages to determine if they are spam-like. If so, the message will be modifed and marked as spam. In either event,

the filtered message is then send to the user's mailbox by the procmail program. Additional rules can be added to each user's personal procmail file also.

If you do not wish to reboot you must start the spamd program manually with the command "**/usr/bin/spamd -d -c -a**".

Additional options for SpamAssassin can be controlled by editing its configuration file located at **/etc/mail/spamassassin/local.cf**.

SquirrelMail Installation

This is a web-based interface that allows remote users to login to a web page and check their e-mail. It is not too hard to install by itself, although some of the available plugins can be tricky to get working correctly. Here are the quick steps:

- 1: Download the software from http://www.squirrelmail.org
- 2: Change to Apache's HTML directory: **cd /var/www/htdocs**
- 3. Extract the program: **tar xzvf <*PATH*>/squirrelmail-*version*.tgz**
- 4. Make a link to the squirrelmail folder: **ln -s squirrelmail-*version* mail**
  - NOTE: This is just for convenience. It's probably easier for users to remember http://www.bamafolks.com/mail than it is to remember http://www.bamafolks.com/squirrelmail-1.40rc2.
- 5: Change into the new directory: **cd mail**
- 6: Finalize the settings by running: **./configure**
- 7: Change ownership of all files/directories to that of the Apache user account: **chown -R nobody.nobody ***

Provided Apache is running and has support for PHP, SquirrelMail should now be active and ready to use. Visit the main website for additional plugins you may wish to download and install.

Adding Login Support to Sendmail

Remember, sendmail does not normally require users to login before sending messages to other users. This works fine, except for one thing. Most installations of sendmail also don't allow users to send out messages unless they are actually logged into the server itself. If you are attempting to allow users to connect from home, or other remote

sites, you must all them to relay message through your server.

There are several ways to accomplish this, but most of them also open the door up for people to use your server to relay spam messages along with real messages from your users. For this reason, I highly recommend adding login support to sendmail. This way, you can configure sendmail to allow relaying of messages only from people who have provided a user name and password.

Sendmail itself cannot check user's passwords, but it does have support for using a 3rd party library to do so. The 3rd party library of choice is called Cyrus SASL (Simple Authentication and Security Layer). It can be downloaded from ftp://ftp.andrew.cmu.edu/pub/cyrus-mail. I have tested and found version 1.5.28 to work well.

Here are the instructions:

- 1: Download Cyrus SASL.
- 2: Extract the software: **tar xzvf <*PATH*>/cyrus-sasl-1.5.28.tar.gz**
- 3: Change to the new directory: **cd cyrus-sasl-1.5.28**
- 4: Configure the program: **./configure --prefix=/usr --enable-login**
- 5: Build the software: **make**
- 6: Install the software: **make install**

Next, you must rebuild sendmail (with extra options to turn on login support):

- 1: Download sendmail from http://www.sendmail.org
- 2: Extract the software: **tar xzvf <*PATH*>/sendmail-*version*.tar.gz**
- 3: Change to the new directory: **cd sendmail-*version***
- 4: Add the following lines to the file named: **devtools/Site/site.config.m4**:
  - APPENDDEF(`conf_sendmail_ENVDEF',`-DSASL')
  - APPENDDEF(`conf_sendmail_LIBS',`-lsasl')
- 4: Build sendmail: **sh Build**
- 5: Create a new sendmail.cf file as follows:
  - Change to the cf/cf directory: **cd cf/cf**
  - Copy a sample .mc file: **cp generic-linux.mc sendmail.mc**

- Edit sendmail.mc and add the following:
  - define(`confAUTH_OPTIONS', `A')dnl
  - define(`confAUTH_MECHANISMS', `LOGIN PLAIN')dnl
  - TRUST_AUTH_MECH(`LOGIN PLAIN')dnl
- Build a new sendmail.cf file: **sh Build sendmail.cf**
- Install the new sendmail.cf file: **sh Build install-cf**
- Change back to the top sendmail directory: **cd ../..**

- 6: Install the new sendmail program: **sh Build install**
- 7: Configure the authentication for sendmail as follows:
  - Change to the SASL library directory: **cd /usr/lib/sasl**
  - Create a new Sendmail.conf file: **vi Sendmail.cf**
    - NOTE: This file must have a capital 'S'.
  - Add the following line:
    - pwcheck_method: shadow

Finally, restart sendmail (**killall -HUP sendmail** or **/etc/rc.d/rc.sendmail restart**).

Users will now need to authenticate before they can send out message through your server to remote sites on the Internet.

Conclusion

There are many other features, options and tools that can be used to control and enhance mail handling. Visit the web sites of the various tools mentioned for additional information. Hopefully these instructions will be enough to get you started safely however.