Mailadm explained

The official

Courier Mail Server User Administration

documentation

Jarle (jgaa) Aase
Jgaa's Internet

jgaa@jgaa.com

Mailadm explained: The official Courier Mail Server User Administration documentation

by Jarle (jgaa) Aase

Published v0.01.01 2002/06/17

This manual is produced using DocBook (http://www.docbook.org/). The original manuscript is included in the Mailadm source code distribution.

Table of Contents

List of Tables

About this Manual

1. Purpose / Scope of this Document

This is the official documentation on *the Courier Mail Server User Administration*. It is primarily targeted against system administrators.

2. Copyrights and Trademarks

Chapter 1. Introduction to Mailadm

1.1. Background

The Courier mailserver is an excellent piece of software. It is designed for speed and reliability. But it does also provide lot's of advanced features. In the spring 2002, I decided to move my main mailserver from sendmail/qpopper + some proprietary software I once wrote to syncronize a Microsoft Access database with Unix shell user accounts, and sendmail virtusers and aliases. The mailserver served some 20 companies and a few hundred mail accounts. The growing use of huge email-attachments caused more and more problems with the old mailserver. I looked at some of the alternatives, and choosed Courier. I then searched for a simple web-frontend to manage users and mail-accounts - with no luck. There was a few projects at Sourceforge, but none which had released any files. So I started my own project.

The aim was to make a system that is simple to use, that ultilize the MySQL-features in Courier and that would make migration from the old mailserver as simple (read cheap) as possible. I ended up with a simple PhP-based web-application that manipulates a MySQL database that is very similar to the sample database from the Courier documentation. I had to add a few columns to handle a new dimension for companies, and a secondary login-name. Courier uses user@domain as both email-address and pop/imap account name. The old mailserver used Unix shell accounts as pop account names. Since it would have been expensive to change a few hundred email setups on client computers, I wanted to use the old pop-names on the new server. With this design, we could continue to use the old approach with email-addresses organized by company - and the old pop-names. The web-application handles everything from managing companies, email-addresses and aliases, to creating Maildirs on the server and dynamically configure Courier with the domains that are in use. We also allow third-party programs to update the database, and creates Maildirs and configures Courier based on the data that was added. (This was how we transferred the account-information from the old Access database to MySQL).

In order to handle privileged operations (like special Courier maintainance commands)

- I wrote a small c program with root-privileges that is called from the PhP scripts. I've tried to make this program as simple and secure as possible to avoid exploits.

1.2. Platforms

The following platforms are supported:

- Linux
- Other Unix/Posix variants (may need some porting efforts)

1.3. Licensing

The program is released under the Gnu Public License (http://www.gnu.org/copyleft/fdl.html) (GPL). The program and source code is in other words free. There is no fee, no user-registration, no royalties, no nothing.

1.4. Download

The latest version of Mailadm can be downloaded from sourceforge.net/projects/courier-mailadm/ (http://sourceforge.net/projects/courier-mailadm/) The files can of course also be downloaded from ftp.jgaa.com (ftp://ftp.jgaa.com).

1.5. Obtaining support and reporting bugs

Mailadm is supported at support.jgaa.com (http://support.jgaa.com/). This site has a modern bug-reporting and bug-tracking facility, and you can also suggest features there, and subscribe to mailing lists. There is also a newsgroup: alt.comp.jgaa where you can ask for support.

If you have discovered a bug that can compromise the security in Mailadm please contact <`jgaa@jgaa.com`> directly. If you don't get a reply within 12 hours, please resend the message.

1.6. Author

Mailadm is written by Jarle (jgaa) Aase, best known for the original free FTP server for windows, War FTP Daemon. You can visit my homepage at www.jgaa.com (http://www.jgaa.com/).

Chapter 2. Compilation and installation

2.1. Installation under Linux with Apache and MySQL installed

Download the latest tarball and unpack it in a loction of your choise. I use Debian GNU/Linux, and have all my web-locations under /var/www. In my case the installation would be something like:

```
$ cd /var/www
$ tar -xvzf /tmp/courier-mailadm.latest.tar.gz
$ cd mailadm/src
$ make
$ su
# make install
```

> **Note:** You must not put the package in a location that is accessible by users of your webserver. If /var/www is your DocumentRoot in Apache, you must use another location!

When the package is unpacked, you must compile the utilty in `./mailadm/src`, su to root and run **make** `install`. Then you must edit two configuration files, `/etc/mailadmutil.conf` and `./mailadm/www/setup/config.php`. You must also modify the configuration file for the MySQL based authentication in Courier, `/etc/courier/authmysqlrc`.

