## VSFTPD – Restricting access to folders in our FTP Server

Sometimes, we want to allow selected users to upload via ftp and nothing else. This is common in webservers for example.

Many hosting providers provide ftp access to the web server so that their customers can upload files (html, js) to the webserver. Of course, we don’t want these users browsing to other directories (other customer websites, system files etc).

We can define a directive in vsftpd.conf called “**local\_root**”: **sudo vi /etc/vsftp.conf**, then restart.

Text

Description automatically generated

Next we need to create the /var/ftp/ directory: **sudo mkdir /var/ftp**

Graphical user interface

Description automatically generated with medium confidence

Create a file called Rules.txt. Now, when users log in to the ftp server, they should automatically have **/var/ftp/** as the current working directory on the server:

Text

Description automatically generated

What if we wanted to restrict the user to **ONLY** that directory?

We can use something called “chroot jail”. This allows admins to set the root directory of a connected account to something other than “/” on the local system.

To enable this, we need to add another directive to **vsftpd.conf** called “**chroot\_local\_user=YES**”:

Text

Description automatically generated

Then restart vsftpd: **sudo systemctl restart vsftpd**.

And ftp as ftp-guy once again, instead of it saying /var/ftp/ as our working directory, it will say “/”:

Text

Description automatically generated

No matter what, we can’t get out:

Text

Description automatically generated

Trying to change directories above the root (for example, cd .. to change to /var from /var/ftp) will say it was successful, but printing our working directory shows us that we didn’t do anything. We’re still securely trapped in chroot jail at “/” (which is really /etc/ftp/).

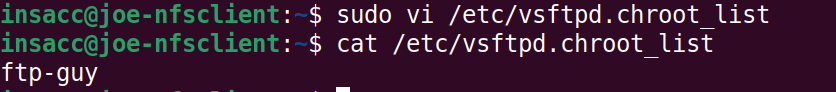
Note: This change would be effective for **ALL** users. What if we want some users chrooted, and others not?

We can use the directive “**chroot\_list\_enable**” which, by default, will reference a file “**/etc/vsftpd.chroot\_list**” as a list of users that are jailed. Also change **chroot\_local\_user** to **NO**

Text

Description automatically generated

Create the file “/etc/vsftpd.chroot\_list” and adding a user to it, we can see the following behaviour:



When we connect with “ftp-guy”, we’ll notice that they have been “chrooted”. If we connect with a different user (ie: insacc), we can see that they are not “chrooted”.

Text

Description automatically generated

We can reverse behavior (so, maintain a list of users that are NOT chrooted) by changing the directive “chroot\_local\_user=YES”:

Text

Description automatically generated

Now, ftp-guy is not chrooted, but everyone else is:

Text

Description automatically generated

## Enable Anonymous uploads

FTP has a built-in feature to allow for anonymous access.

To enable anonymous access, we would need to enable two directives. “**anonymous\_enable=YES**” as well as “**anon\_root=<somedir>**” (in this case, we’ll do **anon\_root=/var/ftp/pub**)

A picture containing text

Description automatically generated

Restart and add the “**anonymous**” user to **/etc/vsftp/allowed\_user\_list**

Text

Description automatically generated

To login to the ftp server “anonymously”, use the username (Name) “anonymous” and just press enter to use a blank password. We can see that anonymous users are automatically chrooted to /var/ftp/pub based on the directive changes we made previously to **vsftpd.conf**

Text

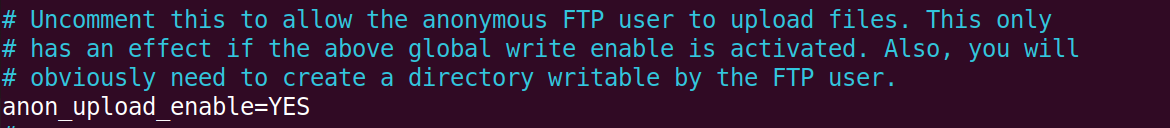
Description automatically generated

What happens if we try to upload something anonymously?

