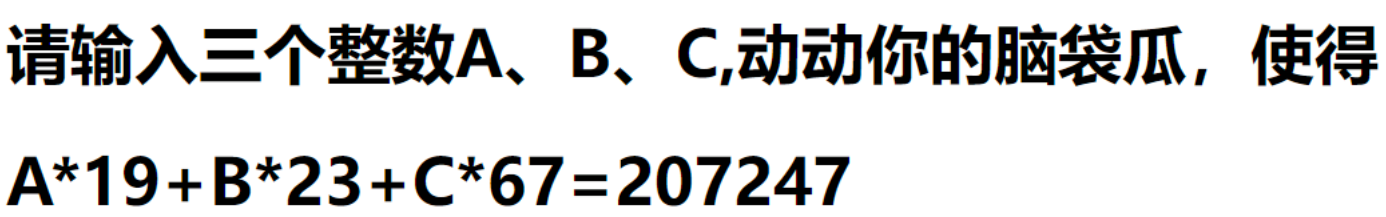
SDBCTF2021 writeup

原创

电子young 于 2022-01-15 19:17:09 发布 2140 收藏
分类专栏： ctf比赛 文章标签： 网络安全

WEB

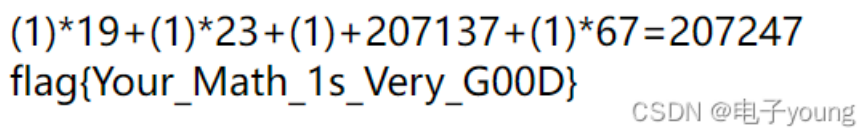
Ez_math



下方有相应计算语句，那么构建

a=1，b=1，c=1)+207137+(1

语句为：



得出flag

Ez_math2

草率了，貌似能爆破出来，我修改一下数据，看你怎么办！！！！ 😭😭🤣🤣🤣🤣🤣

A*207247+B*207213+C*207212=2072

A: 0

B: 0

C: 0)+2072+(0

Submit

(0)*207247+(0)*207213+(0)+2072+(0)*207212=2072
flag{1_am_A_j0k3r!!!}

和上面的方法一样

Ez_login

尝试万能密码

```
admin' or '1'='1
```

提交得到flag

套娃

```php
<?php
error_reporting(0);
$cr0fy = $_REQUEST['cr0fy'];
is_numeric($cr0fy)?die("no way"):NULL;
if($cr0fy>=2072)
{
        echo file_get_contents('../1.php');
        echo "How's that possible";
}
highlight_file(__FILE__);
//研究研究is_numeric()函数
?>
```


is_numeric()检测变量是否为数字或数字字符串

阅读代码后构造payload

```
https://69ab9160.cpolar.io/?cr0fy=2072a
```

进入第二个页面

Just g1ve it a try. <?php

```php
error_reporting(0);
if (isset($_GET['N0vice']) and isset($_GET['Elli0t']))
{
        if ($_GET['N0vice'] == $_GET['Elli0t'])

                echo 'no no no,you can not do this';

        else if (sha1($_GET['N0vice']) === sha1($_GET['Elli0t'])) {
                echo 'ok ok ok,I glve you f14g!';
                echo file_get_contents('../2.php');
        }
        else
                echo 'not right,may be you need google';
}
else
        echo 'Just glve it a try.';
highlight_file(__FILE__);
?>
//嘿嘿啊哈哈哈哈哈哈哈哈，sha1强碰撞来咯 🐥🐥🐥😍😍
```

//嘿嘿啊哈哈哈哈哈哈哈哈，sha1强碰撞来咯 🐥🐥🐥😍😍

可以用数组绕过sha1检测，payload：

```
https://69ab9160.cpolar.io/come_baby.php/?N0vice[]=1&&Elli0t[]=2
```

进入第三个页面

Just g1ve it a try.

```php
<?php
error_reporting(0);
if (isset($_GET['NOvice']) and isset($_GET['Elli0t']))
{
        if ($_GET['NOvice'] == $_GET['Elli0t'])
                echo 'no no no,you can not do this';
        else if(is_array($_GET['NOvice']) || is_array($_GET['Elli0t']))
        {
                echo 'too young too simple';
        }
        else if (sha1($_GET['NOvice']) === sha1($_GET['Elli0t'])) {
                echo 'well!!!,I will give you real flag:';
                echo "<br/>";
                echo file_get_contents('../flag.php');
        }
        else
                echo 'May be you need google or baidu?';
}
else
        echo 'Just g1ve it a try.';
highlight_file(__FILE__);
?>

//太可恶了，太可恶了，刚刚大意了，没有设置数组检测，让你给过来了，不讲武德，来偷袭！！！
//我这次检测了数组，看你怎么办，哼😢 😈😈😈 //太可恶了，太可恶了，刚刚大意了，没有设置数组检测，让你给过来了，不讲武德，来偷袭！！！ //我这次检测了数组，看你怎么办，哼😢 😈😈😈
```

