第三届江西省高校网络安全技能大赛 部分wp&Crypto的疑惑

原创

monster663 于 2020-09-01 18:21:19 发布 1629 收藏 9

本文链接：https://blog.csdn.net/monster663/article/details/108347291

2020 第三届江西省高校网络安全技能大赛

目录

本来实在是不想写这个比赛的wp，最后因为我们队伍解题分配以及战术的问题无缘决赛，赛后我们（也就两个人）也进行了积极的总结和反思，复盘以后发现其实还有好几个题目都能写，最后还是很遗憾的错失进入决赛的机会，准备明年再来吧。

Misc1-Hello

直接把题目拿去base64，得flag

Base64: Q01JU0NDVEZ7V2VsY29tZV9DVEZlciF9

Flag: CMISCCTF{Welcome_CTFer!}

Misc2-encrypt

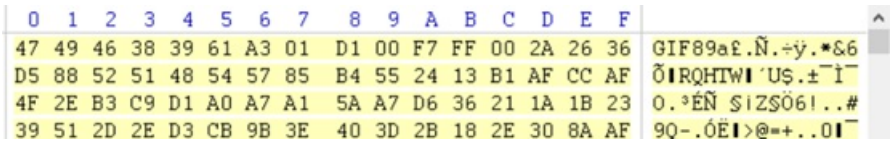
本来是想考zip伪加密，通过winhex修改加密标志位解除伪加密，奈何我的360压缩无视了它的伪加密，直接打开了压缩包，把题目拿去解base64两次，得flag

Base64两次: UTAxSlUwTkRWRVo3Um1GclpWOWxibU55ZVhCMGFXOXVmUT09

Flag: CMISCCTF{Fake_encryption}

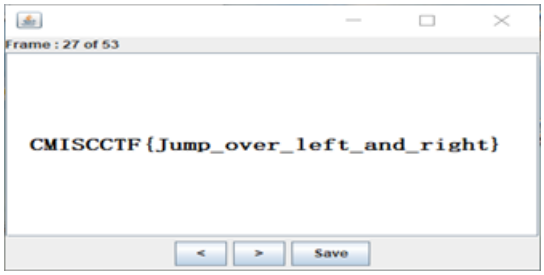
Misc3-jump

题目图片叫jump.png,通过winhex发现gif文件头

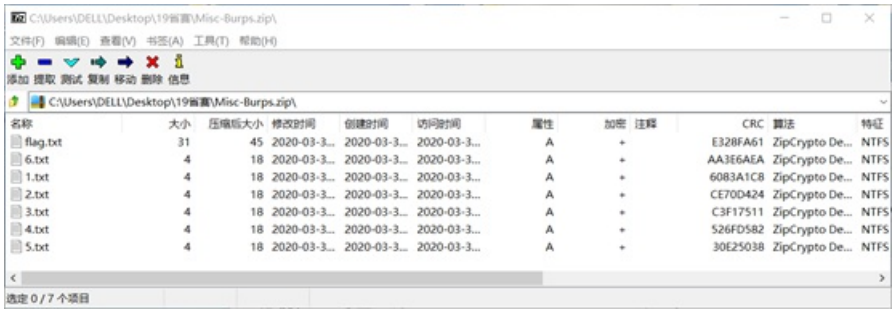


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
	47	49	46	38	39	61	A3	01	D1	00	F7	FF	00	2A	26	36	GIF89a£.Ñ.÷ÿ.*&6
	D5	88	52	51	48	54	57	85	B4	55	24	13	B1	AF	CC	AF	Õ‡RQHTW…´U$.±¯Ì¯
	4F	2E	B3	C9	D1	A0	A7	A1	5A	A7	D6	36	21	1A	1B	23	O.³ÉÑ §¡Z§Ö6!..#
	39	51	2D	2E	D3	CB	9B	3E	40	3D	2B	18	2E	30	8A	AF	9Q-.ÓË›>@=+..0Š¯

