Tim Reeves
Internet Services

Inhaltsverzeichnis

Mailserver General Strategy – what do we want to achieve?

This document covers only the basic use of IMAP and SMTP protocols and how to configure your (Plesk-Ubuntu 12.04) server for those purposes. In particular, it does not cover supplementary methods of securing E-Mail contents such as Authentication, Integrity, or Confidentiality (here in German) such as PGP, S/MIME, SPF and DKIM; or anti-spam techniques for users and or servers.

This is what happens when Alice sends an E-Mail to Bob:

```
Alice => Send (SMTP) => MSA => MTA     =>     MTA => Mailbox <=> MDA  <=> View (IMAP) Bob
(MUA)    <Internet>      Server-1  <Internet>        Server-2         <Internet>     (MUA)
```

MSA = Mail Submission Agent MTA = Mail Transfer Agent

MDA = Mail Display Agent MUA = Mail User Agent

Using her Mail User Agent (e.g. Thunderbird, Outlook, Android-Mail etc.) Alice composes an E-Mail and sends it off to Bob. Her MUA connects to her configured MSA (Submission Agent) using the SMTP protocol and passes the E-Mail on to it for forwarding. Typical MSA's are Postfix, Sendmail or Qmail. Although logically a new step is required – transferring the mail onwards – this is normally done by the same programm acting in a new role, because the transferring to the next (or final) destination also uses the SMTP protocol.

Normally there will only be one transferring “hop”, assuming the final destination server is directly available on the internet. But sometimes, e.g. for a large company, all mail will be

accepted by a proxy server (often for security reasons), and passed on to the final destination which is not publicly accessible.

When the mail reaches its final destination server, the MTA there is responsible for putting it in the user's mailbox (Maildir) there. Thus the way the MTA accesses the Mailbox must be coordinated with the MDA (Mail Display Agent) to ensure that they do not lock each other out or cause race conditions. A very popular Unix/Linux MDA is “courier imap”.

Finally, Bob runs up his PC (or Smartphone) and starts his MUA, which connects to his configured server and MDA via the IMAP protocol, and initiates a dialog to check for any new E-Mails or changes to E-Mails in any other E-Mail folders, and displays these changes to Bob – in our case, the new E-Mail from Alice.

Goals

1. End Users should be able to connect with their E-Mail clients via SSL or TLS, so that display and sending of mails happens encrypted, to protect the private sphere.
2. While doing so there should be no complaints from the Client about wrong SLL certificates.
3. The Server, after acting as MSA then acts as (boundary) MTA, passing mails on to another server, for final delivery (or further transfer). To do so it has to locate the target host, for which it uses the domain name system (DNS) to look up the mail exchanger record (MX record) for the recipient's domain (the part of the E-Mail address on the right of @). The returned MX record contains the name of the target host. That host accepts connections only for E-Mails whose recipients are local (or, for a proxy, which it knows about).
4. Thus it is not important which IP or Certificate the MTA uses when transferring mail to another SMTP server – that Server will only accept mails for local or known recipients. It does not check if the sending server is using an IP and Certificate matching the sender's domain (which could not be the case if the transfer requires more than one hop). Rather, for more strict control of origin, the DKIM method can be used.

Scenario with several domains on the same Server and IP

When you have several domains on the same IP address, it does not matter which of them your users supply as the server address in their E-Mail clients, because they will reach the same instances of submission and display agents running behind that IP address. Those instances know the data of all the mailboxes on all the domains of that IP.

E-Mail client configuration

So, for example, for abc.org and def.org, you can give the server name in both cases as [mail.]abc.org (rather than in one case mail.def.org). This is the best approach, be-

cause the server can only present one SSL certificate per IP address, and in our case, it is presenting a certificate valid for [mail.]abc.org (meaning that it covers both abc.org and mail.abc.org, so you can use either form). Remember to change to [mail.]abc.org on all your accounts at that server, both incoming (IMAP) and outgoing (SMTP). Then you should have no more problems with the client complaining about a wrong certificate. Regarding the server name, it should present itself as [mail.]abc.org, the same as the certificate and the way clients are configured.

