ProFTPd HOWTO

Setting up FTP accounts for users using ProFTPd

Florian Schiessl

<olfi (at) debianhowto (dot) de>

2003-06-13

Revision History

Revision 1.0	2003-02-20	olfi
Startrelease		

Transferring files over file transport protocol (FTP) is widespread. This howto will describe how the ProFTP server http://www.proftpd.org will be installed and configured so that only users, which are members of a special group, can establish a connection. Additional they only can view their own homedir as rootdir from which they shouldn't be able to escape using current tools.

Table of Contents

Introduction

ProFTPd (www.proftpd.org) is a FTP server which configuration is similar to apache webserver. It's flexible and applies to be relative secure. Here, the installation and configuration of user access to his webdir will be described. Beside the alternative to authenticate ProFTPd using PAM there are special packages for MySQL and PostgresSQL authentication.

Copyright and License

Disclaimer

No liability for the contents of this document can be accepted. Use the concepts, examples and information at your own risk. There may be errors and inaccuracies, that could be damaging to your system. Proceed with caution, and although this is highly unlikely, the author(s) do not take any responsibility.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark. Naming of particular products or brands should not be seen as endorsements.

Feedback

Feedback is most certainly welcome for this document. Send your additions, comments and criticisms to the following email address : `<olfi (at) debianhowto (dot) de>`.

Translations

Pointers to available translations are nice. Also your translators tend to give very important inputs.

- German Translation [http://www.debianhowto.de/index.de.html] provided by Florian Schiessl `<olfi (at) debianhowto.de>`

Installing and configuring ProFTPd

Installing ProFTPd

The usual debian way.

```
obelix:~# apt-get install proftpd
```

Answer the question if the configfiles should be edited with YES. If you use FTP only sometimes to up- and download your files then say *inetd* at the following question - but if you'll use your server mostly for ftp then say *standalone*. Normally you don't need anonymous access so say NO to this question. Only if you know exactly why you would use anonymous access you can say YES, but then you have to choose very secure settings so that your server wouldn't misused for warez (and the traffic will be adequate high). The `/etc/proftpd.conf` should even be updated - YES.

A basic configuration option

So far there was created a standard configuration file. To grant only some users access over ftp, who only can view their homedir on the server (chroot environment), put this users together in one secondary group. Access will only be able for users of this group. At first, create this group.

```
obelix:~# addgroup ftpuser
```

Then, every user who should have ftp acces have to be at least a secondary member of this group. Either you choose the option **-G ftpuser** at creating the user or additional you add users to the secondary group using

```
obelix:~# usermod -G ftpuser username
```

An adapted configuration file could look like this:

```
# This is a basic ProFTPD configuration file (rename it to
# 'proftpd.conf' for actual use. It establishes a single server
# and a single anonymous login. It assumes that you have a user/group
# "nobody" and "ftp" for normal operation and anon.

ServerName "meinserver"
ServerType inetd
DeferWelcome off

ShowSymlinks on
MultilineRFC2228 on
DefaultServer on
AllowOverwrite on
```

```
TimeoutNoTransfer 600
TimeoutStalled 600
TimeoutIdle 1200

DisplayLogin welcome.msg
DisplayFirstChdir .message
LsDefaultOptions "-l"

DenyFilter \*.*/

# Uncomment this if you are using NIS or LDAP to retrieve passwords:
#PersistentPasswd off

# Port 21 is the standard FTP port.
Port 21

# To prevent DoS attacks, set the maximum number of child processes
# to 30. If you need to allow more than 30 concurrent connections
# at once, simply increase this value. Note that this ONLY works
# in standalone mode, in inetd mode you should use an inetd server
# that allows you to limit maximum number of processes per service
# (such as xinetd)
MaxInstances 30

# Set the user and group that the server normally runs at.
User nobody
Group nogroup

# Normally, we want files to be overwriteable.
<Directory /*>
# Umask 022 is a good standard umask to prevent new files and dirs
# (second parm) from being group and world writable.
Umask 022 022

AllowOverwrite on
</Directory>

# here are my improvements

# chroot for all users of the group ftpuser
DefaultRoot ~ ftpuser

# grant login only for members of the group
<Limit LOGIN>
DenyGroup !ftpuser
```

```
</Limit>

# disable root login and require a valid shell (from /etc/shells)
<Global>
RootLogin off
RequireValidShell on
</Global>

# increase
UseReverseDNS off
IdentLookups off

# Logging formats
LogFormat default "%h %l %u %t \"%r\" %s %b"
LogFormat auth "%v [%P] %h %t \"%r\" %s"
LogFormat write "%h %l %u %t \"%r\" %s %b"

# activate logging

# every login
ExtendedLog /var/log/ftp_auth.log AUTH auth

# file/dir access
ExtendedLog /var/log/ftp_access.log WRITE,READ write

# forr paranoid (big logfiles!)
#ExtendedLog /var/log/ftp_paranoid.log ALL default
```

If the users don’t get a normal login shell (/bin/sh, /bin/bash etc.), then I recommend to create a pseudo shell **/bin/ftp** as a copy of **/bin/false** and add it in `/etc/shells`.

```
obelix:~# cp /bin/false /bin/ftp
obelix:~# echo "/bin/ftp" >> /etc/shells
```

Already created users can be assigned this shell using **usermod -s /bin/ftp username**

Restart the inetd and you’re ready ;-)

```
obelix:~# /etc/init.d/inetd restart
```

Usinf xinetd instead of inetd

As far as you have already installed the more modern **xinetd** instead of the **inetd** you have to alter its configuration at this point!

More you can find at our xinetd Howto [../xinetd/].

Summary of user configuration

At this point I summarize what affects the so far created users and the user who will be created in future; which shell should be choosen and how to create new users and alter existing users.

user access only over ftp. This is the most probable alternative: Users have only access over ftp to their directories. If there are still any users who should be reconverted to "only ftp" access use the following command **usermod -s /bin/ftp username**. At creating new users think of the -s beside other specific options: **useradd ... -s /bin/ftp ... username**.

user have ftp and shell access. I advise against this if it not absolute neccessary. Exiating users with shell access have already a valid shell. At creating new users think of the -s beside other specific options: **useradd ... -s /bin/bash ... username**. Instead of `/bin/bash` you can use every valid shell. The shell have to be listed in `/etc/shells`.

Users have only ftp access but are able to change their pass using an shell. As far as users should have the possibility to change their ftp pass using a shell but shouldn't do anything further using the shell the best choice will be to give them a special shell `/usr/bin/passwd`. There they will be prompted for a new pass and after retyping this the connection will be closed. If this fits to your users is your choice, but often users are overcharged using ftp so that this possibility won't make them happy in any sense. As mentioned above, the argument -s have to be given at command line **useradd/usermodd ... -s /usr/bin/passwd ... username**. `/usr/bin/passwd` have to be added at `/etc/shells`:

```
obelix:~# echo "/usr/bin/passwd" >> /etc/shells
```

Use SSH instead of telnet!

If the user has shell access don't use telnet, use ssh ;-)