

好多压缩包-writeup

打开题目

Challenge

232 Solves

×

好多压缩包
200

123.zip

Flag

Submit

得到一个压缩包附件，进行解压，查看到里面有68个压缩包

名称	压缩后大小	原始大小	类型	修改日期
out22.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out23.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out24.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out25.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out26.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out27.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out28.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out29.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out30.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out31.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out32.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out33.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out34.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out35.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out36.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out37.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out38.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out39.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out40.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out41.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out42.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out43.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out44.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out45.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out46.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out47.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out48.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out49.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out50.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out51.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out52.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out53.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out54.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out55.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out56.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out57.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out58.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out59.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out60.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out61.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22
out62.zip	132	132	ZIP 压缩文件	2016/8/29 15:07:22

文件: 68, 文件夹: 0, 压缩包大小: 17.4 KB

打开68个文件发现里面每个压缩包里都有一个四个字节大小的txt文档（加密），首先尝试下是不是伪加密，发现不是尝试下爆破无果。。。然后想了下，这么多压缩包，不可能让我们爆破吧，尝试下crc32碰撞

CRC32碰撞可以参考我写的文章[zip压缩包--加密篇](#)

data.txt* 18 4

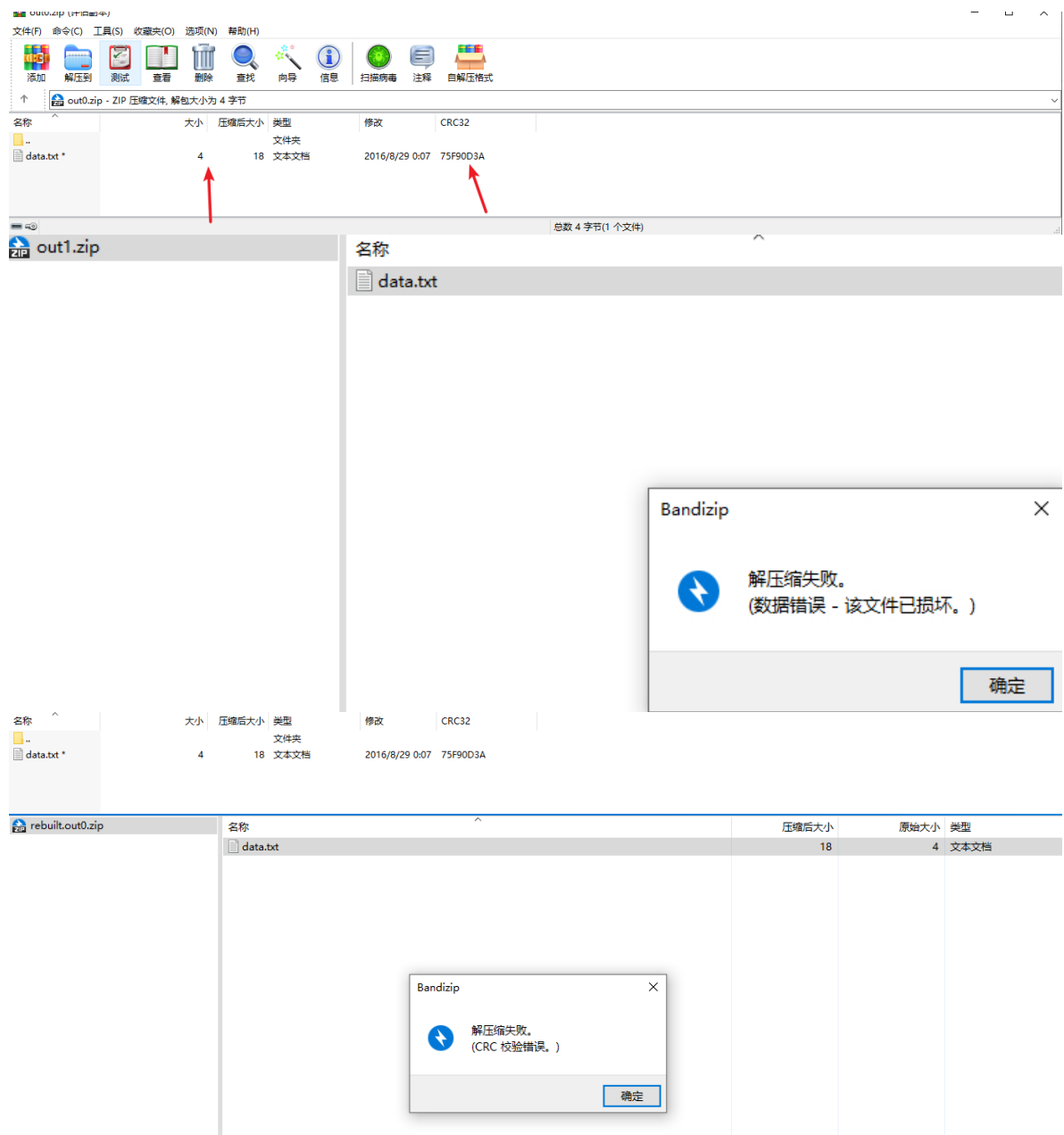
输入密码

data.txt

请输入密码

☐ 用星号隐藏密码(H)

确定 取消



分析里面得文件得四个字节，想着应该是英文，中文字符得话跟爆破没区别...

猜测里面为四个字符（英文），获取zip文件得crc32值，进行crc碰撞

碰撞出来发现是一串base64，拿去解密(有些网站解码不了，解码了也是有问题的，多尝试几个)，放入winhex，发现导入进去的时候老是有问题，我不知道怎么处理，莫名其妙的多了很多空格，搜索了下感觉他们很顺利的就放进去了

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	ANSI ASCII
00000000	CF 00	90 00	73 00	0D 00	0A 00	AA 00	3E 00	7A 00	I	s	*	>	z				
00000010	80 00	23 00	49 00	54 00	02 00	86 00	34 00	AB 00	e	#	I	T	t	4	«		
00000020	FE 00	6B 00	63 00	1D 00	49 00	1D 00	33 00	03 00	p	k	c	I	3				
00000030	01 00	43 00	4D 00	54 00	09 00	15 00	14 00	CB 00	C	M	I	E					
00000040	DD 00	41 00	4F 00	95 00	24 00	48 00	D3 00	E8 00	Y	A	C	•	ç	H	Ç	è	
00000050	8F 00	98 00	45 00	11 00	51 00	41 00	46 00	F7 00	"	E	Q	A	F	÷			
00000060	9F 00	1D 00	20 00	42 00	7C 00	6D 00	2B 00	B8 00	ÿ	B		+	.				
00000070	69 00	CA 00	9F 00	28 00	2C 00	33 00	28 00	FC 00	i	È	ÿ	(,	3	(ü	
00000080	48 00	16 00	99 00	1F 00	1B 00	18 00	1D 00	8F 00	H	"							
00000090	38 00	2C 00	46 00	76 00	E1 00	C5 00	ED 00	67 00	8	,	F	v	á	Ã	i	g	
000000A0	4D 00	72 00	DE 00	4D 00	4A 00	D5 00	82 00	74 00	M	x	ß	M	J	Ö	,	t	
000000B0	BE 00	92 00	BD 00	1F 00	0D 00	0A 00	94 00	CD 00	¾	'	½	"	í				
000000C0	BE 00	AE 00	F7 00	3F 00	22 00	80 00	4A 00	F7 00	¾	÷	?	"	€	J	÷		
000000D0	74 00	20 00	90 00	2D 00	1D 00	1D 00	02 00	62 00	t	-							
000000E0	D1 00	E7 00	D5 00	4F 00	63 00	1D 00	49 00	1D 00	Ñ	ç	Ö	c	c	I			
000000F0	30 00	08 00	20 00	66 00	6C 00	61 00	67 00	2E 00	0								
00000100	74 00	78 00	74 00	B0 00	34 00	69 00	66 00	66 00	t	x	t	°	4	i	f	f	
00000110	69 00	78 00	20 00	74 00	68 00	65 00	20 00	66 00	i	x							
00000120	69 00	6C 00	65 00	20 00	61 00	6E 00	64 00	20 00	i	l	e						
00000130	67 00	65 00	74 00	20 00	74 00	68 00	65 00	20 00	e	t							
00000140	66 00	6C 00	61 00	67 00	C4 00	3D 00	7B 00	40 00	f	l	a	g	Å	=	{	0	
00000150	07 00	00 00															

