

# i春秋 第二届春秋欢乐赛 CryMisc Writeup

原创

XhyEax 于 2019-01-26 21:22:19 发布 550 收藏 1

分类专栏: [CTF](#) 文章标签: [CTF CryMisc i春秋 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/XhyEax/article/details/86661259>

版权



[CTF 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

## 解压

把压缩文件解压, 得到 `crypto.zip` 和 `jiami.py`, 而前者中包含有 `jiami.py`, 构造一个zip出来查看CRC, 发现是同一文件, 所以进行明文攻击。

然后将其中的 `gogogo.zip` 解压, 得到三个文件

| 名称   | 修改日期            |
|--|-----------------|
|  AES.encrypt   | 2017/5/15 17:24 |
|  AESencrypt.py | 2017/5/15 17:24 |
|  RSA.encrypt   | 2017/5/15 17:24 |

## 查看代码

打开 `AESencrypt.py`, 查看代码:

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

s=open('next.zip','rb').read()
BS=16
pad_len=BS-len(s)%BS
padding=chr(pad_len)*pad_len
s+=padding

key='我后来忘了'
n=0x48D6B5DAB6617F21B39AB2F7B14969A7337247CABB417B900AE1D986DB47D971
e=0x10001
m=long(key.encode('hex'),16)
c=pow(m,e,n)
c='0{:x}'.format(c).decode('hex')
with open('RSA.encrypt','wb') as f:
    f.write(c)

obj=AES.new(key,AES.MODE_ECB)
with open('AES.encrypt','wb') as f:
    f.write(obj.encrypt(s))
```

由 `c='0{:x}'.format(c).decode('hex')` 发现是python2的代码（似乎还多写了个0...）

发现key被RSA加密了，给出了n和e，将n分解（当p、q的取值差异过大或过于相近的时候，使用yafu可以快速的把n值分解出p、q值）

```
***factors found***
P39 = 177334994338425644535647498913444186659
P39 = 185783328357334813222812664416930395483
ans = 1
```

解出两个质数是 177334994338425644535647498913444186659 和 185783328357334813222812664416930395483

使用 gmpy2 解出 `d = 21459038613121460434132216103140795081593356519819592462521069311922260546829`

```
d = gmpy2.invert(e, (p-1)*(q-1))
#21459038613121460434132216103140795081593356519819592462521069311922260546829
```

（其实用 RSA-Tool2 也行，这样就没必要装 gmpy2 了）

以16进制读取 RSA.encrypt 文件，得到 68c2e12fadebbd344e82fa9e1eac0f0bde5aecbd7840f18352cf761f872233d 再转化为数字

## 转化代码

```
#python3
byte = open("RSA.encrypt", "rb").read()
hexstr = binascii.b2a_hex(byte).decode("utf-8")
c = int(hexstr,16)
```

```
#python2.7
c = int(open("RSA.encrypt", "rb").read().encode('hex'),16)
```

使用 RSA-Tool2 （需要把字母转为大写，神奇的bug...）或使用python代码解密

## 解密key

```
#python3
from Crypto.Cipher import AES
import binascii

def HextoAscii(hexnum):
    hexStr = str(hexnum).replace("0x", "")
    return binascii.a2b_hex(hexStr).decode("utf-8")

c = 0x068C2E12FADEBB344E82FA9E1EAC0F0BDE5AECBD7840F18352CF761F872233D
n = 0x48D6B5DAB6617F21B39AB2F7B14969A7337247CABB417B900AE1D986DB47D971
e = 0x10001
p = 185783328357334813222812664416930395483
q = 177334994338425644535647498913444186659
d = 21459038613121460434132216103140795081593356519819592462521069311922260546829
m=pow(c,d,n)
print(m)
hexnum = hex(m)
print(HextoAscii(hexnum))
```

```
#python2.7
from Crypto.Cipher import AES

c = 0x068c2e12fadebbd344e82fa9e1eac0f0bde5aecbd7840f18352cf761f872233d#read RSA.encrypt
n = 0x48D6B5DAB6617F21B39AB2F7B14969A7337247CABB417B900AE1D986DB47D971
e = 0x10001
p = 185783328357334813222812664416930395483
q = 177334994338425644535647498913444186659
d = 21459038613121460434132216103140795081593356519819592462521069311922260546829
m=pow(c,d,n)
print(m)
key = "{:x}".format(m).decode('hex')
print(key)
```

m=132172197780003798270878941356862694777

16进制就是 636F70795F5F77686974655F5F6B6579

HexDecode得到 copy\_\_white\_\_key

## 解密AES.crypt文件

修改一下 AEsencrypt.py，然后运行，得到 next.zip

代码如下：


