404gg - WP

行业排名：56

得分：51

签到题

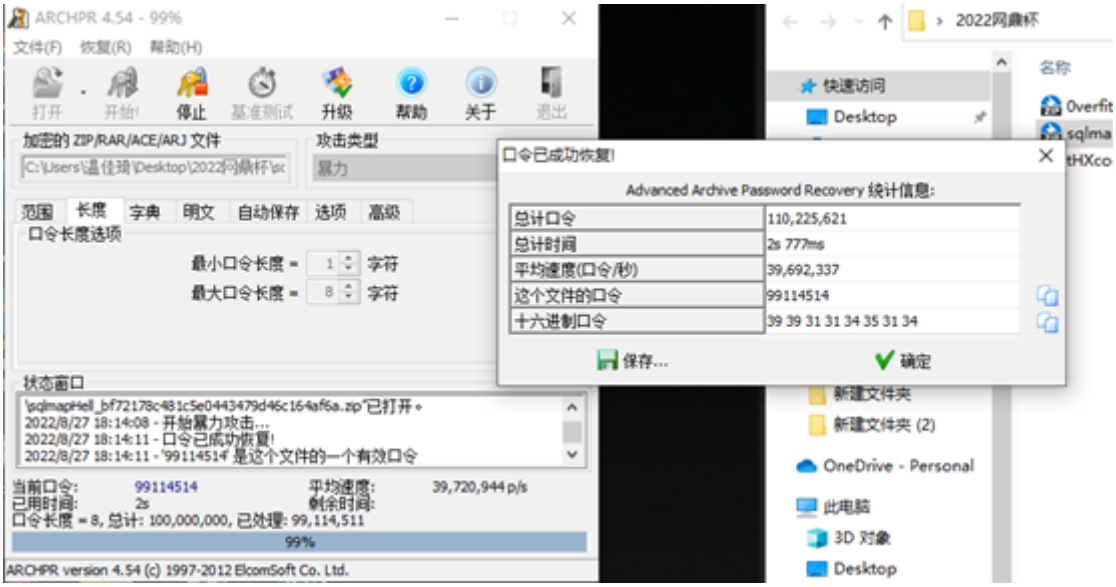
提交队员：温佳琦

Misc签到：

从网上找相关的答案即可。

Misc620

提交队员：温佳琦



尝试爆破下，得到8位纯数字密码：99114514

解压缩的到两个文件：

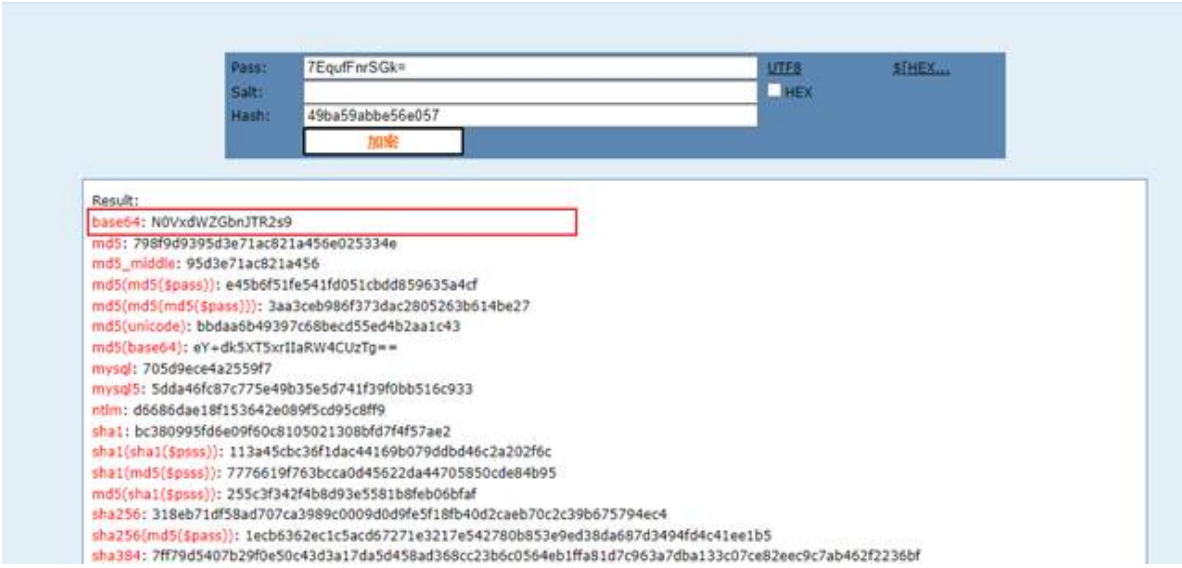
直接打开csv格式的文件有乱码，用txt打开

表头和内容对应上，得到：

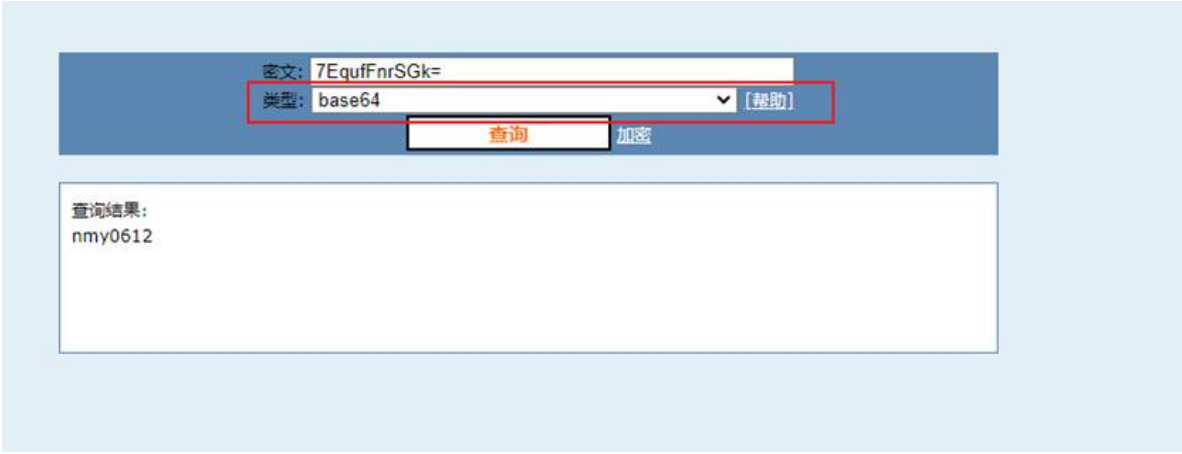
password：7EqufFnrSGk=

用base64解码得到乱码。本来想试试压缩包密码是否为md5(7EqufFnrSGk=)

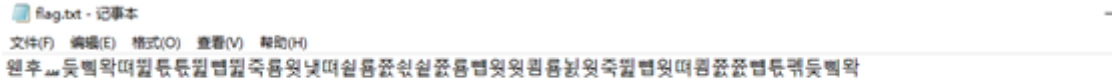
在md5网站上看到也有base64选项



尝试用cmd5网站上的base64解码：



得到flag.7z压缩包密码，输入密码得到flag.txt 文件



经过查询编码发现

Universal online Cyrillic decoder - recover your texts (2cyr.com)

可以解码。

Output (source encoding : WINDOWS-1251) Copy

旗帜左大括号地二一一二九二三杠七八地六杠四零六四杠九七七细杠必七三二九七地细四四九一右大括号

crypto582

提交队员：胡旻

根据费马小定理

$$x = (m1 + 2022)^m \ mod\ m * m1,$$

根据费马小定理可以变成

$$x\ mod\ m = (m1 + 2022)\ mod\ m$$

得到$x - 2022 = m1 + km$

同理，

$$y - 2022 = m2 + km$$

根据二项式定理和费马小定理，

$$c1\ mod\ m = m1^{2022}\ mod\ m$$

$$c1 + km = m1^{2022}$$

同理$c2$.

这样$GCD(c1 - (x - 2022)^{2022}, c2 - (y - 2022)^{2022}) = m$

得到m之后带入x - 2022，y - 2022费马小定理之后的式子，可以得到m1，但是y-2022得到的是m2 - m，所以要加上以一个m。

```
c1 =
85139434329272123519094184286276070319638471046264384499440682030525456122476228
32446276912616762812100621353115392788487030799910601543090936179209358189509144
58293795476333047379166759260042987536742681413995504059343760724860864681869073
26396270307581239055199288888816051281495009808259009684332333344687
```

```
c2 =
104554808380721645840032269336579549039995977113982697194651690041676187039363703190743891658905715473980017457465221488358016284891528960913854895940235089108270134689312161783470000803482494370322574472422461483052403826282470850666418693908817591349159407595131136843764544166774390400827241213500917391144
c3 =
