

Betreff: CYRUS-README.txt
Von: jschmidt@home.revamp-it.ch (Jan Schmidt)
Datum: 26.07.2017 12:53
An: jan.schmidt@revamp-it.ch
Return-Path: <jschmidt@home.revamp-it.ch>
Received: from home.revamp-it.ch ([unix socket]) by flunder-mail (Cyrus git2.4.17+0-Debian-2.4.17+nocaldav-0+deb8u2revamp1) with LMTPA; Wed, 26 Jul 2017 12:53:57 +0200
X-Sieve: CMU Sieve 2.4
Received: by home.revamp-it.ch (Postfix, from userid 1114) id 473552502AD; Wed, 26 Jul 2017 12:53:56 +0200 (CEST)
Nachricht-ID: <20170726105356.473552502AD@home.revamp-it.ch>

Cyrus IMAP for Debian, Simple Install Guide

"All systems administrators have their horror stories. For me, it was setting up a HP Color Bubblejet under Linux using ghostscript before linuxprinting.org was alive. Well that was a piece of cake compared to what I am about to describe in this document."

-- "Hosting email for virtual domains using Postfix and Cyrus"
Haim Dimermanas, 2001-08-01

"I warned you to read all the documentation first, didn't I?"
-- Henrique M. Holschuh, 2002-10-01

This document describes how to get Cyrus running with a simple configuration that you can then tweak to your real needs.

READ README.Debian AS WELL. I MEAN IT! Cyrus is easy, all the trouble is in SASL, and even that becomes easy after you understand how SASL works.

IMPORTANT: Cyrus is a closed-box email system. Your system will access your email through LMTP, IMAP and POP3 ***only***. No direct file access to the email store is supposed to take place.

To setup Cyrus so that you can administer it (i.e. create users), and get email inside it for those users:

1. Make sure libsasl2-modules, libsasl2 and sasl2-bin are installed
2. Make sure /etc/sasl2 is readable by group sasl. Pay attention to overrides (dpkg-statoverride)!
3. Make sure user cyrus belongs to group sasl (cyrus-common-2.x's install tries to do this automatically for you).
4. Edit /etc/cyrus.conf, and make sure the services you need are enabled. These are most probably "imap", "pop3", "lmtpunix".
5. Edit /etc/imapd.conf, and make sure you have some admin users listed in the entry "admins:". I suggest using "cyrus" as your admin.

I also suggest enabling plain text logins, and setting
sasl_minimum_layer: 0

If you have unixhierarchysep enabled in imapd.conf, change all

"." in mailbox names mentioned on this document to "/", since Cyrus will use "/" as the hierarchy separator instead of the default ".". I suggest you just leave unixhierarchysep set to false for now.

6. Restart Cyrus (/etc/init.d/cyrus-imapd restart)
7. Use saslpasswd2 -c to create an account for your admin:
saslpasswd2 -c cyrus
8. Use sasldblistusers2 to make sure step 7 worked fine.
9. Add other users to SASL likewise (saslpasswd2 -c).
10. Log in cyrus as the administrator, and create the mailboxes:
cyradm --user cyrus localhost
cm user.bob
cm user.anna
cm user.clark...
^D

(notice that there is an "user." in front of the mailbox name!)
You must use "user/bob", "user/anna" instead if you have the unixhierarchysep option enabled in imapd.conf.

For this to work, you obviously need the cyrus-admin-2.x package installed.

11. Try to login as a normal user, using imtest or a IMAP/POP3 client. If you have trouble with mutt and CRAM-MD5 or DIGEST-MD5, edit /etc/imapd.conf, and look for sasl_mech_list. Set it to:
sasl_mech_list: plain cram-md5
(this will disable digest-md5, which causes trouble with mutt).
12. Setup your MTA to deliver email inside Cyrus. Basically that can be done (easily) by:
 - a) running /usr/sbin/cyrdeliver (SLOW)
You need the lmtpunix service enabled in /etc/cyrus.conf for this to work.
 - b) delivering using LMTP to /var/run/cyrus/socket/lmtp
You need the lmtpunix service enabled in /etc/cyrus.conf for this to work.

Just make sure (and use dpkg-statoverride to do that) that your MTA can get to /var/run/cyrus/socket/lmtp. It works just like any file in a Unix system.

Cyrus REQUIRES a valid RFC2822 message, and will refuse messages with bad headers (such as that From foobar header, notice the missing ':'), embedded NULLs or any other crap.

That's it. See /usr/share/doc/cyrus-common-2.x/README.{postfix,exim,sendmail} for help on how to setup your MTA to correctly deliver to Cyrus.

LDAP SETUP -----

First, do the steps above and verify that your system is working fine.

