Courier Patch for automatic maildir creation

Carlo Contavalli <ccontavalli at commedia.it> $Revision: 1.12 $ – $Date: 2002/10/31 23:44:22 $

Contents

1 License, copyright and...

2 What is authmkhome?

authmkhome is an authentication module for courier able to create home directories when the users first log in with the pop3 or imap protocol.

It relies on a script created by the system administrator and is thus suitable for most configurations.

In the past few months, authmkhome has been available as a patch for courier. Since it is much more easier to compile an authentication module than the whole courier and since a module is something you can easily plug in an already working system without troubles, the patch is no longer maintained and will be removed on January 13th 2003 from my site.

This module was developed in less than one evening moving stuff from the old patch. Although it is a very simple bounce of ansi C lines, bugs may exist.

However, authmkhome has been used in production environments with mysql without problems, while it has never been tested with ldap or any other authentication method (any feedback is welcome). Well, from the module perspective it doesn't make any difference what method you used to authenticate the user... there shouldn't be any problem anyway.

The latest version of this document and of authmkhome is available at `http://www.commedia.it/ccontavalli/`.

The module was meant to be used with courier pop3d or imapd and I have no clue on using it with courier-mta (is this possible at all? does this make any sense?). I personally use postfix with the virtual-agent and I'm quite happy with that, although I'd like to move to postfix+maildrop.

If you have troubles/suggestions/corrections feel free to mail me at <**ccontavalli at commedia.it**>.

3 Installation

3.1 From scratch

Well, the code is ansi C, it is quite simple, and as long as you have a POSIX operative system and an ansi C compiler with make, it should be as simple as

```
# make
```

However, you must copy the resulting binary (authmkhome) in the right directory by hand. The right directory is that used by courier to store other authentication modules. You may find it using "locate" or trying to remember the parameters you used to configure courier. On my system, the "right" directory is /usr/lib/courier/authlib. Another possibility is to use "courier-configure" and append a "/authlib" to the "libexecdir" variable.

As you probably already know, there is no configure script since authmkhome is just a few lines, and it would take more effort to write a configure.ac file than to rewrite authmkhome in another language. The only configurable parameter is "STD_CREATOR". It indicates the path of the default script to execute to create home directories and it is set by default to "/usr/sbin/authmkhome-creator". You can change it by modifying authmkhome.h or by calling make with something like:

```
# DEFS='-DSTD_CREATOR=\"/fuffa-path/fuffa-creator.sh\"' make
```

where "/fuffa_path/fuffa-creator.sh" is the full path of a creator script.

3.2 With debian

For your convenience, debian binary and source packages have been provided. Just download them and run something like:

```
# dpkg -i courier-authmkhome.deb
```

and you should almost be done.

4 Usage

Now that you have successfully compiled and installed authmkhome, you need to

- tell courier you want to use it
- tell authmkhome how to create home directories

There's one important thing you must be aware of before configuring authmkhome: courier uses a list of modules to authenticate the user. The first module that successfully authenticates the user, enters the home directory of the user. Once there, the authentication goes on and authmkhome is called. However, if the first module fails to change the directory, the authentication will immediately stop, without a chance for authmkhome to create the needed directories. There are two solutions to this problem:

1. You set in your database a generic home directory (like /home) and you let your script change to the correct directory.
2. You use authmkhome just to create the "Maildir" subdirectory under the home of the user.

The first method is not quite usable if you have shell users and you use the standard /etc/passwd file to retrieve authentication data for courier. However, it should be trivial to correctly set up authmkhome in any other case and the second method should be more than enough if you have shell users.

4.1 Talking to courier

So, let's start from the first step...

Before anything else, you should open courier-imap and courier-pop3 configuration files (on my system, `/etc/courier/imapd` and `/etc/courier/pop3d`).

Look for a line like

```
AUTHMODULES="authdaemon"
  or
AUTHMODULES="any_fancy_authentication_module"
```

and change it in

```
AUTHMODULES="authdaemon authmkhome"
  or
AUTHMODULES="any_fancy_authentication_module authmkhome"
```

Beware! authmkhome relies on other modules authenticating the users. Thus, it must be called as the last authentication module (unless you want to see funny things happening to your system).

Watch out that once courier is configured to use authmkhome, any authenticated user whose home directory does not exist won't be allowed in in case authmkhome can't find a valid maildir-creator script.

4.2 Talking to authmkhome

By default, authmkhome will look for a script in `/usr/sbin/authmkhome-creator`. It will call this script with the name of the user as the first argument and the directory to create as the second argument. Many other parameters may be available through the environment but you shouldn't rely on them too much since they may change from version to version of courier.

Anyway, you can specify a different homedir creator with the parameter "MAILDIR_CREATOR". You can put it in any of courier-imap or pop3 configuration files, as long as the script is executable. Actually, if you put this parameter in pop3d you should modify the init script in order to force the variable to be exported. In this case, you may want to modify your /etc/init.d/courier-pop3 to look like:

```
/usr/bin/env - MAILDIR_CREATOR="$MAILDIR_CREATOR" PATH="...
```

while the original should be quite similar to:

```
/usr/bin/env - PATH="$PATH" SHELL="$SHELL" POP3AUTH="$POP3AUTH" \
     $TCPD -pid=$PIDFILE -stderrlogger=${sbindir}/courierlogger \
                         -maxprocs=$MAXDAEMONS -maxperip=$MAXPERIP \
                                  $TCPDOPTS -address=$ADDRESS $PORT \
 ${prefix}/lib/courier/courier/courierpop3login $AUTHMODULELIST \
                 ${prefix}/lib/courier/courier/courierpop3d Maildir
```

If you want to, although useless in most cases, you can also specify two different creators, one for the pop3 daemon and one for the imap daemon. Just put a different "MAILDIR_CREATOR" in the correct configuration files. Make sure to read the following sections on how to write a creator since it can be quite tricky.

