Sendmail 8.13.3
Secure Mailing Solution

HP Part Number: 5992-3190
Published: October 2007
Edition: 1.0

Table of Contents

List of Tables

List of Examples

Executive Summary

This white paper discusses the `STARTTLS` and `AUTH` features that are supported in Sendmail 8.13.3. It also describes how to configure these features on HP-UX systems, to provide an effective secure mailing solution. In addition, this white paper includes selected usage models and examples, and discusses the benefits of using these Sendmail 8.13.3 features on HP-UX systems. This whitepaper also describes how to create a Certificate Authority (CA) and to prepare or sign certificates for Sendmail 8.13.3 servers.

Intended Audience

This white paper is intended for HP customers who are using or planning to use Sendmail 8.13.3 to ensure mail security. This white paper is also intended for system administrators, HP support personnel for Sendmail 8.13.3, HP field engineers, and consultants who advise customers on security solutions. Readers of this document must be familiar with using Sendmail 8.13.3.

Introduction

Sendmail 8.13.3 is the latest version of Mail Transfer Agent (MTA) available on the HP-UX operating system. It offers enhanced security, performance, and anti-spamming capabilities.

Following are the salient security features of Sendmail 8.13.3:

- Transport Layer Security (TLS)/Secure Sockets Layer (SSL) support
- CyrusSASL support

This white paper describes how to configure these security features in Sendmail 8.13.3.

This white paper addresses the following topics:

TLS/SSL Support

Sendmail 8.13.3 uses the Transport Layer Security (TLS) and the Secure Socket Layer (SSL) to encrypt not only the user name and password, but the entire mail transmission. To signal the beginning of an encrypted TLS conversation, Sendmail 8.13.3 uses the `STARTTLS` command within an *SMTP* conversation.

`STARTTLS` feature is an extension of the SMTP service that enables an SMTP server and client to use the transport layer security in providing private and authenticated

communication over the Internet. It enables the SMTP agents to protect some or all of their communications from eavesdroppers and attackers.

The STARTTLS feature offers the following benefits:

- Verifies the identity of the client and server in a mail transmission.
- Authenticates a user for relaying through a mail server.
- Encrypts mail transmissions.
- Encrypts transmissions between two mail servers over the Internet.

Sendmail 8.13.3 relies on the OpenSSL implementation for cryptographic algorithms. The cryptographic algorithms used for encrypting messages are completely transparent to Sendmail 8.13.3.

OpenSSL is an open source implementation of the SSL and TLS protocols. The version of OpenSSL available on the HP-UX 11i v1 and HP-UX 11i v2 operating systems is OpenSSL A.00.09.07l. The version of OpenSSL available on the HP-UX 11i v3 operating system is OpenSSL A.00.09.08d. Both OpenSSL A.00.09.07l and A.00.09.08d include a general-purpose cryptography library and implementation of the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols.

This section addresses the following topics:

- "Cryptography Algorithm" (page 10)
- "Certificates and Authorities" (page 11)

Cryptography Algorithm

The TLS subsystem uses the following components to provide services, such as integrity checking, authentication, and confidentiality:

- **Private key algorithms**, or symmetrical cryptography. This component uses a shared secret and the key, for both encryption and decryption of a message. Input data is mathematically processed using the algorithm and the key, to produce the ciphertext output that must be decrypted for the recipient. Commonly used private key algorithms include *DES*, `Blowfish`, *AES*, and *IDEA*.
- **Public key algorithms**. These algorithms use two mathematically related keys to separate the processes of encryption and decryption. By using functions that are easy to perform in one direction but difficult to perform in the opposite direction, the two keys provide a high level of security if large numbers are used. Commonly used public key algorithms include *RSA*, `El Gamal`, and `Diffie-Hellman`.

  While establishing a TLS session, you can use public key cryptography to exchange a session key that is used in a private key algorithm. You can also use these public keys to authenticate the server and, if required, the client, and to provide session-level encryption and confidentiality for the entire session.
- **Hash algorithms**. These algorithms are a set of one-way functions that accept a variable length input, and, after mathematical processing, produce a fixed length output. The transformations of the data produce a fingerprint of the input. The

minor changes to the input appear as large changes in the output. Popular hash algorithms include SHA-1, *MD5*, and *RIPEMD*.

Hash algorithms are used for integrity checking; that is, to ensure that data is not tampered during transmission.

Certificates and Authorities

A certificate is a collection of information that uniquely identifies a client or a server. It includes descriptive fields, such as the name of an organization and its location, as well as cryptographic information, such as keys and signatures.

The private key of an asymmetrical key pair can be used to sign the content that, when decrypted using the public key, establishes the signature. This signature can be used to offer proof of identity. The public key infrastructures (PKI) use a hierarchy of trustworthiness for the validation of identities, in addition to signing certificates and keys. This is in contrast to the web of trust used in pretty good protection (PGP), which has no central authority.