SASL is perfectly capable of trying various authentication methods one after another. We will change our Cyrus setup for SASL to use a LDAP server lookup through saslauthd.

1. Create the configuration for saslauthd to know what it must do:

Write the following file to /etc/saslauthd.conf:

```
----CUTHERE----
ldap_servers: ldap://127.0.0.1/
ldap_version: 3
ldap_timeout: 10
ldap_time_limit: 10
ldap_cache_ttl: 30
ldap_cache_mem: 32768
ldap_scope: sub
ldap_search_base: ou=mail,o=mydomain
ldap_auth_method: bind
ldap_filter: maildrop=%u
----CUTHERE----
```

And of course, edit it to fit your LDAP setup.

2. Now, configure saslauthd to use LDAP mode and our config file:

Modify /etc/default/saslauthd so that it reads:

```
MECHANISMS="ldap"
PARAMS="-0 /etc/saslauthd.conf"
```

(MECHANISMS can be a space-separated list of authentication mechanisms. If you wanted saslauthd to try LDAP, then PAM, you could use MECHANISMS="ldap pam")

3. Start saslauthd

```
/etc/init.d/saslauthd restart
```

4. Make sure Cyrus will be able to talk to saslauthd

Set the following options in /etc/imapd.conf:

```
sasl_mech_list: PLAIN
allowapop: no
allowplaintext: yes
sasl_minimum_layer: 0
sasl_pwcheck_method: saslauthd
```

And restart Cyrus. You'd better understand that the above allows plaintext logins over the network. There is a LDAP SASL auxprop plugin being worked on that might fix this issue. As it stands right now, you're better off by only accepting IMAPS (secure IMAP) connections.

(sasl_pwcheck_method is a space separated list of SASL methods to try. If you want to have some local users in /etc/sasl2, for example, you could have "sasl_pwcheck_method: auxprop saslauthd" and also "sasl_auxprop_plugin: sasldb")

One ***extremely*** important point to notice is that saslauthd works **ONLY** with plaintext. APOP, CRAM-MD5, OTP, DIGEST-MD5 and any other "auxprop" SASL mech will ***not*** work through saslauthd.

5. That's it. There is a LDAP auxprop module in the works which can deal with APOP, CRAM-MD5, OTP, DIGEST-MD5 and so on, look for it in the SASL docs and openldap's contrib stuff.

```
-- Henrique de Moraes Holschuh hmh@debian.org
Cyrus IMAP for Debian
-----
```

"All systems administrators have their horror stories. For me, it was setting up a HP Color Bubblejet under Linux using ghostscript before linuxprinting.org was alive. Well that was a piece of cake compared to what I am about to describe in this document."

-- "Hosting email for virtual domains using Postfix and Cyrus"
Haim Dimermanas, 2001-08-01

"I warned you to read all the documentation first, didn't I?"
-- Henrique M. Holschuh, 2002-10-01

IMPORTANT: Cyrus is a closed-box email system. Your system will access your email through LMTP, IMAP and POP3 ***only***. No direct file access to the email store is supposed to take place.

For more information, please consult <http://cyrusimap.org/>. There is also Cyrus-HOWTO (Cyrus-IMAP.txt) available as part of the LDP HOWTO collection. Upgrade hints are in UPGRADE.Debian.gz Outdated documentation will cause you much grief, so beware of that when hunting anywhere else than the Cyrus mailinglist for information.

Backports of the latest packages for Debian Stable are available from <http://www.backports.org>

WARNING: For one to get Cyrus IMAPd to work correctly, one must first get the SASL layer to work correctly. This is far from trivial, so if you don't manage at first, don't go around filling bugs against Cyrus IMAPd before you make damn sure it is not a SASL configuration error. Read the hintlist later on this file as well. Start by reading README.Debian.simpleinstall.

The Debian packaging of Cyrus has a few quirks which are important to know about:

1. Renaming of some Cyrus IMAP utilities

The quota, reconstruct, master and deliver utilities have been renamed to cyrquota, cyrreconstruct, cyrmaster and cyrdeliver, following the template set by upstream with "cyradm". This was done because both Cyrus Debian maintainers found the original names to be too generic and likely to cause namespace collisions later.

Since documentation may refer to these utilities using their original name, you must be aware of this fact. Also, installsieve is deprecated and will be removed from the Debian package in the future; use sieveshell instead.

2. Relocation of many Cyrus IMAP files

The default Cyrus install scatters files all over the place. The Debian package installs only a few files in /usr/bin (cyradm, sieveshell). IMAP/email administrator utilities are installed in /usr/sbin (such as cyrreconstruct)[*]. Programs that must be run by cyrmaster are installed in /usr/lib/cyrus (such as imapd and pop3d). Sockets go into /var/run/cyrus/socket, per FHS 2.2. Sieve files go in /var/spool/sieve, but an /etc/sieve compatibility symlink is also installed just in case.

