长安“战疫”网络安全卫士守护赛部分wp

原创

元元努力向上 于 2022-01-08 22:37:49 发布 2573 收藏
分类专栏： 笔记 文章标签： 经验分享

本文链接：https://blog.csdn.net/m0_56194555/article/details/122385961


笔记 专栏收录该内容
4 篇文章 0 订阅
订阅专栏

摘要：长安“战疫”网络安全卫士守护赛部分wp

然后就是朴实无华的取证那个题 不知道是大小写的原因还是啥交不上，无字天书卡到最后那个长得好像摩斯密码的地方，收获满满，继续努力。

misc:

八卦迷宫

西安加油

binary

密码：

no_cry_no_can

no_can_no_bb

RE

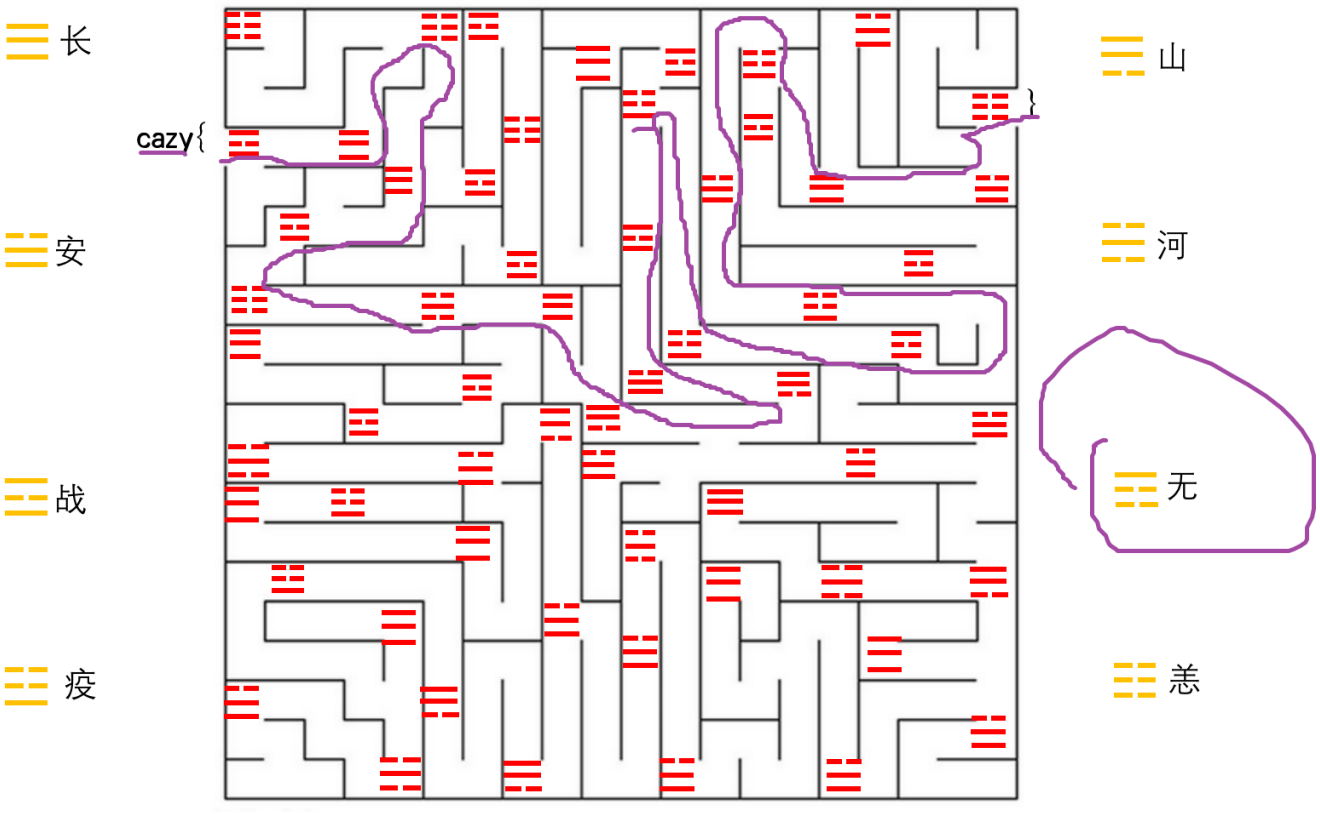
hello_py

cute_dog

misc:

八卦迷宫

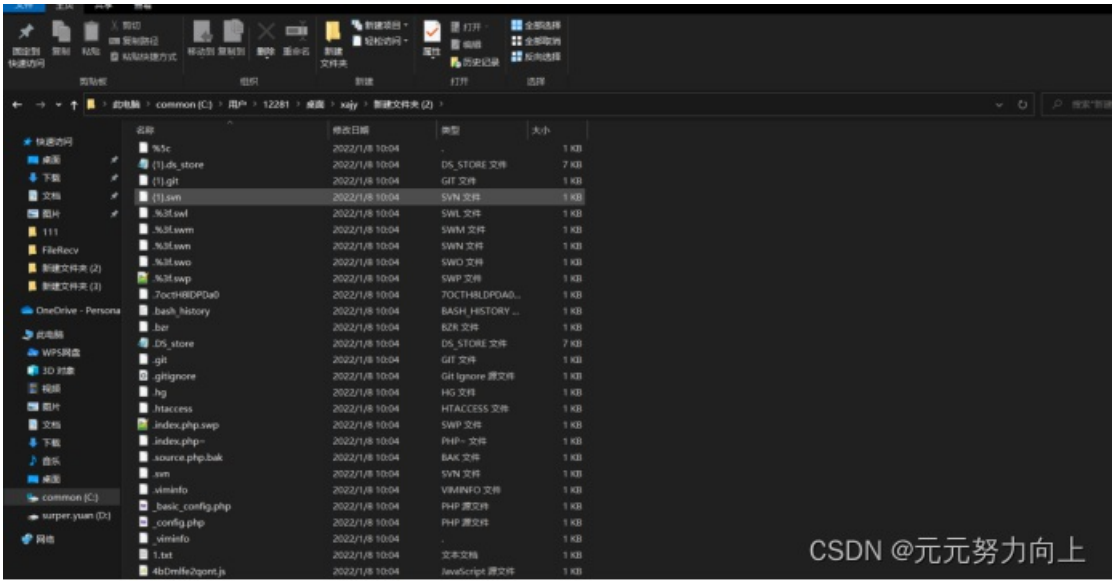
走迷宫，路线即是答案，但是要把中文转化成拼音才能交上flag，哈哈哈，当时还在那疑惑了：



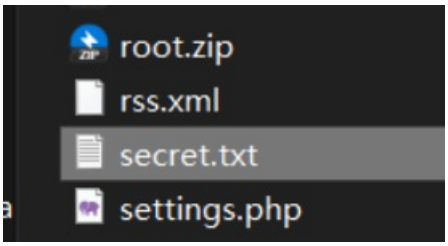
cazy{zhanchangyangchangzhanyanghechangshanshananzhanyiyizhanyianyichanganyang}

