week1

web

谁吃了我的flag :50

Description

呜呜呜，Mki一起床发现写好的题目变成这样了，是因为昨天没有好好关机吗T_T **hint: 据当事人回忆，那个夜晚他正在用vim编写题目页面，似乎没有保存就关机睡觉去了,现在就是后悔，十分的后悔。**

URL http://118.25.111.31:10086/index.html

Base Score 50

Now Score 50

User solved 250

错误姿势：

（一开始没有hint）刚拿到题，直接bp看请求头和回应，发现

```
GET /index.html HTTP/1.1
Host: 118.25.111.31:10086
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
```

```
HTTP/1.1 200 OK
Server: nginx/1.15.8
Date: Fri, 01 Feb 2019 11:43:46 GMT
Content-Type: text/html
Content-Length: 319
Last-Modified: Fri, 25 Jan 2019 08:10:00 GMT
Connection: close
ETag: "5c4ac458-13f"
Accept-Ranges: bytes

<!DOCTYPE HTML>
<html>
    <head>
        <title>who eat my flag???</title>
    </head>
    <body>
        <p>damn...hgame2019 is coming soon, but the stupid Mki haven't finished his web-challenge...</p>
        </br>
        <p>fine, nothing serious, just give you flag this time...</p>
        </br>
        <p>the flag is hgame{3eek_diScl0Sure
    </body>
</html>
```

F5刷新后，头变成了：

```
GET /index.html HTTP/1.1
Host: 118.25.111.31:10086
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: BL_D_PROV=undefined; BL_T_PROV=undefined
Upgrade-Insecure-Requests: 1
X-Forwarded-For: 127.0.0.1
If-Modified-Since: Fri, 25 Jan 2019 08:10:00 GMT
If-None-Match: "5c4ac458-13f"
Cache-Control: max-age=0
```

然后花了大把时间去研究 If-Modified-Since，If-None-Match，Last-Modified，ETag这些东西的含义和神奇作用，然后就没有然后了。。。。

正确解法：

看到hint瞬间想到vim未保存就退出造成的问题，

具体见：参考博客

结论：payload(http://118.25.111.31:10086/.index.html.swp)

打开下载的文件即可看到：

b0VIM 8.0tPE\\\(mki603arrival~mki603/mki/hgame/week1/web1/index.htmlutf-8 U3210#"! Utp

the flag is hgame{3eek_diScl0Sure_fRom+wEbsit@}

fine, nothing serious, just give you flag this time...

damn...hgame2019 is coming soon, but the stupid Mki haven't finished his web-challenge...

flag：**hgame{3eek_diScl0Sure_fRom+wEbsit@}**

换头大战 :100

Description

想要flag嘛 工具: burpsuite postman hackbar 怎么用去百度，相信你可以的

URL http://120.78.184.111:8080/week1/how/index.php

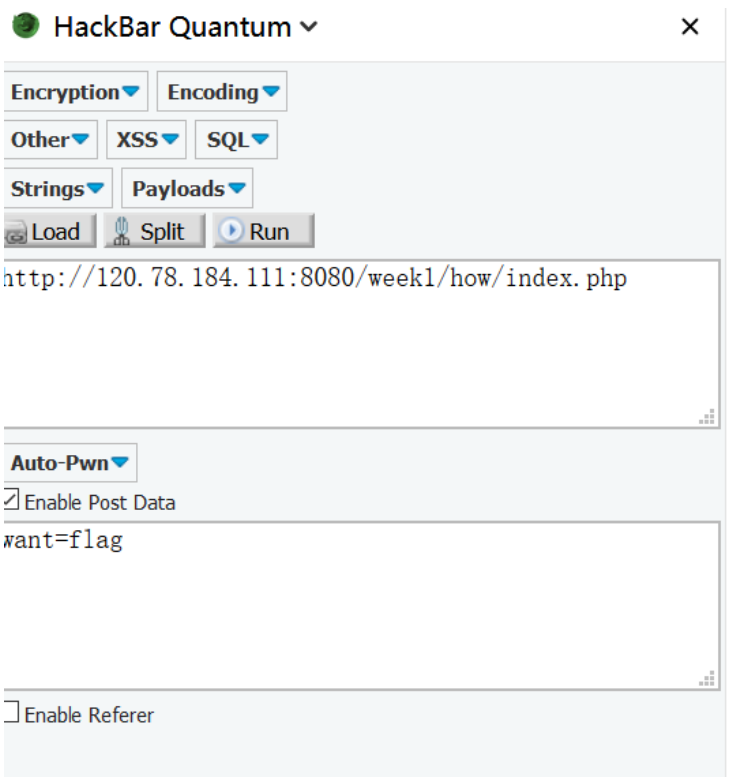
Base Score 100

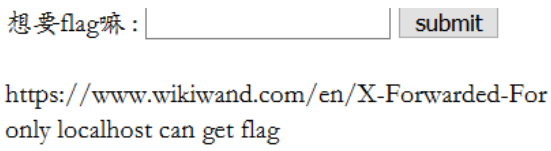
Now Score 100

User solved 267

解题姿势：

先输入“flag”提交，发现：





URL里面多了“?want=flag”,说明现在是get请求，明显提示要求我们改成post请求，我们可以用hint给的那些工具，我就用HackBar Quantum示范一下，这个工具是火狐里面的免费插件，直接去扩展插件里面找就好了。然后就是该X-Forwarded-For: 127.0.0.1，然后就是referer...