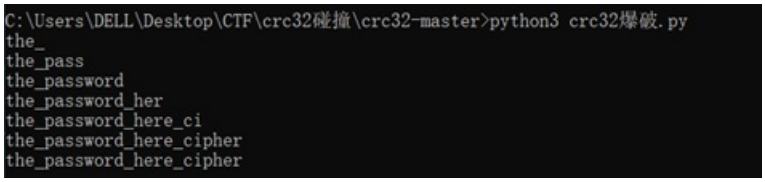
直接修改后缀为gif,通过Stegsolve按帧查看，最后得到flag



Misc4-Brups



通过四字节得crc爆破



另附爆破脚本：

```
import binascii
import string
import os,sys
dic=string.printable
crc5=[0x6083A1C8,0xCE70D424,0xC3F17511,0x526FD582,0x30E25038,0xAA3E6AEA]
password=''
for crc in crc5:
 for i in dic:
  for j in dic:
   for k in dic:
    for l in dic:
     s=i+j+k+l
     if crc==((binascii.crc32(s.encode()))&0xffffffff):
        password+=s
       print(password)
print(password)
```

得到password：the_password_here_cipher

拿去解压打开flag.txt即可得flag:

CMISCCTF{how_to_burp_by_coding}

Misc5-Trees

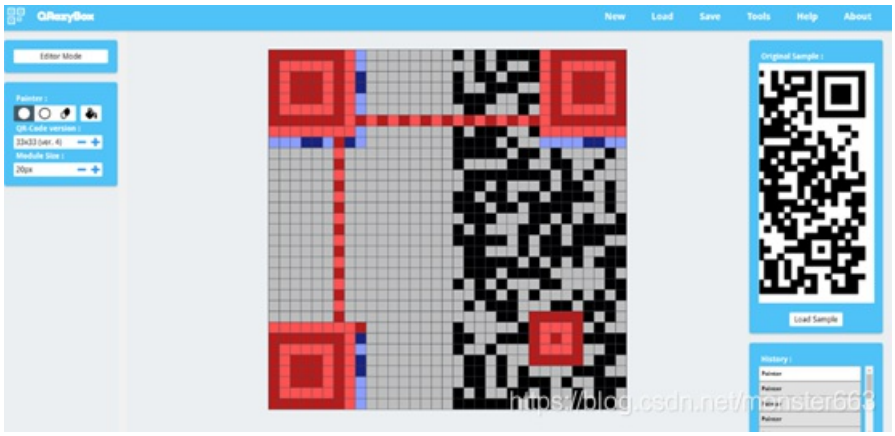
用Stegsolve查看颜色通道时发现部分文字，CMISCCTF{co******t_tree}
上网查单词，得到coconut(椰树)(又暴露英语水平了)，猜测flag为CMISCCTF{coconut_tree}，提交，正确。

赛后群里大佬给出了正解

```
from PIL import Image

img = Image.open('enc.png')
w = img.width
h = img.height
img_obj = Image.new("RGB",(w//16,h//16))

for x in range(w//16):
    for y in range(h//16):
        (r,g,b)=img.getpixel((x*16,y*16))
        img_obj.putpixel((x,y),(r,g,b))

img_obj.save('ok.png')
```

