

BUUCTF：黑客帝国

原创

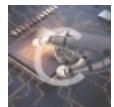
未初 于 2020-11-01 16:31:31 发布 2115 收藏 2

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/mochu777777/article/details/109429148>

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#%E9%BB%91%E5%AE%A2%E5%B8%9D%E5%9B%BD>

Challenge

339 Solves

X

黑客帝国

1

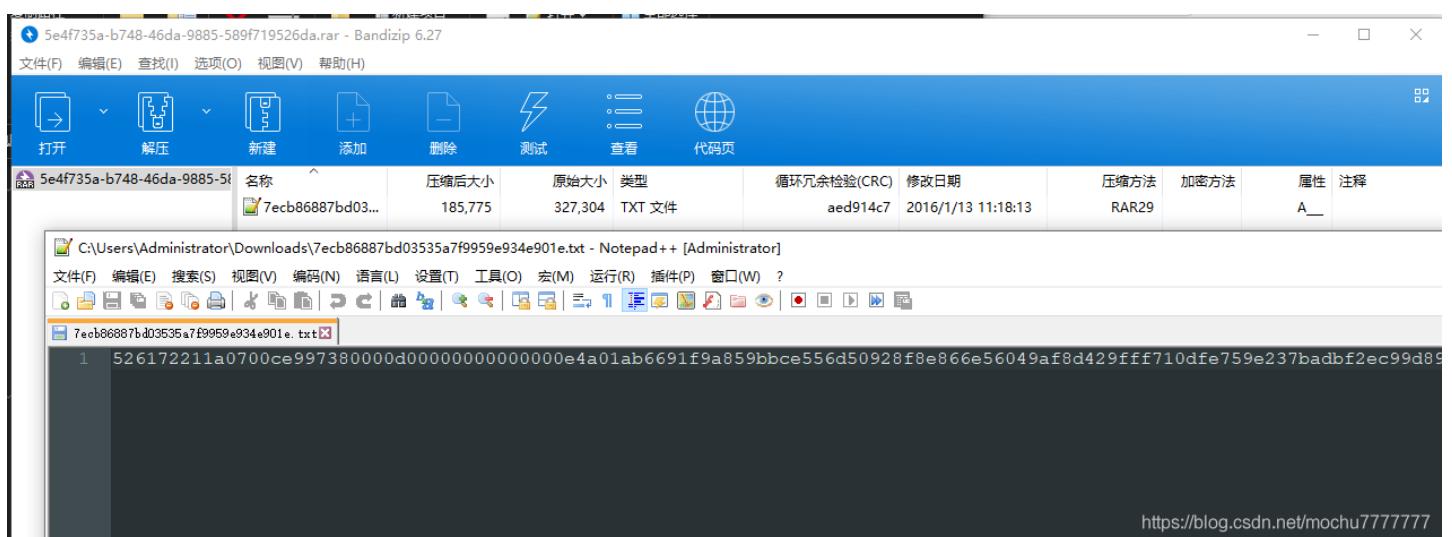
Jack很喜欢看黑客帝国电影，一天他正在上网时突然发现屏幕不受控制，出现了很多数据再滚屏，结束后留下了一份神秘的数据文件，难道这是另一个世界给Jack留下的信息？聪明的你能帮Jack破解这份数据的意义吗？注意：得到的 flag 请包上 flag{} 提交

 5e4f735a-b7...