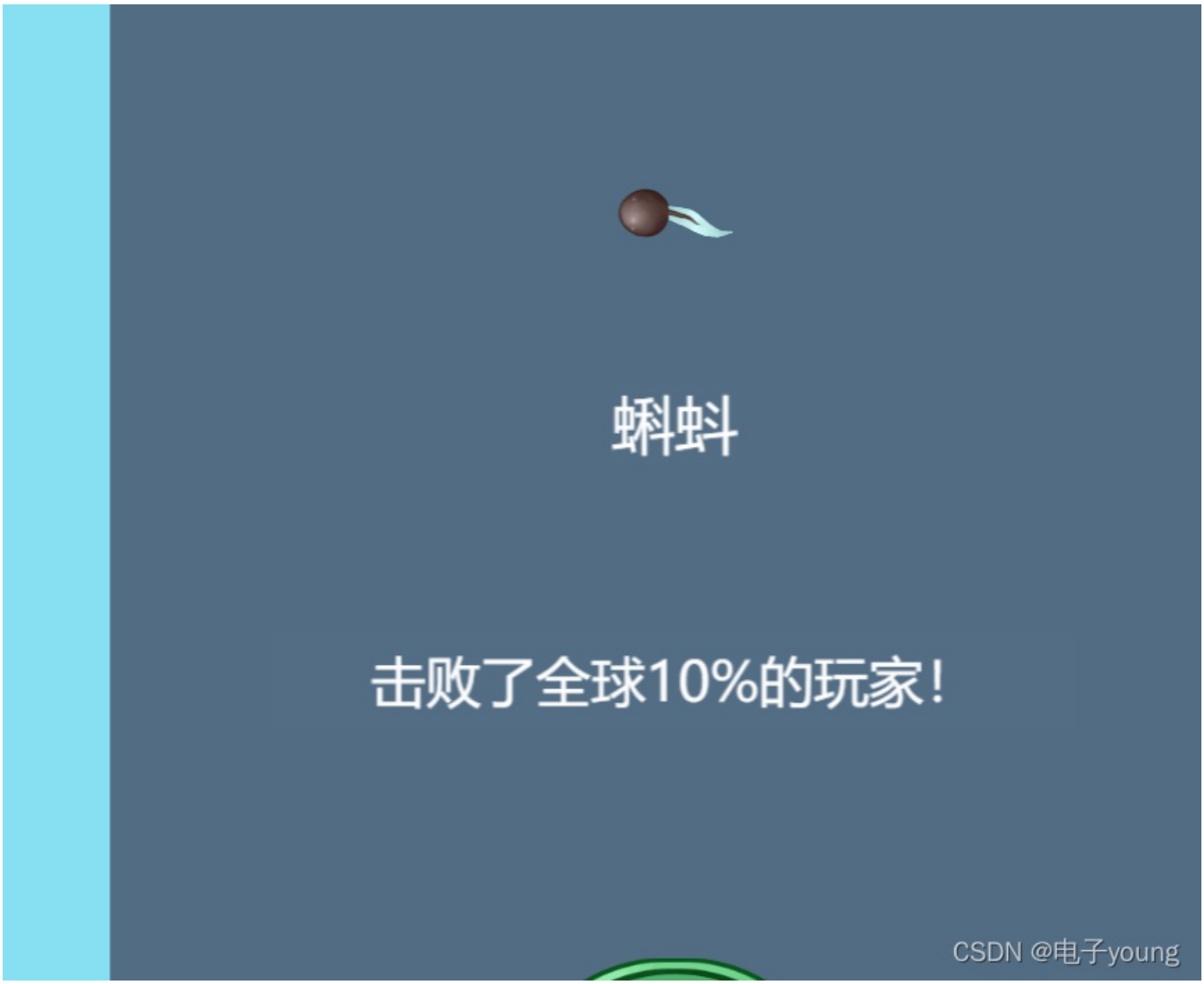

发现存在is_array检测数组

只能用sha1碰撞，找到谷歌发布的sha1

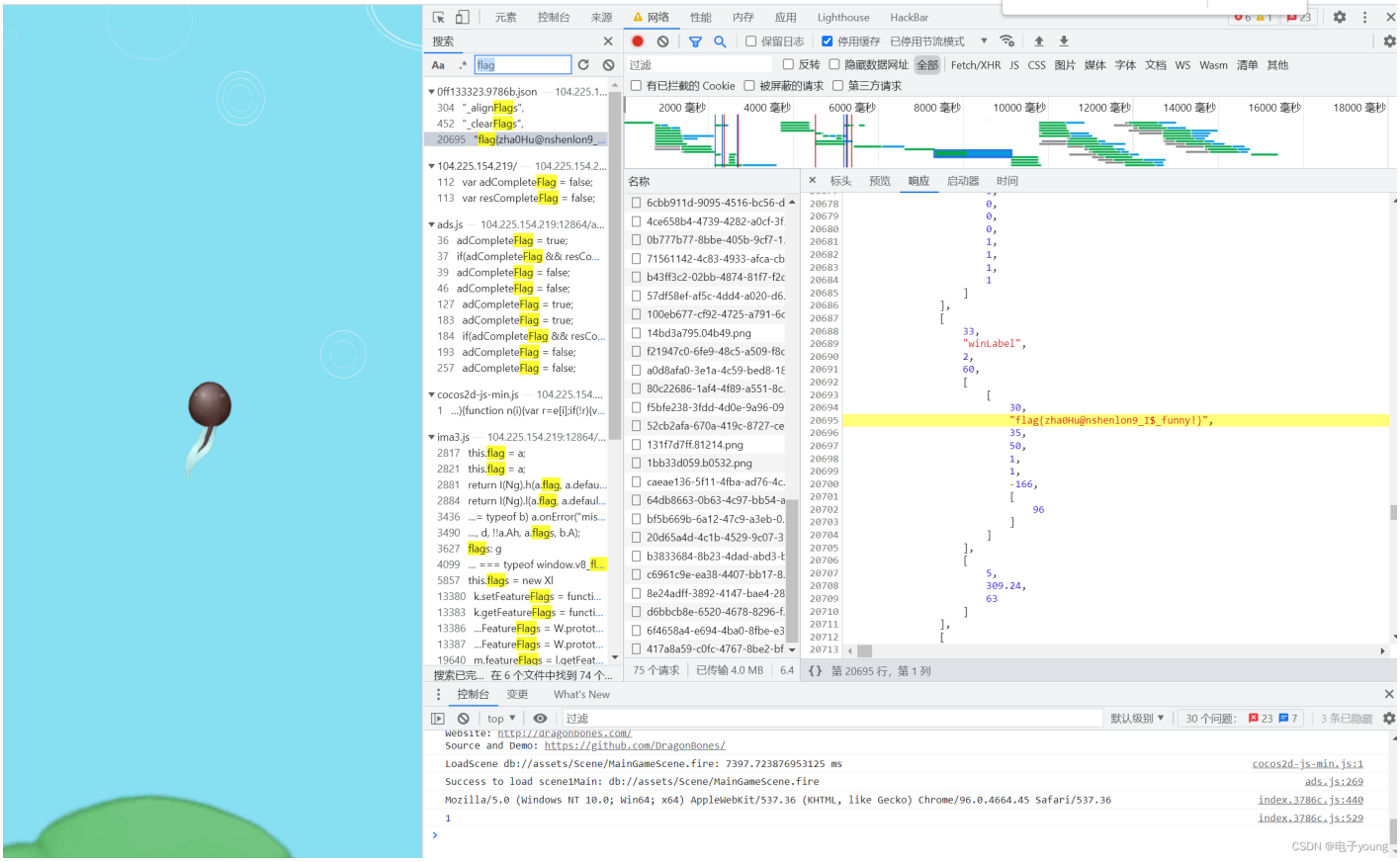
```
N0vice=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%
Elli0t=%25PDF-1.3%0A%25%E2%E3%CF%D3%0A%0A%0A1%200%20obj%0A%3C%3C/Width%202%200%20R/Height%203%200%20R/Type%
```