Misc6-qrcode

直接上工具补全二维码



Flag: CMISCCTF{qr_c0de_r3c0very}

Misc7-blind

考查图片盲水印隐写以及PNG文件修复

扔进kali中去binwalk,发现两张图片，foremost分离

使用decode.py版盲水印脚本

查看得到压缩密码Q@CTF@NX，解压图片：

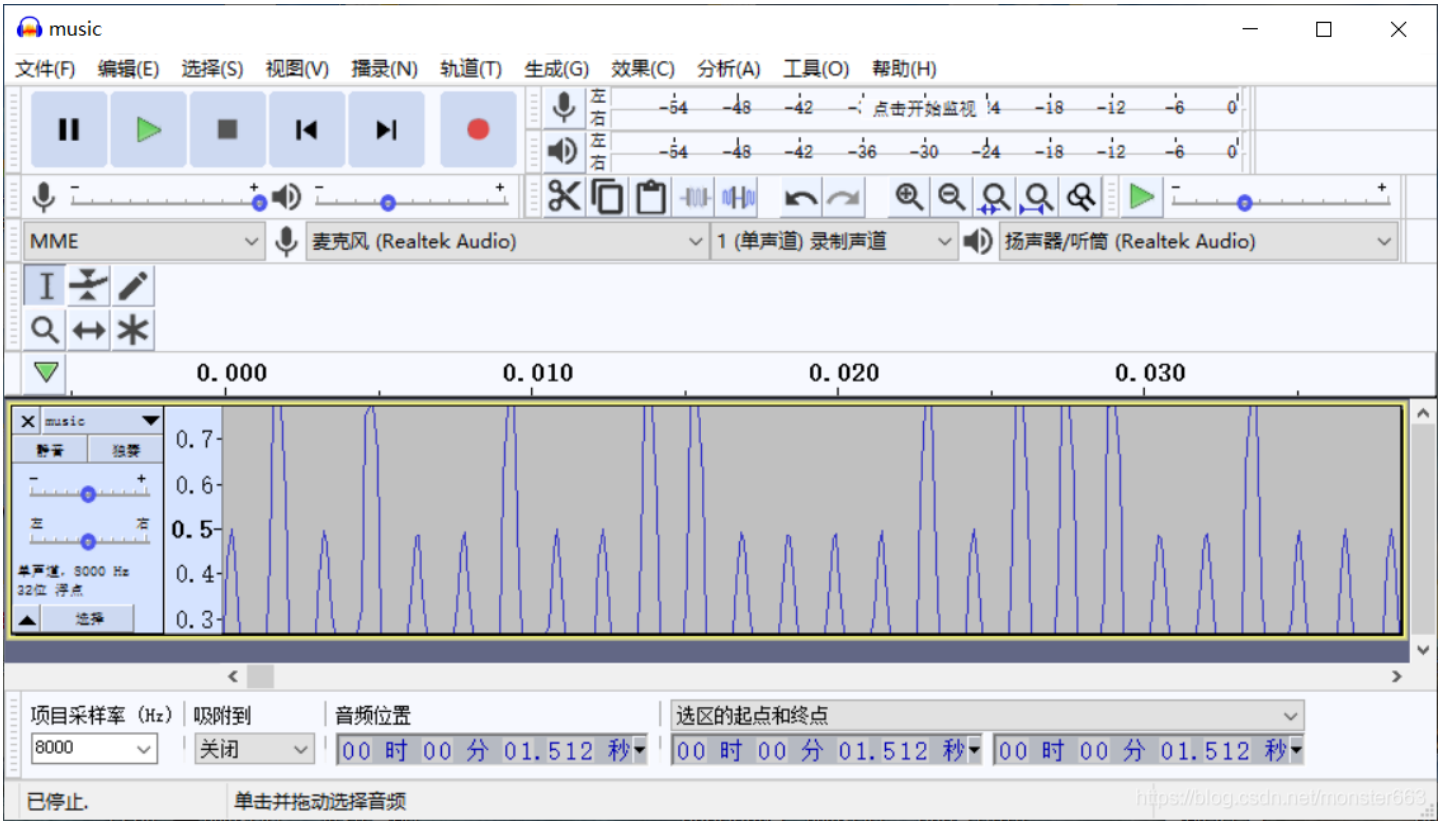
Winhex查看并修复文件头，

补齐文件头，得到flag

Misc8-music

这题是最伤心的，也是我们这次比赛最大的败笔

用audactity分析，发现了音频中有很多高低频



猜测为二进制，用高频表示1，低频表示0，8位位一组，可以发现前面的数据转出来是Rar!，奈何不清楚如何提取其中的二进制数据，手撕三个半小时未果，赛后复现看到大佬用python脚本提出了压缩包数据，后面是NTFS文件流隐写以及PNG图片改高得二维码，这里不再详细叙述了，需要请大佬队伍的wp

web,pwn和reverse可以参考大佬队伍的wp写的也很详细

引用的wp如下：

```
import numpy as np
import struct
import wave
import re


def write_records(records, format, f):
    #Write a sequence of tuples to a binary file of structures.
    record_struct = Struct(format)
    for r in records:
        f.write(record_struct.pack(*r))

path = "./music.wav"
f = wave.open(path, "rb")
# 读取格式信息
# (nchannels, sampwidth, framerate, nframes, comptype, compname)
params = f.getparams()
nchannels, sampwidth, framerate, nframes = params[:4]
# 读取波形数据
str_data = f.readframes(nframes)
f.close()
#将波形数据转换为数组
wave_data = np.fromstring(str_data, dtype=np.short)
b = ''
# arr = [elem for elem in wave_data if elem >0]
max = 0
d = ''
for i in wave_data:
    if i <0:
        if max !=0:
            if max<25000:
                d +='0'
            else:
                d += '1'
                pass
        max = 0
    if max < i:
        max = i

print(d)
print("\n\n\n\n")
a = re.findall(r'.{8}',d)
hex_list=[]
for i in a:
    res = hex(int(i,2))
    hex_list.append(res)
print(hex_list)

with open("result.txt","wb") as f:
    for x in hex_list:
        s = struct.pack('B',int(x,16))
        f.write(s)
```