4.3 Writing the creator

Beware! The creator script is called with a simple exec. Thus, they cannot be "inlined" bash scripts. Example:

```
THIS IS BAD: MAILDIR_CREATOR="mkdir $(echo 'SELECT * FROM ...'|cut -f);
chmod..."
```

There are few things to keep in mind when writing the mailcreator script:

- Unless you are using a suid script (with some kind of wraparound), the script will have the user's privileges. Starting from this assumption, I would suggest you setup the home directories to be owned by the recipient and by a generic mailgroup. Thus, you can create a `/home/mail` owned by root with privileges 0770 (rwxrwx—) where every user is part of the mailgroup and owns is own directory, set with permissions similar to 0700, leading to something like

  ```
  drwxrwx--- root mailgrp /home/mail
  drwx------ usr1 mailgrp /home/mail/usr1
  drwx------ usr2 mailgrp /home/mail/usr2
  ```

  Using this scheme, no user would be able to read somebody else mails, no user would be able to remove anybody else maildirs, however, any mailgrp user could be able to create any number of directories inside `/home/mail` without giving the right to courier to write in there leading to a denial of service. This method is thus suggested to those of you who don't give shell accounts to their mail users.Other better solutions probably exist, but keep in mind that authmkhome runs with user privileges.

- Beware that the username passed to the creator is the name used to authenticate the user with pop3 or imap. Thus, in most cases, it won't be a valid system username. Use the UID environment variable instead.
- The maildir creator script can take a lot of information from the environment. However, the content of the environment may change from version to version of courier. Before you use those variables, I suggest you put some line like the following in your script, just to make sure that the variable you want to use exists:

```
set >> /tmp/state.log
```

Here is an incomplete list of variables available in courier-0.36.0 and their values (most of them are just crap from our point of view):

```
ADDRESS=0
AUTHADDR=ccontavalli@localhost  # Mail address of the logged in user
AUTHARGC=4                      # See man authlib
AUTHARGV0=/usr/lib/courier/courier/imaplogin
AUTHARGV1=/usr/lib/courier/authlib/authdaemon
AUTHARGV2=/usr/bin/imapd
AUTHARGV3=Maildir
AUTHENTICATED=ccontavalli@localhost # Username
AUTHEXPIRE=1009760251
AUTHFULLNAME='Carlo Contavalli' # Full name of the user (if provided by the db)
AUTHMODULES=authdaemon
AUTHMODULES_ORIG=authdaemon
AUTHUSER=/usr/lib/courier/courier/imaplogin
EUID=1051                       # Effective user id of the process
                                # (provided by your system)
GROUPS=()                       # Additional groups (provided by your system)
HOSTNAME=caronte                # Hostname (provided by your system)
IMAPDSTART=YES
IMAPLOGINTAG=001
IMAP_CAPABILITY='IMAP4rev1 CHILDREN NAMESPACE \
        THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT'
IMAP_CAPABILITY_ORIG='IMAP4rev1 CHILDREN NAMESPACE \
        THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT \
        AUTH=CRAM-MD5 AUTH=CRAM-SHA1 IDLE'
IMAP_CAPABILITY_TLS='IMAP4rev1 CHILDREN NAMESPACE \
        THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT AUTH=PLAIN'
IMAP_CAPABILITY_TLS_ORIG='IMAP4rev1 CHILDREN NAMESPACE \
        THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT AUTH=CRAM-MD5 \
        AUTH=CRAM-SHA1 IDLE AUTH=PLAIN'
IMAP_CHECK_ALL_FOLDERS=0
IMAP_DISABLETHREADSORT=0
IMAP_EMPTYTRASH=Trash:7
IMAP_IDLE_TIMEOUT=60
IMAP_MOVE_EXPUNGE_TO_TRASH=0
IMAP_OBSOLETE_CLIENT=0
IMAP_STARTTLS=NO
IMAP_ULIMITD=65536
IMAP_USELOCKS=0
MAILDIR=1051/
MAXDAEMONS=40
MAXPERIP=4
OPTERR=1
OPTIND=1
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

```
PIDFILE=/var/run/courier/imapd.pid
PIPESTATUS=([0]="0")
PORT=143
PPID=668
TCPDOPTS='-nodnslookup -noidentlookup'
TCPLOCALIP=::ffff:127.0.0.1
TCPLOCALPORT=143
TCPREMOTEIP=::ffff:127.0.0.1
TCPREMOTEPORT=1030
UID=1051
```

Finally, here is an example of maildir creator that uses the provided environment variables and the suggested scheme of ownerships and rights:

```
#!/bin/bash

username=$1
maildir=$2
maildirmake /home/mail/$maildir
chown -R $UID:mailgrp /home/mail/$maildir

logger -p auth.notice -t courier Automagically created homedir "$maildir"\
         for uid "$UID" aka "$username".
```