The central authority in a PKI issues a Certificate Authority (CA), a definitive certificate that contains the information and the public key of the server. This CA can be used to sign other certificates, by signing the public key of a requesting body, such as your server, with the private key. The trust in identity is transitive, because the CA is recognized by all the involved parties as authoritative: *"I trust the CA, and the CA says that it is you, so it must be true."*

Certificates can be revoked because of expiration or compromise in security. To do this, the issuing body provides a certificate revocation list (CRL) that identifies the certificates to be invalidated. This is also trusted because strong proof is provided through the trust mechanisms.

Certificates are available in different formats, though PEM is the most widely used format. The PEM encoding is an ASCII text representation of the binary data in the ASN.1 format. The X.509 standard defines the distinguished name (DN) format used in these certificates.

A certificate contains the following information that accompanies the cryptographic keys:

- Common name (CN) being certified
- Organization (O) associated
- Organizational unit (OU), such as a department within an organization
- City or location (L) where an organization is located
- State or province (SP) where the city is located
- Country (C) in the ISO (International Organization for Standardization) format (such as U.S.)

The DN is a combination of the different certificate information. The PEM-encoded certificate contains this information along with the DN of the issuer, the validity period

of the certificate, various administration information, such as a serial number of the certificate, and any other required information, such as Netscape-specific tags. These certificates are used to establish the identity and trustworthiness of the presenter, such as a server or a client. These certificates are also used to authenticate the connecting party and to take appropriate action, such as allowing a connection to proceed, and mail relaying, or entry into a network. You can either use the commercial TLS/SSL certificates (certs) to verify the identity of the Sendmail 8.13.3 server, or create your own certificates for the Sendmail 8.13.3 servers.

RFC 2487 (*SMTP Service Extension for Secure SMTP over TLS*) describes how to use the method for using TLS/SSL as a popular mechanism for enhancing TCP communications with privacy and authentication. In most cases, the communication passes through one or more routers that are not controlled or trusted by either entity. Such untrusted routers can be a serious security threat. If SMTP agents are able to authenticate each others' identities, the server can allow only communications from other SMTP agents it is aware of or it can treat messages received from the known agents differently, compared to messages received from unknown agents.

Cyrus SASL Support

The Simple Authentication and Security Layer (SASL), is a generic mechanism that enables application protocols, such as SMTP and *IMAP*, to accomplish authentication. Sendmail 8.13.3 uses Cyrus SASL, a product implementation of the SASL protocol, to accomplish authentication.

Applications such as Sendmail use the SASL framework to accomplish the SASL protocol exchange. The specific SASL mechanisms govern the exact protocol exchange. If a framework contains `n` protocols and `m` different ways of authentication, SASL attempts to make the framework simple so that you need to write only `n` plus `m` different specifications, instead of `n` times `m` different specifications. With the Cyrus SASL library, you need to write the authentication mechanisms only once, because they work with all the servers that use the authentication mechanisms.

The way SASL works is governed by the mechanism that the client and server select to use and the exact implementation of that mechanism.

A client application interacts with the SASL library (also known as the SASL glue layer) as follows:

1. A client application makes a few calls to initialize the SASL library.
2. Each time the client application makes a new connection, it creates a new context that is stored for the lifetime of that connection.
3. The client application requests the server for the list of supported mechanisms.
4. The client application feeds this list to the SASL library.
5. The client application starts the authentication with the mechanism selected by the SASL library.
6. The server returns some bytes, which are provided to the SASL library.

7. The SASL library returns some bytes to the client application.
8. The client application transmits these bytes over the network.
9. The client application repeats steps 7 – 9 until the server informs the application that the authentication is successful.

An application in the server interacts with the SASL library as follows:

1. A server makes a few calls to initialize the SASL library.
2. When the server establishes a new connection, the server makes a new context for that connection immediately.
3. The client requests a list of mechanisms the server supports and specifies the mechanism it wants to use. The client also requests to start the authentication process after finalizing on the authentication mechanism.
4. The server negotiates this authentication and retains the authentication information for subsequent encoding and decoding operations.

RFC 2554 (*SMTP Service Extension for Authentication*) specifies that the `AUTH` command indicates an authentication mechanism to the server. If the server supports the requested authentication mechanism, it performs an authentication protocol exchange to authenticate and identify the user. Optionally, it also negotiates a security layer for subsequent protocol interactions. If the requested authentication mechanism is not supported, the server rejects the `AUTH` command with a `504` reply.

Versions of Sendmail starting with 8.10 support the `SMTP AUTH` command, as defined in RFC 2554.

Configuring Sendmail 8.13.3 with TLS and SSL

This section describes how to configure SMTP over TLS, as defined in RFC 2487. It also describes how to verify the TLS/SSL configuration.

This section addresses the following topics:

Prerequisites

Following are the prerequisites for configuring the TLS/SSL security feature:

- The KRNG11i strong random number generator

NOTE: The KRNG11i strong random number generator is required only for the HP-UX 11i v1 operating system. For the HP-UX 11i v2 and HP-UX 11i v3 operating systems, the random number generator is available as part of the core HP-UX operating system.