Crypto的疑惑

这里只想再提一下密码学，同时也以虚心学习的态度请教大佬
Crypto-Factor题目
n = 3454083680130687060405946528826790951695785465926614724373
e = 3
c = 1347530713288996422676156069761604101177635382955634367208
gcd(m, n) = 1
n可以由yafu或在线分解得

p=11761833764528579549
q=17100682436035561357
r=17172929050033177661
使用多因子RSA脚本以及小指数爆破攻击均未果，没有思路了，求教

Crypto-Change题目

```
from flag import FLAG
from Crypto.Util.number import *
import gmpy2
import random
while True:
    p = int(gmpy2.next_prime(random.randint(10**399, 10**400-1)))
    q = int(str(p)[200:]+str(p)[:200])
    if gmpy2.is_prime(q):
        break
m = bytes_to_long(FLAG)
n = p*q
e = 65537
c = pow(m,e,n)
with open("enc","wb") as f:
    f.write(str(c))
    f.write("\n")
    f.write(str(n))
```

赛后发现是2019高校运维赛的题目使用该脚本的时候发现，那个题目的n恰好为800位，所以在计算的时候可以直接取n的前两百位和后两百位，这个题目的话由于q,p可能是400位出头的质数，相乘以后的n为798位，在提取n的位数的时候就有些犹豫，目前还没有复现解出这个题目，求大佬帮我答疑解惑。
我又回来了，赛后找大佬积极复现了题目，发现这个RSA真是有意思(头秃，再也不写RSA了)，得到的解释是这样的，由于p是生成的四百位质数，q为p的前后两百位交换，所以有可能p的第201位刚好为0，导致q为399位的质数，最后导致n为798位（太强了吧），大佬的wp如下:

```
c='182812482972168423884795132699225934365072979206288632257180603530046820174392675977209758378734399146216742345585898385168866887000708558119959898992294085847474548306743585711154035585848291290988967352517174312220756638881837930962458861193652684492265539096477345065113556380573776423787885892688197584678128636231428194711357642971544417113415626331810909274966752557628893585198569815939514862013512237657828262360291726912615575646318630641527418369988268899879152029186728850816178597399494254385226049249357897840618728804680238123954207656671747782543031545429711152272581734051959578453680011676521727918037340906791388178004979453256050227967701258768070039292546964652071924183467364467145178290753361477912582242961929982420950384199259355122986865808523351306098081481072454093823090'
n=43898039703131539222945390804850954083224604163143287850957966566418274746310023016082386562179805316498932508608500394018173172108970138074369876144381252302414481720590238090306205413873065845128690434753621083316092491734763314898305201555035491315431216290155587049427390371434986974679386187425720108577789396171546895066164177851211032545737144620337976745886205919394643468332457853016365054163726115803704120564242880294229501156227708468702521362669884952624066375407350810222906647577389363871684517646907093880329851515514024097083638778540108591936974152089027190233295166995341137363368894416247099485665460487228710374692204184406505327405999059549615986620655111936103623743128983098517438452242336481199724125500551424819844792539637819291555389899375866004122339316870738058001243 7
s=str(n)[:198]+str(n)[-200:]
print(len(str(n)))
xy=int(s)
t=n-xy*(10**400)-xy
print(t)
xy2=(t)//(10**200)
import gmpy2
x_y=gmpy2.iroot(xy2-2*xy,2)[0]
x__y=gmpy2.iroot(xy2+2*xy,2)[0]
x=(x__y+x_y)//2
y=x__y-x
p=x*10**200+y
q=n//p
e=65537
d=gmpy2.invert(e,(p-1)*(q-1))
from Crypto.Util import number
print(number.long_to_bytes(gmpy2.powmod(int(c),d,n)))
```

运行即可得到flag