

Courier Auth Postfix

Carlo Contavalli <ccontavalli at commedia.it>

Sat May 4 17:36:36 CEST 2002

This file documents authpfix, a couple programs written to allow only users authenticated with courier to relay mail on a postfix server (aka pop-before-smtp). This manual is Copyright © Carlo Contavalli 2001-2002. Please read the following sections for more details on the copyright of the programs provided. **Note that this is free software and authors hold no responsibility for any damage or loss, direct or indirect, caused by using this software. Use it only on your OWN risk and AFTER carefully reading this document.** The latest version of this document can be found at <http://www.commedia.it/ccontavalli/>.

Contents

1	License, copyright and...	1
2	About courier-auth-postfix...	2
2.1	How it works	2
2.2	Known problems	2
3	Installation	3
3.1	From scratch	3
3.2	With the Debian packages	4
3.3	Building new Debian packages	5
4	More details	5
4.1	Persistent entries	5
5	Optimizers	6
5.1	MySql optimizer	6
5.1.1	Table format	6
6	Testing	6
6.1	Checking your db	6
6.2	Troubleshooting	6

1 License, copyright and...

First of all:

- **Courier** is copyright © 1998-2001 Double Precision, Inc. and was created by Sam Varshavchik <mrsam@courier-mta.com>.

- **Postfix** is copyright © 1999 International Business Machines Corporation and others and it was created by Wietse Venema <wietse@porcupine.org>.

courier-auth-postfix (the set of programs documented in this paper) and this paper are copyright © 2001-2002 Carlo Contavalli and can be distributed under the terms of the GNU General Public License.

The mysql optimizer of courier-auth-postfix is a modified version of the dict module that provides mysql support in postfix, and was originally written by Scott Cotton <scott@icgroup.com> and Joshua Marcus <josh@icgroup.com>. Refer to the postfix documentation for more details about the license of this module.

The Debian package for courier-auth-postfix was obtained modifying the original debian package for postfix available on the main debian archives on www.debian.org and maintained by LaMont Jones <lamont@debian.org>.

Note that courier-auth-postfix is free software and authors hold no responsibility for any damage or loss, direct or indirect, caused by using this software. Use it only on your OWN risk and AFTER carefully reading this document.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts and no Back-Cover Texts.

Any example of program code available in this document should be considered protected by the terms of the GNU General Public License.

courier-auth-postfix is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

courier-auth-postfix is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Trademarks are owned by their respective owners.

2 About courier-auth-postfix...

courier-auth-postfix allows administrators to restrict relaying only to those users that successfully accessed their mailbox either with pop3 or imap (that successfully authenticated against courier-imap or courier-pop3, aka pop-before-smtp).

It was meant for those internet providers that do not want to leave their smtp server as “open-relay” but do not want to use smtp user authentication and do not want to set restrictions based on IP addresses.

courier-auth-postfix is provided as a patch for postfix, but, strictly speaking, it is not a real patch. It is a standalone module that is provided as a patch just to simplify the task of compiling it. Postfix nor courier would be modified by this patch and it would be possible to compile everything independently. However, no support is provided in this manual (not too hard anyway – good luck). From courier point of view, you will end up with an additional authentication module that can be used exactly like any other authentication module (help will be provided in the following sections).

In order to optimize the usage of some kind of databases and in order to provide additional features, courier-auth-postfix supports “optimizers”. An optimizer is just a small module compiled in courier-auth-postfix able to handle requests more efficiently. Actually, the only optimizer provided adds complete support for MySQL (a delete query is much faster than scanning all records).

Once compiled, in order to use courier-authpostfix, you need to modify just a couple of lines in main.cf and in courier configuration files (mainly pop3d and imapd).

You can find the latest version of this document and the mentioned software at <http://www.commedia.it/ccontavalli/>. If you have troubles/suggestions/corrections feel free to mail me at <ccontavalli at commedia.it>.

2.1 How it works

When a user logs in to check his emails on pop3 or imap his IP address and login time is added to a postfix-style database using the same libraries as used by postfix (for example, you can have a mysql:/etc/postfix/mysql.conf database). Postfix is then instructed to allow relaying only to those hosts inserted into that database.

Every once in a while, a cron job takes care of removing expired entries from the database and the removed host must authenticate again before being able to relay new emails.

2.2 Known problems

courier-auth-postfix is able to use both the dict or the map interface of postfix (look to postfix documentation). However, many dict or map modules are only able to read databases and thus cannot be used with courier-auth-postfix. At the time of writing, none of the LDAP or MySql modules support write operations.