西安加油

打开流量包，然后导出http到文件夹：



然后里面有一个secret.txt：

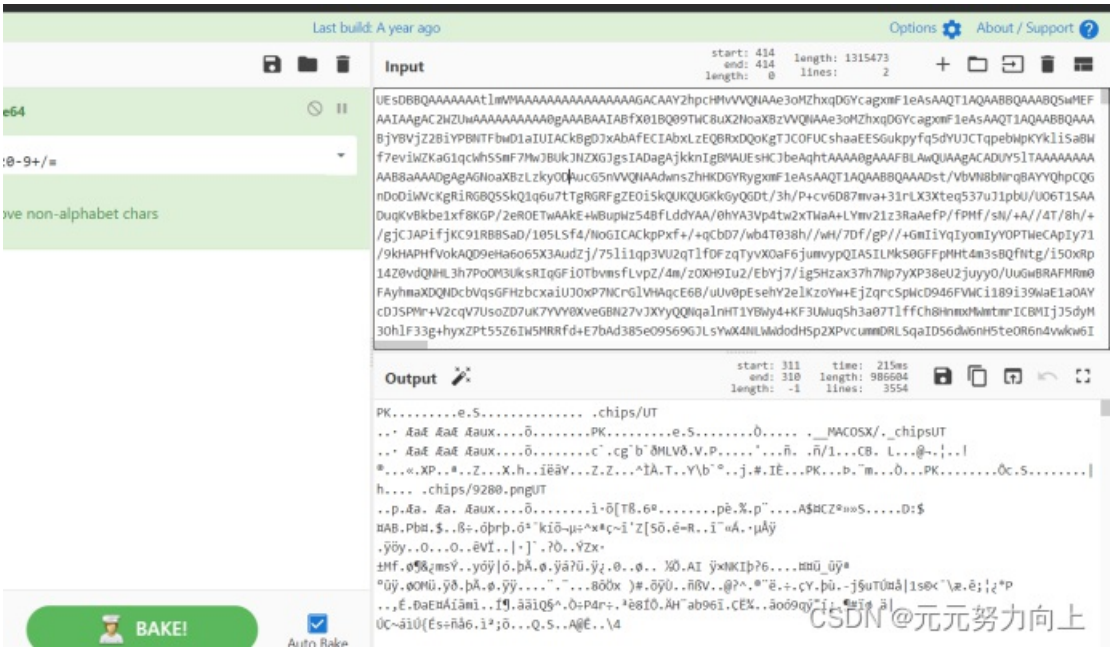


打开：

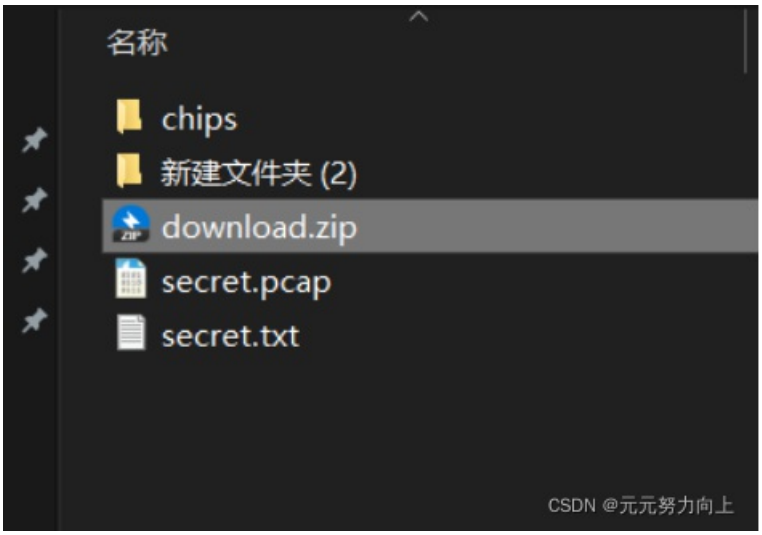
UEsDBBQAAAAAAAAtlmVMAAAAAAAAAAAAAAAAGACAAY2hpcHMvVVQNAAe3oMZhxqDGYcagxmF1eAsAAQT1AQAABBQAAABQSwMEFAAIAAgAC2WZUwAAAAAAAAAA0gAAABAAIABfX01BQ09T
Y2elKzoYw+EjZqrc5pWcD946FVWCi189i39WaE1aOAYcDJSPMr+V2cqV7UsoZD7uK7YVY0XveGBN27vJXYyQQNqaInHT1YBWy4+KF3UWuqSh3a07TlffCh8HnmxMWmtmrICBMIjJ5dyM3OhlF33g+hyx2
6Vzs18mtPUenF3YnTXqpUbOrtqR/Nej07XPey27PME2hzcsrDafoX4v9RQR759zpi9bu7JrYvB7ZrWz29KSk6BK5XXK8SvcZjknSUWq+ULM0MjLy8vISLvZscm1tzX6mqrKiQkCwEma7621cgM0Pqj7fTADAy
l6mvFnLYFU6213oZoEO7cjArXYqSE7K82HrihhWSwkZ0/zaV0fhsWzrpBiDKUrm1RgSH9nZ3E4haMelTCLSrJ+76hPLx7OlvV+W2GhH96oyc+O+/OxfdJTil7B3MNDod0LDRVCSCemyDvE4gGRlhxhDxHus9
6CIL3XGsXtihjOy2qsvWLO1edPZ2+Rb4MKrbVlY1t6vB0G3yQGTYbWO88/f+jwMrDLIwgTvcvCyeAGCPCq63JNaAlWAJAHGbvneC9/qGpg2tjXLW4VB0fUfT+BztbTOuGp5xd+XgeiOVO0N/xvFXixD6NoS9
PHaIYLBCmANVpTm2A+Ux1EFBEtVJVeK97I2AjLHov5WZg65vaAnf7T9CVapMBjjfWVlrFSLcHrGePIWUyNYgd/MUOd8fuTg99Whf0nqeQf1nVmvSz7HKe22gNNWziuUkIqKEDU4vZDxhu0uyt8FdSHobYy7
dGQpQj7yHJuKKcPNSVOQdQ9ABSW2GfhFx6ZSz/LM2U1z7JW76Ff/ZuXlfE4wbDVCVY+vszB0Wfr6+rFFWbm584s/H82yDY3OFLta3nx8VaUrJ/9rackENcxcrG0DENt+9ZOt0//00c+ZXqi+oVdEIcNSCWxV
/OK0dJI5ijGIlY1WySeoZ4SxPDsZyHNL6dBnFUtbQSZryQZr5qyhMr6l22EM44s+LuRK4HOk8KzCeJNIhckrMB0QH+N2+f6jXDdyjF6PVwTsAYPSV3hAgrcHHD8od6pE6jleL9+8mGwjz3wrqj8ZDyhnMYgE4C
00KMNeNpsnl3V2rb96OWfti1X26tKm/ErpNT9GtN/luok96WAaT7X3UTBLhjY50FB5CPwTkmqkUYMyCEGa13Qk+jxUuZppCIS0+feRx8b2o/kLzuDTzSutnoxZ0sLCwMZgojqIPUojo6EqdJgrK7RaiVGKyzGV
PbDX9ildTNiTGm1aGRfUIlvpINvlOJrw1Xe8XkZeqyxOf4sUYKgDgD1KMjaenfteWzP+8UObtcbclpJMMt+xRuc0Aj6YMfnjS9cRboxmSoocB01jZedaIZGymkYWPRzJ+vW16ywKXDlVENZxxlw0Gb6nXvyAI4K
mlYSefqr7iebym4R8keBcqyjwkLRiuR2UyqtRSU1OxmE93LZW9DyKDPN7CV/Y/u6RBiAjwXXQNDNTEtAyMyToiSkQqQn+PotiHMqXk/W6iax40MzBGN+xGiYAZqkRJM1nHBTSNmXWVMXZoRMApq0z90/
ZC95LSUthE2KgxtN/vrUj4uP+SUVeQVPB1Mtz6VOkzS9oFARtdpQ/71wN91ZP93ZsgYfqb+lcO5N3vPEmlvCNGFyx95EkZGuGlb+9OoDdMzeU+k0uLMnhcSkP2uIsFk/gzEdcdsMAKADvIORhih8M07O1QI
fezedGl6K1aRzRv+WvCS0q+02ZoQGC4t0w1my/QJNWvi8BG1uVUtONJ7B5x3hZ+P9OdUUPtikDxZn7s4y1g06wee4c3ye6ExRmq2Al5KKAJN/x/yBvwDd7iftCPvlxtqkKSdgJKU1TOp91/qvfhogiO7jngG40
iAw+rRLYhFqdd+309PTP4ferX0VmJ872nj82hiSxucNNtZp/7HRXNVapfBDevdLILXIiOBX6bR+jSgE8sD8ucR6ilqOu+kvhTXo3xJYZmle4WK9f/642loW0iLVUlp/8A7Y1QnYvHsNy1qnhmF88uAQ93o+OdJLv
ePwsxZCjNwFFL2SPDugfLvq7XXZi6hudQmK011QrSEPdoJzsvjTYGf1Rf4fKBp+KBgzG+7prYGr7X33/Q0IWOHpiVkYeVqa0Is1/bJPjkVrrkidtsLSopWLoo8A3EsdasuKdY2Kkq6PbJf149mebGzbAOhbterBpcv
N0v+ZNttzdA7IEA0YaXdbydyyf8kV0vuoIde/ubb0CyilSrNvu3h8/0/TSuPwKS6dTeDNhbMKlY58m8OBagmbaFhqmpDoura4IJCrcgXdzVKYPI1HWmWZGuuVTi5UcYPbmgxVTOxggo2lvhEmVTHmfqcojEl
