第二届BJD CTF做题总结与题目复现-MISC&Crypto

原创

Qwzf 于 2020-03-25 23:46:03 发布 1215 收藏 8

分类专栏： CTF BJDCTF 文章标签： CTF

本文链接：https://blog.csdn.net/qq_43625917/article/details/105058217

0x00 前言

上周参加了第二届BJD CTF，本Web dog太垃圾，就做出两道Web。不过还好MISC和Crypto做的还行。那就先总结复现一下。

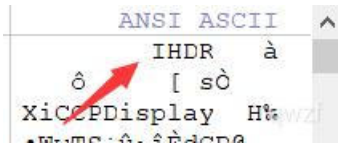
标 * 表示未作出的

0x01 MISC

这个做的还行，不过有三道未做出(其中一道是mikutap。听，然后找按键盘进行对应，这个就算了，不想做)，好了开始复现

最简单的misc-y1ng

就是简单的文件头填充。



看到IHDR，很明显想到png图片。添加文件头89504E47，改后缀，得到16进制

16进制转字符串即可得到flag

A_Beautiful_Picture

改图片高度即可

EasyBaBa

这个就比较有趣了，直接用winrar打开得到`里面都是出题人.jpg`，进行格式分析发现是avi视频文件，改后缀播放，发现有东西，扔到Pr逐帧分析得到四张二维码。扫描得到4串十六进制
6167696E5F6C
6F76655F59
424A447B696D
316E677D
Hex->ASCII排序即可。

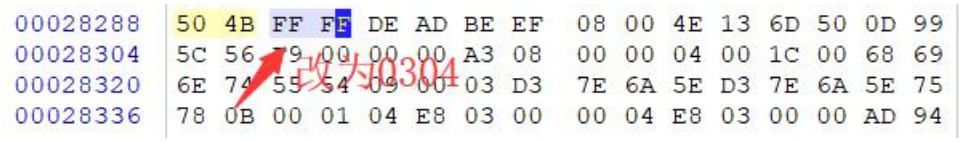
小姐姐-y1ng

winhex搜索BJD即可

*Real_EasyBaBa

这个当时没做出来。看了官方wp发现最后一步去除所有0没有想到。。。。

png文件用winhex打开，搜索504B，将FFFF改为0304(可以先用binwalk分析下发现有zip，因为之前我做过类似的，所以就直接改),直接winrar打开