提交得到flag

召唤神龙

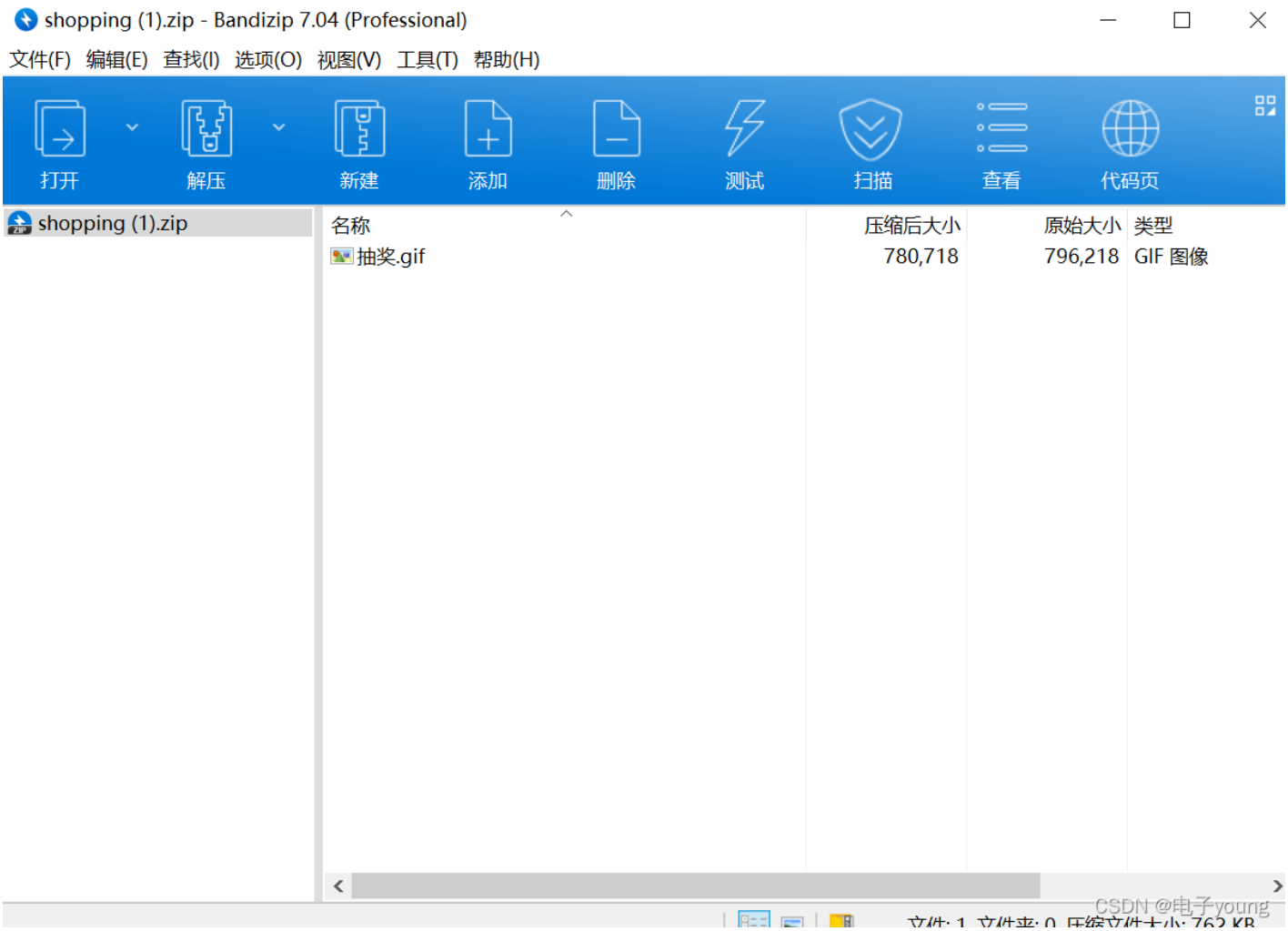


进入检查----猜测flag在网页某个代码中----搜索flag得到


flag{zha0Hu@nshenlon9_I$_funny!}
Website: http://dragonbones.com/
Source and Demo: https://github.com/DragonBones/
LoadScene db://assets/Scene/MainGameScene.fire: 7397.723876953125 ms
Success to load scene1Main: db://assets/Scene/MainGameScene.fire
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36
1
CSDN @电子young