Ap3H9NCGm2O8gW0Yr42rcGKm3LmvryVYn0Wwx0APn6JjMHoM2A8kVN5DnKPl4e+dXgnGDBzUzJubPzXWjh2kzjlyi1FKBd0VeU+zLUFDDUAAKRiRmcal6c2RyZVfZZaAcPXu7h7JhsA3fB1/2D7V9lglJxC
1qnzEoJc0fyHHM+Q5QRw/qvvXUrsd5nih0aCoTXTVBx4XhM7TvI2PpuhjdWPvGpeVEsGXli3STAnSn3SO8oLde/io+ySBqTLklOyAvDNvR32m/Qf40a6zwJOsYG7U2xt07sevStlde47oZROi189T7fPr23S7npjb
49FhdmxRUwhzgPsZWlGOWayGzSTgYNLkBhd47mZywQzDaZYyEXfaiNHFBPy7SEyYW6gccW9b1ee7u5NVcrG8Hjk4GGpYfA+OpunTYnkp7XE5EqmcfELWPFjd7no1IJdGHiZB4lgXxKo8XRiIG52Q43d91F
uLk9BysWel6xXg3rP8591y+YBLKoyQjkiAUB0s8mEeiiykce2z02EgHbB9zzO7vr6x8G6gace6A/v1aokcsmbSZXUwQFn+vgdSlEdaAySlB6TmkhPHiNPlc7nktlkky48ay0wN/Z9CGi+zpI7DtwFl50tDaToBoqYv
v966ua6+ubg5/G6eeXp6uui1Eu/RzJL+tPv3/np+s479QbZWAOmqxhOR7oSs8yVHJZWkyRYGz/IWR/nBz+lo9EfamXsnEs/YSOYF8P/EgfJQh3h/x/V6GN2DE2ExvcMaRFE2IcqQIXBEYeB5ehwMAlNMnaOG
zV7sUM3bICXDOqL1P9FHmRRsi5eLnsa4neL2p4iElLKGbzXzZtgSeIpdgON7TM2XtrbHjAkrUgkrn5+/P3nyhjEJEgwyIMIaBKyWAABjZPJU71k3x2BiggzSZ6dYk1bZ2wwH3cTMk01F/O5BjaTWasd0kLAzy2qp
dJowOwKaFaLlcpVZ+L4/Gl6LomiOokmeSikhhzMnT126dCkv0mSapdog4qDTvX37dlaYYyeOV5r1T7z4yfX19T/46h+eOnOQgqA0v5CiKJgtgsQZZD7rk8ADawdHX3j+iTtv/Oj1P/nKbvcwUPLL/4u/1m7V//l
cBi/d/P+x85ezbVDRY8ATXaSyDmUAoU1o4Pde2+86Y07nk9xHl4mo+9++7vf/bNvV6NwuVH+4s98XgZ+qVRqH7SLLFeBGAyG5XIZ2GqdozUhQjroHj1cr12ZNywsABEDspnxsAwG1dypcwB3b2xv3Hvq/
HCzlwH/v2jsHu1vHVtfm5ubu3r03N7dQCaKjvSM/9EzupMLRaJDnuR+XBjwfH+SiMqKSzx9Z9cVHmd4heNX5+urpPNmKBAWRb52RRKtLy1sPH1gH5blyzh+Op1nQevqlTyVOACITGPCklMAGCEFI0NPp9
j4rpcNDdq4GDjxjjR5QisgMwKDGlD7vDXmjXgKJqddDfuf/gwYvPv/D4Y48Nh+P1dm9cQH3tdGZdAaxoNoUL2LF1zpNEyXj79oeBc3maQPcgPXgoq9Wq1vrhw4fW6c7BkdY6nU6VUktLCwcH+8YWvU7nx
mlvb24vLSxceu3y/0p0yoBSCiJGRmUgwg7agjCuG3fUb750+uTZWWgr3pVc+Lv/V7/4Ph4eHnU6n3++z1cbqxaV5ZI5aWXTosjSNytHVp54JI//73/9uv9+fJMZLS0snVk8uNuc279299+BefzBJWU5ldPUXV5
Zj44OvzRa6/Nt1pW6/vb2+12O8uyY6trX/rSl/7oq3/YO+ym07EuiqxlYq5EcXlxtXHi4uN9zaIUpVkuSAHPRgZgYaCiJAz3j+7deOXjLzQ9ev2tHzeqtbVzlxfnF4bD4Ztv/RjAxaXQ88Miy7/2//s3m5sPO52OMYbI
AIMFFhIRwHfODI7+x//yvyhPu7WwUq1WpSeY8dpb7zjy9WSF3og+OaNG/3OoFGr7e9uPdi4H8fxuXPnut1unucAbnFxPgiC/ElmE9gBUmZO0ik4ufHOX1565rmsEo4N5AaAQQF44Ep29NqPvi3QGSIlha88
0BIASiVlvvbf/j/f+JOvml+RUqHMBj1nc2tNcPBRAlRiYLhaOiQyuWqx3J1bdm44uVPv/z441eazWae5/v7+xsP169duzZ7o2C/3/u5n/u5+fn5999/v1artQ/33/jha4WlDqH89Kc/IxindY56evvHPywVl8/ICDxzU
MC5wbmdVVA0AB3CexmEcoMZhxqDGYXV4CwABBPUBAAAEFAAAAGNgFWNnYGJg8E1MVvAPVohQgAKQGAMnEBsBcRUQg/hrGlgCjiEhQVAmSMeVjB2ZBUBKiEBnmzs/USyvjnEnlfXyvHLErMKBnMS
VoawRgkXOtHDR82KPG3WzCDCZEulpXuyFXEZDEzZORBWeoeIHtY84cifezyCMDQ0vI0V682+518XhLeUxwylKtB8QWvIvDqa3y6mDMPmG8nM7xZ4QgLbSbBYUZ45UlSPnhbQBZFJF5UW2YbDzuuDsdo
iCLJXQCsAYaCsrwjoliMDIjAq/V5XGsN4l0facZTUkXUF5ube6fdwx7273PFva7uWCqW4P1JQWUqLmN/lpZYwcdHDQ+P6GH9ON32PaG7KqRaP6hr9m3MOW3/alRKdfmnD7S7UriLzvbW1uOR3hG9WdHlo