* - As of cyrus-imapd-2.4 (>= 2.4.6) there's a universal tool called /usr/sbin/cyrus which can be used to call all cyrus utilities. Invoke /usr/sbin/cyrus --help without any argument to learn more.

The imapd.conf and cyrus.conf configuration files are in /etc. The PAM policy files are in /etc/pam.d.

Feel free to use dpkg-statoverride to change the permission of

/var/run/cyrus/socket, the cyrus packages will not override your configuration if dpkg-statoverride is used. In fact, you will most likely have to do so for postfix to deliver to Cyrus, for example.

3. Cyrus Murder, the Cyrus IMAPd/POP3 aggregator is available.

However, you will have to configure it yourself. No pre-packaged configuration of Murder is available at this time... The documentation is all there, and the Cyrus packages will happily preserve your Cyrus Murder configuration. You do not have to install the cyrus-imapd or cyrus-pop3d packages in hosts that only need the proxy daemons running, but do note that the /etc/pam.d/imap and /etc/pam.d/pop files are in those packages (and they are needed by the proxies), so you will have to create the files manually.

One important note: MUPDATE doesn't support TLS, so you won't be able to use plaintext authentication methods. The easiest thing to do is to put an entry for your mupdate user in sasldb2 and use DIGEST-MD5.

4. Configurable idled support.

Cyrus IMAPd supports three options of using IDLE in IMAP sessions. The first option is not to support IDLE at all. The second is to use internal polling in the IMAP daemon. The third option is to use an external daemon, idled. Upstream only supports configuration of this during compilation, Debian however includes a patch which makes this runtime-configurable. Please set the 'idlemethod' imapd.conf option according to your needs and enable idled in cyrus.conf if you want to use it.

General notes and hints:

- o *** ALWAYS READ /usr/share/doc/cyrus-common/NEWS.Debian *** after you upgrade the package. This, and every other NEWS.Debian can automatically be shown to you before the upgrade, see the apt-listchanges package for more information.
- o QUOTAS ARE LIMITED TO 2GB on some platforms.
Be careful to not set quotas over that amount if your platform doesn't support the C datatype "long long". Things will break in very bad ways. Yes, it is a big glitch, and no, there are no easy workarounds.
see https://bugzilla.andrew.cmu.edu/show_bug.cgi?id=1212
This has been fixed for the upcoming Cyrus 2.4.
- o Either turn off logging of the DEBUG level, or don't complain about cyrus verbosity on the logs. Don't ever ask in the mailing lists about messages logged in the DEBUG level before reading the source code.
- o Watch out for your /dev/random bitbucket! SASL may use it, and if it empties, it will hang the processes wrapped up by SASL. This means just about every Cyrus service (lmtp, imap, pop3, sieve)... Disable APOP in /etc/imapd.conf if you don't need it, as it is a serious draw on randomness resources.
- o One extremely important point to notice is that saslauthd works ONLY with plaintext. APOP, CRAM-MD5, OTP, DIGEST-MD5 and any other "auxprop" SASL mech will ***not*** work through saslauthd. This can and will cause serious issues in Cyrus murder environments.
- o When using ext3, Cyrus really wants data=journal. However, up to kernel 2.4.20 there are dangerous bugs in that option, so you're better off not using that. xfs is faster and better for Cyrus, anyway. Please note that sarge was shipped with 2.4.27, and etch will not ship any 2.4 kernels anymore."

2.4 kernels are NOT shipped with Debian Etch.

- o nscd users: nscd is highly incompatible with ldap, and somewhat buggy otherwise. If you use nscd and Cyrus segfaults on you, try restarting nscd, or disabling it.
- o "The Debian libldap2 and cyrus-imapd packages are both compiled using the SASL library. If you use cyrus-imapd together with libnss-ldap, or saslauthd together with libpam-ldap, the resulting double calls to SASL library functions can trigger a double-free bug which may cause the calling process to crash. To avoid such a crash, you must recompile the libldap2 package --without-cyrus-sasl." -- <http://bugs.debian.org/145766> [!@#%!!! I didn't expect SASL 2.1 to still have this annoying problem]
- o The lmtpl service (allocated in Debian Woody to port 2003, and non-existent on Debian Sarge and Etch) is non-standard. It has no port officially allocated anywhere; it is usually run bound to the localhost interface, unless one needs it for clustering and high-availability scenarios. If you need it elsewhere, by all means move it -- you only need to edit /etc/services, or change the port for the lmtpl service in /etc/cyrus.conf.
- o The lmtpl service will only allow Cyrus lmtpl administrators to authenticate. Set them in /etc/imapd.conf.
- o Cyrus can now use two different namespaces (the standard one, where all subfolders are children of INBOX, and one where they are all in the same hierarchical level).