There are also some known problems with hash or btree tables, since there are issues with locking and related to the creation of the databases (you should create the database by hand before using courier-auth-postfix, either with postmap or any other tool).

Right now, I've used this module in production environments only making use of optimizers (optimizers can increase speed, add supports for different kind of databases or supply missing functionalities in the dict modules).

The interface with other databases has just been tested for "correctness" and not for "usability". In practice, courier-auth-postfix should be able to write to any database postmap is able to write to without problems as long as postfix does not have any trouble in handling a database currently being modified (postfix should correctly take care of locking).

I suggest you test the database you want to use before actually using it (help will be provided in the following sections). Any feedback would be appreciated and I'll be happy to eventually add your own optimizers or suggestions. Just mail me at <ccontavalli at commedia.it>.

Watch out! Right now courier-authpostfix should be considered beta-quality software. Although I've been using the mysql optimizer on a production system, bugs may exist, even in the debian packages that haven't been deeply tested!

3 Installation

You have 2 different ways of installing courier-auth-postfix. The first one should work on any system or any distribution, and involves getting the source from www.postfix.org and compiling it. The second one consists in installing the .deb packages on any debian distribution. You should find at least one method that suits your needs. The debian packages used to be available on my own site, but they are now available upon request. If you need them, just mail me... I'll send you either precompiled binaries or a patch to build the debian packages.

Note, however, that since courier-auth-postfix is linked against some libraries provided with postfix, you must have installed the same version of postfix you used to compile courier-authpostfix with.

3.1 From scratch

- Get the source for courier-auth-postfix from <http://www.commedia.it/ccontavalli/#AuthPfix>

- Untar it somewhere on your hard drive with something like:

```
$ gzip -cd courier_authpostfix.tar.gz | tar -xv
```

- Get the source of postfix from www.postfix.org
- Untar it somewhere on your hard drive with something like:

```
$ gzip -cd postfix-x.x.x.tar.gz | tar -xv
```

- Ok, now you need to do 2 things: first, you need to copy authpfix in the postfix tree, second, you need to tell postfix to compile courier-auth-postfix. So, let's start from the first step...

```
$ cp -a courier_authpostfix/authpfix postfix-x.x.x/src
```

- Now, you need to change postfix-x.x.x/Makefile.in in order to make the line that looks like

```
DIRS      = src/util src/global src/dns src/master src/postfix src/smtstone \
```

to look like

```
DIRS      = src/util src/global src/authpfix src/dns src/master src/postfix src/smtstone
```

(Note the additional src/authpfix) That way, you tell postfix makefiles to build authpfix right after src/util and src/global.

- Follow the instructions provided with postfix to compile postfix **Whatch out!** there are still some problems with library dependencies in the authpfix makefile. If you experience troubles compiling the module, please send me a mail with the output of make.
- You should end up with two additional files in postfix-x.x.x/bin: authpostfix and authcleanup.
- Copy authcleanup in /usr/sbin while put authpostfix in courier authentication modules directory (depends on your distribution and os). It should be somewhere in /usr (in debian is /usr/lib/courier/authlib) and you can probably find it with something like:

```
$ locate auth | grep ^/usr | grep courier
```

or using the "libexecdir" parameter printed by

```
$ courier-config
```

No special rights are needed (0755 should be good for both, with root:root as owner).

- Now, you need to modify the /etc/init.d/pop3d (or courier-pop or whatever) script in order to support more configuration parameters by changing:

```
/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
```

in something like:

```
/usr/bin/env - LOGINDB="$LOGINDB" PATH="...and so on
```

Note, however, that this file may be different from system to system.

- Edit courier-auth-postfix.cron you can find in courier_authpostfix/authpfix and modify the name of the database you want to use. The name of the database must follow postfix standards. For example, a valid database name may be: mysql:/etc/postfix/mysql/mysql.dict. Look to courier-authpostfix(1) for more details.

- Copy `courier-auth-postfix.cron` in `/etc/cron.d/`.
- Now, you can add a new configuration parameter: “LOGINDB”. You must add this to the `courier-imap` configuration file (`imapd?`) and `courier-pop` configuration file (`pop3d?`). Its value must be the same database as used with the cron script. For example, you may set “`LOGINDB=mysql:/etc/postfix/mysql/mysql.dict`”.
- Add the `authpostfix` as the **last** authentication module both in `imapd` and `pop3d`. Look the section “Usage” for more details.

