

Courier Auth PCRE

Carlo Contavalli <ccontavalli at commedia.it>

\$Revision: 1.12 \$ - \$Date: 2002/10/31 23:39:31 \$

Contents

1 License, copyright and...	1
2 What is authpcr?	1
3 Installation	2
4 Configuration	3
5 Configuration file format	3
6 Testing	4

1 License, copyright and...

This document and authpcr were written by Carlo Contavalli <ccontavalli at commedia.it> and are thus Copyright © Carlo Contavalli 2001-2002.

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.

authpcr 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.

authpcr 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 What is authpcr?

authpcr is an authentication module for courier that modifies logging in usernames as specified by a list of perl regular expressions. The most common use of this module is to rewrite usernames to some standard form in order for courier to allow many different usernames to be authenticated.

For example, by just setting up few regular expressions, it would be possible to allow `someuser@somedomain.com` to log in even if he specifies as a username:

- `someuser@mail.somedomain.com`
- `someuser@pop.somedomain.com`
- `someuser%somedomain.com`
- `someuser@mail.somedotherdomain.com`

where `somedotherdomain` users should move to `somedomain.com`. Obviously, some of you may find this useful while some others may find this as a security threat. Just beware and think twice before writing any regular expression.

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

If you have troubles/suggestions/corrections feel free to mail me at [<ccontavalli at commedia.it>](mailto:ccontavalli@commedia.it).

3 Installation

`authpcre` is not a patch, and you may install it on a running system without too much trouble. Ok, to get it working you need the `pcre` library, available on <ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/> and on most linux distributions (in debian, “`apt-get install libpcre3-dev`”).

You also need an ANSI C compiler (`gcc` should be good enough :-)) and a POSIX system with all the standard include files.

Ok, now that you are sure you have everything installed, just:

1. `tar -xvzf authpcre-version.tar.gz`
2. `cd authpcre-version`
3. `./configure`
4. `make`
5. `make install`

`configure` will try to figure out everything for you. Anyway, there are few cases when you must specify some parameters:

- If the directory where `courier` stores modules is not “`/usr/lib/courier/authlib`”. In this case, specify a “`--with-mod-path=/correct/path/to/use`”. You can figure out the path by looking where the `authdaemon` module is stored (something like `locate authdaemon`) or by using `courier-config` and appending “`/authlib`” to `$libexecdir`.
- If you want the manual pages in a different place than `/usr/local/man`. In this case, use the “`--mandir=/usr/share/man`” to specify a different location.
- If you want the `authpcre-test` binary to be installed in a different place than `/usr/local/bin`. In this case, use the “`--bindir=/something/else`”.
- If you have a different `pcre` library. Use the parameter “`--with-pcre=someotherlibrary`” to specify the library to use. Note that library must be in your linker search path (default is `libpcre`).

- If the header “pcre.h” has a different name or is in some non-standard place. In this case, use the parameter “--with-hpcre=/path/file_to_use.h” to specify the header to use and its location (default is pcre.h, in any of the compiler search paths).
- If you want the configuration file to be in some other place than the default /etc/courier/authpcre. Just use the parameter “--with-conf-file=/path/name/of_file”.

All the other parameters (as printed by `configure --help`) are completely ignored. For example, you may configure `authpcre` with something like

```
# ./configure --with-mod-path=/usr/local/courier/lib \
               --with-conf-file=/usr/local/courier/etc/authpcre
```

Watch out that `make install` will install the binary itself and an example configuration file. Beware that the `configure.ac` file was written in a very short time and is currently being worked on to make it more “standard”.

Precompiled binary packages for debian may be made available for your convenience.

There is no mailing list for `authpcre`, so feel free to mail me if you have any trouble with it.

4 Configuration

Ok, now you need to configure `authpcre`. First of all, you need to tell `courier imapd` and `courier pop3` you want to use `authpcre` as an authentication module. To do this, just open `imapd` and `pop3d` configuration files (on my system, /etc/courier/imapd and /etc/courier/pop3d) with your preferred editor, look for a line like

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

and change it in

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

Beware! `authpcre` **must** be the **first authentication module**. In fact, any preceding authentication module wouldn't see the modified username but the original one. Do it at your own risk.

There is one additional configuration parameter that may be specified (only if you want/wish/need to), “PCRECONFIG”. It allows you to explicitly ask for a configuration file to be used. This is useful if you have different instances of `imap` servers that should use different configuration files (is there anybody out there with something like this? Why did you want to do something like that?) or if you want to specify two different configuration files for `imapd` or `pop3d`. Watch out that at time of writing the `pop3d` init script does not pass additional parameters over to the daemon when it starts it up, and thus the `PCRECONFIG` parameter would be lost. It is however trivial to modify the init script to pass `PCRECONFIG` over. Have fun!

Just to give you a hint, the standard “/etc/whatever_init.d/pop3d” should contain a line that looks like:

```
/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
```

that should be changed to look like:

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

5 Configuration file format

The configuration file of `authpcre` (usually `authpcre.rc`) is a list of comments and regular expressions. A comment is a line starting with any number of whitespaces followed by a “#”. Whitespaces outside regular expressions are ignored. A regular expression starts with any nonblank character beside “#” and “\” that acts as a terminator. It is followed by a perl style regular expression (man `perlre`), the terminator, the string to substitute, another terminator and a list of options.

The string to substitute may contain a \$1, \$2, \$3 and so on that respectively indicate the first matched string, the second one and so on (\$0 indicates the whole match)...

Characters are escaped using a “\”. Supported options are:

- **i** for case insensitive matching
- **a** for anchored matching (look at the `pcre` manual)
- **x** for extended matching (again, look at the `pcre` manual)
- **y** asks for non-greedyness (this time, look at the `perlre` manpage)

For example a valid configuration file may look like:

```
# This is a
# valid configuration file

/\%/@/                                # Replaces the first % found with an @
    $mail\.$$i                          # gets rid of any mail. in the username or domain part
                                          # in a case insensitive way (watch out if you have an
                                          # domain like mail.com or mail.org, or if any user
                                          # uses mail. as his login name :-)
```

```
/^([^@]*)@mail\.([^\.]*\..*)$/i        # same thing as above, but matches only the domain part
                                          # and ensures that we are removing only subdomains
~@foo.com$~@bar.com~i                 # changes domain foo.com in bar.com
```

6 Testing

If you want to test your configuration before using it, you can use `authpcre-test`.

`authpcre-test` applies all the regular expressions provided in `authpcre` configuration file to each line of the standard input, printing the result on standard output.

If you want to test a different configuration file, you can set the environment variable `PCRECONFIG` before calling `authpcre-test`. For example:

```
$ export PCRECONFIG=/home/fuffolandia/fuffolo.rc
$ authpcre-test
```

If something goes wrong, I suggest you double check your regular expression. If still you think there's some kind of problem, you can recompile `authpcre` using “`make debug`”.