See /usr/share/doc/cyrus-doc/html/altnamespace.html for details. If you deal with a large population of winboze users, this option can save you some headaches.

- o One can also chose between netnews-style notation for folders (INBOX.subfolder), where the "." character is reserved to separate folders; or UNIX-style notation (INBOX/subfolder), where dots are allowed in names, and the slash separate folders (the "^" character is reserved in this mode).

See /usr/share/doc/cyrus-common/html/altnamespace.html for details.

- o When using SASL, do keep in mind that cyrus runs under user cyrus, and not root. It cannot read shadow files (unless you add the user cyrus to group shadow), or perform any root-only operations directly. You need to use the saslauthd (or, if available, auxpropd) mechanism to authenticate against root-only data. And that also means user cyrus must be able to talk to the unix socket saslauthd uses (which is controlled by SASL, not Cyrus IMAPd).
- o Any of the SASL configure options can be inserted in imapd.conf, just prefix it with "sasl_" (e.g.: sasl_mech_list: PLAIN). The list of SASL options is in /usr/share/doc/libsassl2/options.html.
- o The services are tcp-wrapped. Their hosts.allow/hosts.deny id is the service name in /etc/cyrus.conf. See hosts_access(5).
- o The PAM service names for use with SASL (via saslauthd) are: "imap", "sieve", "lmtpl", "pop", "mupdate".
- o You need to specify your admin users in /etc/imapd.conf before you can add mailboxes, or deliver through authenticated lmtpl. Do NOT use root. We suggest user cyrus, which is already used by the system for all things Cyrus IMAPd... but it need not be an existing user. As long as SASL will authenticate against it, it will work.

- o Do NOT read your admin user's email via IMAP (see the FAQ for details).
- o Don't export your mail store over NFS or AFS (read the FAQ for more info). You have been warned. You really want a journaled (as in journaling for the metadata), local filesystem for the store. Failing that, you need something with very strict and correct lock semantics, and full mmap support.
- o Ext2 is slow on very large directories (right now), and sync metadata writes enabled are a huge performance hit. If you need high IO throughput from Cyrus, you will need to use ext3, reiserfs, xfs or something similar.
- o You may want to enable/disable synchronous metadata writes to your mail store dirs (check /usr/share/doc/cyrus-doc/html/install.html for more info, in package cyrus-docs). The cyrus-makedirs script tries to do the right thing for ext2 and ext3 filesystems. Failure to correctly update the metadata in the right order can completely screw up your Cyrus store on a power-loss or another disk failure.
- o Try mounting the store and cyrus database filesystems with noatime for performance gains. Load-balance the store using multiple partitions on different physical devices for even better performance gains.
- o Cyrus IMAPd should be fed mail through LMTP. If at all possible, use the Unix socket for that -- it automatically authenticates as user postman and that will help wonders. cyrdeliver can also be used to inject mail, but it will simply open an LMTP socket to cyrus and deliver through that -- this is much slower than using LMTP directly. The UNIX socket is in /var/run/cyrus/socket/lmtp. Use dpkg-statoverride if you need to change the permissions of the socket directory.
- o You can use /usr/sbin/cyrus-makedirs to generate the needed directories for cyrus partitions. It is run automatically by the package postinst, and it knows to parse the /etc/imapd.conf file to verify if hash subdirectories are needed or not. It cannot detect what kind of hashing should be used yet. If you recompile the package with full hashing, change it.
- o Refer to cyrus-utils.sourceforge.net and the info-cyrus mailinglist for mailbox/imap to cyrus conversion scripts.
- o If you don't have pop3 or something else enabled by default in cyrus.conf, installed, disable it. Otherwise, Cyrus master will log warnings that the service could not be started.
- o If you want to run something that is not in /usr/lib/cyrus/bin in cyrus.conf, just use the full path in cyrus.conf (e.g.: cmd="/usr/sbin/squatter").
- o Sieveshell is really lacking on auth capabilities, and timsieved is quite strict on what auth capabilities it offers. So, pay attention to sasl_minimum_layer, and see bug #151295 for more details (<http://bugs.debian.org/151295>). Also, make sure you have the correct set of SASL2 modules installed in in your system.
- o uw-mailutils has some nice utilities to migrate mail stores from/to imap servers. You might find it quite useful to migrate a site to Cyrus.

Known bugs

Please see
<http://bugs.debian.org/cgi-bin/pkgreport.cgi?which=src&data=cyrus-imapd-2.4&archive=no&version=>

for a list of any known bugs.

