***Problem 2***

**Figure out why the “passwd” command needs to be a Root Set-UID program. What will happen if it is not? Login as a regular user and copy this command to your own home directory (usually “passwd” resides in /usr/bin); the copy will not be a Set-UID program. Run the copied program, and observe what happens. Describe your observations and provide an explanation for what you observed.**

**Answer1**

The passwd command is belonging to the root as set-UID. As learnt in the class, if the setuid is set on an executable file, then any users able to execute the file will automatically execute the file with the privileges of the files owner. This allows the system designer to permit trusted programs to be run which a user would otherwise not be allowed to execute.

After coping the passwd to the own home directory, we can see that the passwd program has lost its setuid bit, and the access is just -rwxr-xr-x 1; I believe this prevents access are not getting transferred down.

**(b1) zsh is an older shell, which unlike the more recent bash shell does not have certain protection mechanisms incorporated.**

**Login as root, copy /bin/zsh to /tmp, and make it a Set-UID program with permissions 4755. Then login as a regular user, and run /tmp/zsh. Will you get root privileges? Please describe and explain your observation.**

**Answer2**

When we copy the zsh file as logged as root and grant the set-UID, the access prevailed when we log as a normal user. We can see that the file zsh under /tmp/ folder has the set-UID enabled, -rwsr-xr-x.

**Commands used**

[10/11/20]seed@VM:~$ sudo su

root@VM:/home/seed# cp /bin/zsh /tmp

root@VM:/tmp# chmod 4755 zsh

root@VM:/tmp# exit

[10/11/20]seed@VM:~$ cd /tmp

[10/11/20]seed@VM:/tmp$ ./zsh

[10/11/20]seed@VM:/tmp$ ls -l /tmp/zsh

-rwsr-xr-x 1 root root 756476 Oct 11 22:35 /tmp/zsh

**(b2) Login as root and instead of copying /bin/zsh, this time, copy /bin/bash to /tmp, make it a Set-UID program. Login as a regular user and run /tmp/bash. Will you get root privilege? Please describe and provide a possible explanation for your observation.**

**Answer3**

No, we did not get the root previldege.While copying the /bin/bash and giving the set-ud permission, the access is not granted like zsh.

Bash seems to have been protected. /bin/bash has certain built-in protection that prevent the abuse of the Set-UID mechanism.This would prevent the hackers to not get a hold of the accounts and resources.

[10/11/20]seed@VM:/tmp$ sudo su

root@VM:/tmp# cp /bin/bash /tmp/

root@VM:/tmp# chmod u+s bash

root@VM:/tmp# exit

exit

[10/11/20]seed@VM:/tmp$ ls -al bash

-rwsr-xr-x 1 root root 1109564 Oct 11 23:24 bash

[10/11/20]seed@VM:/tmp$ ./bash

bash-4.3$ id

uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)

bash-4.3$

**(c1) In most Linux distributions (Fedora and Ubuntu included), /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:**

**login as root**

**cd /bin**

**rm sh**

**ln –s zsh sh**

**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;**

**}**

**Login as root, create a new directory /tmp1 and set it to have the same permissions as /tmp, write this program into a file named bad\_ls.c, compile it (using gcc –o bad\_ls bad\_ls.c) and copy the executable as a Set-UID program into /tmp1 with permissions 4755.**

**Is it a good idea to let regular users execute the /tmp1/bad\_ls program (owned by root) instead of /bin/ls ? Describe an attack by which a regular user can manipulate the PATH environment variable in order to read the /etc/shadow file.**

**Answer**

[10/13/20]seed@VM:~$ cd /bin

[10/13/20]seed@VM:/bin$ sudo su

root@VM:/bin# ls -l sh

lrwxrwxrwx 1 root root 4 Jul 25 2017 sh -> dash

root@VM:/bin# rm sh

root@VM:/bin# ln -s zsh sh

root@VM:/bin# ls -al sh

lrwxrwxrwx 1 root root 3 Oct 13 12:47 sh -> zsh

root@VM:/# mkdir /tmp1

root@VM:/# chmod --reference=tmp tmp1

***Problem 3***

**Show how someone who is on the no-fly list can manage to fly provided that boarding passes could be generated online (as an HTML page) and then printed. Please provide a step-by-step description of the attack.**

1. Buy a ticket online, using a prepaid credit card purchased with cash, for a fake passenger name. Make sure you do not use the name which are already on the no-fly list.
2. Check in online 24 hours before departure and print out this boarding pass.
3. Edit the HTML of the boarding pass and put your real name to make up an identical boarding pass same as the boarding pass printed in step#2. Print this out as well.
4. Go to the airport, and present the boarding pass with your real name and your real ID before entering the departure area of the airport, where passengers go through a security check and they have to present a government-issued ID and a boarding pass. As the check done is by visual inspection, they will check that they match, and then let you through the checkpoint with a minimal search.
5. Before boarding a flight, present the fake-name boarding pass to the gate-agent when you board the flight. As the check done here is to ensure the scanned information from the boarding pass matches an existing reservation in the system, so everything will look good.

**Which additional security measures should be implemented in order to eliminate this vulnerability?**

1. Provide the security employees scanners/computers at the gate, so that they can verify the validity of the boarding passes along with the ID proof when someone reaches at the gate.
2. Do NOT allow people to print out boarding passes online.
3. Facial ID scan along with boarding pass scan at the gate to find out any discrepancies.