.com

SSH 15

Restrict SSH User Access to Certain Directory Using Chrooted Jail

by Aaron Kili | Published: March 10, 2017 | Last Updated: March 10, 2017

There are several reasons to restrict a SSH user session to a particular directory, especially on web servers, but the obvious one is a system security. In order to lock SSH users in a certain directory, we can use **chroot** mechanism.

change root (**chroot**) in Unix-like systems such as Linux, is a means of separating specific user operations from the rest of the Linux system; changes the apparent root directory for the current running user process and its child process with new root directory called a **chrooted jail**.

In this tutorial, we'll show you how to restrict a SSH user access to a given directory in Linux. Note that we'll run the all the commands as root, use the sudo command if you are logged into server as a normal user.

Step 1: Create SSH Chroot Jail

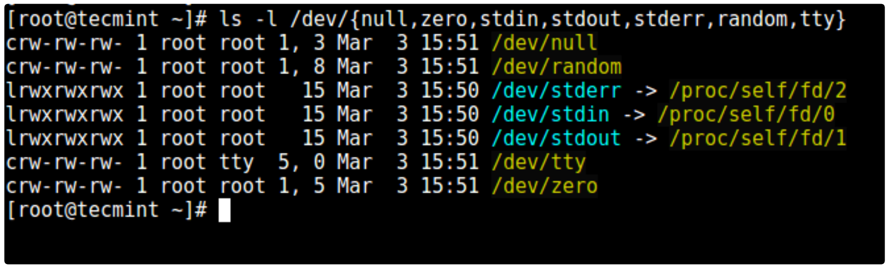
1. Start by creating the chroot jail using the mkdir command below:

```
# mkdir -p /home/test
```

2. Next, identify required files, according to the **sshd_config** man page, the `ChrootDirectory` option specifies the pathname of the directory to chroot to after authentication. The directory must contain the necessary files and directories to support a user's session.

For an interactive session, this requires at least a shell, commonly `sh`, and basic `/dev` nodes such as null, zero, stdin, stdout, stderr, and tty devices:

```
# ls -l /dev/{null,zero,stdin,stdout,stderr,random,tty
```

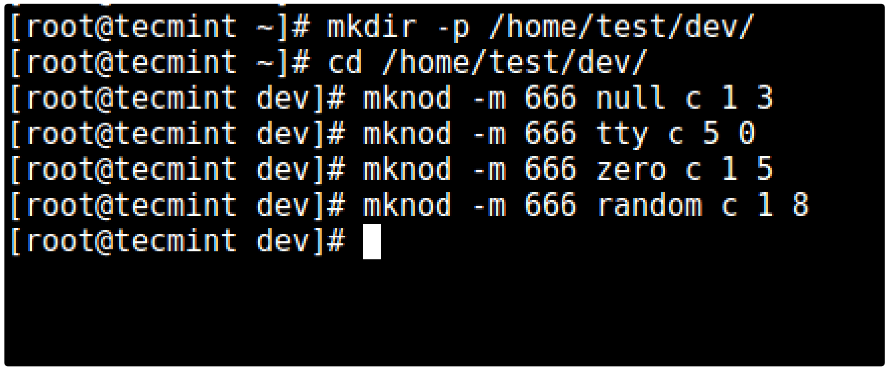


```
[root@tecmint ~]# ls -l /dev/{null,zero,stdin,stdout,stderr,random,tty}
crw-rw-rw- 1 root root 1, 3 Mar  3 15:51 /dev/null
crw-rw-rw- 1 root root 1, 8 Mar  3 15:51 /dev/random
lrwxrwxrwx 1 root root   15 Mar  3 15:50 /dev/stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root   15 Mar  3 15:50 /dev/stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root   15 Mar  3 15:50 /dev/stdout -> /proc/self/fd/1
crw-rw-rw- 1 root tty  5, 0 Mar  3 15:51 /dev/tty
crw-rw-rw- 1 root root 1, 5 Mar  3 15:51 /dev/zero
[root@tecmint ~]#
```

Listing Required Files

3. Now, create the `/dev` files as follows using the **mknod command**. In the command below, the `-m` flag is used to specify the file permissions bits, `c` means character file and the two numbers are major and minor numbers that the files point to.

```
# mkdir -p /home/test/dev/
# cd /home/test/dev/
# mknod -m 666 null c 1 3
# mknod -m 666 tty c 5 0
# mknod -m 666 zero c 1 5
# mknod -m 666 random c 1 8
```