尝试多次发现无果。然后无奈就自己写python实现直接base64解密后直接写入文件里
python脚本:

```
# coding:utf-8

import binascii
import string
import zipfile
import base64

dicts = string.printable # 可打印字符的字符串。ascii码33-126号

def collision_crc(crc):
    global out_file
    for a in dicts:
        for b in dicts:
            for c in dicts:
                for d in dicts:
                    strings = a + b + c + d
                    strings = strings.encode('utf-8')
                    if crc == (binascii.crc32(strings)):
                        out_file.write(strings.decode('utf-8'))
                        # print(strings)
                    return
    # 以上定义一个方法，组合随机字符与CRC进行碰撞，判断如果相等及写入

文件

def obtain_zip():
    for i in range(68):
        file = 'out' + str(i) + '.zip'
        zip_file = zipfile.ZipFile(file, 'r') # 读取创建zip_file对象
        get_crc = zip_file.getinfo('data.txt') # 压缩文件夹里的data.txt文件，获取文
        档内指定的文件信息
        crc = get_crc.CRC
        # 以上定义一个方法，获取68个zip的CRC的值
        collision_crc(crc) # 再调用collision方法传参
```

```

out_file = open('out.txt', 'w')
obtain_zip()
out_file.close()

out_file2 = open('out.txt', 'r')

with open('flag.rar', 'wb') as rar:
    rar.write(base64.b64decode(out_file2.read())) # 二进制将转换后的base64位写入文件

```

运行完成后将写出的文件，打开发现打开失败，导入16进制编辑器，观察数据，发现存在rar的文件尾 C4 3D 7B 00 40 07 00，但缺少文件头，于是补上rar的文件头52 61 72 21 1A 07 00，发现文件修复成功，解压发现是一个flag在压缩包注释上。二进制上也可以看到CMT，CMT即为comment（注释）

The screenshot displays a hex editor interface with the following data:

Address	Hex	ASCII
0000h	52 61 72 21 1A 07 00	Rar!...
0010h	00 00 00 00 AA 3E 7A 00>z.e#.I..T
0020h	00 00 00 00 86 34 AB FEf4&pkc.I.3..
0030h	01 00 00 00 43 4D 54 09CMT...EYAO+S
0040h	48 D3 E8 81 98 45 11 51	H0e."E.DAF=Y. B
0050h	6D 2B B8 69 CA 9F 28 2C	m+.iEY(. (QH.™..
0060h	18 1D 8F 38 2C 46 76 E1	..8.FvAqMrBMJ
0070h	D5 82 74 BE 91 BD 1F 0A	Ö,t%". "I??"E
0080h	4A F7 74 20 90 2D 00 1D	J=t
0090h	62 D1 E7 D5 4F C3 1D 49	bNgÖOc.I.0....
00A0h	66 6C 61 67 2E 74 78 74	flag.txt."diffi
00B0h	20 74 68 65 20 66 69 6C	the file and ge
00C0h	74 20 74 68 65 20 66 6C	t the flagA={.0.
00D0h	00	.

The file explorer shows the following files:

- 00000000.rar:1
- 00000000.rar:2
- 5.rar*
- flag.rar
- flag.rar
- RAR.bt1
- RAR.bt2
- RAR.bt3
- RAR.bt4
- RAR.bt5
- RAR.bt6
- RAR.bt7
- zip.rar

The output window shows the following results:

```

Executing template 'C:\Users\
Result = 0 [0h]

```

这里说明下我之前base64导入进去的出现的问题，因为心里一直纳闷，所以去多次尝试发现需要利用notepad++32位的进行base64解密，就可以得到正确的值，保存为.rar文件，然后进行导入十六进制编辑工具里是可以实现的。（之前用的时notepad++64位的会出现解密不出来的情况，几个在线工具解密出来的值都是错误的或者跟上面导进去不知道为什么多了很多空格这些）这里举几个对比：