可以自己操作一下看看。

very easy web :100

Description

代码审计初♂体验

URL http://120.78.184.111:8080/week1/very_ez/index.php

Base Score 100

Now Score 100

User solved 286

题目

```
 <?php
error_reporting(0);
include("flag.php");

if(strpos("vidar",$_GET['id'])!==FALSE)
  die("<p>干巴爹</p>");

$_GET['id'] = urldecode($_GET['id']);
if($_GET['id'] === "vidar")
{
  echo $flag;
```

```
}
highlight_file(__FILE__);
?>
```

二次URL加密即可（浏览器会自动解密一次）

can u find me :100

Description

为什么不问问神奇的十二姑娘和她的小伙伴呢 学习资料: https://www.cnblogs.com/yaoyaojing/p/9530728.html https://www.cnblogs.com/logsharing/p/8448446.html https://blog.csdn.net/z929118967/article/details/50384529

URL http://47.107.252.171:8080/

Base Score 100

Now Score 100

User solved 255

F12看源码，发现f12.php跳转，用bp,仔细看回复（你会发现很神奇的东西）。

题目提示很明显，自己操作试试。

re

brainfxxker :100

Description

Ouch! What is this? I don't think that I am pretty good at C++, what a brain fxxker it is! 学习资料: https://zh.wikipedia.org/wiki/Brainfuck https://zh.wikipedia.org/zh/ASCII 读懂我的代码逻辑答案就出来了 补充说明: 判定答案是否正确的是 Notice 2，即“不执行 [+.] 这个部分”，不要单纯看有没有输出 orz

URL http://plir4axuz.bkt.clouddn.com/hgame2019/brainfucker.cpp

Base Score 100

Now Score 100

User solved 111

hint很详细，我就不赘述了，直接上脚本

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

// Orz... I haven't learnt C++ before.
// It seems like my brain was fxxked by these codes...
```

```
// Notice:
// 1. the answer is your input when nothing strange was printed
// 2. that is, wrong inputs will encounter with the part "[+.]"
// 3. [!!!] REMEMBER TO WRAP YOUR ANSWER WITH "hgame{" AND "}"
//    [!!!] BEFORE YOU SUBMITTED IT

// oyiadin, Jan 18, 2019
// enjoy it! ;)

namespace bf {

class Parser {
 public:
  Parser() = default;
  ~Parser() = default;
  void execute(const std::string &buf);

 protected:
  uint8_t data[100] = {0};
  int ptr = 0;
};


void Parser::execute(const std::string &buf) {
  for (auto i = buf.cbegin(); i != buf.cend(); ++i) {
    switch (*i) {
      case '>':
        ++ptr;
        break;
      case '<':
        --ptr;
        break;
      case '+':
        ++data[ptr];
        break;
      case '-':
        --data[ptr];
        break;
      case '.':
        putchar((256-data[ptr]));
        ptr=0;
        memset(data, 0, 100);
        break;
      case ',':
        while ((data[ptr] = getchar()) == '\n') ;
        break;
      case '[':
        if (!data[ptr]) {
          while (*i++ != ']') continue;
          --i;
        }

        break;
```

```
      case ']':
        if (data[ptr]) {
          while (*(i-1) != '[') --i;
          --i;
        }
        break;
      default:
        break;
    }
  }
}

}


int main() {
  bf::Parser parser;
  parser.execute(">++++++++++[<---------->-]<++.");
  parser.execute(">+++++++++[<---------->-]<-.");
  parser.execute(">+++++++[<------->-]<---.");
  parser.execute(">++++++[<------>-]<+++.");
  parser.execute(">++++++++[<---------->-]<++.");
  parser.execute(">++++++++++[<---------->-]<--.");
  parser.execute(">++++++++++[<--------->-]<-----.");
  parser.execute(">++++++++++[<---------->-]<+.");
  parser.execute(">+++++++++[<--------->-]<---.");
}
```

flag:

```
hgame{bR4!NfUcK}
```

HelloRe :100

Description

Welcooooooome!

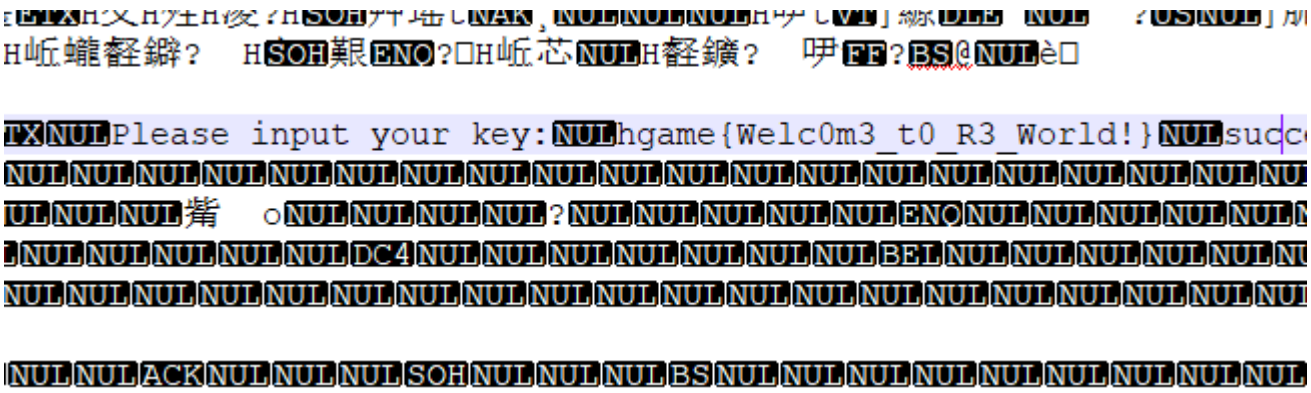
URL http://plps4kyke.bkt.clouddn.com/HelloRe

Base Score 50

Now Score 50

User solved 205

这个题没什么意思，flag直接以字符串在里面，直接notepad++就可以看到



flag:

```
hgame{Welc0m3_t0_R3_World!}
```

r & xor :100

Description

论r 与 xor 的重要性 ida里奇怪的大数字?不如按r试一试

URL http://plir4axuz.bkt.clouddn.com/hgame2019/xor

Base Score 100

Now Score 100

User solved 93

直接丢ida，F5看代码，代码逻辑很简单，就是异或，还有题目给的提示，可以提取其中一个异或字符串，另一个要看它们的地址，一个一个提取出来，主要看懂代码逻辑，然后仔细将另一个字符串数据提取出来即可，具体看脚本。

脚本:

```
#include<iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
int main()
{
    int result; // eax@2
    signed int i; // [sp+8h] [bp-138h]@3
    int v6[6]; // [sp+10h] [bp-130h]@1
    int v7; // [sp+28h] [bp-118h]@1
    int v8; // [sp+30h] [bp-110h]@1
    int v9; // [sp+38h] [bp-108h]@1
    int v10; // [sp+3Ch] [bp-104h]@1
    int v11; // [sp+40h] [bp-100h]@1
    int v12; // [sp+44h] [bp-FCh]@1

    int v13; // [sp+48h] [bp-F8h]@1
```

```
    int v14; // [sp+4Ch] [bp-F4h]@1
    int v15; // [sp+50h] [bp-F0h]@1
    int v16; // [sp+54h] [bp-ECh]@1
    int v17; // [sp+5Ch] [bp-E4h]@1
    int v18; // [sp+60h] [bp-E0h]@1
    int v19; // [sp+64h] [bp-DCh]@1
    int v20; // [sp+68h] [bp-D8h]@1
    int v21; // [sp+6Ch] [bp-D4h]@1
    int v22; // [sp+70h] [bp-D0h]@1
    int v23; // [sp+74h] [bp-CCh]@1
    int v24; // [sp+78h] [bp-C8h]@1
    int v25; // [sp+80h] [bp-C0h]@1
    int v26; // [sp+84h] [bp-BCh]@1
    int v27; // [sp+88h] [bp-B8h]@1
    int v28; // [sp+8Ch] [bp-B4h]@1
    int v29; // [sp+90h] [bp-B0h]@1
    int v30; // [sp+94h] [bp-ACh]@1
    __int64 v31; // [sp+A0h] [bp-A0h]@1
    __int64 v32; // [sp+A8h] [bp-98h]@1
    __int64 v33; // [sp+B0h] [bp-90h]@1
    __int64 v34; // [sp+B8h] [bp-88h]@1
    int v35; // [sp+C0h] [bp-80h]@1
    char s[104]; // [sp+D0h] [bp-70h]@1
    __int64 v37; // [sp+138h] [bp-8h]@1
    v31 = 348395146230480 2664LL;
    v32 = 685993493088052 0053LL;
    v33 = 356022345849145 8926LL;
    v34 = 238722599700715 0963LL;
    v35 = 8200481;
    string flag="hgame{Y0u_mayb3_need_th1s_0ne!!!!!}";
    memset(v6, 0, 0x90uLL);
    int v[]=
{0,0,0,0,0,0,1,0,7,0,92,18,38,11,93,43,11,23,0,23,43,69,6,86,44,54,67,0,66,85,126,72,85,30,0};

    /*for ( i = 0; i < 35; ++i )
    {
        if(i<8){
            s[i] = v[i] ^ *((char*)&v31 + i) ;
            cout<<"s["<<i<<"]:"<<s[i]<<endl;
        }
        else if(i<16){
            s[i] = v[i] ^ *((char*)&v32 + i-8) ;
            cout<<"s["<<i<<"]:"<<s[i]<<endl;
        }
        else if(i<24){
            s[i] = v[i] ^ *((char*)&v33 + i-16) ;
            cout<<"s["<<i<<"]:"<<s[i]<<endl;
        }
        else if(i<32){
            s[i] = v[i] ^ *((char*)&v34 + i-24) ;
            cout<<"s["<<i<<"]:"<<s[i]<<endl;
        }

        else{
```

```
            s[i] = v[i] ^ *((char*)&v35 + i-32) ;
            cout<<"s["<<i<<"]:"<<s[i]<<endl;
        }
    }
    puts("You are right! Congratulations!!");
    for ( i = 0; i < 35; ++i )
    {
        cout<<s[i];
    }
    cout<<endl;*/
    for ( i = 0; i < 35; ++i )
    {
        s[i] = v[i] ^ flag[i];
        cout<<s[i];
    }
    return 0;
}
```

flag:

```
hgame{X0r_1s_interest1ng_isn't_it?}
```

Pro的Python教室（一）:100

Description

Easiest Python Challenge!

URL http://plqbnxx54.bkt.clouddn.com/first.py

Base Score 100

Now Score 100

User solved 211

送分题，直接读源码，一步一步往下做即可。(notepad++就可以解base64)

题目源码

```
import base64
import hashlib

enc1 = 'hgame{'
enc2 = 'SGVyZV8xc18zYXN5Xw=='
enc3 = 'Pyth0n}'

print 'Welcome to Processor\'s Python Classroom!\n'
print 'Here is Problem One.'
print 'There\'re three parts of the flag.'
```

```
print '-----------------------------------------------'

print 'Plz input the first part:'
first = raw_input()
if first == enc1:
    pass
else:
    print 'Sorry , You\'re so vegatable!'
    exit()

print 'Plz input the secend part:'
secend = raw_input()
secend = base64.b64encode(secend)
if secend == enc2:
    pass
else:
    print 'Sorry , You\'re so vegatable!'
    exit()

print 'Plz input the third part:'
third = raw_input()
third = base64.b32decode(third)
if third == enc3:
    pass
else:
    print 'Sorry , You\'re so vegatable!'
    exit()

print 'Oh, You got it !'
```

flag:

```
hgame{Here_1s_3asy_Pyth0n}
```

pwn

aaaaaaaaaa :50

Description

pwn很简单的，a上去就完事了 nc 118.24.3.214 9999

URL http://plps4kyke.bkt.clouddn.com/aaaaaaaaaa

Base Score 50

Now Score 50

User solved 134

pwn我是什么都不懂，直接把东西丢到ida，然后F5 main函数，看代码，

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  signed int v3; // eax@4
  signed int v5; // [sp+Ch] [bp-4h]@1

  setbuf(_bss_start, 0LL);
  signal(14, handle);
  alarm(0xAu);
  puts("Welcome to PWN'world!let us aaaaaaaaaa!!!");
  v5 = 0;
  while ( 1 )
  {
    v3 = v5++;
    if ( v3 > 99 )
      break;
    if ( getchar() != 97 )
      exit(0);
  }
  system("/bin/sh");
  return 0;
}
```

大致意思应该是输入大于99个a即可getshell。

接下来就是cat flag

misc

Hidden Image in LSB :50

Description

Here are some magic codes which can hide information in an ordinary picture, can you extract the hidden image in the provided picture? 其实本来想让大家写写代码，后来干脆就送分了 有个神器叫 stegsolve，利用它可以直接提取本题 flag