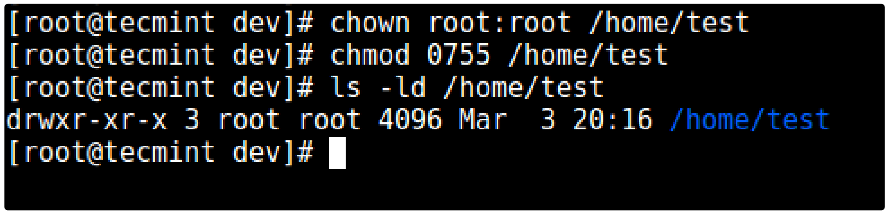


```
[root@tecmint ~]# mkdir -p /home/test/dev/
[root@tecmint ~]# cd /home/test/dev/
[root@tecmint dev]# mknod -m 666 null c 1 3
[root@tecmint dev]# mknod -m 666 tty c 5 0
[root@tecmint dev]# mknod -m 666 zero c 1 5
[root@tecmint dev]# mknod -m 666 random c 1 8
[root@tecmint dev]#
```

Create /dev and Required Files

4. Afterwards, set the appropriate permission on the chroot jail. Note that the chroot jail and its subdirectories and subfiles must be owned by **root** user, and not writable by any normal user or group:

```
# chown root:root /home/test
# chmod 0755 /home/test
# ls -ld /home/test
```



```
[root@tecmint dev]# chown root:root /home/test
[root@tecmint dev]# chmod 0755 /home/test
[root@tecmint dev]# ls -ld /home/test
drwxr-xr-x 3 root root 4096 Mar  3 20:16 /home/test
[root@tecmint dev]#
```

Set Permissions on Directory

Step 2: Setup Interactive Shell for SSH Chroot Jail

How to Add Linux Host to Nagios Monitoring Server Using NRPE Plugin

Nagios 4.2.0 Released – Install on RHEL/CentOS 7.x/6.x/5.x and Fedora 24-19

Install Cacti (Network Monitoring) on RHEL/CentOS 7.x/6.x/5.x and Fedora 24-12

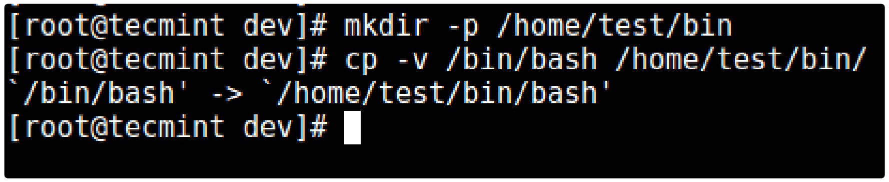
Google Chrome 59 Released – Install on RHEL/CentOS 7/6 and Fedora 25-20

How to Install Ubuntu 16.10/16.04 Alongside With Windows 10 or 8 in Dual-Boot

Tecmint’s Guide to Red Hat RHCSA / RHCE Certification Preparation Study Guide

5. First, create the `bin` directory and then copy the `/bin/bash` files into the `bin` directory as follows:

```
# mkdir -p /home/test/bin
# cp -v /bin/bash /home/test/bin/
```

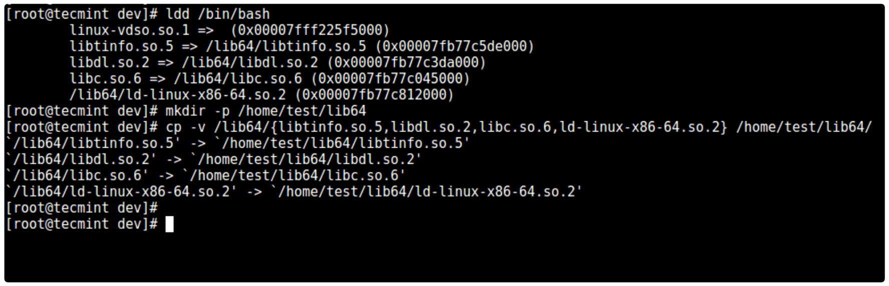


```
[root@tecmint dev]# mkdir -p /home/test/bin
[root@tecmint dev]# cp -v /bin/bash /home/test/bin/
`/bin/bash' -> `/home/test/bin/bash'
[root@tecmint dev]# 
```