`/etc/mailadmutil.conf` is read by the privileged program **/usr/local/bin/mailadmutil**. It must be owned by root, and only root must have write privileges to this file. Below is an example of this file.

```
# mailadmutil.conf
```

```
#
# sample configuration file for Debian GNU/Linux
#
# Created by Jarle (jgaa) Aase 2002



# Root for the virtual mailusers
#
virtuser-root = /var/lib/courier/spool-virtusers

# Full path to the courier makealiases program
#
makealiases-path = /usr/sbin/makealiases

# Full path to the courier makeacceptmailfor program
#
makeacceptmailfor-path = /usr/sbin/makeacceptmailfor

# Full path to the courier makesmtpaccess program
#
makesmtpaccess-path = /usr/sbin/makesmtpaccess

# Full path to the courier makehosteddomains program
#
makehosteddomains-path = /usr/sbin/makehosteddomains

# Full path to a command to the courier control program
courier-path = /usr/sbin/courier

# Uncomment below to set the courier servers user and group
# If these are not set, the program will look for a "courier"
# user and group, and then a "daemon" user and group.
#
courier-user = daemon
courier-group = daemon
```

```
# Uncomment below to allow deletion of mailspool-dirs.
# This should only be allowed on secure, dedicated
# mailservers with no normal/untrusted users, as
# any logged in user, or user with access to the
# file system will be able to delete any mailspool dir.
# On dedicated servers this option will ease maintainance
# a bit, as users can be 100% maintained from the
# web-interface
#
allow-delete = yes

# This is the uid (userid) that is shared for all the
# virtual mail users. The uid must correspond to an
# actual user-id in the /etc/passwd file. This can
# be a fake user like "noone", or a user without
# login privileges that is created for this purpose.
# The ID must also be used as default for the uid field
# in the mail.passwd mysql table.
#
mailbox-uid = 65534

# This is the gid that is shared for all virtual
# mail users. The gid must correspond with an
# actual group id in the /etc/group file. This
# can be a real group created for this purpose, or
# a fake group like "nogroup". The gid must
# correspond with the default value for gid in the
# mail.passwd mysql table.
#
mailbox-gid = 65534
```

`./mailadm/www/setup/config.php` contains information needed by the PhP scripts. It contains the password to the MySQL database - so access should be limited to the www-data user (provided that this is the user-account that PhP runs under). This does not make the content 100% safe - but there are really no ways to secure passwords needed ny PhP scripts - at least not that I'm aware of. Below is an example of this file:

```
<?
// NOTE: This file contain a password in clear text
// and must be read-only to the owner and the group
// www-data

global $db, $DbName, $DbHost, $DbPasswd, $DbUser,
$courier_aliases_file, $courier_domains_file,
$mailadmutil, $virtuser_root;

$DbName = "mail"; // Name of database
$DbHost = "localhost"; // Host for mysql server
$DbUser = "mail"; // User to log on rto mysql as
$DbPasswd = "secret"; // Password to log on with

// Path to the root of the vir-
tual user dirs (which in turn cointains
// the maildirs. This setting must correspond with the setting
// in /etc/mailadmutil.conf
$virtuser_root = "/var/lib/courier/spool-virtusers";

// Path to the mailadmutil helper-program
// This is suid-root program that can run
// privileged commands and create/delete
// mailspool directories with the correct
// ownership and permissions.
$mailadmutil = "/usr/local/bin/mailadmutil";

// Name of a file we can use in the courier
// alieses directory to store our default domains.
$courier_aliases_file = "/etc/courier/aliases/mailadm_generated";

// Name to the file containing the hosted domains.
// We extract these from the database.
$courier_domains_file = "/etc/courier/hosteddomains/mailadm_generated";
```

```
?>
```

The Apache server must also be configured. I will suggest to use SSL on this virtual host. If you don't have SSL configured on your Apache server, just disregard the SSL* lines in the sample below and use :80 in stead of :443.

```
<VirtualHost myip:443>
        ServerName mailadm.jgaa.com
        DocumentRoot /var/www/mailadm/www
        SSLEngine on
        SSLProtocol all
        SSLCertificateFile /etc/apache/ssl.crt/server.crt
        SSLCertificateKeyFile /etc/apache/ssl.key/server.key
        SSLVerifyClient none
        SSLLog /var/log/apache/modssl.log
        SSLLogLevel warn
</VirtualHost>
```

Remember to restart Apache when you have changed the configuration.