SNMP logging

cyrmaster is an agentx SNMP subagent, and it can interface to a agentx SNMP master. It will export data at OID .1.3.6.1.4.1.3.6.1 (cyrusMasterMIB).

The ucd-snmp daemon (package snmpd) is NOT configured to work as agentx master agent by default -- you have to do that manually, by adding "master agentx" to the /etc/snmp/snmpd.conf file.

cyrmaster will register with the snmp agentx master when it is started, so if the snmp master is restarted after cyrmaster, it will not forward the snmp requests to cyrmaster anymore. Check your system for any cron scripts that might be restarting the snmp process if that happens.

See /usr/share/snmp/mib/CYRUS-MASTER-MIB.txt for more details.

Backing up for rainy days

Cyrus automatically checkpoints and backups some of its databases, using the `ctl_cyrusdb(8)` utility (EVENTS in /etc/cyrus.conf). It is supposed to be also capable of recovering automatically from these backups, and to attempt to do so at startup. However, `ctl_cyrusdb -r` is NOT FULLY IMPLEMENTED YET... you are on your own to recover from corrupt databases.

This recovery can be done using the db3 utilities, and even by smart usage of `cvt_cyrusdb(8)` and `ctl_mboxlist(8)`. The automatic backups are useful, too, even if they are not restored automatically.

The database backups are stored at /var/lib/cyrus/db.backup*, you may want to copy the files there to backup media in a cronjob, or something like that. You can kill the TLS cache database, as long as Cyrus is stopped when you do it. Loss of the delivery database is not very bad, it just means some users might get duplicated messages.

Cyrus does NOT backup the mail store automatically. To backup the mail store partitions, you must stop Cyrus and dump the entire partition to your backup media. The MH-like structure of the Cyrus store do make them suitable for incremental backups. Hot-backups of the store can be made, but you risk losing some non-critical metadata when the restore is done.

You can backup all Cyrus non-text databases to a flat text file format using the `cvt_cyrusdb` utility (and recover back from the flat text file format), but you should stop Cyrus first.

If you ever need to recover the mail store from backup, you should run `cyrreconstruct(8)` to rebuild the mailbox indexes.

A daily maintenance cronjob uses `ctl_mboxlist(8)` to dump the mailboxes database to /var/backup. That backup copy can be used as a last-resort copy if the hot backups become corrupted somehow.

Debian source package quirks

Patching is done using `dpatch`. Patches from upstream CVS are marked as such.

THANKS

Thanks go to the CMU crew for producing Cyrus IMAPd in the first place; Michael-John Turner ~mj@debian.org for maintaining the v1.5 branch and setting the groundstones for the v2.1 package; David Parker ~david@neongoat.com and David D. Kilzer ~ddkilzer@theracingworld.com for their huge help in getting the v2.1 packages out-of-the-door, and the upgrade from v1.5 guide; Fabian Fagerholm ~fabbe@paniq.net for stress testing the daemons, and useful feedback; Gilles Bouthenot ~gilles.bouthenot@fcomte.iufm.fr for good feedback; and Henrique de Moraes Holschuh for maintaining the 2.1 branch.

-- Ondřej Surý ~ondrej@debian.org, Tue, 29 Jul 2014 15:32:41 +0200

This is the README for the amavisd-milter(8).

NAME

amavisd-milter -- sendmail milter for amavisd-new

SYNOPSIS

```
amavisd-milter [-fhv] [-d debug-level] [-D delivery-care-of]
               [-m max-conns] [-M max-wait] [-p pidfile] [-P]
               [-q backlog] [-s socket] [-t timeout] [-S socket]
               [-T timeout] [-w directory]
```

DESCRIPTION

The amavisd-milter is a sendmail milter (mail filter) for amavisd-new 2.4.3 and above and sendmail 8.13 and above (limited support for 8.12 is provided).

Instead of older amavis-milter helper program, full amavisd-new functionality is available, including adding spam and virus information header fields, modifying Subject, adding address extensions and removing certain recipients from delivery while delivering the same message to the rest.

For more information you can visit amavisd-milter website:

<http://amavisd-milter.sourceforge.net/>

and SourceForge project:

<http://sourceforge.net/projects/amavisd-milter>

Options

The options are as follows:

-d debug-level

Set the debug level to debug-level. Debugging traces become more verbose as the debug level increases. Maximum is 9.

-D delivery-care-of

Set AM.PDP request attribute `delivery_care_of` to client (default) or server. When client method is used then amavisd-milter is responsible to forward the message to recipients. This method doesn't allow personalized header or body modification.

When server method is used then amavisd-new is responsible to forward the message to recipients and can provide personalized header and body modification. `$forward_method` in `amavisd.conf` must point to some place willing to accept mail without further checking in amavisd-new.