解压hint文件。打开是二维码，扫描得到命令`od -vtx1 ./draw.png | head -56 | tail -28`

linux之od命令

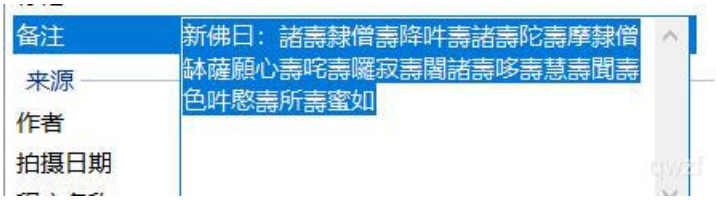
然后执行`od -vtx1 ./ezbb_r.png | head -56 | tail -28`这个命令，得到

```
root@qwzf:/test# od -vtx1 ./ezbb_r.png | head -56 | tail -28
0000700 01 00 02 10 03 10 00 00 01 ee c0 b8 a6 00 00 00
0000720 ff ff ff 00 ff ff ff ff 00 ff ff 00 00 00 ff ff
0000740 ff 00 ff 00 00 00 ff 00 00 ff 00 ff 00 00 ff 00
0000760 ff 00 ff 00 00 00 ff 00 00 ff 00 ff 00 00 ff 00
0001000 ff ff 00 00 00 00 ff 00 00 ff 00 ff 00 ff 00 00
0001020 ff 00 ff 00 00 00 ff 00 00 ff 00 ff 00 00 ff 00
0001040 ff 00 ff 00 ff 00 ff 00 00 ff 00 ff 00 00 ff 00
0001060 ff ff ff 00 ff ff ff 00 00 ff ff 00 00 00 ff ff
0001100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001120 ff ff ff 00 ff ff ff 00 ff ff ff 00 ff ff 00 00
0001140 ff 00 00 00 00 00 ff 00 00 00 ff 00 00 ff 00 00
0001160 ff ff ff 00 00 00 ff 00 ff ff ff 00 00 ff 00 00
0001200 00 00 ff 00 00 00 ff 00 ff 00 00 00 00 ff 00 00
0001220 ff ff ff 00 00 00 ff 00 ff ff ff 00 ff ff ff 00
0001240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001260 ff ff ff 00 ff 00 ff 00 ff ff ff 00 ff ff ff 00
0001300 ff 00 00 00 ff 00 ff 00 ff 00 ff 00 00 00 ff 00
0001320 ff ff ff 00 ff ff ff 00 ff ff ff 00 00 00 ff 00
0001340 00 00 ff 00 00 00 ff 00 00 00 ff 00 00 00 ff 00
0001360 ff ff ff 00 00 00 ff 00 ff ff ff 00 00 00 ff 00
0001400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0001420 ff ff ff 00 ff ff 00 00 00 00 00 00 00 00 00 00
0001440 ff 00 00 00 00 ff 00 00 00 00 00 00 00 00 00 00
0001460 ff ff ff 00 00 ff 00 00 00 00 00 00 00 00 00 00
0001500 ff 00 ff 00 00 ff ff 00 00 00 00 00 00 00 00 00
0001520 ff 00 ff 00 00 ff 00 00 00 00 00 00 00 00 00 00
0001540 ff ff ff 00 00 ff 00 00 00 00 00 00 00 00 00 00
0001560 00 00 00 00 ff ff 00 63 da e9 3c 36 b1 aa 93 59
```

然后就是我没想到的地方：把00字节去掉，得到图案即是flag

*圣火昭昭

这个找新佛语解密网站找了好久，以前都是佛语。。。。
下载图片查看属性，得到`新佛曰：`，以前要么没有这仨字，要么是`佛曰：`。



所以找到新佛语解密网站，解密得到(hint说key去掉com，key应该就是gemlove)

试了下不是flag，应该是某种隐写的密码。然后没思路了，于是看看题目描述和群里管理员“猜”

发现“猜“这个字很魔性。然后我没见过类似的隐写，然后就断这里了。。
看了一眼官方wp发现是`outguess 隐写`，没听说过。查了下，发现直接用一个工具即可。
outguess的安装：

```
git clone https://github.com/crorvick/outguess 进行下载
下载完成后进入outguess文件夹，右击打开终端，执行命令：
./configure && make && make install 进行编译及安装。
```

outguess的常见使用：

```
对图片信息进行破解：
outguess -r /root/qwzf.jpg hidden.txt
带key的：
加密：outguess -k "abcd" -d hidden.txt demo.jpg out.jpg
解密：outguess -k "abcd" -r out.jpg hidden.txt
```

好了，我直接执行

执行成功，打开qwzf.txt得到flag

*TARGZ-y1ng

`.tar.gz`文件直接用winrar解压发现需要输入密码，并不是伪加密，也爆破不出来。于是看看题目描述

`tar zxvf`，即`tar -zxvf 压缩文件名`。还好linux学过，是`.gz`格式的解压缩并解打包。
试下linux解压缩并并解打包命令，然后发现依旧需要密码。密码在哪呢？
试试文件名。果然是解压密码。一直解一直解。好吧，我自闭了，不做了。
看官方wp写个脚本挺省事。但官方脚本有时候会报错(有关系统操作的脚本我还没学过怎么写，抽时间补下相关知识，然后再改下)。
`decompress.py`

