

BUUCTF: zip

原创

末初 于 2020-11-01 21:11:23 发布 1164 收藏 1

分类专栏: [CTF_MISC_Writeup](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/mochu7777777/article/details/109435117>

版权



[CTF_MISC_Writeup](#) 专栏收录该内容

246 篇文章 45 订阅

订阅专栏

题目地址: <https://buuoj.cn/challenges#zip>

Challenge 329 Solves

zip

1

拼在一起解下base64就有flag 注意: 得到的 flag 请包上 flag{}
提交

↓ b2ca8799-1...

Flag Submit

<https://blog.csdn.net/mochu7777777>

b2ca8799-13d7-45df-a707-94373bf2800c.zip - Bandizip 6.27

文件(F) 编辑(E) 查找(I) 选项(O) 视图(V) 帮助(H)

打开 解压 新建 添加 删除 测试 查看 代码页

名称	压缩后大小	原始大小	类型	循环冗余检验(CRC)	修改日期
out0.zip	132	132	ZIP 压缩文件	6088211c	2016/8/29 0:07:22
out1.zip	132	132	ZIP 压缩文件	e871366d	2016/8/29 0:07:22
out2.zip	132	132	ZIP 压缩文件	6c37aeb0	2016/8/29 0:07:22
out3.zip	132	132	ZIP 压缩文件	aa71a72d	2016/8/29 0:07:22
out4.zip	132	132	ZIP 压缩文件	43478366	2016/8/29 0:07:22
out5.zip	132	132	ZIP 压缩文件	4e0271ae	2016/8/29 0:07:22
out6.zip	132	132	ZIP 压缩文件	0f3d249f	2016/8/29 0:07:22
out7.zip	132	132	ZIP 压缩文件	f3c16bf9	2016/8/29 0:07:22
out8.zip	132	132	ZIP 压缩文件	7e69ade9	2016/8/29 0:07:22
out9.zip	132	132	ZIP 压缩文件	a4fce05b	2016/8/29 0:07:22
out10.zip	132	132	ZIP 压缩文件	fc7bdffa	2016/8/29 0:07:22
out11.zip	132	132	ZIP 压缩文件	7c48adf1	2016/8/29 0:07:22
out12.zip	132	132	ZIP 压缩文件	1b6f7562	2016/8/29 0:07:22
out13.zip	132	132	ZIP 压缩文件	865ebf48	2016/8/29 0:07:22
out14.zip	132	132	ZIP 压缩文件	68b80ad8	2016/8/29 0:07:22

名称	大小	压缩后大小	原始大小	类型	循环冗余检验(CRC)	修改日期
out0.zip	132	132	4	TXT 文件	75f90d3a	2016/8/29 0:07:22
out1.zip	132	132	4	TXT 文件	10aae3ca	2016/8/29 0:07:22
out2.zip	132	132	4	TXT 文件	ff882104	2016/8/29 0:07:22
out3.zip	132	132	4	TXT 文件	711bde25	2016/8/29 0:07:22
out4.zip	132	132	4	TXT 文件	f7937ca8	2016/8/29 0:07:22
out5.zip	132	132	4	TXT 文件	1c063911	2016/8/29 0:07:22
out6.zip	132	132	4	TXT 文件	93f5319b	2016/8/29 0:07:22
out7.zip	132	132	4	TXT 文件	edea78ce	2016/8/29 0:07:22
out8.zip	132	132	4	TXT 文件	6e946dcb	2016/8/29 0:07:22
out9.zip	132	132	4	TXT 文件	f2d1101c	2016/8/29 0:07:22

<https://blog.csdn.net/mochu7777777>



<https://blog.csdn.net/mochu7777777>

很多压缩包，但是里面的内容非常小，小于5字节，可以尝试使用 **CRC32爆破** 得到其内容

先以 **out0.zip** 做个例子，**out0.zip** 的 **CRC32** 校验码为：**0x75f90d3a**

```
PS D:\Tools\Misc\crc32> python .\crc32.py reverse 0x75f90d3a
4 bytes: {0x7a, 0x35, 0x42, 0x7a}
verification checksum: 0x75f90d3a (OK)
alternative: 2BHS9N (OK)
alternative: 4zPLa0 (OK)
alternative: 7gJsJx (OK)
alternative: 9hUCvv (OK)
alternative: Gzefs4 (OK)
alternative: MqgwNY (OK)
alternative: QoY76E (OK)
alternative: RORDuU (OK)
alternative: VK0Et6 (OK)
alternative: XDPUH8 (OK)
alternative: bmE3S_ (OK)
alternative: guV_H1 (OK)
alternative: j6113k (OK)
alternative: jgSP_w (OK)
alternative: zV8BC1 (OK)
PS D:\Tools\Misc\crc32> php -r "var_dump(hex2bin('7a35427a'));"
string(4) "z5Bz"
```