Text

Description automatically generated

We get a permission denied error. There are a few things we need to do in order to allow anonymous users to upload files. First, we need to add the directive “**anon\_upload\_enabled=YES**”:



Let’s try again:

Text

Description automatically generated

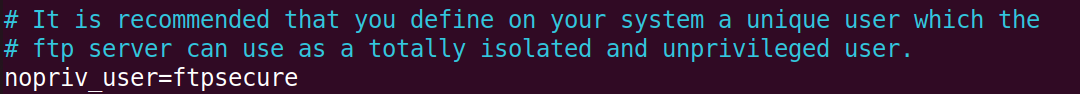
\*\***THIS PART IS INCORRECT:** *VSFTPD runs as “nobody” for anonymous users by default.* **(vsftpd runs as the user “ftp” for anonymous users, not “nobody”. The “ftp” user has permissions of r-x, or read and execute.)**  *Taking a look at the permissions of /var/ftp/pub:* **(Note that when we ftp in as “anonymous” we use the “ftp” user permissions to be able to list the directory contents (ls).**

Text

Description automatically generated

We can see that the only person that has write permissions for /var/ftp/pub is “owner” (in this case “root”). This means we can not upload the file to this location.

We need to create a directory that the ftp user can write to… **The following is incorrect again:** *but “nobody” user is used by a lot of other stuff – let’s create a user only for anonymous access to the ftp. How do we create users again? Let’s try a different method:*

**

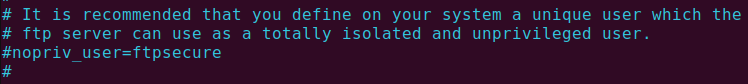
*Adding the directive* ***nopriv\_user=ftpsecure*** *and adding a user* ***ftpsecure*** *to the server will allow anonymous users to be logged in as the user “ftpsecure”, to which we can then restrict permissions. Notice also, that we can not READ from that directory, as most sysadmins will not allow anonymous users to view the files that others users have uploaded anonymously*.

Text

Description automatically generated **🡨 INCORRECT STOPS HERE!**

**To fix the last two incorrect steps we need to:**

1. Edit the vsftp.conf file to remove the incorrect directive:
   1. **sudo vi /etc/vsftp.conf**
   2. Reinstate the # for the “***nopriv\_user=ftpsecure***” directive



* 1. Save the changes and restart the vsftpd: **sudo systemctl restart vsftpd**

1. Remove the “ftpsecure” user:
   1. **sudo deluser ftpsecure**

Text

Description automatically generated

1. Double check your /var/ftp/pub permissions are 755 and the owner is root, we did a lot of finagling and if these were changed they should be changed back:
   1. **sudo chmod 0755 /var/ftp/pub**
   2. **sudo chown root /var/ftp/pub**
   3. Confirm:

Text

Description automatically generated

**And now moving on with correct instructions… 😊**

We need to create a directory that the ftp user can write to: **mkdir /var/ftp/pub/upload**

Next change the owner of this folder to the “ftp” user: **chown ftp /var/ftp/pub/upload**

Text

Description automatically generated

Ensure you are in your home directory where the ftp\_test.txt file is located, ftp to the localhost as “anonymous” (leave the password blank).

After this you should now be able to **ls** and see the “**uploads**” folder.

Go ahead and “**cd uploads**” then “**put ftp\_test.txt**.” The transfer should now complete successfully. 😊

Text

Description automatically generated

**EXERCISE:**

1. Find a partner.

2. Create an account with their name and the password "secret" on your machine.

3. Make it so only anonymous and your buddy can log in.

3. Make it so your partner is chroot jailed to /group\_project, and can only upload files to /group\_project/contributions.

4. Have your buddy put a file called hard\_work in your /group\_project/contributions.

5. Make it so anon users can't upload anything.

6. set the anon\_root=/group\_project.