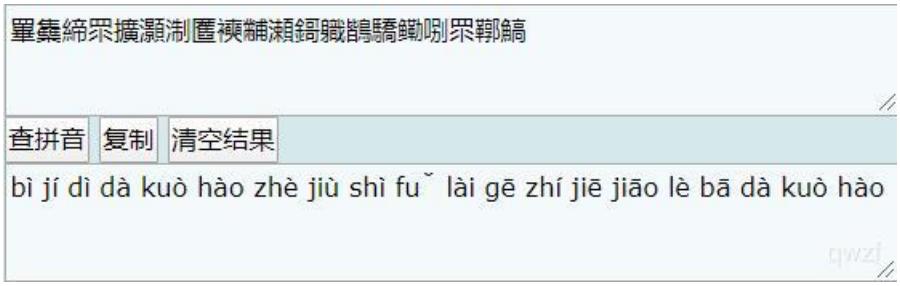
```
import os
import filetype
import time

while 1:
 aa = os.popen('ls')
 filename = aa.read().replace('decompress.py','').replace('\n', '')
 a = filename.replace('.tar.gz', '')
 kind = filetype.guess(filename)
 if kind.extension is 'zip':
  os.system("mv {} {}.zip|unzip -P {} {}.zip".format(filename, a, a, a))
  os.system("rm *.zip")
  time.sleep(0.1)
 else:
  print('解压完成')
  break
```

执行脚本，得到flag文件。

0x02 Crypto

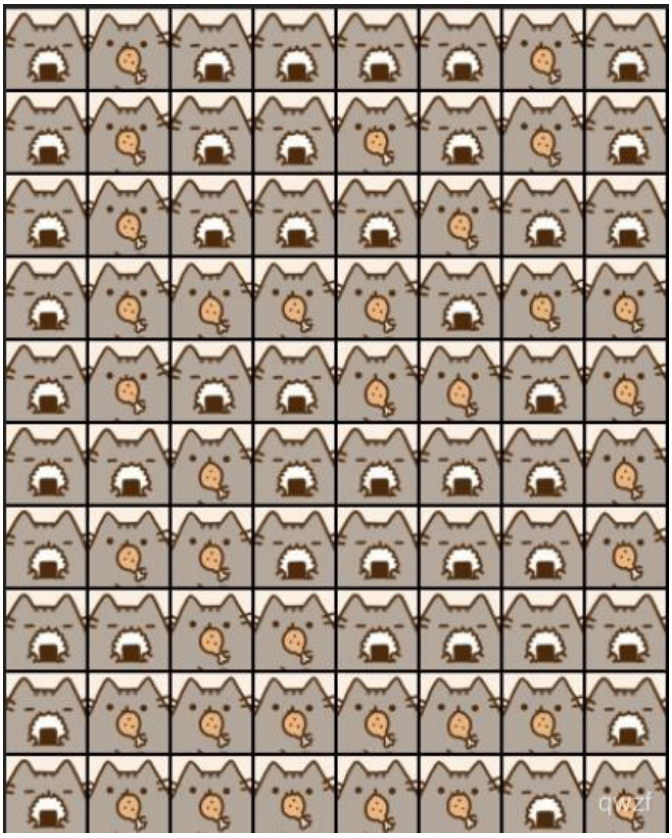
*老文盲了

主要就是生僻汉字拼音，找个在线汉字转拼音即可



然后发现拼音再转换成汉字是`BJD{这就是flag直接交了吧}`，我交，不对。。。问了下群管理员，让我考虑下`"这"还可能是什么？`emmmm，试了好多不对。。
看官方wp说花括号`{}`里的汉字再与之前的生僻字对应，也就是`BJD{淛匱襫黼瀨錯膱鶛驕鰳唰}`
确定这是Crypto，感受到一丝MISC的味道。。

cat_flag



这个呢，就比较简单了。每行8张小图片，有吃的、无吃的。很明显可能就是二进制嘛，8bit一个字节，刚刚好。手工转换下，得到

```
0100001001001010010001000111101101001101001000010110000100110000011111100111110 1
```

然后二进制转ASCII，得到flag。还是感觉这道题也有MISC的感觉。

灵能精通

说实话，没见过，感觉是猪圈变种，但是搜不到。经别人提醒，知道是圣堂武士密码

对照着解密就ok了。。emmmmm，突然发现上边这解密图上的水印似乎是我的某位学长？？！！！
于是好奇的搜了下，发现一篇总结很全面的好文章：
CTF中Crypty（密码类）入门必看