MISC

抽奖



发现是gif文件，进行分离，之前的工具不知道为啥分离不了，去在线工具找gif分离

找到个二维码，扫描得到flag

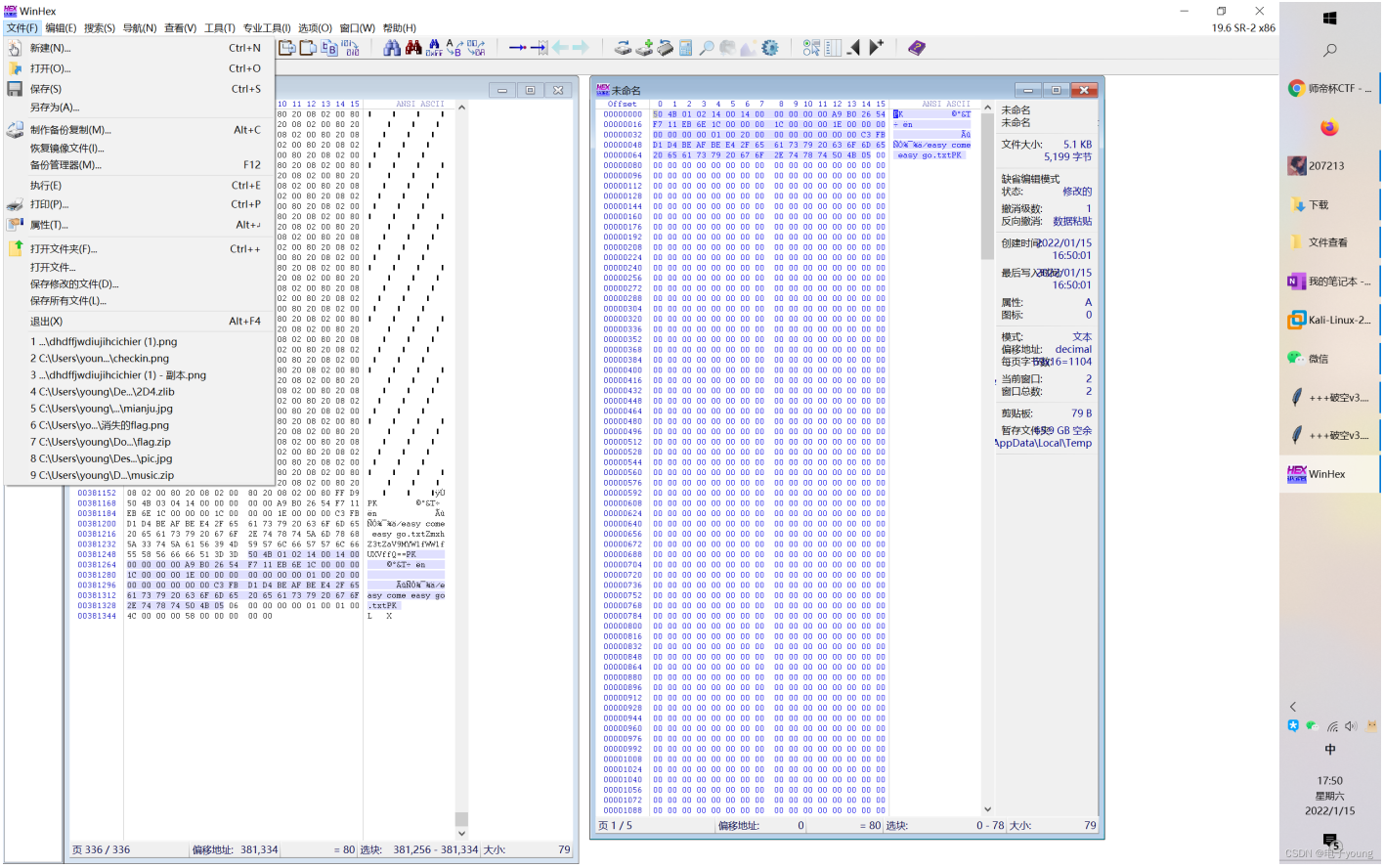
藏猫猫

使用工具找

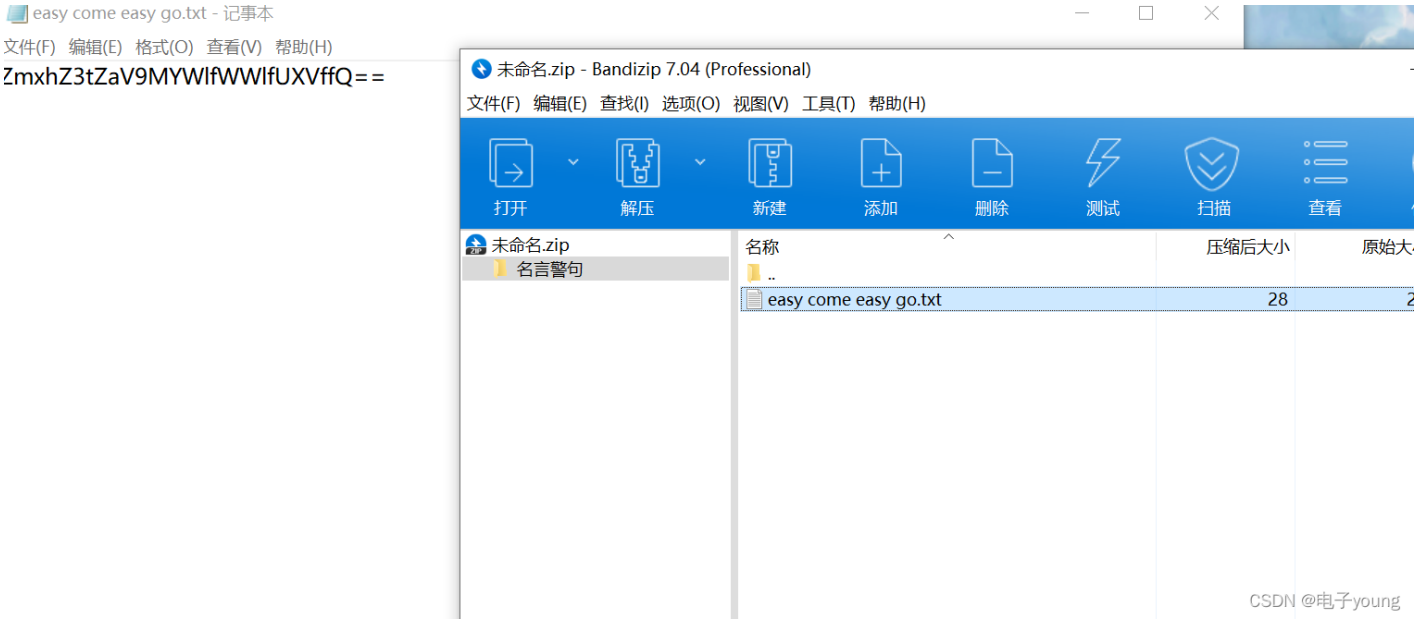
倘来之物

打开发现是一个图片，丢入binwalk分析

发现有zip存在，把它提取出来

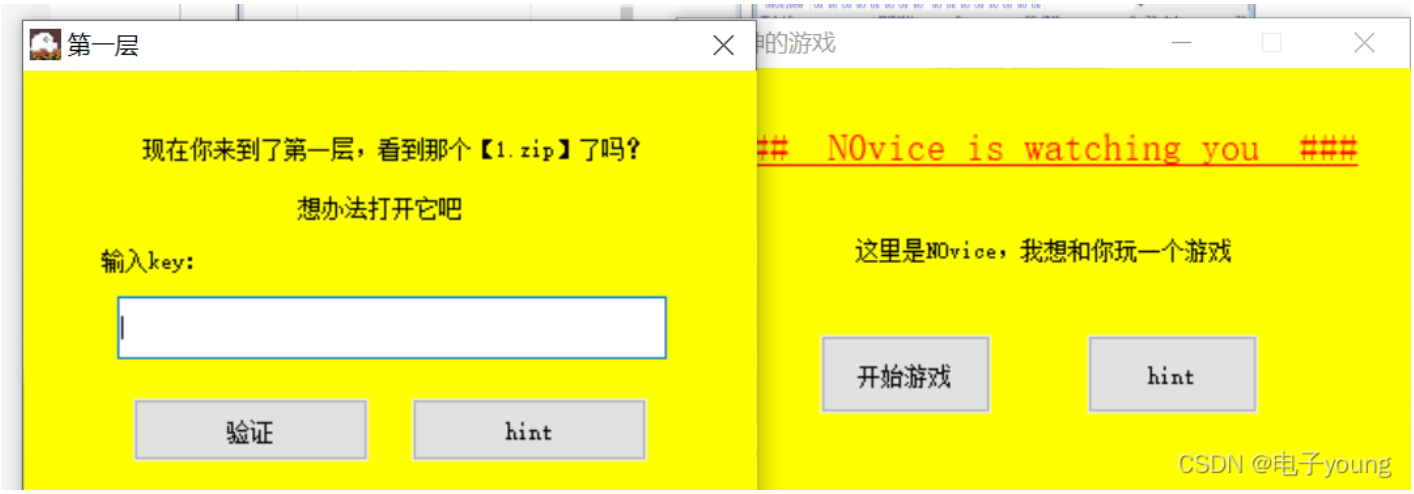


发现



一串base64编码，解码得到flag

N0vice的游戏



一个压缩包，爆破密码为123456，得到二维码，扫码得到key

第二个为一张图片，放入binwalk分离

```
root@kali: ~
File  Actions  Edit  View  Help

┌──(root💀kali)-[~]
└─# binwalk /home/kali/Desktop/2.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1209 x 1920, 8-bit/color RGBA, non-i
nterlaced
41            0x29            Zlib compressed data, default compression
1551846       0x17ADE6        Zip archive data, at least v2.0 to extract, com
pressed size: 49, uncompressed size: 44, name: key_2.txt
1552025       0x17AE99        End of Zip archive, footer length: 22


┌──(root💀kali)-[~]
└─# binwalk -e /home/kali/Desktop/2.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             PNG image, 1209 x 1920, 8-bit/color RGBA, non-i
nterlaced
41            0x29            Zlib compressed data, default compression
1551846       0x17ADE6        Zip archive data, at least v2.0 to extract, com
pressed size: 49, uncompressed size: 44, name: key_2.txt
1552025       0x17AE99        End of Zip archive, footer length: 22
```


