BUUCTF misc zip

原创

Warning 于 2019-10-05 10:35:10 发布
分类专栏： 杂项

这道题跟之前在bugku上做的“好多压缩包”是一样的，但当时做并没有好好理解，现在再总结一下。

涉及到的知识点：

- CRC校验爆破
- rar文件头尾的熟悉
- notepad++插件的使用

首先解压发现题目给了68个压缩包……68个，提示的是：

好吧，那我们就看看压缩包里是什么，然后发现压缩包有密码，无法解压，用winhex查看后发现也并不是伪加密，而文中也没有关于压缩包密码的提示。

关于加密的zip压缩包，一般有以下几种情况（后续会慢慢补充）：

- 伪加密，用winhex查看全局加密标志和局部加密标志
- 弱密码，题目中会有提示或者给出字典，直接爆破
- zip + 图片，一般是隐写题目，从图片中找出密码
- CRC32爆破，一般用于压缩包里文件内容较短时

查看压缩包里的文件，发现只有4个字节，考虑CRC32校验爆破

CRC32校验爆破原理说明：

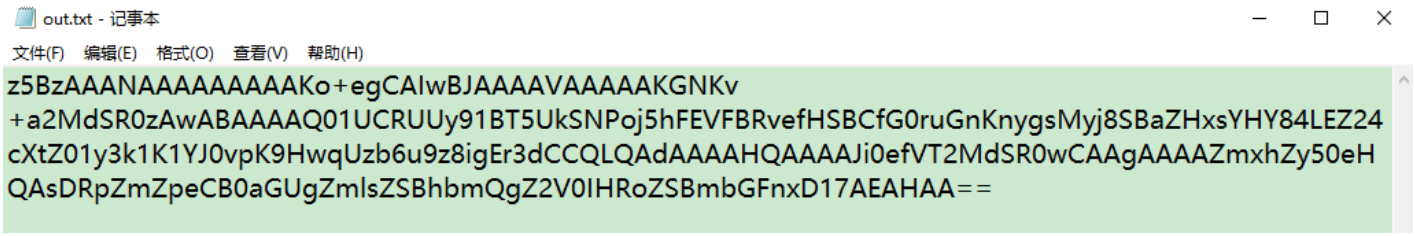
CRC32:CRC本身是“冗余校验码”的意思，CRC32则表示会产生一个32bit（8位十六进制数）的校验值。

在产生CRC32时，源数据块的每一位都参与了运算，因此即使数据块中只有一位发生改变也会得到不同的CRC32值，利用这个原理我们可以直接爆破出加密文件的内容。

爆破脚本如下（来自网上的大牛）：

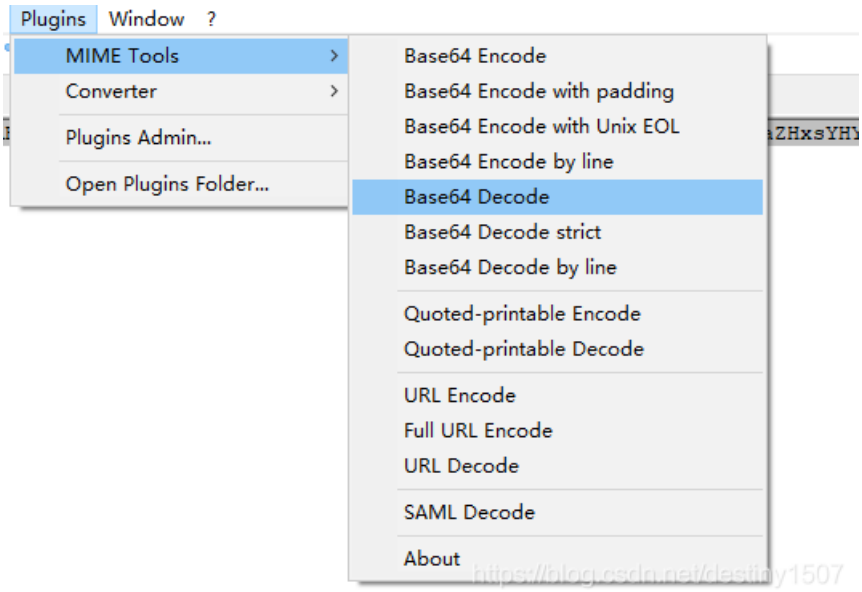
```
import zipfile
import string
import binascii

def CrackCrc(crc):
    for i in dic:
        for j in dic:
            for p in dic:
                for q in dic:
                    s = i + j + p + q
                    if crc == (binascii.crc32(s) & 0xffffffff):
                        #print s
                        f.write(s)
                        return

def CrackZip():
    for I in range(68):
        file = 'out' + str(I) + '.zip'
        f = zipfile.ZipFile(file, 'r')
        GetCrc = f.getinfo('data.txt')
        crc = GetCrc.CRC
        #以上3行为获取压缩包CRC32值的步骤
        #print hex(crc)
        CrackCrc(crc)

dic = string.ascii_letters + string.digits + '+/='

f = open('out.txt', 'w')
CrackZip()
f.close()
```

然后我们发现得到了一串base64编码的字符串：

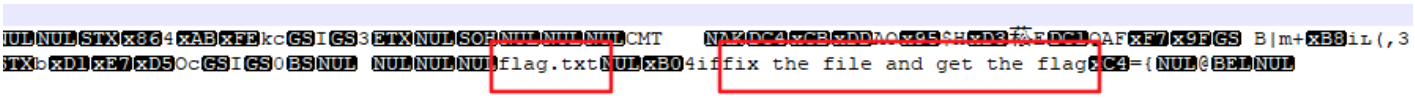


这里有一个知识点，如果用在线base64解码之后再复制，会出现误差，如果用脚本进行解码，总是会多一个字节。（我就是卡在这里卡了好久……）

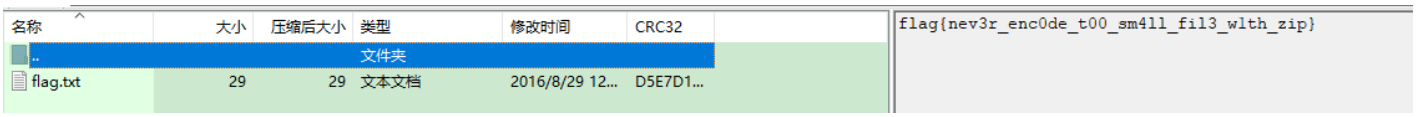
后来百度知道notepad++有一个解码的插件，解码后直接保存就不会有问题了：



发现文件内容中能看清的：有一个flag.txt和一段话



提示我们接下来需要修复文件，用winhex打开，发现了rar文件的文件尾，但是却没有文件头，加上之后修改文件后缀，在注释里就可以看到flag;



文件头：526172211A0700

文件尾：C43D7B00400700