Use these values for IMAP: Server = [mail.]abc.org, Port = 993, Security method: SSL/TLS, Password: Normal or encrypted (should not matter).

For SMTP you need only define one account and set it as standard, and use it for all sending of mail (unless you have an account at e.g. google or freenet or such as well, then also define their SMTP server for those accounts): Server = [mail.]abc.org, Port = 587, Security method = STARTTLS, Password: Normal or encrypted (should not matter).

Ubuntu Mail Setup in general

See also article here

Postfix sends and receives mail from Internet and stores them in the user mailboxes while clients on the Internet can retrieve their mails via Courier IMAP or POP3. The user authentication is performed by Courier Authdaemon.

Since version 12 Plesk offers Dovecot as an alternative to Courier, and since Dovecat in general has the better reviews I also recommend it. This article however describes – for historical reasons – the setup of Courier.

Where are the Plesk-Ubuntu mailboxes? (Maildir)

All the user mailboxes are (under Plesk) at /var/qmail/mailnames

The pattern under that root directory (all files belonging popuser:popuser) is:

1 Dir: <domain-name> (or: <subdomain-name>) - using dots

2 Mailbox-Name (w/o domain,e.g. "tim.reeves")
Contains (files) .qmail .spamassassin, (dirs) @attachments, Maildir

3 Maildir:
Dirs: .<Folder> (that's dot folder) ... courierimapkeywords cur new tmp
Files: courierimapsubscribed courierimapuiddb maildirsize

Courier Imap Config (Vn. 4.12.0)

Re courierimapuiddb & courierimapkeywords see here and here.

- /etc/courier-imap - this is the main config dir for imap and pop3 (subdirs “shared” + “shared.tmp” remain empty)
- TLS_CERTFILE: **/usr/share/imapd.pem** /usr/share/pop3d.pem

- /usr/sbin/courierlogger - Courier syslog wrapper (several daemon instantiations) - accepts incoming network connections, and runs program after establishing each network connection. Runs as a daemon several times (each apparently started by courierlogger):
  - 143 /usr/sbin/imaplogin (=> /usr/bin/imapd)
    (IMAP-Port mit STARTTLS bzgl. Verschlüsselung)
  - 993 /usr/bin/couriertls (=> /usr/bin/imapd)
    (IMAP-Port für grundsätzliche SSL-Verschlüsselung)
    This makes using IMAP via Port 993 with TLS in Clients a good idea...
  - 995 /usr/bin/couriertls (=> /usr/lib/plesk-9.0/pop3login)
    (POP-Port bei SSL-Verschlüsselung)
- /usr/lib/couriertcpd - Courier mail server TCP server daemon
- /usr/lib/courier-authlib - directory with programs / modules for authorisation stuff ; see also Dir: /etc/courier-imap/authlib
- /usr/lib/courier-authlib/authdaemond runs in 5 (!) daemon threads
- /usr/bin/couriertls
- man courierlogger ; man couriertcpd
- Logs: /var/log/mail.{ err | info | log | warn } /usr/local/psa/var/log/maillog

Courier and SSL Certificates

Firstly we need to remember that Courier-Imap is purely an IMAP Server, it is connected to only by MUAs and is used to manipulate the users E-Mails on the server, e.g. deleting or moving them, adding new subfolders etc. So the only certificate issue is between Courier and MUAs; as we have seen above, since all Maildirs on the server are (normally) handled by the one instance of Courier on that server, one can connect to the root IP of the machine, e.g. mail.abc.org, and provide authentication via any local E-Mail account, even if it is associated with a domain whose website is provided on another IP on the same server.

However, in a larger environement one may not want to imply the root domain / IP of the server to all users, in which case you can either (a) make it a special domain extra for that purpose (e.g. imap.strato.de) or (b) at least present the right certificate for the main domain (virtual host) of another IP on the machine when accessed over that IP. Courier allows you to configure this as follows.

Excerpt from config file /etc/courier-imap/imapd-ssl:

WARNING: **Peer certificate verification has NOT yet been tested.**
Only the basic SSL/TLS functionality is known to be working.

