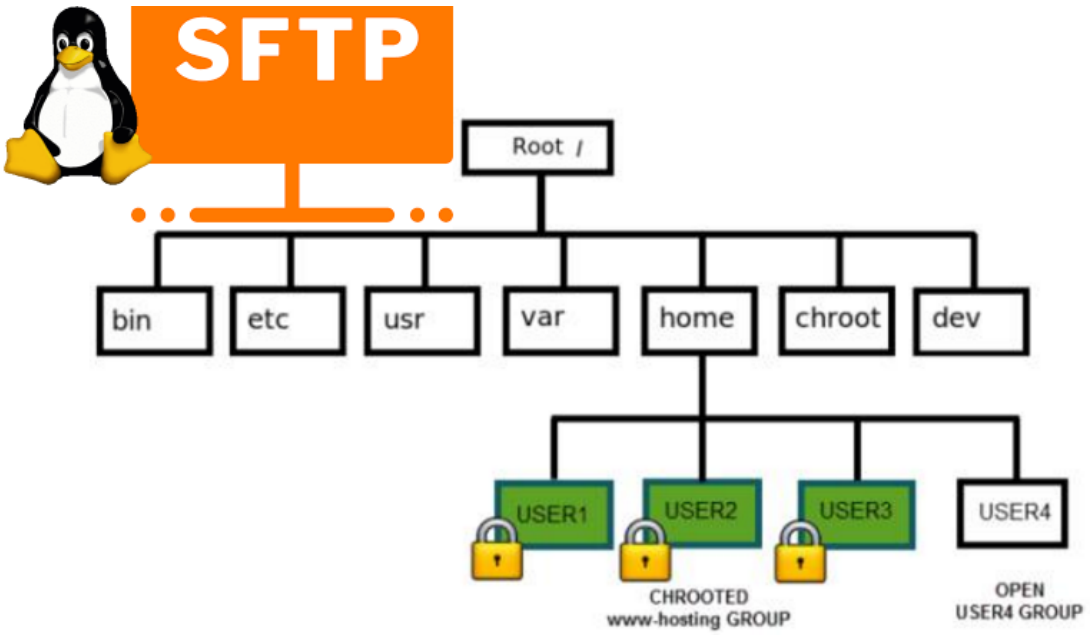
Create SFTP CHROOT Jail User for data transfer to better Linux shared web hosting server security

Author : admin



Adding user SFTP access to a Linux system is often required and therefore a must **for multi users or web hosting environments it is an absolute requirement to have SFTP user space separation (isolation) out of the basic Linux system environment** this is done using a fake **CHROOT Jail**.

Purpose of this article is to show how to **create SFTP Chroot JAIL in few easy configurations**.

By isolating each user into his own space you will *protect the users to not eventually steal or mistakenly leak information such as user credentials / passwords etc*.

Besides that it is useful **to restrict the User to his own File / Web Space** to have granted **only access to Secure FTP (SFTP) only and not SSH login access** and togheter with the **chroot jail environment to protect your server from being attempted to be hacked (*rooted / exploited*)** through some (0day) ***zero-day kernel 1337 vulnerability.***

1. Setup Chrooted file system and do the bind mount in /etc/fstab

```
# chown root:root /mnt/data/share
# chmod 755 /mnt/data/share
# mkdir -p /sftp/home
# mount -o bind /mnt/data/share /sftp/home
```

Next add to **/etc/fstab** (e.g. **vim /etc/fstab**) and add following line:

```
/mnt/data/share /sftp/home  none   bind   0   0
```

To mount it next:

```
# mount -a
```

/mnt/data/share is a mounted HDD in my case but could be **any external attached storage**

2. Create User and sftpgroup group and add your new SFTP Jailed user accounts to it

To achieve SFTP only CHROOT Jail environment you need some UNIX accounts new group created such as **sftpgroup** and use it to assign proper ownership / permissions to newly added SFTP restricted accounts.

```
# groupadd sftpgroup
```

Once the group exists, next step is to *create the desired username / usernames* with **useradd** command and assign it to **sftpgroup:**

```
# adduser sftp-account1 -s /sbin/nologin -d /sftp/home
# passwd sftp-account1
```

```
# usermod -G sftpgroup sftp-account1
```

Above both commands could be also done in one line with adduser

```
# adduser sftp-account1 -g sftpgroup -s /sbin/nologin -d /sftp/home
```

Note the **/sbin/nologin** which is set to prevent SSH logins but still allow access via **sftp / scp** data transfer clients Once the user exists it is a good idea to prepare the jailed environment under a separate directory under root File system system lets say in **/sftp/home/**

3. Set proper permissions to User chrooted /home folder

```
# mkdir -p /sftp/home
# mkdir /sftp/home/sftp-account1
# chown root:root /sftp/
# chown sftp-account1:sftpgroup /sftp/home/sftp-account1
```

For each new created uesr (in this case **sftp-account1**) make sure the permissions are properly set to make the files readable only by the respective user.

```
# chmod 700 -R /sftp/home/sftp-account1
```

For every next created user don't forget to do the same

3. Modify SSHD configuration file to add Chroot match rules

Edit **/etc/ssh/sshd_config** file and to the end of it add below configuration:

```
# vim /etc/ssh/sshd_config
Subsystem sftp internal-sftp
Match Group sftpgroup
ChrootDirectory /sftp/home
ForceCommand internal-sftp
X11Forwarding no
AllowTcpForwarding no
```

Restart **sshd** to make the new settings take effect, to make sure you don't ed up with no access (if it is a remote server) run the sshd daemon on a secondary port like so:

```
# /usr/sbin/sshd -p 2208 &
```

Then **restart sshd** - if it is old Linux with **Init V support**

```
# /etc/init.d/sshd restart
```

- For **systemd** Linux systems

 # systemctl restart sshd

4. Verify Username (sftp-account1) could login only via SFTP and his environment is chrooted

ssh sftp-account1@pc-freak.net

This service allows sftp connections only.
Connection to 83.228.93.76 closed.

sftp sftp-account1@pc-freak.net Connected to 83.228.93.76. sftp>

5. Closure

The quick summary of What we have achieved with below is:

restrict Linux users from having no /bin/shell access but still have Secure FTP copy in few steps to summarize them

a. create new user and group for SFTP chrooted restricted access only
b. set proper permissions to make folder accessible only by user itself
c. added necessery sshd config and restarted sshd to make it working d. tested configuration

This short guide was based on documentation on Arch Linux's wiki SFTP chroot you can check it here.