3.2 With the Debian packages

- Install `courier`

```
# apt-get install courier-base courier-pop courier-imap
```

- Install `postfix`, either using `dpkg` with the packages provided on my site (safest, after downloading them), with something like:

```
# dpkg -i postfix*
```

or using `apt` with:

```
# apt-get install postfix.*
```

- Install `courier-authpostfix.deb` with:

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

- Answer the questions and you should be done.
- You may want to set pins up in `/etc/apt/preferences` in order to avoid having the packages overwritten on the first `apt update-upgrade`.

3.3 Building new Debian packages

In order to do this, just download the latest `postfix` sources from www.debian.org and patch them with the diff provided on the same site.

Apply the patch on my site only at last, before compiling everything.

Follow the standard procedure to rebuild debian packages.

4 More details

If you used the debian packages, you shouldn't be reading this unless something went really wrong. You can specify the database to use with the environment variable `LOGINDB`.

- For `courier` authentication modules, you need to set this parameter both in `/etc/courier/imapd` (or wherever your configuration files for `courier` are) and `/etc/courier/pop3d` and to modify `/etc/init.d/courier-pop` as indicated above.
- For the cron job, modify `/etc/cron.d/courier-authpostfix.cron` (the number indicates how old, in minutes, can at most the entries be, while cron regulates how often the database should be cleaned up) and set the name of the database.
- For `postfix`, add the following lines to `/etc/postfix/main.cf`:

```
smtpd_recipient_restrictions =
    permit_mynetworks
    check_client_access type:nameofdb
    check_relay_domains
```

where type:nameofdb is the name of the database. These lines instruct postfix to allow relaying only from “mynetworks” and to those host listed in the database.

Ok, now you should have something like “LOGINDB=type:nameofdb” both in /etc/courier/{pop3d,imapd}, you should have a cron job in /etc/cron.d that looks like “authcleanup 10 type:nameofdb” and you should have modified /etc/init.d/courier-pop (or pop3d) as indicated above. Now, you only need to tell courier to load authpostfix as the **last** authentication module. You should check the documentation provided with courier or with your distribution in order to correctly do this. Usually you should add authpostfix as the last option of the AUTHMODULES parameter in the configuration files of imap and pop3. Note that authpostfix **must be the last authentication module to load**, otherwise it won't work.

4.1 Persistent entries

Although there are many other ways to achieve this, if you want a host to be always allowed to relay mail with your SMTP server, you may add an entry in your database with LOGINTIME set to 0. courier-authpostfix would then respect this entry and never remove it.

5 Optimizers

Actually, only a mysql optimizer is provided.

5.1 MySQL optimizer

In order to make use of the mysql optimizer, just configure postfix to use mysql. Then, you just have to pass the name of the database (as used with postfix) to courier-authpostfix as outlined above (editing the configuration files of pop3d and imapd and modifying the cron job).

The MySQL optimizer maintains support for multiple MySQL servers in case one dies. However, keep in mind that courier-authpostfix will not take care of maintaining all the databases in sync, and some other method should be used (like binary logging).

Keep also in mind that little if no test have been made to verify reliability of this module (for short, it may not work if the main database dies).

If you are wondering why to use a distributed database such as MySQL for pop-before-smtp, think to an environment where you have a cluster of more than one SMTP server and you distribute users using something like perdition with courier-imap or pop3.

5.1.1 Table format

Ok, a 2 fields table with a char(12) for the ip address and a varchar(20) for the logintime should be enough. In postfix, select field should be set to logintime while the where field should be set to the ip address.

6 Testing

6.1 Checking your db

To check if authpostfix can write in your db, just:

```
# export AUTHENTICATED=someuser@domain
# export TCPREMOTEIP=:ffff::127.0.0.1
# export LOGINDB=type:nameofdb
```

and manually run authpostfix with something like `./authpostfix`. You should see nothing on the screen and have the prompt right back. If everything worked right, you should check the database to see if an entry for 127.0.0.1 has been added.

You may then cleanup the same db waiting a minute (or using 0) and executing `/usr/sbin/authcleanup 1 type:nameofdb`.

Again, you should see no output and it should immediately return. Check the db if everything went ok.

Once these tests have been made, any eventual error should be looked up in the configuration parameters.

6.2 Troubleshooting

If something doesn't work, I suggest you "telnet localhost 110" and first try the pop3 protocol manually (main commands are: USER username, ASS password, LOGOUT). You should see no errors and if the user is authenticated it should be added to the specified database.

You should then try to send an email with postfix to some remote user, like `abc@hotmail.com` and check the logs for errors.

If you can't find the error, strace could be your best friend. Good Luck.