Copy Files to bin Directory

6. Now, identify bash required shared `libs`, as below and copy them into the `lib` directory:

```
# ldd /bin/bash
# mkdir -p /home/test/lib64
# cp -v /lib64/{libtinfo.so.5,libdl.so.2,libc.so.6,ld-
```



```
[root@tecmint dev]# ldd /bin/bash
        linux-vdso.so.1 =>  (0x00007fff225f5000)
        libtinfo.so.5 => /lib64/libtinfo.so.5 (0x00007fb77c5de000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007fb77c3da000)
        libc.so.6 => /lib64/libc.so.6 (0x00007fb77c045000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fb77c812000)
[root@tecmint dev]# mkdir -p /home/test/lib64
[root@tecmint dev]# cp -v /lib64/{libtinfo.so.5,libdl.so.2,libc.so.6,ld-linux-x86-64.so.2} /home/test/lib64/
`/lib64/libtinfo.so.5' -> `/home/test/lib64/libtinfo.so.5'
`/lib64/libdl.so.2' -> `/home/test/lib64/libdl.so.2'
`/lib64/libc.so.6' -> `/home/test/lib64/libc.so.6'
`/lib64/ld-linux-x86-64.so.2' -> `/home/test/lib64/ld-linux-x86-64.so.2'
[root@tecmint dev]#
[root@tecmint dev]# 
```

Copy Shared Library Files

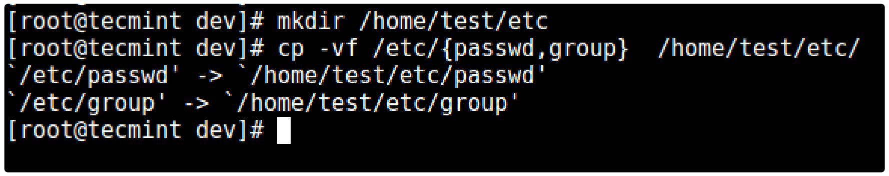
Step 3: Create and Configure SSH User

7. Now, create the SSH user with the useradd command and set a secure password for the user:

```
# useradd tecmint
# passwd tecmint
```

8. Create the chroot jail general configurations directory, `/home/test/etc` and copy the updated account files (**/etc/passwd** and **/etc/group**) into this directory as follows:

```
# mkdir /home/test/etc
# cp -vf /etc/{passwd,group} /home/test/etc/
```



```
[root@tecmint dev]# mkdir /home/test/etc
[root@tecmint dev]# cp -vf /etc/{passwd,group}  /home/test/etc/
`/etc/passwd' -> `/home/test/etc/passwd'
`/etc/group' -> `/home/test/etc/group'
[root@tecmint dev]# 
```

Copy Password Files

Note: Each time you add more SSH users to the system, you will need to copy the updated account files into the `/home/test/etc` directory.

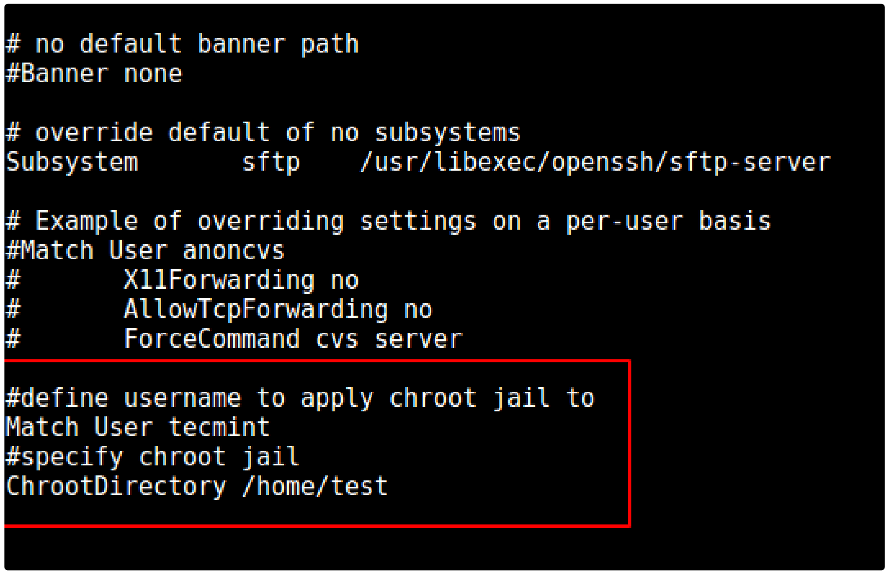
Step 4: Configure SSH to Use Chroot Jail

9. Now, open the `sshd_config` file.

```
# vi /etc/ssh/sshd_config
```

and add/modify the lines below in the file.

```
#define username to apply chroot jail to
Match User tecmint
#specify chroot jail
ChrootDirectory /home/test
```



```
# no default banner path
#Banner none

# override default of no subsystems
Subsystem       sftp    /usr/libexec/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       ForceCommand cvs server

#define username to apply chroot jail to
Match User tecmint
#specify chroot jail
ChrootDirectory /home/test
```

