Restrict SSH User Access to Certain Directory Using Chrooted Jail

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:

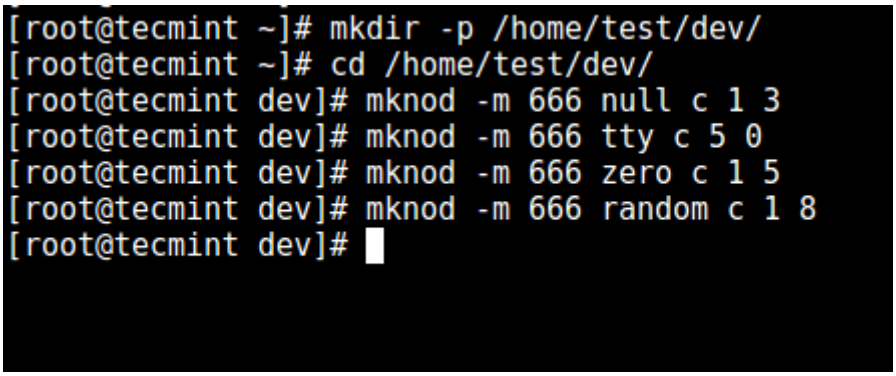
```
# ls -l /dev/{null,zero,stdin,stdout,stderr,random,tty}
```

```
[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 ~]#
```

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.

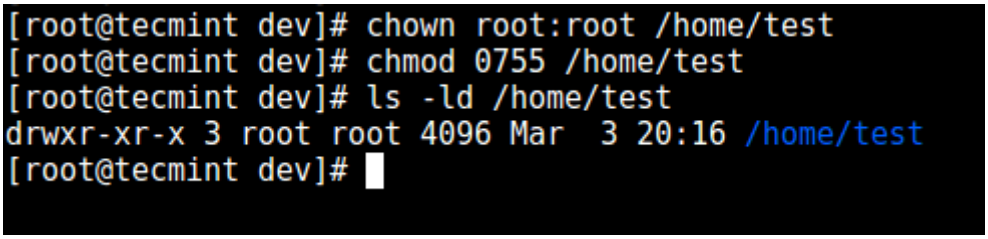
```
# mkdir -p /home/test/dev/
# cd /home/test/dev/
# mknod -m 666 null c 1 3
```

```
# mknod -m 666 tty c 5 0
# mknod -m 666 zero c 1 5
# mknod -m 666 random c 1 8
```



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:

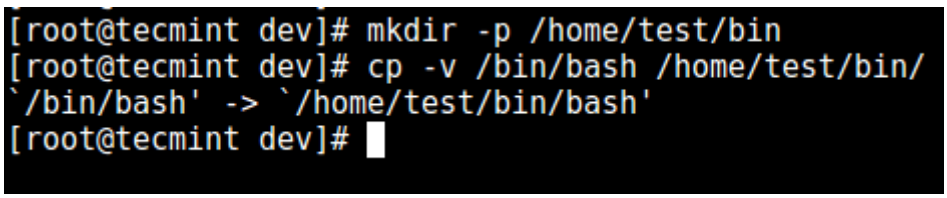
```
# chown root:root /home/test
# chmod 0755 /home/test
# ls -ld /home/test
```



Step 2: Setup Interactive Shell for SSH Chroot Jail

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/
```



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-linux-x86-64.so.2}
```

```
/home/test/lib64/
```

```
[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]#
```

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 tuxi
# passwd tuxi
```

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]#
```

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
```

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