-f

Run amavisd-milter in the foreground (i.e. do not daemonize). Print debug messages to the terminal.

- h Print help page and exit.
- m max-conns
Maximum concurrent amavisd connections (default 0 - unlimited number of connections). It must agree with the \$max_servers entry in amavisd.conf.
- M max-wait
Maximum wait for connection to amavisd in seconds (default 300 = 5 minutes). It must be less than sending MTA timeout for a response to the final "." that terminates a message on sending MTA. sendmail has default value 1 hour, postfix 10 minutes and qmail 20 minutes. We suggest to use less than 10 minutes.
- p pidfile
Use this pid file (default /var/amavis/amavisd-milter.pid).
- P When amavisd-new fails mail will be passed through unchecked.
- q backlog
Sets the incoming socket backlog used by listen(2). If it is not set or set to zero, the operating system default is used.
- s socket
Communication socket between sendmail and amavisd-milter (default /var/amavis/amavisd-milter.sock). The protocol spoken over this socket is MILTER (Mail FILTER). It must agree with the INPUT_MAIL_FILTER entry in sendmail.mc

The socket should be in "proto:address" format:
 - o {unix|local}:/path/to/file - A named pipe.
 - o inet:port@{hostname|ip-address} - An IPV4 socket.
 - o inet6:port@{hostname|ip-address} - An IPV6 socket.
- S socket
Communication socket between amavisd-milter and amavisd-new (default /var/amavis/amavisd.sock). The protocol spoken over this socket is AM.PDP (AMavis Policy Delegation Protocol). It must agree with the \$unix_socketname entry in amavisd.conf.

The socket should be in "proto:address" format:
 - o {unix|local}:/path/to/file - A named pipe.
- t timeout
sendmail connection timeout in seconds (default 600 = 10 minutes). It must agree with the INPUT_MAIL_FILTER entry in sendmail.mc and must be greater than or equal to the amavisd-new connection timeout. When you use other milters (especially time-consuming), the timeout must be sufficient to process message in all milters.
- T timeout
amavisd-new connection timeout in seconds (default 600 = 10 minutes). This timeout must be sufficient for message processing in amavisd-new. It's usually a good idea to adjust them to the same value as sendmail connection timeout.
- v Report the version number and exit.
- w directory
Set working directory (default /var/amavis).

Limited support for sendmail 8.12

- o smfi_addheader() is used instead of smfi_insheader() for insheader and addheader AM.PDP responses. This works well with amavisd-new 2.4.3 or newer.
- o smfi_progress() isn't called when amavisd-milter wait for amavisd-new communication socket.
- o AM.PDP response quarantine isn't implemented.

FILES

/var/amavis/amavisd-milter.pid
The default process-id file.

/var/amavis/amavisd-milter.sock
The default sendmail communication socket.

/var/amavis/amavisd.sock
Th default amavisd-new communication socket.

/var/amavis
The default working directory.

POLICY BANK

When remote client is authenticated, amavisd-milter forward this information to amavisd-new through AM.PDP request attribute policy_bank:

SMTP_AUTH
Indicate that the remote client is authenticated.

SMTP_AUTH_<MECH>
Remote client authentication mechanism.

SMTP_AUTH_<MECH>_<BITS>
The number of bits used for the key of the symmetric cipher when authentication mechanism use it.

EXAMPLES

Configuring amavisd-new

In amavisd.conf file change protocol and socket settings to:

```
$protocol = "AM.PDP";           # Use AM.PDP protocol
$unix_socketname = "$MYHOME/amavisd.sock"; # Listen on Unix socket
### $inet_socket_port = 10024;   # Don't listen on TCP port
```

Then (re)start amavisd daemon.

Configuring sendmail

To the sendmail.mc file add the following entries:

```
define(`confMILTER_MACROS_ENVFROM',
      confMILTER_MACROS_ENVFROM`, r, b')
INPUT_MAIL_FILTER(`amavisd-milter',
      `S=local:/var/amavis/amavisd-milter.sock,
      F=T, T=S:10m;R:10m;E:10m')
```

Then rebuild your sendmail.cf file, install it (usually to /etc/mail/sendmail.cf) and (re)start sendmail daemon.

Running amavisd-milter

This example assume that amavisd-new is running as user amavis. It must agree with the entry \$daemon_user in amavisd.conf.