URL http://plir4axuz.bkt.clouddn.com/hgame2019/lsb.zip

Base Score 50

Now Score 50

User solved 227

直接上神器stegsolve，在Red plane 1时可以看到：

flag:

```
hgame{LSB_is_easy_for_u}
```

打字机 :50

Description

Aris(划掉)牌打字机，时尚时尚最时尚~ hint:谷歌有个以图搜图功能很不错，百度识图好垃圾的。。。

URL http://plps4kyke.bkt.clouddn.com/打字机.zip

Base Score 50

Now Score 50

User solved 152

我是直接看的，直接对着键盘对应打出来就好，注意大小写。（没有尝试谷歌。。。）

```
hgame{My_vi0let_tyPewRiter}
```

Broken Chest :50

Description

这个箱子坏掉了！快用你无敌的[疯狂钻石]想想办法啊！ 更新一波学习资料https://ctf-wiki.github.io/ctf-wiki/misc/archive/zip/

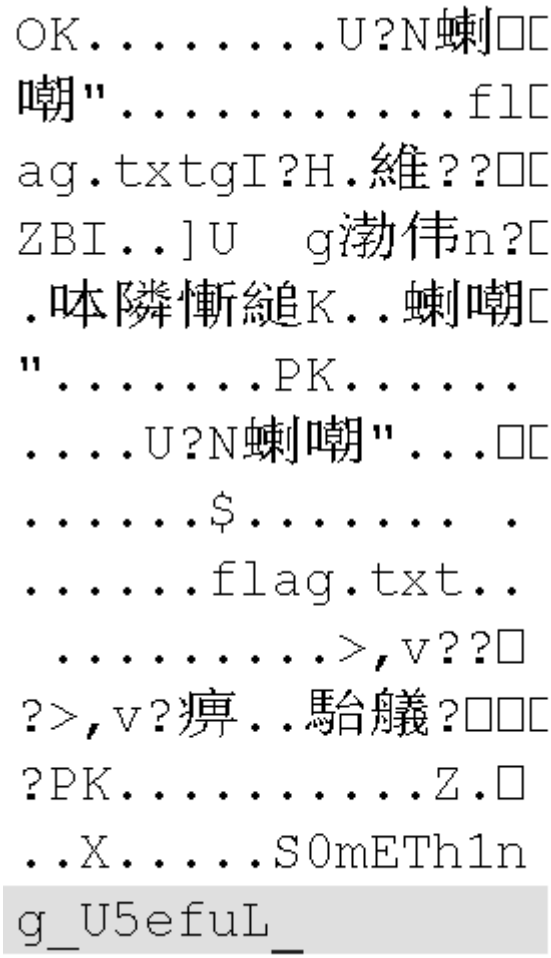
URL http://plqfgjy5a.bkt.clouddn.com/Broken-Chest.zip

Base Score 50

Now Score 50

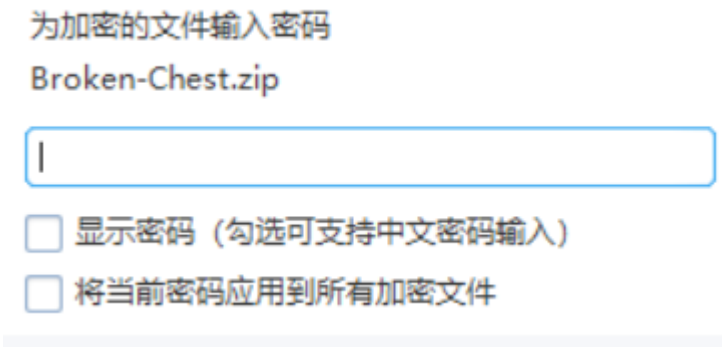
User solved 150

学习资料上面的东西就够了，我们用notepad++或者winhex之类看zip的文件头部和尾部，这里使用的是notepad++加上hex插件发现：

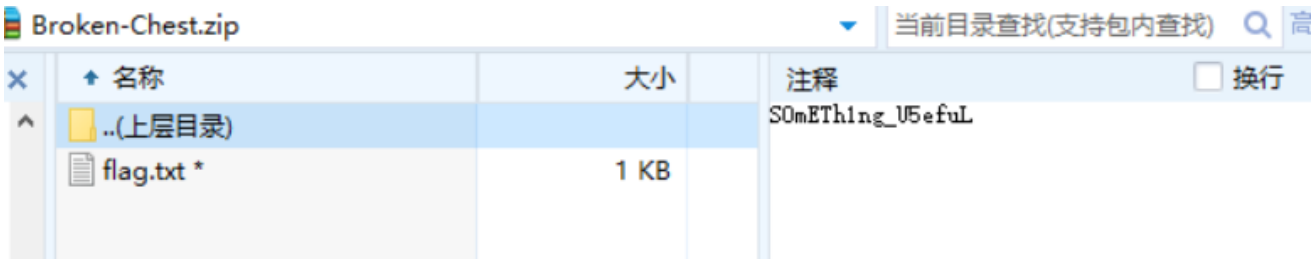


明显文件头错了，将“O"改为”P“,即可打开，尾部有一串奇怪的东西，先不管。

解压zip发现加密了：



用好压打开发现注释：



其实就是文件尾部那串东西，大概就是密码了。

操作一通，果然是，done!