Configure SSH Chroot Jail

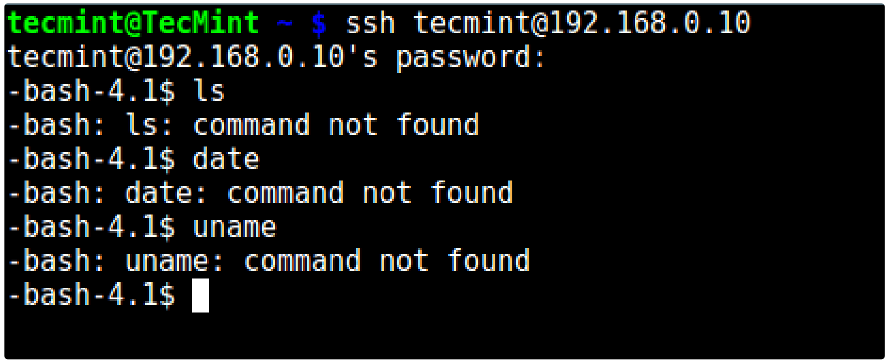
Save the file and exit, and restart the SSHD services:

```
# systemctl restart sshd
OR
# service sshd restart
```

Step 5: Testing SSH with Chroot Jail

10. At this point, test if the chroot jail setup is working as expected:

```
# ssh tecmint@192.168.0.10
-bash-4.1$ ls
-bash-4.1$ date
-bash-4.1$ uname
```



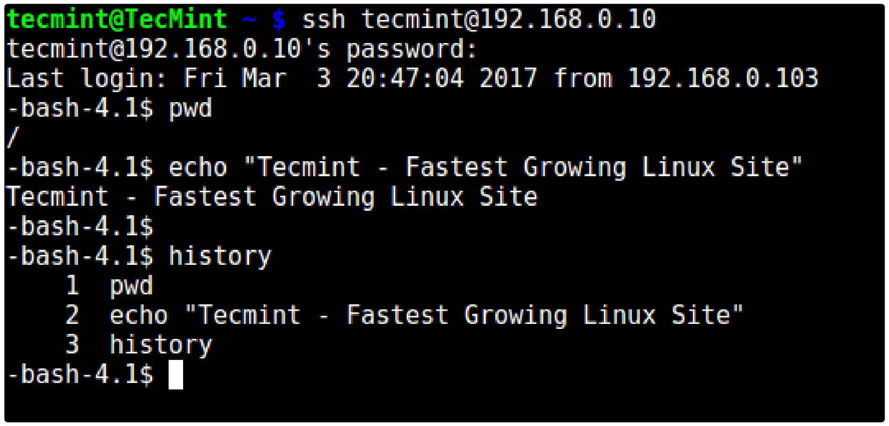
```
tecmint@TecMint ~ $ ssh tecmint@192.168.0.10
tecmint@192.168.0.10's password:
-bash-4.1$ ls
-bash: ls: command not found
-bash-4.1$ date
-bash: date: command not found
-bash-4.1$ uname
-bash: uname: command not found
-bash-4.1$
```

Testing SSH User Chroot Jail

From the screenshot above, we can see that the SSH user is locked in the chrooted jail, and can't run any external commands (ls, date, uname etc).

The user can only execute bash and its builtin commands such as(pwd, history, echo etc) as seen below:

```
# ssh tecmint@192.168.0.10
-bash-4.1$ pwd
-bash-4.1$ echo "Tecmint - Fastest Growing Linux Site"
-bash-4.1$ history
```



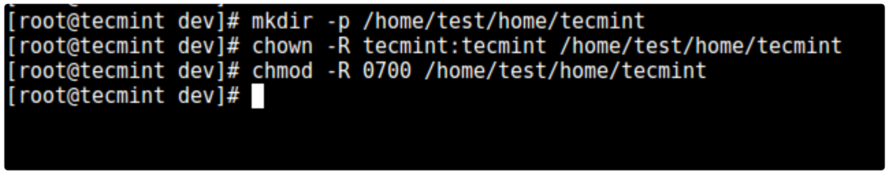
```
tecmint@TecMint ~ $ ssh tecmint@192.168.0.10
tecmint@192.168.0.10's password:
Last login: Fri Mar  3 20:47:04 2017 from 192.168.0.103
-bash-4.1$ pwd
/
-bash-4.1$ echo "Tecmint - Fastest Growing Linux Site"
Tecmint - Fastest Growing Linux Site
-bash-4.1$
-bash-4.1$ history
    1  pwd
    2  echo "Tecmint - Fastest Growing Linux Site"
    3  history
-bash-4.1$ 
```