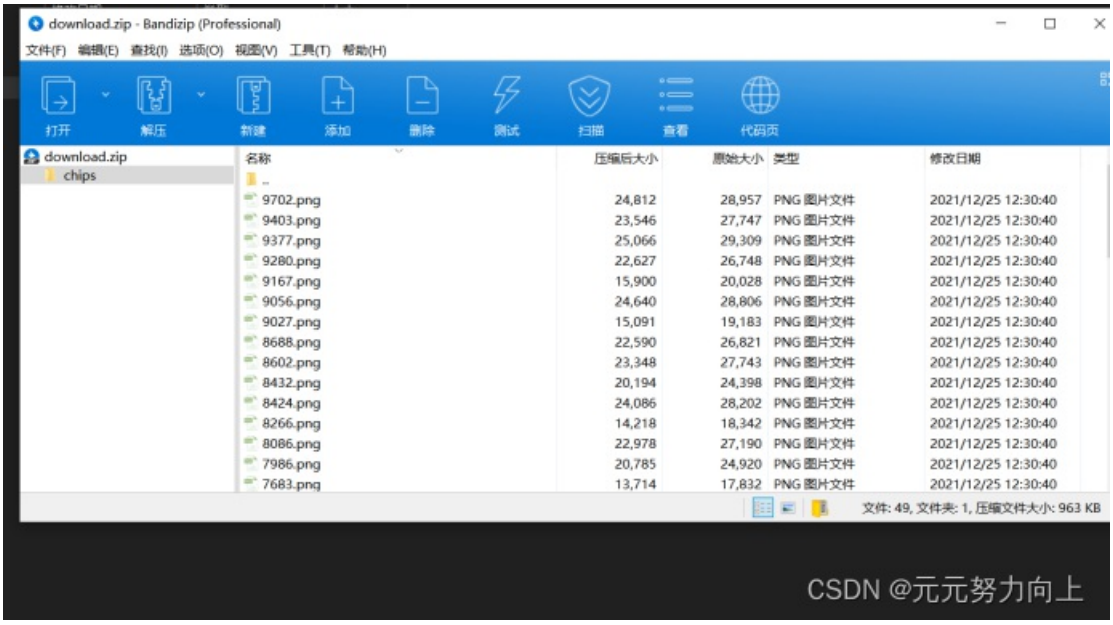
很明显base64

解密：



是一个压缩包，导出：



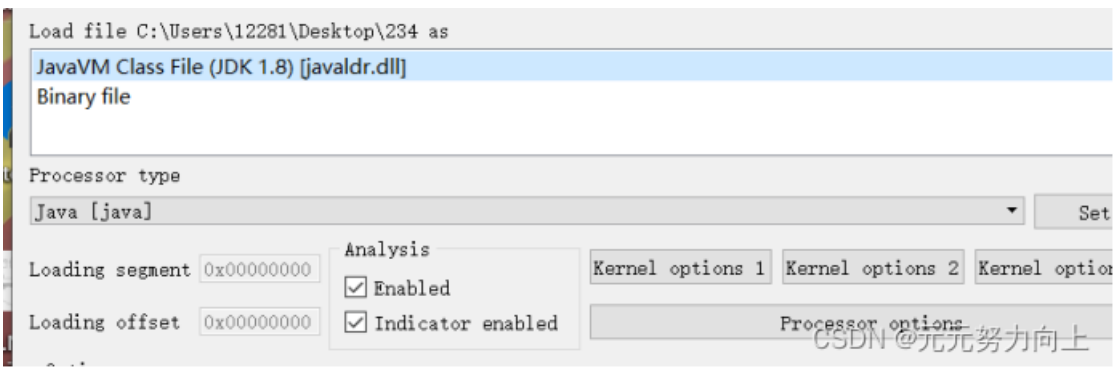


都是照片，一张一张看发现每一张都有涂鸦，拼一块就行：

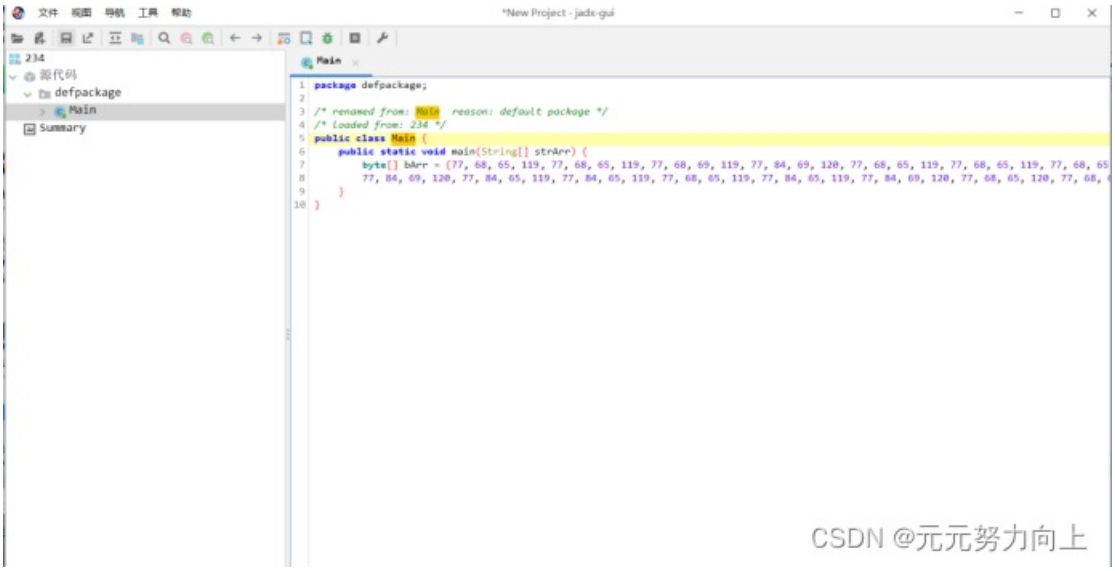
cazy{make_XiAN_great_Again}

binary

给了个234的未知文件，看题目是说是二进制，用ida打开看看：



Javavm，Java逆向，用jadx-gui打开：



只有一个主函数，里面是一个数组

思来想去的说是二进制，那么把这一些数都转换成二进制数据流得到

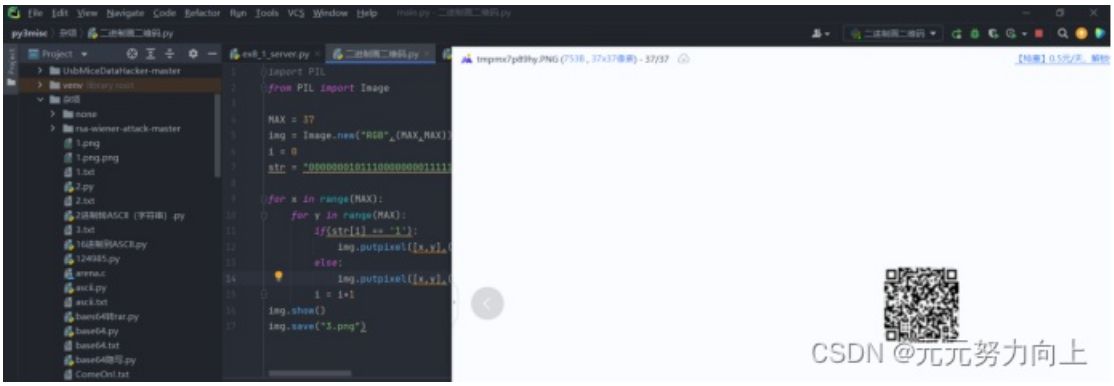
```
"00000001011100000000011111101110000000001111101011010101111100011101101111100100010100001111000111010110110100
```

然后试了好多种方法最后在网上找资料的时候找到大佬的二进制数据流转化成二维码的脚本：

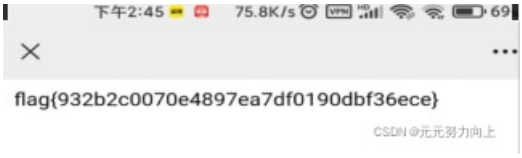
还得是大佬哈哈哈

```
mport PIL
from PIL import Image

MAX = 37
img = Image.new("RGB",(MAX,MAX))
i = 0
str = "00000001011100000000011111101110000000001111101011010101111100011101101111100100010100001111000111010110110100