TLS_CERTFILE - certificate to use. TLS_CERTFILE is required for SSL/TLS servers, and is optional for SSL/TLS clients. TLS_CERTFILE is usually treated as confidential, and must not be world-readable. Set TLS_CERTFILE instead of TLS_DHCERTFILE if this is a garden-variety certificate.

VIRTUAL HOSTS (This applies; *GnuTLS only* does not apply, Ubuntu uses OpenSSL)

Due to technical limitations in the original SSL/TLS protocol, a dedicated IP address is required for each virtual host certificate. If you have multiple certificates, install each certificate file as

- **$TLS_CERTFILE.aaa.bbb.ccc.ddd**
  where "aaa.bbb.ccc.ddd" is the IP address for the certificate's domain name.

So, if TLS_CERTFILE is set to **/usr/share/imapd.pem**, then you'll need to install the actual certificate files as e.g. /usr/share/imapd.pem.87.230.18.99, /usr/share/imapd.pem.217.115.133.24 and so on, for each IP address.

A further Config tip

See: http://forum.parallels.com/showthread.php?289918-After-update-to-Plesk-11-5-Courier-IMAP-no-longer-works So I edited /etc/courier-imap/imapd and bumped up concurrency as suggested. Seems Plesk made a booboo and made 4 instead of 40 concurrent connections.

Postfix Config (Vn. 2.9.6** – 4 Feb. 2013**)

Not a bad idea to read up some background on Postfix and configuring it, here are some good links:

- http://de.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol
- http://en.wikipedia.org/wiki/SMTPS
- http://www.360is.com/06-postfix.htm
- http://superuser.com/questions/536995/why-is-smtp-over-ssl-between-email-servers-not-so-popular
- http://security.stackexchange.com/questions/12087/if-i-send-an-email-over-ssl-will-it-always-be-transmitted-over-ssl
- http://stackoverflow.com/questions/13437484/why-dont-googles-mx-servers-match-the-ssl-certificate-cn
- http://www.postfix.org/documentation.html
- http://www.postfix.org/VIRTUAL_README.html
- http://www.postfix.org/ADDRESS_CLASS_README.html
- http://blog.wpkg.org/2013/07/31/postfix-and-multiple-ssl-certificates/
- http://postfix.1071664.n5.nabble.com/multiple-ssl-certificates-for-multiple-domains-but-just-one-IP-td3615.html
- http://superuser.com/questions/664182/multiple-ssl-certificates-on-one-server

First and foremost: Postfix acts as MSA, accepting mail submissions from authenticated users on port 587; and as MTA, transferring accepted mails for non-local destinations to the destination SMTP server, normally using port 25. What we see from

some of the above articles is that while TLS is preferred for the MSA and MDA connections, especially for authentication, the MTA to MTA transfers may often be unencyrpted anyway, for 2 reasons: (a) it's technically and organisationally very challenging; and (b) the mails will be available unencrypted on each server they pass through anyway. Thus for real mail privacy, the mail content should be encrypted before it ever leaves the client computer, e.g. using PGP (OpenPGP or GnuGP in Thunderbird and Android).

Regarding SSL/TLS encryption on the MSA side, the Internet Protocolls involved look at SSL certificates before communicating the domain name, so one can only present the correct certificate for ONE domain on that IP (i.e. use only one domain on an IP, or at least assign the certificate for the most important mail domain on each IP).

- If you have a company server, then all internal mails will never be transferred to another server, so encrypting the submission and display of mails will suffice (assuming your server is well protected).
- But **for sending sensitive mails** to another server (e.g. another company), then you should really **encrypt the mail itself.**

There are other topics to take care of to ensure that your SMTP server is well accepted, find tips here and here and here.

- An own PTR Record matching Hostname and IP (as used for reverse delegation).
  See smtpd client
  restrictions: reject_unknown_client_hostname and
  reject_unknown_reverse_client_hostname
- The host name should match the certificate proffered by the server (e.g. mail.gemeinwohl.info)
- Do NOT set up any wildcard or catchall mail accounts on the server.
- abuse@yourdomain.tld and postmaster@yourdomain.tld addresses must exist and be functional.
- **TODO:** Deploy Domain Keys and SPF for outgoing messages.