flag在flag.txt里面。

flag：

```
hgame{Cra2y_D1aM0nd}
```

Try :100

Description

无字天书

URL http://plqfgjy5a.bkt.clouddn.com/try-it.pcapng

Base Score 100

Now Score 100

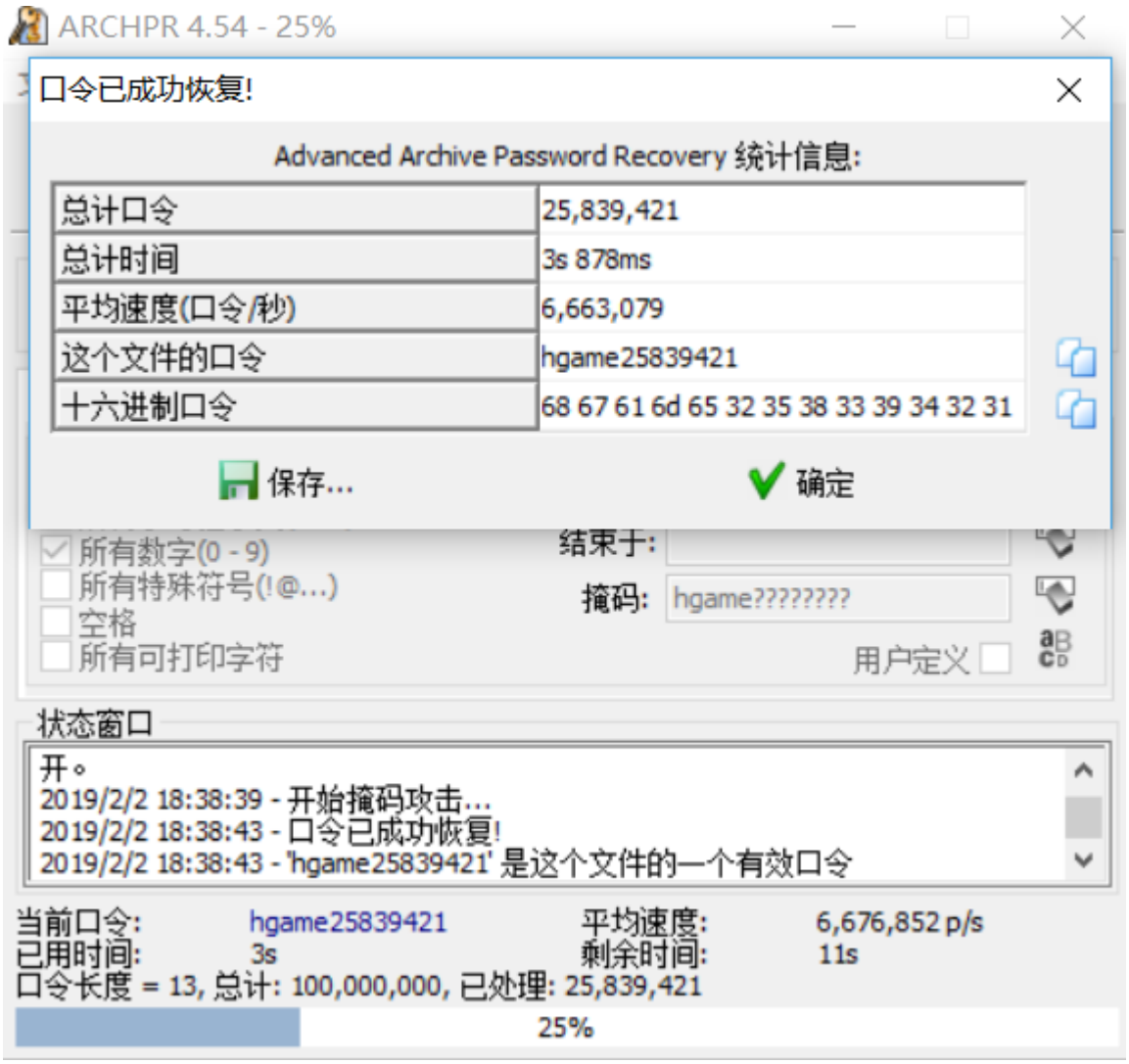
User solved 97

用wireshark打开，追踪tcp协议慢慢查看里面的请求过程，找到奇怪zip (其实可以直接使用过滤查找特殊字符如tcp contains "hgame"),找到目标，先将zip导出，然后解压，发现password.txt，open-it.zip.

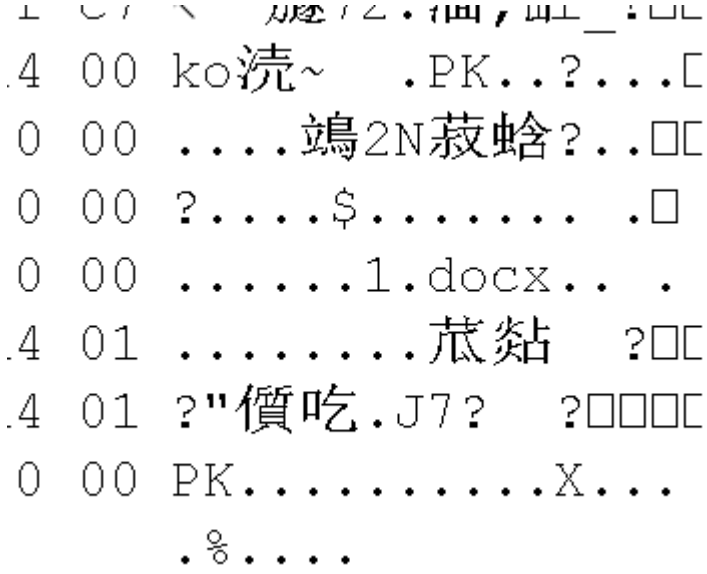
password.txt

```
hgame********
```

open-it.zip加密了，明显的掩码攻击，使用ARCHPR，可以百度下载，先尝试数字简单的东西，发现



解压zip，发现一个jpg文件，但是这个jpg文件有点大，用notepad++打开发现



发现里面有一个1.docx的文档，用binwalk分离，打开1.docx发现“什么都没有”，都是骗人的。。。

其实有经验的知道docx其实有一个隐藏功能，只要一个小勾就可以解决，具体百度“word显示隐藏文字”。

flag:

```
hgame{59d28413e36019861498e823f3f41406}
```

crypt

Mix :50

Description

--.../....-/....-/-.../--.../...--/...../..-./-..../-../-..../..-./--.../----./....-/....-/--.../.----/-..../-.../--.../-.../-..../..---/...../.----/-..../-..../...--/....-/...--/-----/-..../...../--.../-.. So easy