SSH Built-in Commands

Step 6. Create SSH User's Home Directory and Add Linux Commands

11. From the previous step, we can notice that the user is locked in the root directory, we can create a home directory for the the SSH user like so (do this for all future users):

```
# mkdir -p /home/test/home/tecmint
# chown -R tecmint:tecmint /home/test/home/tecmint
# chmod -R 0700 /home/test/home/tecmint
```

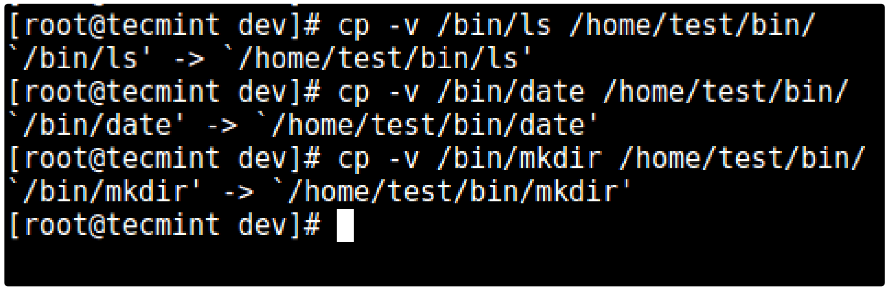


```
[root@tecmint dev]# mkdir -p /home/test/home/tecmint
[root@tecmint dev]# chown -R tecmint:tecmint /home/test/home/tecmint
[root@tecmint dev]# chmod -R 0700 /home/test/home/tecmint
[root@tecmint dev]# 
```

Create SSH User Home Directory

12. Next, install a few user commands such as ls, date, mkdir in the `bin` directory:

```
# cp -v /bin/ls /home/test/bin/
# cp -v /bin/date /home/test/bin/
# cp -v /bin/mkdir /home/test/bin/
```

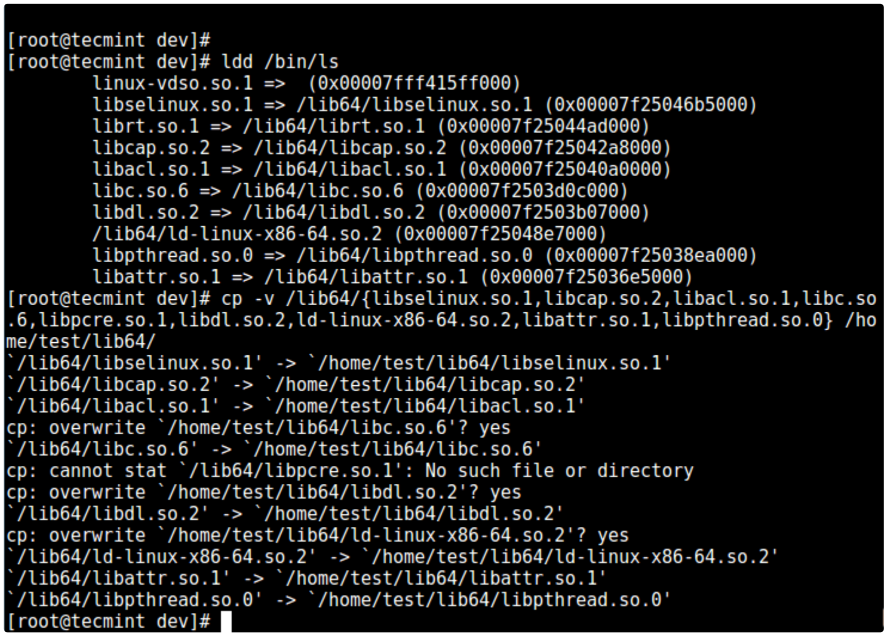


```
[root@tecmint dev]# cp -v /bin/ls /home/test/bin/
`/bin/ls' -> `/home/test/bin/ls'
[root@tecmint dev]# cp -v /bin/date /home/test/bin/
`/bin/date' -> `/home/test/bin/date'
[root@tecmint dev]# cp -v /bin/mkdir /home/test/bin/
`/bin/mkdir' -> `/home/test/bin/mkdir'
[root@tecmint dev]# 
```