If you notice warnings like this from Postfix in /var/log/maillog:
untrusted issuer /C=US/O=Equifax/OU=Equifax Secure Certificate Authority
then follow the instructions here and add the following line to /etc/postfix/main.cf:
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
Note that Plesk may overwrite this file when you change the SMTP setup in Plesk!

Dirs and Files on Plesk - Ubuntu

- **/etc/postfix** - here are the main config files – including a **postfix_default.pem** (/etc/ssl from OpenSSL seems NOT used)
- **/var/qmail/mailnames** - Postfix uses this qmail place to hold the actual mail directories
  (assume Plesk did that for interoperability, between Qmail, Postfix and Courier)

- /var/qmail/mailnames/<domain-name>
- /var/qmail/mailnames/<domain-name>/<mailboxname>
- /var/qmail/mailnames/<domain-name>/<mailboxname>/.qmail
  This file is essential to Postfix working – without it mails remain in the deferred queue
- /var/qmail/mailnames/<domain-name>/<mailboxname>/.spamassassin (dir, SpamAssassin)
- /var/qmail/mailnames/<domain-name>/<mailboxname>/@attachments (dir, seems unused)
- /var/qmail/mailnames/<domain-name>/<mailboxname>/Maildir (dir, mails live under here)
  e.g. /var/qmail/mailnames/lasslos.net/shunyata/Maildir
- /usr/lib/postfix - $daemon_directory - helper binaries, the "local (unix) services" in master.cf
- /var/lib/postfix - $data_directory – Postfix keeps its .db files here
- /usr/sbin/postfix - this is the binary for Postfix control program, see "man postfix" (and my scripts pfstart / pfstop / pfreload)
- /var/spool/postfix - $queue_directory, with lots of subdirs, for all types and states of delivery
- /usr/share/postfix - just some template config files
- Logs: /var/log/maillog and /var/log/mail.{ err | info | log | warn }
- SASL = Simple Authentication and Security Layer
- /etc/ssl/certs – OpenSSL maintains a list of known CAs here
- Sendmail: /usr/lib/sendmail symlink to /usr/sbin/sendmail (this is the Postfix compat version)

Configuring /etc/postfix/main.cf

See man 5 postconf

Here we see that Plesk is using the Non-Postfix Mailbox Store method, defining:

Delivering mail to local mailboxes (MTA)

- **virtual_transport = plesk_virtual**
  specifies ... the name of a master.cf entry that executes non-Postfix sw via the pipe delivery agent
- **virtual_mailbox_domains** = $virtual_mailbox_maps, hash:/var/spool/postfix/plesk/virtual_domains

This trail leads to further places where Plesk stores data about mailboxes and user authentication, which has no relevance to the SSL certificate presented when delivering mail to other servers.

Accepting mails for delivery to other hosts/domains (MSA)

- **mynetworks** = 127.0.0.0/8 [::1]/128 **87.230.18.99/32** [2a01:488:66:1000:57e6:1263:0:1]/128
  Specifies the list of "trusted" SMTP clients that have more privileges than "strangers".
  127.0.0.0/8 means 8 bits Prefix, leaving 24 to contain host addresses (/32 leaves none)
  TODO See if Plesk bumps this up when more IPs are added, if not do it manually
- **inet_interfaces** = all
  specifies all network interface addresses that the Postfix system should listen on.
- smtpd_tls_cert_file = /etc/postfix/postfix_default.pem || smtpd_tls_key_file = $smtpd_tls_cert_file
- postfix_default.pem is NOT modified by Plesk.
  Leave config as is and copy your main domain certificate to /etc/postfix/postfix_default.pem

Configuring /etc/postfix/master.cf

Plesk has added these lines to master.cf: (See: man 5 master)

Delivering mail to local mailboxes (MTA)

plesk_virtual unix - n n - - pipe flags=DORhu user=popuser:popuser argv=/usr/lib/plesk-9.0/postfix-local -f ${sender} -d ${recipient} -p /var/qmail/mailnames

See: http://www.postfix.org/virtual.8.html for some tips, however Parallels has obviously adapted the source code to their own requirements.

plesk_saslauthd unix y y y - 1 plesk_saslauthd status=5 listen=6 dbpath=/plesk/passwd.db

NOTE that these are of type "unix" i.e. internal services to the local machine, using the "pipe" binary at /usr/lib/postfix (see "man 8 pipe" for details).