URL http://example.com

Base Score 50

Now Score 50

User solved 151

没什么好说的摩尔斯电码，随便找一个在线解密网站，解开为

```
tKs_moyDqk{bQf40e}
```

明显接下来是栅栏（每2个一组，总共9组）加变异凯撒（注意大小写，以及{}是不变的），加上开头一定是hgame{可以推测出偏移量。

栅栏解密后为

```
tsmyq{Q4eK_oDkbf0}
```

凯撒解密后为

```
hgame{E4sY_cRypt0}
```

perfect_secrecy :100

Description

Mom told me OTP is perfect secrecy!（结果加上hgame{}）

URL http://plqbnxx54.bkt.clouddn.com/easy_otp.py

Base Score 100

Now Score 100

User solved 42

题目代码

```
import binascii
import string
import random

def strxor(a, b):
    return "".join(hex(x ^ y)[2:].zfill(2) for (x, y) in zip(a, b))


fp = open('poem.txt', 'rb')
flag = "**********************************"
strings = fp.readlines()
key = hex(random.randint(2**511, 2**512))[2:]
strs = [strxor(i[:-3], binascii.unhexlify(key)) for i in strings]
result = strxor(flag.encode('utf-8'), binascii.unhexlify(key))
print(strs)
print(result)

'''
output:
['daaa4b4e8c996dc786889cd63bc4df4d1e7dc6f3f0b7a0b61ad48811f6f7c9bfabd7083c53ba54',
'c5a342468c8c7a88999a9dd623c0cc4b0f7c829acaf8f3ac13c78300b3b1c7a3ef8e193840bb',
'dda342458c897a8285df879e3285ce511e7c8d9afff9b7ff15de8a16b394c7bdab920e7946a05e9941d8308e',
'd9b05b4cd5ce7c8f938bd39e24d0df191d7694dfeaf8bfbb56e28900e1b8dff1bb985c2d5aa154',
'd9aa4b00c88b7fc79d99d38223c08d54146b88d3f0f0f38c03df8d52f0bfc1bda3d7133712a55e9948c32c8a',
'c4b60e46c9827cc79e9698936bd1c55c5b6e87c8f0febdb856fe8052e4bfc9a5efbe5c3f57ad4b9944de34',
'd9aa5700da817f94d29e81936bc4c1555b7b94d5f5f2bdff37df8252ffbecfb9bbd7152a12bc4fc00ad7229090',
'c4e24645cd9c28939a86d3982ac8c819086989d1fbf9f39e18d5c601fbb6dab4ef9e12795bbc549959d9229090',
'd9aa4b598c80698a97df879e2ec08d5b1e7f89c8fbb7beba56f0c619fdb2c4bdef8313795fa149dc0ad4228f',
'cce25d48d98a6c8280df909926c0de19143983c8befab6ff21d99f52e4b2daa5ef83143647e854d60ad5269c87',
'd9aa4b598c85668885df9d993f85e419107783cdbee3bbba1391b11afcf7c3bfaa805c2d5aad42995ede2cdd8297724
4',
'e1ad40478c82678995df809e2ac9c119323994cffbb7a7b713d4c626fcb888b5aa920c354be853d60ac5269199',
'c4ac0e53c98d7a8286df84936bc8c84d5b50889aedfebfba18d28352daf7cfa3a6920a3c',
'd9aa4f548c9a609ed297969739d18d5a146c8adebef1bcad11d49252c7bfd1f1bc87152b5bbc07dd4fd226948397',
'c4a40e698c9d6088879397d626c0c84d5b6d8edffbb792b902d49452ffbec6b6ef8e193840',
'c5ad5900df8667929e9bd3bf6bc2df5c1e6dc6cef6f2b6ff21d8921ab3a4c1bdaa991f3c12a949dd0ac5269c']
'c2967e7fc59d57899d8bac852ac3c866127fb9d7f1e5b68002d9871cccb8c6b2aa'
'''
```