燕言燕语

题目：
79616E7A69205A4A517B78696C7A765F6971737375686F635F73757A6A677D20
hex->ascii

```
yanzi ZJQ{xilzv_iqssuhoc_suzjg}
```

yanzi应该是密钥，后边的是密文。于是开始尝试，由于最近刚看过维吉尼亚密码，所以尝试。果然是维吉尼亚密码加密，解密即可。
维吉尼亚密码加解密原理及其实现

*Y1nglish-y1ng

```
Nkbaslk ds sef aslckdqdqst. Sef aslckdqdqst qo lzqtbw usf ufkoplkt zth oscpslsfko. Dpkfk zfk uqjk dwcko su dscqa
o qt dpqo aslckdqdqst, kzap su npqap qo jkfw mzoqa. Qu wse zfk qtdkfkodkh qt tkdnsfw okaefqdw, nkbaslk ds czfdqa
qczdk. Bkd lk dkbb wse z odsfw.
Q nzo pzjqtv hqttkf zd z fkodzefztd npkt Pzffw Odkkbk azlk qt, pk qo z Izcztkok ufsl Izczt med tsn pk qo tsd bqj
qtv qt Izczt, lzwmk Pzffw qot'd z Izcztkok tzlk med pk qo fkzbbw z Izcztkok. Pzffw nsfwkh qt z bznwkf'o suuqak w
kzfo zvs, med pk qo tsn nsfwqtv zd z mztw. Pk vkdo z vssh ozbzfw, med pk zbnzwo msffsno lstkw ufsl pqo ufqktho z
th tkjkf czwo qd mzaw. Pzffw ozn lk zth azlk zthozdzd dpk ozlk dzmbk. Pk pzo tkjkf msffsnkh lstkw ufsl lk. Npqbk
 pk nzo kzdqtv, Q zowkh pql ds bkth lk &2. Ds lw oefcfqok, pk vzjk lk dpk lstkw qllkhqzdkbw. 'Q pzjk tkjkf msfff
snkh ztw lstkw ufsl wse,' Pzffw ozqh,'os tsn wse azt czw usf lw hqttkf!' Tsn q nqbb vqjk wse npzd wse nztd.
MIH{cwdp0t_Mfed3_u0fa3_sF_geqcgeqc_ZQ_Af4aw}
```

最近大致看了单表替换密码，然后尝试一下，发现不对，emmmm。。。
题目说是英语改过来的，然后就断了。现在看了眼wp：
直接找个在线的 cryptogram solver 即可解密，比如quipqiup

Solve

```
0    -1.433   Welcome to our competition. Our competition is mainly for freshmen and
              sophomores. There are five types of topics in this competition, each of which
              is very basic. If you are interested in networy security, welcome to
              participate. Let me tell you a story. I was having dinner at a restaurant when
              Harry Steele came in, he is a Japanese from Japan but now he is not living in
              Japan, maybe Harry isn't a Japanese name but he is really a Japanese. Harry
              woryed in a lawyer's office years ago, but he is now worying at a bany. He
              gets a good salary, but he always borrows money from his friends and never
              pays it bacy. Harry saw me and came andsatat the same table. He has never
              borrowed money from me. While he was eating, I asyed him to lend me &2. To my
              surprise, he gave me the money immediately. 'I have never borrrowed any money
              from you,' Harry said,'so now you can pay for my dinner!' Now i will give you
              what you want. BJD{pyth0n_Brut3_f0rc3_oR_quipquip_AI_Cr4cy}
```

> 最后一单词是错误的，hint 也告知有个地方需要自己修正。 可以看上面那段话也可以发现 worying at a bany，应该是 working at a bank，还有 networy，很明显应该是 network，y 要改成 k；直接读也发现 cracy 这个单词不对劲，应该和暴力破解是同类型的词，所以改成 Cr4ck

rsa0

这个和下一个RSA题确实不难，比较基础。然而我自己写的脚本没一点问题，但就是跑不出来flag。。。

```
e=14136631
p+q=203940443535403193446987115970007630572414362611423320487832030859361719636376906703013119976098449664178987
132770462554783430389881233594776560584568340000
p-q=552244924992059024168200759326277068616872560260080430800699105330461139310933849775066186314947560213554227
471329071554174881308635889212960014266500803 4
c=575641602744042192641774594504480191832756932007684160514366407555222029636532921723918409125368315711529133 28
785130833906878324669454383965866712212130750436153700289781965212722910455077455251018758850662752081666673050 5
098504784701732829570569887798514018986048242476312812014973216828996721726586973663 50
flag=??????
```