- The OpenSSL software

NOTE: You must install the latest version of the OpenSSL software from http://www.software.hp.com lists to avoid errors while running the `CA.pl` script. Table 1 (page 14) lists the version of OpenSSL that you must install for different HP-UX operating systems.

Table 1 OpenSSL Versions

Operating System Name	OpenSSL Version
HP-UX 11i v1	A.00.09.07l
HP-UX 11i v2	A.00.09.07l.001
HP-UX 11i v3	A.00.09.08d.001

- The latest version of the Sendmail 8.13.3 web upgrade.

NOTE: For the HP-UX 11i v3 operating system, Sendmail 8.13.3 is available as part of the core HP-UX operating system. For the HP-UX 11i v1 and HP-UX 11i v2 operating systems, Sendmail 8.13.3 is available as a web upgrade at:

http://www.software.hp.com

Generating Certificates and Keys

The OpenSSL script, `/opt/openssl/misc/CA.pl`, can be used to generate the certificates and keys. By default, the certificates are encrypted using the DES encryption. You must log in as a superuser and modify the `CA.pl` script to prevent the certificates from being DES encrypted.

Follow this procedure to generate certificates and keys:

1. To change the directory to `/opt/openssl/misc`, enter the following command:

   ```
   cd /opt/openssl/misc
   ```

2. To copy the `CA.pl` script to the `CA.pl.ORIGINAL` script, enter the following command:

   ```
   cp CA.pl CA.pl.ORIGINAL
   ```

3. Replace the entries marked with numbers in the following `CA.pl` script:

```
exit 0;
} elsif (/^-newcert$/) {
  # create a certificate

system ("$REQ -new -x509 -keyout newkey.pem -out newcert.pem $DAYS"
);1

$RET=$?;
 print "Certificate is in newcert.pem, private key is in newkey.pem\n"
  } elsif (/^-newreq$/) {
system ("$REQ -new -keyout newkey.pem -out newreq.pem $DAYS");2

$RET=$?;
print "Request is in newreq.pem, private key is in newkey.pem\n";
 } elsif (/^-newreq-nodes$/)
```

1 Replace this line with the following:

```
system ("$REQ -new -nodes -x509 -keyout newkey.pem -out newcert.pem $DAYS");
```

2 Replace this line with the following:

```
system ("$REQ -new -nodes -keyout newkey.pem -out newreq.pem
$DAYS");
```

The only change is the addition of the `-nodes` option while generating certificates. If you do not include this option, Sendmail 8.13.3 cannot load the encrypted key during startup. As a result, it logs the following error message in the `/var/adm/syslog/mail.log` file:

```
"May 4 11:55:20 XXXXX sm-mta[23544]: STARTTLS=server, error:
SSL_CTX_use_PrivateKey_file(/etc/mail/certs/servername-key.pem) failed"
```

NOTE: You must modify the first line in the `CA.pl` script to the location of the perl interpreter on your system. Otherwise, the following error message is logged in the `/var/adm/syslog/syslog.log` file:

```
interpreter "/opt/perl/bin/perl" not found
```

---

4. Follow this procedure to create your own CA, and to create certificates and keys for your Sendmail 8.13.3 server:

   a. To create a CA, enter the following command:

   ```
   $ ./CA.pl –newca
   ```

   The following message displays:

   ```
   CA certificate filename (or enter to create)
   ```

   Enter the file name or press **Enter**.

   The following message displays:

   ```
   Making CA certificate...
   Generating a 1024 bit RSA private key
   ..................++++++...++++++
   ```

```
writing new private key to
'./demoCA/private/cakey.pem'
Enter PEM pass phrase:
```

Enter the passphrase.

NOTE: Select a unique passphrase so that no one can abuse your CA and sign a certificate.

The following message displays:

```
Verifying - Enter PEM pass phrase:
```

Enter the passphrase again.

The following message displays:

```
You are about to be asked to enter information
that will be incorporated into your certificate request.
```

Enter the organization name, location, and your name.

After you answer the questions prompted by the `./CA.pl -newca` command, the following files are created:

- The `./demoCA/cacert.pem` file. This is the CA certificate file that you can exchange with communication partners for TLS authentication or verification.
- The `./demoCA/private/cakey.pem` file. This is the private key file of the CA and is passphrase-protected. You can use this private key to sign or revoke certificates.