Add Commands to SSH User

13. Next, check the shared libraries for the commands above and move them into the chrooted jail libraries directory:

```
# ldd /bin/ls
# cp -v /lib64/{libselinux.so.1,libcap.so.2,libacl.so
```



```
[root@tecmint dev]#
[root@tecmint dev]# ldd /bin/ls
        linux-vdso.so.1 =>  (0x00007fff415ff000)
        libselinux.so.1 => /lib64/libselinux.so.1 (0x00007f25046b5000)
        librt.so.1 => /lib64/librt.so.1 (0x00007f25044ad000)
        libcap.so.2 => /lib64/libcap.so.2 (0x00007f25042a8000)
        libacl.so.1 => /lib64/libacl.so.1 (0x00007f25040a0000)
        libc.so.6 => /lib64/libc.so.6 (0x00007f2503d0c000)
        libdl.so.2 => /lib64/libdl.so.2 (0x00007f2503b07000)
        /lib64/ld-linux-x86-64.so.2 (0x00007f25048e7000)
        libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f25038ea000)
        libattr.so.1 => /lib64/libattr.so.1 (0x00007f25036e5000)
[root@tecmint dev]# cp -v /lib64/{libselinux.so.1,libcap.so.2,libacl.so.1,libc.so
.6,libpcre.so.1,libdl.so.2,ld-linux-x86-64.so.2,libattr.so.1,libpthread.so.0} /ho
me/test/lib64/
`/lib64/libselinux.so.1' -> `/home/test/lib64/libselinux.so.1'
`/lib64/libcap.so.2' -> `/home/test/lib64/libcap.so.2'
`/lib64/libacl.so.1' -> `/home/test/lib64/libacl.so.1'
cp: overwrite `/home/test/lib64/libc.so.6'? yes
`/lib64/libc.so.6' -> `/home/test/lib64/libc.so.6'
cp: cannot stat `/lib64/libpcre.so.1': No such file or directory
cp: overwrite `/home/test/lib64/libdl.so.2'? yes
`/lib64/libdl.so.2' -> `/home/test/lib64/libdl.so.2'
cp: overwrite `/home/test/lib64/ld-linux-x86-64.so.2'? yes
`/lib64/ld-linux-x86-64.so.2' -> `/home/test/lib64/ld-linux-x86-64.so.2'
`/lib64/libattr.so.1' -> `/home/test/lib64/libattr.so.1'
`/lib64/libpthread.so.0' -> `/home/test/lib64/libpthread.so.0'
[root@tecmint dev]#
```

Copy Shared Libraries

Step 7. Testing SFTP with Chroot Jail

SHARE

+

14. Do a final test using sftp; check if the commands you have just installed are working.

Add the line below in the `/etc/ssh/sshd_config` file:

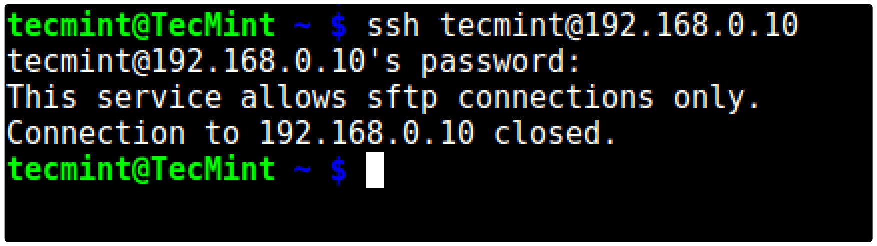
```
#Enable sftp to chrooted jail
ForceCommand internal-sftp
```

Save the file and exit. Then restart the SSHD services:

```
# systemctl restart sshd
OR
# service sshd restart
```

15. Now, test using SSH, you'll get the following error:

```
# ssh tecmint@192.168.0.10
```

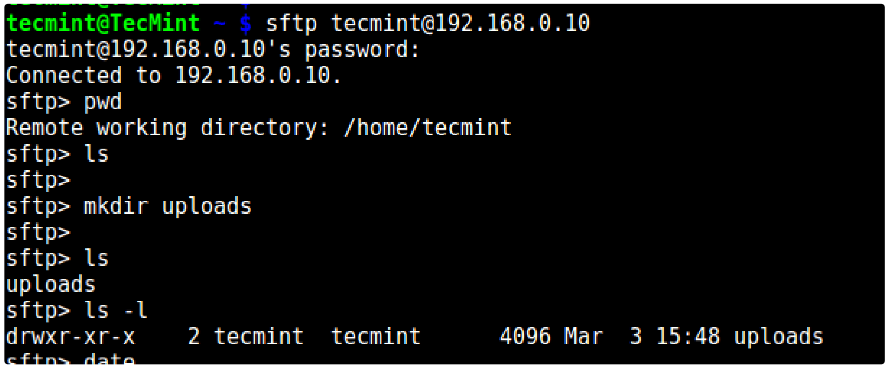


```
tecmint@TecMint ~ $ ssh tecmint@192.168.0.10
tecmint@192.168.0.10's password:
This service allows sftp connections only.
Connection to 192.168.0.10 closed.
tecmint@TecMint ~ $
```