得到key

第三个是修改图片高度

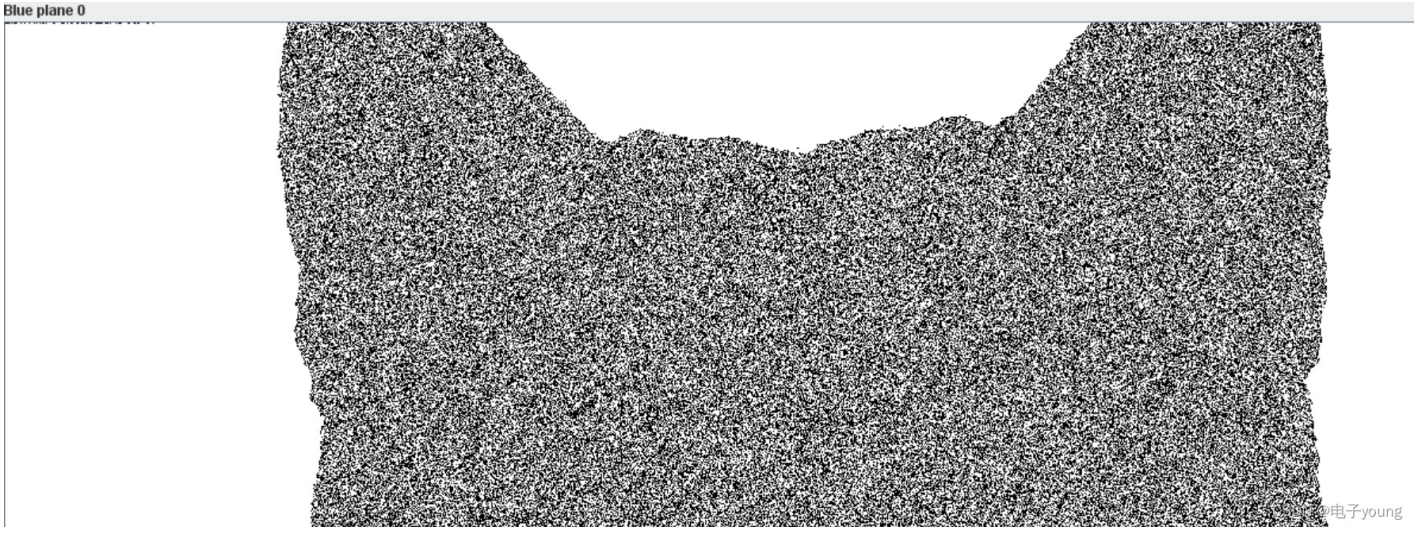
可以看出下方缺少，winhex修改高度得到key

最终得到flag

---

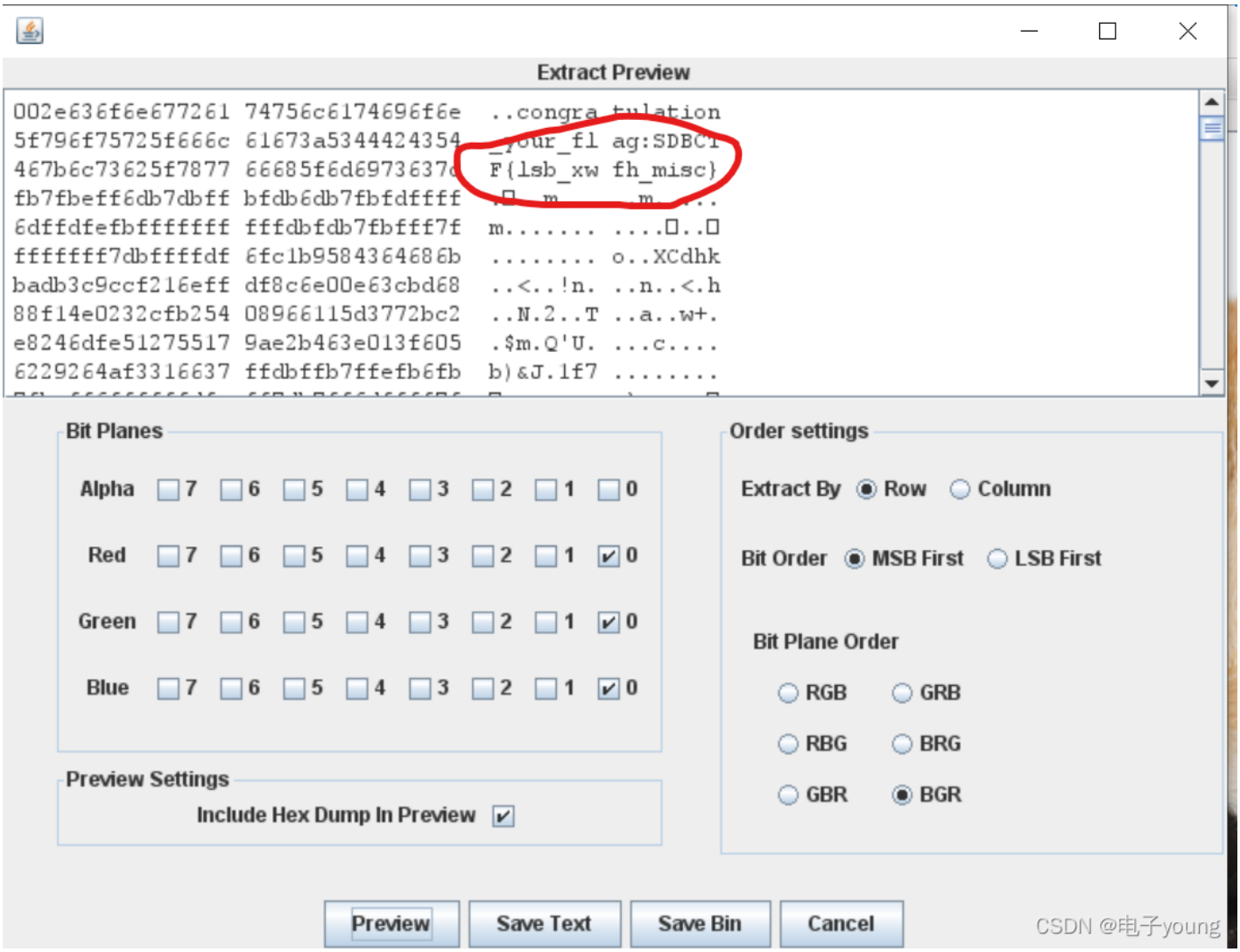
b1ndog

lsb隐写



上方明显有东西

一开始RGB发现没有啥东西，然后看到反向的{ }猜测是BGR，得到flag



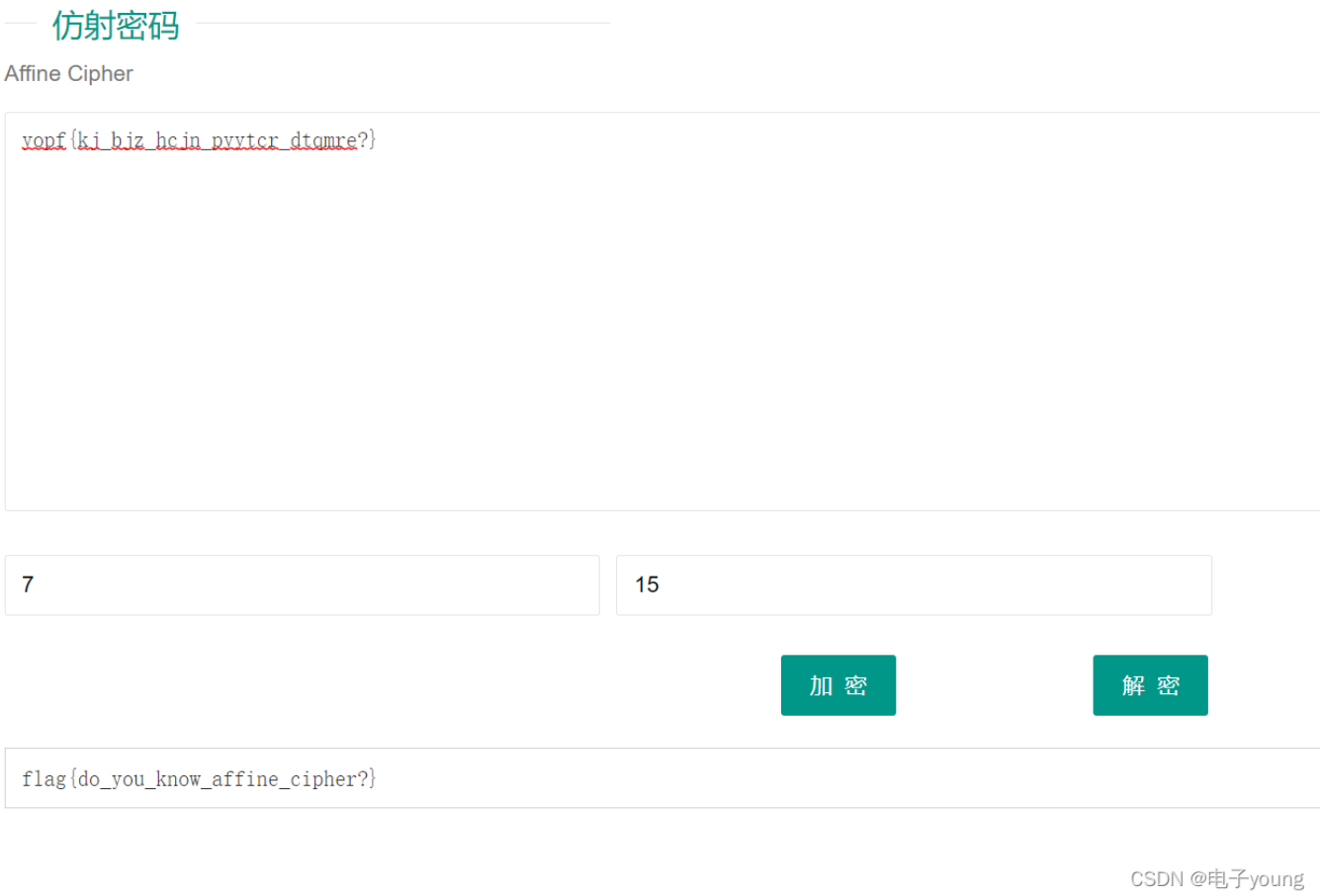
CRYPTO

affine

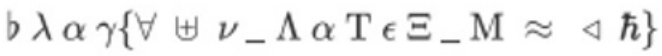
提示是affine的话就试试仿射

yopf{kj_bjz_hcjn_pyytcr_dtqmre?}

在线工具解码



Signin



一开始懵了，然后看到图片名字symbols，一直找，找到了Latex symbol这个东西，对了很久很久很久，总算对照出来了

±	\pm	∩	\cap	◇	\diamond	⊕	\oplus
∓	\mp	∪	\cup	△	\bigtriangleup	⊖	\ominus
×	\times	⊎	\uplus	▽	\bigtriangledown	⊗	\otimes
÷	\div	⊓	\sqcap	◁	\triangleleft	⊘	\oslash
*	\ast	⊔	\sqcup	▷	\triangleright	⊙	\odot
⋆	\star	∨	\vee	◁	\lhd	◯	\bigcirc
∘	\circ	∧	\wedge	▷	\rhd	†	\dagger
•	\bullet	\	\setminus	⊴	\unlhd	‡	\ddagger
·	\cdot	/	\wr	⊵	\unrhd	⨿	\amalg
+	+	−	-				

。其中一些，例如<和>，是通过键入相应的键盘字符产生的。其他通过下表中的命令获取：