The last thing to do is to change the permissions on two directories used by the courier server. The reason for this is that PhP needs to write textfiles with domains and aliases in order to dynamically reconfigure Courier. On my Debian installation, these paths are `/etc/courier/hosteddomains` and `/etc/courier/aliases`

```
drwxrwx---    2 daemon   www-data  aliases
drwxrwxr-x    2 daemon   www-data  hosteddomains
```

> **Note:** These permissions must be set each time you upgrade the Courier mailserver, as the upgrade procedure resets the ownership and modes.

`/etc/courier/authmysqlrc` contains information needed by the MySQL based authentication module in Courier I have modified the SQL query to look for both the old pop-name and the new account-name. The query also checks if an account is disabled. An example file is present in `./mailadm/mysql`. Below is an example of this file:

```
MYSQL_SERVER    localhost
MYSQL_USERNAME  mail
MYSQL_PASSWORD  secret
MYSQL_PORT      0
MYSQL_OPT       0
MYSQL_DATABASE  mail
MYSQL_USER_TABLE        passwd
MYSQL_CLEAR_PWFIELD     clear
DEFAULT_DOMAIN  ldp.no
MYSQL_UID_FIELD uid
MYSQL_GID_FIELD gid
MYSQL_LOGIN_FIELD       id
MYSQL_HOME_FIELD        home
MYSQL_NAME_FIELD        name
MYSQL_MAILDIR_FIELD     maildir
MYSQL_QUOTA_FIELD       quota
MYSQL_WHERE_CLAUSE      active = ’yes’

MYSQL_SELECT_CLAUSE     \
SELECT P.id, P.crypt, P.clear, P.uid, P.gid, \
P.home, P.maildir, P.quota, P.name \
    FROM mail.passwd AS P, mail.organization AS O \
    WHERE P.organization_id = O.organization_id \
    AND (P.id = CONCAT(’$(local_part)’, ’@’, ’$(domain)’) \
    OR P.old_popname = ’$(local_part)’) \
    AND O.hosted_here = ’yes’ \
    AND P.active = ’yes’
```

The last thing to do is to create the actual database. I assume that mysql is installed and operational. Create a database named "mail", and run the script to create the required tables: **$ mysql** `-p mail < ./mailadm/mysql/create_maildb.sql`.

When all the configuration steps are done, run **courier** `restart` to reload the configuration data.

Chapter 3. Using Mailadm

Point your web-browser to the address you assigned to the site and start using Mailadm.

3.1. Organizations

Each mail-account must belong to an organization. Typically, you will have one "Organization" for each domain you handle. When you create a new organization, you also specify a domain-name and which mail-servers the organization will use. The domain-name is used to dynamically configure Courier to use this domain. The mailservers are just informative. If you handle mail-accounts for different companies, it is useful to have this information when you configure email-clients. If you configure Microsoft Outlook clients, Mailadm will use this information to create handy `.ins` files that allow you to just clik once to do the entire configuration. In you uncheck the "Hosted Here" checkbox when editing an organization, the organization will not be activated in Courier This can be done to add information to pending domains, or to use Mailadm to maintain information about mail-addresses on other servers (just like a simple database).

3.2. Mailboxes and aliases

When you click on the mailbox-icon on an organization, you get a listing of all of it's mailboxes, and the choise to make new ones. When you edit mailboxes, you can enter the following informtion:

Table 3-1. Mailbox options

Users Real Name	Informative field.

Primary mail-address	This is both an email address, and a mail-account name used for pop’ing or accessing the mailbox from webmail or imap. If you don’t specify a domain, Mailadm will add the default domain for the current organization.
Aliases	Any number of mail-aliases for this mailbox. The aliases can be in the same domain, or in other domains. Mailadm will not dynamically configure Courier to use alian domains used here - so if you use domains not used for any of your organizations, you must add the domain to Courier mnanually. If you don’t specify a domain name, Mailadm will add the default domain for the current organization.
Quota	Please consult the Courier documentation.
Old pop-account	This option is designed to allow migration from other mailservers that use name-convensions other than Courier. You will normally not enter anything here for new mailboxes.
Password	Password for the mailbox. If you leave this field blank, Mailadm will create a moderately secure password. If you don’t check the "Encrypt" checkbox, the password will be stored in clear text. This is useful for the helpdesk in companies where the users need to get their passwords all the time.

Active	If not checked, the mailbox is ignored by Courier
Path	This is an informative (read-only) field that simply tells you where the Maildir for the user is on the server.

3.3. Resynch Aliases and Domains

The main menu item "Resynch Aliases and Domains" will dynamically reconfigure Courier with the domains and aliases in the database. Missing Maildirs will also be created. This is useful if you use third-party software to add mailbosex and/or organizations to the SQL database. You must run this command whenever you have made changes in domains or aliases.