Test SSH Chroot Jail

Try using SFTP as follows:

```
# sftp tecmint@192.168.0.10
```



```
tecmint@TecMint ~ $ sftp tecmint@192.168.0.10
tecmint@192.168.0.10's password:
Connected to 192.168.0.10.
sftp> pwd
Remote working directory: /home/tecmint
sftp> ls
sftp>
sftp> mkdir uploads
sftp>
sftp> ls
uploads
sftp> ls -l
drwxr-xr-x    2 tecmint  tecmint      4096 Mar  3 15:48 uploads
sftp> date
```

Testing sFTP SSH User

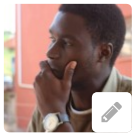
> **Suggested Read:** Restrict SFTP Users to Home Directories Using chroot Jail

That's it for now!. In this article, we showed you how to restrict a SSH user in a given directory (chrooted jail) in Linux. Use the comment section below to offer us your thoughts about this guide.

3. Get your own self-hosted blog with a Free Domain at ($3.95/month).
4. Become a Supporter - Make a contribution via PayPal
5. Support us by purchasing our premium books in PDF format.
6. Support us by taking our online Linux courses

We are thankful for your never ending support.

Tags: SSH Tips



Aaron Kili **View all Posts**

Aaron Kili is a Linux and F.O.S.S enthusiast, an upcoming Linux SysAdmin, web developer, and currently a content creator for TecMint who loves working with computers and strongly believes in sharing knowledge.

Your name can also be listed here. Got a tip? Submit it here to become an TecMint author.

PREVIOUS STORY

How to Change Root Password of MySQL or MariaDB in Linux

NEXT STORY

6 Best PDF Page Cropping Tools For Linux

YOU MAY ALSO LIKE...

FireSSH – A Web Browser SSH Client Plugin for Firefox

26 JUN, 2013

9

How to Install and Configure OpenSSH Server In Linux

5 Best Practices to Secure and Protect SSH Server

14 DEC, 2012

13 NOV, 2013

15 RESPONSES

Comments 15 Pingbacks 0

Iulian Murgulet March 25, 2017 at 4:57 pm

Hello John,

I can not say for sure it is possible (because I do not has a such case), but I guess, that is possible, because scponly is only a shell like bash. But if you can describe your test case I will try to give more help.

Reply

John March 24, 2017 at 9:06 pm

Can cronjobs or scripts run for the user configured to use scp only. The user is configured to use sftp/scp only and ssh is not allowed.
Thank you.

Reply

Aaron Kili March 27, 2017 at 2:20 pm

@John

As @Iulian has mentioned, try to describe your use case, it could be possible to find a solution for it.

Reply

Iulian Murgulet March 14, 2017 at 3:44 pm

Thx. @Aaron, I appreciate your remarks!

The link shared by you for Ahmed could be not useful in these days. At least me, I can not find likewise-open in the default repos for Linux-mint (last version). Maybe I am wrong ;) But for sure Ahmed can use SSSD: “The System Security

Services Daemon (SSSD) provides access to different identity and authentication providers”:

```
# apt install sssd-ldap sssd-ad sssd-krb5 sssd-ipa
```

After that he can integrate any LINUX desktop/server, in any LDAP/AD(ldap)/IPA(ldap), and maybe more others …. ! Then the rest of the tutorial(without the likewise-open part) can be used. And with SSSD, you can also have cache credential for any authenticated user, even if the AD/LDAP server is DOWN for some time. As a final word, I think that likewise-open is discontinued (if I remember correctly). In my case likewise-open has fail many years ago!

Reply

Aaron Kili ⊙ March 15, 2017 at 12:36 pm

@Iulian

Many thanks man for the heads up, will surely try this out.

Reply

Iulian Murgulet ⊙ March 15, 2017 at 1:48 pm

No problem, you can try it, sssd is very simple to setup and it is KISS(keep it simple stupid – sorry, no offence to anyone)

Reply

Aaron Kili ⊙ March 15, 2017 at 3:27 pm

@Iulian

Sure, thanks for always following us.

Reply

Iulian Murgulet ⊙ March 13, 2017 at 11:12 pm

Hello, this tutorial is ok to show how a chroot can be use. But from practical point of view is complicated, and does not scale.