Flag

Submit

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

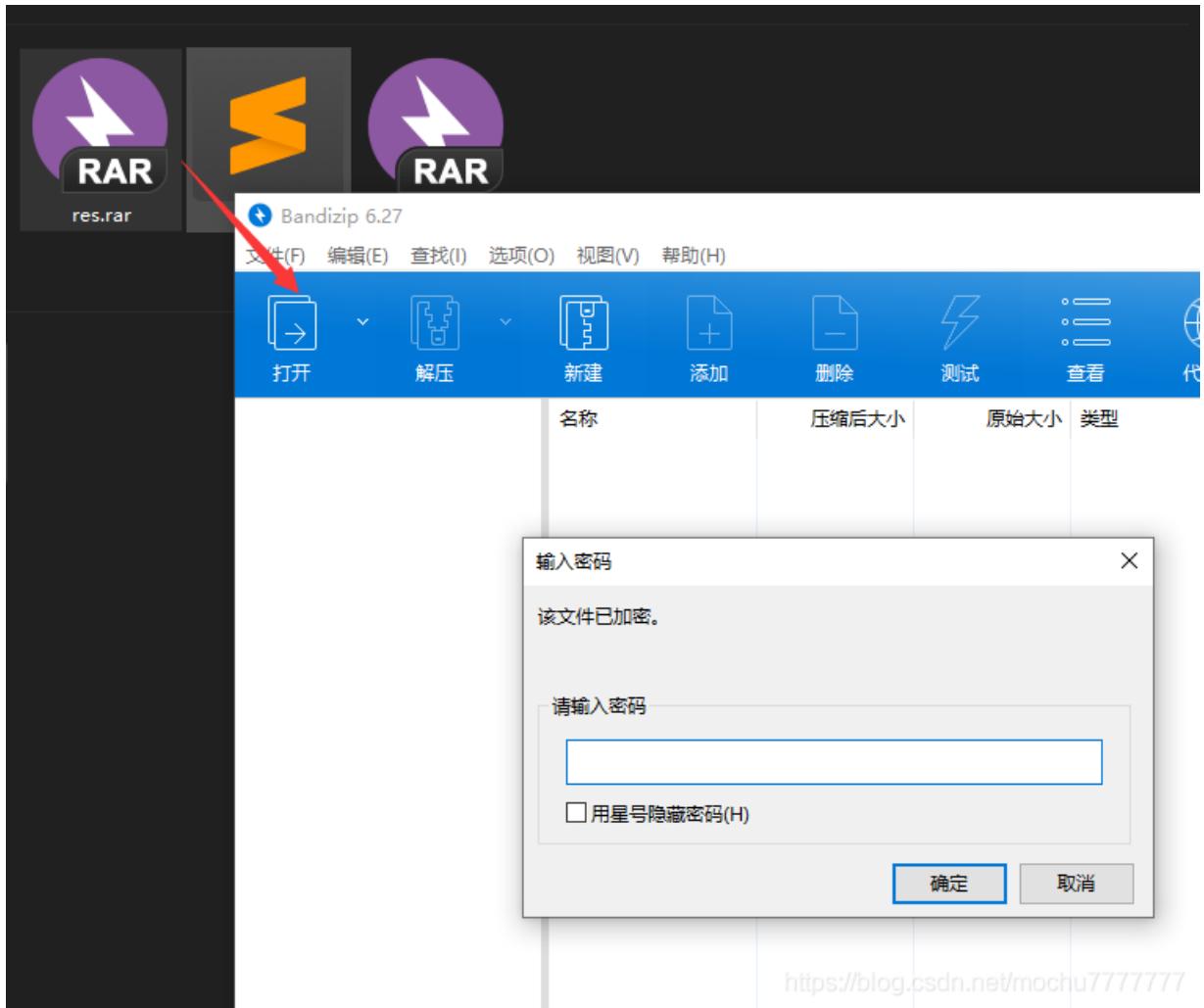


开头很像 RAR 文件的十六进制数据，利用脚本将这些数据以二进制流形式写入文件

```
import binascii

hex_data='这里填十六进制数据'
out=open('res.rar','wb')
out.write(binascii.unhexlify(hex_data))
out.close()
```