今天又试了下发现把除号`/`改为`//`就行了。。。吐血了。。。
我写的脚本如下：

```
import gmpy2
#import binascii
from Crypto.Util import number

x = 203940443535403193446987115970007630572414362611423320487832030859361719636376906703013119976098449664178987
132770462554783430389881233594776560584568340000
y = 552244924992059024168200759326277068616872560260080430800699105330461139310933849775066186314947560213554227
471329071554174881308635889212960014266500803 4
c = 575641602744042192641774594504480191832756932007684160514366407555222029636532921723918409125368315711529133
287851308339068783246694543839658667122121307504361537002897819652127229104550774552510187588506627520816666730 5
050985047847017328295705698877985140189860482424763128120149732168289967217265869736635 0
p=(x+y)//2
q=(x-y)//2
e=14136631

n=p*q
phi_n = (p-1)*(q-1)
d = gmpy2.invert(e,phi_n)
m = pow(c, d, n)
print(m)
print(number.long_to_bytes(m))
#print(binascii.a2b_hex(hex(m)[2:]).decode("utf8"))
```

```
============= RESTART: D:\网安\做题\BUUCTF\BJD2\Crypto\rsa0\rsa0.py ===
=
143376365551180833086311036443467716183954502863274023836211450953972042
047737698820653153615114
b'flag{fba47903-09e6-4436-9e5a-9ba77284ce71}\n'
>>>
```

rsa1

已知`p^2+q^2`和`p-q`，联立方程组可解出p,q
测试发现每访问一次，e和c都会变，但是`p^2+q^2`和`p-q`不变，于是考虑共模攻击：

```
import binascii
from gmpy2 import invert

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

c1 = 513406459866806488674134444342283048519927392777825474543426977418658773929005112742021061348981128768257723192295570024250957596742322011852187575246109336374517826040591789758360619509417131123303340860564021314144102554577972058083395963449916206844661959207863504181866914142217531999306584932060622160 77
e1 = 13718357

c2 = 346738734184926791893158418982573027221111714665843267932777180579773998311772946545200648667891289632441039114806917781433956407647935567589480863265180188158683203249381155205103693228511307775776370157108924353339611869060327537708929386498423217210534396884277690818506372603237307774087185642953951469 91
e2 = 13103833

A = 120799346838375010895229335442462202865240248511497399986136417952902545991062894320724425829562051895373363854631133655790120865759504895601823054884863801257685969042479894585894024020955762745984438766369627115385495533880991887182178078779931382196022613790869270179113382812710158113446233023928865487250
B = -173310740922423028953019224890542442439053023706195585309017244357388125399188801058695125058319631820311398914638875212914669298040078201193928522639873 6

n = (A-B**2)//2

s = egcd(e1, e2)
s1 = s[1]
s2 = s[2]
if s1<0:
    s1 = - s1
    c1 = invert(c1, n)
elif s2<0:
    s2 = - s2
    c2 = invert(c2, n)

m = pow(c1,s1,n)*pow(c2,s2,n) % n
print(binascii.a2b_hex(hex(m)[2:]).decode("utf8"))
```

```
'''
============== RESTART: D:\网安\做题\BUUCTF\BJD2\Crypto\rsa0\:
=
flag{e0e1f1ef-80dc-4641-b739-5a8483a22ea7}

>>>
```

0x03 后记

随便写写就那么多字了。。。算了，Web复现再另起一篇进行记录。做题和复现BJD MISC和Crypto，我又收获一波新知识：`新佛语解密`、`outguess隐写`、`多次输入密码解压压缩包脚本`、`圣堂武士密码`和`在线的cryptogram solver(quipqiup)`，似乎就收获那么多。。。