资料链接：安全客

直接上脚本，链接里面有两种解法，以下分别实现（python3）

脚本1

```
import binascii
import string
import random
```

```
def str_to_hex(s):
    return ' '.join([hex(ord(c)).replace('0x', '') for c in s])

def hex_to_str(s):
    return ''.join([chr(i) for i in [int(b, 16) for b in s.split(' ')]])

def str_to_bin(s):
    return ' '.join([bin(ord(c)).replace('0b', '') for c in s])

def bin_to_str(s):
    return ''.join([chr(i) for i in [int(b, 2) for b in s.split(' ')]])
def strxor(a, b):
    return "".join(hex(x ^ y)[2:].zfill(2) for (x, y) in zip(a, b))

str=['daaa4b4e8c996dc786889cd63bc4df4d1e7dc6f3f0b7a0b61ad48811f6f7c9bfabd7083c53ba54',
'c5a342468c8c7a88999a9dd623c0cc4b0f7c829acaf8f3ac13c78300b3b1c7a3ef8e193840bb',
'dda342458c897a8285df879e3285ce511e7c8d9afff9b7ff15de8a16b394c7bdab920e7946a05e9941d8308e',
'd9b05b4cd5ce7c8f938bd39e24d0df191d7694dfeaf8bfbb56e28900e1b8dff1bb985c2d5aa154',
'd9aa4b00c88b7fc79d99d38223c08d54146b88d3f0f0f38c03df8d52f0bfc1bda3d7133712a55e9948c32c8a',
'c4b60e46c9827cc79e9698936bd1c55c5b6e87c8f0febdb856fe8052e4bfc9a5efbe5c3f57ad4b9944de34',
'd9aa5700da817f94d29e81936bc4c1555b7b94d5f5f2bdff37df8252ffbecfb9bbd7152a12bc4fc00ad7229090',
'c4e24645cd9c28939a86d3982ac8c819086989d1fbf9f39e18d5c601fbb6dab4ef9e12795bbc549959d9229090',
'd9aa4b598c80698a97df879e2ec08d5b1e7f89c8fbb7beba56f0c619fdb2c4bdef8313795fa149dc0ad4228f',
'cce25d48d98a6c8280df909926c0de19143983c8befab6ff21d99f52e4b2daa5ef83143647e854d60ad5269c87',
'd9aa4b598c85668885df9d993f85e419107783cdbee3bbba1391b11afcf7c3bfaa805c2d5aad42995ede2cdd8297724
4',
'e1ad40478c82678995df809e2ac9c119323994cffbb7a7b713d4c626fcb888b5aa920c354be853d60ac5269199',
'c4ac0e53c98d7a8286df84936bc8c84d5b50889aedfebfba18d28352daf7cfa3a6920a3c',
'd9aa4f548c9a609ed297969739d18d5a146c8adebef1bcad11d49252c7bfd1f1bc87152b5bbc07dd4fd226948397',
'c4a40e698c9d6088879397d626c0c84d5b6d8edffbb792b902d49452ffbec6b6ef8e193840',
'c5ad5900df8667929e9bd3bf6bc2df5c1e6dc6cef6f2b6ff21d8921ab3a4c1bdaa991f3c12a949dd0ac5269c']
flag='c2967e7fc59d57899d8bac852ac3c866127fb9d7f1e5b68002d9871cccb8c6b2aa'
text=''
for i in range(4,16):
    text+=strxor(binascii.unhexlify(flag), binascii.unhexlify(str[i]))
def bxor(a, b):     # xor two byte strings of different lengths
    if len(a) > len(b):
        return bytes([x ^ y for x, y in zip(a[:len(b)], b)])
    else:
        return bytes([x ^ y for x, y in zip(a, b[:len(a)])])


def hamming_distance(b1, b2):
    differing_bits = 0
    for byte in bxor(b1, b2):
        differing_bits += bin(byte).count("1")
    return differing_bits


def score(s):
    freq = {}
    freq[' '] = 700000000

    freq['e'] = 390395169
```

```
    freq['t'] = 282039486
    freq['a'] = 248362256
    freq['o'] = 235661502
    freq['i'] = 214822972
    freq['n'] = 214319386
    freq['s'] = 196844692
    freq['h'] = 193607737
    freq['r'] = 184990759
    freq['d'] = 134044565
    freq['l'] = 125951672
    freq['u'] = 88219598
    freq['c'] = 79962026
    freq['m'] = 79502870
    freq['f'] = 72967175
    freq['w'] = 69069021
    freq['g'] = 61549736
    freq['y'] = 59010696
    freq['p'] = 55746578
    freq['b'] = 47673928
    freq['v'] = 30476191
    freq['k'] = 22969448
    freq['x'] = 5574077
    freq['j'] = 4507165
    freq['q'] = 3649838
    freq['z'] = 2456495
    score = 0
    string=bytes.decode(s)
    for c in string.lower():
        if c in freq:
            score += freq[c]
    return score


def break_single_key_xor(b1):
    max_score = 0
    english_plaintext = 0
    key = 0

    for i in range(0,256):
        b2 = [i] * len(b1)
        try:
            plaintext = bxor(b1, b2)
            pscore = score(plaintext)
        except Exception:
            continue
        if pscore > max_score or not max_score:
            max_score = pscore
            english_plaintext = plaintext
            key = chr(i)
    return key
```

```
b = binascii.unhexlify(text)


normalized_distances = []


for KEYSIZE in range(2, 40):
    # 我们取其中前6段计算平局汉明距离
    b1 = b[: KEYSIZE]
    b2 = b[KEYSIZE: KEYSIZE * 2]
    b3 = b[KEYSIZE * 2: KEYSIZE * 3]
    b4 = b[KEYSIZE * 3: KEYSIZE * 4]
    b5 = b[KEYSIZE * 4: KEYSIZE * 5]
    b6 = b[KEYSIZE * 5: KEYSIZE * 6]
    b7 = b[KEYSIZE * 6: KEYSIZE * 7]

    normalized_distance = float(
        hamming_distance(b1, b2) +
        hamming_distance(b2, b3) +
        hamming_distance(b3, b4) +
        hamming_distance(b4, b5) +
        hamming_distance(b5, b6)
    ) / (KEYSIZE * 5)
    normalized_distances.append(
        (KEYSIZE, normalized_distance)
    )
normalized_distances = sorted(normalized_distances, key=lambda x: x[1])


for KEYSIZE, _ in normalized_distances[:5]:
    block_bytes = [[] for _ in range(KEYSIZE)]
    for i, byte in enumerate(b):
        block_bytes[i % KEYSIZE].append(byte)
    keys = ''

    for bbytes in block_bytes:
        keys += break_single_key_xor(bbytes)
    key = bytearray(keys * len(b), "utf-8")
    plaintext = bxor(b, key)
    print("keysize:", KEYSIZE)
    print("key is:", keys, "n")
    s = bytes.decode(plaintext)
    print(s)
```

