FTP Server

chroot ftp user

By default any account with SSH access already has SFTP access as well. The problem with just adding a user and letting them have at it is that by default along with SFTP access users which you only wish to grant file management access also have shell access and may be able to install and run processes you may not wish them to run on your server. The following is a command by command walk through on how to allow users only file management access via SFTP while not allowing them to gain shell access on your server.

These command by command instructions are Ubuntu/Debian centric but only with the in the case of apt-get and process restart commands and should be no problem to adapt to any other distro. The first thing we will want to do is install SSH which generally is installed by default but just in case…

```
sudo apt-get install openssh-server
```

Once we know SSH is installed we can then begin to add users we wish to allow only SFTP access only to. We will start by adding a new user group to our system. I will use “sftponly” for the name of this group but the name of the group is up to you. Just be sure that if you use a different name to modify any proceeding referances to the “sftponly” group name with the one you used.

```
sudo groupadd sftponly
```

Now open the file /etc/ssh/sshd_config in your favorite text edit. I prefer nano for such trivial edits.

```
sudo nano /etc/ssh/sshd_config
```

Look for the line near the bottom of this file that looks like this.

```
Subsystem sftp /usr/lib/openssh/sftp-server
```

And change it to this.

```
Subsystem sftp internal-sftp
```

At the very bottom of this file you will need to add the following lines which restrict the “sftponly” user groups access when logging in via SSH.

```
# Rules for sftponly group
Match group sftponly
ChrootDirectory %h
X11Forwarding no
AllowTcpForwarding no
ForceCommand internal-sftp
```

Once you have added these lines save the file and restart the SSH process.

```
sudo /etc/init.d/ssh restart
```

Next we will create the home directory for the user we are about to add where they will be allowed file access to as well as where they will be chrooted or jailed. This directory can be located wherever but for the sake of this tutorial I will use the following directory.

```
sudo mkdir /var/www/vhost/domain.com/
```

Now we will need to add a user as we normally would under Linux. At this time we will also specify the users home directory which we just created using the “-d” flag. Remember to replace “sally” with the login name for the user you wish to add.

```
sudo useradd -d /var/www/vhosts/domain.com/ sally
```

Then we will need to change the group which the user we just added will belong to.

```
sudo usermod -g sftponly sally
```

Next we will set our new user’s shell to /bin/false which will not allowing our new user shell login.

```
sudo usermod -s /bin/false sally
```

Set a password this user will use.

```
sudo passwd sally
```

We will now need to give ownership to any files and folders within the new users home directory which may exist such as folders brought over from a skel or created by an administrator setting up the users home directory manually.

```
sudo chown sally:sftponly -R /var/www/vhosts/domain.com/
```

Now comes an important part the newly created users home directory MUST be owned by root. If this directory is not owned by root then the newly created user may not be chooted or jailed within their home directory possibly allowing them access to other directories under theirs.

```
sudo chown root:root /var/www/vhosts/domain.com/
```

And that is it you now have set up a user allowed only SFTP access over SSH with no shell access and chroot them within their directory. You are now much safer then you were when running FTP.

From:
https://studie.famvisser.net/ - **Studie Pagina Chavoerah Maqor**

Permanent link:
https://studie.famvisser.net/doku.php?id=li_ftp&rev=1396448306

Last update: **2022/05/06 21:58**