First create working directory:

```
mkdir /var/amavis/tmp
chmod 750 /var/amavis/tmp
```

```
chown amavis /var/amavis/tmp
```

Then start amavisd-milter as non-privileged user amavis:

```
su - amavis -c "amavisd-milter -w /var/amavis/tmp"
```

Limiting maximum concurrent connections to amavisd

To limit concurrent connections to 4 and fail after 10 minutes (10*60 secs) of waiting run amavisd-milter with this options:

```
su - amavis -c "amavisd-milter -w /var/amavis/tmp -m 4 -M 600"
```

Troubleshooting

For troubleshooting run amavisd-milter on the foreground and set debug level to appropriate level:

```
su - amavis -c "amavisd-milter -w /var/amavis/tmp -f -d level"
```

where debug levels are:

- 1 Not errors but unexpected states (connection abort etc).
- 2 Main states in message processing.
- 3 All amavisd-milter debug messages.
- 4-9 Milter communication debugging (smfi_setdbg 1-6).

SEE ALSO

<http://amavisd-milter.sourceforge.net>
<http://www.ijs.si/software/amavisd/>
<http://www.milter.org/developers>
<http://www.sendmail.org>

AUTHORS

This manual page was written by Petr Rehor <rx@rx.cz> and is based on Jerzy Sakol <jerzy.sakol@commgraf.pl> initial work.

BUGS

A community mailing lists are available at:

http://sourceforge.net/mail/?group_id=138169

Enhancements, requests and problem reports are welcome.

If you run into problems first check the users mailing list archive before asking questions on the list. It's highly likely somebody has already come across the same problem and it's been solved.

Setting up Cyrus IMAPd for Postfix (Debian)

=====
Cyrus works wonderfully well with Postfix, both in single-system stores (where Postfix runs in the same host as Cyrus IMAPd), and remote mail stores (where Cyrus IMAPd is in a different host than Postfix).

Delivery to Cyrus should always be done through direct LMTP from Postfix. It is far simpler and faster to do such delivery using Unix sockets, but Unix sockets are only an option for single-system stores. Cyrdeliver is just a stdio-to-LMTP proxy, and it slows down mail delivery greatly.

Cyrus requires LMTP deliveries to be authenticated. It assumes that any deliveries done through an Unix socket are trustable, and pre-authenticates

them as if coming from the "postman" (fictitious) user (but you can re-authenticate them as another lmtplib admin user, if you wish).

Deliveries done through TCP sockets are not limited to the same host, unlike the Unix socket ones, and can therefore be used in remote mail stores. However, Cyrus requires that the LMTP session be authenticated using one of the SASL mechs available to Cyrus (you can switch this off by giving a "-a" parameter to lmtplib in cyrus.conf, but that is unsafe since anyone can bypass any user authentication controls you might have on mail delivery, that way).

TCP-socket LMTP sessions should be authenticated as one of the Cyrus LMTP admins, normal Cyrus users are not enough. This requires Postfix with SASL support.

Setting up Postfix for LMTP delivery to Cyrus

=====

Just set up a transport (either using a transport map, or the default_transport configuration directive of Postfix). Do not use cyrdeliver.

I suggest that the lmtplib transport be duplicated and renamed to cyrus if you use it to talk to anything else (such as amavisd-new, or amavis-ng). That way, LMTP connection caching to the Cyrus store gets optimized, and you can use the lmtplib-named LMTP transport for something else.

WARNING: Postfix 2.0 does not downcase the recipient in LMTP deliveries, so if your users require it, you will have to set lmtplib_downcase_rcpt: yes in imapd.conf.

Unix sockets:

For Unix sockets, the Postfix transport is specified as "lmtplib:unix:/var/run/cyrus/socket/lmtplib", (we are using the default Cyrus unix socket location as an example, you can change it in /etc/cyrus.conf and /etc/imapd.conf).

You need a Cyrus lmtplib service listening on that socket, of course, so make sure something like:

```
lmtplibunix      cmd="lmtplib" listen="/var/run/cyrus/socket/lmtplib"
```

is in the SERVICES section of the /etc/cyrus.conf file. You also need to make sure both Cyrus and Postfix can talk through that socket. Unix sockets work just like files, so that translates to making sure both the user "cyrus" and the user Postfix is using for LMTP delivery can both read and write to that file.

WARNING: Since Cyrus pre-auths anything coming through the Unix socket, anyone who can write to it will be able to inject email into Cyrus directly.

Use dpkg-statoverride to make sure your configuration for the socket permissions will not be overwritten by the Cyrus packages. Do remember that Postfix usually runs the LMTP transport as user "postfix" (configurable in /etc/postfix/master.cf). Also, do not run the postfix lmtplib transport chrooted if the socket is not inside the chroot.