For a scp run in a jail, is more simple to use scponly. For your test case, a webserver, we can use any container technology (lxc is one possibility), or even better kvm. But you know, each solution have good points, and bad points.

Any Linux admin must think what is the best for his particular case. This is the most important for me is ok the solution A or B? How I can reduce the risk for A and for B? I have the skills for A/B? I have the proper resources for A/B (time, servers, storage, and so on)?

Reply

Aaron Kili March 14, 2017 at 11:52 am

@Iulian

Your are right, from a practical point of view, implementing this may by be complicated especially when used with ssh, scp and other related commands. And also when you need to install additional commands for users and create a PATH for them to run commands without specifying the absolute path to the commands.

Therefore, it would effectively and reliably work in test cases for testing certain programs in an isolated environment on the system. Thanks for sharing your thoughts with us.

Reply

Ahmed March 10, 2017 at 9:33 pm

When I test SFTP connection.

```
"subsystem request failed on channel 0
Couldn't read packet: Connection reset by peer"
```

logs – /var/log/secure:

```
"error: subsystem: cannot stat /usr/libexec/openssh/sftp-server: No such file or directory
subsystem request for sftp failed, subsystem not found"
```

When I change sshd_config: (https://www.tecmint.com/restrict-sftp-user-home-directories-using-chroot/)

```
"# override default of no subsystems
Subsystem    sftp internal-sftp
#/usr/libexec/openssh/sftp-server
ForceCommand internal-sftp"
```

It's work for me. But, why user see all folder/files from jail ? ex: **bin/etc/dev** ?
"cd /" move him to /home/test.

And How Can I run this Jail with LDAP/geten passwd user from LDAP ?

Reply

Aaron Kili ⊙ March 11, 2017 at 3:05 pm

@Ahmed

Remember in this guide, we didn't block user from viewing files in the chrooted jail(which is the apparent root directory), but it is possible to configure this.

This guide can give you a fair start to using LDAP with Chrooted jail:
https://heitorlessa.com/sftp-jail-chroot-with-active-directory-authentication-832ebf93dfa8#.duimrfrmr

And we will create a guide for this soon.

Reply

Ldap ⊙ March 10, 2017 at 1:12 pm

Can I please auth with ldap ?

Reply

Aaron Kili ⊙ March 11, 2017 at 3:09 pm

@Ldap

This guide should give you a fair start to using LDAP with Chrooted jail:
https://heitorlessa.com/sftp-jail-chroot-with-active-directory-authentication-832ebf93dfa8#.duimrfrmr

We don't have a guide for this yet, however, we'll create one in the near future.

Reply

Iulian Murgulet ⊙ March 13, 2017 at 11:23 pm

Hello Ahmed,

Basically a chroot jail is useful only for simple application (read like with few dependencies) For your test case (ldap) is more simple to setup a RO (read-only) ldap server in a container or in a kvm guest.

This is my opinion, and it was working in my case for many years. ldap has many dependencies / libraries and is hard to make a chroot for this.

Reply

Aaron Kili March 14, 2017 at 11:44 am

@Iulian

Many thanks for sharing your experience with us, we'll look into this as you have suggested and i hope @Ahmed will as well.

Reply

GOT SOMETHING TO SAY? JOIN THE DISCUSSION.

Comment

Name *

Email *

Website

□ **Notify me of followup comments via e-mail. You can also subscribe without commenting.**

Post Comment

LINUX MONITORING TOOLS

- Install Munin (Network Monitoring) in RHEL, CentOS and Fedora
- Darkstat – A Web Based Linux Network Traffic Analyzer
- How to Setup and Manage Log Rotation Using Logrotate in Linux
- linux-dash: Monitors “Linux Server Performance” Remotely Using Web Browser
- nload – Monitor Linux Network Bandwidth Usage in Real Time

LINUX INTERVIEW QUESTIONS

- 15 Interview Questions on Linux “ls” Command – Part 1
- Shilpa Nair Shares Her Interview Experience on RedHat Linux Package Management
- 11 Basic Linux Interview Questions and Answers
- 10 Core Linux Interview Questions and Answers
- 10 Useful SSH (Secure Shell) Interview Questions and Answers

OPEN SOURCE TOOLS

- 9 Best Twitter Clients for Linux That You Will Love to Use
- 9 Tools to Monitor Linux Disk Partitions and Usage in Linux
- 4 Good Open Source Log Monitoring and Management Tools for Linux
- 11 Best Open Source Web Browsers I Discovered for Linux in 2016
- 18 Best IDEs for C/C++ Programming or Source Code Editors on Linux