脚本2

```
import binascii
import string
import random


def str_to_hex(s):
```

```
    return ' '.join([hex(ord(c)).replace('0x', '') for c in s])

def hex_to_str(s):
    return ''.join([chr(i) for i in [int(b, 16) for b in s.split(' ')]])

def str_to_bin(s):
    return ' '.join([bin(ord(c)).replace('0b', '') for c in s])

def bin_to_str(s):
    return ''.join([chr(i) for i in [int(b, 2) for b in s.split(' ')]])
def strxor(a, b):
    return "".join(hex(x ^ y)[2:].zfill(2) for (x, y) in zip(a, b))

str=['daaa4b4e8c996dc786889cd63bc4df4d1e7dc6f3f0b7a0b61ad48811f6f7c9bfabd7083c53ba54',
'c5a342468c8c7a88999a9dd623c0cc4b0f7c829acaf8f3ac13c78300b3b1c7a3ef8e193840bb',
'dda342458c897a8285df879e3285ce511e7c8d9afff9b7ff15de8a16b394c7bdab920e7946a05e9941d8308e',
'd9b05b4cd5ce7c8f938bd39e24d0df191d7694dfeaf8bfbb56e28900e1b8dff1bb985c2d5aa154',
'd9aa4b00c88b7fc79d99d38223c08d54146b88d3f0f0f38c03df8d52f0bfc1bda3d7133712a55e9948c32c8a',
'c4b60e46c9827cc79e9698936bd1c55c5b6e87c8f0febdb856fe8052e4bfc9a5efbe5c3f57ad4b9944de34',
'd9aa5700da817f94d29e81936bc4c1555b7b94d5f5f2bdff37df8252ffbecfb9bbd7152a12bc4fc00ad7229090',
'c4e24645cd9c28939a86d3982ac8c819086989d1fbf9f39e18d5c601fbb6dab4ef9e12795bbc549959d9229090',
'd9aa4b598c80698a97df879e2ec08d5b1e7f89c8fbb7beba56f0c619fdb2c4bdef8313795fa149dc0ad4228f',
'cce25d48d98a6c8280df909926c0de19143983c8befab6ff21d99f52e4b2daa5ef83143647e854d60ad5269c87',
'd9aa4b598c85668885df9d993f85e419107783cdbee3bbba1391b11afcf7c3bfaa805c2d5aad42995ede2cdd8297724
4',
'e1ad40478c82678995df809e2ac9c119323994cffbb7a7b713d4c626fcb888b5aa920c354be853d60ac5269199',
'c4ac0e53c98d7a8286df84936bc8c84d5b50889aedfebfba18d28352daf7cfa3a6920a3c',
'd9aa4f548c9a609ed297969739d18d5a146c8adebef1bcad11d49252c7bfd1f1bc87152b5bbc07dd4fd226948397',
'c4a40e698c9d6088879397d626c0c84d5b6d8edffbb792b902d49452ffbec6b6ef8e193840',
'c5ad5900df8667929e9bd3bf6bc2df5c1e6dc6cef6f2b6ff21d8921ab3a4c1bdaa991f3c12a949dd0ac5269c']
flag='c2967e7fc59d57899d8bac852ac3c866127fb9d7f1e5b68002d9871cccb8c6b2aa'
text=''
for i in range(4,16):
    text+=strxor(binascii.unhexlify(flag), binascii.unhexlify(str[i]))

def bxor(a, b):     # xor two byte strings of different lengths
    if len(a) > len(b):
        return bytes([x ^ y for x, y in zip(a[:len(b)], b)])
    else:
        return bytes([x ^ y for x, y in zip(a, b[:len(a)])])


def hamming_distance(b1, b2):
    differing_bits = 0
    for byte in bxor(b1, b2):
        differing_bits += bin(byte).count("1")
    return differing_bits


def break_single_key_xor(text):
    key = 0
    possible_space=0

    max_possible=0
```

```
    letters = string.ascii_letters.encode('ascii')
    for a in range(0, len(text)):
        maxpossible = 0
        for b in range(0, len(text)):
            if(a == b):
                continue
            c = text[a] ^ text[b]
            if c not in letters and c != 0:
                continue
            maxpossible += 1
        if maxpossible>max_possible:
            max_possible=maxpossible
            possible_space=a
    key = text[possible_space]^ 0x20
    return chr(key)



b = binascii.unhexlify(text)


normalized_distances = []


for KEYSIZE in range(2, 40):
    #我们取其中前6段计算平局汉明距离
    b1 = b[: KEYSIZE]
    b2 = b[KEYSIZE: KEYSIZE * 2]
    b3 = b[KEYSIZE * 2: KEYSIZE * 3]
    b4 = b[KEYSIZE * 3: KEYSIZE * 4]
    b5 = b[KEYSIZE * 4: KEYSIZE * 5]
    b6 = b[KEYSIZE * 5: KEYSIZE * 6]

    normalized_distance = float(
        hamming_distance(b1, b2) +
        hamming_distance(b2, b3) +
        hamming_distance(b3, b4) +
        hamming_distance(b4, b5) +
        hamming_distance(b5, b6)
    ) / (KEYSIZE * 5)
    normalized_distances.append(
        (KEYSIZE, normalized_distance)
    )
normalized_distances = sorted(normalized_distances,key=lambda x:x[1])


for KEYSIZE,_ in normalized_distances[:5]:
    block_bytes = [[] for _ in range(KEYSIZE)]
    for i, byte in enumerate(b):
        block_bytes[i % KEYSIZE].append(byte)
    keys = ''
    try:

        for bbytes in block_bytes:
```

```
            keys += break_single_key_xor(bbytes)
        key = bytearray(keys * len(b), "utf-8")
        plaintext = bxor(b, key)
        print("keysize:", KEYSIZE)
        print("key is:", keys, "n")
        s = bytes.decode(plaintext)
        print(s)
    except Exception:
        continue
```

注：由于题目给的一回合方式将第一位给去掉了，所以第一位需要去猜，还有keysize=33是确定的，可以不需要明文间距那些函数，直接设置

flag：

```
hgame{OTP_is_not_safe_if_more_than_once}
```

Base全家 :50

Description

全家老小

URL http://plir4axuz.bkt.clouddn.com/hgame2019/enc.txt

Base Score 50

Now Score 50

User solved 89

这个题目需要判断base加密方式，主要考察base64，base32,base16 (这个其实就是将16进制转ASCII)，我这里有个代码示范。至于如何区分，详情百度“base系列”。。。

base16

```
import base64

with open('base32de2.txt', 'r') as f:
    str = f.read()
    flag = base64.b16decode(str)
    f.close()
    with open('base16de3.txt', 'w') as f1:
        f1.write(flag)
```

base32

```
import base64

with open('base32de2.txt', 'r') as f:
    str = f.read()
    flag = base64.b32decode(str)
    f.close()
    with open('base32de3.txt', 'w') as f1:
        f1.write(flag)
```

base64

```
import base64

with open('base32de2.txt', 'r') as f:
    str = f.read()
    flag = base64.b16decode(str)
    f.close()
    with open('base16de3.txt', 'w') as f1:
        f1.write(flag)
```

最终会解密为：

```
base58：2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8BnC
```

直接去base58在线解密网站，我没找到。。。

都是找到了一个脚本，稍微改了一下，（python2）

```
__b58chars = '123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz'
__b58base = len(__b58chars)


def b58encode(v):
    """ encode v, which is a string of bytes, to base58.
    """

    long_value = int(v.encode("hex_codec"), 16)

    result = ''
    while long_value >= __b58base:
        div, mod = divmod(long_value, __b58base)
        result = __b58chars[mod] + result
        long_value = div
    result = __b58chars[long_value] + result

    # Bitcoin does a little leading-zero-compression:
    # leading 0-bytes in the input become leading-1s
    nPad = 0
    for c in v:
        if c == '\0':
```

```
            nPad += 1
        else:
            break

    return (__b58chars[0] * nPad) + result


def b58decode(v):
    """ decode v into a string of len bytes
    """

    long_value = 0
    for (i, c) in enumerate(v[::-1]):
        long_value += __b58chars.find(c) * (__b58base ** i)

    result = ''
    while long_value >= 256:
        div, mod = divmod(long_value, 256)
        result = chr(mod) + result
        long_value = div
    result = chr(long_value) + result

    nPad = 0
    for c in v:
        if c == __b58chars[0]:
            nPad += 1
        else:
            break

    result = chr(0) * nPad + result
    return result

if __name__ == "__main__":
    print  b58decode("2BAja2VqXoHi9Lo5kfQZBPjq1EmZHGEudM5JyDPREPmS3CxrpB8BnC")
```

最终结果为

```
hgame{40ca78cde14458da697066eb4cc7daf6}
```