947716258454491288120813452912189733019791525771315684977404761237291586193247531285172226927509005246890490786069783177425459974827636008843629924684065775247086220460334097134160261453777401822336748900633335346469276012623336722336958632866378174712703140937208274094741789179693265569399426221125118193304
x =
782373294083519554659270928059950769098260110293717832564543221666003981491326234846797233625626000689617604100392415542325880115778541684023998959923317613537724159825605229125118793049773622255975524463978688432751290272487652527845038411142913928220525068371320939602902373356863540124484148040309388737654
y =
100442166633632319633494450595418167608036668647704883492068692098914206322465717138894302011092841820156560129280901426898815274744523998613724326647935591857728931946261379997352809249780159136988674034759483947949779535134522005905257436546335376141008113285692888482442131971935583298243412131571769294029
z =
104712661985900115750011628727270934552698948001634201257337487373976943443738367683435788889160488319624447315127992641805597631347763038111352925925686965948545739394656951753648392926627442105629724634607023721715249914976189181389720790879720452348480924301370569461741945968322303130995996793764440204452

from Crypto.Util.number import *
m1_mod_m = x - 2022
tmp_m1 = pow(m1_mod_m, 2022)
#print(tmp)

m2_mod_m = y - 2022
tmp_m2 = pow(m2_mod_m, 2022)
m = GCD(tmp_m1 - c1, tmp_m2 - c2)
m1 = (x - 2022) % m
m2 = (y - 2022) % m
print(m1)
m2 = m2 + m
print(m2)
e = 2022
assert x == pow(m1 + 2022, m, m * m1)
assert c1 == pow(m + m1, e, m * m1)
assert z == pow(m + 2022, m1, m * m1)
assert y == pow(m2 + 2022, m, m * m2)
flag = m + m1 + m2
print(long_to_bytes(flag))
import hashlib
flag = hashlib.md5(str(flag).encode('utf-8')).hexdigest()
print(flag)
#flag{27979a70ef9152b759d9340779256dc8}
```