这样就可以爆破出 **out0.zip** 中 **data.txt** 的内容，然后利用网上找的一个脚本

```

#python3
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for k in dic:
                for h in dic:
                    s = i + j + k + h
                    if crc == (binascii.crc32(s.encode())):
                        f.write(s)
                        return

def CrackZip():
    for i in range(0,68):
        file = 'out'+str(i)+'.zip'
        crc = zipfile.ZipFile(file, 'r').getinfo('data.txt').CRC
        CrackCrc(crc)
        print('\r'+ "loading: {:.%}" .format(float((i+1)/68)),end='')

dic = string.ascii_letters + string.digits + '+/='
f = open('out.txt', 'w')
print("\nCRC32begin")
CrackZip()
print("CRC32finished")
f.close()

```

得到所有压缩包的 `data.txt` 中的内容。并拼接在一起

```

z5BzAAANAAAAAAAAAKo+egCAIwBJAAAAVAAAAAKGNkv+a2MdSR0zAwABAAAAQ01UCRUUy91BT5UkSNPoj5hFEVfBRvefHSBCfG0ruGnKnygsMyj8
SBaZHxsYHY84LEZ24cXtZ01y3k1K1YJ0vpK9HwqUzb6u9z8igEr3dCCQLQAdAAAAHQAAAAJi0efVT2MdSR0wCAAgAAAAZmxhZy50eHQAsDRpZmZp
eCB0aGUgZmlsZSBhbmQgZ2V0IHRoZSBmbGFnxD17AEAHAA==

```

使用这个网站在线解密base64: <https://the-x.cn/base64>

Base64 在线解码、编码



常规Base64

CSS Base64

DES加密/解密

3DES加密/解密

AES加密/解密

RSA加密/解密

点击关闭

```
z5BzAAANAAAAAAAAAKo+egCAIwBJAAAAVAAAAKGNKv+a2MdSR0zAwABAAAAQ01UCRUUy91BT5UKSNPoj5hFEVFBFvfeHSBCfG0ruGnKnygsMyj
8SBaZHxsYHY84LEZ24cXtZ01y3k1K1YJ0vpK9HwqUzb6u9z8igEr3dCCQLQAdAAAAHQAAAAJi0efVT2MdSR0wCAAgAAAAZmxhZy50eHQAsDRpZmZpe
CB0aGUgZmlsZSBhbmQgZ2Z0IHRoZSBmbGFuZD17AEAHAA==
```

编码源格式: 文本 Hex 解码结果:

自动检测

中文编码:

UTF-8

编码

解码

```
00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
-----
CF 90 73 00 00 0D 00 00 00 00 00 00 00 00 AA 3E 7A | ..s.....>z
00 80 23 00 49 00 00 00 54 00 00 00 02 86 34 AB | ..#.I...T....4.
FE 6B 63 1D 49 1D 33 03 00 01 00 00 00 43 4D 54 | .kc.I.3.....CMT
09 15 14 CB DD 41 4F 95 24 48 D3 E8 8F 98 45 11 | .....AO.$H....E.
51 41 46 F7 9F 1D 20 42 7C 6D 2B B8 69 CA 9F 28 | QAF... B|m+.i..(
2C 33 28 FC 48 16 99 1F 1B 18 1D 8F 38 2C 46 76 | ,3(.H.....8,Fv
E1 C5 ED 67 4D 72 DE 4D 4A D5 82 74 BE 92 BD 1F | ...gMr.MJ..t....
0A 94 CD BE AE F7 3F 22 80 4A F7 74 20 90 2D 00 | .....?"J.t.-.
1D 00 00 00 1D 00 00 00 02 62 D1 E7 D5 4F 63 1D | .....b...Oc.
49 1D 30 08 00 20 00 00 00 66 6C 61 67 2E 74 78 | I.0...flag.tx
74 00 B0 34 69 66 66 69 78 20 74 68 65 20 66 69 | t..4ifix the fi
6C 65 20 61 6E 64 20 67 65 74 20 74 68 65 20 66 | le and get the f
6C 61 67 C4 3D 7B 00 40 07 00 | lag.={.0..
```

未能识别的数据

当前编码: [Hex + Ascii]