♣	#club	⇔	#Leftrightarrow	←	#leftarrow	∇	#nabla	å	#aa
℘	#voidn	|	#void8	⊗	#otimes	⌟	#downleftarrow	/	#/
≤	#leq	ℏ	#hbar	⇐	#Leftarrow	—	#topbar	\	#backslash
≈	#approx	♦	#diamond	∏	#prod	⌠	#arcbar	·	#upoint
∈	#in	ℵ	#aleph	□	#Box	↑	#uparrow	∂	#partial
⊃	#supset	≥	#geq	||	#parallel	⊕	#oplus	¬	#corner
∩	#cap	≠	#neq	♥	#heart	⇑	#Uparrow	⌊	#ltbar
©	#ocopyright	∉	#notin	ℑ	#Jgothic	∑	#sum	⌊	#bottombar
™	#trademark	⊆	#subseteq	⟨	#LT	⊥	#perp	→	#rightarrow
×	#times	∪	#cup	≡	#equiv	∀	#forall	√	#surd
•	#bullet	©	#copyright	⊂	#subset	♠	#spade	⇒	#Rightarrow
ƒ	#voidb	™	#void3	⊇	#supseteq	ℜ	#Rgothic	∫	#int
″	#doublequote	÷	#divide	∧	#wedge	⟩	#GT	⊙	#odot
|	#lbar	°	#circ	®	#oright	∝	#propto	∃	#exists
⌈	#arcbottom	∞	#infty	Å	#AA	⊄	#notsubset	+	#plus
↓	#downarrow	∠	#angle	±	#pm	∅	#oslash	−	#minus
↔	#leftrightarrow	|	#cbar	∓	#mp	∨	#vee		
⇓	#Downarrow	⌈	#arctop	⋯	#3dots	⊕	#void1		