for x in range(MAX):
    for y in range(MAX):
        if(str[i] == '1'):
            img.putpixel([x,y],(0,0,0))
        else:
            img.putpixel([x,y],(255,255,255))
        i = i+1
img.show()
img.save("3.png")
```

得到图片：



然后扫码得到：

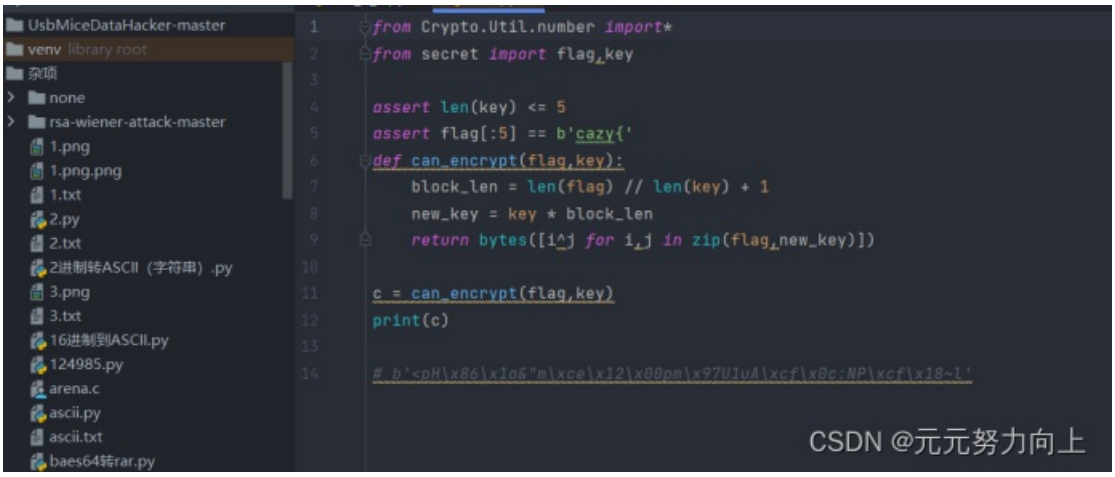


cazy{932b2c0070e4897ea7df0190dbf36ece}

密码：

no_cry_no_can

下载下来打开是一个python脚本：



分析一下 就是key有五位 然后这个加密函数的功能就是 将flag里面的每一位与key按位进行异或5个一循环，那么要先求出来key，因为给出来flag的前五位cazy{ 和加密后的c

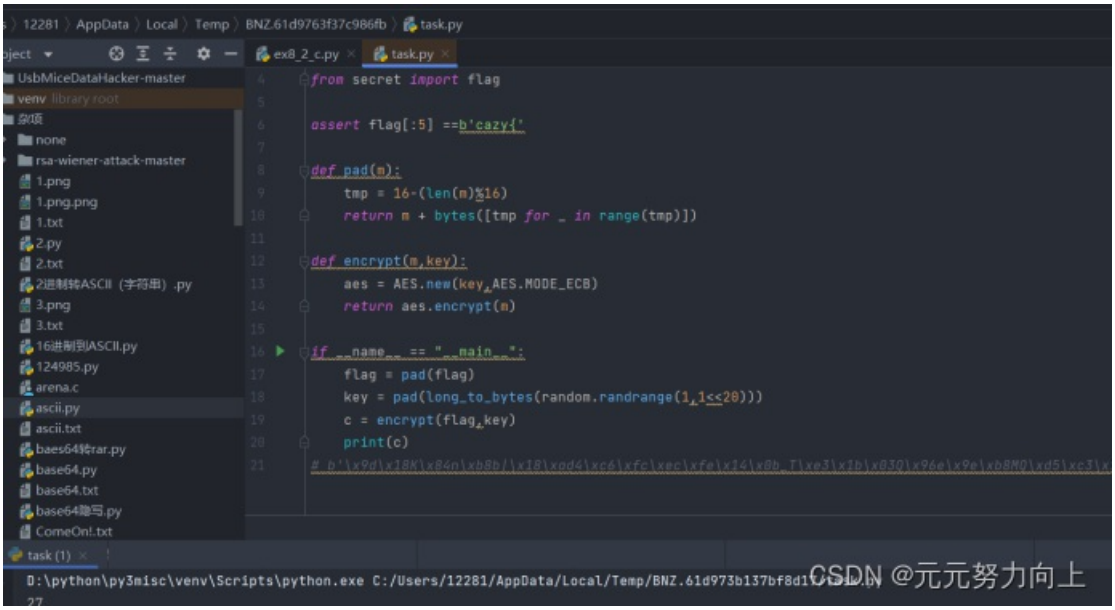
很容易求出来key，求出来key之后就可以写脚本了：

```
s='cazy{'
a='<pH\x86\x1a&"m\xce\x12\x00pm\x97U1uA\xcf\x0c:NP\xcf\x18~l'
print(len(a))
key=[]
for i in range(len(s)):
    key.append(ord(s[i])^ord(a[i]))
print(key)
m=''
for i in range(len(a)):
    m+=chr(ord(a[i])^key[i%5])
print(m)
```