```
from Crypto.Cipher import AES


s=open('AES.encrypt','rb').read()
BS=16
pad_len=BS-len(s)%BS
padding=chr(pad_len)*pad_len
s+=padding


key='copy__white__key'

obj=AES.new(key,AES.MODE_ECB)
with open('next.zip','wb') as f:
    f.write(obj.decrypt(s))
```

解压缩 next.zip 得到三个文件

 [encrypt.py](#)

 first

 second

查看 encrypt.py，代码如下

```
from base64 import *

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

发现是把flag.jpg拆成两部分，都使用base16 decode了一次

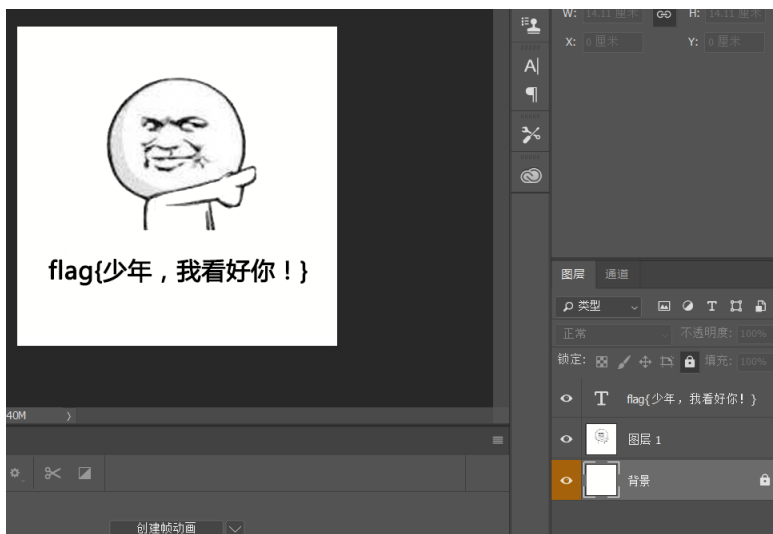
## 合并出flag.jpg

```
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:
    f.write(s)
```

喂！老子在问你话啊！ 求婚戒指都准备好了 这段时间我过的很开心 看，妈妈，那是什么？  
等一切结束后，我有些话想跟你说 犯人也在我们之中，怎么可能一起睡！我回我自己房间去  
你看Jacky的鼻子很大对吧 哟！这位小哥匆匆忙忙要去哪儿啊？ 小西克幸  
谁敢杀我？谁敢杀我？谁敢杀我？ 才一个人也敢说出这种大话 XXX由我来保护！  
答应我，一定要好好活下去 这次的工作的报酬是以前无法比较的  
这里就交给我，你们快走！ 嗯？从未见过的武器？ 什么声音？去看一下吧  
自爆模式启动 明天是女儿的生日啊 我已经是天下无敌啦啦啦 顺便一提，我家的阳台不太稳固  
等这场战争结束，俺就要回老家结婚了！  
你这家伙是什么人？ 到了天竺之后，首先得把肚子填饱  
回想中 明天就去约她出来吧 什么嘛，原来是错觉  
区区人类 咦？... 怎么样？我说过没什么异常的吧 这次做完就金盆洗手  
已经死了么 CV很贵 嗯？老鼠么 成功了吗？ 这工作结束后我们两人一起生活吧  
我没想杀人，都、都是因为那家伙不好！全都是那家伙的错！所以我才【略  
吕、吕布啊啊啊啊啊啊啊啊啊啊啊啊！ 桑岛法子 大丈夫だ 问题ない  
叫破喉咙也不会有人来救你的  
原来是这么回事！得去告诉他们 放心，这艘船非常坚固，绝对不可能沉没的！

(可自行保存分析)

按照惯例，肯定是藏在文件尾，010Editor打开，提取出尾部文件  
发现是psd文件



key为 `copy__white__key`，这是一个提示，把背景导出为png

[查看bg.png](#)

使用stegsolve打开，点一下左箭头，得到一个二维码，解码得到flag

参考：

[CryMisc\\_\\_writeup](#)

[CTF中RSA套路](#)

[RSA史上最强剖析，从小白变大神，附常用工具使用方法及CTF中RSA典型例题](#)