δ \delta

ϵ \epsilon

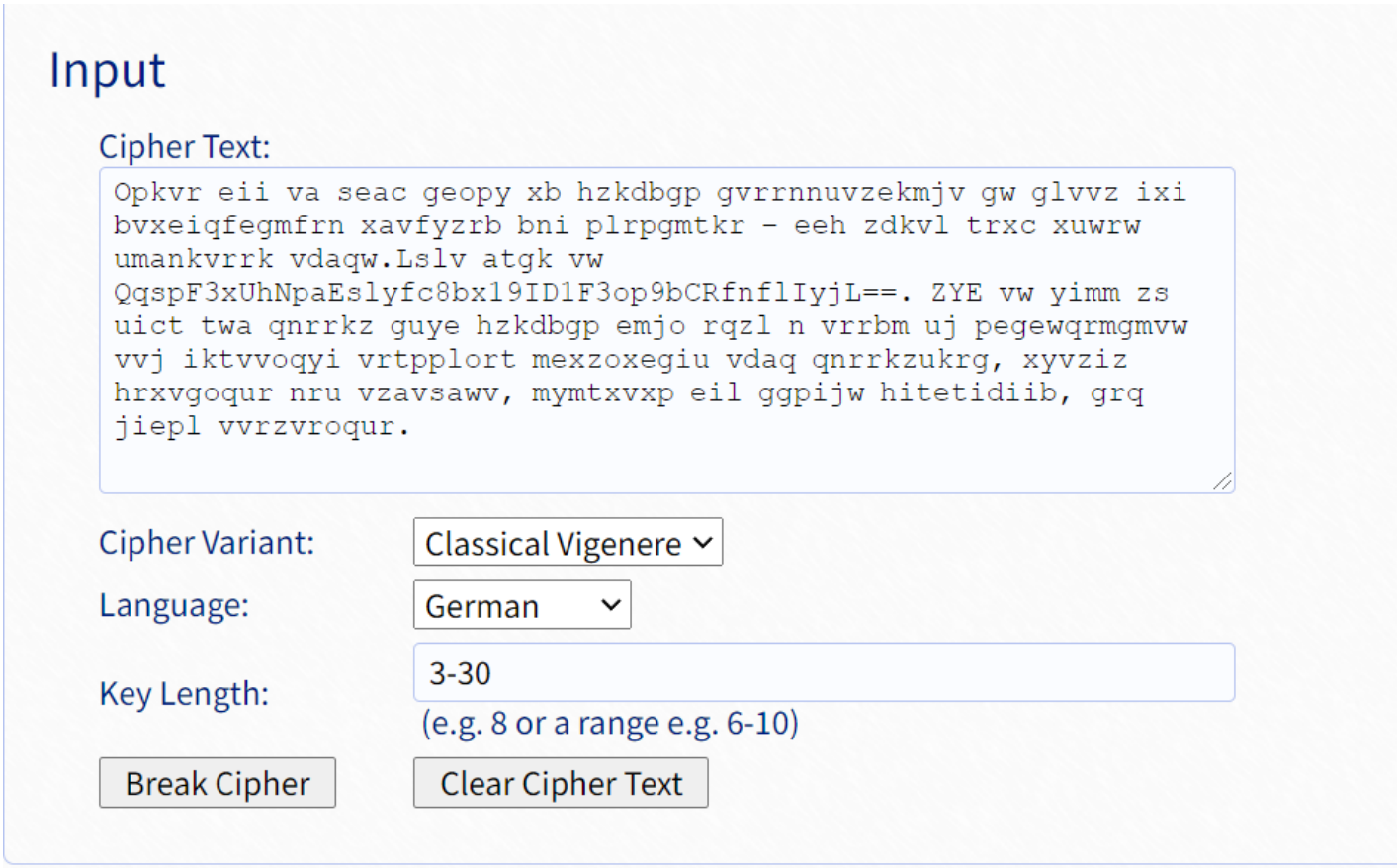
flag{fun_LaTeX_Math}

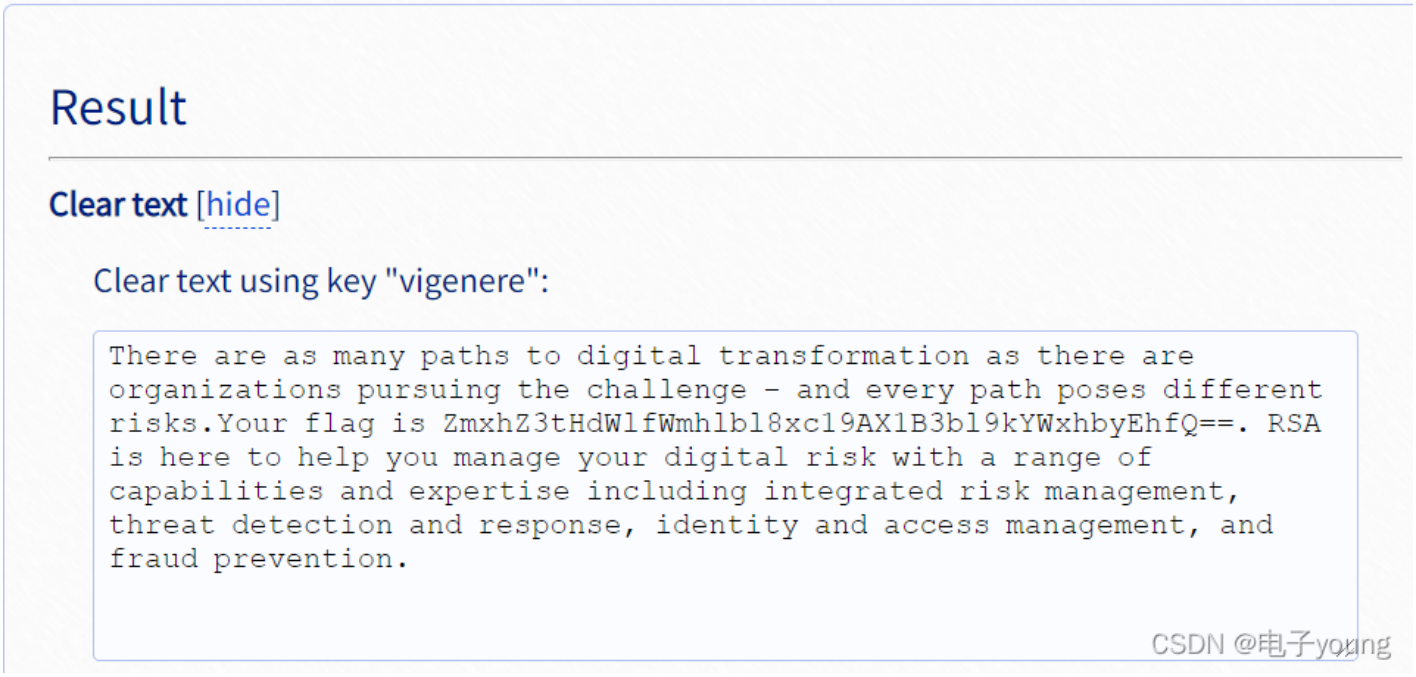
Ez_classical

```
Opkvr eii va seac geopy xb hzkdbgp gvrrnnuvzekmjv gw glvvz ixi bvxeiqfegmfrn xavfyzrb bni plrpgmtkr – eeh z
```

古典密码的话应该是维吉尼亚密码

但是我也不知道密钥是啥，最后找到个网站Vigenere Solver - www.guballa.de在线爆破





看到个base64，解码得到flag

Basic_RSA

```
n = 71687100457195407860469819444789606170290157387683684916906107530942406676860709254712769885858066624040
c = 68674580990932158766103042615106094896367460229852102108353388855376474807270391611440380160789695077800
e = 0x10001
```