NOTE: Do not exchange the private key file with communication partners.

b. To generate the certificate and the key pair for the Sendmail 8.13.3 server, enter the following command:

```
$ ./CA.pl -newreq
```

The following output displays:

```
Generating a 1024 bit RSA private key...
+++++.........................++++++
writing new private key to 'newkey.pem'
-----
You are about to be asked to enter information that will
be incorporatedinto your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave
some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Enter the your organization name, location, and name.

The `./CA.pl -newreq` command creates the following files:

- The private key of the Sendmail 8.13.3 server (`./newkey.pem`)
- The original (unsigned) certificate request (`./newreq.pem`)

c. To sign the certificate using the CA created in Step a, enter the following command:

```
$ CA.pl -sign
```

A signed public certificate, `./newcert.pem` (with its public key), is created for the Sendmail 8.13.3 server.

5. To create a subdirectory `certs` under the `/etc/mail` directory, enter the following command:

```
mkdir -p /etc/mail/certs
```

6. To set the appropriate permissions to the `certs` subdirectory, enter the following command:

```
chmod 755 certs
```

7. To change the directory location to `certs`, enter the following command:

```
cd /etc/mail/certs
```

8. To copy the previously created CA certificate, the Sendmail 8.13.3 server certificate, and the key from the `/opt/openssl/misc/` directory to the `/etc/mail/certs` directory, enter the following commands:

```
cp /opt/openssl/misc/demoCA/cacert.pem
/etc/mail/certs/cacert.pem
cp /opt/openssl/misc/newkey.pem
/etc/mail/certs/servername-key.pem
cp /opt/openssl/misc/newcert.pem
/etc/mail/certs/servername-cert.pem
```

9. To create a hashed symbolic link to the CA certificate, enter the following command:

```
ln -s cacert.pem `openssl x509 -noout -hash < cacert.pem`.0
```

This command reads the `cacert.pem` file and creates an 8-character cryptographic hash, which is used as the filename (with '.0' appended) that links to the CA certificate. During a certificate exchange in an SSL handshake, Sendmail 8.13.3 computes the hash of the received public key of the CA certificate, appends '.0' to the hash, and compares the computed hash with its own copy of the public key of the CA certificate.

10. To verify whether the symbolic link to the CA certificate is created properly, enter the following command:

```
ll *.0
```

Ensure that you obtain an output similar to the following:

```
2197 lrwxrwxrwx 1 root sys 10 Jul 9 09:44 fea4e1bb.0 -> cacert.
pem
```

NOTE: The link name `fea4e1bb.0` is only an example. The link name must be of the format `<characters>.0`.

The Sendmail 8.13.3 server is now ready with the signed public certificate and the private key pair. If you have multiple Sendmail 8.13.3 servers (for example, relay and forwarders), you can either create an individual key pairs and a signed certificate for each Sendmail 8.13.3 server and get it signed by the CA, or use the cryptographic keys to be distributed across the Sendmail 8.13.3 servers in your environment.

The previously mentioned Sendmail 8.13.3 configuration option considers that you are using the same certificate and key, irrespective of whether Sendmail 8.13.3 acts in a client mode or a server mode. If you need different pairs of certificate and keys for these two operational modes, you must create them using the procedure described in the "Generating Certificates and Keys" (page 14) and rename them appropriately (such as `clientname-cert.pem` and`clientname-key.pem`). You must also configure the file names against `ClientKeyFile` and `ClientCertFile` options in the Sendmail 8.13.3 configuration file, as described in "Configuring Sendmail 8.13.3 with TLS and SSL" (page 13).

Do not store the private key of the CA (`/opt/openssl/misc/demoCA/private/cakey.pem`) in the Sendmail 8.13.3 servers.

Configuring the Sendmail 8.13.3 Server with TLS/SSL

To configure the Sendmail 8.13.3 server with TLS/SSL, you must create a new Sendmail 8.13.3 configuration file with `STARTTLS` feature enabled using the HP-UX `gen_cf` utility.

NOTE: If you do not have a `/etc/mail/submit.cf` file, you cannot enable the Mail Submission Program (MSP). Hence, you can skip the `submit.cf` additions or changes discussed in this white paper.

If you have any site-specific customized configuration in your `/etc/mail/sendmail.cf` file or `/etc/mail/submit.cf` file, ensure that you back up the customized changes. Obtain a backup of the existing Sendmail 8.13.3 configuration files (`/etc/mail/sendmail.cf` and `/etc/mail/submit.cf`) enables you to revert to the original state in case you encounter any configuration issues.

Follow this procedure to configure the Sendmail 8.13.3 server with TLS/SSL:

1. To back up the existing Sendmail 8.13.3 configuration file, enter the following command:
   ```
   cp -p /etc/mail/sendmail.cf /etc/mail/sendmail.cf.BACKUP
   ```
2. To change the directory to the `/usr/newconfig/etc/mail/cf/cf` directory, enter the following command:

```
cd /usr/newconfig/etc/mail/cf/cf
```

3. To run the `gen_cf` utility, enter the following command:
   ```
   $ ./gen_cf
   ```
4. Select the `2: STARTTLS` option under the `4: Security Options` option in the main menu.
5. Select the `5: Generate sendmail.cf` option and press **Enter** to generate the Sendmail 8.13.3 configuration file (`sendmail.cf.gen`) with the `STARTTLS` feature enabled.
6. Repeat Steps 1–4.
7. Select the `6: Generate submit.cf` option and press **Enter** to generate the configuration file of the Sendmail 8.13.3 client queue runner (`submit.cf.gen`) with the `STARTTLS` feature enabled.
8. Copy the previously created `sendmail.cf.gen` file and the `submit.cf.gen` file to the `/etc/mail/sendmail.cf` file and the `/etc/mail/submit.cf` file, respectively.
9. Open the Sendmail 8.13.3 configuration files (`/etc/mail/sendmail.cf`and `/etc/mail/submit.cf`) and edit the `UseTLS`, `CACertPath`, `CACertFile`, `ServerCertFile`, `ServerKeyFile`, `ClientCertFile`, `ClientKeyFile`, and `RandFile` options, as follows:

   ```
   # If set, Sendmail enables the TLS feature
   UseTLS=True
   # CA directory
   CACertPath=/etc/mail/certs
   # CA file
   CACertFile=/etc/mail/certs/cacert.pem
   # Server Cert
   ServerCertFile=/etc/mail/certs/servername-cert.pem
   # Server private key
   ServerKeyFile=/etc/mail/certs/servername-key.pem
   # Client Cert
   ClientCertFile=/etc/mail/certs/servername-cert.pem
   # Client private key
   ClientKeyFile=/etc/mail/certs/servername-key.pem
   # Random data source (required for systems without /dev/urandom under OpenSSL)
   RandFile=egd:/dev/random
   ```

10. Follow this procedure if Mail Submission Program (MSP) is enabled for Sendmail 8.13.3:
    a. To change the directory to `/etc/mail/certs`, enter the following command:
       ```
       cd /etc/mail/certs
       ```
    b. To change the mode to `640` for all the private keys, enter the following command:
       ```
       chmod 640 *.pem
       ```
    c. To change the group for all the private keys, enter the following command:
       ```
       chgrp smmsp *.pem
       ```
    d. To update the configuration file of the Sendmail 8.13.3 MTA (`/etc/mail/sendmail.cf`) and the configuration file of the MSP

(`/etc/mail/submit.cf`), use the following option in the Sendmail 8.13.3 configuration file:
`DontBlameSendmail=GroupReadableKeyFile`

11. Follow this procedure if MSP is disabled in Sendmail 8.13.3:
    a. To change the directory to `/etc/mail/certs`, enter the following command:
    `/ cd /etc/mail/certs`

NOTE: For more information about configuring Sendmail 8.13.3, see the *HP-UX Mailing Services Administrator's Guide* at:
http://www.docs.hp.com/en/netcom.html#Internet%20Services

    b. To change the mode for all the private keys, enter the following command:
    `chmod 600 *.pem`
    c. To change the group for all the private keys, enter the following command:
    `chgrp root *.pem`
    d. To restart the Sendmail 8.13.3 daemons, enter the following commands:
    `/sbin/init.d/sendmail stop`
    `/sbin/init.d/sendmail start`

Verifying the TLS/SSL Configuration

Follow this procedure to verify the TLS/SSL configuration:

1. Examine the output of the `mtail` command to ensure that Sendmail 8.13.3 does not contain any error or warning after configuring TLS/SSL.
2. Send a test mail using Sendmail 8.13.3 and verify if the mail is delivered to the destination address.
3. Ensure that you notice STARTTLS in certain Sendmail 8.13.3 log entries to ascertain the proper configuration of STARTTLS.
4. Establish a Telnet session to port `25` or `*587` of the server configured recently, to ensure that it offers the `STARTTLS` support in response to the `EHLO` command. The `*587` port is used if MSP is enabled for Sendmail 8.13.3.

   Following is a sample Telnet session, which ascertains the STARTTLS support:

```
$ telnet localhost 25
Trying...
Connected to localhost.<domain_name>
Escape character is '^]'.
220 <hostname>.<domain-name> ESMTP Sendmail @(#)Sendmail
version 8.13.3 - Revision
 2.005 - 12 January 2007/8.13.3; Fri, 4 May 2007 18:00:30 +
0530 (IST)
```

   where:

<hostname> Specifies the host name.

<domainname> Specifies the domain name.

Enter the EHLO command, as follows:

```
$ EHLO localhost
```

The following output displays:

```
250-<hostname><domain-name> Hello localhost [127.0.0.1],
pleased to meet you
250-ENH ANCEDSTATUSCODES
 250-PIPELINING
 250-EXPN
 250-VERB
 250-8BITMIME
 250-SIZE
 250-DSN
 250-ETRN
 250-AUTH DIGEST-MD5 CRAM-MD5
250-STARTTLS
 250-DELIVERBY
 250 HELP
```

You must ensure that you obtain a response similar to 250-STARTTLS from the Sendmail 8.13.3 server. Additionally, ensure that you get the 250-STARTTLS line in response to the SMTP EHLO command. This indicates that STARTTLS is configured correctly.

Enter the following to indicate that you want to quit the Telnet session:

```
$ QUIT
```

The following output displays:

```
221 2.0.0 <hostname>.<domain-name> closing connection
 Connection closed by foreign host.
```

5. Send a mail using the STARTTLS configured Sendmail 8.13.3 server and ensure that you get the (version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256 verify=OK) line in the following message header:

```
From root @<sender_name>.<domain_name> Thu Jul 5 19:19:30
IST 2007
Received: from <sender_name>.<domain_name> by <hostname>.
<domain_name>
(@(#)Sendmail version 8.13.3 - Revision 2.005 - 12 January
2007/8.13.3) with ESMTP id l65DnTLe028546
(version=TLSv1/SSLv3 cipher=DHE-RSA-AES256-SHA bits=256
verify=OK)
for <hostname>.<domain_name>; Thu, 5 Jul 2007 19:19:30
+0530 (IST)
Received: (from root@localhost) by <sender_name>.<domain_name>
((@(#)Sendmail
version 8.13.3 - Revision 1.000 - 1st August,2006/8.13.3)id
l65Dbpdc008315
```

```
for root@<hostname> Thu, 5 Jul 2007 19:07:51 +0530 (IST)
Date: Thu, 5 Jul 2007 19:07:51 +0530 (IST)
From: <server_name>.<domain_name>
```

The `verify` macro in the message header in the `mtail` command output contain the result of the verification of the presented certificate.

The `verify` macro can contain the following values:

`OK`	Verification succeeded.
`NO`	No certificate presented.
`FAIL`	Certificate presented, but is not be verified. For example, CA is missing.
`NONE`	`STARTTLS` was not performed.
`TEMP`	Temporary error has occurred. For example, Sendmail 8.13.3 has received a `454` message from its peer.
`PROTOCOL`	Protocol error occurred.
`SOFTWARE`	Problems incurred during the handshake at the TLS level. In this case, the connection is dropped.

Configuring Sendmail 8.13.3 with SASL

This section addresses the following topics:

Prerequisites

Following are the prerequisites for configuring SASL in Sendmail 8.13.3:

NOTE: The prerequisites are common for both the Sendmail 8.13.3 server and client configurations.

- The KRNG11i strong random number generator

NOTE: The KRNG11i strong random number generator is required only for the HP-UX 11i v1 operating system. For the HP-UX 11i v2 and 11i v3 operating systems, the random number generator is available as part of the core HP-UX operating system.

- The OpenSSL software

NOTE: You must install the latest version of the OpenSSL software from http://www.software.hp.com, to avoid errors while running the `CA.pl` script. Table 2 (page 23) lists the version of OpenSSL that you must install for a particular HP-UX operating system.

Table 2 OpenSSL Versions

Operating System Name	OpenSSL Version
HP-UX 11i v1	A.00.09.07l
HP-UX 11i v2	A.00.09.07l.001
HP-UX 11i v3	A.00.09.08d.001

- The CyrusSASL version A.06.00-2.1.21 product in the HP-UX Internet Express bundle.

NOTE: Do not use the latest version of CyrusSASL (A.06.00-2.1.22), because it poses some issues with Sendmail 8.13.3.

- The latest version of the Sendmail 8.13.3 web upgrade.

NOTE: For the HP-UX 11i v3 operating system, Sendmail 8.13.3 is available as part of the core HP-UX operating system. For the HP-UX 11i v1 and HP-UX 11i v2 operating systems, Sendmail 8.13.3 is available as a web upgrade at:

http://www.software.hp.com.

Setting up the Sendmail 8.13.3 Server with SASL

Before setting up the Sendmail 8.13.3 server with SASL, ensure that you have completed the following tasks:

- The STARTTLS feature is set up, as discussed in "Configuring Sendmail 8.13.3 with TLS and SSL" (page 13), before using the `DIGEST-MD5` algorithm.
- The SASL libraries are installed in a default location accessed by Sendmail 8.13.3. The libraries must be safe, that is, they must be owned by the superuser and must be writable only by the superuser. The path of the SASL libraries must also be safe.
- The existing Sendmail 8.13.3 configuration files (`/etc/mail/sendmail.cf` and `/etc/mail/submit.cf`) are backed up so that you can revert to the original state if you encounter any configuration issues and you do not loose any site-specific customized configuration.

Follow this procedure to set up the Sendmail 8.13.3 server with SASL:

1. Uncomment the following entries in the `/etc/mail/sendmail.cf` file:

```
C{TrustAuthMech}GSSAPI DIGEST-MD5 LOGIN PLAIN
O AuthMechanisms=GSSAPI KERBEROS_V4 DIGEST-MD5 CRAM-MD5
 LOGIN PLAIN EXTERNAL
O DefaultAuthInfo=/etc/mail/default-auth-info
O AuthOptions=A
```

NOTE: If these entries are already uncommented, ensure that you assign the previously mentioned values to the `TrustAuthMech`, `AuthMechanisms`, `DefaultAuthInfo`, and `AuthOptions` options.

2. Create the `/usr/lib/sasl2/Sendmail.conf` file with following entries:

```
pwcheck_method: auxprop mech_list:
login plain cram-md5 digest-md5
```

Ensure that you have provided permission only for the superuser to access the `/usr/lib/sasl2/Sendmail.conf` file, as follows:

```
# cd /usr/lib/sasl2/
# chmod 600 Sendmail.conf
```

3. To set the SASL password of the user for server programs and SASL mechanisms that use the standard `libsasl` database of user secrets, enter the following command:

`saslpasswd2 -a` *`appname`* `-c [–u]` *`<server_domain_name>`* *`userid`*

The `saslpasswd2` command prompts and accepts the SASL password, and creates the `/etc/sasldb2` file.

Example 1 Sample saslpasswd2 Command

Following is a sample `saslpasswd2` command:

```
saslpasswd2 -a Sendmail -c –u <domain-name> root
```

where:

`Sendmail`	Specifies the application name.
`<domain-name>`	Specifies the domain name of the Sendmail 8.13.3 server.
`root`	Specifies the user ID.

4. To restart the Sendmail 8.13.3 server, enter the following commands:

```
/sbin/init.d/sendmail stop
/sbin/init.d/sendmail start
```

Verifying the SASL Configuration

Follow this procedure to ensure that SASL is set up properly on the Sendmail 8.13.3 server:

1. To establish a Telnet session with the `localhost`, enter the following command:

   ```
   % telnet localhost 25
   ```

   The following output displays:

   ```
   Trying...
   Connected to localhost.<domain-name>
   Escape character is '^]'.
   220 <hostname> <domain-name>ESMTP Sendmail @(#)Sendmail
   version 8.13.3 - Revision 1.000
    - 1st August,2006/8.13.3; Thu, 5 Jul 2007 18:37:50 +0530
   (IST)
   ```

2. To send an `EHLO` message to the Sendmail 8.13.3 server, enter the following at the Telnet prompt:

   ```
   EHLO localhost
   ```

   The following output displays:

   ```
   250-<hostname> <domain-name> Hello localhost [127.0.0.1], pleased to meet you its
   250-ENHANCEDSTATUSCODES
   250-DSN
   250-AUTH CRAM-MD5 LOGIN PLAIN
   250-AUTH CRAM-MD5 LOGIN PLAIN
   250 HELP
   ```

NOTE: Ensure that you get a response similar to the previous output from the Sendmail 8.13.3 server

3. To quit the Telnet session, enter the following command:

   ```
   Quit
   ```

   The following output displays:

   ```
   221 2.0.0 <hostname> <domain-name> closing connection
   Connection closed by foreign host.
   ```

NOTE: If you encounter any problem while verifying the SASL setup, check the syslog file using the `mtail` command for any security problems (for example, unsafe files). If you are unable to identify the problem, increase the log level to `13` in the `/etc/mail/sendmail.cf`, and restart the Sendmail 8.13.3 server using the following commands:

```
/sbin/init.d/sendmail stop
/sbin/init.d/sendmail start
```

Using Sendmail 8.13.3 with AUTH

This section discusses how to use Sendmail 8.13.3 with the `AUTH` feature.

Follow this procedure if you use Sendmail 8.13.3 only to transfer mail from your local computer to a mail server that requires `SMTP AUTH` authentication:

1. Follow this procedure to generate the /etc/mail/sendmail.cf file:

NOTE: Ensure that you obtain a backup of the existing Sendmail 8.13.3 configuration files (/etc/mail/sendmail.cf and /etc/mail/submit.cf) so that you can revert to the original state if you encounter any configuration issues and you do not loose any site-specific customized configuration.

   a. To change the directory to the /usr/newconfig/etc/mail/cf/cf directory, enter the following command:
      ```
      cd /usr/newconfig/etc/mail/cf/cf
      ```
   b. To copy the generic-hpux10.mc file to the generic-hpux10.mc.ORIG file, enter the following command:
      ```
      cp generic-hpux10.mc generic-hpux10.mc.ORIG
      ```
   c. Add the following entries to the generic-hpux10.mc file and enter the sendmail-sasl server name:
      ```
      define(`SMART_HOST',<` sendmail-sasl server name'>dnl
      define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM
      - MD5 LOGIN PLAIN')dnl
       FEATURE(`authinfo',`hash /etc/mail/auth/client-info')dnl
      ```
   d. To generate the sendmail.cf.gen file, enter the following command and select the "5: Generate sendmail.cf" option in the main menu:
      ```
      ./gen_cf
      ```
   e. To copy the /etc/mail/sendmail.cf file to the /etc/mail/sendmail.cf.PREVIOUS file, enter the following command:
      ```
      cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.PREVIOUS
      ```
   f. To copy the sendmail.cf.gen file to the /etc/mail/sendmail.cf file, enter the following command:
      ```
      cp sendmail.cf.gen /etc/mail/sendmail.cf
      ```
   g. To restore the generic-hpux10.mc file, enter the following command:
      ```
      cp generic-hpux10.mc.ORIG generic-hpux10.mc
      ```
2. Follow this procedure to set up Sendmail 8.13.3 as a SASL client:
   a. To change directory to the location where the Sendmail 8.13.3 configuration files are located (usually the /etc/mail/. directory), enter the following command:
      ```
      cd /etc/mail/
      ```
   b. To create a safe subdirectory called auth under the /etc/mail/ directory, enter the following commands:
      ```
      mkdir auth
      chmod 700 auth
      ```
   c. To create a file called client-info under the /etc/mail/auth directory, enter the following command:
      ```
      vi client-info
      ```

d. Enter your authentication information in the `client-info` file using the following syntax:

```
AuthInfo: <server_name> "U:root" "I: <username>" "P:
<password>" "M: <auth_mech>"
```

where:

`server_name`	Specifies the Sendmail 8.13.3 SASL server name.
`username`	Specifies the user name to which the authentication information applies.
`password`	Specifies the password that is configured using the `saslpasswd2` command in the Sendmail 8.13.3 SASL server.
`auth_mech`	Specifies the list of client preferred authentication mechanisms in the ascending order.

Example 2 Sample Authentication Information

Following is a sample authentication information that you can enter in the `client-info` file:

```
AuthInfo:hostname.domain-name "U:root" "I:root" "P:abc"
 "M:DIGEST-MD5 CRAM-MD5 PLAIN LOGIN"
```

NOTE: In this example, the client first uses the `DIGEST-MD5` mechanism. If this mechanism fails or the server does not support this mechanism, the client uses the `CRAM-MD5` mechanism. Similarly, the client uses the subsequent authentication mechanisms if the current authentication mechanisms fail.

e. To update the `DefaultAuthInfo` option in the `/etc/mail/sendmail.cf` file, use the following entry:

```
#O DefaultAuthInfo=/etc/mail/auth/client-info
```

f. To generate the authentication database and to provide readable permission only to the superuser, enter the following commands:

```
# cd auth
# makemap hash client-info
# chmod 600 client-info*
# cd ..
```

g. To restart the Sendmail 8.13.3 server, enter the following commands:

```
/sbin/init.d/sendmail stop
/sbin/init.d/sendmail start
```

NOTE: If you use the `FEATURE(`authinfo')` option, the hostname in the map entry must match exactly with the hostname of the ISP mailserver, as explained in the `../cf/README` file.

Sendmail 8.13.3 searches only for domain parts or IP nets if you use the access map. If you use the `authinfo` feature, Sendmail 8.13.3 performs only three lookups, one default and two exact matches.

Verifying the Cyrus SASL Setup

This section discusses how to verify the Cyrus SASL setup.

Follow this procedure to verify the Cyrus SASL setup:

1. From the system where you have configured the client to the Sendmail 8.13.3 SASL server, enter the following commands to send mails to the Sendmail 8.13.3 server:
   ```
   echo "TEST MAIL" | sendmail -v username@server.domain |
   test.log
   echo "TEST MAIL" | sendmail –v root@XXX.yyy.com | test.log
   ```
   You can also use other mail clients, such as `elm`, `mailx`, and `mail`, to send mails.
2. Ensure that you receive a similar message, specified as follows, to denote that the Cyrus SASL setup is working properly:
   ```
    ">>> AUTH CRAM-MD5
   334 PDM2MzU1OTY5NDEuNDkxNjA3OEBpbmV0MTEuaW5kaWEuaHAuY29tPg== >
   >> cm9vdCBmNWFkMGIyMDQzZGE2YTNkZmUwOTUxYWU2ZTU5NTg2Yg
   == 235 2.0.0 OK Authenticated"
   ```

NOTE: This message varies with the `AUTH` mechanism used, but you must receive the `235 2.0.0 OK Authenticated` message to denote that the Cyrus SASL setup is working properly. On the peer system, ensure that you receive the `authenticated bits=0` message in the corresponding message header.

Related Information

For more information on TLS/SSL and Cyrus SASL, see the following websites:

- http://www.sendmail.org/~ca/email/auth.html
- http://www.sendmail.org/~ca/email/starttls.html
- http://docs.hp.com/en/netcom.html#Internet%20Services
- http://www.sendmail.org/misc/other-non-sendmail-links.php

Glossary

AES	Advanced Encryption Standard (AES) also known as Rijndael, is a encryption method that operates on fixed-length groups of bits and termed blocks.
DES	Data Encryption Standard (DES) is a method for encrypting information.
ESMTP	Extended Simple Mail Transfer Protocol (ESMTP) specifies extensions to the original protocol (SMTP) for sending mail that supports graphics, audio, and video files, and text in various national languages.
IDEA	IDEA (International Data Encryption Algorithm (IDEA) is a 64-bit iterative block cipher with a 128-bit key.
IMAP	Internet Message Access Protocol (IMAP) is a method for accessing electronic mail or bulletin board messages stored in a mail server.
MD5	Message Digest algorithm 5 (MD5) is a widely used cryptographic hash function with a 128-bit hash value.
RIPEMD	RACE Integrity Primitives Evaluation Message Digest (RIPEMD) is a message digest algorithm and cryptographic hash function.
RSA	Rivest Shamir Adleman (RSA) is an algorithm for public-key cryptography. It is the first algorithm suitable for signing and encryption, and one of the first great advances in public key cryptography.
SMTP	Simple Mail Transfer Protocol (SMTP) is a simple, text-based protocol for sending mail messages.