数据长度: 202 Bytes

插件数: 18, 耗时: 1ms

<https://blog.csdn.net/mochu77777777>

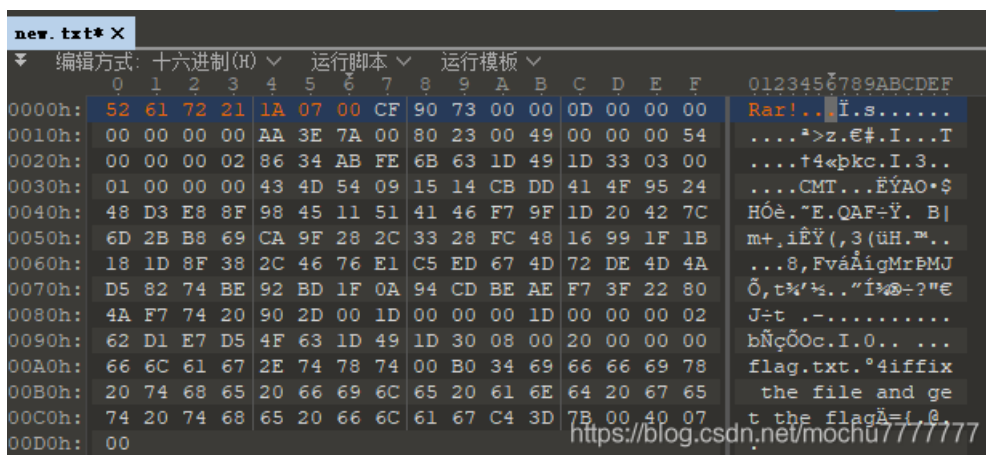
发现base64解出来貌似是字节流数据，而且结尾是RAR文件的结尾，使用脚本将base64解码并以字节流的形式写入一个新的文件

```
import base64

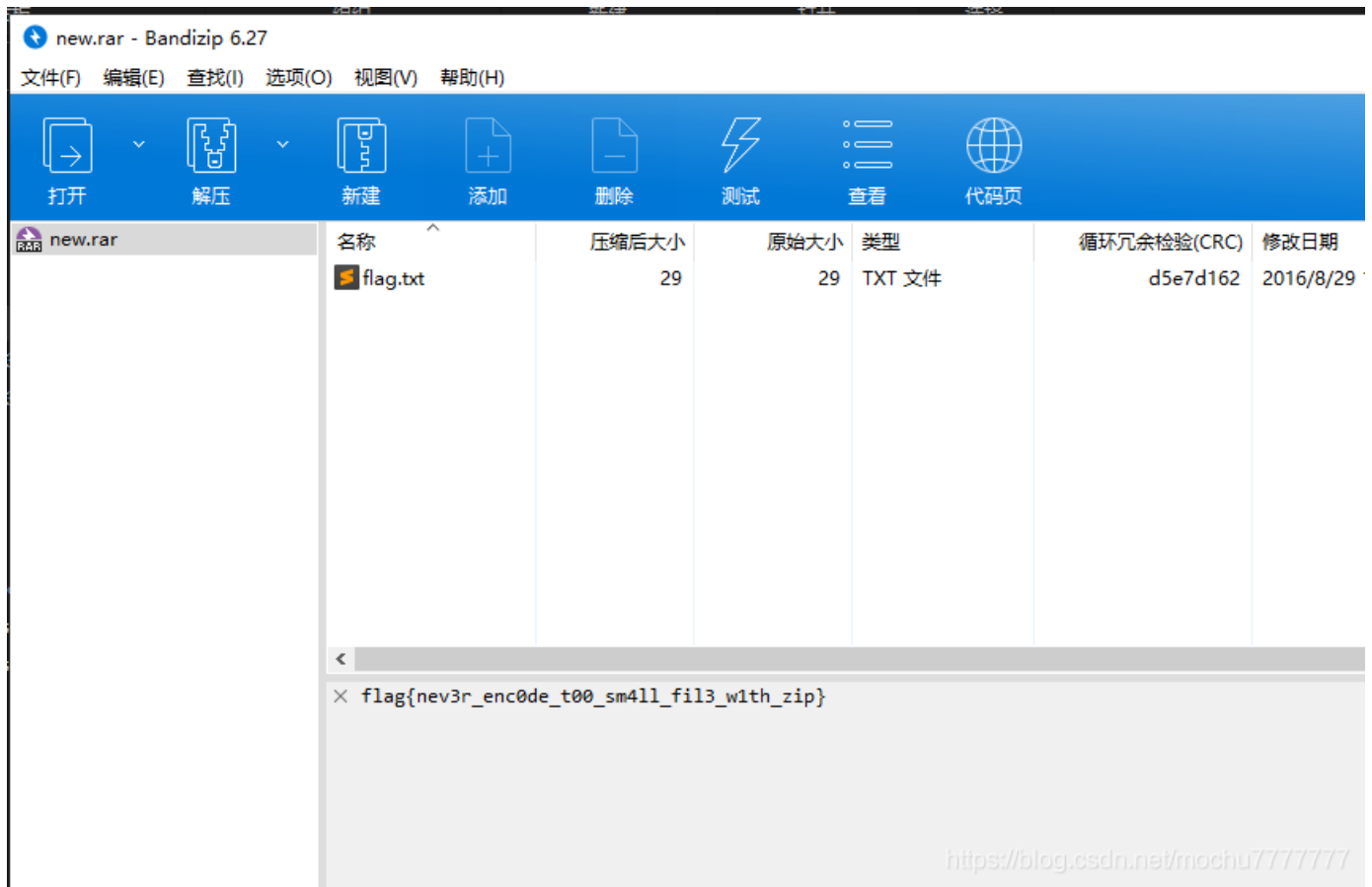
base64_text = open('out.txt', 'r').read()
byte_stream = base64.b64decode(base64_text)
open('new.txt', 'wb').write(byte_stream)
```

使用 010 Editor 打开，添加一个RAR文件的头部

```
52 61 72 21 1A 07 00
```



保存，修改后缀为 rar



在注释中发现flag

```
flag{nev3r_enc0de_t00_sm4ll_fil3_w1th_zip}
```