在线网站分解n无效，可能是接近，调用yufu进行分解，得到p，q

```
P77 = 846682351636051333731023345322384324552705075654991127358762747415006172077 61

P77 = 846682351636051333731023345322384324552705075654991127358762747415006172077 13
```

```
import gmpy2 as gp
import binascii
p =  gp.mpz()
q =  gp.mpz()
e =  gp.mpz()
c =  gp.mpz()
n = p*q
phi = (p-1) * (q-1)
d = gp.invert(e, phi)
m = pow(c, d, n)
print(m)
```

转字符得到**flag{Th1s_1s_B@sic_R54!}**

RSA's door

```
n = 834601703327895433594450604114970917058043473648142812454705923711336631064935306085294094691029549371 8
c = 262507574231085677927245013780315446984706165425832229497910287340593637000924061623758114413501473277 6
e = 3
```

e特别小，那就用低加密指数攻击

```
#python3
import  RSAwienerHacker
n=
e=
d =  RSAwienerHacker.hack_RSA(e,n)
if d:
    print(d)
import hashlib
flag = "flag{" + hashlib.md5(hex(d)).hexdigest() + "}"
print flag
```

或者用一些工具更方便，最后得出

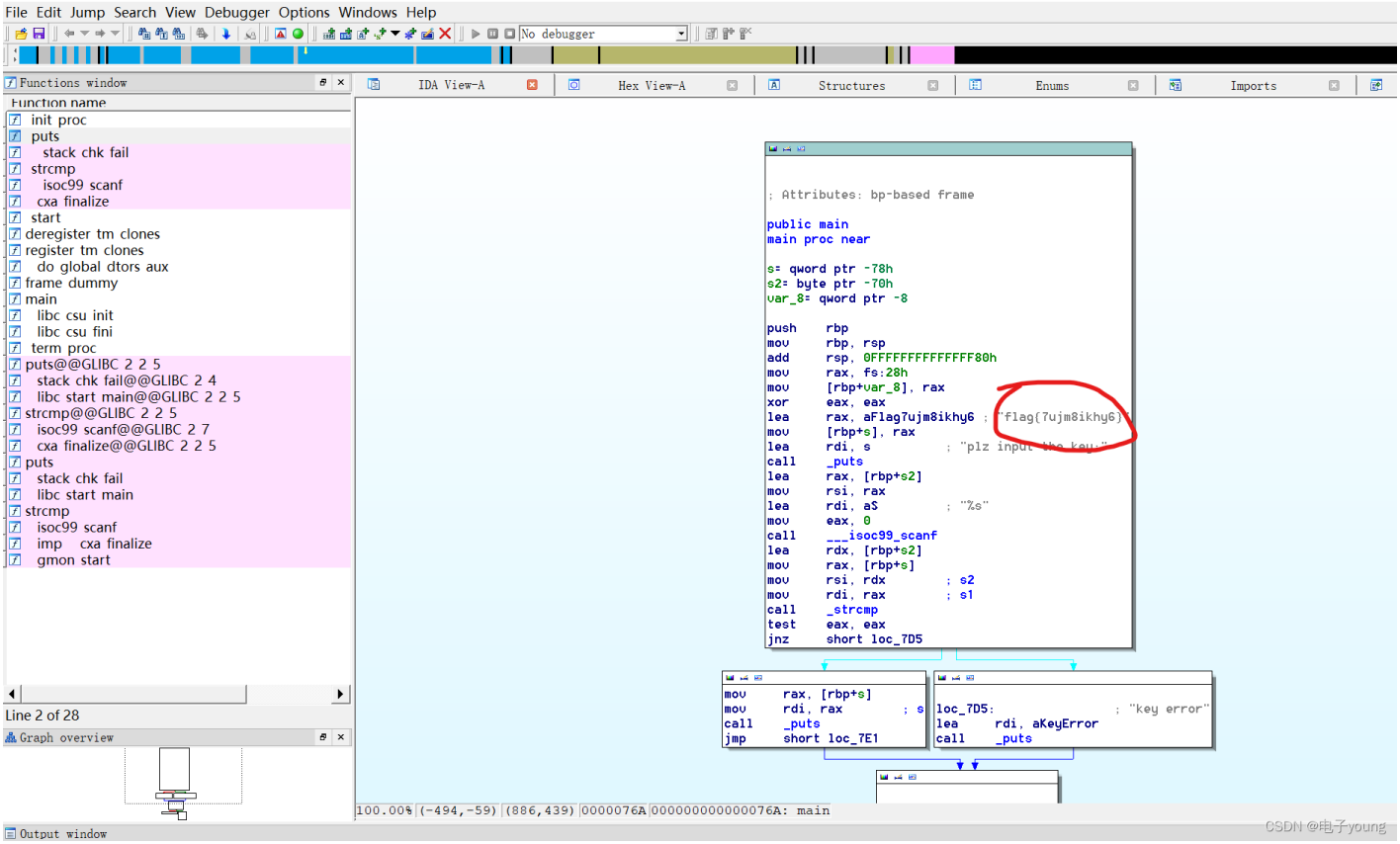
1433763655511617498648135287891321364534462577872359854642976836020073844911403039243128
转字符串**flag{We1c0me_To_C0oooooola_Gouliang_Cuppp!}**

Reverse

逆向签到处

打开压缩包发现有密码，直接破解密码得出123456

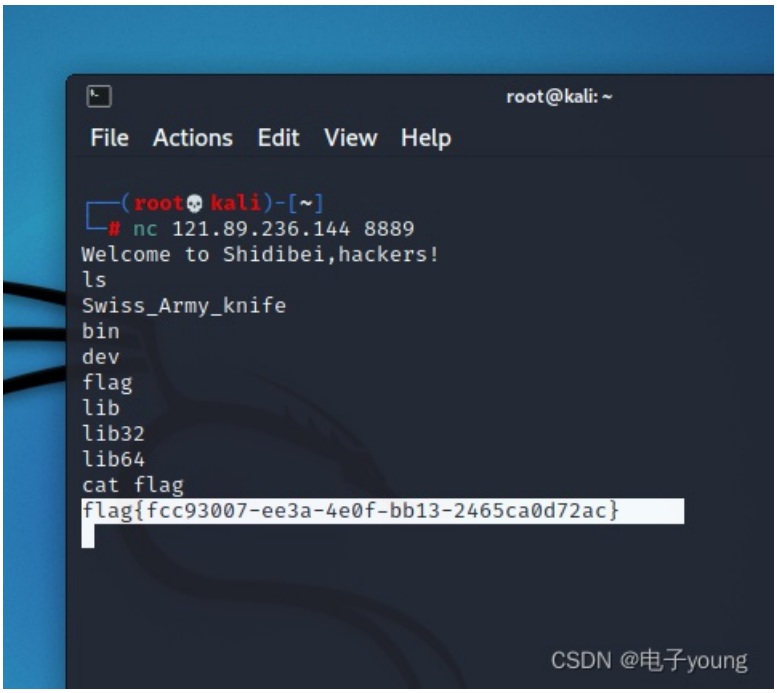
ida打开文件找到flag



> PWN

给出了个端口，kali中用nc进入

121.89.236.144 8889



ls看看目录，cat flag得到flag

flag{fcc93007-ee3a-4e0f-bb13-2465ca0d72ac}