得到被加密的 `res.rar`



ARCHPR 尝试爆破，老样子没提示的话先尝试掩码爆破四位数字

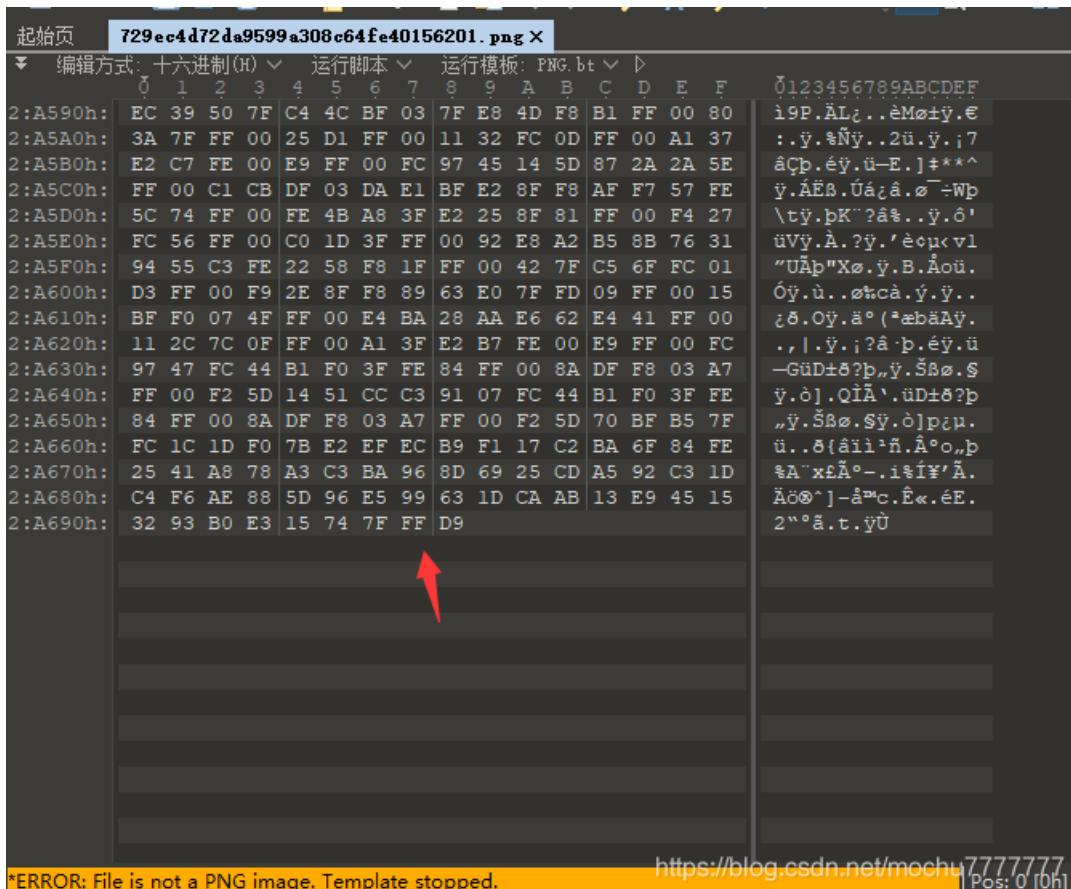
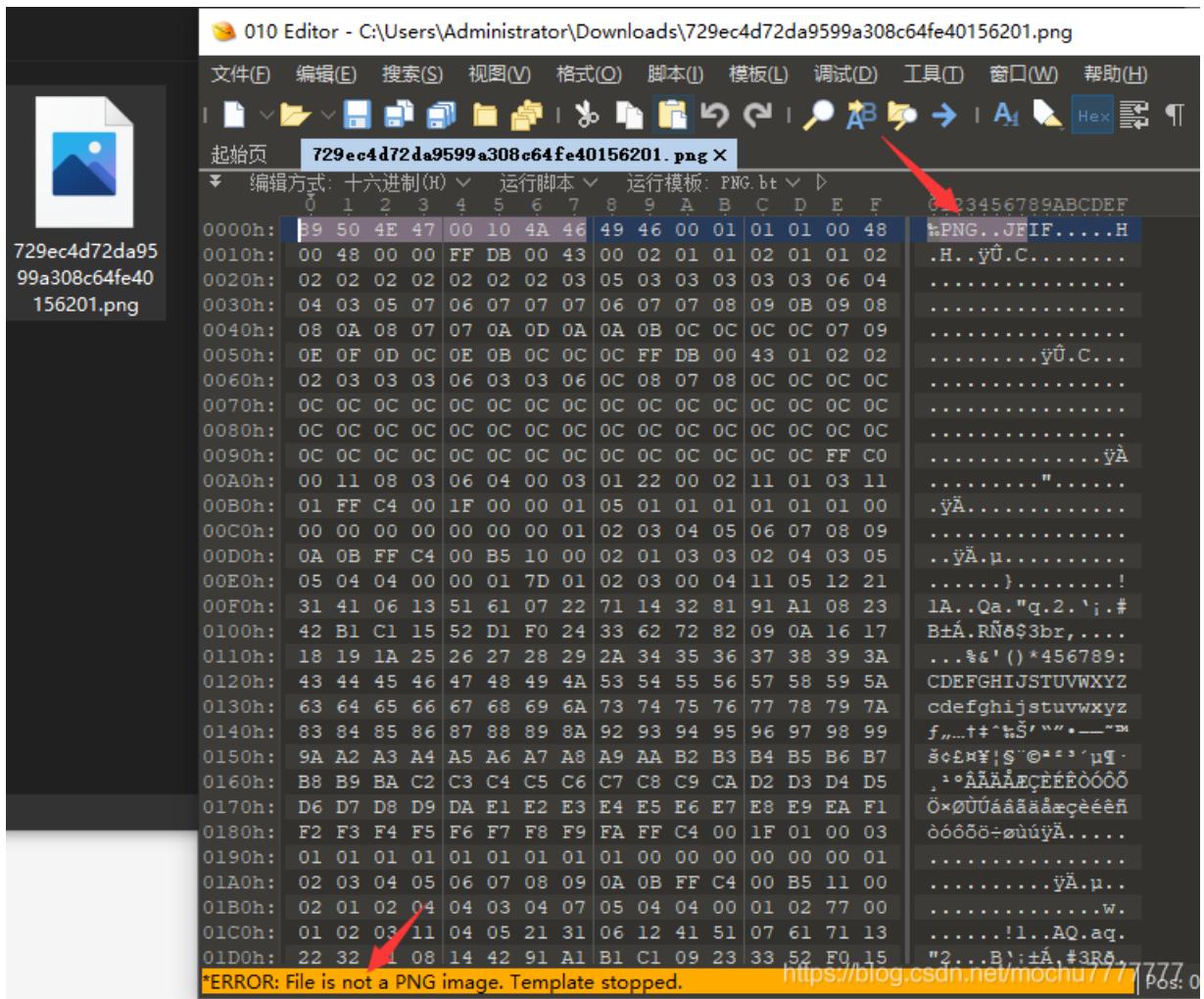
A screenshot of the ARCHPR 4.54 password recovery tool. On the left, a modal dialog box titled "口令已成功恢复!" (Password recovered successfully!) displays "Advanced Archive Password Recovery 统计信息:" (Statistics:). It shows the following data:

总计口令	3,693
总计时间	11s 955ms
平均速度(口令/秒)	308
这个文件的口令	3690
十六进制口令	33 36 39 30

At the bottom of this dialog are "保存..." (Save...) and "确定" (OK) buttons. On the right, the main ARCHPR interface shows the following details:

- File: C:\Users\Administrator\Downloads\res.rar
- Attack Type: Mask
- Characteristics: Range, Length, Dictionary, Plain Text, Auto Save, Options, Advanced
- Mask Settings: Mask: ???? (highlighted)
- Status Window:
 - 2020/11/1 16:24:11 - 文件"C:\Users\Administrator\Downloads\res.rar"已打开。
 - 2020/11/1 16:24:11 - 开始掩码攻击...
 - 2020/11/1 16:24:23 - 口令已成功恢复!
 - 2020/11/1 16:24:23 - '3690' 是这个文件的一个有效口令
- Statistics:
 - 当前口令: 3690
 - 平均速度: 308 p/s
 - 已用时间: 11s
 - 剩余时间: 20s
 - 口令长度 = 4, 总计: 10,000, 已处理: 3,693
 - 进度: 36%

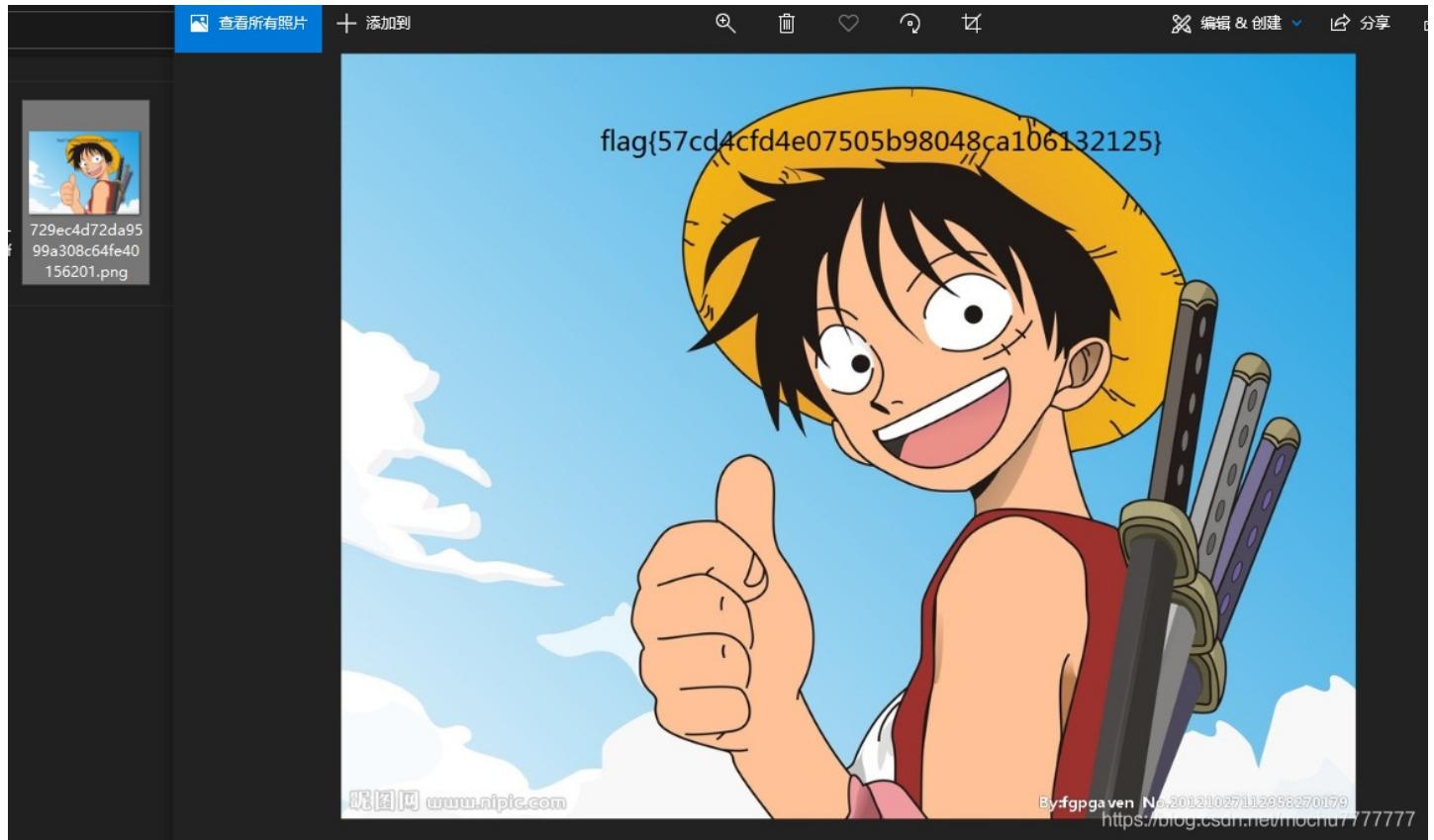
成功得到口令: 3690



根据以上信息可知这是一张 `jpg` 的图片被加了 `PNG` 的头，修改成 `JPG` 的头：`FF D8`

起始页 729ec4d72da9599a308c64fe40156201.png*																
		编辑方式: 十六进制(H)		运行脚本		运行模板: PNG, bt										
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000h:	FF	DB	4A	46	49	46	00	01	01	01	00	48	00	48	00	00
0010h:	FF	DB	00	43	00	02	01	01	02	01	01	02	02	02	02	02
0020h:	02	02	02	03	05	03	03	03	03	03	06	04	04	03	05	07
0030h:	06	07	07	07	06	07	07	08	09	0B	09	08	08	0A	08	07
0040h:	07	0A	0D	0A	0A	0B	0C	0C	0C	0C	07	09	0E	0F	0D	0C
0050h:	0E	0B	0C	0C	0C	FF	DB	00	43	01	02	02	02	03	03	03
0060h:	06	03	03	06	0C	08	07	08	0C							
0070h:	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C	0C

保存即可得到正常的图片



flag{57cd4cf4e07505b98048ca106132125}