1. Create a lmtplib group:


```
# addgroup lmtplib
```
2. Put user postfix in that group:


```
# adduser postfix lmtplib
```

3. Fix the socket directory permissions:


```
# dpkg-statoverride --force --update --add \
  cyrus lmtpl 750 /var/run/cyrus/socket
```
4. Restart Postfix and Cyrus IMAPd


```
# /etc/init.d/postfix restart
# /etc/init.d/cyrus-imapd restart
```

TCP sockets:

TCP sockets are easier on the Cyrus side, and more complicated on the Postfix side. For Cyrus, it is enough to have an "lmtpl" service listening on the desired IP interface (or in all of them, if you leave the interface unspecified as in the example below), that means something like this in /etc/cyrus.conf SERVICES area:

```
lmtpl          cmd="lmtpl" listen="lmtpl"
```

(do note that you MUST have an lmtpl entry in /etc/services for this to work. This can be any available port, for historical reasons, 2003 is most often used).

Also, remember that Cyrus observes tcpwrapper permissions (/etc/hosts.allow and /etc/hosts.deny), make sure they're set up correctly or Cyrus might refuse the connections.

To configure Postfix' lmtpl transport to authenticate using SASL, do the following:

Note: On Sarge systems, you will also need the postfix-tls package for SASL support. On etch and above, the support is included in the postfix package.

1. Configure the lmtpl transport SASL layer:


```
(add to /etc/postfix/main.cf):
lmtpl_sasl_auth_enable = yes
lmtpl_sasl_password_maps = hash:/etc/postfix/sasl_passwd
lmtpl_sasl_security_options =
lmtpl_destination_concurrency_limit = 100
lmtpl_destination_recipient_limit = 0
```

The *_limit values should match whatever you configured as limits in Cyrus. The above configuration will allow plain text logins.

Create the password map /etc/postfix/sasl_passwd to tell postfix of a Cyrus LMTPL administrator user and password to use.

e.g.:
echo "mycyruspool.my.domain.org postman:foobar" >sasl_passwd
postmap sasl_passwd

2. Configure Cyrus to accept that user as a lmtpl administrator


```
(add to /etc/imapd.conf)
lmtpl_admins: postman
```
3. Tell postfix to use the lmtpl transport to deliver email using transport maps or something else. I suggest making a copy of the postfix lmtpl transport in master.cf, renaming it to "cyrus", and using that.
4. Note that to use the virtual domains feature in Cyrus, you need to tell postfix to use the lmtpl transport as the virtual_transport. If you also want "local" mail recipients, i.e. those that are addressed to hosts listed in mydestination, to be delivered to

Cyrus, you also need to use lmtpl as the mailbox_transport.

That's it!

To build this package, configure was called as follows:

```
configure '--build=x86_64-linux-gnu' '--prefix=/usr' '--includedir=\${prefix}/include'
'--mandir=\${prefix}/share/man' '--infodir=\${prefix}/share/info'
'--localstatedir=/var' '--libdir=\${prefix}/lib/x86_64-linux-gnu'
'--libexecdir=\${prefix}/lib/x86_64-linux-gnu' '--disable-maintainer-mode' '--disable-
dependency-tracking' '--build' 'x86_64-linux-gnu' '--with-extraident=Debian-
2.4.17+nocaldav-0+deb8u2revamp1' '--prefix=/usr/share' '--exec-prefix=/usr'
'--libexecdir=/usr/sbin' '--bindir=/usr/sbin' '--sbindir=/usr/sbin' '--includedir=/usr
/include/cyrus' '--datadir=/usr/share/cyrus' '--sysconfdir=/etc' '--sharedstatedir=/usr
/share/cyrus' '--localstatedir=/var/lib/cyrus' '--mandir=/usr/share/man' '--with-cyrus-
prefix=/usr/lib/cyrus' '--enable-gssapi' '--with-gss_impl=mit' '--with-ldap=/usr'
'--with-lock=fcntl' '--with-perl=/usr/bin/perl' '--with-openssl=/usr' '--with-bdb=db'
'--with-bdb-libdir=/usr/lib' '--with-bdb-incdir=/usr/include' '--enable-murder'
'--enable-nntp' '--enable-replica
tion' '--disable-listtext' '--with-sasl=/usr' '--enable-idled' '--enable-http' '--with-
opendkim-libdir=/usr/include/opendkim' '--with-opendkim-incdir=/usr/lib' '--with-cyrus-
user=cyrus' '--with-cyrus-group=mail' '--with-com_err=' '--with-pidfile=/var
/run/cyrmaster.pid' '--with-syslogfacility=MAIL' '--with-ucdsnmp=/usr'
'build_alias=x86_64-linux-gnu' 'CFLAGS=-g -O2 -fPIE -fstack-protector-strong -Wformat
-Werror=format-security -Wall -Wextra -g -fno-strict-aliasing -pipe -O2' 'LDFLAGS=-fPIE
-pie -Wl,-z,relro -Wl,-z,now -Wl,-z,defs -Wl,--as-needed' 'CPPFLAGS=-
D_FORTIFY_SOURCE=2'
```