cazy{y3_1s_a_h4nds0me_b0y!}

no_can_no_bb

下载下来打开依旧是一个python脚本，很明显的AES加密



这个是多次（2**20）加密直接写脚本爆破：

前面不变后面把加密函数换成自己写的解密函数，结果如下：

```
mport random
from Crypto.Util.number import long_to_bytes
from Crypto.Cipher import AES
def pad(m):
    tmp = 16-(len(m)%16)
    return m + bytes([tmp for _ in range(tmp)])
c = b'\x9d\x18K\x84n\xb8b|\x18\xad4\xc6\xfc\xec\xfe\x14\x0b_T\xe3\x1b\x03Q\x96e\x9e\xb8MQ\xd5\xc3\x1c'
for i in range(1,2**20):
    key = pad(long_to_bytes(i))
    aes = AES.new(key,AES.MODE_ECB)
    m = aes.decrypt(c)
    if m[:5] ==b'cazy{':
        print(m)
```

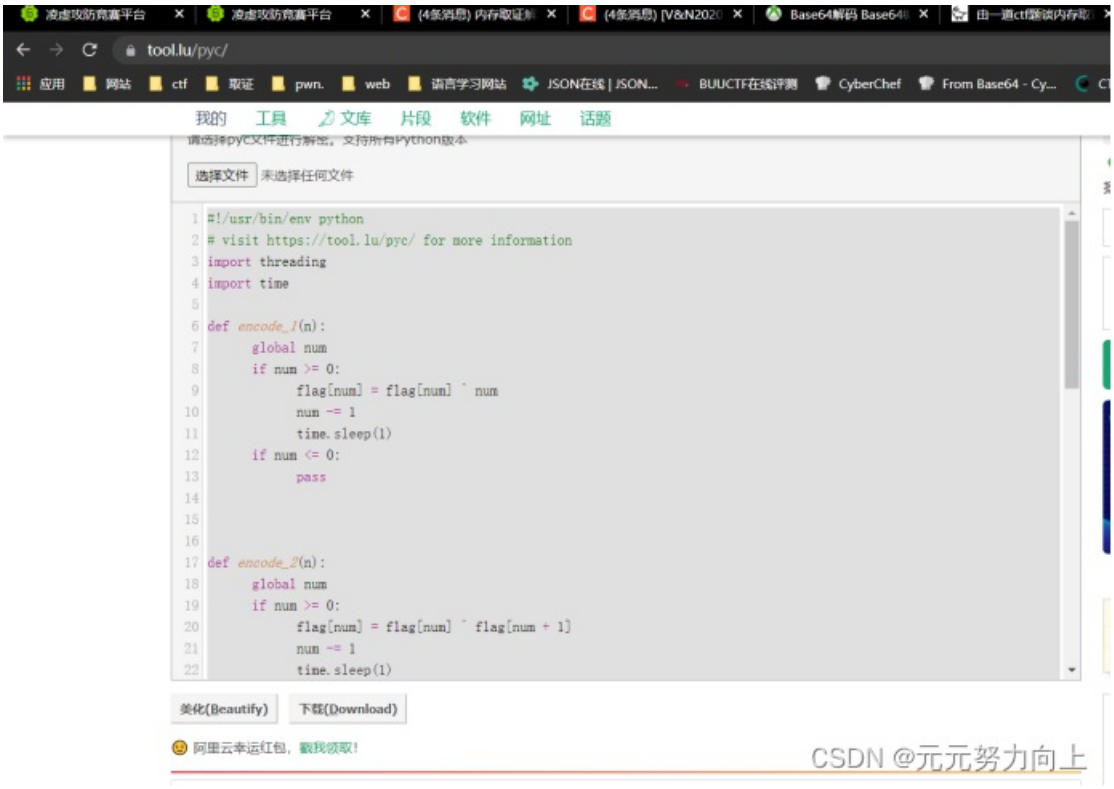
cazy{n0_c4n,bb?n0p3!}

RE

hello_py

下载下来是一个.pyc的程序，在线反汇编一下

得到：


tool.lu/pyc/
我的 工具 文库 片段 软件 网址 话题
选择文件 未选择任何文件
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
import threading
import time
def encode_1(n):
global num
if num >= 0:
flag[num] = flag[num] ^ num
num -= 1
time.sleep(1)
if num <= 0:
pass
def encode_2(n):
global num
if num >= 0:
flag[num] = flag[num] ^ flag[num + 1]
num -= 1
time.sleep(1)
美化(Beautify)
下载(Download)
阿里云幸运红包，戳我领取！

```
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
import threading
import time

def encode_1(n):
    global num
    if num >= 0:
        flag[num] = flag[num] ^ num
        num -= 1
        time.sleep(1)
    if num <= 0:
        pass



def encode_2(n):
    global num
    if num >= 0:
        flag[num] = flag[num] ^ flag[num + 1]
        num -= 1
        time.sleep(1)
    if num < 0:
        pass


Happy = [
    44,
    100,
    3,
    50,
    106,
    90,
    5,
    102,
    10,
    112]
num = 9
f = input('Please input your flag:')
if len(f) != 10:
    print('Your input is illegal')
    continue
flag = list(f)
j = 0
print("flag to 'ord':", flag)
t1 = threading.Thread(encode_1, (1,), **('target', 'args'))
t2 = threading.Thread(encode_2, (2,), **('target', 'args'))
t1.start()
time.sleep(0.5)
t2.start()
t1.join()
t2.join()
if flag == Happy:
    print('Good job!')
    continue
print('No no no!')
Continue
```

