Chroot Jail Tutorial (with user and group creation and deletion, and SSH key logins)

sources:
http://allanfeid.com/content/creating-chroot-jail-ssh-access
https://linuxconfig.org/how-to-automatically-chroot-jail-selected-ssh-user-logins
https://krisko210.blogspot.com/2014/04/ssh-chroot-jail.html

what's a jail? just a way of “locking” a user or process to some portion of the disk space
this, of course, has the potential to restrict access to resources
it can be implemented just for running processes, local users, or remote users (e.g., SSH)
we often use it to setup customized versions of software (e.g., Linux)
we can create an entire directory hierarchy for the system in the jail!
here, we'll try it for SSH

first, we need to be root:

```
sudo -i
cd ~
```

let's make the jail:

```
mkdir /var/jail
```

**we need to put the shell in the jail (everything we want the user to access must be in the jail)
so let's see what dependencies bash has:**

```
ldd /bin/bash
```

and let's get them in the jail, along with bash:

```
cd /var/jail/
mkdir bin lib lib64
cp /lib/x86_64-linux-gnu/libtinfo.so.5 lib
cp /lib/x86_64-linux-gnu/libdl.so.2 lib
cp /lib/x86_64-linux-gnu/libc.so.6 lib
cp /lib64/ld-linux-x86-64.so.2 lib64
cp /bin/bash bin
```

let's go to jail:

```
chroot /var/jail/
```

and try stuff:

```
ls -al
```

[ctrl+d]

manually copying resources and their dependencies is time consuming, so here's a script:

source: http://linuxcareer.com
make sure the file (jail.sh) is in ~ (/root) and that it is executable (chmod u+x jail.sh)
watch the line breaks when copying/pasting from the PDF!

```
#!/bin/bash
# This script can be used to create simple chroot environment
# Written by LinuxCareer.com <http://linuxcareer.com/>
# (c) 2013 LinuxCareer under GNU GPL v3.0+

#!/bin/bash

CHROOT='/var/jail'
mkdir $CHROOT

for i in $( ldd $* | grep -v dynamic | cut -d " " -f 3 | sed 's/://'
| sort | uniq ); do
     cp --parents $i $CHROOT
done

# ARCH amd64
if [ -f /lib64/ld-linux-x86-64.so.2 ]; then
     cp --parents /lib64/ld-linux-x86-64.so.2 /$CHROOT
fi

# ARCH i386
if [ -f  /lib/ld-linux.so.2 ]; then
     cp --parents /lib/ld-linux.so.2 /$CHROOT
fi

echo "Chroot jail is ready. To access it execute: chroot $CHROOT"
```

now, let's try it with the script:

```
cd ~
rm -rf /var/jail/
./jail.sh /bin/{ls,cat,echo,rm,mkdir,bash} /usr/bin/vim
/usr/bin/whoami /usr/bin/scp /etc/hosts
```

add useful “special” files:

```
mkdir /var/jail/dev
mknod -m 0666 /var/jail/dev/null c 1 3
mknod -m 0666 /var/jail/dev/random c 1 8
mknod -m 0444 /var/jail/dev/urandom c 1 9
```

and let's add a new user to the system to test everything out:

the user will be sphincter; its password will be sphincter
we'll add a group to place all jailed users (chrootjail)

```
groupadd chrootjail
adduser sphincter
adduser sphincter chrootjail
```

we can check that the user is added:

```
ls -alh /home
```

now we copy the system's password and group files to the jail:

```
cp /etc/passwd /etc/group /var/jail/etc/
```

and let's remove users and groups that we don't need in the jail:

```
vim /var/jail/etc/passwd /var/jail/etc/group
```

remove everything except **root**, **sphincter**, and **chrootjail** references

now, let's add the new user's home directory in the jail:

```
mkdir /var/jail/home /var/jail/home/sphincter
```

and copy default shell configuration files:

```
cp -r /etc/skel/ /var/jail/home/sphincter
mv /var/jail/home/sphincter/skel/.* /var/jail/home/sphincter
rm -rf /var/jail/home/sphincter/skel/
```

let's also copy useful bash and vim configuration files:

```
cp /home/jgourd/.bashrc /home/jgourd/.bash_aliases
/home/jgourd/.vimrc /var/jail/home/sphincter/
```

we need to make sure that the new user owns it's home directory:

```
chown -R sphincter:sphincter /var/jail/home/sphincter
```

finally, we need the SSH server (if it's not already installed):

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

and we'll need to alter its configuration to jail users in the chrootjail group:

```
vim /etc/ssh/sshd_config
```

add the following to the configuration file:

```
Match group chrootjail
     ChrootDirectory /var/jail/
     X11Forwarding no
     AllowTcpForwarding no
```

restart SSH:

```
service ssh restart
```

and try logging in:

```
ssh sphincter@localhost
```

[ctrl+d]

notice that the jail shell doesn't recognize the new user, so let's fix that:

```
cp /lib/x86_64-linux-gnu/libnss_* /var/jail/lib/x86_64-linux-gnu
```

and try again:

```
ssh sphincter@localhost
```

play around!

what about denying password logins and only allowing keys?

let's create a key on the client (we'll need to be logged in as the user we wish to allow access to):
when prompted for a password, just press Enter

```
ssh-keygen -t rsa
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
```

finally, we can change the SSH configuration:

```
vim /etc/ssh/sshd_config
```

change the following lines (if necessary) as follows:

```
PasswordAuthentication no
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile %h/.ssh/authorized_keys
```

and restart SSH:

```
service ssh restart
```

and try again:

```
ssh sphincter@localhost
```

lastly, let's undo everything that we did:
first, let's remove the jail:

```
rm -rf /var/jail
rm ~/jail.sh
```

then, let's remove the new user:

```
deluser --remove-home sphincter
```

also, let's remove the chrootjail group:

```
delgroup --only-if-empty chrootjail
```

finally, let's remove the SSH server (or fix the configuration as needed):

```
sudo apt-get remove --purge openssh-server
```