Accepting mails for delivery to other hosts/domains (MSA)

Finally the following lines (at EOF) with (the only) inet types governing external access to Postfix (the service name syntax depends on the service type):

87.230.18.99- unix - n n - - smtp -o smtp_bind_address=87.230.18.99 -o smtp_bind_address6= -o smtp_address_preference=ipv4

87.230.18.99- “a pathname relative to the Postfix queue directory” – hm don't believe that

unix The service listens on a UNIX-domain socket and is accessible for local clients only

- Private

n Not unprivileged – runs as (mail_owner?) root or postfix

n Don't Chroot to the mail queue directory (/usr/lib/postfix)

- - No automatic wakeup, use default process limit

smtp Run /usr/lib/postfix/smtp, the "Postfix SMTP+LMTP client", see "man 8 smtp"

smtp inet n - - - - smtpd

smtps inet n - - - - smtpd -o smtpd_tls_wrappermode=yes

submission inet n - - - - smtpd -o smtpd_**enforce_tls=yes** -o smtpd_**tls_security_level=encrypt** -o smtpd_sasl_auth_enable=yes -o smtpd_client_restrictions=permit_sasl_authenticated,reject -o smtpd_sender_restrictions= -o smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject_unauth_destination

The above mirror standard values as in the start-of-file comment, except "submission", which got a lot of (good looking) specific options. 1st is for port 25 (acting as MTA), 2nd for (deprecated) SMTPS (**SSL**) on port 465, 3rd for ESMTPS submission with START**TLS** on **port 587** (acting as MSA). See here why you should NOT use port 465. Anyway, certificate presentation will be needed on ALL these ports!

Managing Postfix

- mailq (same as...)
- postqueue -p List up the current mail queue
- postfix check Warn about bad directory/file ownership or permissions, and create missing directories
- Look at the logs: Logs: /var/log/maillog + /var/log/mail.{ err | info | log | warn }

ADD THIS SECTION TO MASTER.CF FOR EACH EXTRA IP

```
<ip>:smtp inet n - - - - smtpd
 -o smtpd_tls_cert_file=/etc/postfix/<yourCertForThisIp>.pem

<ip>:smtps inet n - - - - smtpd
 -o smtpd_tls_wrappermode=yes
 -o smtpd_tls_cert_file=/etc/postfix/<yourCertForThisIp>.pem

<ip>:submission inet n - - - - smtpd
 -o smtpd_enforce_tls=yes
 -o smtpd_tls_security_level=encrypt
 -o smtpd_etrn_restrictions=reject
 -o smtpd_sasl_auth_enable=yes
 -o smtpd_client_restrictions=permit_sasl_authenticated,reject
 -o smtpd_sender_restrictions=
 -o
smtpd_recipient_restrictions=permit_mynetworks,permit_sasl_authenticated,reject_unauth_destina
tion
 -o smtpd_tls_cert_file=/etc/postfix/<yourCertForThisIp>.pem
```

BTW 2014 Apache 2.2.22, Roundcube: 0.9.5 [me 0.9.0, Avail 0.9.5]) so HostEurope have done us proud on Roundcube, so just use standard webmail.

BTW Thunderbird: Seems that TB by default allows up to 5 concurrent connections to a particular mail server. When TB is configured with several mail accounts – at different domains on the same IP – then the Server easily reaches its connection limit on that IP for a particular client. I have found that reducing the max allowed number per email account from 5 can 3 help:

Extras | Konten-Einstellungen | <Konto> | Server-Einstellungen | Erweitert | Maximale Anzahl der aufrecht erhaltenen Serververbindungen