然后分析一下这个python脚本的功能

就是对输入的flag进行异或 下标为奇数用自己异或下标，下标为偶数则用自己异或下一个，然后和Happy比较

逻辑很简单

逆向过来就行，脚本如下：

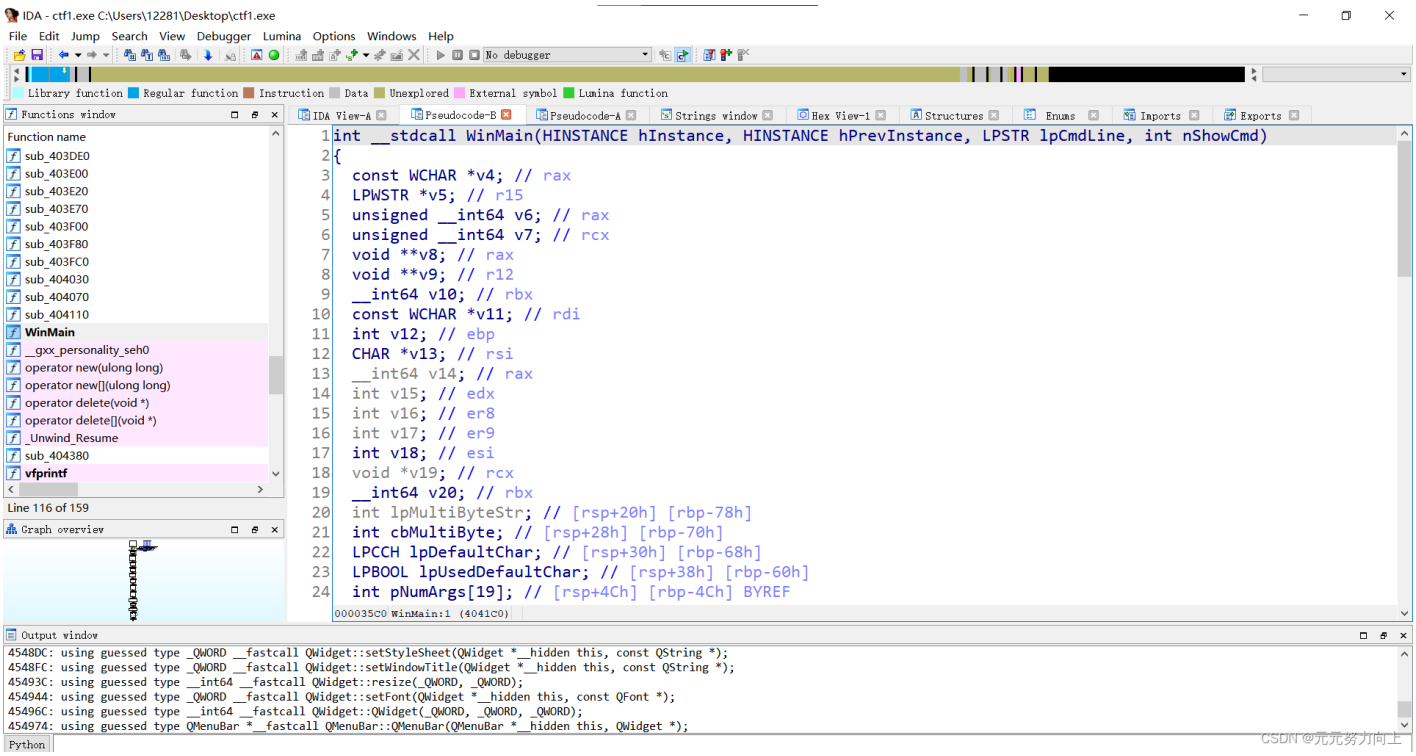
```
Happy = [
    44, 100, 3, 50, 106, 90, 5, 102, 10, 112]
flag=''
for i in range(len(Happy)):
    if i%2!=0:
        flag+=chr(Happy[i]^i)
    else:
        flag+=chr(Happy[i+1]^Happy[i])
print(flag)
```

