# i春秋第二届春秋欢乐赛CryMisc题目WriteUp

转载

iqiqiya 于 2018-04-24 19:49:16 发布 4043 收藏
分类专栏： ------i春秋第二届春秋欢乐赛 我的CTF进阶之路 文章标签： i春秋第二届春秋欢乐赛CryMisc题目WriteUp CryMisc题目WriteUp CryMisc解题思路 i春秋第二届春秋欢乐赛CryMisc题目解题思路


------i春秋第二届春秋欢乐赛 同时被 2 个专栏收录
3 篇文章 0 订阅
订阅专栏
我的CTF进阶之路
108 篇文章 18 订阅
订阅专栏

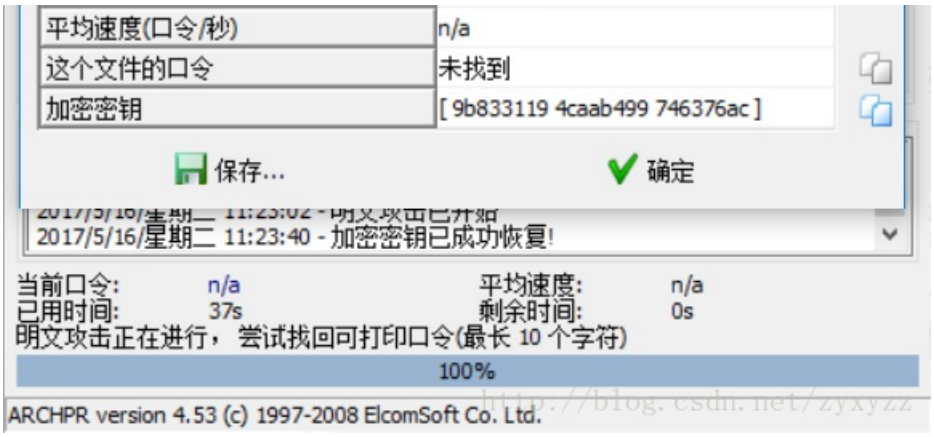
```
CryMisc__writeup
```

把CryMisc.zip解密后得到crypto.zip和jiami.py，crypto.zip是被加密过的，而jiami.py则是加密脚本，

```python
# -*- coding:utf8 -*-

import pyminizip
from hashlib import md5
import os

def create(files, zfile):
    password = os.urandom(15)
    password = md5(password).hexdigest()
    pyminizip.compress_multiple(files, zfile, password, 0)
    pass

if __name__ == '__main__':
    files = ['jiami.py','gogogo.zip']
    zfile = 'crypto.zip'
    create(files, zfile)
```



其中密码是随机的15位字符的md5值，基本不可能爆破，而这里是使用本身这个jiami.py和gogogo.zip一起加密成crypto.zip，这里就存在着zip的明文攻击，用winrar（不能用7zip或者好压）把jiami.py压缩成jiami.zip，使用archpr选择“明文攻击”




因为密码长度过长，导致无法直接获取到，但是点击“确定”可以另存为已经解密好的crypto_decrypted.zip，解压其中的gogogo.zip得到AESencrypt.py、AES.encryt、RSA.encrypt，其中AESencrypt.py内容是模仿勒索软件，先AES加密，然后把key通过RSA进行加密。这里RSA文件只需要先分解即可，直接上脚本。

```python
# -*- coding:utf8 -*-
from Crypto.Cipher import AES
import gmpy2

c=long(open('RSA.encrypt','rb').read().encode('hex'),16)
n=0x48D6B5DAB6617F21B39AB2F7B14969A7337247CABB417B900AE1D986DB47D971
e=0x10001
#通过yafu来分解n
p=185783328357334813222812664416930395483
q=177334994338425644535647498913444186659
d=gmpy2.invert(e,(p-1)*(q-1))
m=pow(c,d,n)
key='{:x}'.format(m).decode('hex')

s=open('AES.encryt','rb').read()
obj=AES.new(key,AES.MODE_ECB)
s=obj.decrypt(s)
BS=16
pad_len=ord(s[-1])
s=s[:-1*pad_len]
with open('next.zip','wb') as f:
    f.write(s)
```



运行后可以得到AES的key为copy__white__key（这里也是稍微给后面的解密一点提示）、得到next.zip，解压后得到encrypt.py、first、second，其中encrypt.py就是对flag.jpg文件进行zip处理为2个文件。只要熟悉python代码，写个解密代码很简单。

```python
# -*- coding:utf8 -*-
from base64 import *

s=[0,1]
with open('first','rb') as f:
    s[0]=b16encode(f.read())
with open('second','rb') as f:
    s[1]=b16encode(f.read())
s=map(''.join,zip(*s))
s=b16decode(''.join(s))
with open('flag.jpg','wb') as f:
```

```
12      f.write(s)
```



得到flag.jpg


这里我为了容易做，使用了一张flag的百度百科上面的图片，如果可以细心的看到图片左下角的百度百科字样，去下载原图，然后对比后发现flag.jpg后面多出的数据其实是一个psd文件（原理是用copy命令把jpg文件和psd文件合并在一起）

```
00021F20  B3 C9 73 24 91 CB F6 75  87 CB 89 8F CB 1E 17 AE  ³Es$ ËöuËÉ Ë  ®
00021F30  39 E4 FA FA 0A 1B B8 1B  B4 00 50 01 40 05 00 14  9äúú  ¸ ´ P @
00021F40  00 50 01 40 05 00 7F FF  D9 38 42 50 53 00 01 00   P @   ÿÙ8BPS
00021F50  00 00 00 00 00 00 03 00  00 01 90 00 00 01 90 00
00021F60  08 00 03 00 00 00 00 00  00 71 44 38 42 49 4D 04          qD8BIM
00021F70  04 00 00 00 00 00 0F 1C  01 5A 00 03 1B 25 47 1C          Z   %G
00021F80  02 00 00 02 00 00 00 38  42 49 4D 04 25 00 00 00        8BIM %
00021F90  00 00 10 CD CF FA 7D A8  C7 BE 09 05 70 76 AE AF     ÍÏú}¨Ç¾  pv®¯
00021FA0  05 C3 4E 38 42 49 4D 04  24 00 00 00 00 3C B1 3C   ÃN8BIM $     <±<
00021FB0  3F 78 70 61 63 6B 65 74  20 62 65 67 69 6E 3D 22  ?xpacket begin="
```



也可以通过8BPS头和8BIM字样推测是psd文件，用photoshop打开


最顶层的文字是假的，这里关键在于锁定的“背景”层，看似是新建图片时所留下的默认背景图，而本题就是把flag隐藏在里面，把上面2层隐藏掉，然后对背景色另存为png格式（这样才能完好的保留颜色），使用stegsolve打开，并按下向左的按钮


就呈现出一幅二维码，扫描就得到flag{409d7b1e-3932-11e7-b58c-6807154a58cf}

这图的原理是前景色为(255,255,254)，人眼无法识别它跟白色的区别，而如果使用photoshop的油漆桶填充的时候，注意把默认的容差32改为容差0才可以看出区别。

本文转载自https://blog.csdn.net/zyxyzz/article/details/72629354