```
# systemctl restart sshd
```

Step 5: Testing SSH with Chroot Jail

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

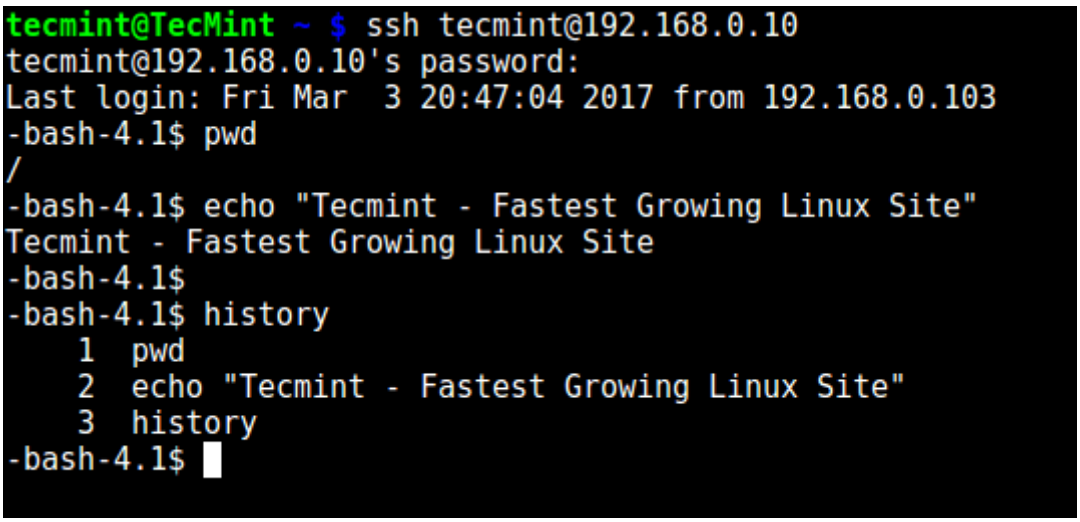
```
# ssh tecmint@192.168.0.10
-bash-4.1$ ls
-bash-4.1$ date
-bash-4.1$ uname
```

```
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$
```

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:

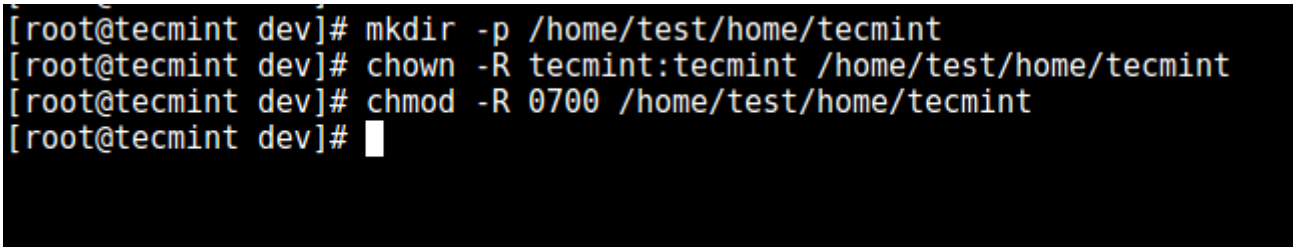
```
# ssh tecmint@192.168.0.10
-bash-4.1$ pwd
-bash-4.1$ echo "Tecmint - Fastest Growing Linux Site"
-bash-4.1$ history
```



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
```



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]#
```

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.1,libc.so.6,libpcre.so.1,libdl
.so.2,ld-linux-x86-64.so.2,libattr.so.1,libpthread.so.0} /home/test/lib64/
```

```
[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]#
```

Step 7. Testing SFTP with Chroot Jail

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
```

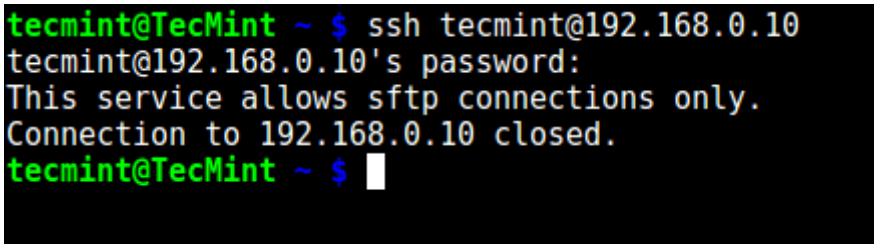
```
ForceCommand internal-sftp
```

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

```
# systemctl restart sshd
```

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

```
# ssh tecmint@192.168.0.10
```



Try using SFTP as follows:

```
# sftp tecmint@192.168.0.10
```