flag{He110_cazy}

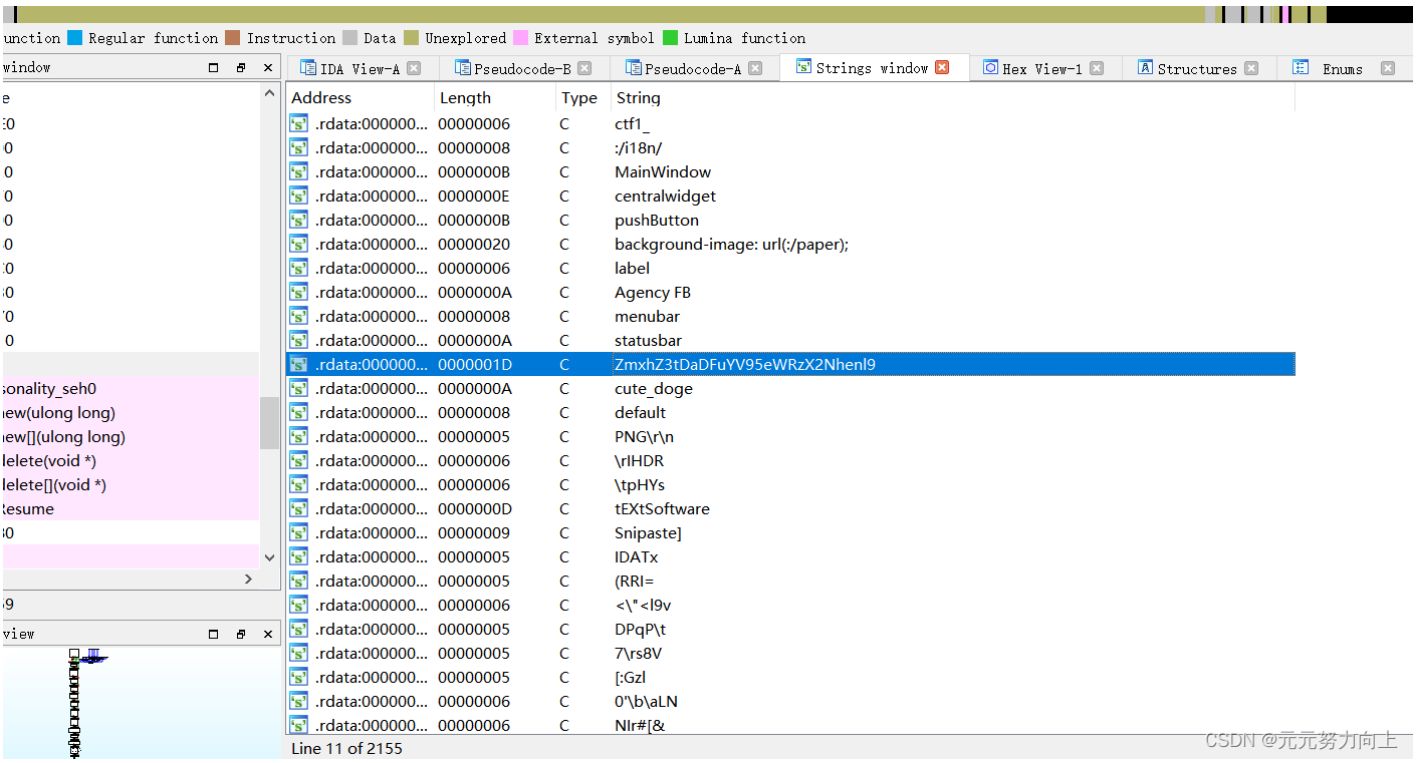
cute_dog

经典第一步查壳：

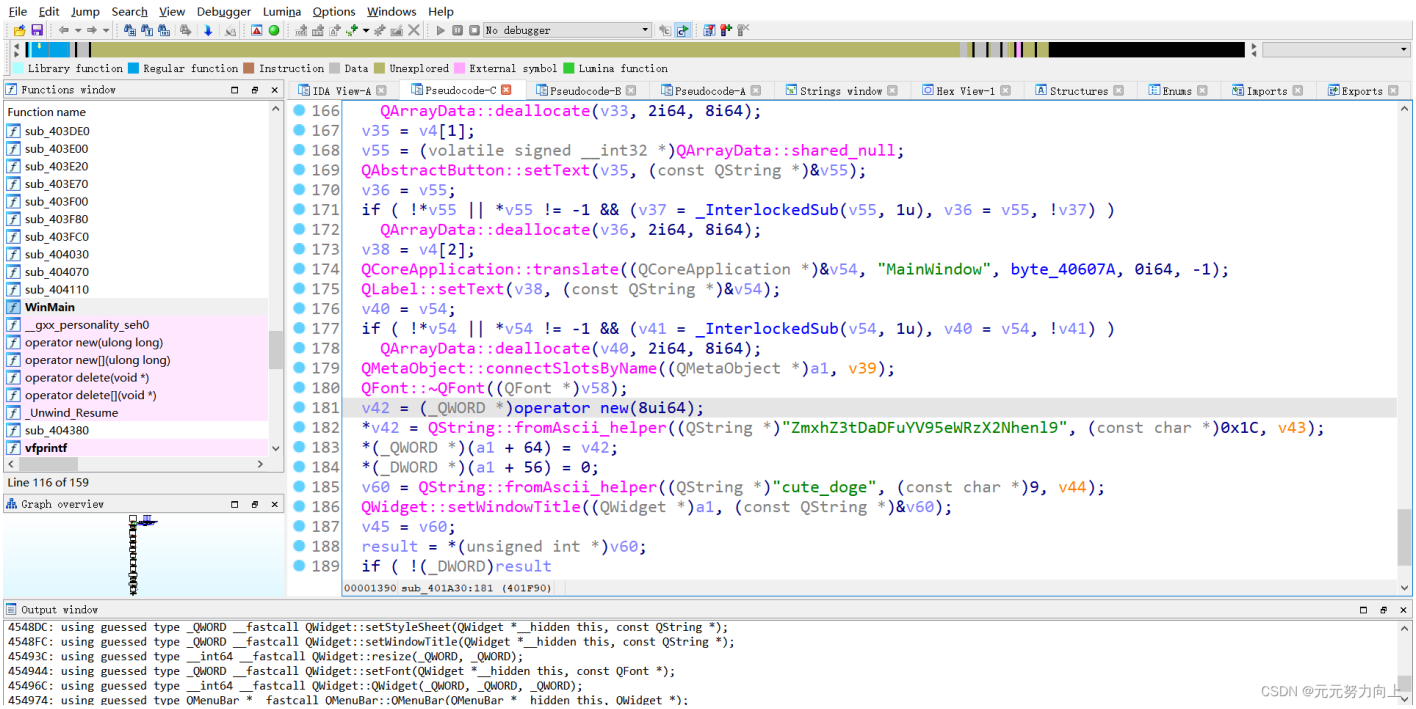
无壳64位，扔ida64：



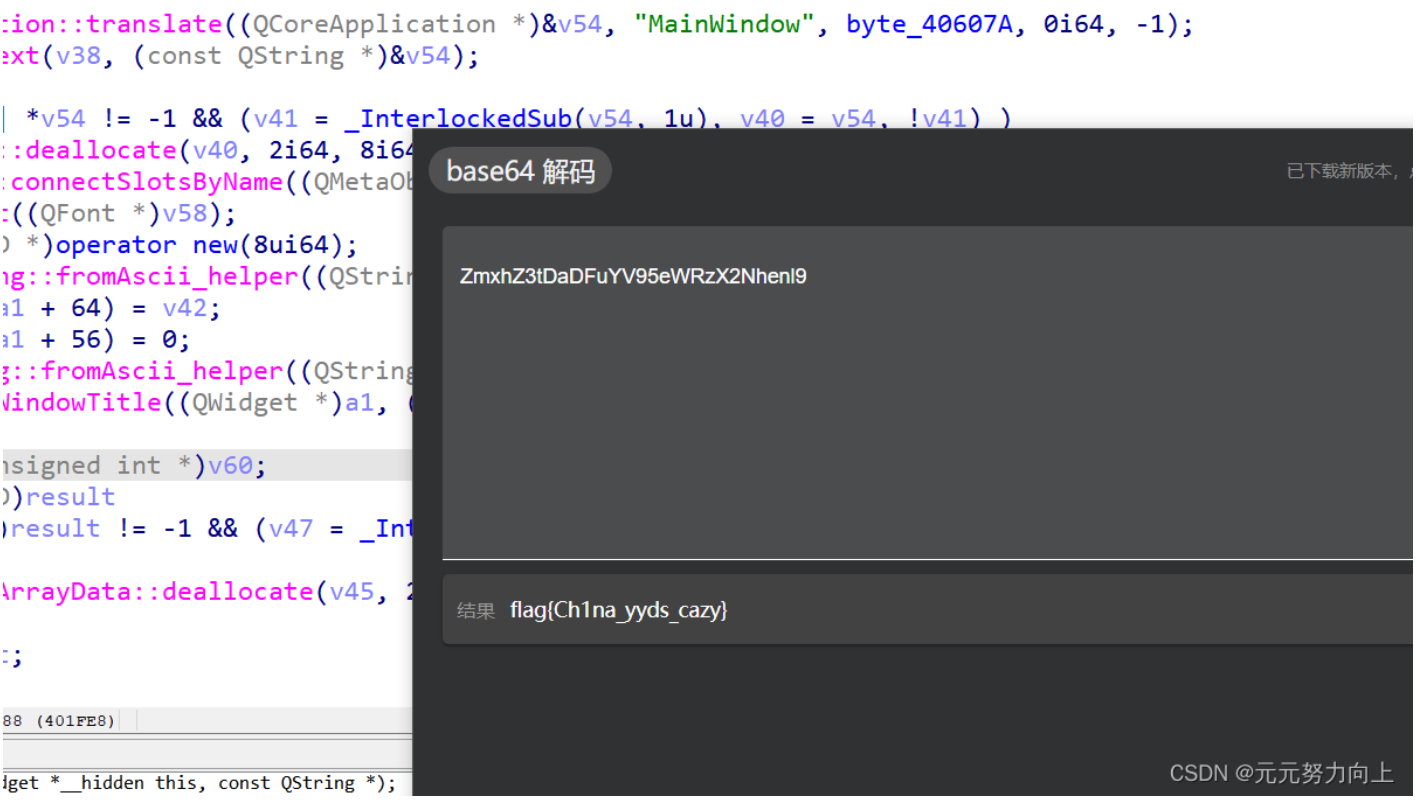
主函数没啥东西，shitf f12看看：



可疑字符串，去它函数里面看看：



也没啥特殊函数，就猜测是加密之后的字符串，先试一下base64：



得到答案： flag{Ch1na_yyds_cazy}